

University of Portsmouth
School of Creative Technologies

**Conducting the performance:
Setting up a motion capture real-time
performance**

By
Group 3

May 2009

Contents

CONTENTS	2
CHAPTER 1: INTRODUCTION	3
CHAPTER 2: PIPELINE CONSIDERATIONS	6
CHAPTER 3: VICON IQ.....	7
SECTION 3.1: SETTING UP	8
<i>Sub - Section 3.1.1: Volume set up.....</i>	<i>9</i>
SECTION 3.2: ACTORS AND PROPS.....	13
<i>Sub - Section 3.2.1: Actor makers.....</i>	<i>13</i>
<i>Sub - Section 3.2.2: Prop markers</i>	<i>15</i>
SECTION 3.3: ACTOR CALIBRATION	15
<i>Sub - Section 3.3.1: Range of Motion test.....</i>	<i>15</i>
<i>Sub - Section 3.3.2: Calibrating the Actor</i>	<i>16</i>
CHAPTER 4: 3DS MAX.....	18
SECTION 4.1: CHARACTER MESHES	18
SECTION 4.2: RIGGING THE MESHES	20
SECTION 4.3: UNWRAPPING THE CHARACTERS.....	22
SECTION 4.4: EXPORTING THE CHARACTERS.....	23
CHAPTER 5: MOTION BUILDER.....	24
SECTION 5.1: SETTING UP THE ACTOR	24
SECTION 5.2: IMPORTING A CHARACTER FROM MAX	25
<i>Sub - Section 5.2.1: Importing the model and rig</i>	<i>25</i>
<i>Sub - Section 5.2.2: Characterization and Re-Targeting.....</i>	<i>25</i>
SECTION 5.3: USER INTERACTIONS	26
<i>Sub - Section 5.3.1: Controlling the cameras</i>	<i>26</i>
<i>Sub - Section 5.3.2: Stage effects.....</i>	<i>27</i>
<i>Sub - Section 5.3.3: Light Show.....</i>	<i>27</i>
SECTION 5.4: THE REAL-TIME PLUG-IN.....	27
REFERENCES.....	29
APPENDIX A – THE G-53 MARKER SET.....	30

Chapter 1: Introduction

Motion capture is the recording of markers within a globally defined space. These markers are recorded using special 2d cameras fitted with ring of red LEDs attached to the camera.

The markers are made of highly reflective scotch tape. Optical motion capture works by the reflective markers reflect the red LED light back to the camera. The motion capture system used by the University of Portsmouth is a Vicon set up with 8 inferred V-Cams which record black and white at a resolution of 640 x 480 at 119.9 frames per second.

These 2d camera feeds are feed into a Vicon Data Station which will also triangulated the white dots to create a point marker in 3d space. The Data Station connections to a program called TARSUS to allow the labeling of point data to the Data Station and to send out the marker coordinates and labels when required for 3rd party programs such as Autodesk 3ds Max and Autodesk Motion Builder.

The Vicon optical motion capture system is controlled and setup by a program called Vicon IQ. Vicon IQ connects to the TARSUS program to send data to the Data Station such as marker labels and Vicon Skeleton Kinematic Model (VSK) data.

TARSUS sends out marker labels and marker position data and this data will become the driving force of digital actor which will be retargeted to the final 3d mesh inside of Autodesk's Motion Builder.

Fig 1.1 shows the complex Vicon Motion Capture network setup and how it was used to spread the processing of data to produce a faster labeling solution to help label the data in near real-time.

Conducting the performance: Setting up a motion capture real-time performance

“The Camera records the data, which is then sent to the data station (connection shown in red).

The data station then will triangulate the data, and send it to the TARUS (connection shown in blue).

This data is sent from TARSUS to Vicon IQ Control software to be displayed and configured in real-time (connection shown in green). The user sets up their marker labels here, and it is sent back to TARSUS (connection shown in yellow).

These marker labels are sent back to the data station to be labeled in real-time (connection shown in pink). And the labeled real-time data is sent back to TARSUS to be used by 3rd party plug-ins and Vicon IQ (connection shown in blue).

The Real-Time machine, used to display models driven by motion data uses Autodesk’s Motion Builder to connect to TARSUS (connection shown in orange) to collect the number of markers, their names and their positions in space.

This is solved in real-time and show on a projector using a long VGA cable (connection shown in turquoise).

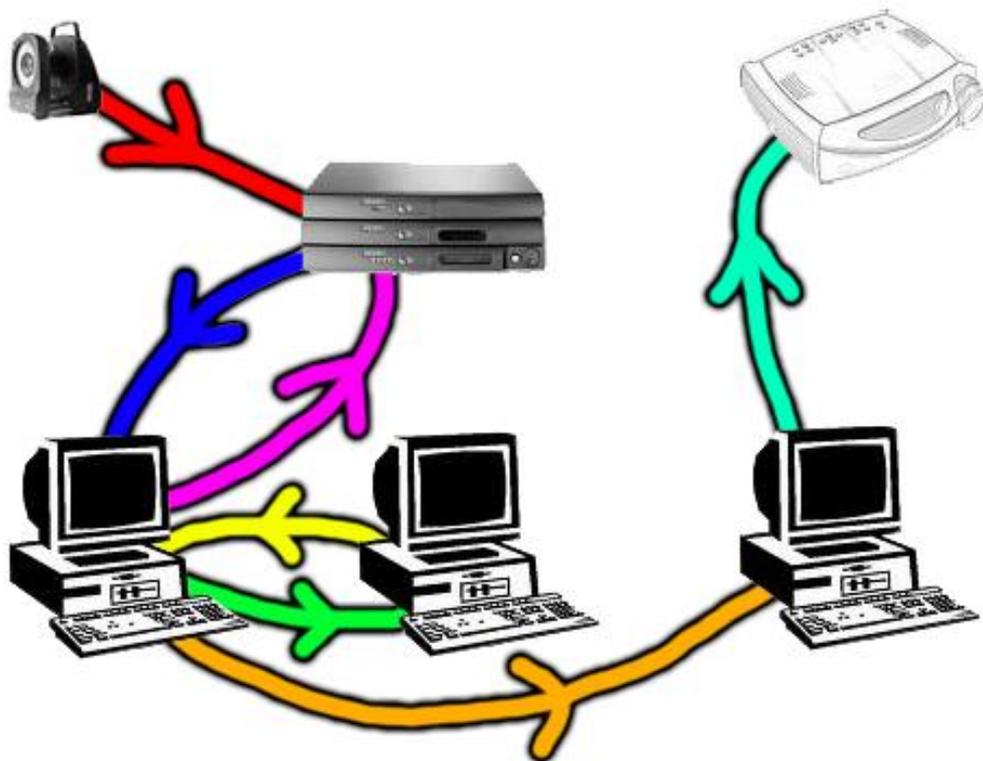


Fig1.1 The Vicon Network setup” (RTGROP Group 3, 2009)

Conducting the performance: Setting up a motion capture real-time performance

In this book we will walk you through all the steps required to set up a virtual real-time performance using a Vicon V-cam system and running the final real-time performance inside of Autodesk's Motion Builder.

With the basics of motion capture described, we will next look at the pipeline needed to ensure all models; rigs and textures will work inside the final engine.

Chapter 2: Pipeline Considerations

As the process of motion capture was described in the last chapter, we need to establish the pipeline for creating the models and assets and bring them into the final program.

We established that the creative content creation program we were going to use was Autodesk 3ds Max due to the improved compatibility between 3ds Max and Motion Builder in the form of the FBX export/ import updates.

An important point to mention when working with a group of artist is to ensure that all artists are working on the same scale and all models are in proportion to one another. This was done in our pipeline by building a simple basic 3d stage which would be used to measure all the rest of the characters and assets.

A know bug for converting character meshes between 3ds Max and Motion Builder is when the 3d mesh is skinned using the “physique” modifier, as this data isn’t read in Motion Builder, but the skinning data from the “skin” modifier is read and used. With this in mind, all character mesh data must be exported using the “skin” modifier to allow the mesh to be deformed in real-time in Motion Builder.

Using 3ds Max to produce the creative content means that the most idea file format to use to get the data into Motion Builder is the FBX file format, which Autodesk has adopted as their format of choice, greatly increasing the compatibility between 3ds Max, Maya and Motion Builder.

The program used to setup and configure the real-time motion data is Vicon IQ, and is the process of setting up is explained next.

Chapter 3: Vicon IQ

Vicon IQ is the program used to set up, label and clean motion data when using a Vicon Motion Capture system.

Vicon IQ has its own set pipeline users must follow to be able to use the motion data. The first thing is to set up a new shoot data within the in-built content management database called “eclipse”. The Icons at the top of the screen are used to help keep your data organized. Fig 3.1 shows a cut down eclipse interface with the buttons and the relationship between each of the different sections and stubs.

The data is split up into; Projects; Capture Days; Sessions; and Sub-Sessions. This way all relevant data is kept together such as configuration files, calibration files and calibrated subject files.

Vicon IQ requires a Session to be selected before any data can be saved, so the data management side is compulsory.

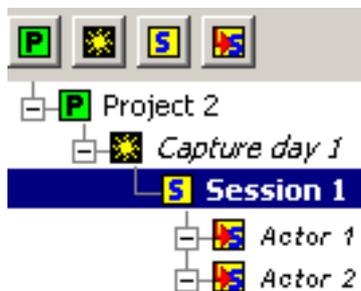


Fig 3.1 shows how the data is structured and stored



Fig 3.2 is the New Project button from eclipse



Fig 3.3 is the New Capture Day button from eclipse



Fig 3.4 is the New Session button from eclipse



Fig 3.5 is the New Sub-Session button from eclipse

Using the buttons at the top of eclipse will allow for new node to be created, and double clicking on a particular node will turn it blue and set it as the active session or sub session like in fig 3.1.

Section 3.1: Setting up

With a session or sub session selected, we can move from the data management tab to the setup tab. The only setting that should be changed in this tab is the IP address of TARSUS computer. For the set up our motion capture studio is using, the IP address should be 192.168.0.1.

The next tab along the top is the Calibrate tab, this is where the capture volume is defined, and the world space axis is set.

To do this we must connect to TARSUS, Fig 3.1.1 shows the button to connect to start the connection to the TARSUS machine. Without doing this, no real-time information will be displayed or calculated.

Fig 3.1.3 shows a fully calibrated and connected calibration screen in Vicon IQ. Please bear in mind that a calibration will be automatically copied over from the last calibration used, this could be greatly wrong as it will not be updated, and so could be misleading.



Fig 3.1.1 Connect to TARSUS button



Fig 3.1.2 Close Connection to TARSUS button

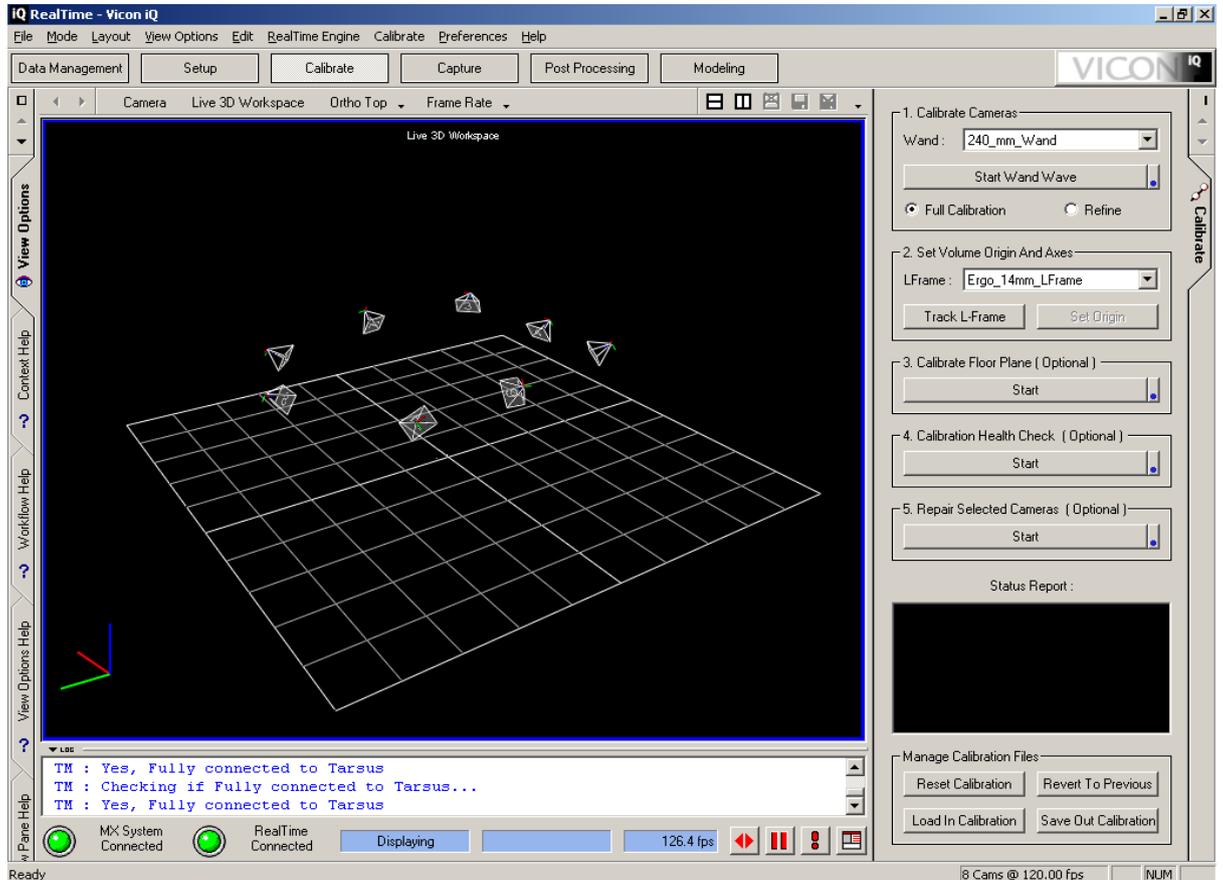


Fig 3.1.3 Calibration Tab and connected to TARSUS

To setup the capture volume we must now define the area using the calibration “240mm wand” and then use the “Ergo 14mm LFrame” to set the world space origin and axis.

Sub - Section 3.1.1: Volume set up

With the cameras connected and in the calibration tab, the next bit is to define the capture volume which is going to be used by the actors by performing a wand wave.

A wand wave is where a wand is waved around the capture volume. The wand used by our studio is the same as Fig 3.1.1.1, it is a 240mm wand and

Conducting the performance: Setting up a motion capture real-time performance

this is important as the distance between the markers is set as such, the computers in the cameras can quickly work out which marker is which.



Fig 3.1.1.1 A Vicon 240mm Calibration Wand.

This process is very important as it sets up each camera's position relative to each other, so the more frames that can be captured during the wand wave, the more accurate the calibration will be.

Vicon IQ requires a minimum of 500 frames to be recorded per camera, but with 3 years of motion capture experiences, I would suggest at least 1500 per camera, as it provides a much more accurate calibration. Accuracy in the calibration is needed to reduce marker noise during real-time, that is the ambient movement of a marker due to inaccuracies in the triangulation of the data.

When performing a wand wave, it is best to look at the camera view to see how the 3D capture volume is being described to the Vicon system. Ideally you will want to completely fill in the camera, but only within the capture

Conducting the performance: Setting up a motion capture real-time performance

volume. Fig 3.1.1.2 shows the wand wave from the camera view, but was taken with use of a Vicon MX system, but there should not be any major changes. Note the frame count on the bottom right of the picture.

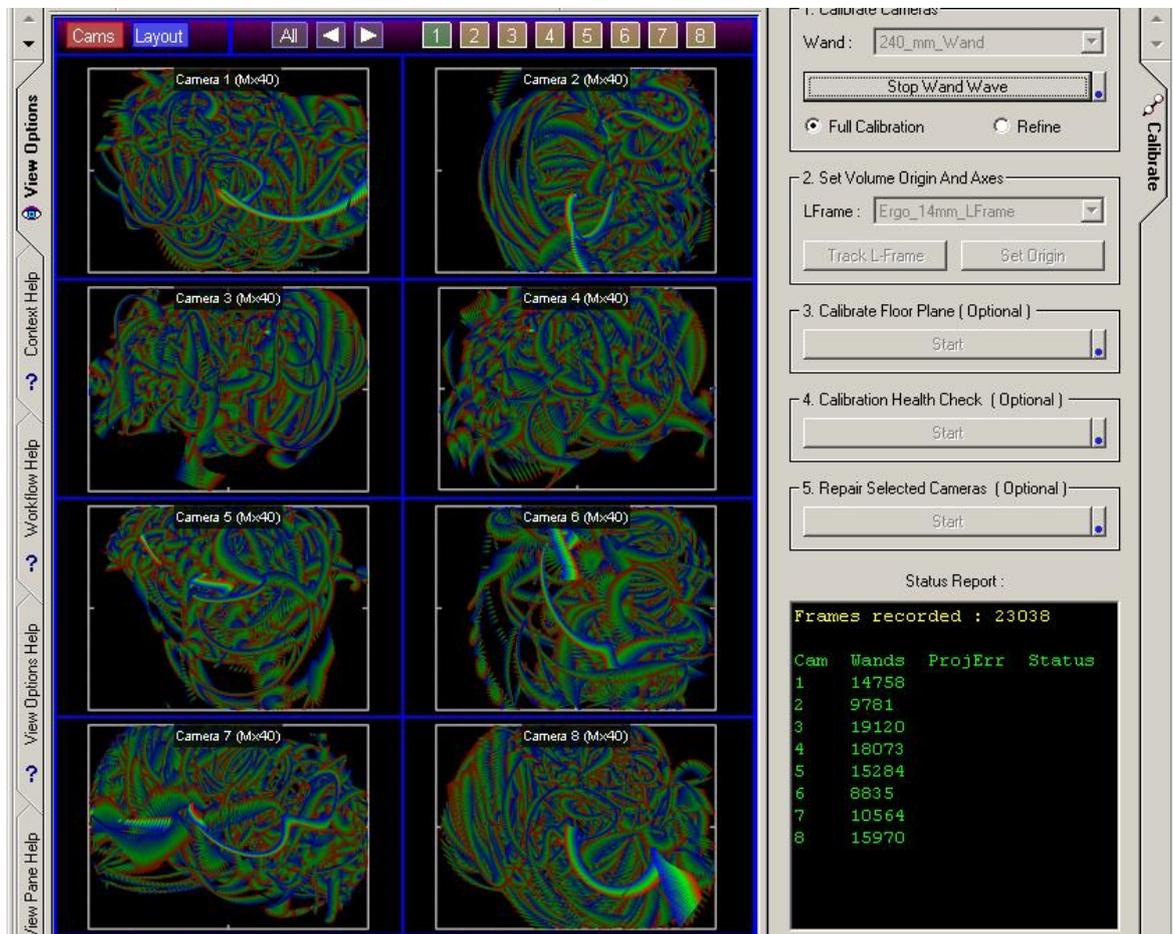


Fig 3.1.1.2 A Wand wave from the view of the cameras.

With the cameras set up, all the cameras will be in the right place based of the world space of Camera one, so many cameras will appear to be in the air when viewing using the 3D view. This is due to the world origin and axis not being set. To do this we use the Ergo 14mm LFrame, Fig 3.1.1.3.



Fig 3.1.1.3 The Ergo 14mm LFrame

The LFrame has two knobs to adjust the height to the LFrame as well as two spirit levels to help you ensure the LFrame is perfectly level when you set it. If the LFrame is not level, you may find your real-time character walking up or down a slope. The 3 markers in a row represent the X Axis and the 2 markers in a row represent the Z Axis. For easy use in Motion Builder the actor should be facing the positive Z direction, so to fit in with the way the Actor is created in Motion Builder. In our studio, the X Axis is parallel to the desk of computers with the Z axis facing towards the computers. The Hypotenuse of the LFrame should be facing towards the Vicon Data Station.

When this is set, the Track L Frame button should be pressed, and the Set Origin button 5 seconds after. This is to counteract a known bug in the software where the origin isn't set for some unknown reason.

With the calibration complete, the calibration set up should be saved out, but will be automatically saved after your first capture.

Section 3.2: Actors and Props

Sub - Section 3.2.1: Actor makers

To capture an actor, we have to use a specially designed motion capture suit and attach reflective markers to record the motions of joints and bones.

The suits we use at the studio are a newer version of the ones used in the production of Polar Expresses by Sony. Fig 3.2.1.1 shows Tom Hanks wearing an old motion capture suit of the same type of the ones we use.



Fig 3.2.1.1 Tom Hanks from the Polar Express

The advantages these suits have are that they are covered in Velcro for the majority of the body, but have a breathable blue material for some areas where markers should not be placed. This helps the actor wearing the suit keep a more stable body temperature.

For a single human actor, we have a tried and tested marker set that we use. A marker set is a list of marker names and general positions used to help reconstruct the data, but not too many that it compromises the data.

Conducting the performance: Setting up a motion capture real-time performance

Our custom defined marker set for the studio is called the G53 marker set, which means it's a general use marker set with 53 markers. The placement of the g53 can be found in appendix A.

The markers used are 19mm reflective markers with Velcro bases allowing them to be stuck straight onto the actor's motion capture suit without having to worry about any additional adhesive and allow for the markers to be changed and tweaked with relative ease.

The shoes are provided by the studio and already have Velcro strips glued to them so markers can quickly be attached to the shoes.

When placing markers, every marker must be a minimum of knuckles distance away from any other marker; this is so the Vicon system does not read the two markers as one.

Makers should be placed on bones when possible, as bones do not move whereas muscle over the bone does.

When recording more than one actor, as we did, it is important the marker number 27 ("Pelvis") is placed in a different position as the Pelvis marker is used to identity different actors being recorded.

Fig 3.2.1.2 shows the marker placement for the G53 Marker set.

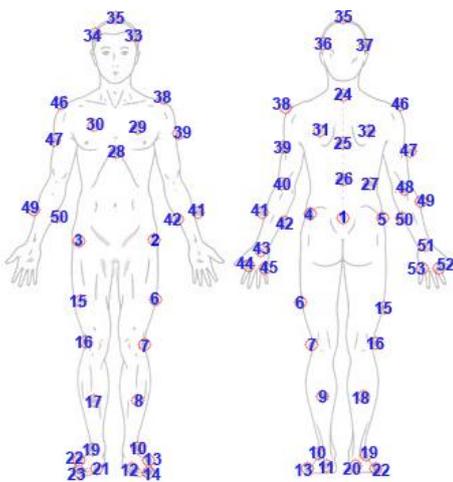


Fig 3.2.1.2 The G53 Marker set

Sub - Section 3.2.2: Prop markers

Props are easier to record than actors due to the rigid nature of most props. All props must have at least 3 markers on to be used as a rigid body in either 3ds Max or Motion Builder, and I suggest having an additional one or two markers on the prop in case of marker occlusion.

The markers used to define a prop are the same 19mm markers used on the subject, and should be attached using Velcro pads, but there are smaller markers with a hard plastic base, suitable for being taped or glued to the prop.

The Ergo 14mm LFrame is a good example of marker placement on a prop, as a rigid body because the markers do not move independently as well not all markers being placed in a straight line.

Unlike an actor, a prop does not require a range of motion test or a calibration, instead the prop is defined inside of the capture tab. Inside of the capture tab, there is the Create/Edit Objects side tab on the right hand side of Vicon IQ. Within that tab there is a button called "Create Object", and hitting this button will cause the Real-Time to be paused allowing you to select the markers you want to use to make up the object. Once all the relevant objects are selected, typing in the prop name at the bottom of the screen will set the name of the prop and hitting the Finish button at the top will resume the Real-Time feed.

It is THAT easy to define a prop!

Section 3.3: Actor Calibration

Sub - Section 3.3.1: Range of Motion test

Unlike the props, Actors need to be calibrated in order for Vicon IQ to label the marker data in real-time or post production.

For the actor to be calibrated, they must perform what is known as a range of motion test (ROM), where the actor is recorded doing a range of different motions that would, typically, be applicable in the motions being recorded. For our performance, we recorded a standard range of motions which include:

Conducting the performance: Setting up a motion capture real-time performance

- Rotating of the head
- Swinging of the arms, both forward and back
- Rotating of the torso
- Leaning back
- Leaning forward and touch toes
- Legs kicking out and rotated around
- Lunge forward
- A step forward
- A step backwards

These motions are likely to appear during the performance and as they will have been calibrated, the computer should find it easier to label.

To capture a ROM, Vicon IQ should be set on the capture tab, and the side tab should be set on Capture. Under group box 4 there is a Start button, clicking on this will start the recording and the same button will now read Stop. Clicking on the Stop button will stop the recording.

Once the ROM test has been captured, clicking on the Load into Post button will open up the file in the Post Processing tab, as the file will have to be reconstructed and cleaned before it can be calibrated.

Sub - Section 3.3.2: Calibrating the Actor

To use the recorded data, the 2D data must be reconstructed and trajectory fitted. What this means is to analyze the 2D position of each marker and use the other cameras to build up a 3D marker position at 119.9 frames per second. To do this we run two operations in the pipeline, the top two operations, the reconstruct trajectory fit frame range and the trajectory fit frame range. These two operations will reconstruct the 3D data which then needs to be labeled.

With the 3D data reconstructed, the marker data needs to be labeled in order to be calibrated. To do this we need a Vicon Skeleton Template (VST), one set up to match the marker set used to place the markers on the actor. For us we used the G53 VST to load in the appropriate marker labels.

Conducting the performance: Setting up a motion capture real-time performance

The side tab called subject contains all the necessary tools and buttons to do this! Under the Subject tab, there is a button called import VST, and by clicking it and selecting the G53, all the relevant marker labels will be loaded in. Now it's just a task of going through the data and labeling all the markers up.

Care should be taken to label the markers only in one time direction as labeling marker data for the whole range can cause data to be relabeled.

Using operations such as Fill gaps with spline fill and Fill gaps with rigid bodies, the ROM test should be relatively easy and quick to clean. Use the Continuity chart view to quickly find gaps in the labeled marker data.

Once the ROM test has been fully labeled and "cleaned", you should go back to the Subject tab on the right hand side, and at the bottom hit the button with a G on it and the button just below it with a T on it. What this does is it sets a General T-Pose so the computer knows this is the generic ambient posture of the actor. Hitting the Calibrate Subject button will calibrate the ROM.

When the Subject is fully calibrated, under the Capture top tab, there is a side tab called Active Subjects. Opening up that tab, you should find your actor and a check box, checking that box will enable real-time labeling for your actor.

Chapter 4: 3ds Max

Section 4.1: Character meshes

For the virtual performance, you will need four character models. One drummer, one singer and two guitarists. Two of the characters will use pre recorded motion capture data for their animations and the other two will be animated in real-time. It is up to you what characters you wish to have real-time and pre-recorded, we choose to pre-record the drummer and guitarist leaving the singer and basest being real-time animated.

The characters for the real time performance should be around 2000 polygons in each. The low polygon count is necessary to enable 25 frames per second during the performance.

Characters can be modelled by any method you wish. Our characters were modelled using 'box modelling', a process where characters start as basic primitives and are then sculpted and detailed by adding edge loops. It is best to model your characters in the T-pose as this makes rigging them latter on much easier.

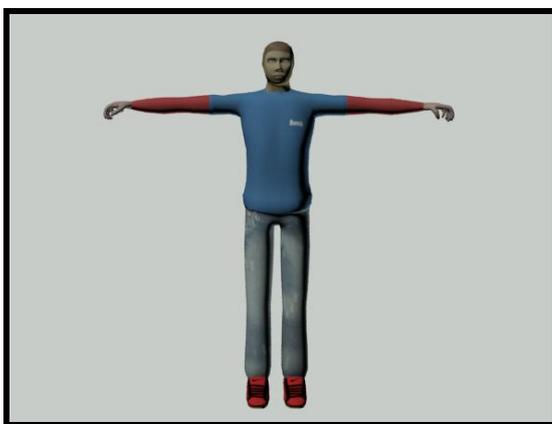


Fig 4.1.1 T-Pose

Conducting the performance: Setting up a motion capture real-time performance

While modelling the characters, it is extremely important to keep the mesh comprised of quads (four sided polygons). Non-quads can produce animation errors and cause a mesh to act badly when animated. If a non-quad cannot be avoided, try and position it on part of the mesh that won't have much animation deforming it, such as the top and back of the skull.

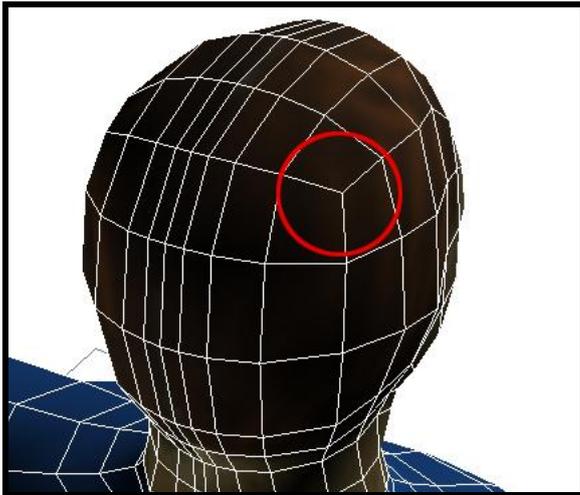


Fig 4.1.2 A pole on the back of the head.

Poles should also be avoided where possible. A pole is a vertex that has a number of edges connected to it that does not equal four. Like non-quads, a pole will cause a mesh to deform badly when animated and should therefore try to be positioned on part of the mesh that has little or no animation (skull, base of feet).

Section 4.2: Rigging the meshes

When working within Real Time projects and using Autodesk Motion Builder it is important to make sure all models are rigged using the modifier skin. While most people would use physique, there are compatibility issues regarding it when imported into Motion Builder. Skin however, does not have this same issue.

A skeletal deformation tool that lets you deform one object with another, Skin focuses on the distribution of weight between the bones, and thus how each bone affects the surrounding vertices. Each vertex is given a starting weight of 1.0, and then by using the weight table you can determine which bones affect which vertices and change the strength of each bone.

When the skin modifier is first applied to the mesh and bones are assigned, each bone is given a capsule-shaped envelope. To assign all the bones, click add on the bones rollout on the right hand side of the screen. You then need to select all the bones you wish to include from the newly appeared window. It is important not to choose Bip1 however, as this can affect the skinning process.

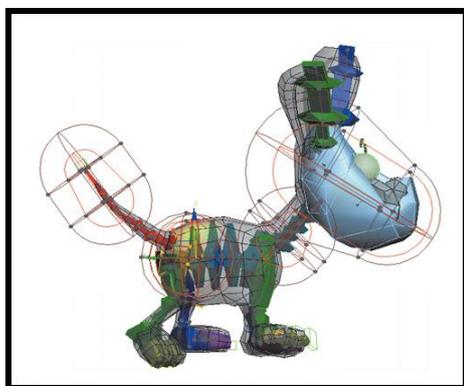


Fig 4.2.1 Envelopes

Conducting the performance: Setting up a motion capture real-time performance

All of the vertices that are in each envelope move with the bone. The size of the envelopes can be determined by using the select and move tool to make them bigger or smaller. The size and position of the bone can also be changed in the same manner. If two envelopes overlap, all affected vertices will move through a blend of the two.

By deselecting the modifier and selecting one of the bones (the hand for example) you can see the result of the Skin. If when moved stray vertices move as well, you can increase the size of the envelopes. If this still proves unsuccessful, you can select the vertices option and select each renegade vertex individually. By changing its weights, you can assign the vertex to the bone that should affect it.



Fig 4.2.2 Standard Biped

Section 4.3: Unwrapping the Characters

Once the models were rigged, the textures were applied to the mesh. This is done by unwrapping the character mesh so that it is laid flat out; this makes it easier to apply the textures. This is broken down into several stages:

- 1) After selecting the mesh the Unwrap UVW modifier was applied to the object. After doing this the Unwrap UVW modifier can be seen in the objects modifier stack.
- 2) In the sub object list there are three options including Vertex, Edge and Face. For the characters 'Face' was selected.
- 3) Then under the map parameters rollout each face was defined to the appropriate mapping type (planar, pelt, cylindrical, spherical). The different types of mapping are how the texture map is applied to the mesh. In the case of these models; cylindrical was selected. This cylindrical shape was then fitted around whatever section of the mesh that is being unwrapped using the scale tool.
- 4) 'Edit' was selected under the parameters rollout. This is because the UVW's of the unfolded mesh then need to be moved and scaled. This is done in the edit UVW window. This is a complex and time consuming process as sections of the characters mesh will be out of place and will need to be separated and fitted together so that texture can be applied to the mesh as easily as possible (without having to alter the shape of the texture very much).
- 5) Once the mesh unfolded mesh had been setup in the correct position on the edit UVW window, the UV template was then rendered to an appropriate size and saved.
- 6) The textures which had been created were then moved and scale in order to be applied to the template.
- 7) The bitmap of the texture was then found in the material editor and applied to the mesh.

Section 4.4: Exporting the Characters

After the characters were completed in 3ds studio max, they then needed to be exported so that they could be used in motion builder. When exporting; the model needs to be saved in '.fbx'. After selecting to save and export the model in this format a list of options are available to select. This is a list of options showing what to include when exporting the model. It is import to make sure that anything involved with the model in max are selected for instance in this case, the 'Skin' checkbox needed to ticked as all the character models had the skin modifier. When selecting the options the checkbox 'embed textures in export file' will not be selected, this needs to be ticked as the textures will need to be included.

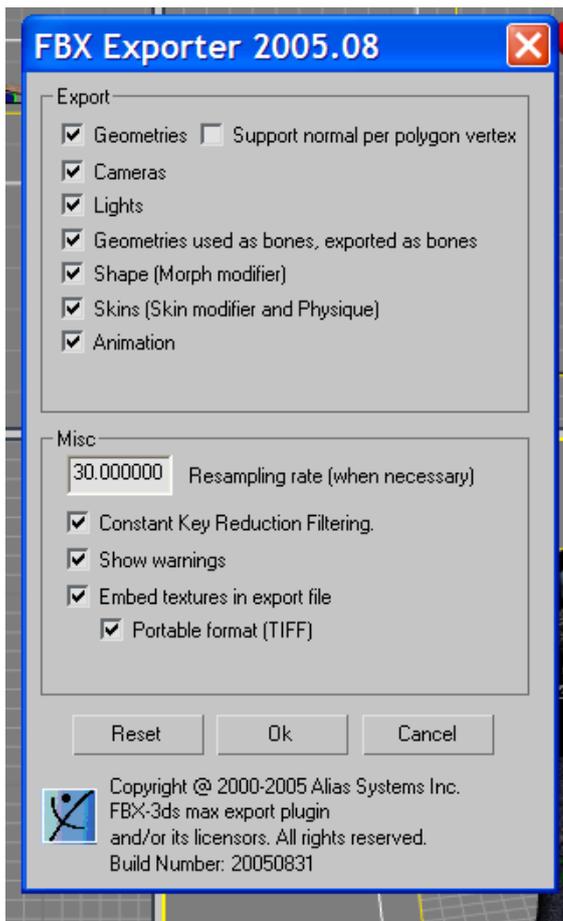


Fig 4.4.1 The FBX Exporter Settings

Chapter 5: Motion Builder

Section 5.1: Setting up the Actor

To set up the actor for real-time retargeting, we must import the ROM C3D file into Motion Builder. With the file imported we need to create an actor to map the data onto. By dragging an actor from the asset browser into the 3D view space will create an actor, which should be lined up with your data.

The actor must be translated (moved), Rotated and transformed (scaled) to fit the marker set up. The idea is to for the actor to match the human actor rather than using the markers to represent joints. So with this in mind the actor limbs should be placed to the human actors' limbs relative to the markers, so the markers should sitting onto of the 3d actor.

Opening the actor's settings in the Navigator panel will allow you to assign markers to varies parts of the body. Fig 5.1.1 shows the user interface for this.

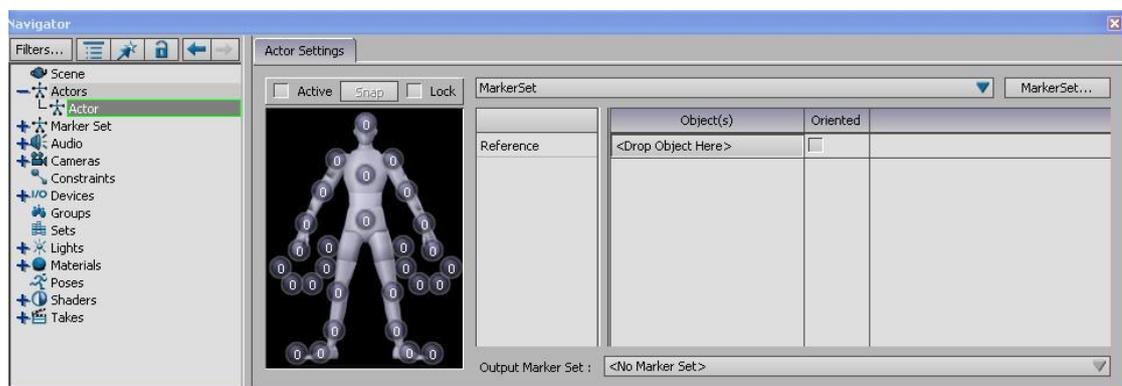


Fig 5.1.1 the Motion builder actor marker settings

All that is left to do is to select the markers and alt-drag them into the relevant circles on the actors settings.

Once all markers are in place, hitting the snap button and selecting TR will calculate the rigid body solver and get the 3D actor moving along with the marker data.

Section 5.2: Importing a character from Max

Sub - Section 5.2.1: Importing the model and rig

To stop any problems with conflicting object, node or texture names, each model was imported into motion builder using the FBX importer separately, set up within motion builder and given its own namespace.

A namespace is similar to namespace in C++ where the namespace precedes the name of the object, node or texture. This removes any chance of conflicting names.

All objects were given a positing node at the top of the hierarchy so to apply all translation, rotation and transforming operations to the node and all the rest would follow suit.

Textures are mostly exported from 3ds Max using the FBX plug-in and therefore are created automatically in Motion Builder.

Sub - Section 5.2.2: Characterization and Re-Targeting

As all characters were skinned and rigged using a standard biped, the 3ds Max biped character template was used to quick characterize all bipeds. As each is saved in its own namespace, this was a quick and easy process. Once the rig had been characterized, pre-recorded motions were then merged onto the rig using pre solved data from the Motion Capture Studio.

The two characters which were to be driven in real-time have their character motion targets set to their respective actors and the scene was saved in case of any problems.

Motion Builder handles all the real-time solving and retargeting so you don't have to worry about any of that as long as it is set up properly.

Section 5.3: User Interactions

Sub - Section 5.3.1: Controlling the cameras

To allow the user to interactably move the cameras and switch to different cameras during the performance, it was necessary to employ some type of user interaction device. For this we used a Microsoft X-Box 360 wireless controller. The 360 controller is familiar to many people as is easy to use.

Motion Builder already has a controller device, so it was a simple case of linking the 360 controller up within motion builder, do to this, a relationship constraint was designed and employed. Fig 5.3.1.1 shows the complex network of mathematical and logical nodes to handle the camera controls.

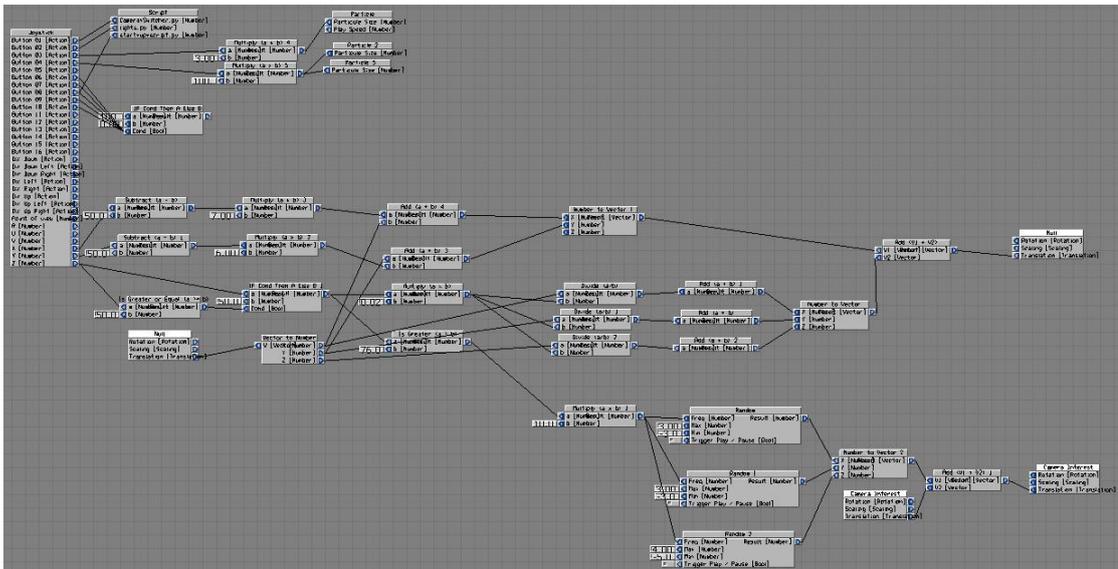


Fig 5.3.1.1 The Camera Controls Relationship Constraint

Motion builder already has the ability to switch between cameras, but only using a preset shortcut key, and with not ability to map that shortcut onto the 360 controller, a python script was written to handle this. The script simply gets an array of all the cameras in the scene, and removes the ones premade by motion builder (top, front, left and right) and searches the array for the active camera. The script then gets the next camera in the array and set that as the active camera. In all the script is only 40 lines long. In order to call this

Conducting the performance: Setting up a motion capture real-time performance

in real-time a script device was added and called within the Camera Controls Relationship constraint.

Sub - Section 5.3.2: Stage effects

Our setup includes two different stage effects, Smoke and Sparks. Both are created using shader based particle systems within Motion Builder and controlled and activated via the Camera Control Relationship constraint.

Each particle system is attached to a box which is used as an emitter, so to be able to position the location of the particle system.

In the Camera Control Relationship constraint, each particle system is brought into the constraint and set as a receiver with the size being the value that is changed.

This allows the systems to be tweaked in an interactive manor allowing us to see and tweak the effects as they are being triggered.

Sub - Section 5.3.3: Light Show

The lights are simple spot light with a volumetric cone to add effect.

To give the user the ability to change the light color interactively, another python script was written to handle this. 3 randomly generated float values were made for each light so each light had a different RGB color value.

Like the script to switch between cameras, the light changing script was loaded into a script device so it could be controlled by the Camera Controls Relationship constraint.

Section 5.4: The Real-Time Plug-in

The real-time plug-in for Motion Builder 2009 was built in house and is still at a beta release. The functionality of the real-time plug-in for 2009 is the same as the official motion builder 7.5 plug-in.

With the c3d data imported and applied to the actor, the real-time deviled should be dragged from the Asset Browser into the 3D scene. This will create a instance of the real-time plug-in in the scene. The IP address of the plug-in

Conducting the performance: Setting up a motion capture real-time performance

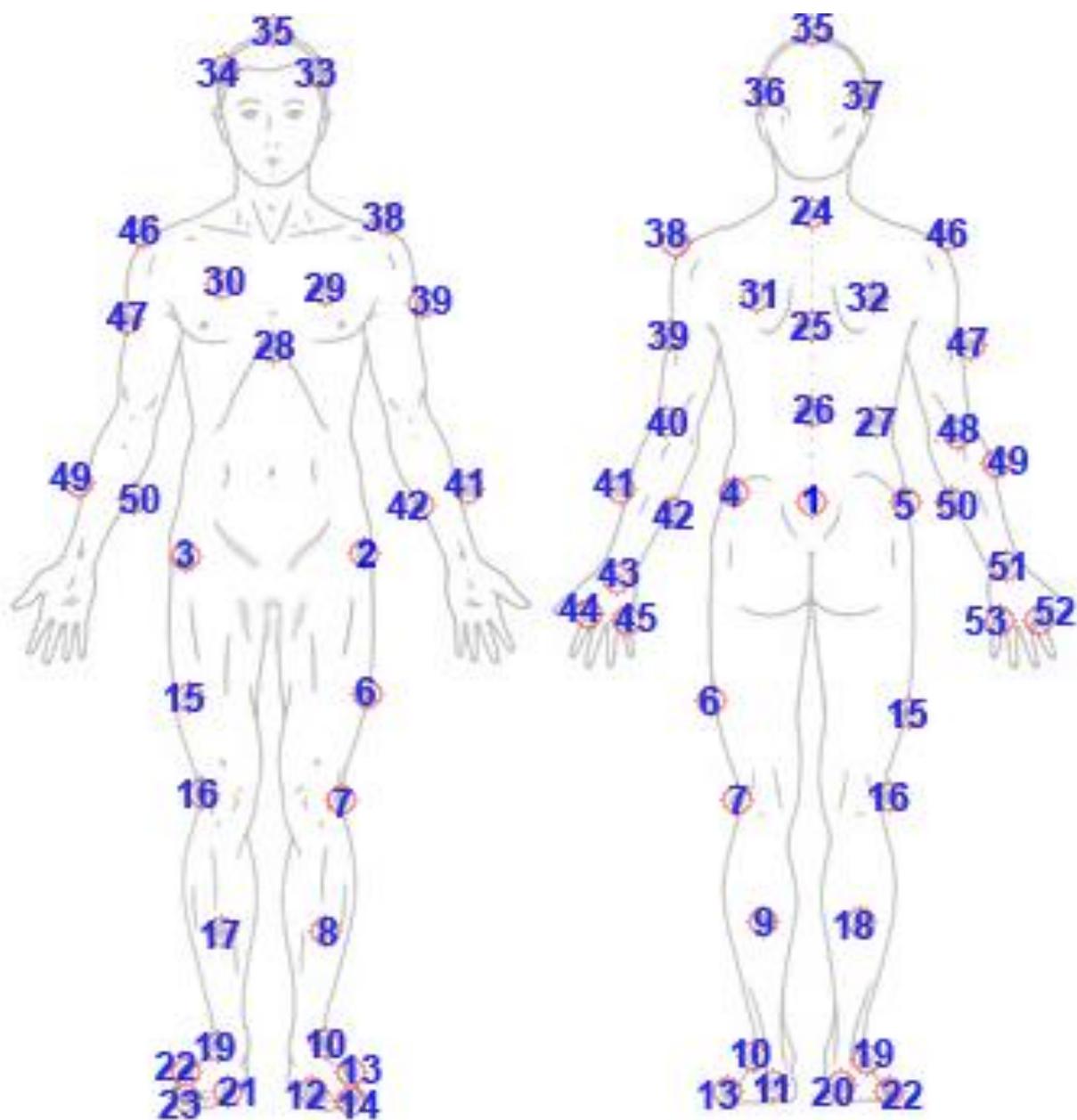
should be set 192.168.0.2, the TARSUS machine, and the optical root to the c3d optical root. Hitting update node will start the real-time feed.

That is a real-time performance!

References

RTGROP Group 3. (2009). *RTGROP group 3 website*. Retrieved 05 24, 2009, from <http://www.piersshepherd.net/rtgrp/>

Appendix A – The G-53 Marker Set



Marker Number	Marker Name	Marker Code
1	Root	Root
2	Left Front Waist	LFWT
3	Right Front Waist	RFWT
4	Left Back Waist	LBWT

Conducting the performance: Setting up a motion capture real-time performance

5	Right Back Waist	RBWT
6	Left Thigh	LTHI
7	Left Knee	LKNE
8	Left Shin	LSHN
9	Left Calf	LCALF
10	Left Ankle	LANK
11	Left Heel	LHEE
12	Left Metatarsus 5	LMT5
13	Left Metatarsus 1	LMT1
14	Left Toe	LTOE
15	Right Thigh	RTHI
16	Right Knee	RKNE
17	Right Shin	RSHN
18	Right Calf	RCALF
19	Right Ankle	RANK
20	Right Heel	RHEE
21	Right Metatarsus 5	RMT5
22	Right Metatarsus 1	RMT1
23	Right Toe	RTOE
24	Top Spine	TopSpine
25	Middle Spine	MidBack
26	Lower Spine	LowerBack
27	Character ID	Pelvis
28	Sternum	STRN
29	Left Front Shoulder	LFTShould
30	Right Front Shoulder	RFTShould
31	Left Back Shoulder	LRRShould
32	Right Back Shoulder	RRRShould
33	Left Front Head	LFHD
34	Right Front Head	RFHD
35	Top of head	ARIEL
36	Left Back Head	LBHD
37	Right Back Head	RBHD

Conducting the performance: Setting up a motion capture real-time performance

38	Left Shoulder	LSHO
39	Left Upper Arm	LUPA
40	Left Elbow	LELB
41	Left Forearm Inner	LFRMInner
42	Left Forearm Outer	LFRMOuter
43	Left Wrist	LWRIST
44	Left Thumb	LTHUMB
45	Left Pinkey	LPINKY
46	Right Shoulder	RSHO
47	Right Upper Arm	RUPA
48	Right Elbow	RELB
49	Right Forearm Inner	RFRMInner
50	Right Forearm Outer	RFRMOuter
51	Right Wrist	RWRIST
52	Right Thumb	RTHUMB
53	Right Pinkey	RPINKY