
Desarrollo de flujos de trabajo para el uso y aplicación de la tecnología de captura de movimiento en animación digital

Alejandro Micheo Marves



UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Desarrollo de flujos de trabajo para el uso y aplicación de la tecnología de captura de movimiento en animación digital

Trabajo de graduación presentado por Alejandro Micheo Marves para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2025

UNIVERSIDAD DEL VALLE DE GUATEMALA
Facultad de Ingeniería



Desarrollo de flujos de trabajo para el uso y aplicación de la tecnología de captura de movimiento en animación digital

Trabajo de graduación presentado por Alejandro Micheo Marves para optar al grado académico de Licenciado en Ingeniería Mecatrónica

Guatemala,

2025

Vo.Bo.:



(f)

M. Sc. Carlos Esquit

Tribunal Examinador:



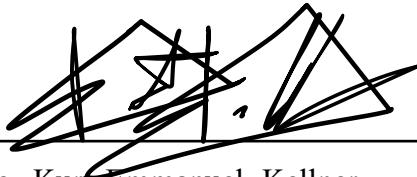
(f)

M.Sc. Carlos Esquit



(f)

M. Sc. Miguel Enrique Zea Arenales



(f)

Ing. Kurt Emmanuel Kellner

Fecha de aprobación: Guatemala, 13 de febrero de 2025.

Mis éxitos en la universidad no hubieran sido posibles sin el apoyo de mis papás, que siempre se han esforzado en darme las mejores oportunidades para estudiar y crecer. Mi ambición por lograr nuevos objetivos es gracias a mi mamá, Claudia Marves, y el optimismo para ver siempre el lado bueno de la vida es gracias a mi papá, Juan José Micheo. Mi desempeño no hubiera sido el mismo sin los desayunos, almuerzos y cenas de mi abuelita. Mi hambre no por desayunos, sino por el conocimiento, ha sido alimentada por las esporádicas conversaciones con Joseph Ayala. Mi entusiasmo y energía no estaría ahí si no fuera por el constante apoyo de mis amigos. Mi confianza se la debo a mis hermanas, Ximena y Fátima, que no fallan en hacerme ver mi potencial. Mis ganas de mejorar son gracias a Sara, que me acompañó a lo largo de este trayecto.

Mis logros son un reflejo de las personas que lo dan todo por mí.

Prefacio	III
Lista de figuras	IX
Lista de cuadros	X
Resumen	XI
Abstract	XII
1. Introducción	1
2. Antecedentes	2
2.1. Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica	2
2.2. Motive: Optical Motion Capture Software	3
2.3. Unreal Engine	4
2.4. Autodesk Maya y MotionBuilder	5
2.5. Unity	6
3. Justificación	8
4. Objetivos	10
4.1. Objetivo general	10
4.2. Objetivos específicos	10
5. Alcance	11
6. Marco teórico	12
6.1. Animación 3D	12
6.2. Proceso de producción de animación 3D	12
6.3. <i>Rigging</i>	13
6.3.1. Sistema de esqueleto	14
6.3.2. <i>Parenting</i>	18

6.3.3.	Cinemática directa e inversa	18
6.4.	Técnicas de captura de movimiento	19
6.5.	Sistema de captura óptico pasivo	19
6.6.	Preparación para la captura	21
6.6.1.	Volumen de captura	21
6.7.	Cámaras Prime x41 de OptiTrack	22
6.8.	Configuración de marcadores	23
6.9.	<i>Retargeting</i>	25
7.	Preparación del volumen de captura	26
7.1.	Configuración de las cámaras	26
7.1.1.	Enfoque	27
7.1.2.	Posicionamiento de las cámaras	28
7.1.3.	Altura de las cámaras	28
7.1.4.	Volumen de captura por cuadrantes	29
7.2.	Proceso de calibración	32
7.2.1.	<i>Masking</i>	32
7.2.2.	<i>Wanding</i>	32
7.2.3.	Establecer el origen	34
8.	Configuración y exportación de esqueleto	35
8.1.	Configuración de marcadores	35
8.1.1.	Selección de esqueleto y preparación de marcadores	36
8.1.2.	Colocación de marcadores	37
8.1.3.	Resultado de la configuración del esqueleto	39
8.2.	Grabación y exportación de información de movimiento	40
8.2.1.	Carpeta de sesión	40
8.2.2.	Grabación de movimiento en Motive	41
8.2.3.	Exportación de datos	41
9.	Transmisión en tiempo real de la información de movimiento (Motive)	43
9.1.	Data Streaming Pane	43
10.	Transmisión en tiempo real de la información de movimiento a los programas de animación digital	46
10.1.	Transmisión en tiempo real de la información de movimiento (MotionBuilder)	47
10.1.1.	Autodesk MotionBuilder: OptiTrack Skeleton Plugin	47
10.1.2.	Redireccionamiento de animación del esqueleto de Motive a un personaje en MotionBuilder	49
10.2.	Transmisión en tiempo real de la información de movimiento (Unity)	51
10.3.	Transmisión en tiempo real de la información de movimiento (Unreal Engine)	53
10.3.1.	Unreal Engine: OptiTrack Live Link Plugin	53
11.	Pruebas de movimientos en el volumen de captura	57
11.1.	Listado de movimientos	57
12.	Conclusiones	60
13.	Recomendaciones	61

Lista de figuras

1.	Interfaz de Motive [4]	4
2.	Personaje en Unreal Engine animado con Live Link [4]	5
3.	Transmisión de esqueleto de Motive a MotionBuilder [4]	6
4.	Transmisión en tiempo real de Motive a Unity [4]	7
5.	Una imagen del <i>rig</i> de un arma (izquierda) y un humano (derecha) [10]	13
6.	Ejemplo de sistema de esqueleto [10]	14
7.	Ejemplo de junta de bisagra [10]	15
8.	Ejemplo de junta articulada [10]	15
9.	Ejemplo de junta de pivote [10]	16
10.	Ejemplo de sistema de junta de rótula [10]	16
11.	Un <i>rig</i> jerárquico sencillo en Maya [10]	17
12.	<i>Rig</i> simple de cabeza con un pivote en el centro de la esfera [10]	17
13.	<i>Rig</i> simple de cabeza con un pivote correctamente colocado en la parte inferior de la esfera [10]	17
14.	Una relación padre/hijo en Maya [10]	18
15.	Marcador reflectivo de OptiTrack [3]	19
16.	Participante con traje de captura óptico posicionado en el centro del volumen de captura del salón CIT-116 en la Universidad del Valle de Guatemala [3] . .	20
17.	Volumen de captura con 8 cámaras apuntando al centro [11]	21
18.	Volumen de captura por cuadrantes[11]	22
19.	Cámara Prime x41 de OptiTrack [3]	22
20.	Vista frontal de la colocación de las cámaras del Robotat [3]	23
21.	Sensor óptico pasivo de <i>OptiTrack</i> [3]	24
22.	Traje de captura de OptiTrack [3]	24
23.	Diagrama de la configuración de las cámaras en el volumen de captura antes de los cambios	27
24.	Demostración de un marcador desenfocado, moderadamente enfocado y co- rrectamente enfocado, en ese orden [3]	27
25.	Patatas del trípode de una de las cámaras, mostrando sus calcomanías corres- pondientes para determinar la posición exacta en la que se debe colocar . . .	28

26.	Visualización en escala de grises de la cámara 2 con los Pololu 3pi debajo de cada cámara y el participante humano en el centro de la plataforma	29
27.	Diferencia de altura entre cámaras centrales (3 m) y laterales (2.6 m)	29
28.	Diagrama de la plataforma de captura separada en cuadrantes	30
29.	Vista en planta de la posición de cada cámara y el punto en el que se enfocó cada una	31
30.	Visualización en escala de grises de la vista de la cámara 2 con los Pololu 3pi debajo de cada cámara	31
31.	Reflejos no deseados antes y después del proceso de enmascarado [3].	32
32.	Ejemplo de <i>wanding</i> en el volumen de captura.	33
33.	Trayectorias de la varilla en el proceso de <i>wanding</i> en el Camera Viewport . .	34
34.	Escuadra usada para definir el origen en el centro de la plataforma	34
35.	Configuración de marcadores para esqueleto de cuerpo completo con 37 marcadores	36
36.	Traje de captura usado para las pruebas, incluye un sombrero, una chaqueta, un pantalón y un par de guantes. La fotografía fue tomada después de las pruebas, los marcadores siempre se deben colocar cuando el participante ya tiene puesto el traje	37
37.	Ejemplo de participante con traje de OptiTrack adoptando una pose de calibración <i>T-Pose</i> [3]	38
38.	Marcadores de segmento asimétricos para diferenciar la pierna derecha de la izquierda [3]	38
39.	Participante en <i>T-Pose</i> para la calibración del esqueleto	39
40.	Resultado de la configuración del esqueleto en Motive	39
41.	Carpetas de sesión ordenadas en el <i>Data Management Pane</i> [3]	40
42.	Posición en la pantalla de Motive del Control Deck	41
43.	Animación de salto exportada de Motive a un personaje en Motion Builder . .	42
44.	Animación de salto exportada de Motive a un personaje en Unity	42
45.	Configuraciones para la transmisión de datos en el Data Streaming Pane de Motive	44
46.	Asset Browser en MotionBuilder	47
47.	Ventana para configurar el Skeleton Plugin en la ventana de Navigator de MotionBuilder	48
48.	Transmisión en vivo de datos de Motive a MotionBuilder	48
49.	Esqueleto transmitido de Motive a MotionBuilder	49
50.	Personaje <i>mia_rigged</i> viendo hacia el eje Z positivo en el Viewer Pane de MotionBuilder	49
51.	Modelo de <i>mia_rigged</i> caracterizado	50
52.	Ventana de Character Controls de <i>mia_rigged</i> con el esqueleto de OptiTrack seleccionado	50
53.	Personaje animado por el esqueleto transmitido desde Motive	51
54.	Optitrack Unity Plugin en el folder de Assets en la pestaña llamada Project de Unity	51
55.	Configuraciones en el Optitrack Streaming Client	52

56.	Menú de OptiTrack Skeleton Animator (Script) del personaje seleccionado en la escena	53
57.	Botón de reproducción en la interfaz de Unity	53
58.	Live Link Pane en Unreal Engine para configurar los datos de la red	54
59.	Live Link Pane con listado de los <i>assets</i> que se están transmitiendo desde Motive	54
60.	Live Link Pane con listado de los <i>assets</i> que se están transmitiendo desde Motive	55
61.	Live Link Display de la data transmitida desde Motive	56
62.	Comparación entre la información de movimiento recibida de Motive (esqueleto verde) con los movimientos realizados por el personaje (esqueleto blanco)	56

Lista de cuadros

1. Evaluación del cumplimiento de los movimientos en el volumen de captura,
con el número de *take* correspondiente en FBX. 58

En este trabajo de graduación se implementó el uso de la tecnología de captura de movimiento para aplicaciones de animación digital. En la Universidad del Valle de Guatemala, se reconfiguró el volumen de captura del salón CIT-116 compuesto por seis cámaras OptiTrack, con el objetivo de utilizar la plataforma para la captura de movimiento en humanos.

Para facilitar la transmisión de los datos de movimiento, se emplearon *plugins* proporcionados por OptiTrack, los cuales permitieron establecer una conexión entre el *software* de captura Motive y diversos programas de animación digital. Se estableció un flujo de trabajo para los programas de MotionBuilder, Unity y Unreal Engine. Gracias a esta integración, se logró transferir un esqueleto con datos de movimiento en tiempo real grabado por el sistema de captura de movimiento de la universidad.

Asimismo, se documentó todo el proceso en un manual detallado que describe paso a paso las configuraciones necesarias para utilizar la plataforma de captura de movimiento, desde el posicionamiento de las cámaras y la calibración del sistema y los trajes de captura de movimiento, hasta la integración con los programas de animación digital. El manual proporciona instrucciones claras sobre cómo capturar y exportar datos de movimiento utilizando el programa Motive, al igual que cómo transmitir en tiempo real estas animaciones a los distintos programas de animación digital a través de la red de comunicación del Robotat.

The main objective of this thesis was to implement the use of motion capture technology for digital animation applications. The capture volume, composed of six cameras located in the CIT-116 laboratory at the Universidad del Valle de Guatemala, was reconfigured so as to be able to use the platform for human motion capture as well as robotic applications.

To facilitate the transmission of motion data, plugins provided by OptiTrack were used to establish a connection between Motive (the motion capture software provided by OptiTrack) and various digital animation programs. A pipeline was established for MotionBuilder, Unity, and Unreal Engine programs. Thanks to this integration, a skeleton with real-time motion data recorded by the university's motion capture system was transferred via a Wi-Fi network to the 3D animation softwares.

The entire process was documented in a detailed step-by-step guide that describes the configurations required to use the motion capture platform, from camera positioning and calibration of the motion capture system and suits to the integration with the digital animation programs. The manual provides clear instructions on how to capture and export motion data using Motive, as well as how to transfer these animations to the various digital animation programs via the Robotat Wi-Fi network.

CAPÍTULO 1

Introducción

En el laboratorio CIT-116 se implementó un ecosistema de pruebas y desarrollo de captura de movimiento denominado Robotat. Esta plataforma utiliza cámaras de captura y el programa Motive de OptiTrack para procesar datos de movimiento y posición de sensores ópticos pasivos. Este trabajo se centra en aprovechar la plataforma del Robotat para la captura de movimiento en humanos, adaptándola a aplicaciones de animación digital. Con este fin, se estudió y documentó el uso de Motive y sus funcionalidades específicas para la captura de movimiento en humanos. Además, se integraron programas de animación digital con Motive para establecer una transferencia fluida de información entre ambas plataformas. Todo el proceso se recopiló en un manual de usuario que detalla paso a paso los procedimientos necesarios para capturar movimiento mediante el sistema de cámaras, trajes y software de OptiTrack.

Primero, se describen las modificaciones realizadas a la configuración de las cámaras en el volumen de captura de la plataforma Robotat, con el objetivo de que esta pueda capturar de manera efectiva tanto a actores humanos como a agentes robóticos. Una vez configurado un volumen de captura adecuado, se explica cómo posicionar los marcadores en los trajes de captura pasivos para asegurar la precisión en la captura de movimientos de un actor humano. Finalmente, se detalla el proceso de exportación de los datos de movimiento a los programas de animación digital MotionBuilder, Unity y Unreal Engine. Además, se presenta una lista de movimientos que se probaron en la plataforma, junto con los resultados obtenidos al transmitir el movimiento desde Motive al programa de MotionBuilder.

La tecnología de captura de movimiento se utiliza para simular el movimiento de personas u objetos en un ambiente virtual. En la actualidad se utiliza en la industria del cine, los videojuegos e incluso la biomecánica para interpretar de una manera exacta y auténtica los movimientos grabados [1].

La Universidad del Valle de Guatemala cuenta con un ecosistema de pruebas y desarrollo de captura de movimiento denominado Robotat, cuya infraestructura puede ser aprovechada para la captura de movimiento de humanos.

2.1. Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica

Perafán (2022) describe que el ecosistema del Robotat del laboratorio de robótica CIT-116 de la Universidad del Valle de Guatemala se desarrolló con el fin de que la institución contara con un lugar especializado para la investigación e implementación de procesos robóticos. Con una red *wifi* de por medio, se estableció una comunicación entre el sistema de captura de movimiento de OptiTrack junto con cualquier tipo de agente robótico con el que se desea experimentar. Asimismo, se implementó un protocolo de conexión *wifi* para que múltiples clientes se conectaran a la red al mismo tiempo.

Para crear el ecosistema, el primer paso fue configurar el *hardware* y *software* del sistema de captura de movimiento. El *hardware* que se instaló consta de 6 cámaras Prime×41, 6 abrazaderas Manfrotto, 6 trípodes de cámara, 1 switch NETGEAR ProSafe GS728TPPv2 y 1 tarjeta de red Intel I210-T1. Después de colocar las cámaras en sus respectivas posiciones, se configuró el *software* de captura de movimiento de OptiTrack llamado Motive. Con el fin de que todo el laboratorio se comunicara por una misma red, se configuró una red *wifi* que habilita el intercambio bidireccional de información entre la computadora que ejecuta el *software* de Motive y los agentes robóticos.

Para poder determinar si la conexión entre el *hardware* y el programa fueron exitosas, se utilizó un cuerpo rígido compuesto por tres sensores ópticos pasivos. Este se colocó en distintas posiciones del área de grabación, se comparó la posición obtenida por Motive con las mediciones reales. El mayor porcentaje de error obtenido fue de 5.28 %. Para determinar si la red funcionaba con múltiples clientes, se realizó el envío de información de la pose de dos cuerpos rígidos en el ambiente de captura, a dos microcontroladores ESP32. Funcionando al mismo tiempo, ambos microcontroladores obtuvieron la pose de su respectivo identificador, por lo que la implementación de la red de comunicación multiagente fue exitosa.

El ecosistema del Robotat permite a los usuarios realizar pruebas para distintos tipos de robot, ya sea para el desarrollo de robots móviles o la robótica de enjambre. Sin embargo, las pruebas realizadas fueron solo con cuerpos rígidos, no se investigó la funcionalidad del sistema con trajes de captura de movimiento para humanos [2].

2.2. Motive: Optical Motion Capture Software

La empresa de OptiTrack que desarrolla las cámaras Prime x41 que se operan actualmente en el laboratorio de robótica CIT-116 tiene su propio *software* de captura de movimiento llamado Motive. El programa se encarga de rastrear el movimiento de marcadores reflectivos y traducirlo al mundo digital con una precisión óptima [3].

El *software* se encuentra en su versión 3.1, en la que ofrece una amplia gama de funciones. Principalmente, se caracteriza por su calibración continua, en la que automáticamente le indica al usuario cuando la captura de un marcador no ha sido óptima y cómo se puede corregir, incluso en ambientes desafiantes. Así mismo, han implementado captura asistida por aprendizaje automático, que le permite al usuario rastrear el movimiento de personas, animales o incluso cualquier tipo de objeto móvil. Además de los sensores ópticos pasivos, OptiTrack ofrece sensores que combinan las propiedades de marcadores reflectivos con sensores IMU (Inertial Measurement Unit). Esta fusión de sensores le ofrece al usuario la capacidad de obtener una captura de datos fluida [3].

Para aplicaciones con humanos, Motive cuenta con una herramienta llamada *solver*. Con ella se estima la pose de cada segmento del cuerpo que tenga un marcador. Así mismo, define con exactitud los movimientos y rastrea los marcadores incluso en casos en los que están momentáneamente fuera de la visibilidad de las cámaras. Para poner en uso la data capturada por el *software*, se ofrecen una variedad de *plugins* para distintas ramas de trabajo. Para aplicaciones de animación digital se puede conectar con Unreal Engine, Unity, Motion Builder y Maya. Además, cada *plugin* cuenta con documentación exhaustiva por parte de OptiTrack para poner en uso cada una de las herramientas [3].



Figura 1: Interfaz de Motive [4]

2.3. Unreal Engine

Desarrollado por Epic Games, Unreal Engine ofrece a sus usuarios un ambiente especializado en el diseño de personajes y entornos computarizados. Inicialmente, el *software* fue creado por Sweeney (1995) como un editor de programación en C++ especializado en videojuegos, en el que el usuario podía renderizar sus creaciones al igual que modificar el escenario de su juego mientras lo visualizaba. En la actualidad la herramienta es usada no sólo para videojuegos, sino también en la industria del cine, arquitectura, eventos en vivo o simulaciones físicas [5].

En el caso de la animación en 3D por medio de *motion capture*, Unreal Engine cuenta con procesos definidos para usar la tecnología en la creación de los personajes computarizados. Para delimitar las partes móviles y las articulaciones de un cuerpo digital se usa una malla esquelética. Las mallas son representaciones simples de los eslabones que componen el cuerpo de un personaje. Estas se pueden manipular hasta llegar al movimiento deseado. Cuando se ha definido una malla esquelética Unreal Engine, ofrece un sistema de animación llamado plano de animación, en el que le permite al usuario crear guiones visuales para el movimiento de sus personajes. Mediante el plano de animación el usuario puede anclar la malla esquelética a la información de movimiento transmitida por un programa de movimiento de captura [6].

Para utilizar el plano de animación con el programa de Motive, se puede activar el *plugin* de Live Link [6]. Esta es una herramienta que le permite a Unreal Engine recibir datos transmitidos por Motive en tiempo real; lo que le proporciona al usuario la opción de previsualizar su animación mientras se está actuando. Con herramientas como estas, el animador puede hacer su proceso de trabajo más rápido y eficiente [4].

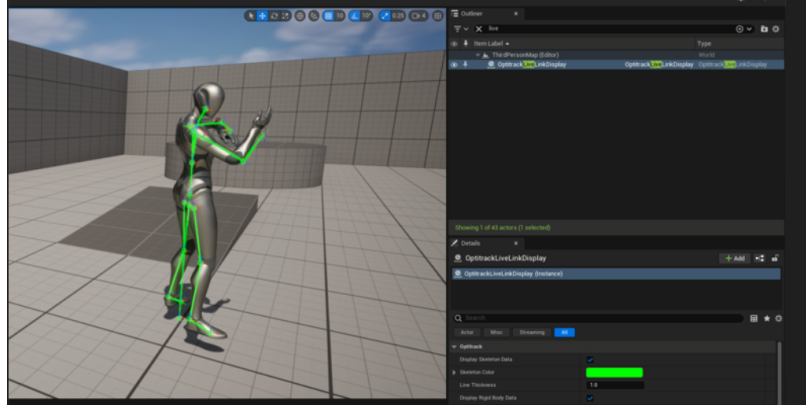


Figura 2: Personaje en Unreal Engine animado con Live Link [4]

2.4. Autodesk Maya y MotionBuilder

Autodesk tiene su propio *software* para animar y modelar personajes en 3D llamado Maya, en donde el usuario puede generar simulaciones y renderizar sus creaciones. El *software* de Maya fue creado por una compañía llamada Alias Systems Corporation fundada en 1983. Hoy en día Maya se ha convertido en un estándar en la industria del cine [7].

El objetivo de este programa es proveer herramientas para facilitar los procesos iterativos del modelado 3D y así darle la libertad al cliente de enfocarse en lo creativo. Esto lo logra mediante el uso de su propio lenguaje de programación llamado Maya Embedded Language (MEL). Así mismo, ofrece la capacidad de agregar detalles minuciosos de modelado a los personajes y escenarios para crear mundos complejos y realistas. Además, cuenta con un API para los lenguajes de C++ y Python que el usuario puede usar para automatizar procesos tediosos [7].

OptiTrack cuenta con un *plugin* para Maya, sin embargo, este solo brinda una integración de cámara virtual en tiempo real. Lo que quiere decir que el usuario tendrá control de la posición y orientación de las cámaras en el ambiente virtual. Si se desea usar las cámaras y trajes de OptiTrack para aplicaciones más complejas se recomienda usar el *software* de Autodesk, MotionBuilder. Con MotionBuilder es posible la transmisión en tiempo real de los marcadores, cuerpos rígidos, esqueletos y cámaras virtuales del sistema de Optitrack [4].

A diferencia de Maya, en donde el *software* acompaña al usuario en todo el proceso de creación de un personaje desde el modelado hasta sus movimientos e interacción con el ambiente, MotionBuilder se utiliza únicamente para la animación del personaje. Por lo tanto, el procedimiento para transformar la información capturada por las cámaras de OptiTrack es más simple y completo. Los datos de los sensores del traje de captura de movimiento se pueden asociar directamente con los nodos del esqueleto virtual usando la herramienta de Auto-Characterize de MotionBuilder [8].

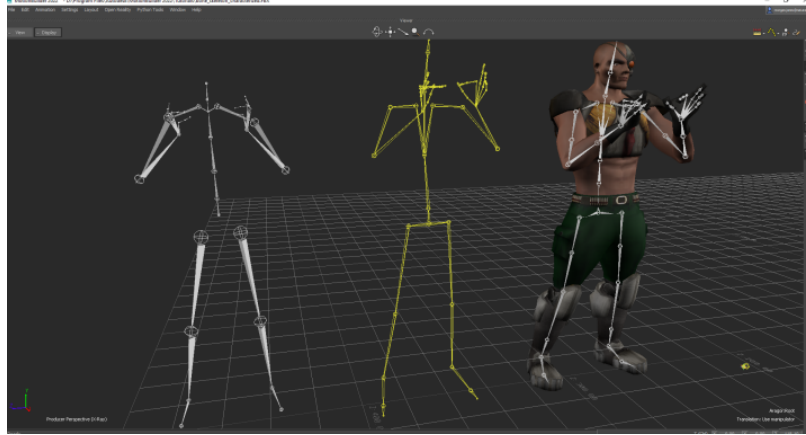


Figura 3: Transmisión de esqueleto de Motive a MotionBuilder [4]

2.5. Unity

Con datos obtenidos en el 2022, Unity declara que el 70 % de los mil juegos más destacados para teléfono móvil se desarrollaron con Unity. La herramienta es una de las más completas en el mercado para el desarrollo de ambientes y personajes generados por computadora en 3D. Aunque inicialmente se desarrolló como un software para democratizar la industria de los videojuegos, actualmente se utiliza también en aplicaciones de cine o en la industria automóvil y aeroespacial, entre otras [9].

Para la creación de videojuegos, Unity ofrece un servicio versátil en el que el usuario puede crear un juego para cualquier tipo de plataforma, ya sea para computadora, teléfono móvil, consola o incluso realidad virtual. Además, es muy amigable para programadores de todos los niveles. El lenguaje de programación que usa es C# [9].

Unity también ha evolucionado su plataforma para que se puede utilizar en aplicaciones de cine. El usuario puede crear personajes con facilidad gracias a avances en simulaciones físicas y algoritmos de aprendizaje automático. Además, los personajes se pueden traer a la vida con la implementación de herramientas de animación. El proceso de animación en Unity se caracteriza por ser fácil de iterar y por su habilidad de crear tomas cinematográficas en tiempo real debido a su proceso de renderizado de alta definición [9].

El *plugin* de OptiTrack en Unity ofrece una interfaz en tiempo real para transmitir los datos de cuerpos rígidos y esqueletos con seis grados de libertad de un sistema de captura de movimiento. Las animaciones obtenidas con los trajes de captura se pueden aplicar automáticamente a los personajes usando el sistema de Avatars de Unity [4].

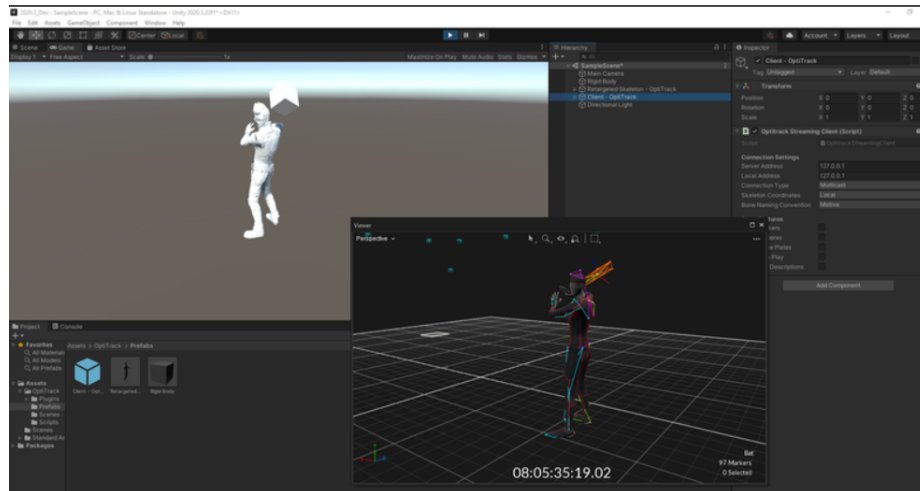


Figura 4: Transmisión en tiempo real de Motive a Unity [4]

Actualmente, la tecnología de captura de movimiento se ha convertido en un estándar en la industria del cine y los videojuegos, permitiendo lograr movimientos realistas y complejos en la animación de personajes tridimensionales. Aún cuando la tecnología ya lleva un tiempo considerable en el mercado, su implementación en Centro América ha sido lenta o incluso inexistente. Al aprovechar la plataforma del Robotat con la que cuenta hoy la Universidad del Valle de Guatemala se puede desarrollar lo que podría ser un estudio de captura de movimiento. Hay un interés dentro de la institución para usar la tecnología de captura de movimiento en cursos de animación digital, videojuegos, mercadeo y producción de video.

Para los cursos de animación digital y videojuegos, la tecnología se puede implementar para establecer flujos de trabajo que le permitan a los usuarios animar a sus personajes tridimensionales. Les ofrecerá la oportunidad de obtener movimiento realistas para crear animaciones fluidas. Muchos de los movimientos de un avatar virtual en un videojuego son repetitivos y sutiles, por lo que pueden ser difíciles de replicar manualmente. Con la tecnología se puede ahorrar tiempo al acelerar el proceso de animación. Los cursos se verían beneficiados e incluso podrían emprender proyectos más ambiciosos al tener las herramientas necesarias para recrear animaciones complejas en un corto periodo de tiempo.

En Guatemala, los sistemas de captura de movimiento no están disponibles y el uso de personajes animados mediante esta tecnología no se ha explorado en publicidad digital. Por lo tanto, en los cursos de mercadeo, esta tecnología podría emplearse para crear personajes animados en anuncios, brindando a los estudiantes o incluso a empresas externas, la oportunidad de diseñar campañas creativas y únicas. Este enfoque no solo enriquecería el área de mercadeo, sino que también podría extenderse a otras áreas como la industria del cine, donde existe un interés por integrar herramientas de animación digital.

La industria del cine en Guatemala ha crecido con en el tiempo, sin embargo en la actualidad la Universidad del Valle de Guatemala no ofrece ninguna clase relacionada a cine. La más parecida es la de producción de video, en la que también existe interés para implementar la tecnología. Usando herramientas como Unreal Engine, Unity o Motion Builder, se puede incorporar el uso de animaciones en 3D como parte del curso. Aun así, existen

otras instituciones que buscan desarrollar el cine en el país, como la Universidad Francisco Marroquín, que ofrece su propia escuela de cine. Con la tecnología disponible para facilitar el proceso de animación, se puede enriquecer la educación audiovisual, permitiendo a los estudiantes experimentar con la tecnología y elevar la calidad de sus producciones. Esto no solo potenciará la creatividad y la innovación en los proyectos estudiantiles, sino que también puede contribuir al crecimiento de la industria cinematográfica guatemalteca. Del mismo modo, empresas de cine extranjeras podrían contratar al estudio de la universidad para desarrollar sus animaciones.

La tecnología puede beneficiar a la universidad en un ámbito académico, al igual que darle la posibilidad de ofrecer el servicio a empresas extranjeras y ser pioneros en la implementación de la tecnología en el área de Centro América.

4.1. Objetivo general

Integrar las herramientas de *software* y documentar los procesos que permitan utilizar la información obtenida por trajes de captura de movimiento de OptiTrack en animación digital.

4.2. Objetivos específicos

- Estudiar y documentar el uso del programa Motive y las funcionalidades que ofrece para la captura de movimiento en humanos.
- Integrar programas de animación digital con el programa Motive para establecer una transferencia de información de una plataforma a la otra.
- Desarrollo y documentación de flujos de trabajo para facilitar el aprendizaje y uso de la tecnología.

El trabajo busca integrar las herramientas de *software* y documentar los procesos que permitan utilizar la información obtenida por trajes de captura de movimiento de OptiTrack en animación digital. Para ello se reconfiguró el volumen de captura compuesto por seis cámaras Prime x41, que se encuentran en el salón CIT-116 de la Universidad del Valle de Guatemala. Al realizar las modificaciones, se logró que el volumen de captura se pueda usar para la captura en humanos, al igual que para agentes robóticos. Además se logró obtener información de un esqueleto usando el sistema de captura compuesto por las cámaras, un traje de sensores pasivos y el *software* de Motive. La información se pudo utilizar en los programas de MotionBuilder, Unity y Unreal Engine, ya sea como archivo FBX o mediante la red de comunicación del Robotat.

Asimismo, se redactó un manual de usuario que describe en detalle los procesos necesarios para capturar movimiento con precisión mediante el sistema de cámaras, trajes y *software* de OptiTrack. Este manual abarca la preparación adecuada del volumen de captura, la configuración de marcadores en los trajes, la exportación de una animación como archivo .FBX y los pasos para la transmisión de datos vía *wifi* hacia diversos programas de animación digital. La configuración del volumen de captura descrita en el manual corresponde a las seis cámaras disponibles en la universidad al momento de realizar este trabajo; si se adquieren más cámaras, será necesario definir una nueva configuración.

Para las grabaciones de movimiento, se necesitó un participante que utilizara el traje de captura. Los trajes con los que se realizaron las pruebas incluyen un sombrero, una chaqueta, un pantalón y un par de guantes. Debido a la falta de calzado con Velcro, se usaron calcetines con adhesivos de Velcro para colocar los marcadores en los pies del participante. Esto resultó en discrepancias en la animación final debido a desplazamientos en los marcadores. Cabe resaltar que las pruebas se realizaron con un solo participante, no se llevaron a cabo pruebas con múltiples personas actuando simultáneamente en la plataforma.

6.1. Animación 3D

Una animación es un conjunto de imágenes que se ordena en una secuencia específica con el fin de crear la ilusión de movimiento. Por lo tanto, es el trabajo del animador mover a los personajes tridimensionales fotograma por fotograma para que la audiencia perciba que los objetos se están moviendo. En el cine, la frecuencia estándar de fotogramas debe ser de 24 fotogramas por segundo para videojuegos la frecuencia sube a 30 fotogramas por segundo. En el caso de una película para que el movimiento se vea fluido, el animador debe mover a su personaje 24 veces por segundo. Por ello en una película de noventa minutos se deben planear 129,900 fotogramas para cumplir con dicho estándar. Esta es la razón por la cual la animación consume tanto tiempo y recursos [10].

Existen muchos procesos o herramientas para lograr una animación; para fines de este trabajo el único relevante es el de la animación por medio de captura de movimiento.

6.2. Proceso de producción de animación 3D

Con el fin de desarrollar una animación de una manera eficiente y rápida, el proceso de producción se segmenta en etapas en las que cada animador está a cargo de una actividad específica. El proceso completo puede llevarse a cabo por dos personas o incluso llegar a requerir más de 400 contribuyentes. La cadena de producción está compuesta por los programas, el *hardware* y el grupo de personas que trabajan en conjunto para producir una animación. Las tres etapas en las que se separa el proceso son [10]:

- Preproducción.
- Producción.

- Posproducción.

Para la industria cinematográfica, la publicidad y el desarrollo de videojuegos, las tres fases se desarrollan de una forma similar. En la preproducción, el equipo tiene el objetivo de escribir una historia atractiva con personajes interesantes. Es entonces el trabajo del equipo de producción llevar a cabo el trabajo necesario para traer la historia y a sus personajes a la vida. Por último, en la posproducción se busca afinar el producto para darle un aspecto realista a la animación al agregarle efectos especiales, correcciones de color u otros detalles faltantes [10].

El sistema de captura de movimiento a implementar forma parte de la etapa de producción, debido a que asiste a los animadores a obtener los movimientos necesarios para animar a los personajes digitales [10].

6.3. *Rigging*

Para animar cualquier objeto en 3D es necesario que se diseñe un sistema para controlar y mover las geometrías que lo forman. El sistema de control es conocido como un *rig*. Es el objetivo de los especialistas en *rigging* crear un esqueleto con restricciones y mecanismos de control para que los animadores puedan mover únicamente las articulaciones necesarias de su modelo [10].

El proceso comienza al colocar en el objeto a animar una red de uniones y huesos que juntos conforman un esqueleto. Cada unión se coloca en el lugar donde se desea que pivoten los eslabones. En el caso de un personaje humanoide o animal, cada junta se posiciona en las articulaciones sobre las que rotan los huesos. Después, se determina si el movimiento en cada punto de pivote será traslacional o rotativo. A continuación, se utiliza un deformador (herramienta que modifica la forma de un objeto), para conectar la geometría del modelo con el sistema de articulaciones, con el fin de que al mover el *rig* se mueva el personaje. Por último, se utiliza nuevamente la herramienta del deformador para agregarle detalles a la geometría y lograr que se mueva de manera más realista [10].

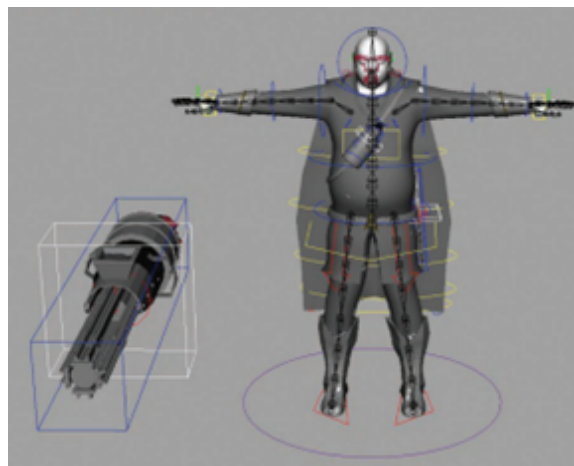


Figura 5: Una imagen del *rig* de un arma (izquierda) y un humano (derecha) [10]

A continuación se listarán conceptos básicos de los objetos que conforman un *rig*.

6.3.1. Sistema de esqueleto

La base de cualquier *rig* es el esqueleto. Este es un sistema compuesto por un conjunto jerárquico de huesos, juntas y puntos de pivote. En el que se definen los puntos de articulación como los codos, rodillas u hombros del humanoide. Su función es proporcionar una estructura básica que delimita los movimientos de un personaje. En la Figura 6 se puede observar un ejemplo de un sistema de esqueleto y en la Figura 5 se puede observar el sistema de esqueleto dentro del *rig* de un personaje [10].

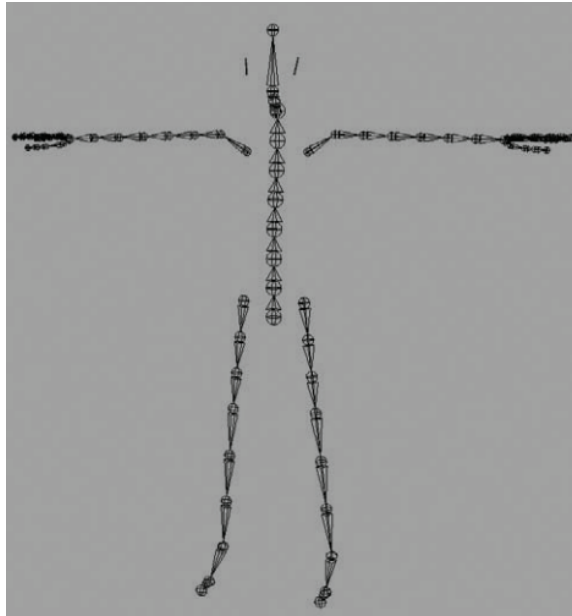


Figura 6: Ejemplo de sistema de esqueleto [10]

Para comprender mejor el funcionamiento del esqueleto, es importante entender los componentes esenciales que lo conforman: los puntos de pivote y las juntas. Estos elementos permiten definir cómo se mueven y rotan los huesos entre sí.

Juntas

Una junta o articulación es una unión estructural del esqueleto, representa un punto anatómico que determina dónde se produce una flexión o rotación entre dos huesos.

Cuatro tipos básicos de articulaciones:

- Junta de bisagra: rota en un eje, de la misma manera que lo haría un codo o una rodilla. (Figura 7).

- Junta articulada: juntas que individualmente tienen poco movimiento, pero al rotarlas en conjunto crean un movimiento con mayor amplitud. Un ejemplo de esta junta sería una columna vertebral. (Figura 8).
- Junta de pivote: rota alrededor de un eje como en el caso de un antebrazo. (Figura 9).
- Junta de rótula: puede rotar en cualquier dirección, como lo haría un hombro. (Figura 10).

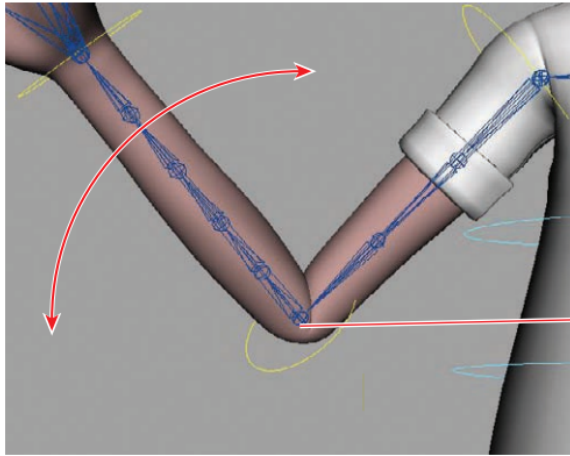


Figura 7: Ejemplo de junta de bisagra [10]

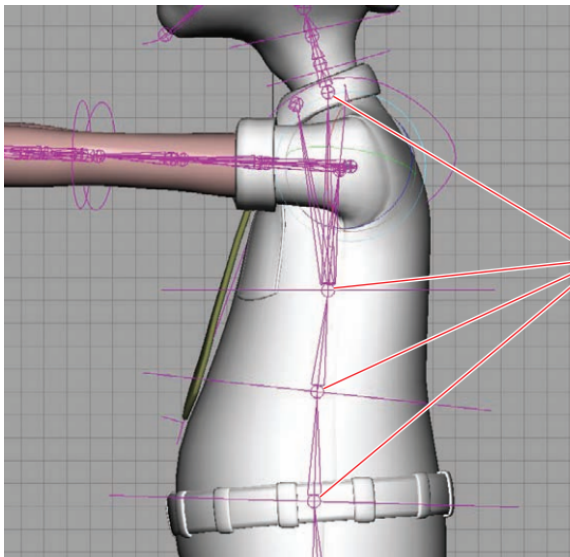


Figura 8: Ejemplo de junta articulada [10]

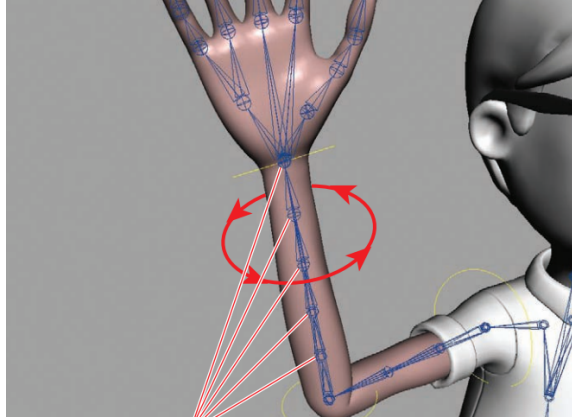


Figura 9: Ejemplo de junta de pivote [10]

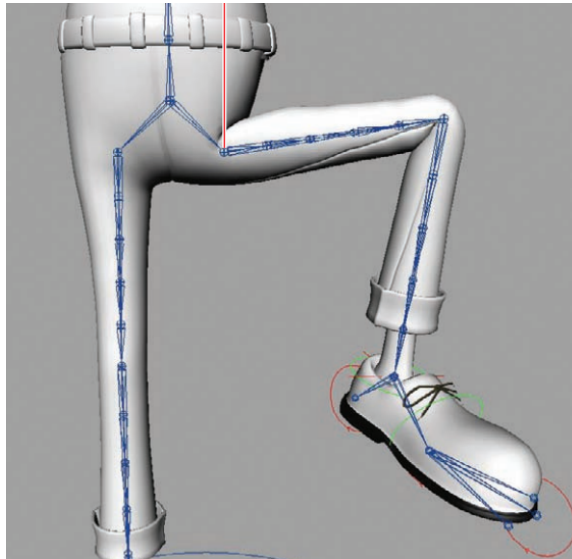


Figura 10: Ejemplo de sistema de junta de rótula [10]

Posición de pivote

En la animación 3D, la posición de pivote es un punto de referencia sobre el que un objeto rota, al igual que la posición desde la cual otros manipuladores moverán o escalarán el objeto. Es muy importante colocar correctamente la posición de pivote al momento de crear un *rig*, para crear un punto de manipulación adecuado para los animadores y un movimiento que se vea realista [10].

Como ejemplo, se puede observar en la Figura 11 una relación jerárquica básica padre/hijo en la que todas las geometrías están conectadas. Para mover la cabeza de la forma que la rotaría un humano, el punto de pivote debe estar entre el cubo rojo y la esfera que representa la cabeza. Sin embargo, el punto de pivote se encuentra por defecto en el centro de la esfera, como se puede ver en la Figura 12. Para lograr el movimiento deseado, se debe trasladar el punto de pivote a la parte inferior de la esfera, como muestra la Figura 13 [10].

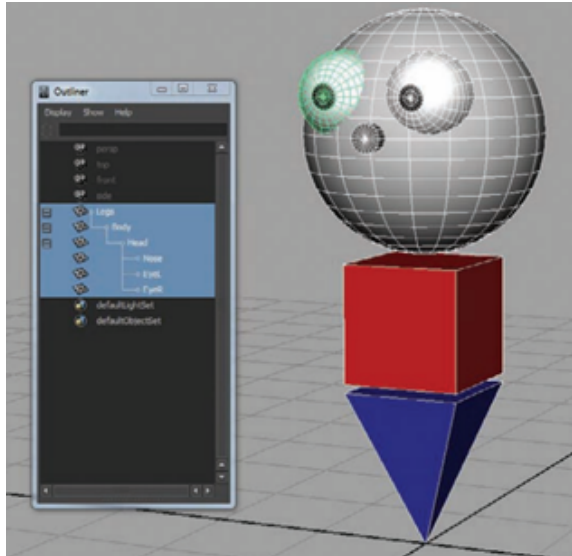


Figura 11: Un *rig* jerárquico sencillo en Maya [10]

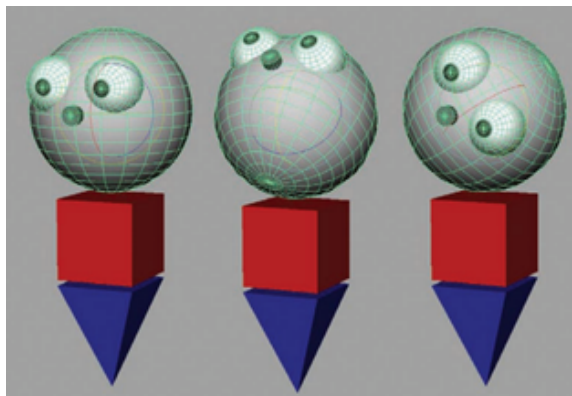


Figura 12: *Rig* simple de cabeza con un pivote en el centro de la esfera [10]

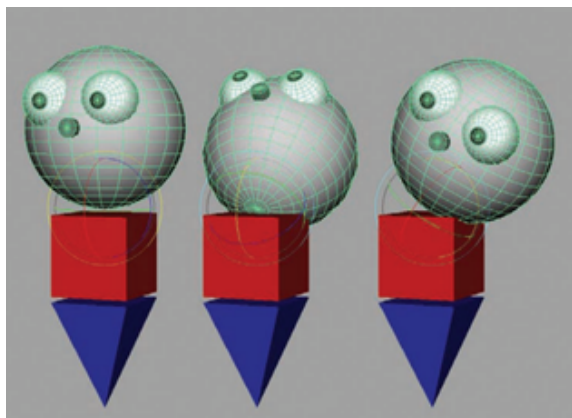


Figura 13: *Rig* simple de cabeza con un pivote correctamente colocado en la parte inferior de la esfera [10]

6.3.2. Parenting

Para un funcionamiento realista, las relaciones en los *rigs* se conectan usando un sistema de jerarquía para crear las articulaciones del objeto. La jerarquía establece una relación de padre/hijo entre las articulaciones. Un objeto hijo puede moverse, rotar y escalar independientemente del objeto padre, pero cuando el objeto padre se mueve, el elemento hijo lo debe seguir. Esta técnica de agrupación se conoce como *parenting*. Es posible agrupar múltiples objetos hijo a un mismo objeto padre, a esto se le conoce como una relación de hermanos. Incluso es posible que un objeto hijo tenga en un nivel más pequeño dentro de su jerarquía otros objetos hijo, como se puede observar en la Figura 14. En la mayoría de herramientas de animación digital, es posible modificar las relaciones entre las articulaciones en cualquier momento [10].

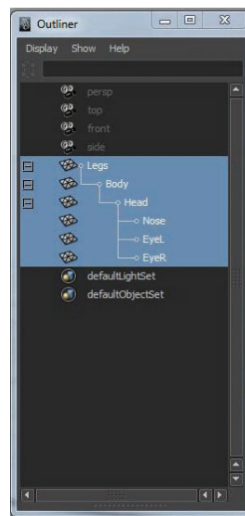


Figura 14: Una relación padre/hijo en Maya [10]

6.3.3. Cinemática directa e inversa

El movimiento de las juntas puede describirse de dos formas: cinemática directa y cinemática inversa. En la cinemática directa el animador tiene control de todas las articulaciones. Para crear un movimiento, debe mover una articulación a la vez hasta llegar a la posición deseada. En la cinemática inversa, el movimiento de una junta puede desencadenar el movimiento de otras articulaciones que estén conectadas a la junta que se movió. Por ejemplo, si la muñeca de un personaje se mueve a una posición, el brazo del personaje seguirá el movimiento [10].

El animador puede cambiar entre cinemática directa e inversa dependiendo del movimiento que quiera lograr [10].

6.4. Técnicas de captura de movimiento

En el presente, se emplean cuatro técnicas principales de captura de movimiento [1]:

- Técnicas ópticas pasivas
- Técnicas ópticas activas
- Técnicas sin marcadores
- Técnicas inerciales

Las primeras dos se caracterizan por el uso de marcadores. En el caso de la técnica óptica pasiva, los marcadores son retroreflectivos, lo que quiere decir que estos reflejan la luz infrarroja emitida por las cámaras. Una vez el reflejo es detectado, el programa calcula la posición del indicador en el espacio tridimensional del que forma parte. A diferencia de este método, el óptico activo utiliza sensores que emiten su propia luz en vez de reflejarla, por lo que necesitan su propia fuente de poder. Sin embargo, el proceso de detección sigue siendo el mismo [1].

Las técnicas restantes no usan marcadores para calcular la posición de un cuerpo. Para lograr la captura de los movimientos requieren de cámaras sensibles y un software especializado. Aún que este método no requiere de una inversión inicial tan grande, es menos preciso que las alternativas ópticas. Por último, la técnica inercial se puede usar incluso sin la presencia de cámaras. Calcula la posición mediante unidades de medición inerciales (IMU). Estos sensores incorporan giroscopios, magnetómetros y acelerómetros para determinar la posición de un cuerpo en movimiento [1].

6.5. Sistema de captura óptico pasivo

Un sistema de captura óptico pasivo consiste de 4 a 32 cámaras y una computadora que las controla. En la actualidad, la Universidad del Valle de Guatemala cuenta con una plataforma de captura que consta de 6 cámaras y dos trajes de captura de movimiento con un set de marcadores reflectivos [11].



Figura 15: Marcador reflectivo de OptiTrack [3]

Los marcadores pasivos se fabrican usando un material reflectivo y suelen ser de forma esférica, pero su forma y tamaño puede variar dependiendo de la aplicación. Por ejemplo,

para la captura de movimiento en los dedos de las manos o las expresiones faciales, se requiere de sensores de un menor tamaño que los que se usan en el resto del cuerpo. Los marcadores pasivos se fijan al actor mediante adhesivos de velcro que se adhieren a los trajes de captura [11].



Figura 16: Participante con traje de captura óptico posicionado en el centro del volumen de captura del salón CIT-116 en la Universidad del Valle de Guatemala [3]

Las cámaras en un sistema de captura pasivo emiten luz que es reflejada por los sensores del traje de captura, como los que se pueden observar en la Figura 15. Las cámaras capturan la luz reflejada a una velocidad entre 30 a 2000 muestras por segundo. Dos cámaras deben de poder ver un marcador en todo momento para calcular su posición en el espacio tridimensional. Cuando un marcador no está a la vista de alguna de las cámaras del sistema se pierde su información de posición. Existen técnicas de edición que permiten al usuario completar la información faltante, sin embargo, cuando el marcador permanece fuera del campo de visión de las cámaras durante un período prolongado, no es posible corregir la pérdida de datos. La información obtenida de un sistema óptico pasivo es muy precisa cuando no se sufre de dichos problemas de oclusión [11].

Para generar la información de movimiento, las cámaras capturan una secuencia de imágenes en escala de grises de las luces reflejadas por los marcadores. Luego, se aplican métodos de procesamiento de imágenes para reducir el ruido y separar los marcadores del fondo. Los datos de cámara en 2D se obtienen determinando las coordenadas 2D del centro de cada marcador con respecto a la vista de cada cámara que lo pueda visualizar. Para determinar el centro de cada marcador, se realiza un cálculo de centroides o una técnica de procesamiento de imágenes llamada *cicle fitting*. A partir de los datos en 2D de la cámara, las coordenadas de cada marcador obtenidas por las cámaras se triangulan para obtener las coordenadas en 3D [11].

6.6. Preparación para la captura

La preproducción es una parte muy importante del proceso de producción para obtener animaciones 3D por medio de captura de movimiento. Si no se desarrolla un plan adecuado previo a la captura de datos, puede que un pequeño descuido en el volumen de captura resulte en datos erróneos y ruido que afecten en la obtención precisa de los datos de los sensores ópticos. Por lo que es indispensable que se planifique detalladamente el volumen de captura, la posición de las cámaras y el posicionamiento de los marcadores [11].

6.6.1. Volumen de captura

El volumen de captura es el espacio tridimensional que el sistema de captura de movimiento puede abarcar. Si solo se cuenta con un total de 6 a 8 cámaras es recomendado que el espacio sea lo más cercano posible a 10 pies por 10 pies. Es necesario verificar la posición de las cámaras del volumen de captura antes de cada grabación para una obtención precisa de la información de posición. En la Figura 17, se puede observar un volumen de captura simple usando 8 cámaras [11].

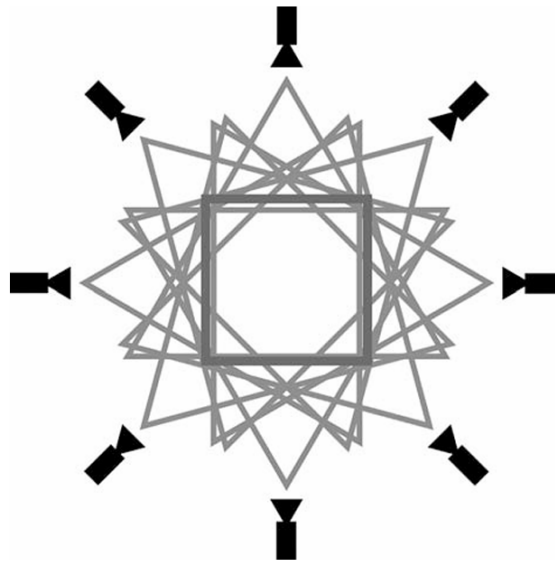


Figura 17: Volumen de captura con 8 cámaras apuntando al centro [11]

Si se cuenta con más cámaras, es posible agrupar y enfocar las cámaras en cuadrantes en vez de apuntarlas todas hacia el centro. Si se usa esta configuración de zonas se debe tomar en cuenta que si las cámaras no están bien posicionadas puede que haya una pérdida de datos en el centro del espacio de captura dependiendo del tamaño del volumen y del número de cámaras que se estén usando. En ciertas configuraciones es posible que sólo tres o cuatro cámaras puedan ver el centro del espacio. Al usar la configuración de zonas se debe de consultar a las directrices del fabricante para el posicionamiento adecuado de las cámaras [11]. Se puede observar un ejemplo en la Figura 18.

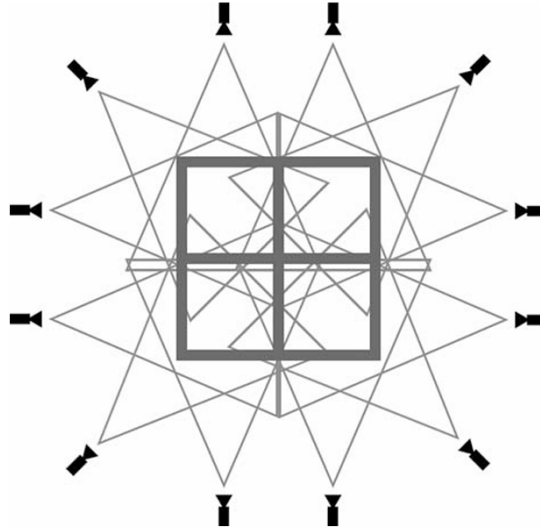


Figura 18: Volumen de captura por cuadrantes[11]

6.7. Cámaras Prime x41 de OptiTrack

El sistema de OptiTrack con el que cuenta el Departamento de Ingeniería Electrónica, Mecatrónica y Biomédica de la Universidad del Valle de Guatemala, está compuesto por seis cámaras Prime x41 de OptiTrack [3].



Figura 19: Cámara Prime x41 de OptiTrack [3]

Las especificaciones técnicas se encuentran a continuación [3]:

- Resolución: 2048x2048.
- Velocidad de fotogramas: 180 Hz.

- Latencia: 5.5 ms.
- Precisión 3D: +/- 0.10 mm.
- Rango para marcadores pasivos: 30 m.
- Rango para marcadores activos: 45 m.
- Anillo de LED:
 - 20 LED.
 - Luces del tipo infrarrojo de 850 nm.
- Tamaño: 12.6cm x 12.6 cm x 13.2 cm.
- Peso: 1.36 Kg.

En la actualidad, las cámaras están colocadas en sus respectivos trípodes que pueden alcanzar hasta una altura de 3 metros. Se inclinó cada estructura a 30° de forma que las cámaras apuntaran ligeramente hacia el suelo. Se optó por una colocación rectangular de las cámaras para que cubrieran toda el área de la plataforma de grabación.



Figura 20: Vista frontal de la colocación de las cámaras del Robotat [3]

6.8. Configuración de marcadores

Los marcadores pasivos consisten en una base con una superficie de Velcro en la parte posterior y un sensor pasivo esférico en la parte superior, como el que se puede observar en la Figura 21. Una configuración de marcadores es un set definido de varios sensores que en conjunto van a capturar un movimiento.



Figura 21: Sensor óptico pasivo de *OptiTrack* [3]

Para que una captura sea precisa y exacta todos los marcadores se deben colocar en su posición específica (dependiendo de la configuración) y deben permanecer fijos en todo momento. En el sistema de captura de OptiTrack, si en algún momento se mueve un marcador este va a dejar de contribuir al seguimiento del objeto. Además, las superficies de los marcadores deben de permanecer limpias y en buen estado para reflejar en su mayor capacidad la luz infrarroja emitida por las cámaras. Si se retira o se mueve algún marcador, se desmarcará del conjunto de marcadores y dejará de contribuir al seguimiento del objeto fijado [3].

El tamaño de los marcadores afecta su visibilidad. Los marcadores con mayor tamaño son más fáciles de detectar a mayores distancias, sin embargo, no son óptimos para el seguimiento de movimientos finos con detalles minuciosos. Se da el caso opuesto usando marcadores pequeños [3].

La universidad cuenta con dos trajes de captura de movimiento uno de talla L y otro de talla S. Para colocar los marcadores en el traje, es necesario que el participante adopte una pose de calibración conocida como T-Pose. De esa forma se podrán colocar los marcadores en sus respectivas posiciones sin mayor problema.



Figura 22: Traje de captura de OptiTrack [3]

6.9. *Retargeting*

Al aplicar los movimientos capturados a un personaje con *rig*, es común que el esqueleto fuente (esqueleto obtenido del sistema de captura) y el esqueleto objetivo tengan proporciones distintas. Para que el movimiento se adapte al esqueleto objetivo, es esencial ajustar las diferencias de escala y proporciones entre ambos esqueletos [11].

El *retargeting* o redireccionamiento de la animación implica mucho más que simplemente transferir las rotaciones y traslaciones de un esqueleto a otro. Este proceso requiere ajustar cuidadosamente las diferencias proporcionales entre los esqueletos, manteniendo la integridad del movimiento original [11].

Los programas de animación digital incluyen sus propias herramientas para el redireccionamiento de la animación del esqueleto obtenido por captura de movimiento al esqueleto del personaje que se desea animar.

Preparación del volumen de captura

Antes de realizar cualquier captura de movimiento, con el fin de obtener los mejores resultados, se debe preparar adecuadamente el área de captura. Por ello, este capítulo busca describir detalladamente el proceso que se llevó a cabo para adecuar el espacio y las cámaras con las que cuenta actualmente el campus de la Universidad del Valle de Guatemala en el salón CIT-116 para obtener la mejor área de captura. Es importante recordar que la plataforma no se usa únicamente para la captura en humanos; fue diseñada inicialmente con el propósito de crear un ecosistema para la investigación e implementación de procesos robóticos. Por lo tanto, al configurar y calibrar las cámaras infrarrojas se debe tomar en cuenta que la captura debe funcionar para actores humanos, al igual que para agentes robóticos.

7.1. Configuración de las cámaras

Cuando se instalaron las cámaras en el 2022, se posicionaron de tal modo que los seis trípodes en los que se colocaron estuvieran a la misma altura. De igual forma, la inclinación de todas las cámaras se configuró a 30° con respecto a un eje horizontal paralelo a la plataforma, con el fin de que enfocaran hacia el centro de la plataforma de captura. Probó ser una buena configuración para los agentes robóticos, sin embargo, la altura de las cámaras no era suficiente para grabar la parte superior del cuerpo humano.

Por lo tanto, se optó por modificar la altura de los trípodes y el punto al que enfoca cada una de las cámaras. En lugar de que todas apuntaran al centro de la plataforma, se decidió que con el número limitado de cámaras con las que cuenta la universidad, sería mejor cambiar la configuración de estas a un volumen de captura por cuadrantes. Con el objetivo de cubrir más zonas, se redujo la redundancia entre los campos de visión de cada cámara, siempre tomando en cuenta que para capturar el movimiento de los marcadores estos deben

de poder verse en todo momento por dos o más cámaras. En la Figura 23 se puede observar la configuración original de las cámaras con el punto de enfoque señalado en rojo.

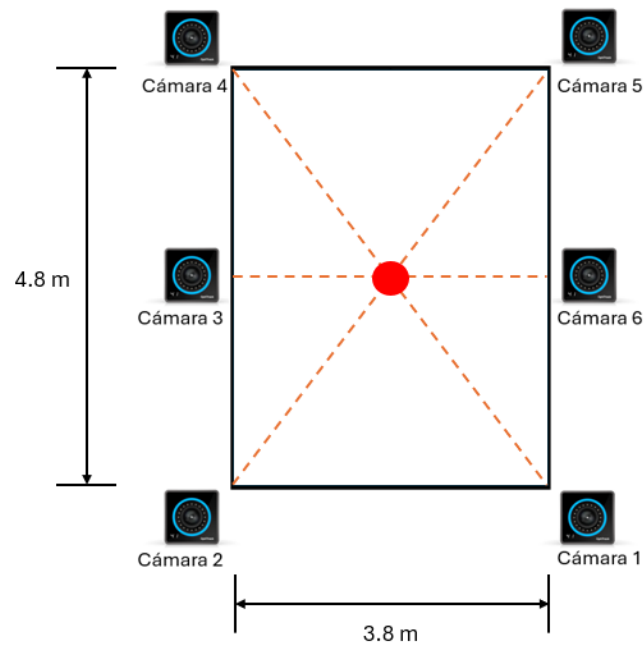


Figura 23: Diagrama de la configuración de las cámaras en el volumen de captura antes de los cambios

7.1.1. Enfoque

Se verificó que las cámaras pudieran visualizar correctamente los marcadores. Para ello, se colocaron dos marcadores de diferentes tamaños en el centro de la plataforma. Luego, se configuraron las cámaras para mostrar la vista en escala de grises y utilizando el *zoom*, se acercó el lente a los marcadores para comprobar si la imagen estaba desenfocada. Si la imagen estaba borrosa, se ajustó el enfoque hasta que coincidiera con la imagen de referencia del manual de OptiTrack 24. Era importante que ambos marcadores fueran claramente visibles, ya que algunas aplicaciones requieren del uso de marcadores más pequeños que los utilizados en el traje de captura de movimiento.

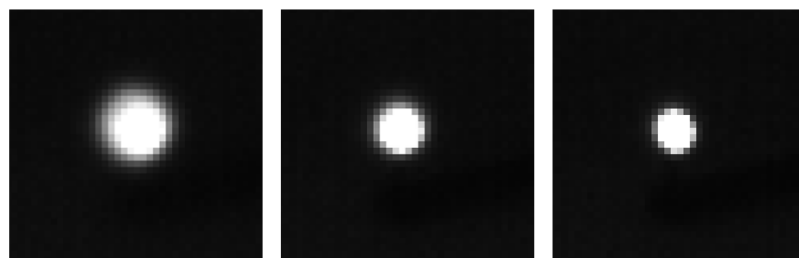


Figura 24: Demostración de un marcador desenfocado, moderadamente enfocado y correctamente enfocado, en ese orden [3]

7.1.2. Posicionamiento de las cámaras

Se estableció una nueva posición para cada trípode con el fin de orientar las cámaras hacia los puntos deseados en la configuración por cuadrantes que se explicará más adelante. Las cámaras no se movieron de sus ubicaciones definidas en la configuración anterior: se mantienen cuatro cámaras en las esquinas y dos en lados opuestos del centro de la plataforma, como se muestra en la Figura 23. Únicamente se ajustó ligeramente la posición de cada trípode. Por si se mueven las cámaras, se marcó la posición en la que debe ir cada pata del trípode, utilizando calcomanías de colores. La calcomanía de cada pata debe coincidir con la calcomanía del mismo color que se colocó en el piso del laboratorio.



Figura 25: Patas del trípode de una de las cámaras, mostrando sus calcomanías correspondientes para determinar la posición exacta en la que se debe colocar

7.1.3. Altura de las cámaras

Al tener las cámaras en sus posiciones establecidas, el siguiente paso fue ajustar la altura de los trípodes. El objetivo era permitir que las cámaras identificaran los marcadores en la parte superior del cuerpo humano, que no eran visibles en la configuración anterior, así como los marcadores de los agentes robóticos y de los pies de los actores humanos. Para ello, se colocaron seis robots Pololu 3pi, cada uno con un cuerpo rígido compuesto por tres marcadores reflectivos, debajo de cada cámara. Además, un participante humano colocó un marcador en el punto más alto de su cabeza y se situó en el centro de la plataforma. Cuando ambos tipos de marcadores, los de los robots y los del participante, fueron identificados por las cámaras, se determinó que la altura de 2.6 m era adecuada para las cámaras posicionadas en las esquinas. Las cámaras centrales se instalaron a una mayor altura de 3 m, ya que se buscaba que cubrieran la mayor parte de su respectiva mitad. En la Figura 26 se puede observar el marcador colocado sobre la cabeza del participante, así como los agentes robóticos situados debajo de cada cámara, ambos visibles para la cámara.

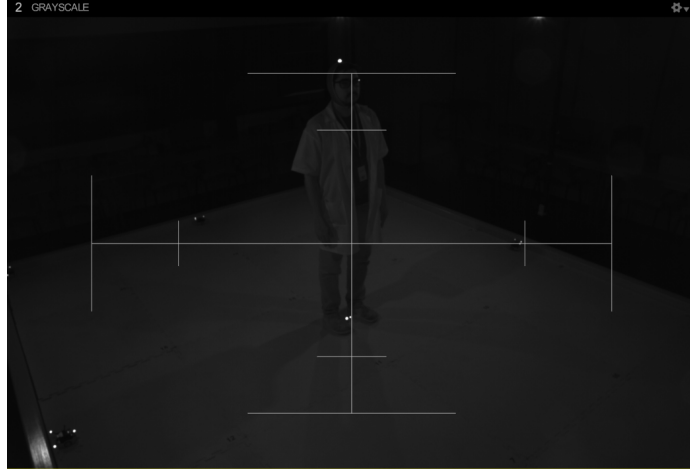


Figura 26: Visualización en escala de grises de la cámara 2 con los Pololu 3pi debajo de cada cámara y el participante humano en el centro de la plataforma



Figura 27: Diferencia de altura entre cámaras centrales (3 m) y laterales (2.6 m)

7.1.4. Volumen de captura por cuadrantes

Después de enfocar todas las cámaras y ajustar sus alturas, se orientó cada una hacia distintos puntos de la plataforma, en lugar de todas al centro como en la configuración anterior. Por ello, se dividió la plataforma en cuadrantes, de modo que cada cámara ubicada en las esquinas apuntara a un cuadrante distinto de la plataforma; mientras que las cámaras centrales, situadas a mayor altura, cubrieran la mayor parte de la mitad de la plataforma hacia la que están orientadas.

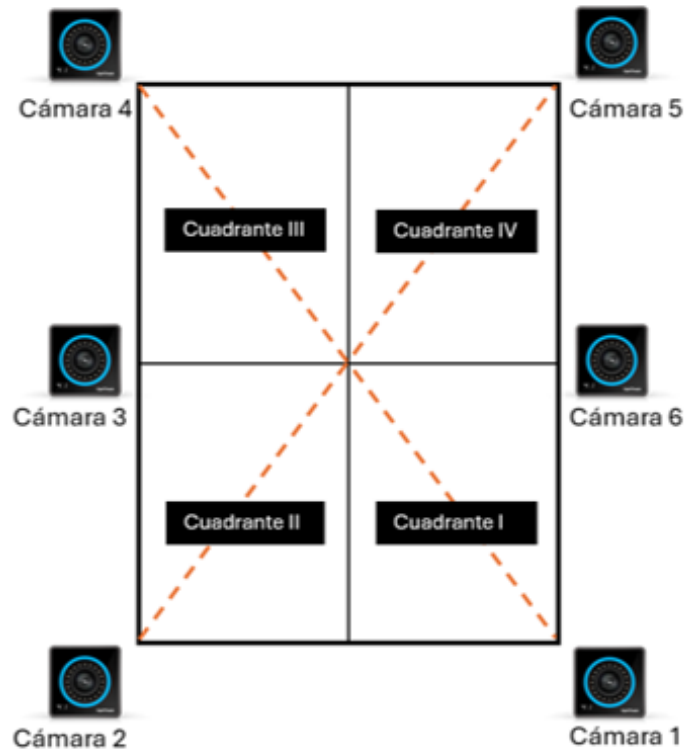


Figura 28: Diagrama de la plataforma de captura separada en cuadrantes

Se utilizó la herramienta Aim Assist de Motive para alinear las cámaras con los puntos deseados. Como referencia, se colocó un marcador en el centro de la plataforma. Luego, utilizando el centro del Aim Assist, se ajustó angularmente cada cámara hasta que apuntara al punto medio de su respectivo cuadrante, como se muestra en la Figura 29. En la Figura 30 se puede observar cómo el centro del Aim Assist se desplazó ligeramente del centro de la plataforma para generar un pequeño sesgo en la orientación de cada cámara. Este ajuste permitió que las cámaras tuvieran un enfoque más específico hacia sus respectivos cuadrantes, mejorando la cobertura de la plataforma.

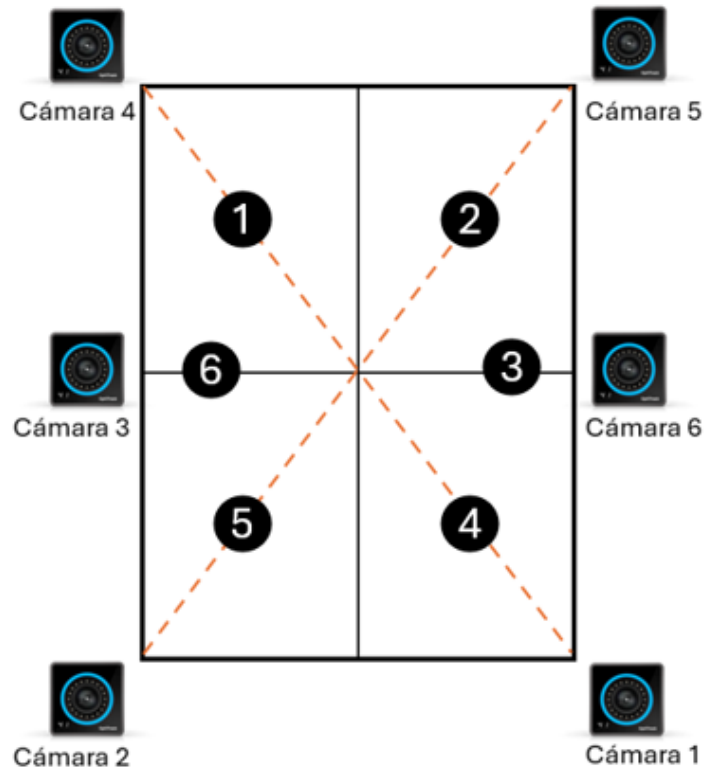


Figura 29: Vista en planta de la posición de cada cámara y el punto en el que se enfocó cada una

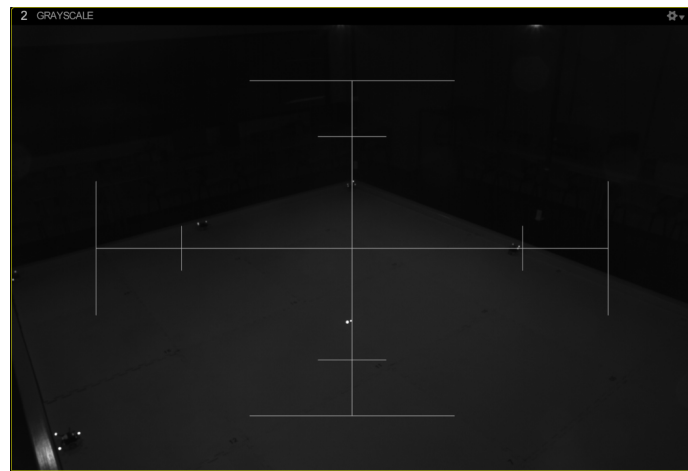


Figura 30: Visualización en escala de grises de la vista de la cámara 2 con los Pololu 3pi debajo de cada cámara

7.2. Proceso de calibración

Después de que las cámaras han sido posicionadas y enfocadas, se deben de calibrar para que el sistema pueda calcular la posición y orientación de cada marcador, al igual que recrear una representación digital del espacio de captura. Si en algún momento se cambia la configuración de las cámaras, es necesario volver a hacer el proceso de calibración. De igual modo, cambios en el ambiente como fluctuaciones en la temperatura pueden reducir la precisión del conjunto de cámaras. Por lo tanto, se recomienda hacer la calibración periódicamente o incluso antes de cada sesión de captura.

7.2.1. *Masking*

Previo a la calibración, se cubrieron todos los marcadores o reflejos identificados por las cámaras. En el Camera Viewport de Motive, se localizaron los reflejos visibles como puntos blancos en la vista de la cámara y se procedió a cubrirlos. Además, se ajustó el valor del *threshold* de las cámaras, el cual es un umbral que determina qué reflejos la cámara considera o descarta como marcadores.

Cuando ya no fue posible cubrir los reflejos o filtrarlos con el *threshold* se usó la herramienta de *masking*. Motive permite enmascarar los reflejos visibles mediante la función *Mask* en el Calibration Pane. Con la función, se aplicaron máscaras rojas sobre los reflejos detectados y se filtraron de los datos de captura. Es importante recordar que se deben borrar las máscaras de la calibración previa antes de aplicar una nueva para evitar la pérdida innecesaria de información valiosa [3].

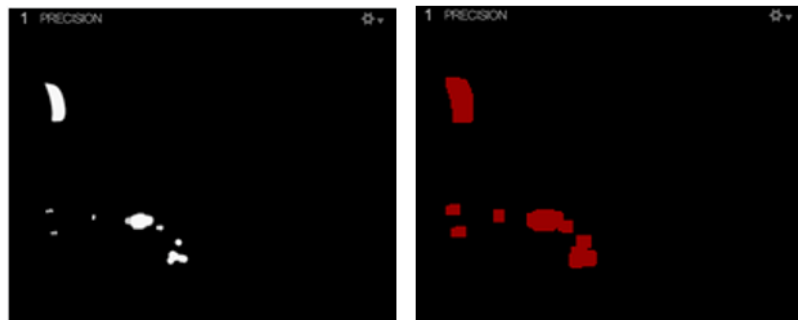


Figura 31: Reflejos no deseados antes y después del proceso de enmascarado [3].

7.2.2. *Wanding*

Una vez el volumen de captura esté libre de reflejos no deseados, se puede comenzar el proceso de *wanding*. Este consiste en usar una varilla de calibración que en su extremo cuenta con tres sensores reflectivos separados por una distancia conocida para calibrar el volumen de captura. En el campus se usa la varilla CW-250 que tiene un ancho de 250 mm y está diseñada específicamente para la calibración de volúmenes de captura pequeños o medianos. La varilla debe moverse de una posición a otra, cubriendo la mayor parte del

volumen de captura para asegurar que las cámaras registren los marcadores de calibración desde diferentes ángulos. Para que los resultados sean óptimos es necesario que por lo menos dos cámaras vean los tres marcadores de la varilla al mismo tiempo. Además, si durante el proceso se detecta algún reflejo que no sea perteneciente a la varilla no se registrará la muestra. Por esta razón, se debe de evitar el uso de ropa o cualquier artículo reflectivo que puede afectar el proceso. De igual manera, es importante que no estén dañados los sensores de la varilla o el sistema no podrá recopilar las muestras necesarias.



Figura 32: Ejemplo de *wanding* en el volumen de captura.

Los pasos para calibrar el sistema son los siguientes [3]:

1. Asegurar que no haya reflejos no deseados en el área de captura. Si los hay, se debe usar la herramienta de *mask* para filtrarlos.
2. Garantizar que las cámaras no estén seleccionadas.
3. Seleccionar la opción de *full calibration* en el panel de calibración.
4. Seleccionar el tipo de varilla, para el caso de la universidad sería la varilla CW-250.
5. Presionar *start wanding* para empezar el proceso.
6. Mover gentilmente la varilla de una posición a otra buscando cubrir la mayor parte del volumen. Es necesario cubrir alturas pequeñas y grandes.
7. En la ventana Camera Viewport se debe de poder apreciar el trayecto de la varilla que ha visualizado cada cámara. Es importante que la vista de cada cámara se cubra lo más posible con muestras.
8. Presionar *start calculating* en el panel de calibración para obtener el ambiente digital esperado.

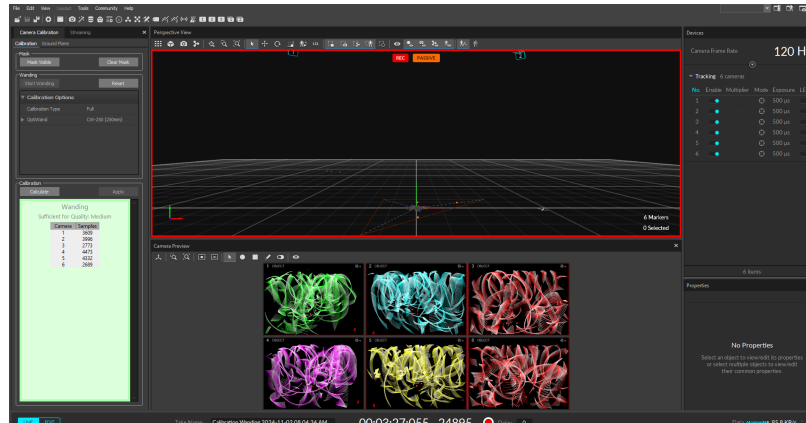


Figura 33: Trayectorias de la varilla en el proceso de *wanding* en el Camera Viewport

7.2.3. Establecer el origen

Al terminar la calibración se colocó una escuadra de calibración en el centro de la plataforma para establecerlo como el origen. El brazo largo de la escuadra se posicionó viendo hacia lo largo de la plataforma para establecerlo como el eje y. El brazo corto se usó para determinar la dirección del eje x, como se puede observar en la Figura 34 Después, usando la opción de *set ground plane* en el panel de calibración se definió el origen. Por lo tanto, el eje z es perpendicular al plano xy que se determinó con la escuadra. Por último, se guardó la configuración de calibración en un archivo .CAL. Al contar con la opción de guardar la calibración en un archivo, es posible rehusarla siempre y cuando la configuración de cámaras no haya cambiado o variables externas no ameriten volver a hacer la calibración.

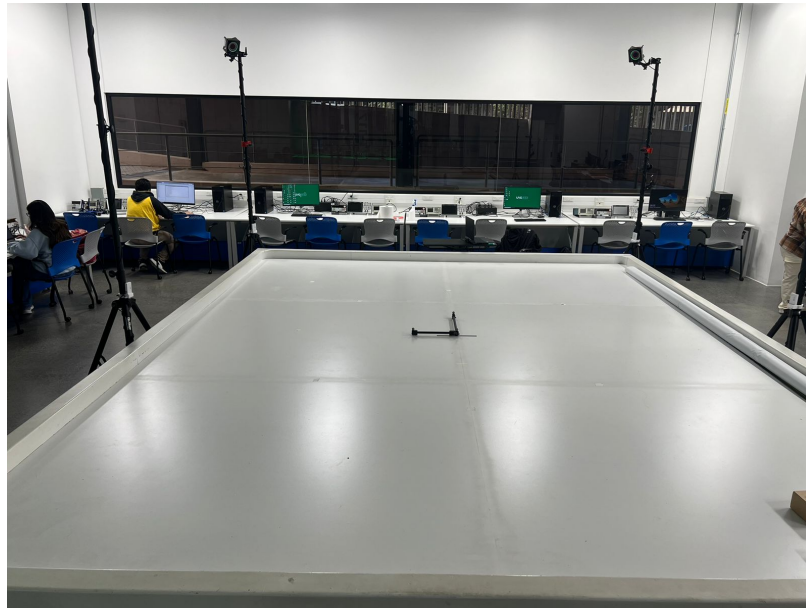


Figura 34: Escuadra usada para definir el origen en el centro de la plataforma

Configuración y exportación de esqueleto

Al contar con un volumen de captura con cámaras calibradas y enfocadas, sin reflejos que interfieran en la captura y con una cobertura total del área, se garantiza un entorno adecuado para la detección precisa de los marcadores. Sin embargo, esto no es suficiente para asegurar una captura de movimiento óptima en humanos. Este capítulo tiene como objetivo describir los pasos necesarios para lograr una captura precisa de los movimientos de un actor humano, al igual que detallar las herramientas que ofrece Motive para facilitar el proceso. Además, se detallará el proceso de exportación del archivo .TAK, que contiene la información de movimiento de cada toma para que pueda ser utilizado en otros programas de animación digital.

A continuación, se detalla el proceso que se siguió para grabar una toma de captura de movimiento utilizando la plataforma del Robotat, el programa Motive y los trajes de captura de OptiTrack. Para ello, fue necesario contar con un participante que usara el traje de captura, al cual se le configuraron los marcadores pasivos de forma que se asegurara una detección precisa de sus movimientos.

8.1. Configuración de marcadores

Para capturar el movimiento de humanos, una buena colocación de los marcadores es fundamental. Estos deben permanecer fijos durante toda la grabación y posicionarse en puntos específicos del cuerpo humano para obtener una representación precisa y fiel del movimiento. Una colocación incorrecta o el desplazamiento de los marcadores durante la captura pueden generar distorsiones en los datos, afectando la calidad del seguimiento esquelético y el resultado de la animación final.

8.1.1. Selección de esqueleto y preparación de marcadores

Motive ofrece una selección de configuraciones de marcadores que se pueden elegir según el tipo de movimiento que se desea capturar. Estas configuraciones incluyen guías detalladas sobre la posición exacta en la que debe colocarse cada marcador y el número total de marcadores necesarios para generar el esqueleto. Para crear una configuración se deben de seguir los pasos descritos a continuación:

1. En Motive, abrir la ventana Builder Pane y seleccionar la pestaña Create.
2. En las opciones disponibles para *type* se debe de seleccionar *skeleton*.
3. Seleccionar la configuración de marcadores que se desea replicar.
4. Después de seleccionar una configuración en el *Builder Pane*, aparecerá un avatar con las posiciones exactas del cuerpo donde deben colocarse los marcadores. Usando el *click* derecho del *mouse*, es posible rotar el avatar para obtener una mejor visualización de la ubicación de los marcadores desde diferentes ángulos.

Para las pruebas, se seleccionó una configuración de cuerpo completo que utiliza un total de 37 marcadores, como se muestra en la Figura 35. El traje de captura de movimiento utilizado incluye un sombrero, una chaqueta, un pantalón y un par de guantes. Debido a la falta de calzado con velcro, se usaron calcetines con adhesivos de velcro para colocar los marcadores en los pies del participante. Aunque estos resultaron adecuados para las pruebas, no son ideales para lograr una captura de movimiento con mayor precisión.

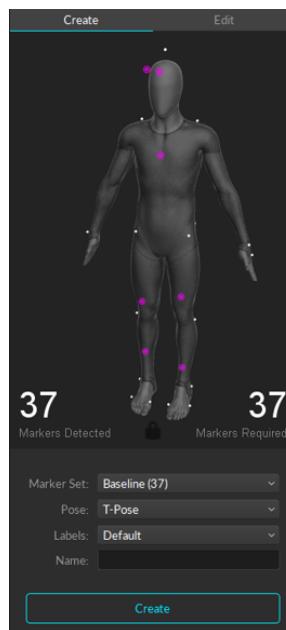


Figura 35: Configuración de marcadores para esqueleto de cuerpo completo con 37 marcadores



Figura 36: Traje de captura usado para las pruebas, incluye un sombrero, una chaqueta, un pantalón y un par de guantes. La fotografía fue tomada después de las pruebas, los marcadores siempre se deben colocar cuando el participante ya tiene puesto el traje

8.1.2. Colocación de marcadores

Al colocar los marcadores en el participante humano, es fundamental que se posicionen de la manera más similar posible a la visualización del avatar en el Builder Pane. Los marcadores de color blanco, mostrados en la Figura 35 tienen una posición fija, lo que significa que deben ubicarse en la misma posición para cada esqueleto creado. Estos son cruciales para el proceso de auto-etiquetado de los marcadores del esqueleto. En cambio, la posición de los marcadores representados en color magenta es relativa, lo que permite que su ubicación varíe según las proporciones del actor, generando así un esqueleto único adaptado a cada participante.

Durante la colocación de los marcadores, el participante adoptó una pose de calibración, que puede ser la *A-pose* o la *T-pose*, tal como se muestra en la Figura 37. Para posicionar correctamente los marcadores en las articulaciones, el participante tuvo que extender y flexionar cada articulación, lo que permitió identificar el lugar adecuado para colocar cada marcador. Además, se tuvo que identificar el eje en el que se mueve la articulación para asegurar que los marcadores se coloquen a lo largo de ese eje, donde el desplazamiento de la piel es mínimo durante el movimiento. Los marcadores identificados con color magenta, conocidos como marcadores de segmento, no se colocaron alrededor de una articulación. Estos se posicionaron de manera asimétrica en su respectivo segmento del cuerpo. Es necesario que la colocación sea asimétrica para que al momento de resolver el esqueleto se pueda diferenciar fácilmente la izquierda de la derecha. Se puede observar un ejemplo de la asimetría descrita en la Figura 38.



Figura 37: Ejemplo de participante con traje de OptiTrack adoptando una pose de calibración *T-Pose* [3]

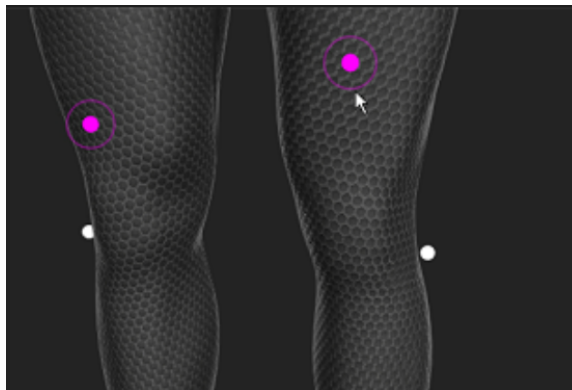


Figura 38: Marcadores de segmento asimétricos para diferenciar la pierna derecha de la izquierda [3]

Consejos adicionales para la colocación de los marcadores [3]:

- Si los marcadores se van a colocar directamente sobre la piel del participante, es importante limpiarla previamente para asegurarse que esté libre de humedad, ya que podría causar que los marcadores se desplacen de su posición durante la captura.
- No usar ropa o zapatos con material reflectivo.
- Recogerse el pelo que pueda ocultar los marcadores.
- Remover cualquier tipo de joyería.
- En el Builder Pane, el número de marcadores necesarios y el número de marcadores detectados debe coincidir. Si los marcadores del esqueleto no se detectan automáticamente, se deben seleccionar manualmente desde la ventana 3D Perspective View.

8.1.3. Resultado de la configuración del esqueleto

Una vez el participante tenía puesto el traje con los marcadores correspondientes, se posicionó en el centro de la plataforma y adoptó nuevamente la pose de calibración *T-pose*, como se puede observar en la Figura 39. Al estar en la plataforma y con los 37 marcadores visibles, se seleccionaron todos los marcadores que conforman al esqueleto y se habilitó el botón *create* en el Builder Pane, permitiendo la creación del esqueleto 40.



Figura 39: Participante en *T-Pose* para la calibración del esqueleto

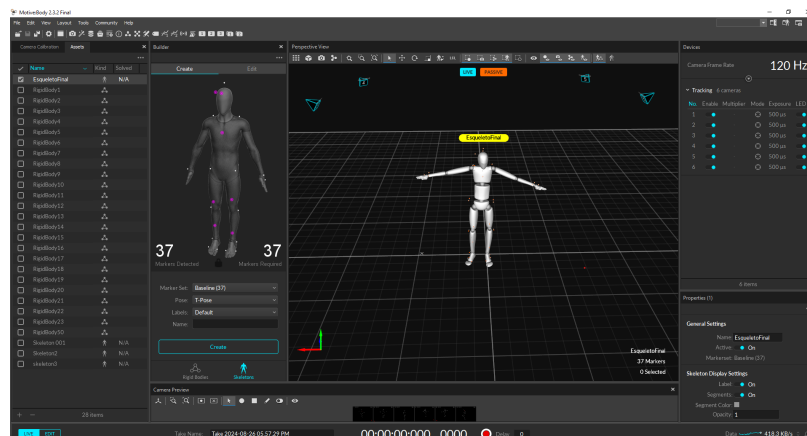


Figura 40: Resultado de la configuración del esqueleto en Motive

8.2. Grabación y exportación de información de movimiento

A continuación se presentará como se guardó una grabación de los movimientos capturados en un archivo con extensión .TAK desde Motive. Al igual que como exportar los datos en otros formatos para su uso en otros programas de animación digital.

8.2.1. Carpeta de sesión

Previo a grabar los movimientos, es buena práctica mantener un orden adecuado en los archivos. Cada grabación se guarda en un archivo con extensión .TAK, todas las grabaciones de una misma sesión deben almacenarse en el mismo directorio. Por lo tanto, es recomendable iniciar una sesión creando una nueva carpeta. Esta carpeta puede crearse en cualquier directorio deseado de la computadora y luego cargarse utilizando el botón de creación de carpeta de sesión en el *Data Pane* o simplemente arrastrándola al *Data Pane*. Si no se crea la carpeta de la sesión todas las tomas se guardarán en el directorio predeterminado (Documents/OptiTrack/Default).

En la figura a continuación se tiene un ejemplo de un proyecto ordenado por carpetas. Dentro de una carpeta principal llamada *motive takes*, se crean subcarpetas por día, y dentro de estas, carpetas específicas para cada sesión. Este sistema de organización facilita el acceso a las grabaciones y evita la pérdida de archivos generados en Motive.

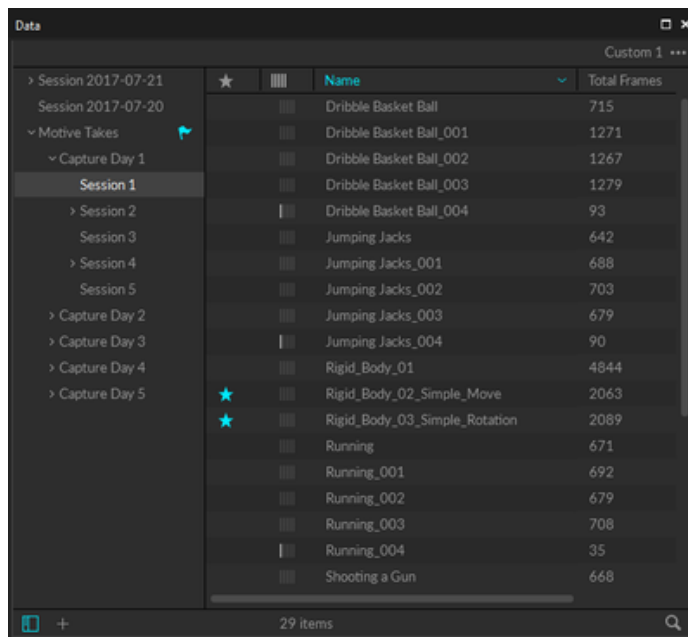


Figura 41: Carpetas de sesión ordenadas en el *Data Management Pane* [3]

8.2.2. Grabación de movimiento en Motive

Con el esqueleto definido, el volumen de captura calibrado y la carpeta de sesión configurada, se procedió a realizar la grabación de movimientos. El actor se posicionó en el centro de la plataforma en la pose de calibración *T-Pose* y en el Control Deck se presionó el botón rojo para iniciar la grabación. En la Figura 42 se puede observar el Control Deck con el botón rojo en la interfaz de Motive. Si Motive está en *live mode*, es decir cuando las cámaras están activas y el sistema está procesando datos en tiempo real, es posible iniciar la grabación presionando la tecla de espacio. El botón rojo del Control Deck se iluminó con un rojo más brillante al comenzar la grabación.

El actor realizó una serie de movimientos de prueba para verificar que el sistema funcionaba correctamente. Al finalizar, se presionó nuevamente el botón rojo del Control Deck para detener la grabación. El archivo con extensión *.TAK*, que contiene los movimientos capturados, se guardó automáticamente en la carpeta de la sesión. Este archivo puede reproducirse, editarse, reconstruirse y exportarse en distintos formatos para su uso en programas de animación digital.

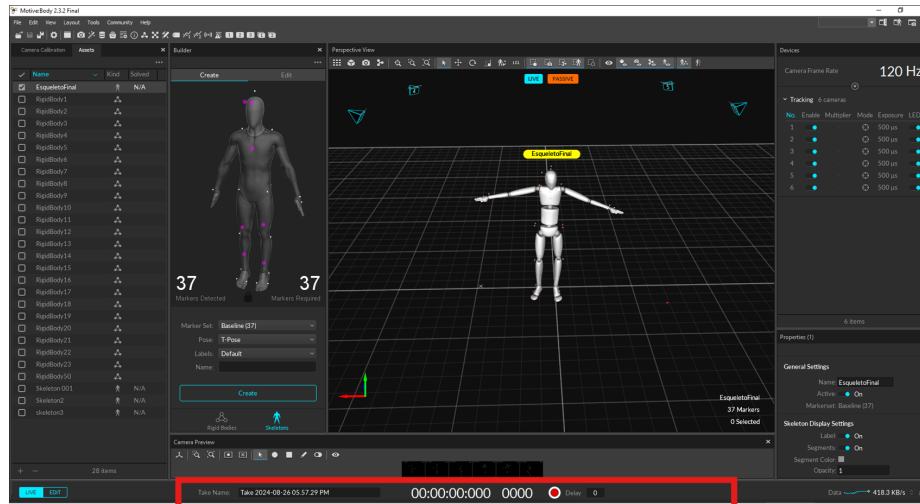


Figura 42: Posición en la pantalla de Motive del Control Deck

8.2.3. Exportación de datos

Después de grabar el movimiento y guardarlo en un archivo con extensión *.TAK*, se exportaron los resultados a un programa de animación digital. La ventana de exportación se encuentra en la pestaña *File*, donde se debe seleccionar la opción *export tracking data* para comenzar el proceso. Hay que recordar que para exportar el movimiento del esqueleto o cuerpo rígido estos deben de estar resueltos, es decir se cuenta con la información de posición y orientación de cada objeto. Si hay un objeto no resuelto en la ventana de *Assesst* este se puede deseleccionar para no incluirlo en la exportación.

Se guardaron todos los *takes* actuados como archivos con extensión *FBX*. De las opciones disponibles, se decidió exportar los movimientos en este formato, ya que permite incluir los

datos de posición de los marcadores, los datos de Cuerpo Rígido con 6 Grados de Libertad y la información del esqueleto. Además, este formato es uno de los estándares en la industria del cine y los videojuegos para la exportación de animaciones, por lo que muchos programas de animación digital, como Unity o MotionBuilder, pueden leerlo. A continuación, se presentan dos ejemplos de un archivo .FBX exportado desde Motive aplicado a distintos personajes en Unity y MotionBuilder. Es importante destacar que, para que la animación pueda utilizarse en diferentes personajes, es necesario pasar por un proceso de *rigging* y *retargeting* en su programa preferido de animación digital.

En el caso de Unreal Engine, no es posible exportar solo el archivo .FBX con la animación, ya que este programa requiere que la animación esté vinculada a un esqueleto para que pueda utilizarse. Por otro lado, en Unity y MotionBuilder es posible cargar los datos de posición y orientación del esqueleto, sin la necesidad de tener un esqueleto preexistente.

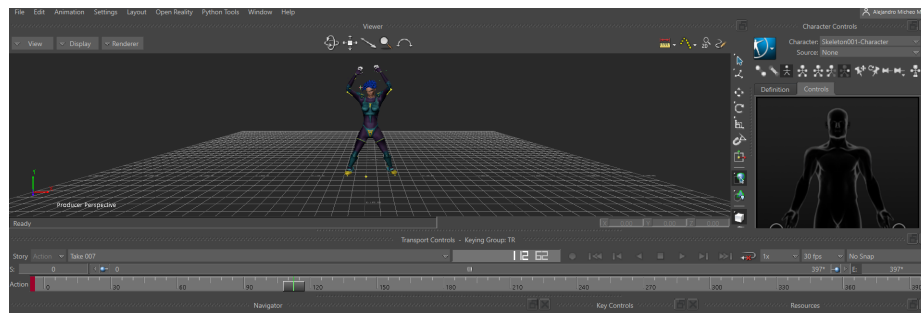


Figura 43: Animación de salto exportada de Motive a un personaje en Motion Builder

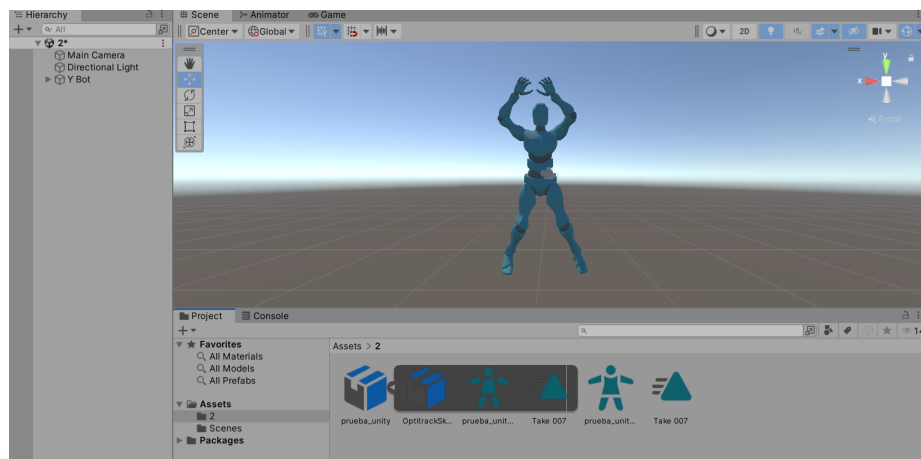


Figura 44: Animación de salto exportada de Motive a un personaje en Unity

Transmisión en tiempo real de la información de movimiento (Motive)

Además de exportar los datos mediante archivos FBX, es posible hacerlo en directo a través de una transmisión en tiempo real utilizando un protocolo de red llamado NatNet. Motive ofrece una variedad de *plugins* para transmitir los datos a múltiples aplicaciones de animación digital. Este capítulo tiene como objetivo describir las configuraciones que deben seleccionarse en Motive para lograr una transferencia de datos exitosa a los programas de MotionBuilder, Unity y Unreal Engine.

9.1. Data Streaming Pane

Para habilitar la transferencia de datos hacia los programas de animación digital, primero se realizaron las configuraciones necesarias en Motive, utilizando el Data Streaming Pane para establecer la conexión. En esta sección se detallarán las configuraciones seleccionadas en el Data Streaming Pane para lograr la transmisión de información de un esqueleto a través de la red de comunicación *wifi* del ecosistema de Robotat, con la que cuenta la Universidad del Valle de Guatemala en el salón CIT-116.

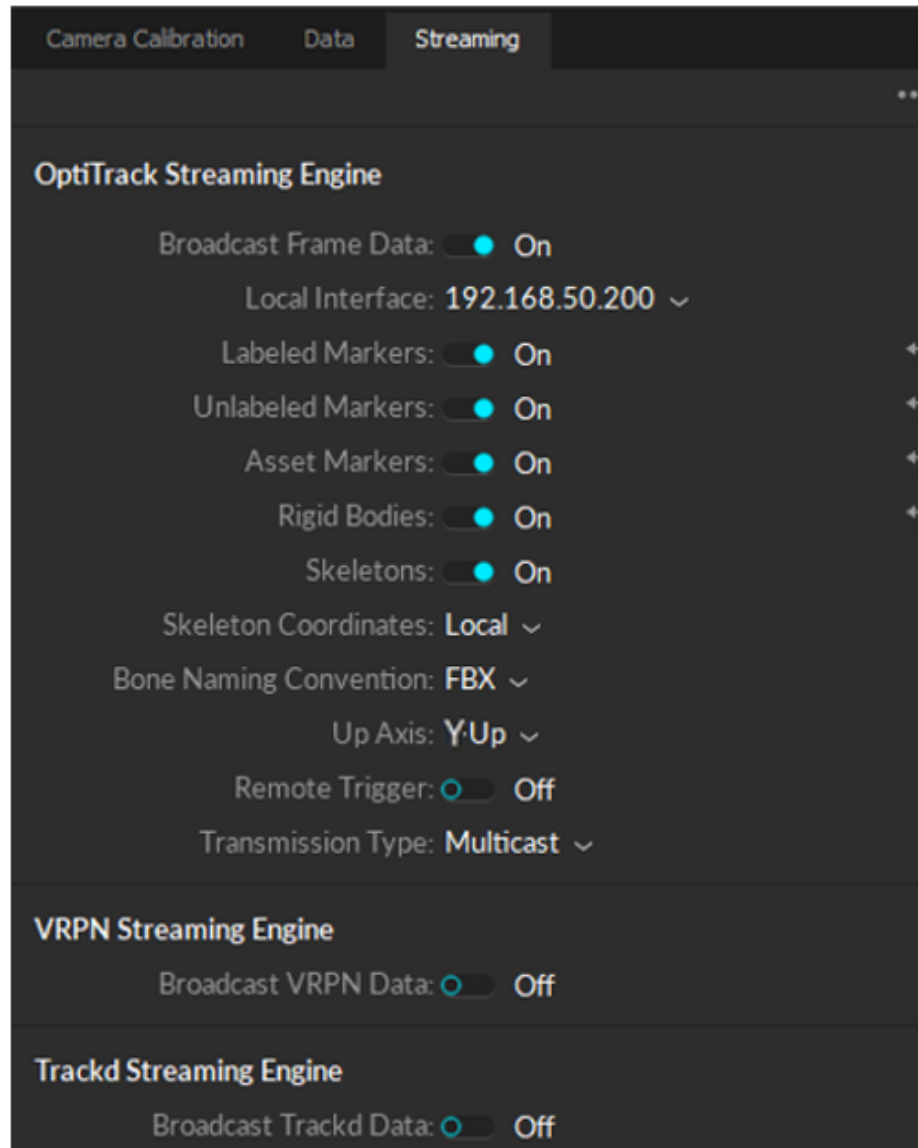


Figura 45: Configuraciones para la transmisión de datos en el Data Streaming Pane de Motive

Para activar la transmisión, se habilitó el selector Broadcast Frame Data. Luego, en el campo Local Interface, se seleccionó la dirección IP de la red de comunicación del Robotat, ya que el objetivo era transmitir la información a programas que se ejecutaron en otra computadora mediante esa red. Si se desea recibir la información en la misma computadora, se puede utilizar la opción de *loopback*.

A continuación, se encontraron una serie de selectores para elegir el tipo de información que se desea transmitir por la red: marcadores etiquetados, marcadores sin etiqueta, marcadores asociados a un *asset*, cuerpos rígidos o esqueletos. Para la transmisión de un esqueleto es necesario que todos estén activos.

Como su nombre lo indica, *skeleton coordinates* es una configuración que únicamente afecta a los esqueletos que se están transmitiendo y ofrece dos opciones: *Local* y *Global*.

Esta configuración controla cómo se procesan los datos de movimiento en los programas de animación que reciben la transmisión. La opción predeterminada es *Global*, en la cual los datos de posición y rotación de cada hueso se calculan en relación al sistema de coordenadas global, definido en la calibración del volumen de captura. Por otro lado, en la opción *Local*, los datos de posición y rotación de cada hueso son relativos a su hueso padre inmediato en la jerarquía del esqueleto. Esto resulta útil para animaciones, ya que en estas se necesita conservar las relaciones entre los huesos, independientemente de la ubicación global del esqueleto. Por lo tanto, para transmitir el esqueleto a los programas de animación digital se optó por usar el sistema de coordenadas local.

El siguiente campo a configurar fue Bone Naming Convention. Dado que los tres programas de animación a los que se transmitió la información de movimiento utilizan archivos con la extensión .FBX para leer datos de movimiento, se eligió esta opción. Esta convención asegura que los nombres de los huesos y la estructura del esqueleto sean compatibles con el formato FBX, facilitando la importación y sincronización de los movimientos. A continuación, se seleccionó el *Up Axis*, esta configuración determina qué eje será tomado como el eje vertical al momento de recibir la información en los programas de animación digital. Los tres programas de animación digital usados toman el *Eje Y* como el eje vertical. A diferencia de las aplicaciones de robótica realizadas en el mismo volumen de captura que usan el *Eje Z*.

Por último, se seleccionó el Transmission Type, que ofrece dos opciones: Multicast y Unicast. Ambas opciones pueden utilizarse para la transmisión del esqueleto en tiempo real; sin embargo, Unicast, al ser una conexión punto a punto, está diseñada para enviar los datos directamente a un receptor específico. Por lo que la hace la opción más eficiente al momento de enviar la información a un único receptor. Por el otro lado, Multicast permite que varias direcciones IP reciban los mismos datos simultáneamente, lo que la hace ideal cuando se necesita que múltiples dispositivos reciban la transmisión de manera eficiente.

Transmisión en tiempo real de la información de movimiento a los programas de animación digital

Después de realizar las configuraciones necesarias en el Data Streaming Pane de Motive para la transmisión en tiempo real del esqueleto, se descargaron los *plugins* proporcionados por OptiTrack para recibir los datos en los programas de animación digital: MotionBuilder, Unity y Unreal Engine. En este capítulo se presentarán los pasos seguidos para recibir la información del esqueleto en tiempo real en dichos programas.

Para poner a prueba el flujo de información en los tres programas, se estableció la conexión mientras un participante, equipado con el traje de captura de movimiento, se ubicó en el centro de la plataforma del volumen de captura en el salón CIT-116 de la Universidad del Valle de Guatemala. El participante realizó una serie de movimientos de prueba para verificar que la información se estaba recibiendo en tiempo real y evaluar si se puede usar la información recibida para la animación de un personaje digital.

Cabe resaltar que, dado que no se lograba conectar correctamente los programas con Motive mediante una conexión inalámbrica a la red de comunicación del Robotat, se optó por conectar la computadora que ejecutaba los programas de animación directamente al *router* de la red mediante un cable Ethernet. Al realizar esta conexión por cable, se logró establecer la comunicación exitosamente. Además, se debe considerar que si no se establece la transferencia de datos correctamente, una posible causa puede ser la interferencia de un antivirus, que a veces bloquea o restringe el tráfico de red necesario para la comunicación entre los programas.

10.1. Transmisión en tiempo real de la información de movimiento (MotionBuilder)

En esta sección se presentarán los pasos que se siguieron para usar el Autodesk Motion Builder: OptiTrack Skeleton Plugin para transmitir un esqueleto capturado con el traje de OptiTrack al *software* de MotionBuilder.

10.1.1. Autodesk MotionBuilder: OptiTrack Skeleton Plugin

Se realizaron las configuraciones en el Data Streaming Pane de Motive y se descargó el *plugin* de MotionBuilder proporcionado por OptiTrack. El *plugin* abre un instalador, se deben de seguir las direcciones del instalador para usar el *plugin*. Dentro del Asset Browser de MotionBuilder, se navegó a la carpeta de Templates y luego a Devices, donde se seleccionó el OptiTrack Skeleton Plugin, como se puede observar en la Figura 46.

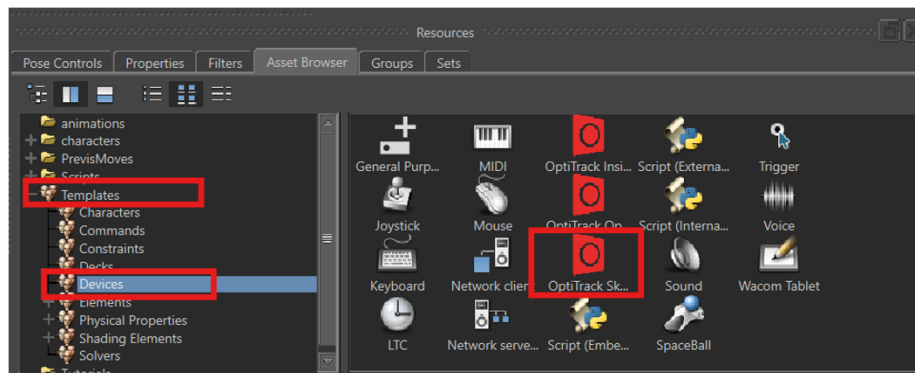


Figura 46: Asset Browser en MotionBuilder

El *plugin* desplegó en la ventana de Navigator de MotionBuilder un menú con las configuraciones necesarias. En la Figura 47 se puede observar la ventana de configuración del *plugin*. En el apartado de Local Address, se configuró la dirección IP de la computadora que ejecuta MotionBuilder. En el campo Server Address, se seleccionó la dirección IP de la red de comunicación del Robotat, que es la misma que está en el campo Local Interface en el Data Streaming Pane de Motive. Luego, en el campo Server Type, se seleccionó Unicast para optimizar la transferencia de datos hacia un único cliente. Por último, se activó la opción Auto-Characterize, lo cual permite que MotionBuilder reconozca los datos del esqueleto transmitido desde Motive como un personaje y facilita la redirección de la animación hacia otro modelo con rig [12].

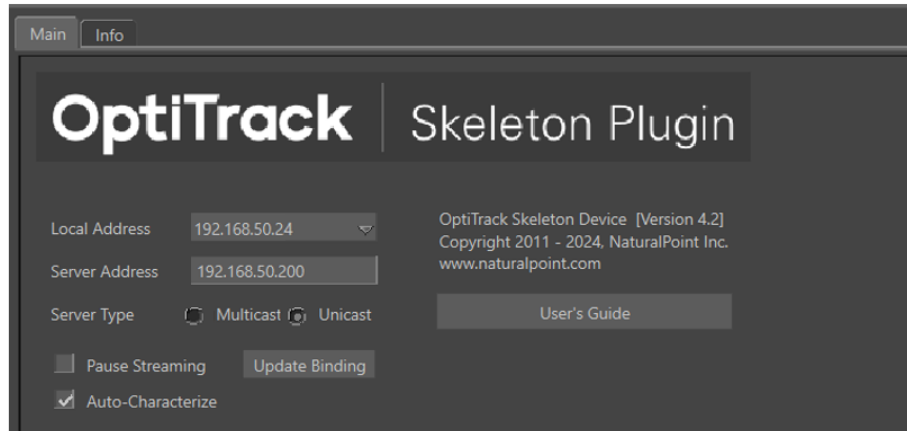


Figura 47: Ventana para configurar el Skeleton Plugin en la ventana de Navigator de MotionBuilder

Una vez configurados correctamente los datos de la red, se seleccionó el botón Online ubicado al lado izquierdo del Skeleton Plugin en la ventana de Navigator de MotionBuilder. Al establecer la conexión, el cuadro de Online cambió a un tono verde, indicando que la conexión se había establecido correctamente y que los datos se estaban transmitiendo. Como en este caso los datos de movimiento se estaban transmitiendo en vivo, con un participante equipado con el traje de captura en la plataforma, se seleccionó el encuadre de Live. Cabe destacar que, si los datos se transmiten en modo de edición desde Motive (es decir, desde una grabación de movimiento pregrabada), no se debe seleccionar el encuadre Live.

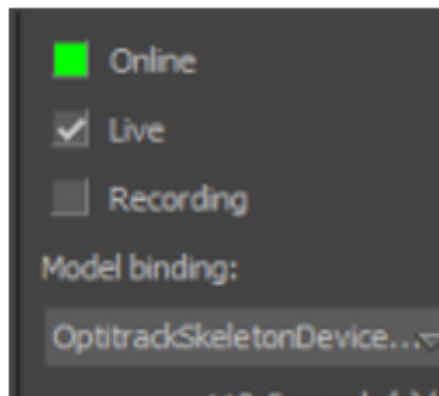


Figura 48: Transmisión en vivo de datos de Motive a MotionBuilder

Al haber realizado las configuraciones y establecido la transmisión de datos, se generó un esqueleto en el Viewer Pane de MotionBuilder que replicaba en tiempo real los movimientos que el participante realizaba en la plataforma del volumen de captura.

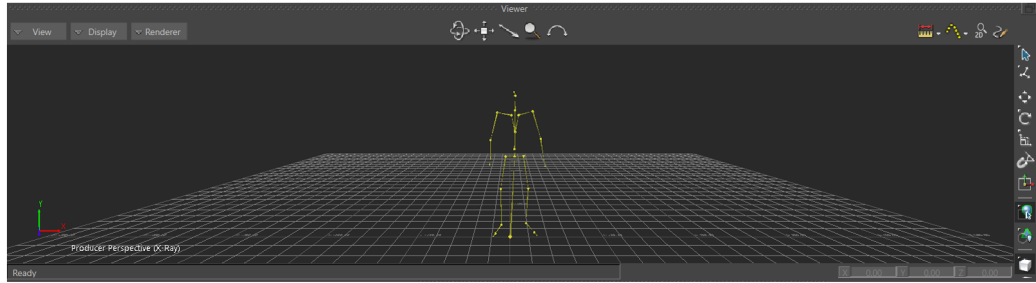


Figura 49: Esqueleto transmitido de Motive a MotionBuilder

10.1.2. Redireccionamiento de animación del esqueleto de Motive a un personaje en MotionBuilder

El esqueleto transmitido desde Motive se puede usar para animar un personaje virtual en tiempo real. Para lograrlo, se seleccionó Auto-characterize en las configuraciones del Skeleton Plugin y se arrastró a la escena el personaje deseado, asegurándose de que este contara con un *rig* para que pudiera ser animado correctamente.

El personaje utilizado fue *mia_rigged*, el cual se puede encontrar en la pestaña de Asset Browser dentro de la ventana de Resources de MotionBuilder. En la carpeta Tutorials, proporcionada por MotionBuilder, se encuentran varios modelos, entre ellos *mia_rigged*. Este personaje se arrastró a la escena en *T-Pose*, orientado hacia el eje Z positivo. También es importante que el modelo esté caracterizado. Se aseguró que lo estaba al buscar al personaje en la ventana de Navigator al desplegar el menú de Characters, como se muestra en la Figura 51.



Figura 50: Personaje *mia_rigged* viendo hacia el eje Z positivo en el Viewer Pane de MotionBuilder

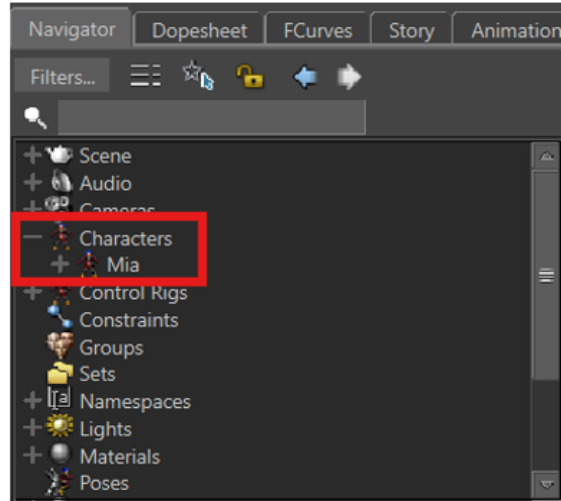


Figura 51: Modelo de mia_rigged caracterizado

Luego, se abrió la ventana de Character Controls del personaje (Window → Actor/Character Controls). En el campo Source, se seleccionó el esqueleto transmitido desde Motive, cuyo nombre coincidía con el del esqueleto en el Assets Pane de Motive. Como se puede observar en la Figura 52. Una vez anclado el esqueleto al personaje mia_rigged, este comenzó a animarse, replicando los movimientos realizados por el participante con el traje en la plataforma de captura.

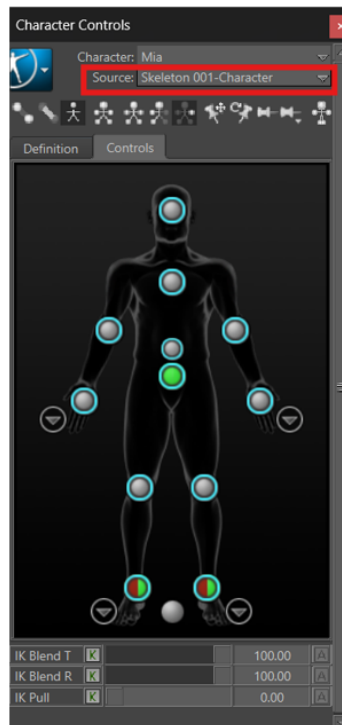


Figura 52: Ventana de Character Controls de mia_rigged con el esqueleto de OptiTrack seleccionado

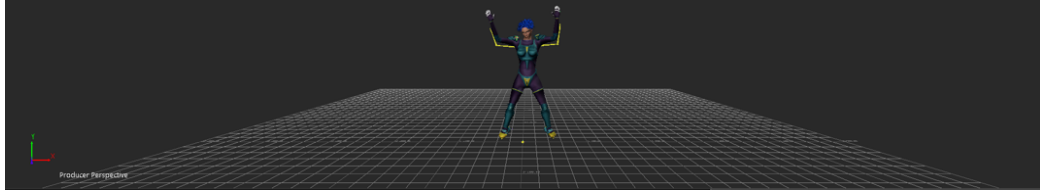


Figura 53: Personaje animado por el esqueleto transmitido desde Motive

10.2. Transmisión en tiempo real de la información de movimiento (Unity)

En esta sección se presentarán los pasos que se siguieron para usar el OptiTrack Unity Plugin para transmitir un esqueleto capturado con el traje de OptiTrack al *software* de Unity.

En Unity se creó un nuevo proyecto y se descargó el *plugin* proporcionado por OptiTrack. En este caso el *plugin* es un archivo en formato unitypackage, por lo tanto, se importó al proyecto de Unity para usar su contenido. Dentro del proyecto se generó un *folder* llamado OptiTrack Unity Plugin. Este contiene el objeto cliente para establecer la conexión con Motive, un modelo 3D para un humanoide con *rig* y los scripts de C# que se procesan en Unity para establecer la conexión.

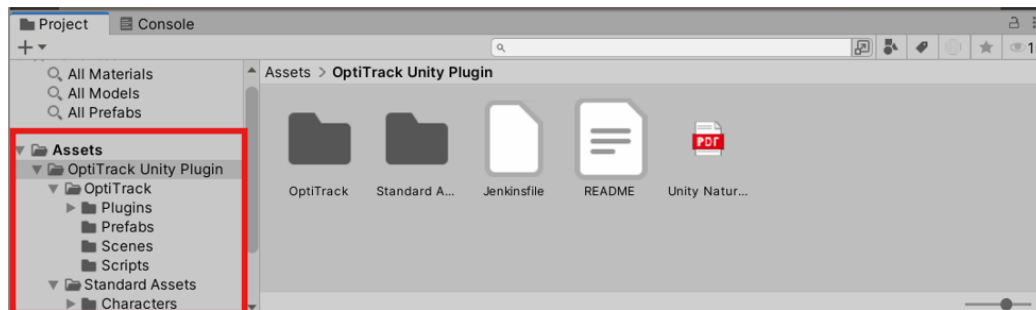


Figura 54: Optitrack Unity Plugin en el folder de Assets en la pestaña llamada Project de Unity

Para establecer la conexión con Motive, primero se utilizó el objeto cliente, Client – OptiTrack.prefab, ubicado en la carpeta Prefabs del *plugin*. Para usarlo se arrastró a la escena y se configuraron los siguientes parámetros:

En el apartado Server Address, se ingresó la dirección IP de la computadora que ejecuta Motive, la cual es la misma que se ingresó en el campo Local Interface en el Data Streaming Pane de Motive. En Local Address, se ingresó la dirección IP de la computadora que corre Unity. En el caso de los campos de Connection Type, Skeleton Coordinates y Bone Naming Convention se configuraron de acuerdo con las opciones seleccionadas en el Data Streaming Pane de Motive. Connection Type se configuró en Unicast, Skeleton Coordinates en local y Bone Naming Convention en FBX. Se configuró con los mismos parámetros que los establecidos para la transmisión del esqueleto a MotionBuilder.

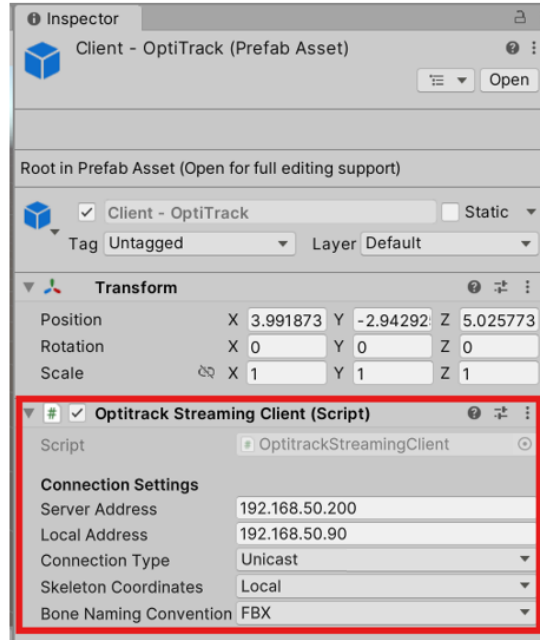


Figura 55: Configuraciones en el Optitrack Streaming Client

Después de configurar el objeto cliente, se arrastró el personaje a animar a la escena. A diferencia de MotionBuilder, en Unity no es necesario realizar un procedimiento específico de redireccionamiento de la animación para que los datos del esqueleto transmitidos desde Motive se apliquen al personaje. La transmisión se dirige directamente al personaje mediante un *humanoid avatar*.

El sistema Avatar de Unity permite que un modelo se identifique como humanoide, reconociendo automáticamente partes como brazos, piernas, torso y cabeza. Dado que los humanoides en Unity siempre comparten una estructura similar, el sistema de Avatar permite redireccionar las animaciones de manera sencilla a otros personajes humanoides [13].

Para las pruebas, se utilizó un humanoide incluido en el OptiTrack Unity Plugin, ubicado en la carpeta de Prefabs del *plugin*. Una vez colocado en la escena, se seleccionó el modelo y se configuraron los parámetros en el apartado OptiTrack Skeleton Animator (Script). En el campo de Streaming Client, se asignó el objeto Client – OptiTrack previamente configurado. En el campo Skeleton Asset Name, se ingresó el nombre del esqueleto transmitido desde Motive. Finalmente, en Destination Avatar, se generó un Avatar para el humanoide, y se seleccionó en el campo correspondiente para completar la configuración.

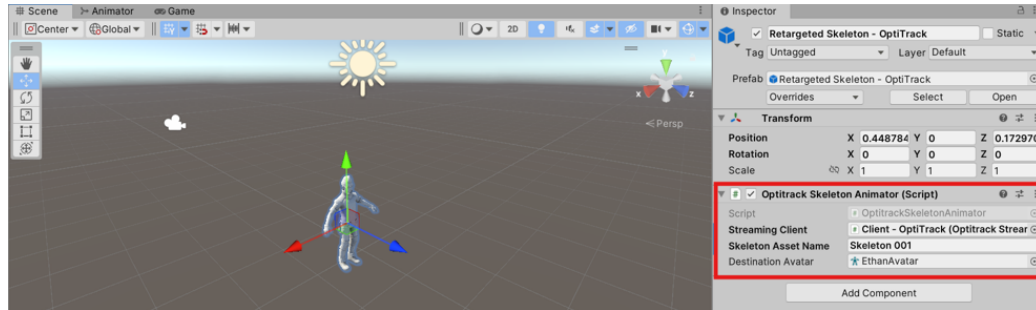


Figura 56: Menú de OptiTrack Skeleton Animator (Script) del personaje seleccionado en la escena

Una vez finalizada la configuración de los parámetros en el humanoide, se ejecutó la escena utilizando el botón de reproducción, como se muestra en la Figura 57. Esto permitió observar en tiempo real los movimientos del participante reflejados en el modelo 3D dentro de la interfaz de Unity.

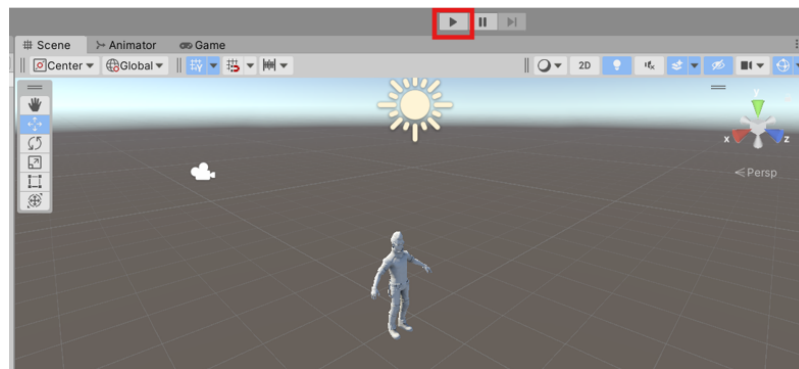


Figura 57: Botón de reproducción en la interfaz de Unity

10.3. Transmisión en tiempo real de la información de movimiento (Unreal Engine)

En esta sección se presentarán los pasos que se siguieron para usar el Unreal Engine: OptiTrack Live Link Plugin para transmitir un esqueleto capturado con el traje de OptiTrack al *software* de Unreal Engine.

10.3.1. Unreal Engine: OptiTrack Live Link Plugin

Se descargó el *plugin* de Unreal Engine en el directorio de *plugins* de la computadora que se usó para correr el programa de Unreal. Para activarlos se inició un nuevo proyecto y se habilitaron los *plugins*: OptiTrack - Live Link y Live Link.

En el menú del *plugin* de Live Link que se habilita en Window → Virtual Production → Live Link se estableció la conexión con la red de comunicación del Robotat por la que

Motive estaba transmitiendo la información. En el apartado de Server Address se ingresó la dirección IP de la computadora que estaba corriendo Motive. En el Client Address se ingresó la dirección IP de la computadora que está corriendo Unreal Engine. Por último, se colocó en Connection Type sí la transferencia de datos iba a ser por medio de Unicast o Multicast.

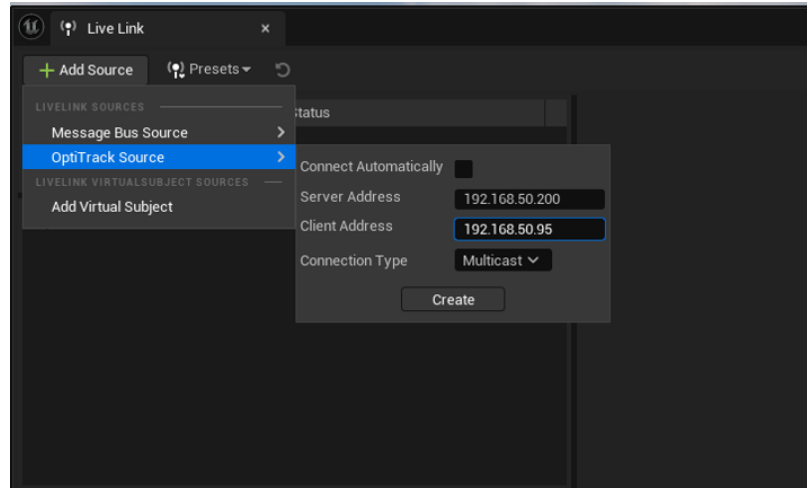


Figura 58: Live Link Pane en Unreal Engine para configurar los datos de la red

Luego, se seleccionó el botón de Create en el Live Link Pane, generando un listado de todos los *assets* presentes dentro de la zona de captura que estaban configurados en Motive para transmitirse a través de la red. En esta prueba, solo se transmitió un esqueleto. Al lado del nombre del esqueleto, un punto verde indicó que la transmisión y recepción de la información a través de la red se estaban realizando correctamente.

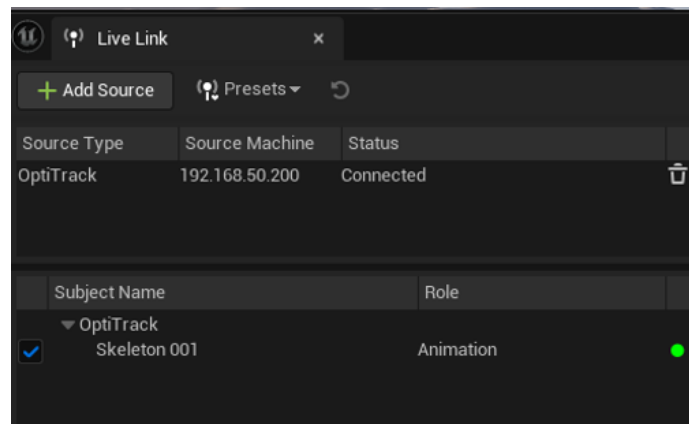


Figura 59: Live Link Pane con listado de los *assets* que se están transmitiendo desde Motive

Se importó el modelo 3D del personaje con *rig* que se desea animar al proyecto de Unreal Engine. En el Content Drawer del proyecto, se creó un objeto denominado Animation Blueprint, el cual se vinculó a la malla esquelética del personaje importado. El Animation Blueprint es un diagrama de bloques programable que se utiliza en Unreal Engine para controlar la animación de una malla esquelética durante una simulación [6]. Dentro del

Animation Blueprint, se colocó un bloque llamado Live Link Pose y se conectó al bloque de Output Pose. En la configuración de Live Link Subject Name, se seleccionó el esqueleto transmitido desde Motive. Además, el *plugin* incluye un objeto que permite redireccionar automáticamente la animación del esqueleto al personaje. Para ello, se seleccionó en el apartado Retarget Asset del bloque Live Link Pose la opción OptiTrackLiveLinkRetarget. Esta configuración facilita la adaptación de los datos de movimiento al modelo, permitiendo que el personaje siga en tiempo real los movimientos capturados.

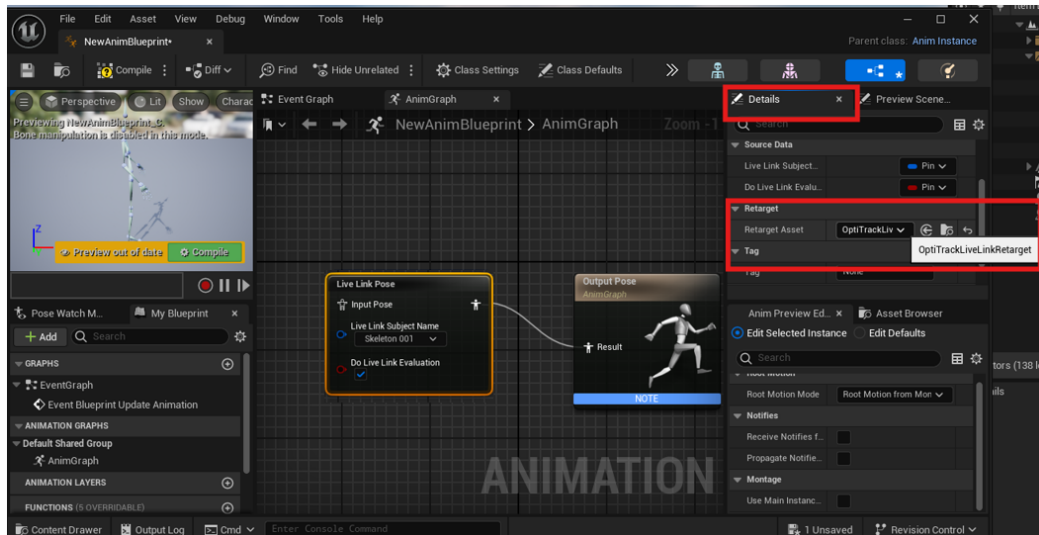


Figura 60: Live Link Pane con listado de los *assets* que se están transmitiendo desde Motive

Una vez guardado el Animation Blueprint, se arrastró desde el Content Drawer hacia la escena. Luego, para visualizar la animación en la escena, se seleccionó el Animation Blueprint y en el panel Details, se agregó el componente Live Link Skeleton Animation. Esto permitió ver en tiempo real cómo el personaje respondía a los datos de movimiento transmitidos desde Motive. También se añadió un display de los datos de movimiento transmitidos desde Motive. Esto permite comparar la información de movimiento recibida de Motive con los movimientos realizados por el personaje en la escena, facilitando la verificación y ajuste de la precisión en tiempo real entre la captura y la animación del personaje.

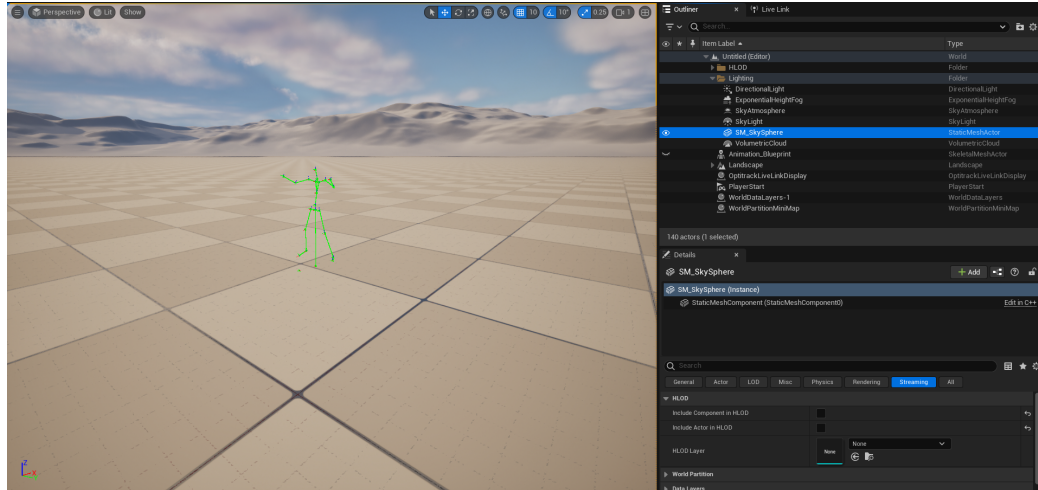


Figura 61: Live Link Display de la data transmitida desde Motive

Al colocar el display, se verificó que la información se estaba recibiendo correctamente en Unreal Engine; sin embargo, esta no se aplicaba adecuadamente al personaje. Aunque los datos de movimiento eran precisos, el mapeo presentaba desajustes. Esto puede deberse a que el objeto de redirección de la animación en el Animation Blueprint no se estaba adaptando correctamente el esqueleto del personaje. Además, una diferencia significativa en la escala o proporciones entre el personaje y el esqueleto transmitido podría estar afectando cómo se aplican los movimientos, provocando inconsistencias en la animación.



Figura 62: Comparación entre la información de movimiento recibida de Motive (esqueleto verde) con los movimientos realizados por el personaje (esqueleto blanco)

Para lograr un buen mapeo del esqueleto al personaje, como en los demás programas, se recomienda explorar la opción de hacer el redireccionamiento manualmente. Es decir, colocar manualmente el nombre de los huesos del esqueleto transmitido en un *blueprint*, asegurándose de que coincidan correctamente los huesos que deben ser mapeados. También se podría utilizar el Unreal Engine: MotionBuilder Workflow, que aprovecha a MotionBuilder como un preprocesador de la data de movimiento para ajustar y orientar los huesos antes de enviar los datos transformados a Unreal Engine.

Pruebas de movimientos en el volumen de captura

Para poner a prueba el volumen de captura de la Universidad del Valle de Guatemala y la transmisión de datos en tiempo real desde Motive hacia los programas de animación digital usando la red de comunicación del Robotat, se seleccionó un conjunto de 22 movimientos comunes que podrían ser de interés en futuros proyectos. Esto se logró al establecer un intercambio de información entre Motive y MotionBuilder siguiendo los procedimientos detallados en el capítulo anterior. Adicionalmente, se redireccionaron las animaciones en tiempo real para poder visualizar cómo se aplicaban a un personaje, lo que permitió evaluar la precisión de la captura de movimiento sobre el modelo animado.

11.1. Listado de movimientos

A continuación, se presenta un listado de 22 movimientos que fueron probados en el volumen de captura. Para cada movimiento, se evaluó el grado de cumplimiento del resultado de la animación final utilizando una escala que clasifica el desempeño en tres categorías: No cumplido, Parcialmente cumplido, y Cumplido. Esta evaluación permitió identificar las fortalezas y limitaciones del sistema en la captura y transmisión de los movimientos. Además, se lista el número de *take* con el que se identifica cada animación en el archivo FBX en que se guardaron.

Movimientos	No Cumplido	Parcialmente Cumplido	Cumplido	Número de <i>Take</i> en FBX
Rotación de cabeza y cuello				2
Elevación y rotación de muñeca con flexión del brazo				4
Caminar				5
Correr				6
Salto vertical				7
Giros				8
Salto de tijera				9
Sentarse y levantarse desde una posición en silla				10
Sentarse en el suelo con piernas cruzadas				11-13
Caída al suelo y pararse de nuevo				14-15
Movimiento de golpe				16
Movimiento de patada				17
Vueltegado hacia delante				18
Salto con giro				19
Simulación de caída por explosión				20
Arrastre estilo militar				21
Baile				22
Caminata en cuclillas				23
Despechadas				24
Acostarse boca abajo				25
Movimiento de brazos y piernas simulando "ángeles de nieve"				26
Gesto de reflexión (mano en la cabeza en actitud pensativa)				27

Cuadro 1: Evaluación del cumplimiento de los movimientos en el volumen de captura, con el número de *take* correspondiente en FBX.

Hay tres razones por las cuales los movimientos no se lograban traducir correctamente al personaje: oclusión de marcadores, desplazamiento de los marcadores o un mal redireccionamiento de los movimientos del esqueleto de Motive al personaje.

Casos en los que se dio la oclusión de marcadores incluyen movimientos como *sentarse en el suelo con las piernas cruzadas*, *arrastre militar* o el *movimiento de brazos y piernas simulando ángeles de nieve*. En estos movimientos, el participante se colocó en una posición en la que los marcadores quedaron cubiertos por su propio cuerpo, lo que impidió su detección por las cámaras del sistema de captura. Esta oclusión afectó la precisión de la captura de datos y dificultó la correcta transmisión de los movimientos al personaje. En particular, la oclusión en las piernas o los brazos. En las animaciones finales se pudo observar como estas partes del cuerpo se movían de manera no natural debido a la falta de información en las articulaciones ocultas.

La siguiente razón por la que hubo discrepancias en la animación final fue el desplazamiento de marcadores. En movimientos como *correr*, *vueltegado hacia adelante* y *caída al suelo y pararse de nuevo* (movimientos muy dinámicos), los marcadores específicamente de los pies, se desplazaron. Debido a la falta de calzado adecuado con velcro en el traje de captura, se utilizaron calcetines con adhesivos de velcro para fijar los marcadores en los pies del participante. Sin embargo, debido a que el calzado no era el óptimo, durante la grabación de las animaciones se presentaron problemas recurrentes en la captura del movimiento de los pies.

Por último, movimientos como la *rotación de cabeza y cuello* o el *gesto de reflexión* no se representaron correctamente en la animación final debido a un mal redireccionamiento del esqueleto. Al analizar la rotación de la cabeza, se podía observar que el punto de pivote de la cabeza estaba mal colocado, ya que la rotación no se veía natural. En el caso del gesto de reflexión, dado que las dimensiones del modelo no son las mismas que las del esqueleto de captura, las proporciones no se alinearon adecuadamente. Este desajuste provocó que el movimiento de la mano en la cabeza no se viera como el movimiento actuado, resultando en una animación que no replicaba la intención del movimiento original.

CAPÍTULO 12

Conclusiones

- Se logró reconfigurar el volumen de captura compuesto por seis cámaras *Prime x41* a un volumen de captura por cuadrantes, con el fin de que el sistema se pueda usar para la captura en humanos al igual que para agentes robóticos.
- Se obtuvo información de un esqueleto con el sistema de captura y se exportó como archivo .FBX para su uso en Unity y MotionBuilder.
- Se logró establecer una transferencia de información de un esqueleto, mediante la red de comunicación del Robotat del programa de OptiTrack Motive a MotionBuilder, Unity y Unreal Engine.
- Se redireccionó la información del esqueleto transmitida por Motive para animar modelos de humanoides en tiempo real en los programas de MotionBuilder y Unity.

- Con el fin de obtener animaciones fieles al movimiento real del actor, se recomienda adquirir más cámaras para el volumen de captura. Al momento de posicionarlas, se debería buscar que estén a alturas variadas para obtener una mayor cobertura y reducir problemas de captura; especialmente cuando el actor esté en posiciones complejas con posibles oclusiones de los marcadores.
- Diseñar una estructura que rodee la plataforma de captura para que las cámaras puedan colocarse a una mayor altura, proporcionando un mayor campo de visión y cobertura de la plataforma.
- Para evitar que reflejos externos interfieran con la captura de movimiento, se recomienda instalar cortinas en las ventanas del salón CIT-116 de la Universidad del Valle de Guatemala. Esto minimizaría las posibles fuentes de luz externa que puedan crear reflejos no deseados en el volumen de captura.
- Se recomienda adquirir calzado con velcro para los trajes de captura. Debido a la falta de un método de sujeción para los marcadores de los pies, se colocaron los marcadores usando calcetines con adhesivos de velcro, lo cual resultó en desplazamientos constantes de los marcadores durante las grabaciones. Para obtener animaciones más precisas y fieles al movimiento real, es preferible contar con un calzado con velcro diseñado específicamente para este propósito que mantenga los marcadores fijos.
- La computadora en la que se descargó el programa Motive está ubicada en el salón CIT-114, mientras que la plataforma de captura está en el salón CIT-116. Esta separación complica el monitoreo de la captura de movimiento, ya que el operador debe desplazarse entre ambos espacios para realizar cualquier modificación o verificación en el programa de Motive. Para optimizar el flujo de trabajo, se sugiere transferir el equipo de cómputo al salón CIT-116
- Para lograr un buen mapeo del esqueleto al personaje en el programa de Unreal Engine, se recomienda explorar la opción de hacer el redireccionamiento manualmente. Es decir, colocar manualmente el nombre de los huesos del esqueleto transmitido en un *blueprint*,

asegurándose de que coincidan correctamente los huesos que deben ser mapeados. También se podría utilizar el Unreal Engine: MotionBuilder Workflow, que aprovecha a MotionBuilder como un preprocesador de la data de movimiento para ajustar y orientar los huesos antes de enviar los datos transformados a Unreal Engine.

-
- [1] M.-S. E. Ltd., *What is motion capture and how does it work?* Consultado: 2024-03-25, 2020. dirección: <https://www.mo-sys.com/news/what-is-motion-capture-and-how-does-it-work/>.
 - [2] C. P. Montoya, “Robotat: un ecosistema robótico de captura de movimiento y comunicación inalámbrica,” Tesis de licenciatura, Universidad Del Valle de Guatemala, 2022.
 - [3] I. NaturalPoint, *OptiTrack*, Consultado: 2024-03-18, 2024. dirección: <https://optitrack.com/>.
 - [4] I. NaturalPoint, *OptiTrack*, Consultado: 2024-03-18, 2024. dirección: <https://docs.optitrack.com/>.
 - [5] M. Thomsen, *History of the Unreal Engine*, Consultado: 2024-03-20, 2012. dirección: <https://www.ign.com/articles/2010/02/23/history-of-the-unreal-engine?page=1>.
 - [6] I. Epic Games, *Unreal Engine*, Consultado: 2024-03-20, 2024. dirección: <https://www.unrealengine.com/en-US>.
 - [7] A. Inc., *Maya*, Consultado: 2024-03-23, 2024. dirección: <https://www.autodesk.com/products/maya/overview?term=1-YEAR&tab=subscription>.
 - [8] A. Inc., *Motion Builder*, Consultado: 2024-03-23, 2024. dirección: <https://www.autodesk.com/products/motionbuilder/overview?term=1-YEAR&tab=subscription>.
 - [9] U. Technologies, *Unity*, Consultado: 2024-03-24, 2024. dirección: https://unity.com/products/unity-engine?_rsc=6ze0e.
 - [10] B. N. y P. (Firm), *3D Animation Essentials* (Serious skills). J. Wiley y Sons, 2012. dirección: <https://books.google.com.gt/books?id=d7WWzQEACAAJ>.
 - [11] B. W. Midori Kitagawa, *MoCap for Artists Workflow and Techniques for Motion Capture*. Focal Press, 2008.
 - [12] A. Inc., *Motion Builder*, Consultado: 2024-10-20, 2024. dirección: <https://help.autodesk.com/view/MOBPRO/2024/ENU/>.

- [13] U. Technologies, *Unity Documentation*, Consultado: 2024-11-02, 2024. dirección: <https://docs.unity3d.com/Manual/index.html>.

