(CV-CG joint project)

Animating and compositing motion capture actor performances using OptiTrack and camera motion tracking.

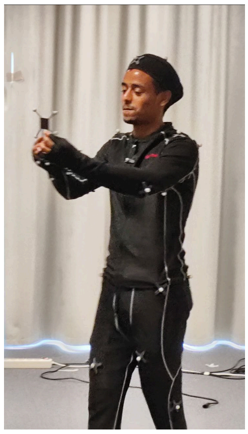
Our project consisted of using a commercial OptiTrack motion capture system to record a set of 3D animated performances, which we then retargeted, rendered, and composited in order to create animated characters performing realistic motion in lifelike environments. Additionally, we applied camera tracking in order to seamlessly composite the character onto a moving video footage.

Motion Capture

Motion capture is a technology that allows for accurately tracking the movement of people and objects in 3D space using specialized equipment, and it is usually combined with computer vision techniques. For our project, we used an **OptiTrack system**[1] designed to capture human motion with high precision.

Hardware setup

The university's Multisensory Laboratory is equipped with eight ceiling-mounted infrared cameras with built-in infrared LEDs capable of recording at 360 frames per second (*left*). The controlled environment setup allows the cameras to **focus only on the** retroreflective infrared **markers**, filtering out most other visual information which **reduces the amount of data** to process and **simplifies** the computer vision problem of tracking the markers in the 3D space. **Blackout curtains** prevent sunlight interference, and the system captures scenes from **multiple angles** to minimize occlusion. The system's **calibration**, which was completed before our session, ensures accurate camera positioning and scale. For motion capture, the actor wears a **specialized velcro suit** (*right*) with markers placed on key body points for precise tracking.



Software setup

Paired with the OptiTrack cameras we used a **licensed** copy of **Motive software**[2] which delivers real-time 3D motion data from the cameras with top-level accuracy. The software offers a user-friendly interface that allows users to select the type of asset needed for the recording session and choose the number of markers to use to guarantee different levels of precision.

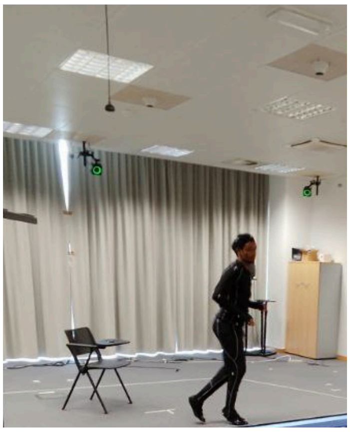
Motion capture session

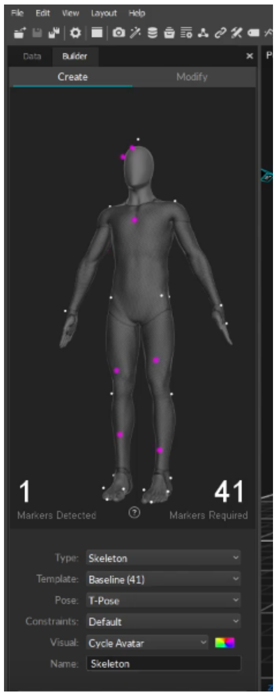
The first step of our session was to run the **software** and **prepare the environment**. Namely, we taped the blinds shut and removed any reflective item that might interfere with

[1] OptiTrack - Motion Capture Systems

[2] Motive software

the cameras. The next step was to prepare the actor for a full-body capture. In Motive, we selected the template marker set and then we followed the 3D guide (*right*) to place the 41 markers on the suit. We encountered a few issues with **marker positioning**, which required minor adjustments to ensure proper alignment. An issue we faced while working with the captured data is the lack of realism in the hand movements, due to the limited amount of **markers on the hands,** which is most noticeable on the handstand walking scene. Once we made sure that the markers were correctly positioned on the suit, we created our asset. Finally, we started recording the actor performing different movements within the motion capture field. The cameras tracked the reflective markers in real time, creating a digital skeleton based on his movements. We recorded several takes, also including untracked props such as a chair (*left*) and a cardboard box . The captured motion data was saved in .tak format, a proprietary format used by OptiTrack, but it was exported into .bvh and .fbx for better compatibility with animation software.





Retargeting (w/Rokoko)

To automate the retargeting by mapping between the mocap skeleton and the 3D character's rig, we used **Blender**[3] with the **Rokoko plugin**[4]. We imported the captured data in .bvh format and a 3D character model from **Mixamo**[5] in .fbx format. The main issue that we faced was that the **difference in proportions** of the **character** and the **actor** led to unrealistic overlaps and clipping when performing even simple motions, such as crossing the arms, as well as some apparent slipping during walking.

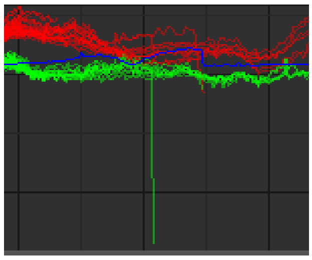
Perspective and light matching (w/fSpy)

To align the camera perspective in Blender with that of our background image we utilized **fSpy**[6]. This tool analyzes user-placed control points in the image in order to compute the intrinsic camera parameters as well as the rotation, distance and scale from the origin scene. Using the **importer add-on for Blender**[7], we imported those parameters as a Blender camera. To ensure that the lighting in the 3D scene matched that of the background image we manually positioned and adjusted the Blender light sources to replicate the lightning.

Camera motion tracking (w/Blender)

With the suggestion of our advisor, we decided to explore the possibility of compositing the animation over a **moving camera video** instead of a static background to create a more

[3] [Blender software]
[4] [Rokoko plugin]
[5] [Mixamo]
[6] [fSpy]
[7] [fSpy importer add-on for Blender]



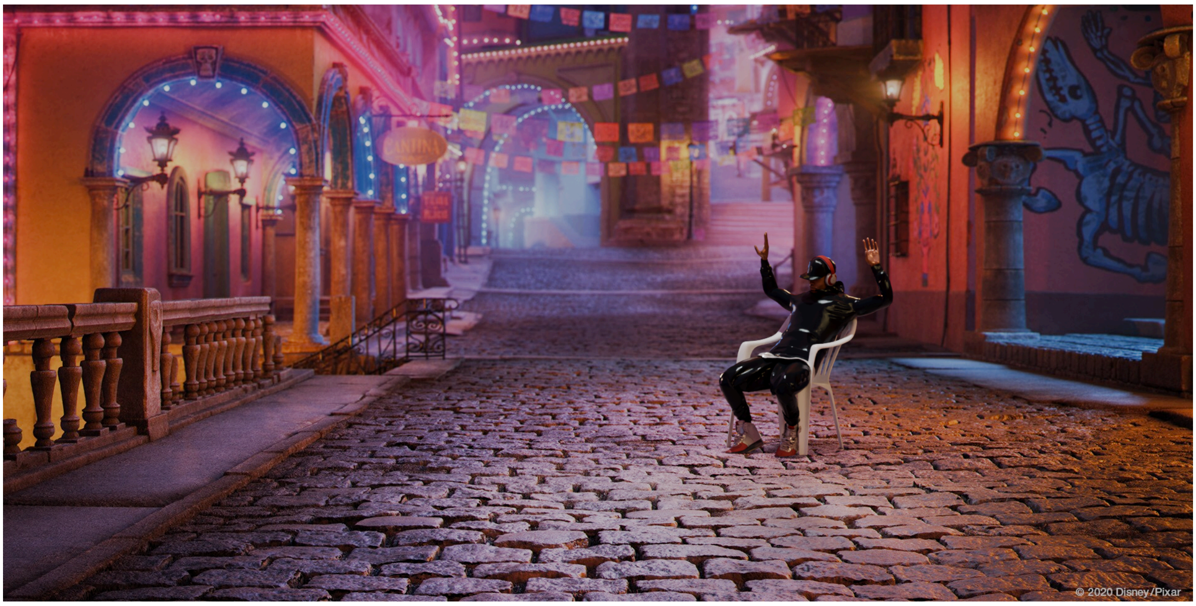
realistic demo. We recorded a video behind the Povo 0 building and captured a photosphere to match the lighting and reflections. To accurately place 3D objects in the scene, we needed to calculate the **camera**’s intrinsic **parameters** and its **position for every frame** to mimic the real life motion in the 3D scene. We used Blender’s **motion tracking workspace**, which employs a feature tracking algorithm to track the apparent motion of objects between frames. Once enough features are tracked accurately, the motion of these trackers is used to estimate **camera motion**. Blender offers a **Good Features to Track** algorithm to find easy to track points. Many of these were however unstable or quickly lost. Using the x and y deflection graph (*right*), we removed obvious **outliers** and **badly tracked** features. To mitigate the loss of tracking, we changed the reference from keyframe to the **previous frame**, helping track features despite minor distortion from camera motion. While **automatic features** were mostly useful for **foreground points,** they were lacking in the background. To address this, we also selected features **manually** on intuition, strategically selecting points like fence post **tips**, window **corners**, manholes, and picnic benches to help align the 3D scene. With these adjustments, we were able to get a camera solve **error** of about **1 pixel,** which is not ideal, but we were likely limited by the quality of the smartphone camera. After that, we needed to set the **scale and rotation**. We used the distance **between fence posts** to set the scale of the scene, while the tracked points on the ground helped determine the floor position and axis direction.

Final results

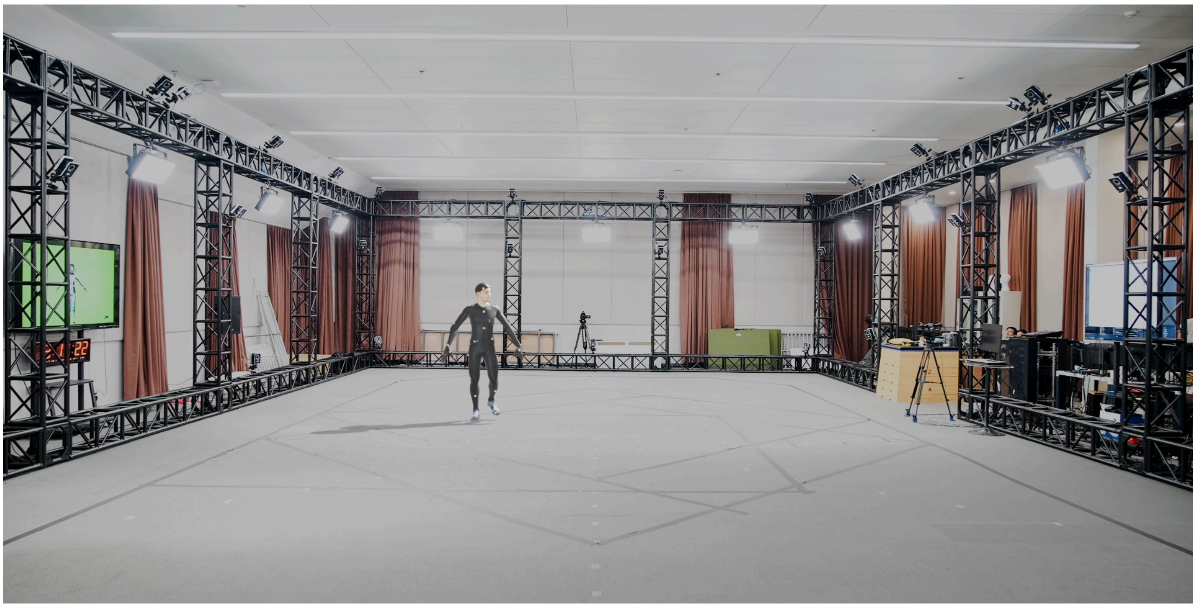
The final results are in the form of the **three video files**, each utilizing a slightly different **technique**. The first video was created as a test and was composited over a CGI background taken from Disney’s Coco. The second featured a real **motion capture setup** photo as background, while the final one used a video. We used different rendering engines and techniques for each, each with its own limitations. We took **seven** motion capture **recordings** and used three of them for our videos to experiment with a variety of **real-life motions**. They included one recording of the actor **sitting** in a chair and standing up, another of him **walking** around and **waving** in the capture area, and finally the hand walking scene. These actions represent fast, realistic, and **complex movements** that would be hard to track without motion capture techniques, especially the handwalking scene. However, some visual imperfections still persist from the capture and retargeting process. In all videos, the actor and character model are of different sizes. This led to **clipping and slipping**, as mentioned in a previous paragraph. Each video also shows specific limitations. In the Coco video, the **chair** used is not the same model as the one from the lab, this causes the character’s hands to clip into the environment. In the hand walking scene, the lack of **trackers on the hand** is evident as the hand appears inaccurately posed and the character also appears to be hovering on the ground. In conclusion, our results appear rather realistic at first sight, as far as interactions go. However, upon a closer look, some **limitations** inherent to motion capture are visible.

On the following page, we will display a series of **meaningful screenshots** from our results to provide a better explanation of the points discussed in this last paragraph, as well as a link to the final video itself on Google Drive.

static-background-coco.mp4



mocap-lab-animation.mp4



motion-tracked-handwalk.mp4