Tampere University

Sanna Salonen

MOTION CAPTURE IN 3D ANIMATION

Bachelor's thesis
Faculty of Information Technology and Communication Sciences
Examiner: Pia Niemelä
May 2021

ABSTRACT

Sanna Salonen: Motion Capture in 3D Animation
Bachelor's thesis, 25 pages
Tampere University
Computing and Electrical Engineering
May 2021

3D-modelling has become a popular way to create graphics for many IT-based applications. This can be credited to its reusability, detailedness, and capability to portray depth. These models are often used in animation production and at the same time the popularity of motion capture has also risen. Motion capture is a way to transform motion of real physical objects to digital data. Thanks to benefits such as interactivity, realism, and automatization it has made it an increasingly more appealing way to create animation. These methods have found use, for instance, in in entertainment, sports, and medical applications. However, combining motion capture with 3D-animation is a complicated process, which comes with several stages and a set of development challenges.

This work was done as a literature review, which set out to discover what kind of challenges and development possibilities exist in using motion capture for 3D-animation. The work first explored the fundamentals of 3D-modelling and -animation. Secondly, it showcased motion capture and its different methods of implementation. Finally, the work focused on three central challenges of the animation process: accessibility, cleaning the data and fitting it to a model. These challenges span the entire production process from acquiring the data to processing it.

Result was a thorough review of different parts of the process as well as their development possibilities. Along with the strengths and weaknesses of each motion capture system it was revealed that accessibility and simplicity of a system is inversely proportional to the quality of the acquired data. This is a problem, because in future the expansion of motion capture usage relies largely on consumer products. The priority therefore is to simplify the systems without compromising the quality. Machine learning algorithms, for instance, can be used to understand the details and nuance of human motion, even with inadequate data. In the future they have the possibility to further advance technological development of motion capture, so simple systems with small amount of equipment have their appeal. Additional benefits for this technology can also be found in other parts of the production pipeline. However, even more complex systems will continue finding use in high budget productions, because their high-quality data is reliable.

Keywords: 3D-model, 3D-animation, skeletal animation, motion capture

CONTENTS

LIST OF SYMBOLS AND ABBREVIATIONS

2D	Two-dimensional
3D	Three-dimensional
4DPC	4D performance capture
CG	Computer-generated
CNN	Convolutional neural network
FK	Forward Kinematics
IK	Inverse Kinematics

.

1. INTRODUCTION

With the development of computer graphics it has become possible to create increasingly more intricate animation than ever before. Digital three-dimensional (3D) models have revolutionized the field and proven themselves to be extremely fast and efficient to use in the creation of animation. Not only do they provide depth information and great detail, but they can also be reused in separate frames of animation without having to be reconstructed from scratch. The graphics of huge amount of modern video games, for instance, are done with heavy emphasis on 3D. This high demand for 3D animation has raised interest in making it more efficient to produce while also increasing its perceivable level of quality.

As a result motion capture technology has become an essential part of animation production and will become even more important in the future. Motion capture refers to tracking motion of real objects and turning it into digital data. Tracked human motion can, for instance, be fitted on a 3D character model that it moves accordingly. Many films have combined live action with computer-generated (CG) characters. To make the animation realistic enough to blend in with its environment, the motion of real actors is tracked. Examples of such films include Star Wars (1999), Lord of the Rings (2002), King Kong (2005), Pirates of the Caribbean (2006), Avatar (2009), Pacific Rim (2018) and many more. Highly life-like results can be attained, but motion capture can be used for more cartoony animation as well with some adjustments to the data [1]. Motion capture is also used in video games, virtual reality, live shows, sports analysis, medical applications and more. Motion capture is an essential part of virtual reality applications for instance, allowing the player to move in real life while having that motion displayed in real-time CG graphics. Motion capture has already been utilized in some well-known older video game applications such as Nintendo Wii (2006) and Microsoft Kinect (2010).

With the use of motion capture comes great benefits. Firstly, animating character models by hand is a tedious process that can be aided and even automated with the use of motion capture data, helping to reduce production costs [2]. Secondly, motion capture can track even the most subtle movements of the human body, making it possible to create very natural and realistic animation. Finally, interactivity and real-time possibilities are also motion capture's major selling points. All these benefits combined make motion

capture desirable in various applications, all the way from high budget film productions to average game consoles consumers can buy. However, with all these benefits also come challenges that keep motion capture from reaching its true potential.

This work as a whole is meant to provide understanding of how motion capture can be utilized in 3D animation as well as direct attention to areas where new developments are being researched. The goal is to find answers to the research question of what kind of challenges exist in utilizing motion capture in 3D animation and what is being done to solve those issues. Second chapter explains the fundamentals of 3D modelling and how these models are set up for animation. Third chapter covers how motion capture works and introduces some of the most popular systems used in animation production. Fourth chapter addresses common challenges one faces when using motion capture in 3D animation. Finally, in the fifth chapter conclusions are made from the results.

2. FUNDAMENTALS OF 3D ANIMATION

Digital animation was almost exclusively produced in two-dimensional (2D) format for decades. Each frame was manually drawn by hand and shown in a quick sequence to create an impression of motion. Technique known as cel animation utilized transparent plastic sheets that were drawn on, layered, and photographed before being used as frames in digital format. At the turn of the millennia, purely digital animation methods started becoming more commonly used than traditional methods. This can be contributed to the development of computer software, which managed to simplify and automate parts of the tedious creation process. Digital keyframing made it possible to draw reference frames, such as poses, and automatically interpolate rest of the frames between them [3]. Additionally, computer software and drawing tablets allowed artists to draw digitally just like on paper, making cel animation obsolete. The next natural step in the digital world of animation was moving to 3D format.

3D animation in this case refers to using three-dimensional data to model graphical objects for motion. Its traditional counterpart can be thought of as stop motion animation with physical puppets. Storing the data in 3D format makes it possible for scenes to have depth, although the data is converted to 2D in rendering so that it can be displayed on a monitor. In animation this feature is useful for determining what the depicted scene looks like in motion. Scene refers to the rendered environment and any other objects in it. 3D models can be used to render very intricate, even photorealistic, objects and scenes and lend themselves to reusability. The models do not need to be recreated from scratch for every new frame, which is a major drawback for traditional 2D animation.

The industry demand for 3D animation is high and will only keep increasing as years go on. 3D art often looks distinctly different from typical hand-drawn highly stylized 2D art, but it has over the years gotten better at emulating its feel. Preference over whether 2D or 3D is more visually appealing is a matter of subjective taste, but both methods can be utilized in unison and have their place in the industry.

2.1 Polygonal modelling

3D models can be constructed and represented in various ways, but by far the most widely used one is a polygonal model. Due to its simple structure, it is cheap to render and is generally the only option for real-time engines [4, p. 59]. This work will focus on polygonal models due to their popularity and suitability for animation.

In the polygonal model three-dimensional data points, vertices, are connected to each other by edges. These edges form polygons, also referred to as faces of the model, that are either three-sided triangles or four-sided quads. All individual faces are flat planes that form the surface of the hollow wireframe structure. This is more commonly referred to as the polygon mesh.

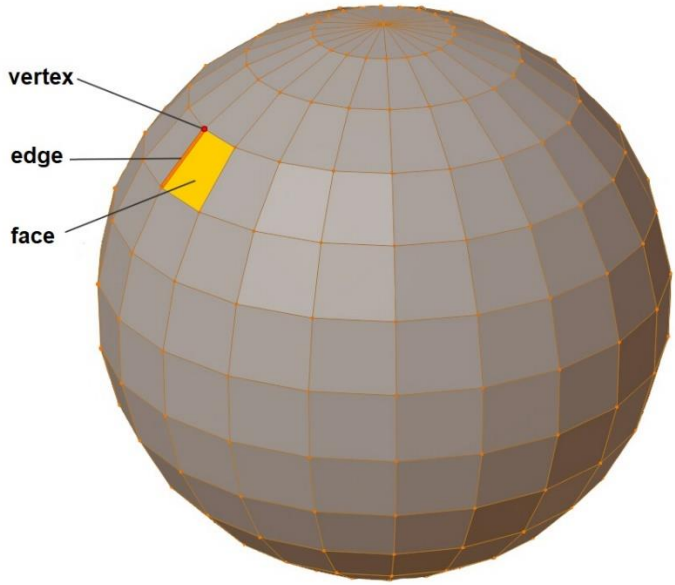

Figure 1. *3D model of a sphere created in Blender.*

The performance of a 3D model is typically measured with its polygon or vertex count. Resolution refers to the polygon density in a specific local or global area of the model. Essentially, the higher these parameters are, the more complex the model is. Estimation of the appropriate resolution is case specific and often related to what the hardware can reasonably handle, but as hardware develops over time, so does its capability to render more complex 3D models [4, p. 61].

If the model's resolution is high, it becomes heavy to render. On the other hand, if the model's resolution is low, it is visibly blocky and applying motion to the mesh is harder, because high-movement areas are impaired [4, p. 60]. These encompass, for instance, joints and other flexible areas on character models. High-movement areas are modelled while keeping their potential motion ranges in mind, because possible mesh deformations, such as stretching, should look natural. Choosing the right resolution for the model is a matter of compromise. If real-time rendering is necessary, simplicity is desirable. Topology refers to the structure and layout of the mesh and its quality is essential in animation.

The model is visually enhanced from the simple polygonal mesh in the final rendering. Textures are generally added by mapping a 2D image onto the surface of the 3D model by linking coordinates of the image to specific mesh vertices. The object can also have effects such as shaders applied to it with the render engine. Complexity concerns for real-time applications have to be considered here as well and can affect the choice of a render engine and applied effects. Some render engines strive more for realism and complexity while others for computational simplicity. Video games, for instance, often have historically had lower visual quality than TV-shows and movies, because of the real-time constraints. Complexity is not as big of an issue for prerendered footage.

2.2 Skeletal animation

To animate a polygonal model, vertices in the mesh need to be manipulated and moved to produce different poses. Animation input interface for the mesh is generally provided by a separate motion system. The most popular method for this is skeletal animation, which represents the motion system as a skeleton. The mesh in this case represents the skin that follows the skeleton's movements.

A digital character can be thought to consist of three connected systems overall: the motion system that provides the skeletal animation, the control system that can be used to manipulate the skeleton and deformations system that defines how the mesh will move in relation to the skeleton. [4, p. 56, 87] Commonly this setup is referred to as a character rig. The mesh with this method is not manipulated directly as that would be a tedious process with a high number of vertices. Instead the skeleton simplifies the process by providing much fewer controls for the animation input. Methods to achieve mesh deformation without a separate skeleton have been developed, but skeletal animation is the common practice in many of the most popular 3D modelling software [5]. This work will cover the fundamentals of skeletal animation, because of its established popularity. With motion capture data separating the animation control input from the mesh deformations simplifies the process considerably.

The skeleton consists of joints that are connected to each other by bones. The joints, like in real human bodies, can bend while the bones stay rigid. The bones themselves are attached to mesh. All the vertices of the mesh are segmented into different categories based on their association with one or more bones, meaning that the movement of a specific bone only affects the vertices it is directly linked to [6, p. 57-59]. The skeleton as a whole represents a hierarchical tree with joints as the nodes and bones as the edges [7]. Bones and joints in this case form what is referred to as joint chains together.

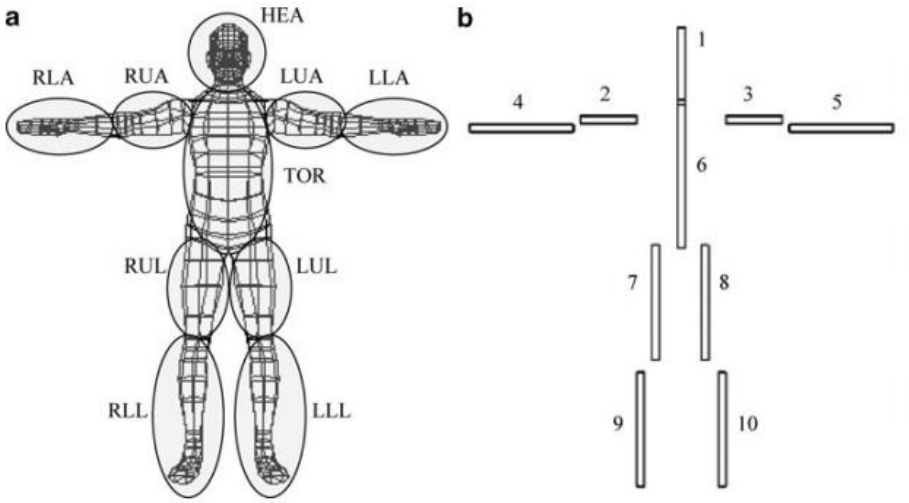

Figure 2. *Grouping of the mesh vertices (a) based on the bones (b) [6, p. 58].*

The process of binding the mesh to the skeleton is referred to as skinning. Bones and joints are placed in any location of the 3D model that requires mesh deformations and their amount determines the complexity of the motion. To create a face rig for instance, bones and joints are tightly placed in key locations of the face so that by manipulating them a wide range of different facial expressions can be created.

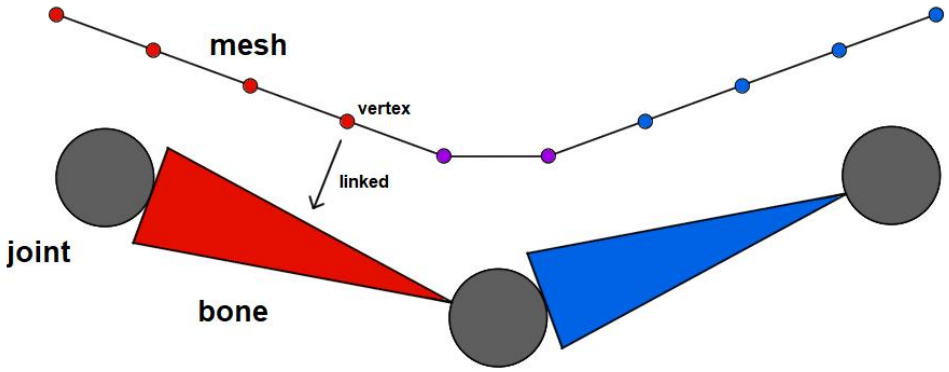

Figure 3. *Relationship between the skeleton and the mesh*

Joints dictate the directions the bones point to and serve as the controls for animation input. They can be represented by homogeneous matrices to which rotational and translational transformations are applied [7]. According to Euler's theorem, any rotation transformation can be represented by a maximum number of three rotations around x, y, and z axes. Euler rotation can therefore be applied to a joint with separate rotations in a successive order. However, the rotation order for the different directions can influence the end results, causing ambiguity. [6, p. 84]

Although Euler rotation is the more common method, the rotation can be applied with the Quaternion rotation instead to avoid issues related to rotation order ambiguity and certain rotation combinations cancelling each other out. The method can select any axis to rotate the object around instead of having only x, y, and z axes to choose from. A joint with lots of free movement such as the shoulder can benefit from this, but for a joint with one rotation direction such as the knee it is not as useful. [4, p. 90-92][8] Each joint can have six possible degrees of freedom in total, three in rotation and another three in translation. Constraints are commonly placed on joints to restrict their degrees of freedom and how far they can move in each axis to mimic the function of real bodies.

Euclidean translation selects a specified direction and moves every point a certain distance towards it in the three-dimensional space. However, translation is generally only applied directly to the root joint to move the whole model around if necessary, not to individual joints. Instead, joints get their respective translations according to the rotations in the joint chain. [7] The bones themselves are rigid objects so they are not deformed when a rotation is applied on a connected joint. A bone simply moves as dictated by a joint and passes the motion to its other connected joint in the chain.

The skeleton's hierarchical tree is controlled either by forward (FK) or inverse kinematics (IK), although systems can use of both in unison and flip between them. Kinematics in this case refers to how rotational and translational motion is passed along the joint chain [6, p. 113]. In FK rotations are applied starting from the chosen parent joint and passed down to its children in an accumulative fashion. More commonly IK is used where the motion is applied to the end joint and passed up the chain to its root. With IK, a solver is used to compute the locations of the joints down the chain. This gets more complicated the longer the chain is, because the rotation is only applied to the end joint and rest need to be computed independently. [4, p. 101-103] If, for instance, the hand is moved to a specific location, the elbow has several different directions it can point to. The solving process can be simplified with the use of constraints, because they narrow down the number of possible solutions a joint rotation can have. This can be achieved by removing some degrees of freedom or by restricting how far on a certain axis a joint can rotate.

IK, in summary, solves a posture for a character by estimating each degree of freedom for all the individual joints. IK computation problem is a challenge of its own, because it needs to solve a posture as smoothly and computationally lightly as possible. The end result will also have to look as natural and realistic as possible to achieve a satisfactory level of perceivable quality. [9]

After the character rig has been constructed, skinning needs to be applied to it to make the mesh vertices follow the movements of the skeleton. The placement of the joints and the bones needs to be chosen carefully so that the resulting mesh deformation is appropriate. Each vertex is linked with a corresponding bone, even several near joint areas, and transformations are applied based on these weighted links.

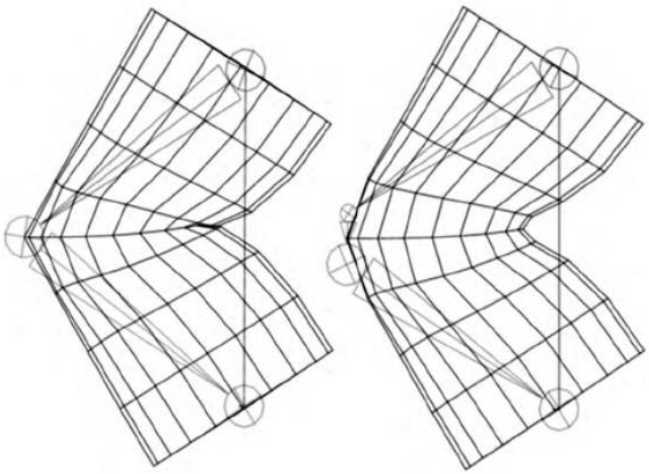

Figure 4. *Joint and bone placement can affect the mesh deformation [4, p. 93].*

Like with IK solvers, numerous different computation methods for mesh deformations have been developed. As an example, the most notable method is linear blend skinning (LBS). It is very widely used due its simplicity and effectiveness and is thus suitable for real-time applications. As a downside it suffers from artifacts. It can produce a twisting effect on the mesh, often referred to as the “candy-wrapper” effect, or even lead to loss of volume. LBS also does not capture non-kinematic motions such as stretching or bulging. It is possible with physics-based simulations, but it is still complicated and computationally expensive. Data-driven methods on the other hand can be used to produce realistic kinematic skinning functions with the help of motion capture data. [10][11] As there are many factors that need to be considered when skinning, picking a method is a matter of compromise.

Animation that utilizes 3D models is often produced with manual keyframing. To produce poses that can be used as keyframes, transformations are applied to the model’s skeleton by an animator with FK or IK techniques. Additional in-between frames needed to make the footage smooth are automatically interpolated. While it is an artform that has value in being created by hand, for many applications there is merit in automating the animation process. One way to achieve this is with motion capture solutions.

3. MOTION CAPTURE METHODS

Motion capture evolved from rotoscoping, a technique where video footage is traced over by hand to create animation. The principle behind motion capture is the same, using real motion as a reference for animation, but this time in three-dimensional format. [4, p. 221][8][12] Motion capture tracks and samples motion with cameras or sensors and turns it into digital data. In practical terms this refers to capturing the motion of a human performer, although motion capture can be utilized on animals and inanimate objects as well. Another term, performance capture, is often used interchangeably, but refers to capturing more subtle details than what motion capture is capable of. This entails details such as facial expressions and finger movement. In this work, for the sake of consistency, the term motion capture will also encompass performance capture.

The captured data represents a set amount points that move in a three-dimensional space over a period of time. These points should be located in pivotal places on the body so that they can later be used to reconstruct its pose. [12] This simple representation is enough to be useful in a wide range of applications, such as 3D animation where the data points can be fitted on the character rig's joints. Rotation and translation transformations are applied to the joints with the motion capture data, effectively making the model copy the movements of what was originally being tracked. Motion capture data can be used to animate a model either entirely on its own or in combination with some input from an artist.

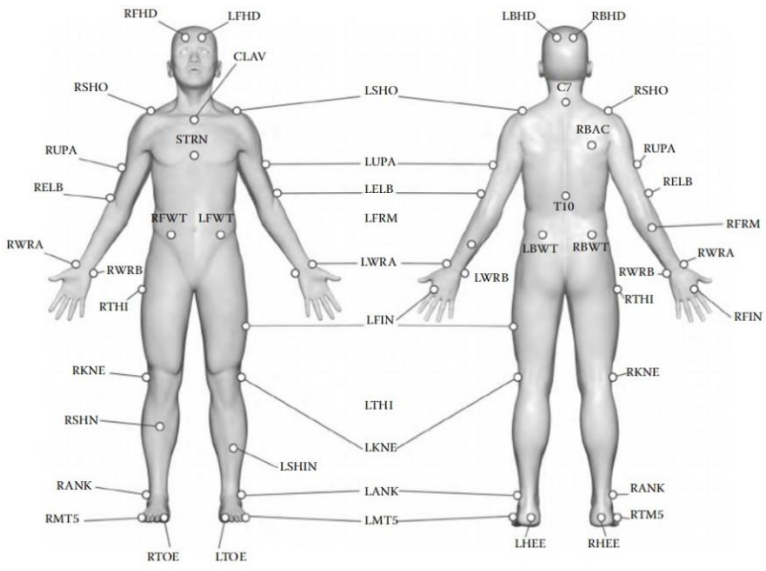

Figure 5. *Example placement and labelling for the trackable points [4, p. 223].*

Initially mostly mechanical motion capture systems, heavy and restrictive exoskeletons, were used, but soon other systems saw increase in their popularity [3]. Each motion capture system has its own advantages and disadvantages that determine what kind of applications it is most suitable for. Differences are typically found in systems' accessibility, complexity, and quality of the acquired data. This chapter will cover the most notable motion capture methods that can be utilized in the creation of 3D animation.

Motion capture methods can be divided in two different ways: optical and non-optical systems as well as marker-based and markerless systems. Markers in this case refer to physical objects, typically sensors, that are placed on the performer's body. This is often done with a special kind of wearable suit. Most motion capture systems utilize some kind of markers, but markerless systems can bypass this with cameras and computer vision techniques.

3.1 Non-optical systems

Non-optical systems rely on various kinds of sensors rather than cameras to track motion. These sensors are typically placed on the performer, making these systems marker-based. Notable non-optical motion capture systems that have been in used in the past decades include acoustical, mechanical, magnetic, and inertial systems [2]. More systems naturally exist, but will not be covered in this work.

An acoustical motion capture system consists of sound transmitters on the performer's main joints, three receptors placed in the capture site and a data processing module. [2][3][13] Receptors calculate sensors' locations from time and phase differences by picking up the characteristic frequencies the emitters fire sequentially [13]. Due to this acoustical systems tend to suffer from interferences, such as noise caused by reflections [2][3][13]. The system's lower cost works as a compromise for the quality of the acquired data [2].

Mechanical motion capture system is an exoskeleton that consists of rigid straight rods, potentiometers that capture their angles and often an accelerometer that captures global translation [3][8]. When worn, rods represent the bones and potentiometers the joint rotations of a skeleton. The angle data is given to a kinematic algorithm that determines body posture [14]. Mechanical system does not suffer from environmental interferences and is simple, meaning that it can be used anywhere in real time. As a downside its ability to determine global translation and to get high-quality data is poor. The exoskeleton easily restricts the movements of its wearer, resulting in stiff motion. [2][5][8][12][13] Despite

its issues the mechanical system is relatively popular thanks to its low cost and simplicity [2][12].

In a magnetic motion capture system the place of the markers is taken by magnets. These sensors measure their spatial relationship in respect to a magnetic transmitter, making it possible to compute rotation and translation data in real time. [3][8] Magnetic systems, however, are easily susceptible to electrical and magnetic interferences of the environment. For this reason the environment needs to be kept free from high-conductivity metals [8][13]. Additionally, magnetic motion capture tends to have a low sampling rate and the data is noisy [8].

Finally, inertial measurement units (IMU) are used in an inertial motion capture systems. IMU typically consists of a combination of accelerometers, gyroscopes, and magnetometers. It has the capability to measure the direction of Earth's gravity and magnetic field so orientation can be determined with great accuracy. Motion based on acceleration and turning rates can be determined as well. An inertial system is very simple and can be used in any environment. The main issue with inertial systems is an effect referred to as drifting, where error cumulatively rises the longer the system is in consecutive use. The system therefore needs to be calibrated often to provide it usable reference points. [15] Drifting in inertial systems is reduced slightly by the complementary sensors IMUs consist of [14].

These kinds of complementary systems are an effective way to make up for the weaknesses found with specific types of sensors. A mechanical system, for instance, can be combined with accelerometers on limbs to produce more convincing animation [5].

3.2 Optical marker-based systems

Optical marker-based systems determine the location of physical markers with an assortment of high-speed cameras placed around the set. These markers are placed on the performer’s body and their positions are determined from camera footage with triangulation. Triangulation is a mathematical way of reconstructing 3D objects from numerous 2D images based on camera location and projection matrix information. This process mimics the function of human eyes and how they perceive depth and spatial positions [13]. At least two cameras are required, but any more than that increases the system’s accuracy [8]. High camera count increases post-processing time, but in addition to higher quality of the acquired data it also prevents marker occlusion [12]. Marker occlusion refers to a camera not having a direct line of sight to the tracked marker, which causes

gaps in the recorded data. This issue does not need to be considered with most non-optical systems.

Two types of optical motion capture markers are used: active and passive. They both rely on light emitted by LEDs to track the markers. Active markers emit light that is captured by the cameras while passive markers reflect the light emitted from the cameras back to them [8]. Although only active LEDs allow cameras to directly identify markers to prevent confusion between them, passive optical is the most prevalently used method in animation productions. This is primarily due to its accuracy, flexibility, and capability to capture even high-detail performances. [5] Marker count is theoretically limitless as well, although reflectors too close to each other get easily mixed up.

Figure 6. *Performers in Vicon's motion capture suits. [16]*

Despite this preference both optical systems have a very high sampling rate and freedom of movement for the performer, making it possible to produce dynamic motion data. [2][3] However, like with electromagnetic systems, optical systems require strict environmental control in regard to lighting, and the equipment setups are complex [2][8][13]. Fair amount of post-processing is also required, which increases operating costs and makes utilization in real time more difficult [2][3][8][12][15]. Non-optical systems in comparison have a lower barrier of entry than optical marker-based systems. Significantly higher costs and complexity come as a compromise for improved quality of the acquired data.

3.3 Optical markerless systems

All previous motion capture systems require equipment that either the performer has to wear or is part of the calibrated environment. This makes the motion capturing process quite complicated. However, in recent years there has been rapid development in optical markerless systems, which try to solve these issues. Only a simple camera setup is required to use computer vision and machine learning algorithms to track features that can

be reconstructed in digital format. Basic principle is the same as in optical marker-based systems, but as the name suggests, the performer does not have to wear physical markers on their body.

Optical markers make feature detection a relatively simple task so without them the features have to be recognized from distinct natural shapes [5]. In the case of humans, features of the face and body. In the face these would correspond with facial features such as corners of the eyes and mouth for instance. Additional dots can also be drawn on the face to aid detection, creating a hybrid of the marker-based and markerless systems. Feature extraction is generally done using convolutional neural networks (CNN) [17]. As with other methods, chosen features from the data should represent the pivotal points of the human body so that their motion can be easily fitted on the character rig.

Cameras utilized in markerless motion capture can take RGB, grayscale, depth, or even infrared images [17]. The number of cameras used varies and affects the acquired data's quality, just like with optical marker-based systems. However, lesser amount of equipment also reduces hardware costs and gives the performers much more freedom [2]. This makes it easier to use motion capture in everyday consumer products. Additionally, facial motion capturing is easier, because physical sensors or markers on a performer's face can be considered intrusive.

Computer vision algorithms are also capable of detecting greater detail from an image than just markers, which makes it possible to utilize motion capture in new innovative ways. An example of this is 4D performance capture (4DPC) that can be used to reconstruct the surface mesh of a moving 3D model [18]. Markerless systems can even capture motion from objects that physical markers cannot be placed on. Topic subject to active research is the automation of the mesh modelling and animation process with the help of 3D reconstruction [11].

However, human pose estimation, especially the monocular kind, comes with great challenges. These issues include image disturbances, viewpoint changes, depth loss and difficulty of determining the complex structure of the human body. [17] The effect of depth loss can be lessened by using more cameras or depth images, but the other issues still persist. Estimating the accurate location of the chosen features from video footage is a hard task, but machine learning has become significantly more accurate over the years. The future of motion capture will lead to more wide use optical markerless systems due to their high potential. Although, they still have ways to go before they can replace any of the more traditional systems.

4. CHALLENGES

Previous chapters explained the fundamentals of 3D models and motion capture, but additional actions need to be taken to combine them for animation. This 3D animation production pipeline consists of three main stages of data processing: acquirement, cleaning, and fitting the data to the character rig [4, p. 226]. Each stage comes with its specific set of challenges that make the utilization of motion capture in 3D animation more difficult. These problem areas are subject to active research and development in an attempt to perfect the technology. This following chapter will address the most common challenges found in the production pipeline while paying attention to the significance of real-time usage and automation.

4.1 Accessibility

From the descriptions of different motion capture methods it can be concluded that accessibility is a prevailing issue. Equipment and software required are complicated to use and expensive, often restricted to indoor use. Markers, special equipment, and calibration require extensive preparation time. [19] This is not a major issue for big film productions, but challenges many smaller productions and consumer products. Average person that wants to use an animated virtual avatar, for instance, cannot realistically acquire and use such equipment. Accessibility varies heavily between different motion capture methods and some have more future development potential than others. Accessibility as a concept covers the usability of the hardware and software as well as their monetary costs.

What most motion capture methods share in common with each other is being expensive. Not only due to the equipment, but the system may also need professionals to maintain and use it. If the desired motion capture system is too expensive and the production budget does not allow it, a company will have to consider outsourcing their work. If motion capture is used often, it is generally worth the purchase. Buying a system is an investment, which will have to be assessed on a case-by-case basis. [5] However, many of the more complex setups are still barred from being used in consumer applications. Cost is not a major issue when creating animation for films and other high budget projects, so expensive optical marker-based optical systems are often used for their high-quality results. Prices can go up to tens or even hundreds of thousands of dollars if many cameras are used. Non-optical systems therefore still see development despite their

lower quality of data, because they are considered to be very cost-effective [15]. Low-cost solutions aim to reduce the amount of equipment needed, but this often comes as a trade-off to quality. Many of these solutions utilize inertial sensors or simple markerless setups with fewer cameras or scanners for instance. Magnetic systems are also relatively inexpensive [3]. Quality improvement is an active research topic for such systems. Expensive systems will always be desirable, but low-cost solutions will also see popularity in the future due to being able to be used with more freedom.

Hardware requirements along with the price determine what is and what is not accessible to the masses. The amount of equipment as well as the expertise required to use the system and maintain it are a common issue. Especially marker-based optical systems are complex, but inertial systems are from the simplest end, corresponding with the matter of price. However, development in monocular markerless motion capture ensures that optical systems can be used to certain degree with even a simple setup of a singular webcam for instance. Markerless technology also has another advantage in accessibility as wearing markers on one’s body can be considered intrusive and difficult. [17] Especially mechanical systems are restrictive to the performer when it comes to freedom of movement so markerless systems are preferred.

Many of the systems also require lots of equipment laid out in a carefully calibrated environment. Optical, magnetic, and acoustic systems' quality of data easily suffers from external interferences. Optical systems in particular are not yet suitable for large-scale outdoor scenes [20]. Portability is therefore more applicable to inertial and mechanical systems, which can be used in any environment. Range of use is also large, unlike with optical systems where the performer has to be in front of the stationary cameras at all times while making sure that the markers do not become occluded.

There is major interest in reducing the amount of equipment needed to capture motion and this is possible with non-optical systems as well. Developments have been made to reduce the number of sensors as statistical analysis can make up for the lesser amount of data inputs. Attempts have been made so that full body motion can be fetched from a database or even directly synthesized with adequate results based on the input of a single IMU. [19]

Along with the equipment also software is required, divided into acquisition and post-processing software. Acquisition software can, for instance, perform camera calibration in the case of optical systems while post-processing software prepares the acquired data for use. [5] The more automated this whole process is, the more accessible the motion capture is as processing data can require both time and expertise. Additionally, lack of

extensive post-processing makes it possible to use motion capture in real-time applications. Direct acquisition refers to these systems that do not require any post-processing, such as magnetic, acoustic, and mechanical systems. As a downside these systems are generally more obtrusive to the performer and have lower sampling rates, which in turn affect the data's quality. Optical systems on the other hand require extensive amount of post-processing with a high camera count, but produce precise data. [3][12] Regardless of the system used, all of them require at least some processing, such as cleaning, fitting the data and solving the posture to be used in animation. If real-time usability is required, quality might have to be compromised by cutting down the post-processing time.

The ideal motion capture system would therefore be as simple and cheap as possible without leading to compromises in quality. Research has been put into IMUs, for instance, because of their low cost and simplicity [19]. As optical systems require cameras and are sensitive to lighting, non-optical systems are driving the expansion in portable motion capture [15]. If lack of portability and sensitivity to interferences is not an issue, using computer vision for feature detection in markerless systems provides a relatively easy way to capture motion, especially in monocular cases. These two systems in particular are perfect for accessible products and any developments in quality improvement will be an added bonus. This in particular is a major challenge developers face. In some cases, such as film productions, lack of accessibility is not a major issue so quality can be prioritized. Due to this there are benefits in different systems having different market niches.

4.2 Accuracy and cleaning

The goal of motion capture is to determine the specific location of chosen features in a 3D space over a sequence of time. However, due to the equipment's natural inaccuracies and interferences of the environment, the acquired data is not a perfect one-to-one representation of reality. Even with expensive high-fidelity equipment there will inevitably be errors, noise and outliers that lower the data's quality [21]. Quality is determined with perceptual metrics so there is not much objectivity to it. What level of quality is acceptable for a specific application is decided on a case-by-case basis. Human visual perception is very sensitive even to the most minor distortions so quality is a metric that should not be completely ignored [20].

If the acquired data's quality is not good enough as is, the unavoidable challenge of using motion capture is having to clean the data. Optical marker-based systems have higher accuracy than their competition, but only after extensive post-processing. No motion capture method is immune to corrupted data, but the nature of the dominant errors can differ

between them [4, p. 223]. Regardless of the system used, post-processing of the data can be costly and time-consuming so preventing noise with the right equipment and a setup is a priority [4, p. 224]. Cleaning is a process that can be done by hand, but large parts of it can be automated thanks to developments in computer software. A major motivation is to make cleaning as fast as possible so that the data can be used in real-time applications. Without automation this would not be possible and manual cleaning also includes the possibility of user induced errors [21][22]. Data can technically be used without cleaning, but the perceived quality of the animation in this case tends to be very poor.

High sampling rate, also known as framerate, helps the cleaning process and increases the data's initial quality. In the case of temporal filters, for instance, average values are found easier with higher framerates. [5] This also leaves room for downsampling, which can be used to smoothen out errors [4, p. 223]. Additionally it makes it easier to do various operations with filters, both for cleaning noise and applying visual effects [12]. As a downside, higher frame count takes more storage space and involves more processing. Animation is generally delivered at framerates of 30 or 24 frames per-second or lower, so data naturally needs to be sampled at a higher rate [4, p. 223][12]. Starting at a high frame count makes it easier to determine what the delivery rate will be. Optical systems are often favoured over non-optical systems due to their high sampling rates [2]. However, they in particular have a unique set of issues that other systems do not generally have. The following chapter covers common types of errors and methods used to clean the data, starting with the ones typical to optical systems.

These issues concern problems with identifying the trackable features. Each of the data points needs to have a unique identity, in other words a label. This label is consistent as the point moves around in a 3D space over a sequence of time. This is necessary so that the data can be used to control a character rig. The data points firstly need to know their spatial relationship to their peers and secondly be connected to specific segments of the 3D model. Optical systems commonly have issues that cause problems with the tracking of specific features and their identities: occlusion and confusion.

Occlusion is a problem where cameras require a clear line of sight to the features they are tracking. If the feature is blocked momentarily, it results in a gap in the data point's trajectory. [5] Marker identity confusion on the other hand is most commonly caused by rapid occlusion of features the cameras are trying to track [4, p. 224]. Especially in the case of multiple performers interacting closely, the system may have issues trying to track and identify several sets of markers at once [22]. Active markers can solve this problem by assigning each LED-marker a distinct identity. With passive markers and markerless systems, however, data points can mistakenly switch with each other. This

mislabelling is a typical issue for optical features that have no automatic way of identifying themselves from their peers [4, p. 224][21]. Due to reliance on sight, even lighting can affect the quality of the acquired data. Markers can be misinterpreted and markerless systems especially might have trouble recognizing the correct features under certain conditions.

To clean the data, trajectories are reconstructed. Any gaps caused by occlusions are filled and the labels are corrected if mislabelled. Gaps in the data can also be caused by reasons other than occlusion, like after some erroneous data points have been removed in another part of the cleaning process [8]. To fill the gaps, the placement of missing data points can for instance be interpolated or calculated with rigid body math. This refers to placing four or more markers in a rigid formation with each other so that if one goes missing, its location can be determined based on its peers. [5] Software generally has the ability to correctly fill any gaps in the data by guessing where the missing points should be. As with IMUs, which include complementary sensors, additional technologies can be combined with optical motion capture systems to improve the quality of the acquired data in the cases of occlusion and confusion. Other ways exist too, such as data-driven methods. However, in complicated cases with multiple performers it is still a common practice to have specialists manually fix errors in the data. Automatic detection of errors will only become more desirable as markerless systems see more use. [22]

Aforementioned issues are more specific to optical systems, but other systems are also easily susceptible to data corruption as well. Issues with camera or sensor calibration are present across all systems [4, p. 223]. Magnetic and inertial systems especially suffer from sensor noise, output drifting and environmental disturbances [21]. Across all systems most common types of data corruption include high-frequency noise and trajectory spikes. Spikes refer to drastic and brief changes in values. These can be crudely removed, but it will cause a gap in the data that needs to be filled. Shaking will also inevitably occur, which refers to data points moving slightly despite the performer trying to stand still. This can be fixed by removing all the shaking data and interpolating the created gap. [8] Standstill reference points can be used as well. This, however, still leaves in the issue of noise.

Many algorithms to denoise motion capture data exist, such as matrix low rank filling and data-driven approaches. The common way to denoise data and smoothen out noise is to follow a filtering strategy. [23] As with filters generally, they should be used carefully to avoid removing details from the original data. Too much filtering and the motion stops looking realistic and nuanced, but when done carefully, it has great benefits. Gyroscope data for instance is often high-pass filtered to reduce the effects of drifting in inertial

systems [15]. Noise generally exists in high frequencies so low-pass filters like Gaussian or Kalman filter networks to sequentially filter noise are popular. Both methods can successfully be used in real-time applications as well. [21]

However, filtering is bad at retaining spatial characteristics natural to human motion. Each degree of freedom of the joints is related to each other, but filters are used on them separately. Methods to filter noise with data-driven techniques in an attempt to retain spatial information therefore have been developed. Goal is to learn spatial-temporal patterns of human motion with robust statistics to filter noise, outliers and fill in missing values. [21] Data-driven approaches naturally require large databases for training purposes, but in recent years such resources have become available [22]. Reliance on the sample variety of the database is still a problem so room for improvement exists. Despite this they have become the mainstream choice for denoising motion capture data, mostly due to their ability to understand the complexity and nuances of human motion. [23]

Complete prevention of errors in the data is not possible, but attempts should be made to minimize their effect. If markers are used, they should be placed rigidly near the performer's joints so that they do not move. Markers should also be kept as far away as possible from each other so that their rotation in relation to each other is clearer. [5][8] Additionally, distancing the markers from each other prevents mislabelling while using more cameras prevents occlusion. Any environmental disturbances should be removed to alleviate noise, such as lighting that could disturb the cameras. With magnetic systems anything that could interfere with the sensors, causing electrical or magnetic disturbances, needs to be removed [13]. Metallic beams underneath the floor for instance can make it more difficult to determine foot-to-floor contact [4, p. 221-222]. Preventing as many errors as possible can save a lot of trouble in post-processing.

In conclusion, while some issues can be avoided by changing the setup, some amount of noise and errors is still inevitable. For that reason the data needs to be cleaned in post-processing. Methods to filter data while retaining all the nuances of human motion are highly desirable. Processing, however, limits the ability to use the data for real-time animation unless some amount of delay or lower quality data is acceptable.

4.3 Fitting and retargeting the data to a model

Once the data has been captured and cleaned, the points have to be fitted to a 3D model's skeleton to produce animation. Ideally the results would not have to be modified after the solving algorithm has been applied, but manual adjustments are often required for best-looking animation. The data fitting process comes with many issues to consider.

Fitting differs based on what motion capture data format is being used. Body parts in the data are segmented and stored separately in vectors, but varying file and storage formats naturally exist [3]. Data points themselves can be purely translational or also include rotational and hierarchical aspects. Translational data only includes the locations of the markers and is commonly acquired with optical systems. It is possible to solve rotational information from this data as well, but it requires extra computing. A combination of translational and rotational data can be directly acquired with magnetic systems for instance. Rotational data makes it easier to solve the motion on the skeleton directly, but purely translational data allows the creation of more complex character setups. One format, a tier higher in terms of complexity, includes hierarchical information and can be referred to as skeletal data. A skeleton has been fitted to this data, so it already has all the information necessary to produce animation. [8][12] However, the acquired data requires some computation to get to this point.

Solving algorithm's job in this case is to determine the rotation and translation of the skeleton's joints based on the relative positions of its associated motion capture data points [4, p. 224-225]. In essence, a link between the data and joints is established. The software knows which data points are associated with which joints with the help of labels and body part segmentation. Many factors have to be considered in this process. For instance, the local rotation axes of joints have to be identically oriented with the rotation data and the rotation order between them has to match in order to avoid the possible ambiguity of Euler angles [8]. In addition, challenges arise from solving proportion differences as well as from finding the most efficient method for computation. Fitting therefore is a complicated process.

The data can be applied to the model's skeleton directly or indirectly via a mediator skeleton [8]. Many pieces of animation software are implemented so that the skeleton is automatically animated if rotational data is applied on it. With translational data an internal skeleton will have to be constructed for the data points beforehand. [12] Various different computation methods to fit data points to joints exist and the best choice is determined on a case-by-case basis. In addition to data format differences, the relationship between joints and data points is never the same. This relationship can be influenced by joint type, suit slide, muscle bulge, skin stretch, joint's degrees of freedom and many more factors. Due to this variety a huge number of algorithms exist. A rigid body solver, for instance, determines joint rotations from a rigid formation of three or more data points, a method often used for translational data acquired with optical systems. [5] Regardless of the method chosen, a character rig can be fully controlled with even a very small amount of data points.

This process is possible with the limited amount of data points, because the skeleton's bones are rigid objects connected to each other by rotating joints [4, p. 224-225][17]. This consistency of proportions and bone lengths allows the solver to find a best-fit solution between the data and the skeleton easier. Constraints related to translation and degrees of freedom for joints still apply. Real human bodies, however, cannot be perfectly modelled as a rigid chain of joints. Spines provide more complex movement and joints allow some amount of laxity in all directions. [12][14] This is something to consider with the acquired motion capture data that may be precise enough to capture such movements.

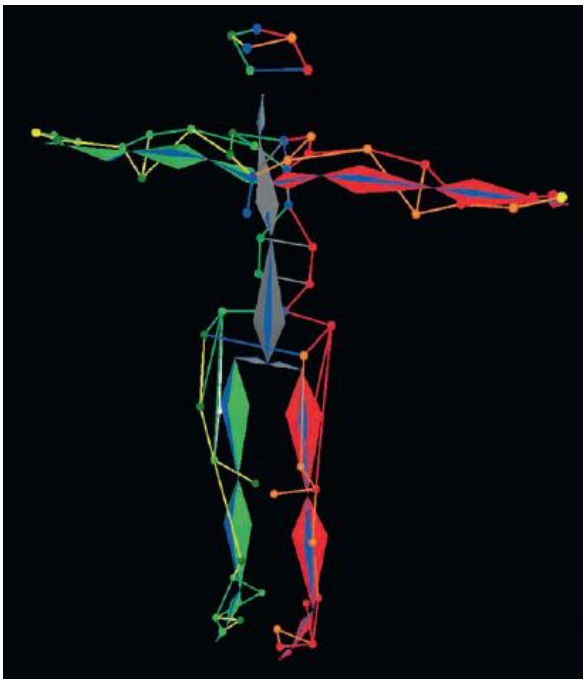

Figure 7. *Skeleton fitted to motion capture data points using Vicon's Blade [5].*

In addition to fitting the data points to the skeleton, the floor constraint is also determined. This makes it so that the model collides with the floor of the scene and does not float or pass through it. However, with the collision it is more difficult to make the animation look natural and it also makes the calculations more complex. [4, p. 225] Like with matching rotations between the data and the model, the translational relationship with the camera and world's coordinate systems needs to be accounted for as well. If the goal is to track the motion of the face only, the data needs to be stabilized to account for the motion of the performer's head. For solving there needs to be a stationary reference point that the data points move in relation to. [5] Additional issues may also arise from when multiple characters interact in one scene and their collisions need to be determined. Knowing the exact relationship between the different coordinate systems is therefore critical.

What makes this process even more difficult is the difference between the performer whose motion was tracked and the character rig. Not only are there differences in scale and length of the bones, but also in joint configurations [4, p. 227]. With motion retargeting it is possible, for instance, to adapt the motion data of an adult to a child. Modifying the data increases its reusability and makes it possible to achieve more realistic results. [24] There is no single best way to do retargeting as it depends heavily on what kind of motion is being worked on [8]. The motion is retargeted by establishing a relationship between the skeleton of the 3D model and a skeleton specifically fitted to the motion capture data. A rotation of a single joint passed from one skeleton to the other is less susceptible to the effects of scale and proportion differences, so the use of mediator skeletons in this case helps. [4, p. 227] Numerical IK solvers are a popular choice but have often require manual adjustments due to appearing unnatural. So that solvers would have the knowledge of actual human movements, deep learning solutions are being actively researched. [24]

Another issue with motion targeting is related to the mesh of the character model. Generally it is assumed that the shape of the mesh and the pose of the skeleton are two separate elements. Shape is determined purely based on the skeleton's pose and a skinning algorithm. This, however, does not account for how motion of a skinny character can be applied on a fat character. Mesh collisions in this case would cause clipping and artifacts. In the reverse case of fat data being applied on a skinny character, this would cause different parts of the mesh not to touch each other despite this being the intention. Skeletal pose and mesh shape therefore cannot be treated independently in the case of motion retargeting. Referred to as skeletal motion retargeting, it deforms the mesh purely based on the skeleton's pose and requires manual adjustments. Surface mesh retargeting on the other hand can help to make the deformations seem more natural. [25] Incorrect contact with the environment and other characters can become an issue as well [22]. Collisions should therefore be adjusted accordingly.

To summarize, solving and retargeting requires computation that limits its uses in real time, just like with other parts of the production pipeline. High quality results are inevitably associated with heavy computation and complexity. Real-time computation meanwhile places approximations and limitations on the results. Especially passive optical systems are not suited for it. [5] As apparent, the problem of perceptual quality applies to all stages of the pipeline and the 3D models themselves.

5. CONCLUSIONS

This thesis provided a general overview of what using motion capture in 3D animation entails, along with its possibilities and challenges. As a general rule it was noted how high perceptual level of quality inevitably increases the cost and complexity of a system. Quality and real-time usability's importance should therefore be determined when deciding what methods to use for a specific application. Due to the variety in systems and processing methods, there is almost bound to be something any application can use on a satisfactory level.

The major goal of motion capture technology is to make systems both precise and accessible. Some of this development has been accomplished with advancements in technology, such as machine learning solutions. Efficiency of motion capture technology is also rising, making it more and more appealing to use in a wide variety of applications. This makes it clear that motion capture's popularity is only going to rise over the coming years and the highest potential for expansion lies in consumer products. This increases the importance of accessibility to an even higher level. Additionally, future developments will most certainly involve the use of machine learning in several stages of the production pipeline. This is will therefore not only improve markerless motion capture that relies on it in the acquirement process, but it will also create new possibilities for motion capture and 3D animation in general.

Due to the work covering a very wide area, details about many of the technologies were left on a general level without delving into specifics. While motion capture can be used in 2D animation as well, it was considered out of scope for this work. As a general overview the work provides useful material to anyone who wishes to start using motion capture in their own applications.

REFERENCES

[1] J. Wang, S. Drucker, M. Agrawala, M. Cohen, The Cartoon Animation Filter, ACM Transactions on Graphics, 25.3, 2006, pp. 1169-1173

[2] L. Quanzhi, Research on Animation and Its Motion Capture Technology, Artificial Intelligence and Communications, International Conference on Data Processing, 2018

[3] P. Nogueira, Motion Capture Fundamentals A Critical and Comparative Analysis on Real-World Applications, 4th International Conference on Information Society and Technology, 2011

[4] R. O'Neill, Digital Character Development: Theory and Practice, CRC Press, 2nd ed., 2016

[5] J. Okun, S. Zwerman, Performance and Motion Capture, Visual Effects Society Handbook, Routledge, 2010, pp. 361-412

[6] R. Mukundan, Advanced Methods in Computer Graphics with Examples in OpenGL, Springer London, 1st ed., 2012

[7] L. Kavan, J. Žára, Real Time Skin Deformation with Bones Blending, WSCG ShortPapers Proceedings, 2003

[8] M. Kitagawa, B. Windsor, MoCap for Artists Workflow and Techniques for Motion Capture, Elsevier/Focal Press, 1st ed., 2008

[9] A. Aristidou, J. Lasenby, FABRIK: A Fast, Iterative Solver for the Inverse Kinematics Problem, Graphical Models, 73.5, 2011, pp. 243-260

[10] N. A. Rumman, M. Fratarcangeli, Skin Deformation Methods for Interactive Character Animation, Computer Vision, Imaging and Computer Graphics Theory and Applications, Springer International Publishing, 2017, pp. 153-174

[11] P. Fechteler, A. Hilsmann, P. Eisert, Markerless Multiview Motion Capture with 3D Shape Model Adaptation, Computer Graphics Forum 38.6, 2019, pp. 91-109

[12] A. Menache, Understanding Motion Capture for Computer Animation, San Francisco: Elsevier Science & Technology, 2011

[13] Q. Fu, X. Zhang, J. Xu, H. Zhang, Capture of 3D Human Motion Pose in Virtual Reality Based on Video Recognition, Complexity (New York, N.Y.), 2020

[14] D. Roetenberg, H. Luinge, P. Slycke, Xsens MVN: Full 6DOF Human Motion Tracking Using Miniature Inertial Sensors, Xsens Motion Technologies BV, Tech. Rep., 2009

[15] G. Shi, Y. Wang, S. Li, Development of Human Motion Capture System Based on Inertial Sensors 2125, Sensors & transducers 173.6, 2014, pp. 90-97

[16] I. Failes, What Mocap Suit Suits You?, VFX Voice, 2019 [Online] (Accessed 29.4.2021): https://www.vfxvoice.com/what-mocap-suit-suits-you/

[17] W. Gong, X. Zhang, J. Gonzàlez, A. Sobral, T. Bouwmans, C. Tu, E. Zahzah, Human Pose Estimation from Monocular Images: A Comprehensive Survey, Sensors (Basel, Switzerland) 16.12, 2016

[18] P. Huang, M. Tejera, J. Collomosse, A. Hilton, Hybrid-Skeletal-Surface Motion Graphs for Character Animation from 4D Performance Capture, ACM Transactions of Graphics 34.2, 2015, pp. 1-14

[19] C. Mousas, Full-Body Locomotion Reconstruction of Virtual Characters Using a Single Inertial Measurement Unit, Sensors (Basel, Switzerland), 17.11, 2017, p. 2589-

[20] X. Shihong, G. Lin, L. Yu-Kun, Y. Ming-Ze, C. Jinxiang, A Survey on Human Performance Capture and Animation, Journal of Computer Science and Technology, 32.3, 2017, pp. 536-554

[21] L. Hui, C. Jinxiang, Example-Based Human Motion Denoising, IEEE Transactions on Visualization and Computer Graphics, 16.5, 2010, pp. 870-879

[22] A. Aristidou, D. Cohen-Or, J.K. Hodgins, A. Shamir, Self-similarity Analysis for Motion Capture Cleaning, Computer Graphics Forum, 37.2, 2018, pp. 297-309

[23] Y. Zhu, Denoising Method of Motion Capture Data Based on Neural Network, Journal of Physics. Conference Series, 1650.3, 2020

[24] S. Uk Kim, H. Jang, J. Kim, A Variational U-Net for Motion Retargeting, Computer Animation and Virtual Worlds, 31.4-5, 2020

[25] J. Basset, S. Wuhrer, E. Boyer, F. Multon, Contact Preserving Shape Transfer for Rigging-Free Motion Retargeting, Motion Interaction and Games, 2019, pp. 1-10