

# Motion Capture für die Masse

Eine Analyse preiswerter Motion Capture Systeme im  
Vergleich zum Industriestandard

## Diplomarbeit

Ausgeführt zum Zweck der Erlangung des akademischen Grades  
**Dipl.-Ing. für technisch-wissenschaftliche Berufe**

am Masterstudiengang Digitale Medientechnologien an der  
Fachhochschule St. Pölten, **Digitale Medientechnologien**  
**[Postproduktion]**

von:

**[Martin Konrad, BSc]**

[dm161524]

Betreuer/in und Erstbegutachter/in: [Mag. Franz Schubert]

Zweitbegutachter/in: [Dipl. -Ing. (FH) Mario Zeller]

[Linz, 12.01.2019]

# Ehrenwörtliche Erklärung

Ich versichere, dass

- ich diese Arbeit selbständig verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und mich auch sonst keiner unerlaubten Hilfe bedient habe.

- ich dieses Thema bisher weder im Inland noch im Ausland einem Begutachter/einer Begutachterin zur Beurteilung oder in irgendeiner Form als Prüfungsarbeit vorgelegt habe.

Diese Arbeit stimmt mit der vom Begutachter bzw. der Begutachterin beurteilten Arbeit überein.

.....

Ort, Datum

.....

Unterschrift

# Kurzfassung

Die vorliegende Diplomarbeit vergleicht zwei preiswerte Motion Capture Systeme für die Herstellung von 3D-Charakter Animationen und deren Unterschiede zur Performance des Industriestandards.

Die ersten beiden Abschnitte verleihen einen groben Überblick über die Grundprinzipien der Charakter Animation sowie die unterschiedlichen Methoden zur Bewegungserfassung, mit ihren jeweiligen Einsatzgebieten. Im Hauptteil werden die Günstig-Systeme ganzheitlich analysiert und daraufhin simultan, unter kontrollierten Produktionsbedingungen, getestet. Die erhaltenen Daten werden gegeneinander sowie mit Motion Capture Animationen eines State-of-the-Art-Systems verglichen.

Die Ergebnisse fließen anschließend in eine Nutzwertanalyse ein, welche die untersuchten Systeme über ein Punktemodell bewertet. Die vorliegenden Ergebnisse definieren den Eignungswert der Systeme und liefern einen Anhaltspunkt für zukünftige Projektplanungen.

Aus Kostengründen wurde, stellvertretend für die Performance des Industriestandards, auf Animationen einer Onlinedatenbank referenziert. Deshalb basieren vereinzelte Ergebnisse auf den subjektiven Interpretationen des Autors.

Nur eines der beiden preiswerten Motion Capture Systeme kann dem Vergleich zum Industriestandard annähernd standhalten. Es hat jedoch das am besten bewertete Preis-Leistungs-Verhältnis. Das zweite Günstig-System kann die Anforderungen einer professionellen Produktionsumgebung nicht erfüllen.

# Abstract

The present thesis compares two affordable Motion Capture systems for use in 3D-Character Animation, and analyses their performance differences to the industry standard.

The first two sections provide a brief overview of the basics of Character Animation as well as the different techniques of Motion Capturing and their related fields of use. In the main section, the low-priced systems are holistically analysed and simultaneously tested under the conditions of a controlled production environment. The collected data is then compared against each other and furthermore a state-of-the-art-system.

The conclusions are later integrated into a use-value analysis, which evaluates the systems through a scoring model. The obtained results define the systems suitability and offer a reference point for upcoming projects in the industry.

Due to economic reasons, animations of an online Motion Capture database perform substitutionally for the industry standard. Hence, isolated results are based on subjective interpretations of the author.

Just one tested system can barely compete against the performance of the industry standard, it has the highest ranked cost effectiveness though. The second affordable system does not meet the requirements of a professional production environment.

# Inhaltsverzeichnis

<b>Ehrenwörtliche Erklärung</b>	<b>II</b>
<b>Kurzfassung</b>	<b>III</b>
<b>Abstract</b>	<b>IV</b>
<b>Inhaltsverzeichnis</b>	<b>V</b>
<b>1 Einleitung</b>	<b>8</b>
1.1 Problemstellung	9
1.2 Motivation	9
1.3 Forschungsziele/-Fragen	10
1.4 Methodik	10
1.5 Gliederung der Arbeit	11
<b>2 Grundlagen der Charakter Animation</b>	<b>12</b>
2.1 Zeichentrickfilm	12
2.2 Keyframe Animation	13
2.2.1 Squash & Stretch	13
2.2.2 Anticipation	13
2.2.3 Staging	13
2.2.4 Straight Ahead & Pose-to-Pose	14
2.2.5 Follow through & Overlapping Action	14
2.2.6 Secondary Action	14
2.2.7 Timing	14
2.2.8 Slow In & Slow Out	15
2.2.9 Arcs	15
2.2.10 Exaggeration	15
2.2.11 Solid Drawing	15
2.2.12 Appeal	15
2.3 Digitale Charaktere	16
2.3.1 Rigging	16
2.3.2 Kinematik	17
2.3.3 Keyframing	18
2.3.4 Motion Graph	18
<b>3 Grundlagen des Motion Capturing</b>	<b>19</b>
3.1 Einführung	19
3.2 Motion Capture Systeme	20
3.2.1 Optisches Motion Capture	20
3.2.2 Radiofrequenz Positionssysteme	31
3.2.3 Elektromagnetische Tracker	32

3.2.4	Inertial Motion Capture Anzüge	33
3.3	Konfigurationen	39
3.3.1	Hand Motion Capture	39
3.3.2	Facial Motion Capture	40
3.4	Anwendung in der Unterhaltungsindustrie	42
3.4.1	Gaming	42
3.4.2	Virtual Reality	45
3.4.3	Spielfilm	46
3.5	Datenformate	49
3.5.1	C3D	49
3.5.2	ASF/AMC	50
3.5.3	BVH	52
3.5.4	FBX	53
3.6	Planung	54
3.6.1	Script	54
3.6.2	Storyboard	54
3.6.3	Shotlist	55
3.7	Nachbearbeitung	56
3.7.1	Data Cleanup und Post Processing	56
3.7.2	Retargeting	58
<b>4</b>	<b>Praktische Analyse</b>	<b>60</b>
4.1	Einführung	60
4.2	Vicon – Markerbasiertes Motion Capture	61
4.2.1	Hardware	61
4.2.2	Setup	64
4.2.3	Kalibrierung	66
4.2.4	Softwareaufbau	67
4.3	Perception Neuron – Inertiales Motion Capture	71
4.3.1	Hardware	71
4.3.2	Setup	73
4.3.3	Kalibrierung und Wartung der Sensoren	75
4.3.4	Softwareaufbau und Feineinstellungen	77
4.4	iPi Soft – Markerloses Motion Capture	80
4.4.1	Hardware	81
4.4.2	Setup	82
4.4.3	Kalibrierung	83
4.4.4	Softwareaufbau und Feineinstellung	84
4.5	Durchführung	87
4.5.1	Kriterien	87
4.5.2	Szeneninhalt	88
4.5.3	iPi Soft Setup	91

4.5.4	Perception Neuron Setup	94
4.6	Zusammenfassung	98
4.6.1	Vicon	98
4.6.2	Perception Neuron	99
4.6.3	iPi Soft	102
<b>5</b>	<b>Nutzwertanalyse</b>	<b>105</b>
5.1	Einsatzbereich	105
5.2	Zielsetzung	105
5.3	Grundregeln	106
5.4	Kriterien	106
5.4.1	Bestimmung	106
5.4.2	Gewichtung	107
5.4.3	Bewertung	110
5.5	Ergebnisse	112
<b>6</b>	<b>Zukünftige Entwicklungen</b>	<b>115</b>
6.1	Deep Learning in neuronalen Netzwerken	115
6.2	Quantencomputer	119
<b>7</b>	<b>Fazit</b>	<b>121</b>
<b>8</b>	<b>Glossar</b>	<b>123</b>
<b>9</b>	<b>Anhang</b>	<b>127</b>
9.1	Kriterienbewertung	127
	<b>Literaturverzeichnis</b>	<b>132</b>
	<b>Abbildungsverzeichnis</b>	<b>137</b>
	<b>Tabellenverzeichnis</b>	<b>140</b>

# 1 Einleitung

Motion Capturing (MoCap) beschreibt eine präzise Methode zur Erfassung realer Bewegungsabläufe, welche anschließend digitalisiert und in eine 3D-Computeranimation gewandelt werden. (Kitagawa & Windsor, 2008, S. xiii)

Seit dem Aufschwung optischer, markerbasierter Systeme in den 1980er Jahren, hat sich die Firma *Vicon Motion Systems* ([www.vicon.com](http://www.vicon.com)) als Markt- und Innovationsführer im Bereich der hochpräzisen Bewegungserfassung etabliert. Die Systeme liefern überragende Genauigkeit in Relation zu immens hohen Investitionskosten (ab 72.000 € bis mehrere Mio. €). (Berger, 2016, S. 14), ([www.prophysics.ch](http://www.prophysics.ch))

Es gibt mittlerweile weitaus günstiger vermarktete Systeme (ab <1.000 €), welche sich gezielt an professionelle Produktionsumgebungen mit begrenztem Budget richten. Die Herangehensweisen zum Erfassen der Bewegungsdaten sind dabei teilweise völlig unterschiedlich.

Der **Industriestandard** arbeitet mit auf dem Objekt angebrachten, hochreflektierenden Markern, die von mehreren Infrarotkameras erfasst und anschließend in digitale Koordinaten konvertiert werden.

**Markerlose Systeme** errechnen die Bewegung des Objekts entweder über Mustererkennung, Silhouettracking, Triangulation, Radiowellen, Tiefeninformation oder einer Kombination der Techniken.

**Inertiale MoCap Anzüge** errechnen die translatorischen und rotatorischen Bewegungsanteile mehrerer, getrennter Sensoren in einem selbsterzeugten Referenzmagnetfeld, und verzichten dabei ebenfalls auf reflektierende Marker sowie Kameras.

Die Arbeit richtet sich an jene Zielgruppen, welche auf der Suche nach preiswerten Alternativen zur Anwendung von Motion Capturing, die Qualifikation der unterschiedlichen Techniken für ihre persönlichen Einsatzzwecke prüfen wollen.



### 1.1 Problemstellung

Das oberste Ziel ist eine präzise und gleichzeitig preiswerte sowie unkomplizierte Bewegungserfassung. Herausragende Präzision kann durch markerbasierte State-of-the-Art Systeme (*Vicon Motion Systems*) bereits erfüllt werden. Der finanzielle und organisatorische Aufwand ist jedoch alles andere außer massentauglich.

Motion Capturing auf Produktionsniveau bleibt damit einem zahlungskräftigen Anwenderkreis vorbehalten. Der rasante Fortschritt markerloser Motion Capture Systeme bringt gegenwärtig revolutionäre Forschungsprojekte und kommerzielle Produkte hervor, welche professionelle Ergebnisse zu einem hervorragenden Preis-Leistungs-Verhältnis versprechen.

Gegenwärtige Forschungen fokussieren sich allerdings hauptsächlich auf die Präzision der Daten. Stichhaltige Aussagen über die Gesamtpformance besagter Günstig-Systeme unter produktionsnahen Bedingungen, sind aufgrund der geringen Forschungsgrundlage nur bedingt möglich.

### 1.2 Motivation

Professionelle Character-Animation ist ein zeit- und kostenintensiver Prozess. In einigen Fällen lässt sich diese effizienter mit Motion Capturing umsetzen. Optische State-of-the-Art Systeme sind erst ab einem Preisniveau von mindestens 72.000 € erhältlich, was den finanziellen Rahmen klein- bis mittelgroßer Unternehmen oder schulischer Einrichtungen womöglich übersteigt.

Konstant anwachsende Multimediainhalte machen Motion Capturing zu einem gefragten Werkzeug zur Beschleunigung von Animationspipelines. Deshalb werden Alternativen vorangetrieben, welche die Kosten drastisch reduzieren und gleichzeitig keine Abstriche gegenüber der Präzision in Kauf nehmen sollen.

Die Leistung eines Motion Capture Systems wird nicht ausschließlich durch seine Präzision bestimmt. Deshalb vergleicht diese Arbeit, unter produktionsnahen Bedingungen, die Gesamtpformance zweier preiswerter Systeme und prüft, inwieweit diese in eine professionelle Produktionsumgebung integriert werden können. Die Ergebnisse sollen als Entscheidungsgrundlage für zukünftige Projektplanungen in schulischen Einrichtungen sowie Unternehmen in der Multimediabranche dienen.

*„Any technology that makes things **Better, Faster or Cheaper** will eventually **become standard.**“ (Price, 2018)*

### 1.3 Forschungsziele/-Fragen

Die Arbeit widmet sich die Frage, welches der beiden preiswerten Systeme unter denselben kontrollierten Bedingungen die bessere Gesamtperformance aufweist.

Zudem sollen die Leistungs differenzen zwischen preiswerten Motion Capture Systemen und dem Industriestandard eruiert werden.

Motion Capture Daten werden nur selten direkt weiterverwendet und unterlaufen für gewöhnlich einige Nachbearbeitungsschritte. Anhand der Rohdaten kann festgestellt werden, welcher Aufwand nötig wird, um ein professionelles Endergebnis zu erhalten.

Darauf aufbauend wird geklärt, wie weit ein preiswertes Motion Capture System für ein professionelles Produktionsumfeld geeignet ist und ob dessen Leistungsumfang die gängigsten Aufnahmeszenarien abdeckt.

Nutzwertanalysen sind ein nützliches, betriebswirtschaftliches Instrument, um nicht quantifizierbare Größen in eine aussagekräftige Bewertung einfließen zu lassen. Das dazu verwendete Punktesystem soll die in der Arbeit erforschten Systeme ihrer Gesamtperformance nach reihen und gleichzeitig deren Nutzwert ermitteln.

Das erhaltene Leistungsprofil gibt Auskunft darüber, ob sich preiswerte Motion Capture Systeme für Bildungseinrichtungen sowie klein- bis mittelständische Unternehmen qualifizieren können.

### 1.4 Methodik

Eine ausführliche Literaturrecherche gibt Aufschluss über technische Grundprinzipien sowie aktuelle Projekte im Bereich der Bewegungserfassung. Sie ermittelt zudem die wichtigsten Anforderungen eines professionellen Motion Capture Systems.

Der praktische Vergleichstest findet anhand einer praxisnahen Produktion im Videostudio der Fachhochschule St. Pölten statt. Rein technisch ist es möglich, das System von *iPi Soft* simultan mit dem *Perception Neuron* Anzug zu testen. Als Referenz dienen Daten eines *Vicon* Setups der *Simon Fraser University & National University of Singapore* ([www.mocap.cs.sfu.ca](http://www.mocap.cs.sfu.ca)). Diese umfasst humane Bewegungsabläufe wie Gehen, Laufen, Springen, Tanz und Weitere.

Um die technischen Grenzen der beiden Systeme zu ermitteln, sind rund sieben Choreografien mit unkomplizierten bis hin zu komplexen Bewegungsabläufen geplant. Die Choreografien werden eingehend anhand der Vorgaben der Datenbank einstudiert, um sie mit dem *Perception Neuron* Set sowie gleichzeitig mit der *iPi Soft* Variante zu reproduzieren. Die erhaltenen Bewegungsdaten werden anschließend in ihrer Rohform in eine 3D-Animation konvertiert und optisch miteinander verglichen. Falls möglich, werden sie soweit aufbereitet, dass sie der Qualität des Industriestandards entsprechen.

Neben dem optischen Vergleich werden ebenfalls qualitativen Eigenschaften wie Nutzerfreundlichkeit, Störanfälligkeit und Leistungsumfang analysiert und dokumentiert.

Zur objektiven Bewertung aller Erkenntnisse wird anschließend eine Nutzwertanalyse erstellt, welche die in der Arbeit erfassten Systeme (*Vicon*, *iPi Soft* und *Perception Neuron*) nach einem Punktemodell bewertet. Die dazu nötigen Kriterien ergeben sich aus der Literaturrecherche sowie dem praktischen Vergleichstest und dessen Auswertung.

## 1.5 Gliederung der Arbeit

Um die Zusammenhänge der Fragestellung und technische Hintergründe besser zu verstehen, wird in den ersten beiden Kapiteln mit theoretischem Basiswissen an das Thema 2D- und 3D-Animation, sowie Motion Capturing herangeführt.

Den Hauptteil bildet der praktische Vergleichstest der zwei preiswerten Motion Capture Systeme. Der gesamte Produktionsablauf, inklusive Vor- und Nachbereitung, wird schriftlich und mit Videoaufnahmen dokumentiert.

Danach soll eine Nutzwertanalyse die drei erforschten Systeme nach einem Punktesystem bewerten und eine wirtschaftliche Entscheidungshilfe für zukünftige Projektplanungen liefern.

Im letzten Abschnitt findet ein Ausblick auf gegenwärtige sowie zukünftige Entwicklungen auf dem Gebiet der Bewegungserfassung statt, um mit allen gesammelten Fakten ein aussagekräftiges Fazit zu ziehen.

## 2 Grundlagen der Charakter Animation

### 2.1 Zeichentrickfilm

Der Ursprung der Computeranimation liegt in der klassischen Zeichentrickanimation, welche Anfang des 20. Jahrhunderts ins Leben gerufen wurde. Mit „Gertie the trained Dinosaur“ produzierte Winsor MacCay 1914 die erste, komplett von Hand gezeichnete Trickfilmanimation, welche aus 10.000 Einzelbildern bestand. (Dietmar Jackèl, Neunreither, & Wagner, 2006, S. 1)

Mit der sogenannten Cel Animation entwickelte dessen Erfinder Earl Hurd noch im selben Jahr ein Verfahren, welches den bewegten Bildvordergrund vom zumeist statischen -Hintergrund trennte und somit den enormen Zeitaufwand erheblich verkürzte. Dabei wurden die Vordergrundobjekte auf transparente Celluloid Folien gezeichnet und anschließend auf den Bildhintergrund gelegt und abfotografiert. 1916 ließ Otto Messner mit „Felix the Cat“ den nächsten Meilenstein in der Zeichentrickgeschichte entstehen. Das besondere an Felix war der „Studio Character“, ein speziell zugewiesenes Persönlichkeitsprofil. (Dietmar Jackèl u. a., 2006, S. 1, 2)

1928 folgte Walt Disney mit dem ersten vertonten Trickfilm „Steamboat Willie“, in dem Mickey Mouse sein Debüt feierte und damit den heute noch anhaltenden, enormen Erfolg seines Unternehmens sicherte. Vier Jahre später entstanden „Flowers and Trees“ sowie „Three little Pigs“, welche erstmals das Drei-Farben-Technicolor-Verfahren für Trickfilme einsetzten. Die Aufzeichnung wurde über eine Kamera auf drei separate Filmstreifen mit jeweils einem davor platzierten Farbfilter aufgezeichnet. Mit dem Oskar-prämierten „Snowwhite and the seven Dwarfs“ erreichte der klassische Trickfilm 1937 vorerst seinen Höhepunkt. Die darauffolgenden Filme wie „Fantasia“, „Pinocchio“, „Jungle Book“ uvm. zählen heute zu den großen Klassikern des Trickfilms. (ebd.)

The knowledge that went into making little drawings come to life is in the early Disneys. Nobody taught us how to articulate these fanciful characters. We had to discover the mechanics ourselves and pass them around amongst each other. There are many styles but the mechanics of the old Disney animation remain. (Williams, 2001, S. 46)

## 2.2 Keyframe Animation

Keyframe Animation bzw. Frame to Frame Animation wird seit Beginn der kommerziellen Produktion von Zeichentrickfilmen verwendet und zählt zu den ältesten, nach wie vor gebräuchlichen Animationsverfahren. Der Bewegungsablauf unterteilt sich in unterschiedliche Phasen, welche durch Keyframes oder Hauptphasen abgegrenzt werden. Im Zeichentrick übernimmt dies der Hauptphasenzeichner, alle Zwischenbilder werden vom Inbetweener erstellt. (Maestri, 2006, S. 111,112)

Die wichtigsten Techniken, welche Disney in den zwanziger und dreißiger Jahren hervorbrachte, haben heute noch ihre Gültigkeit. Sie entstanden durch genauestes Studieren natürlicher Objekte und Lebewesen. Im Buch „The Illusion of Life: Disney Animation“, verfassten Ollie Johnston und Frank Thomas zwölf Grundprinzipien für glaubwürdige Animationen, welche zum Basiswissen eines jeden guten Animators zählen sollten. Sie werden im Folgenden kurz erläutert.

### 2.2.1 Squash & Stretch

Elastische Körper, die auf anderen Objekten aufprallen, müssen zusammengestaucht oder in ihrer Länge gedehnt werden. Das Gesamtvolumen des Körpers soll erhalten bleiben. Dieses Prinzip dient zur Verdeutlichung von Geschwindigkeit. (Johnston & Thomas, 1981, S. 47-51)

### 2.2.2 Anticipation

Eine Hauptbewegung entsteht aus einer Gegenbewegung. Beispielsweise setzt ein Charakter erst zum Sprung an, bevor er zur eigentlichen Flugphase übergeht. Der Zuschauer wird damit auf die kommende Bewegung vorbereitet und kann schnelle Aktionen leichter wahrnehmen. Sie wirken besser lesbar und natürlicher. (Johnston & Thomas, 1981, S. 51–53)

### 2.2.3 Staging

Damit wird eine Handlungsidee unmissverständlich dargestellt bzw. die Aufmerksamkeit des Zuschauers in den Mittelpunkt der Handlung geführt. Es ist wichtig, dass sich der Charakter vom Rest der Szene abhebt. Dazu gibt es unterschiedliche Methoden, wie der gezielte Einsatz von Bewegungs- und Farbkontrasten (Vordergrund/Hintergrund). Idealerweise ist der Körperausdruck des Charakters eindeutig, auch wenn nur Umrisse der Figur sichtbar sind. (Johnston & Thomas, 1981, S. 53–56)

### 2.2.4 Straight Ahead & Pose-to-Pose

Bei *Straight Ahead* wird eine Bewegungssequenz vom Anfang bis zum Ende durchgängig animiert, ohne auf Schlüsselpositionen zu achten. Die Technik findet unter anderem bei unvorhersehbaren Animationen wie Feuer, Wasser, Explosionen, usw. Anwendung.

*Pose-to-Pose* verwendet besagte Keyframes. Dabei wird die Sequenz in Abschnitte unterteilt, um daraufhin Schlüsselpositionen des Charakters zu definieren. Die Zwischenbilder werden im Nachhinein vom Inbetweener ergänzt. (Johnston & Thomas, 1981, S. 57–59)

### 2.2.5 Follow through & Overlapping Action

*Follow through* bezeichnet die eigenständige Bewegung unbelebter Elemente eines Objekts oder Charakters (Haare, Gliedmaßen, ...), welche erst vor oder nach der Hauptbewegung agieren und zeitverzögert zum Stillstand kommen. Beim *Overlapping* überlappen die unterschiedlichen Bewegungsphasen mehrerer dieser Elemente im Zeitverlauf. Diese Technik wird angewandt, um die visuelle Lücke zwischen der aktuellen und der folgenden Aktion zu schließen und somit das Interesse des Zuschauers aufrecht zu erhalten. (Johnston & Thomas, 1981, S. 60–63)

### 2.2.6 Secondary Action

Sie wird häufig mit der *Overlapping Action* verwechselt und dient vielmehr dazu, die Hauptbewegung zu unterstützen, um der Animation eine größere Dimension zu verleihen.

Beispielsweise kann sich ein Charakter vor dem Heben einer Kiste (Hauptbewegung) die Hände reiben (unterstützende Bewegung). Es gilt darauf zu achten, dass die unterstützende Bewegung das Gesamtbild nicht dominiert. (Johnston & Thomas, 1981, S. 64, 65)

### 2.2.7 Timing

Die Persönlichkeit und Natur eines Charakters lässt sich maßgeblich davon beeinflussen, wie viele Zwischenbilder bei einer vorgegebenen Bildwiederholrate zur Darstellung einer Handlung eingesetzt werden. Mehr Frames resultieren in einer langsamen Bewegung, weniger Frames in einer schnellen.

Beim *Timing* geht es darum, ob für jeden einzelnen Frame erneut animiert wird (*drawing on ones*), oder nur für jeden zweiten (*drawing on twos*), jeden dritten (*... threes*), usw.

Ein klassischer Spielfilm bedient sich 24 Bilder in der Sekunde. *Drawing on twos* ist daher weiter verbreitet als *-ones*, da sich die Hälfte der Arbeit einsparen lässt. Außerdem wirken langsame Bewegungen ästhetischer, da durch ungenaue Animationen ein zittriger Effekt (*Jitter*) entstehen kann.

*Drawing on ones* wird verwendet, sobald schnelle Bewegungen für den Zuschauer trotzdem gut erkenntlich bleiben sollen. (Johnston & Thomas, 1981, S. 65, 66)

### 2.2.8 Slow In & Slow Out

Nahezu jeder Bewegungsablauf variiert in seiner Geschwindigkeit. Sind Bewegungen linear, wirken sie auf den Zuschauer mechanisch und unnatürlich. Zeichentrick bedient sich bei Anfangs- und Endpunkten einer erhöhten Bewegungsauflösung bzw. mehrerer Keyframes. Bei 3D-Animationen und Motion Graphics werden die Bewegungskurven in der Software S-förmig angepasst, bekannt unter *Easy Ease*. (Johnston & Thomas, 1981, S. 63)

### 2.2.9 Arcs

Bewegungsabläufe lebendiger Charaktere verhalten sich nicht nur im Zeitverlauf nonlinear, sondern ebenfalls in der Position. Gliedmaßen rotieren beispielsweise um ein Gelenk herum und bewegen sich eher bogenförmig als geradlinig. Diese Technik verleiht dem animierten Objekt mehr Persönlichkeit. (Johnston & Thomas, 1981, S. 63, 64)

### 2.2.10 Exaggeration

Um Bewegungen zu verdeutlichen, kommen oftmals Überbetonungen zum Einsatz. Dabei geht es nicht darum, die Posen zu verzerren, sondern überzeugender darzustellen. Diese Technik gilt es sparsam einzusetzen, um einen unrealistischen Gesamteindruck zu vermeiden. (Johnston & Thomas, 1981, S. 66, 67)

### 2.2.11 Solid Drawing

Dieses Prinzip gilt gleichermaßen für gezeichnete und computeranimierte Grafiken. Der Charakter soll sich gefühlt im dreidimensionalen Raum befinden. Dazu muss dieser richtig proportioniert und perspektivisch korrekt dargestellt sein. Symmetrien am Objekt oder in der Bewegung wirken unnatürlich und sind zu vermeiden. (Johnston & Thomas, 1981, S. 67–69)

### 2.2.12 Appeal

Der Charakter soll in seinem Aussehen und seinen Handlungen dem Zuschauergeschmack entsprechen bzw. das Charisma eines realen

Schauspielers besitzen. Das ist natürlich Geschmackssache. Ein guter Anhaltspunkt ist es, dem Charakter Dynamik zu verleihen. Appeal muss nicht automatisch Schönheit implizieren, sondern kann beispielsweise über interessante Details erreicht werden. (Johnston & Thomas, 1981, S. 69, 70)

## 2.3 Digitale Charaktere

### 2.3.1 Rigging

Sobald ein Charakter modelliert wurde, muss dieser für seine Animation vorbereitet werden. In diesem Prozess werden ihm das Skelett und dessen Steuerelemente vergeben, damit der Animator ihn im Anschluss manipulieren kann. (Maestri, 2006, S. 65)

Ein guter Character Rigger braucht gleichzeitig das Verständnis eines Animator, Programmierer und Interface Designer, um ein effizientes Charakter-Setup zu kreieren. Dies gibt dem Animator später die nötige Kontrolle, ohne ihn mit irrelevanten Nebenaufgaben zu beschäftigen. (ebd.)

Die Animation digitaler 3D-Charaktere/Objekte läuft grundsätzlich immer nach der *Skin and Bones* Technik ab. Das bedeutet, dass das zu animierende Objekt (Körper, Kleidung, Haare, uvm.) in eine Vielzahl von Polygone unterteilt wird. Diese werden wie ein Netz über das später unsichtbare Skelett gelegt. (Fischer, 2016, S. 18)

Das virtuelle Skelett besteht wie beim menschlichen Körper aus *Bones*, welche über *Joints* miteinander verbunden sind. In den meisten 3D-Programmen ist das Skelett ein ungerendertes Helfer-Objekt, welches im Charakter platziert wird und somit dessen Polygone bzw. *Mesh* ansteuert und deformiert. (Maestri, 2006, S. 68)

Ein 3D-Skelett entspricht in der Regel nicht 1:1 einem realen Skelett, da sich dutzende kleine Knochen durch virtuelle Knochen zusammengefasst lassen. Hochrealistische Objekte können jedoch weitaus komplexere Knochenstrukturen beinhalten. In den verschiedenen 3D-Programmen sind *Bones* und *Joints* unterschiedlich dargestellt, die tetrahedrale Form findet in den meisten Fällen Anwendung. (ebd.)



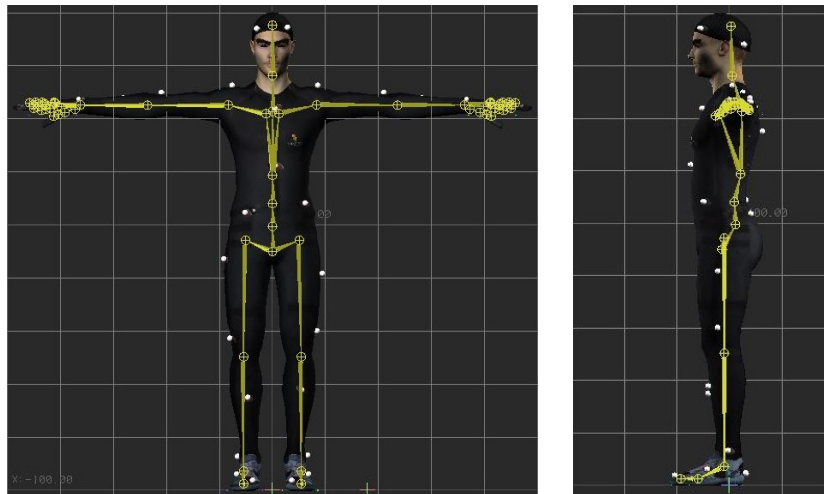


Abbildung 1. Virtueller 3D-Charakter inklusive 3D-Skelett

### 2.3.2 Kinematik

Die *Forward Kinematic* (FK) bezeichnet die standardmäßige Manipulation von Skelett-Hierarchien. Die Gliedmaßen des menschlichen Körpers werden über sich drehende Gelenke bewegt. Nach demselben Prinzip funktioniert die FK. Soll beispielsweise ein Charakter einen Gegenstand greifen, so werden die Gelenke so lange abwärts ihrer Hierarchie nach rotiert (z.B. Schulter > Oberarm > Unterarm > Hand), bis die Hand den Gegenstand erreicht. (Maestri, 2006, S. 70)

Das Gegenstück dazu ist die *Inverse Kinematic* (IK). Anstelle von Rotation wird hier Translation eingesetzt. Für den Griff nach einem Objekt wird die Hand, das unterste Glied der Hierarchie, am Objekt platziert. Alle darüberliegenden Elemente richten sich automatisch entsprechend dieser Bewegung neu aus. (Maestri, 2006, S. 71)

Die IK Kette besteht grundsätzlich aus der *Root* (Ursprung), dem *Joint*, dem *Bone* und dem *Effector* in gegebener Reihenfolge. In den meisten Programmen führt die Umpositionierung der *Root* zur Verschiebung der gesamten IK Kette. Der *Effector* steuert die Bewegung der darüberliegenden *Bones* bzw. die Drehung der *Joints* und ist im Regelfall das gekeyframe-Element. Der *Effector* lässt sich mit externen Objekten verknüpfen (z.B. Fahrradpedal). (Maestri, 2006, S. 72)

Häufig entstehen an Gliedmaßen mit ausgeprägten Freiheitsgraden (z.B. Arme) Interpolationsprobleme, da diese mehrere Möglichkeiten haben, an ein gewünschtes Ziel zu gelangen. Abhilfe schaffen manuelle Restriktionen der *Joint*-Winkel. In einigen Programmen lassen sich die *Joints* zudem als Scharnier- (zweidimensional) oder Kugelgelenk (dreidimensional) definieren. (Maestri, 2006, S. 72, 73)

Die unterschiedlichen Körperteile eines Charakters erfordern oftmals spezifische Animationsmethoden, weshalb zwischen FK und IK gewechselt werden muss. Gute Programme liefern die Möglichkeit zur Überblendung von IK auf FK mithilfe eines Prozentreglers zwischen 0% bis 100%. (Maestri, 2006, S. 75)

### 2.3.3 Keyframing

Die Animation digitaler Charaktere basiert ebenfalls auf Keyframing, wobei der Keyframe die exakte Position eines Objekts zu einem definierten Zeitpunkt bestimmt. Ein Interpolationsalgorithmus übernimmt die Funktion des Inbetweeners. Die Art des Algorithmus muss vom Anwender der Software vorgegeben werden. Keyframes können eine Vielzahl weiterer Attribute wie Position, Farbe, Transparenz uvm. vergeben werden. (Maestri, 2006, S. 112)

### 2.3.4 Motion Graph

Über *Motion Graphs* lässt sich exakt bestimmen, wie sich ein Objekt innerhalb der Hauptphasen verhält. Der Graph selbst wird üblicherweise über *Bézier Handles* beliebig manipuliert. Anhand der *Motion Graphs* kann ein guter Animator alle Probleme eines Bewegungsablaufs auslesen und gegebenenfalls reparieren. In den unterschiedlichen Programmen ist die Handhabung der Bewegungskurven zumeist ähnlich. Die horizontale Achse des Graphen kennzeichnet den Zeitverlauf, die vertikale Achse beinhaltet Parameter wie Rotation, Position, Skalierung, uvm. (Maestri, 2006, S. 113, 114)

Oftmals wird die Bewegung des Objekts in X, Y und Z Achsen aufgeteilt, um absolute Kontrolle zu gewährleisten. Da die Bewegungskurven meist glatte oder runde Formen aufweisen, können über auffällige Beulen und Kanten im Graphen Fehler erkannt und ausgebügelt werden (Cleanup). (ebd.)

Eine anschauliche Darstellung der Keyframes findet sich im sogenannten *Dope Sheet*, in welchem nur die Keyframes als Blöcke ohne *Motion Graphs* im Zeitverlauf angezeigt und verändert werden. Das *Dope Sheet* ermöglicht eine effiziente Manipulation gesamter Hierarchiegruppen (z.B. Arm). (Maestri, 2006, S. 115)

## 3 Grundlagen des Motion Capturing

### 3.1 Einführung

Bei Motion Capturing handelt es sich um eine Technik, welche zur Erfassung von natürlichen bzw. realistischen Bewegungssequenzen zum Einsatz kommt. Ein mit sogenannten Markern (reflektierende oder leuchtende Flächen) oder auffälligen Merkmalen ausgestatteter Darsteller bzw. Gegenstand gibt hierbei die Bewegung vor. Die exakte Position des Objekts wird fortlaufend mit optischen, mechanischen, akustischen oder elektromagnetischen Sensoren aufgezeichnet und in ein für den Computer interpretierbares Format gewandelt. (Dietmar Jackèl u. a., 2006, S. 9)

Die gesammelten Bewegungsdaten müssen im Anschluss für gewöhnlich einer Nachbearbeitung unterzogen werden. Daraufhin stehen die optimierten Daten zur Steuerung des Charakter-Setups zur Verfügung. Häufig werden die Daten in Bibliotheken abgespeichert und kommerziell vermarktet. (ebd.)

MoCap erzeugt FK-Keyframes für jede einzelne Bewegung eines *Joints*, während IK die Möglichkeit bietet, gesamte *Bone*-Ketten nachträglich zu manipulieren und dabei die originalen Keyframes zu überschreiben. Das Mischen der beiden Techniken zielt somit auf die bestmögliche Performance der Animation ab. Dieselbe Technik wurde bei der klassischen Zeichentrickanimation im Zuge der Rotoskopierung, bei der reale Filmaufnahmen über einer Mattglasscheibe abgepaust wurden, angewandt. (Kitagawa & Windsor, 2008, S. 130)

Die MoCap Daten dienen als Ausgangsgrundlage, um in weiterer Folge die Bewegungen des Charakters laut der im Kapitel 2.2 beschriebenen Animationsprinzipien nachträglich zu manipulieren. Damit wird ihm mehr Tiefe verliehen oder die Anforderungen der vorgegebenen Handlung erfüllt. (ebd.)

Nachträglich animierte Finger sind dahingehend ein typisches Beispiel. Hände werden vorwiegend als starre Objekte mit einer Drei-Achsen-Beweglichkeit am Handgelenk dargestellt. Ein Originalmitschnitt der Choreografie dient als Referenz für realitätsgetreue Animationen der Finger. Ein weiteres Beispiel für nachträgliche Anpassungen sind Gesichtsausdrücke, welche für gewöhnlich nur bei *Performance Capture* gleichzeitig vom System erfasst werden. (Kitagawa & Windsor, 2008, S. 130, 131, 134)

Motion Capturing ist kein Allheilmittel und soll die klassische Keyframe Animation nicht ersetzen, da die Methoden zumeist eine unterschiedliche Ästhetik verfolgen. Keyframe Animation eignet sich für Cartoon-artige, humorvolle oder unrealistische

Bewegungen. Im Gegensatz dazu kommt MoCap bei realistischen, ernsthaften oder generischen Bewegungsabläufen zum Einsatz. Die Performance des erfassten Subjekts beeinflusst dabei wesentlich die Qualität des Endresultats.

## 3.2 Motion Capture Systeme

Grundsätzlich werden MoCap Systeme in outside-in, inside-out und inside-in Systeme unterteilt. Dabei wird die Position der Signalquelle und der zugehörigen Sensoren definiert.

Beim outside-in Verfahren erfassen Sensoren von außen die Bewegungen des Subjekts innerhalb des Aufnahmebereichs. Beispiele dafür sind kamerabasierte Systeme, wobei die Kameras die Sensoren darstellen und die Marker oder auffällige Merkmale die Bewegungsdaten bekanntgeben. Inside-out Systeme platzieren die Sensoren nicht außen sondern direkt am Körper. Von dort aus empfangen sie ein Signal eines externen Senders (z.B. elektromagnetisches Feld) und berechnen ihre Bewegung im Raum. Das Inside-In Prinzip verpackt das gesamte System am Körper, wobei die Bewegung der Gliedmaßen den Sender und eine Kombination aus Sensoren wie Potentiometer, Gyroskope, Beschleunigungssensoren oder Magnetometer den Empfänger darstellt. Beispiele dafür sind elektromechanische- und inertielle Anzüge (z.B. Perception Neuron). (Menache, 2011, S. 16, 17)

### 3.2.1 Optisches Motion Capture

#### 3.2.1.1 Marker Systeme

Sofern ein ausgereiftes System zum Einsatz kommt, ist die optische, markerbasierte Bewegungserkennung eine herausragend präzise und damit bevorzugte Methode. Markerbasierte Daten können trotz allem einen enormen Nachbearbeitungsaufwand verursachen.

Das System arbeitet über Kameras mit CCD (Charged Coupled Device) Sensoren, welche das Licht der retroreflektierenden Marker durch spezielle Filter aufnehmen und die Koordinaten an einen Computer zur Auswertung zu senden. Eine wichtige technische Anforderung an den Sensor ist dessen Auflösung bzw. Pixelanzahl. Theoretisch gilt, je höher die Auflösung desto besser.

Das gegenwärtige Topmodell der *Vantage*-Familie von *Vicon Motion Systems* arbeitet mit 16 Megapixeln pro Kamera bei einer Bildrate von 120 Bildern pro Sekunde. (Vicon Motion Systems Ltd., 2016, S. 31)



Abbildung 2. Passive Marker, Vicon Vero (Low-End), Vicon Vantage (High-End)

In der Regel besteht ein Setup aus mindestens 8 bis maximal 32 Kameras. In Ausnahmefällen können jedoch Setups mit mehreren hundert Kameras möglich werden. Eine Vielzahl an Kameras verursacht enorme Datenmengen. Einige moderne MoCap Kameras (z.B. *Vantage*) besitzen deshalb eine integrierte Datenverarbeitung, wodurch sich Bandbreite und Rechenleistung reduzieren lässt. (Menache, 2011, S. 17, 18)

Neben der Auflösung spielt die Bildfrequenz oder Bildrate, welche in Frames per Second (fps) angegeben wird, eine wesentliche Rolle. Diese muss bei optischen Systemen hoch genug sein, um schnelle Bewegungen noch fein genug aufzulösen. Für die meisten Fälle sind 60 fps ausreichend. Ungewöhnliche Situationen erfordern Bildraten bis zu 420 fps oder darüber hinaus. Bei hohen Bildraten muss gleichzeitig die Auflösung verringert werden. Weiters ist die Verschluss synchronisation wichtig, da hier die einzelnen Kameras miteinander sowie mit der Frequenz externer Lichtquellen wie LEDs angeglichen werden.

Gewöhnlich sind die LEDs in Form von Ringleuchten an den Kameras verbaut. Als Lichtfarbe wird sichtbares Rot vorgezogen, da es vernachlässigend störend auf das Subjekt wirkt und gegenüber Infrarot effektiver strahlt. Bei passiven Systemen sendet die Ringleuchte an der Kamera ihre Strahlen direktional zu den Markern. Das reflektierte Licht gelangt wiederum auf dem CCD Sensor, wodurch sich die Koordinaten der Marker über Vektoren berechnen lassen. Die Marker selbst sind hauptsächlich Kugeln mit einer retroreflektierenden Beschichtung, ähnlich dem *Scotch Brite Tape*, und variieren in ihrem Durchmesser von wenigen Millimetern bis einigen Zentimetern. (Menache, 2011, S. 18)

Optisches MoCap wurde ursprünglich für medizinische Zwecke entworfen und konnte daher den Anforderungen von Computergrafikanwendungen nicht gerecht werden. Deshalb entstand das *Vicon 8* System, welches das erste System mit *SMPTE* Timecode Funktion war. Der Timecode erlaubte ein problemloses Editieren der MoCap Aufnahmen sowie ein genaues Abstimmen der virtuellen Charaktere mit austauschbaren Hintergründen, sogenannten Background Plates. Außerdem konnte mit *Vicon 8* erstmals ein Referenzvideo als AVI Datei abgespeichert und anschließend mit den MoCap Daten synchronisiert werden.

Auch heute noch bezieht *Vicon* als Spezialist in medizinischen Anwendungsgebieten rund 50 Prozent seiner Gesamteinnahmen aus dem Bereich der Biomechanik (v.a. Ganganalysen). (Menache, 2011, S. 18, 19), (Vicon Motion Systems Ltd., 2018c)

Damit das System perfekt aufeinander abgestimmt ist, muss es einen Kalibrierungsprozess durchlaufen. Dazu wird ein Objekt (z.B. ein Stab oder Würfel), dessen Abmessungen der Software bekannt sind, mit Markern ausgestattet. Ist dieses Objekt in jeder Kameraperspektive sichtbar, kann die Software daraus die exakte virtuelle Position für jede Kamera im Raum ermitteln. Dieser Prozess sollte vor und nach jeder Aufnahme wiederholt werden, da die kleinste Positionsänderung einer Kamera zu völlig verzerrten Ergebnissen führen kann. (Menache, 2011, S. 19)

Um die 3D-Position eines Marker zu errechnen, werden zumindest zwei kalibrierte Kameras benötigt. Die zusätzlichen Kameras soll dieses Verhältnis durchgehend aufrechterhalten. Kommt es dennoch zu großflächigen Verdeckungen, sind in der Software gegebenenfalls Algorithmen zur Überbrückung von Aufzeichnungslücken vorhanden. (ebd.)

Sobald die Aufnahme digitalisiert wurde, versucht die Software einen exakten Bewegungspfad der Marker zu reproduzieren. Dabei werden alle Pixel, deren Helligkeitswert einen vorgegebenen Schwellenwert unterschreitet, gelöscht und damit die Marker isoliert. Im nächsten Schritt bestimmt die Software die 2D-Koordinaten eines jeden Marker in jeder Kamera. Kombiniert ergeben sich daraus die 3D-Koordinaten der Marker. (Menache, 2011, S. 20)

Die Marker müssen die gesamte Sequenz hindurch eine eindeutige Identität aufweisen. Der MoCap Operator, welcher die Aufnahme am Computer betreut, definiert vor dem Aufnahmestart die Identität eines jeden Marker. Die Software versucht daraufhin diese Identität beizubehalten. Durch Verdeckungen oder Überblendungen, kann diese verloren gehen, worauf der Operator sie erneut zuweisen muss. Das Tracking ist abgeschlossen, sobald über die gesamte Sequenz hinweg jeder Marker eine eindeutige Identität sowie Koordinaten aufweist. (ebd.)

Die gewonnenen Daten beinhalten dabei nur absolute Positionskoordinaten der Marker auf der X, Y und Z Achse im Zeitverlauf. Zur Verarbeitung in 3D-Softwarepaketen sind gewöhnlich zusätzliche Verarbeitungsschritte nötig, bei denen zumindest die Rotation der Glieder sowie die Hierarchie des Skeletts definiert wird. Die meisten optischen Trackingsysteme bieten dazu dementsprechende Funktionen. (ebd.)

Im Allgemeinen wird bei *Vicon* Systemen von Genauigkeitsabweichungen knapp über 1 Millimeter gesprochen. In einer aktuellen Studie konnte eine

Gesamtgenauigkeit unter 2 Millimeter bewiesen werden. Die Genauigkeit variiert dabei zwischen statischen und schnellen Bewegungen. Bei statischen Experimenten konnte eine absolute Fehlerrate von 0.15mm und eine niedrige Abweichung von 0.015mm bewiesen werden. Für derart präzise Ergebnisse sollten Markergröße und Samplingrate, im Bezug auf die voraussichtlichen Beschleunigungen des erfassten Objekts, sachgemäß miteinander abgestimmt werden. (Merriaux, Dupuis, Bouteau, Vasseur, & Savatier, 2017, S. 2, 17, 18)

#### 3.2.1.2 Markerlose Systeme

Ein Beispielprodukt eines optischen, markerlosen MoCap Systems ist das in dieser Arbeit analysierte System von *iPi Soft*, welches die Bewegungsinformation aus der Berechnung einer Multikameraaufnahme oder alternativ aus RGB-D Kameraaufnahmen gewinnt. Auf das Prinzip von RGB-D Kameras wird im Kapitel 4.3.2 *Facial Motion Capture* näher eingegangen.

Objekttracking hat sich zu einer der populärsten Forschungsthemen im Bereich des Motion Capturing entwickelt. Die Erfassung menschlicher Bewegungen konnte in den letzten Jahren große Fortschritte verzeichnen, während generische Objekte immer noch ein schwer zu lösendes Problem darstellen. Gründe dafür können Helligkeitsschwankungen, Bildrauschen, komplexe Bewegungen oder komplexe Oberflächen sein. (Agren, 2017, S. 7), (Yilmaz, Javed, & Shah, 2006)

Um ein Objekt in einer Aufnahme zu tracken, muss dessen Bewegungspfad durch die Analyse der Position in jedem einzelnen Frame des Videos ermittelt werden. Nahezu jeder Trackingalgorithmus vermutet eine ausgeglichene Bewegung ohne plötzliche Richtungsänderung. Daten über das Erscheinungsbild und Verhalten des Objekts unterstützen diese Aufgabe zusätzlich. Das Tracking findet in drei Etappen statt, **Objektdarstellung**, **Objekterfassung** und **Objekttracking**. (ebd.)

##### 3.2.1.2.1 Objektdarstellung

Im ersten Schritt muss das Beobachtungsobjekt in eine für den Computer verständliche Sprache übersetzt werden. Gewöhnlich werden dazu Eigenschaften der Gestalt und/oder der Form als Basis herangezogen. Äußere Faktoren wie Anwendungsbereich, Zweck und Ziel definieren daraufhin den entsprechenden Algorithmus.

Objektdarstellung = Form + Gestalt
------------------------------------

(Agren, 2017, S. 8)

Um die Form zu beschreiben, gibt es je nach Anwendungsbereich und Objekt mehrere gängige Möglichkeiten:

- **Punkte (a,b)** – Ein Punkt (a) bzw. ein Set aus Punkten (b). Problematisch bei mehreren Objekten aufgrund von Verdeckung
- **Geometrische Formen (c,d)** – Darstellung durch primitive Geometrien wie Ellipse oder Rechteck. Geeignet für simple Objekte
- **Silhouette und Kontur (g,h,i)** – Konturdarstellung für die Außengrenzen- und Silhouettendarstellung für den inneren Bereich eines Objekts. Geeignet für komplexe Formen
- **Gegliederte Modellstruktur (e)** z.B. Darstellung menschlicher Glieder durch simple geometrische Formen wie Ellipsen
- **Skelett Modell (f)** – Bezieht sich auf die Silhouette eines Objekts und extrahiert durch die Mittelachsen-Transformation daraus ein virtuelles Skelett. Gebräuchliches Model bei der Objekterkennung

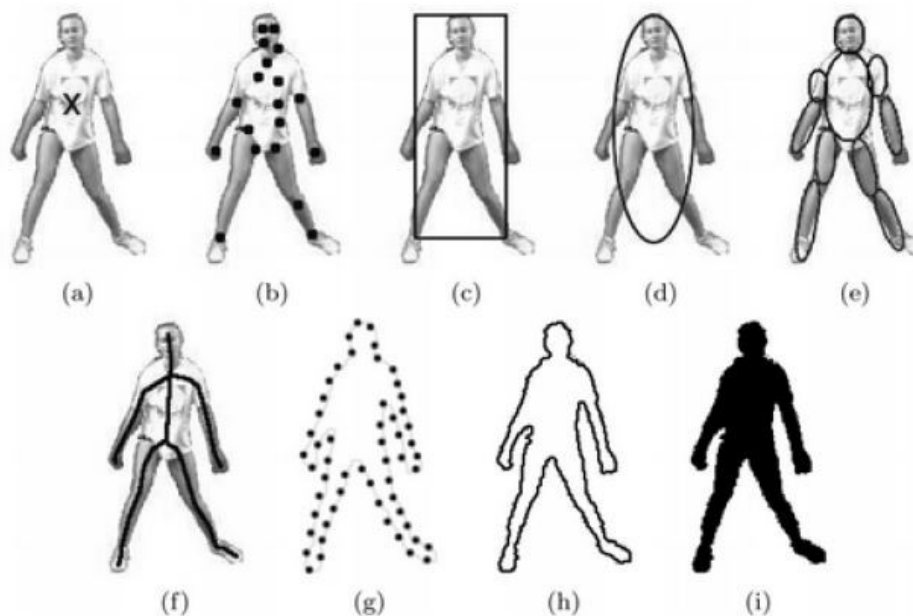


Abbildung 3. Verschiedene Formmodelle

(Yilmaz u. a., 2006)



Gleich der Form gibt es auch unterschiedliche Methoden die Gestalt zu beschreiben:

- **Wahrscheinlichkeitsdichte der Objektgestalt** – Ist die mathematische Funktion der Wahrscheinlichkeit einer zufälligen Variable in einem definierten Auswahlbereich vieler Variablen. Wird der vom Formmodell definierte Innenbereich eines Bildes (z.B. Kontur) genutzt, lässt sich eine Vorhersage über die Dichte gewisser Merkmale wie Farbe oder Textur anstellen
- **Vorlage** – Werden aus simplen geometrischen Formen erstellt. Eine Vorlage kann Informationen über Raum und Gestalt speichern. Führt zu Problemen bei drastisch abweichender Erscheinung in unterschiedlichen Perspektiven
- **Aktive Erscheinungsmodelle** – Eine Sammlung an Markierungspunkten im Objektinneren oder dessen Außengrenzen definieren die Form. Für jeden Punkt werden Gestalt-Vektoren vergeben, welche Farbe, Textur oder Gradientenausmaß beinhalten. Die Methode erfordert zuvor ein Training durch ein Set aus Samples
- **Multi-View Erscheinungsmodelle** – Errechnet die Gestalt aus unterschiedlichen Perspektiven

(Agren, 2017, S. 10), (Yilmaz u. a., 2006)

#### 3.2.1.2.2 Objekterfassung

Um ein willkürliches Objekt in einem Video zu Tracken, muss dieses zuvor identifiziert werden. Dazu gibt es zwei Ansätze. Entweder wird das Objekt in jedem Frame erneut detektiert oder nur einmal zu Beginn der Sequenz. (Agren, 2017, S. 12)

Die folgenden Absätze beschreiben die gängigsten Methoden der Objekterfassung.

**Punkt-Detektoren** beziehen sich auf *Interest Points*, welche bei Helligkeitsschwankungen und Perspektivenwechseln optimalerweise stabil bleiben. Der Begriff *Interest Point* wird oft verwirrenderweise mit dem Begriff *Corner* gleichgestellt, da sich die Punkte auf Ecken beziehen und daher ähnlich bis gleich beschreiben lassen. Es gibt einige Methoden zur Erfassung von Ecken und Punkten, wobei der *Harris Detektor* einer der populärsten ist. Er wird in mehreren Varianten ausgeführt. (ebd.)

Die **Hintergrund-Subtraktion** wird zum Trennen des Vordergrunds vom Hintergrund verwendet. Dabei wird der Hintergrund ermittelt und daraufhin ein Modell desgleichen erstellt. Der Algorithmus reagiert auf Veränderungen sensibel, weshalb das Modell stets auf dem aktuellsten Stand gehalten werden muss. Demnach dürfen sich keine beweglichen Objekte im Hintergrund befinden. Jeder Frame wird daraufhin mit dem Hintergrundmodell abgeglichen, sodass jegliche Bewegung als das zu erfassende Objekt interpretiert wird. (ebd.)

Die Technik ist entweder rekursiv- oder non-rekursiv. Rekursiv bedeutet, dass Änderungen im Hintergrund für jeden Frame erneut in das Modell mit einberechnet werden. Somit wird das Endmodell vom Ursprungsmodell abweichen und gleichzeitig irrelevante/fehlerhafte Inhalte mitschleppen. Im Gegenzug wird der Arbeitsspeicher weniger belastet. Die non-rekursive Technik beliefert laufend einen Buffer mit einer  $n$ -Anzahl an Frames, auf welchen sich das Hintergrundmodell ausschließlich bezieht. Das Modell aktualisiert sich dadurch schneller, der Arbeitsspeicher wird bei wachsender Buffergröße hingegen zunehmend belastet. (Agren, 2017, S. 12, 13)

Die **Segmentierung** unterteilt den gesamten Bildinhalt in mehrere Segmente oder Regionen. Diese Methode ist hilfreich zur Lokalisierung von Objekten und Abgrenzungen. Dafür gibt es verschiedene Ansätze. Einer davon nennt sich aktive Konturen bzw. *Snakes*, mit der Aufgabe, Außengrenzen von Objekten zu ermitteln. Die *Snake* wird unter der Anweisung externer Bedingungen entlang von Linien oder Kanten angeheftet. Aktive Konturen können das Objekt im Bild alleine nicht ausfindig machen. Der Algorithmus benötigt zusätzliche Informationen über die Gestalt des Objekts. (Agren, 2017, S. 13)



Abbildung 4. Bild Segmentierung (li.) und Aktive Konturen (re.)

**Überwachtes Lernen** beschreibt einen Algorithmus, welcher sich automatisch das Erscheinungsbild eines Objekts aus verschiedenen Perspektiven beibringt und demnach auf jegliche Vorlagen, die das Objekt mehr oder weniger gut

repräsentieren, verzichtet. Dem Computer werden dazu Lernbeispiele zur Verfügung gestellt, die ebenfalls die zuvor vom Anwender definierte Lösung enthalten. Unklassifizierbare Daten werden in Kombination mit der Lösungsfunktion als weitere mögliche Lösungsausgänge berechnet und klassifiziert. Der *Feature-Selection* Prozess hat einen entscheidenden Einfluss auf die Performance der Klassifizierung. Als bekannte Lernansätze gelten neuronale Netzwerke, adaptive Verstärkung, Entscheidungsbäume und unterstützende Vektormaschinen. (Agren, 2017, S. 14)

**Zeitliche Differenzierung** identifiziert pixelgenau die Unterschiede zwischen zwei oder drei aufeinanderfolgenden Frames. Diese Methode zeigt ihre Schwächen bei sich langsam bewegenden Objekten und versagt völlig, sobald das Objekt zum Stillstand kommt. (ebd.)

#### 3.2.1.2.3 Objekttracking

Es existiert eine Vielzahl an Algorithmen zum Verfolgen von Objekten, wobei sie grundsätzlich entweder für ein- oder mehrere Objekte in der Szene konzipiert sind. Die Algorithmen lassen sich in drei Hauptgruppen unterteilen, **Punkt-Tracking**, **Kernel-Tracking** und **Silhouetten-Tracking**, welche sich wiederum in Subgruppen aufteilen. (Agren, 2017, S. 14)

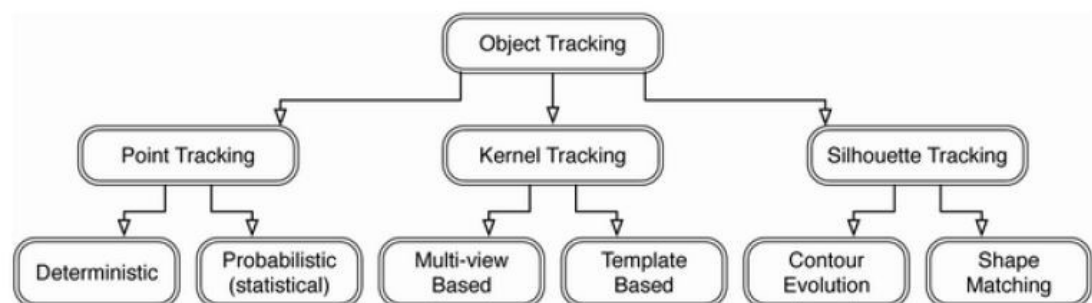


Abbildung 5. Kategorien gebräuchlicher Trackingverfahren

**Punkt-Tracking** evaluiert den Zustand des Objekts bezüglich dessen Position und Bewegungen, indem Punkte über die Frames hinweg verteilt werden. Es weist folgende Eigenschaften auf:

- Wird genutzt, wenn das Objekt in einem oder mehreren Punkten dargestellt werden soll
- Passend für Objekte, welche nur einen geringen Anteil am gesamten Bild einnehmen
- Hervorragend bei kleinen Objekten, die durch nur einen einzigen Punkt repräsentiert werden können
- Unempfindlich gegenüber Helligkeitsschwankungen
- Problematische Erfassung bei Verdeckungen (vor allem bei Multipunktdarstellung), Unklarheiten und dem Ein- und Ausstieg des Objekts in der Szene
- Schwierige Unterscheidung zwischen mehreren Objekten und zwischen Objekt und Hintergrund

(Agren, 2017, S. 23, 24)

*a. Deterministische Methode*

Wird dann angewandt, sobald die Bewegungskurven des Objekts mit einem vorhandenen Modell vorhergesagt werden können. Da mehrere Modelle parallel nebeneinander laufen, ist diese Methode normalerweise anpassungsfähig. Bei unvorhergesehenen bzw. komplexen Bewegungsabläufen kommt es schnell zu Problemen. Für menschliche Bewegungsabläufe existieren weitaus mehr Modelle als für tierische oder generische. (Agren, 2017, S. 24)

*b. Statistische Methode*

Eignet sich bei verrauschten bzw. gestörten Messungen oder ungenauen Vorhersagen durch unruhige Bewegungen. (ebd.)

**Kernel-Tracking** basiert auf dem Vergleich aufeinanderfolgender Frames, wodurch sich Aussagen über die Bewegungspfade treffen lassen:

- Wird genutzt, wenn primitive Formen wie Ellipsen oder Rechtecke die Form des Objekts repräsentieren
- Eignet sich hervorragend für simple, starre Objekte
- Objekt muss nur einmal zu Beginn der Szene erfasst werden
- Führt in Einzelfällen zum Auslassen von Objektpartien durch zu simple, geometrische Darstellung und Miteinberechnung des Hintergrunds
- Probleme treten bei starren sowie beweglichen Objekten auf
- Verdeckungen sind problematisch

(ebd.)

*a. Vorlagen-basierend*

Wird angewandt, wenn keine drastischen Erscheinungs- oder Bewegungsänderungen aus unterschiedlichen Perspektiven stattfinden. Die Methode kann als relativ simpel und unkompliziert angesehen werden. Bei komplexen Vorlagen wird sie zeitintensiv. Außerdem treten Probleme bei Verdeckungen und beim temporären Verlassen des Objekts in der Szene auf.

*b. Multi-View-basierend*

Effizient bei Objekten, die aus einer Kameraperspektive nicht ausreichend beschrieben werden können (z.B. Bewegungs- oder Erscheinungsänderungen). Das Tracking reagiert hier anpassungsfähiger auf plötzliche Veränderungen. Es ist komplexer in der Berechnung.

(Agren, 2017, S. 25)

**Silhouetten-Tracking** wird verwendet, wenn die Darstellung mithilfe simpler Geometrien oder Punkten unzureichend genau ist. Hier wird ein Modell anhand der Silhouette des Objekts in einem Frame erstellt, während im darauffolgenden Frame nach einer dazu passenden Region gesucht wird:

- Wird genutzt, wenn das Objekt anhand einer Silhouette oder Kontur dargestellt werden soll
- Eignet sich für komplexe und bewegliche Objekte
- Flexible Objektdarstellung
- Kann eine Vielzahl an Objektformen verarbeiten
- Die Objekterfassung muss nur einmal zu Beginn der Aufnahme erfolgen
- Schlechte Verarbeitung von Verdeckungen

(Yilmaz u. a., 2006), (Kersting, Smith, Brox, Klette, & Seidel, 2005, S. 17)

*a. Gestalt-Anpassung*

Eignet sich dann, wenn die Darstellung des gesamten Objekts notwendig ist. Dies ist beispielsweise dann der Fall, wenn Farbinformationen ein Bestandteil der Formbeschreibung sind. Die Methode lässt sich gut mit einer Objekterkennung kombinieren, da die gesamte Region evaluiert wird. Der Berechnungsaufwand ist hoch und kann zeitintensiv werden. (ebd.)

*b. Kontur-Evolution*

Ist in Situationen angemessen, bei denen die Kontur zur Darstellung des Objekts ausreicht, wie es z.B. bei einer Bewegungsmustererkennung der Fall ist. Objekterkennung ist hier unwesentlich. Diese Methode ist robuster und schneller als die Gestalt-Anpassung, außerdem unempfindlicher gegenüber Helligkeitsschwankungen. Das Prinzip funktioniert allerdings nicht, sobald das Objekt zur Gänze verdeckt wird. (ebd.)

#### 3.2.1.2.4 Feature Selection

Damit Tracking effizient funktioniert, müssen auffällige Merkmale eines Objekts bekanntgegeben werden, was für gewöhnlich manuell geschieht. Feature Selection ist eng mit der Objektdarstellung verwandt und wird auch maßgeblich von ihr beeinflusst. Merkmale können sein:

- **Kanten** – Erzeugen messbare Kontraste in der Bildamplitude und sind daher relativ leicht zu erfassen. Sind auch weniger störanfällig als Farbe
- **Optischer Fluss** – Bezieht sich auf Helligkeitsmuster und berechnet diese als Faktor bei Pixelunterschieden zwischen Frames mit ein
- **Farbe** – Wird zumeist in RGB, oder YCbCr und HSV repräsentiert. Ist anfällig gegenüber Helligkeitsschwankungen und Reflektionseigenschaften
- **Textur** – Gleichmäßigkeit und Regelmäßigkeit sind relevant und quantifizierbar

(Agren, 2017, S. 10, 11, 12)

#### 3.2.2 Radiofrequenz Positionssysteme

Diese Technik wird vor allem bei der Positionserfassung eingesetzt. Sie spielt aufgrund zu geringer Genauigkeit im Vergleich zu optischen Systemen derzeit eine untergeordnete Rolle. (Menache, 2011, S. 22)

Ein bekanntes Beispiel eines RF-Systems ist das Global Positioning System, kurz GPS. Ursprünglich für militärische Zwecke erfunden, wurde es 1983 der zivilen Bevölkerung in einer zunächst abgeschwächten Version zur Verfügung gestellt. Die GPS-Einheit am Boden berechnet durch Signale der GPS-Satelliten ihre relative Position zu einem festgelegten Koordinatensystem. Damit die Positionsermittlung nach heutigen Standards funktioniert, müssen zumindest vier Satelliten gleichzeitig die Position berechnen.

In seiner orbitalen Ausführung ist GPS für MoCap Anwendungen zu ungenau. Befände sich das System in einem begrenzten Areal, in dem Satelliten durch Transmitter ersetzt und die Empfänger am Subjekt angebracht werden, ließe sich eine ausreichende Präzision theoretisch erfüllen. (Menache, 2011, S. 22, 23)

Echtzeiterfassungssysteme bzw. Real-Time Location Systems (RTLS) existieren in vielen Ausführungen, das bekannteste Schema basiert auf dem Wireless Internet (Wi-Fi). Für MoCap ist RTLS nach dem jetzigen Stand mit Genauigkeiten von rund einem Meter zu ungenau und konzentriert sich vielmehr auf Tracking-Anwendungen mit beweglichen Inventar in Krankenhäusern, Lagerhallen und dergleichen. Die Bedeutung der Unterhaltungsbranche ist für die

Technologietreiber derzeit gering. Das MIT arbeitet jedoch an einem System, welches die 3D-Bewegung eines Körpers mithilfe der Reflektionen eines Wi-Fi Signals interpretiert. Damit lässt sich bereits das Zentrum des Körpers mit einer Genauigkeit von 10 bis 20 cm lokalisieren. (Menache, 2011, S. 23), (Götz, 2015, S. 28)

Local Positioning-Systeme (LPS) befinden sich noch in ihrer Reifephase und haben das Potenzial, die MoCap-Technik der Zukunft zu werden. Sie basieren auf hochfrequenten Radiowellen und bestimmen die exakte Position (1-2 mm) eines Objekts. Der wesentliche Unterschied zu den anderen RF-Systemen liegt darin, dass Sendeeinheiten am bewegten Objekt selbst die Signale über mehrere hundert Impulse pro Sekunde zu den Empfängern aussenden. Damit die nötige Genauigkeit erzielt wird, müssen die Position sowie die Distanz der Empfänger zu den Sendern bekannt sein. Inklusive der Signallaufzeit sowie dessen Charakteristik wird daraus die Bewegung des Objekts kalkuliert. Aufgrund seiner Funktionsweise und relativ preiswerten Hardware ergeben sich viele neue Möglichkeiten. So ließen sich beispielsweise sportliche Bewegungsanalysen in Echtzeit direkt am Spielfeld durchführen oder Virtual Reality Anwendungen revolutionieren. (Menache, 2011, S. 25, 26)

#### 3.2.3 Elektromagnetische Tracker

Das System wurde ursprünglich für militärische Flugzeughelme entwickelt und gehört zur Kategorie der Six Degree of Freedom (6DoF) Systeme. Sie bestehen aus einer elektronischen Kontrolleinheit, einer Software und mehreren am Objekt befindlichen Empfängern, welche ihre relative, räumliche Position zu einem Transmitter errechnen. Für gewöhnlich sind die Sensoren über Kabel mit der Kontrolleinheit verbunden. Der Transmitter erzeugt ein niederfrequentes Magnetfeld, welches durch die Sensoren aufgenommen und zur Kontrolleinheit gesendet wird. Von dort aus lässt es sich filtern, verstärken und zur Berechnung der Translations- sowie Rotationsdaten an die zentrale Recheneinheit bzw. dessen Software weitersenden.

Die Systeme werden entweder mit Wechsel- oder Gleichstrom betrieben. Sie besitzen alle dieselbe Störanfälligkeit gegenüber künstlich erzeugter, metallischer Gegenstände (magnetisches Wirbelfeld). Verschiedene Kompensationsalgorithmen versuchen das Problem zu vermindern. Grundsätzlich sollte Störquellen nach Möglichkeit aus dem Weg gegangen werden.

Elektromagnetische Tracker werden zum Beispiel in Controllern für Konsolen- und PC Spiele, Echtzeit-Bühnenperformances oder zur Berechnung der globalen Position bei inertialen MoCap Anzügen (*Perception Neuron*, ...) verwendet. (Menache, 2011, S. 26, 27, 28)



#### 3.2.4 Inertial Motion Capture Anzüge

Sie sind Inside-In basierend und arbeiten mit Gyroskopen, Beschleunigungssensoren oder Potentiometern, mittlerweile zumeist umgesetzt über kleine und präzise **Micro Electro Mechanical System (MEMS)** Sensoren. Diese werden in sogenannten Inertial Measurement Unit's (IMU) kombiniert. Die IMU's sind an strategischen Punkten auf einem Spandex Anzug oder über ein Gurtsystem am Körper befestigt und messen die Beschleunigungen und Ausrichtungen des Subjekts. Da sie keine zusätzlichen Kameras oder fixe Verbindungen zum Computer benötigen, besitzen derartige Anzüge eine praktische Handhabung und können in größeren Arealen, Innen sowie Außen eingesetzt werden. (Menache, 2011, S. 28, 29), (Götz, 2015, S. 28)

Zur Berechnung der globalen Translationskoordination wird eine unterstützende Größe benötigt. Zumeist übernehmen elektromagnetische- oder Ultraschallsensoren diese Funktion, was alle Anfälligkeiten dieser Techniken wie *Drift* und Ungenauigkeit mit einschließt. Auch wenn die Sensitivität der heutigen Sensoren drastisch gestiegen ist, treten Störungen bei feinen Bewegungen mit geringen Beschleunigungen auf. Darüberhinaus gehen die Anzüge grundsätzlich von simplen Scharniergelenken an den Körperteilen aus und berücksichtigen keine anatomisch komplexen Mechanismen wie die der Schulter, Handgelenke ö.ä. (ebd.)

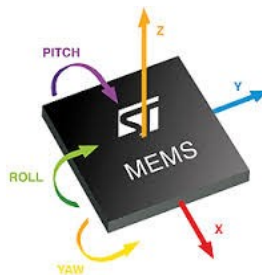


Abbildung 6. Illustration und Funktion einer MEMS Komponente

##### 3.2.4.1 MEMS Sensortechnik

Aufgrund ihrer unhandlichen Größe und hohen Kosten wurden Trägheitssensoren früher hauptsächlich in der Luft- und Schifffahrt eingesetzt. Während des letzten Jahrzehnts haben intensive Forschungen und die Einführung der MEMS Sensoren auf diesem Gebiet diese Hindernisse relativiert, weshalb die Komponenten mittlerweile in unzähligen Produkten der Industrie und des Massenmarkts verbaut werden. MEMS Komponenten sind klein, robust, leicht, preiswert, energiesparend und haben kurze Startzeiten. Ihre Hauptnachteile sind eine reduzierte Genauigkeit und Anfälligkeit gegenüber Störeinflüssen, weshalb eigenständige MEMS Komponenten ihren typischen Abweichungen unterliegen. (Woodman, 2007, S. 3)

Darum werden sie zumeist mit zusätzlichen, unterstützenden Sensoren kombiniert. Beispiele für derartige Helfer sind bildverarbeitende Sensoren, Ultrawideband-Signale (UWB) oder das GPS. Die Art der unterstützenden Sensoren richtet sich dabei nach dem jeweiligen Einsatzzweck. (Hol, 2011, S. 1)

Die Kombination mehrerer Sensoren bringt zwei wesentliche Probleme mit sich. Zum einen müssen die Datenströme der einzelnen Sensoreinheiten zusammengeführt werden, um eine zuverlässige Positions- und Orientierungsaussage treffen zu können. Zum anderen benötigen die Sensoren eine Kalibrierung, da die Daten selten an denselben Koordinaten im 3D-Raum gemessen werden. Dies bedeutet, dass die Ausrichtung und relative Position der Sensoren laufend an Ort und Stelle neu ermittelt werden muss, um im Anschluss exakte Daten in für uns lesbaren Einheiten zu liefern. Der Kalibrierungsprozess muss für jede Kombination aus Sensoren eigens gestaltet werden. Darum ist es wichtig, diese Prozeduren mit unterstützender Hardware und so simpel wie möglich durchzuführen. (Hol, 2011, S. 5, 6)

Für MoCap Anwendungen werden die IMU's strategisch auf den Gliedmaßen angebracht, wodurch die Rotation sowie relative Position eines Objekts ermittelt werden kann. Aus medizinischer Sicht erlaubt es Spezialisten, die Bewegungsabläufe eines Patienten zu überwachen und zu analysieren oder die Technik von Athleten zu studieren und zu verbessern. In der Unterhaltungsindustrie dienen die gewonnenen Daten als Grundlage für virtuelle Charaktere oder Spezialeffekte. (Hol, 2011, S. 4)

Sie enthalten ein 3D-Gyroskop sowie einen 3D-Beschleunigungssensor (G-Sensor) und in vielen Fällen auch ein 3D-Magnetometer. Beinahe alle IMU's sind entweder feststehend oder entkoppelt verbaut. Der Unterschied liegt am Referenzkörper, auf welchen sich das Gyroskop und der G-Sensor beziehen. (Woodman, 2007, S. 5)

Bei stabilen Plattform Systemen werden IMU's von externen Rotationen entkoppelt, d.h. die Sensoren bleiben stets zum Hauptrahmen hin ausgerichtet. Das wird über einen sogenannten *Gimbal* realisiert, welcher die Plattform in alle drei Achsenrichtungen frei bewegen lässt (siehe Abbildung 7.). Die einzelnen Sensorkomponenten erkennen jede kleinste Bewegung und senden die Signale weiter an kleine Drehmotoren (Torquemotoren), welche die Bewegungen ausgleichen und somit die Plattform durchgehend in der Waage halten. (ebd.)

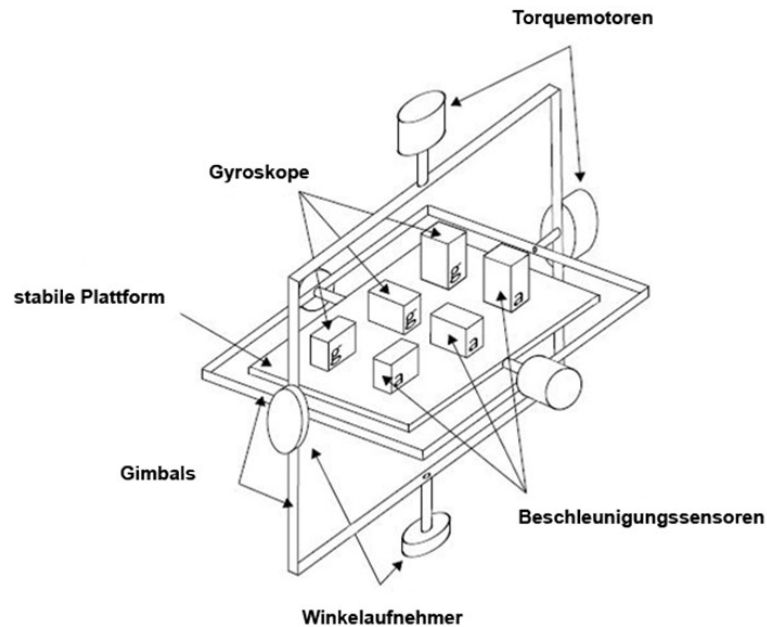


Abbildung 7. Gimbal-stabilisierte IMU-Plattform

Festsitzende IMU's sind hingegen fix mit dem beweglichen Objekt verbunden. Sie sind mechanisch simpler aufgebaut und können dadurch in kleineren Dimensionen gefertigt werden. Dafür ist wiederum mehr Berechnungsaufwand nötig, was zum heutigen Zeitpunkt keinen entscheidenden Kostenfaktor mehr darstellt. Deshalb sind festsitzende IMU's dominierend bei inertialen Navigationssystemen. (Hol, 2011, S. 3 - 6)

IMU Einheiten basieren auf der MEMS-Technologie und werden hauptsächlich für Navigationszwecke eingesetzt. MEMS funktioniert nach simplen mechanischen Prinzipien. Die Gyroskope (siehe Abbildung 8.) bestehen aus einem vibrierenden Massekörper, welcher auf einer vorgegebenen Frequenz oszilliert. Der Körper schwingt in einem mechanischen Federsystem und wird wie beim G-Sensor über ein kapazitives System ausgelesen. Rotiert der Körper, kann über die aus dem Coriolis-Effekt entstehende sekundäre Schwingung der Rotationswinkel errechnet werden. Ein Gyroskop berechnet jeweils die Rotation einer Achse, weshalb ein 3-Achsen Gyroskop (3D) die Rotation der X, Y, und Z Achse misst. Ein typisch auftretender Fehler ist die hohe Gravitationssensitivität, welche durch die Verformung der Federung auftritt und zu unerwünschten Abweichungen führt. Um dem entgegenzuwirken ist eine absolute Referenzgröße (z.B. Erdmagnetfeld) notwendig. (Hol, 2011, S. 23, 24)

Der Beschleunigungssensor funktioniert ebenfalls durch eine auf mechanischen Federn, in einem Vakuum gelagerten Masse, welche bei Beschleunigung verschoben wird. Die Auflösung und somit Genauigkeit des G-Sensors richtet sich direkt nach der Größe des verwendeten Masse-Feder-Systems. Die Sensoren stehen in 1D-, 2D- oder 3D-Varianten zur Verfügung. 3D-

Beschleunigungssensoren sind zum jetzigen Zeitpunkt nicht genau genug um in Hochpräzisions-MEMS AHRS's (Attitude and Heading Reference System) verwendet zu werden. (Xsens North America Inc., 2018)

Die entstehenden Strukturen können daraufhin mit entsprechenden Signalaufnehmern und elektronischen Schaltkreisen auf Silikonchips integriert werden. Das Verfahren dazu nennt sich Thick Epitaxial Layer for Micro-gyroscopes and Accelerometers (ThELMA), welches unterschiedlich dicke und dünne poly-Silikon Schichten miteinander kombiniert. (STMicroelectronics, 2018)

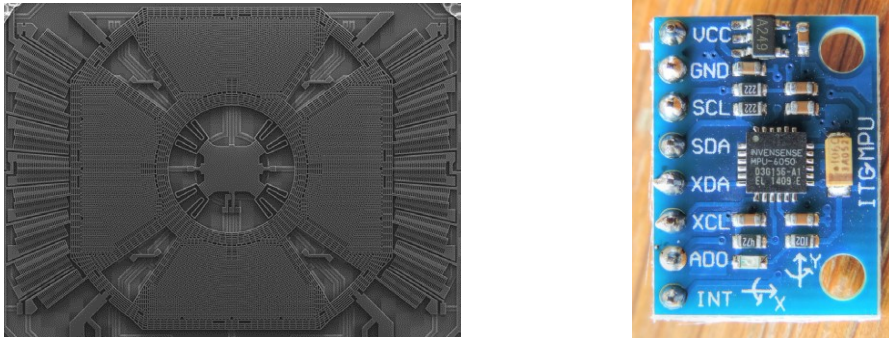


Abbildung 8. Mikroskopansicht eines 3D-MEMS Gyroskops (© STMicroelectronics) (li.) und IMU-Einheit mit MEMS Gyroskop und -G-Sensor (re.)

AHRS's sind die kosteneffizientere Alternative zu high-grade IMU's. Die Besonderheit ist eine on-board Datenverarbeitung, welche Informationen über die horizontale Ausrichtung und den Kurs liefert, während IMU's auf externe Geräte zur Berechnung dieser Parameter angewiesen sind. (Xsens North America Inc., 2018)

Bei MEMS Geräten sind weißes Rauschen und unkorrigierte Verzerrungen hauptverantwortlich für fehlerhafte Daten (z.B. *Drift*). Das weiße Rauschen bewirkt eine zufällige Winkeländerung, dessen Standardabweichung proportional zu Quadratwurzel der Zeitkomponente zunimmt. Eine unkorrigierte Verzerrung verursacht einen Fehler, welcher sich linear zum Zeitverlauf ausbreitet. Außerdem schleichen sich Quantisierungsfehler in die Berechnung mit ein. (Woodman, 2007, S. 23)

Je nach Anwendungsfall variiert die Qualität der verbauten Sensorsysteme, weshalb sie nach Kosten/Störanfälligkeit klassifiziert werden (siehe Abbildung 9.). Strategische und taktische Sensoren sind gewöhnlich in Hochrisikoanwendungen wie Personenflugverkehr oder Militärflugzeugen verbaut. Sensoren für den Massenmarkt besitzen typischerweise eine Störanfälligkeit von >30 Grad/h und

sind in Mobiltelefonen, Controller für Spielekonsolen (Nintendo Wii) uvm. zu finden. Consumer-grade Gyroskope driften während der Koppelnavigation zu schnell ab und sind daher für professionelle Anwendung ungeeignet. Im Zuge dessen wurden die Industrial-grade Gyroskope ins Leben gerufen, welche für unbemannte Fahrzeuge (Luft, Land, Wasser) oder zur Bemessung und Korrektur weiterer Sensorsysteme eingesetzt werden können. (Xsens North America Inc., 2018)

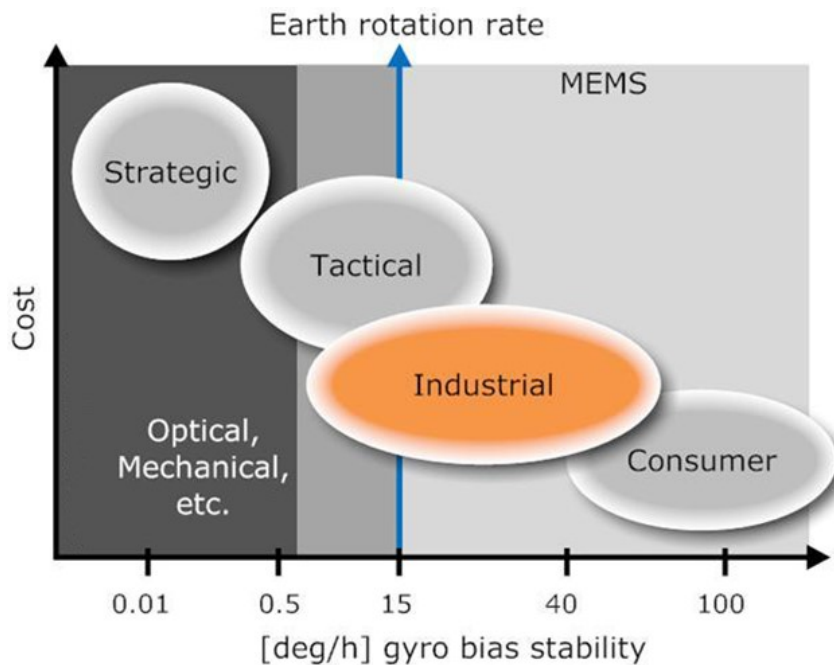


Abbildung 9. Qualitätsstufen von Inertial Sensoren (© xsens)

#### 3.2.4.2 Fehlerkompensation

IMU's benötigen zur genauen Positionsbestimmung eine physikalische Bezugsgröße. Kombiniert man eine IMU mit einem Magnetometer, gilt es darauf zu achten, dass dabei das ungestörte Erdmagnetfeld herangezogen wird. Platziert man die Sensoreinheit z.B. auf einem beweglichen magnetischen Objekt oder einem metallischen Gegenstand, so induziert sich eine magnetische Störspannung, welche nach Möglichkeit im Vorhinein in die Kalibrierung mit einberechnet werden muss. (Woodman, 2007, S. 33)

Die Basiskomponenten einer IMU für Motion Capturing sind demnach ein Gyroskop, ein Beschleunigungssensor und ein Magnetometer, wobei die beiden Letzteren für die Kompensation von Störeinflüssen und den Abweichungen des Gyroskops zuständig sind. Kurze Änderungen der Orientierung werden vom

Gyroskop erfasst, während der G-Sensor und der Magnetometer für die Langzeitstabilität verantwortlich sind. (Schepers, Guiberti, & Bellusci, 2018, S. 1)

Zusätzlich zu den orthogonalen Gyroskopen und Beschleunigungssensoren, sind in IMU's oftmals drei orthogonale Magnetometer verbaut. Diese messen die Stärke und Richtung des anwesenden Magnetfeldes, um so die Ausrichtung nach Norden festzustellen. Das Magnetometer würde aufgrund seiner Störanfälligkeit gegenüber künstlichen Magnetfeldern zu ungenaue Positionsdaten liefern und kann das Gyroskop daher nicht ersetzen. In Kombination arbeiten diese ausreichend präzise. Die Genauigkeit hängt dabei von einer sachgemäßen Kalibrierung und den vorherrschenden Umgebungsfaktoren ab. Falsche Neigungswinkel treten durch längere Beschleunigungsphasen auf, Ausrichtungsfehler durch magnetische Störfelder. Der Vorteil der Magnetfeldmessung liegt in einer integrierten und unabhängigen Funktionsweise, während absolute Positionssysteme wie das GPS auf externe Infrastrukturen zurückgreifen müssen. (Woodman, 2007, S. 33), (Schepers u. a., 2018, S. 1)

### 3.3 Konfigurationen

Aufgrund der hohen Komplexität werden die Hände sowie das Gesicht häufig nachträglich manuell animiert oder mit separaten MoCap Aufnahmen zur Hauptanimation hinzugefügt. Die folgenden Punkte beschreiben kurz die wesentlichen Merkmale dieser Sondersituationen.

#### 3.3.1 Hand Motion Capture

Das Erfassen der Hand ist nicht trivial, da sie nur eine begrenzte Fläche zum Platzieren von Markern bietet. Subtile Hand- und Fingerbewegungen vermitteln eine Vielzahl unterschiedlicher Emotionen. (Kitagawa & Windsor, 2008, S. 137)

Der größte Hinderniss beim Tracken mit optischen Systemen sind Verdeckungen. Es gibt viele Situationen, in denen sich die Finger oder Hände gegenseitig verdecken. Ein anschauliches Beispiel ist das Deuten mit dem Zeigefinger. Während ein ausgestreckter Finger dessen Marker sichtbar lässt, ballen sich die restlichen Finger zu einer Faust. Es gibt hier keine Möglichkeit, die Geste ohne Verdeckungen zu Erfassen. Treten derartige Situationen begrenzt auf, muss nicht auf Hand-MoCap verzichtet werden. Aus wirtschaftlichem Interesse gilt zu prüfen, ob die Hände nachträglich animiert werden sollen. (Kitagawa & Windsor, 2008, S. 149, 150)

Es gibt verschiedene Konfigurationen, um eine Hand für das MoCap vorzubereiten. Alle Arten haben gemeinsam, dass die Marker für eine gleichmäßige Sichtbarkeit auf dem Handrücken platziert werden. Ebenso ist eine zentrierte Anordnung der Fingermarker vorteilhaft, um gegenseitige Verdeckungen so gering wie möglich zu halten. (Kitagawa & Windsor, 2008, S. 141)

Um zeitaufwendige Reparaturversuche zu vermeiden, wird das Subjekt wenn möglich auf potentielle Verdeckungen hingewiesen, ohne dabei die Aufmerksamkeit ausschließlich auf die Marker zu lenken. Außerdem gilt es den Ablauf der Szenen zuvor gedanklich durchzugehen, um mögliche Komplikationen von Grund auf zu vermeiden. Es ist nicht sinnvoll, eine gute Aufnahme aufgrund einer kurzen Verdeckung abzubrechen. (Kitagawa & Windsor, 2008, S. 150)

Eine vorwiegend angewandte Methode für das Handtracking nennt sich *Rigid Hand* (= starre Hand). Hierbei repräsentieren drei auf dem Handrücken positionierte Marker die gesamte Hand. Die Finger verhalten sich zusammengebunden und unbeweglich. Ein ungleicher Abstand der Marker zueinander erleichtert dem Tracking System, die Finger zu differenzieren. Die Rigid Hand Konfiguration liefert nur eine grob simplifizierte Darstellung der tatsächlichen Handbewegungen. (Kitagawa & Windsor, 2008, S. 141, 142)

#### 3.3.2 Facial Motion Capture

Gesichts-MoCap ist ein etablierter Geschäftsbereich und Gegenstand intensiver Forschungsarbeiten. Es gibt mittlerweile eine Vielzahl markerbasierter und markerloser Lösungen zum Scannen von Gesichtern. Nachwievor muss ein Kompromiss zwischen der Genauigkeit der Daten und der Größe des dafür nötigen Setups eingegangen werden. Das obere Ende markieren Systeme, welche für die größtmögliche Qualität zur Verwendung in Spielfilmen ausgelegt sind. (Bouaziz, 2015, S. 9)

Aufgrund ihrer Robustheit haben sich markerbasierte Techniken als ein weit verbreitetes und überzeugendes Werkzeug für Echtzeit-Gesichtsanimationen in den Bereichen nicht-menschlicher Kreaturen und schlichter Videospielcharaktere bewährt. Die Gesichtserkennung mit Markern simplifiziert wesentlich den Trackingprozess, schränkt allerdings die räumliche Detailauflösung ein. Hervorragende Ergebnisse lassen sich erzielen, indem die Marker in ausreichender Anzahl entlang der Gesichtsmuskelpartien angebracht werden. Die Gesichtsdaten sind rein örtlich, sodass es keinen Unterschied macht, ob sich der Schauspieler während der Aufnahme bewegt oder nicht. Die Daten müssen vor dem Transfer auf den digitalen Charakter stabilisiert und aufbereitet werden. (Menache, 2011, S. 200), (Bouaziz, 2015, S. 9)

Für präzise Digitalisierungen menschlicher Gesichtsstrukturen haben sich 3D-Scanverfahren mithilfe strukturierter Lichtscanner oder Multi-View-Kamera-Systeme als ein effizientes Instrument behauptet. Hochpräzise Gesichtserkennungen werden für gewöhnlich über verschiedene Varianten von Registrierungs- und Trackingalgorithmen, angewandt auf Geometriedaten, Texturen oder beides, realisiert. Sie sind durch ihren Fokus auf Präzision nicht für interaktive Anwendungen ausgelegt. Dazu wäre es nötig, derartige Systeme mit Markern zu kombinieren. (Bouaziz, 2015, S. 10)

Die am meisten verwendeten Gesichtsanimationssysteme basieren auf vormodellierter Shape-Interpolation oder Cluster- bzw. Gitterdeformation. Viele heutige Systeme verwenden beide Techniken gleichzeitig. Das State-of-the-Art Modell ist ein Muskelsystem kombiniert mit Shape-Interpolation. (Menache, 2011, S. 200)

Das Endziel, markerloses und hochpräzises Motion Capturing zu erhalten, gilt als „**der heilige Gral**“ in der VFX-Industrie. Die gegenwärtigen Entwicklungen bei RGB-D Geräten (z.B. Microsoft Kinect, Asus Xtion Live) eröffnen völlig neue Perspektiven im Bezug auf markerlose, preiswerte und portable Echtzeit MoCap Lösungen. RGB-D Geräte zeichnen ein Bild sowie gleichzeitig die zugehörige Tiefeninformation (Depth) auf. Dadurch lässt sich das Motion Tracking über eine 2D-/3D-Registrierung eines beweglichen Modells beschreiben. In Kombination mit einem ausgeklügeltem Algorithmus lassen sich damit bislang beispiellose



Gesichtserkennungen zu einem herausragend günstigen Preis umsetzen. (Bouaziz, 2015, S. iii), (Götz, 2015, S. 35)

Das Echtzeit-Remapping auf einen virtuellen Charakter funktioniert nur dann zuverlässig, wenn zuvor das Gesicht des Anwenders in markanten Gesichtsausdrücken (Lachen, Wut, Angst, usw.) gescannt und davon ein Modell erzeugt wurde. Zusätzlich verursachen preiswerte RGB-D Geräte hohe Rauschpegel. Deshalb müssen Algorithmen entwickelt werden, welche eine Synergie zwischen Computersimulation, künstlicher Intelligenz und Computergrafik schaffen. (Bouaziz, 2015, S. 2)

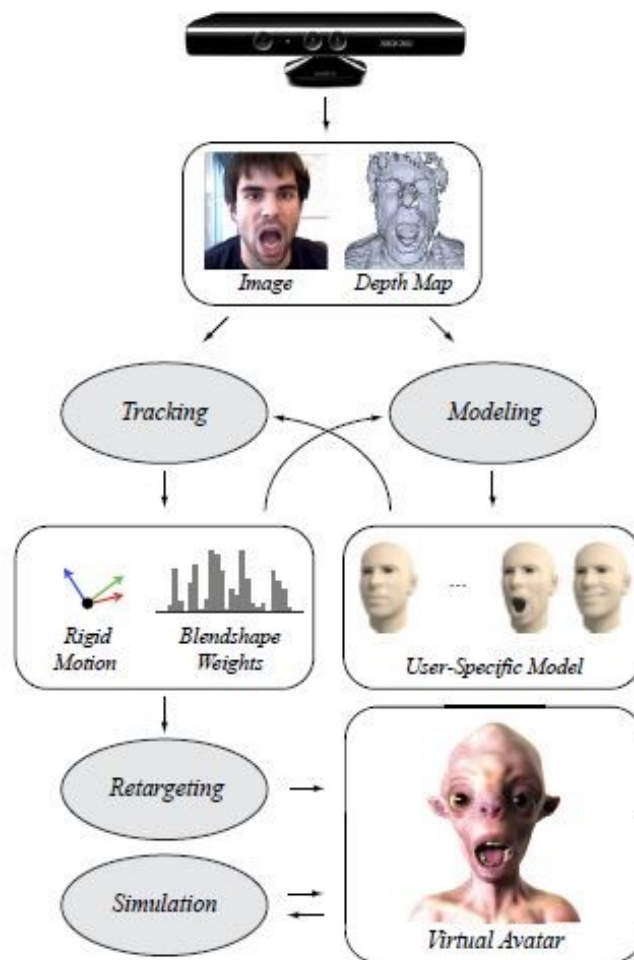


Abbildung 10. Animationspipeline einer Echtzeit-Gesichtserkennung

## 3.4 Anwendung in der Unterhaltungsindustrie

Derzeit profitieren hauptsächlich die Sparten Medizin, Sport, Entertainment, Rechtswesen und kleinere Nischenmärkte von der Technik. Im Entertainmentbereich entwickelt sich MoCap nicht zuletzt aufgrund des boomenden Gaming-Marktes zu einem zunehmend gefragten Werkzeug. (Menache, 2011, S. 3, 37)

### 3.4.1 Gaming

Die Videospielindustrie war die erste treibende Kraft, welche Motion Capturing großflächig in ihre Produktionen integrierte. Die Technik wird gegenwärtig in nahezu jedem Spiel mit menschlichen Charakteren eingesetzt. Der Ursprung liegt in einer Zeit, in der MoCap mäßige Ergebnisse zu einem gerade noch erschwinglichen Preis lieferte. Videospiele konnten aufgrund der damals begrenzten Rechen- und Speicherleistung keine allzu komplizierten Bewegungsdaten verarbeiten. Mit der stetigen Weiterentwicklung technischer Ressourcen verbesserten sich die Kapazitäten der MoCap-Systeme, welche für die heute gewohnte Präzision nötig sind. So entwickelten sich die beiden Industrien seither nebeneinander. Videospiele sind mittlerweile mühelos in der Lage, dieselben hochauflösten Bewegungen wie in Spielfilmen darzustellen. (Menache, 2011, S. 41)

Darüber hinaus kann der Spieler mit MoCap selbst zum „Controller“ eines Spiels werden. 2003 brachte die Firma Sony als Ergänzung zu ihrer Konsole PlayStation 2 die Eye Toy Kamera auf den Markt, welche unter bestimmten Voraussetzungen die Bewegungen des Spielers erfasst und gleichzeitig in 2D-Koordinaten wandelt. So ließen sich speziell dafür programmierte Spiele interaktiv ansteuern. Nintendo ging 2006 einen Schritt weiter und brachte den *Wii Controller* auf den Markt, welcher mithilfe eines Beschleunigungs- sowie Infrarotsensors die Bewegungen im 3D-Raum trackt. (ebd.)

#### 3.4.1.1 Game Engines

Eine Game Engine versteht sich als ein Framework, welches zur Entwicklung von Videospielen oder anderen Echtzeitanwendungen eingesetzt wird. Game Engines bestehen in der Regel aus einer Physik-, Sound-, Skript- und Rendering Engine. Die zwei am häufigsten verwendeten heißen *Unreal Engine* von **Epic Games** und *Unity* von **Unity Technologies**. (Przemyslaw & Leibetseder, 2016, S. 1)

Beim Import von 3D-Assets (Geometrie und/oder Skelettdaten) fixieren sich beide Engines auf dieselben Datenformate, FBX und OBJ. *Unity* ermöglicht zusätzlich den direkten Stream-Import von proprietären Datenformaten aus bekannten 3D-

Programmen wie *Blender*, *Maya*, *3ds Max*, etc. (Przemyslaw & Leibetseder, 2016, S. 9)

Im Gegensatz zu zeitaufwendig gerenderten Filmszenen, müssen Game Engines jeden einzelnen Frame in den Bruchteilen einer Sekunde berechnen. Das Hauptaugenmerk liegt daher auf einer möglichst effizienten Nutzung verfügbarer Rechenleistung, was in der Regel mit einer reduziert realistischen Darstellung einhergeht.

Game Engines sind in Verbindung mit Motion Capturing dahingehend interessant, da sie die Bewegungsdatenströme eines Ganzkörpersetups (Körper und Gesicht) mittlerweile in Echtzeit verarbeiten und in In-Game-Grafik (spielfertige Auflösung/Textur) wiedergeben können. In Kombination mit der originalen Tonaufnahme wird dieser Vorgang als Performance Capture bezeichnet, welches 2006 im Titel *Heavenly Sword* erstmals in einem Videospiel Anwendung fand. (Ninja Theory, 2015a)

#### 3.4.1.2 *Hellblade: Senua's Sacrifice*

Das aktuelle Vorzeigebeispiel an Echtzeit Performance Capture auf Low-Budget Niveau spiegelt sich im 2017 veröffentlichten, mehrfach ausgezeichneten Triple AAA Computerspiel ***Hellblade: Senua's Sacrifice*** wider.

Die Anforderungen an das Spiel bzw. die Hauptcharakterin waren hoch. Senua sollte real, nuanciert, komplex und glaubhaft wirken, um eine größtmögliche Verbindung zum Spieler aufzubauen. Normalerweise stellen sich Zwischenszenen in Triple AAA Titeln aus Tonaufnahmen eines Tonsprechers, Bewegungsaufnahmen eines Schauspielers und den manuell gekeyten Gesichts- und Kameraanimationen des Animators zusammen. Zum Zusammensetzen aller Sequenzen wird häufig ein Team aus über 30 Personen benötigt. (Ninja Theory, 2015a)

Das britische Entwicklerstudio Ninja Theory konnte auf nur 13 Personen zurückgreifen und musste daher die Kosten drastisch senken. Performance Capture ist immens kostspielig und wird selbst bei großen Videospielproduktionen, mit Gesamtbudgets von mehreren hundert Millionen USD, nicht oder nur sparsam eingesetzt. (ebd.), (Price, 2018)

Ninja Theory bauten aus 12 *Vicon Bonita* Kameras (Low-End), IKEA Gerüsten, Günstig-LED Panelen und handelsüblichen Trainingsmatten auf wenigen Quadratmetern ein eigenes MoCap Studio, mit tadellosen Ergebnissen. Für das Gesichtstracking wurde ein Prototyp eines Helms im 3D-Drucker angefertigt und mit zwei Kleinkameras versehen. Das Gesicht wurde über aufgemalte Markerpunkte erfasst und über einen eigens programmierten Echtzeit-Face-Solver auf den 3D-Charakter übertragen. Die Kamera wurde ebenfalls mit Markern

getrackt und dessen Position bzw. das Framing zeitgleich in die Game-Engine eingespeist. Das gerenderte Bild inklusive digitaler Szeneninhalte konnte daraufhin als Live-View auf den Vorschaumonitor der Kamera ausgegeben werden. (ebd.)



Abbildung 11. Echtzeit Performance Capture für das Computerspiel Hellblade

Das selbstgebaute Studio hatte seine Limitationen. Durch das geringe Aufnahmevolumen ließ es hauptsächlich nur eine Person und keine großräumigen Bewegungen zu. Akrobatische Aufnahmen wurden deshalb in das große MoCap Studio des Imaginarium in London ausgelagert und innerhalb von zwei Tagen abgedreht. Die Produktionskosten dafür betrugen knapp die Hälfte des gesamten, eigenen Studios. Ein zweiter wesentlicher Faktor ist die Anlaufzeit des Systems. Da einige Komponenten wie die Software in Eigenregie gebaut wurden, waren die Anlaufzeiten bis zum Aufnahmestart hoch. Das *Vicon Siren* System wirbt damit, denselben Zweck in einem Zehntel der Zeit umzusetzen. (Ninja Theory, 2015c) (Vicon Motion Systems Ltd., 2017d, S. 2)

Die Daten wurden zur Nachbearbeitung in *Autodesk MotionBuilder* importiert und aufbereitet. Die Gesichtsanimationen und Feinheiten wurden mit *Autodesk Maya* bearbeitet. Anschließend wurden die fertigen Animationen inklusive Audio in die *Unreal Engine* importiert und mit Licht und Effekten kombiniert. (Vicon Motion Systems Ltd., 2017d, S. 2)

Die Hardware für die Tonaufnahme war die teuerste Komponente im gesamten Setup. Inklusive Schauspieler konnte das Aufnahmeset von nur vier Personen betrieben werden. Die Gesamtinvestition entsprach laut CCO Tameem Antoniades dem eines neuen Kleinwagens (keine offiziellen Zahlen veröffentlicht). Das Entwicklerstudio hat damit bewiesen, dass mit gegenwärtig verfügbaren Mitteln, High-End Produktionen zu einem Bruchteil eines Multimillionen-Budgets möglich sind. (Ninja Theory, 2015b)

#### 3.4.2 Virtual Reality

Virtual Reality, oder kurz VR, beschreibt eine künstlich erschaffene Welt, in welche die Anwender durch verschiedenste Wahrnehmungstäuschungen „eintauchen“ sowie mit ihr interagieren können. Eine besondere Form der VR ist die Mixed Reality oder MR, welche wiederum in Augmented Reality (AR) oder Augmented Virtuality (AV) aufgeteilt wird. Augmented Reality fügt über einen transparenten Bildschirm (z.B. Brille), zusätzlich zur realen Umgebung, künstlich erschaffene Elemente (z.B. Informationen, Gegenstände, etc.) hinzu. Die Technik findet beispielsweise in der Sozialforschung, dem Ingenieurwesen und im Gamingbereich Anwendung. Augmented Virtuality integriert dagegen Gegenstände aus der Außenwelt in die künstliche Umgebung. (Fischer, 2016, S. 5, 14)

Ein VR-System setzt sich aus drei Komponenten zusammen. Zum Ersten das Tracken der Bewegungen des Anwenders über ein beliebiges Motion Capture Verfahren. Die erfassten Bereiche können einzelne Körperpartien (Kopf, Hände, ...) oder Ganzkörpertracking beinhalten. Die Bewegungsdaten werden daraufhin in Echtzeit in eine Game Engine gestreamed, auf den künstlichen Avatar übertragen und das gerenderte Bild wiederum dem Anwender zumeist über eine geschlossene VR-Brille (Head-Mounted Display) ausgegeben. (ebd.)

Für VR-Anwendungen sind markerlose Inertial MoCap Anzüge interessant, da die Bewegungen eines oder mehrerer Anwender preiswert, latenzarm und unkompliziert direkt in die Game Engine eingespeist werden können. Die Handhabung wird anhand des *Perception Neuron* Anzugs in Kapitel 4.3 genauer erläutert. (Fischer, 2016, S. 14)

Optische, markerbasierte Systeme können ihre Bewegungsdaten bereits in Echtzeit übertragen. Infolge wird der boomende VR-Markt intensiv bewirtschaftet. *Vicon* hat im August 2018 auf der *SIGGRAPH* Messe ihr erstes High-End Location Based Virtual Reality System (LBVR) namens *ORIGIN* vorgestellt.

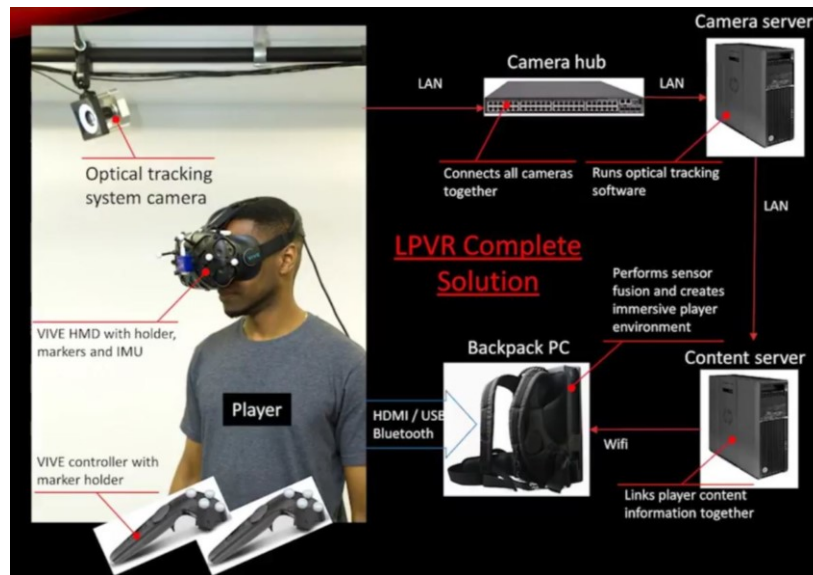


Abbildung 12. Beispiel eines Location Based VR-Systems der Firma LP-RESEARCH Inc.

#### 3.4.3 Spielfilm

MoCap erfüllt in diesen Genres die unterschiedlichsten Zwecke. Angefangen bei digitalen Extras, -Stunts oder -Hintergrundcharakteren bis hin zu hochkomplexen, durchwegs animierten Spielfilmheld\*innen. (Menache, 2011, S. 43)

Die Filmindustrie befindet sich seit geraumer Zeit in einem drastischen Umschwung, in welcher digitale Szeneninhalte traditionelle Produktionsumgebungen zunehmend verändern. Dadurch wird das Filmset immer kahler und stellt keine realen Anhaltspunkte für die gesamte Produktionsbesetzung inklusive Schauspieler bereit. Seit einigen Jahren wird deshalb an verschiedensten virtuellen Lösungen gearbeitet, die dieses Problem zumindest vermindern sollen. (Götz, 2015, S. 5)

Virtuelle Produktionen wirken wie ein neuartiges Konzept, wobei diese seit ihrer Entstehung bereits eine umfassende Probephase absolviert haben. Sie basieren auf einer Kombination aus Motion Capturing, Compositing, Echtzeit-Ray Tracing und leistungsfähigen Ein- und Ausgangsschnittstellen. (Götz, 2015, S. 26)

##### 3.4.3.1 Virtuelle Kamera

Eine Studiokamera, die einen Greenscreen filmt, welcher augenblicklich von einer virtuellen Studioumgebung ersetzt wird, kann als die simpelste Form einer virtuellen Produktion angesehen werden. Derartige Systeme kommen häufig in Studio- und Serienproduktionen fürs Fernsehen zum Einsatz. Die Echtzeit-Komposition entspricht in diesem Fall zumeist der finalen Ausgabe.

In Filmen ist der Szenenaufbau hingegen in der Regel weitaus komplexer. Da die Zusammenarbeit zwischen Regisseur und Kameramann eine entscheidende Rolle

im Entstehungsprozess spielt und Filmsets währenddessen immer leerer werden, würden diese von einer Live-Vorschau aus computergenerierten und realen Bildern enorm profitieren. Eine virtuelle Kamera übernimmt genau diese Funktion. (Götz, 2015, S. 30)

Die reale Kamera wird mithilfe optischer Marker (aktiv oder passiv) getrackt und wichtige Parameter (v.a. Brennweite und Sensorgröße) auf das virtuelle Äquivalent übertragen. Dies erlaubt einem einen Echtzeit-Einblick in das virtuelle Set/Studio. Optische outside-in Systeme leiden gelegentlich unter Problemen durch Verdeckungen und Reflexionen. Der Nachteil kommt bei virtuellen Kamerasystemen umso mehr zu tragen, da selbst kleinste Fehler störend wirken. (Götz, 2015, S. 31)

Im Gegensatz dazu erscheint die inside-out Technik am vielversprechendsten. Diese evaluiert die Raum-zeitliche Position der Kamera aus dessen aufgenommenen Bildern bzw. aus den Bildern einer zusätzlichen Kamera, welche direkt über der Hauptkamera montiert wird. Dabei bezieht sich die zweite Kamera auf ein Muster aus Markern, welches an der Decke angebracht wurde und dessen Markerabstände bekannt sind. Diese Methode ist nur in kontrollierten Produktionsumgebungen wie TV-Studios oder Filmsets sinnvoll. (ebd.)

Die jüngsten Bemühungen versuchen gänzlich ohne Marker auszukommen, indem natürlich vorkommende Objekte im aufgenommenen Filmmaterial in Echtzeit erkannt und getrackt werden sollen. Ein vollkommenes Tracken einer Kamera erfordert nicht nur extrinsische Parameter wie Position und Ausrichtung, sondern ebenfalls intrinsische Messungen der Blende, Fokus, Brennweite und Sensorgröße. (ebd.)

*Ncam Technologies* bedient seit 2014 mit einem derartigen System den Markt. Dazu wurde ein ausgeklügelter Algorithmus für bildbasiertes Featuretracking, unterstützt von Sensoren aus der Flugzeugindustrie, entwickelt. Das System besteht aus zwei sogenannten *Witness-Cameras*, welche unterhalb der Hauptkamera angebracht werden und Graustufenvideos mit 100 fps aufzeichnen. Die Prozessoreinheit trennt automatisch bewegliche und unbewegliche Objekte und zieht letztere als Berechnungsgrundlage für die Position der Kamera im 3D-Raum heran. (Götz, 2015, S. 32)

Sind die Graustufenbilder zu einheitlich, wird als Unterstützung der Fluglagensensor hinzugezogen. Die Verzögerung bis zum Erhalt der Bild- und Optikdaten wird zur Glättung der Trackingergebnisse genutzt. Die Abbildungsfehler (z.B. Aberration) der Optiken an der Hauptkamera müssen in einem Kalibrierungsprozess (mit Punktematrix) identifiziert und via Software ausgeglichen werden. Dieser Prozess kann für Zoomoptiken mit variabler Brennweite mehrere Stunden in Anspruch nehmen. (ebd.)

Die intrinsischen Daten werden in Echtzeit an ein Kamera Kontrollsystem wie z.B. das *FI + Z* von *Preston Cinema Systems* geschickt. Das Hybridsystem liefert robuste Daten, ist jedoch anfällig gegenüber Kontrast- und Helligkeitsschwankungen, was bei Live-Action Szenen problematisch ist. Außerdem ist die Trackingqualität maßgeblich vom Bildrauschen der *Witness-Cameras* abhängig. (ebd.)

Nachdem jeglicher Parameter gesammelt wurde, werden die Informationen zu einem 3D-Software Package, z.B. einer Gamengine, gestreamt und das kombinierte Bild wiederum auf einem beliebigen Vorschaumonitor ausgegeben.



*Abbildung 13. markerlose NCam (li.) und markerbasiertes System Optitrack Insight (re.)*



## 3.5 Datenformate

Grundsätzlich lassen sich Motion Capture Daten in Rotations- oder Translationswerten beschreiben und abspeichern. Optische Systeme erzeugen im Allgemeinen Translationsdaten, mechanische Systeme erzeugen Rotationen und magnetische Systeme beides. Rotationsdaten eignen sich optimal bei Ganzkörperanimationen mit einem Skelett, Translationen bevorzugt bei Gesichts-MoCap, *Digital Puppetry* und Echtzeit-Anwendungen. C3D, ASF/AMC, BVH und FBX sind die am häufigsten anzutreffenden Datenformate im Bereich des Motion Capturing und werden daher nun ausführlicher beschrieben. (Kitagawa & Windsor, 2008, S. 181)

### 3.5.1 C3D

Dieses binäre Datenformat findet bei Animationen, Biomechanik und Ganganalysen Anwendung und kann 3D-Daten sowie gleichzeitig jegliche numerischen Werte abspeichern (z.B. 3D- und analoge Daten). Ursprünglich wurde es exklusiv für optische MoCap Systeme entwickelt, um die rohen Kameraaufnahmen und gleichzeitig die Markerdaten zu speichern. Es besteht aus einem oder mehreren 512-byte Blöcken. Die Datei wird am Anfang in eine Header-, anschließend in die Parameter- und zuletzt die Datensektion unterteilt. (Kitagawa & Windsor, 2008, S. 181), (Menache, 2011, S. 138)

Im Header befinden sich Informationen über den Startpunkt der Parametersektion, Trajektorien (math. Lösungskurven), die Anzahl analoger Kanäle und Sampleraten. Ein Bruchteil der Parameter ist für den Unterhaltungsbereich relevant. (ebd.)

Die Parametersektion beinhaltet alle nötigen Informationen zum Auslesen der Datensektion wie Datentyp, Dimension und Position eines jeden Parameter. Die Parameter sind dabei Beschreibungen der Datenarten, welche im File abgespeichert sind. (ebd.)

Die Datensektion besteht aus Daten wie 3D-, 2D- und analogen Daten in beliebiger Kombination. Das C3D Format ist frei zugänglich und implementierbar. (ebd.)

#### 3.5.2 ASF/AMC

ASF (Acclaim Skeleton File) und AMC (Acclaim Motion Capture) sind ein von *Acclaim Entertainment* entwickeltes MoCap-Datenformat. Das Computerspielunternehmen trug maßgeblich zur Entwicklung von optischen Motion Capture bei, meldete 2004 allerdings Insolvenz an. Nachdem das Format lizenzfrei zugänglich wurde, adoptierte es *Vicon* als Ausgabeformat seiner MoCap Systeme. Das Fileformat ist unter den nicht-binären Massenformaten vergleichsweise schlicht aufgebaut und wird von den meisten, kommerziellen Animationsprogrammen unterstützt. (Menache, 2011, S. 143), (Kitagawa & Windsor, 2008, S. 181,182)

Es besteht deshalb aus den Dateien ASF und AMC, da dasselbe Skelett oftmals für mehrere, unterschiedliche Bewegungen verwendet wird. So kann dieses in nur einer anstatt mehrerer Dateien untergebracht und wiederverwendet werden. Mehrere AMC Dateien beziehen sich in Folge immer auf eine bestimmte ASF Datei. Es darf pro Datei nur eine *Root* vorhanden sein, sodass sich zwei unterschiedliche Skelette derselben *Root* unterordnen müssen. (ebd.)

Die AFS Datei enthält sämtliche Informationen über die Hierarchy sowie betreffende, mechanische Funktionen des Skeletts. Darunter fallen *Bone*-Benennung, Freiheitsgrade, Multiplikatoren, Grenzwerte, Dokumentationen und Dateinamen von Skins. Zusätzlich enthält es die Ausgangsposition sowie -Ausrichtung. AFS teilt sich in acht Sektionen, vorangestellt von folgenden englischen Schlüsselwörtern, auf:

- **version** – Version der Skelettdefinition, z.B. version 1.10
- **name** – Name des Skeletts, unabhängig vom Dateinamen
- **unit** – deklariert eine Sektion, in welcher die Einheiten für die unterschiedlichsten Datentypen definiert sind. Die Interpretation dieser Sektion ist oft programmabhängig.
- **documentation** – erlaubt das Abspeichern von Dokumentationen. Die Information bleibt über mehrere Dateiversionen hinweg bestehen.
- **root** – Ist ein Element der *Bone*-Hierarchie und gleichzeitig der Ausgangspunkt eines Skeletts. Die *Root* verhält sich wie jeder andere *Bone*, enthält jedoch keine Richtungs- oder Längeninformation.

Das Schlüsselwort *Axis* beschreibt die Rotationsreihenfolge innerhalb der *Root*, *order* definiert die Zuordnung und Reihenfolge der Translationskanäle, z.B. TX TY TZ RX RY RZ.

- **bonedata** – Definiert alle *Bones*/Segmente relativ zur *Root*. Sie wird von einem *begin*- und *end*-node eingeklammert. Innerhalb der Klammer befinden sich weitere Definitionen:

„*id*“ deklariert jedes Segment mit einer eigenen Nummer/Namen.

„*name*“ vergibt jedem Segment einen Namen und macht „*id*“ eigentlich redundant. Häufig werden Segmente von Füllsegmenten begleitet. Diese haben den gleichen Namen wie das *Parent*-Segment. Sie sind mit einer zusätzlichen Nummer am Ende benannt (z.B. hips1). Sie enthalten zumeist keine Daten und agieren als Bindeglieder zwischen *Parent*- und *Child*-Segmenten.

„*direction*“ bestimmt die Darstellung und Ausrichtung des jeweiligen *Bone*

„*length*“ arbeitet in Kombination mit *direction* und bestimmt die dargestellte Länge des *Bones*

„*Axis*“ beschreibt die globale Achsausrichtung und die Reihenfolge der Ausgangsrotationen

„*dof*“ steht für den erlaubten Freiheitsgrad (degree of freedom) und dessen Transformationskanäle (tx ty tz rx ry rz l)

„*limits*“ setzt die Ober- und Untergrenzen der *dof* – Transformationskanäle fest

- **hierarchy** – Definiert die *Parent-Child*-Beziehungen innerhalb eines Skeletts
- **skin** – Beinhaltet eine Liste der passenden/zugehörigen 3D-Modelle

(Menache, 2011, S. 144)

Die Einheiten Sektion enthält die drei Felder *mass* (= Masse), *length* (= Länge) und *angle* (= Winkel). *Mass* wird gewöhnlich von der 3D-Software ignoriert und agiert als Multiplikator für segmentierte Masse. Die *length* dient als Multiplikator für die *Bone*-Länge. Das *angle*-Feld gibt an, ob die Orientierung in Grad oder Radiant definiert ist. Die gesamte Hierarchie steht im Verhältnis zur Position und Orientierung der *Root*. (ebd.)

Die AMC Datei enthält die eigentlichen Bewegungsdaten, welche sich alle auf die Definitionen in der ASF Datei beziehen. Die Felder sind gleichermaßen angeordnet wie in der *dof*-Sektion der ASF Datei. Für jeden Frame wird eine eigene Sektion gespeichert. Der Hauptvorteil des Acclaim-Format ergibt sich aus der beliebigen Reihenfolge an Rotationen oder Transformationen innerhalb der Skeletthierarchie. (Menache, 2011, S. 146)

#### 3.5.3 BVH

Dieses Datenformat wurde ursprünglich vom Unternehmen BioVision, spezialisiert auf Sportanalysen und Animationen, entwickelt. Im Gegensatz zum Vorgänger BVA (ASCII), enthält das binäre BVH Format Bewegungs- **und** Skelettdaten. Es ist ein optimales „all around“ Format, kann allerdings keine komplette Ausgangsposition definieren, da das Offset der *Child*-Segmente nur Translationen beschreibt. (Kitagawa & Windsor, 2008, S. 182), (University of Wisconsin Graphics Lab, 1999)

Die Datei teilt sich einerseits in die Hierarchie (HIERARCHY) - und zum Zweiten in die Bewegungsdaten (MOTION)-Sektion. Neben dem Skelettaufbau ist in der Hierarchie-Sektion der Quellcode in der Programmiersprache C eingebettet. Eine BVH-Datei kann theoretisch mehrere Skelette/Hierarchien enthalten, wird jedoch praktisch davon begrenzt, wie die Bewegungs-Sektion formatiert ist. Dort werden die Transformationen zeilenweise pro Segment vergeben, was bei unterschiedlichen Ausleseverfahren zu Problemen führen kann. (ebd.)

Jedes Segment enthält eine spezifische Reihenfolge an Transformationen und teilt sich auf in das OFFSET- und CHANNELS-Feld. OFFSET definiert die Ausgangswerte/-stellung mit Bezug auf das übergeordnete *Parent*-Segment. CHANNELS definiert die Art der Transformationskanäle (Translation und/oder Rotation) und deren gegebene Reihenfolge. Eine Skalierbarkeit ist nicht vorgesehen. (ebd.)

Die Bewegungsdaten-Sektion enthält die Dauer der Animation, gegeben in Frames, zugehörige Framerate und eine numerische Wertezeile in derselben Reihenfolge wie im CHANNELS-Feld, sodass die jeweiligen Segmente angesteuert werden. In den allermeisten Fällen besitzt die *Root* sechs-, und alle restlichen *Child*-Segmente drei Kanäle. (Kitagawa & Windsor, 2008, S. 183) (University of Wisconsin Graphics Lab, 1999)

#### 3.5.4 FBX

Der Hauptzweck des Formates ist es, 3D-Szenen zu beschreiben, es war ursprünglich nicht für MoCap-Daten vorgesehen. Die Entwicklung fand im Unternehmen *Kaydara* für deren Software *FilmBox* statt. Heute ist *FilmBox* unter dem Namen *MotionBuilder* bekannt, welches zu *Autodesk* gehört. *MotionBuilder* besitzt die Fähigkeit, alle anderen MoCap-Datenformate in FBX zu übersetzen. FBX wird von einer breiten Masse an 3D-Programmen unterstützt. (Kitagawa & Windsor, 2008, S. 183)

Das Format unterstützt:

- Geometrien
- Texturen
- Kameras
- Lichter
- Marker
- Skelettdaten
- Animationsdaten

Es ist wahlweise ASCII oder binär codiert und in der Programmiersprache C formatiert. FBX wird wiederum in mehrere Sektionen, inklusive Untersektionen unterteilt. Diese starten mit einem Schlüsselwort, gefolgt von einem Doppelpunkt. Die Hauptsektionen heißen **FBXHeadExtension** (Dateikopf-Erweiterung), **Definitions** (= Definitionen), **Relations** (= Relationen), **Connections** (= Verbindungen) und **Takes** (= Aufnahmen). Die gegenseitigen Beziehungen in der Skeletthierarchie werden in **Connections** bestimmt. Markerdaten (Translationen) sind unter **Relations** zu finden. Die Rotationswinkel der *Joints* werden in der **Takes**-Sektion gespeichert. (ebd.)

Motion Capture Daten können in diesem Format mit maximal 120 fps exportiert werden, auch wenn die Aufnahmebildrate weit darüber liegt. (Vicon Motion Systems Ltd., 2018a, S. 94)

### 3.6 Planung

Es besteht nicht selten die überstürzte Meinung, dass MoCap „einfach und schnell“ funktioniert. Das trifft bestenfalls dann zu, sofern einiges an Routine mit dem verwendeten System besteht. Der Prozess ist nicht vollautomatisch. (Kitagawa & Windsor, 2008, S. 13)

Unabhängig vom System, zählt die Planung zu den wichtigsten Vorbereitungsschritten einer erfolgreichen MoCap Aufnahme. Hier werden alle relevanten Themen aufgeteilt und organisiert. Grob lässt sich diese Phase in Vor-Aufnahmeplanung, Aufnahmeplanung und Entwurf der Produktionspipeline aufteilen. Eine gute Vorbereitung erspart Zeit, Geld und liefert bessere Endresultate.

#### 3.6.1 Script

Zu Beginn wird ein **Script** erarbeitet, welches die zu erzählende Geschichte in eine Art Fahrplan für alle Beteiligten verpackt. Es kann unterschiedliche Inhalte wie Drehbücher, Ideen zu Kurzfilmanimationen oder eine Shotlist für Videospiele usw. darstellen. Das aufzunehmende Subjekt sollte das Script so bald als möglich zur Verfügung gestellt bekommen, um die psychischen und physischen Anforderungen an seine Performance zeitnah klarzustellen.

Ein Script enthält üblicherweise folgende Bausteine:

- Szenen Kopf - Kurzbeschreibung wo und wann eine Szene stattfindet
- Handlung - Beschreibung aller sich bewegenden Elemente in einer Szene
- Dialog - Gespräche zwischen Charakteren
- Subjektnamen - steht über der jeweiligen gesprochenen Zeile
- Übergänge - Beschreibung der Übergänge in Einstellungen und Szenen
- Kameraeinstellung - z.B. *close-up*, *mid-shot*, *long-shot* etc.

(Kitagawa & Windsor, 2008, S. 14)

#### 3.6.2 Storyboard

Aufbauend auf dem Script entsteht dessen 2D-Visualisierung, das **Storyboard**. Es besteht aus einer Reihe an skizzenhaften Zeichnungen mit schriftlichen Hinweisen auf den Inhalt in jeder Zeichnung wie Subjektpersonal, Bühnenbild, Kameraeinstellungen, Filmschnitt und Audiodesign. Gewöhnlich unterliegt jedes Storyboard einigen Änderungen und Anpassungen, weshalb dieses in einem dementsprechend leicht modifizierbaren Format entworfen werden sollte. Jedes Storyboard verfolgt das Ziel, die eigenen Ideen zu konkretisieren und dem Team sowie Entscheidungsträgern so zu kommunizieren, dass jeder Beteiligte eine

unmissverständliche Vorstellung vom fertigen Endprodukt erhält. (Kitagawa & Windsor, 2008, S. 15)

Ein *Animatic* ist die animierte Version des Storyboards, welches zusätzlich grobe, visuelle Elemente (rohe Zeichnungen, Kamerafahrten, VFX, ...) und ein erstes Sounddesign enthält. Es hilft einzuschätzen, ob die Story effektiv erzählt wird. Häufig werden zu lange oder komplexe MoCap Aufnahmen durchgeführt, ohne der Story dienlich zu sein. Sollte sich herausstellen, dass eine Szene in einer Länge von zwei Minuten ebenso gut in 30 Sekunden funktioniert, so kann dies neben einer besseren Dynamik einen erheblichen positiven Einfluss auf das Budget mit sich bringen. (Kitagawa & Windsor, 2008, S. 16)

Zur Anfertigung von Animatics werden häufig Game-Engines herangezogen. In Kombination mit dem Echtzeit-Rendering der Engine und bereits vorhandenen MoCap Daten, lassen sich Änderungen schnell und unkompliziert visualisieren. Um Zeit zu sparen, kann für zukünftige Projekte eine Bibliothek mit typischen Bewegungsabläufen angelegt werden. (ebd.)

#### 3.6.3 Shotlist

Die **Shotlist** ist eine Liste der Ereignisse, welche im Gesamten die Szene ergeben. Sie liefert einen Überblick über die Komplexität sowie die benötigten Ressourcen innerhalb der Szene. Darunter fallen unter anderem die Anzahl der Schauspieler, der Tagesplan, die Zeit per Shot und die Requisiten. (Kitagawa & Windsor, 2008, S. 23, 24)

Die Shotlist ist einer der frühesten und wichtigsten Schritte in der Projektorganisation. Aktionsreiche Szenen benötigen längere Drehzeiten. Komplexe Inhalte erfordern wiederum eine entsprechende Schauspielperformance. (ebd.)

Im Sinne der Cinematographie definiert sich ein Shot als ein ohne Unterbrechung, über eine einzige Kamera geführter Blickwinkel. Ein- oder mehrere zusammenhängende Shots ergeben demnach die Szene. Es benötigt ein tieferes Verständnis dafür, wie die Shots anhand des Storyboards durchgeführt werden sollen, um später in der Postproduktions- und Schnittphase zu funktionieren. Die physikalischen Grenzen in Bezug auf Länge, Breite und Höhe des Aufnahmevervolumens sind dahingehend ein typischer Faktor. Die Shotlist wird daher als eine Ressourcenliste der Szenen betrachtet. (ebd.)

## 3.7 Nachbearbeitung

### 3.7.1 Data Cleanup und Post Processing

Je nach System werden nach der Aufnahme Marker-Daten oder Skelett-Daten abgespeichert. Es müssen beiderseits fehlerhafte Daten nachträglich entfernt oder fehlende Daten ersetzt werden. Dieser Prozess wird fachsprachlich als Cleanup bezeichnet. Je nach Performance eines Systems, kann sich der nötige Zeitaufwand erheblich summieren. Nachdem die Daten bereinigt wurden, stehen die Animationen für weitere Manipulationen bereit. (Kitagawa & Windsor, 2008, S. 39)

Der nötige Detailaufwand richtet sich nach dem gewünschten Endprodukt. Ist dieses für Kino- oder Fernsehproduktionen vorgesehen, kann eventuell Nachbearbeitung eingespart werden, da durch die verbindlich festgelegten Kameraperspektiven nicht jeder Bereich zwangsläufig sichtbar wird. Bei Videospielen und interaktiven Medien hat der Anwender allerdings umfangreiche Kontrolle über die Ansicht des Charakters, weshalb die Daten detailliert aufzubessern sind. (Kitagawa & Windsor, 2008, S. 39, 40)

Am häufigsten sind Probleme wie Verdeckungslücken, Störungen, falsch identifizierte Marker und hochfrequenter *Jitter* in den Daten inkludiert, welche die Software vorerst selbstständig auszugleichen versucht. Gelingt ihr dies nicht, müssen manuelle Korrekturschritte durchgeführt werden. Aus Zeitgründen ist die Handkorrektur stets zuletzt, nach der Verwendung aller zur Verfügung stehenden, automatischen Werkzeuge in Betracht zu ziehen. (Kitagawa & Windsor, 2008, S. 39)



Für gewöhnlich besitzt jede proprietäre Software eines MoCap Systems eigene, maßgeschneiderte Werkzeuge. Sollten diese nicht ausreichend sein, muss auf weitere Softwarepakete wie *MotionBuilder* ausgewichen werden.

In Kombination mit dem *Graph Editor* stehen in *MotionBuilder* automatische Filter zur Verfügung, welche sich jeweils auf bestimmte Aufgaben in den Bewegungspfaden der Marker bzw. erzeugter Keyframes spezialisieren:

- Butterworth Filter
- Constant Key Reducer Filter
- Cut Filter
- Gimbal Killer Filter
- Key Reducing Filter
- Key Sync Filter
- Peak Removing Filter
- Reinterpolate Filter
- Resample Filter
- Smooth Filter
- Smooth Translation Filter
- Time Shift Filter und Scale Filter
- Unroll Rotations Filter

(Autodesk Inc., 2017)

Filter sind mathematische Funktionen, welche hoch- und niederfrequente Störungen beseitigen oder andere Modifikationen an den ursprünglichen Daten vornehmen. Jedes MoCap System enthält einen festen Störanteil in den Daten, der systembedingt unterschiedliche Formen und Ausprägungen annehmen kann. Oberflächlich betrachtet erscheinen die Daten oft präzise. Im Detail sind erwähnte Störungen hingegen sichtbar. Der *Butterworth* Filter ist ein effizienter und häufig angewandter Lo-Pass Filter zum Eliminieren von hochfrequenten Störungen. Filter sollten immer nur so viel als nötig eingesetzt werden, um die subtilen Feinheiten einer nonlinearen Bewegung nicht zu zerstören. (Kitagawa & Windsor, 2008, S. 59, 60, 61)

Ein weiteres, häufiges Artefakt betrifft ein Abdriften der Füße bzw. die Fehlinterpretation der Fußkontakte mit dem Boden, bekannt unter *Footskate*. Dies hat systembedingt unterschiedliche Ursachen. Bei markerbasierten Systemen liegt die Ursache zumeist in *Retargetingprozessen* oder einem schlecht kalibrierten System. Bei inertialen Systemen trägt zusätzlich eine Störung des Referenzmagnetfelds oder ein schlecht kalibriertes Subjekt (z.B. falsche Vermessungen, falsche Position der Sensoren ...) zum Abdriften bei. Darüberhinaus durchlaufen die Skelettdaten im späteren Verlauf eine Vielzahl an

Manipulationen wie *Retargeting*, Zeitverzerrungen, Kurskorrekturen, Überblendungen und Filterprozesse, welche allesamt zur Verstärkung oder Entstehung von Artefakten beitragen. *Footskate* ist ein störender Fehler, da Fußkontakte den Charakter mit seiner Umgebung verbinden und bereits geringe Abweichungen als unrealistisch empfunden werden. (Kovar, Gleicher, & Schreiner, 2002, S. 1)

Markerdaten und Skelettdaten weisen neben ihrem unterschiedlichen Informationsgehalt eine alternative Hierarchie auf. Die Hierarchie beschreibt ein System aus Beziehungen von mehreren Elementen zueinander, wobei jedes Element einem anderen stärkeren Element untergeordnet wird. Für gewöhnlich besteht eine skelettale Hierarchie aus einer universellen *Root* und mehreren untergeordneten *Child*-Elementen. (Kitagawa & Windsor, 2008, S. 40)

Markerbasierte Systeme arbeiten mit einer flachen Hierarchie, d.h. alle der *Root* untergeordneten Elemente sind auf einer gemeinsamen Ebene und somit voneinander unabhängig. Der entscheidende Vorteil liegt darin, dass etwaige Fehler eines Markers nicht auf benachbarte Marker übergreifen. (ebt.)

Skelettale Daten hingegen verwenden eine weitaus komplexere *Joint*-Struktur. Die *Root* bildet für gewöhnlich der *Hip-Joint*. Von dort ausgehend sind Ober- und Unterkörper dem *Hip-Joint* untergeordnet. Diese Hierarchie verzweigt sich in immer tiefere Abstufungen, bis die untersten Elemente wie Zehen- oder Fingerspitzen erreicht werden. Fehlerhaft getrackte *Hip-Joints* wirken sich automatisch auf das gesamte Skelett aus. (ebt.)

#### 3.7.2 Retargeting

Nach dem praktischen Vergleichstest aus Kapitel 4, sollen die Daten in *Autodesk MotionBuilder* geplottet, sprich Rotationen und Translationen der Exportdatei auf das Skelett einer repräsentativen 3D-Figur aufgetragen werden. *Autodesk MotionBuilder* ist häufig in MoCap-Pipelines vertreten und bietet eine Vielzahl an professionellen Werkzeugen zur Weiterverarbeitung der rohen Bewegungsdaten. (Menache, 2011, S. 189)

Vorerst ist es nötig, die einzelnen *Bones* des Source-Skeletts dem Target-Skelett zuzuweisen. Dieser Prozess wird *Retargeting* genannt. Der Aufwand für das *Retargeting* richtet sich nach der Gleichheit in der *Bone*-Struktur. Die Zuweisung wird entweder manuell oder vorzugsweise automatisch durchgeführt, indem die Exportdatei auf das Target-Skelett gezogen wird. Dazu muss die Namenskonvention des Source-Skelett in der Trackingsoftware der des Target-Skelett in der externen 3D-Animationssoftware (z.B. *MotionBuilder*, *Maya*, *Blender*...) entsprechen. Nicht jede 3D-Animationssoftware benutzt dieselbe Namenskonvention. Deshalb sollte die proprietäre Trackingsoftware zumindest die

gängigsten Formate beherrschen, welche in die Exportdatei implementiert werden. (Kitagawa & Windsor, 2008, S. 43, 44, 69)

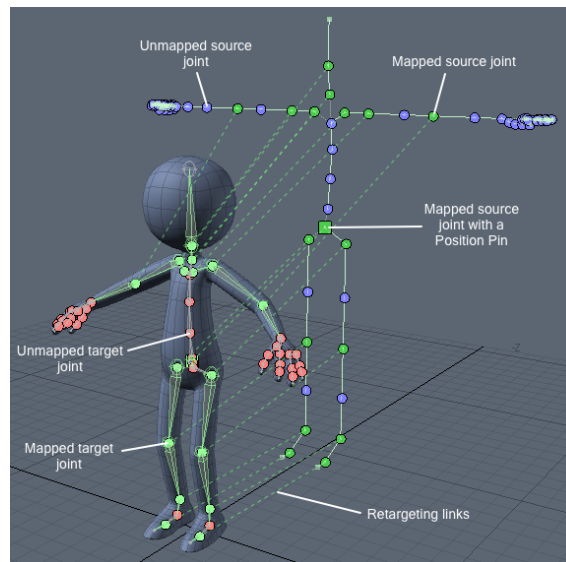


Abbildung 14. Retargeting Beispiel (learn.foundry.com)

Der *Retargetingprozess* variiert je nach Datentyp. Der vorangegangene Absatz beschreibt die Übertragung von skelettalen Daten, welche aus Translationen sowie Rotation bestehen (iPi Soft, Perception Neuron). Markerdaten hingegen beinhalten nur X, Y und Z Koordinaten, weshalb diese erst über einen zusätzlichen Zwischenschritt konvertiert werden müssen, besser bekannt unter **Solving**. (Kitagawa & Windsor, 2008, S. 63,64)

In *MotionBuilder* lässt sich dazu ein sogenannter *Actor* in die Szene importieren, welcher ein human geformtes Helferobjekt darstellt. Das Helferobjekt wird anschließend nach Augenmaß an die Markerpositionen angepasst bzw. ausgerichtet. Optimalerweise steht die Markerwolke anhand einer *T-Pose* zur Verfügung, da so die Ausrichtung wesentlich genauer vorgenommen werden kann. Sobald die Bewegungen auf das Helferobjekt umgerechnet wurden, können dessen Translations- und Rotationsdaten wiederum in skelettale Daten konvertiert werden. (ebd.)

## 4 Praktische Analyse

### 4.1 Einführung

Das folgende Kapitel analysiert und vergleicht zwei unterschiedliche Günstig-MoCap Techniken miteinander und eruiert zugleich die Unterschiede zur Performance des Industriestandards. Dazu wird vorab die Konstruktion der einzelnen Systeme aufgeschlüsselt.

Danach findet ein praktischer Direktvergleich zwischen *iPi Soft* und dem inertialen *Perception Neuron* Anzug anhand einer Simultanaufnahme, unter den kontrollierten Bedingungen eines professionellen Videostudios statt. Die virtuelle Szene, inklusive virtueller Kamerapositionen, lässt sich aus der Software *iPi Mocap Studio* als FBX Datei verpacken und einem externen 3D-Softwarepaket wie *Autodesk MotionBuilder* rekonstruieren. Damit kann zur Nachbearbeitung die Originalaufnahme als Video-Overlay geladen und als reale Referenz herangezogen werden.

Nach der Wandlung der Daten in skelettale Animationen, werden diese für jedes System aus zwei unterschiedlichen Kameraperspektiven über die Originalaufnahme gelegt und die dabei entstehende Multikameraaufnahme zu einem Videofile kompiliert. Damit ist ein einheitlicher 1:1 Präzisionsvergleich der beiden Systeme gegeben.

Am Ende des Kapitels wird anhand aller bisherigen Erkenntnisse eine Gewichtung und Bewertung der Kriterien erstellt, welche die Grundvoraussetzungen eines professionellen MoCap Systems definieren und zudem die Basis für eine ganzheitliche Nutzwertanalyse liefern. Das Resultat soll einen wirtschaftlichen Anhaltspunkt für zukünftige Projekte im Bereich des Motion Capturing bieten.

Die Animationen der SFU-Datenbank dienen als Referenz für qualitative Aussagen über die optische Qualität des Industriestandards.

Auf die rohen Markerdaten kann keine Einsicht genommen werden, da sich das proprietäre Datenvormat HDF der *Vicon Blade* Software (Vorgänger von *Vicon Shogun*) nur über eine kostenpflichtige Lizenz bearbeiten lassen. Alternativ stehen vorbereitete Daten im C3D Format zur Verfügung.

Die Datenbank bietet keine spezifischen Angaben zum verwendeten *Vicon* System. In der Nutzwertanalyse soll dafür stellvertretend das Leistungsspektrum des *Vicon Vero 2.2* Systems herangezogen werden, da dieses am ehesten der Zielgruppe von *Perception Neuron* und *iPi Soft* entspricht. (Vicon Motion Systems Ltd., 2017e)

## 4.2 Vicon – Markerbasiertes Motion Capture

Das Unternehmen konzentriert sich auf drei wesentliche Kernbereiche, **Life-Sciences**, **Entertainment** und **Engineering**. Für jeden dieser Bereiche wird eine entsprechende Trackingsoftware (*Nexus/Shogun/Tracker*) sowie variierende Kamertypen empfohlen. Die Typen richten sich nach der Raumbeschaffenheit, Objektanzahl oder Objektart. ([www.vicon.com](http://www.vicon.com))

Der folgende Abschnitt geht von einer Verwendung im Entertainmentbereich aus.

### 4.2.1 Hardware

Gegenwärtig stellt *Vicon* zwei Produktgruppen zur Verfügung, die *Vantage*- (High-End) und die *Vero*-Familie (Low-End). Daraus ergeben sich sieben Kamertypen mit divergenten Eigenschaften bezüglich Auflösung, Framerate, Brennweite, Latenz uvm. Ein simples Ganzkörper-MoCap für 1 – 2 Personen lässt sich mit dem Modell *Vero v2.2* und einer Anzahl von 10 Kameras ausreichend umsetzen. Die maximale Bildrate liegt mit einer maximalen Auflösung von zwei Megapixeln bei 330 fps. Mit diesem Setup sind Aufnahmevolumen bis 5 x 5 Meter vorgesehen. Ab drei Personen sollte auf 16 Stück *Vantage V5* Kameras (V5 = fünf Megapixel/Kamera) zurückgegriffen werden. Für Performance Capture wird das High-End Topmodell *Vantage V16* in einer Anzahl von 12 Kameras empfohlen. (ebd.)

Die Investitionssummen der drei Systeme sind nach Angaben des Schweizer Vertriebspartners *prophysics AG* ([www.prophysics.ch](http://www.prophysics.ch)) wie folgt:

- 10 x *Vero v2.2* Setup > 60.000 – 80.000 EUR zzgl. MwSt.
- 16 x *Vantage V5* Setup > 150.000 – 180.000 EUR zzgl. MwSt.
- 12 x *Vantage V16* Setup > 180.000 – 210.000 EUR zzgl. MwSt.

Die Setups beinhalten alle Geräte und Kabel, Kameramontage, Software, Installation und Support.

Die Kamera selbst setzt sich aus Hauptgehäuse, Ringleuchte, Kameralinse und einem optischen Filter zusammen. Die Ringleuchte existiert in unterschiedlichen Ausführungen und kann beliebig an die Aufnahmesituation angepasst werden. Unterstützt wird der proprietäre *Vicon*- oder *C-Mount* Objektivanschluss. Alle *Vantage* Kameras sind mit Festkörpersensoren ausgestattet, welche Linearität, Sensitivität und Anwesenheit von Erschütterungen prüfen sowie gegebenenfalls melden. (*Vicon Motion Systems Ltd.*, 2016, S. 30, 31, 33, 34)

Die Standard-Ringleuchte an der Vorderseite der Kamera besteht aus Hochleistungs-LEDs, welche das Aufnahmevolumen mit infraroten Lichtimpulsen in einer Wellenlänge von 850nm bestrahlt. Pro Frame wird ein Impuls genau dann ausgesandt, sobald sich der globale, elektronische Verschluss des Bildsensors öffnet. Der optische Filter sorgt dafür, dass ausschließlich die Wellenlänge der retroreflektierenden Marker den Bildsensor erreicht. Sollte das System im Freien zum Einsatz kommen, kann natürliches Licht die Präzision beeinträchtigen. (Vicon Motion Systems Ltd., 2016, S. 35, 36)

Die proprietären Linsen sind speziell für MoCap angefertigt und besitzen einen großen Abbildungskreis, sodass das gesamte Bild gleichmäßig ausgeleuchtet wird. Linsen von Drittherstellern können unter Umständen das Bild nicht fein genug auflösen und verursachen ein Verschmelzen von eng benachbarten Markern. (Vicon Motion Systems Ltd., 2016, S. 42)

*Vantage* Kameras evaluieren das gesamte Bild in Graustufen. Gegenüber einem Schwarz-Weiß Bild können so mehr Informationen ausgelesen und eine höhere Präzision erreicht werden. Das genaue Zentrum der daraus erzeugten Lichtpunkte wird über Zentroid-Algorithmen errechnet. Erst danach stehen die Daten zur Weiterverarbeitung in der Trackingsoftware des Host-PC bereit. (Vicon Motion Systems Ltd., 2016, S. 29)

Für perfekt synchronisierte Video-Overlays mit einer Bildauflösung bis zu 1080p, können *Vicon's* Kameramodelle *Vue* oder *Bonita* als Bestandteil eines *Vicon* Systems kalibriert werden. *Vue* und *Bonita* übertragen dabei reine Videosignale und keine MoCap Daten. Die Möglichkeit eines Video-Overlays bietet eine enorme Hilfestellung bei manuellen Korrekturmaßnahmen. (Vicon Motion Systems Ltd., 2018a, S. 16)

Mit *Vicon's Lock+ Sync Boxen* (Lock Lab und Lock Studio) lassen sich bis zu acht analoge Geräte von Drittherstellern (z.B. Forceplates) über *VESA*-Schnittstellen mit dem bestehenden System verbinden, synchronisieren, integrieren und ansteuern. (Vicon Motion Systems Ltd., 2018b)

Die Daten der Kameras sowie der Lock Boxen laufen auf einem PoE oder PoE+ Switch (Power over Ethernet) zusammen, welcher wiederum mit dem Host-PC inklusive Trackingsoftware verbunden ist. Ein einzelner PoE+ Switch hat 16 Netzwerkeingänge und kann insgesamt 12 MoCap-Kameras aufnehmen. Ab 13 Kameras wird demnach eine weitere PoE+ Einheit sowie ein zusätzlicher Gigabit Switch benötigt. *Vicon* Systeme können nur in einem dedizierten Netzwerk und über einen fest zugeteilten Netzwerkeingang am Host-PC agieren. Digitale Drittgeräte können zusätzlich direkt mit dem Host-PC verbunden werden. (Vicon Motion Systems Ltd., 2016, S. 9, 10, 11)

Die Mehrheit der Geräte wird über Cat5e Netzkabel mit RJ-45 Verbindungen miteinander verbunden. Diese sorgen für die Stromzufuhr, den Datenaustausch und die Synchronisationssignale. Die Kabel sollten gut geschirmt sein und einen Ferritkern am Kameraende tragen. (Vicon Motion Systems Ltd., 2016, S. 64)

Jedes System hat eine Master-Einheit, welche die Synchronisationszeiten für die restlichen Geräte vorgibt. Je nach Konfiguration, kann der Master eine Kamera, eine Lock+ Box oder ein Gigabit Switch sein. (Vicon Motion Systems Ltd., 2016, S. 18)

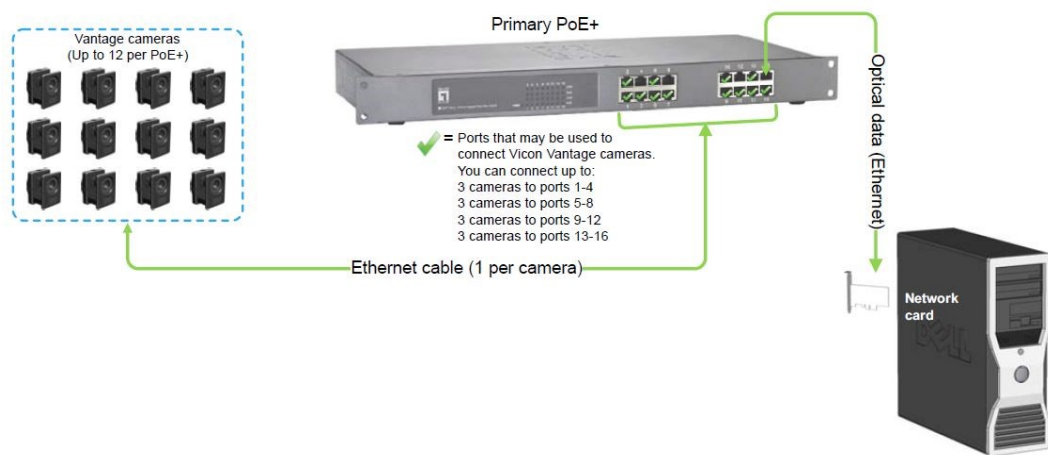


Abbildung 15. Basis-Setup mit 12 Vicon Vantage Kameras

Zum Kalibrieren der MoCap- sowie Videokameras gibt es einen speziellen, mit fünf LED Paaren bestückten Kalibrierungsstab in T-Form. Damit lassen sich der Ursprungspunkt und die Achsenausrichtung des Aufnahmevolumens definieren. (Vicon Motion Systems Ltd., 2016, S. 76)

Sofern ein eigener Computer als Host-PC eingesetzt werden soll, sind dessen hohen Hardwareanforderungen zu beachten. Je nach Umfang des Systems muss vor allem in Grafikleistung investiert werden. Die minimalen Systemanforderungen sind wie folgt:

- mATX 4 drive bay Gehäuse
- INTEL® CORE™ i9-7900X Prozessor
- Corsair Hydro H100i Cooler
- 32GB (2x16GB) 2666MHz RAM
- GeForce 1050Ti
- 2 off 250GB SSD Boot Drive Read 530MB/s
- 1 off 3TB WD Black 7,200RPM HDD
- 650W 80 PLUS Platinum Netzteil
- Asus X99 Motherboard
- 1 off 4 Port Intel I350-T4 Gigabit Netzwerkkarte
- 64 bit Windows 10 Betriebssystem

(Vicon Motion Systems Ltd., 2016, S. 58), ([www.vicon.com](http://www.vicon.com))

### 4.2.2 Setup

Die Genauigkeit des Systems wird wesentlich von der Anatomie des erfassten Körpers beeinflusst. Das Ziel ist es, dessen Skelett so genau als möglich anhand der Markergruppen oder sonstiger Konfigurationsmöglichkeiten darzustellen. Befinden sich die Marker an ungeeigneten Stellen (z.B. ausgeprägte Muskelpartien, komplexe Rotationen, gelenksnah, ...), wird die Genauigkeit der Daten negativ beeinflusst. Jeder Hersteller empfiehlt für sein System ein eigenes Marker- bzw. Konfigurations-Schema, von welchem ausgegangen werden sollte. Außerdem ist es ratsam, sich mit den muscolo-skelettalen Zusammenhängen des Menschen anhand einschlägiger Fachliteratur vertraut zu machen. (Kitagawa & Windsor, 2008, S. 19, 20)

Vicon definiert die Markerpositionen in einem *Vicon Labeling Skeleton Template* (VST) für generische Subjekte oder einem *calibrated Vicon Labeling Skeleton* (VSK) für spezifische Subjekte. Die Dokumentation der jeweiligen Trackingsoftware gibt Aufschluss darüber, wie die Marker entsprechend dieser Vorgaben anzubringen sind. Anhand der VSK- bzw. VST- Dateien kann *Blade* die Bewegungen richtig interpretieren und automatisch ein 3D-Skelett generieren,



welches wiederum ein 3D-Mesh ansteuert. (Vicon Motion Systems Ltd., 2016, S. 26)

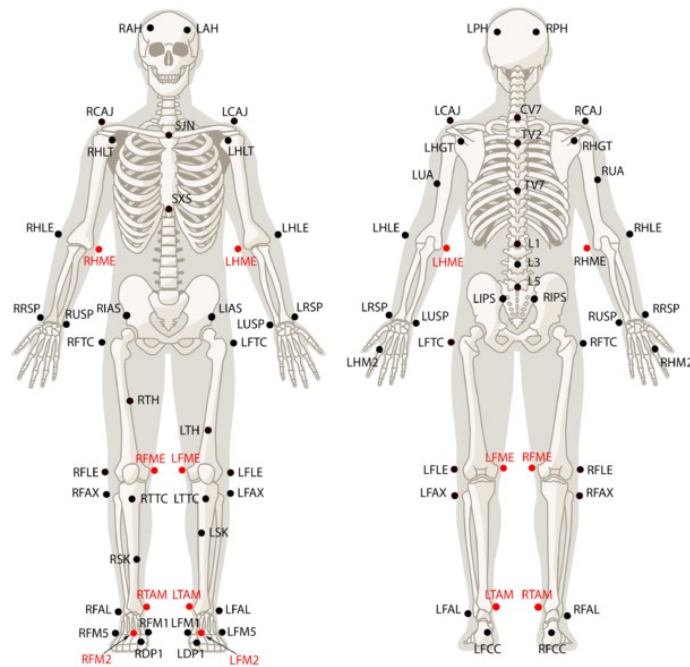


Abbildung 16. Beispiel einer Markerkonfiguration

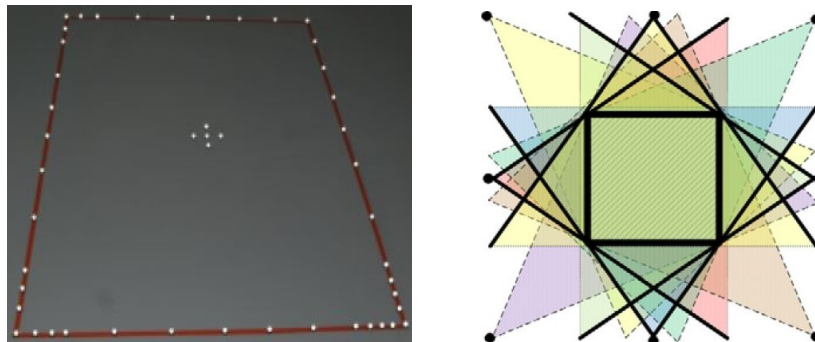
Die Art der Bewegungen kann es nötig oder sinnvoll machen, die Position der Marker zu ändern bzw. an bestimmten Stellen stärker zu konzentrieren. Liegt der Schauspieler beispielsweise nur am Bauch, so ist es ratsam, die Brustmarker auf den Rücken umzulagern. Diese Maßnahme sollte nur dann durchgeführt werden, wenn derartige Sequenzen einen überwiegenden Teil im gesamten Aufnahmezeitraum einnehmen. (Kitagawa & Windsor, 2008, S. 19)

Für Aufnahmen mit mehreren Subjekten kommen sogenannte Marker-Cluster zum Einsatz. Diese Cluster sind individuelle Muster aus mindestens vier Markern in einem eingegrenzten Bereich, wodurch die Software die Subjekte voneinander unterscheiden kann. Für gewöhnlich werden Cluster am Torso angebracht. (Vicon Motion Systems Ltd., 2016, S. 30, 33)

Um das Aufnahmenvolumen in den Kameras sichtbar zu machen, werden mehrere Marker rund um dessen Außengrenzen platziert. Der Ursprungsmittelpunkt wird über fünf Marker als Kreuz dargestellt. (Vicon Motion Systems Ltd., 2016, S. 8)

Die genaue Position der Kameras kann je nach Größe des Aufnahmevolumens, Kameratyp und Kameraoptik variieren. Grundsätzlich werden die Kameras rund um das Aufnahmevolumen angeordnet und zielen dabei alle zu dessen Ursprungsmittelpunkt. Die erforderliche Aufnahmehöhe muss ebenfalls berücksichtigt werden. Je nach Bedarf können die Kameras bodennah auf Stativen

oder hoch auf einem Gestänge mit einer zusätzlichen Universalklammer (z.B. Manfrotto Superclamp) montiert werden. (Vicon Motion Systems Ltd., 2016, S. 27)



*Abbildung 17. Aufnahmevolumen und Kamerapositionen eines Vicon Setups*

Nach dem Starten der Trackingsoftware, wird das Live-Bild der jeweiligen Kameras sichtbar. So lässt sich die optimale Schärfe der Marker über den manuellen Fokusring an der Kamera ermitteln. Die Blende wird danach soweit geschlossen, bis so wenig Hintergrundrauschen wie möglich erkenntlich wird, ohne dabei die Qualität der Markerabbildungen zu beeinträchtigen. Die Marker sollten gerade soweit ausgeleuchtet sein, dass sie im Zentrum ca. 90% Weiß und zum Rand hin abfallend Grau dargestellt werden. (Vicon Motion Systems Ltd., 2018a, S. 11, 12)

#### 4.2.3 Kalibrierung

Nachdem die Kameras sachgemäß ausgerichtet und adjustiert wurden, muss das System eine Zwei-Phasen-Kalibrierung durchlaufen. Die Dimensionen und relativen Markerpositionen des spezifischen Kalibrierungsgeräts sind in einer VSK Datei definiert und müssen zuvor in der jeweiligen Trackingsoftware des Host-PC geladen werden. (Vicon Motion Systems Ltd., 2016, S. 79)

Danach kann das Gerät bzw. dessen LEDs aktiviert und mit einer Kombination aus welligen und kreisenden Bewegungen solange durch das leere Aufnahmevolumen geführt werden, bis in jedem Kamerabild eine gleichmäßige Verteilung aller Lichtpunkte vorhanden ist. Anhand der Bewegungen des Kalibrierungsstabs, kalkuliert die Software die physische Position, Ausrichtung, Brennweite und Linsenverzerrung jeder *Vicon* Kamera im Aufnahmevolumen. Linsenverzerrungen und andere nonlineare Fehler werden über eine Korrektur-Matrix für jeden einzelnen Frame erneut ausgeglichen. (Vicon Motion Systems Ltd., 2016, S. 80)

In der zweiten Phase muss der Stab flach am zuvor festgelegten Ursprungsmittelpunkt des Aufnahmevolumens gelegt werden. Die T-Form definiert die X- und Y-Achse des Koordinatensystems, aus welcher automatisch die vertikale Z-Achse errechnet wird. (Vicon Motion Systems Ltd., 2016, S. 81)

### 4.2.4 Softwareaufbau

Für den Entertainment- bzw. Animationsbereich lautet die Kernsoftware *Vicon Shogun*. Sie teilt sich wiederum in die Programme *Shogun Live* und *Shogun Post* auf. (Vicon Motion Systems Ltd., 2017a)

*Shogun Live* deckt alle Bereiche rund um das System-Setup, Kalibrierung, Bewegungserfassung und Aufzeichnung ab. *Shogun Post* befasst sich mit der Datenanalyse sowie -Aufbereitung und dem Export zu Softwarepaketen von Drittherstellern. (ebd.)

Das GUI ist, wie bei den meisten heutigen professionellen Softwarepaketen, modular aufgebaut und lässt sich in benutzerdefinierten Presets abspeichern. Die Oberfläche teilt sich in drei wesentliche Bereiche. Im 3D-Arbeitsbereich [1] lassen sich die MoCap Daten Live in unterschiedlichen Darstellungsmodi wie *Centroid*, *Greyscale*, *3D-Overlay* usw. betrachten. Ebenso finden sich in benachbarten Reitern die Streams sämtlicher Kameraperspektiven. (ebd.)

Links befinden sich die Panele für globale Einstellungen [2] betreffend dem System, dem Subjekt oder der Echtzeit-Datenverarbeitung. Unter *System* lassen sich z.B. alle im System integrierten Geräte wie Kameras und Switches anzeigen und spezifische Änderungen an denselben vornehmen. *Subject* listet unter anderem alle in der Szene vorhandenen Objekte wie Personen und Gegenstände (Schwerter usw.) sowie deren Aufbau anhand der Marker und *Bones*. (ebd.)

Der rechte Bereich beinhaltet die Panele für Kalibrierungs-, Aufzeichnungs- und Streamingaufgaben. Alle Aufnahmen in *Shogun Live* werden im proprietären MCP Format abgespeichert. Die Datei enthält die aktuellen Kalibrierungsdaten, das aktive Subjekt und verarbeitete Echtzeitdaten mit Pfaden zu den XML Dateien der rohen 2D Kameradaten. Alternativ lassen sich die Skelettdaten in Echtzeit zu den Game Engines *Unity* und *Unreal*, sowie *Autodesk MotionBuilder* oder direkt auf eine Festplatte streamen bzw. speichern. (Vicon Motion Systems Ltd., 2018a, S. 40)

Die Echtzeitperformance richtet sich nach dem Umfang des verwendeten Systems und der Fähigkeit des Host-PC, die aufkommenden Datenmengen schnell genug zu verarbeiten. Je unausgeglichener das System, desto mehr *Drop Frames* (= verworfene Bilder) entstehen. Das Echtzeit-*Solving* in *Shogun* ist stabil. Erst eine Droprate ab 30% wird als kritisch erachtet. Zudem ist die Software in der Lage, die Bewegungspfade großflächig verdeckter Marker sofort zu rekonstruieren (Occlusion Fixing). (Vicon Motion Systems Ltd., 2017c)

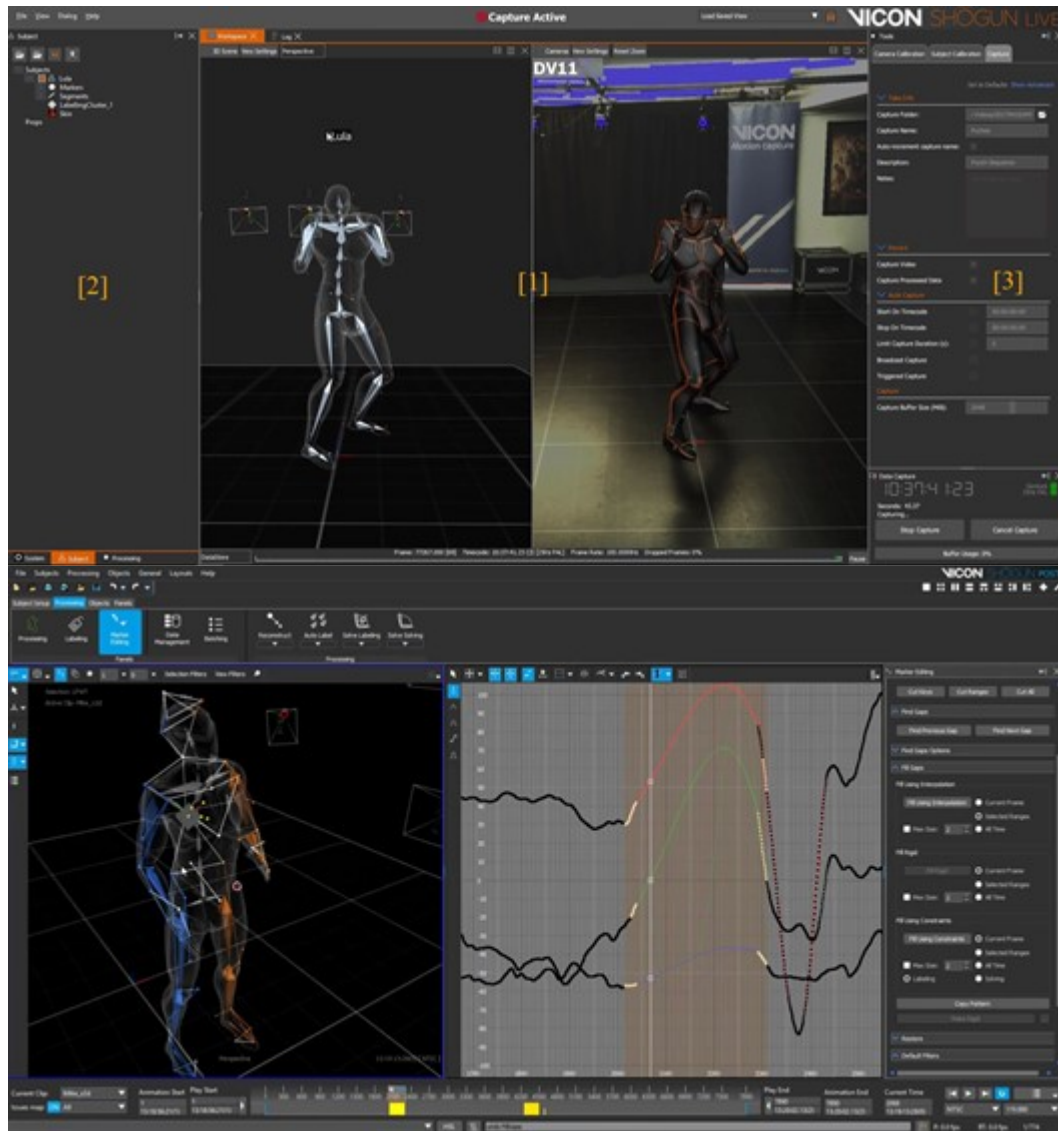


Abbildung 18. GUI der Shogun Live- (o.) und Shogun Post Software (u.)

Die Werkzeuge zur Datennachbereitung in *Shogun Post* sind umfangreich und leistungsfähig. Nach dem Import der MCP Datei steht man vor der Wahl, Aufzeichnungslücken automatisch oder manuell zu füllen. Eine automatische Korrektur rekonstruiert und identifiziert die fehlenden Marker der *Drop-Frames*. Danach muss die Qualität des automatischen Occlusion Fixing überprüft werden. Der Algorithmus ist eine an- und abschaltbare Einstellungsebene, weshalb ein direkter Abgleich mit den Rohdaten ermöglicht wird. Er wirkt sich auf **alle** Markerdaten eines Subjekts aus, sodass fehlerfrei getrackte Marker ebenfalls von einer Korrektur betroffen sein können. Occlusion Fixing betrachtet die Markerdaten am Anfangs- und Endpunkt einer Aufzeichnungslücke. Wurden hier die Markernamen vertauscht oder beinhalten sie sonstige Unstimmigkeiten, kann der Algorithmus die fehlenden Marker nicht richtig rekonstruieren. Darum müssen

derartige Fehler vor der Anwendung identifiziert und manuell behoben werden. (Vicon Motion Systems Ltd., 2018a, S. 57, 58, 59)

Für eine effiziente Fehlersuche und -Korrektur stehen drei grundsätzliche Werkzeuge zu Verfügung, eine *Time Bar Issue Map*, ein *Data Health View* und ein *Graph Editor*.

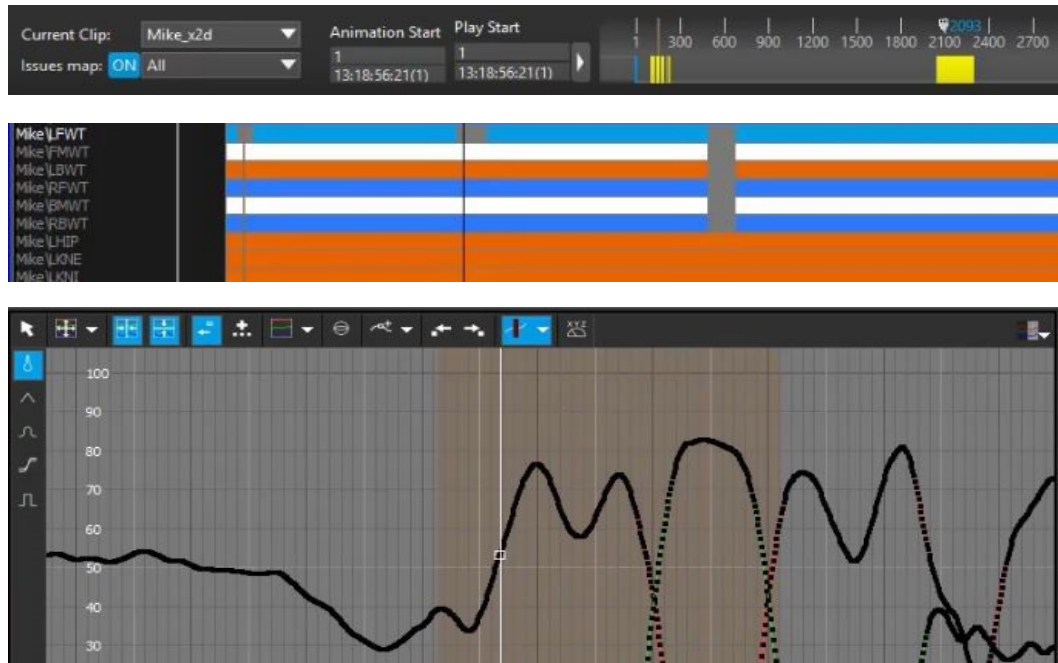


Abbildung 19. *Time Bar Issue Map*, *Data Health View* und *Graph Editor* (v.o.n.u.)

Der obere Balken der *Time Bar Issue Map* gibt Auskunft über den Prozentsatz benannter Marker. Der untere Balken zeigt den Prozentsatz an Markern mit einer hohen Fehlerrate, verursacht durch Fehlbenennungen, schlechtes *Solving* usw. Je länger ein Fehler auftritt, desto länger breitet sich der gelbe Balken im Zeitverlauf aus. Der Farbcode reicht von Gelb bis Rot. Je roter sich der Balken färbt, desto mehr Marker sind von Fehlern betroffen. (Vicon Motion Systems Ltd., 2018a, S. 66)

Welche Marker in welchem Zeitraum genau von Fehlern oder Aufzeichnungslücken betroffen sind, kann im *Data Health View* herausgefunden werden. (Vicon Motion Systems Ltd., 2018a, S. 68)

Die Keyframes, Bewegungspfade und Lücken der Marker auf ihrer X, Y und Z Achse lassen sich im *Graph Editor* detailliert darstellen und gegebenenfalls reparieren. Der Interpolationsalgorithmus eignet sich perfekt für kleinere Lücken mit linearem Bewegungsverlauf. Große Lücken mit nonlinearer Charakteristik lassen sich anhand von Mustererkennungen in vordefinierten *Rigid Bodies* rekonstruieren. *Rigid Bodies* sind feste Körperbereiche (z.B. Arm, Kopf, Torso, ...), die durch mindestens vier Marker repräsentiert werden, welche automatisch in

Beziehung zueinander stehen. Ist einer dieser Marker durch einen Fehler betroffen, kann dessen Position anhand der Koordinaten der restlichen Marker innerhalb des *Rigid Bodies* errechnet werden. (Vicon Motion Systems Ltd., 2018a, S. 88, 89), (Kitagawa & Windsor, 2008, S. 56)

Fehlen mehrere Marker eines *Rigid Bodies*, versucht die *Fill Using Constraints* Option, anhand des *Solving*-Skeletts, die Position der Marker vorherzusagen. Der Erfolg dieser Methode hängt davon ab, ob umliegend ausreichend Markerdaten vorhanden sind, und ob sich die betroffenen *Bones* sowie deren fehlenden Marker in einer sinnvollen Position befinden. (Vicon Motion Systems Ltd., 2018a, S. 89)

Durch Kalibrierungsfehler oder sonstige Störeinflüsse kann hochfrequenter *Jitter* in den Daten auftreten. *Shogun Post* bietet dazu eigene Filter in verschiedenen Ausprägungen an. Die Filter verhalten sich additiv und können beliebig oft angewandt werden. (Vicon Motion Systems Ltd., 2017b)

## 4.3 Perception Neuron – Inertiales Motion Capture

### 4.3.1 Hardware

Das Set eignet sich laut Hersteller für verschiedenste MoCap Anwendungen. Im VFX Bereich können die getrackten Daten via dem BVH oder FBX Format, in alle gängigen 3D-Programme wie *Maya*, *MotionBuilder*, *Blender* etc. importiert werden. Die Integration in ein Head Mounted Display (HMD) für VR/Game Interaction funktioniert ebenso fließend. Die Outputs der rohen G-Sensor- und Gyroskop Daten eignen sich ebenfalls für medizinische oder sportliche Analysen. Das Auswerten und Vergleichen der Daten wird über ein frei verfügbares Analysewerkzeug ermöglicht. Der Datenstrom lässt sich in Echtzeit via Wi-Fi an externe Geräte weiterleiten, um z.B. bei Bühnenperformances eingesetzt zu werden. (Noitom LTD, 2018a, S. 7)

*„PERCEPTION NEURON is the world's most versatile, adaptable and affordable motion capture system offering user friendly technology for various applications.“ (Noitom LTD, 2018a, S. 6)*

Grundsätzlich besteht der Anzug aus bis zu 32 kleinen IMU Sensoren namens Neuronen, den zugehörigen Sensorsockeln mit einem Klettband, zur individuellen Befestigung an den Gliedmaßen und dem HUB (Knotenpunkt), welcher die Datenströme synchronisiert und an den Host-PC weiterleitet. Die Stromversorgung der Geräte erfolgt über eine externe, mobile Stromquelle (Powerbank). Je nach verwendeter Sensoranzahl kann zwischen einer Samplerate von 60 fps (< 18 Neuronen) oder 120 fps (> 18 Neuronen) gewählt werden. (ebd.)



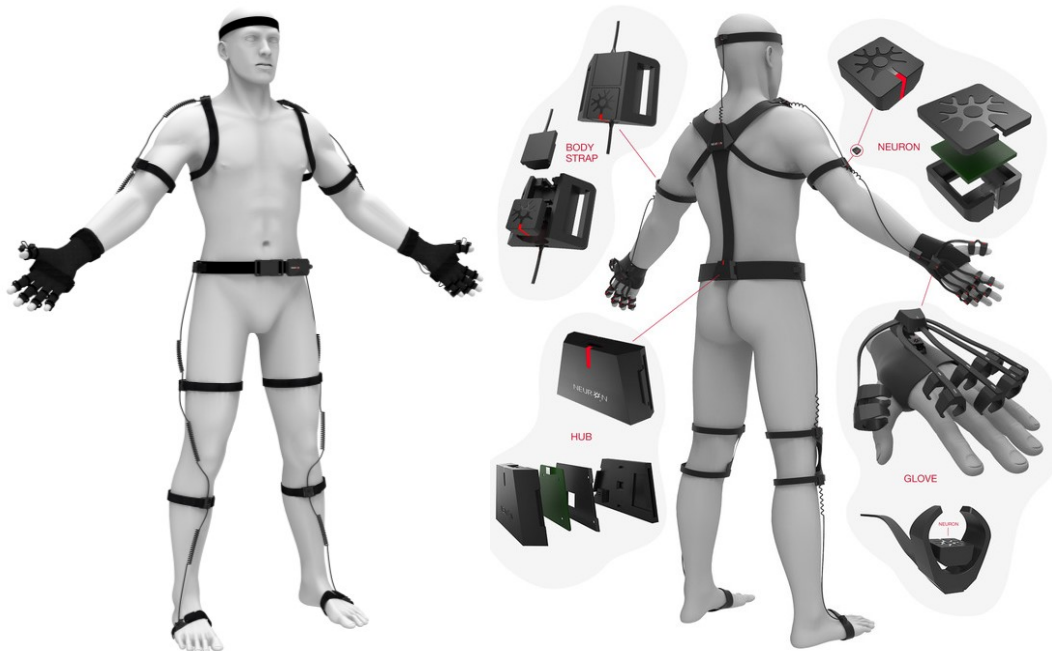


Abbildung 20. Hardware des Perception Neuron MoCap Anzug

Die Neuronen selbst weisen folgende technischen Spezifikationen auf:

- 3-Achsen Beschleunigungssensor, 3-Achsen Gyroskop Sensor und 3-Achsen Magnetometer (9-DOF)
- kalkulierte Orientierung (Quaternion)
- werkskalibriert, leichte Kalibrierung über ein Wartungs-Tool
- statische Genauigkeit: +/-1 Grad für Roll/Nick Winkel, +/-2 Grad für Gier Winkel (Eulerwinkel)
- Messbereich der Winkelgeschwindigkeit und Beschleunigung: +/-2000 °/Sek., +/-16G
- Hardware Schnittstelle: 6-pin Pogo-pin Verbindung
- Schnittstelle: UART Sendeempfängerrate Rate bei 3Mbps, TTL und CMOS konform
- Stromversorgung: 3.3 V
- Stromverbrauch: 20-50 mA bei 3.3 v abhängig von der internen Auslastung
- Gehäuse: Aluminium Legierung
- Abmessungen: 12.5mm\*13.1mm\*4.3mm
- Gewicht: 1.2 Gramm

(Noitom LTD, 2018c)



Die Verbindung mit dem Host-PC lässt sich über ein USB Kabel oder über Wi-Fi herstellen. Dazu muss das HUB vorerst über den USB Anschluss in der proprietären Software *Axis Neuron* registriert werden. Außerdem funktioniert die drahtlose Verbindung zum Host-PC nur dann, sofern ein Wireless Router an den Rechner angeschlossen ist. Das Aufnahmevolumen wird lediglich durch die Reichweite des Routers begrenzt, es ist theoretisch beliebig erweiterbar. Es können bis zu fünf Anzüge gleichzeitig an einer Session teilnehmen, wobei für den Vergleichstest nur ein Anzug zur Verfügung steht. (Noitom LTD, 2018a, S. 29–35)

Die minimalen Hardwareanforderungen für den Host-PC sind wie folgt:

- Intel i5-4590 Prozessor
- Intel HD Graphics 4000
- 8GB RAM
- 120 GB SSD oder 500 GB HDD
- 3x USB 3.0 ports und 1x USB 2.0 port
- Windows 7 SP1 64-bit

(Noitom LTD, 2018b)

Ein dementsprechender Standrechner ist mit Peripheriegeräten um rund 400 € erhältlich. Für die drahtlose Verbindung kann jeder 2.4 GHz Standard-Router in das System integriert werden. Diese sind ab rund 20 € erhältlich. Außerdem muss das HUB mit einer Powerbank versorgt werden. Zur erforderlichen Kapazität gibt es keine Angaben. Im offiziellen Benutzerforum wurde die Anfrage eines Nutzers so beantwortet, dass eine Powerbank mit 1.300 mAh, den Anzug mit 32 Neuronen, rund 3 – 4 Stunden mit Strom versorgen sollte. Diese sind um rund 30 € erhältlich. (amazon.com), (Noitom LTD, 2015)

Inklusive dem *Perception Neuron* Anzug, ergibt sich daraus eine **Gesamtinvestition von rund 1.759 € inkl. Mehrwertsteuer.**

### 4.3.2 Setup

Mit dem Anzug sind verschiedene Setups möglich. Zum leichteren Verständnis werden vier Beispiele veranschaulicht.

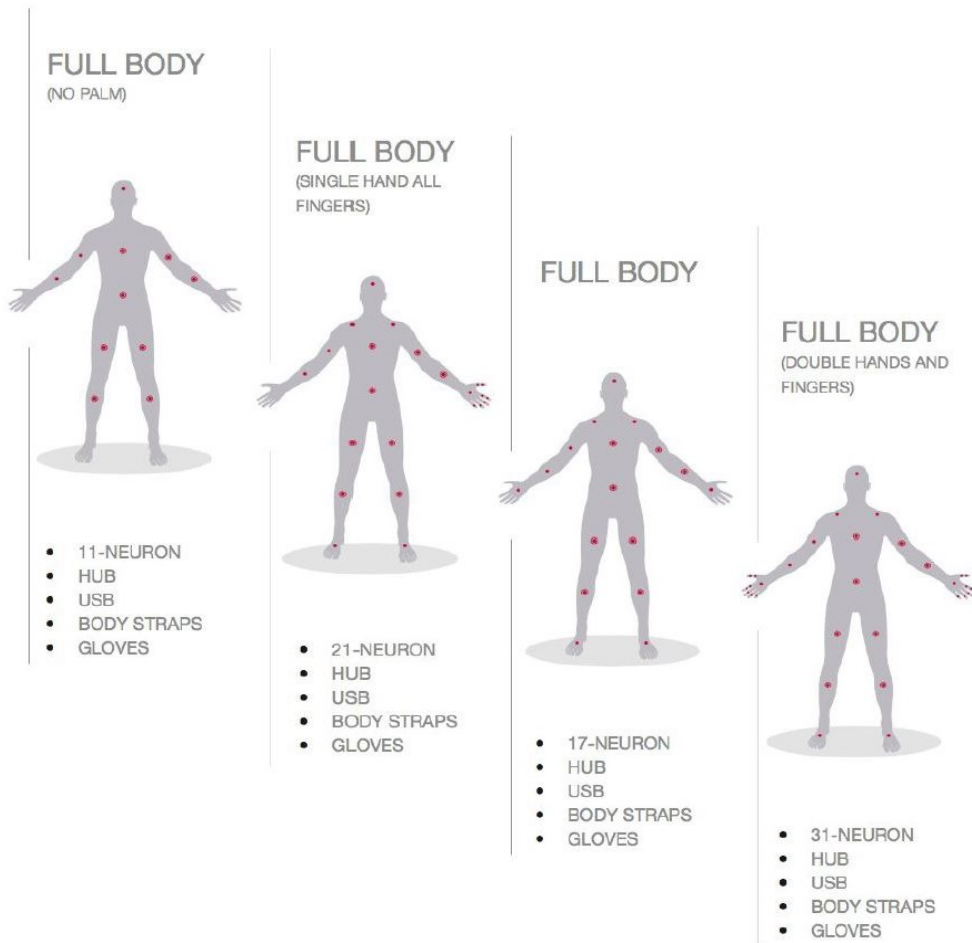


Abbildung 21. Varianten eines Ganzkörpersetups des Perception Anzugs

Es ist überaus wichtig, alle vorhandenen Konfigurationsmöglichkeiten zur Anpassung des Systems an die Anatomie des Subjekts wahrzunehmen. Die *Axis* Software verfügt hierfür über einige Presets zur Genauigkeitssteigerung des Systems. Einer der einflussreichsten Werkzeuge ist der *Body Size Manager*, welcher es erlaubt, aus einer Vielzahl an Presets von Körpergrößen zu wählen oder ein individuelles Profil zu erstellen. Um anatomische Unklarheiten zu beseitigen, stehen im offiziellen Manual exakte Illustrationen zur richtigen Vermessung der Körperpartien bereit. (Noitom LTD, 2018a, S. 64 – 69)

Wie sich herausstellt, ist es aufgrund von Haut und Muskeln zum Teil nicht möglich, eine exakte Aussage über die Endpunkte eines Knochens oder das Zentrum eines Gelenks zu treffen. Dennoch wird der Körper des Subjekts gewissenhaft vermessen und ein persönliches Profil angelegt. Dieses kann zu einem beliebigen Zeitpunkt wieder in die Software importiert und zugewiesen werden.

Wie die Gurte mit den Sensorsockeln am Körper zu befestigen sind, kann anhand einer Illustration sowie an der Rückseite der Sockel ausgelesen werden. Fehlende Maßangaben verursachen einen Interpretationsspielraum.

### 4.3.3 Kalibrierung und Wartung der Sensoren

Bei einem Ganzkörper-Setup sieht *Axis Neuron* eine Vier-Stufen-Kalibrierung anhand einer *Steady-Pose*, *A-Pose*, *T-Pose* und *S-Pose* vor, welche die gesamte Körperhaltung des Subjekts beeinflusst (siehe Abbildung 22.). (Noitom LTD, 2018a, S. 35)

Im Zuge der *Steady-Pose* müssen sich die Sensoren während der Kalibrierung im möglichst regungslosen Zustand befinden. Sie entfernt Verzerrungen an den Sensor-zu-Sockel Verbindungen. Sobald alle Sensoren initialisiert sind, muss der Vorgang kein weiteres Mal wiederholt werden. Der Anzug kann alternativ regungslos auf den Boden gelegt werden. (ebd.)

Die *A-Pose* benötigt einen aufrecht und parallel stehenden, mit den Armen nach unten und den Handflächen zu den Oberschenkeln gerichteten Körper. (ebd.)

Bei der *T-Pose* müssen zusätzlich die Arme parallel und waagrecht zur Seite hin ausgestreckt werden. Die Handflächen deuten nach unten. Bei einem kompletten Handtracking sind die Daumen 45 Grad nach unten abzuwinkeln. (ebd.)

Die *S-Pose* bildet eine leichte Hocke. Die Füße stehen parallel und die Arme werden nach vorne hin waagrecht ausgestreckt. Die Handflächen deuten nach unten.

In der *Axis Neuron* Software werden alle Posen automatisch der Reihe nach durchlaufen, sobald die *calibrate* Schaltfläche ausgewählt wird. Das Subjekt steht daraufhin zum Tracken bereit.

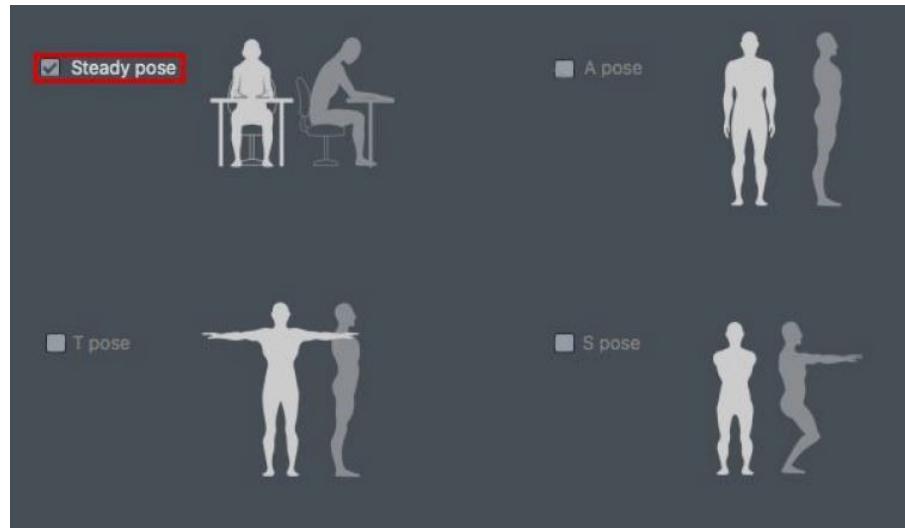


Abbildung 22. Kalibrierungssequenz des Perception Anzugs

Die Neuronen reagieren empfindlich auf künstliche Magnetfelder, welche Störungen des Systems verursachen. Dieses Problem tritt nicht ausschließlich temporär auf, sondern kann den Sensor auf Dauer magnetisieren und zur Gänze unbrauchbar machen. Ab diesem Moment beginnt der Sensor ohne erkennlichen Grund spürbar zu drift. Danach muss dieser über einen speziellen Reset Vorgang zurückgesetzt werden. Dazu wird er zuerst im kompletten Ruhezustand und daraufhin in kreisenden Bewegungen ausgelesen und kalibriert. Der Vorgang ist nicht ganz risikofrei, da ausdrückliche Warnungen auf einen möglichen Totalverlust der Sensoreinheit hinweisen. (Noitom LTD, 2018a, S. 99–102)

Damit eine Magnetisierung der Sensoren während der Lagerung von vornherein vermieden wird, empfiehlt der Hersteller die Sensoren nach Gebrauch stets aus dem Anzug zu entfernen und in einer speziellen, antimagnetischen Elektrostahlbox aufzubewahren.

Des Weiteren sollen während der Aufnahme folgende Störquellen vermieden werden:

- elektrische Geräte oder Generatoren
- jede Form eines Magneten
- Sicherungskästen oder Hochspannungsleitungen
- Lautsprecher, Kühlschränke, Kompressoren oder Klimaanlage
- Computer, -Peripherie, TV- und Computerbildschirme

(Noitom LTD, 2018a, S. 111)

### 4.3.4 Softwareaufbau und Feineinstellungen

Das GUI von Axis ist modular aufgebaut. Die Standard Ansicht (siehe Abbildung 23.) bietet ein Fenster zur Voransicht des Live-Inputs oder der gespeicherten Aufnahmen [1]. Hier kann lediglich die Perspektive verändert werden. Der Hilfscharakter ist nicht anwählbar. (Noitom LTD, 2018a, S. 17)

Am rechten Rand ist eine Sensorkarte sowie die MoCap Leiste [2]. Die Karte signalisiert die Anzahl der verwendeten Sensoren an der jeweiligen Körperposition. Dessen Ampelsystem von Grün nach Rot kennzeichnet die Signalstärke des Sensors. In der MoCap Leiste sind die Bedienelemente zur Aufnahme, Verbindung, Kalibrierung und Fehlerkorrektur des Anzugs. (ebd.)

Unter der Voransicht befindet sich die Kontaktbearbeitung [3]. Hier lassen sich bereits erste Cleanup-Schritte nach der Aufnahme in Form von Kontaktfehlerbehebungen bzw. dem *Contact Editing* durchführen. Im gegebenen Beispiel kennzeichnen die schwarzen Balken den Bodenkontakt der Füße. Gleichzeitig wird bei jedem Kontakt ein roter Ball am betroffenen Körperglied sichtbar. Die Kontaktphasen der Gliedmaßen können im Nachhinein beliebig anpasst werden. Die Funktion steht ebenfalls für die Hände und die Hüfte zur Verfügung. (ebd.)

Gleich darunter befinden sich die Werkzeuge zur Abgrenzung und Wiedergabe der ROI [4]. Im unteren linken Bereich befindet sich das Dateiverzeichnis [5] mit bereits gespeicherten Aufnahmen. Rechts daneben sind die drei wichtigsten Einstellmöglichkeiten zur Fehlerkorrektur der Körperhaltung [6]. (ebd.)

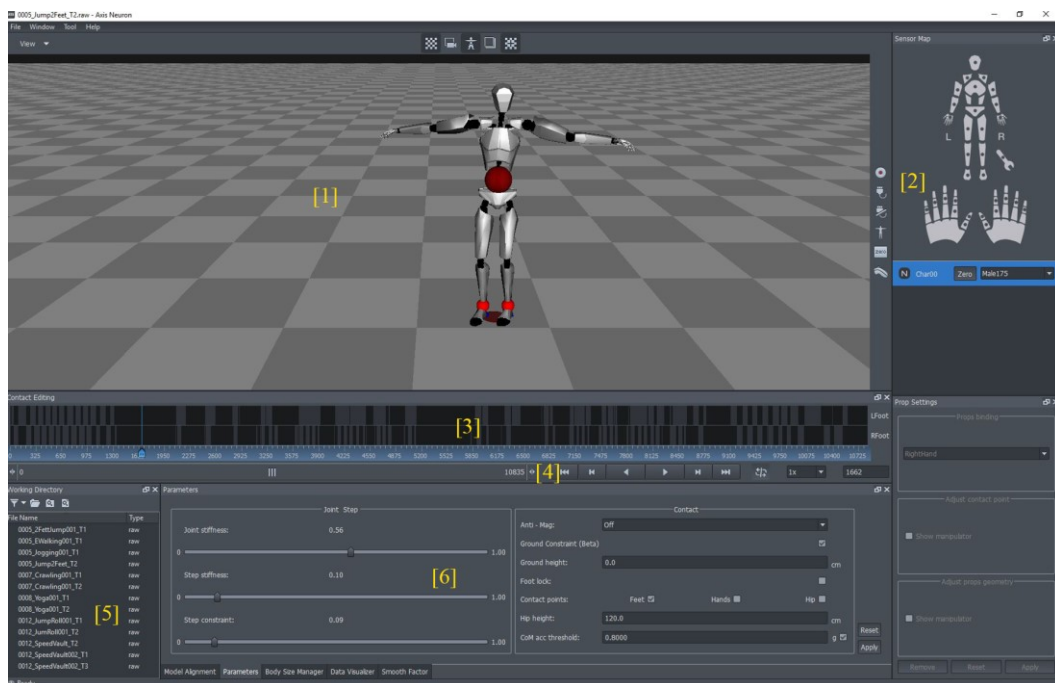


Abbildung 23. GUI der Axis Neuron Software

Die Vier-Stufen-Kalibrierung ist die Grundeinstellung zur korrekten Interpretation der Körperhaltung. Die *Axis Neuron* Software bietet zusätzliche Möglichkeiten, bestimmte Parameter gezielt an das Subjekt anzupassen, um so das MoCap zu optimieren.

Im Bereich *Model Alignment* lassen sich Abweichungen der Pose mit den Parametern *Yaw*, *Pitch* und *Roll* ausgleichen. *Yaw* richtet den Charakter in eine gewünschte Himmelsrichtung aus. *Pitch* beschreibt den inkludierten Winkel zwischen dem Sensor und dem Boden in vertikaler Richtung. Der Einstellbereich liegt innerhalb von 90 Grad und lässt den Körper vorwärts oder rückwärts neigen. *Roll* ist der inkludierte Winkel zwischen dem Sensor und dem Boden in horizontaler Richtung. Der Einstellbereich liegt innerhalb von 90 Grad und lässt den Körper nach links oder rechts hin ausgleichen. (Noitom LTD, 2018a, S. 53)

Für weitere Genauigkeitssteigerungen lässt sich der *Body Size Manager* konfigurieren. Im Detail bedeutet dass eine bessere Fußkontaktberechnung, eine korrekte Schrittweite, eine minimierte Fehlerquote bei Auftreten und ein allgemein besseres Verständnis der Software über die Bewegungsabläufe des Charakters. (Noitom LTD, 2018a, S. 64)

Die anzupassenden Bereiche umfassen Kopf, Nacken, Torso, Schulterbreite, Ober- und Unterarm, Handfläche, Hüftbreite, Oberschenkel, Schienbein, Fersenhöhe und Fußlänge. Die gesamte Vermessung lässt sich speichern, importieren, exportieren und nachträglich bearbeiten. Nachdem die Körpermaße dem Avatar zugewiesen wurden, müssen die Kalibrierphasen erneut durchlaufen werden. Für spontane MoCap Aufnahmen stehen optional diverse Presets zur Verfügung. (Noitom LTD, 2018a, S. 54–56)

Hinzu kommen die Parameter zur Feineinstellung des Bodenkontakts und der Eigenschaften der Gelenke (siehe Abbildung 24.). Je höher die Werte der *Joint Stiffness* eingestellt werden, desto schwerer lassen sich die Bewegungen der einzelnen Glieder voneinander unterscheiden. Genaue Körpermaße erlauben die Verwendung höherer Werte. Die *Step Stiffness* ist ein Algorithmus, der anhand von Mustererkennung die Dauer des Kontakts der Füße mit dem Boden während einer Gehbewegung errechnet. Höhere Werte lassen die Füße schwerer vom Boden abheben. (Noitom LTD, 2018a, S. 57)

Dabei definiert die *Step Constraint*, ab wann der Bodenkontakt vermutlich hergestellt wird. Sollten die Ergebnisse der *Step Stiffness* nicht zufriedenstellend sein, lässt sich über die *Foot Lock* Funktion die vertikale Z-Achse der Füße blockieren, damit diese leichter am Boden fixiert bleiben. Mit der Vergabe von *Contact Points* lässt sich der Bodenkontakt noch detaillierter bestimmen. Für die Füße stehen hier insgesamt zehn Kontaktpunkte am Fußballen und der Ferse zur Verfügung. (Noitom LTD, 2018a, S. 57, 58)

Hände und Hüfte lassen sich ebenfalls mit Kontaktpunkten ausstatten. Sind im Aufnahmevolumen magnetische Störfelder anwesend, kann mit der *Anit-Mag* Funktion der *Footskate* minimiert werden. Während das Subjekt Stufen betritt, sollte der Wert der *Ground Height* angepasst werden, damit das 3D-Skelett den Boden nicht unterwandert. (Noitom LTD, 2018a, S. 57, 58)

Das *Model Alignment* und weitere Feineinstellungen müssen **vor dem Contact Editing** durchgeführt werden, da ansonsten sämtliche vergebenen Werte hinfällig werden. (Noitom LTD, 2018a, S. 61)



Abbildung 24. Feineinstellung des Skeletts sowie des Bodenkontaktes

Für gewöhnlich entstehen durch Vibrationen, lockere Gurte und Messfehler eine fehlerhafte Interpretation der Beschleunigungen sowie Gravitation. Der *Smooth Factor Filter* wirkt dem entgegen. (Noitom LTD, 2018a, S. 59)

### 4.4 iPi Soft – Markerloses Motion Capture

*iPi Motion Capture* ist ein Softwarepaket der Firma *iPi Soft LLC* und versteht sich als markerloses Werkzeug zum Tracken von Bewegungen im 3D-Raum. Je nach Lizenz können bis zu 16 handelsübliche und vor allem preiswerte RGB Kameras im Webcam-Format, bis zu vier Tiefensensoren wie die der Microsoft Kinect, Bewegungs-Controller wie Playstation Move und Action Kameras wie GoPro genutzt werden. (iPi Soft LLC, 2014, S. 1)

Aufgrund des markerlosen Prinzips kann iPi Motion Capture auf spezielle MoCap Anzüge und konventionelle Aufnahmeräume verzichten. Ebenso ist das System portabel, hat kurze Aufbauzeiten (rund 30 Minuten) und ist durch nur eine Person bedienbar. Das Aufnahmevolumen erstreckt sich von mindestens 3 x 1.5 Meter auf maximal 10 x 10 Meter. (ebd.)

Die Kernkompetenzen des Softwarepakets sind wie folgt:

- Multikameraaufzeichnungen in gewöhnlichen Büro- oder Heimumgebungen
- Aufzeichnung von bis zu 3 Charakteren gleichzeitig direkt auf dem Computer mit Grafikkarten auf Gaming-Niveau
- Interne Keyframe-Werkzeuge und Verbesserungsalgorithmen
- Anfertigung einer Skelett-Animation für virtuelle 3D-Charaktere
- Inkludiert die gängigsten Dateiformate wie .fbx, .bvh, .dmx, .smd und Collada
- Kompatibel mit den meisten gängigen 3D-Anwendungsprogrammen wie *Maya*, *Cinema 4D*, *Blender*, *Unreal Engine* usw.
- Detaillierte biomechanische Analysen des menschlichen Körpers mit der **iPi Biomech** Erweiterung
- Integration der Software in größere Programmstrukturen und komplexe automatisierte Prozesse mit der **iPi Automation** Erweiterung
- Echtzeit-Preview mit einer RGB-D Kamera möglich

(ebd.)

iPi Motion Capture ist in drei verschiedenen Varianten erhältlich, wobei in der für die Arbeit relevanten 30-Tage Testversion die Funktionen des **Basic** Abo enthalten sind. Damit lassen sich bis zu zwei Tiefensensoren oder bis zu sechs RGB Kameras zur Aufzeichnung verwenden und somit alle Anforderungen an ein präzises Motion Capturing erfüllen. Der gegenwärtige Preis für eine perpetuale Lizenz liegt bei **804,31 € inklusive MwSt. und ohne Hardware**. ([www.ipisoft.com](http://www.ipisoft.com))



### 4.4.1 Hardware

Zur Aufnahme mit RGB Kameras wird die *Playstation Eye* Kamera der *Sony PlayStation 3* Konsole empfohlen. Im System integriert schafft sie bis zu 60 Bilder in der Sekunde und hat gleichzeitig eine hervorragende Auto-Belichtung und Auto-Verstärkung bei einer ausreichend guten Restlichtempfindlichkeit. Die Linsenverzerrung ist mit  $< 1\%$  niedrig und der diagonale Betrachtungswinkel von  $75^\circ$  für MoCap optimal. Der Stückpreis liegt bei rund 10 Euro (Stand 19.06.2018, amazon.com). (iPi Soft LLC, 2017, S. 2, 3)

Es wird empfohlen, stets die maximale Bildrate auszunutzen, da so Bewegungsunschärfe vermieden und feine Details besser erkannt werden. Theoretisch sind die *Playstation Eye* Kameras zu 120 fps in der Lage, allerdings würde der Kamerasensor zu hohes Bildrauschen verursachen und die MoCap Aufnahme unbrauchbar machen. Ebenso sind Bildraten unter 30 fps nicht empfehlenswert. (iPi Soft LLC, 2017, S. 13)

Die minimalen Anforderungen an den Host-PC sind ein x86/x64 Intel Pentium 4 Prozessor, bevorzugt in einer Dualcore oder Quadcore Ausführung, sowie eine DirectX 11 fähige Grafikkarte auf Gaming-Niveau. Im Vergleichstest werden sechs Kameras mit einer Auflösung von  $640 \times 480$  Pixel und einer Bildrate von 60 fps benutzt. Dafür ist ein Speichermedium mit einer Übertragungsrate von nicht weniger als 25 Mbyte/s erforderlich. (iPi Soft LLC, 2015b, S. 3, 4)

Um ein Setup aus sechs PS Eye Kameras im Format 480p60 zu betreiben, sollte die CPU zumindest  $4 \times 2$  GHz Leistung aufweisen. Die Daten sollen auf einer speziell zugeteilten, vom Betriebssystem getrennten Festplatte mit ausreichender Schreib- und Lesegeschwindigkeit gespeichert werden. Die Schnittstelle zwischen Kamera und Host-PC stellen sogenannte USB Controller her. Beim erwähnten Setup aus sechs Kameras können über einen USB 2.0 Controller jeweils zwei und über manche USB 3.0 Controller vier Kamerasignale empfangen werden. Ein zeitgemäßer Desktop Computer verfügt über mindestens zwei USB 2.0 Controller.

Ab rund drei Meter Kabellänge nimmt die Signaldämpfung deutlich zu, weshalb für längere Strecken eine aktive Verstärkung (Repeater) integriert werden muss. Unzureichende Systemressourcen führen zu Problemen wie *Drop-Frames*, einer instabilen Bildwiederholrate oder zu vorzeitigen Aufnahmestops. (iPi Soft LLC, 2015b, S. 13, 14)

Ein moderner Einsteiger-Desktop PC kann die minimalen Systemanforderungen mühelos erfüllen und ist inklusive Peripheriegeräte (Bildschirm, Maus, Tastatur) um rund 350 € erhältlich. (amazon.com)

Eine aktiv verstärktes USB 2.0 Verlängerungskabel ist um rund 14 € pro Stück erhältlich. (amazon.com)

Die im Test verwendeten Dreibeinstative sind mit einem Stückpreis von rund 15 €, im Vergleich zu produktionsüblichen Licht- oder Videostativen, preiswert. Mit 700 Gramm Gesamtgewicht sind sie hingegen äußerst leicht und damit anfällig gegenüber Erschütterungen jeglicher Art. Zugunsten einer erhöhten Stabilität sollten Videostative ab 1,5 kg in Betracht gezogen werden. Diese sind ab rund 27 € erhältlich. (amazon.com)

Mit der in dieser Arbeit getesteten Basislizenz und einem Setup aus sechs Playstation Eye Kameras, ergeben sich inklusive zusätzlicher Hardware ein **Gesamteinkaufspreis von rund 1.465 € inkl. MwSt.**

### 4.4.2 Setup

Besteht das System nur aus optischen RGB-Kameras, erfasst *iPi Motion Capture* unter anderem die Kleidungsfarbe des Darstellers und differenziert diese vom Hintergrund. Darum sollte das Subjekt satte, dunkle Farben tragen, welche sich eindeutig vom Hintergrund abheben. Dazu bietet sich ein schwarzes, dunkelgrünes oder -blaues Langarmshirt oder T-Shirt in Kombination mit einer blauen Jeanshose und schwarzen Schuhen an. Rote Kleidung ist aufgrund der Ähnlichkeit zum menschlichen Hautton ungeeignet. (iPi Soft LLC, 2015, S. 11, 12, 13)

Für markerloses MoCap sind helle Hintergründe ideal. Die Software ist so konzipiert, dass gleichmäßig ausgeleuchtete Wände in Alltagsumgebungen ausreichend sind, um das Tracking erfolgreich durchzuführen. Hat die Kleidungsfarbe des Subjekts einen niedrigen Kontrast zu Hintergrundobjekten, kann die Software zu Fehlinterpretationen neigen. Abhilfe schafft ein einheitlich hellgrüner oder hellblauer Hintergrundvorhang. Die gleichmäßige Deckenbeleuchtung einer Büroumgebung reicht im Regelfall zur Aufnahme aus. Bei Heimanwendungen sind unter Umständen zusätzliche Lichtquellen nötig. Außenaufnahmen sind aufgrund der Anfälligkeit des Systems gegenüber Sonnenlicht nur an bedeckten Tagen möglich. (iPi Soft LLC, 2015a, S. 5)

Für die Aufnahme mit mehreren RGB Kameras ist eine minimale Aufbaufläche von 4 x 4 Metern vorgesehen, da das Subjekt ansonsten nicht in das Bild passt. Bei einer Auflösung von 640 x 480 Pixel können bis zu 7 x 7 Meter Aufnahmevolumen genutzt werden, was für Bewegungen wie Laufen, Tanzen usw. ausreichend sein sollte. (iPi Soft LLC, 2015a, S. 4)

Der Szenenaufbau fällt je nach Kamerazahl unterschiedlich aus. Er beginnt grundsätzlich halbkreisförmig. Ab vier Kameras kann zur Verbesserung der Genauigkeit eine der Kameras höher (ca. 3m) platziert werden. Ab fünf Kameras wird je nach Platzangebot eine Anordnung zwischen einem Halb- oder Vollkreis gewählt. Zusätzlich kann noch eine zweite Kamera nach oben hin platziert werden.

*Playstation Eye* Kameras besitzen kein Stativgewinde. Dieses Problem wird beispielsweise mit einem professionellen, einseitigen oder doppelseitigen Klebeband umgangen. (iPi Soft LLC, 2015b, S. 17, 18)

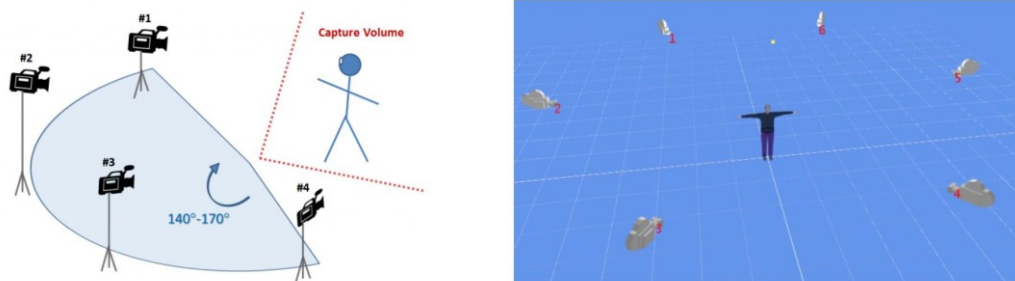


Abbildung 25. Beispiel eines halb- und vollkreisförmigen Aufbaus

### 4.4.3 Kalibrierung

Für eine korrekte Datenberechnung ist eine globale Kalibrierung des Systems notwendig. Dieser Schritt ist kritisch und sollte nach jeder kleinsten Änderung der Kameraposition, vorzugsweise vor und nach jeder Aufnahme, erneut durchgeführt werden. (iPi Soft LLC, 2015a)

Zuallererst ist darauf zu achten, dass die Linse der *Playstation Eye* Kamera sich im Weitwinkelmodus (Drehung der Linse zum blauen Punkt) befindet. Daraufhin muss der Reflektorkopf einer Taschenlampe abgeschraubt werden, sodass nurmehr die Glühbirne vorsteht. Alternativ ist die leuchtende Kugel eines PlayStation Move Controller ebenfalls zulässig. (ebd.)

Danach wird die *iPi Recorder* Software gestartet und der *Darkening* Modus aktiviert, um ausschließlich die hellsten Bereiche der Aufnahme eindeutig sichtbar zu machen (optimalerweise die Taschenlampe). Nachdem das Aufnahmevervolumen zuvor ermittelt und abgegrenzt wurde, kann die Aufnahme gestartet und die Außengrenzen der Reihe nach mit der Lichtquelle von Oben nach Unten abgewandert werden. Zum Schluss wird eine spiralförmige Bewegung von Oben nach Unten zum Ursprungsmittelpunkt hin aufgenommen. Sobald die Aufnahme stoppt, wird das „Kalibrierungsvideo“ zur späteren Verwendung abgespeichert. (ebd.)



Abbildung 26. Auswertung des Kalibrierungsvideos

### 4.4.4 Softwareaufbau und Feineinstellung

Das Softwarepaket teilt sich in die *iPi Recorder*- und die *iPi Mocap Studio* Software. Erstere ist für die Aufnahme der Videosstreams zuständig, welche anschließend zu einer Multikameradatei zusammengefügt und in *Mocap Studio* zur Auswertung importiert werden.

Das Interface von *iPi Mocap Studio* (siehe Abbildung 27.) ist fest aufgebaut. Es gibt keine unterschiedlichen Darstellungspresets. Die verfügbaren Elemente sind jedoch übersichtlich strukturiert und weitgehend nachvollziehbar.

Die *Standard Tool Bar* [1] enthält neben den Projekteinstellungen alle Werkzeuge zur Interaktion mit den verschiedenen Kameraaufnahmen sowie dem *Actor* bzw. dessen Skelett. Der Arbeitsbereich [2] ist interaktiv.

Unterhalb des Arbeitsbereichs befinden sich die *Timeline Controls* [3], mit welchen sich z.B. die *Region of Interest (ROI)* eingrenzen oder die Aufnahme in mehrere Takes aufteilen lässt. Außerdem können manuelle Keyframes für individuelle Animationen der Hände vergeben werden.

Die wichtigsten Parametereinstellungen [4] finden sich rechts in funktionspezifischen Tabs. Im *Scene Tab* sind alle globalen Einstellungen und Daten über die Ausrichtung des Koordinatensystems, der Kameras, der Grundebene und der Lichtquelle zu finden.

Die Einstellungen des *Actor Tab* beeinflussen wesentlich die Objekterkennung und damit die Qualität des Trackings. Der *Actor* ist ein Hilfscharakter, ähnlich dem in *MotionBuilder* für das Solven der Markerdaten. Durch die Überblendung des *Actor*

mit den Originalaufnahmen, lässt sich dieser exakt an sein reales Vorbild anpassen. Die Farben des Subjekts in der Originalaufnahme können anschließend über Farbpipetten auf den *Actor* übertragen werden. Eine *T-Pose* ist für ein effizientes Ausrichten des *Actors* vorteilhaft. Sobald der *Actor* weitestgehend mit dem Subjekt übereinstimmt, ist das Tracking bereit. Der Vorgang wird im Punkt 4.5.3.1 genauer erläutert.

Wie bei jedem MoCap System, entstehen hier mehr oder weniger häufige Fehler, welche an einzelnen bestimmten Frames entstehen und sich auf mehrere nachfolgende Frames ausbreiten. Die Software bietet dazu interne Werkzeuge zur manuellen Korrektur dieser Keyframes sowie Algorithmen zur Glättung von Ungenauigkeiten wie *Jitter*.

The screenshot displays the Microsoft Kinect v2 software interface, which is used for motion capture and tracking. The main window shows a 3D virtual environment with a person standing in the center. The person's pose is tracked and visualized with a green skeleton overlay. The interface is divided into several panels:

- Top Panel:** Contains the main 3D view and a toolbar with various icons for navigation and settings.
- Left Panel:** Displays camera settings for 'Camera #1', including height over ground, position (XYZ), target (XYZ), azimuth/pan, altitude/tilt, roll, pixel aspect ratio, and field of view. It also includes synchronization offset, reprojection error, and marker misdetections.
- Right Panel:** Contains actor settings, including gender, height, body dimensions, and appearance. It also includes a 'Track and Refine' section with options for tracking, refining, and post-processing.
- Bottom Panel:** Displays a list of body parts and their corresponding tracking status, including Torso, Head, Left Arm, Right Arm, Left Leg, and Right Leg.

Red annotations are present in the image:

- [1] points to the 'Save Scene' button in the top right corner.
- [2] points to the 'Refit Pose' button in the 'Track and Refine' panel.
- [3] points to the 'Refit Pose' button in the 'Track and Refine' panel.

86

# 4.5 Durchführung

## 4.5.1 Kriterien

Ein 1:1 Vergleich der Daten ist nur zwischen den zwei preiswerten Systemen möglich, da ein High-End Motion Capturing System aus Kostengründen nicht zur Verfügung steht.

Stellvertretend werden Aufnahmen der SFU Motion Capture Datenbank gewählt, da die Markerdaten bereits in das BVH Format konvertiert wurden und für jede vorhandene Aufnahme eine interaktive 3D-Voransicht anhand eines 3D-Animationsskeletts zur Verfügung steht. Dadurch lassen sich geeignete Sequenzen auswählen, einstudieren und daraufhin eine Shotlist für den praktischen Vergleich erstellen.

Für kontrollierte Bedingungen sind ein ausreichend großes Areal, mit gleichmäßig ausgeleuchtetem Hintergrund sowie möglichst wenig magnetischer Störstrahlung erforderlich. Die individuellen Beleuchtungsmöglichkeiten und Größe des Videostudios der Fachhochschule St. Pölten erfüllen diese Anforderungen. Trotz der umfangreichen technischen Infrastruktur des Studios, konnte die Auswertung einer elektronischen Magnetfeldmessung keine für den *Perception Neuron* Anzug problematischen Störfelder feststellen.

Die Praktische Analyse findet in drei Etappen statt. Zuerst werden alle Auffälligkeiten während der Aufnahme sowie die allgemeine Handhabung der beiden Systeme reflektiert.

Die zweite Phase beinhaltet den Vergleich der reinen Rohdaten bzw. Animationen, ohne jegliches Post Processing oder sonstige Veränderungen. Damit soll festgestellt werden, wie präzise die beiden Systeme das Tracking grundsätzlich durchführen. Die Skelette werden dazu innerhalb einer FBX Datei aus *Axis* und *Mocap Studio* exportiert und in *Autodesk MotionBuilder* importiert. Die Rohanimationen der beiden Testsysteme werden anhand eines digitalen Skeletts visualisiert und in einer Videodatei gegenübergestellt. (siehe Abbildung 28.) Die Datei kann auf dem beiliegenden Datenträger begutachtet werden.

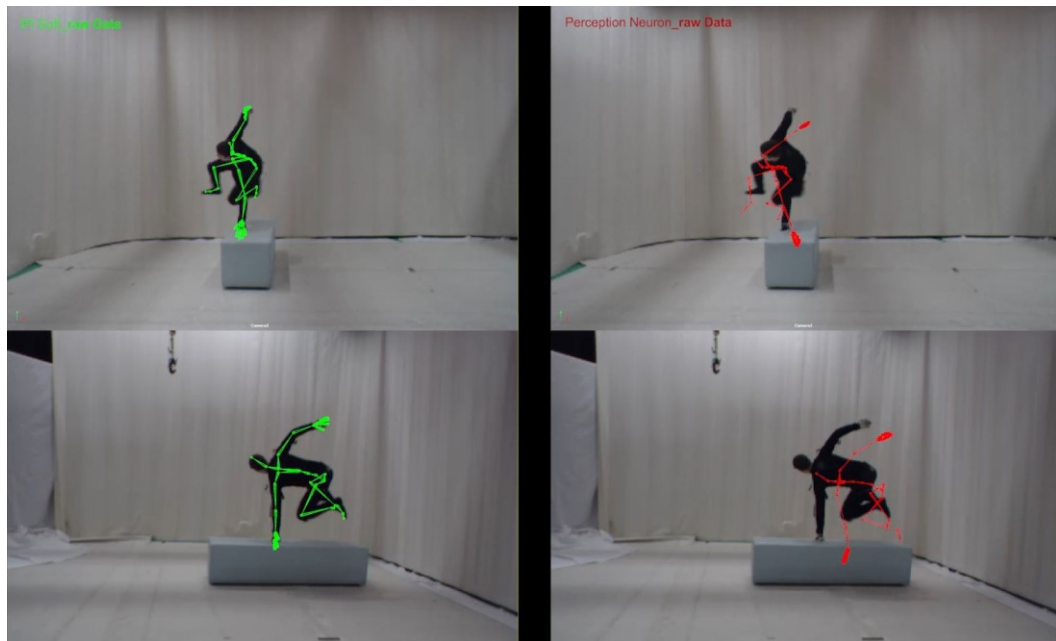


Abbildung 28. Screenshot einer Vergleichsaufnahme

In der letzten Phase wird die Szene *0007\_Crawling001\_T1* nach Möglichkeit soweit aufbereitet, dass sie der optischen Qualität der *Vicon* Referenzaufnahme entspricht. Die Sequenz eignet sich als Bewertungsgrundlage, da sie einen komplexen Bewegungsablauf (Bodenkontakt, Verwinden des gesamten Körpers, ruckartige Richtungswechsel usw.) enthält.

Es werden sämtliche Cleanup- und Post Processing Werkzeuge geprüft, die in den proprietären Programmen zur Verfügung stehen. Der entstandene Nachbearbeitungsaufwand liefert weitere Erkenntnisse über die Gesamtpformance der Systeme. Die Videodateien der Roh- und Endanimation, übertragen auf einen 3D-Charakter, stehen ebenfalls am beiliegenden Datenträger zur Beurteilung bereit.

### 4.5.2 Szeneninhalt

Für die Testaufnahmen müssen Szenarien gefunden werden, welche die gängigsten Bewegungsabläufe sowie typische Fehlerquellen einer MoCap Session abdecken und die Grenzen der Günstig-Lösungen ermitteln.

Sie sollen demnach Gehen, Springen, Laufen und ebenfalls problematische Szenarien wie Verdeckungen, Ganzkörperkontakt mit dem Boden, Interaktion mit Gegenständen und Treppensteigen bzw. Abheben von der Grundebene meistern.

Die Aufnahmen beinhalten einzelne Szenen/Sequenzen mit ansteigender Komplexität. Jede Szene besteht aus zwei bis drei Takes, um auf ausreichend Reserven aufgrund unbemerkter Aufnahmefehler zurückgreifen zu können. Die Takes sind aus praktischen Gründen in einer durchgehenden Aufnahme



untergebracht. Visuelle Handzeichen signalisieren den aktuellen Take. Dadurch muss später nicht unzählige Male die *T-Pose* erneut, sondern einmalig pro Szene zugewiesen werden. Außerdem lassen sich unnötige Einzeldateien vermeiden.

Die *T-Pose* wird oft zu Beginn und am Ende einer Aufnahme eingebracht und ermöglicht bei markerbasierten Systemen einen übersichtlichen Blick auf alle Marker, sodass sich diese später leichter identifizieren lassen. Für ein *Retargeting* erwartet die 3D-Software den Körper ebenfalls in dieser Stellung. Die *T-Pose* wird anschließend verworfen und nicht im finalen Produkt zu sehen sein. (Kitagawa & Windsor, 2008, S. 48)

Bei Großprojekten ist auf eine detaillierte Dateibenennung zu achten, da nicht jedes MoCap System ein Video-Overlay zulässt und eine Differenzierung der Aufnahmen schwer bis unmöglich wird. Darum sollten zumindest Datum, Szene, Take und besser noch Schlagwörter im Namen vorhanden sein.

Für die Aufnahme bekommt jedes System einen eigenen Host-PC. Da *Axis* die Bewegungen in Echtzeit analysiert und speichert, sind diese sofort nach der Aufnahme abrufbereit. Damit hat *Axis* einen wesentlichen Geschwindigkeitsvorteil gegenüber dem *iPi Soft* System, welches die Multikameraaufnahme erst analysieren und auswerten muss. Je nach verbauter GPU und Szenenlänge, können dabei einige Minuten verstreichen. Die GPU (Nvidia GT 430) des *iPi Soft* Host-PCs ist mit 700 MHz Grafiktakt für eine zeitnahe Analyse der Multikameraaufnahme zu leistungsschwach. (iPi Soft LLC, 2018)

Durch ihre jeweils simple Bedienung, können beide Testsysteme durch nur eine Person gesteuert und simultan betrieben werden. Dazu wird in *iPi Recorder* ein Timer zum Starten der Aufnahme angelegt. Während dieser runterzählt, lässt sich *Axis* zur Aufnahme vorbereiten. Danach positioniert sich das Subjekt mittig in einer *T-Pose* im Aufnahmevolumen und hält diese für 2 – 3 Sekunden. Anschließend wird mit der Choreografie fortgefahren.

Begonnen wird mit einer simplen Gangsequenz. Theoretisch sollte diese Bewegung für beide Systeme keine Herausforderung darstellen und eine Grundlage für erste Schlussfolgerungen bilden. Die nächste Stufe ist eine Jogging Bewegung. Hier könnten Schwierigkeiten für die *Playstation Eye* Kameras entstehen, da sie mit maximal 60 fps aufzeichnen. (iPi Soft LLC, 2017, S. 2)

Bei der darauffolgenden Sequenz werden kurze und lange Vor-, Rückwärts- und Seitwärtssprünge getestet.

Eine bedeutende Schwäche der *iPi Software* ist die Differenzierung des Subjekts vom eigenen Schatten, vor allem bei bodennahen Aktionen wie Liegestütze. Die dadurch fehlerhaften Frames können teilweise irreparabel sein. (iPi Soft LLC, 2015a, S. 19)

Darum wird geprüft, ob sich dieses Problem mit einem optimal ausgeleuchteten Hinter- bzw. Untergrund beheben lässt. Als Handlung wird eine Yoga- und Kriechsequenz mit langsamen und bodennahen Bewegungen ausgewählt.

Im nächsten Versuch sollen Interaktionen mit Gegenständen sowie schnelle Bewegungen getestet werden. Dazu wird eine Langbank mittig in der Aufnahmefläche positioniert, welche nach einer kurzen Anlaufphase, mit einer Hand gestützt, übersprungen wird, um danach auf der gegenüberliegenden Seite abrupt zum Stillstand zu kommen.

Die letzte Aufnahme beinhaltet eine gespielte Szene, in der das Subjekt nach einem Schuss in die Brust zu Boden fällt und auf allen Vieren kriechend den Arm in Richtung Hilfe streckt. Diese Sequenz enthält ausgeprägte Verdeckungen und -Verdrehungen, häufige Bodenkontakte und abrupte Richtungswechsel.

Da sich die komplette Szene inklusive animierten 3D-Skelett und virtueller Kameras gemeinsam in eine FBX Datei aus *iPi Soft* exportieren lassen, sollen mögliche, unkorrigierbare Fehler mit den Werkzeugen in *MotionBuilder* bearbeitet werden.

Dazu wird in zwei der sechs virtuellen Kameras ein Video-Overlay geladen. Auf diese Weise lassen sich grobe Trackingfehler unkompliziert identifizieren. (Abbildung 29.).

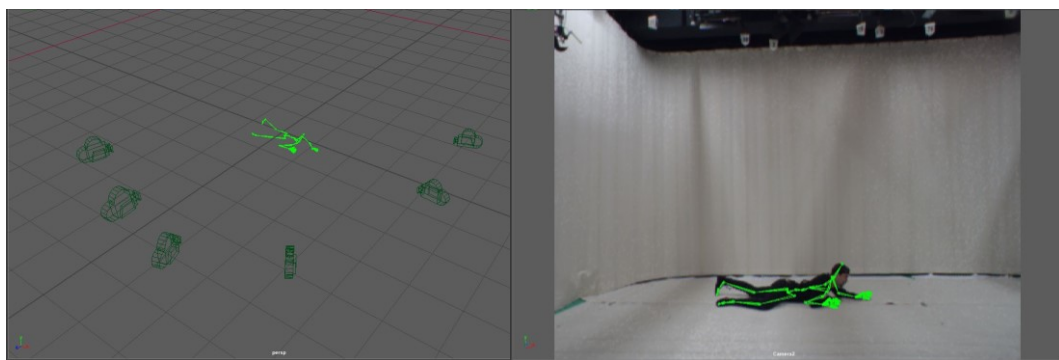


Abbildung 29. Perspektiven- (li.) und Kameraansicht der 0007\_Crawling Szene (re.)

### 4.5.3 iPi Soft Setup

Die Choreografien beinhalten ausgedehnte Bewegungen wie Joggen, Sprünge und Walkcycles, weshalb die Fläche ausreichend groß geplant werden muss. Es gilt zu beachten, dass sich die Aufnahmefläche aus der Schnittmenge der Perspektiven aller Kameras ergibt, weshalb die erforderliche Produktionsfläche inklusive Kameras bei ausgedehnten Szenen enorm anwachsen kann. Im beschriebenen Test sind die Kameras halbkreisförmig zum Subjekt mit einem Kreisdurchmesser von rund 8 Meter aufgestellt, wodurch sich eine effektive Aufnahmefläche von rund 3.7 x 2.5 Meter ergibt (rechteckiger Grundschnitt des Studios).

Nach einer groben Ausrichtung der Kameras mithilfe eines Stand-In, werden diese von links nach rechts (hier #1 - #6) durchnummeriert. Bei einer halbkreisförmigen Anordnung limitieren hauptsächlich die Perspektiven der äußeren Kameras #1 und #6 den Aktionsbereich. Um die exakten Außengrenzen zu ermitteln, wird die *iPi Recorder* Software gestartet und danach das Live-Bild aller Kameras über den Bildschirm des Host-PC geladen.

Zuerst werden Kamera #1 und #6 möglichst weit zur Mitte des Areals hin ausgerichtet. Danach bewegt sich das Subjekt im Aktionsbereich und beobachtet im Live-Bild der Kameras #1 und #6, wie weit es sich bewegen kann, ohne dass es aus einer der beiden Perspektiven hinausreicht.

Durch Feinjustieren der Kameras wird der größtmögliche Schnitt der beiden Perspektiven festgelegt. Die Außengrenzen werden zur besseren Sichtbarkeit mit Klebeband oder einem Edding am Boden markiert. Die restlichen Kameras #2 bis #5 müssen danach möglichst zur Mitte der definierten Aufnahmefläche hin ausgerichtet werden. Die optimale Positionierung ist abgeschlossen, sobald das Subjekt aus keiner der sechs Perspektiven hinausreicht.

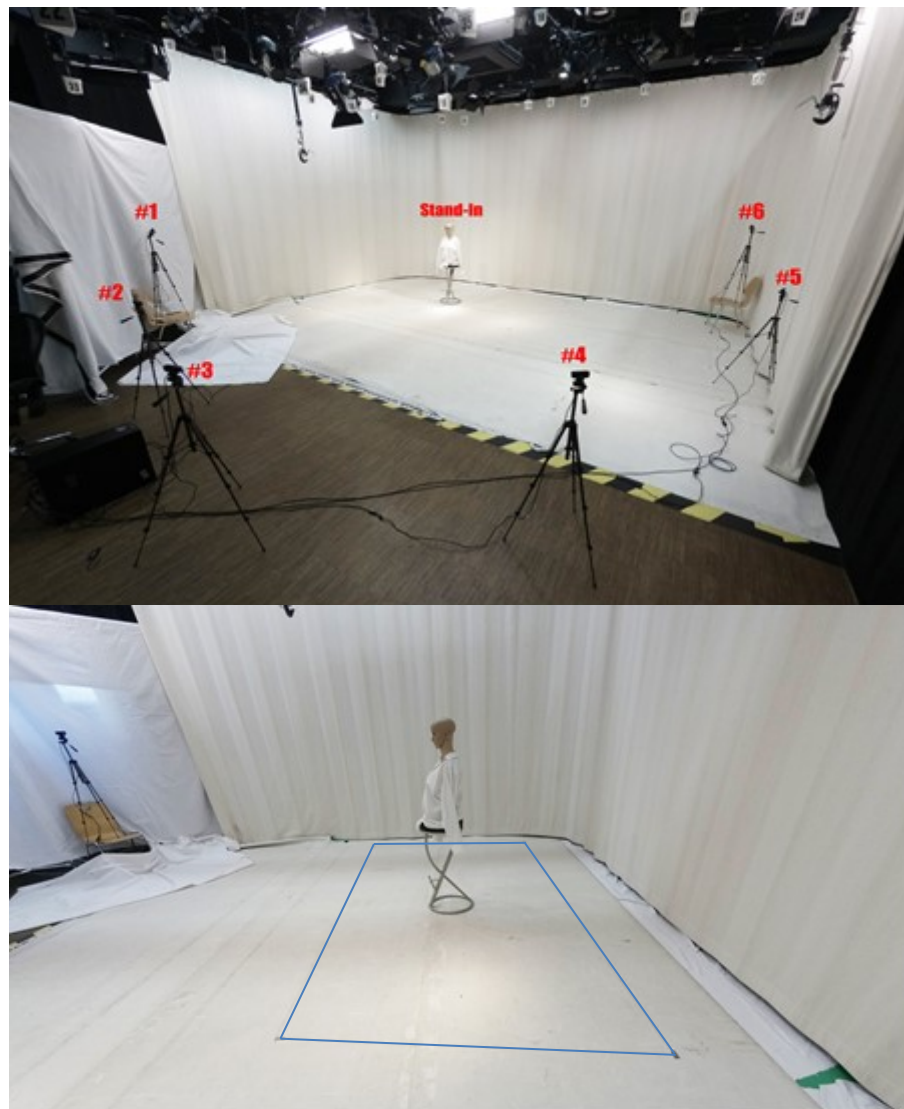


Abbildung 30. Ausrichten der Kameras und Einstellen der Beleuchtung

Der Hintergrund wird durchgehend weiß gewählt, damit sich Fehlinterpretationen so weit als möglich vermeiden lassen. Zusätzlich wird dieser mit den vorhandenen Studioscheinwerfern gleichmäßig aufgeleuchtet. Hier macht sich der schwache Dynamikumfang der *Playstation Eye* Kameras bemerkbar, da sie bei zu hoher Lichtintensität zum Übersteuern neigen. Die Software kann bei abgedunkelten Lichtverhältnissen jedoch ausreichend Informationen auslesen.

Für die optimale Kleidung wird exakt auf die Empfehlungen des *iPi Soft Manual* geachtet, d.h. das Subjekt trägt ein langärmeliges, schwarzes Shirt, eine blaue Jeans und schwarze Socken. (iPi Soft LLC, 2015a, S. 6)

An den Händen wird auf weiße Handschuhe ausgewichen, da die schwarzen Handschuhe des *Perception Neuron* Anzugs möglicherweise keine ausreichende Abgrenzung zu den Armen zulassen und trotzdem eine Möglichkeit zur Befestigung des Sensorsockels vorhanden sein muss. Weiße Handschuhe sind aufgrund des schlechten Kontrasts zum Hintergrund nicht optimal. Der Umstand wird jedoch in Kauf genommen.

### 4.5.3.1 Tracking

Das Tracking erfolgt nicht in Echtzeit, sondern muss über einen extra Zwischenschritt durch die Berechnung der Multikameraaufnahme umgesetzt werden. Nachdem alle Szenen mit der *iPi Recorder* Software aufgenommen wurden, lassen sich diese in die Software *iPi Mocap Studio* importieren. Zuvor muss das Kalibrierungsvideo geladen und interpretiert werden. Dieser Prozess wird über die Grafikkarte des Computers vollzogen, deren Leistung folglich die Berechnungsdauer beeinflusst. (iPi Soft LLC, 2015a, S. 16, 17)

Dazu wird ein zusätzlicher PC mit einer Hochleistungs-Grafikkarte *GeForce GTX 980 Ti* und einem GPU Takt von rund 1300 MHz verwendet. Damit ließ sich eine 1 ½ Minuten lange Multikameraaufnahme in rund 45 Sekunden berechnen. Zum Vergleich wurde eine wesentlich schwächere *GeForce GTX 430* des Host-PC getestet, welche für dieselbe Aufnahme rund 10 Minuten in Anspruch nahm. Um einen effizienten Workflow sicherzustellen, sollte demnach auf ausreichend Grafikleistung geachtet werden.

Der nächste Schritt ist die Vergabe der Grundebene, die den virtuellen Boden der Szene bestimmt. Sofern optimale Bedingungen (Licht, Kontrast, Sichtbarkeit zum Kalibrierungsmarker, ...) herrschen, sollte die Software die Grundebene sowie die Position der Kameras automatisch erkennen. Dafür werden die ermittelten Groundpoints und die errechnete Höhe der Kameras kontrolliert. Sollte die Höhe deutlich vom zuvor vermessenen Wert abweichen, deutet dies auf eine Fehlinterpretation der Groundpoints hin. In Folge müssen diese manuell definiert werden. (iPi Soft LLC, 2015a, S. 12, 13, 14)

Um die korrekte Szenengröße zu ermitteln, wird im letzten Schritt die zuvor gemessene Kamerahöhe einer der Kameras in das zugehörige Feld eingetragen. Daraus werden automatisch die Höhenmaße der restlichen Kameras berechnet. In eigenen Test stimmen die Werte, bis auf eine Abweichung von +/- 2 cm, mit den eigenen Messungen überein.

Ist die virtuelle Szene definiert, wird die eigentliche Aufnahme importiert. Nach Bekanntgabe der passenden Körpergröße, kann mit dem Zuweisen der *T-Pose* begonnen werden. Dazu wird eine passende Perspektive angewählt und der *Actor* über die Silhouette des Subjekts im Video platziert. Feine Unterschiede können danach mit den vorhandenen Werkzeugen *Move*, *Rotate*, *IK* sowie den

Parametern im *Actor Tab* angepasst werden. Zuletzt wird jede Kameraperspektive kontrolliert und etwaige Abweichungen korrigiert. Das Subjekt steht daraufhin zum Tracken bereit.

Die durchschnittliche Berechnungsrate liegt bei 3.2 Frames/Sekunde, wodurch sich eine gesamte Berechnungsdauer von rund 15 Minuten ergibt. Ebenfalls anzumerken ist der hohe Speicherbedarf einer Multikameradatei. Ein 2 ½ Minuten langes Video nimmt 1.32 GB Festplattenspeicher in Anspruch.

### 4.5.3.2 Cleanup

Nach dem Tracken steht die Rohanimation eines vordefinierten 3D-Skeletts zur Verfügung, welches sich je nach Komplexität der Choreografie mehr oder weniger gut mit der Originalaufnahme deckt. Die noch enthaltenen Trackingfehler (Lücken, *Jitter*, ...) werden im nächsten Schritt direkt mit den vorhandenen Werkzeugen in *iPi Mocap Studio* bearbeitet.

Zunächst müssen die Trackinglücken identifiziert werden. Sie entstehen durch Interpretationsfehler in spezifischen Frames, welche sich über mehrere Frames hinweg fortsetzen können. Fehlerhafte Bereiche lassen sich theoretisch nach manuellen Positionskorrekturen der betroffenen Gliedmaßen erneut Tracken. In schweren Fällen kann selbst *Mocap Studio* die Fehler nicht mehr korrigieren.

Bei lückenlosen Aufnahmen bietet *Mocap Studio* die Möglichkeit, basierend auf den bisherigen Trackingdaten, die Frames mit einem feineren Algorithmus erneut zu evaluieren. Damit sollen genauere Resultate erzielt werden. Der Prozess benötigt mindestens dieselbe Berechnungsdauer des ersten Durchgangs.

Vor dem Export werden die Daten mit einem variablen Smooth- sowie Peak Filter geglättet.

### 4.5.4 Perception Neuron Setup

Damit sich die Tests ohne störende Kabelverbindungen durchführen lassen, muss zuallererst ein WLAN Netzwerk erstellt werden. Dies setzt einen Drahtlosrouter voraus, welcher über ein LAN-Kabel mit dem Rechner verbunden wird. Nur so kann sich das HUB des Anzugs über Funk mit dem zugehörigen Programm *Axis* verbinden.

Das Ganzkörpersetup mit 17 Sensoren entspricht dabei derselben Funktion wie die der *iPi Soft* Aufnahme, d.h. die Hände werden jeweils nur durch einen Sensor vertreten. Die Finger müssen bei Bedarf nachträglich manuell animiert werden.

Um sicherzustellen, dass im Test keine Ungenauigkeiten durch magnetisierte Sensoren entstehen, werden diese zuvor einzeln mit dem mitgelieferten Werkzeug über *Axis* kalibriert bzw. zurückgesetzt. Nach dem Zurücksetzen tritt kein

sichtbarer *Drift* an den Sensoren auf. Anschließend werden die Sensoren in die vorgesehenen Sockel eingerastet und die einzelnen Komponenten des Anzugs miteinander verbunden, um daraufhin die Gurte den Anweisungen nach am Körper zu befestigen. Vor der Vier-Phasen-Kalibrierung muss das personalisierte Profil mit den genauen Körpermaßen geladen werden.

Nach den ersten Prohebewegungen entsteht ein Problem in den Rotationen der Beine, da sich diese beim Anwinkeln unnatürlich verdrehen. Die erste Vermutung liegt in einer ungenauen Position der Sensorsockel, weshalb diese erneut ausgerichtet, sachgerecht befestigt und die Vier-Phasen-Kalibrierung durchgeführt werden. Der Fehler bleibt, unabhängig von der Sockelposition, weiterhin bestehen.

Nach längeren Testversuchen werden die Sockel und Sensoren der Füße mit den Ersatzeinheiten ausgetauscht. Es lässt sich eine leichte Verbesserung feststellen, Fehlinterpretationen der Rotationen bleiben jedoch weiterhin bestehen. Zusätzlich werden das benutzerdefinierte Profil des Avatars verworfen und auf ein Preset gewechselt, welches dem Subjekt am nächsten kommt (*Male 175*). Wieder lassen sich geringe Verbesserungen feststellen. Naturgetreu sind die Bewegungen hingegen nicht. Vermutlich ist ein Messfehler involviert. Neben den fehlerhaften Beinen addiert sich ein Versatz des Rumpfbereichs hinzu, welcher sich mit keiner Neupositionierung des zuständigen Sockels beheben lässt.

Da weitere Verbesserungsversuche keine Wirkung zeigen, sollen die Aufnahmen mit gegebener Konstellation durchgeführt und etwaige Fehler nach Möglichkeit später manuell korrigiert werden.

##### 4.5.4.1 Cleanup

Bevor die Rohdaten in einer externen 3D-Software weiterverwendet werden können, sollten sie zuvor in der proprietären Software *Axis* von den größten Fehlern befreit werden. Hier lassen sich bestimmte Parameter anpassen und daraufhin den Algorithmus den Trackingvorgang neu evaluieren, bis ein nach den Vorgaben entsprechendes Endergebnis entsteht. Der Vorgang entspricht entfernt dem Identifizierungsprozess von verdeckten oder falsch interpretierten Markern bei optischen Systemen. Zur Voransicht der Rohanimation steht ein Hilfscharakter zur Verfügung. Der Prozess findet in drei Stufen statt: Parameter-Feineinstellung > Kontaktpunktbearbeitung > Filterung

#### 4 Praktische Analyse

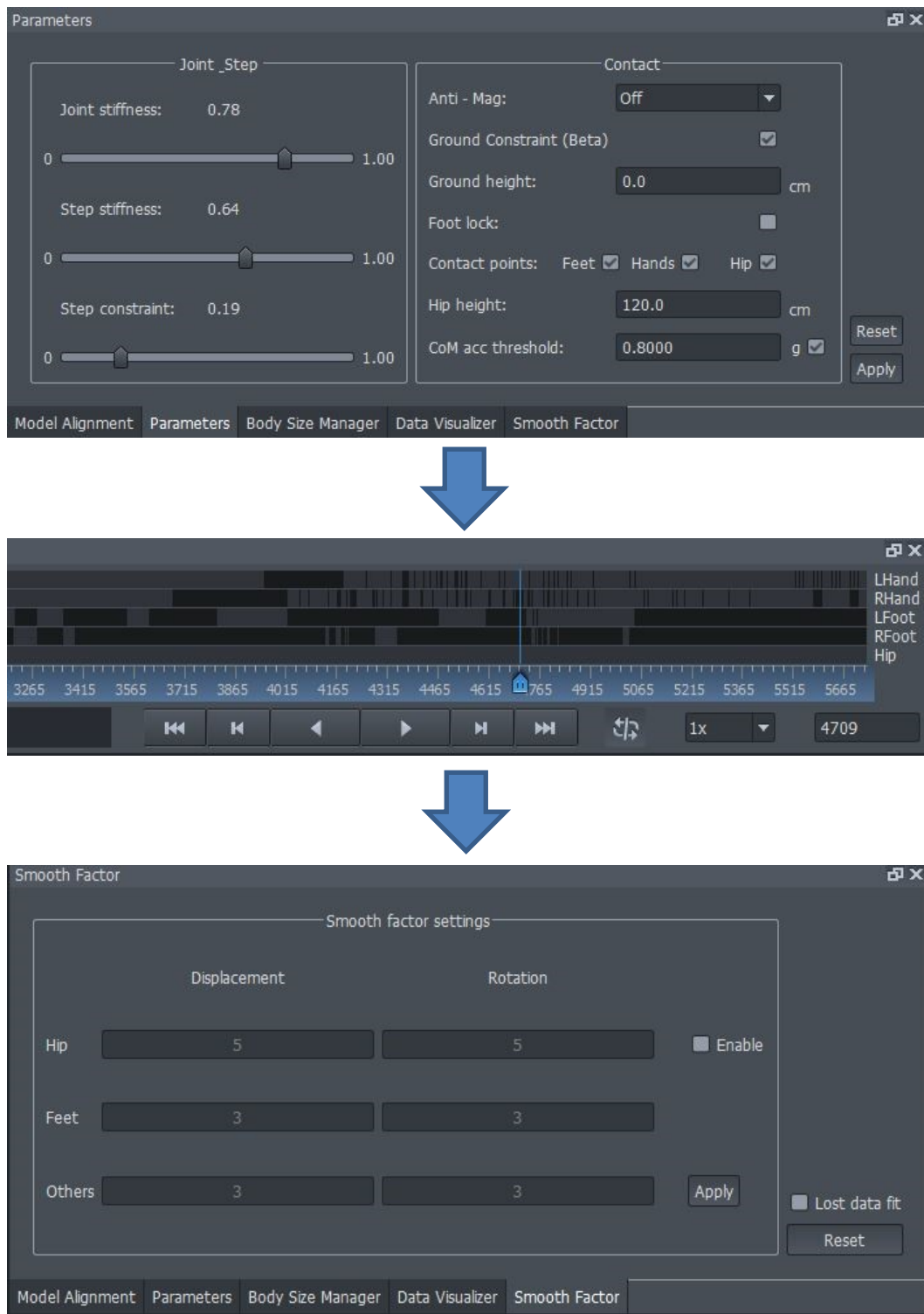


Abbildung 31. Axis Post Processing Workflow

Hauptsächlich betrifft dies die Kontakte der Hände, Füße oder der Hüfte mit dem Boden oder anderen generischen Gegenständen sowie die Akzentuierung der



Bewegung selbst. Durch die Korrektur von Kontaktpunkten können einzelne Artefakte zur Gänze beseitigt werden.

Der Hilfscharakter von *Axis* ist für eine Identifikation subtiler Fehlstellungen, aufgrund seiner Darstellungsform, ungeeignet. Die Daten werden deshalb, wie in Kapitel 3.7.2 erläutert, auf einen 3D-Charakter übertragen. Um Fehlstellungen auszubessern, wird dem Charakter ein sogenanntes *Control-Rig* vergeben, welches als „Arbeitsskelett“ verstanden werden kann. Die Rohdaten des Originalskeletts werden dazu auf das Control-Rig übertragen und dieses bis zum gewünschten Grad manipuliert. Danach lassen sich die veränderten Daten auf das Ursprungs-Skelett oder beliebig viele weitere Skelette übertragen. Die Ursprungsdaten werden bei diesem Vorgang mit den neuen Daten überschrieben bzw. ergänzt.

Mithilfe der FK/IK Funktionen des Control-Rig lassen sich oberflächliche Fehlstellungen in der Körperhaltung dauerhaft korrigieren. So kann beispielsweise ein nach innen eingedrehtes Bein wieder geradegestellt werden.

Der Trackingalgorithmus des Anzugs bezieht sich auf die Ausgangswerte nach der Kalibrierung, weshalb bei groben Kalibrierungsfehlern schwere Probleme wie *Gimbal Lock* oder *Flipping* entstehen können. Diese sind durch markante Ausschläge einzelner Keyframes, innerhalb ihrer Bewegungsgraphen, ersichtlich und lassen sich vorzugsweise mit einem Filter oder per Hand korrigieren.

## 4.6 Zusammenfassung

### 4.6.1 Vicon

Laut der Simon Fraser University Datenbank wurden deren Animationen bereits einem Post-Processing unterzogen. Es kann davon ausgegangen werden, dass zumindest Aufzeichnungslücken und grobe Markerfehler zuvor beseitigt wurden. ([www.mocap.cs.sfu.ca](http://www.mocap.cs.sfu.ca))

Gegenwärtig übernehmen die automatischen Funktionen der *Vicon Blade* Software teilweise diesen aufwendigen Prozess.

Für einen ganzheitlichen Vergleich des Systems ist deshalb eine eigens angefertigte Aufnahme aussagekräftiger, welche als Gegenstand weiterführender Forschungsarbeiten in Betracht gezogen werden sollte. Als optische Referenz sind die Daten jedoch ausreichend.

Es gibt kein Video-Overlay zu den Bewegungsdaten, die Körperhaltung der 3D-Figur entspricht nach dem persönlichen Empfinden des Autors jedoch Großteils dem menschlichen Vorbild.

Die Bewegungen sind nach dem *Solving* in *MotionBuilder* akzentuiert und glaubwürdig. Bei Bodenkontakten unterwandern Arme und Beine an verschiedenen Stellen zeitweise den Boden oder erreichen diesen nicht. Derartige Probleme entstehen unter anderem während dem *Solving*- und *Retargeting*prozess, aufgrund unterschiedlicher Proportionen zwischen Source- und Target-Charakter. Je weniger diese übereinstimmen, desto mehr Komplikationen treten im Nachhinein auf.

Für derartige Fälle gibt es in *MotionBuilder* eine Vielzahl an Feineinstellungen, welche allerdings einen geübten Umgang erfordern. Einzelne Stellen müssen manuell angepasst werden. Unnatürliche Verdrehungen sind teilweise an den Fußgelenken zu beobachten. Trotz präziser Daten sind Korrekturen der Arme und Beine notwendig.

Hoch- und niederfrequente Störungen sind ebenfalls in überschaubaren Ausmaß vorhanden. Ein *Butterworth*- sowie *Peakfilter* und anschließende Interpolationen der Keyframes können diese glätten, ohne dass realistische Details gelöscht werden.

### Vorteile

- äußerst präzise auch in Echtzeit
- vielseitige Anwendungsszenarien
- veränderbare Markerkonfiguration
- stabiles Tracking und Solving
- große Aufnahmeflächen
- hohe Sampleraten
- leistungsstarke Softwarepakete

### Nachteile

- extensiver Hardwareeinsatz
- hoher Einstiegspreis
- hoher Vorbereitungsaufwand
- stationär
- hoher Trainingsaufwand
- störungsanfällig (Licht, Reflexionen, Vibrationen, ...)

### 4.6.2 Perception Neuron

Vor allem im Ein-Mann-Betrieb können Störeinflüsse nicht vollständig ausgeschlossen werden. Beim Kalibrieren und Starten der Aufnahme wird der Host-PC ständig berührt. Im Vorschaubild sind deshalb Abweichungen zur Ausgangsstellung zu beobachten. Ab wenigen Zentimetern Entfernung zu den Geräten sind die Störeinflüsse augenscheinlich nicht mehr wahrnehmbar. Die Choreografie findet mehrere Meter entfernt zu den Aufnahmegeräten statt. Zur Minimierung von additiven *Drift* im Zeitverlauf wird zusätzlich die Vier-Phasen-Kalibrierung vor jedem Take neu durchgeführt.

Bei allen Aufnahmen sind nach wenigen Sekunden Translationsprobleme des gesamten Skeletts zu verzeichnen. Vermutlich entsteht das Problem durch Interpretationsfehler der Bodenkontakte. Dieser Umstand führt bei längeren Sequenzen zu deutlichen Versätzen in der Szene. Die Füße beginnen zwangsläufig zu driften. Zudem werden die Hände nicht richtig interpretiert, da diese trotz identifizierter Kontaktpunkte unrealistisch unterhalb der Grundebene verschwinden.

Der Kontaktpunkteditor ist lediglich ein grobes Hilfsmittel und mehr dazu geeignet, die auffälligsten *Drifts* und plötzliche Relokationen richtigzustellen. Um unklare Fehler effizient zu beseitigen, ist ein Video-Overlay nötig. *Axis* bietet dazu keine Funktion, weshalb die Korrekturen hauptsächlich über Schätzungen durchgeführt werden.

Die Bewegungen wirken steif. Mit der Feineinstellung *Joint Stiffness* in *Axis* kann dem bedingt entgegengewirkt werden. Jeder angepasste Parameter korreliert unweigerlich mit anderen Faktoren wie Bodenkontakten und komplexen Rotationswinkeln. Es bedarf einiges an Erfahrung mit dem *Perception Neuron* System, um hier adäquate Einstellungen zu treffen.

Es stellt sich heraus, dass die Leistung der zur Verfügung stehenden Werkzeuge die entstandenen Probleme nicht abdecken kann. Es ist notwendig, die Aufräumarbeiten auf das Nötigste zu begrenzen und weitere Reparaturen in *MotionBuilder* durchzuführen.

Der Export der MoCap Daten aus *Axis* wirft ein merkwürdiges Problem auf. Sobald die FBX Datei in *MotionBuilder* importiert wird, stimmt die Subjekt-Framerate von 120 fps nicht mit dem entsprechenden Zeitraster von *MotionBuilder* überein, sodass die Animationen mit dem *iPi Soft-* Equivalent nach wenigen Sekunden asynchron auseinanderlaufen.

Da es in *Axis* keinerlei Funktion gibt die Framerate zu ändern, wird die Geschwindigkeit des *Axis*-Skeletts, mithilfe der TimeWarp Funktion, an das Video-Overlay angeglichen.

Bei komplexen Bewegungen, in denen sich der Körper häufig verwindet, sind bereits nach dem ersten Take Veränderungen in der Stellung der Gliedmaßen wahrnehmbar. Es wird vermutet, dass die Gurte verrutschen und dadurch Fehlinterpretationen zustande kommen. Vor allem an den Beinen neigt das Gurtsystem dazu, sich durch ständige Erschütterungen zu lockern. Soll eine konstante Performance des Systems gewährleistet werden, ist ab diesem Zeitpunkt die Aufnahme abubrechen und eine erneute Kalibrierung vorzunehmen.

Die Verbindung des HUB's zur externen Stromquelle wird über ein USB 2.0 Kabel mit Steckerverbindung Typ-B (Mini) hergestellt. Die Verbindung ist für die auftretenden Belastungen während einer Aufnahme unzureichend. Aufgrund einer mangelnden Verarbeitung des HUB's entstehen Wackelkontakte zur Stromversorgung. Die Aufnahme wird dadurch zweimal unterbrochen.

Die Daten bzw. die Körperhaltung sind mit unnatürlichen Rotationen durchzogen. Auch der Versuch, die Haltung über ein IK/FK-Rig nachträglich zu korrigieren, bringt nur marginale Verbesserung, da der Trackingalgorithmus jede ausgedehnte Bewegung fehlinterpretiert und eine unrealistische Darstellung verursacht. Es müssen daher großflächig Daten gelöscht und manuelle Keyframes gesetzt werden, um die Körperhaltung dem Video-Overlay anzugleichen.

Ab diesem Punkt muss der Bearbeiter einen versierten Umgang mit der Software beherrschen und Fachwissen für menschliche Bewegungsabläufe mitbringen, da

ansonsten die Gefahr besteht, die Animation zu verschlechtern. Im eigenen Testlauf tritt dieser Fall ein.

Es müssen so viele Bereiche mit manuellen Keyframes ersetzt werden, dass die Originaldaten nur mehr ansatzweise repräsentativ für die Performance des gesamten Systems stehen. Der *Perception Neuron* Anzug kann einem Vergleich mit *Vicon* nicht standhalten. Auch das *iPi Soft* System liefert stabilere Daten.

Wie effizient ein geübter MoCap-Artist die auftretenden Probleme gelöst hätte, kann der Autor dieser Arbeit aufgrund mangelnder Praxis nicht beurteilen. Es bleibt fraglich, ob sich ein Geschwindigkeitsvorteil gegenüber klassischer Handanimation bereits relativiert hat.

##### **Vorteile**

keine Verdeckungen

günstiger Einstiegspreis

Echtzeit-Streaming

großes Aufnahmevolumen

Multicharakteraufnahme

wenige Störeinflüsse

portabel

##### **Nachteile**

hochempfindlich gegenüber  
Störungen des Referenzmagnetfeldes

Einschränkungen durch  
Kabelverbindungen

empfindliche Hardware

fixe Sensorkonfiguration

kontrollierte Umgebung erforderlich

eingeschränktes Verständnis der  
Software über komplexe Bewegungen

kein Video-Overlay

### 4.6.3 iPi Soft

Das Interface der Software *Mocap Studio* ist übersichtlich und dessen Funktionen im Manual ausführlich beschrieben. Die Aufnahmen können effizient verarbeitet werden.

Bei unkomplizierten Bewegungsabläufen wie die einer Gangsequenz funktioniert die Bewegungserkennung, trotz des geringen farblichen Kontrasts innerhalb der Silhouette (Torso, Arme, Beine, usw.), stabil. Das 3D-Skelett befindet sich überwiegend innerhalb der Silhouette und verformt sich bei ausgedehnten Beugungen flexibel mit dem Darsteller mit. Die Animation enthält keinen *Drift*.

Schwierigkeiten ergeben sich bei Verdeckungen. Sobald die betroffenen Bereiche in mindestens zwei Kameras sichtbar sind, werden sie wieder automatisch erfasst. An den Händen kommt es mehrfach zu Fehlinterpretationen. Die kontrastarmen Handschuhe sind vermutlich ein negativer Einflussfaktor. Dunkle Farbtöne wie dunkelblaue Jeans werden als Schwarz interpretiert, da die Kamera bei hellen Hintergründen automatisch abdunkelt. Womöglich führt eine hellere Farbe zu einer präziseren Subjekterkennung.

Bei bodennahen Bewegungen kommt es zu großflächigen Verdeckungen im Bereich der Hüfte, Arme und Beine. Die Bodenkontakte können nicht richtig berechnet werden. Es entstehen unnatürliche Rotationen und Flipping. Die Grundebene wird daraufhin häufig unterwandert. Durch die für den Algorithmus unlogischen Verdrehungen der betroffenen Stellen, breiten sich die Fehler auf die übrige Subjekterfassung aus. Einige Trackinglücken lassen sich durch erneutes Tracken der betroffenen Bereiche in *Mocap Studio* nicht beheben und müssen in einer externen Software wie *MotionBuilder* korrigiert werden.

Sollte eine Systemerweiterung durch *MotionBuilder* in Betracht gezogen werden, gilt es jährliche Lizenzgebühren von 1.720,- USD zu berücksichtigen. ([www.autodesk.com](http://www.autodesk.com)). Alternativ kann auf das lizenzfreie 3D-Softwarepaket *Blender* ([www.blender.org](http://www.blender.org)) zurückgegriffen werden.

Das Skelett besitzt systembedingt einen ausgeprägten Anteil an *Jitter*, da die Position für jeden Frame neu evaluiert wird. Der *Jitter* verstärkt sich bei schwer zu erfassenden Bereichen zunehmend. In Folge ist ein Smooth Filter unumgänglich. Der in *Mocap Studio* bereitgestellte Filter greift bereits bei geringen Einstellungswerten hart in die Rohdaten ein. Dadurch sind die Bewegungen sichtlich linearisiert und verlieren realistische Feinheiten. Inwieweit diese ursprünglich vorhanden waren, ist aufgrund des ausgeprägten *Jitter* zugleich schwer zu beurteilen. Ein Werkzeug zur Datenanalyse steht in der Form der kostenpflichtigen **Biomech** Erweiterung zur Verfügung. Einen Graph-Editor zum Bearbeiten der Daten gibt es nicht.

Da die Leistung der proprietären Software nicht ausreichend ist, wird das Post Processing in *MotionBuilder* durchgeführt.

Nachdem die Rohanimation in *MotionBuilder* auf einen 3D-Charakter geplottet wurde, lassen sich Unreinheiten und Fehler im Bereich von Bodenkontakten und des gesamten Rumpfes eindeutig feststellen. In Abbildung 32. erkennt man, dass das *iPi Soft* System einen deutlich höheren Anteil an *Jitter* gegenüber dem *Perception Neuron* Anzug beinhaltet. Zum Teil versagt das Tracking völlig und liefert unbrauchbare Daten in verschiedenen Körperbereichen. Die Cleanup-Zeiten sind stellenweise vergleichbar mit dem *Perception Neuron* System, da sekundenlange Bereiche völlig gelöscht und manuell ersetzt werden müssen. Ab hier können einzelne Keyframes bestenfalls als grober Anhaltspunkt herangezogen werden.

Die Angaben des Herstellers, dass bodennaher Körperkontakt zu Problemen führt, kann in diesem Test bestätigt werden. Nach dem hohen *Jitter*-Anteil sind derartige Verdeckungen die größte Schwachstelle des *iPi Soft* Systems und müssen großflächig ersetzt werden. Das *Occlusion Fixing* von *Vicon Blade* arbeitet hier weitaus effizienter.

Ebenfalls anzumerken ist die Anfälligkeit der Kamera-Stativ-Kombination, da diese durch ihre leichte Bauweise empfindlich auf äußere Einflüsse reagiert. Die Aufnahme musste einmalig aufgrund einer Erschütterung des Stativs abgebrochen und das System erneut kalibriert werden.

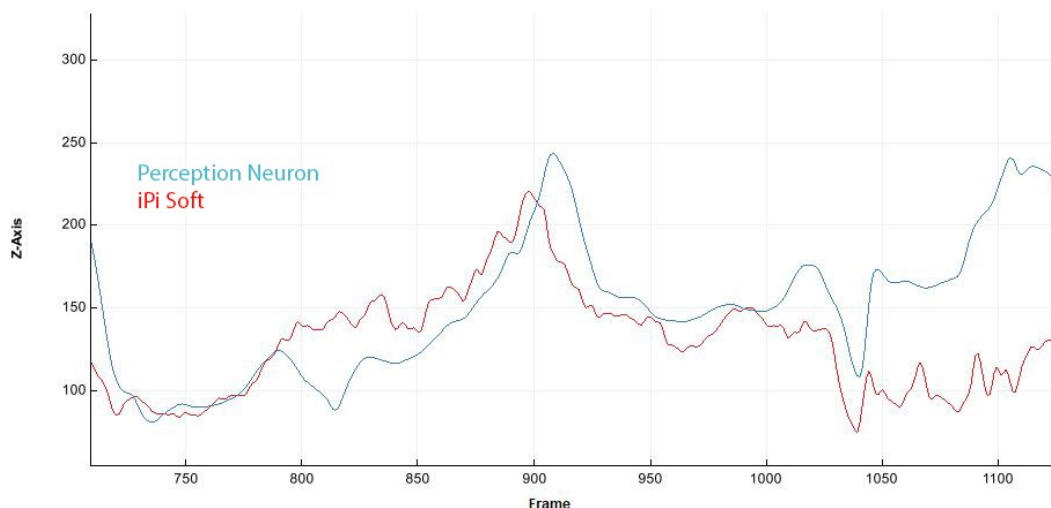


Abbildung 32. Vergleich des *Jitter*-Anteils in den Originaldaten

##### **Vorteile**

portabel und schneller Aufbau

günstiger Einstiegspreis

FBX Export der Szene

Multicharakteraufnahme

unempfindliche Hardware

stabiles Tracking

Video-Overlay

##### **Nachteile**

eingeschränkte Echtzeit  
Datenverarbeitung

kontrollierte Umgebung erforderlich

empfindlich auf Störungen

beschränkt auf menschliche Subjekte

umfangreiche Filterung notwendig

kein Stativgewinde bei günstigen  
(Web-)Kameras

intensives Daten-Processing



# 5 Nutzwertanalyse

## 5.1 Einsatzbereich

Bei diversen Auswahlverfahren ist die Nutzwertanalyse ein gebräuchliches und vielseitiges Werkzeug. Sie wird häufig mit quantitativen Bewertungsverfahren kombiniert. Der Nutzwert beschreibt dabei den subjektiven Eignungswert, welcher die Befriedigung von bestimmten Bedürfnissen widerspiegeln soll. Die Nutzwertanalyse verfolgt das Ziel, verschiedene Lösungsalternativen, in Abhängigkeit zu den Präferenzen einer Entscheidungsperson, in eine geordnete Reihenfolge zu bringen. Je höher der Gesamtnutzwert einer Alternative, desto mehr entspricht diese den Vorstellungen und Zielen des Entscheidungsträgers. Qualitative sowie quantitative Kriterien sind in der Nutzwertanalyse zulässig. Zahlungs- bzw. Kosten-/Ertragsgrößen sind allerdings nicht mit einzubeziehen, da diese in einem separaten, quantitativen Verfahren (z.B. Kapitalwertmethode) erfasst werden und ansonsten doppelt einfließen würden. Hauptsächlich wird die Nutzwertanalyse dann herangezogen, wenn Lösungsalternativen nicht ausschließlich monetär zu bewerten sind oder wenn mehrere, ungleich gewichtete Kriterien erfüllt werden sollen. (Bundesministerium des Innern & Bundesverwaltungsamt, 2018, S. 317)

## 5.2 Zielsetzung

Je nach Zweckmäßigkeit kann die Gewichtung und Beurteilung der Kriterien variieren. Die Zielsetzung legt deshalb den Beurteilungsrahmen fest.

Die folgende Nutzwertanalyse geht von einem klein- bis mittelständischen Unternehmen oder einer Hochschuleinrichtung mit einem allgemeinen Interesse an Motion Capturing aus, welches aufgrund begrenzter finanzieller Mittel bereit ist, Abstriche in mehreren Kategorien in Kauf zu nehmen. Muss-Kriterien sind demnach nur das grundlegende Vermögen zur präzisen Erfassung von Ganzkörperbewegungen, der Einsatzbereich in einer professionellen Studioumgebung und eine ausgewogene Gesamtperformance. Alle getesteten Systeme sollten die Mindestanforderungen laut Hersteller erfüllen können.

### 5.3 Grundregeln

Das Gesamtproblem wird in mehrere Teilprobleme zerlegt. Die Anzahl an Lösungsalternativen ist mindestens zwei und typischerweise fünf. Alles darüber hinaus erschwert zunehmend die Konzentration auf die Individualitäten der jeweiligen Alternativen. Tendenzen und graduelle Entscheidungen, wie die Frage nach der Höhe einer Investitionssumme, lassen sich mit der Nutzwertanalyse nicht beantworten. (Kühnapfel, 2014, S. 2, 6)

### 5.4 Kriterien

#### 5.4.1 Bestimmung

Um die Güte einer Lösungsalternative zu ermitteln, werden bestimmte Kriterien aufgestellt, die für die Entscheidung eines Problems relevant sind. Einen Ausgangspunkt bildet dabei das definierte Zielsystem, welches die Kann- und Muss-Ziele einer Untersuchung beinhaltet. Eine Sammlung aus 10 – 20 Kriterien ist erfahrungsgemäß sinnvoll und sollte das Problem ausreichend durchleuchten, ohne dabei das Verfahren zu verkomplizieren. (Kühnapfel, 2014, S. 9)

In einem 2015 veröffentlichten Bericht des *International Journal of Mathematics and Computers Simulation* stellten dessen Verfasser, unabhängig von der verwendeten Technik, eine Liste aus den 11 wichtigsten Anforderungen an ein MoCap System zusammen:

- **Genauigkeit** – Die Fehlerquote ist das Basisproblem eines jeden MoCap Systems
- **Realtime** – Die Möglichkeit des 3D-Modells, den Bewegungen in Echtzeit zu folgen, bringt einen immensen Effizienzvorsprung
- **Freiheitsgrad** – Je weniger Bewegungseinschränkungen vorhanden sind, desto natürlicher wirken die Animationen
- **Bildfrequenz** – Eine höhere Bildfrequenz lässt schnelle und feine Bewegungen fehlerfreier erfassen
- **Unterbrechungen** – Die Aufnahme darf nicht unterbrochen werden
- **Identifikation** – Die Fähigkeit des Systems, verschiedene Körperteile richtig zu identifizieren
- **Störeinflüsse** – Die Empfindlichkeit des Systems gegenüber äußeren Einflüssen wie Fremdlicht, Magnetfelder usw.
- **Preis** – Das Ziel ist es, hochqualitatives MoCap zu einem möglichst günstigen Preis anzubieten

- **Training** – Die Dauer zur Vertrautmachung des Subjekts mit den Anforderungen des Systems
- **Portabilität** – Die Möglichkeit das System auf verschiedene Orte zu verlagern
- **Software Komplexität** – Die Komplexität der Software zur Aufnahme und Nachbearbeitung der Animation

(Kostov, Yordanova, & Kalchev, 2015, S. 1)

Die Kriterien dienen im weiteren Verlauf als Entscheidungsgrundlage der getesteten Systeme. Da monetäre Interessen nicht direkt in die Nutzwertanalyse miteinfließen dürfen, werden über eine Kostenwirksamkeitsanalyse die Anschaffungspreise mit den errechneten Teilnutzenwerten in Beziehung gesetzt.

### 5.4.2 Gewichtung

Die Muss-Ziele bedürfen in der Nutzwertanalyse keiner weiteren Untersuchung, da die Erfüllung Voraussetzung ist. Die Kriterien aus den Kann-Zielen müssen hingegen mit Punkten gewichtet werden. Eine höhere Punktezahl bestimmt dabei einen höheren *Gewichtungsfaktor*  $G_f$  und somit einen größeren Einfluss auf den Gesamtnutzenwert. Die Summe aller Gewichte beträgt dabei 100%. (Bundesministerium des Innern & Bundesverwaltungsamt, 2018, S. 317), (Kühnapfel, 2014, S. 10)

Um die Gewichtung objektiv zu gestalten, erhält jedes Kriterium eine Relevanz/Gewichtung anhand einer Skala von z.B. 1 – 5, wobei 5 den höchsten Punktwert markiert. Die Punkte aller Kriterien werden anschließend summiert und mit einer Dreisatzrechnung die relativen Prozentanteile der jeweiligen Kriterien an der Gesamtsumme errechnet. (Kühnapfel, 2014, S. 10, 11)

Die Gewichtung wird anhand von allgemeinen Zusammenhängen und den bisherigen Erkenntnissen aus den vorangegangenen Kapiteln vorgenommen. Sie spiegelt deshalb nur die subjektive Meinung des Autors dieser Arbeit wider. Mehrere Teilnehmer sorgen für ein ausgewogeneres Ergebnis. Zur besseren Nachvollziehbarkeit werden die Gedankengänge grob angeführt.

#### 5.4.2.1 Genauigkeit

Ein genaues System verursacht weniger Fehler und verringert automatisch den Berechnungs- bzw. Reparaturaufwand. Umso schneller kann ein fertiges Produkt geliefert werden. Außerdem wird ein realistischer Gesamteindruck wesentlich von feinen Details in den Bewegungen definiert. Sofern ein gewisser Grenzwert kein grundsätzliches Ausschlusskriterium darstellt, sollte Präzision, unabhängig vom System, zumindest die höchstmögliche Relevanz besitzen.

### 5.4.2.2 *Realtime*

Realtime bringt einen beachtlichen Performancevorteil. Auf Probleme kann augenblicklich reagiert und wenn nötig Änderungen vorgenommen werden. Das Risiko für einen Nachdreh oder Reparaturen durch unbemerkte Fehler verringert sich. Die Effizienz am Aufnahmeset steigt. Allgemein ist die Sinnhaftigkeit für Echtzeitmöglichkeiten hoch. Zum Erzeugen von MoCap Daten ist die Funktion nicht zwingend notwendig. Der Parameter wird mit durchschnittlicher Relevanz bewertet.

**Anmerkung:** In dieser Analyse wird lediglich die Möglichkeit zur Echtzeit-Datenverarbeitung bewertet. Über die Performance selbst kann aufgrund fehlender Tests keine Aussage getroffen werden. Vermutlich würde sich andernfalls eine Punkteverschiebung zugunsten des *Vicon*- und *Perception*-Systems ergeben. In diesem Beispiel sind alle Systeme echtzeitfähig und erhalten hier eine vollkommene Zielerfüllung.

### 5.4.2.3 *Freiheitsgrad*

Motion Capturing soll im Wesentlichen komplexe, realistische oder generische Bewegungsabläufe erfassen, welche über klassische Handanimation nicht, oder nur durch einen hohen Aufwand umzusetzen sind. Schränkt das System das Subjekt in irgendeiner Weise ein, werden dessen Animationen dementsprechend unnatürlich wirken. Solange noch ein spürbarer Geschwindigkeitsvorteil herrscht, können Bewegungseinschränkungen in Kauf genommen werden. Der Stellenwert wird durchschnittlich bewertet.

### 5.4.2.4 *Bildfrequenz*

Eine Bildfrequenz von 60 fps ist für gängige MoCap Szenarien ausreichend und sollte mittlerweile von jedem professionellen System unterstützt werden. 30 fps markieren die Untergrenze. Bei schnellen Bewegungen können allerdings höhere Bildraten nötig werden, sodass sich hier eine mögliche Fehlerquelle ergibt. Fehlerquellen gilt es aus Zeit- bzw. Kostengründen zu vermeiden. Alternativ könnte man nach Möglichkeit problematische Bewegungen streichen, was wiederum indirekt den Freiheitsgrad einschränkt. Die Relevanz des Parameters wird hoch bewertet.

### 5.4.2.5 *Unterbrechungen*

Aufnahmeunterbrechungen wirken sich auf sämtliche Bereiche eines MoCap Systems, inklusive erfasstem Subjekt, negativ aus. Unterbrechungsfreiheit erhält deshalb die höchstmögliche Relevanz.

### 5.4.2.6 Identifikation

Eine schlechte Identifikation führt unweigerlich zu Aufnahmeverzögerungen und fehlerhaften Daten. Auf eine stabile Subjekterkennung sollte ein höchstmöglicher Wert gelegt werden.

### 5.4.2.7 Störeinflüsse

Sofern es die Gegebenheiten zulassen, können negative Einflüsse gezielt vermieden werden. Außerdem müssen Störungen nicht zwangsläufig die Präzision beeinträchtigen, sofern das automatische Post Processing die Probleme selbst beheben kann.

Dieses Kriterium kann je nach vorgesehenem Einsatzzweck des Systems zum Ausschlusskriterium werden. Ein möglicher Einflussfaktor wirkt sich gleichzeitig auf mehrere Kriterien wie den Freiheitsgrad, die Identifikation und damit auf Unterbrechungen und/oder die Genauigkeit aus.

Dem Kriterium wird ein hoher Stellenwert eingeräumt.

### 5.4.2.8 Training

Langwierige Eingewöhnungszeiten verzögern die Aufnahme. Unter Umständen führt ein schlechtes Verständnis des Subjekts, über die Anfälligkeiten eines Systems, zu Aufnahme Fehlern oder -Unterbrechungen. Ein Echtzeit-Preview könnte dem entgegenwirken. Die Relevanz ist durchschnittlich.

### 5.4.2.9 Portabilität

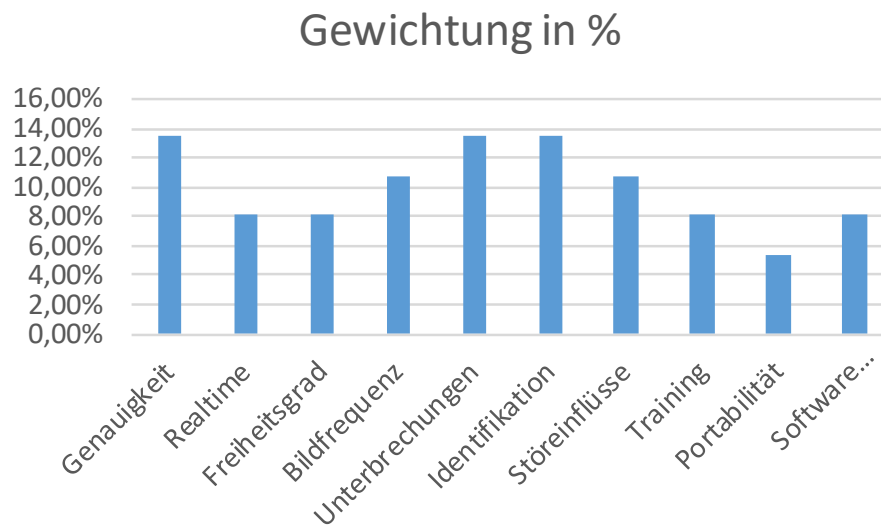
Da das System voraussichtlich nur in einem Studio betrieben wird, ist der Faktor vernachlässigbar und wird unterdurchschnittlich bewertet.

### 5.4.2.10 Software Komplexität

Wie sich im praktischen Versuch herausstellte, neigen zu viele oder verwirrende Einstellmöglichkeiten dazu, die Genauigkeit und in Folge die Performance eines Systems zu beeinträchtigen.

Übernimmt nur ein ausgewählter Personenkreis die Aufnahme am Host-PC, kann diesem Problem mit ausreichend Erfahrung entgegengewirkt werden. Bei vielen verschiedenen Anwendern, wie es in Hochschuleinrichtungen stets der Fall ist, sollten alle grundlegenden Funktionen intuitiv und ohne lange Anlernzeiten zur Verfügung stehen. Die Priorität wird durchschnittlich bewertet.

Daraus ergibt sich folgende Verteilung:



*Abbildung 33. Gewichtung der Kriterien*

### 5.4.3 Bewertung

Der Zielerfüllungsfaktor  $Z_f$  beschreibt den Grad, mit welchem die Kriterien einer Lösungsalternative erfüllt werden. In der Regel verwendet man dazu ein Punktesystem von 0 bis 10, jedes Kriterium unterliegt derselben Spanne. 10 steht dabei für eine überragende Erfüllung des Ziels. Die Vergabe des Faktors kann über eine Diskussion bzw. eine subjektive Schätzung durch eine oder besser mehrere Personen erfolgen. Alternativ lassen sich die Kriterienausprägung mit dem Zielerfüllungsfaktor, mithilfe eines Diagramms, in Beziehung zueinander setzen. Letztere Methode eignet sich eher für quantifizierbare Größen wie beispielsweise eine höchstmögliche Bildrate. (Bundesministerium des Innern & Bundesverwaltungsamt, 2018, S. 318, 319)

<b>Erfüllungsgrad</b>	<b>Zf</b>
nicht erfüllt	0
gerade noch erfüllt	1
ausreichend	2
ausreichend - befriedigend	3
befriedigend	4
befriedigend – gut	5
gut	6
gut – sehr gut	7
sehr gut	8
sehr gut – überragend	9
überragend	10

*Tabelle 1. Definition des Zielerfüllungsfaktors*

Um den Gesamtnutzwert *GN* zu errechnen, müssen zuerst alle Teilnutzwerte *TN* der einzelnen Kriterien einer Lösungsalternative ermittelt werden. Diese ergeben sich aus der Multiplikation der Gewichtungsfaktoren *Gf* mit den Zielerfüllungsfaktoren *Zf*. Daraus lassen sich folgende Formeln ableiten:

$$\mathbf{TN = Gf * Zf}$$

$$\mathbf{GN = \sum TN}$$

(ebd.)

Die Bewertung findet wie in Punkt 5.4.2, anhand der Erkenntnisse und Ergebnisse aus den bisherigen Kapiteln, statt. Für eine objektive Bewertung werden die Kriterien wenn möglich in qualitative Unterkriterien aufgeteilt und diese jeweils in einer Skala zwischen 1, 0.5 und 0 quantifiziert. Die Gesamtwertung der Unterkriterien lässt sich daraufhin über eine lineare Transformation (2. Strahlensatz) der Zielerfüllung (siehe Tabelle 2.) nach zuordnen. Die entsprechende Dokumentation steht im Anhang zur Ansicht bereit.

## 5 Nutzwertanalyse

KRITERIUM	Gf %	Zf	TN	Zf	TN	Zf	TN
		Vero 2.2	Vero 2.2	iPi Soft	iPi Soft	Perception	Perception
Genauigkeit	13,51	9	122	3	39	1	19
Realtime	8,11	10	81	10	81	10	81
Freiheitsgrad	8,11	6	45	5	41	5	36
Bildfrequenz	10,81	10	108	2	20	4	39
Unterbrechungen	13,51	9	126	7	90	3	45
Identifikation	13,51	10	135	6	84	3	34
Störeinflüsse	10,81	3	27	3	36	5	54
Training	8,11	3	24	7	57	4	32
Portabilität	5,41	1	5	7	38	7	38
Software Komplexität	8,11	6	47	6	47	3	27
<b>GESAMTNUTZWERT</b>	<b>100</b>		<b>721</b>		<b>532</b>		<b>406</b>

Tabelle 2. Zuweisung des Zielerfüllungsfaktors und Ermittlung des Gesamtnutzwerts

## 5.5 Ergebnisse

Das *Vicon Vero 2.2* System erzielt mit 721 Punkten den höchsten Gesamtnutzwert und erfüllt damit die Zielvorgaben zu rund 72%. In der Genauigkeit kann es den größten Abstand zu den anderen Systemen verbuchen. Die hohe Bildrate sowie die Subjekterkennung sind ebenfalls herausragend. Die auffälligsten Schwächen zeigen sich in der Portabilität, dem Training und der Anfälligkeit gegenüber Störeinflüssen.

*iPi Soft* überzeugt durch Portabilität, niedrige Trainingszeiten und einen unkomplizierten Umgang mit der proprietären Software. Die Identifikation befindet sich knapp über dem Mittelfeld. Der Freiheitsgrad wird durch die Empfindlichkeit gegenüber Verdeckungen und Schatten vermindert. Die Gesamtgenauigkeit leidet ebenfalls unter diesem Problem. Zusammengefasst ergibt sich ein Nutzwert von 532 Punkten und eine Zielerfüllung von rund 53%.

Trotz der hohen Anfälligkeit durch Magnetfelder bleibt der *Perception Neuron* Anzug, gegenüber den zahlreichen Störeinflüssen optischer Systeme, unempfindlich und kann sich hier in der einzigen Kategorie abheben. Die übrigen Werte sind mittelmäßig bis unterdurchschnittlich. Das größte Problem liegt in der Interpretation der Sensoren sowie den verwirrenden Einstellungen innerhalb der Software. Sie wirken sich direkt auf die Gesamtgenauigkeit aus. Außerdem fließen Unterbrechungen während der Aufnahme negativ in die Gesamtwertung mit ein. Mit 406 Punkten und einem *Zf* von 41% kann das System weniger als die Hälfte der Zielvorgaben erfüllen und positioniert sich damit an letzter Stelle.

Zur **Kostenwirksamkeitsanalyse** müssen nun die Anschaffungskosten durch die Gesamtnutzenpunkte geteilt werden.



## 5 Nutzwertanalyse

---

Durch die vergleichsweise hohe Investition von mindestens 72.000 € ist ein Nutzenpunkt des *Vicon Systems* mit rund 100 € zu bezahlen. Die Kosten von 1.759 € des *Perception Neuron* Anzugs werden mit 4,33 € pro Nutzwert gedeckt. Bei einer Investition von 1.465 € kann *iPi Soft* mit 2,75 € die höchste Kostenwirksamkeit erzielen.

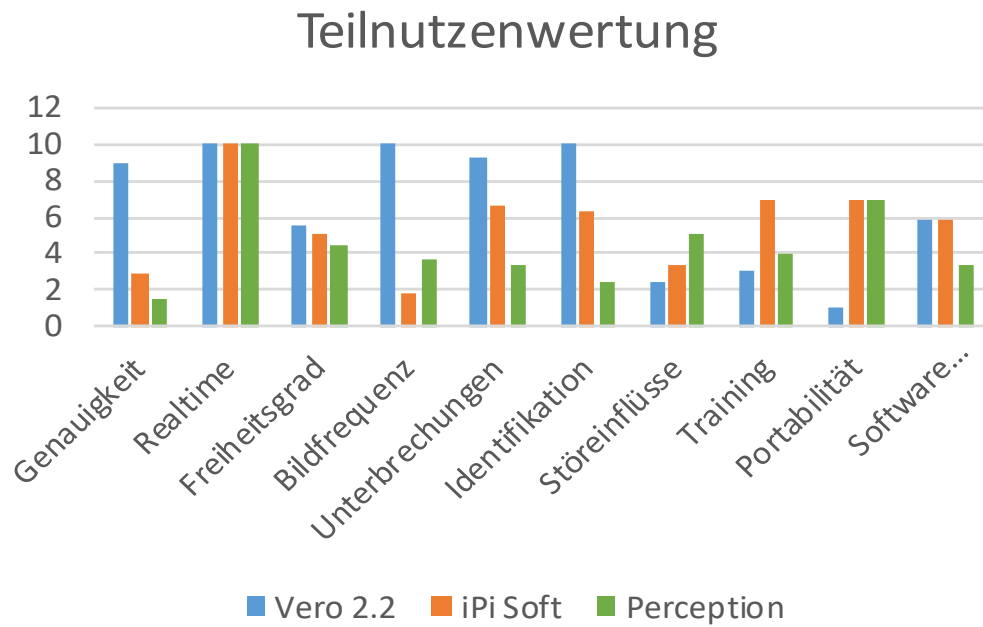


Abbildung 34. Teilnutzenwertung ohne Gewichtung

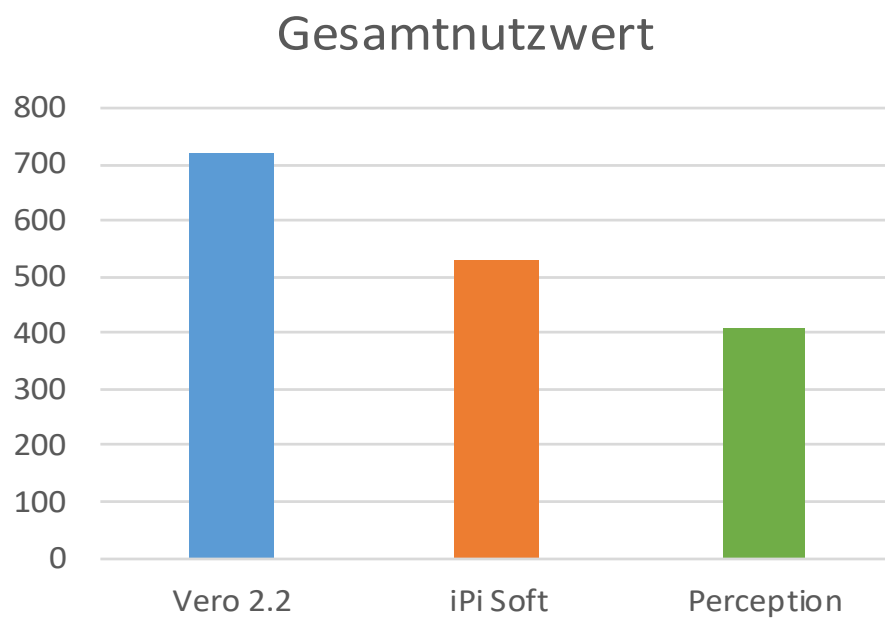


Abbildung 35. Gesamtnutzwert inklusive Gewichtung

## 6 Zukünftige Entwicklungen

### 6.1 Deep Learning in neuronalen Netzwerken

Die Begriffe entstammen dem Bereich des *Machine Learning*. Ein standardisiertes, neuronales Netzwerk besteht aus vielen simplen und miteinander verbundenen Prozessoren bzw. Knotenpunkten, den sogenannten Neuronen. Jedes dieser Neuronen produziert eine Sequenz aus reell wertigen Aktivierungen. Eingangs-Neuronen werden durch Reize von Sensoren aktiviert, welche wiederum Neuronen der darunterliegenden Zwischenschichten (= hidden layers), über gewichtete Verbindungen, auslösen. Die Neuronen der Ausgabeschicht können daraufhin Ereignisse aktivieren und damit ihre Umgebung beeinflussen (z.B. autonomes Fahren). (Schmidhuber, 2014, S. 86)

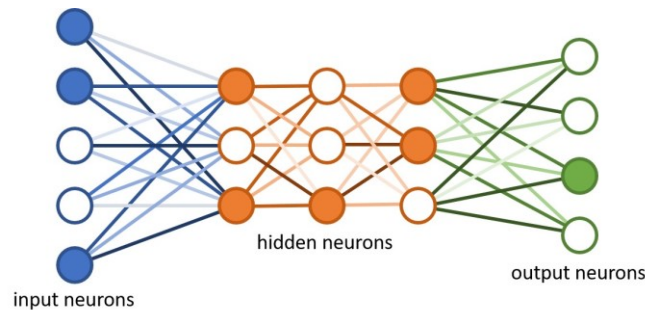


Abbildung 36. künstliches neuronales Netzwerk

Beim *Learning* (= Lernen) geht es darum, die passenden Gewichtungen zu verteilen, um dem neuronalen Netzwerk ein angestrebtes Verhaltensmuster beizubringen. Je nach Komplexität des Problems können dazu unzählige zusammenhängende, rechnerische Schichten nötig werden. Jede Schicht beeinflusst dabei den Aggregatzustand des gesamten Netzwerks. *Deep Learning* verteilt somit präzise Gewichtungen über viele Schichten hinweg. Je verzweigter die Netzwerke sind, desto mehr Trainingsaufwand und Rechenleistung wird dafür erforderlich. Aufgabenlösungen über hochkomplizierte Zusammenhänge, wie beispielsweise das Finanzwesen, sind immer noch problematisch. (ebd.)

Flache neuronale Netzwerkmodelle gibt es bereits seit den sechziger Jahren. Ab dem Millennium bekamen tiefe neuronale Netzwerke großflächige Aufmerksamkeit, indem sie alternative Methoden der künstlichen Intelligenz wie Kernel-Maschinen, in vielen Anwendungsfällen übertrafen. Seit 2009 haben *Supervised Deep Neural Networks* viele offizielle Wettbewerbe der Mustererkennung für sich entschieden. (ebd.)

Dieses Jahr entwickelte ein Zusammenschluss an Forschern deutscher und amerikanischer Universitäten einen künstlich intelligenten Algorithmus, der jegliches Lebewesen mit Motion Capturing erfassen können soll. Optische MoCap Verfahren beziehen sich auf 2D-Videoaufnahmen, welche Bild für Bild identifiziert werden müssen. Mit der öffentlich zugänglichen Software **DeepLabCut** haben die Forscher eine automatisierte Technik entwickelt, die diesen Prozess umgeht und damit wesentlich schneller funktioniert. (Yirka, 2018, S. 1)

Um das Programm *DeepLabCut* zu erschaffen, wurde ein künstlich neuronales Netzwerk mit Informationen aus der **Imagenet** Datenbank trainiert, welche eine enorme Sammlung an Bildern mit zugehörigen Metadaten enthält. Der daraufhin entwickelte Algorithmus kann anhand der antrainierten Metadaten präzise Vorhersagen über die Pose des Subjekts treffen. Die Software selbst steuert den Algorithmus, interagiert mit dem Benutzer und gibt die finalen Markerdaten aus. (ebd.)

Der Benutzer muss Beispiele wie Bilder des zu erfassenden Subjekts hochladen und die wichtigsten Körperteile markieren. Danach werden noch allgemeine Bewegungsaufnahmen desgleichen und letztendlich die Videodatei mit dem relevanten Trackingbereich benötigt. Die prognostizierten Marker entsprechen mindestens der Qualität von Hand gesetzten Markern, wobei dazu ein außerordentlich geringer Trainingsaufwand der Software nötig ist (ebd.)



*Abbildung 37. Sechs Beispiele für automatisch applizierte Marker durch DeepLabCut*

Ein weiteres Beispiel für künstliche Intelligenz in Kombination mit Motion Capturing, ist der von Daniel Holden und seinem Team entwickelte Cleanup-Algorithmus. Optische Systeme wie die von *Vicon* sind äußerst präzise, besitzen jedoch einen entscheidenden Nachteil. Die Rohdaten erfordern durch verdeckte oder falsch identifizierte Marker teilweise eine intensive Nachbearbeitung.

Das Werkzeug überspringt den *Solvingprozess* (Markerdaten > Actor > Skelett) vollautomatisch und ist dabei unempfindlich gegenüber Fehlern in den Markerdaten. Dazu haben die Entwickler ein sogenanntes **deep denoising feed-forward neural network** trainiert. (Holden, 2018, S. 1)

Es arbeitet mit einer Denoising-Funktion, welche die typischen Fehler einer MoCap Aufnahme emuliert, sodass das Netzwerk diese gegebenenfalls erkennt und automatisch überblendet. Für das Training wurden umfangreiche Datensätze wie die der *Carnegie Mellon University MoCap Database* ([www.mocap.cs.cmu.edu](http://www.mocap.cs.cmu.edu)) und deren Markerkonfigurationen herangezogen. Weitere essentielle Berechnungsfaktoren sind das Offset der Marker zu den Gelenken und die zugehörigen *Skinning Weights* (= Verformung des Poly-Mesh). Daraufhin können künstliche Markerpositionen rekonstruiert und enthaltene Fehler in den Originaldaten nachgebildet werden, welche wiederum das Netzwerk trainieren. (Holden, 2018, S. 2)

Abbildung 38. illustriert schemenhaft das Training des Algorithmus. Anhand der Input-Pose  $Y$  wird eine Markerkonfiguration  $Z$  errechnet, woraus sich die Markerpositionen  $X$  rekonstruieren lassen. Diese werden anschließend bewusst korumpiert als Pose  $\hat{X}$  definiert. Zusammen mit den vorgewichteten Marker-Offsets  $\hat{Z}$  werden die neu erhaltenen Daten in das neuronale Netzwerk eingespeist. Das Netzwerk erzeugt daraus die Gelenkstransformationen  $\hat{Y}$ .  $\hat{Y}$  wird daraufhin mit der Originalpose  $Y$  verglichen und der errechnete Messfehler  $L$  zur Neugewichtung der Neuronen im Netzwerk herangezogen.

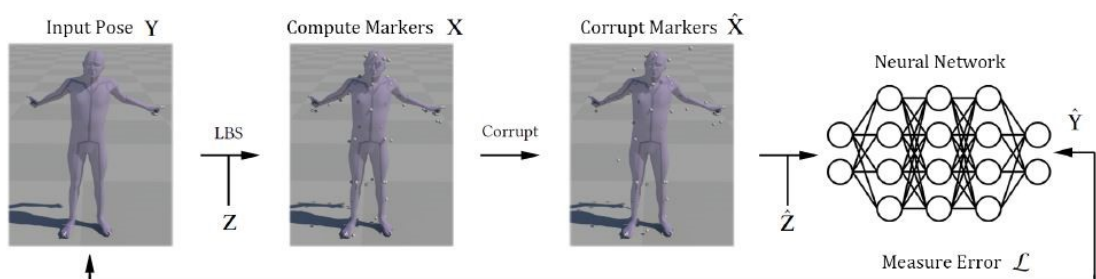


Abbildung 38. Trainingsbeispiel eines neuronalen Netzwerks

Die Fehlerquote misst wenige Millimeter und hat somit Produktionsqualität. Darüber hinaus benötigt das System kaum Prozessorleistung sowie Arbeitsspeicher. Durch seine Robustheit eliminiert die Technik jeglichen Bedarf an manuellen Verbesserungsarbeiten und ermöglicht einen wesentlich höheren Datendurchsatz als es bisher möglich war. (Holden, 2018, S. 1)

Eine weitere Ausführung integrierter, neuronaler Netzwerke würde den Umfang dieser Arbeit übersteigen. Auf die Forschungsprojekte ***DensePose: Dense Human Pose Estimation In The Wild*** sowie ***Mode-Adaptive Neural Networks for Quadruped Motion Control*** soll aufgrund des hohen Forschungspotentials an dieser Stelle verwiesen werden. (Güler, Neverova, & Kokkinos, 2018), (Zhang, Starke, Komura, & Saito, 2018)

### 6.2 Quantencomputer

*Machine Learning* und Quantencomputer sind zwei Technologien, die sich jeweils auf bisher schwer- oder unlösbare Probleme in der Datenberechnung fokussieren. In Kombination mit *Support Vector Machines* (SVMs) sind Kernel-Methoden normalerweise die am meisten bekannten, binären Methoden für Klassifizierungen bei Mustererkennungsaufgaben. Trotz allem stoßen diese Systeme durch zu große Merkmalsbereiche und rechenintensive Kernel-Funktionen immer wieder an ihre Grenzen. Quantencomputer eröffnen hier neue Möglichkeiten. Die Kreuzung zwischen *Machine Learning* und Quantencomputer wird als *Quantum Machine Learning* bezeichnet, welches in den letzten Jahren zunehmend an Aufmerksamkeit gewonnen hat. Daraus entstanden mittlerweile einige bedeutende Quanten-Algorithmen. (Havlicek u. a., 2018, S. 1)

Das übliche Berechnungsmodell unserer Computer basiert auf dem allgemein bekannten Prinzip von Bits bzw. deren Zuständen 0 oder 1, und der Verarbeitung mit simplen logischen Gattern (AND, OR, NOT, NAND). Quantencomputer arbeiten hingegen mit kontrollierten Elementarteilchen wie Elektronen, die einen weitaus größeren Bereich an möglichen Zuständen annehmen können. Eine Einheit nennt sich Quanten-Bits oder qubits und kann die Zustände 0,1 oder eine lineare Kombination aus beiden annehmen. Letzteres Stadium wird als **Superposition** bezeichnet, eine beliebige Überlagerung von 0 und 1. Bei Quantencomputern hat die Position eine wesentliche Besonderheit, da sie eine exponentiell hohe Anzahl an möglichen logischen Zuständen annehmen kann, d.h. alle Zustände zwischen  $|00\dots 0\rangle$  und  $|11\dots 1\rangle$  (die Halbwinkel-Klammer  $| \rangle$  impliziert qubits). Kein konventioneller Computer ist dazu in der Lage. (IBM Research, 2017)

Außerdem bedienen sich Quantencomputer einem weiteren Phänomen aus der Quantenphysik, dem **Entanglement** bzw. der Quantenverschränkung. Sobald zwei Teilchen miteinander verschränkt sind, können Veränderungen an dem einem Teilchen absolut gleichzeitig beim anderen Teilchen ausgelesen werden. Die Fähigkeit der Verschränkung ist ein wesentlicher Faktor der überragenden Leistung von Quantencomputern. Um die Rechenleistung bzw. das **Quantenvolumen** des gesamten Systems zu erhöhen, bedarf es einer höheren qubits-Anzahl und einer niedrigeren Fehlerquote. (ebd.)

Quantencomputer mit weniger als rund 50 qubits können von den größten heutigen Computern simuliert werden. Alles darüber hinaus schlägt sich in einem deutlichen

Geschwindigkeitsvorteil nieder, wobei sich der Leistungsvorsprung mit jedem zusätzlichen qubit verdoppelt. Dieser Schwellenwert wird als *quantum advantage* oder *-supremacy* bezeichnet. (Strohm & Rölver, 2018, S. 3)

Google hat im März 2018 diese Grenze überschritten und seinen bisher stärksten Prozessor-Prototypen mit 72 qubits vorgestellt. Google setzt dabei wie der Technikriese IBM auf supraleitende qubits, deren physikalischen Zustände bei Temperaturen nahe dem absoluten Nullpunkt (-273 °C) mithilfe von Mikrowellen gesteuert werden.

Im Moment stehen via dem IBM Q Experience Netzwerk insgesamt vier Quantencomputer (drei davon öffentlich von 5 bis 14 qubits) und ein Simulator mit 32 qubits für wissenschaftliche Forschungsarbeiten zur Verfügung. ([www.research.ibm.com](http://www.research.ibm.com))

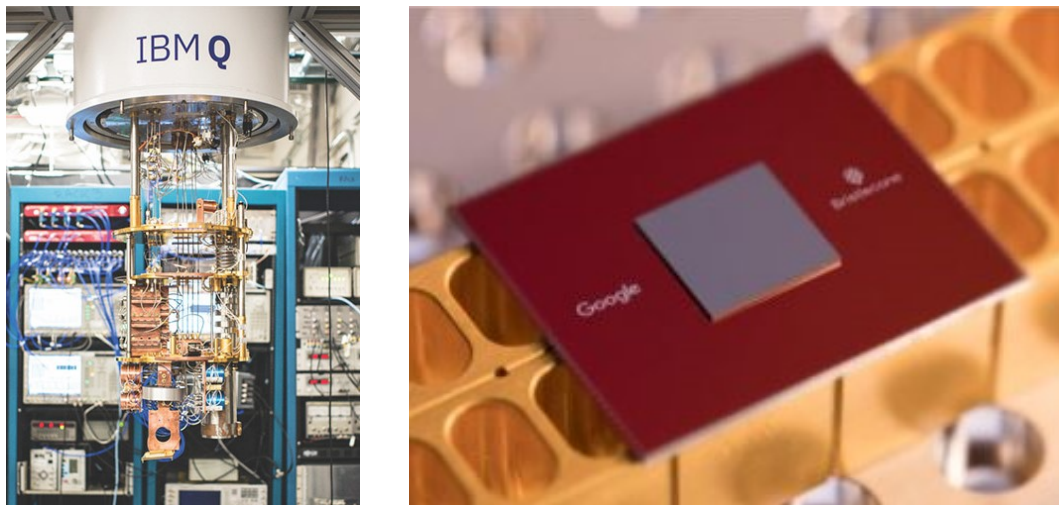


Abbildung 39. IBM Q Quantencomputer (li.) und Google's 72 qubit Prozessor (re.)

Obgleich die kommerzielle Nutzung der Quantentechnologie noch am Anfang steht, werden gegenwärtig erste profitable Geschäftsfelder in der Chemie, Wirtschaft, Finanz, Logistik und künstlichen Intelligenz erforscht. Da Quantencomputer eine große Fläche beanspruchen, wird Rechenzeit auf selbigen vermutlich als Dienstleistung in Form von Rechenzentren angeboten. Der nötige Aufwand ist aktuell unverhältnismäßig hoch. Es wird jedoch davon ausgegangen, dass in den nächsten fünf Jahren simple, für konventionelle Computer unlösbare Aufgaben, berechenbar sind. (Filipp, 2018), (Strohm & Rölver, 2018, S. 2)



# 7 Fazit

*Vicon's* markerbasierte Systeme beinhalten hohe Anschaffungskosten, einen erheblichen Vorbereitungsaufwand und einen intensiven Hardwareeinsatz. Sie sind dagegen ausgereift sowie vielseitig und liefern herausragend präzise Bewegungsdaten. Die bisherige Fehlerrate wurde durch intelligente Algorithmen soweit verbessert, dass hochpräzises Tracking unter suboptimalen Bedingungen und in Echtzeit möglich ist. Die Usability im Soft- und Hardwarebereich konnte ebenfalls deutlich verbessert werden. Die Gesamtpformance erreicht in der Nutzwertanalyse mit Abstand die höchste Bewertung.

Das markerlose System von *iPi Soft* ist preiswert, portabel und liefert eine robuste Trackingperformance. Aufbau und Aufnahme können von nur einer Person durchgeführt werden. Schnelle Bewegungen werden ausreichend erkannt. Bei Verdeckungen, geringem Szenenkontrast oder Abschattungen, versagt das System zum Teil völlig. Durch den hohen Anteil an hochfrequentem *Jitter*, müssen die Daten umfangreich gefiltert werden, worunter der realistische Gesamteindruck leidet. Einige Trackingfehler können in der proprietären Software nicht behoben werden. Für High-End Szenarien mit einem ausgeprägten Bedarf an Präzision ist das System ungeeignet. Bei weniger anspruchsvollen Einsätzen, und einer Erweiterung des Systems durch leistungsstarke Softwarepakete, ist *iPi Soft* eine interessante Alternative zum Industriestandard.

Der *Perception Neuron* Anzug besitzt einen verhältnismäßig günstigen Preis, eine kompakte Hardware und einen moderates Rauschverhalten der Sensoren. Theoretisch kann das System ortsungebunden und im Freien eingesetzt werden. Die Gesamtpformance des Systems schneidet allerdings schlecht ab. Die Komponenten des Anzugs sind fragil und verschieben sich. Die Vier-Stufen-Kalibrierung führt zu keiner korrekten Interpretation der Körperhaltung. Störungen gegenüber dem Referenzmagnetfeld können nicht ausreichend abgeschirmt werden, sodass schwere Fehler wie *Drift* über die gesamte Animation hinweg auftreten. Die proprietäre Software *Axis* kann Bodenkontakte nicht ausreichend präzise berechnen. Die Feineinstellungen des Trackingalgorithmus sind wenig intuitiv. Die eigenen Werkzeuge können einige Trackingfehler nicht beseitigen.

Wäre eine adäquate Abschirmung gegenüber Magnetfeldern vorhanden, ließen sich möglicherweise ähnlich akkurate Daten wie mit *iPi Soft* oder sogar darüber hinaus erzielen. Inertial Motion Capturing, in der Preisklasse des *Perception Neuron* Anzugs, ist nach den Erkenntnissen dieser Thesis, für professionelle Anwendungen, nicht ausgereift.

Im Laufe der Ausarbeitung dieser Thesis ergaben sich spannende Fragen, die aufgrund ihrer Reichweite nicht mehr ausreichend erforscht werden können. Worin liegen beispielsweise die Unterschiede in der Echtzeitperformance der verschiedenen Techniken und was ändert sich bei mehreren Subjekten? Wie ließe sich die Präzision markerloser Systeme drastisch erhöhen? Wieweit können inertielle MoCap Systeme gegen Störfelder abgeschirmt werden (XSens)? Wie ließe sich die Rechenleistung von Quantencomputern, für Motion Capturing verfügbar machen und wieweit verändert künstliche Intelligenz die Methoden zur Bewegungserfassung?

Motion Capturing ist ein faszinierendes Themengebiet, mit einer Vielzahl an möglichen Anwendungsbereichen. Die verwendeten Systeme besitzen ihre spezifischen Stärken und Schwächen. Der aktuelle Marktführer limitiert sich durch sein Unvermögen, auf Marker zu verzichten. Währenddessen ist es durch die rasanten Entwicklungen hinsichtlich künstlicher Intelligenz und Quantencomputer absehbar, dass hochpräzises, markerloses Motion Capturing in Echtzeit, innerhalb der nächsten zehn Jahre Realität werden könnte. Der „heilige Gral“ scheint damit in greifbarer Nähe.

## 8 Glossar

<b>Attitude</b>	engl. für Fluglage/-Horizont
<b>Augmented Reality</b>	engl. für erweiterte Realität
<b>Augmented Virtuality</b>	engl. für erweiterte Virtualität
<b>Bones</b>	engl. für Knochen
<b>Buffer</b>	hier engl. für temporär genutzter Zwischenspeicher für Ein- und Ausgangsdaten während der Übertragung
<b>Cartoon</b>	engl. für Karikatur
<b>CCO = Chief Creative Officer</b>	engl. für ranghöchster Kreativchef im Marketing
<b>Cel Animation</b>	Animationstechnik mithilfe von Celluloidfolie
<b>Character-Animation</b>	hochwertige Animation einer Figur im Animationsfilm
<b>Charged-Coupled-Device</b>	engl. für ladungsträgergekoppelte Schaltung
<b>Cleanup</b>	engl. für Aufräumarbeiten
<b>Compositing</b>	hier engl. für Mischung/Zusammensetzung
<b>Corner</b>	engl. für Ecke/Winkel
<b>Deep Learning</b>	engl. für tiefgehendes Lernen
<b>Degree of Freedom</b>	engl. für Freiheitsgrad
<b>Digital Puppetry</b>	engl. für Echtzeitmanipulation- und Rendering von digitalen 2D- und 3D-Charakteren in einer virtuellen Umgebung

<b>Effector</b>	engl. für Effektor, Element zum Manipulieren von Joint-Hierarchien
<b>Ethernet</b>	Familie an Computer-Netzwerktechniken
<b>Facial</b>	engl. für im Gesicht
<b>Feature-Selection</b>	hier engl. für Auswahl anhand von relevanten Merkmalen, um Modelle zu konstruieren
<b>Flipping</b>	hier engl. für schlagartig verdrehte Gelenke
<b>Flipping</b>	engl. für schlagartige Verdrehungen
<b>Forceplate</b>	Messinstrument für Bodenreaktionskräfte
<b>Forward Kinematic</b>	Vorwärts Kinematik
<b>Frame</b>	Einzelbild
<b>Framedrop</b>	engl. für verworfene Bilder
<b>Frames per Second</b>	engl. für Bilder pro Sekunde
<b>Framework</b>	hier engl. für Programmiergerüst
<b>Framing</b>	engl. für Bildausschnitt einer Kamera
<b>Gimbal</b>	engl. für kardanischer (entkoppelter) Rahmen
<b>Gimbal Lock</b>	engl. für blockierte Rotationsachse
<b>GPU = Graphics Processing Unit</b>	engl. für Grafikprozessor
<b>GUI = Graphic User Interface</b>	engl. für grafische Benutzeroberfläche
<b>Heading</b>	engl. für Kursanzeige
<b>Head-Mounted-Display</b>	engl. für am Kopf angebrachter Bildschirm
<b>Hierarchy</b>	engl. für Hierarchie
<b>Host</b>	hier engl. für Hauptcomputer
<b>In between</b>	engl. für dazwischen
<b>Inertial Measurement Unit</b>	engl. für Trägheitssensor

<b>inside-in</b>	engl. für innerlich
<b>inside-out</b>	engl. für von Innen nach Außen
<b>Inverse Kinematic</b>	Inverse Kinematik
<b>Jitter</b>	hier engl. für hochfrequente Genauigkeitsschwankungen/Zittern
<b>Joints</b>	engl. für Gelenke
<b>Keyframes</b>	Schlüsselbilder
<b>LED</b>	Light Emitting Diode
<b>Local Positioning System</b>	engl. für lokales Positionierungssystem
<b>Machine Learning</b>	engl. für maschinelles Lernen
<b>Machine Learning</b>	engl. für Maschinenlernen
<b>Mesh</b>	hier engl. für ein Netz aus Polygonen, welche die Hülle des virtuellen Charakters/Objekts ausmachen
<b>Motion Capturing (kurz MoCap)</b>	engl. für Bewegungserfassung
<b>Motion Graph</b>	engl. für Gradiationskurve einer Bewegung im Zeitverlauf
<b>Occlusion Fixing</b>	Algorithmus zur automatischen Rekonstruktion von physisch verdeckten Markern
<b>outside-in</b>	engl. für von Außen nach Innen
<b>Pipeline</b>	hier engl. für Aneinanderreihung verschiedener Prozesse innerhalb einer Produktionskette
<b>Preset</b>	engl. für Voreinstellung
<b>Ray Tracing</b>	Algorithmus basierend auf Strahlenausendung zur Ermittlung von Verdeckungen an 3D-Objekten
<b>Remapping</b>	engl. für das Umrechnen/Übertragen von Eigenschaften bestimmter Referenzpunkte eines Subjekts auf Referenzpunkte eines anderen Subjekts

<b>retroreflektierend</b>	Einfallrichtung entspricht Ausfallrichtung einer elektromagnetischen Welle
<b>ROI = Region of Interest</b>	engl. für relevanter Bereich
<b>Root</b>	hier engl. für Wurzel/Ursprung
<b>Samplingrate</b>	Abtastfrequenz
<b>Skins</b>	hier engl. für Polygonhülle eines 3D-Skeletts
<b>SMPTE Timecode = Society of Motion Picture Television Engineers</b>	Standardisierter Zeitstempel zur Synchronisation von Audio- und Videogeräten bzw. -Inhalten
<b>Source- und Targetskelett</b>	engl. für Ausgangs- und Zielskelett
<b>Stand-In</b>	engl. für Platzhalterfigur des eigentlichen Subjekts
<b>State-of-the-Art</b>	auf dem neuesten Stand der Technik
<b>Thick Epitaxial Layer</b>	engl. für dicke Halbleiterschicht
<b>Timecode</b>	engl. für Zeitstempel
<b>Tracker</b>	Algorithmus zum Identifizieren bzw. Verfolgen eines Objekts
<b>Triple AAA</b>	höchster Qualitätsstandard in der Videospielindustrie
<b>Ultrawideband</b>	engl. für Ultrabreitband
<b>VESA = Video Electronics Standards Association</b>	einheitlicher Videostandard
<b>VFX = visual effects</b>	engl. für visuelle Effekte
<b>Video-Overlay</b>	engl. für Überblendung mit einer Videoaufnahme
<b>Witness-Camera</b>	engl. für Zeugenkamera

# 9 Anhang

## 9.1 Kriterienbewertung

### Legende:

1 = trifft zu, 0.5 = trifft teilweise zu, 0 = trifft nicht zu

SW = schlechtester Wert, BW = bester Wert, K = Kriteriumswert,  $x = Z_f(K)$

HP = höchste Punktzahl = 10, NP = niedrigste Punktzahl = 0

Punktetransformation nach dem 2. Strahlensatz:  $x = (K - SW) \cdot \frac{HP - NP}{BW - SW} + NP$

### GENAUIGKEIT

Zf	K		Kriterien	Vero 2.2	iPi Soft	Perception
0	7	= SW	Aufzeichnungsluecken	0,5	1	1
1	6,3		hochfrequente Stoerungen	0,2	1	0,5
2	5,6		Drift durch System	0	0	1
3	4,9		Flipping	0	1	1
4	4,2		unrealistischer Gesamteindruck	0	0,5	0,5
5	3,5		Durchdringen der Grundebene	0	0,5	1
6	2,8		kein Occlusion Fixing	0	1	1
7	2,1					
8	1,4					
9	0,7					
10	0	= BW				
K (riteriumswert) ->				0,7	5,0	6,0
Zf =				Zf =	Zf =	Zf =
Zf (K) gerundet ->				9	3	1

FREIHEITSGRAD

Zf	K		Kriterien	Vero 2.2	iPi Soft	Perception
0	10	= SW	Marker-/Sensorverlust	1	0	1
1	9,0		Kabelverbindungen Körper	0	0	1
2	8,0		Aktionsradius/Sichtkontakt	1	1	0,5
3	7,0		Fehlinterpretation durch Algorithmus	0	0,5	1
4	6,0		begrenzte Kleidungsfarbe	0,5	1	0
5	5,0		begrenzter Kleidungsstil	1	0,5	0,5
6	4,0		mechanische Einschränkungen	0	0	0
7	3,0		kein dynamischer Szeneninhalt	0,5	1	0
8	2,0		starre Marker-/Sensorposition	0,5	0	0,5
9	1,0		Limitation auf humane Subjekte	0	1	1
10	0	= BW				
K (riteriumswert) ->				4,5	5,0	5,5
				Zf =	Zf =	Zf =
Zf (K) gerundet ->				6	5	5

BILDFREQUENZ

Zf	K		Kriterien	Vero 2.2	iPi Soft	Perception
0	0	= SW	max Framerate	330	60	120
1	33					
2	66					
3	99					
4	132					
5	165					
6	198					
7	231					
8	264					
9	297					
10	330	= BW				
K (riteriumswert) ->				330	60	120
				Zf =	Zf =	Zf =
Zf (K) gerundet ->				10	2	4



### UNTERBRECHUNGEN

Zf	K		Kriterien	Vero 2.2	iPi Soft	Perception
0	3	= SW	De-Kalibrierung	0,2	1	1
1	2,7		Software Absturz	0	0	0
2	2,4		sonstige Unterbrechung	0	0	1
3	2,1					
4	1,8					
5	1,5					
6	1,2					
7	0,9					
8	0,6					
9	0,3					
10	0	= BW				
K (riteriumswert) ->				0,2	1,0	2,0
				Zf =	Zf =	Zf =
Zf (K) gerundet ->				9	7	3

### IDENTIFIKATION

Zf	K		Kriterien	Vero 2.2	iPi Soft	Perception
0	4	= SW	unzureichende Feineinstellungen	0	0	0
1	3,6		fehlerhafte Kalibrierung	0	0,5	1
2	3,2		ungenauere Kalibrierungsmethode	0	0,5	1
3	2,8		inkonsistente Algorithmen	0	0,5	1
4	2,4					
5	2,0					
6	1,6					
7	1,2					
8	0,8					
9	0,4					
10	0	= BW				
K (riteriumswert) ->				0,0	1,5	3,0
				Zf =	Zf =	Zf =
Zf (K) gerundet ->				10	6	3

## 9 Anhang

### STOERUNGEN

Zf	K		Kriterien	Vero 2.2	iPi Soft	Perception
0	6	= SW	Tageslicht/Studiolicht	1	1	0
1	5,4		Störfelder	0	0	1
2	4,8		Verrutschen/Verlust der Marker/Sensoren	1	0	1
3	4,2		Verdeckungen	0,5	1	0
4	3,6		Vibrationen der Kamera/Sensoren	1	1	1
5	3,0		Hintergrund (Reflexion, Änderungen)	1	1	0
6	2,4					
7	1,8					
8	1,2					
9	0,6					
10	0	= BW				
K (riteriumswert) ->				4,5	4,0	3,0
Zf =				Zf =	Zf =	Zf =
Zf (K) gerundet ->				3	3	5

### TRAINING

Zf	K		Kriterien	Vero 2.2	iPi Soft	Perception
0	5	= SW	Limits durch Sensoren/Marker	1	0	1
1	4,5		Limits durch Kleidung	1	0,5	0,5
2	4,0		Limits durch Kabelverbindungen	0	0	1
3	3,5		professionelles Acting durch Präzision (Erfassung von Nuancen)	1	0,5	0,5
4	3,0		leeres Bühnenbild durch Location-Limits	0,5	0,5	0
5	2,5					
6	2,0					
7	1,5					
8	1,0					
9	0,5					
10	0	= BW				
K (riteriumswert) ->				3,5	1,5	3,0
Zf =				Zf =	Zf =	Zf =
Zf (K) gerundet ->				3	7	4

PORTABILITÄT

Zf	K		Kriterien	Vero 2.2	iPi Soft	Perception
0	5	= SW	unbewegliche Hardware	0,5	0	0
1	4,5		umfangreiche Hardware	1	0,5	0,5
2	4,0		empfindliche Hardware	1	0	1
3	3,5		schwere Hardware	1	0,5	0
4	3,0		großes Packmaß	1	0,5	0
5	2,5					
6	2,0					
7	1,5					
8	1,0					
9	0,5					
10	0	= BW				
K (riteriumswert) ->				4,5	1,5	1,5
				Zf =	Zf =	Zf =
Zf (K) gerundet ->				1	7	7

SOFTWAREKOMPLEXITÄT

Zf	K		Kriterien	Vero 2.2	iPi Soft	Perception
0	6	= SW	umfangreiche Einstellmöglichkeiten	1	0,5	0,5
1	5,4		unintuitive Einstellmöglichkeiten	1	0,5	1
2	4,8		kompliziertes GUI	0,5	0	0
3	4,2		fehlende Unterstützung Video-Overlay	0	0	1
4	3,6		ungenügende Tool-Performance	0	0,5	1
5	3,0		fehlende Graphen-Ansicht/Editor	0	1	0,5
6	2,4					
7	1,8					
8	1,2					
9	0,6					
10	0	= BW				
K (riteriumswert) ->				2,5	2,5	4,0
				Zf =	Zf =	Zf =
Zf (K) gerundet ->				6	6	3

# Literaturverzeichnis

Agren, S. (2017). Object tracking methods and their areas of application: A meta-analysis. Universität Umeå. Abgerufen von <http://www8.cs.umu.se/education/examina/Rapporter/SannaAgrenFinal.pdf>

Autodesk Inc. (2017). MotionBuilder Help 2017 Help: Filtering. Abgerufen 25. November 2018, von <http://help.autodesk.com/view/MOBPRO/2017/ENU/?guid=GUID-B1768B7C-9469-485D-9636-707205CEB4C7>

Berger, R. T. (2016). *Ein Low-Budget Mocap System als Alternative zu den Industrie Standards*. St. Pölten.

Bouaziz, S. (2015). *Realtime Face Tracking and Animation*. École Polytechnique Fédérale de Lausanne. Abgerufen von [https://lgg.epfl.ch/publications/2015/Sofien\\_Thesis/thesis.pdf](https://lgg.epfl.ch/publications/2015/Sofien_Thesis/thesis.pdf)

Bundesministerium des Innern, & Bundesverwaltungsamt. (2018). *Handbuch für Organisationsuntersuchungen und Personalbedarfsermittlung*. Berlin: Bundesministerium des Innern. Abgerufen von [https://www.orghandbuch.de/OHB/DE/ohb\\_pdf.pdf;jsessionid=E0DDEB37F02A6A6A3B6BF9CA8D4D3921.2\\_cid350?\\_\\_blob=publicationFile&v=28](https://www.orghandbuch.de/OHB/DE/ohb_pdf.pdf;jsessionid=E0DDEB37F02A6A6A3B6BF9CA8D4D3921.2_cid350?__blob=publicationFile&v=28)

Dietmar Jackèl, Neunreither, S., & Wagner, F. (2006). *Methoden der Computeranimation*. Springer.

Filipp, S. (2018). Quantencomputer: Der Beginn der kommerziellen Quanten-Ära. Abgerufen 4. Dezember 2018, von undefined

Fischer, T. (2016). *Technische Analyse eines Full-Body immersiven VR-Systems realisiert mit Motion Capturing, Unity 3D und Oculus Rift*. Hamburg: Hamburg University of Applied Sciences.

Götz, K. (2015). *Virtual Production - Possibilities and Limitations of Virtual Production Environments Optimization through Implementation of Innovative Interfaces*. Stuttgart: Hochschule der Medien.

Güler, R. A., Neverova, N., & Kokkinos, I. (2018). DensePose: Dense Human Pose Estimation In TheWild. Cornell University. Abgerufen von <https://arxiv.org/abs/1802.00434>

Havlicek, V., Córcoles, A. D., Temme, K., Harrow, A. W., Kandala, A., Chow, J. M., & Gambetta, J. M. (2018). Supervised learning with quantum enhanced feature

spaces. IBM T.J. Watson Research Center, Center for Theoretical Physics, Massachusetts Institute of Technology.

Hol, J. (2011). *Sensor Fusion and Calibration of Inertial Sensors, Vision, Ultra-Wideband and GPS*. Linköping: Linköping University.

Holden, D. (2018). Robust Solving of Optical Motion Capture Data by Denoising. Association for Computing Machinery. Abgerufen von <https://dl.acm.org/citation.cfm?id=3201302>

IBM Research, I. Q. (2017). IBM Q Experience Documentation. Abgerufen 4. Dezember 2018, von <https://quantumexperience.ng.bluemix.net/qx/tutorial?sectionId=beginners-guide&page=002-Introduction~2F001-Introduction>

iPi Soft LLC. (2014, 2015). iPi Motion Capture Brochure. Abgerufen von [ipisoft.com/pr/iPi\\_Motion\\_Capture\\_Brochure.pdf](http://ipisoft.com/pr/iPi_Motion_Capture_Brochure.pdf)

iPi Soft LLC. (2015a). User Guide for Multiple PS Eye Cameras Configuration - iPi Docs. Abgerufen 17. November 2018, von [http://docs.ipisoft.com/User\\_Guide\\_for\\_Multiple\\_PS\\_Eye\\_Cameras\\_Configuration#Glowing\\_Marker](http://docs.ipisoft.com/User_Guide_for_Multiple_PS_Eye_Cameras_Configuration#Glowing_Marker)

iPi Soft LLC. (2015b, Juli 24). iPi Soft User Guide for Multiple PS Eye Cameras Configuration. Abgerufen 17. Juni 2018, von [http://docs.ipisoft.com/User\\_Guide\\_for\\_Multiple\\_Depth\\_Sensors\\_Configuration](http://docs.ipisoft.com/User_Guide_for_Multiple_Depth_Sensors_Configuration)

iPi Soft LLC. (2017, Oktober). Cameras and Accessories - iPi Docs. Abgerufen 19. Juni 2018, von [http://docs.ipisoft.com/Cameras\\_and\\_Accessories#USB\\_controllers](http://docs.ipisoft.com/Cameras_and_Accessories#USB_controllers)

iPi Soft LLC. (2018). Video Cards Tracking Speed. Abgerufen 27. Dezember 2018, von [/store/gpu/](http://store.gpu/)

Johnston, O., & Thomas, F. (1981). *The Illusion of Life: Disney Animation*. Disney Editions.

Kersting, U., Smith, A., Brox, T., Klette, R., & Seidel, H.-P. (2005). A silhouette based human motion tracking system. The University of Auckland.

Kitagawa, M., & Windsor, B. (2008). *MoCap for Artists*. Focal Press.

Kostov, N. T., Yordanova, S. M., & Kalchev, Y. D. (2015). MoCap – The Advantages of Accelerometers and Accuracy Improvement. International Journal of Mathematics and Computers in Simulation.

Kovar, L., Gleicher, M., & Schreiner, J. (2002). Footskate Cleanup for Motion Capture. University of Wisconsin.

Kühnapfel, J. B. (2014). *Nutzwertanalysen in Marketing und Vertrieb*. Wiesbaden: Springer Gabler.

Maestri, G. (2006). *(digital) Character Animation* (3. Aufl.). New Riders.

Menache, A. (2011). *Understanding Motion Capture for Computer Animation* (2. Aufl.). Morgan Kaufmann.

Merriaux, P., Dupuis, Y., Boutteau, R., Vasseur, P., & Savatier, X. (2017). A Study of Vicon System Positioning Performance. Normandie University, Department of Multimodal Transportation Infrastructure. Abgerufen von <https://www.mdpi.com/1424-8220/17/7/1591/pdf>

Ninja Theory. (2015a). *Hellblade Development Diary 10: Capturing Performance*. Abgerufen von <https://www.youtube.com/watch?v=YBprDIzcXxw>

Ninja Theory. (2015b). *Hellblade Development Diary 18: The Shoot Set Up*. Abgerufen von <https://www.youtube.com/watch?v=FACTByOjqyQ>

Ninja Theory. (2015c). *Hellblade Development Diary 22: Combat Revisited | Part 1*. Abgerufen von <https://www.youtube.com/watch?v=xRMdJtwLins>

Noitom LTD. (2015, August 22). User Forum: 2.1 Amp USB Portable Power Supply for 32 Neuron Setup... [Text]. Abgerufen 12. Januar 2019, von <https://neuronmocap.com/forum/21-amp-usb-portable-power-supply-32-neuron-setup>

Noitom LTD. (2018a). Axis Neuron Manual.

Noitom LTD. (2018b). What are the minimum hardware requirements? Abgerufen 28. Dezember 2018, von <http://support.neuronmocap.com/hc/en-us/articles/205948897-What-are-the-minimum-hardware-requirements->

Noitom LTD. (2018c, Juni 17). Perception Neuron [Text]. Abgerufen 17. Juni 2018, von <https://neuronmocap.com/content/product/perception-neuron>

Price, A. (2018). *The Next Leap: How A.I. will change the 3D industry*. Amsterdam. Abgerufen von <https://www.youtube.com/watch?v=FlgLxSLsYWQ>

Przemyslaw, G., & Leibetseder, L. (2016). Unreal vs Unity. Technische Universität Wien. Abgerufen von [https://www.cg.tuwien.ac.at/research/publications/2016/Przemyslaw\\_Gora\\_2016\\_UVU/Przemyslaw\\_Gora\\_2016\\_UVU-thesis.pdf](https://www.cg.tuwien.ac.at/research/publications/2016/Przemyslaw_Gora_2016_UVU/Przemyslaw_Gora_2016_UVU-thesis.pdf)

Schepers, M., Guiberti, M., & Bellusci, G. (2018). Xsens MVN: Consistent Tracking of Human Motion Using Inertial Sensing. Xsens Technologies B.V.

Schmidhuber, J. (2014). *Deep learning in neural networks: An overview*. Manno-Lugano, Schweiz: Elsevier Ltd. Abgerufen von [https://ac.els-cdn.com/S0893608014002135/1-s2.0-S0893608014002135-main.pdf?\\_tid=3708713a-002e-4d8e-aa74-68b64cc53310&acdnat=1543675104\\_0e8ea5295484245a0e4692a5d22342f1](https://ac.els-cdn.com/S0893608014002135/1-s2.0-S0893608014002135-main.pdf?_tid=3708713a-002e-4d8e-aa74-68b64cc53310&acdnat=1543675104_0e8ea5295484245a0e4692a5d22342f1)

STMicroelectronics. (2018). MEMS - STMicroelectronics. Abgerufen 9. Juni 2018, von [http://www.st.com/content/st\\_com/en/about/innovation---technology/mems.html](http://www.st.com/content/st_com/en/about/innovation---technology/mems.html)

Strohm, T., & Rölver, R. (2018). Quantencomputing und andere Quantentechnologien. *Digitale Welt*, (4). <https://doi.org/10.1007/s42354-018-0133-z>

University of Wisconsin Graphics Lab. (1999). Biovision BVH. Abgerufen 6. November 2018, von <https://research.cs.wisc.edu/graphics/Courses/cs-838-1999/Jeff/BVH.html>

Vicon Motion Systems Ltd. (2016). Vicon Vantage Reference. Abgerufen von <https://www.prophysics.ch/wp-content/uploads/2017/06/ViconVantageReference.pdf>

Vicon Motion Systems Ltd. (2017a). 1 - *Shōgun Live - Introduction*. Abgerufen von <https://vimeo.com/218944959>

Vicon Motion Systems Ltd. (2017b). 6 - *Shōgun Post - Marker Editing*. Abgerufen von <https://www.youtube.com/watch?v=ZoYIdRNvWuc&t=362s>

Vicon Motion Systems Ltd. (2017c). 7 - *Shōgun Live - Capture Direct to Disk*. Abgerufen von <https://vimeo.com/218944993>

Vicon Motion Systems Ltd. (2017d). *Vicon Case Study: Vicon helps Ninja Theory battle for independence with Hellblade: Senua's Sacrifice*.

Vicon Motion Systems Ltd. (2017e). Vicon Vero Brochure. Abgerufen von <https://www.vicon.com/file/vicon/vero-family-6pp-99394.pdf>

Vicon Motion Systems Ltd. (2018a). Getting started with Vicon Shogun.

Vicon Motion Systems Ltd. (2018b). Lock Sync Box. Abgerufen von <https://www.vicon.com/file/vicon/lockstudiolab-1106web-97223.pdf>

Vicon Motion Systems Ltd. (2018c). *Vicon Siren: The History of Mocap*. Abgerufen von <https://www.engadget.com/video/5b06094447c83018ebd13cd0/>

Williams, R. (2001). *The Animator's Survival Kit*. London: Faber & Faber.

Woodman, O. J. (2007). An introduction to inertial Navigation. University of Cambridge. Abgerufen von <https://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-696.pdf>

Xsens North America Inc. (2018). Inertial sensors. Abgerufen 10. Juni 2018, von <https://www.xsens.com/tags/inertial-sensors/>

Yilmaz, A., Javed, O., & Shah, M. (2006). Object tracking: A Survey. ACM Computing Surveys 38(4):1-45.

Yirka, B. (2018). Applying deep learning to motion capture with DeepLabCut. Abgerufen 30. November 2018, von <https://phys.org/news/2018-08-deep-motion-capture-deeplabcut.html>

Zhang, H., Starke, S., Komura, T., & Saito, J. (2018). Mode-Adaptive Neural Networks for Quadruped Motion Control. University of Edinburgh, Adobe Research. Abgerufen von <http://homepages.inf.ed.ac.uk/tkomura/dog.pdf>



# Abbildungsverzeichnis

Abbildung 1. Virtueller 3D-Charakter inklusive 3D-Skelett.....	17
Abbildung 2. Passive Marker, Vicon Vero (Low-End), Vicon Vantage (High-End) .....	21
Abbildung 3. Verschiedene Formmodelle .....	24
Abbildung 4. Bild Segmentierung (li.) und Aktive Konturen (re.).....	26
Abbildung 5. Kategorien gebräuchlicher Trackingverfahren .....	27
Abbildung 6. Illustration und Funktion einer MEMS Komponente.....	33
Abbildung 7. Gimbal-stabilisierte IMU-Plattform .....	35
Abbildung 8. Mikroskopansicht eines 3D-MEMS Gyroskop (© STMicroelectronics) (li.) und IMU-Einheit mit MEMS Gyroskop und -G-Sensor (re.) .....	36
Abbildung 9. Qualitätsstufen von Inertial Sensoren (© xsens).....	37
Abbildung 10. Animationspipeline einer Echtzeit-Gesichtserkennung .....	41
Abbildung 11. Echtzeit Performance Capture für das Computerspiel Hellblade .	44
Abbildung 12. Beispiel eines Location Based VR-Systems der Firma LP- RESEARCH Inc.....	46
Abbildung 13. markerlose NCam (li.) und markerbasiertes System Optitrack Insight (re.).....	48
Abbildung 14. Retargeting Beispiel (learn.foundry.com).....	59
Abbildung 15. Basis-Setup mit 12 Vicon Vantage Kameras .....	63
Abbildung 16. Beispiel einer Markerkonfiguration .....	65
Abbildung 17. Aufnahmevervolumen und Kamerapositionen eines Vicon Setups...	66
Abbildung 18. GUI der Shogun Live- (o.) und Shogun Post Software (u.) .....	68
Abbildung 19. Time Bar Issue Map, Data Health View und Graph Editor (v.o.n.u.) .....	69
Abbildung 20. Hardware des Perception Neuron MoCap Anzug .....	72
Abbildung 21. Varianten eines Ganzkörpersetups des Perception Anzugs .....	74
Abbildung 22. Kalibrierungssequenz des Perception Anzugs .....	76

Abbildung 23. GUI der Axis Neuron Software .....	77
Abbildung 24. Feineinstellung des Skeletts sowie des Bodenkontaktes .....	79
Abbildung 25. Beispiel eines halb- und vollkreisförmigen Aufbaus .....	83
Abbildung 26. Auswertung des Kalibrierungsvideos .....	84
Abbildung 27. GUI der iPi Motion Capture Studio 4 Software .....	86
Abbildung 28. Screenshot einer Vergleichsaufnahme .....	88
Abbildung 29. Perspektiven- (li.) und Kameraansicht der 0007_Crawling Szene (re.) .....	90
Abbildung 30. Ausrichten der Kameras und Einstellen der Beleuchtung .....	92
Abbildung 31. Axis Post Processing Workflow .....	96
Abbildung 32. Vergleich des Jitter-Anteils in den Originaldaten .....	103
Abbildung 33. Gewichtung der Kriterien .....	110
Abbildung 34. Teilnutzenwertung ohne Gewichtung .....	114

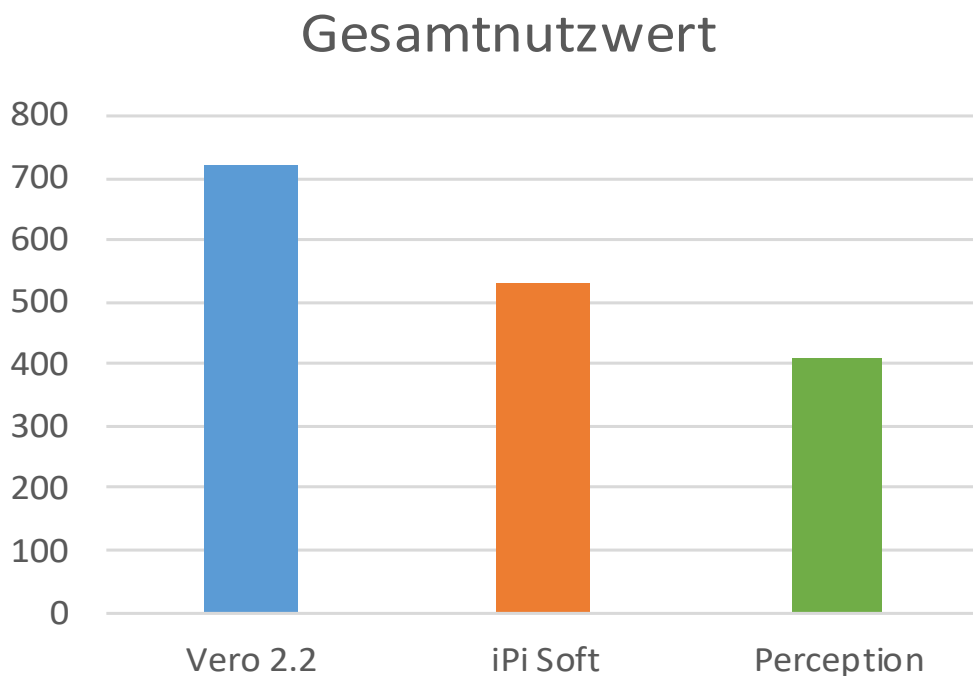


Abbildung 35. Gesamtnutzwert inklusive Gewichtung .....	114
Abbildung 36. künstliches neuronales Netzwerk .....	115
Abbildung 37. Sechs Beispiele für automatisch applizierte Marker durch DeepLabCut .....	116

## 9 Anhang

---

Abbildung 38. Trainingsbeispiel eines neuronalen Netzwerks.....	117
Abbildung 39. IBM Q Quantencomputer (li.) und Google's 72 qubit Prozessor (re.) .....	120

# Tabellenverzeichnis

Tabelle 1. Definition des Zielerfüllungsfaktors .....	111
Tabelle 2. Zuweisung des Zielerfüllungsfaktors und Ermittlung des Gesamtnutzwerts .....	112