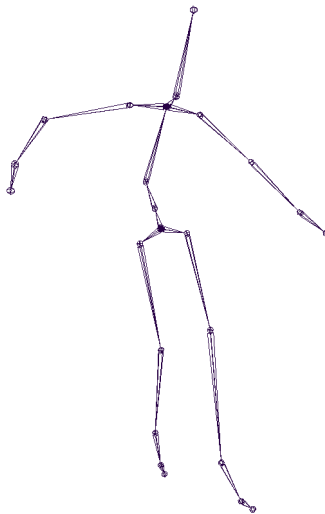


# Diplomarbeit

Fachhochschule Stuttgart  
Hochschule für Druck und Medien

## MotionLab



### Bearbeitung von Motion-Capture-Daten mittels prozeduraler Verfahren

Vorgelegt von  
Florian Linner

1. Prüfer: Prof. Dr. Thomas Keppler
2. Prüfer: Dipl. Inf. Peter Kattwinkel

# Inhaltsverzeichnis

1 Zusammenfassung .....	1
2 Einleitung .....	3
3 Unternehmen .....	5
3.1 I-D Media AG .....	5
3.2 I-D TV .....	6
4 Maya .....	8
4.1 Alias Wavefront .....	8
4.2 Die Software .....	9
4.2.1 Benutzerinterface .....	10
4.2.2 Modeling .....	12
4.2.3 Dynamics .....	12
4.2.4 Animation .....	14
4.2.5 Rendering .....	15
4.3 MEL .....	16
4.3.1 Allgemeines .....	16
4.3.2 Syntax, Datentypen, Operatoren und Flowcontrol .....	17
5 Projektbeschreibung .....	19
5.1 Allgemeines .....	19
5.1.1 Gestik und Körpersprache .....	19
5.2 Motion-Capturing .....	20
5.2.1 Einleitung .....	20
5.2.2 Die ersten Einsätze von Motion-Capture .....	21
5.2.3 Technischer Hintergrund .....	22

5.2.4 Wie unterscheidet sich Motion Capturing von synthetischer Animation? .....	23
5.2.5 Optische Motion-Capture Systeme .....	24
5.2.6 Magnetische Motion Capture-Systeme .....	28
5.2.7 Elektromechanisches Motion-Capture .....	31
5.2.8 Zum Verständnis des teilweise schlechten Rufs von Motion Capture .....	32
5.3 MotionLab .....	33
5.4 Möglichkeiten .....	33
5.5 Ziele .....	34
5.6 Aufbau .....	34
5.6.1 Motion Equalizer .....	35
5.6.2 Cycle .....	41
5.6.3 Edit .....	44
5.6.4 Blend-Tool .....	44
5.6.5 Speed .....	46
6 Implementierung .....	47
6.1 Allgemeines .....	47
6.2 Scripts und Prozeduren von MotionLab .....	48
6.2.1 SourceMotionLab.mel .....	48
6.2.2 MotionEqualizer.mel .....	48
6.2.3 Blend.mel .....	52
6.2.4 Cycle.mel .....	53
6.2.5 Speed.mel .....	55
6.2.6 Edit.mel .....	55
6.3 Zusammenfassung .....	57
7 Anwendungsbeschreibung .....	58
7.1 MotionEqualizer .....	58

7.2 Edit .....	59
7.3 Blend .....	61
7.4 Speed .....	63
7.5 Cycle .....	65
8 Fazit .....	66
8.1 Ergebnis .....	66
8.2 Ausblick .....	67
9 Anhang .....	68
9.1 Begriffsdefinition .....	68
9.2 Maya Lizenzen .....	71
9.3 MEL-Script .....	73
9.4 Filmverzeichnis .....	77
9.5 Abbildungsverzeichnis .....	78
9.6 Literaturverzeichnis .....	80

# 1 Zusammenfassung

Computergrafik wird ein zunehmend wichtigeres Hilfsmittel, wenn es um die glaubhafte Realisierung von Lebewesen in den Audiovisuellen Medien geht. Im Computer erzeugte Charaktere agieren als Moderatoren, als menschliches Interface in Applikationen, oder als Darsteller in Werbung, Fernseh- und Kinofilmen.

Ein sehr wichtiger Faktor, der zur Glaubwürdigkeit und Natürlichkeit dieser Charaktere beiträgt, ist die Gestik. Durch sie werden Nachrichten zum gesprochenen Wort kommuniziert, wodurch die Worte unterstrichen oder widerlegt werden können.

Gegenstand dieser Arbeit ist die Entwicklung eines Werkzeuges zur Bearbeitung, Veränderung und Bereinigung von Motion-Capture-Daten im Rahmen der Software Maya und der Scriptsprache MEL.

Diese Arbeit entstand im Unternehmensbereich ID-TV der ID Media AG. Dieser ist, neben den Bereichen Video-/Postproduktion und Visualisierung, auch im Bereich Charakteranimation tätig.

Nach einer kurzen Einführung in die Leistungsfähigkeit der Software Maya und Beschreibung der Skriptsprache MEL, wird in der Projektbeschreibung auf die unterschiedlichen Voraussetzungen für ein solches Werkzeug eingegangen und ein Konzept ausgearbeitet.

Das entwickelte Werkzeug soll ein Arbeiten auf einem höheren Level, als es die bisherige Maya-Funktionalität bietet, aufbauend auf Motion-Capture-Daten ermöglichen. Die Motion-Capturing-Technologie dient als Grundlage für das Werkzeug. Auf sie wird später eingegangen.

Im anschließenden Kapitel wird die Implementation eines Prototyp beschrieben und diskutiert. Nach einer Anwendungsbeschreibung des gegenwärtigen Stands des Motion-Editing-Tools erfolgt eine Beurteilung

der Arbeit und ein Ausblick auf eine eventuelle Weiterentwicklung.

## 2 Einleitung

3D-Computergrafik nimmt heutzutage in vielen Bereichen einen immer höheren Stellenwert ein. Ob in der Automobilindustrie, wo Karosserien billiger im Computer entworfen und getestet werden können, im Produktdesign im Allgemeinen, der Medizin oder der Architekturvisualisierung. Eine sehr wichtige Rolle spielt 3D-Computergrafik auch in den Medien. Im Internet, in Computerspielen, Fernseh- und Kinofilmen begegnen wir immer öfter computergenerierten Spezialeffekten, künstlichen dreidimensionalen Welten und 3D-Charakteren.

Dabei entwickelt sich die Technologie immer schneller und somit auch die Anforderungen an die Qualität der Produkte. Immer mehr Realismus wird gefordert. Viele der heutigen Kinofilme leben zu einem großen Teil von ihren digitalen Effekten. „Jurassic Park“ ist durch seine Dinosaurier zu einem Meilenstein in der Geschichte der Computeranimation und des Filmtricks geworden und hat der Computeranimation im Kino zu einem regelrechten Boom verholfen.

Die ersten Animationsprogramme, die Mitte der Neunziger auf den Markt kamen, erforderten ein hohes technisches Know How und konnten wenig Bedienungskomfort bieten. Die Designer und Animatoren forderten jedoch eine einfache Handhabung, die möglichst auch ihrem bisherigen Arbeitsablauf ähnelte.

Inzwischen sind die Softwarehersteller diesen Forderungen nachgekommen. 1995 wurde erstmals die 3D-Software Maya auf der SIGGRAPH von Alias|Wavefront als revolutionäre Software angekündigt, die neue Maßstäbe in der Produktivität, der Bedienung sowie der Erweiterbarkeit und neuer Simulations- und Darstellungstechnologien setzt.

In allen Bereichen der Unterhaltungsindustrie sind Charaktere in welcher Form auch immer zu finden. Mal ist es ein sprechendes WC, Aliens, die uns erschrecken oder die Heldin eines Computerspiels, mit der wir viele Abenteuer gemeinsam erleben. Was diese Charaktere ausmacht, sind ihre Emotionen, die sie vermitteln. Auch wenn der Charakter die Form einer Shampooflasche hat, so bekommt er durch die Art, wie er sich bewegt Charakter und ist so in der Lage, uns in seinen Bann zu ziehen, ihn für uns abstoßend oder sympathisch wirken zu lassen. Das Phänomen Lara

Croft ist ein gutes Beispiel dafür, wie ein virtueller Charakter die Menschen begeistern kann, sogar zu einem Idol werden kann, und weltweit Berühmtheit erlangen kann.

Eine der Stärken der Software Maya liegt ohne Zweifel auf dem Gebiet der Charakteranimation, und Maya bietet mit seiner Programmiersprache MEL vielfältige Möglichkeiten zur individuellen Erweiterung der Funktionalität.

Gerade bei Computerspielen, vor allem solchen, die sich mit menschlichen Bewegungsabläufen beschäftigen, wie Sportsimulationen, ist eine glaubwürdige Darstellung menschlicher Bewegungen äußerst wichtig.

Mittels Motion-Capturing ist eine wirklichkeitsgetreue Wiedergabe natürlicher Bewegungsabläufe möglich. Leider ist eine Nachbearbeitung der erhaltenen Bewegungsdaten nicht oder nur unter sehr großem Aufwand möglich.

Im Rahmen dieser Diplomarbeit soll, anhand der von MEL gebotenen Möglichkeiten, ein Werkzeug konzipiert und entwickelt werden, das dem Anwender die Möglichkeit gibt, Motion-Capture-Daten beliebiger Form zu bearbeiten, zu bereinigen und nach eigenen Wünschen zu verändern, ohne dabei den natürlichen Charakter der Bewegungen zu verlieren.

Hierzu wird zunächst näher auf das Thema Motion-Capturing eingegangen, werden die Vor- und Nachteile der einzelnen Systeme beleuchtet. Desweiteren werden menschliche Bewegungsmuster analysiert und nach diesen Erkenntnissen wird ein Konzept für das Motion-Editing-Tool ausgearbeitet. Darauf folgt die Beschreibung des in MEL implementierten Prototyps.



## 3 Unternehmen

### 3.1 I-D Media AG

Bereits 1988 gründete der geschäftsführende Gesellschafter Bernd Kolb die beiden Firmen I-D Werbeagentur GmbH und I-D Media GmbH, die heute unter dem Dach der I-D Gruppe zusammengefaßt sind. I-D steht hierbei für „Innovativ Digital“. 1993 wurde ein Multimedia-Produktionsstudio eingerichtet und ein internationales Team nach Deutschland geholt, die ihr Know How im Umgang mit Screendesign, CD-ROM-Produktion und Online-Programmierung einbrachten. Die Firma entwickelt und vermarktet neue Produkte für innovative und digitale Medienformate, die vom enormen Wachstumspotential des Internets profitieren.

Im Frühjahr 1999 ging das Unternehmen mit 180 festangestellten Mitarbeitern, mit Hauptsitz in Essingen im Bereich Neue Märkte erfolgreich an die Börse. Aufsichtsratsvorsitzender der somit entstandenen I-D Media AG ist Lothar Späth. Inzwischen hat sich die Zahl der Mitarbeiter auf über 400 erhöht. Der Hauptsitz wurde von Essingen nach Berlin verlegt. Außer in Berlin ist die I-D Media AG noch in Potsdam, Hamburg, Stuttgart, London und Singapur vertreten.

Die I-D Media AG ist eine der größten unabhängigen Multimedia-Agenturen Europas. In der von „W+V“ (Werben & Verkaufen) veröffentlichten Rangliste der größten deutschen Multimedia-Agenturen belegte die I-D Media AG, mit einem Umsatz von 15 Millionen Mark im Geschäftsjahr 1999, den dritten Platz.

Das Unternehmen ist in vier Geschäftsbereiche gegliedert:

- Dienste und Formate
- Software
- Agentur
- I-D TV

Der Geschäftsbereich Dienste und Formate pflegt vor allem den Cycosmos [[www.cycosmos.com](http://www.cycosmos.com)], das bekannteste Projekt der I-D-Media AG. Es handelt sich hierbei um eine digitale Community im Internet. Hier können die Mitglieder sich einen Avatar mit ihren Hobbies, Vorlieben oder Interessen erschaffen und mit diesem andere Avatare im Cycosmos, die eventuell die gleichen Interessen haben in kürzester Zeit ausfindig machen und mit ihnen kommunizieren.

Im Geschäftsbereich Software wird zur Zeit das Software-Tool „Living Screen“ entwickelt. PC-Benutzer erhalten mit dieser Software neueste Nachrichten und Informationen, sobald dieser Online geht. Hierzu kann sich der Benutzer in entsprechende Newsgroups eintragen, die für ihn relevante Informationen behandeln. Living Screen ermöglicht Unternehmen, Benutzer direkt und ohne Umweg über die Unternehmens-Webseite zu erreichen.

Die Agentur ist der ursprüngliche Geschäftsbereich der I-D Gruppe. Seit 1988 erarbeite das I-D Team innovative Marketingkonzepte und Designs für Marken wie Chevignon, Lee, oder Mustang. Hier werden Webseiten konzipiert, entworfen und realisiert. Zahlreiche Auszeichnungen sprechen für den Erfolg dieses Geschäftsbereiches.

I-D TV (interactive digital television) ist ein Produktionshaus in den Bereichen Computeranimation, Video-/Film- und Postproduktion, sowie Softwareentwicklung und interaktives Fernsehen.

## 3.2 I-D TV

I-D TV als eine Tochter von I-D Media existiert in der heutigen Form seit 1997. Sie entstand aus dem früheren I-D-3D-Lab, einem Full-Service rund um computergenerierte Graphiken und Animationen. I-D TV ist ein eigenständiges Produktionshaus in den Bereichen Computeranimation, Video/Film- Postproduktion. Mit 45 Mitarbeitern gehört I-D TV heute zu den großen digitalen Animationshäusern Deutschlands.

I-D TV besteht aus den vier Einheiten:

- Art Department
- Digital Post
- Animation Studios
- Visual Devedge

Das Art Department beschäftigt sich mit allem, was in Verbindung steht mit Gestaltung und Umsetzung von Entwürfen im zweidimensionalen Bereich. Hier werden Storyboards gezeichnet, Charakter am Zeichentisch entworfen und Bildschirmwürfe gestaltet. Es werden hier aber auch Texturen gezeichnet und Typografien entworfen.

In der Digital Post stehen zwei Postproduktionsplätze vom Typ Softimage Digital Suite zur Verfügung. Auf ihnen werden Animationen, Filme und Videos geschnitten, digital bearbeitet und können dann auf die Formate VHS, SVHS, DV, miniDV, Beta SP, Digi Beta und MPEGII ausgespielt werden.

In den Animationstudios werden 3D-Animationen realisiert. Die Abteilung ist noch einmal aufgeteilt: Der Creatureshop, in dem Charakter entworfen und animiert werden, ein Motion-Capture-Department, in dem ein optisches 10-Kamera-System der Firma Motion Analysis zur Verfügung steht und der Bereich Visualisierung. Gearbeitet wird auf SGI und Intergraph NT-Workstations mit den Programmen MAYA, Lightwave, Softimage 3D, Studio Paint und PowerAnimator.

Die Visual Devedge hat die Aufgabe, Tools zu entwickeln, die ein einfacheres und effizienteres Arbeiten im 3D- und im Videobereich ermöglicht. So entstand hier zum Beispiel „Crowds“, ein Tool, mit dem sich Massenszenen generieren lassen. Entwickelt wird unter C/C++, MEL, Pascal und Visual Basic.

# 4 Maya

## 4.1 Alias|Wavefront

Alias|Wavefront ist einer der weltweit führenden Anbieter von Software für Computergrafik. Die Firma entwickelt Software für die Branchen Film und Video, Spiele und interaktive Medien sowie für Industriedesign und Architektur.

Die Entstehung des Unternehmens in seiner heutigen Form geht auf das Jahr 1984 zurück. Damals brachte die Firma Wavefront aus Santa Barbara das erste kommerzielle 3D-Softwarepaket auf den Markt. Im selben Jahr wird in Paris die Firma Thomas Digital Image (TDI) gegründet, welche von Wavefront im Jahre 1993 aufgekauft wird. 1995 werden Wavefront und Alias aus Toronto von Silicon Graphics, Inc. (SGI) übernommen und zur heutigen Firma Alias|Wavefront zusammengeschlossen.

Alias|Wavefront ist nun eine eigenständige, hundertprozentige Tochtergesellschaft von Silicon Graphics, Inc. mit Sitz in Toronto, Kanada. Das Unternehmen betreibt Forschungs- und Entwicklungsabteilungen in Toronto, Mountain View, Vancouver, Santa Barbara, Seattle und Paris.

Von Alias werden für die Bereiche Design und Entertainment Produkte angeboten. Für den Designbereich steht mit der Studio 9-Familie, bestehend aus StudioPaint, AutoStudio und SurfaceStudio, ein Softwarepaket für Industrie- und Automobildesign sowie zum Gestalten technischer Oberflächen zur Verfügung.

Im Bereich Entertainment bietet Alias im 2D-Compositing-Bereich Maya Composer und Maya Fusion, im 3D-Bereich Explore, PowerAnimator und Maya. Vom PowerAnimator existiert eine spezielle Version zur Entwicklung von Spielen, und auch Maya dringt mittlerweile vehement in diesen Markt vor. [AW]

## 4.2 Die Software

Maya ist Alias|Wavefront's professionelles Animationstool zur Generierung dreidimensionaler Szenen. Maya ist für das SGI-eigene Betriebssystem Irix und Windows NT erhältlich. Die Software ist speziell für Charakteranimation sehr geeignet. Beispiele der Leistungsfähigkeit kann man in großen Kinoproduktionen wie Matrix, Die Mumie, Stuart Little, A Bug's Life oder Titanic sehen. Aber auch im Spielesektor hat sich Maya inzwischen etabliert.

Maya verbindet erstmals die Module Modeling, Animation, Dynamics, Rendering, die in früheren Software-Paketen wie Explore getrennt waren, unter einer Oberfläche, sodaß die verschiedenen Arbeitsschritte bis zur fertigen Animation innerhalb eines Programms erfolgen können. Trotzdem bleibt die Software in diese Modi aufgegliedert, was für bessere Übersichtlichkeit und direkten Zugriff auf die innerhalb der Arbeitsschritte benötigten Funktionen sorgt.

Die Software ist inzwischen bei der Version 2.5 angekommen. Dabei sind weitere Komponenten wie PaintEffects und Cloth dazugekommen. Sie werden ebenfalls als weitere Module in die Oberfläche eingebunden, sodaß direkt darauf zugegriffen werden kann. Durch diesen modularen Aufbau bleibt Maya offen für weitere Entwicklungen, ohne daß diese große Veränderungen im Interface erfordern würden.

Die Systemarchitektur von Maya ist sehr offen gehalten. Es wird mit abstrakten Objekten, sogenannten Nodes gearbeitet. Es existieren Geometrie-Nodes, Nodes für Animationskurven, Lichter, Texturen, Shader oder Expressions. Über die C++-API-Schnittstelle können neue Nodes, sowie Plugins programmiert werden. Desweiteren verfügt Maya über die Scriptsprache MEL. Mit ihr lassen sich die meisten Aufgaben lösen, aber um beispielsweise einen speziellen Shader für den Maya-Renderer zu programmieren, ist MEL als Scriptsprache zu langsam. Außerdem läßt sich die Oberfläche beliebig erweitern und den persönlichen Bedürfnissen anpassen.

Momentan ist Maya in zwei verschiedenen Lizenzversionen erhältlich:

Maya Complete

Maya Unlimited

Im Folgenden soll hier grob auf die vier Hauptmodule Modeling, Animation, Dynamics und Rendering, sowie auf das Benutzerinterface eingegangen werden, um einen Überblick über den Aufbau und die Leistungsfähigkeit des Programms zu bekommen. Hierbei wurde auf die Quellen [Maya W] und [Knappe] zurückgegriffen.

#### 4.2.1 Benutzerinterface

Das Benutzerinterface von Maya ist sehr flexibel und kann beliebig erweitert und den individuellen Bedürfnissen angepaßt werden. So können Elemente, die momentan nicht gebraucht werden, ausgeblendet werden, sodaß der ganze Bildschirm für Ansichtsfenster zur Verfügung steht. Die Elemente können in einzelnen Fenstern aufgerufen werden, oder sie können ins Layout integriert werden. Das Layout des Hauptfensters kann ohne Einschränkungen variiert und die einzelnen Elemente beliebig platziert werden.

Sind die Elemente ausgeblendet, lassen sich trotzdem alle Funktionen über die Hotbox, die bei längerem Drücken der Leerzeilentaste erscheint, aufrufen. Diese Hotbox läßt sich wie das gesamte Benutzerinterface individuell gestalten und den jeweiligen Bedürfnissen anpassen. Natürlich sind auch die meisten Tasten mit Funktionen belegt (Hotkeys). Diese Belegung läßt sich individuell verändern, sodaß ein sehr zügiges Arbeiten möglich ist

In der Standardeinstellung enthält das Hauptfenster zunächst oben die Hauptmenüleiste, die alle Funktionen des jeweils aktiven Moduls enthält. Darunter befindet sich die Stausline. In ihr kann zwischen den einzelnen Modulen hin- und hergeschaltet werden. Außerdem enthält sie ein Auswahltool, mit dem man die Auswahl in den Ansichtsfenstern einschränken kann, sodaß in komplexen Szenen nur bestimmte Objekttypen auf das Auswahlwerkzeug reagieren.

Im zentralen Bildschirmbereich befinden sich die Ansichtsfenster (Panels), in denen die Szene in Echtzeit dargestellt wird. Auch sie lassen sich beliebig anordnen und konfigurieren. Es sind orthographische und perspektivische Projektionen möglich, sowie die Projektion aus der Sicht einer Kamera oder auch einer Lichtquelle.

Die Ansichtsfenster können ebenso Editoren und Kontrollfenster jeder Art enthalten.

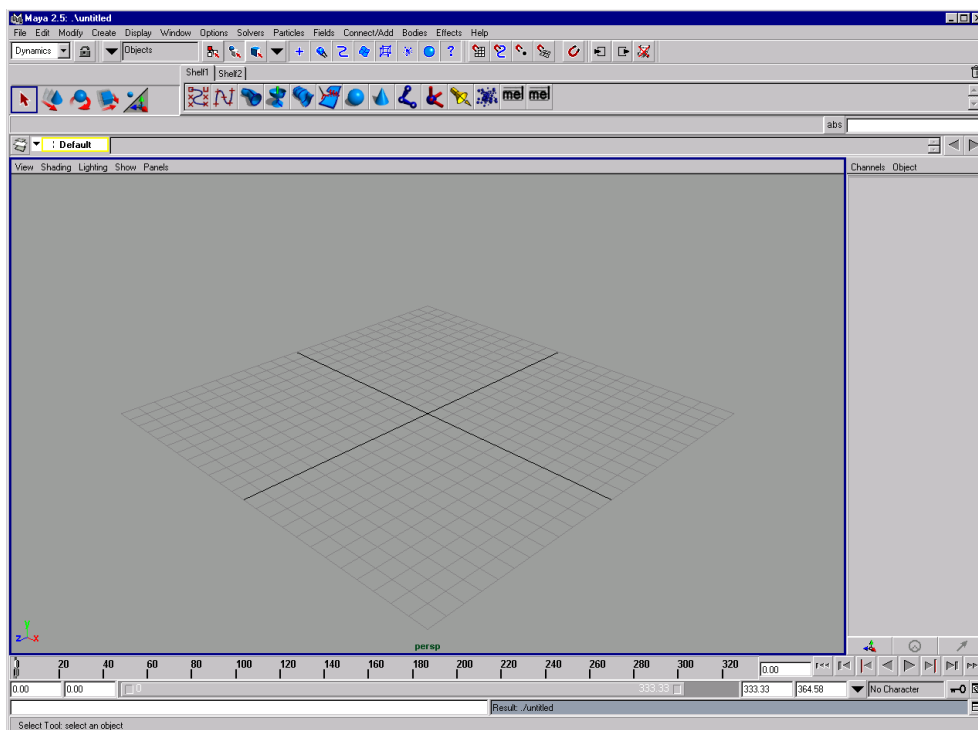


Abb 1: Benutzerinterface

Unterhalb der Ansichtsfenster befinden sich Range- und Timeslider, mit denen man Kontrolle über die Zeit hat. Darunter schließlich die Feedback- und Commandlines. Durch die Commandline kann der Anwender MEL-Befehle eingeben, ohne den platzeinnehmenden Script-Editor öffnen zu müssen. Durch die Feedbackline erhält der Anwender Nachrichten aus dem System.

Maya unterstützt auch Eingabegeräte wie Maus, Tablett, MIDI-Geräte etc.

### 4.2.2 Modeling

In Maya ist es möglich zwei Arten von Geometrien zu erstellen. NURBS- oder Polygon -Geometrien.

Beim NURBS-Modeling dienen als Ausgangsobjekte meist Splinekurven. Diese können mit verschiedenen Werkzeugen zu Flächen weiterverarbeitet werden. Sie können auch als Animationspfade oder eigenständige Objekte Verwendung finden.

Außerdem können auch Primitiven wie Kreis, Zylinder, Kegel, Ebene, Würfel, Kugel als Ausgangsobjekte verwendet werden und dann mit den vielen teilweise vom PowerAnimator bekannten Translations- und Deformationswerkzeugen bearbeitet und in die gewünschte Form gebracht werden.

Für das Polygonmodeling können die gleichen Werkzeuge benutzt werden, es stehen jedoch auch noch spezielle Polygonwerkzeuge zur Verfügung. NURBS-Flächen können auch in Polygone umgewandelt werden.

Die Objekte in Maya erhalten bei ihrer Generierung eine History, in der alle Bearbeitungsschritte gespeichert werden. So kann nachträglich ein Faktor einer schon früher stattgefundenen Operation verändert werden. Diese Veränderung wirkt sich dann durch die gesamte History auf alle nachfolgenden Operationen aus. Ein aus einer Kurve entstandenes Loft-Objekt kann also verändert werden, indem man die Kurve, aus der es entstanden ist, verändert. Die Construction History kann auch jederzeit gelöscht werden, da sie viel Speicher in Anspruch nimmt, und die Operationen aufwendiger berechnet werden müssen.

### 4.2.3 Dynamics

Physikalische Bewegungsabläufe, die auf Trägheit, Trägheitsmoment, Haft-/ Gleit-Reibung, Dämpfung, Elastizität etc. Rücksicht nehmen, lassen sich durch Keyframe-Animation nur unter großem Aufwand glaubwürdig realisieren. Das menschliche Auge nimmt solche Bewegungen tagtäglich wahr und erkennt daher unnatürliche Bewegungen sofort. Interagieren mehrere Objekte miteinander, oder soll gar ein Partikelsystem einen Festkörper in Bewegung versetzen oder ein Soft-Object teilelastisch



deformieren, ist es für den Animator nahezu unmöglich, all diese Faktoren zu berücksichtigen.

Mit Dynamics erhält der Animator ein Dynamik-System, das Naturkräfte wie Gravitation, Wind und physikalische Abläufe, die daraus resultieren innerhalb von Maya komplett simuliert. Alle Parameter sind frei einstellbar, die Berechnung der Bewegungsabläufe und Interaktionen zwischen Objekten, basierend auf physikalischen Gesetzmäßigkeiten erfolgt, abhängig von der verwendeten Hardware, in Echtzeit. Es können von den errechneten Bewegungen auch Keyframes und Bewegungspfade erzeugt werden, die dann wiederum editiert werden können, um zum Beispiel Bewegungen zu akzentuieren.

Dynamics läßt sich in drei unterschiedliche Systeme einteilen:

Das Dynamiksystem für weiche Körper (Softbodies). Jedes 2D- oder 3D-Objekt kann als deformierbarer Softbody definiert werden, um Haare, Muskeln oder Fett zu simulieren. Wird ein Softbody erzeugt, werden auf dessen Oberfläche Partikel aufgebracht zwischen denen eine elastische Verbindung besteht. Die Anzahl bzw. Reichweite dieser Verbindungen, ihre Elastizität und Dämpfung legen das Verhalten fest. Zum Beispiel Haare, die auf einem Kopf sitzen, der geschüttelt wird.

Das Dynamiksystem für Festkörper (Rigid Bodies) mit einer Reihe Constraints wie „Pin“, „Nail“, „Hinge“, „Spring“ oder „Barrier“, welche die gegenseitigen Einflüsse von Objekten definieren. Der Constraint vom Typ Nail bewirkt z.B., daß sich ein Objekt zum anderen so verhält, als ob es an einer Schnur an diesem hängen würde.

Ein Partikelsystem. Mit Partikeln lassen sich realistische atmosphärische und pyrotechnische Effekte wie Rauch, Feuer, Regen oder Explosionen erzeugen. Die Partikel können von einem Emitter generiert werden, welcher an Objekte gebunden werden kann, die über das Dynamiksystem animiert werden. Außerdem können sie auch mit dem Partikel Tool erzeugt werden. Partikel können durch viele Parameter beeinflusst werden, wie Position, Geschwindigkeit, Beschleunigung oder Lebensdauer. Sie können auf verschiedene Arten dargestellt werden, wie z.B. Punkte, Striche, Kugeln oder Blobs. Außerdem kann jede beliebige Geometrie an einen Partikel gehängt werden.

Alle Elemente des Dynamiksystems können physikalisch miteinander interagieren. Die Beziehungen zwischen den Elementen können im Relationship-Editor festgelegt werden.

#### 4.2.4 Animation

Bei der Animation kommt neben den 3 Dimensionen des Raumes noch die vierte Dimension, die Zeit, ins Spiel. Nahezu alle Parameter bis hin zur Farbe der Fenster können in Maya über die Zeit verändert, animiert werden. Das funktioniert nach dem Prinzip der Keyframe-Animation, die wiederum auf der traditionellen Animation beruht. Dabei werden an einem markanten Bewegungszustand Schlüsselbilder gesetzt.

Bei der traditionellen Animation werden die Bilder zwischen den Keyframes gezeichnet, in der Computeranimation übernimmt diese Aufgabe der Computer. Der Animator legt also zu verschiedenen Zeitpunkten verschiedene Parametereinstellungen fest. Die Zustände dazwischen werden vom Computer durch Interpolation errechnet. Dabei liegen diese Werte auf Spline-Kurven, die im Graph-Editor bearbeitet werden können.

Die Kontrollelemente für das Abspielen der Animation sind in erster Linie der Time- und Rangeslider. Im Timeslider können Keys gesetzt, markiert und editiert werden. Außerdem können Sounddateien in ihm angezeigt werden.

Ein weiterer wichtiger Editor ist neben dem Graph-Editor der Dopesheet-Editor. Während im Grapheditor die Zeit, sowie die Werte einer Animation beeinflusst werden können, können im Dopesheet-Editor die Keys nur über die Zeit dargestellt und editiert werden. Das macht ihn für diesen Zweck weitaus übersichtlicher als den Graph-Editor.

Für die Animation eines Charakters kann mit dem Joint-Tool ein Skelett erstellt werden, über das die Geometrie animiert wird. Das Skelett kann traditionell durch Vorwärtskinematik animiert werden, es stehen in Maya jedoch auch Solver für inverse Kinematik zur Verfügung. Über unterschiedliche Constraints können Bewegungen und Einflußnahmen von Objekten zueinander definiert werden. Beziehungen zwischen Attributen von Objekten werden über Expressions oder set-Driven Keys erzeugt. Wird dabei eins der Attribute verändert, wird das mit ihm in Beziehung

stehende Attribut automatisch entsprechend verändert.

#### 4.2.5 Rendering

Das Rendering ist der letzte Arbeitsschritt in der Produktionspipeline einer Computeranimation. Jetzt werden den Objekten Materialien mit bestimmten Materialeigenschaften zugewiesen, Lichter werden gesetzt. Dann kann die Berechnung der Bilder beginnen

Alle Objekte erhalten ein Material, das bestimmte Eigenschaften aufweist, durch die ein bestimmtes Aussehen erzielt wird. Hierfür wird in Maya ein Shading-Netzwerk aufgebaut, das aus vielen Faktoren bestehen kann. Material-Nodes bestimmen die Art des Materials (Surface-, Volumetric-, Displacement- Materials). Maya bietet verschiedene Beleuchtungsmodelle (Phong, Phong E, Blinn, Lambert), die im Surface-Node festgelegt werden, Texturen werden über Texture-Nodes definiert. Es gibt 2D-, 3D- und Environment-Texturen. Unter Utilities sind unterschiedliche Nodes für die Oberflächengestaltung wie Bump, Place Texture, Projection etc. realisiert.

Als Lichtquellen stehen dem Benutzer Ambient (Umgebungslicht), Point (punktförmige Lichtquelle) und Spot (vergleichbar mit einem Spotlicht) zur Verfügung.

Alle diese Nodes werden bei ihrer Erzeugung standardmäßig verknüpft, lassen sich jedoch im Connection-Editor auch per Hand verknüpfen. Auf diese Art lassen sich sehr komplexe Shading-Netzwerke erzeugen.

Maya bietet dem Benutzer einen Hardware- und einen Software-Renderer. Der Hardware-Renderer erlaubt dem Benutzer das interaktive Bearbeiten der Szene im schattierten Modus. Texturen und Lichtquellen werden dabei von der Hardware unterstützt. Bei SGI-Maschinen sind das spezielle Grafikkarten mit großem Cache. Jedoch ist eine Open-GL-Darstellung von Szenen bei genügend Grafik-RAM auch auf Personalcomputern möglich.

Das eigentliche Bild und Endprodukt einer Animation wird mit dem Software-Renderer in der geforderten Qualität gerendert. Der in Maya integrierte Renderer liefert hervorragende Bilder auch in komplexen Szenen.

Bei Partikeln berücksichtigt er den gegenseitigen Schattenwurf, Blendenflecken können nachträglich simuliert und Lichter sichtbar gemacht werden. Der Renderer unterstützt objektbezogenes Raytracing. Nur wenn notwendig, wird der Raytrace-Algorithmus aufgerufen, um Spiegelungen, Brechungen oder Raytrace-Schatten zu berechnen. Weitere Funktionen sind 3D Motionblur, Antialiasing und weiche, realistische Schatten.

Die Berechnung der Bilder erfolgt in Form von Rechtecken. Dies ermöglicht ein intelligentes Speichermanagement und erfordert nicht die gesamte Szene zur Berechnung im Speicher. Der Renderer ermöglicht zudem Batchrendering, d.h. Maya muß nicht gestartet sein während des Renderns, und somit steht mehr Speicher für die Berechnung der Szene zur Verfügung

## 4.3 MEL

### 4.3.1 Allgemeines

Mayas Scriptsprache MEL bildet die Grundlage für alle Operationen in Maya. MEL setzt sich aus Commands (Befehlen), Funktionen und einer der Programmiersprache C ähnlichen, Syntax zusammen. Mit über 600 Commands und weiteren 75 Funktionen deckt MEL die Funktionalität von Maya weitestgehend ab. Sie rufen entsprechende Operationen des Programms auf und bilden die Schnittstelle zwischen Userinterface und System. Man unterscheidet vier Gruppen:

Scene Commands

Administration Commands

User-Interface Creation Commands

Functions.

Unter Scene Commands sind mit 500 Befehlen sämtliche Befehle zusammengefaßt, die direkt auf die Szene Einfluß nehmen. Mit ihnen können sämtliche Operationen durchgeführt werden, die Maya bietet. Es können Objekte erzeugt, modifiziert, deformiert oder animiert werden. Alle Operationen, die über das Benutzerinterface aufgerufen werden, werden intern über MEL aufgerufen.

Über die Administration Commands kann auf das Betriebssystem zugegriffen werden. Mit ihnen wird die Dateistruktur verwaltet. Diese Commands werden nochmals in File Input/output-, General- und System-Befehle unterteilt.

Die User-Interface Creation Commands enthalten Befehle, die die Benutzeroberfläche betreffen. Mit ihnen lassen sich alle Elemente des Maya Benutzerinterface erzeugen, bestehende Layouts verändern, Fenster erzeugen etc. Das Benutzerinterface von Maya ist zu einem großen Teil auch mittels MEL aufgebaut.

Functions umfaßt Befehle zum Erzeugen und Editieren von Arrays und Kugen, sowie eine Anzahl von mathematischen Funktionen. Hier findet man auch sämtliche String-Funktionen zum Bearbeiten von Zeichenketten.

Maya bietet mehrere Möglichkeiten MEL-Commands einzugeben und auszuführen. Zur Eingabe sind der Scripteditor, der Expressioneditor und die Commandline die gebräuchlichsten Werkzeuge. Ausführen lassen sich Commands über Scriptfiles, Shelficons, Hotkeys, Menubuttons oder Expressions.

#### 4.3.2 Syntax, Datentypen, Operatoren und Flowcontrol

Im wesentlichen findet sich in MEL in eingeschränkter Form die Syntax der Programmiersprache C wieder. MEL kennt jedoch keine Zeiger und nur fünf definierte Datentypen.

Das Erstellen eines MEL-Scripts oder einer Prozedur funktioniert grundsätzlich nach den gleichen Prinzipien wie in jeder anderen Programmiersprache. Der erstellte Code sollte übersichtlich organisiert und ausreichend dokumentiert sein, so daß dieser nachvollziehbar ist. Hierzu

stehen dieselben Instrumente wie in den gängigen Programmiersprachen zur Verfügung. Zeilenweise Kommentare werden in Mel mit `'//'` erzeugt. Ein Kommentarblock darf keinen Zeilenkommentar enthalten.

Aufeinander folgende Commands eines Scripts müssen in keinem direktem Zusammenhang stehen. Jeder Command muß aber durch ein Semikolon abgeschlossen werden. Sind Commands nicht in einer Prozedur zusammengefaßt, können diese nur explizit im Scripteditor, der Commandline oder im Expressioneditor ausgeführt werden. Eine Prozedur wird im allgemeinen wie folgt definiert:

```
[global] proc [return-type] <procedure name> ([arguments])
{
    [MEL Statements]
}
```

Einer Prozedur können Argumente übergeben werden und sie kann auch einen Rückgabewert besitzen, der - gleich wie in C - mit dem Ausdruck `'return'` übergeben wird.

MEL verfügt über die fünf Datentypen Integer, Float, String, Vector und Matrix. Jeder Variablen muß ein `$`-Zeichen vorgesetzt werden, um diese als solche zu identifizieren. Ausgenommen des Datentyps Matrix können auch Arrays gebildet werden. Diese sind in der Handhabung gleich den Arrays in C. Bei einer impliziten Deklaration von Array belegt Maya automatisch den Speicher für 16 Variablen des entsprechenden Typs. Zur besseren Speichernutzung sollte, speziell bei globalen Arrays eventuell eine explizite Deklaration in Betracht gezogen werden. Grundsätzlich sind bei Variablen- wie Prozedur-Namen die Konventionen von C einzuhalten.

Variablendeklaration: `<Datentyp> $ <Variablenname>;`

Arraydeklaration: `<Datentyp> $ <Variablenname>[]; (implizit)`

Desweiteren stehen eine Anzahl von arithmetischen, logischen und relationalen Operatoren zur Verfügung. Soweit vorhanden, haben sie die gleiche Präzedenz und Assoziativität wie die Operatoren in C. Ebenfalls wurden die Möglichkeiten zur Flowcontrol von C übernommen. So stehen zur Fallunterscheidung `if-then-else`-Anweisungen, `switch`-Anweisungen und die bedingte Bewertung zur Verfügung sowie die drei Schleifen-Typen `while`, `for` und `do while`. [Maya M]

# 5 Projektbeschreibung

## 5.1 Allgemeines

### 5.1.1 Gestik und Körpersprache

Die Gestik eines Menschen ist ein sehr bedeutsames Mittel zur Kommunikation. Sie vermittelt einen Subtext (unterbewußte Nachrichten) in vielerlei Hinsicht. Zum einen kann sie die Kommunikation unterstützen und zum anderen aber auch Gesagtes widerlegen.

Die menschliche Sprache hat sich möglicherweise von einem gestischen Kommunikationssystem aus entwickelt. Ein neuronaler Kreis, der sowohl bei Affen als auch bei Menschen nachgewiesen wurde, ist dafür verantwortlich. Forscher haben bei Versuchen mit Affen festgestellt, daß die Gruppe der Neuronen, welche die Sprachentwicklung auslösen, immer dann aktiv sind, wenn der Affe einen Gegenstand ergreift oder mit ihm hantiert. Diese Nervenzellen feuern jedoch auch dann, wenn das Tier den Versuchsleiter ähnliche Handlungen ausführen sieht. [Arb]

Durch kraftvolle und bestimmte Gesten verraten wir Selbstsicherheit. An sich herumzupfen, sich kratzen und in anderer Weise nervös agieren enthüllt Anspannung und Unsicherheit. Wer Gesten eher vermeidet, wirkt eingeschüchtert. Leidet jemand unter einer depressiven Störung, setzt er kaum noch Gesten ein. Unsere Hände untermalen und kommentieren im Gespräch unsere Worte – und bringen damit eindeutiger und deutlicher zum Ausdruck, was gemeint ist, als die Sprache das könnte. Nach vorn gestreckte Arme und eine einen Schlag andeutende Handbewegung unterstreichen entschiedene Worte oder eine Entscheidung. Nach oben offene Hände symbolisieren Geben, Nehmen und Bitten. Sind die Hände nach unten geöffnet, zeigen sie Zudecken, Beschwichtigen oder Herabmindern an.

Menschliche Bewegungsabläufe sind sehr komplex. Sehr viele Faktoren spielen eine Rolle dabei, wie ein Bewegungsablauf wirkt und was damit ausgedrückt wird. Schon ein einfaches Neigen des Kopfes um ein paar

Grad zur Seite, geben der Körperhaltung einen völlig anderen Charakter.

So wurde die Körpersprache von Politikern auf einen computergenerierten Charakter übertragen, danach wurden Studenten die Bewegungen vorgespielt, und ohne die Menschen, von denen diese Bewegungen stammten, überhaupt gesehen zu haben, konnten die Leute Aussagen über Sympathie, Intelligenz, und sogar äußere Attraktivität machen.

Dies deutet auf die Komplexität der menschlichen Bewegung hin und was genau dabei eine Rolle spielt, wie wir eine Körperhaltung oder Geste interpretieren, ist bis heute noch nicht geklärt [Frey].

## 5.2 Motion-Capturing

### 5.2.1 Einleitung

Motion Capture ist eines der meistdiskutierten Themen der Computeranimation heutzutage. Es wird jedoch oft nicht richtig verstanden oder überbewertet. Deshalb soll hier allgemein erklärt werden, was Motion Capture ist.

Bei Motion Capture Systemen werden die Bewegungen eines Objektes im dreidimensionalen Raum gemessen. Ursprünglich wurde diese Technologie für die Biomechanik und die Orientierung in virtuellen Welten entwickelt, sie hat sich jedoch in den letzten Jahren rapide weiterentwickelt.

Bisher waren die Bewegungen eines virtuellen Charakters nur so real wie es die Nachahmungsfähigkeit eines Animators zuließ, doch nun bietet Motion-Capture die Möglichkeit, reale Bewegungen auf virtuelle Darsteller zu übertragen. Motion-Capture-Systeme dienen somit als Schnittstelle zwischen Mensch und Maschine und ermöglichen so die Erschaffung synthetischer Figuren mit den individuellen Bewegungscharakteristiken lebender Schauspieler.

Der entscheidende Faktor für Motion-Capture ist also Realismus. Die Bewegungen virtueller Charaktere wirken jedoch oft künstlich. Das liegt vor allem an den Vergleichsmöglichkeiten des Betrachters. Bei Schreibtisch-



lampen oder Fantasiefiguren fallen Abweichungen im Bewegungsmuster nicht so auf, geht es jedoch um Menschen oder Tiere, sind dem Betrachter deren natürliche Bewegungen sehr wohl vertraut. So bald also menschen- oder tierähnliche Charaktere animiert werden, muß dies entweder sehr übertrieben oder in engen Toleranzen realistisch erfolgen. Bei den vielen verschiedenen Ausdrucksmöglichkeiten, die allein beim Gang eines Menschen auftreten, läßt sich nachvollziehen, was es für ein Problem darstellt, einen virtuellen Charakter real anmutend zu animieren. Motion Capture bietet sich hier als nützliches Werkzeug an.

Motion Capture wird aber auch in anderen Gebieten eingesetzt, z.B. in der Medizin oder bei der Überwachung und Analyse von technischen Bewegungsabläufen. Um Motion Capture Systeme sinnvoll und effektiv einsetzen zu können, ist es jedoch notwendig, ihre Funktionsweisen und Eigenschaften zu verstehen. [MoCap 1].

### 5.2.2 Die ersten Einsätze von Motion-Capture

Es gibt Motion-Capture schon seit ungefähr 15 Jahren im Bereich Computeranimation. Die erste kommerzielle Verwendung für Motion-Capture war 1984, als Bob Abel und sein Team bei Abel and Associates für den National Canned Food Information Council den Spot „Brilliance“ produzierten, der 1985 während der Super Bowl lief. 1984 war Keyframe-Animation noch keine Alternative und die wenigen existierenden Computer-Grafik-Firmen hatten keinen Zugang zu den damaligen Motion-Tracking-Technologien, sodaß Abel eine eigene Lösung für das Problem finden mußte. Der Spot besteht aus einem weiblichen Chrom-Roboter, dem die Schauspielerin Kathleen Turner ihre Stimme lieh.

Nachdem „Brilliance“ gezeigt wurde, begannen sich einige Firmen für Motion-Capture zu interessieren. Bei Omnibus, einem neuen Computer-Grafik-Studio, experimentierten Brad deGraf und Jeff Kleiser für einige Kunden damit, unter anderem für Marvel Comics und die Jim Henson Company. Die Firma schloß ihre Türen 1987, und beide deGraf und Kleiser gründeten jeweils die Firmen deGraph/Wahrman und Kleiser/Walczak. 1988 bekam deGraph/Wahrman von Silicon Graphics den Auftrag, für ihre neuen 4D Computer eine Demonstration zu machen. Die Demo nannte sich „Mike the Talking Head“ und wurde auf der SIGGRAPH 1988 gezeigt. Sie bestand aus einem Charakter, der mit dem Publikum inter-

agierte. Er wurde von einem Puppenspieler auf der Konferenz gespielt und wurde in Echtzeit auf der neuen SGI-Hardware gerendert.

Während der letzten Jahre nahm die Verwendung der Technologie behutsam aber stetig zu. Firmen wie Pacific Data Images, Digital Domain und Industrial Light & Magic setzten die Technik erfolgreich in Filmen wie "Batman Forever", "Batman und Robin", "Titanic", "Die Mumie" und Star Wars: Episode I" ein. Die Anwendung der Technologie wird weiterhin wachsen, besonders auf dem Gebiet der digitalen Stunts und der Darstellung von Massenszenen im Hintergrund betrifft.

### 5.2.3 Technischer Hintergrund

Motion Capture beinhaltet das Vermessen der Position und Orientierung eines Objekts im physikalischen Raum, gefolgt vom Aufnehmen dieser Information in einer durch den Computer verwendbaren Form. Interessante Objekte hierfür sind menschliche und nichtmenschliche Körper, Gesichtsausdrücke, Kamera- oder Lichtpositionen, und andere Elemente einer Szene.

Sind die Daten einmal in einer vom Computer verwendbaren Form aufgenommen, können Animatoren sie verwenden, um Elemente in einer computergenerierten Szene zu kontrollieren.

Für Echtzeit-Motion-Capture-Systeme können diese Daten interaktiv verwendet werden (mit minimaler Verzögerung), um ein Echtzeit-feedback zu erhalten. Alias|Wavefront's Kinemation oder MotionSampler, in Verbindung mit einem Flock of Birds Eingabegerät, ist ein Beispiel einer Echtzeit-Motion-Capture-Umgebung.

Andere Motion-Capture-Geräte sind nicht echtzeitfähig, entweder dadurch, daß die Bewegungsdaten zusätzliche Nachbearbeitung benötigen um in einer Animation verwendet werden zu können, oder dadurch, daß die ermittelten Daten nur ein Standbild der vermessenen Objekte liefern. Das optische System von Motion Analysis und der DID monkey sind Beispiele für nicht echtzeitfähige Systeme.

#### 5.2.4 Wie unterscheidet sich Motion Capturing von synthetischer Animation?

Bei der synthetischen Animation kontrolliert der Endanwender die Animationskurven oder Attribute von Szenen-Elementen direkt (meistens durch Keyframes oder Bewegungspfade) oder durch numerische Simulationstechniken.

Motioncapture-basierte Animation benutzt aufgezeichnete Positionen und Orientierungen von realen Objekten, um den synthetischen Animationsprozess zu verbessern, indem sie Grundinformationen für Animationskurven, Timing, oder Kontrolle von Attributen liefert.

Animation, die sich nur auf Motion Capture stützt, benutzt aufgezeichneten Positionen und Orientierungen von realen Objekten, um Animationskurven zu erzeugen, die von synthetischen Objekten in der computergenerierten Szene übernommen werden. Aufgrund von nicht zu vereinbarenden Geometrien, Qualitätseinbußen bei den Motion Capture-Daten und Anforderungen von der kreativen Seite sind Animationen selten nur auf Motion Capture basierend.

## 5.2.5 Optische Motion-Capture Systeme



Abb 2: Optisches Motion-Capture - Full Body Tracking

Beim optischen Motion-Capture System werden reflektierende Marker an dem Objekt, dessen Bewegung aufgezeichnet werden soll, befestigt. Diese Marker werden dann abgefilmt, und mit nur einem Bit (sehr hoher Kontrast) weiterverarbeitet. Die Marker sind kleine Kugeln, die in reflektierendes Material eingewickelt sind. Die Marker werden von Hochgeschwindigkeits-Kameras aufgenommen. Die Zahl der benutzten Kameras hängt von der Art des Motion-Capture ab. Für Gesichtsmotion Capture (Facetracking) reicht meistens eine Kamera, manchmal zwei. Beim Ganzkörper Motion-Capture (Full Body Tracking) werden vier bis sechs (oder mehr) Kameras verwendet, um die Abdeckung des ganzen aktiven Bereiches zu gewährleisten. Um den Kontrast zu verbessern, ist jede Kamera mit Infrarot- (IR) LEDs ausgestattet und IR (pass) Filter sind über der Kameralinse angebracht. Die Kameras sind an Controller-Karten

angeschlossen, die sich typischerweise in einem PC-Chassis befinden. Abhängig vom System wird Hochkontrast- (1 bit) Video auf dem Host aufgezeichnet. Bevor mit dem Motion Capture begonnen wird, wird ein Kalibrierungs-Gerüst – eine sorgfältig ausgemessene und konstruierte 3D-Anordnung von Markern aufgenommen. Diese dient als Referenz für die Motion-Capture Session. Nach einer Motion-Capture Session, müssen die Daten nachbearbeitet (getrackt) werden. Mittels Triangulation kann anschließend die Position jedes Markers ausgehend von den Markerbildern der Kameras ermittelt werden.

Beim Tracken können einige Probleme auftreten, wie zum Beispiel das Verwechseln von Markern, fehlende oder fehlerhafte Daten, und unerwünschte Reflexionen. Dadurch kann das Tracken je nach Qualität der aufgezeichneten Daten ein ziemlich zeitaufwendiger Prozess sein.



Abb 3: Optisches Motion-Capture - Face Tracking

Vorteile von optischem Motion Capture:

- Große Aktionsfläche (Fläche, auf der sich der Performer bewegen kann) möglich
- Abhängig vom benutzten System und der erfordernten Präzision kann die Aktionsfläche beliebig groß sein.

- Keine störenden Kabel

Der Performer ist physikalisch nicht mit dem Motion Capture-System verbunden. Damit ist es für den Performer zum Beispiel möglich, außerhalb des Capture-Bereiches zu beschleunigen und wieder abzubremesen, wenn zum Beispiel wirklich schnelles Laufen aufgezeichnet werden soll.

- Marker sind keine große Investition

Marker kosten nicht viel, und so könnte man theoretisch hunderte von Markern in einer Szene verwenden. Die Grenzen liegen hier eher in der Videoauflösung und den Grenzen der Trackingsoftware, sodaß das praktische Maximum wahrscheinlich bei deutlich weniger liegt.

- Hohe Samplingrate

Bei einer Samplingrate von 120 – 250 Hz können die meisten menschlichen Bewegungen leicht erfasst werden. Bei Würfeln und Schlägen jedoch kommt auch diese Samplingrate an ihre Grenzen. Wenn ein Ball schnell geworfen wird, legt die menschliche Hand 33 cm in 1/120 Sekunde zurück. Für Schlagereignisse wie Trommeln, Schlagen und hartes Fallen können die Beschleunigungen Frequenzen weit über 120 Hz erreichen. Glücklicherweise sind solche Bewegungen auch für das menschliche Auge schwer nachzuvollziehen und der Verlust an Genauigkeit ist meistens unmerklich.

#### Nachteile von optischem Motion Capture

- Anschaffungskosten

Mit \$150,000 bis \$250,000 US sind optische Systeme die teuersten der beschriebenen Systeme. Die Produktionskosten sind auch höher, da der Vorgang Ähnlichkeit mit Film- oder Videoproduktionen aufweist.

- Lichtempfindlichkeit

Da die heutigen optischen Systeme auf Kontrast basieren, sind Hintergrund, Kleidung und Raumbelichtung von Bedeutung.

- Empfindlichkeit gegenüber Reflexionen

Nasse oder glänzende Oberflächen (Spiegel, Böden, Schmuck usw) können falsche Ergebnisse verursachen. Da ein Marker mindestens von zwei Kameras gesehen werden muß (für 3D-Daten), kann ganze oder teilweise Verdeckung durch den Performer selbst, Bodenmatten oder andere Marker zu fehlenden oder fehlerhaften Daten oder vertauschten Markern führen. So wird zum Beispiel oft die Hüfte durch die Hand (im Stehen) oder die Hüfte durch den Ellbogen (im Sitzen) oder die Hand durch einen Gegenstand, der in der Hand gehalten wird, verdeckt.

- Tracking-Zeit

Wie oben erwähnt, kann das Tracking abhängig von der Komplexität der Bewegungen, den Anforderungen an die Genauigkeit und der Qualität der ermittelten Rohdaten weit mehr Zeit in Anspruch nehmen als das eigentliche Capturing.

- Nicht echtzeitfähig

Da es kein unmittelbares Feedback bezüglich der Qualität der aufgezeichneten Daten gibt, ist es unmöglich, zu wissen, ob eine Bewegung angemessen aufgezeichnet wurde. Deswegen müssen in jedem Fall mehrere Takes aufgezeichnet werden, um sicher zu gehen, daß auf jeden Fall ein Brauchbarer dabei ist.

- Nur Positionsdaten

Während einerseits zusätzliche Marker relativ billig sind, werden jedoch andererseits mehr Meßpunkte benötigt, da die Winkel der Gelenke durch die Abstände der Marker zueinander

berechnet werden müssen. Jüngste Entwicklungen der Trackingsoftware erlauben zwar das Erstellen von Rotationsdaten, was aber die Komplexität des Trackingprozesses wiederum erhöht.

- Empfindlichkeit der Kalibrierung

Da mehrere Kameras verwendet werden, muß der Referenzrahmen für jede Kamera exakt vermessen werden. Ist eine Kamera falsch ausgerichtet (z.B. weil gegen das Stativ gestoßen wurde), wird diese Kamera widersprüchliche Daten in Bezug auf die anderen Kameras liefern. Dies ist besonders störend zu dem Zeitpunkt, wenn ein Marker vom Gesichtsfeld der einen Kamera in das der Nächsten wandert, da dann zwei Punkte aus einem Marker erzeugt werden können, oder der Markerpfad springt.

### 5.2.6 Magnetische Motion Capture-Systeme

Magnetische Motion Capture-Systeme verwenden Sensoren, um exakt das magnetische Feld zu vermessen, das von einem Emitter erzeugt wird. Beispiele für magnetische Motion Capture Systeme sind der Ascension Bird/xaa und Polhemus Fastrak/xaa und Ultratrak/xaa. Solche Systeme sind echtzeitfähig, d.h. sie können zwischen 15 und 120 Samples/Sekunde liefern (abhängig vom Modell und der Anzahl der Sensoren) und sie liefern 6D-Daten (Position und Orientierung) mit minimaler Transport-Verzögerung.



Abb 4: Magnetisches Motion-Capture - Emitter



Ein typisches magnetisches Motion Capture-System hat eine oder mehrere elektronische Kontrolleinheiten, an die der Emitter und die Sensoren angeschlossen sind. Die Kontrolleinheiten wiederum sind durch ein Netzwerk oder eine serielle Schnittstelle mit einem Host Computer verbunden. Die Motion Capture – oder Animationssoftware kommuniziert mit diesen Geräten über ein Treiber-Program. Die Sensoren sind am Performer oder den Gegenständen, die getrackt werden sollen angebracht. Der Emitter sitzt entweder über oder seitlich der Active Area. Es darf sich kein Metall innerhalb der Active Area befinden, da es das Magnetfeld stören kann.



Abb 5: Magnetisches Motion-Capture

#### Typisches Setup für Charakteranimation

Die einleuchtendste Methode, die Sensoren zu platzieren, ist, an jedem Gelenk einen zu befestigen. Die physikalischen Beschränkungen des menschlichen Körpers (Arme müssen mit der Schulter verbunden sein usw.) erlauben jedoch eine exakte Methode mit weitaus weniger Sensoren. Da ein magnetisches System Positions- und Orientierungsdaten liefert, ist es möglich, die Jointpositionen durch Kenntnis der Längen der Gliedmaßen des Performers zu berechnen.

Die typische magnetische Motion Capture Session verläuft ähnlich wie ein Film-Shooting. Durch sorgfältig vorbereitete Proben wird sichergestellt, daß die Performer mit den Einschränkungen durch die Kabel und dem verfügbaren Active Space, der zur Aufzeichnung zur Verfügung steht, vertraut sind. Geprüft werden muß auch oft das Befestigen der Sensoren, um sicherzugehen, daß die Kabel alle Bewegungen des Performers mitmachen. Das Drehbuch wird in kleine Stücke aufgeteilt, die bearbeitbar sind, und das Storyboard ist oft unter dem Gesichtspunkt Motion Capture gemacht. Jeder Shot muß oft mehrere Male aufgezeichnet werden, und ein Audiotrack wird oft zur Synchronisation verwendet.

Da magnetische Systeme Echtzeitdaten liefern, können sich der Regisseur und die Schauspieler die Ergebnisse der Motion-Capture-Session während dem eigentlichen Take und sofort danach mit Audioplayback und uneingeschränkter Freiheit, was die Kameraführung betrifft, ansehen. Dieser schnelle Interaktionszyklus macht magnetisches Motion Capture ideal für Situationen in denen das Bewegungsfeld nicht so groß sein muß und direkte Interaktivität zwischen Regisseur, Performer und Computer-Charakter eine Rolle spielt.

#### Vorteile von magnetischem Motion Capture

- Positions- und Orientierungsinformation

Man braucht weniger Aufzeichnungspunkte (Sensoren)

- Minimale Kalibrierung erforderlich

Magnetische Motion Capture Systeme messen Entfernungen und Rotationswinkel in Bezug auf ein einziges Objekt, den Emitter. Die Erfassung von anderen Daten erfordert nur die Kenntnis der Position des Emitters (und natürlich Meßgenauigkeit).

- Keine große mechanische Anfälligkeit

Die Systeme wurden in vielfältigen Produktionsumgebungen eingesetzt, auch bei on-air- und Live-Auftritten, und das über mehrere Jahre hinweg. Auch wenn die Eigenschaften, was

Transport, Wartung, und Qualität des Materials angeht, noch verbessert werden können, sind sie doch besser als bei den optischen Systemen.

- Echtzeitfähig

Diese Methode erlaubt interaktive Anzeige

- Anschaffungskosten

Das typische System kostet unter \$40,000 US – wesentlich weniger als bei optischen Full-Body Systemen.

#### Nachteile von magnetischem Motion Capture

- Empfindlichkeit gegenüber Metall

Dieser Mangel wurde was die erforderliche Genauigkeit betrifft für Entertainment Produktion überbewertet. Ascension hat sich als weniger empfindlich gegenüber Metallen erwiesen als Polhemus. Beachtet werden muß allerdings, daß die Bühne, die Wände und die sonstigen Gegenstände für eine Motion-Capture Session nicht metallisch sind.

- Begrenzter Bewegungsraum

Der maximale effektive Aktionsradius liegt im wesentlichen bei weniger als das Maximum für optische Systeme

#### 5.2.7 Elektromechanisches Motion-Capture

Eine weitere Variante von Motion-Capture-Geräten ist der elektromechanische Anzug. Diese Systeme sind Armaturen, die vom Performer während des Capturings getragen werden. Sie verwenden Potentiometer, um die Rotation der Gelenke zu messen. Ihr Ausgang kann Bewegungsdaten in Echtzeit generieren, aber ihre Fehleranfälligkeit ist groß und ihre Samplingrate ist niedrig. Diese Systeme sind nicht besonders teuer. Weil sie

nur die Rotation der Gelenke messen, haben die meisten elektromagnetischen Anzüge keine Möglichkeit, die globale Position der Performance zu messen, sodaß sie zu diesem Zweck noch eine andere Tracking-Technology benötigen. Manche Anzüge enthalten einen elektromagnetischen Sensor, der sie anfällig für Störungen des Magnetfeldes macht. Ein Beispiel für einen elektromagnetischen Anzug ist das Gypsy 2.5 Motion-Capture-System von Analogus.

### 5.2.8 Zum Verständnis des teilweise schlechten Rufs von Motion Capture

Ob Performance Animation (Motion Capture) eine Kunstform ist, ist Kernthema vieler Diskussionen. Manche glauben, daß das künstlerische Wesen der Performance Animation die Performance selbst ist. „Ein Motion Capture-System ist ein Aufnahmegerät wie eine 35mm- oder Videokamera, die Kunst liefern der Regisseur und die Schauspieler.“ sagte Francesco Chiarini, Manager des Electronic Arts Worldwide Motion Capture Studios. „Vielleicht waren Maler und Fotografen vor ein paar Jahrzehnten noch in einem Konkurrenzkampf miteinander, doch das ist nicht mehr der Fall, und dasselbe gilt auch für Motion Capture. Die jüngste Generation von Animatoren ist Motion Capture gegenüber nicht mehr so feindselig eingestellt, wie es die älteren während der letzten zehn Jahre waren.“

Tim Johnson, co-Regisseur von „Antz“ sagte einmal, Charakteranimation mit Performance-Animation zu vergleichen ist, wie Micky Maus mit Kermit dem Frosch zu vergleichen. Beide Charaktere sind verschieden und gehören nicht in den gleichen Cartoon. Der Punkt ist, daß Kermit nicht vorgibt, Micky Maus zu sein und sein eigenes, andersgeartetes Publikum hat. Performance-Animation versucht, mit der Keyframe-Animation zu konkurrieren und hier liegt die Wurzel des Problems.

Wann sollte Motion Capture verwendet werden?

Motion Capture sollte in Situationen herangezogen werden, wenn hyperrealistische Bewegung erforderlich ist oder für Charaktere, die dafür gedacht sind, durch einen Puppenspieler bewegt anstatt handanimiert zu werden. Bei dieser Einführung in Motion Capture wurde auf die Quellen [MoCapWP] und [MoCap 2] zurückgegriffen.

## 5.3 MotionLab

MotionLab ist ein Tool, das sowohl auf der Technik des Motion-Capturings aufbaut, als auch die Werkzeuge der Keyframe-Animation aufgreift. Es ist für die Software Maya und in deren Scriptsprache MEL implementiert. Aufgabe dieses speziellen Werkzeuges ist es, den großen Vorteil, den die Motion-Capture-Technologie bietet, nämlich ein perfektes Abbild von menschlichen (oder nichtmenschlichen) Bewegungsabläufen zu liefern, mit den Freiheiten, die der Animator bei der KeyFrame-Animation hat, zu verbinden.

MotionLab soll schnell und unkompliziert zum Ziel führen und in seiner Bedienung möglichst intuitiv sein. Es wird davon ausgegangen, daß der Anwender mit den grundlegenden Eigenschaften und Funktionalitäten von Maya und der Charakteranimation vertraut ist

Bei der traditionellen Keyframeanimation ist es relativ einfach, eine Bewegung durch Translations- und Rotationskurven zu definieren und zu modifizieren. Trotzdem ist es eine sehr komplexe Aufgabe, die hohe Anforderungen an die Fähigkeiten des Animators stellt, durch Keyframing von Splinekurven die Gliedmaßen eines Charakters einzeln so zu koordinieren und manipulieren, daß er glaubwürdig erscheint. Somit wäre eine etwas allgemeinere Kontrolle über den Charakter sinnvoll. Solch eine allgemeinere Kontrolle würde es auch sinnvoller machen, vorgefertigte Sequenzen und Animationslibraries zu benutzen, was momentan noch ziemlich schwierig ist, da die Animationen aus der Library selten auf die jeweiligen Bedürfnisse passen.

Aus den oben genannten Gründen scheint es sinnvoll, Tools zu entwickeln, mit denen schon vorhandene Bewegungen wiederverwendet und an die individuellen Bedürfnisse angepaßt werden können.

## 5.4 Möglichkeiten

Zu diesem Zweck lassen sich Techniken aus dem Bereich der Bild -und Signalverarbeitung übernehmen, die neue Möglichkeiten zum Bearbeiten, Mischen, und Anpassen von Bewegungen eröffnen.

Dabei wird ein Freiheitsgrad eines Gelenks eines Charakters als aufgezeichnetes Signal behandelt.

Ein Signal enthält an jedem Frame einen Wert für den bestimmten Freiheitsgrad, zu dem das Signal gehört. Diese Werte ergeben sich entweder aus einer Splinekurve bei der traditionellen Keyframe-Animation, oder entsprechen den getrackten Markern eines Motion-Capture-Systems. Die Signal-bearbeitungs-Methoden, die ich hier verwandt habe, sollen Kontrolle über die Bewegungen auf einer höheren Ebene schaffen, also z.B. die Geschwindigkeit einer Bewegung direkt kontrollieren zu können.

Ein durch ein Motion-Capture-System erhaltenes Bewegungssignal kann also mit den Signalbearbeitungsmethoden aus der Bildbearbeitung modifiziert werden.

## 5.5 Ziele

Ziel von MotionLab ist es, die Nachbearbeitung von Motion-Capture-Daten so einfach wie möglich und auf einer niedrigen Spezialisierungsstufe zu halten. Hierdurch wird die vielseitige Einsatzmöglichkeit gewährleistet. Es soll außerdem eine allgemeinere Kontrolle über den Charakter schaffen.

MotionLab soll auf alle Arten von Charakteren und Animationen anwendbar sein. Es soll keine Einschränkung, was die Namensgebung der Joints, die Art der Attribute (obwohl es sich wohl in den meisten Fällen um Rotationsdaten von Joints handeln wird), die Anzahl der Joints oder die Länge der Animation betrifft, geben.

Außerdem soll das Interface bedienungsfreundlich und übersichtlich sein.

## 5.6 Aufbau

MotionLab ist modular aufgebaut. Es besteht aus mehreren gleichberech-

tigten Modulen was auch im Design des Interface erkennbar sein sollte. Diese Module sind voneinander unabhängig. Sie können in beliebiger Reihenfolge oder auch einzeln eingesetzt werden. Die Ergebnisse bauen jedoch trotzdem aufeinander auf und je nach Reihenfolge variiert das Endergebnis.

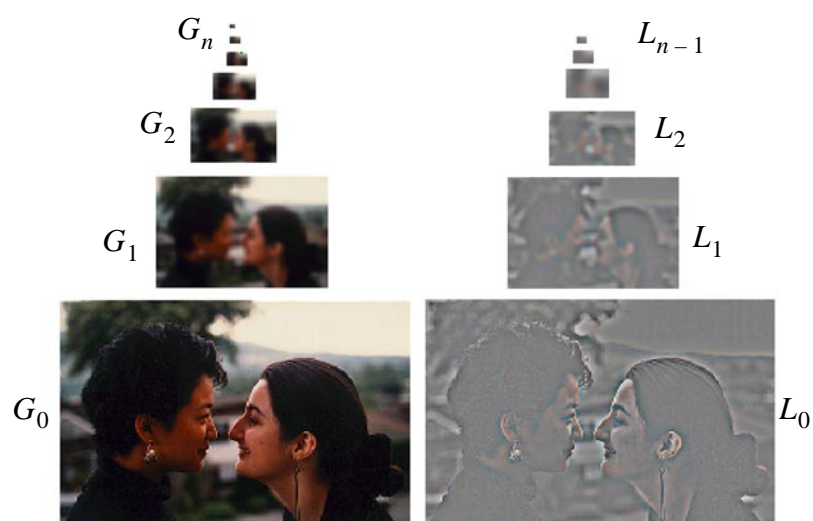
Diese Module sind im einzelnen:

- Motion Equalizer
- Edit
- Blend
- Speed
- Cycle

### 5.6.1 Motion Equalizer

Wie oben erwähnt, lassen sich Bewegungssignale mit Methoden aus der Bildbearbeitung modifizieren. Ein solcher Ansatz besteht im Isolieren einzelner Frequenzbänder. Er basiert auf digitaler Filterung. Ein Filter ist allgemein eine frequenzselektive Schaltung. Digitale Filter verarbeiten im Gegensatz zu analogen Filtern keine zeitkontinuierlichen, sondern zeitdiskrete Datenwerte. Signale mit bestimmten Frequenzen durchlaufen das Filter, während Signale mit anderen Frequenzen gesperrt oder gedämpft werden. Die Signalfrequenzen, die ein Filter durchlaufen, liegen in Bereichen oder Bändern. Diese Frequenzbänder enthalten Charakteristiken einer Bewegung, so lassen sich zum Beispiel aus einer Gehbewegung qualitative Faktoren wie "träge" und "nervös" herausziehen. Dabei wird das Signal durch eine Ablaufkette von Tiefpassfiltern geschleust, um eine Reihe von Bandpass-Signalkomponenten zu erzeugen. Wendet man diese Filter rekursiv auf das Ergebnis der sukzessiven Filterstufen an, kann man damit ungefähr dasselbe Ergebnis erhalten, wie mit einer Fourier-analyse. Dieser Vorgang wird als Multi-Resolution-Filtering bezeichnet. Wendet man das Multi-Resolution-Filtering auf ein Bild an, besteht der erste Schritt darin, das Bild sukzessive durch die Tiefpässe laufen zu lassen, wobei das Bild jeweils mit dem Faktor 2 bei jeder Wiederholung

heruntergerechnet wird. Dieser Prozess wird wiederholt, bis die Bildgröße auf ein Pixel reduziert ist, welcher der Durchschnittshelligkeit entspricht. Die Bandpasspyramide wird dann berechnet, indem jeweils zwei aufeinanderfolgende Tiefpassbilder voneinander subtrahiert werden und das Subtrahenten-Bild als erstes Frequenzband herausgezogen wird.



$$G_0 = L_0 + L_1 + L_2 + \dots + L_{n-1} + G_n$$

Abb 6: Links: Tiefpasspyramide; rechts: Bandpasspyramide

Danach kann das Bild ohne weitere Manipulation rekonstruiert werden, indem alle Bandpass-Bilder aufaddiert werden plus der Durchschnittshelligkeit.

Das Prinzip des Multi-Resolution-Filterings kann also auch auf Bewegungs-Parameter eines Charakters angewendet werden, was aus folgendem Grund auch Sinn macht:

Niedrige Frequenzen enthalten allgemeine, grobe Bewegungsmuster, wohingegen hohe Frequenzen Details, Feinheiten und (im Fall von digitalisierter Bewegung) das meiste Rauschen enthalten. Jeder Bewegungsparameter wird als eindimensionales Signal behandelt, von dem aus die



Tiefpass- und Bandpassebenen berechnet werden. Die Abbildungen 7 bis 12 zeigen am Bewegungssignal der X-Rotation eines Oberschenkel-Joints, wie sich die verschiedenen Charakteristiken einer Bewegung auf die verschiedenen Bänder aufsplitten lassen. Addiert man die einzelnen Bänder, erhält man wieder das Originalsignal.

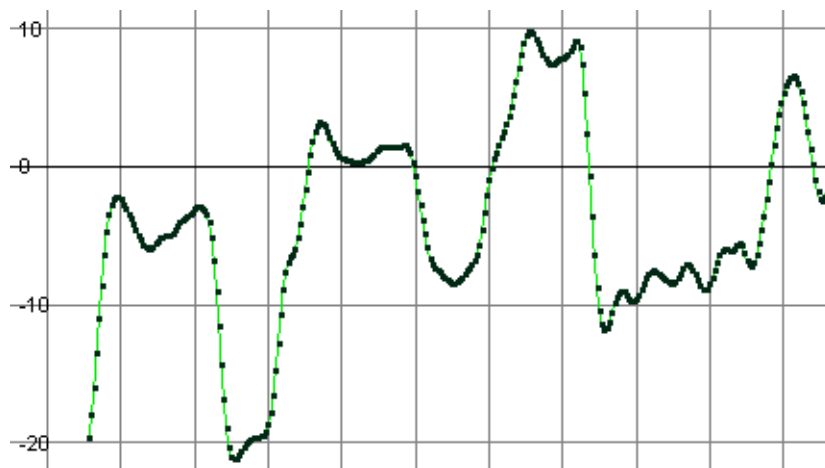


Abb 7: Originalsignal (X-Rotation eines Oberschenkel-Joints)

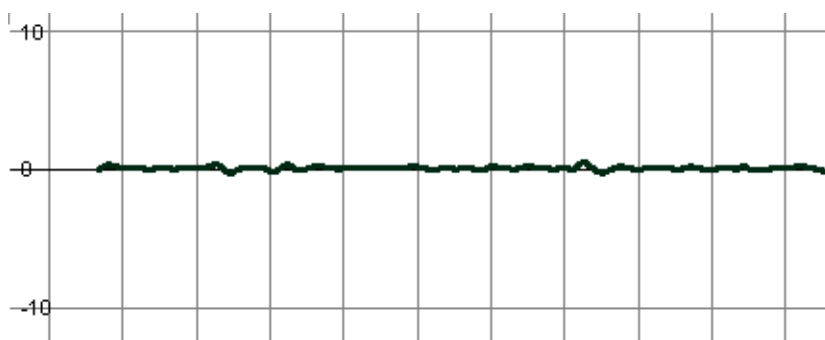


Abb 8: Band1 (enthält Rauschen (X-Rotation eines Oberschenkel-Joints))

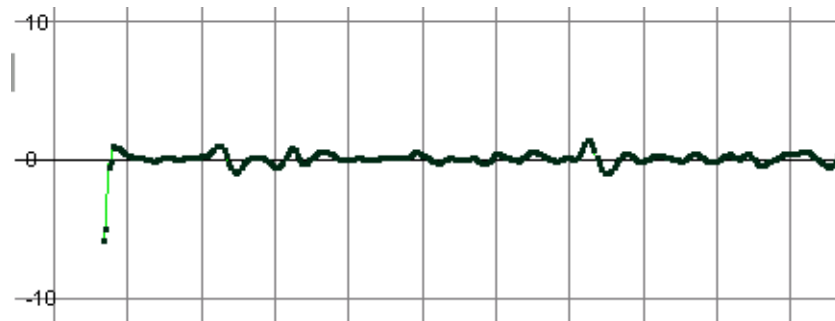


Abb 9: Band 2 (X-Rotation eines Oberschenkel-Joints)

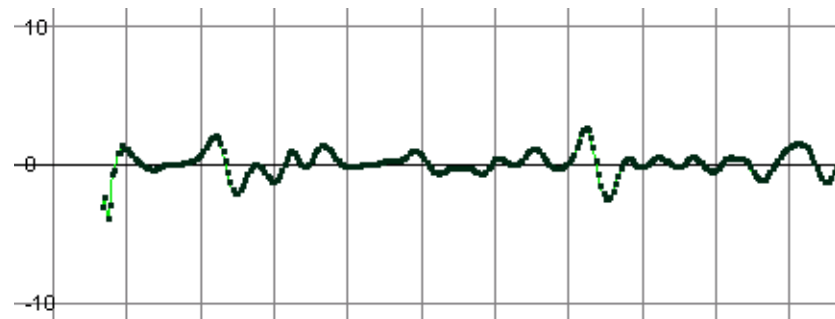


Abb 10: Band 3 (X-Rotation eines Oberschenkel-Joints)

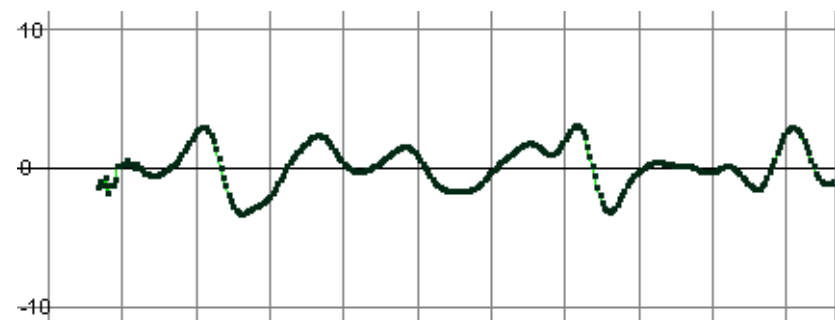


Abb 11: Band 4 (X-Rotation eines Oberschenkel-Joints)

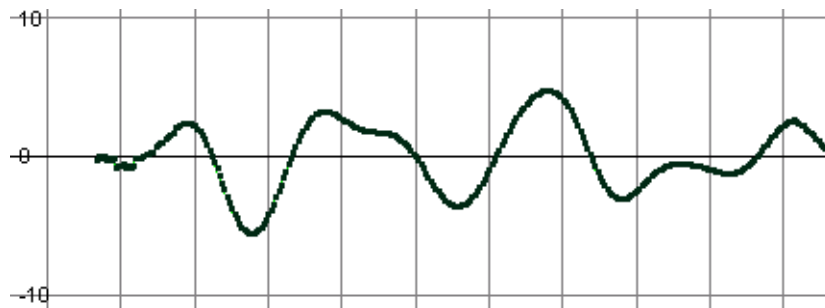


Abb 12: Band 5 (Hauptbewegung (X-Rotation eines Oberschenkel-Joints))

Die Amplituden der Bänder lassen sich nun separat anheben oder absenken. Die Applikation, die sich für dieses Verfahren anbietet, ist ein Equalizer. So wie der Equalizer in einem Audio-Verstärker ist es eine Art Grafik-Equalizer für Bewegung, wo die Amplitude (Gain) jedes Frequenzbandes individuell mittels eines Sliders angepaßt werden kann, bevor alle Bänder wieder zusammengefaßt werden, um die endgültige Bewegung zu erhalten.

Ein paar Beispiele sollen den Effekt von Anheben oder Absenken der einzelnen Frequenzbänder verdeutlichen:

Wurden die Mittleren Frequenzen einer Gehsequenz angehoben, war das Ergebnis ein geglätteter aber betonterer Gang. Wurden im Gegensatz dazu die hohen Frequenzen angehoben, fügte das ein nervöses Zucken der Bewegung hinzu.

Durch Anheben der tiefen Frequenzen erhielt man einen gedämpften, eingeschränkten Gang mit verminderter Bewegung der Gelenke.

Durch Eliminieren der hohen Frequenzen läßt sich Rauschen aus dem Bewegungssignal entfernen. Es kann auf diese Weise gereinigt und geglättet werden.

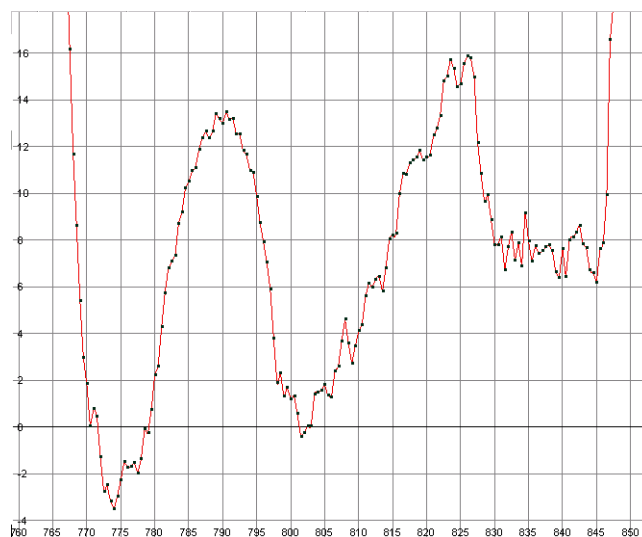


Abb 13: X-Rotation, Unterarm links ungeglättet

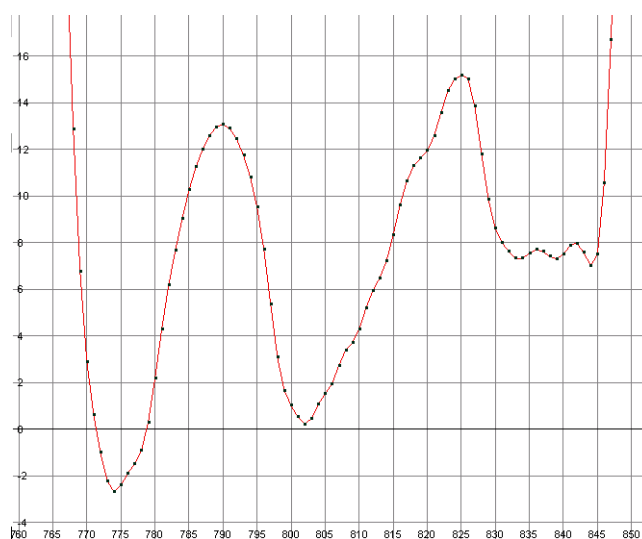


Abb 14: X-Rotation, Unterarm links geglättet

Im Gegensatz dazu kann durch Verstärkung der hohen Frequenzen auch Zittern erst erzeugt werden. So kann man beispielsweise einer Kamerafahrt ihre Sterilität nehmen und sie wie aus der Hand gefilmt aussehen lassen.

## 5.6.2 Cycle

Jede Bewegung läßt sich in verschiedene Bewegungsabschnitte einteilen, die jeweils für das Gelingen der Bewegung eine unverzichtbare Funktion haben.

Bei zyklischen Bewegungen wiederholen sich gleichartige Teilbewegungen (Beispiel: Laufen, Rudern). Der Bewegungsablauf läßt sich in zwei Phasen einteilen. Es kommt zu einer Überlagerung von Vorbereitungs- und Endphase (Phasenverschmelzung).

Zur Vorbereitungsphase:

Für die Vorbereitungsphase ist vor allem ihre Bedeutung charakteristisch: Durch sie werden Voraussetzungen geschaffen, die die in der nachfolgenden Hauptphase auszuführenden Aktionen optimal vorbereiten. Diese optimalen Voraussetzungen ähneln sich bei den meisten Menschen: Zuerst findet eine "Ausholbewegung" statt, die gegen die eigentlich gewollte Bewegungsrichtung gerichtet ist.

Ob man einen Ball oder einen Speer weitwerfen will, nach oben oder aus der Schrittstellung nach vorne springen will, oder wenn man am Reck einen Umschwung ausführen will, dann führt man vorweg in die Gegenrichtung eine Ausholbewegung aus. Man kann einen Ball natürlich auch ohne Ausholbewegung werfen und man kann auch ohne Ausholbewegung einen Umschwung beginnen. Die Folgen davon sind jedoch: Die nachfolgende Wurfbewegung erbringt nicht die volle Leistung und das Umschwingen wird sogar möglicherweise mißlingen.

An diesen beiden Beispielen kann man erkennen, wie die Vorbereitungsphase die nachfolgenden Phasen der Bewegung beeinflusst.

Wenn man die Bewegungen in dieser Phase allerdings übertreibt, also z.B. beim Werfen extrem weit ausholt oder beim Reck sehr hoch aufschwingt, kann die nachfolgende Leistung auch wieder schlechter werden.

Dies läßt sich erklären, wenn man analysiert, wozu diese (wohldosierte) Ausholbewegung da ist. Ein Grund liegt darin, den optimalen Beschleunigungsweg zu finden. Vergrößert man die Ausholbewegung, verlängert

sich damit auch der anschließend nutzbare Beschleunigungsweg. Dadurch wirkt die Kraft länger auf den beschleunigten Körperteil bzw. das Sportgerät, und dadurch entsteht eine höhere Endgeschwindigkeit.

Je nachdem, wie der Trainingszustand der Muskeln ist, kann ein langer Beschleunigungsweg die Muskulatur ermüden, sodaß sie nur noch geringere Kräfte aufbringen kann. So wird der Vorteil des längeren Beschleunigungsweges durch den Nachteil einer überforderten Muskulatur aufgehoben. Also kann man die Ausholbewegung nicht beliebig vergrößern. Ein weiteres Problem liegt darin, daß die Ausholbewegung entgegen gesetzt der Hauptbewegung verläuft, und man sie infolgedessen abbremsen und in die Hauptrichtung überleiten muß.

Die Hauptphase:

Das Kernstück einer sportlichen Bewegung ist die Hauptphase. Durch sie muß die entsprechende Bewegungsaufgabe unmittelbar bewältigt werden. Das Ziel, das man mit einer Bewegung erreichen will, z.B. beim Gehen vorankommen, oder beim Aufheben eines Gegenstandes die Position des Gegenstands zu ändern, wird durch die in der Hauptphase ausgeführten Aktionen erreicht: zum Beispiel das möglichst schnelle Strecken in Sprung-, Knie- und Hüftgelenk beim Springen, durch die vom Körper weg gerichtete Armstreckung beim Stoßen, oder die nach hinten gerichteten Arm- bzw. Beinbewegungen im Wasser beim Schwimmen oder beim Laufen und Gehen auf der Straße.

Man kann hierbei zwischen zwei Typen von Zielen unterscheiden, die in den jeweiligen Hauptphasen zu erreichen sind. Bei dem einen Typ liegt das Ziel darin, sich selbst einen Bewegungsimpuls zu erteilen, um von einem Ort zum anderen zu gelangen, um also sogenannte Ortsveränderungen auszuführen. Dazu gehört z.B. Laufen, Springen, Schwimmen oder Rudern.

Beim zweiten Typ geht es nicht um die Bewegung des eigenen, sondern eines anderen Körpers. Der eigene Körper oder auch nur Teile davon müssen so bewegt werden, daß ein anderer Körper, der natürlich in direktem Kontakt zum eigenen Körper steht in gezielter Weise eine Beschleunigung erfährt und somit bewegt wird.

Beispiele hierfür sind Werfen, Kicken, Schlagen eines Tennisballs oder

eines (Box-) Gegners.

### Endphase

Die Endphase ist zwar die letzte, jedoch nicht weniger wichtige Phase einer Bewegung. Sie dient als Überleitung von der Hauptphase in einen stabilen Ruhezustand oder zu einer weiteren Hauptbewegung (bei zyklischen Bewegungen). Das eigentliche Ziel der Bewegung wurde bereits in der Hauptbewegung erfüllt, und nun befindet sich die Person in einem Zustand, der erst durch eine Übergangsaktion beruhigt werden muß. Diese Übergangsaktion ist einerseits nötig, um die Sicherheit des Sportlers zu gewährleisten. Nachdem man eine Latte übersprungen hat, kann die Bewegung noch nicht am Ende sein, sondern es muß, um Verletzungen zu vermeiden, auch noch gewährleistet sein, das der Sportler sicher auf dem Erdboden ankommt. Wenn die Endphase einen Übergang zu einer weiteren Hauptbewegung darstellen soll, muß die Endphase der vorherigen Bewegung mit der Vorbereitungsphase der nächsten Bewegung überlagert sein. Beispiele hierfür sind Torlauf beim Skifahren, wo auf einen Schwung gleich der nächste folgt, oder beim Turnen am Reck, wo nach einem Umschwung der nächste folgt. Außerdem eben die zyklischen Bewegungen wie Gehen, Laufen, Rudern, wo das Zusammenfallen von Vorbereitungs- und Endphase charakteristisch und mit dem Begriff der Phasenverschmelzung gekennzeichnet ist.[Göhner]

Beim Cycle-Tool kann aus einer Bewegung, die eine sich wiederholende Bewegung enthält (wie z.B. Gehen) ein Cycle isoliert und zu einer Endlos-Bewegung gemacht werden. Dabei überlagern sich Vorbereitungsphase und Endphase (Phasenverschmelzung). Der Vorgang erinnert an das Tiling in der Bildbearbeitung, wo ein Bild wie bei Kacheln immer wieder an sich selbst angefügt wird und so eine ganze Wand gekachelt werden kann muß darauf geachtet werden, daß Anfang und Ende der Kachel gleich sind, damit die Übergänge von einer Kachel zur nächsten nicht sichtbar sind.

Das gleiche gilt beim Cycling einer Animation. Anfang und Ende müssen gleich sein.

Da das in den seltensten Fällen der Fall sein wird, wird beim Cycle-Modul folgendermaßen vorgegangen: Das Stück Animation, daß gecycelt werden soll, wird in sich selbst übergeblendet. Dadurch erhält man vor und

hinter dem Übergang absolut identische Animationssegmente, wodurch man die Animation nahtlos an sich selbst fügen kann.

Die beiden entscheidenden Faktoren beim Cyceln sind die Auswahl des zu cyclenden Animationsstücks selbst und die Länge des Übergangs. Beides wird dem Anwender überlassen, da eine Automation hier eine Einschränkung bedeuten würde.

### 5.6.3 Edit

Mit dem Edit-Modul kann ein Signal lokal verändert werden, während die Kontinuität und der Charakter des Signals beibehalten werden. Um eine Bewegung abzuändern, ändert der Animator nur die Pose des Charakters an ein paar Keyframes. Über diesen Versatz an mehreren Stellen wird für jeden Freiheitsgrad eine Spline-Kurve gelegt, und die Werte dieser Spline-Kurve werden auf die Original-Bewegung aufaddiert, um eine neue, veränderte Bewegung zu erhalten.

Da dieser Prozess nicht viel Rechenaufwand erfordert, erhält man einen schnellen Interaktionszyklus.

Mit diesem Modul erhält man ein Mittel, mit dem eine Bewegung wie das Ergreifen eines Objekts an einer Stelle des Tisches, sehr einfach in das Ergreifen eines Objekts irgendwo sonst auf dem Tisch umgändert werden kann. Das erlaubt eine einfache und direkte Veränderung von Motion-Capture Daten durch ein normales Keyframe-Interface. Deshalb ist das Interface des Edit-Moduls sehr einfach gehalten, da die Funktionalität des Keyframings einzelner Attribute durch das Interface von Maya sehr gut gelöst ist. So muß sich der Animator in diesem Bereich nicht einmal umstellen, obwohl damit eine völlig neue Funktionalität verbunden ist.

### 5.6.4 Blend-Tool

Eine weitere sehr wichtige Komponente von MotionLab bietet die Möglichkeit, zwei Animationen mit weichem Übergang ineinander überzublen- den.

Das macht den Einsatz von Bewegungsbibliotheken erst sinnvoll, da so



verschiedene Bewegungsmuster mit nahtlosem Übergang aneinandergereiht werden können.

So läßt sich zum Beispiel aus einer Geh-Animation und einer Lauf-Animation eine Animation erstellen, in der der Charakter erst geht und dann langsam anfängt zu laufen. Die Überlagerung erfolgt durch Interpolation. Dabei werden zu einem Zeitpunkt die Werte der beiden Ausgangsanimationen nach einer Blendfunktion durch folgende Formel miteinander vermischt:

$$(1 - \text{blendFunction}) * \text{input}[0] + \text{blendFunction} * \text{input}[1]$$

Die Blendfunktion wird durch eine Kurve, der BlendCurve dargestellt, die in einem GraphEditor, dem Blend-Editor innerhalb von MotionLab beliebig editiert werden kann. Die Ergebnis-Animation wird auf einen Ziel-Charakter gelegt und läßt sich in Echtzeit betrachten. Durch diesen Interaktionszyklus in Echtzeit wird ein optimaler Workflow erreicht.

Dieses Tool erfordert auch ein Interface, in dem alle Parameter übersichtlich dargestellt und editierbar sind.

Um das Ineinanderblenden von zwei Animationen sinnvoll zu machen, müssen die Animationen zeitlich aufeinander abgestimmt sein. Aus diesem Grund werden die beiden Animationen in einem Editor, dem Timing-Editor in MotionLab dargestellt. Dieser Timing-Editor, erinnert ein wenig an einen Editor eines Video-Editing-Systems. Das ist ziemlich naheliegend, da das ganze Verfahren dasselbe ist. Im Timing-Editor kann das Timing der Animationen aufeinander abgestimmt werden und die Animationen können skaliert werden. Ist ein aufwendigeres Editieren über die Zeit einer Animation nötig, so muß dieses zuvor im Speed-Modul erfolgen.

Der Editor reagiert wie alle Editoren von MotionLab nicht auf Änderungen der aktuellen selectionList, d.h. es kann zwischendurch an anderen Dingen gearbeitet werden, wobei natürlich andere Objekte selektiert werden, und danach wieder zu MotionLab zurückgekehrt werden, ohne die Darstellung der zu blendenden Objekte, sowie der BlendingCurve verloren zu haben. Dies erfordert natürlich, daß zuvor die zu blendenden Objekte als solche dem System bekanntgegeben werden. Daraufhin werden sofort die Editoren aktualisiert.

Alle Veränderungen von Parametern, die innerhalb des Blend-Moduls gemacht werden, beeinflussen in Echtzeit die anderen Parameter und sind von keiner Reihenfolge abhängig. Nach einer Veränderung im Timing-Editor kann man die BlendCurve manipulieren und danach in den Timing-Editor zurückkehren. Erst wenn man mit dem Ergebnis zufrieden ist, wird man auf die Zielanimation die BakeSimulation-Funktion anwenden und das Blending-Netzwerk, auf das hier nicht näher eingegangen wird, abgebaut.

Das zeitliche aufeinander Abstimmen der Animationen, kann auch automatisch erfolgen. Dabei werden die höchsten Ausschläge (Peaks) eines frei wählbaren Joints beider Skelette ermittelt. Falls es sich z.B. um ein Skelett mit Beinen und bei der Animation um eine Gehbewegung handelt, empfiehlt es sich den Oberschenkel zu nehmen. Danach werden der letzte Peak der ersten Animation und der erste Peak der zweiten Animation zeitlich übereinander gebracht.

### 5.6.5 Speed

Mit einer weiteren Komponente von MotionLab läßt sich die Geschwindigkeit einer Animation mittels einer Spline-Kurve beliebig editieren.

Da bei jeder lokalen Änderung die komplette Kurve für jeden Freiheitsgrad jedes Joints neu berechnet werden muß, ist diese Funktion leider nicht in Echtzeit möglich.

# 6 Implementierung

## 6.1 Allgemeines

MotionLab ist eine in MEL implementierte Applikation. Eine Entwicklungsumgebung steht, abgesehen von einem Scripteditor, nicht zur Verfügung. Zum Erzeugen von MEL-Scripts kann ein normaler Texteditor, wie z.B. Ultra-Edit verwendet werden. Ein MEL-Script sollte eine Länge von 500 Zeilen nicht überschreiten [Maya M]. Aus Gründen der Übersichtlichkeit und Wiederverwendbarkeit einzelner Elemente sowie der Zeilenbeschränkung wurde die Funktionalität von MotionLab auf mehrere Script-Files verteilt.

Da in MEL keine Zeiger zur Verfügung stehen, läßt sich die Verwendung globaler Variablen kaum vermeiden. Funktionen und Variablen, die direkt von Elementen der Benutzeroberfläche eines Windows angesprochen werden, müssen als globale Variablen und Funktionen definiert werden.

Die Elemente der Windows werden in Layouts definiert. Diese lassen sich beliebig schachteln und können unterschiedliche Bedienungs- und Darstellungselemente wie Buttons, Checkboxes oder Textelemente besitzen. Sie unterliegen einer strengen Hierarchie, die definiert, welches Layout sich in welchem befindet.

Hier ein Beispiel zur Definition eines einfachen Windows mit Scrollbars:

```
window;
    frameLayout;
        scrollLayout;
            columnLayout;
                button;
                button;
            columnLayout;
                text;
                button;
showWindow;
```

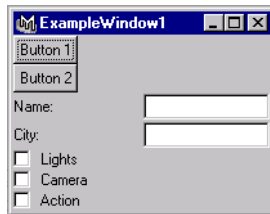


Abb 15: Beispiel-Fenster

## 6.2 Scripts und Prozeduren von MotionLab

### 6.2.1 SourceMotionLab.mel

Das Script `SourceMotionLab.mel` enthält die Prozedur `MotionLab`. Diese ist die erste Prozedur, die durch Eingabe des Wortes `MotionLab` in die Commandline aufgerufen wird. Sie liest die anderen zur Erzeugung und Durchführung von MotionLab notwendigen Prozeduren der jeweiligen Scripts ein und ruft die Prozedur `MotionLab` auf. Die `source`-Aufrufe der Prozedur `sourceMotionLab` hätten ohne weiteres in die Prozedur `MotionLab` gelegt werden können. Da es aber nicht möglich ist, das Argument eines `source`-Befehls aus einer Variablen und einer Konstanten Zeichenkette zusammensetzen, müssen alle Argumente der unterschiedlichen `source`-Aufrufe explizit eingegeben werden, d.h. der Anwender muß zu jedem `source`-Aufruf den gesamten Pfad des entsprechenden Files eingeben. Durch die Isolation dieser Aufrufe erhält der Anwender eine einfache und übersichtliche Prozedur, die abgesehen von dem parameterfreien Aufruf der `MotionLab`-Prozedur nur die jeweiligen `source`-Aufrufe enthält.

### 6.2.2 MotionEqualizer.mel

Das Script `MotionEqualizer` enthält die Prozedur `calculateBands`.

Nachdem über die Slider im `MotionEqualizer`-Interface die Multiplikatoren für die einzelnen Bandpass-Bänder festgelegt wurden, wird über den `Process`-Button die Prozedur gestartet.

Als erstes wird die Liste der selektierten Joints, auf die die Bandpasspyramide angewendet werden soll, übergeben. Danach wird der erste und letzte Keyframe der Animation ermittelt, um den Anfangs- und Endwert für die entsprechenden Schleifen zu erhalten. Anschließend werden in einem Array die Attribute, auf die die Bandpasspyramide angewendet werden soll festgelegt. Das sind im allgemeinen nur die Rotationsdaten rx, ry, rz. Optisches Motion-Capturing liefert zwar eigentlich nur die Positionsdaten der einzelnen Marker, aber im HTR-Format, in dem die Daten nach Maya gelangen, liegen die Daten schon als Rotationsdaten der einzelnen Joints vor. Nur der root-Joint, der höchste Joint in der Skelett-Hierarchie hat auch Translationsdaten.

Also ist es nötig, dem System mitzuteilen, wann es außer den Rotationsdaten auch die Translationsdaten berücksichtigen muß.

Der gesamte Algorithmus ist in vier Ebenen von Schleifen verschachtelt.

In der äußersten Ebene wird eine Schleife durchlaufen, die das gesamte Array der Elemente des Skeletts durchgeht und somit jeden Joint abarbeitet. Im Falle, daß die Hierarchy below-Checkbox angeklickt ist, ist dies die komplette Joint-Hierarchie, ansonsten besteht das Array nur aus einem Element, nämlich dem aktuellen Joint.

Innerhalb dieser Schleife befindet sich eine if-Abfrage, die feststellt, ob es sich bei dem momentanen Joint um den Root-Joint handelt, oder nicht. Dabei ist die Namensgebung der Joints völlig irrelevant, da sich der Root-Joint immer an letzter Stelle in dem Array, in dem die Namen aller Elemente des Skeletts liegen, befindet.

Danach folgt, immer noch innerhalb der ersten Schleife eine weitere Schleife, die durch die Attribute der Joints geht.

Wie schon erwähnt, wird jeder Bewegungsparameter als eindimensionales Signal behandelt, und so muß die Bandpasspyramide für jedes Attribut durchlaufen werden.

Im Fall, daß es sich um den Root-Joint handelt Rotations- und Translationsdaten, ansonsten nur Rotationsdaten. In der nächsten Schleifenebene befinden sich mehrere Schleifen, die jeden Frame der Animation durchlaufen.

In der ersten wird zunächst für jeden Frame der Wert des Quellcharakters in ein Array gelesen.

Nun folgt der eigentliche Algorithmus:

Die Länge  $m$  (Anzahl der Frames) von jedem Signal legt fest, wie viele Frequenzbänder ( $f_b$ ) berechnet werden:

Anstatt eine Pyramide von Tiefpass- und Bandpass-Sequenzen zu generieren, wo jede sukzessive Sequenz mit dem Faktor zwei reduziert wird, werden die Sequenzen in derselben Länge gehalten und der Kernel wird in jeder Ebene durch das Einsetzen von Nullen zwischen den Werten des Filter-Kernels ausgeweitet. Zum Beispiel mit einem Kernel der Breite 5, wie in MotionLab verwendet:

$$w_1 = [cbabc],$$

$$w_2 = [c0b0a0b0c],$$

$$w_3 = [c000b000a000b000c], \text{ etc.},$$

wobei  $a = 3/8$ ,  $b = 1/4$  und  $c = 1/16$ .  $a$ ,  $b$ ,  $c$  sind die Filterkoeffizienten. Sie können mittels einer Fouriertransformation berechnet werden. Sie bestimmen die Charakteristik des Filters. Durch sie wird z.B. der Frequenzbereich festgelegt, ab dem gesperrt wird.

Da es sich hier um Signale und nicht um Bilder handelt, ist die Verkleinerung auf ein Pixel nicht notwendig. Außerdem ist die Rekonstruktion des Signals schneller, wenn das Signal nicht in jeder Ebene wieder ausgeweitet werden muß. Im Folgenden wird der Motion-Multi-Resolution-Algorithmus im Detail dargestellt.

Die Schritte 1 bis 5 werden für jedes Bewegungs-Parameter-Signal durchgeführt:

1. Die Tiefpass-Sequenz von allen  $f_b$ -Signalen (Frequenzbändern) ( $0 \leq k < f_b$ ) berechnen, indem sukzessive das Signal durch die mit Nullen ausgeweiteten Kernels geschleust wird, wobei  $G_0$  das originale Bewegungssignal ist und  $G_{f_b}$  der Durchschnittswert (siehe oben):

$$G_{k+1} = w_{k+1} \times G_k$$

Das kann effizient berechnet werden, indem die Breite des Kerns konstant gehalten wird, und stattdessen Frames im Signal übersprungen werden ( $i$  geht über alle Frames des Signals).

$$G_{k+1}(i) = \sum_{m=-2}^2 w_1(m) G_k(i + 2^k m)$$

2. Durch Subtrahieren von Signalen, die nacheinander den Tiefpass durchlaufen haben, erhält man die Bandpass-Filterbänder. ( $0 < k < fb$ )

$$L_k = G_k - G_{k+1}$$

3. Die Amplituden für jedes Band mit dem Multiplikator multiplizieren (Anheben oder Absenken der Amplituden durch die Slider am Equalizer).

4. Das Bewegungssignal durch addieren der einzelnen Bänder plus dem Durchschnittswert wieder rekonstruieren.

$$G_0 = G_{fb} + \sum_{k=0}^{fb-1} L_k$$

[MSP]

### 6.2.3 Blend.mel

Das Script Blend enthält die Prozedur blend.

Auch sie enthält wieder mehrere Schleifen, die teilweise ineinandergeschachtelt sind.

Zunächst wird das Ziel-Skelett erstellt. Dazu wird zunächst die Hierarchy-Below-CheckBox im Blend-Interface abgefragt. Ist sie aktiv, wird die Liste der Elemente des ersten zu blendenden Skeletts bezüglich der Anzahl ihrer Elemente abgefragt. Danach wird das aktuelle Skelett dupliziert und an alle Elemente des Duplikats die Endung Blend anhängt.

Jetzt wird das Blending Netzwerk erstellt.

Dazu werden in der ersten Schleife, die durch alle Elemente des Skeletts geht die Namen der beiden zu blendenden Joints und des Ziel-Joints, sowie die Namen der Parameter zu Variablen zusammengefaßt. Mit Hilfe dieser Variablen werden die Quell-Joints in einem blendtwoAttr-Node verknüpft. Dieser Node mischt zwei Eingangswerte und weist das Ergebnis einem Ausgangswert zu.

Das Blenden erfolgt nach folgender Formel:

$$(1 - \text{blendFunction}) * \text{input}[0] + \text{blendFunction} * \text{input}[1]$$

Der blendtwoAttr-Node verfügt außerdem über ein Attribut namens Blender, welches bestimmt, wieviel von dem jeweiligen Ausgangssignal in das gemischte Signal einfließt. Dieser Blender wird mit der BlendCurve verbunden, die sich im unteren Grapheditor des Blend-Interfaces befindet. So werden alle sich entsprechenden Joints mit Blend-Nodes verbunden und alle Blend-Nodes schließlich mit der BlendCurve.

In der nächsten Schleife, die auch wieder über alle Elemente der jeweiligen Skelette geht, wird dieses Blending-Netzwerk wieder gelöscht.



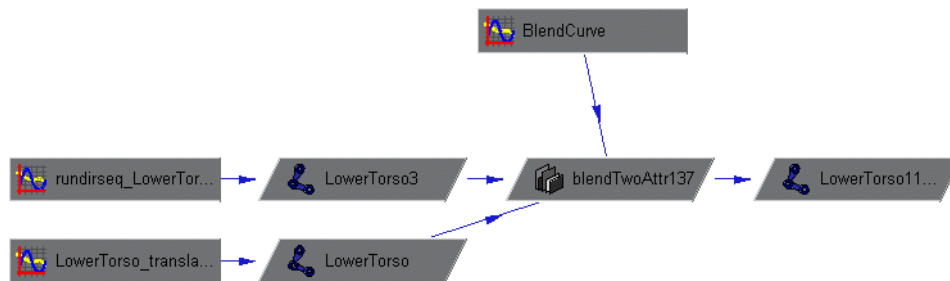


Abb 16: blendTwoAttr-Node

Desweiteren enthält die Datei die Methode `Peaks`. Für den Fall, daß der Anwender das zeitliche Aufeinanderabstimmen der Animationen automatisieren möchte, ein Joint mit dazugehörigem Attribut, auf den die Animationen synchronisiert werden sollen ausgewählt wurde und der `Autoadjust`-Button, wird die Methode `Peaks` aufgerufen.

In einer Schleife, die über die Länge der Animation geht, wird das Bewegungssignal zuerst der einen Animation in ein Array gelesen. In derselben Schleife wird noch in einer `if`-Abfrage geprüft, ob die Werte vor und nach dem momentanen Frame kleiner sind als der Wert am momentanen Frame. Ist das der Fall, handelt es sich um ein lokales Maximum.

Diese Schleife wird für beide Bewegungen durchlaufen.

## 6.2.4 Cycle.mel

Das Script `Cycle` enthält die Prozedur `makeCycle`.

Nachdem durch das `Cycle`-Interface die Länge der zu wiederholenden Animation, sowie die Länge des weichen Übergangs (`Transition`) festgelegt wurde, wird das Signal der Animation in ein Array gelesen. Dies geschieht wieder in zwei ineinander geschachtelten Ebenen von Schleifen. Die erste arbeitet alle Joint-Elemente des Skelettes durch. Innerhalb dieser sind nun mehrere Schleifen auf einer Ebene, die die Frames durchmachen. Die erste davon liest alle Daten des Ausgangssignals in ein Array.

In einer weiteren Schleife, die vom Anfang der Animation bis zum Anfang der Transition geht, werden die Daten der Quellanimation ohne Veränderungen auf das Zielskelett übertragen.

In einer weiteren Schleife, die über die Transition geht, wird zuerst überprüft, ob der momentane Parameter die x- oder z-Translation ist. Diese ist für die Fortbewegung des Charakters zuständig. Ist das der Fall, kann also die Translation auch über die Transition hinweg unverändert übernommen werden.

Ist dies nicht der Fall, wird mittels linearer Interpolation zwischen den Werten am Anfang der Transition und den Werten am Ende der Transition weich übergeblendet. Das ergibt den gleichen Effekt, als hätte man zwei identische Animationen gegen einander verschoben und im Bereich der Überlappung eine Blende stattfinden lassen.

In einer letzten Schleife, die vom Ende der Transition zum Ende der Quellanimation reicht, wird das Signal wieder unverändert vom Quellskelett auf das Zielskelett übertragen.

Anschließend wird die Animation auf den zu wiederholenden Abschnitt verkürzt. Anschließend werden die Bewegungen aller Joints bis auf die Translation des Root-Joints auf Cycle gesetzt. Dadurch werden die Animationskurven unendlich oft wiederholt.

Die Translation des Root-Joints wird auf Cycle with offset gesetzt. Dadurch werden die Animationskurven unendlich oft wiederholt, und der erste Key der gecyclten Kurve wird an den letzten Key der Originalkurve angehängt.

In diesem Fall wird auf das Erstellen eines Blend-Netzwerkes verzichtet. Die Interpolation erfolgt also nicht in Echtzeit. Durch den linearen Ablauf der Methode bei der nicht unterbrochen werden kann und darf, wäre eine Interaktivität ohnehin nicht möglich.

### 6.2.5 Speed.mel

Das Script Speed enthält die Prozedur speed.

Bevor diese Prozedur aufgerufen wird, muß im Speed-Interface zuerst der Root-Joint, dessen Geschwindigkeit verändert werden soll, festgelegt werden. Dann wird zunächst überprüft, ob die Hierarchy-Below-Checkbox aktiv ist. Daran wird erkannt, wie viele Joints in der darauffolgenden Schleife bearbeitet werden müssen.

Nun folgt eine Schleife, die über alle zu bearbeitenden Joints geht. Auch hier wird zunächst wieder festgestellt, ob es sich bei dem aktuellen Joint um den Root-Joint handelt. Ist dem so, werden außer den Rotationsdaten auch die Translationsdaten berechnet.

Die nächste Schleife liegt innerhalb der Ersten und reicht über alle Attribute der Animation. Innerhalb dieser liegen wieder mehrere Schleifen auf einer Ebene die sich über alle Frames der Animation erstrecken.

Die erste davon liest alle Daten des Ausgangssignals in ein Array. In einer weiteren Schleife wird der momentane Wert der SpeedCurve ausgelesen, die zuvor vom Anwender im SpeedCurve-Editor verändert werden konnte. Dieser Wert dient als Divident für die Zeitmarke, an der der nächste Frame gesetzt wird. Ist dieser Faktor z.B. 2, so wird der nächste Keyframe schon nach einem halben Frame gesetzt. Ist er 0.5, so wird der nächste Keyframe erst nach zwei Frames gesetzt.

Das wurde auf diese Weise gelöst, da der Wert in der SpeedCurve die Geschwindigkeit darstellen soll. Bei höherer Geschwindigkeit verkürzt sich also die Animation, bei niedrigerer verlängert sie sich.

### 6.2.6 Edit.mel

Das Script edit enthält drei Prozeduren. Die Erste, die Prozedur transform wird aufgerufen durch das Drücken des Transform-Buttons. In dieser Prozedur wird zunächst ein Transform-Node erstellt. Dieser Transform-Node erhält die Translations-Koordinaten des Root-Joints, um den es geht. Danach wird der Root-Joint in der Hierarchie unter den Transform-Node gestellt. Nun kann der Anwender den Transform-Node

nach Belieben positionieren und ausrichten, wie er gerne möchte

In der nächsten Prozedur `Process` wird die Animation mit den neuen Positions- und somit auch Translationsdaten auf die entsprechenden Animationskurven gelegt. Da Translationsdaten sich immer auf das Weltkoordinatensystem beziehen, also absolut sind, müssen sich mit einer Veränderung der Position auch die Translationsdaten ändern.

Dies wird hier folgendermaßen gelöst:

Der Root-Joint hängt momentan in der Hierarchie unter dem Transform-Node. Seine gekeyten Translationsdaten, also die Motion-Capture-Daten, beziehen sich nun auf den Transform-Node. Um herauszufinden, wie sie sich nun auf das Weltkoordinatensystem beziehen, existiert in Maya der `X-From`-Befehl. Mit diesem Befehl kann man unter anderem die Transformationswerte eines Objekts im Weltkoordinatensystem ermitteln.

Die Prozedur `process` enthält nun eine Schleife, die über alle Frames der Animation geht. Innerhalb dieser Schleife wird nun an jedem Frame der absolute Wert des jeweiligen Attributs ermittelt und danach am Root-Joint mit diesen Werten ein neuer Schlüsselwert gesetzt.

Anschließend kann der Transform-Node gelöscht werden.

Die letzte der Prozeduren `overAnimate` wird durch den `Change-Attribut`-Button im Interface aufgerufen.

Bei dieser Prozedur wird zunächst abgefragt, ob das jeweilige Attribut mit einer Animationskurve verbunden ist. Wenn dem so ist wird für dieses Attribut mittels dem Befehl `addAttr` ein zweites Attribut erzeugt. Danach wird ein Node vom Typ `addDoubleLinear` generiert. Diesem werden als Input die beiden Attribute zugewiesen, und er gibt als Ausgabe die Summe dieser Werte an den Joint weiter. Somit lassen sich über die durch die Motion-Capture-Daten gegebene Animation weitere Animation legen.

## 6.3 Zusammenfassung

In seinem momentanen Zustand ist MotionLab als ein Prototyp zu sehen, der die grundlegenden Eigenschaften und Forderungen an ein Tool dieser Art erfüllt. Es sind jedoch noch Erweiterungen der Funktionalität denkbar. So könnte man die Blendingfunktion auf beliebig viele Objekte ausweiten, ähnlich wie bei einem Soundprogramm mit beliebig vielen Kanälen. Es sind außerdem noch nicht sämtliche Fehler abgefangen. Diese Funktionalität muß im Hinblick auf eine verbreitete Verwendung noch vervollständigt werden.

Allgemein kann gesagt werden, daß die Qualität des Ergebnisses, das MotionLab liefert, immer von der Qualität der Ausgangsbewegungen sowie von der Parametereinstellung durch den Anwender abhängen wird.

Wünschenswert wäre im Hinblick auf die Anwendungsfreundlichkeit wäre eine Bedienung, die mehr über Drag'n'Drop funktioniert. Die Selektion über Buttons ist etwas zeitaufwendig und auf die Dauer etwas entnervend. Denkbar wäre zumindest auch eine Selektion über einen Cursor, der für das Tool spezifisch ist. Somit wäre trotzdem eine Selektion in einem anderen Zusammenhang möglich, die die Selektionsliste von MotionLab nicht beeinflußt. Beide Lösungen scheinen aber nicht in MEL vorgesehen und somit in diesem Rahmen nicht realisierbar zu sein.

## 7 Anwendungsbeschreibung

Das Tool wird gestartet, indem zunächst das sourceMotionLab.mel im Scripteditor in den Speicher gelesen wird, und anschließend entweder im Scripteditor oder von einem Shelf aus aufgerufen wird.

Als erstes Script in der Liste wird nun MotionLab.mel gestartet, wo das Benutzerinterface definiert wird. Es besteht aus einem TabLayout, das die verschiedenen Module enthält. Defaultmäßig ist das MotionEqualizer-Modul beim Starten an oberster Position im TabLayout.

### 7.1 MotionEqualizer

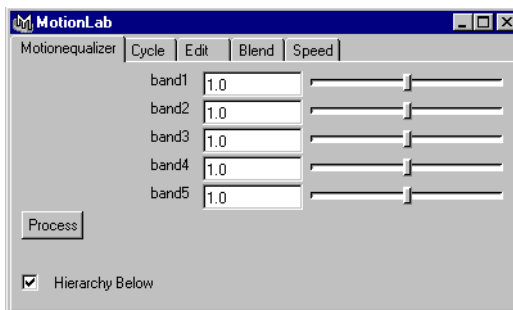


Abb 17: MotionEqualizer

Der MotionEqualizer besteht aus fünf übereinander angeordneten Slidern, die jeweils ein Frequenzband repräsentieren. Über diese Slider kann ähnlich wie bei einem AudioEqualizer die Amplitude (Gain) des jeweiligen Frequenzbands individuell eingestellt werden. Sind diese Einstellungen vorgenommen, wird die Berechnung der einzelnen Bänder über den Process-Button ausgelöst. Der Process-Button ist nur aktiv, wenn zuvor ein Objekt, auf dem mindestens ein Key liegt, selektiert wur-

de. Man kann durch die „Hierarchy Below“-Checkbox festlegen, ob MotionEqualizer nur auf den selektierten Joint, oder auch auf alle in der Hiwerarchie darunterliegenden Joints anwenden will. Ist kein Objekt selektiert, hat das Drücken des Process-Buttons eine Fehlermeldung zur Folge. Nach dem Drücken des Process-Buttons werden die Bänder berechnet, wieder zum Bewegungssignal zusammengefaßt und auf das Skelett übertragen.



Film 1: MotionEqualizer (Band 3 und 4 angehoben)



Film 2: MotionEqualizer (Band 1 und 2 eliminiert)

## 7.2 Edit

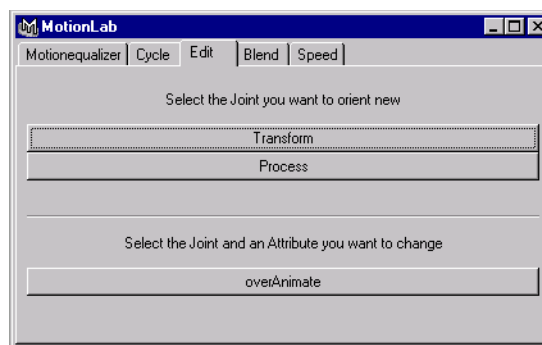


Abb 18: Edit

Dieses Modul dient zur nachträglichen Veränderung von Bewegungen. So kann zum Einen die Orientierung eines Charakters nachträglich verändert werden. Läuft zum Beispiel ein Charakter in der Originalbewegung nach Norden, kann man ihn ohne Probleme nach Südwesten laufen lassen. Hierzu muß der RootJoint selektiert werden. Danach wird der Trans-

form-Button gedrückt: Jetzt kann der Joint ausgerichtet werden.

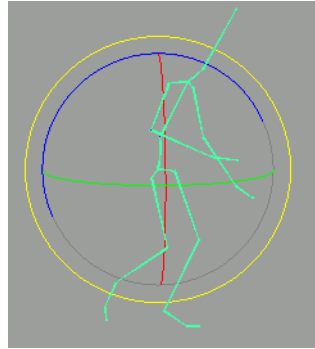


Abb 19: Modul für Orientierung

Ist die Transformation abgeschlossen, wird der Process-Button gedrückt. Daraufhin werden die Keys neu berechnet.

Zum Ändern kann auf eine bestehende Animation eine zusätzliche Animation darübergerlegt werden. Wenn zum Beispiel ein Charakter in der Originalbewegung ein Buch aus einem Regal holt, das Regal aber in einer anderen Höhe ist, kann mit diesem Modul die Höhe, in die der Charakter greift, angepaßt werden. Dazu muß ebenfalls zunächst ein Joint selektiert werden. Außerdem muß in der Channelbox das Attribut, das angepaßt oder verändert werden soll, selektiert werden.

LowerTorso	
Translate X	12.308
Translate Y	-12.005
Translate Z	-134.511
Rotate X	-9.887
Rotate Y	-9.111
Rotate Z	7.518
Scale X	1
Scale Y	1
Scale Z	1
Visibility	on

Abb 20: ChannelBox



Drückt man dann den overAnimate-Button, wird neben dem Originalattribut (z.B. RotateX) ein zusätzliches Attribut angelegt, das nun über den Graph-Editor mittels einer Animationskurve fein abgestimmt werden kann. Die beiden Attribute werden in einem Node namens AddDoubleLinear zusammengeführt, addiert und an den entsprechenden Joint weitergeleitet. Das Ganze geschieht in Echtzeit, man erhält also zu jedem Zeitpunkt ein visuelles Feedback über das Ergebnis seiner Manipulationen.



Film 3: Edit



Film 4: Edit (Anwendung)



Film 5: Edit - overAnimate (Anwendung)

## 7.3 Blend

Mit der Blendkomponente des Tools lassen sich zwei verschiedene Bewegungen entweder miteinander vermischen oder, was öfter von Nutzen sein wird, weich ineinander überblenden. So kann zum Beispiel aus einer Gehbewegung und einer Laufbewegung eine Bewegung generiert werden, bei der aus einem Gehen langsam ein Laufen wird. Die Handhabung dieses Moduls sieht folgendermaßen aus:

Zunächst müssen dem Tool die beiden zu blendenden Skelette bekannt gemacht werden. Dafür sind die beiden Bestätigungsbuttons im oberen Teil des Layouts gedacht, die mit Joint1 und Joint2 beschriftet sind. Man selektiert einen Joint, und definiert mit dem Drücken des Joint1-Buttons diesen Joint als den ersten der beiden zu blendenden Joints. Dabei kann es sich um ganze Skelette, sowie um einzelne Joints handeln. Um das festzulegen, dient die Checkbox „Hierarchy Below“. Ist sie aktiv, betrifft das Blending nicht nur den selektierten Joint, sondern auch alle in der Joint-Hierarchie unter ihm liegenden Joints. Entsprechend wird bei dem zweiten Joint verfahren. Man kann während des Selektierens beobach-

ten, daß in dem Timing-Editor unterhalb der Buttons die Animationen der beiden QuellJoints als Balken über die Zeit angezeigt werden.

Dieser Editor wird bei jeder Änderung der aktuellen Blend-Selektionsliste aktualisiert. Im nächsten Arbeitsschritt werden in diesem Editor die beiden Animationen zeitlich aufeinander abgestimmt. Dies erfolgt durch Verschieben der Balken über die Zeitachse. Hierbei sollte vor allem darauf geachtet werden, daß z.B. bei Gehbewegungen die Schrittfolgen ineinander übergehen. Dieser Arbeitsschritt kann auch automatisiert werden. Dazu muß der Button Autoadjust gedrückt werden. Auch wenn dieser Modus benutzt wird, kann im Falle eines Nichtgefallens die zeitliche Abstimmung hinterher trotzdem noch einmal manuell erfolgen. Bis auf die Selektion der Joints sind die Arbeitsschritte an keine feste Reihenfolge gebunden. Die hier beschriebene Reihenfolge erscheint jedoch als die Logischste.

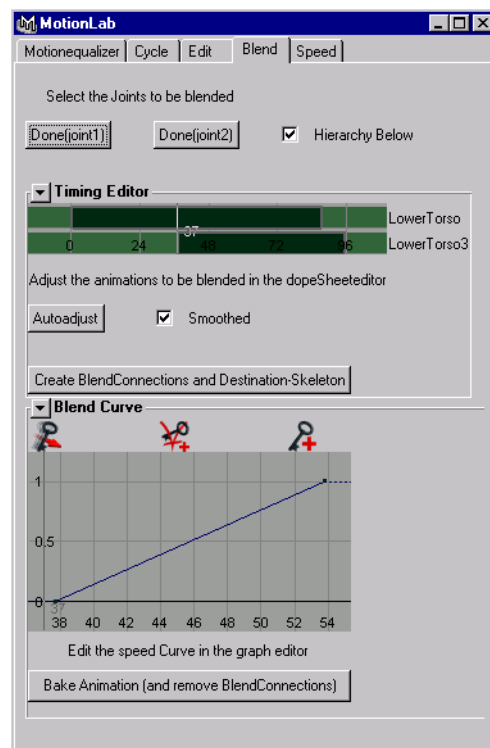


Abb 21: Blend

Weiterhin kann im BlendCurve-Editor, der sich unter dem Timing-Editor befindet, der Verlauf der Überblendung über eine Spline-Kurve individuell

eingestellt werden. In dieser Kurve bedeutet null eine hundertprozentige Übertragung der ersten Kurve mit keiner Berücksichtigung der Zweiten. Eins bedeutet den umgekehrten Fall. In seiner Anfangseinstellung hat die Kurve einen Key am Anfang des Übergangs mit dem Wert null und einem Key am Ende des Übergangs mit dem Wert eins. Dieser Editor bietet die wichtigsten Funktionen des Grapheditors. Es können Keys in die Spline-Kurve eingefügt werden, verschoben und gelöscht werden. Auch hier erfolgt ein grafisches Feedback in Echtzeit. Das bedeutet, alle Editoren sowie die 3D-Ansichtsfenster werden in Echtzeit aktualisiert.

Hat man auf diese Art die Animationen feinabgestimmt und ist mit dem Ergebnis, das man jederzeit in einem der 3D-Ansichtsfenster am Zielskelett überprüfen kann, zufrieden, wird in einem letzten Arbeitsschritt die Zielanimation gebaket, d.h. es werden in jedem Frame alle Parameter gekeyt. Anschließend wird noch das Blending-Netzwerk abgebaut. Das Baken und Abbauen des Blending-Netzwerkes geschieht durch das Drücken des Buttons „Bake Animation and remove Blend-Connections“.



Film 6: Blend



Film 7: Blend (Anwendung)

## 7.4 Speed

Im Speed-Modul kann die Geschwindigkeit einer Animation völlig frei über eine Spline-Kurve verändert werden.

Zunächst muß auch hier über den Done-Button eine Selektion des zu bearbeitenden Joints erfolgen. Auch hier kann man über eine „Hierarchy Below“-Checkbox einstellen, ob man nur einen einzelnen Joint oder auch die komplette Hierarchie unter ihm bearbeiten möchte.

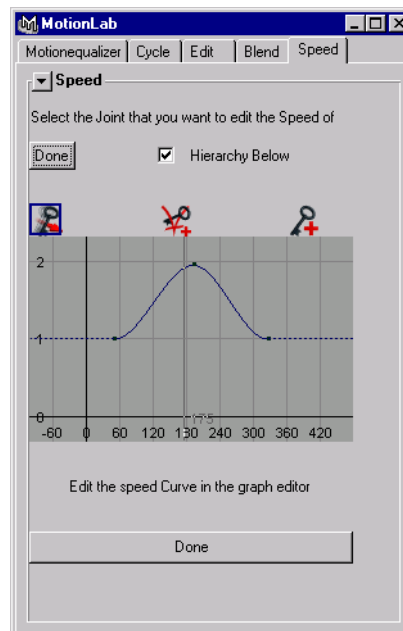


Abb 22: Speed

Auch in diesem Layout ist ein Editor zum Editieren einer SplineKurve zu finden. In diesem Speed-Editor stellt die Kurve die Geschwindigkeit der Animation dar. Bei der Initialisierung des Editors enthält die Kurve daher nur zwei Keys am Anfang und Ende der Animation mit jeweils dem Wert eins. Die Senkrechte Achse des Editors stellt die Geschwindigkeit dar, die Waagrechte die Zeit. Der Wert eins stellt den Originalzustand dar, Werte zwischen Null und Eins eine Verlangsamung der Animation, Werte über eins eine Beschleunigung. Auch hier erfolgt das Feedback wieder in Echtzeit.



Abb 23: Keys nach der Bearbeitung mit Speed



Film 8: Speed

## 7.5 Cycle

Das Cycle-Tool dient dazu, aus einem Stück Animation eine Endlos-Animation zu generieren, beispielsweise von einer Gehbewegung.

Dazu wird zuerst der Joint ausgewählt, dessen Animation bearbeitet werden soll. Danach muß im Timeslider das Stück Zeit selektiert werden, das aus der Animation extrahiert und zu einem Zyklus gemacht werden soll. Schließlich kann im Transition-Slider noch eingestellt werden über wieviel Frames sich der Übergang zwischen den momentan noch nicht aneinander passenden Randstücken der Animation erstrecken soll. In dieser Transition wird mittels Interpolation die Differenz zwischen den Endstücken geglättet.

Nun hat das Tool genügend Informationen und es kann der Done-Button gedrückt werden. Daraufhin wird ein Zielskelett generiert und auf dieses die Cycle-Animation gelegt.

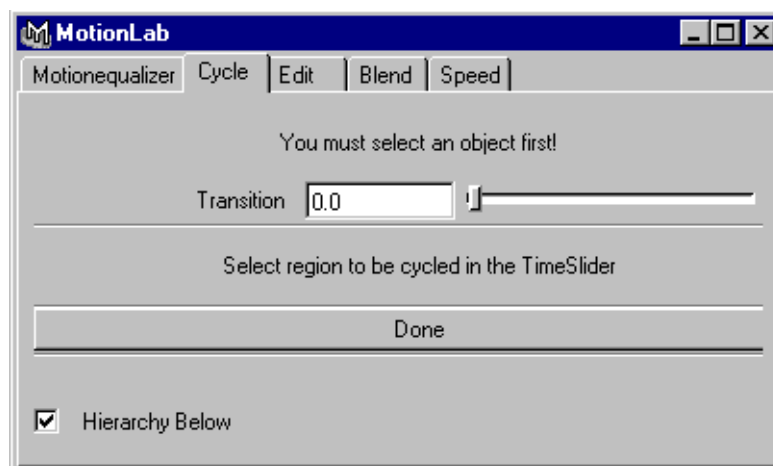


Abb 24: Cycle



Film 9: Cycle

## 8 Fazit

### 8.1 Ergebnis

Mit der Scriptsprache MEL wurde dem Anwender ein mächtiges Werkzeug an die Hand gegeben. Es erlaubt ihm einen weiten Eingriff in die Funktionalität der Software und eröffnet die Möglichkeit, diese auch zu erweitern. Besitzt der Anwender Grundkenntnisse im Programmieren, sind ihm im Rahmen Mayas keine Grenzen gesetzt. So lassen sich damit Werkzeuge zum Im- und Exportieren von fremden Dateiformaten für den Datentransfer zwischen verschiedenen 3D-Programmen herstellen, wie Werkzeuge zum Animieren großer Massen von Charaktern, wie sie z.B. in einem Fußballstadion benötigt werden. Für Aufgaben, bei denen große Mengen von Arbeitsschritten durch einen Algorithmus automatisiert werden können, bietet sich MEL als Lösung an.

So eben auch für die Bearbeitung von Motion-Capture-Daten, da dort grundsätzlich große Mengen von Daten anfallen.

Grenzen sind allerdings durch die Geschwindigkeit gesetzt. So ist für viele Anwendungen der Interaktionszyklus zu langsam, um die Anwendung echtzeitfähig zu machen, was natürlich für Werkzeuge aller Art wünschenswert ist.

MotionLab selbst demonstriert den Einsatz und einen Bruchteil der Möglichkeiten von MEL und zeigt die Stärken der Sprache. Es ist ein Werkzeug entstanden, daß dem Animator viele neue Möglichkeiten zur Benutzung von Motion-Capture-Daten bietet und somit Motion-Capturing noch attraktiver und effektiver werden läßt.

Außerdem kann durch den Einsatz von MotionLab enorm viel Zeit und Geld gespart werden, da eine Bewegung, die nicht gefällt, nicht zwingend ein weiteres Mal per Motion-Capturing aufgezeichnet werden muß, sondern solange bearbeitet werden kann, bis sie gefällt.

MotionLab in seiner gegenwärtigen Form erreicht im Großen und Ganzen die angestrebte Funktionalität, die Geschwindigkeit ist jedoch bei manchen Komponenten eine gewisse Behinderung, beim Cycle-Tool z.B.

wäre ein visuelles Feedback über die Parameter, die man einstellt, sehr wünschenswert.

## 8.2 Ausblick

Motion-Editing steht erst am Anfang. Es existieren noch so gut wie keine kommerziellen Werkzeuge zu diesem Thema. So läßt sich vermuten, daß auf diesem Gebiet noch eine ganze Menge Entwicklung stattfinden wird. Gerade der Ansatz, die Parameter für die Animation auf einem höheren, allgemeineren Level anzulegen, ist sehr vielversprechend. Dadurch wird sich Motion-Capturing von seinem ansatzweise schlechten Ruf erholen.

MotionLab wird innerhalb der Firma zur Verfügung stehen. Hierbei wird sich das Tool noch im praktischen Einsatz bewähren sowie dessen Akzeptanz sich zeigen müssen. Da das Tool in MEL implementiert ist, läßt es sich nicht auf andere Plattformen wie Lightwave oder Softimage übertragen.

Für MEL-Scripte gibt es gegenwärtig keine kommerziellen Märkte und somit ist an einen Vertrieb von in MEL implementierten Tools vorerst nicht zu denken.

## 9 Anhang

### 9.1 Begriffsdefinition

#### Attribute

Durch Attribute werden die Eigenschaften eines Objekts beschrieben. Die Attribute sind über die Zeit veränderbar und somit animierbar. Über sie werden Objekte animiert. Attribute lassen sich direkt über MEL oder den AttributEditor verändern. [Maya A]

#### Attributeditor

Der AttributEditor ist eine grafisch aufgearbeitete Schnittstelle, die Zugriff auf sämtliche Attribute eines Objekts ermöglicht. Außerdem kann über ihn auf die Construction History sowie deren Attribute zugegriffen werden.

#### Avatar

Der Begriff stammt aus dem Buddhismus und bezeichnet die Verkörperung eines Teils der Essenz Gottes. Auf den Bereich der Computeranimation übertragen, steht der Begriff Avatar für einen künstlichen, in 3D modellierten Menschen.

#### Constraints

Es gibt zwei Gruppen von Constraints: Basic und Dynamic Constraints. Durch Constraints werden Objekte gezwungen, bestimmte Werte bezogen auf andere Objekte nicht zu überschreiten. Das kann Orientierung, Position oder Skalierung betreffen. So lassen sich zum Beispiel Kameras oder Lichter an Objekte oder Gruppen von Objekten binden. Insgesamt stehen in Maya 8 Basic Constraints zur Verfügung.

Die Dynamic Constraints definieren das Verhalten von z.B. Aktive-/Passive Body zueinander. Hier stehen weitere 5 Constraints zur Verfügung



### Grapheditor

Im GraphEditor werden die Animationskurven der gekeyten Objektattribute angezeigt. Es handelt sich dabei um Splinekurven, mit den Keys als Kontrollpunkten. Man kann diese Kurven editieren, deren Form beeinflussen oder von hier aus die zugehörigen Attribute bearbeiten.

### Key / Keyframes

In einem Key wird der Wert eines Attributs zu einem bestimmten Zeitpunkt gespeichert. Man definiert als Keyframes markante Bewegungszustände. Durch mehrere Keyframes entsteht eine Animation, indem die Werte zwischen den Keyframes interpoliert werden.

### Manipulatoren

Durch Manipulatoren können Objekte transformiert werden. Die wichtigsten Manipulatoren Move, Rotate, Scale. Die Manipulatoren sind direkt in den 3D-Ansichtsfenstern sichtbar und es können von dort aus verschiedene Funktionen aufgerufen werden.

### Nodes

Alle Elemente in Maya sind aus Nodes aufgebaut. Es gibt beispielsweise Shape Nodes, Render Nodes, Transform Nodes, AnimCurve Nodes, Shader Nodes usw. Diese Nodes werden in der Szene verknüpft. Der Node einer Kugel im Raum beispielsweise besteht mindestens aus einem Shape Node, welcher die Geometrie, und einem Transform Node, der die Position in Beziehung zum Weltursprung definiert. Die Nodes können im ConnectionEditor individuell verknüpft werden. Auf diese Weise lassen sich alle Elemente in Maya wie in einem Baukasten zusammenbauen und lassen so alle Freiheiten. Es können individuelle Shader erstellt werden usw.

### Outliner

Im Outliner wird, wie im Hypergraph, die Szenenhierarchie grafisch dargestellt. Außer der Szenenhierarchie können über entsprechende Filter auch die Attribute der einzelnen Objekte aufgezeigt werden. Die Struktur

des Outliners ist mit der des Explorers von Windows Betriebssystemen vergleichbar. Untergeordnete lassen sich entsprechend ausblenden. Der Outliner eignet sich zum Auswählen von Objekten bei komplexen und unübersichtlichen Szenen. [Maya H]

#### Scanline-Renderer

Scanline-Rendering ist ein Renderverfahren, bei dem das Bild einer Szene Zeile für Zeile berechnet wird. Im Gegensatz zum Field-Rendering wird hierbei meist die gesamte Szene im Arbeitsspeicher behalten, wodurch die Nutzung des Speichers während des Renderings eingeschränkt ist.

#### ScriptEditor

Der Scripteditor dient zur Eingabe von MEL-Befehlen und zur Wiedergabe von Feedback. Aus diesem Grund ist der Scripteditor in zwei Bereiche aufgeteilt. Im oberen werden die Befehle ausgeführt, Resultate und Fehlermeldungen zurückgegeben. Im unteren Teil lassen sich Befehle und Prozeduren in beliebiger Anzahl eingeben. Er hat die grundsätzliche Funktionalität eines Texteditors und stellt die Schnittstelle zu Maya auf MEL-Basis dar.

#### Slider

Slider sind grafische Elemente, die eine Steuerung eines Wertes zulassen. Sie sind in ihrer Wirkungsweise und Aussehen vergleichbar mit Schiebereglern.

#### Tabs / Tab-Layout

Tabs sind ebenfalls grafische Elemente und erinnern in ihrem Aussehen an Karteikarten. Sie haben auch eine ähnliche Funktionalität.

## 9.2 Maya Lizenzen

Maya Complete

Besteht aus den Komponenten:

Maya Artisan

Maya F/X

Fusion Lite Edition (LE)

Invigorator Lite Edition (LE)

Maya Unlimited

Besteht aus den Komponenten:

Maya Artisan

Maya F/X

Maya Cloth

Maya Fur

Maya Live

Advanced Modeling

Maya Paint Effects ist eine Komponente von Maya, um mittels Pinselstrichen partikel-Effekte oder Geometrie auf eine 2D-Fläche oder auf oder zwischen 3D-Geometrie malen kann.

Maya F/X

Mit Maya F/X wird das Standardprogramm um ein ausgereiftes und komplexes Partikelsystem erweitert.

Fusion Lite Edition (LE)

In Zusammenabit mit Eyeon Software Inc. wird Fusion Light als Compositing-Werkzeug mit Maya vertrieben. Maya Fusion Light baut auf der Software Digital Fusion von Eyeon auf, ist aber im Funktionsumfang eingeschränkt. So werden lediglich die grundlegenden Compositing-Funktionalitäten, einfache Farbkorrekturen sowie weitere Bildformate einschließlich AVI und Quicktime unterstützt.

#### Invigorator Lite Edition (LE)

Diese Komponente von Maya dient zum Importieren von Vektorgrafiken im Adobe Illustrator-Format. Die Grafiken werden als SplineKurven importiert, Somit sind sie nachträglich editierbar und können zu 3D-Modellen weiterverarbeitet werden.

#### Maya Cloth

Mit Maya Cloth lassen sich realistische Textilien (Kleidung) erstellen, die sich auch wirklichkeitsgetreu bewegen. Es können auch Eigenschaften wie Stärke des Stoffs eingestellt werden. Die Kleidungsstücke werden wie echte Kleidungsstücke über Schnittmuster entworfen. Dadurch wird naturgetreues Verhalten des Kleidungsstückes auch an den Nähten gewährleistet.

#### Maya Fur

Fur ist ein Werkzeug, mit dem auf NURBS-Geometrien realistisches Fell und kurze Haare generiert werden können. Hierbei können Attribute wie Farbe, Lockigkeit, Länge, Ausrichtung etc. durch das Artisan Painting-Tool bearbeitet werden. Somit können die Haare regelrecht gekämmt werden.

#### Maya Live

Dabei handelt es sich um ein Tool zum Tracken von Kamerafahrten. Das bedeutet, daß die Position und Bewegung der Kamera in einem Realfilm erfasst und in einer 3D-Szene nachgebildet werden kann. Dies geschieht durch das Markieren mehrerer markanter Punkte, durch die per Triangulation die Rauminformation berechnet werden kann. Somit können 3D-Objekte in eine Szenen eingepaßt werden.

## Advanced Modeling

Hier werden die Modeling-Tools, die vom Power-Animator bekannt sind, und die höchste Präzision gewährleisten dem Maya-Anwender zur Verfügung gestellt.

Die Angaben beziehen sich auf [AW1].

## 9.3 MEL-Script

In der vorliegenden Arbeit werden die entwickelten MEL-Scripte funktional beschrieben. Damit sich der Leser ein Bild von der MEL-Syntax machen kann, ist hier beispielhaft das Script MotionEqualizer mit Interface, das sich normalerweise in dem Script MotionLab befindet, abgedruckt. Alle weiteren Scripts befinden sich auf der beigelegten CD.

```
global float $bandmultiplikator[8];

for ($counter = 1; $counter <= 8; $counter++)
{
    $bandmultiplikator[$counter]=1;
}

window;
    columnLayout;

        floatSliderGrp -label "band1" -field true
            -minValue -3.0 -maxValue 5.0
            -fieldMinValue 0.01 -fieldMaxValue 5.0
            -value 1.0
            -cc (" $bandmultiplikator[1]=#1;");

        floatSliderGrp -label "band2" -field true
            -minValue -3.0 -maxValue 5.0
            -fieldMinValue 0.01 -fieldMaxValue 5.0
            -value 1.0
            -cc (" $bandmultiplikator[2]=#1;");

        floatSliderGrp -label "band3" -field true
            -minValue -3.0 -maxValue 5.0
            -fieldMinValue 0.01 -fieldMaxValue 5.0
            -value 1.0
            -cc (" $bandmultiplikator[3]=#1;");

        floatSliderGrp -label "band4" -field true
            -minValue -3.0 -maxValue 5.0
            -fieldMinValue 0.01 -fieldMaxValue 5.0
            -value 1.0
            -cc (" $bandmultiplikator[4]=#1;");

        floatSliderGrp -label "band5" -field true
```

```

        -minValue -3.0 -maxValue 5.0
        -fieldMinValue 0.01 -fieldMaxValue 5.0
        -value 1.0
    -cc (" $bandmultiplikator[5]=#1;");

    button -label "Process" -c ("calculateBands;") -align "center";

showWindow;

//-----
//      Berechnung der Bänder
//-----

global proc calculateBands()
{
    global float $bandmultiplikator[8];

    float $orig[500];

    string $Bone;
    string $selection1[];
    string $Parameter[6];
    string $Bone1;
    string $together;
    global string $selection[];
    int $framecounter, $startFrame, $endFrame, $framecounterArray, $band, $m, $p;
    int $paramLimit;
    float $w [5];
    matrix $g[9] [2000];
    matrix $l [10] [2000];

    clear $selection1;
    clear $selection2;
    $selection1 = `ls -sl`;
    $selection2 = `ls -sl`;

    $startFrame=`findKeyframe -which first $selection1`;
    $endFrame=`findKeyframe -which last $selection1`;

    if ($selection1[0] != "")
    {
        $Parameter = {"rotateX", "rotateY", "rotateZ", "translateX", "translateZ", "translateY"};

        for ($bonecounter = 0; $bonecounter <= size($selection1)-1; $bonecounter++)
        {
            // Außer für LowerTorso nur Rotationsdaten
            //-----

            if ($bonecounter != size($selection1)-1)
                $paramLimit = 2;
            else
                $paramLimit = 5;
            //-----

            for ($paramcounter = 0; $paramcounter <= $paramLimit; $paramcounter++)
            {
                $Bone1 = $selection1[$bonecounter];
                $Bone2 = $selection2[$bonecounter];
                $together=$Bone1+"."+$Parameter[$paramcounter];

```

```

//-----
// Mocap-Daten in Array packen
//-----

for ($framecounter = $startFrame; $framecounter < $endFrame; $framecounter++)
{
    $framecounterArray=$framecounter-$startFrame+32;
    $g[0] [$framecounterArray] = `getAttr -time $framecounter $together`;
}

$w[1] = .0625;
$w[2] = .25;
$w[3] = .375;
$w[4] = .25;
$w[5] = .0625;

for ($band = 0; $band <= 4; $band++)
{
    $p=`pow 2 $band`;

    for ($framecounter = $startFrame; $framecounter < $endFrame; $framecounter++)
    {
        $framecounterArray=$framecounter-$startFrame+32;
        $g[$band+1] [$framecounterArray]=0;

        for ($m = -2; $m <= 2; $m++)
        {

            $g[$band+1] [$framecounterArray]=$g[$band+1] [$framecounterAr
ray]+($g[$band][$framecounterArray+$p*$m])*$w[$m+3];

        }

        $l[$band+1] [$framecounterArray] = $g[$band] [$framecounterArray] - $g[$band+1]
[$framecounterArray];
    }
}

cutKey -at $Parameter[$paramcounter] $selection1[$bonecounter];

for ($framecounter = $startFrame; $framecounter < $endFrame; $framecounter++)
{
    $framecounterArray=$framecounter-$startFrame+32;
    $l[9][$framecounterArray] =
($g[5][$framecounterArray]+$l[5][$framecounterArray])*$bandmultiplikator[5]+$l[4][$framecount
erArray]*$bandmultiplikator[4]+$l[3][$framecounterArray]*$bandmultiplikator[3]+$l[2][$frameco
unterArray]*$bandmultiplikator[2]+$l[1][$framecounterArray]*$bandmultiplikator[1];

    setKeyframe -attribute $Parameter[$paramcounter] -t $framecounter -v $l[9][$frame-
counterArray] $Bone1;
}
}
}
else

```

```
error( "There was nothing selected!\n" );  
}
```



## 9.4 Filmverzeichnis

Film 1:MotionEqualizer (Band 3 und 4 angehoben) .....	60
Film 2:MotionEqualizer (Band 1 und 2 eliminiert) .....	60
Film 3:Edit .....	62
Film 4:Edit (Anwendung) .....	62
Film 5:Edit - overAnimate (Anwendung) .....	62
Film 6:Blend .....	63
Film 7:Blend (Anwendung) .....	63
Film 8:Speed .....	64
Film 9:Cycle .....	65

## 9.5 Abbildungsverzeichnis

Abb 1: Benutzerinterface .....	11
Abb 2: Optisches Motion-Capture - Full Body Tracking .....	24
Abb 3: Optisches Motion-Capture - Face Tracking .....	25
Abb 4: Magnetisches Motion-Capture - Emitter .....	28
Abb 5: Magnetisches Motion-Capture .....	29
Abb 6: Links: Tiefpasspyramide; rechts: Bandpasspyramide .....	36
Abb 7: Originalsignal (X-Rotation eines Oberschenkel-Joints) .....	37
Abb 8: Band1 (enthält Rauschen (X-Rotation eines Oberschenkel-Joints)) ...	37
Abb 9: Band 2 (X-Rotation eines Oberschenkel-Joints) .....	38
Abb 10: Band 3 (X-Rotation eines Oberschenkel-Joints) .....	38
Abb 11: Band 4 (X-Rotation eines Oberschenkel-Joints) .....	38
Abb 12: Band 5 (Hauptbewegung (X-Rotation eines Oberschenkel-Joints)) ...	39
Abb 13: X-Rotation, Unterarm links ungeglättet .....	40
Abb 14: X-Rotation, Unterarm links geglättet .....	40
Abb 15: Beispiel-Fenster .....	48
Abb 16: blendTwoAttr-Node .....	53
Abb 17: MotionEqualizer .....	58
Abb 18: Edit .....	59

Abb 19: Modul für Orientierung .....	60
Abb 20: ChannelBox .....	60
Abb 21: Blend .....	62
Abb 22: Speed .....	64
Abb 23: Keys nach der Bearbeitung mit Speed .....	64
Abb 24: Cycle .....	65

## 9.6 Literaturverzeichnis

- [Arb] Michael A. Arbib  
Vor der Sprache war die Gestik  
[http://www-hbp.usc.edu:8376/HBP/people/faculty/arbib/Arbib\\_books.html](http://www-hbp.usc.edu:8376/HBP/people/faculty/arbib/Arbib_books.html)  
von der University of Southern California
- [AW] Alias|Wavefront-Homepage  
<http://www.aw.sgi.com>
- [Frey] Frey, Siegfried  
Die Macht des Bildes.  
Huber Hans, Goettingen  
1999
- [Göhner] U. Göhner:  
Prinzipien zur Analyse sportlicher Bewegungen.  
In: Sport - Theorie in der gymnasialen Oberstufe, S. 119 ff
- [Knappe] Heiko Knappe  
Maya - Alias|Wavefront's Animations-Tool der nächsten  
Generation, Digital Production Ausgabe 1/1998  
ACT Verlag GmbH, Pullach
- [Maya A] Using Maya Animation  
Alias|Wavefront, Toronto 1999

- [Maya H]            Using Maya Hypergraph, Sets & Expressions  
Alias|Wavefront, Toronto 1999
- [Maya M]            Using Maya MEL  
Alias|Wavefront, Toronto 1999
- [Maya W]            Maya for Windows NT 2.5, Online Tutorial
- [Mocap 1]           Alberto Menache  
Motion Capture for Digital FX Part One:  
The Basics of Motion Capture
- [Mocap 2]           Alberto Menache  
Understanding the controversy:  
An analysis of motion capture's bad reputation
- [MoCapWP]           Scott Dyer, Windlight Studios  
Jeff Martin, Alias|Wavefront  
John Zulauf, Alias|Wavefront  
Motion Capture White Paper  
<http://www.aw.sgi.com>
- [MSP]                Armin Bruderlin  
Motion Signal Processing  
Simon Frasier University  
Lance Williams  
Apple Computer, Inc.

# Erklärung

Ich erkläre, die vorliegende Arbeit selbst verfaßt und keine anderen als die angegebenen Hilfsmittel benutzt zu haben.

Potsdam, 28.06.2000