

Monster - Ein interaktives Projektions-System für bewegte Objekte bei Tanzperformances

MARIAN WEGER

BACHELORARBEIT

eingereicht am

Institut für Elektronische Musik und Akustik
Universität für Musik und darstellende Kunst Graz

im Studiengang

ELEKTROTECHNIK-TONINGENIEUR

in Graz

im September 2012

Diese Arbeit entstand im Rahmen des Gegenstands

Musikinformatik 1

im

Sommersemester 2012

Betreuer:

Winfried Ritsch Ao.Univ.Prof. DI

Inhaltsverzeichnis

Kurzfassung	v
Abstract	vi
1 Einleitung	1
2 Kommunikation bei multimedialen Bühnendarbietungen	3
2.1 Synchronisation	3
2.1.1 Unidirektionale Synchronisation	3
2.1.2 Bidirektionale Synchronisation	4
2.2 Interaktion	4
2.3 Beispiele von Interaktion bei Tanz-Aufführungen mit Video- projektion	4
2.3.1 D.a.v.e. - digital amplified video engine (Klaus Ober- maier und Chris Haring, 1999) [20]	5
2.3.2 Solo 4 > Three (Shadows) (Frieder Weiß, 2003) [30] .	5
2.3.3 Glow (Chunky Move, 2008) [2] [3]	7
3 Technische Realisierung einer adaptiven Videoprojektion	9
3.1 Einführung in die Videoprojektion	9
3.1.1 Projektor-Aufstellung	10
3.1.2 Entzerrung der Projektion	11
3.1.3 Zusammenhängende Projektion mit mehreren Projek- toren	12
3.2 Marktübersicht Projektions- und Tracking-Systeme	12
3.2.1 Hardware-basiert	13
3.2.2 Software-basiert	13
3.3 Marktübersicht: Hardware für optisches Tracking	17
3.3.1 Kamera	17
3.3.2 Passive Infrarot-Systeme	18
3.3.3 Wärmebildkamera	18
3.3.4 Structured Light Coding	19
3.4 Implementierung der adaptiven Videoprojektion	21
3.4.1 Hardware und Verkabelung	22

3.4.2	Das Tiefenbild	23
3.4.3	OpenGL Shading Language	23
3.4.4	Berechnung der Schablone	24
3.4.5	Ausgleich von schiefer Kinect-Position	26
3.4.6	Glättung	27
3.4.7	Einbinden der OpenGL Shader in Pd	28
3.4.8	Füllen der Schablone	29
3.4.9	Entzerrung der Projektion	29
3.4.10	Farbanpassung	29
3.4.11	Live Korrektur	30
3.4.12	Grafische Benutzeroberfläche	30
3.5	Diskussion des entwickelten Systems	31
4	Monster	35
4.1	Ausgangspunkt & Aufgabenstellung	35
4.2	Idee	36
4.3	Implementierung	36
4.3.1	Kamera	36
4.3.2	Beleuchtung für das Kamerabild	36
4.3.3	Technischer Aufbau	38
4.3.4	Optische Rückkopplung	38
4.3.5	Bildschirmanordnung	38
4.3.6	Midi-Steuerung	39
4.4	Choreografie	39
4.4.1	Die ersten Tests	39
4.4.2	Kostüm	41
4.4.3	Ablauf des Stücks	42
4.5	Diskussion der Aufführungen	44
4.5.1	21. Jänner 2012: Semperdepot, Wien	44
4.5.2	März 2012: Fidelio Wettbewerb, Wien	44
4.5.3	3. Mai 2012: Odeon, Wien	45
4.5.4	9. September 2012: Kino Šiška, Ljubljana (ICMC2012)	46
5	Zusammenfassung	48
A	Ergänzende Materialien	50
A.1	Auszüge aus dem Proposal für Monster [13]	50
A.1.1	Abstract	50
A.1.2	Description	50
A.1.3	Musical Idea	51
A.2	Kritiken im Auswahlprozess der ICMC2012	51
	Literaturverzeichnis	54

Kurzfassung

Tanzperformances nutzen immer mehr die Möglichkeiten neuer Medien wie Live-Elektronik und Video-Projektion. Oftmals besteht hierbei jedoch ein unzureichender Zusammenhang zwischen den einzelnen Komponenten. Diese Arbeit beschreibt die theoretischen Grundlagen von Synchronisation und Interaktion, und diskutiert ausgewählte Beispiele moderner Tanzperformances. Hierbei wird ausschließlich auf die Wechselbeziehung zwischen Tänzern und Videoprojektion eingegangen, da die Klangsteuerung durch Bewegung in anderen Arbeiten bereits hinreichend behandelt wurde. Desweiteren werden die nötigen Grundlagen zu Video Projektion allgemein erläutert.

Schließlich soll ein System zur Interaktion eines Tänzers mit Echtzeit-Videoprojektion implementiert und anhand der Aufführung des Werks *Monster* getestet werden. Hierbei soll in enger Zusammenarbeit mit der Komponistin Wen Liu ein hierarchisches Modell geschaffen werden, in dem die Musik den Tänzer synchronisiert, und dieser mit der Videoprojektion interagiert.

Abstract

Dance performances make more and more use of the possibilities, that are gained through new media technologies like live-electronics or video projection. But many times, they lack a proper relation between the individual parts. This work describes the basic principles of synchronisation and interaction and discusses selected examples of modern dance performances. It is important to note, that only the interrelationship between dancers and video projection is considered in this work, as motion control for sound is sufficiently described in other literature. The required basic principles of video projection are described as well.

The aim is to implement a system to enable a dancer to interact with live-video projection. This is to be tested in the performance of the work *Monster*. In close collaboration with the composer Wen Liu, a hierarchical model shall be created, in which the music synchronises a dancer, while selfsame interacts with the video projection.

Kapitel 1

Einleitung

In der Computermusik hat sich der Begriff der Live-Elektronik mittlerweile fest etabliert. Es ist gang und gäbe, dass Musiker mit ihren Bewegungen oder durch ihr Instrument elektronische Systeme steuern und somit zu ihrem Spiel synchronisieren.

Auch die Verwendung von Videoprojektion ein beliebtes Mittel zur Unterstützung von Musik- und Tanzaufführungen. Durch die Komplexität der gegenseitigen Wechselbeziehung der einzelnen Domänen „Klang“, „Videoprojektion“ und „Bewegung“ besteht hierbei jedoch gelegentlich eine unzureichende Verbindung. Oftmals nutzen multimediale Aufführungen ein zentralisiertes Modell der Synchronisation, bei dem die zeitliche Abfolge ausschließlich zentral von einer Domäne synchronisiert wird. Als Beispiel sei hier die Steuerung und Synchronisierung von Klang und Video zu den Bewegungen eines Tänzers oder Musikers angeführt. Bei Beteiligung mehrerer Musiker oder Tänzer ist deren künstlerische Interpretation jedoch stark eingeschränkt und es bleibt wenig Spielraum für spontane Ideen des Einzelnen.

Während interaktive Klangsteuerung vor allem in der elektronischen Musik mittlerweile ein fester Bestandteil ist, wird Interaktion beim Einsatz von Videoprojektionen nur selten genutzt. Dabei würden gerade moderne Tanzperformances stark von einem verstärkten Einsatz von interaktiver Videoprojektion profitieren.

Ziel dieser Arbeit ist nun, die Möglichkeiten der Synchronisation und Interaktion zwischen Tanz und Videoprojektion aufzuzeigen, und anschließend ein einsatzfähiges System hierfür zu implementieren, und an einem spezifischen Projekt zur Anwendung und Aufführung zu bringen.

Anhand des konkreten Beispiels soll ein hierarchisches Modell angewendet werden, bei dem die Musik einen Tänzer synchronisiert, und dieser wiederum mit der Videoprojektion interagiert. Der Tänzer soll also, von der Musik synchronisiert, eigenständig kreativ arbeiten können, indem die Videoprojektion an seine Bewegungen angepasst wird. Es soll ein System entwickelt werden, das es einem Tänzer ermöglicht, seine visuelle Erscheinung

in gleichem Maße durch elektronische Unterstützung zu verändern, wie es Musikern mit ihren Klängen in der Computermusik schon lange möglich ist. Dabei soll der Schwerpunkt weniger auf dem projizierten Material liegen, sondern vielmehr auf der adaptiven Projektion selbst.

Eine Interaktion des Tänzers mit dem Klang ist bei dieser Arbeit ausdrücklich nicht gewünscht. Dieses Thema wird u. a. in verschiedenen Arbeiten von Matthias Kronlachner diskutiert (siehe [12]).

Der Hauptteil dieser Arbeit gliedert sich in drei Bereiche. Im ersten Teil werden die theoretischen Grundlagen der Wechselbeziehung und Kommunikation bei Bühnenwerken im allgemeinen erläutert. Es werden die Begrifflichkeiten und Hintergründe von Synchronisation und Interaktion erläutert. Weiters wird anhand von Beispielen auf die Möglichkeiten der Interaktion von Tanz und Videoprojektion eingegangen.

Der zweite Teil widmet sich der Entwicklung eines Systems zur Interaktion von Videoprojektionen mit den Bewegungen eines Tänzers. Es werden die Grundlagen von Videoprojektion und Video-"Mapping" erklärt, sowie mögliche Systeme zur Erfassung des Körpers diskutiert. Weiters wird eine geeignete Lösung zum Tracking des Tänzers und der adaptiven Projektion gesucht, die den Anforderungen entspricht. Darauf aufbauend wird das System schließlich implementiert und getestet. Abschließend folgt eine Diskussion des entwickelten Systems.

Im dritten und letzten Teil wird das entwickelte System schließlich anhand eines konkreten Stückes getestet und an dessen individuelle Anforderungen angepasst. Ausgangsbasis ist eine Komposition der Komponistin Wen Liu, zu dem eine Tanzperformance mit Videoprojektion entwickelt werden soll. Es werden die notwendigen Erweiterungen und Anpassungen des Systems erörtert und implementiert. Weiters wird mittels der neu entstandenen Ausdrucksmöglichkeiten eine passende Choreographie ausgearbeitet. Das Ergebnis mehrerer Aufführungen wird anschließend diskutiert.

Am Ende der Arbeit folgt schließlich eine Zusammenfassung der Ergebnisse und deren kritische Diskussion. Als Ausblick werden außerdem mögliche zukünftige Erweiterungen und Probleme des entwickelten Systems erörtert.

Kapitel 2

Kommunikation bei multimedialen Bühnendarbietungen

Multimedial unterstützte Bühnendarbietungen setzen eine intensive Verknüpfung mehrerer grundlegend verschiedener komplexer Systeme voraus. Um eine reibungslose Aufführung zu gewährleisten, muss eine wechselseitige Kommunikation zwischen Computer-Systemen, Tänzern und Musikern hergestellt werden. Wesentlich sind hierbei der Austausch von Zeit-Informationen („Synchronisation“) und die soziale Wechselbeziehung („Interaktion“).

Im Folgenden werden diese beiden wesentlichen Arten der Wechselbeziehung genauer erörtert. Anschließend wird im speziellen auf die Interaktion von Tanz und Videoprojektion eingegangen. Hierzu werden verschiedene Beispiele zeitgenössischer Bühnenwerke herangezogen.

2.1 Synchronisation

Für die Kategorisierung verschiedener Formen der Synchronisation sollen zuallererst die allgemeinen Begrifflichkeiten erläutert werden.

Per Definition bezeichnet Synchronisation oder auch Synchronisierung „das zeitliche Aufeinander-Abstimmen von Vorgängen“[32]. Sie kann in einseitig („unidirektional“) oder wechselseitig („bidirektional“) ausfallen.

2.1.1 Unidirektionale Synchronisation

Für eine stabile Synchronisation mehrerer Systeme muss eine feste hierarchische Struktur existieren, in der jedes Element nur einen unangefochtenen Taktgeber besitzt. Vergleichbar ist dies mit Organisationsmodellen zur hierarchischen Struktur von Unternehmen. Bei digitalen Systemen wird der Taktgeber „Master“ genannt. Bei Takt-Empfängern spricht man von „Slaves“

(siehe [35]). Zur einfachen Darstellung werden hierarchische Verknüpfungen fortan mit Pfeilen dargestellt:

$$\textit{Master} \rightarrow \textit{Slave}$$

Ein Slave kann seinerseits wiederum als Master für einen oder mehrere (Sub-)Teilnehmer fungieren. Auf diese Weise können relativ komplexe Strukturen entstehen. Um Konflikte zu vermeiden sollte die Kommunikation bzw. die Befehlskette möglichst einseitig, von oben nach unten, erfolgen.

2.1.2 Bidirektionale Synchronisation

Es gibt natürlich auch Anwendungen, in denen eine adaptive, bidirektionale Synchronisation stattfindet. Hierbei muss jedoch zuvor ein für alle Teilnehmer gültiges Ziel festgelegt werden, auf dessen Erfüllung gemeinsam hingearbeitet wird (vgl. [9]).

Im Bereich der Musik existiert diese Form der Synchronisation z. B. im Zusammenspiels mehrerer Musiker eines Ensembles. Hierbei stehen die Musiker jeweils auf gleicher hierarchischer Stufe und müssen sich gemeinsam durch Anpassung und Fehlerkorrektur auf einen gemeinsamen Takt einigen. Damit eine Einigung erfolgen kann, müssen die einzelnen Teilnehmer jeweils auf das gleiche, vorher festgelegte Ziel hinarbeiten. Dieses Ziel kann z. B. ein bestimmtes ästhetisches Ideal oder eine Partitur darstellen (siehe [9]).

2.2 Interaktion

Falls in einem bidirektionalen Synchronisationsmodell kein eindeutiges, gemeinsames Ziel festgelegt wurde, aber dennoch eine bidirektionale Kommunikation, bzw. Anpassung besteht, so wird eher von „Interaktion“ gesprochen.

Der Duden definiert Interaktion als „aufeinander bezogenes Handeln zweier oder mehrerer Personen“ bzw. die „Wechselbeziehung zwischen Handlungspartnern“ [6]. Interaktion setzt in diesem Kontext also voraus, dass die Handlungspartner soziale Wesen sind. In der Informatik wird dieser Begriff jedoch auch für Wechselbeziehungen zwischen Mensch und Maschine, bzw. Computer benutzt.

Da Interaktion also immer eine Form von sozialer Beziehung bzw. Kreativität voraussetzt, kann der Begriff bei der Kommunikation zwischen zwei Maschinen nur schwer angewendet werden.

2.3 Beispiele von Interaktion bei Tanz-Aufführungen mit Videoprojektion

Durch die fortschreitende Technologisierung bei künstlerischen Darbietungen haben Synchronisierung und Interaktion zwischen Menschen und Computer-

Systemen eine hohe Wichtigkeit erlangt. Musiker, Tänzer, Schauspieler, etc. aber auch elektronische Echtzeitsysteme für Klang-Zuspielungen und Video-Projektion müssen miteinander kommunizieren und sich gegenseitig aufeinander abstimmen.

Durch die technisch bedingte, rechteckige Form von Videos, ergibt sich bei Projektion dieser auf eine Leinwand oftmals der Eindruck eines „Guckkastens“ in eine andere „Welt“. Für den Zuschauer besteht somit immer eine räumliche Trennung der beiden „Welten“, was eine emotionale Bindung erschwert. Da der Tänzer nicht in der Lage ist, diese Welt zu berühren, findet die Interaktion auf einer höheren, für das Publikum schwer erfassbaren Ebene statt.

Die folgenden Beispiele lösen dieses Problem damit, dass die Projektion nicht auf einer konventionellen Leinwand erfolgt, bzw. nutzen eine Leinwand, die das gesamte Blickfeld des Betrachters einnimmt. Anhand dieser Beispiele sollen nun verschiedene Formen der Interaktion bei multimedialen Tanz-Performances diskutiert werden.

Hierbei wird ausschließlich auf die Interaktion zwischen den Domänen „Tanz“ und „Videoprojektion“ eingegangen. Ziel ist es, eine Form der Interaktion zwischen Tanz und Video zu finden, die die Ausdrucksmöglichkeiten von reinem Tanz sinnvoll erweitern kann. Der einfachste Fall, die Synchronisation von Tanz zur Videoprojektion wird aus diesem Grund hier nicht abgehandelt.

2.3.1 D.a.v.e. - digital amplified video engine (Klaus Obermaier und Chris Haring, 1999) [20]

Einen Meilenstein in der Verknüpfung von Tanz und Videoprojektion haben Klaus Obermaier und Chris Haring im Jahr 1999 mit dem Werk *D.a.v.e.* gesetzt (siehe Abb. 2.1. Ein Video ist online verfügbar: [20]).

Der Tänzer fungiert hier unter anderem als Projektionsfläche für Körper anderer Menschen und interagiert selbst mit seinem eigenen Körper, indem er diesen augenscheinlich deformiert und modifiziert. Er ist hierbei größtenteils mit dem Rücken zum Publikum gedreht, um eine möglichst glatte Projektionsfläche zu bieten.

In Wirklichkeit besteht bei diesem Stück jedoch nur wenig echte Interaktion, da das Videomaterial vorproduziert ist und auch deren Projektion nicht adaptiv dem Tänzer folgt, sondern diesem die Choreographie quasi vorgibt. Das Video fungiert hier also als Master, an das sich der Tänzer als Slave strikt anpassen muss.

2.3.2 Solo 4 > Three (Shadows) (Frieder Weiß, 2003) [30]

Interaktion entsteht erst dann, wenn das Video durch die Bewegungen des Tänzers beeinflusst werden kann. Als Beispiel sei hier das Werk *Solo 4 >*



Abbildung 2.1: D.a.v.e. - digital amplified video engine (Klaus Obermaier und Chris Haring, 1999) [20]



Abbildung 2.2: Solo 4 > Three (Shadows) (Frieder Weiß, 2003) [30]

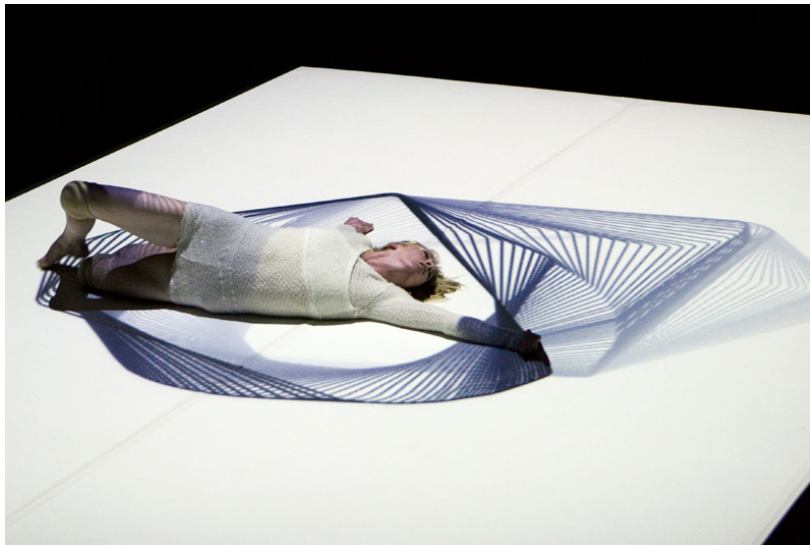


Abbildung 2.3: Glow (Chunky Move, 2008) [2]

Three (Shadows) von Frieder Weiß angeführt (siehe Abb. 2.2). Ein Video hierzu kann unter [30] abgerufen werden.

Bei diesem Stück wird die Tänzerin mit Infrarot-Technik vom Hintergrund separiert, und in Echtzeit eine Art „Schablone“ aus den Umrissen des Körpers erstellt. Diese Schablone wird in verschiedenen Farben in Lebensgröße auf eine bis zum Boden reichende Leinwand projiziert. Es entsteht der Eindruck eines leuchtenden Schattens der Tänzerin. Durch Spiegelung und Verdopplung dieser Schablone auf der Leinwand kann die Tänzerin hier direkt mit sich selbst interagieren.

Ein Problem hierbei ist, dass diese Schatten immer an die zweidimensionale Leinwand gebunden sind, und diese nicht verlassen können. Die Tänzerin muss an diese Wand herantreten, um mit dem Video zu interagieren.

2.3.3 Glow (Chunky Move, 2008) [2] [3]

Der Einsatz von Videoprojektion im modernen Tanz hat sich mittlerweile stark wegbewegt von der klassischen Projektion auf eine Leinwand. Die Projektion übernimmt immer mehr die Aufgabe einer Lichtquelle für die Tänzer, um deren visuelle Erscheinung grundlegend zu verändern.

Gideon Obarzanek, der Choreograph der australischen Tanzgruppe *Chunky Move*, wurde aufmerksam auf Frieder Weiß und startete eine intensive Zusammenarbeit (siehe [19]). Obarzanek wollte ein Stück schaffen, in dem der Tänzer nicht an der Unterkante einer Videoleinwand agiert, sondern sich direkt darin bewegen kann. Da der Tänzer nunmal durch die Schwerkraft an den Bühnenboden gebunden ist, verlagerte Obarzanek auch die Videoprojek-

tion dorthin. Dieser Trick erlaubt es dem Tänzer, sich auf dieser virtuellen Welt frei zu bewegen und intuitiv mit dem ganzen Körper damit zu interagieren. Weiters wollte Obarzanek, dass die Videoprojektion sich an die individuellen Bewegungen des Tänzers anpasst, was nur durch in Echtzeit generiertes Video möglich ist (siehe [19]).

Im Jahre 2008 entstand aus dieser Zusammenarbeit das Stück *Glow* (siehe Abb. 2.3, [3], [2]). Der Tänzer bewegt sich hier die meiste Zeit liegend auf dem Bühnenboden, um besser mit der Projektion zu „verschmelzen“. In verschiedenen Szenen besteht die Bodenprojektion jeweils aus einfachen Strukturen, die sich durch „Berührung“ mit dem Tänzer in ihrer Gestalt verändern.

Als Problem bei Bodenprojektion sei hier genannt, dass das Publikum, um etwas zu sehen, zumindest leicht schräg von oben auf die Bühne herabschauen muss, was in vielen Aufführungsorten evtl. nicht möglich ist.

Kapitel 3

Technische Realisierung einer adaptiven Videoprojektion

Es soll nun ein System entwickelt werden, das eine durch einen Tänzer selbst gesteuerte Videoprojektion ermöglicht. Die Idee ist, eine größtmögliche Synchronisierung von Videoprojektion zu den Körper-Bewegungen zu schaffen, und ihm so eine zusätzliche Ausdrucksmöglichkeit zu verschaffen.

Analog zu einem Musiker, der den Klang seines Instruments mit Live-Elektronik verändert, soll der Tänzer sein äußeres Erscheinungsbild durch Videotechnik verändern können. Diese Verwandlung ist nur möglich, wenn die Videoprojektion direkt auf seinem Körper stattfindet. Die Projektion auf eine Leinwand kommt nicht infrage, da dies den Bezug zwischen Tänzer und Videoprojektion stark beeinträchtigen würde. Das Ziel ist also ein System, das eine Verschmelzung von Video und Tänzer, wie bei *D.a.v.e.* (siehe 2.3.1) in Echtzeit ermöglicht.

Um die Projektion direkt auf einen (beweglichen) Körper bewerkstelligen zu können, muss dieser gleichzeitig von einem Sensor-System erfasst werden. Da der Tänzer nicht in seiner Bewegung eingeschränkt werden soll, kommen für dieses Tracking¹ ausschließlich optische Erfassungs-Systeme in Betracht. Mithilfe der gewonnenen Bewegungsdaten soll schließlich Bildmaterial direkt auf den Körper des Tänzers projiziert werden. Die Projektion soll schließlich mithilfe eines oder mehrerer Video-Beamer ermöglicht werden.

3.1 Einführung in die Videoprojektion

Bilder, auch in Form von bewegten Bildern (Video), werden in der heutigen Digitaltechnik meist in rechteckiger Form verarbeitet. Durch die Anordnung

¹Der Begriff „Tracking“ bedeutet in diesem Zusammenhang die automatische, maschinelle „Verfolgung von bewegten Objekten“ [39]

vieler rechteckiger Pixel² in einer 2-dimensionalen Matrix (Rastergrafik [36]) können Bildinformationen auf digitalem Wege sehr effizient verarbeitet werden. In der Videotechnik fließen viele solcher Bilder in kurzen Zeitabständen hintereinander durch das System. Auch Videoprojektoren liefern deshalb stets ein rechteckiges Bild.

Bei der Videoprojektion haben wir es also mit rechteckigen Bildern zu tun, die mithilfe eines Beamers auf eine geeignete Projektionsfläche (z. B. Leinwand) projiziert werden.

Um im Folgenden die Grundlagen der Videoprojektion zu erläutern, werden die Bilder und Beschreibungen aus [25] herangezogen.

3.1.1 Projektor-Aufstellung

Soll eine Projektion nicht dieses charakteristische rechteckige Format besitzen, so muss das maximal mögliche Rechteck, das der Projektor erreichen kann, das Zielobjekt vollkommen umschließen. Es muss also im Voraus eine geeignete Aufstellung des Projektors gewählt werden. An den überschüssigen Stellen, die der Beamer zwar noch erreichen könnte, an denen aber keine Projektion gewünscht ist, werden die Pixel schwarz eingefärbt.

Der maximale Bereich, den ein Video-Projektor ausleuchten kann, ist also eine 4-seitige Pyramide mit der Spitze im Nodalpunkt der Optik des Projektors. Das bedeutet, das Bild, das die Grundfläche der Pyramide darstellt, wird entsprechend größer, je weiter diese vom Projektor entfernt ist.

Der optische Öffnungswinkel des Linsensystems des Projektors bestimmt also das Verhältnis zwischen Abstand zur Leinwand und Größe der Projektion.

$$\text{Bildbreite} = \frac{\text{Projektionsabstand}}{\text{Projektionsverhältnis}} \quad (3.1)$$

$$\text{Bildhöhe} = \text{Seitenverhältnis} \cdot \text{Bildbreite} \quad (3.2)$$

Daraus lässt sich über die Spezifikationen eines gegebenen Beamers der nötige Abstand zum Tänzer (Höhe ca. 2 m) wie folgt errechnen (siehe A.1, [13], [40]):

$$\text{Projektionsverhältnis} = 0,49$$

$$\text{Seitenverhältnis} = 16 : 10$$

$$\text{Höhe} = 2 \text{ m}$$

²„Mit Pixel (...) werden die einzelnen Farbwerte einer digitalen Rastergrafik bezeichnet sowie die zur Erfassung oder Darstellung eines Farbwerts nötigen Flächenelemente bei einem Bildsensor beziehungsweise Bildschirm mit Rasteransteuerung.“[31]

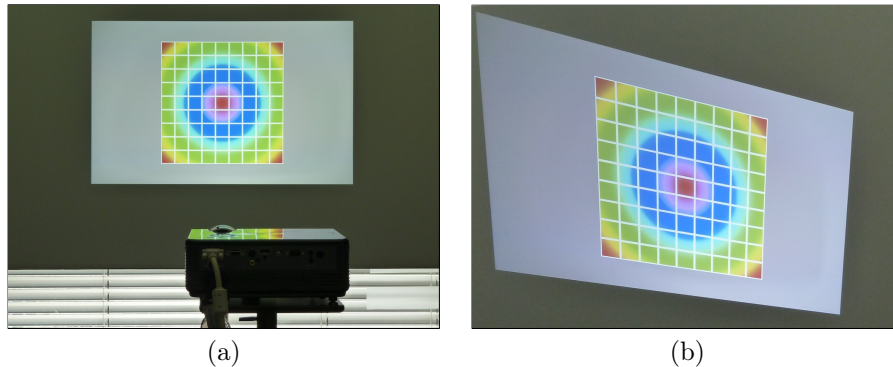


Abbildung 3.1: perfekte Beamer-Aufstellung (a) und geometrische Verzerrung bei schiefer Aufstellung [25]

$$\begin{aligned}
 \text{Abstand} &= \text{Projektionsverhältnis} \cdot \text{Seitenverhältnis} \cdot \text{Höhe} \\
 &= 0,49 \cdot \frac{16}{10} \cdot 2 \text{ m} = 1,568 \text{ m}
 \end{aligned} \tag{3.3}$$

Bei den meisten Projektoren ist die Projektionsebene normal zu dessen Ausrichtung. Um bei einer Montage an der Decke trotzdem eine an der Wand hängende Leinwand perfekt ausleuchten zu können, wird das Bild durch die Optik des Projektors um einen fixen Wert vertikal verschoben. Teurere Geräte besitzen teilweise eine aufwändige Optik, mit der das Bild innerhalb eines bestimmten Bereichs in der Projektionsebene beliebig verschoben werden kann. Dies ermöglicht eine gewisse Toleranz in der Aufstellung.

3.1.2 Entzerrung der Projektion

Wenn der Projektor in einem schiefen Winkel auf die Leinwand ausgerichtet ist, so wird das projizierte Bild geometrisch verzerrt dargestellt (vgl. Abb. 3.1 (a) und (b)). Dies lässt sich ausgleichen, indem noch vor der Projektion in der Gegenrichtung verzerrt wird (siehe Abb. 3.2). Der Maximale Bereich, den der Projektor beleuchten kann, ist in den Abbildungen hellgrau dargestellt.

Diese Verzerrung wird in der Regel durch Software gelöst. Bei Beamern existiert oft eine sog. Trapezkorrektur in horizontaler und vertikaler Richtung. Zu beachten ist bei software-basierter Korrektur jedoch, dass der maximal auszuleuchtende Bereich schon durch die Optik und Aufstellung des Beamers gegeben ist, und sich das entzerrte Bild immer innerhalb dessen Ausleuchtungs-Bereich befindet. Durch starke Korrektur wird das sichtbare Bild hierbei also zwangsläufig kleiner. Weiters ist zu beachten, dass Videoprojektoren keinen idealen Schwarzwert besitzen, und immer etwas Licht durchgelassen wird. Die Folge ist, dass evtl. ein grauer Rand um das entzerrte Bild sichtbar bleibt.

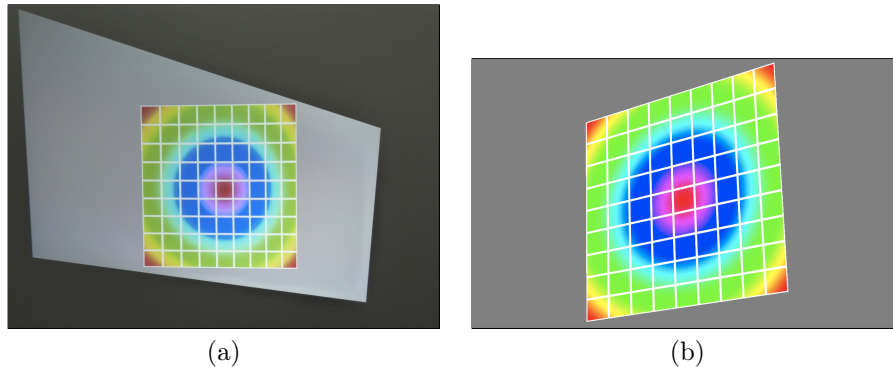


Abbildung 3.2: Entzerrte Projektion bei schiefer Beamer-Aufstellung (a) das vom Projektor ausgegebene, verzerrte Bild, um diese Projektion zu erreichen [25]

Durch eine Projektions-Entzerrung im Wiedergabesystem, außerhalb des Projektors, ergibt sich eine noch höhere Flexibilität. Durch aufwändige Software-Systeme kann Videomaterial auch für eine Projektion auf sehr komplexe Geometrien entzerrt werden. Dieses exakte Einpassen der Projektion in die Form von Objekten wird Video Mapping genannt. Sobald Videomaterial auf eine bestimmte Geometrie „gemappt“ wird, so spricht man oft auch von einer Textur.

3.1.3 Zusammenhängende Projektion mit mehreren Projektoren

Falls ein Projektor nicht ausreicht, um das Zielobjekt komplett abzudecken, können weitere Projektoren herangezogen werden. Der maximal abzudeckende Bereich sollte dann mit den anderen Projektoren leicht überlappen, um eine sanfte Überblendung zwischen den Teilprojektionen zu ermöglichen. Dieses „Soft-Edging“ verhindert im Gegensatz zum „Hard-Edging“ im Falle einer leichten Verschiebung eines Projektors, dass Schnittkanten herausstechen, die die Nahtstellen offenbaren (siehe [25]).

3.2 Marktübersicht Projektions- und Tracking-Systeme

Es ist nun erforderlich, ein geeignetes System zu finden, mit dem sich das aufwändige Mapping auf den Körper des Tänzer realisieren lässt. Das System sollte außerdem über ausreichende Schnittstellen für das Tracking verfügen, welches in Abschnitt 3.3 behandelt wird. Falls alle Aufgaben in einem einzigen System gelöst werden können, so ist dies logischerweise vorzuziehen.



Abbildung 3.3: Coolux Pandoras Box Server (a) und zugehörige, grafische Benutzeroberfläche (b) [4]

3.2.1 Hardware-basiert

Es gibt eine Vielzahl an hardware-basierten Komplettlösungen zur Entzerrung anspruchsvoller Videoprojektionen. Beispielhaft seien an dieser Stelle drei hierzulande bekannte Vertreter genannt: *Green Hippo Hippotizer/HippoCritic*³, *High End Systems Axon*⁴ und *Coolux Pandoras Box* (siehe Abb. 3.3).

Im Normalfall lassen solche Medienserver (bzw. Videoserver) im Bereich des Video Mapping kaum Wünsche offen und besitzen teilweise auch Schnittstellen und eingebaute Funktionen für das Tracking von bewegten Objekten.

Diese Geräte sind jedoch mit relativ hohen Anschaffungskosten verbunden und lassen sich nur schwer mit eigenen Systemen individuell erweitern. Der Benutzer hat hier keinen Einblick in die interne Datenverarbeitung.

3.2.2 Software-basiert

Mittlerweile existieren einige Software-Lösungen speziell für Video Mapping. Aber auch allgemeinere Programmier-Umgebungen aus dem Bereich der Medienkunst besitzen Erweiterungen hierzu, oder lassen sich zumindest in dieser Richtung leicht erweitern.

Proprietäre Software

Proprietäre Software besitzt den Vorteil, dass die eingebauten Funktionen in der Regel stabil und effizient funktionieren. Außerdem kann ein professioneller Support von Vorteil sein. Leider mangelt es aber oft an Schnittstellen oder Möglichkeiten, eigene individuelle Anpassungen für spezielle Vorhaben vorzunehmen.

³Green Hippo Media Servers: <http://www.green-hippo.com/>

⁴High End Systems - Axon: <http://www.highend.com>

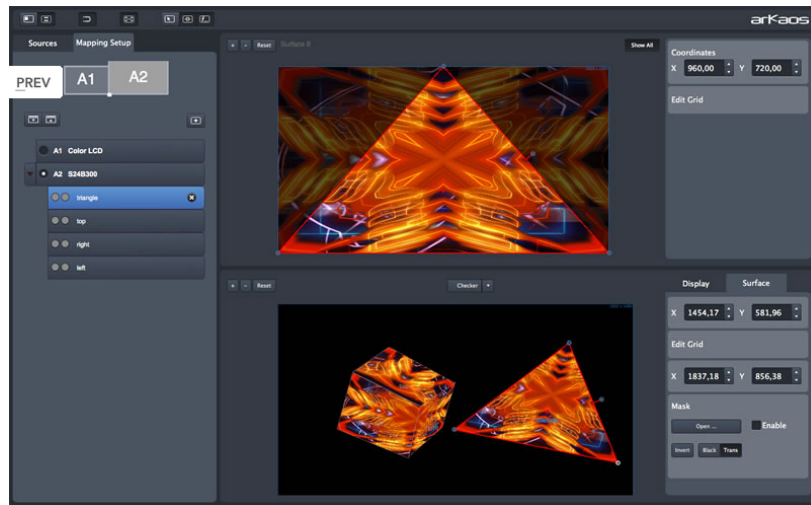


Abbildung 3.4: Arkaos MediaMaster Pro [1]

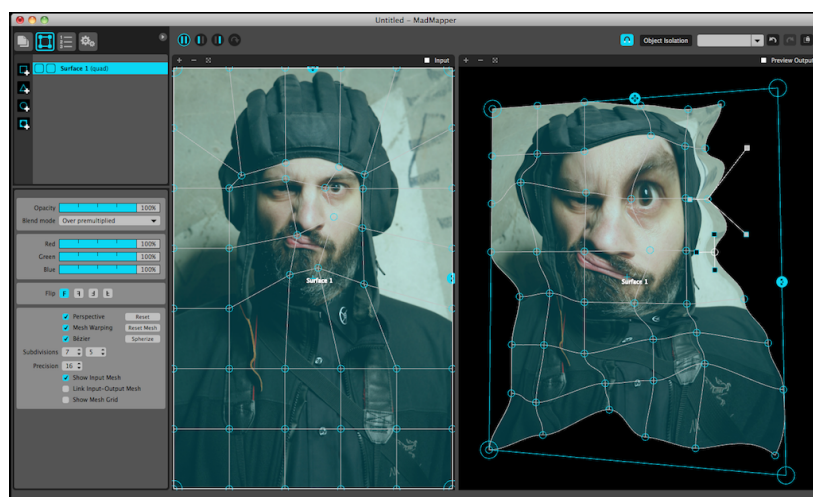


Abbildung 3.5: MadMapper [14]

MadMapper [14] & Arkaos MediaMaster [1] Zwei relativ erfolgreiche Vertreter für Video Mapping sind *MadMapper* (s.Abb. 3.5) und *Arkaos MediaMaster* (s.Abb. 3.4).

Sie übertragen im Grunde die Möglichkeiten von Medienservern auf den PC und laufen auf standard Computer Hardware. Eine Zuspiegelung von Videomaterial ist in Echtzeit über die *Syphon* Schnittstelle möglich. Diese Software erlaubt einen Austausch von Bildmaterial zwischen Applikationen auf Hardware-Ebene. Das Framework ist zwar frei und quelloffen, jedoch leider nur für *Mac OS X* erhältlich (siehe [23]).

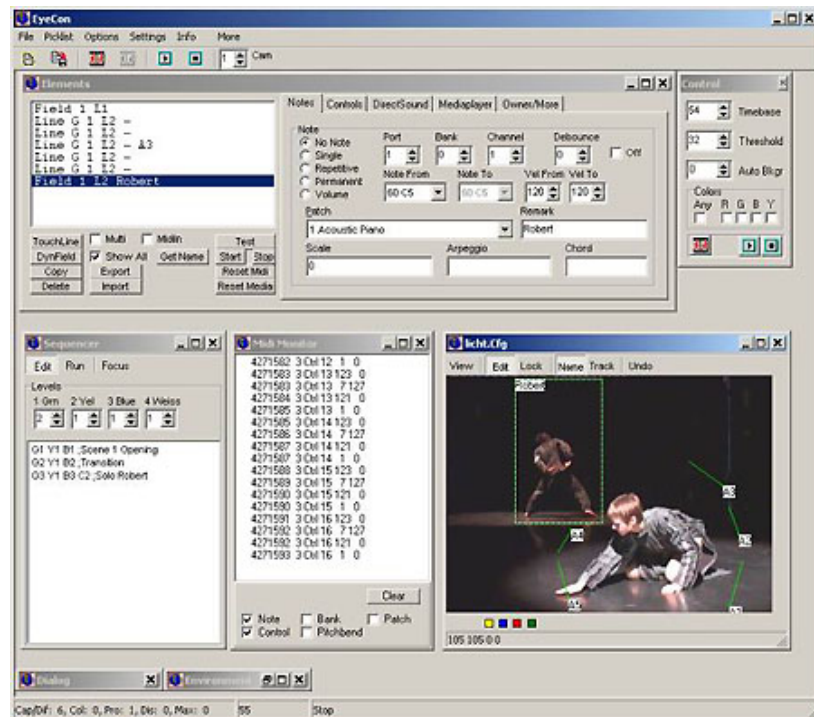


Abbildung 3.6: EyeCon [29]

Eine Möglichkeit des Tracking wäre also über externe Software theoretisch möglich.

EyeCon [29] Eine vielversprechende Lösung für das ausgelagerte Tracking stellt die Software EyeCon (s.Abb. 3.6) dar. Die Software wurde von Frieder Weiß eigens für interaktive Installationen und Tanzperformances entwickelt. Es wird außerdem eine Vielzahl an Tracking-Techniken unterstützt.

Max/MSP [5] Dazu käme z.B. *Max/MSP* (s.Abb. 3.7) infrage, welches die nötigen Schnittstellen für aufwendiges Tracking bereitstellt. Diese Software verfügt aber sogar selbst über ausreichende Erweiterungen für Video Mapping, womit sich der Einsatz spezieller Software hierfür erübrigen würde. Durch die graphische Programmierungsumgebung und die leichte Erweiterbarkeit mit eigenen PlugIns, sog. Externalen, ist diese Software prädestiniert für die schnelle Implementierung interaktiver Computersysteme.

Freie Software

Pure Data [21] Für *Max/MSP* besteht allerdings auch eine freie, quelloffene Alternative: *Pure Data* (Abk.: *Pd*) (s.Abb. 3.8).

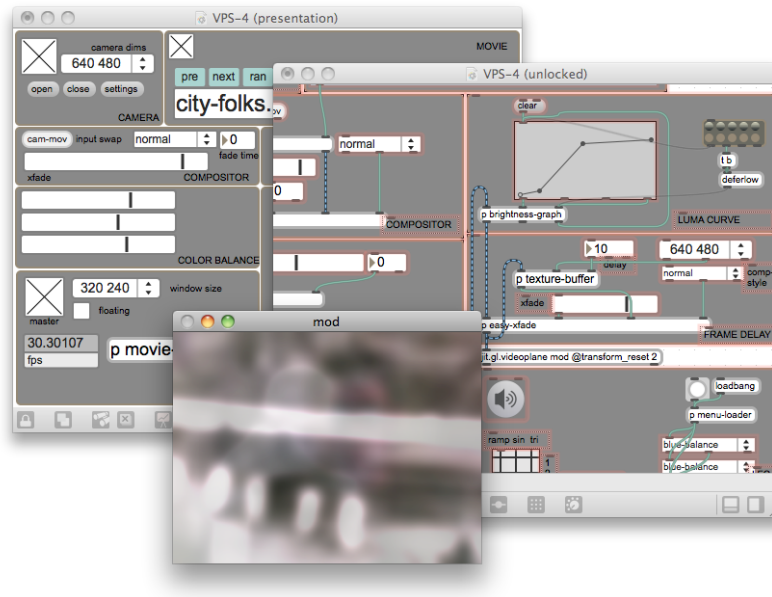


Abbildung 3.7: Cycling '74 Max/MSP mit Jitter [5]

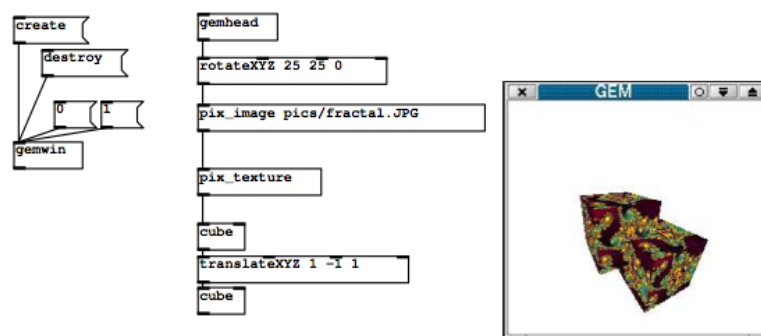


Abbildung 3.8: Pure Data mit dem GEM External [41]

Pd wurde, wie *Max/MSP*, ursprünglich für die Echtzeit-Verarbeitung von Audio-Daten entwickelt, ist jedoch auch individuell durch Externals um weitere Funktionalität erweiterbar. Mit dem *GEM* External (s. [8] und [41]) lassen sich hardware-beschleunigte Video-Anwendungen realisieren. *GEM* nutzt *OpenGL*⁵ für die Darstellung. *Pd* ist mit den meisten verbreiteten Betriebssystemen und Architekturen kompatibel. Für *Pd* existiert außerdem eine umfangreiche Lösung zur Realisierung von aufwendigen Video-Projektionen, namens *Extended View Toolkit* (s. [25] und [24]).

⁵OpenGL - Open Graphics Library: <http://www.opengl.org>

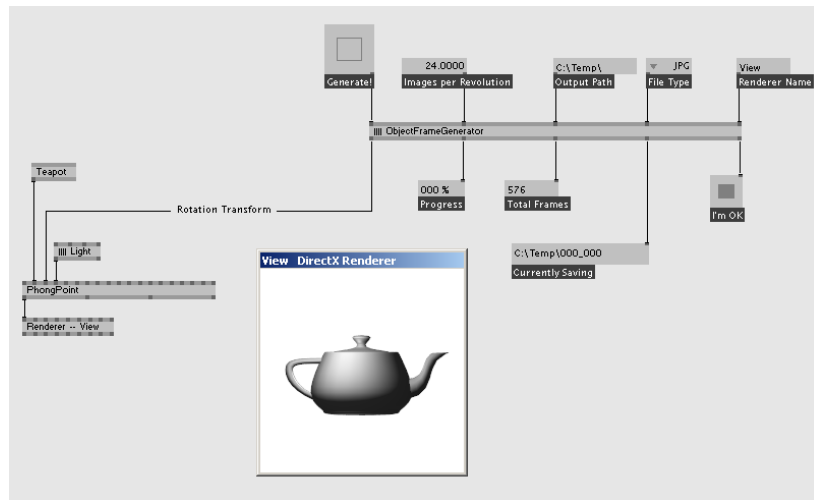


Abbildung 3.9: vvvv - a multipurpose toolkit [27]

vvvv [27] Da keine Audioverarbeitung benötigt wird, kommt die speziell auf Videoverarbeitung abzielende Software *vvvv* (s. Abb. 3.9) in Betracht. Wie *Pd* und *Max/MSP* handelt es sich hier um eine graphische Datenfluss-Programmierungsumgebung. *vvvv* basiert auf der hocheffizienten *Microsoft DirectX Graphics*[15] Schnittstelle, und ist deshalb ausschließlich für *Microsoft Windows* Betriebssysteme erhältlich. Für nicht-kommerzielle Zwecke ist *vvvv* gratis erhältlich. Diese Software beinhaltet viele Erweiterungen für Tracking und aufwändige unkonventionelle Video-Mappings, und ist deshalb für viele in diesem Bereich die erste Wahl.

3.3 Marktübersicht: Hardware für optisches Tracking

Es existiert mittlerweile eine Vielzahl an optischen Tracking-Systemen für verschiedenste Anwendungsbereiche. Die in Betracht kommenden Lösungen sollen nun auf ihre Tauglichkeit für dieses Projekt untersucht werden.

3.3.1 Kamera

Die billigste Lösung scheint die Erfassung mittels einer handelsüblichen Kamera zu sein. USB- und FireWire-Kameras für den PC sind weltweit verfügbar und günstig in der Anschaffung. Es existieren einige Software-Lösungen zur Separierung von bewegten Objekten vor einem statischen Hintergrund (Background Subtraction). Zur reinen Erfassung des Körpers würde dieser Ansatz also prinzipiell funktionieren.

Da dieses System jedoch mit Licht im sichtbaren Bereich arbeitet, würde das Tracking erheblich durch die Videoprojektion und eventuelle Beleuch-

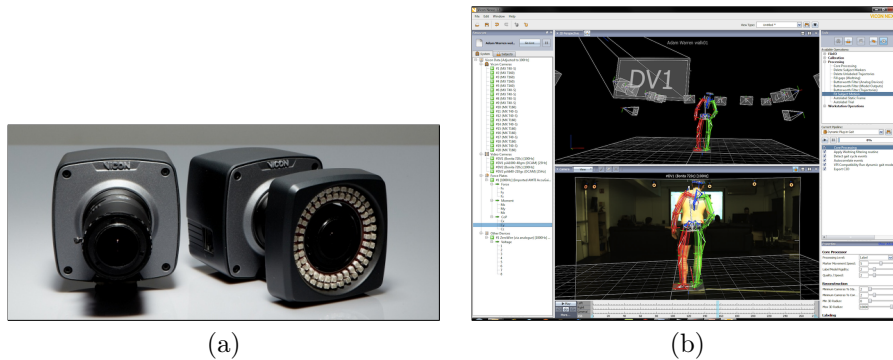


Abbildung 3.10: Vicon Kamera-System (a) und grafische Benutzeroberfläche der Software (b) [26]

tung gestört.

3.3.2 Passive Infrarot-Systeme

Die Probleme, die mit Kameras im sichtbaren Lichtbereich auftreten, können durch Infrarot-basierte Lösungen eliminiert werden.

Um eine Szenerie durch Infrarot-Kameras zu überwachen, muss logischerweise ausreichend im Infrarotbereich ausgeleuchtet werden. Für eine Separierung vom Hintergrund muss sich der Tänzer im Infrarotbereich stark von diesem hervorheben. D.h. der Tänzer ist stark infrarot reflektierend und der Hintergrund stark absorbierend, oder umgekehrt. Dies wird bei spezialisierten Tracking-Systemen wie etwa *Vicon Tracker* (s.Abb. 3.10) durch Reflektoren erreicht, die auf das bewegliche Objekt angebracht werden (s.Abb. 3.11).

Ein solches System stünde im *Cube* des *Institut für Elektronische Musik und Akustik (IEM)* in Graz zur Verfügung⁶. Das zu entwickelnde System soll jedoch aus ästhetischen Gründen möglichst auf am Tänzer angebrachte Leuchtpunkte verzichten. Außerdem wäre der Aufbau dieser Tracking-Lösung bei externen Veranstaltungsorten kompliziert und Zeitaufwendig.

3.3.3 Wärmebildkamera

Wärmebildkameras nutzen im Gegensatz zu herkömmlichen Infrarotkameras eine höhere Wellenlänge. Während Infrarotkameras eher kurzwelliges Infrarot detektieren, lässt sich mithilfe der Thermografie mittleres und langwelliges Infrarot aufnehmen (s.Abb. 3.12 und [38]).

Ein Problem hierbei ist, dass der Bühnenbereich mit heißen Geräten, wie Scheinwerfern, Projektoren, etc. ausgestattet ist, die das Bild evtl. stören

⁶Institut für Elektronische Musik und Akustik - IEM: <http://iem.kug.ac.at/>



Abbildung 3.11: Tänzerin mit Infrarot-Reflektoren für das Vicon Tracking System [18]



Abbildung 3.12: Thermografische Aufnahme der Körpertemperatur einer Frau [38]

könnten. Weiterer Nachteil sind die hohen Kosten, die mit der Anschaffung eines solchen Geräts verbunden sind.

3.3.4 Structured Light Coding

Structured Light Coding ist ein aktives 3D-Scan Verfahren, und arbeitet im einfachsten Fall mithilfe eines Projektors und eines leicht versetzt dazu plat-

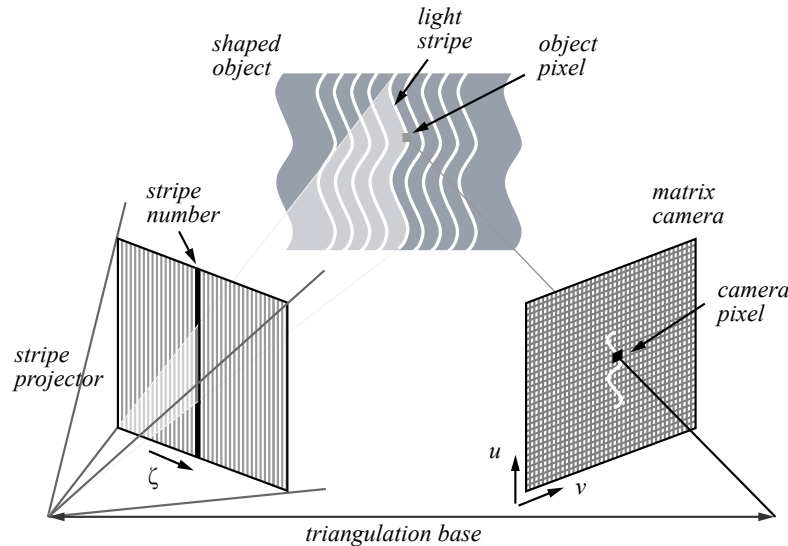


Abbildung 3.13: Prinzip eines Structured Light 3D-Scanner [37]

zierten Kamerasystems. Der Projektor sendet ein bestimmtes geometrisches Pattern aus, z. B. in Form von Streifen oder Punkten, das mit dem leicht versetzt dazu platzierten Kamerasystem aufgenommen wird. Durch Triangulation kann für einzelne Punkte die Entfernung errechnet werden (s. Abb. 3.13 und [37]).

Einige Medienserver sowie Software wie *MadMapper* und *vvvv* besitzen Funktionen, um 3D-Scans mithilfe handelsüblicher Beamer und Webcams zu realisieren. Da sich dies aber bei einem Echtzeit-Einsatz mit der Projektion überschneiden würde, kommen nur Infrarot-basierte Systeme infrage. Projektoren für den Infrarotbereich sind zwar schwer erhältlich, doch es gibt bereits All-in-one Lösungen, die Projektor und Kamera integrieren.

Kinect [17]

Ursprünglich für die berührungslose Bewegungssteuerung der *Xbox* Spielkonsole entwickelt, hat sich der *Microsoft Kinect* Sensor (s. Abb. 3.14) als günstiger 3D-Scanner etabliert.

Das Gerät misst das Tiefenbild mithilfe eines Infrarot-Laser, der ein spezielles Punkt-Muster projiziert. Die eingebaute Infrarot-Kamera erfasst die Reflexion dieses Patterns aus einem leicht versetzten Blickwinkel. Ein direkt im Gerät befindlicher DSP berechnet die sog. Punktwolke (Point Cloud) mit den Distanz-Informationen mittels Korrelation mit gespeicherten Referenz-Patterns (siehe [12]). Der Sensor hat eine Auflösung von 640x480 Pixeln.

Es existieren eine Reihe von Externalen für *Pure Data* und *vvvv*, um die

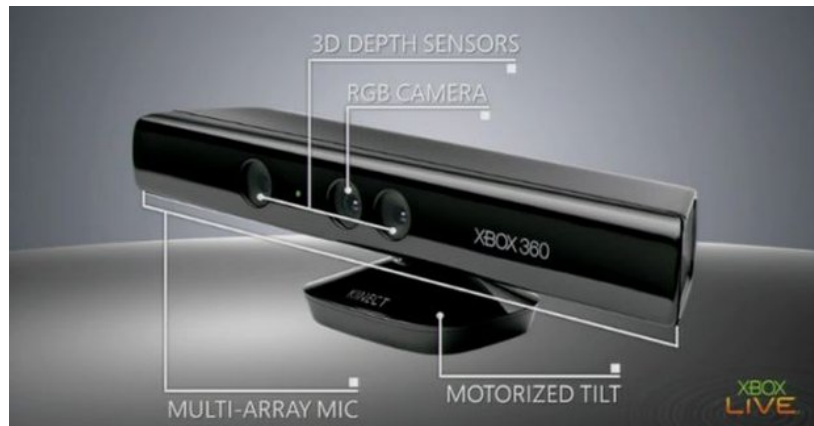


Abbildung 3.14: Microsoft Xbox Kinect Sensor [17]

Entfernungsdaten in Form eines Videobildes weiter zu verarbeiten. Auch *EyeCon* unterstützt die *Kinect* als Eingabequelle.

Ein nahezu identisches und software-seitig weitgehend kompatibles Produkt liefert die Firma *Asus* unter dem Namen *Xtion PRO*⁷.

3.4 Implementierung der adaptiven Videoprojektion

Anhand des bisherigen Wissens kann nun mit der Implementierung des Systems begonnen werden. Einleitend möchte ich noch einmal das Ziel erläutern:

Es soll ein System entwickelt werden, mit dem die Projektion auf den Körper eines Tänzers adaptiv zu seinen Bewegungen möglich ist. Das System soll möglichst mit freier Software realisiert werden, und bei maximaler Funktionalität möglichst preiswert und zukunftssicher sein. Es soll auf einem handelsüblichen PC unter *Microsoft Windows*, *Mac OS X* und *Linux* Betriebssystemen ausgeführt werden können.

Da das System evtl. in Zukunft um interaktive Klangsynthese erweitert werden soll, und bevorzugt alles innerhalb einer einzigen Software stattfindet, wurde die Programmierumgebung *Pure Data* für die Implementierung ausgewählt. Das Tracking wird aufgrund der einfachen Handhabung und des geringen Preises mit *Microsoft Kinect* realisiert. Außer für die *Kinect* Integration sollen möglichst keine Externalen in *Pd* verwendet werden. Die aufwändigen Grafikberechnungen sollen auf der Grafikkarte ausgelagert werden. Hierzu kommt *OpenGL Shading Language*⁸ zum Einsatz, da sich solche

⁷Asus Xtion PRO: <http://www.asus.de/>

⁸„Die OpenGL Shading Language (kurz: GLSL oder glSlang) ist eine C-ähnliche Programmiersprache, um mittels OpenGL auf dem Grafikprozessor eigene Programme, sogenannte Shader, auszuführen.“ [34]

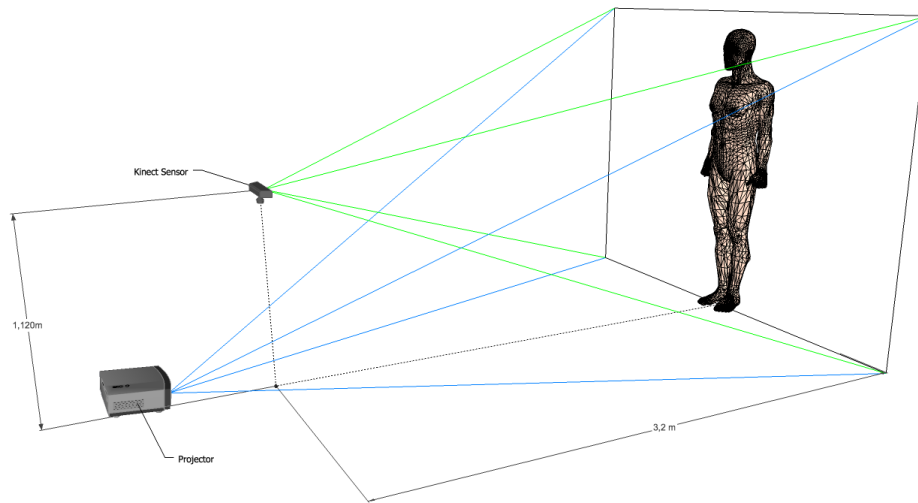


Abbildung 3.15: Technischer Aufbau des Systems

sogenannten Shader einfach in *Pure Data* und *GEM* integrieren lassen.

Die Idee für das Projektions-System ist nun Folgende: Durch die *Kinect* wird ein Tiefenbild des Bühnenbereichs mit dem Tänzer erstellt. Innerhalb dieses Bildes wird nun ein Abstandsbereich festgelegt, innerhalb dessen die Projektion stattfinden soll. Anschließend wird für alle Pixel, die sich innerhalb des festgelegten Bereichs befinden, der RGBA-Wert $[0\ 0\ 0\ 0]$ gesetzt, und für alle außerhalb befindlichen Pixel der Wert $[0\ 0\ 0\ 1]$. Somit entsteht eine schwarze Schablone, in der ein bestimmter Abstandsbereich „durchsichtig“ bzw. „ausgeschnitten“ ist. Wird diese Schablone nun vor ein Bild gelegt, so scheint dieses an Stelle des Tänzers durch. Der Tänzer wird also vereinfacht als ebene Fläche angenommen, damit die Berechnung im 2D-Bereich erfolgen kann.

3.4.1 Hardware und Verkabelung

Die *Kinect* wird über USB an den PC angeschlossen. Über aktive USB-Verlängerungen liefert er noch bei 20 m Kabellänge ein einwandfreies Signal. Sie kann über einen Adapter auf gewöhnlichen Kamera- oder auch Mikrofon-Stativen angebracht werden. Durch Tests hat sich ein Aufbau wie in Abb. 3.15 etabliert. Der Projektor steht dabei auf dem Bühnenboden, um einen schnellen Auf- und Abbau zu garantieren und dem Publikum nicht die Sicht zu versperren.

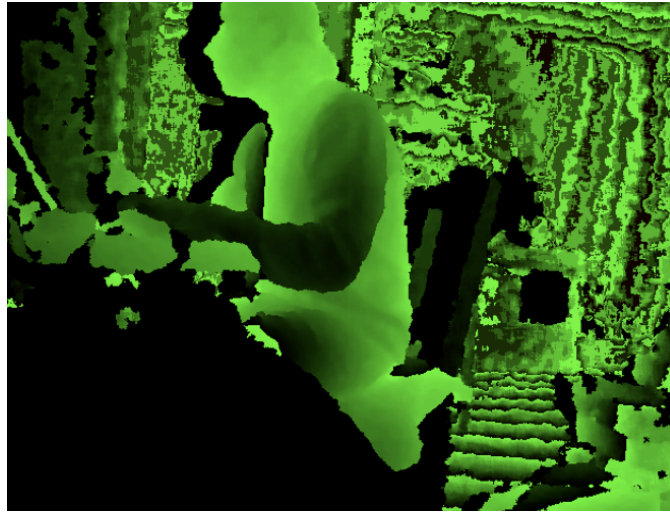


Abbildung 3.16: Tiefenbild aus *pix_freenect* im Raw-Modus

3.4.2 Das Tiefenbild

Um das Tiefenbild des Kinect Sensor in *Pd* verarbeiten zu können, wird das *pix_freenect* External benutzt (siehe [10]). Die Distanz-Information wird vom Sensor als Videostream mit 11 bit Distanz-Werten pro Pixel übergeben. Da *GEM* mit 8 bit Werten je Farbkanal arbeitet (RGBA), schreibt *pix_freenect* die ersten 8 bit in den Rot-Kanal und die restlichen 3 bit in den Grün-Kanal eines RGBA-Videostreams (siehe [12]). Wie in Abb. 3.16 ersichtlich, lässt sich diese Repräsentation mit bloßem Auge nur schwer Informationen herauslesen.

Dieser Videostream kann nun mit einfachen Farb-Operationen weiterverarbeitet werden. Es existieren zwar zum jetzigen Zeitpunkt keine Externals zur umfassenden Bearbeitung dieser Daten, es ist jedoch möglich, eigene Shader in *Pd* zu integrieren, die die Farbinformationen verarbeiten.

3.4.3 OpenGL Shading Language

Es gibt vier unterschiedliche Typen von GLSL Shadern, wobei in diesem Fall nur der Fragmentshader von Bedeutung ist. Er wird immer zuletzt ausgeführt. Soll von der Applikation ein Primitiv gezeichnet werden, so füllt der Fragmentshader die Fragmente (Pixel) dieses Objekts mit Farben einer Textur (siehe [34]).

Da für dieses Projekt nur 2-dimensionale Bildverarbeitung nötig ist, wird der Fragmentshader hier nur auf rechteckige Bilder angewendet. Abb. 3.17 zeigt, wie die Shader in *Pure Data* eingebunden werden können.

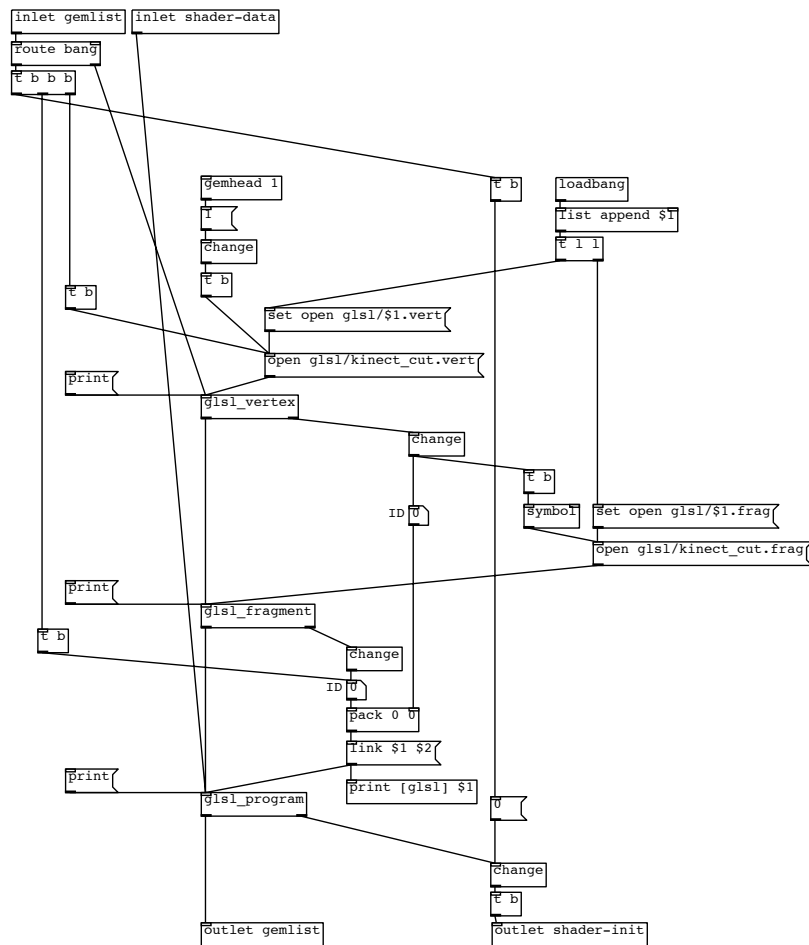


Abbildung 3.17: Einbinden von OpenGL Shadern in Pd

3.4.4 Berechnung der Schablone

Zuallererst müssen aus den Farbwerten der einzelnen Pixel deren *reale* Koordinaten im Raum berechnet werden.

Der Koordinatenursprung wird im Nodalpunkt des *Kinect* Sensor festgelegt. Für die Berechnung soll ein Fragment Shader eingesetzt werden, der die Farbwerte der einzelnen Pixel ausliest und auswertet, und die Pixel schließlich aufgrund der Berechnungen neu einfärbt, um die Schablone zu erstellen. Die z-Achse wird in Blickrichtung des Sensors definiert. *pix_freeneck* kodiert die z-Komponente schon in Millimeter in den Farbwerten. Dieser Wert muss also nur noch aus dem Rot- und Grün-Kanal zusammengesetzt werden:

```
1 // z component
2 real.z = color.r * 65536.0 + color.g * 256.0;
```

Die realen x- und y-Koordinaten können über den horizontalen bzw. ver-

tikalen Aufnahmewinkel (field of view) des *Kinect* Sensor bestimmt werden. Diese Werte können dem offiziellen *Microsoft Kinect for Windows SDK* entnommen werden [16]:

$$\begin{aligned} fov_H &= 1.0144686707507438 \text{ rad} \\ fov_V &= 0.78980943449644714 \text{ rad} \\ x_{real} &= \left(\frac{x_{pos}}{640} - 0.5\right) \cdot z_{real} \cdot \tan \frac{fov_H}{2} \cdot 2 \\ y_{real} &= \left(0.5 - \frac{y_{pos}}{480}\right) \cdot z_{real} \cdot \tan \frac{fov_V}{2} \cdot 2 \end{aligned} \quad (3.4)$$

Der *GLSL* Code sieht demnach folgendermaßen aus:

```
1 // x component
2 float FovH = 1.0144686707507438;
3 float XtoZ = tan(FovH / 2.0) * 2.0;
4 real.x = ((pos.x / 640.0) - 0.5) * real.z * XtoZ;
5
6 // y component
7 float FovV=0.78980943449644714;
8 float YtoZ = tan(FovV / 2.0) * 2.0;
9 real.y = (0.5 - (pos.y / 480.0)) * real.z * YtoZ;
```

Da nun für jeden Pixel die realen, 3-dimensionalen, kartesischen Koordinaten bekannt sind, können nun Grenzen für die Schablone gesetzt werden. Für jede der drei Richtungen wird durch zwei Werte ein quaderförmiger Bereich festgelegt, in dem die Projektion sichtbar sein soll:

```
1 if ((real.x >= xmin) && (real.x <= xmax) && (real.y >= ymin)
2 && (real.y <= ymax) && (real.z >= zmin) && (real.z <= zmax))
3     {
4         gl_FragColor = color_inside;
5     }
6 else
7     {
8         gl_FragColor = color_outside;
9     }
```

Die beiden variablen Farbwerte *color_inside* und *color_outside* sind RGBA-Werte und werden von *Pd* als Variablen gesetzt. Die Grenzen des virtuellen Quaders *xmin*, *xmax*, *ymin*, *ymax*, *zmin* und *zmax* sind Millimeter-Maße, die ebenfalls von außen gesetzt werden.

Wird für *color_inside* weiß, also der RGBA-Wert [1 1 1 1] gewählt, und für *color_outside* schwarz (RGBA: [0 0 0 1]), so ergibt sich ein Bild, wie in Abb. 3.18.



Abbildung 3.18: Rohe Schablone

3.4.5 Ausgleich von schiefer Kinect-Position

Bisher konnte durch die Grenzen nur ein Quader bestimmt werden, dessen Kanten parallel zu den Koordinatenachsen verlaufen. Um eine schiefe Aufstellung der *Kinect* zu ermöglichen, ist es sinnvoll, diesen Quader in drei Freiheitsgraden drehen zu können.

Hierzu werden die realen Koordinaten mithilfe dreier *Drehmatrizen* (siehe [33]) transformiert:

```
1 mat3 xrot = mat3 (  
2   1.0, 0.0, 0.0,  
3   0.0, cos(xangle), sin(xangle),  
4   0.0, -sin(xangle), cos(xangle)  
5 );  
6  
7 mat3 yrot = mat3 (  
8   cos(yangle), 0.0, -sin(yangle),  
9   0.0, 1.0, 0.0,  
10  sin(yangle), 0.0, cos(yangle)  
11 );  
12  
13 mat3 zrot = mat3 (  
14   cos(zangle), sin(zangle), 0.0,  
15   -sin(zangle), cos(zangle), 0.0,  
16   0.0, 0.0, 1.0  
17 );  
18  
19 real = real * xrot * yrot * zrot;
```

xangle, *yangle* und *zangle* sind hierbei die von *Pd* aus setzbaren Neigungswinkel um die x-, y- und z-Achse.

3.4.6 Glättung

Wie in Abb. 3.18 zu sehen ist, ergibt sich durch die harten Grenzen ein stark „ausgefranst“ Rand, der bei bewegtem Bild ein schnelles „Flimmern“ hervorruft. Auch stellt sich durch Mess- und Rechenfehler teilweise ein relativ starkes Bildrauschen ein, das sich in einem Flimmern der eigentlich geschwärzten Anteile der Schablone bemerkbar macht.

Es wird nun versucht, diese Fehler durch einen Tiefpass herauszufiltern. Ein solcher wird in der Bildtechnik üblicherweise durch Gauß'sche Unschärfe implementiert.

Der Chip-Hersteller *Advanced Micro Devices, Inc. (AMD)* empfiehlt für hohe Effizienz bei Shader-Programmierung eine zweistufige Implementierung von Filtern (siehe [22]). Hierbei wird ein einfacher Algorithmus zuerst in horizontaler und anschließend in vertikaler Richtung, ausgeführt.

Da Gauß'sche Unschärfe ein sehr verbreiteter Standard-Filter ist, finden sich im Internet zahlreiche freie Implementierungen. Für dieses Projekt wurde folgender Ansatz gewählt (vertikale Richtung) (siehe [7]):

```
1 void main(void)
2 {
3   vec2 pos = (gl_TextureMatrix[0] * gl_TexCoord[0]).st;
4   vec4 sum = vec4(0.0);
5
6   // blur in y (vertical)
7   // take nine samples, with the distance blur_size between them
8   sum += texture2DRect(MyTex, vec2(pos.x, pos.y - 4.0*blur_size)) * 0.05;
9   sum += texture2DRect(MyTex, vec2(pos.x, pos.y - 3.0*blur_size)) * 0.09;
10  sum += texture2DRect(MyTex, vec2(pos.x, pos.y - 2.0*blur_size)) * 0.12;
11  sum += texture2DRect(MyTex, vec2(pos.x, pos.y - blur_size)) * 0.15;
12  sum += texture2DRect(MyTex, vec2(pos.x, pos.y)) * 0.16;
13  sum += texture2DRect(MyTex, vec2(pos.x, pos.y + blur_size)) * 0.15;
14  sum += texture2DRect(MyTex, vec2(pos.x, pos.y + 2.0*blur_size)) * 0.12;
15  sum += texture2DRect(MyTex, vec2(pos.x, pos.y + 3.0*blur_size)) * 0.09;
16  sum += texture2DRect(MyTex, vec2(pos.x, pos.y + 4.0*blur_size)) * 0.05;
17
18  gl_FragColor = sum;
19 }
```

Analog hierzu wird anschließend ein zweiter Shader in horizontaler Richtung gerechnet.

Das gefilterte Bild sieht zwar generell schon etwas glatter aus, das Flimmern ist jedoch nicht komplett verschwunden. Die Schablone wird durch den Filter offensichtlich nur verwischt: Die Umrisse der Schablone werden stark vergrößert, d. h. die Projektion wird etwas über den Rand überstehen. Wenn also geglättet werden muss, so sollte dies ausschließlich „nach Innen“ geschehen.

In einer erweiterten Version vergleicht der Shader nach der Rechnung, ob der Pixel heller oder dunkler würde. Der neue Farbwert wird nur dann angewendet, wenn die neue Pixelfarbe dunkler als vorher ist:

```

1 void main(void)
2 {
3     vec2 pos = (gl_TextureMatrix[0] * gl_TexCoord[0]).st;
4     vec4 sum = vec4(0.0);
5     vec4 original = texture2DRect(MyTex, vec2(pos.x, pos.y));
6     float sum2 = 0.0;
7
8     // blur in y (vertical)
9     // take nine samples, with the distance size between them
10    sum += texture2DRect(MyTex, vec2(pos.x, pos.y - 4.0*shadow_size)) * 0.05
        * 2.0;
11    sum += texture2DRect(MyTex, vec2(pos.x, pos.y - 3.0*shadow_size)) * 0.09
        * 2.0;
12    sum += texture2DRect(MyTex, vec2(pos.x, pos.y - 2.0*shadow_size)) * 0.12
        * 2.0;
13    sum += texture2DRect(MyTex, vec2(pos.x, pos.y - shadow_size)) * 0.15 *
        2.0;
14    sum += texture2DRect(MyTex, vec2(pos.x, pos.y)) * 0.16 * 2.0;
15    sum += texture2DRect(MyTex, vec2(pos.x, pos.y + shadow_size)) * 0.15 *
        2.0;
16    sum += texture2DRect(MyTex, vec2(pos.x, pos.y + 2.0*shadow_size)) * 0.12
        * 2.0;
17    sum += texture2DRect(MyTex, vec2(pos.x, pos.y + 3.0*shadow_size)) * 0.09
        * 2.0;
18    sum += texture2DRect(MyTex, vec2(pos.x, pos.y + 4.0*shadow_size)) * 0.05
        * 2.0;
19
20    sum2 = clamp(sum.a, 0.0, 1.0);
21
22    if (sum2 > original.a)
23    {
24        gl_FragColor = vec4(sum.r, sum.g, sum.b, sum2);
25    }
26    else
27    {
28        gl_FragColor = original;
29    }
30 }

```

In Tests mit dem *Kinect* Sensor zeigt sich, dass durch diesen Shader sowohl das Bildrauschen verschwindet, als auch die ausgefransteten Kanten der Projektion schön nach Innen abgerundet werden (siehe Abb. 3.19).

3.4.7 Einbinden der OpenGL Shader in Pd

In Abb. 3.20 wird gezeigt, wie die Shader in Pd eingebunden wurden. Um schnell unterschiedliche Shader hintereinander schalten zu können, wurde die Abstraktion *gls-buf* implementiert (Innerer Aufbau siehe 3.21). Hier wird eine durch eine Textur-ID spezifizierte Textur durch eine Fragment-Shader modifiziert und auf einem Rechteck in einen neuen Framebuffer gezeichnet. Dessen Textur-ID kann nun an ein nächstes Modul weitergegeben werden. *kinect_cut* ist der Shader zum Erstellen der Schablone, während *shadow* den



Abbildung 3.19: Gefilterte Schablone

angepassten Glättungsfilter beinhaltet (s.Abb. 3.20).

3.4.8 Füllen der Schablone

Nun muss die Schablone mit Bildmaterial gefüllt werden. Um eine exakte Einpassung des Bildes zu ermöglichen, soll dieses hinter der Schablone frei transformiert werden können. Hierzu werden die Abstraktionen und Shader aus dem *Extended View Toolkit (EVT)* verwendet. Abb. 3.22 zeigt den grundsätzlichen Aufbau. Die aus Abb. 3.20 erhaltene Schablonen-Textur und die zu projizierende „Füllung“ werden auf frei transformierbare Rechtecke gemappt. Das Ganze wird in einen neuen Framebuffer gerendert, der nun weiter verarbeitet werden kann.

3.4.9 Entzerrung der Projektion

Der Framebuffer wird anschließend auf ein neues Modul gerendert, durch dessen Verzerrung die Projektion im Gesamten justiert werden kann, um eine schiefe Projektor-Ausrichtung zu korrigieren.

3.4.10 Farbanpassung

Um bei unterschiedlichen Kostüm- und Hautfarben des Tänzers eine möglichst korrekte Farbdarstellung zu erhalten, wurde eine Funktion zur Farbanpassung der Video-Ausgabe eingebaut. Um die ohnehin schon stark beanspruchte CPU nicht weiter zu belasten, wurde auch dies über einen Shader

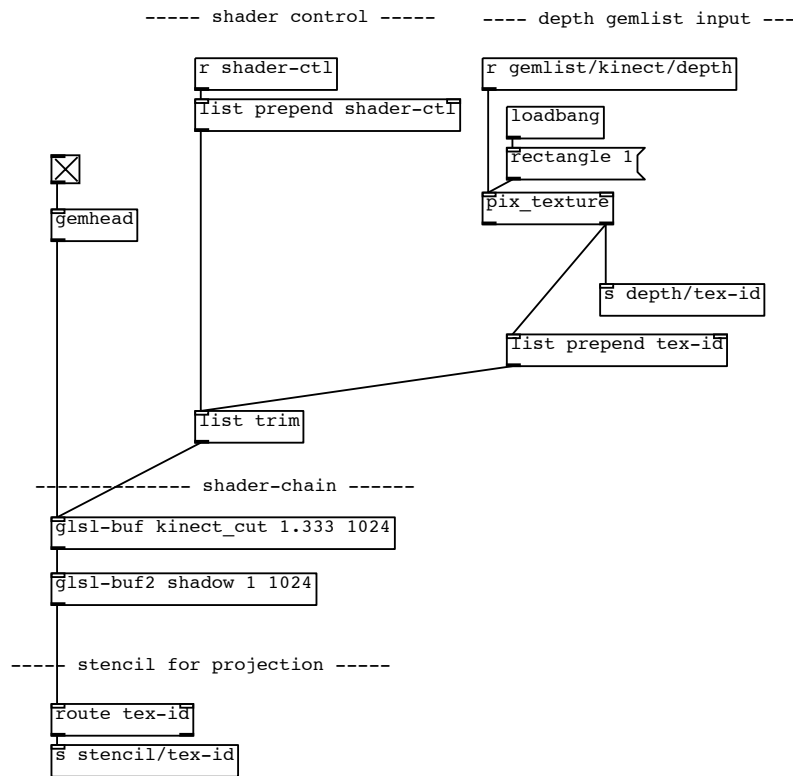


Abbildung 3.20: Die Implementierung der Tiefenbild-Verarbeitung in Pd

auf die Grafikkarte ausgelagert.

3.4.11 Live Korrektur

Da der Tänzer als zweidimensionale Fläche angenommen wird, ergeben sich bei unterschiedlichen Abständen zum Projektor Verzerrungen der Projektion.

Um bei einer Aufführung schnell korrigierend eingreifen zu können, wurden einige einfache Transformationen implementiert (s.Abb. 3.18). Die gesamte Projektion kann in x- und y-Richtung verschoben und skaliert werden. Außerdem ist die Möglichkeit einer Rotation vorhanden.

3.4.12 Grafische Benutzeroberfläche

Für eine benutzerfreundliche Bedienung wurde eine grafische Bedien-Oberfläche implementiert (s.Abb. 3.24). Die Parameter sind außerdem über MIDI durch Hardware-Controller steuerbar.

Um während einer Aufführung die Möglichkeit zur optischen Kontrol-

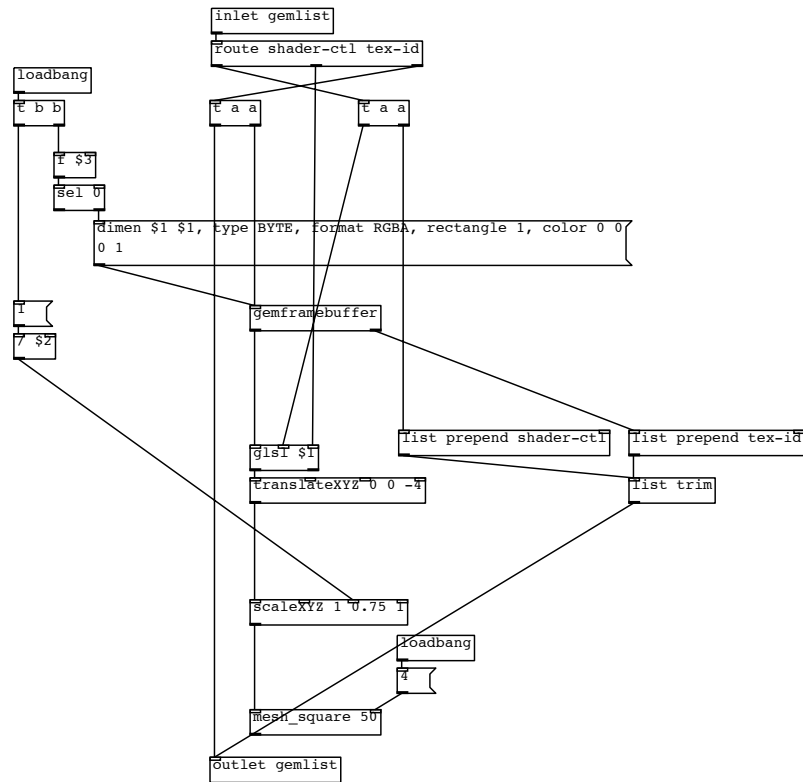


Abbildung 3.21: glsl-buf Abstraktion zum schnellen seriellen patchen von Shadern

le zu erlangen, wird das Video-Ausgabefenster etwas größer eingestellt, um teilweise in den Kontroll-Monitor hineinzuragen. In diesem Teil des GEM Window können kleine Live-Vorschau-Bilder aus verschiedenen Stadien des Systems gerendert werden. Mehr dazu in Abschnitt 4.3.5.

3.5 Diskussion des entwickelten Systems

Ein Problem ergibt sich in der Latenz der Videoverarbeitung. Die *Kinect* hat durch die interne Berechnung eine relativ hohe spürbare Latenz. Hinzu kommt die Latenz durch die Berechnungen in *Pure Data*. Dies führt dazu, dass die Schablone bei schnellen Bewegungen immer ein wenig hinterher hinkt. Für schnelle Tanzbewegungen ist das System demnach nur in Ausnahmefällen geeignet.

Weiters ergibt sich durch den begrenzten Öffnungswinkel der *Kinect*, dass nur ein relativ schmaler Bereich abgedeckt werden kann. Für größere Bühnen wären also zusätzliche *Kinect* Sensoren nötig, deren Bilder zu einem

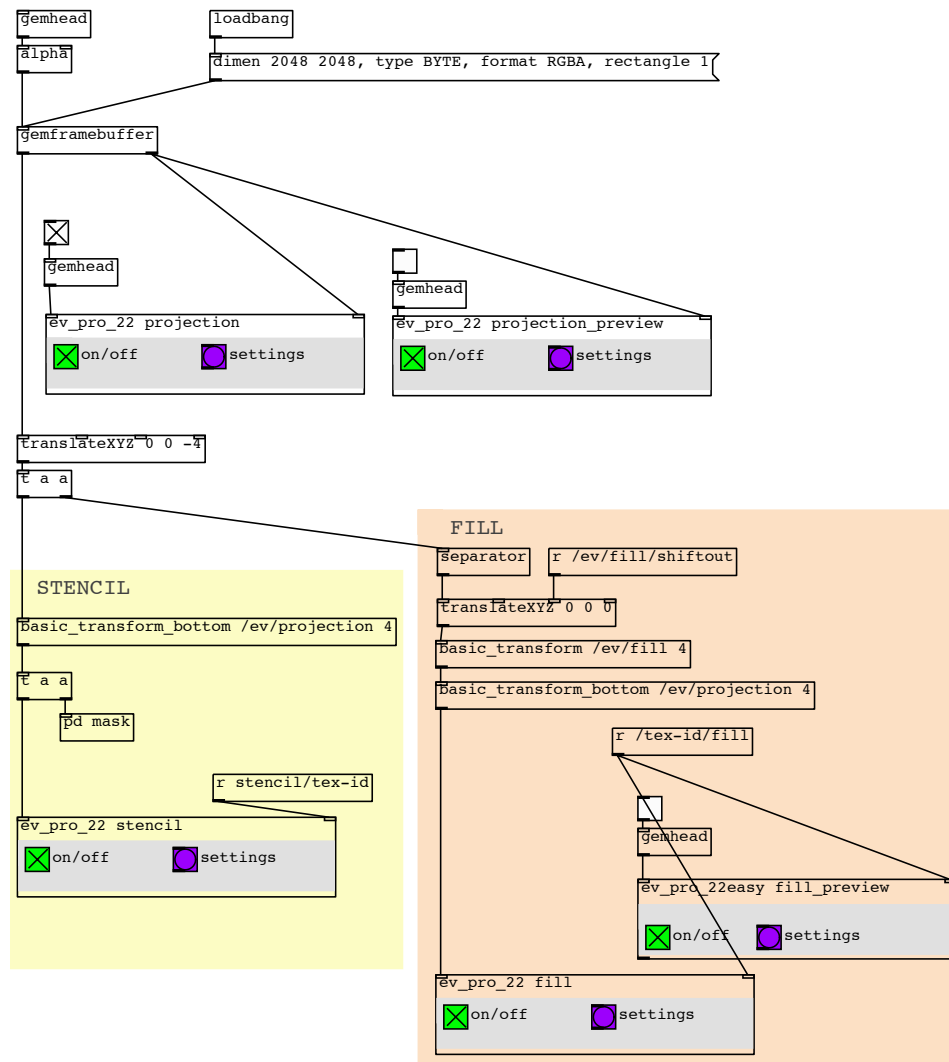


Abbildung 3.22: Pure Data Patch: Projektion

zusammenhängenden Panorama-Bild aneinandergesetzt werden müssten. Eine Abdeckung größerer Flächen durch mehrere Projektoren wäre durch die *Extended View Toolkit* mit *Pd* uneingeschränkt möglich. Dies würde allerdings, abhängig von der verwendeten Grafikkarte, den Einsatz von weiterer Hardware, wie z. B. *Matrox TripleHead2Go*⁹ erforderlich machen.

Die relativ große Ungenauigkeit der Projektion bei weitläufigen Bewegungen ist ebenfalls problematisch. Eine Lösung wäre die zusätzliche Messung der mittleren Distanz des Tänzers zur *Kinect*, um die Projektion je

⁹TripleHead2Go ist ein „Graphics eXpansion Module“ der Firma Matrox, das es erlaubt, bis zu 3 Anzeigegeräte an einem Video-Ausgang zu verwenden: <http://www.matrox.com>

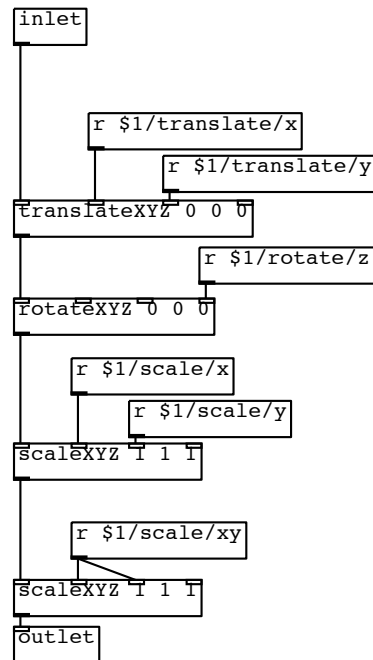


Abbildung 3.23: Einfache geometrische Verzerrungen zur Korrektur

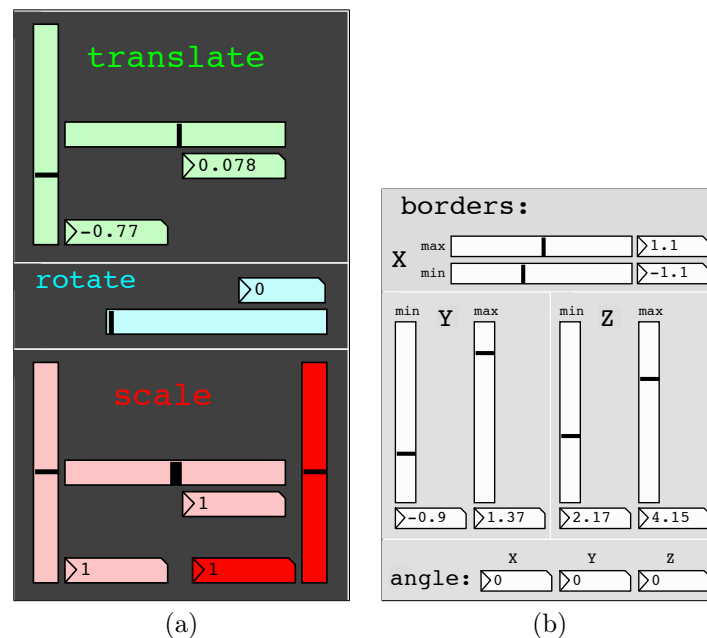


Abbildung 3.24: Grafische Benutzeroberfläche für (a) einfache geometrische Verzerrungen und (b) Erstellung der Schablone.

nach Abstand automatisch in der Größe zu ändern. Dieser Teil wurde zwar implementiert, hat sich jedoch als äußerst fehleranfällig erwiesen, weshalb trotzdem auf die manuelle Korrektur zurückgegriffen wurde.

Problematisch ist auch der relativ große Rechenleistungs-Bedarf des Systems. Es setzt eine teure Grafikkarte aus dem Gaming-Bereich voraus, und lastet einen Rechenkern meines *Intel i5-2500k* Prozessors fast vollständig aus.

Von diesen Probleme abgesehen funktioniert das System jedoch tadellos. Ohne eine spezifische Anwendung lässt sich das System aber nicht ausreichend testen. Es ist eine weitere Evaluierung anhand einer öffentlichen Aufführung erforderlich.

Kapitel 4

Monster

4.1 Ausgangspunkt & Aufgabenstellung

Ausgangspunkt ist eine Kooperation mit der Komponistin Wen Liu. Auf Basis eines Tonbandstückes soll eine audio-visuelle Tanzperformance realisiert werden.

Das Tonbandstück basiert auf auskomponierten Fragmenten für Flöte, die von der Wiener Flötistin Sylvie Lacroix eingespielt wurden. Die Aufnahmen wurden anschließend von Wen Liu in der Sequenzer-Software Apple Logic stark nachbearbeitet und arrangiert. Der Flötenklang wurde an vielen Stellen stark gefiltert und in der Tonhöhe verschoben. Laut dem Proposal (siehe [13]) sollte die Musik die immersive Erfahrung für das Publikum unterstützen. Die Musik sollte subtil direkten Einfluss auf die Gefühle der Zuhörer nehmen und dabei selbst eher im Hintergrund bleiben. Um Raum für Leben in der Visuellen Erscheinung zu schaffen, wurde die Flötenaufnahme in der Nachbearbeitung sämtlicher Lebendigkeit beraubt. Das Stück arbeitet mit der Spannung an der Grenze zwischen natürlichem und synthetischem Klang. Durch die Öffnung von musikalischen Räumen werden neue Klangdimensionen hörbar gemacht (siehe [13]).

Auch eine dazu passende Choreographie sollte in enger Zusammenarbeit entwickelt werden.

Da beim Tonbandstück der zeitliche Ablauf vorgegeben ist, sind hier bei einer Aufführung keine zeitlichen Änderungen oder Interaktionen mit anderen Domänen möglich. Der Tänzer muss sich also nach der Musik richten, und wird somit von dieser synchronisiert. Um trotzdem eine interessante Interaktion zu ermöglichen, soll die Video-Projektion vom Tänzer synchronisiert werden. Dadurch soll eine hierarchische Synchronisierung entstehen: Das Tonband steuert den Tänzer, der Tänzer steuert die Video-Projektion. Oder allgemein:

$$\textit{Musik} \rightarrow \textit{Tanz} \rightarrow \textit{Videoprojektion}$$

Da das Tonbandstück live während der Aufführung abgemischt werden soll, ergibt sich zusätzlich ein Rückkopplungs-Zweig:

$$\textit{Tanz, Musik, Video} \rightarrow \textit{Tonmischung}$$

4.2 Idee

Ein in Echtzeit aufgenommenes Video von der Rückseite des Tänzers soll spiegelverkehrt auf die Vorderseite seines Körpers projiziert werden. Dem Tänzer soll somit die Möglichkeit gegeben werden, mit seinem eigenen „Spiegelbild“ zu interagieren. Wie bei *D.a.v.e.* (siehe 2.3.1) soll der Tänzer mit seinem eigenen Körper interagieren können. Hierzu sollen weder vorgefertigte Videos, noch 3D-Renderings verwendet werden, sondern nur das pure Spiegelbild seines eigenen Körpers.

4.3 Implementierung

Auf Basis der gesammelten Ideen und des zuvor entwickelten Systems kann dieses nun für das Stück *Monster* angepasst und entsprechend erweitert werden.

4.3.1 Kamera

Um ein Spiegelbild des Tänzers zu erstellen, muss dieser also mithilfe einer Kamera live gefilmt werden. Die *Kinect* besitzt zwar eine integrierte RGB-Kamera, diese hat jedoch eine relativ schlechte Auflösung und niedrige Framerate, weshalb eine separate USB-Webcam gewählt wurde.

Die Entscheidung fiel für die *Logitech Webcam Pro 9000*¹, da dies eine der wenigen hochauflösenden USB-2.0 Webcams ist, die über einen 4:3 Sensor verfügen. Eine 16:9 Kamera wäre in diesem Fall relativ unpraktisch, da nur ein schmaler Streifen in der Mitte benutzt würde, und die vertikale Auflösung gering ist. Eine Drehung der Kamera um 90 Grad würde wiederum ein zu schmales Bild erzeugen. Die Kamera wird, wie die *Kinect* auch, über zwei aktive 10m USB Verlängerungen am Computer angesteckt. Längere Kabelverbindungen als 20m wurden nicht getestet.

4.3.2 Beleuchtung für das Kamerabild

Da das Stück für vollkommene Dunkelheit konzipiert ist, ergibt sich ein großes Problem für die Kamera: Um ein schönes Bild zu erhalten, muss die Szenerie gut ausgeleuchtet werden.

¹Logitech Webcam Pro 9000: www.logitech.com

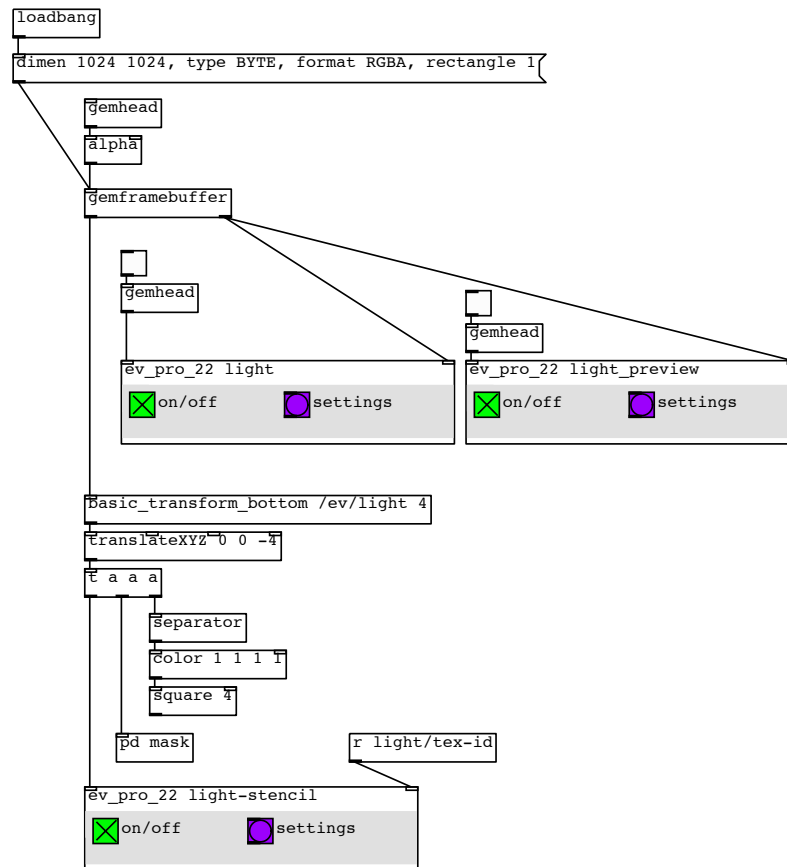


Abbildung 4.1: Pd Patch zur Beleuchtung des Tänzers

Erste Tests mit einem Scheinwerfer, der den Tänzer von hinten beleuchtet, ergaben, dass das Publikum stark geblendet werden würde. Außerdem würde die Bühne wieder zu stark erhellt werden, was die Projektion wiederum abschwächen würde. Es musste also eine Lösung gefunden werden, den Körper des Tänzers von hinten komplett zu beleuchten, ohne daran vorbei zu strahlen.

Es schien also praktisch, für die Beleuchtung das gleiche System, wie für die Projektion zu verwenden (s.Abb. 4.1). Durch Hinterlegung der für die Projektion gedachten Schablone mit einer weißen Fläche ergibt sich eine weiße Silhouette des Tänzers. Wenn diese von einem zweiten Projektor spiegelverkehrt von der Rückseite auf den Tänzer projiziert wird, so wird der Körper exakt ausgeleuchtet, ohne dass das Publikum etwas merkt. Diese „Lichtprojektion“ kann auf die gleiche Weise justiert werden, wie die Projektion für die Vorderseite.

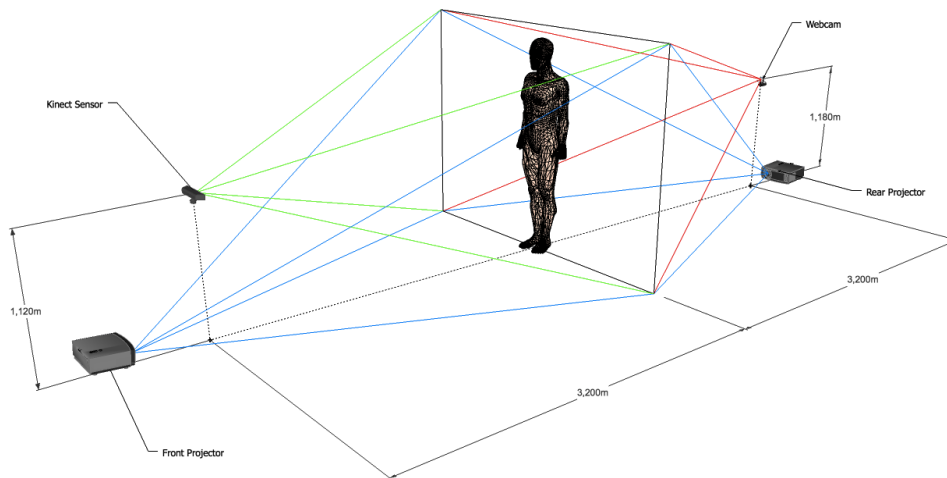


Abbildung 4.2: Technischer Aufbau für *Monster*

4.3.3 Technischer Aufbau

Der Technische Aufbau von Abb. 3.15 musste nun erweitert werden um die zusätzliche Kamera und den zweiten Projektor. Das resultierende System ist in Abb. 4.2 zu sehen.

4.3.4 Optische Rückkopplung

Da die Kamera direkt in den Beamer für die Vorderseite gerichtet ist, kann diese geblendet werden, falls die Projektion ungenau ist und über den Rand des Tänzers hinweg scheint. Leuchtet der Beamer nun also in die Kamera, so erhält diese ein weißes Bild. Da dieses Bild aber von eben jenem Beamer wieder projiziert wird, etnsteht eine optische Rückkopplung. Die Schablone wird dadurch zwar logischerweise nicht gestört, die Projektion wandelt sich jedoch zu einer einfachen Beleuchtung des Tänzers.

Die Einfachste Lösung zur Unterbindung dieser Rückkopplung ist es, das Bild des Projektors an der Stelle, die zur Kamera zeigt, durch einen verschiebbaren, gerenderten Punkt schwärzen. Da die Kamera- und Beamer-Positionen fix sind, braucht der Punkt nur einmal vor der Aufführung justiert werden. Hierzu hat es sich als praktisch erwiesen, den Punkt farbig einzufärben und bei laufendem Projektor so zu verschieben, dass die Kamera beleuchtet wird. Anschließend kann die Farbe wieder zurückgesetzt und die Position gespeichert werden.

4.3.5 Bildschirmordnung

In Abb. 4.3 sieht man einen Screenshot des Kontrollmonitors. Die linke Hälfte beinhaltet die grafische Benutzeroberfläche für den *PD*-Patch. Auf der rech-

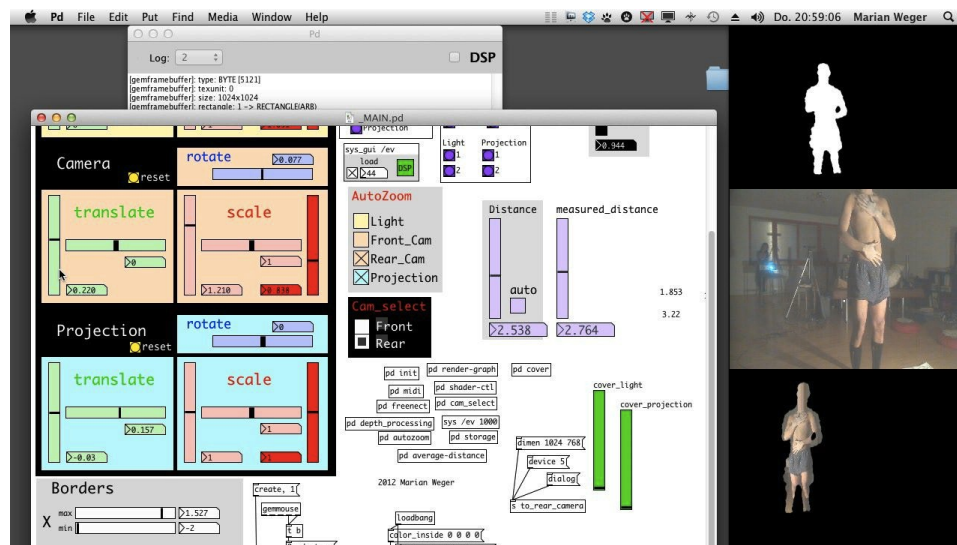


Abbildung 4.3: Monitor Screenshot

ten Seite sind Kontroll-Bilder für die Korrektur während der Performance eingeblendet: Oben sieht man den Video-Ausgang für den hinteren Beamer, in der Mitte, das Kamerabild und unten den Ausgang für den vorderen Beamer. Da der Desktop von der Grafik Hardware automatisch in die einzelnen Video-Ausgänge unterteilt wird, muss in *GEM* ein Ausgabe-Fenster mit der Breite der beiden Projektoren zusammen, plus dem Überhang in den Kontrollmonitor, erstellt werden. Die Berechnungsgrundlage hierfür findet sich in Abb. 4.4.

4.3.6 Midi-Steuerung

Um die wichtigsten Parameter während der Performance live zu steuern, wurden die MIDI-Controller *Korg nanoKONTROL* (s.Abb. 4.5) und *Akai LPD8* (s.Abb. 4.6) genutzt.

4.4 Choreografie

4.4.1 Die ersten Tests

Nachdem das System technisch fertig war, konnten erste Tests mit dem Tänzer (Rino Indrawan Indiono), durchgeführt werden. Zuerst ging es darum, durch Ausprobieren die neu entstandenen Möglichkeiten für den Tänzer kennenzulernen. Da der Tänzer die Projektion auf seinem Körper nur unzureichend sehen kann, erwiesen sich Kamera-Aufnahmen, bzw. ein Spiegel als wertvolle Hilfsmittel. Zusammen mit der Komponistin, Wen Liu, wurde an-

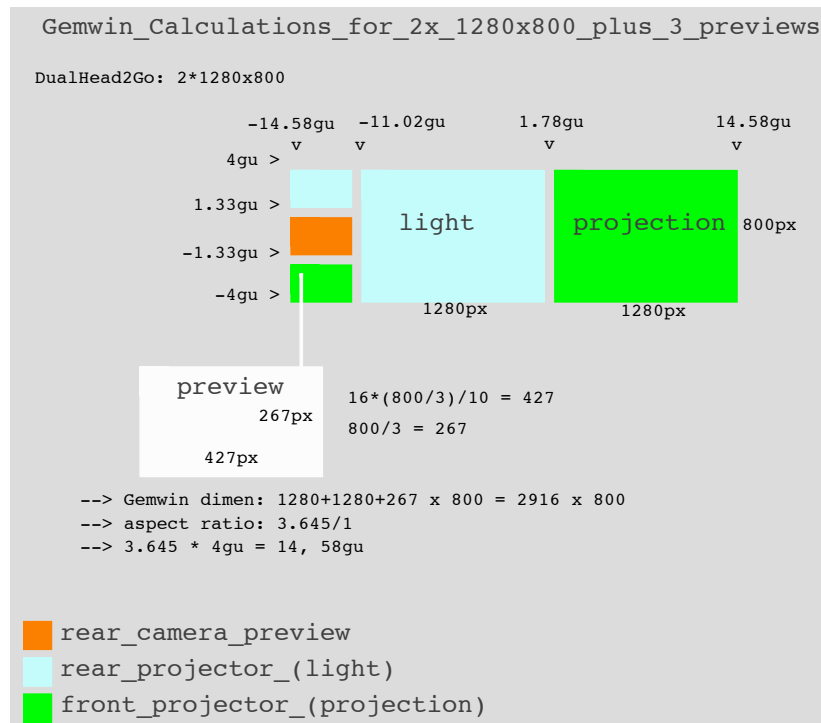


Abbildung 4.4: GEM Ausgabefenster Einstellungen für Preview

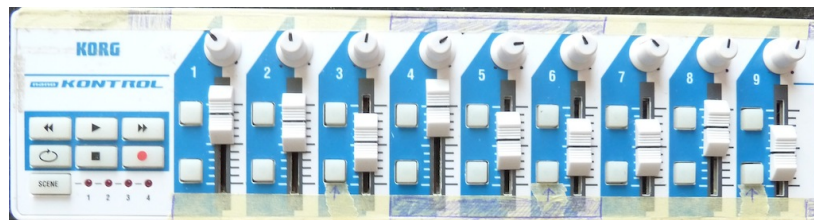
Abbildung 4.5: Der Korg nanoKONTROL wurde für *Monster* in drei Sektoren unterteilt: Die Anpassung für die Beleuchtung, das Kamerabild und die Projektion.Abbildung 4.6: Akai LPD8 mit Beschriftungen für *Monster*



Abbildung 4.7: Choreographie für *Monster*

hand der Musik eine passende Choreographie entwickelt. Es stellte sich heraus, dass langsame, kleine Bewegungen hierbei eine größere Wirkung erzielen, als ausladend große. Auch durch den eher engen, beklemmenden Charakter der Musik, wirkten weitläufige Bewegungen auf der Bühne eher als störend. Die Einschränkung des Systems auf einen relativ kleinen Bühnenbereich ist demnach für dieses Stück nicht von Nachteil. Die größte Wirkung erlangte der Tänzer auf einer festen Position stehend mit kleinen Bewegungen.

4.4.2 Kostüm

Da die Musik eher unauffällig, schlicht und beklemmend wirkt, wurde auch für das Kostüm des Tänzers ein schlichter Ansatz gewählt. Gemeinsam kamen wir zu dem Schluss, dass jegliche Art von Kostüm hier sogar nur störend wirkt und deshalb folgerichtig komplett darauf verzichtet werden sollte. Ein komplett unbedeckter Tänzer wäre zwar vielleicht das ästhetische Ideal, würde aber die Deutung des Stücks eher in Richtung Provokation verdrehen. Also entschieden wir uns für eine schlichte, weiße Unterhose, um den Ausgangspunkt für Interpretationen des Publikums neutral und unbefangen zu halten.

Um die Projektion auf dem Körper besser zum Vorschein zu bringen wurde die gesamte Rückseite des Tänzers mit weißer Körperfarbe bemalt. Wenn der Tänzer mit der Vorderseite zum Publikum steht, ergibt sich so für die Kamera ein besserer Kontrast. Im Falle der Rückwärts zum Publikum gewandten Position des Tänzers entsteht eine bessere Projektionsfläche für den Beamer.

4.4.3 Ablauf des Stücks

Die erarbeitete Choreographie besteht aus mehreren Teilabschnitten, innerhalb dieser der Tänzer relativ frei improvisieren kann. In Einklang mit der Musik entsteht eine Art Zyklus (s.Abb. 4.7).

Die sonderbare Hand

Zu Beginn steht der Tänzer mit dem Gesicht zum Publikum und bewegt langsam eine Hand hinter den Rücken. Die Hand erscheint somit als Projektion auf seinem Bauch. Mit der anderen Hand versucht er sich zaghaft, dieser sonderbaren Hand zu nähern, es gelingt ihm jedoch nicht, diese zu greifen.

Der Zusammenbruch

Er drückt immer stärker vergeblich gegen den Bauch, und geht langsam zu Boden, bis er so klein wie möglich am Boden kauert. Die Wirkung der Projektion fällt hierbei langsam in sich zusammen, da kaum mehr eine sinnvolle Projektionsfläche zur Verfügung steht.

Der Wendepunkt

Immer noch in sich zusammengezogen dreht er sich nun langsam um, bis er die gleiche Haltung, jedoch mit dem Rücken zum Publikum gerichtet, eingenommen hat.

Erwachen

Dann beginnt der Tänzer, sich langsam aufzurichten. Die Projektion entwickelt hierbei ihre volle Wirkung. Langsam wird die Vorderseite des Tänzers in der Projektion erkennbar. Durch den beim Aufrichten noch gekrümmten Rücken wirkt der projizierte Körper ungewohnt deformiert, bis er dann schließlich die volle Größe erreicht. Für den Betrachter ist nun optisch nicht mehr leicht erkennbar, was Projektion ist, und was der echte Körper.

Das Monster

Der Tänzer sinkt noch ein paar Mal in sich zusammen, um sich anschließend wieder aufzurichten, und spielt so damit, seinen Körper visuell zu deformieren.

Die Verschmelzung

Nun versucht der Protagonist abermals, dieses zweite Ich auf seinem Körper zu fassen. Diesmal jedoch mit vollem Körpereinsatz. Sein Körper beginnt, mit der Projektion zu verschmelzen, und es entsteht weniger ein deformierter

Körper, sondern vielmehr ein monströses Gebilde aus Körperteilen. Schließlich hält der Tänzer für wenige Sekunden inne.

Der Ausbruch

Um die choreographische Entwicklung auf eine neue Stufe zu heben, wurden nun einige Erweiterungen am System vorgenommen. Ab diesem Stopp wird das Kamerabild der Webcam um drei Sekunden verzögert. Der Tänzer kann sich nun also von seinem alter Ego losreißen, und ist diesem fortan immer drei Sekunden voraus. Er wartet ein zweites mal, bis die Projektion die gleiche Position inne hat wie er. Nun wird das Kamerabild zu einem Standbild angehalten. Der Tänzer bricht aus, und kehrt mit einem dritten Stop schließlich wieder in dieselbe Position zurück, um die Projektion dort wieder „abzuholen“ und erneut zum Leben zu erwecken.

Die Versöhnung

Das Delay ist immer noch aktiv. Der Tänzer hält ein viertes mal inne, bis die Projektion ihn eingeholt hat. An dieser Stelle wird die Verzögerung wieder abgeschaltet. Der Tänzer vereint sich wieder mit der Projektion, bis er schließlich ein letztes mal, mit der Hand auf dem Bauch, wie in der ersten Szene stehen bleibt.

Zwei Varianten für ein Ende

Von Angesicht zu Angesicht Für die erste Aufführung wurde ein Ende gewählt, in dem sich der Protagonist seinem zweiten Ich stellt. Um dies zu bewerkstelligen, wird die gesamte Bildausgabe zu einem Standbild angehalten. Der Tänzer dreht sich nun langsam um, bis schließlich seine Vorderseite (spiegelverkehrt!) auf seiner Vorderseite zu sehen ist. Nach einer kurzen Pause wird abgeblendet.

Die Flucht Da die erste Version vom Ende nicht schlüssig erschien, wurde ab der zweiten Aufführung ein alternatives Ende gewählt. Es schien schlüssiger, wenn sich die projizierte Gestalt langsam wieder aus dem Körper zurückzieht und schließlich verschwindet. Hierzu wurde das gesamte projizierte Bild langsam nach unten verschoben. Die Projektion wandert langsam über den Rücken hinab, bis der Kopf schließlich über die Beine im Boden verschwindet. Auch die Musik wird langsam leiser, bis komplette Stille und Dunkelheit herrscht.

4.5 Diskussion der Aufführungen

Das Stück wurde mittlerweile mehrere Male erfolgreich aufgeführt und musste dabei an verschiedenste Bühnensituationen angepasst werden. Hierbei kamen immer wieder Schwachstellen des Systems zum Vorschein, die einer Lösung bedurften. Videos und Bildmaterial der Aufführungen können unter [28] abgerufen werden.

4.5.1 21. Jänner 2012: Semperdepot, Wien

Die Uraufführung des Stücks fand am 21. Januar 2012 im Semperdepot in Wien statt. Dies war die Abschlussveranstaltung des Projekts *New_Air*², von welchem das Stück professionell unterstützt wurde.

Bei der Aufführung stellte sich heraus, dass das Einmessen des Systems extrem viel Zeit beansprucht, und auch der Platzbedarf durch die dort verfügbaren schmalwinkligen Projektoren relativ groß war. Die Projektoren mussten viele Meter vom Tänzer entfernt stehen, um ihn optimal ausleuchten zu können. Ein weiteres Problem waren die Kabellängen, die nicht zum Regieplatz reichten. Deshalb musste ein separater Tisch am Bühnenrand eingerichtet werden. Durch den schiefen Blick auf die Bühne war das Einmessen und auch korrigieren während der Aufführung relativ schwierig.

4.5.2 März 2012: Fidelio Wettbewerb, Wien

Nach dem Erfolg im Semperdepot folgten eine Reihe von Auftritten beim *Fidelio-Wettbewerb 2012*³ der *Konservatorium Wien Privatuniversität* in der Sparte „Fidelio.Kreation/Laboratorium/Werkstatt“.

Um das Stück musikalisch auf eine neue Ebene zu heben, wurde ab der zweiten Aufführung eine Live Flöte hinzugefügt. Die Flötistin Melanija Pinter improvisiert nach relativ grob gesetzten Richtlinien mit eher Geräuschhaften Einwürfen zur Tonbandeinspielung dazu. Somit kann sich auch die Musik etwas mehr nach dem Tanz richten und es findet eine umfassende Interaktion zwischen Tanz und Musik statt.

Eine neue Aufstellung für den Kinect Sensor

Die Vor- und Hauptrunde in den Sälen des „Kons“ waren eine sehr gute Möglichkeit, das System im harten Bühnenalltag zu testen.

Es stellte sich heraus, dass die Kinect vor dem Tänzer dem Publikum oftmals die Sicht versperrt. Da dieser in der Berechnung sowieso vereinfacht

²NeW_Air (New experimental Ways_Artists in residence) war ein Projekt im Rahmen von uni:vision an der Universität für Musik & darst. Kunst Wien: <http://newair.at>

³Der Fidelio-Wettbewerb wird seit 2002 von der Hugo-Breitner-Gesellschaft zur Förderung junger KünstlerInnen an der Konservatorium Wien Privatuniversität ausgetragen: <http://www.konservatorium-wien.ac.at>

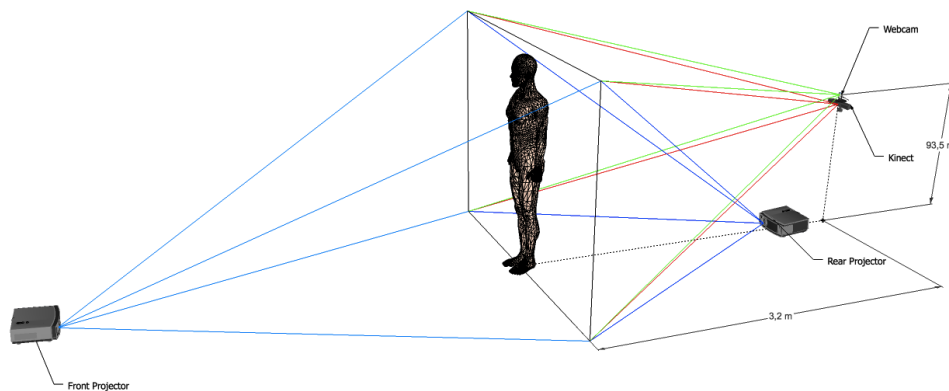


Abbildung 4.8: Überarbeiteter Technischer Aufbau für *Monster*

als 2-dimensional angenommen wurde, lag die Überlegung nahe, die Kinect zusammen mit der Kamera auf dem gleichen Stativ auf Rückseite zu positionieren (s.Abb. 4.8). Dies bewirkte keinerlei Nachteile im Tracking, jedoch erhebliche Vorteile beim Einmessen, da Webcam und Kinect fortan den gleichen Blickwinkel besaßen.

Ergebnis

Der erste Einsatz der neuen Kamera-Aufstellung erfolgte beim Finale im Wiener *Porgy&Bess*. Das System funktionierte Tadellos, und das Stück gewann schließlich ein Stipendium der *Tsuzuki-Stiftung* im Wert von 1500 Euro.

4.5.3 3. Mai 2012: Odeon, Wien

Für die Jubiläumsfeier *20 Jahre Erasmus in Österreich - 25 Jahre Erasmus in Europa* im Wiener *Odeon Theater* sollte eine auf 5 Minuten verkürzte Version von *Monster* gezeigt werden.

Ein effizienteres System für den Aufbau

Da das Stück in ein umfangreiches und aufwändiges Programm eingebunden war, musste ein schnelleres System für den Auf- und Abbau entwickelt werden.

Die Schwachstelle lag mittlerweile hauptsächlich noch bei den verwendeten Projektoren. Es wurden bisher immer die am Aufführungsort verfügbaren Geräte genutzt. Da diese vorher nicht ausprobiert werden konnten, bzw. deren Spezifikationen im voraus nicht bekannt waren, musste bei der Aufstellung zum Teil stark improvisiert werden.

Die Lösung war, fortan nur noch eigene, identische Ultra-Kurzstanz-Projektoren vom selben Typ zu verwenden. Die Ultra-Kurzstanz Optik

führt zu einem großen Öffnungswinkel, der eine Aufstellung sehr nahe am Tänzer erlaubt. Diese Projektoren benötigen nur einen Abstand von ca. 2 Meter, um den Tänzer komplett ausleuchten zu können.

Da nun die gesamte benötigte Technik selbst zur Verfügung gestellt wurde, konnte das System in Ruhe zuhause exakt eingemessen werden.

Ergebnis

Im Odeon wurde das System dann nach den genauen Maßen wie zuhause auf der Bühne aufgebaut werden, und benötigte nur einen kurzen Test von ca. 5 Minuten während des Soundcheck.

Um die Verkürzung des Stücks zu erreichen, musste lediglich die Tonband-Zuspielung von Seiten der Komponistin editiert werden, und mit Tänzer und Flötistin eine neue Struktur der Choreographie festgelegt werden. Das Stück entfaltete in dem großen Saal des *Odeon* eine starke Wirkung.

Einziges Problem herbei war die stärkere Lichtleistung der neuen Projektoren. So wurde der Tänzer von der Rückseite so stark beleuchtet, dass die Kamera übersteuerte und nur mehr ein schwarzes Bild anzeigte. Zufällig bestand aber hierfür schon eine Lösung: Ein Drehregler auf dem MIDI-Controller war schon mit einer Abblend-Funktion belegt, die ursprünglich zum sanften abblenden am Ende des Stücks konzipiert war. Es stellte sich heraus, dass der Projektor hiermit perfekt in der Helligkeit regelbar ist. Durch die schwächere Beleuchtung wurde das Kamerabild wieder kontrastreicher, und ergab natürliche Farben.

4.5.4 9. September 2012: Kino Šiška, Ljubljana (ICMC2012)

Eine große Ehre war es, beim Eröffnungskonzert der *International Computer Music Conference 2012 (ICMC)*⁴ in Ljubljana aufzutreten.

Im Vorfeld durchlief das Stück einen mehrstufigen Review-Prozess. Die Kritiken der Reviewer sind im Anhang A.2 angeführt. Die Kritiker lobten die visuelle Erscheinung des Stücks, kritisierten jedoch teilweise die fehlende Interaktion des Tänzers mit der Musik. Sie einigten sich jedoch darauf, dass die Musik trotzdem stimmig ist, und zu dem Werk passt (vgl. Anhang A.2).

Eine neue Kamera und der Wechsel auf Linux

Auch für diese Aufführung wurden wieder einige Änderungen am System vorgenommen. Prinzipiell wurde der selbe Aufbau gewählt, der schon im *Odeon* so gut funktioniert hatte. Nun war es an der Zeit, das Problem mit der niedrigen Kamera-Framerate anzugehen. Nach Tests mehrerer Kameras verschiedener Hersteller lieferte die *Logitech HD Pro Webcam C920* die

⁴International Computer Music Conference 2012: <http://www.icmc2012.si/>

besten Bilder. Sie liefert dank Hardware-Kompression eine Auflösung bis zu 1920x1080 Pixel bei einer Framerate von 30 Hz über USB-2.0.

Leider stehen für *Mac OS X* keine geeigneten Treiber zur Verfügung. Deshalb wurde ein Wechsel auf *Ubuntu* Linux vorgenommen.

Ergebnis

Leider stellte sich heraus, dass die neue Kamera noch leichter übersteuert als die ursprünglich verwendete, was in einem relativ kontrastarmen Bild bei der Aufführung resultierte. Ansonsten gab es bei diesem Konzert keine weiteren Probleme. Von Veranstalter und Publikum gab es viel Lob und keinerlei Kritik, weshalb auch dieser Auftritt als voller Erfolg gewertet werden kann.

Kapitel 5

Zusammenfassung

Es wurde erfolgreich ein System implementiert, das es einem Tänzer durch weitreichende Interaktion mit dem Computersystem neue Ausdrucksmöglichkeiten verleiht. Die hierarchische Synchronisierung zum Tonbandstück hat gut funktioniert. Das System konnte sich mehrmals auch in schwierigen Aufführungssituationen auf verschiedenen Bühnen beweisen, und hat stets einen reibungslosen Ablauf garantiert.

Erfolgreich wurde es auch bei dem Stück „Vertimas“ von Matthias Kronlachner in Vilnius eingesetzt (siehe [12]).

Monster kann einen Preis beim *Fidelio-Wettbewerb 2012* vorweisen und schaffte es ins Eröffnungskonzert der *International Computer Music Conference 2012* in Ljubljana. Bei allen Aufführungen war eine ausnahmslos positive Resonanz zu vernehmen.

Die Vereinfachung zu einer zweidimensionalen Berechnung der Projektion hat zwar gut funktioniert, bringt jedoch für weitläufigere Bewegungen auf der Bühne zu viele Ungenauigkeiten. Für eine exakte Berechnung sollte lieber komplett im 3D-Raum gearbeitet werden. Somit würde sich die Projektion vollautomatisch regeln, und bedürfte keiner manuellen Korrektur mehr während der Aufführung.

Eine Bereicherung wäre evtl. die zusätzliche Auswertung des Skelett-Modells durch das *pix_openni*-External (siehe [11]). Dem Tänzer könnte so die Möglichkeit gegeben werden, sowohl Video-Projektion als auch Musik selbst zu kontrollieren.

Um in Zukunft größere Bühnen abzudecken, ist es erforderlich, die Videobilder mehrerer *Kinect* Sensoren zu einem zusammengehörigen zu verbinden. Damit wäre es auch möglich, Stücke für mehrere Tänzer gleichzeitig zu realisieren. *pix_openni* bietet die Möglichkeit, verschiedene Personen voneinander zu trennen, um im Blau-Kanal des RGB-Streams zusätzlich eine Tänzer-ID zu übertragen. Somit könnten unterschiedliche Tänzer jeweils unabhängig mit anderem Bildmaterial beleuchtet werden.

Eine Veröffentlichung der *Pd*-Patches ist in naher Zukunft nicht geplant,

das erarbeitete Wissen wird aber nach und nach in *Extended View Toolkit* (siehe [24]) einfließen und in diesem Zusammenhang veröffentlicht werden.

Anhang A

Ergänzende Materialien

A.1 Auszüge aus dem Proposal für Monster [13]

A.1.1 Abstract

Monster is a mixed-media performance for flute, live-electronics, one dancer and realtime-generated video projection. The movements of the dancer are recorded by a multi-camera-system, and re-projected onto his body at the same time. The basic idea of the composition is to enable the dancer to interact directly with himself, and thus, create a strongly immersive visual performance. In an experimental stage setting, an alchemistic laboratory is created, where diverse ingredients like music, dance and video are melted together in order to create a new being, which transcends the original matters from which it is constructed.

A.1.2 Description

The entire piece is subdivided into two main parts. It starts with the dancer standing on stage, facing the audience. His left hand resides behind his back, recorded by a standard webcam, and gets re-projected onto his front body by a video projector in realtime. The central key scene is the anchor point of the performance, where the dancer starts to move downwards gradually, and, at the same time turns around. Whilst he is successively rising again, with his back towards the audience, a new dimension is opened. A new visual phenomena is demonstrated: An optical illusion that tricks and entertains our eyes, as the dancer seems to be mixed up with his own reflection.

During further progress, a delay between real movement and projection is established. The time dimension is split, resulting in two different time bases. Thus, the projection no longer represents the present.

A.1.3 Musical Idea

The music is intended to sublimely support the immersive experience. Standing more in the back, it tries to exert direct influence on the feelings of the audience. To make room for life in the visual component, all liveliness in the music is eliminated as much as possible. Nevertheless, a real flute is chosen, to keep the sound natural and familiar. Analogue to the choreography, a blurring border between the natural and synthesized sound is established, and new dimensions are unveiled by opening up additional musical spaces.

A.2 Kritiken im Auswahlprozess der ICMC2012

Dies ist ein Auszug aus der Annahmeerklärung des *ICMC 2012 Music Program Committee*:

----- REVIEW 1 -----

PAPER: 276

TITLE: Monster

AUTHORS: Wen Liu and Marian Weger

OVERALL RATING: -1 (weak reject)

REVIEWER'S CONFIDENCE: 3 (high)

Artistic Merit: 4 (medium high)

Technical Quality: 5 (high)

Feasibility of Performance: 3 (medium)

The work is very clearly designed, seems (from the trailer) to be effective and also seems to do an excellent job of playing with the reality of the dancers body.

My only concern with the work in terms of its place in the ICMC is the relative unimportance of the musical element. The attention to the musical in the proposal amounts to just a few sentences, compared with that which concerns the dancer - while this makes perfect sense after seeing some of the work, my concern is that the piece is not the ideal fit for a 'music' conference. I would also suggest that the artist alter the line, "Nevertheless, a real flute is chosen, to keep the sound natural and familiar." - the use of the word 'nevertheless' contributed greatly to my feeling that the musical accompaniment to the dancing didn't really matter.

----- REVIEW 2 -----

PAPER: 276

TITLE: Monster

AUTHORS: Wen Liu and Marian Weger

OVERALL RATING: 1 (weak accept)

REVIEWER'S CONFIDENCE: 2 (medium)

Artistic Merit: 4 (medium high)

Technical Quality: 5 (high)

Feasibility of Performance: 4 (medium high)

The video excerpt clearly displays the artistic merit of the performance, a subtle yet mesmerising optical illusion, an innovative and intimate integration of dance and live visuals. Using the body itself as the visual display is a novel and poetic idea, one which is not without its technical difficulties.

The proposal clearly states both the technological and aesthetic preoccupations of the performance in question, with very clear and detailed documentation of the technical aspects of the performance. The supporting video materials demonstrate the feasibility of the performance, however it is assumed that ample set-up time will need to be allocated for calibration of the various visual tracking components, especially considering the exacting nature of the live moment tracking.

Sonically the musical materials are fairly static, however the aesthetic decisions behind the link between both aural and visual components are clear from the documentation provided. Considering the focus of this conference, it is however a little disappointing that the complex technologies involved might not also have been used interactively with respect to the generation of the live audio material. This criticism is also inclusive of the hidden/non-integrated 'presence' of a live instrumental performer as 'support'.

----- REVIEW 3 -----

PAPER: 276

TITLE: Monster

AUTHORS: Wen Liu and Marian Weger

OVERALL RATING: 3 (strong accept)

REVIEWER'S CONFIDENCE: 3 (high)

Artistic Merit: 5 (high)

Technical Quality: 5 (high)

Feasibility of Performance: 4 (medium high)

This is a highly creative and original piece of work which has been executed with a high degree of technical competence. Excellent use of a range of media to produce a compelling performance piece. I strongly recommend this piece for presentation. The presenters have provided a comprehensive list of technical requirements and are able to bring most of the necessary equipment with them needed for presentation.

----- METAREVIEW -----

PAPER: 276

TITLE: Monster

Reviewers split over the role/importance of the music (in the context of a music conference), but the majority agree that the music is apt for the piece and works well: though one reviewer notes the missed

opportunity to integrate the video-tracking data into the music in some way, to make it more generative and involved.

A practical concern was raised that extra rehearsal may be needed for video/tracking calibration (presumably this would have to be done in the performance space).

Literaturverzeichnis

- [1] Arkaos - interactive visual technologies: *MediaMaster Pro*. <http://www.arkaos.net/show-lighting/mediamaster/>.
- [2] Chunky Move. <http://chunkymove.com.au>.
- [3] Chunky Move und F. Weiß: *Glow*, 2008. <http://www.frieder-weiss.de/glow.htm>.
- [4] Coolux Media Systems: *Pandoras Box Server*. <http://www.coolux.de/products/pandoras-box-server/>.
- [5] Cycling '74: *Max/MSP*. <http://cycling74.com/>.
- [6] Duden online: *Interaktion, die*. <http://www.duden.de/rechtschreibung/Interaktion>, besucht: 26.09.2012.
- [7] gameRENDERING: *Gaussian Blur Filter Shader*. <http://www.gamerendering.com/2008/10/11/gaussian-blur-filter-shader/>, besucht: 26.09.2012.
- [8] GEM - Graphics Environment for Multimedia. <http://gem.iem.at/>.
- [9] Keller, P.E.: *Musical ensemble synchronisation*. In: *The inaugural International Conference on Music Communication Science*, 2007.
- [10] Kronlachner, M.: *pix_freenect - PureData/GEM External*. https://github.com/kronihias/pix_freenect.
- [11] Kronlachner, M.: *pix_openni - PureData/GEM External*. https://github.com/kronihias/pix_openni.
- [12] Kronlachner, M. und I. m Zmoelnig: *The Kinect sensor as human-machine-interface in audio-visual art projects*. In: *Live Interfaces: Performance, Art, Music*, 2012. <http://lipam.lurk.org/index.php?conference=LIPAM&schedConf=lipam2012&page=paper&op=view&path%5B%5D=42&path%5B%5D=15>.
- [13] Liu, W. und M. Weger: *Proposal for the piece "Monster"*, 2012.

- [14] MadMapper - The Video Mapping Software. <http://www.madmapper.com/>.
- [15] Microsoft: *DirectX Graphics*. <http://msdn.microsoft.com/directx>.
- [16] Microsoft: *Kinect for Windows*. <http://www.microsoft.com/en-us/kinectforwindows/>.
- [17] Microsoft: *Xbox Kinect*. www.xbox.com/kinect.
- [18] Moar, V., G. Eckel und D. Pirrò: *Bodyscapes*. <http://iaem.at/projekte/egm/bodyscapes>.
- [19] Obarzanek, G.: *Digital Moves*. PopTech Conference, 2009. http://poptech.org/popcasts/gideon_obarzanek%E2%80%99s_digital_moves.
- [20] Obermaier, K. und C. Haring: *D.a.v.e - digital amplified video engine*, 1999. <http://www.exile.at/dave/>.
- [21] Puckette, M.: *Pure Data*. <http://puredata.info/>.
- [22] Story, J., T. Harada und H. Grün: *DirectCompute Accelerated Separable Filters*. In: *Game Developers Conference*, 2011. http://developer.amd.com/gpu_assets/DirectCompute%20Accelerated%20Separable%20Filters.pps.
- [23] Syphon. <http://syphon.v002.info/>.
- [24] Venus, P. und M. Weger: *Extended View Toolkit*. <http://extendedview.mur.at/>.
- [25] Venus, P., M. Weger, C. Henry und W. Ritsch: *Extended View Toolkit*. In: *Pure Data Convention Weimar - Berlin*, 2011. http://www.uni-weimar.de/medien/wiki/PDCON:Conference/Extended_View_Toolkit.
- [26] Vicon Motion Capture Systems: *Tracker*. <http://www.vicon.com/products/vicontracker.html>.
- [27] vvvv - a multipurpose toolkit. <http://vvvv.org/>.
- [28] Weger, M.: *Monster*. <http://marianweger.com/projects/monster.shtml>.
- [29] Weiß, F.: *EyeCon - Video Motion Sensing*. <http://eyecon.palindrome.de/>.
- [30] Weiß, F.: *Solo 4 Three*, 2003. <http://www.frieder-weiss.de/video/projects.htm>.
- [31] Wikipedia: *Pixel*. <http://de.wikipedia.org/wiki/Pixel>, besucht: 26.09.2012.

- [32] Wikipedia: *Synchronisation*. <http://de.wikipedia.org/wiki/Synchronisation>, besucht: 26.09.2012.
- [33] Wikipedia - The Free Encyclopedia: *Drehmatrix*. <http://de.wikipedia.org/wiki/Drehmatrix>, besucht: 26.09.2012.
- [34] Wikipedia - The Free Encyclopedia: *GLSL*. <http://de.wikipedia.org/wiki/GLSL>, besucht: 26.09.2012.
- [35] Wikipedia - The Free Encyclopedia: *Master/Slave*. <http://de.wikipedia.org/wiki/Master/Slave>, besucht: 26.09.2012.
- [36] Wikipedia - The Free Encyclopedia: *Rastergrafik*. <http://de.wikipedia.org/wiki/Rastergrafik>, besucht: 26.09.2012.
- [37] Wikipedia - The Free Encyclopedia: *Structured-light 3D scanner*. http://en.wikipedia.org/wiki/Structured-light_3D_scanner, besucht: 26.09.2012.
- [38] Wikipedia - The Free Encyclopedia: *Thermographic camera*. http://en.wikipedia.org/wiki/Thermographic_camera, besucht: 26.09.2012.
- [39] Wikipedia - The Free Encyclopedia: *Tracking*. <http://de.wikipedia.org/wiki/Tracking>, besucht: 26.09.2012.
- [40] Wikipedia - The Free Encyclopedia: *Videoprojektor - Projektionsverhältnis*. <http://de.wikipedia.org/wiki/Videoprojektor>, besucht: 26.09.2012.
- [41] Zmoelnig, I. m: *GEM - Graphics Environment for Multimedia*, 2003. <http://gem.iem.at/documentation/tutorial/primer>.

Abbildungsverzeichnis

2.1	D.a.v.e. - digital amplified video engine (Klaus Obermaier und Chris Haring, 1999) [20]	6
2.2	Solo 4 > Three (Shadows) (Frieder Weiß, 2003) [30]	6
2.3	Glow (Chunky Move, 2008) [2]	7
3.1	perfekte Beamer-Aufstellung (a) und geometrische Verzerrung bei schiefer Aufstellung [25]	11
3.2	Entzerrte Projektion bei schiefer Beamer-Aufstellung (a) das vom Projektor ausgegebene, verzerrte Bild, um diese Projektion zu erreichen [25]	12
3.3	Coolux Pandoras Box Server (a) und zugehörige, grafische Benutzeroberfläche (b) [4]	13
3.4	Arkaos MediaMaster Pro [1]	14
3.5	MadMapper [14]	14
3.6	EyeCon [29]	15
3.7	Cycling '74 Max/MSP mit Jitter [5]	16
3.8	Pure Data mit dem GEM External [41]	16
3.9	vvvv - a multipurpose toolkit [27]	17
3.10	Vicon Kamera-System (a) und grafische Benutzeroberfläche der Software (b) [26]	18
3.11	Tänzerin mit Infrarot-Reflektoren für das Vicon Tracking System [18]	19
3.12	Thermografische Aufnahme der Körpertemperatur einer Frau [38]	19
3.13	Prinzip eines Structured Light 3D-Scanner [37]	20
3.14	Microsoft Xbox Kinect Sensor [17]	21
3.15	Technischer Aufbau des Systems	22
3.16	Tiefenbild aus pix_freenect im Raw-Modus	23
3.17	Einbinden von OpenGL Shadern in Pd	24
3.18	Rohe Schablone	26
3.19	Gefilterte Schablone	29
3.20	Die Implementierung der Tiefenbild-Verarbeitung in Pd	30

3.21	gls-buf Abstraktion zum schnellen seriellen patchen von Shadern	31
3.22	Pure Data Patch: Projektion	32
3.23	Einfache geometrische Verzerrungen zur Korrektur	33
3.24	Grafische Benutzeroberfläche für (a) einfache geometrische Verzerrungen und (b) Erstellung der Schablone.	33
4.1	Pd Patch zur Beleuchtung des Tänzers	37
4.2	Technischer Aufbau für <i>Monster</i>	38
4.3	Monitor Screenshot	39
4.4	GEM Ausgabefenster Einstellungen für Preview	40
4.5	Der Korg nanoKONTROL wurde für <i>Monster</i> in drei Sektoren unterteilt: Die Anpassung für die Beleuchtung, das Kamerabild und die Projektion.	40
4.6	Akai LPD8 mit Beschriftungen für <i>Monster</i>	40
4.7	Choreographie für <i>Monster</i>	41
4.8	Überarbeiteter Technischer Aufbau für <i>Monster</i>	45