



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Bericht

Julia Pressburger
Interaktive Kunst

Julia Pressburger
Interaktive Kunst

Bericht eingereicht im Rahmen der Seminar Ringvorlesung
im Studiengang Master of Science - Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Kai von Luck

Abgegeben am 8. März 2009

Inhaltsverzeichnis

1 Einführung	4
1.1 Motivation	4
2 Hauptteil	6
2.1 Seamless Interaction	6
2.2 Technik und Technologie	8
2.2.1 Quartz Composer	8
2.2.2 Open Sound Control	9
2.2.3 Kamera zur Bewegungserkennung	11
2.3 Ambient Awareness	13
3 Fazit und Ausblick	14
Literaturverzeichnis	15

1 Einführung

Das Ambient Awareness Projekt der Hochschule für Angewandte Wissenschaften Hamburg im WS 2008/2009 war ein interdisziplinäres Pilotprojekt zwischen den Fakultäten *Design*, *Medien und Information* und *Technik und Informatik*.

Die Aufgabe bestand darin eine interaktive Ausstellung zu erschaffen. Der Besucher sollte auf neue und für ihn vielleicht ungewohnte Weise mit der Technik interagieren. (Pressburger und Tennstedt, 2009)

1.1 Motivation

Die Motivation sich mit Interaktionstechniken und passenden Design-Methoden für Interaktive Kunst zu beschäftigen kam durch das Master-Projekt Emotional Tent. Die Herausforderung lag darin, nach einer kurzen Einarbeitungsphase für einen maximalen Effekt auf der Projekt-Präsentation am 6. Februar 2009 zu sorgen. Nach den Recherchen (siehe (Pressburger, 2009)) kristallisierte sich eine Methode heraus: Die Kamera-/Bild-Erkennung als „Sensor“, die Sensor-Verarbeitung im Quartz Composer und die Breitstellung der Sensor-Daten über das OSC-Protokoll.

Durch Integration dieser Sensor Methode in zwei Einzelprojekte waren zwei Arten von Aktionsmodellen gefordert. Zuerst die grafische Ausgabe, dies war auch im

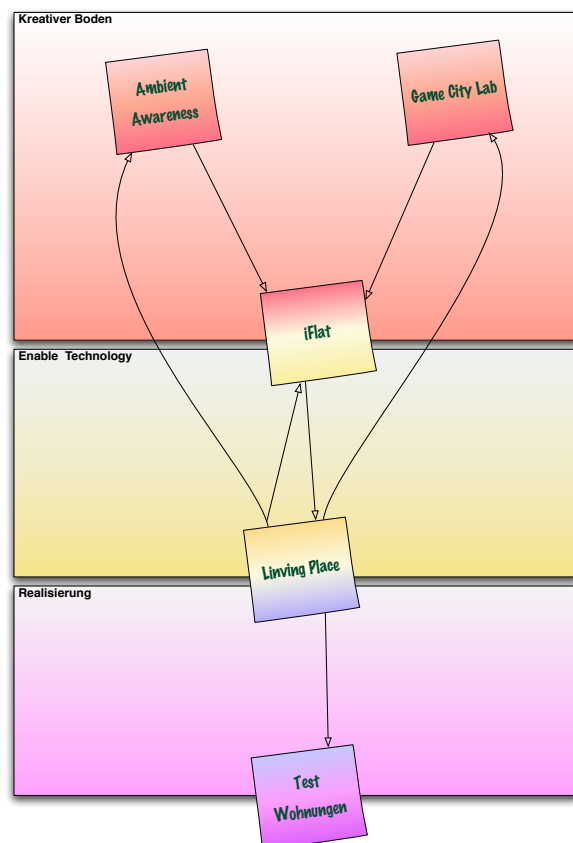


Abbildung 1.1: **Ambient Awareness** im Forschungsschwerpunkt Ambient Assistant Living

Quartz Composer möglich, des Weiteren Sound als Reaktion auf die Sensor-Daten (siehe Abbildung [2.2](#)).

Es galt nun herauszufinden ob die Entscheidung für den Quartz Composer die richtige war. Wie flexibel würde er auch für Aufgaben einsetzbar sein, für die er nicht gedacht war. Würden Probleme auftauchen die es schwierig machen würden funktionierende Prototypen zur Ausstellung fertigzustellen?

2 Hauptteil

2.1 Seamless Interaction

„Seamless Interaction“ als Interaktionstechnik in einem User-Centered Interaktionsmodell, ermöglicht die unmittelbare Interaktion zwischen Benutzer und Technik/Computer aus Sicht des Benutzers. Wenn richtig umgesetzt, ist keine Lernphase nötig. Der Benutzer interagiert/steuert intuitiv.

Für die Umsetzung des Interaktionsmodelles mit Kamera-Erkennung ist das Interfacedesign eine großer Herausforderung. Welche Benutzeraktion führt zu welcher Reaktion?

Für das Projekt konnte das Interface stark vereinfacht werden. Es mussten z. B. keine Gesten erkannt werden. Von Bedeutung war nur, wo sich innerhalb des Kamera-Bildes ein Besucher befand, der sich bewegt.

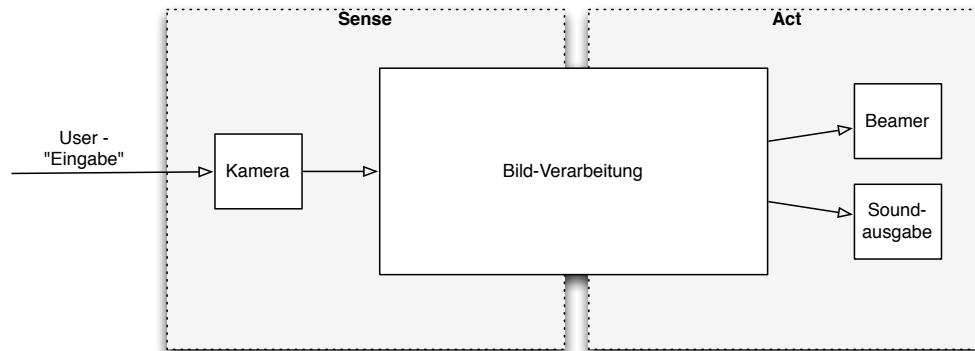


(a) **Particle** Skinsight



(b) **Sound** Little Vintage-Garden

Abbildung 2.1: Fotos zur Ausstellung Ambient Awareness ©Jörg Raasch

Abbildung 2.2: **Sense-Act-Modell**

Sense-Act-Modell für die Anwendungsszenarien

Die Installationen sind als reaktive Systeme (Abbildung 2.2) aufgebaut. Sensor ist die Kamera, deren Daten werden verarbeitet (siehe Kapitel 2.2) und die Reaktion ist:

1. Partikel im Beamerbild (Abbildung 2.1a)
2. Soundausgabe (Abbildung 2.1b)

Die für die Realisierung dieser Systeme verwendeten Techniken und Technologien werden im folgenden Kapitel näher beschrieben. Die genauere Beschreibung „Signal-Verarbeitung“ findet sich in Kapitel 2.2.3.

2.2 Technik und Technologie - Quartz Composer & Open Sound Control

Auf Grund des hohen Zeitdrucks, wurde der Quartz Composer als Technologie für die Realisierung der Installationen ausgewählt. Von allen überprüften Technologien (siehe Bericht zu Anwendungen 2 ([Pressburger, 2009](#))) schien er den geringsten Aufwand zur Einarbeitung, mit dem besten Ergebnis zu ermöglichen. Zur Kommunikation zwischen den Komponenten fiel die Wahl auf das Open Sound Control (OSC) Protokoll.

2.2.1 Quartz Composer

Der Quartz Composer gehört in das Development-Softwarepaket von Mac-OSX. Die Development-Tools sind seit der Betriebssystem-Version 10.4 dabei oder können bei Apple.com in der ADC (Apple Developer Connection: [ADC - Xcode Tools](#)) nach Registrierung frei heruntergeladen werden.

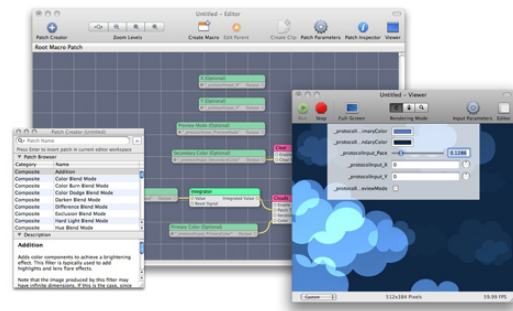


Abbildung 2.3: **Quartz Composer** (Apple Computer, 2007)

Quartz Composer ist eine grafische Entwicklungsumgebung. Compositions werden aus „Patches“ zusammengesetzt und können dann einfach im Quartz Composer ausgeführt, im Quartz Composer Viewer angezeigt oder mit Quicktime abgespielt werden (nur möglich bei nicht-interaktiven Compositions). Patches sind im Grunde grafische Repräsentationen von Sub-Routinen aus den Grafik-Frameworks des Betriebssystems.

Abbildung 2.3: **Quartz Composer** (Apple Computer, 2007)

Apple hat OpenGL ¹ fest in sein Betriebssystem integriert. Dies führt dazu, dass zur Ausführungszeit (wenn der Rechner eine Grafikkarte enthält, die OpenGL implementiert hat) Befehle von der CPU an die GPU (Grafik Processing Unit) ausgelagert werden und im Hintergrund

¹OpenGL (Open Graphics Library) ist eine Spezifikation für eine plattform- und programmiersprachenunabhängige Programmierschnittstelle zur Entwicklung von 3D-Computergrafik. Der OpenGL-Standard beschreibt etwa 250 Befehle, die die Darstellung komplexer 3D-Szenen in Echtzeit erlauben. Zudem können andere Organisationen (zumeist Hersteller von Grafikkarten) proprietäre Erweiterungen definieren. Die Implementierung des OpenGL-API erfolgt in der Regel durch Systembibliotheken wie Mesa, auf einigen Betriebssystemen auch als Teil der Grafikkarten-Treiber. Diese führen entsprechend Befehle der Grafikkarte aus, insbesondere müssen auf der Grafikkarte nicht vorhandene Funktionen durch die CPU emuliert werden. [Wikipedia - OpenGL](#)

direkt von der Grafikkarte ausgeführt werden. Dies macht die Ausführung performanter. ([Apple Computer, 2007](#))

2.2.2 Open Sound Control

Open Sound Control (OSC) ist ein einfaches Kommunikationsprotokoll. Es ist optimiert für den Echtzeit-Einsatz in IP-Netzwerken. Haupteinsatzgebiet ist das Musizieren in verteilten Systemen. Es kann aber auch zur Interprozesskommunikation und innerhalb von Anwendungen genutzt werden. Für die Typisierung der OSC Daten stehen folgende Typen zur Verfügung:

- Integer (32 bit)
- Float (32 bit)
- OSC String (ASCII Zeichenkette)
- OSC Timetag (64 bit Festkomma nach NTP)
- OSC Blob (32 Längenangabe, plus beliebige Anzahl Bytes, plus 0-3 Zero-Padding Bytes)

Es gibt auch nicht Standard-konforme Typen wie z.B. in OSC verpackte MIDI-Nachrichten, Boolean-Typen und Arrays. ([Sukale, 2008](#))

Um OSC mit dem Quartz Composer nutzen zu können, muss man die limitierte Implementation von OSC im Quartz Composer handhaben (siehe Abbildung 2.4).

Wie man in der Abbildung sehen kann, stehen mit dem in QC (Quartz Composer) zur Verfügung stehendem Patch, nur

- String
- Boolean
- Index
- Float
- und Array of Float

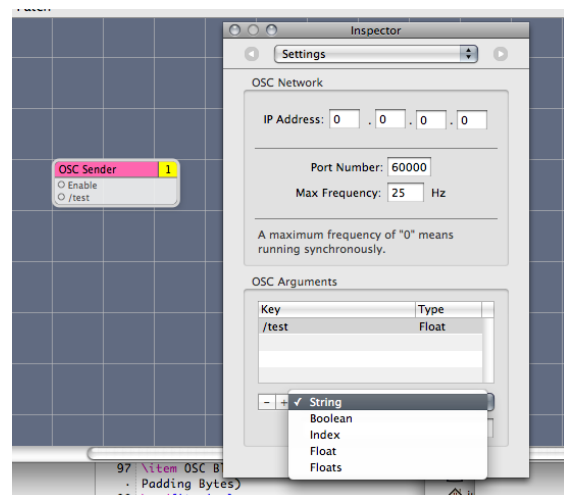


Abbildung 2.4: Implementation des OSC im Quartz Composer

zur Verfügung.

Die eingeschränkte Implementierung von OSC machte bei der Übertragung von MIDI-Noten Probleme. Es wurde ein OSC→MIDI Wandler benötigt der die Note aus einer Float-Struktur auslesen kann. Freie Implementierungen erwarteten eine Struktur von Integer-Werten.

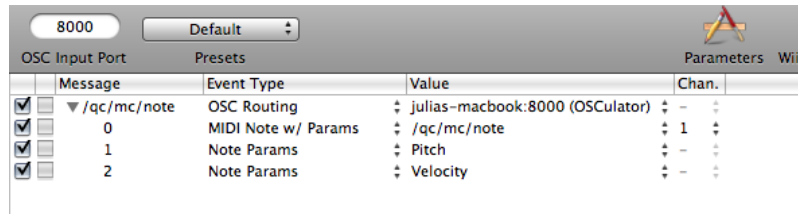
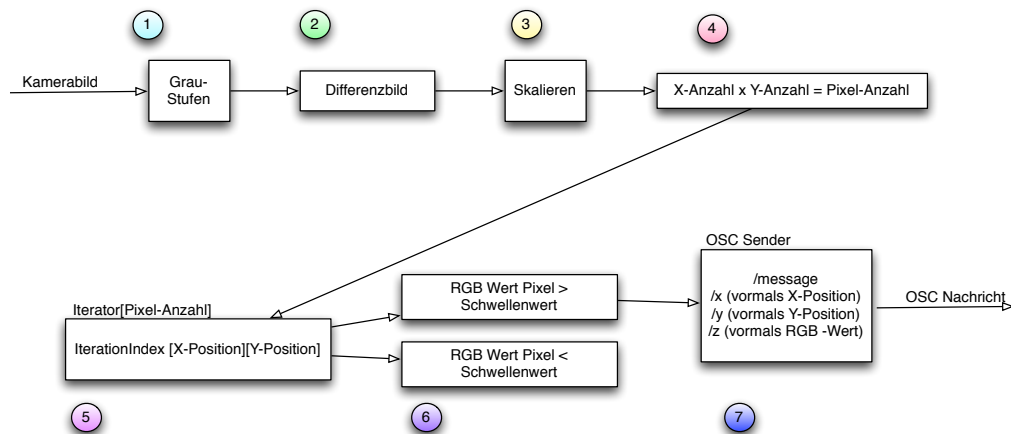


Abbildung 2.5: **OSCulator** MIDI-Note mit Parametern

Für den Mac gibt es eine Lösung. Diese ist nicht frei und musste für die Ausstellung erworben werden.

Abbildung 2.6: **Bewegungserkennung** schematisch

2.2.3 Kamera zur Bewegungserkennung

Die Bewegungserkennung (siehe Abbildung 2.6) funktioniert vereinfacht wie folgt:

1. Das empfangene Kamerabild wird in ein Gray-Scale Bild umgewandelt.
2. Es wird ein Differenz-Bild erstellt, aus dem aktuellen und dem vorangegangenen Bild.
3. Das entstandene Differenz-Bild wird kleiner skaliert. Für höhere Genauigkeit in der Bewegungserkennung wenig, für weniger Genauigkeit mehr.
4. Die Pixel-Anzahl des Bildes wird berechnet.
5. Das Bild wird pixelweise in einen Iterator gegeben (Iterator[Pixelanzahl], Iterator-Index[X-Position][Y-Position]) und der Iterator wird durchgegangen.
6. Der RGB-Wert jedes Pixels wird mit einem Schwellenwert verglichen, ist der Schwellenwert überschritten
7. werden dem OSC-Sender Werte übergeben.

Die Werte, die dem OSC-Sender übergeben werden, sind unterschiedlich. Sie hängen vom Reaktionssystem ab. Für die grafische Reaktion wird Folgendes übergeben:

- X-Position → X
- Y-Position → Y
- RGB-Wert → Z

Für die Sound-Reaktion werden die Informationen aus X-Position, Y-Position und RGB-Wert verwendet, um eine Midi-Note mit den Eigenschaften Pitch und Velocity zu erstellen, die dann dem OSC-Sender übergeben wird.

Der OSC-Sender stellt somit die Daten für die Reaktion des Systems zur Verfügung. Die Daten können sowohl innerhalb des Quartz Composers von einem OSC-Receiver, als auch von anderen Anwendungen oder anderen Rechnern empfangen werden, für den Fall, dass das Ausführen der Reaktion (z. B. das Rendern eines Particle-Systems) auf einem anderen Rechner stattfinden soll.

Bild-Analyse und Rendern der Reaktion auf unterschiedlichen Systemen laufen zu lassen, kann unter Umständen sinnvoll sein. Beide Aktionen erfordern eine hohe Performance, je nach der Genauigkeit, die für die Bewegungserkennung gefordert ist. Ist die Latenz zwischen Aktion des Besuchers und Reaktion des Systems zu hoch, wird nicht das gewünschte Interaktionserlebniss erreicht.

2.3 Ambient Awareness als Forschungsschwerpunkt

Ambient Awareness ist ein Forschungsschwerpunkt aus dem Bereich *Ambient Intelligence* (siehe Abbildung 1.1). In einer Umgebung mit selbst gesteckten Regeln und Bedingungen kann man zunächst Ideen ausformen und ausprobieren. Für das Beispiel der Ausstellung, des Durchgangs vom WS 2008/2009, die Installationen die mit einer Kamera als Sensorik laufen, bedeutet das, wie in Kapitel 2.1 schon kurz erwähnt, keine Gestenerkennung. Das Szenario ist stark vereinfacht. Es kann zuerst ausprobiert werden ob die Ideen mit der gewählten Hardware und der gewählten Technologie funktionieren.

Ideen die sich bewährt haben wandern in die *iFlat*. Dort werden sie in die Laborumgebung integriert. Doch vorher müsste untersucht werden in welchen Szenarien die neue Methode einzusetzen wäre. Für das Beispiel der Bewegungserkennung durch eine Kamera, würde sich die Frage stellen, ob dies als Erweiterung der Sensorik der *iFlat* sinnvoll wäre? Könnte sie z. B. nach Erweiterung um Gestenerkennung, als Alternative oder Ergänzung zur Fernbedienung agieren?

Hat sich eine Technologie in der *iFlat* bewährt wird sie in Zukunft ins *Living Place*, die Musterwohnung für *Ambient Intelligence* der Hochschule für Angewandte Wissenschaften Hamburg, ziehen. In Realexperimenten werden dort die Konzepte aus der *iFlat* getestet. *Living Place* wird wie eine richtige Wohnung aufgebaut sein. Die Experimente werden dort in einer authentischen Umgebung stattfinden. *Living Place* wird außerdem mit den neusten Technologien aus dem Bereich Usability ausgestattet sein, so können die Realexperimente umfassend ausgewertet werden. (Wendt und Stegelmeier, 2008)

3 Fazit und Ausblick

Für die Entwicklung von zwei interaktiven Installationen, mit einer Kamera als Bewegungssensor, den Quartz Composer und zur Kommunikation OSC (Open Sound Control) zu verwenden stellte sich als gute Entscheidung heraus. Zu Ausstellung waren für beide Installationen funktionierende Prototypen fertig. Auch wenn es kleine Probleme mit der limitierten Implementierung von OSC im Quartz Composer gab, konnten diese durch den Einsatz einer kommerziellen Software für die Konvertierung der OSC-Daten in MIDI-Noten behoben werden.

Jetzt ist es interessant auszuprobieren, ob man die Prototypen erweitern kann. Es fehlt noch etwas an Interaktionsgefühl. Wie kann man dies noch verstärken? Ist es möglich die Prototypen um Gestenerkennung zu erweitern? Oder ist es sinnvoll auf openFrameworks (siehe ([Pressburger, 2009](#)), Kapitel 2.2.3) umzusteigen?

Literaturverzeichnis

- [Apple Computer 2007] APPLE COMPUTER, Inc.: *Leopard Developer Graphics & Media Overview*. October 2007. – URL <http://developer.apple.com/leopard/overview/graphicsandmedia.html>
- [Pressburger 2009] PRESSBURGER, Julia: Digital Art Design / HAW Hamburg, Department Informatik. URL <http://users.informatik.haw-hamburg.de/~ubicom>, February 2009. – Forschungsbericht
- [Pressburger und Tennstedt 2009] PRESSBURGER, Julia ; TENNSTEDT, Sven: Projektbericht - Emotional Tent - Ambient Awareness / HAW Hamburg, Department Informatik, Department Design. March 2009. – Forschungsbericht
- [Sukale 2008] SUKALE, Martin: *Konstruktion eines Netzwerkes eingebetteter Systeme für interaktives Design*. www.informatik.haw-hamburg.de, HAW Hamburg, Department Informatik, Diplomarbeit, August 2008. – URL <http://users.informatik.haw-hamburg.de/~ubicom/arbeiten/diplom/sukale.pdf>
- [Wendt und Stegelmeier 2008] WENDT, Piotr ; STEGELMEIER, Sven: iFlat - Eine dienstorientierte Architektur für intelligente Räume / HAW Hamburg, Department Informatik. 2008. – Forschungsbericht