Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Projektbericht

Marcus Rödiger

Optimierungsstrategien für IR-Touchsysteme

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Marcus Rödiger

Optimierungsstrategien für IR-Touchsysteme

Projektbericht eingereicht im Rahmen der Veranstaltung Projekt
im Studiengang Master Of Science Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Erster betreuender Professor : Prof. Dr. rer. Kai von Luck
Zweiter betreuender Professor : Prof. Dr. Gunter Klemke

Abgegeben am 1. März 2009

Inhaltsverzeichnis

Abbildungsverzeichnis

1. Einführung

1.1. Motivation

Die Interaktion zwischen Mensch und Maschine ist bis heute eine der größten Hürden beim Arbeiten mit Computersystemen. Die gängigen Eingabemethoden des Benutzers, zum Beispiel mit Maus und Tastatur, erfordert immer noch einen hohen Grad an Abstraktion und kognitivem Lernaufwand. Einen guten alternativen Ansatz stellen dabei Bildschirme mit berührungsempfindlicher Oberfläche dar. Vogt (2008) Mit der sogenannten Touchbildschirmtechnik ist der Anwender in der Lage mit dem System auf sehr natürliche Weise zu interagieren, welches dazu führt, dass mit nur wenig Lernaufwand der Benutzer in die Lage versetzt wird mit dem System effektiv zu kommunizieren. In der Hochschule für Angewandte Wissenschaften existieren Touchbildschirme auf der Basis von Infrarottechnik, welche jedoch für den Einsatz als Multitouchbildschirm nur bedingt geeignet sind.

1.2. Ziel des Projektes

Ziel des Projekts ist die Entwicklung eines Systems zur effektiven und intuitiven Benutzerinteraktion zwischen Anwender und Maschine mit Hilfe von touchempfindlichen Bildschirmen. Die zur Verfügung stehende Hardware ist von sich aus nicht für den Einsatz als Multitoucheingabegerät geeignet. In dem Projekt geht es vor allem darum Techniken und Strategien zu entwickeln, um die Erkennung von Multitoucheingaben auf der vorhandenen Hardware zu verbessern. Weiteres Ziel ist es Anwender mit dem entwickelten System zu konfrontieren - ohne oder mit sehr wenig Anleitung - um an ihrem Verhalten die Benutzbarkeit und den „Motivationsfaktor“ dieser neuen Technik zu untersuchen.

Für diese Untersuchung wird eine prototypische Anwendung entwickelt und im Rahmen der Ausstellung „Ambient Awareness“ den Benutzern zur Verfügung gestellt. Anschließend die Reaktionen der Benutzer ausgewertet.

2. Aufbau des Systems

In diesem Abschnitt wird der Aufbau des Systems beschrieben. Im ersten Teil wird die verwendete Ein- und Ausgabehardware und die sich aus ihr ergebenden Probleme bei der Benutzung als Multitoucheingabegerät beschrieben. Anschließend wird ein kurzer Überblick über mögliche Hardwarealternativen gegeben. Im zweiten Teil wird auf die verwendete Software eingegangen, welche für das System benutzt wird.

2.1. Hardware

Als Hardware wird ein 42"Flachbildschirm Flatman TFT mit Infrarot basiertem 42"Touchaufsatz und passendem Ständer eingesetzt. (Abbildung 2.1)

Der Touchaufsatz wird von der Firma IRTOUCHSYSTEMS (IRTOUCHSYSTEMS Co. Ltd. (2009)) vertrieben. Es ist möglich, den Touchaufsatz unabhängig vom Bildschirm zu betreiben. Die Benutzung eines Bildschirms mit darauf montiertem Touchaufsatz bietet dem System jedoch die Möglichkeit, auf die Eingaben des Benutzers durch ein direktes Feedback zu reagieren. Als Beispiel sei hier das Verschieben eines Bildes auf dem Bildschirm genannt. Wenn der Anwender das Bild auf dem Bildschirm berührt, verfärbt es sich, um dem Anwender direkt zu signalisieren, dass das Bild nun bewegt werden kann. Wenn der Benutzer seinen Finger nun über den Bildschirm bewegt, wird das Bild entsprechend nachgeführt. Durch diese unmittelbare Kommunikation, wird in einfacher Weise ein Regelkreislauf aufgebaut - mit dem Anwender als Teil von diesem.

2.1.1. Rechenhardware

Für das Auslesen des Aufsatzes wird ein einfacher low cost PC benötigt. Da jedoch die Analyse und Optimierung der Touchdaten und vor allem das Präsentationssystem hohe Rechenleistung erfordern, wird in diesem Fall ein Mac Pro mit entsprechendem 3D-Hardwarebeschleuniger eingesetzt.

Abbildung 2.1.: Touchbildschirm mit Ständer

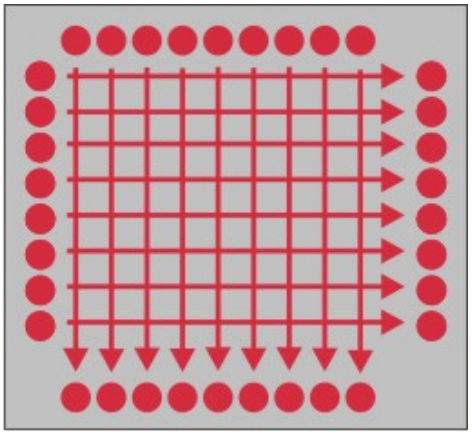

Abbildung 2.2.: Infrarotgitternetz

2.1.2. Touchaufsatz

Zur Objekterkennung nutzt der Aufsatz zwei Reihen von Infrarotlichtschranken, welche - jeweils auf der X- und Y-Achse liegend - ein Infrarotsensorgitter oberhalb der Scheibe des Bildschirms aufspannen. (Abbildung 2.2)

Genauere Informationen über die Eigenschaften der Hardware findet man unter Gehn (2008) im Abschnitt 2.2 und 2.3.1.

Stefan Gehn zeigt in seiner Arbeit unter anderem die Probleme der infrarotbasierten Technik beim Einsatz als Multitouchschirm auf. Im folgendem Abschnitt soll auf die wichtigsten dieser Probleme eingegangen werden.

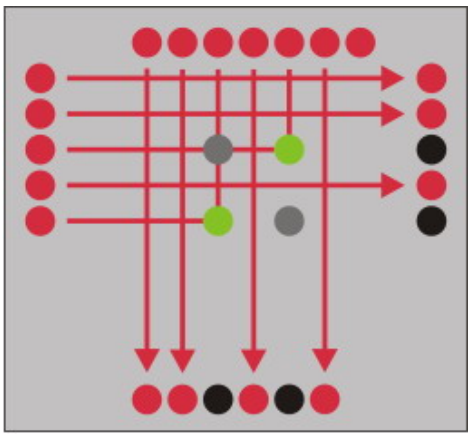

Abbildung 2.3.: Mehrdeutigkeit bei mehr als einem Druckpunkt

Probleme der eingesetzten infrarotbasierten Technik

Mehrdeutigkeit bei mehr als einem Druckpunkt

Ziel der Anwendung ist es, die genaue Position des Druckpunktes in X- und Y-Richtung zu erkennen. Bei der hier benutzten IR-Technik wird versucht, ein Gitter mit n*m möglichen Druckpunkten mittels n+m Sensoren zu erfassen. Die Menge der Zustände der Druckpunkte ist surjektiv zur Menge der Zustände der Sensoren. Bei mehr als einem Druckpunkt entsteht daher das Problem, dass die einzelnen Druckpunkte von den Sensoren nicht mehr eindeutig erfasst werden. Für einen Sensorzustand sind teilweise mehrere Kombinationen aus Druckpunkten möglich. (Abbildung 2.3)

Die Sensoren können also in bestimmten Fällen nicht den realen Zustand des Druckpunktgitters erfassen und die Interpretation erzeugt so nicht existierende Schattenpunkte.

Probleme durch asynchrone Messung der X- und Y-Werte

Die eingesetzte Hardware erfasst zu erst nacheinander die einzelnen Werte in X-Richtung und danach nacheinander die Werte in Y-Richtung. Durch die zeitliche Verzögerung beim Auslesen kommt es in einzelnen Fällen beim Unterbrechen der Lichtschranken zu Fehlmessungen. Es kann also passieren, dass eine Lichtschranke in X-Richtung als nicht unterbrochen gemessen wird, aber die korrespondierende Lichtschranke in Y-Richtung eine Unterbrechung wahr nimmt. Dieses resultiert daraus, dass in der Zwischenzeit der Finger beide Lichtschranken unterbrochen hat. Das Resultat ist die Übertragung eines fehlerhaften Messergebnisses. (Abbildung 2.4) Dieses Problem tritt normalerweise immer nur in jeweils einem Messzyklus auf.

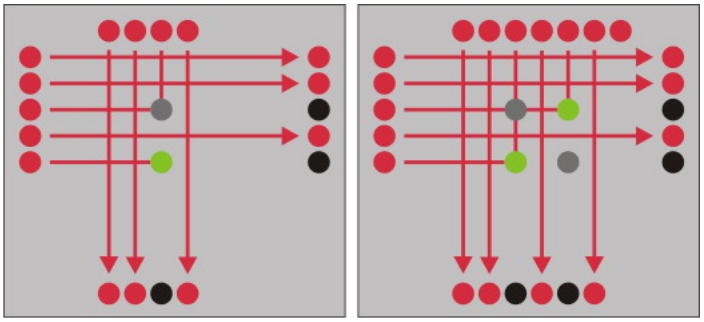

Abbildung 2.4.: Links gemeldetes Ergebnis - Rechts erwartetes Ergebnis

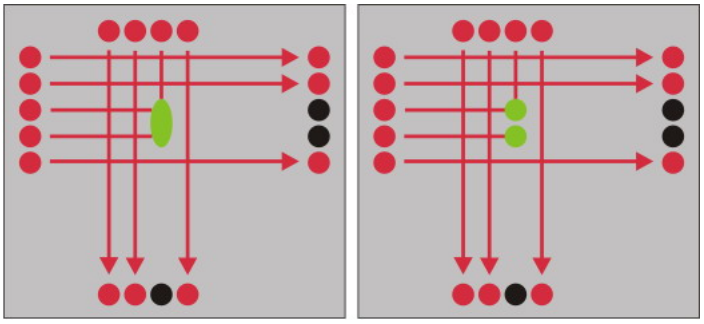

Abbildung 2.5.: Links Ein großer Druckpunkt wird gemessen - Rechts reale Druckpunkte

Erkennung von zwei Druckpunkten als einen
Manchmal kann es vorkommen, dass zwei Druck- oder Schattenpunkte dicht nebeneinander liegen und Ihr Abstand in X-oder Y-Richtung das Rastermaß von 4 mm unterschreitet. Dieses führt dazu, dass bei der Messung und der anschließenden Interpretation beide Punkte zu zwei großen, übereinander liegenden Punkten verschmelzen. (Abbildung 2.5)

2.1.3. Anschlusskabel des Touchaufsatzes

Der Touchaufsatz besitzt einen seriellen Anschluss, über den die Statusinformation der Lichtschranken zyklisch abgefragt werden können. Da moderne Rechner nicht mehr über eine serielle Schnittstelle verfügen, wird ein passender RS232 zu USB Konverter verwendet. In diesem Fall wird konkret ein Konverter der Firma Entrega Technologies Inc Entrega Technologies Inc (2009) mit der Modelnummer U232-P9 eingesetzt. Es kann jedoch auch jeder andere RS232-USB Konverter benutzt werden, der im System eine COM-Schnittstelle simuliert. Darüber hinaus muss zur Stromversorgung des Touchaufsatzes ein spezielles Adapterkabel verwendet werden. Stefan Gehn Gehn (2008) beschreibt im Anhang die benötigte Pinbelegung dieses Adapters, welcher über einen zweiten USB Port den Aufsatz mit der nöti-

gen Spannung versorgt. In diesem Fall wurde einfach ein altes serielles Kabel entsprechend umgebaut.

2.1.4. Hardwarealternativen zur Multitoucheingabe

Es wurde gezeigt, dass die zur Verfügung stehende Hardware Multitoucheingabe nur unzureichend unterstützt. Es gibt jedoch bereits Systeme, welche die Position von mehr als einem Druckpunkt erkennen können. Teilweise sind diese Systeme sogar in der Lage, diese Druckpunkte zu einem bestimmten Benutzer zuzuordnen. Rödiger (2007) In den meisten Fällen, sind diese Alternativen entweder auf kleine Bildschirmdurchmesser beschränkt oder haben einen sehr großen und damit unpraktischen Aufbau zur Folge. Daher ist das Ziel dieser Arbeit die vorhandene IR-Technik zu verbessern.

2.2. Software

2.2.1. Softwareumgebung

Aufgrund von guten Erfahrungen von Mohammed Ali Rahimi und Matthias Vogt (Rahimi (2008), Vogt (2008)) wird als Sprache für den Prototypen C# verwendet. C# ist ein guter Kompromiss aus Performance und objektorientierter Sprache. Darüber hinaus gibt es für C# ein Direkt X SDK zur Erstellung dreidimensionaler Oberflächen. Als Entwicklungsumgebung wird Visual Studio .NET 2009 eingesetzt. Resultierend aus der Wahl der Programmiersprache und der besseren Treiberunterstützung für RS232-USB Konverter wird Windows XP als Betriebssystem gewählt.

3. Entwicklung einer prototypischen Anwendung

In diesem Kapitel wird der Aufbau und die Eigenschaften der entwickelten Anwendung beschrieben. Im ersten Teil werden Softwarekonzepte vorgestellt und bewertet, welche die aufgezeigten Probleme beim Betrieb des Touchaufsatzes im Multitouchumfeld kompensieren sollen. Anschließend werden die entwickelten Konzepte in ihrer Kombination bewertet und ein erstes Resümee über die Qualitätsverbesserung gezogen. Im zweiten Teil wird kurz auf die grafische Repräsentation und die Entwicklung einer Beispielanwendung eingegangen.

Der Prototyp besteht aus zwei Modulen: Das erste Modul liest den Datenstrom vom Touchaufsatz aus der seriellen Schnittstelle aus und transformiert und interpretiert diesen. Die daraus erzeugten Touchereignisse werden dann an das zweite Modul weiter gegeben. Das zweite Modul reagiert entsprechend auf die Touchereignisse und zeigt die Ergebnisse dem Benutzer auf dem Bildschirm an.

3.1. Modul 1 - Auslesen und Aufbereiten der Toucheingabe

Das Protokoll des Touchaufsatzes liefert periodisch ein Paket mit den Sensorinformationen des Touchaufsatzes. (Weitere Informationen über das Übertragungsprotokoll des Touchaufsatzes Siehe Anhang A) Aus diesen Paketen wird eine Liste mit allen für dieses Frame möglichen Druckpunkten generiert.

Im Abschnitt 2.1.2 wurde gezeigt, dass die gemessenen Rechtecke nicht immer den wirklichen Druckpunkten entsprechen. Daher werden im Folgendem Methoden vorgestellt, welche anhand von Regeln und Annahmen versuchen, echte Druckpunkte von Schattendruckpunkten zu unterscheiden.

3.1.1. Regeln zur Unterscheidung von Echten- und Schattendruckpunkten

Ein echter Druckpunkt ist der rechteckige Bereich, welcher vom Benutzer im Sensorgitter wirklich unterbrochen wurde. Ein Schattendruckpunkt ist ein Rechteck an einer Position, welche gar nicht vom Benutzer berührt wurde, jedoch von den Sensoren als eine Unterbrechung an diesem Punkt erkannt wurde.

Im folgenden werden die Regeln beschrieben, um einen echten Druckpunkt zu identifizieren.

Regel 1 - Singletouch

Wenn in diesem Frame nur ein Rechteck erkannt wurde, ist dieses immer ein echter Druckpunkt, da keine Mehrdeutigkeiten auftreten können.

Regel 2 - Einbeziehung von alten Frames

Bei dieser Regel wird das vorherige Frame herangezogen. Es findet also eine Auswertung über die Zeit statt. Dazu wird geprüft, ob es ein aktives Rechteck aus dem alten Frame gibt, welches ein Rechteck des aktuellen Frames unmittelbar überlappt. Ist dies der Fall, ist das aktuelle Rechteck auch aktiv.

Die Regel beruht auf der Annahme, dass der Benutzer mit seinen Fingern eine gewisse Trägheit besitzt. Wenn der Benutzer im vorherigen Frame einen echten Druckpunkt erzeugt hat, welcher erkannt wurde, wird im aktuellen Frame mit hoher Wahrscheinlichkeit der neue Druckpunkt in der unmittelbaren Umgebung des alten Druckpunktes liegen. Wenn dies der Fall ist, wird der neue Druckpunkt ebenfalls als echt anerkannt. Dadurch wird zum Beispiel die Bewegung eines Fingers über den Bildschirm erfasst und aufgezeichnet.

Leider ist dieses Verfahren stark von der Größe des Druckpunktes und der Abtastrate der Hardware abhängig, da bei zu schnellen Bewegungen der neue Druckpunkt nicht mehr zwingend über dem alten Druckpunkt liegt. Er wird so vom System als neuer Druckpunkt erkannt. Zur Kompensation dieses Problems wird jeder Druckpunkt aus dem alten Frame um seinen jeweiligen Richtungsvektor verschoben und anschließend wie oben mit dem aktuellen Druckpunkt auf eine mögliche Überlappung hin getestet. Der Richtungsvektor eines Druckpunktes ist die Richtung und die Geschwindigkeit, in die sich der Druckpunkt relativ zu seinem Vorgänger bewegt hat.

Diese zusätzliche Auswertung führt dazu, dass schnellere Bewegungen erfasst werden können. Jedoch führt sie zu Problemen, wenn der Benutzer abrupt die Richtung ändert. Da

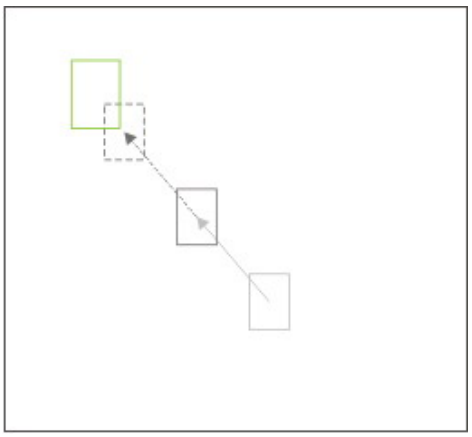

Abbildung 3.1.: Abtastfehler und Kompensation

dieses jedoch seltener der Fall ist und die Erkennungsrate dadurch massiv erhöht wird, ist dieser Fehler akzeptabel. (Abbildung 3.1)

Die Abbildung zeigt den aktuellen Druckpunkt in grün. Den Druckpunkt aus dem letzten Frame in dunkelgrau. Den verschobenen Druckpunkt aus dem letzten Frame in gestricheltem dunkelgrau. Und den Druckpunkt aus dem vorletzten Frame in hellgrau. In diesem Beispiel wird der neue Druckpunkt dem alten Druckpunkt zugeordnet, da sich der aktuelle und der alte verschobene Druckpunkt überdecken.

Regel 3 - Auswertung der Eckpunkte

Die Sensoren erkennen immer ein Druckpunktfeld, in dem die Echten- und Schattendruckpunkte in einem Gitter nebeneinander oder untereinander angeordnet sind. (Abbildung 3.2) Die vier Druckpunkte, welche alle anderen einschließen, die also in den jeweiligen Ecken liegen, haben besondere Eigenschaften. Einer dieser Eckpunkte muss immer ein echter Druckpunkt sein. Das Anwenden von Regel 2 kann dieser eine Druckpunkt bereits gefunden werden. Wenn nun das Druckpunktgitter aus einer gleichen Anzahl von Druckpunkten in X sowie in Y Richtung besteht (Abbildung 3.2) und wir bereits einen echten Eckpunkt ausfindig gemacht haben, dann ist der diesem Eckpunkt gegenüberliegende Eckpunkt mit Sicherheit auch ein echter Druckpunkt. Die Anwendung dieser Regel tritt insbesondere dann auf, wenn erst ein Finger auf den Bildschirm gesetzt wird (Regel 1) und anschließend ein zweiter Finger zusätzlich den Bildschirm berührt. (Regel 2 und 3) Abbildung 3.2 zeigt so eine Situation, in der im vorherigen Frame bereits ein echter Eckdruckpunkt identifiziert wurde und deshalb aktiv bleibt. Druckpunkt 2 wird aufgrund von Regel 3 nun aktiviert. Der mittlere Druckpunkt wird jedoch nicht als ein echter Druckpunkt erkannt. Druckpunkt 3 und 4 können nicht sicher als echte Druckpunkte identifiziert werden. Sie bleiben deshalb in ihrem Zustand.

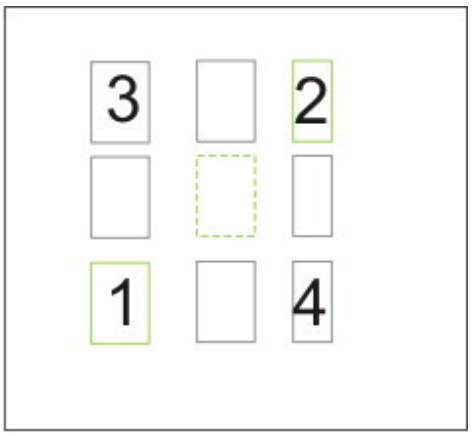

Abbildung 3.2.: Auswertung der Eckpunkte

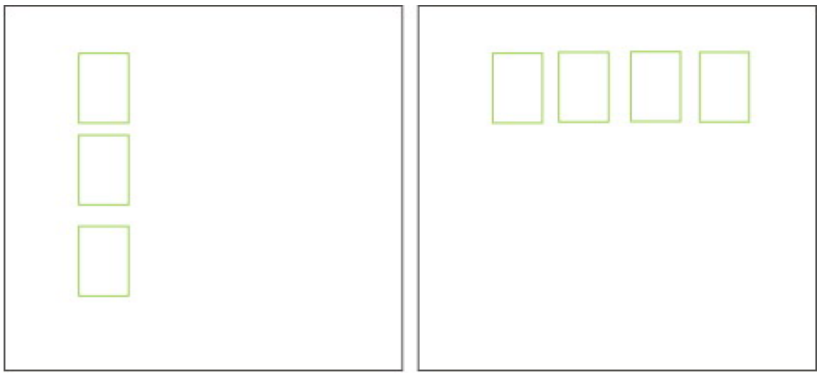

Abbildung 3.3.: Aktive Druckpunkte in horizontaler und vertikaler Richtung

Regel 4 - Druckpunkte liegen nur auf einer Ebene

Die vierte Regel besagt, dass alle Druckpunkte aktiv sind, wenn sie sich auf derselben X oder Y Ebene liegen. (Abbildung 3.3) Aufgrund der Fehlmessung - wie im Abschnitt 2.1.2 beschrieben - muss in diesem Fall die Regel jedoch erweitert werden. Die Regel darf nur Anwendung finden, wenn ähnlich wie bei Regel 2 ein überlappender, aktiver oder inaktiver Druckpunkt aus dem vorherigen Frame vorhanden ist. Dadurch werden die Druckpunkte im ersten Frame erst einmal als Schattendruckpunkte registriert, um dann im folgenden Frame als echte Druckpunkte interpretiert werden zu können. Dieses führt natürlich zu einer leichten Verfälschung der Aktion des Benutzers, da erst das zweite Frame die echten Druckpunkte erkennt und die jeweilige Aktion startet. Dieses ist jedoch akzeptabel, da der Benutzer nur durch ein direktes Feedback den Start einer Aktion erkennt und dieses auch ein Frame später erfolgen kann. (Im Gegensatz zur Maussteuerung, wo der Benutzer durch hörbares Klicken bzw. den Druckwiderstand der Maus erkennt, dass seine Eingabe ausgelöst wurde.)

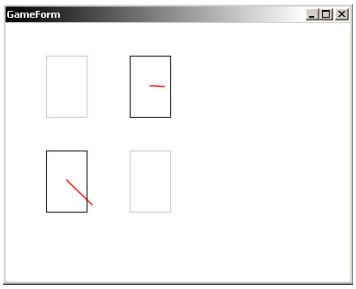

Abbildung 3.4.: Erste grafische Ausgabe

3.1.2. Auswertung der Kombination der Regeln

Im Test des Systems hat sich gezeigt, dass die kombinierte Benutzung der vorherigen Regeln zu einer verhältnismäßig hohen Qualitätsverbesserung geführt haben. Die Benutzung von einem oder meist auch zwei Druckpunkten ist relativ problemlos. In der Praxis hat sich jedoch gezeigt, dass in den meisten Fällen nicht mehr als maximal zwei echte Druckpunkte identifiziert werden können. Die Annahmen über Regel 3 erwiesen sich bei mehr als zwei Druckpunkten als falsch. In einigen Situationen wird der gegenüberliegende Eckpunkt als gedrückt erkannt, obwohl dieses gar nicht der Fall ist. Das Problem - wie im Abschnitt 2.1.2 „Erkennung von zwei Druckpunkten als einen“ beschrieben - konnte nicht kompensiert werden. Dadurch ergeben sich Fehlinterpretationen, wenn zum Beispiel zwei Finger zusammen geführt und wieder auseinander gezogen werden. Trotzdem werden Druckpunkte (bis zum Maximum von zwei gleichzeitig) in den meisten Anwendungsfällen zuverlässig erkannt.

Ein anderer Ansatz, um die auftretenden Fehler zu überspielen, wäre die Anforderung der Anwendung soweit anzupassen, dass das Wissen um eine genaue Position der Druckpunkte nicht erforderlich ist. Ein Beispiel wäre die Benutzung von Wischgesten, welche nur eine präferierte Richtung und keine exakte Lage der echten und falschen Druckpunkte benötigen. In diesem Prototypen sollen jedoch die Druckpunkte eindeutig als echt erkannt werden, um sie so einem jeweiligen Ereignis zuordnen zu können.

3.2. Modul 2 - Grafische Repräsentation

Anfänglich war das Projektziel die Entwicklung eines Prototyps zur verbesserten Benutzung der Touchauflage als Multitouchsystem. Für diesen Zweck wurde eine einfache MVC basierte grafische Repräsentation der vom System erkannten Druckpunkte erstellt. (Abbildung 3.4) Zur Mitte des Projektes wurde das Ziel im Zuge des Gesamtprojektes „Ambient Awareness“ um eine weitere Anforderung erweitert:

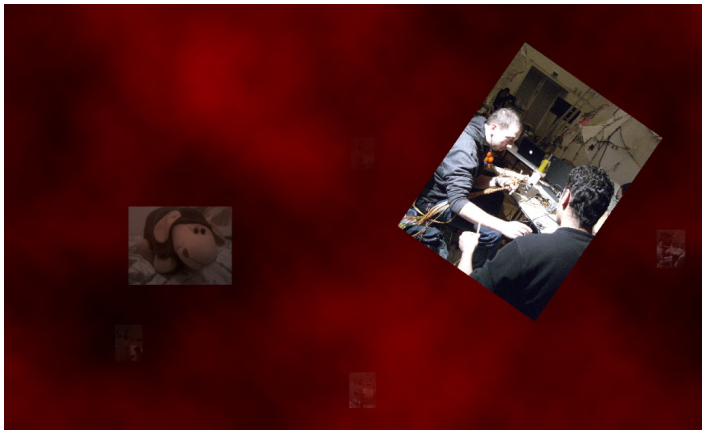

Abbildung 3.5.: Finale grafische 3D - Oberfläche

Erstellung einer Touchanwendung zur Dokumentation und Präsentation der Vorarbeiten des Gemeinschaftsprojektes „Ambient Awareness“.

Der Prototyp wurde um eine grafische 3D - Oberfläche und einen einfachen Gesteninterpreter ergänzt, welche es dem Benutzer ermöglichen durch Bewegung des Fingers auf dem Bildschirm Projektbilder und Projektfotos aus dem Hintergrund nach vorne zu holen, um sie anschließend dort zu betrachten, zu drehen, in X- und Y-Richtung zu bewegen oder wieder zurück in den Hintergrund transferieren zu können. (Abbildung 3.5) Aufgrund der Erkenntnisse bei der Entwicklung des Moduls 1 wurde bei der Erweiterung auf komplexe Multitoucheingabe wie das Vergrößern und Verkleinern von Bildern mit zwei Fingern verzichtet. Alle Aktionen eines Bildes konnten so mit nur einem Finger durchgeführt werden.

3.2.1. Verwendete Technologie Direct X

Für die 3D - Oberfläche wurde Managed Direct X bzw. Managed Direct 3D aus dem Microsoft Direct X SDK[1] verwendet. Die Schnittstelle Direct 3D bietet die Möglichkeit zur Entwicklung komplexer 3D-Anwendung unter Benutzung gängiger Hardwarebeschleuniger. Wie sich im Laufe des Projektes heraus stellte, ist die Unterstützung von Direct X durch die im SDK enthaltende C# Bibliothek nur unzureichend. Darüber hinaus ist die direkte Benutzung der Direkt X Klassen für einfache grafische Anwendung zu komplex und erfordert ein hohes Maß an Einarbeitung und Programmieraufwand. Zur Sammlung von Erfahrungen im Bereich Direkt X Programmierung wurde trotz des Aufwandes an der Benutzung der Schnittstelle festgehalten.

[1] URL http://msdn.microsoft.com/en-us/directx/default.aspx

3.2.2. Direct X Alternativen

Für die Entwicklung eigener grafischer Oberflächen wird im Nachhinein die Verwendung des Windows Presentation Foundation[2] empfohlen. Dieses Framework setzt auf Direkt X auf und kann mit .NET-Sprachen kombiniert werden. Es bietet Methoden und Werkzeuge zur Entwicklung von komplexen 2D oder 3D Benutzeroberflächen.

3.3. Ausblick

Der Prototyp ist in keinster Weise vollständig. Nachfolgend sind einige Maßnahmen aufgelistet, welche in zukünftigen Projekten umgesetzt werden könnten.

- Umstellung der grafischen Oberfläche auf das Microsoft Presentation Foundation Framework.
- Erkennung von anderen Eingabegesten, wie „präferierte Richtung des Benutzers“.
- Umbau des Moduls 1 zu einer eigenständigen Bibliothek, welche von anderen Programmen eingebunden werden kann.
- Hinzufügen von weiteren Regeln, zur Verbesserung der Erkennung von echten Druckpunkten.
- Unterstützung von anderer Touchhardware.
- Erkennung des jeweiligen Benutzers bei der Eingabe.

[2]URL http://msdn.microsoft.com/de-de/netframework/aa663326.aspx

4. Fazit

Die Entwicklung des Prototypen unter Anwendung der erarbeiteten Regeln haben gezeigt, dass eine Verbesserung der Qualität unter bestimmten Voraussetzungen möglich ist. Jedoch kann auch diese Verbesserung die Probleme dieser für den Multitoucheinsatz mangelhaften Hardware nur kompensieren, nicht aber gänzlich lösen. Bei der Verwendung von mehr als zwei Druckpunkten bleibt die Interpretation der Druckpunkte ein Ratespiel, welches wohl nur durch den Einsatz besserer Hardware oder komplexer Softwareverfahren geändert werden kann.

Trotzdem zeigt das Projekt, dass in gewissen Anwendungsgebieten sich mit dieser Hardware- und Softwarkombination brauchbare Ergebnisse schaffen lassen. Auch das Feedback der Benutzer des ersten Prototyps zeigt, dass die Eingabe per Touchbildschirm für viele eine echte Alternative darstellt. Die meisten Anwender haben ohne vorherige Kenntnisse bei der Benutzung des Prototyps sehr gute Resultate erzielt. Dieses ist wohl darauf zurück zu führen, dass Touchbildschirme für den Anwender einfach und intuitiv zu bedienen sind. Aus diesem Grund wird die Interaktion mittels Bildschirmtouchgeräten in Zukunft wohl zunehmen und sich langfristig neben anderen Formen etablieren.

Literaturverzeichnis

[Entrega Technologies Inc 2009] ENTREGA TECHNOLOGIES INC: *Entrega Technologies Inc Website*, 2009. – URL http://www.entrega.com/

[Gehn 2008] GEHN, Stefan: Evaluation einer infrarotbasierten Multitouch-hardware / Hochschule für Angewandte Wissenschaften Hamburg. URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master07-08-proj/gehn/report.pdf, 2008. – Forschungsbericht

[IRTOUCHSYSTEMS Co. Ltd. 2009] IRTOUCHSYSTEMS CO. LTD.: *IRTOUCHSYSTEMS Website*, 2009. – URL http://irtouch.com

[Rahimi 2008] RAHIMI, Mohammed A.: Multitouch: Out of the shelf. Gestenbasierte Interaktion für verfügbare Applikationen / Hochschule für Angewandte Wissenschaften Hamburg. URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master08-09-aw1/rahimi/bericht.pdf, 2008. – Forschungsbericht

[Rödiger 2007] RÖDIGER, Marcus: Multitouch - Technik & Technologien / Hochschule für Angewandte Wissenschaften Hamburg. URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2007/roediger/bericht.pdf, 2007. – Forschungsbericht

[Vogt 2008] VOGT, Matthias: Multitouch-Steuerung moderner Computersysteme / Hochschule für Angewandte Wissenschaften Hamburg. URL http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master08-09-aw1/vogt/bericht.pdf, 2008. – Forschungsbericht

Serial interface touchscreen special protocol

Communication parameters

Interface: RS232
Speed: 38400 baud rate
Start bit: 1bit
Data bit: 8 bits
End bit: 1
No parity

Data structure

Touch status

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6
0xaa	Coordinates of shaded LED on X axis	Coordinates of shaded LED on Y axis	Starting coordinates of the first Touch Point on X axis		Touch Width of the first Touch Point on X axis	
Byte7	Byte8	Byte9	Byte10	……	N-9	N-8
Starting coordinates of the second Touch Point on X		Touch Width of the second Touch Point on X axis		……	Starting coordinates of the second-last Touch Point on Y axis	
N-7	N-6	N-5	N-4	N-3	N-2	N-1
Touch Width of the second-last Touch Point on Y axis		Starting coordinates of the last Touch Point on Y axis		Touch Width of the last Touch Point on Y axis		CRC

Length N=4+ (number of touch points on X axis) *4+ (number of touch points on Y axis) *4=12~60Byte variable

End of touch: 4Byte

Byte0	Byte1	Byte2	Byte3
0x55	0	0	0xaa

CRC=0x55+Byte0+Byte1+……+ByteN-2 Range of X axis and Y axis value: 0~4095,
Definition of Touch Width: Touch Width of a touch point is the value of start point coordinates subtract end point coordinates.
In all dual-byte data, higher byte is in front of lower byte