



Hochschule für Angewandte Wissenschaften Hamburg  
*Hamburg University of Applied Sciences*

# Projekt

Christian Fischer

Validierung eines Konzeptes für eine  
Interaktionskomponente in Collaborative  
Workspaces

# Inhaltsverzeichnis

<b>1 Einführung</b>	<b>3</b>
1.1 Ziel des Projektes . . . . .	3
<b>2 Projekt</b>	<b>4</b>
2.1 Idee . . . . .	4
2.2 Hardware . . . . .	5
2.3 Entwicklungsumgebung . . . . .	5
2.4 Systemaufbau . . . . .	6
2.5 Algorithmen . . . . .	7
2.5.1 Bildverarbeitung . . . . .	7
2.5.2 Extraktion . . . . .	8
2.5.3 Mathematische Interpretation . . . . .	9
<b>3 Auswertung</b>	<b>13</b>
<b>Literaturverzeichnis</b>	<b>15</b>

# 1 Einführung

## 1.1 Ziel des Projektes

Dieses Projekt ist ein Bestandteil der Vorbetrachtung der Masterarbeit mit dem Thema „Multimodale Interaktion in Collaborative Workspaces“. Einzuordnen ist diese in das UbiComp-Projekt<sup>1</sup> der HAW Hamburg.

Ziel dieses Projektes ist die Validierung eines Konzeptes für eine Interaktionskomponente in, durch Computer unterstützten, Arbeitsumgebungen. Die vorliegende theoretische Betrachtung erfolgt im Seminar (Fischer (2007a)), basierend auf den beiden Ausarbeitungen „Anwendung I“ (Fischer (2006)) und „Anwendung II“ (Fischer (2007b)), in denen Grundkenntnisse über mögliche Technologien vermittelt werden, das Resultat ist ein grobes Konzept aus funktionaler und primär technologischer Sicht.

Der Fokus dieser Arbeit liegt auf dem technologischen Aspekt, es ist die Umsetzbarkeit der Idee zu zeigen und mögliche Chancen und Herausforderungen zu erörtern.

---

<sup>1</sup>UbiComp-Projekt der HAW Hamburg: <http://users.informatik.haw-hamburg.de/~ubicomp/>

## 2 Projekt

Bevor das eigentliche Konzept betrachtet werden kann, erfolgt, basierend auf den Vorkenntnissen, eine Auswahl der zu verwendenden Hardware. Mit Sicht auf dieser, ist der zweite Schritt die Bestimmung einer geeigneten Entwicklungsumgebung. Betrachtet werden hierbei angemessene Programmiersprachen, Frameworks und Tools. Projektübergreifend muss auf die gesamte UbiComp-Umgebung geblickt werden, um eine spätere Synergie der einzelnen Teilprojekte zu gewährleisten. Spezieller Fokus liegt auf den Arbeiten von Philipp Rossberger und Oliver Köckritz. Es folgt der konkrete Entwurf des Systemaufbaus gemäß des Seminars. Der nächste Schritt ist die Vermittlung verwendeter Algorithmen auf Grundlagenbasis. Die letzte Sektion eröffnet offene Fragestellungen in technischer Hinsicht und zeigt Chancen sowie Herausforderungen.

### 2.1 Idee

Entwickelt wird ein kamerabasiertes Eingabesystem mit der Möglichkeit der dreidimensionalen Interaktion im Raum. Eine Kamera wird hierbei als Zeigegerät eingesetzt, ausgerichtet wird auf den zu interagierenden Bereich, einem oder mehreren Displays. Für die Validierung des Konzeptes wird der Einsatz in dieser Arbeit auf die Interaktion mit einem Display beschränkt.

Eindeutig erkannt wird der Bildschirm durch eine markante Hervorhebung der Ränder. Erreicht wird dies durch die Verwendung von Infrarotdioden, die in den Ecken der angebracht sind. Mit dem Einsatz eines Infrarotsperrefilters vor dem Objektiv der Kamera ist eine effektive Extraktion dieser „Tags“ möglich.

Mit Hilfe von geometrischen Berechnungen lassen sich der Zeigepunkt auf dem Bildschirm und die Orientierung sowie Bewegung des Zeigegerätes im Raum ermitteln.

## 2.2 Hardware

Eingesetzt wird eine handelsübliche Miniaturkamera mit einer Auflösung von 380 TV Linien. Die Kommunikation erfolgt via Funk auf 2,4GHz, ein Empfänger stellt das Signal über Composite Video (FBAS) zur Verfügung. Eine Video-Grabberkarte mit einer Abtastrate von 25 Bildern pro Sekunde ermöglicht die Verwendung des Videostreams über USB auf einem Computer.

Die Tags am Bildschirmrand bilden Standard-Leds mit einer kurzwelligen Strahlung von 950 nm bei 100 mA und einer Sperrspannung von 5 V, die Filterung des IR-Lichts ermöglicht handelsübliche IR-Folie.

Die Auswertung und Berechnung erfolgt auf einem Notebook mit 1,8 GHz Centrino-Prozessor und 1 GB RAM. Als Betriebssystem wird Microsoft Windows XP SP2 eingesetzt.

## 2.3 Entwicklungsumgebung

Zu entwickeln sind zwei Softwareapplikationen: das Modul der Videoaufnahme und -auswertung sowie eine beispielhafte Anwendung, welche die ausgewerteten Daten grafisch darstellt (z.B. in Form eines Mauszeigers).

**Programmiersprache** Die Auswertung des Videostreams ist zeitkritisch, was die Verwendung von C++ im Bereich der Video-Zerlegung und -Aufarbeitung, sowie der Dateninterpretation gerechtfertigt. Die grafische Oberfläche wird in Java erstellt.

**Bibliotheken** Das Zerlegen des Videostreams in Einzelbilder erfolgt durch „IC Imaging Control“ von Imaging Source<sup>1</sup>. Es handelt sich um eine Bibliothek in C++.

Aufgrund eigener Erfahrungen und des vorhandenen Know-Hows an der HAW Hamburg wird für die Bilderverarbeitung bzw. Bildmustererkennung die „LTI-Lib“ der Technischen Hochschule Aachen<sup>2</sup> eingesetzt. Diese Bibliothek ist in C++ verfügbar und kann bequem mit IC Imaging Control verknüpft werden.

Zur Gestaltung der grafischen Benutzeroberfläche wird das Standard Widget Toolkit (SWT) von Eclipse<sup>3</sup> eingesetzt.

---

<sup>1</sup>IC Imagine Control von Imaging Source: <http://www.imagingcontrol.com/>

<sup>2</sup>LTI-Lib der Technischen Hochschule Aachen: <http://www.techinfo.rwth-aachen.de/Software/Ltilib/>

<sup>3</sup>Standard Widget Toolkit: <http://www.eclipse.org/swt/>

**Tools** Mit C++ bietet sich unter Windows als Entwicklungsumgebung Microsoft Visual Studio 2003 an (VS2005 ist nicht interessant, da die LTI-Lib für VS2003 implementiert ist).

Die Java-Komponenten werden unter Eclipse entwickelt.

## 2.4 Systemaufbau

Das Grobkonzept des System wird im Seminar erarbeitet, dargestellt in Abbildung 2.1. Die Entwicklung des Prototyps erstreckt sich auf 3 Ebenen: die Datenerfassung, die Verteilung und die Verwendung auf der funktionalen Ebene. Diese Arbeit zeigt die Plausibilität des Interaktionskonzepts, wobei der spezielle Fokus auf die Datenerfassungsebene gerichtet ist. Somit ist primär die hardwaretechnische Umsetzung und Videoanalyse von Interesse. Die grafische Oberfläche ist ein Fenster, in dem ein Mauszeiger bewegt und ein dreidimensionaler Würfel gedreht werden kann. Die Daten des Interaktionsgerätes werden auf der Verteilungsebene via UDP an den GUI-Rechner gesendet. Näher wird auf diese beiden Schichten nicht eingegangen.

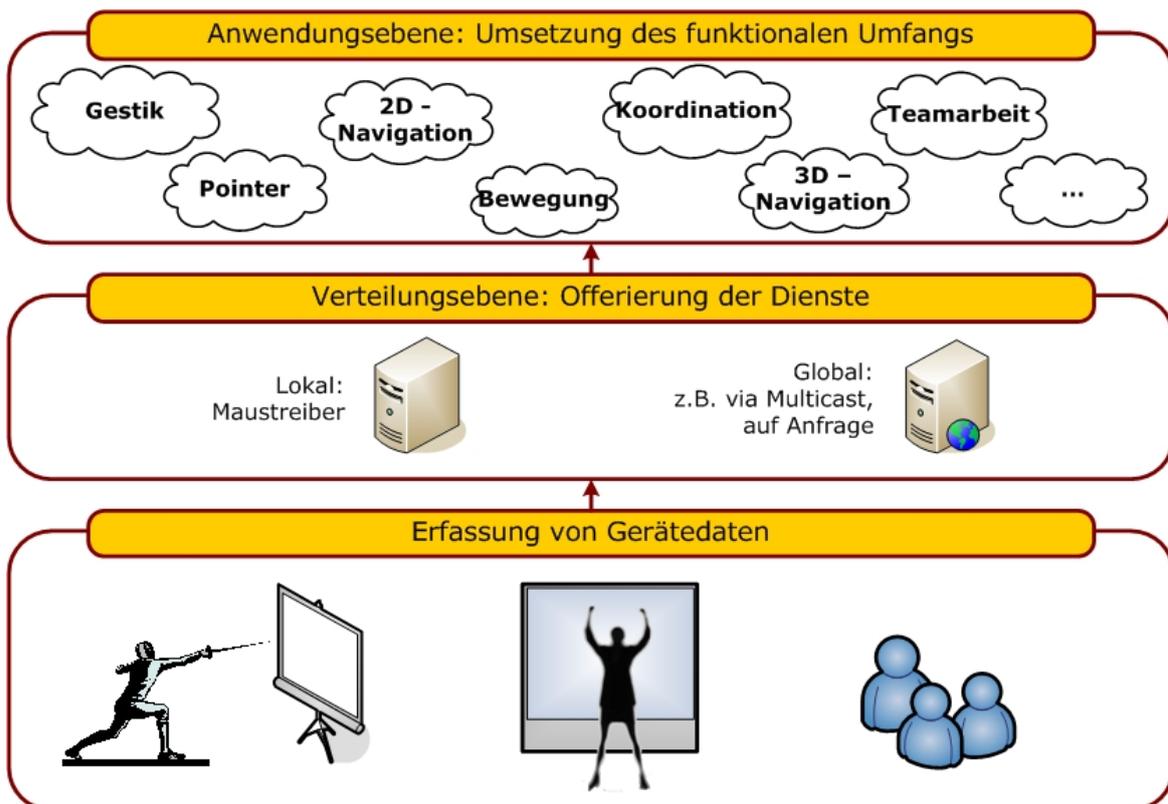


Abbildung 2.1: Schichtmodell des Systems

Die Datenerfassung ist ein Prozess von fünf Schritten:

1. Videoaufnahme: Die Kamera nimmt den gezeigten Bereich als Video auf, der Grabber stellt dem System den Stream zur Verfügung.
2. Bildverarbeitung: Das einzelne Bild wird aufbereitet, um effektiver visuelle Informationen erfassen zu können.
3. Extraktion: Die wesentlichen Informationen des Bildes werden extrahiert und als Datensatz aufbereitet.
4. Mathematische Interpretation: Aus den Informationen des Datensatzes mit den Bildinformationen lassen sich Eigenschaften bezüglich Haltung des Eingabegerätes gewinnen.

Nachfolgend werden die wesentlichen Punkte der Datenerfassung im Detail aufgezeigt, diese sind Bildverarbeitung, Extraktion und die mathematische Interpretation.

## 2.5 Algorithmen

### 2.5.1 Bildverarbeitung

Mit Hilfe der „IC Imagin Control“-Bibliothek können 25 Bilder pro Sekunde des Videostreams extrahiert werden. Da die Bilder nacheinander bearbeitet werden, hängt die eigentliche Framerate von der Verarbeitungs- bzw. Berechnungsgeschwindigkeit des Gesamtsystems ab. Nach einer Graufwertabstufung auf 8 Bit (mit LTI-Lib) zeigt Abbildung 2.2 ein einzelnes Frame, wie es direkt von der Kamera mit installiertem Infrarotsperrfilter erfasst wird.

Neben den, im Bild klar zu erkennenden, vier Infrarot-LEDs fallen weniger helle Lichtschleier und andere Lichtquellen, wie hier das Fenster auf. Um die vier Punkte markant hervorzuheben, ist die Konvertierung in einen kleineren Farbraum nützlich. Durch die Validierung gegen einen geeigneten Schwellwert bleiben die wesentlichen Informationen im Bild bestehen, der Rest wird schwarz. Nach [Sonka u. a. \(2007\)](#) eignet sich dafür der iterativ berechnete, „optimale Schwellwert“. Das Ergebnis ist in Abbildung 2.3 zu sehen.

### 2.5.2 Extraktion

Mit Extraktion ist hier nicht das bekannte Verfahren für Bildmustererkennung gemeint, sondern die Berechnung der konkreten vier einzelnen Mittelpunkte des Bildes. Angewandt wird hier die geometrische Momentenberechnung, um die einzelnen Schwerpunkte zu erhalten.



Abbildung 2.2: Einzelnes Rohbild von der Kamera

Es wird wie folgt vorgegangen: Durch eine Schwerpunktberechnung auf das gesamte Bild erhält man den „Mittelpunkt“ zwischen den vier Eckpunkten, zu sehen in [Abbildung 2.4](#).

Mit einer gedachten horizontalen und einer vertikalen Linie durch den Mittelpunkt kann die Fläche in vier Sektoren aufgeteilt werden. Die erneute Anwendung der Schwerpunktberechnung auf jeden einzelnen Quadranten ergibt die einzelnen Schwerpunkte der Eckpunkte, siehe [Abbildung 2.5](#).

Die Koordinaten der vier Eckpunkte liegen nicht im Ganzzahlenformat vor, sondern als Fließkommawert. Da sich die Eckpunkte in einem Koordinatensystem befinden, können mit ihnen mathematische/geometrische Berechnungen ausgeführt werden.

### 2.5.3 Mathematische Interpretation

Die Koordinaten der Eckpunkte bilden verbunden ein Viereck, welches das Display repräsentiert. Mathematisch/geometrisch lassen sich Informationen über die Verwendung des Eingabegerätes errechnen:

**Zeigerpositionierung** Die Kamera fungiert als Zeigegerät, folglich bildet der Bildmittelpunkt des Kamerabildes den Punkt, auf den gezeigt werden soll. Es muss stets davon ausgegangen werden, dass das durch die vier Eckpunkte aufgespannte Viereck perspektivisch verzerrt ist, da der Benutzer das Eingabegerät von beliebiger Position halten

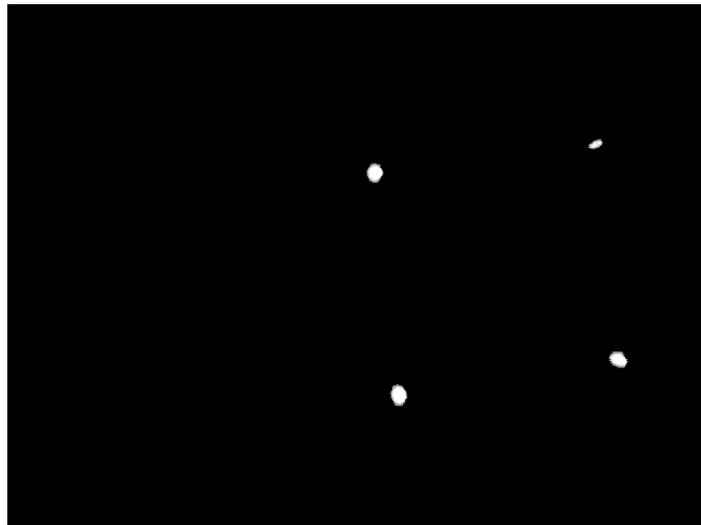


Abbildung 2.3: Einzelnes Bild nach „Optimal Threshold“

kann. Das Viereck ist in einem affinen Koordinatensystem aufgespannt, die Koordinatenachsen sind jedoch nicht orthogonal zueinander, siehe Abbildung 2.6. Punkt P markiert den Mittelpunkt des Kamerabildes.

Der erste Schritt ist eine Transformation der Bildpunkte in ein kartesisches Koordinatensystem. Das Zielobjekt ist in Abbildung 2.7 zu sehen. Die Verhältnisse der Eckpunkte zum Mittelpunkt sind in beiden Vierecken gleich, daher gilt folgende Gleichung zur Berechnung der Koordinaten des Punktes P:

$$P'.x = \frac{\overline{A'M'}}{\overline{A'E'}} * \overline{A'B'}$$

$$P'.y = \frac{\overline{B'M'}}{\overline{B'F'}} * \overline{B'C'}$$

mit

$$\overline{A'M'} = \sqrt{\overline{A'B'}^2 * \overline{B'C'}^2}$$

$$\overline{A'E'} = \frac{\overline{AE}}{\overline{AM}} * \overline{A'M'}$$

$$\overline{A'B'} = \text{Displayauflösung}.x$$

$$\overline{B'M'} = \overline{A'M'}$$

$$\overline{B'F'} = \frac{\overline{BF}}{\overline{BM}} * \overline{B'M'}$$

$$\overline{B'C'} = \text{Displayauflösung}.y$$

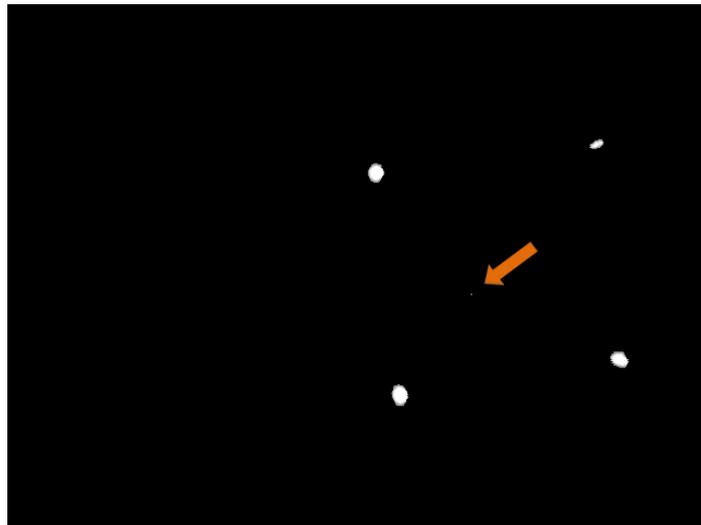


Abbildung 2.4: Einzelnes Bild mit Gesamtschwerpunkt

Die Displayauflösung wird auf 1.0 mit vier Nachkommastellen normiert, somit ist die Datenschnittstelle eindeutig definiert und nicht bezüglich einer Auflösung fixiert bzw. es muss für den einzelnen Anwendungsfall nicht umgerechnet werden.

**Orientierung** Die Orientierung des Eingabegerätes wird durch die Winkel zu der horizontalen und der vertikalen Ebene angegeben. Dies ermöglicht eine Auswertung der Orientierung im dreidimensionalen Raum. In [Abbildung 2.8](#) ist das rotierte Viereck zu sehen, stellt man sich eine Ebene vor und lässt diese um die x-Achse ( $\alpha$ ) und um die y-Achse ( $\beta$ ) rotieren, gelten aufgrund der Linearität folgende Gleichungen:

$$\alpha = 180.0 * \frac{\overline{ME}}{\overline{ME} + \overline{MG}}$$

$$\beta = 180.0 * \frac{\overline{MF}}{\overline{MF} + \overline{MH}}$$

**Lokalisierung** Die Algorithmus zur Entfernungsbestimmung des Benutzers zum Display, ist in dieser Projektarbeit noch nicht behandelt, grob skizziert sollten die Schritte jedoch wie folgt aussehen: Durch Rotation um die beiden Achsen auf die Nullstellung, bildet sich ein Rechteck. Die Seitenlängen geben Aufschluss über die Entfernung der Person bzw. des Eingabegerätes zum Display.

Anmerkungen: Eine Kalibrierung bzw. „Eichung“ des Systems im Vorfeld ist notwendig, um eine reelle Entfernung angeben zu können. Nicht trivial ist die rückwärtige Rotation auf die Nullposition, da die geometrische Berechnung von der Position des Vierecks im Kamerabild abhängig ist.

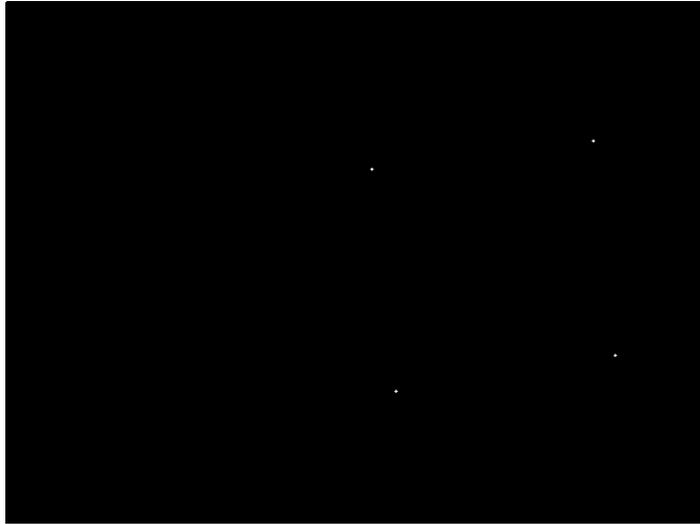


Abbildung 2.5: Einzelnes Bild mit vier Schwerpunkten

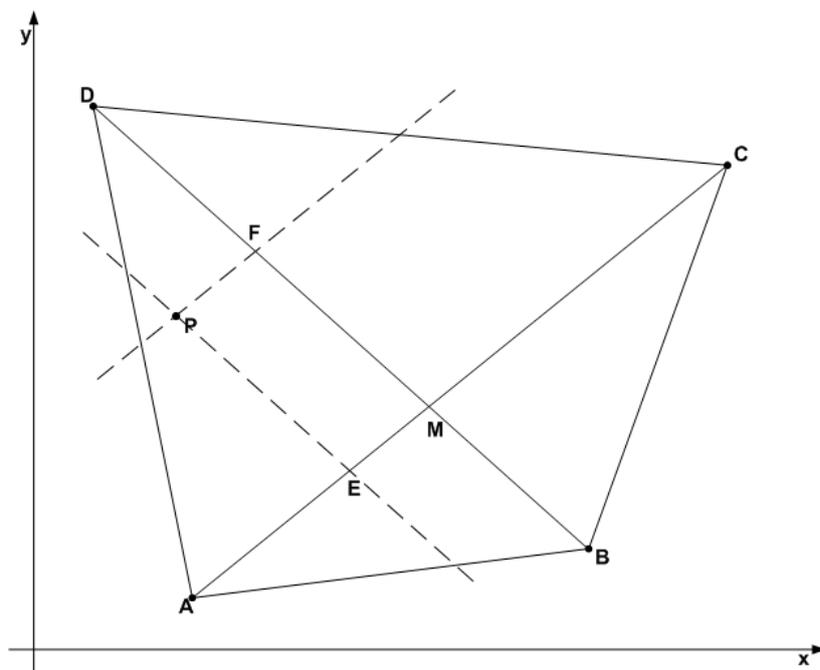


Abbildung 2.6: Perspektivisch verzerrter Bildschirm

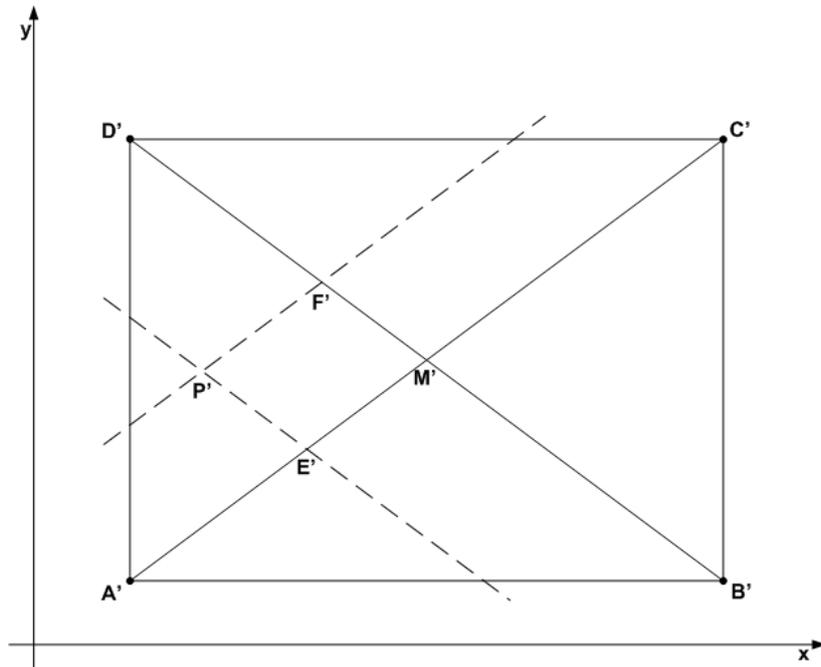


Abbildung 2.7: Entzerrter Bildschirm

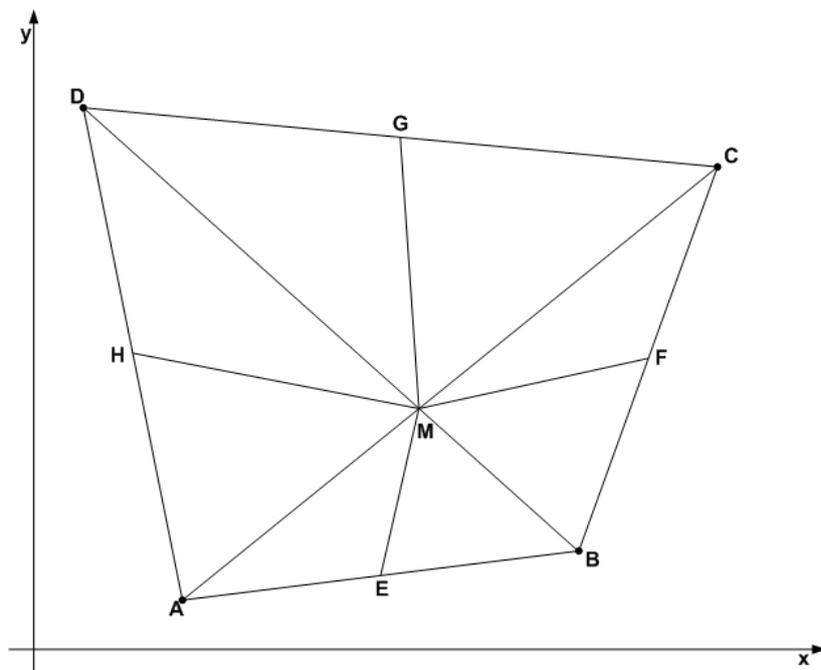


Abbildung 2.8: Rotiertes Rechteck

## 3 Auswertung

Ziel dieses Projektes ist die Validierung eines Konzeptes für eine Interaktionskomponente. Die vorliegende technologische Umsetzung zeigt, dass der theoretische Ansatz konkret zur Praxis ist und das Konzept grundsätzlich funktioniert.

Mit einer Erneuerungsrate von etwa 60 ms pro Datensatz ist die Anwendbarkeit mittelmäßig bis grenzwertig. Gesteigert werden könnte die Performanz durch die Verwendung effizienterer Bilderkennungsalgorithmen, besonders bei der Verarbeitung und Extraktion.

Eine Erhöhung der Zuverlässigkeit und Genauigkeit kann durch den Einsatz stärkerer IR-LEDs (auch mit größerem Abstrahlwinkel) und einem darauf abgestimmten Infrarotsperrfilter erfolgen. Das periodische Triggern der LEDs mit einem Vielfachen der Framerate der Kamera könnte Lichtschleier um die Dioden herum unterdrücken. Die Erweiterung um Softwaremodule zum Filtern und Regeln erhöht die Genauigkeit.

Aktuell bedingt die Verwendung das Einhalten eines relativ hohen und unpraktikablen Abstands. Zurückzuführen ist dies auf die Bedingung, permanent alle vier Eckpunkte auf dem Bild sehen zu können. Das verwendete Objektiv mit einer Brennweite von 12 mm ermöglicht nur einen Bildwinkel von  $18^\circ$ . Eine Optimierung mit 3,6 mm-Objektive erhöht den Blickwinkel auf (für die Anwendung ausreichende)  $60^\circ$ , jedoch tauchen hiermit Probleme mit dem automatischen Weißabgleich der einfachen Miniaturkameras auf. Die Dioden sind nicht mehr eindeutig vom restlichen Bild zu unterscheiden.

Der hohe Bedarf an Rechenleistung, besonders bei der Bildverarbeitung, schlägt sich kritisch auf die Mehrbenutzerfähigkeit nieder. Um mehrere Geräte in das System zu integrieren, ist sehr viel Rechenpower notwendig.

In der Ausarbeitung zu „Anwendung II“ wird der Nintendo Wiimote Controller vorgestellt, der zu Beginn dieser Projektarbeit noch nicht auf dem Markt war. Details über die eingesetzte Technologie und des eigentlichen Interaktionskonzeptes waren ebenfalls nicht verfügbar. Wie in der oben genannten Ausarbeitung beschrieben, ist mit dem Wiimote-Controller eine sehr ähnliche Idee umgesetzt. Die verwendete Hardware beseitigt kritische Punkte, die bei dieser Validierung aufgefallen sind:

Die Erneuerungsrate pro Rohbild ist sehr hoch, da eine spezielle optische Komponente eingesetzt und damit ein Großteil der Berechnung von Hardware erledigt wird. Dieser Fakt ermöglicht gleichzeitig die Mehrbenutzerfähigkeit. Durch den Einsatz von Bluetooth als Kommunikationsmittel ist die Integration von sieben Geräten mit einem Empfänger auf einem System leicht implementierbar. Die Interaktionsentfernung ist deutlich geringer, was auf die Verwendung von einem optimalen Objektiv zurückzuführen ist.

Die Wiimote verwendet als Tags zwei IR-LEDs, was die Funktionalität in Hinblick auf Interaktion verringert. Es ist nicht zu unterscheiden, ob ein Benutzer weiter weg ist, oder von der Seite schaut. Es soll jedoch die Möglichkeit geben, die Wiimote für bis zu vier LEDs zu konfigurieren, womit das Problem behoben ist.

# Literaturverzeichnis

- [Fischer 2006] FISCHER, Christian: *Multimodale Interaktionen in Collaborative Workspaces*. Juli 2006. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2006/fischer/abstract.pdf>. – Anwendung 1
- [Fischer 2007a] FISCHER, Christian: *Entwicklung einer Interaktionskomponente in Collaborative Workspaces*. 02 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07/fischer/report.pdf>. – Semnar
- [Fischer 2007b] FISCHER, Christian: *Interaktionskomponenten für Collaborative Workspaces*. 02 2007. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master06-07-aw/fischer/report.pdf>. – Anwendung 2
- [Sonka u. a. 2007] SONKA, Milan ; HLAVAC, Vaclav ; BOYLE, Roger: *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007. – 129 S. – ISBN 049508252X