

Studiengang Systemtechnik

Vertiefungsrichtung Infotronics

Diplom 2011

Fernando Kummer

*Entwicklung einer auf
Kinect basierenden
Steuerschnittstelle*

Dozent

Pierre Roduit

Experte

Flavio Tarsetti

Es handelt sich um den Originalbericht des Studenten. Er wurde nicht korrigiert und kann folglich Fehler und ungenaue Angaben enthalten.

Entwicklung einer auf Kinect basierenden Steuerschnittstelle

Bericht zur Diplomarbeit

Schuljahr 2010/2011

Projekt Nummer: it/2011/50

Verfasser:

Kummer Fernando

Betreuer:

Dr. Roduit Pierre

(HES-SO Valais // Wallis)

Tarsetti Flavio

(Idiap Research Institute)

SI	TV
X	X

<input checked="" type="checkbox"/> FSI <input type="checkbox"/> FTV	Année académique / Studienjahr 2010/2011	No TD / Nr. DA it/2011/50
Mandant / Auftraggeber <input type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input checked="" type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i> Idiap Research Institute	Etudiant / Student Fernando Kummer <hr/> Professeur / Dozent Pierre Roduit	Lieu d'exécution / Ausführungsort <input checked="" type="checkbox"/> HES—SO Valais <input type="checkbox"/> Industrie <input checked="" type="checkbox"/> Etablissement partenaire <i>Partnerinstitution</i>
Travail confidentiel / vertrauliche Arbeit <input type="checkbox"/> oui / ja ¹ <input checked="" type="checkbox"/> non / nein	Expert / Experte (données complètes) Flavio Tarsetti Idiap Research Institute	

Titre / Titel

Entwicklung einer auf Kinect basierenden Steuerschnittstelle

Description et Objectifs / Beschreibung und Ziele

Das Forschungsinstitut Idiap (<http://www.idiap.ch/>), das sich in erster Linie der Mensch-Maschine-Schnittstelle und dem Management von Multimediainformationen widmet, entwickelt auch Technologiedemonstrationssysteme, um die Öffentlichkeit auf das Institut aufmerksam zu machen.

In diesem Zusammenhang soll im Rahmen dieser Diplomarbeit ein System zur Bedienung und Steuerung einer Bilder- und Videogalerie entwickelt werden, mit dem man nur mittels Gesten die Bilder durchsehen, wiedergeben und vergrössern oder ein Video vorwärts- oder rückwärtsspulen kann usw.

Die Zielsetzungen sind:

- Definition der Funktionalitäten des Demonstrationssystems und der Steuergesten,
- Entwicklung einer Anwendung zur Erkennung dieser Gesten mittels Kinect,
- Entwicklung der Demo-Anwendung, welche die Bilder- und Videogalerien verwaltet.

Délais / Termine

 Attribution du thème / Ausgabe des Auftrags:
16.05.2011

 Exposition publique / Ausstellung Diplomarbeiten:
02.09.2011


 Remise du rapport / Abgabe des Schlussberichts:
11.07.2011 / 12.00 Uhr

 Défense orale / Mündliche Verteidigung:
dès la semaine 36 / ab Woche 36

Signature ou visa / Unterschrift oder Visum

Responsable de l'orientation

Leiter der Vertiefungsrichtung:

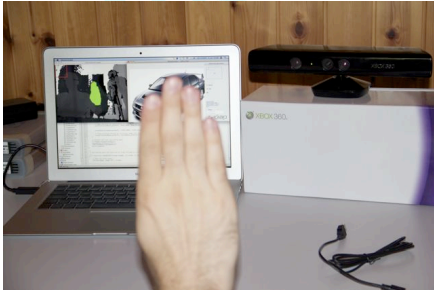

¹ Etudiant/Student:


¹ Par sa signature, l'étudiant-e s'engage à respecter strictement la directive et le caractère confidentiel du travail de diplôme qui lui est confié et des informations mises à sa disposition.

Durch seine Unterschrift verpflichtet sich der Student, die Richtlinie einzuhalten sowie die Vertraulichkeit der Diplomarbeit und der dafür zur Verfügung gestellten Informationen zu wahren.

Entwicklung einer auf Kinect basierenden Schnittstelle

Diplomand/in Fernando Kummer



Ziel des Projekts

Ziel der Diplomarbeit ist die Entwicklung einer Mensch-Maschine-Schnittstelle, welche dazu verwendet wird, eine Video- und Bildergalerie zu steuern, basierend auf dem neu entwickelten Sensorsystem Kinect von Microsoft.

Methoden | Experimente | Resultate

Diese Diplomarbeit wurde in Zusammenarbeit mit dem Forschungsinstitut Idiap in Martinach realisiert. Dieses Institut feiert sein 20-jähriges Bestehen und erweitert in diesem Rahmen seinen Showroom.

In dieser Diplomarbeit wird eine Mensch-Maschine Schnittstelle entworfen, welche die Kinect nutzt. Dieser Sensor ist eine Entwicklung von Microsoft für die Spielkonsole XBOX. Mit Hilfe dieses Sensors wird der Spieler selbst zum Steuergerät.

Diese Idee wird für die Arbeit übernommen und für die Steuerung einer Video- und Bildergalerie genutzt. Das Tiefenbild der Kinect wird genutzt um die Gesten jeder Hand einzeln analysieren zu können. Das Tiefenbild hat den Vorteil, dass das Finden der Hände von der Distanz abhängig ist und nicht mehr von einem Farbschema.

Wie dieses erfolgreich abgeschlossene Projekt eindrücklich zeigt, ist es schon heute möglich Programme ohne Maus und Tastatur zu steuern. Das erstellte Demonstrationsprogramm ist angenehm zu steuern. Es erlaubt eine Bilder- und Videogalerie zu kontrollieren.

Es bleibt jedoch klar festzuhalten, dass für die Zukunft interessante Anwendungsmöglichkeiten vorhanden sind.

Diplomarbeit | 2011 |

Studiengang
Systemtechnik

Anwendungsbereich
Infotronics

Verantwortliche/r Dozent/in
Dr Pierre Roduit
pierre.rodut@hevs.ch

Partner
Idiap Research Institute



Idiap Research Institute in
Martinach



Bild der erkannten und ausge-
werteten Hand, beziehungsweise
Geste

Inhaltsverzeichnis

1	<u>EINLEITUNG</u>	1
2	<u>PFLICHTENHEFT</u>	2
2.1	PROJEKTZIEL	2
2.2	MATERIAL	2
3	<u>HARDWARE – KINECT SENSORSYSTEM</u>	4
4	<u>WAHL DER BIBLIOTHEKEN</u>	6
4.1	OPENCV	6
4.2	OPENNI UND NITE	6
4.3	QT UND QXT	6
5	<u>DEFINITION DER FUNKTIONALITÄTEN DES PROGRAMMS</u>	7
5.1	MENÜFENSTER	7
5.2	GALERIEFENSTER (BILDER- UND VIDEOGALERIE)	7
5.3	BILDFENSTER	8
5.4	VIDEOFENSTER	8
5.5	INFORMATIONSFENSTER	9
5.6	DOCKING FENSTER	11
5.7	DEFINITION DER STEUERGESTEN	12
6	<u>PROGRAMM ZUR ERKENNUNG VON GESTEN</u>	14
6.1	FLUSSDIAGRAMM DER GESTENERKENNUNG	14
6.2	EINLESEN DER BILDDATEN	15
6.3	FUNKTIONSZONE	15
6.4	HANDSUCHE	16
6.5	HANDDISTANZ	17
6.6	HANDBILDER	17
6.7	HANDDATEN	18
6.8	GESTENERKENNUNG	20
6.9	BEWEGUNGSERKENNUNG	22
6.9.1	GESTEN MIT EINER HAND	24
6.9.2	GESTEN MIT ZWEI HÄNDEN	24
7	<u>PROGRAMM FÜR DIE BILDER- UND VIDEOGALERIE</u>	25
7.1	KLASSENDIAGRAMM	25
7.1.1	BILDANALYSE	26
7.1.2	ANZEIGE DER FENSTER	26
7.2	ZUSTANDSMASCHINE	26
7.3	PROGRAMMFENSTER	28
7.3.1	ALLGEMEIN	28
7.3.2	MENÜFENSTER	28
7.3.3	GALERIEFENSTER	28
7.3.4	BILDFENSTER	29
7.3.5	VIDEOFENSTER	29
8	<u>SCHLUSSFOLGERUNG</u>	30
8.1	RESULTAT	30
8.2	ANALYSE	30

8.3	VORSCHLÄGE ZUR WEITERENTWICKLUNG	31
8.4	GEDANKEN ZUR DIPLOMARBEIT	32
9	DATUM UND UNTERSCHRIFT	33
10	DANKSAGUNG	34
11	ANHANG	35
11.1	VERZEICHNISSE FÜR ABBILDUNGEN UND TABELLEN	35
11.1.1	ABBILDUNGSVERZEICHNIS	35
11.1.2	TABELLENVERZEICHNIS	36
11.2	QUELLENVERZEICHNIS - EIBLIOGRAFIE	36
11.3	ERKLÄRUNGEN ZUR DEFINITION DER STEUERGESTEN	38
11.3.1	MENÜFENSTER	38
11.3.2	GALERIEFENSTER (BILDER- UND VIDEOGALERIE)	38
11.3.3	BILDFENSTER	39
11.3.4	VIDEOFENSTER	41
11.4	ERKLÄRUNGEN ZUM PROGRAMM ZUR GESTENERKENNUNG	45
11.4.1	FUNKTIONSZONE	45
11.4.2	HANDZONE	46
11.4.3	HANDDATEN	46
11.4.4	GESTENERKENNUNG	49
11.4.5	BEWEGUNGSERKENNUNG	50
11.5	ERKLÄRUNGEN ZUM PROGRAMM DER BILDER- UND VIDEOGALERIE	53
11.5.1	HAUPTZUSTANDSMASCHINE	53
11.5.2	BILDERGALERIE	54
11.5.3	VIDEOGALERIE	55
11.5.4	BILDANZEIGE	56
11.5.5	VIDEOANZEIGE	57
11.6	SOFTWARE UND ELEKTRONISCHE DOKUMENTATION	58

1 Einleitung

Diese Diplomarbeit wird in Zusammenarbeit mit dem Forschungsinstitut Idiap aus Martinach realisiert. Dieses Institut hat sich vor allem auf die Schnittstelle zwischen Mensch und Maschine spezialisiert. Weitere Tätigkeitsgebiete sind die Sammlung und Auswertung von Multimediainformationen.

Um sich einem breiten Publikum präsentieren zu können, wurden schon verschiedene Demonstrationssysteme entwickelt. Da dieses Jahr das 20 Jahre Jubiläum von Idiap ansteht, soll ein neues Demonstrationssystem den bisherigen Showroom erweitern. Diese Diplomarbeit beschäftigt sich mit der Entwicklung dieses Demonstrationsgeräts.

Das Ziel der Arbeit ist es, ein System zu entwickeln, welches erlaubt, eine Video- und Bildergalerie mit Hilfe von Gesten und Handbewegungen zu steuern. Diese Bewegungen werden mit Hilfe des neuen Kinect Sensorsystems von Microsoft gefilmt. Das zu entwickelnde Programm wertet die aufgenommenen Bilder aus und sendet die entsprechenden Befehle der Gesten und Bewegungen programmintern an die Video- und Bildergalerie weiter.

Der folgende Bericht beschäftigt sich mit der Definition der Funktionalitäten des Programms für die Bilder- und Videogalerie. Das darauf folgende Kapitel beschäftigt sich mit der Definition der Gesten und Bewegungen, welche als Befehle interpretiert werden sollen. Weiter wird auch das Programm zur Erkennung von Gesten besprochen. In einem vierten Teil wird das Programm für die Video- und Bildergalerie entworfen und erklärt.

2 Pflichtenheft

Dieses Kapitel gibt einen Überblick über das genaue Pflichtenheft und zeigt, mit welchem Material gearbeitet wird.

2.1 Projektziel

Das Forschungsinstitut Idiap, das sich in erster Linie der Mensch-Maschine-Schnittstelle und dem Management von Multimediainformationen widmet, entwickelt auch Technologiedemonstrationssysteme, um die Öffentlichkeit auf das Institut aufmerksam zu machen.

In diesem Zusammenhang soll im Rahmen dieser Diplomarbeit ein System zur Bedienung und Steuerung einer Bilder- und Videogalerie entwickelt werden, mit dem man nur mittels Gesten die Bilder durchsehen, wiedergeben und vergrössern oder ein Video vorwärts- oder rückwärtsspulen kann und ähnliche Funktionalitäten.

Die folgenden Punkte verdeutlichen die Ziele dieser Diplomarbeit:

- > Definition der Funktionalitäten des Demonstrationssystems und der Steuergesten,
- > Entwicklung einer Anwendung zur Erkennung dieser Gesten mittels Kinect,
- > Entwicklung der Demo-Anwendung, welche die Bilder- und Videogalerien verwaltet.

2.2 Material

Um diese Diplomarbeit zu realisieren, werden das Kinect-Sensorsystem von Microsoft für die Spielkonsole XBOX und ein MacBook Air verwendet.

Die Spezifikationen des MacBook Air sind folgende:

- > 2.13 GHz Intel Core 2 Duo
- > 4 GB 1067 MHz DDR3 RAM
- > 256 MB NVIDIA GeForce 320 M

Um ein Programm zu schreiben wird das Entwicklungstool XCode 3.2.5 verwendet. Zusätzlich dazu werden für die Durchführung der Diplomarbeit noch die folgenden Bibliotheken genutzt:

- > OpenNI: Ist eine Library, welche benutzt wird um eigene Programme zu entwickeln, welche auf das Kinect-Sensorsystem zugreifen. Die verwendete Version ist 1.0.
- > NITE: Ist eine Middleware, welche von OpenNI gebraucht wird. Diese hat zum Ziel einige Aspekte der Personenerkennung und

Personenverfolgung zu vereinfachen und ist direkt in OpenNI eingebunden.

- > OpenCV: Ist eine Bibliothek, welche sich hauptsächlich mit Computer Vision beschäftigt und in diesem Zusammenhang zur Anzeige der Daten des Kinect-Sensorsystems verwendet wird. Diese Bibliothek vereinfacht auch die Weiterverarbeitung von Bilddaten. Es wird die Version 2.2 verwendet.
- > Qt: Diese Bibliothek wird verwendet um die verschiedenen Fenster und deren Inhalte zu definieren und anzuzeigen. Die verwendete Version ist 4.7.4.
- > Qxt: Ist eine Bibliothek, welche Qt erweitert und einige zusätzliche Funktionen für die Galerieansichten bereitstellt. Die verwendete Version ist 0.6.1.

3 Hardware – Kinect Sensorsystem

Dieses Kapitel befasst sich mit der verwendeten Hardware des Sensorsystems. Es werden kurz einige Eigenschaften des Sensorsystems gezeigt, welche während den Tests entdeckt werden konnten.

Die Hardware des Sensorsystems kann in den folgenden Abbildungen erkannt werden.

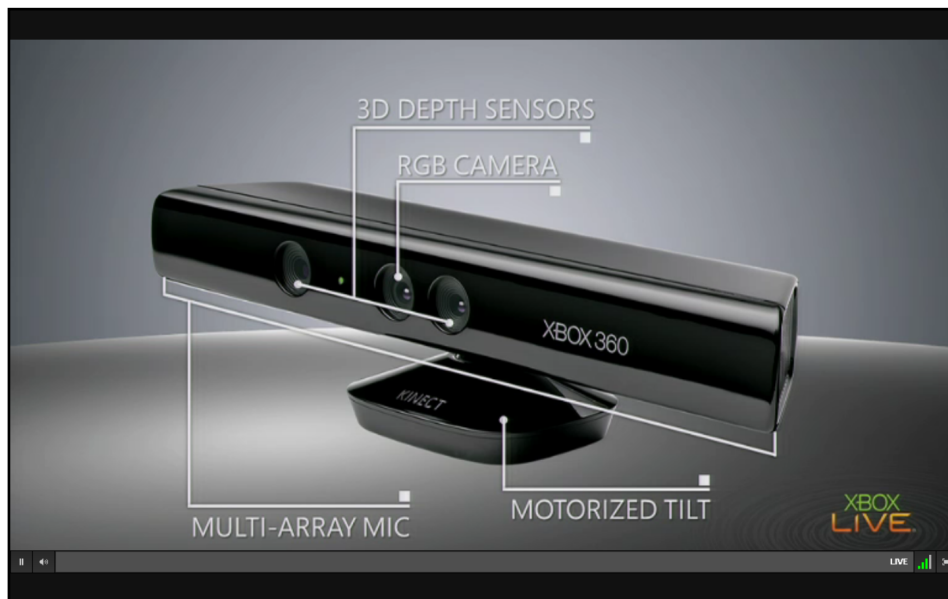


Abbildung 1: Kinect-Sensorsystem mit Beschriftung der Sensorik [18]



Abbildung 2: Kinect-Sensorsystem Innenansicht [19]

Wie man in der Abbildung 1 erkennen kann, besitzt das verwendete Sensorsystem eine RGB Kamera und ein 3D Tiefensensor. Weiter ist auch eine Multi-Array Mikrophon eingebaut. Und um dem Sensor eine grössere Möglichkeit zur Bildaufnahme zu bieten, besitzt das Sensorsystem einen Motor zur vertikalen Bewegung mit Drehpunkt im Fuss des Geräts.

Abbildung 2 zeigt ein wenig mehr von dem Aufbau des Sensorsystems. Es wird gezeigt wie das Innenleben des Sensors aussieht.

Die RGB Kamera nimmt Bilder mit einer Auflösung von 640 mal 480 Pixel mit einer Bildrate von 30 Bildern pro Sekunde auf. Der 3D Tiefensensor basiert auf der Infrarot Technik. Es werden klar definierte Lichtimpulse im Infrarotbereich vom Sender ausgeschiedt. Der Empfänger kann diese Lichtimpulse interpretieren und daraus ein Tiefenbild berechnen. Das Tiefenbild hat ebenfalls eine Auflösung von 640 mal 480 Pixel und eine Bildrate von 30 Bildern pro Sekunde. Um ein Tiefenbild zu erhalten, wird für jedes Pixel die Lichtintensität gemessen und entsprechend dem Messwert dargestellt.

Die folgende Abbildung illustriert ein Problem des Tiefenbilds und der dazugehörigen Infrarotsensoren.

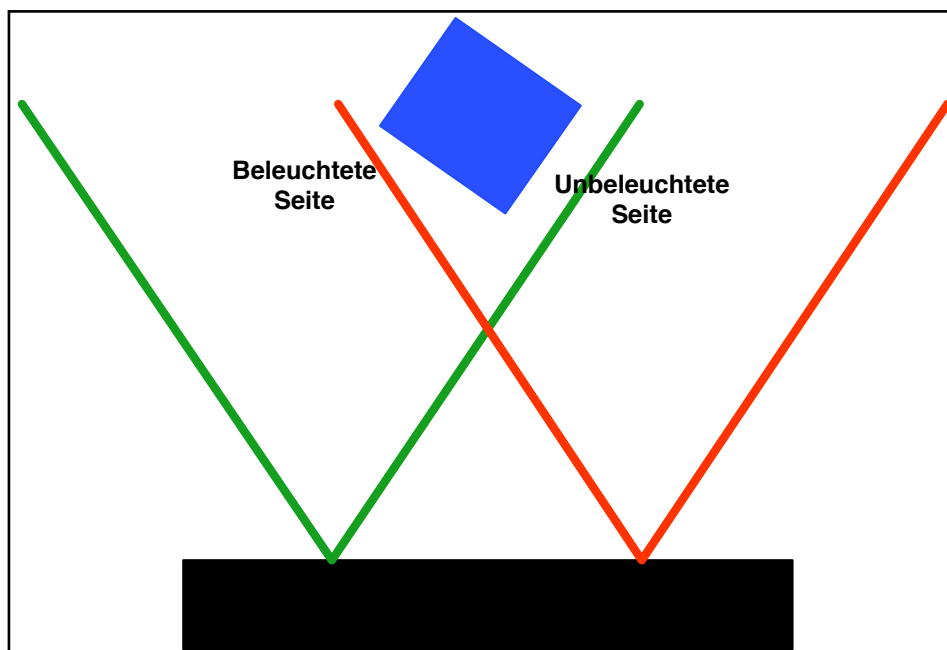


Abbildung 3: Eigenheit des Kinect-Sensorsystems

Das Problem besteht darin, dass der Infrarotsender (grün) nicht immer alle Flächen beleuchten kann, welche vom Infrarotempfänger (rot) gesehen werden. Dadurch wird die unbeleuchtete Fläche innerhalb des Tiefenbilds als schwarze Fläche dargestellt, weil die Lichtintensität dieser Fläche mit dem Standardwert 0 versehen wird. Dies ist dadurch problematisch, da je näher eine Fläche der Kamera ist, umso dunkler wird diese auf dem Bild dargestellt, denn die Lichtintensität geht näher an den Wert 0. Diese Tatsache erschwert die Analyse des Tiefenbilds.

4 Wahl der Bibliotheken

Dieses Kapitel beschäftigt sich einführend in das Projekt kurz mit der Wahl der Bibliotheken.

4.1 OpenCV

OpenCV ist eine Bibliothek, die entwickelt wurde um die Aspekte der Computer Vision zu vereinfachen. Sie stellt sehr viele Funktionen zur Verfügung um Berechnungen auf Bildern und Bilddaten sehr einfach und rasch ausführen zu können. Diese Bibliothek bietet sich deshalb an, weil sie von Intel unterstützt wird. Das bedeutet auch, dass die Berechnungen und deren Übersetzung in Code sehr stark an die Architektur von Intel Prozessoren angelehnt ist und diese optimal ausnützt. Dies ist der Hauptgrund zur Wahl dieser Bibliothek.

4.2 OpenNI und NITE

OpenNI und NITE sind Bibliotheken, beziehungsweise Middleware, um eine Verbindung zum Kinect Sensorsystem aufzubauen. Die Wahl fiel auf diese, weil OpenNI sehr einfach zu gebrauchen ist und während den Tests auch sehr stabil lief (über 1 Stunde bis zur normalen Programmbeendung), trotz der momentan instabilen Version für Mac OS X.

4.3 Qt und Qxt

Qt ist die einfachste Bibliothek um Fenster erstellen zu können. Sie bietet auch genügend Funktionalitäten um das Menü-, Bildanzeige- und Videoanzeigefenster implementieren zu können.

Die Bibliothek Qxt ist eine Erweiterung zu Qt und wird verwendet um die Bilder- und Videogalerie zu erstellen. Sie bietet zusätzlich noch einige Funktionalitäten, die Qt nicht hat und die das Programm interessanter gestalten. Als Beispiel lässt sich hier die Flowview nennen.

5 Definition der Funktionalitäten des Programms

Dieses Kapitel beschäftigt sich mit der Definition der Funktionalitäten des Demonstrationsprogramms. Das Demonstrationsprogramm setzt sich zusammen aus einer Bildergalerie und einer Videogalerie. Nachfolgend werden die entsprechenden Funktionen und Steuermöglichkeiten mit den dazugehörigen Gesten im letzten Unterkapitel definiert.

Es werden die einzelnen Programmfenster getrennt betrachtet und für jedes werden die Funktionen festgelegt.

5.1 Menüfenster

Das Menüfenster hat die Aufgabe den Benutzer vor die Wahl zu stellen, ob er die Bildergalerie oder die Videogalerie anschauen möchte. Logischerweise kann der Benutzer auch das Programm von diesem Fenster aus beenden. Dies wird über Menüpunkte ermöglicht (Auswahl zwischen Bilder- und Videogalerie) oder über eine „aktive Ecke“ (Schliessen des Programms in der Ecke oben links).

Die nachfolgende Abbildung verdeutlicht das eben gesagte.

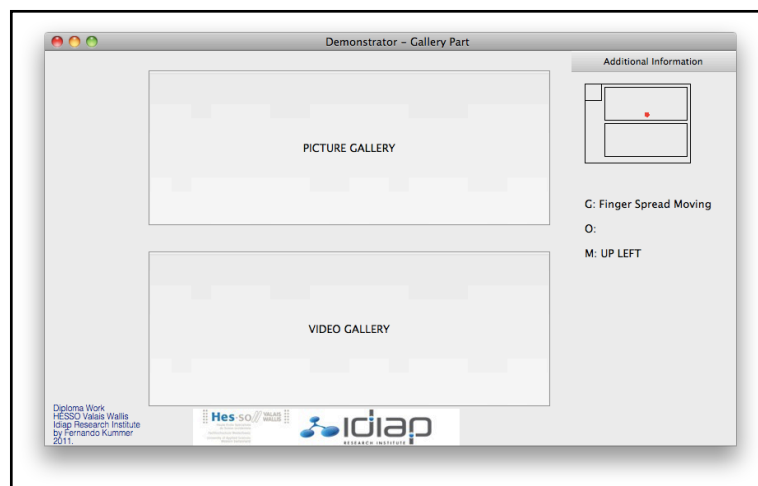


Abbildung 4: Menüfenster des Demonstrationsprogramms

5.2 Galeriefenster (Bilder- und Videogalerie)

Das Galeriefenster enthält eine Auflistung der Bilder oder Videos. Der Benutzer kann nun auswählen, welches Bild im Grossformat angezeigt oder welches Video abgespielt werden soll (Je nach Auswahl im Menüfenster). Als Programmbenutzer kann man auch durch die Galerie scrollen, um alle Bilder oder Videos in der Übersicht anschauen zu können. Weiter kann der Benutzer natürlich auch von hier aus zurück ins Menüfenster gelangen. Dies wird wiederum über die „aktive Ecke“ oben links ermöglicht.

Die folgende Abbildung zeigt das Fenster für die Video- und die Bildergalerie.

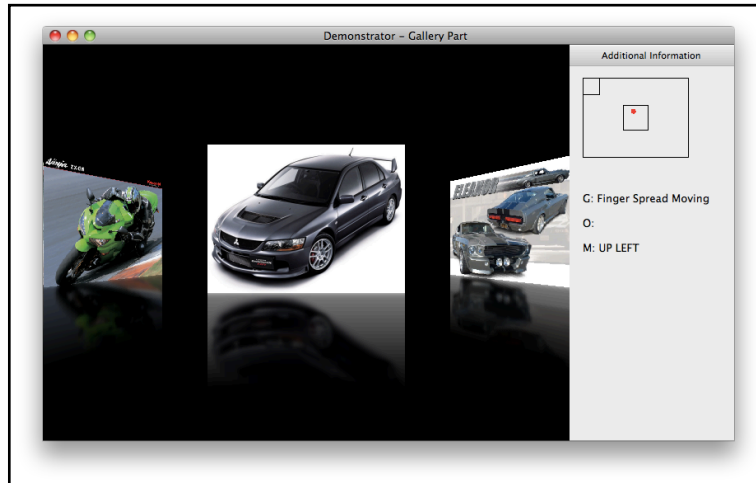


Abbildung 5: Galeriefenster für die Bilder- und Videogalerie

5.3 Bildfenster

Wenn sich der Benutzer entschieden hat, ein Bild im Grossformat anzuschauen, hat er in diesem Fenster die Möglichkeit zum nächsten oder vorherigen Bild in der Liste zu wechseln. Wenn ein Listenende erreicht werden sollte, wird zum anderen Listenende gewechselt. Es wird dem Benutzer auch die Möglichkeit gegeben das Bild zu vergrössern oder zu verkleinern, selbst das Drehen von Bildern wird unterstützt. Da das Vergrössern von Bildern unterstützt wird, kann das Bild logischerweise auch verschoben werden. Von diesem Fenster aus kann der Benutzer auch wieder zur Bildergalerie gelangen.

Die folgende Abbildung zeigt solch ein Fenster zur Bildansicht.

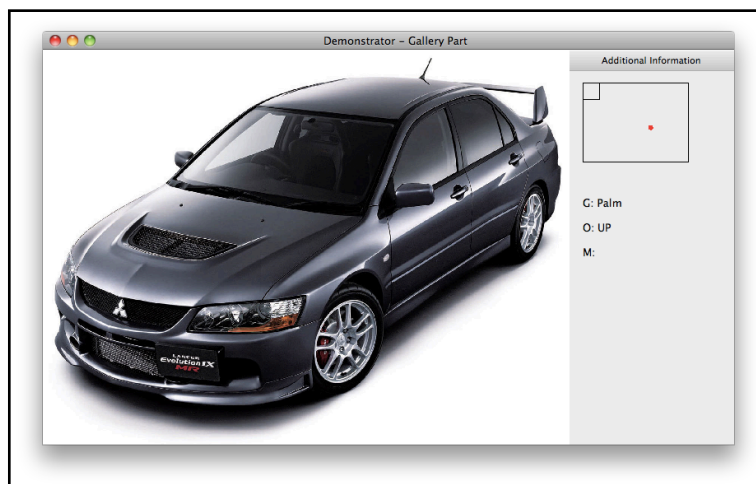


Abbildung 6: Bildfenster zur Präsentation von jedem Bild einzeln

5.4 Videofenster

Das Videofenster gibt dem Benutzer die Möglichkeit sein gewähltes Video anzuschauen. Weiter kann er auch die Lautstärke des Videos ändern. Es besteht auch die Möglichkeit zum nächsten oder vorherigen Video in der Liste zu wechseln. Sollte ein Ende der Liste erreicht werden, wird zum anderen Listenende gewechselt. Natürlicherweise kann das Video auch angehalten (Pause) und daraufhin wieder fortgesetzt werden (Play). Von hier aus kann der Benutzer wiederum zur Videogalerie wechseln.

Die nachstehende Abbildung zeigt ein Fenster für das Anzeigen eines Videos.



Abbildung 7: Videofenster zur Präsentation von Videos

5.5 Informationsfenster

Mit Hilfe dieses Fensters wird der Benutzer darüber informiert, wo sich die Hände befinden, und ob die Hände überhaupt in der Funktionszone liegen. Wenn sich die Hände in der Funktionszone aufhalten, werden sie grün gefärbt. Die Position lässt sich auch mit Hilfe dieses Fensters genügend genau bestimmen.

Die folgenden Abbildungen zeigen das Informationsfenster. Im ersten Bild befindet sich die Hand innerhalb der Funktionszone und im zweiten Bild nicht.



Abbildung 8: Informationsfenster mit Hand innerhalb der Funktionszone

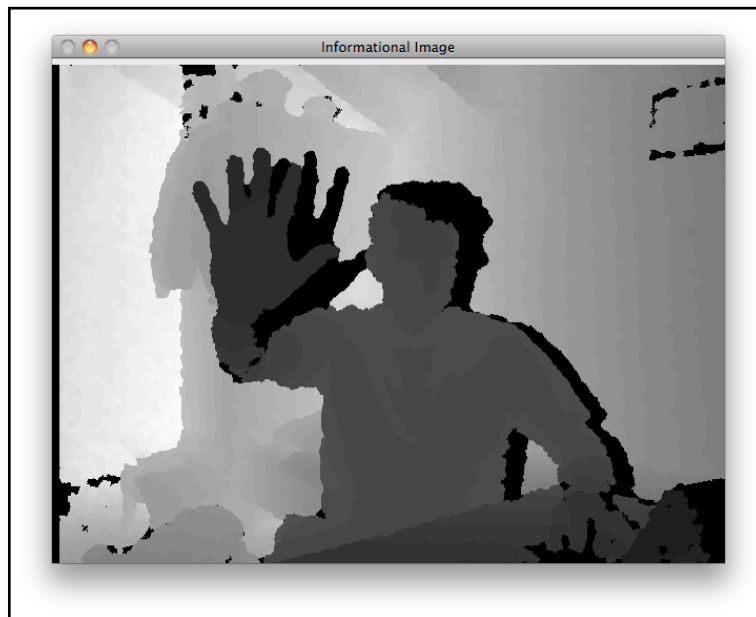


Abbildung 9: Informationsfenster mit Hand ausserhalb der Funktionszone

Zusätzlich werden an den passenden Stellen dieses Fenster noch Rechtecke zur besseren Orientierung eingezeichnet. Diese hängen vom momentanen Zustand ab. Die folgenden Abbildungen verdeutlichen dies.

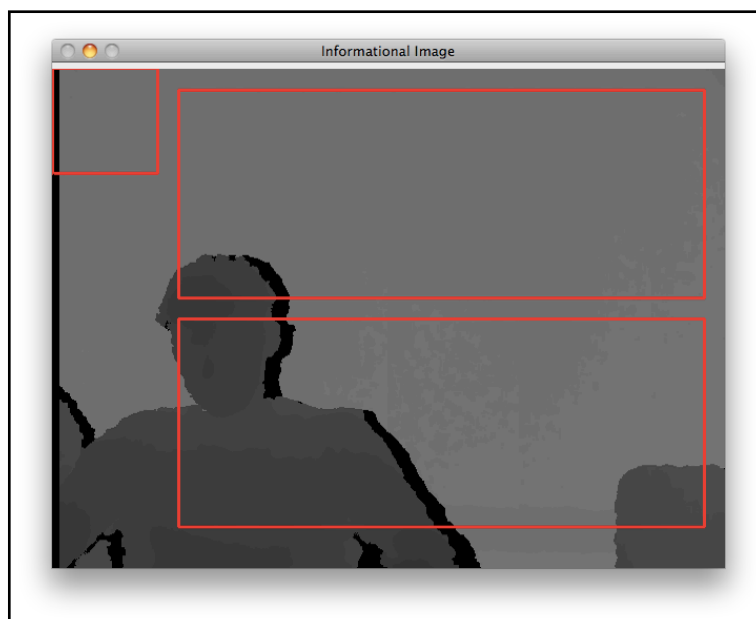


Abbildung 10: Informationsfenster mit Rechtecken für das Startmenü

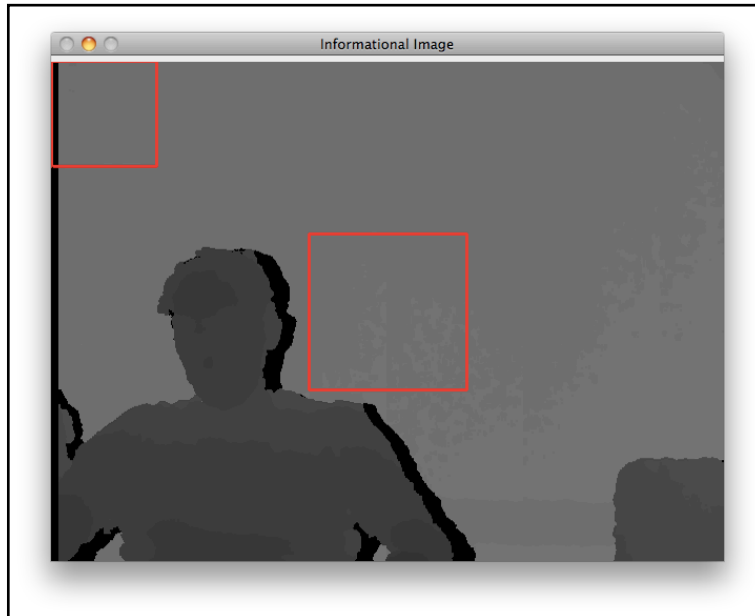


Abbildung 11: Informationsfenster mit Rechtecken für die Galeriefenster

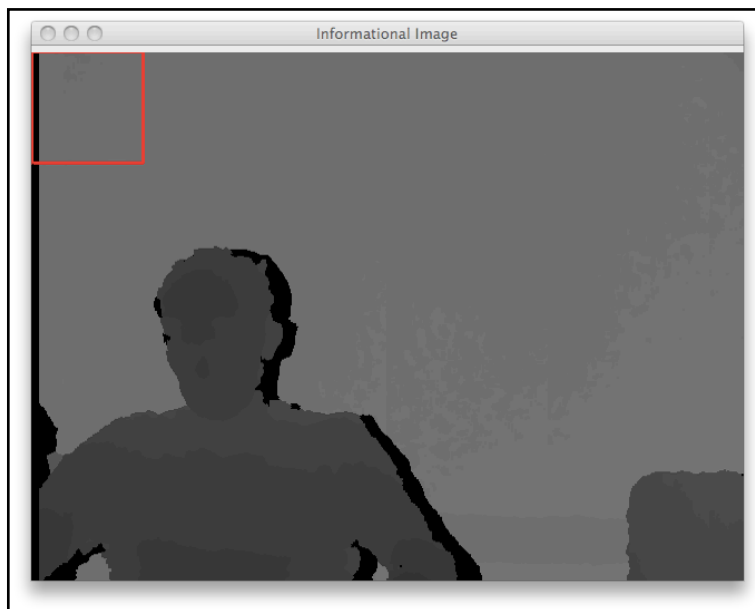


Abbildung 12: Informationsfenster mit Rechtecken für Bild- und Videoanzeige

5.6 Docking Fenster

Wie man in den Abbildungen 4 bis 7 sehen kann ist neben der Darstellung der Benutzeroberfläche noch ein Docking Fenster angebracht. Mit Hilfe dieses Fensters kann sich der Benutzer darüber informieren wo sich die Hand befindet und welche Geste gefunden wurde. Der Zweck und die Funktion dieses Fensters sind dieselben wie für das Informationsfenster aus dem vorhergehenden Kapitel.

Dadurch dass der Benutzer auch sehen kann, was für ein Analysresultat erzielt wurde, wird die Bedienung stark vereinfacht.

Die folgende Abbildung zeigt ein Docking Fenster mit einem Analysresultat.

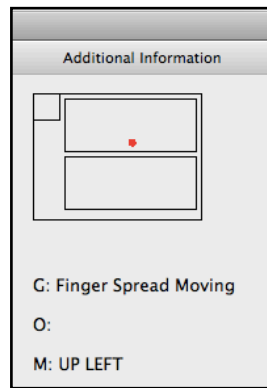


Abbildung 13: Docking Fenster mit Analyseresultat und Positionsanzeige

Das Rechteck zuoberst vom Docking Fenster zeigt an wo sich die Hand befindet (roter Punkt) und wo die aktiven Flächen sind. Darunter ist das Analyseresultat als Text enthalten:

- G steht für die Geste,
- O repräsentiert die Orientierung der Hand,
- M zeigt an, in welche Richtung sich die Hand bewegt

5.7 Definition der Steuergesten

Die Gesten zur Steuerung des Programms sollen möglichst natürlich sein. Dies ist sehr wichtig um auch die Bedienung für den Benutzer angenehm zu gestalten. Ein weiteres wichtiges Kriterium ist die Art der Gesten. Jede Geste, die genutzt wird, muss Charakteristiken aufweisen, welche die Geste von allen anderen unterscheidet. Falls dies nicht der Fall ist, ist die Geste unbrauchbar und kann nicht genutzt werden. Aus diesen Gründen werden nur vier Gesten definiert. Dies sind die folgenden Gesten:

- Faust
- Hand mit gespreizten Fingern
- Hand mit zwei gestreckten Fingern
- Hand mit zusammengelegten und gestreckten Fingern

Um eine möglichst grosse Ausschöpfung dieser Gesten zu erhalten und um die Funktionalitäten des Programms bedienen zu können, werden die zwei letztgenannten Gesten noch mit der Orientierung der Fingerspitzen versehen. Zusätzlich wird auch unterschieden ob sich die Hand bewegt oder nicht. Denn auch die Bewegungsrichtung ist ein wichtiges Kriterium für die Gestik. Weiter kann auch unterschieden werden ob nur eine Hand gebraucht wird oder zwei Hände für diese Geste genutzt werden.

Das bedeutet, dass mit den vier obigen Handformen die folgenden Gesten definiert werden können:

- Eine Hand:
 - o Faust nach links bewegt
 - o Faust nach rechts bewegt
 - o Faust nach oben bewegt
 - o Faust nach unten bewegt
 - o Faust still auf Position
 - o Hand mit gespreizten Fingern still auf Position
 - o Hand mit gespreizten Fingern nach links bewegt

- Hand mit gespreizten Fingern nach rechts bewegt
- Hand mit gespreizten Fingern nach oben bewegt
- Hand mit gespreizten Fingern nach unten bewegt
- Hand mit zusammengelegten und gestreckten Fingern still auf gleicher Position mit der Orientierung
- Hand mit zwei Fingern gestreckt still auf gleicher Position mit der Orientierung
- Zwei Hände:
 - Zwei Fäuste mit Zoomfaktor und Drehwinkel

Diese Gesten genügen um alle geforderten und definierten Funktionalitäten des Demonstrationsprogramms bedienen zu können.

Die Zuordnung der Gesten wurde grösstenteils den heute existierenden Touchscreens auf Smartphones und ähnlichen Geräten nachempfunden. Der Grund hierfür ist offensichtlich. Praktisch jedermann hatte schon einmal das Vergnügen ein solches Gerät zu bedienen und wird mit allergrösster Wahrscheinlichkeit diese Gestik oder ähnliche Gesten in diesem Demonstrationsprogramm versuchen. Aus diesem Grund werden die Gesten folgendermassen den Funktionalitäten zugeordnet:

Funktionalität	Gestik
Überall im Programm	
Auswählen	Faust still auf dem gewünschten Punkt
Im Galeriefenster	
Scrollen nach links	Faust nach links bewegt
Scrollen nach rechts	Faust nach rechts bewegt
Bildanzeige	
nächstes Bild	Hand nach rechts orientiert
vorheriges Bild	Hand nach links orientiert
Bild verschieben	Hand mit gespreizten Fingern bewegt
Zoomen und drehen	Zwei Fäuste bewegt
Videoanzeige	
nächstes Video	Hand nach rechts orientiert
vorheriges Video	Hand nach links orientiert
Lautstärke erhöhen	Faust nach oben bewegt
Lautstärke verringern	Faust nach unten bewegt
Video pausieren	Hand mit gespreizten Fingern

Tabelle 1: Zuordnung der Gesten zu den Funktionalitäten

Die Bilder zu diesen Gesten und eine Bedienungsanleitung finden sich im Anhang in Kapitel 11.3.

6 Programm zur Erkennung von Gesten

Dieses Kapitel beschäftigt sich mit den Grundideen zur Gestenerkennung, die in diesem Projekt angewandt werden.

Weitere Erklärungen sind im Anhang 11.4 zu finden.

6.1 Flussdiagramm der Gestenerkennung

Um den Ablauf der Gestenerkennung besser gliedern zu können, wird folgendes Flussdiagramm zur Veranschaulichung herangezogen.

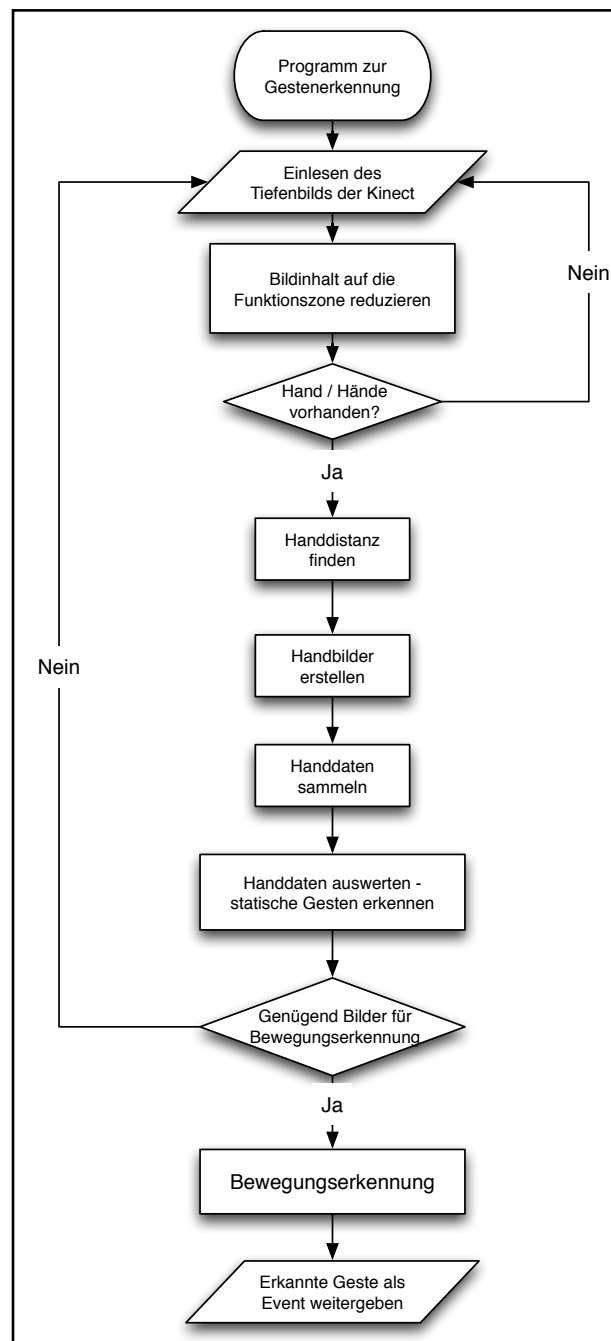


Abbildung 14: Flussdiagramm der Gestenerkennung

Das Flussdiagramm bietet eine erste Übersicht über den Ablauf der Gestenerkennung. Die einzelnen Stationen im Flussdiagramm werden in den nachfolgenden Kapiteln besprochen und erläutert.

6.2 Einlesen der Bilddaten

Für die Realisation der Gestenerkennung werden die Bilddaten von dem Kinect Sensorsystem eingelesen. Es handelt sich hierbei um Tiefenbilder. Jeder Pixel ist 16 Bit gross und stellt eine Distanz zwischen der Kamera und dem realen Punkt dar. Die eingelesenen Bilder haben eine Grösse von 640 auf 480 Pixel.

Es wird hier auf Tiefenbilder zurückgegriffen, da später die Hand nicht nach einem Farbschema, sondern anhand einer Distanz gefunden werden soll. Dies bietet auch den Vorteil, die Analyse vereinfachen zu können, denn es müssen nicht mehr verschiedenste mögliche Farbschemas abgearbeitet werden um eine oder mehrere Hände zu finden.

6.3 Funktionszone

Die Gesten werden nicht auf einem RGB-Bild gesucht, sondern auf dem Tiefenbild, welches von der Kinect aufgenommen wird. Im Tiefenbild hat jeder Pixel eine Grösse von 16 Bit. Da die Computer Vision Bibliothek OpenCV für einige Funktionen nur Bilder mit einer Pixelgrösse von 8 Bit pro Kanal unterstützt, wird in einem ersten Schritt das aufgenommene Bild auf diese Grösse reduziert. Danach wird, gestützt auf der Annahme, dass sich zwischen der Kamera und den Händen des Benutzers nichts befindet, eine Funktionszone angelegt.

Die nachfolgende Abbildung zeigt die 8 Bit Version des Originalbildes.



Abbildung 15: 8 Bit Version des Originalbilds von der Kinect

Die Funktionszone hat eine kameraferne Grenze mit dem Wert von 40. Dieser Wert entspricht ungefähr einer Distanz von 80 Zentimetern zur Kamera. Diese Distanz wurde gewählt, dass sich der Benutzer in einer angenehmen Distanz zur Kamera und dem Computerbildschirm befindet. Somit kann der Benutzer in einer angenehmen Position das Demonstrationsprogramm bedienen und sieht auch gut, wie das Programm reagiert.

Die Breite dieser Funktionszone hat den Wert 20. Dies entspricht einer Breite von ungefähr 40 Zentimetern. Die Funktionszone wurde so klein gewählt, um den Benutzer möglichst auf die Hände und einen kleinen Teil der Unterarme reduzieren zu können. Trotz der kleinen Breite kann der Nutzer seine Hände in der Zone noch in der Distanz variabel bewegen.

Der Vorteil einer Funktionszone ist, dass der Nutzer die Hände in einer genau definierten Distanz zur Kamera bewegen muss. Daher wird auch sichergestellt, dass sich nur zwei Hände auf dem Bild befinden. Durch die Funktionszone befinden sich im Idealfall nur noch die Hände und ein Teil der Unterarme auf dem Bild. Der Körper des Benutzers wird somit aus dem Bild wegreduziert. Das Resultat dieses Vorgangs wird in ein binäres Bild umgewandelt.

Die folgende Abbildung zeigt ein Resultat dieses Algorithmus.

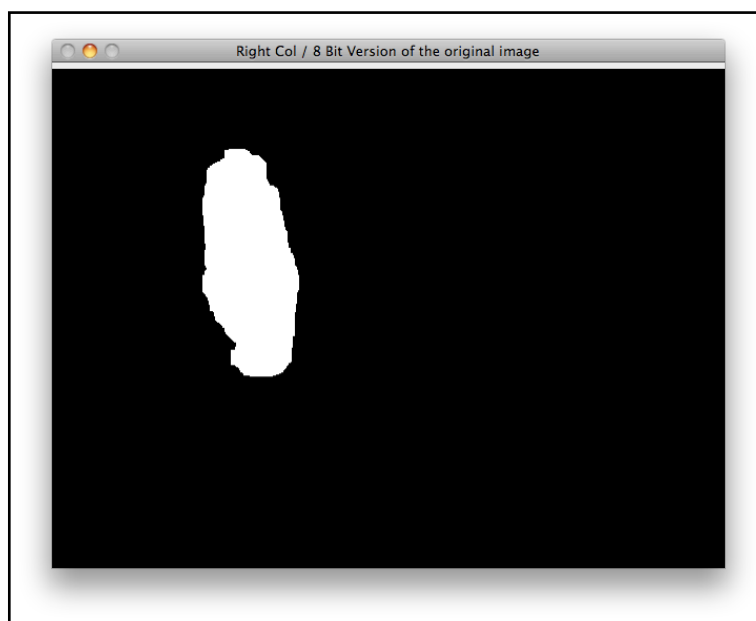


Abbildung 16: Hand in der Funktionszone mit kleinem Teil des Unterarms

6.4 Handsuche

Sobald ein Bild die Reduktion auf die Funktionszone durchlaufen hat, wird nach Händen gesucht.

Auf dem resultierenden Bild des vorherigen Kapitels wird nun nach weissen Flächen, so genannten Blobs, gesucht. Diese Blobs werden mit Hilfe von Sequenzen von Punkten beschrieben, welche den Umriss darstellen.

Mit Hilfe dieser Umrisssequenzen kann die Fläche des Blobs bestimmt werden. Dies ist wichtig, da man mit dieser Charakteristik schon Bildstörungen (kleine Fläche) von Händen (grosse Flächen) unterscheiden kann.

Basierend auf der Annahme, dass sich innerhalb der Funktionszone nur zwei Hände befinden, werden die zwei ersten Blobs mit einer genügend grossen Fläche als Hände betrachtet. Die Mindestgrösse liegt hier bei 7000 Pixel.

Dieser Wert folgt aus Tests, in welchem die Grösse von verschiedenen Handformen ausgegeben wurde. Eine Faust eines erwachsenen Mannes hat eine Fläche von ungefähr

9000 Pixel. Die Tests wurden mit der Faust durchgeführt, da diese Handform die kleinste ist unter den hier verwendeten. Die Grenze wurde etwas tiefer angelegt, da das Programm auch für Jugendliche bedienbar sein soll.

Auch die Annahme, dass sich nur zwei Hände innerhalb der Funktionszone befinden, ist dadurch begründet, dass sich die Funktionszone so nah an der Kamera befindet, dass sich kein zweiter Benutzer daneben stellen kann, um das Programm richtig bedienen zu können. Ein zweiter Benutzer würde den ersten Benutzer nur stören.

Wenn nun ein Blob eine genügend grosse Fläche besitzt, wird die Umrissequenz dieses Blobs gespeichert. Falls sich noch ein zweiter Blob findet, wird auch dessen Umrissequenz gespeichert.

6.5 Handdistanz

Sobald eine Hand oder sogar zwei Hände gefunden wurden, wird für jede Hand der Punkt gesucht, welcher die kürzeste Distanz zur Kamera aufweist. Dies geschieht mit Hilfe der Umrissequenz, welche gespeichert wurde.

Von jedem Punkt in dieser Sequenz wird die Distanz aus dem Tiefenbild gelesen und die kürzeste Distanz wird gespeichert. Dies wird für jede der gefundenen Hände durchgeführt.

Diese ist ein sehr einfacher Algorithmus um die Handdistanz zu finden. Es wird darauf verzichtet ein Histogramm mit allen Distanzen und der dazugehörigen Verteilung anzufertigen. Der Grund für die Wahl dieses einfachen Algorithmus liegt darin, dass sich die Analyse nicht mit dem Finden der Handdistanz aufhalten lassen soll, sondern schnellst möglich zur Erkennung der Gesten übergehen soll.

6.6 Handbilder

Da sich auf der Umrissequenz jedoch nicht nur die Hand, sondern auch ein Teil des Unterarms befindet, wird in diesem Schritt für jede Hand ein Binärbild erstellt, auf dem nur noch diese eine Hand zu finden ist. Dieser Schritt vereinfacht die nachfolgende Erhebung der Handdaten.

Um ein Handbild zu erstellen wird in einem ersten Schritt ein Bild für die Handzone erstellt. Diese Zone ist ähnlich zur Funktionszone, nur ist die Breite für diese Zone kleiner und die Grenze ist dynamisch. Die Grenze hängt nämlich von der Distanz ab, die im vorherigen Kapitel gefunden wurde. Die kameraferne Grenze der Handzone erhält den Wert der minimalen Handdistanz minus 4. Der Wert 4 entspricht ungefähr 8 Zentimeter und erlaubt auch die Hand etwas zu neigen und nicht mit 90° zum Unterarm halten zu müssen. Damit die Bildanalyse keine Probleme aufweist wird die Breite dieser Zone mit dem Wert 6 gewählt. Dies bedeutet, dass die kameranahe Grenze also zirka 4 Zentimeter vor der Minimaldistanz liegt. Grund dafür waren einige Probleme, wenn die kameranahe Grenze gleich der Minimaldistanz ist. Es wurden Finger abgeschnitten oder im Finger und oder der Handfläche entstanden schwarze Flächen und verunmöglichten eine weitergehende Analyse der Geste.

Die Handzone eliminiert schon einige unnütze weisse Flächen ausserhalb der gewünschten Handfläche und reduziert den Unterarm, welcher sich noch im Funktionszonenbild befindet. Da jedoch beide Hände annähernd dieselbe Distanz zur Kamera besitzen können, wird in einem nächsten Schritt eine Binärbildmaske erstellt,

welche den Inhalt der Umrisssequenz aus der Funktionszone der Hand schwarz hat und alles andere ausserhalb ist weiss. Mit Hilfe dieser Maske wird das Bild von der Handzone maskiert und als Resultat erhält man ein Binärbild, welches nur noch die gewünschte Hand enthält.

Auf diesem binären Handbild wird noch ein Open ausgeführt. Das bedeutet, dass zuerst die schwarzen Flächen vergrössert werden und nachher die weisse Fläche (Handfläche) vergrössert wird. Dieser Vorgang ist wichtig, um unsaubere Stellen (schwarze Punkte, kleine schwarze Flächen) innerhalb der Handfläche zu eliminieren. Hierbei ist wichtig, dass die weisse Fläche mindestens einmal mehr erweitert wird als die schwarze Fläche vorher. Im Programm zur Gestenerkennung wird die schwarze Fläche dreimal erweitert. Eine Erweiterung um mehr Pixel würde die Geste mit gespreizten Fingern nicht ermöglichen, denn die Finger würden durch das Erweitern der schwarzen Fläche gelöscht. Daraufhin erfolgt ein fünfmaliges Erweitern der weissen Fläche. Dadurch wird die Hand zwar breiter, aber die Bildstörungen innerhalb der Handfläche werden verschwinden.

Für beide Hände wird ein solches Binärbild erstellt. Mit Hilfe dieses Bildes lassen sich auf einfache Weise für jede Hand die nötigen Charakteristiken berechnen um die Handform und später die Geste erkennen zu können.

Die folgende Abbildung zeigt ein solches binäres Handbild.

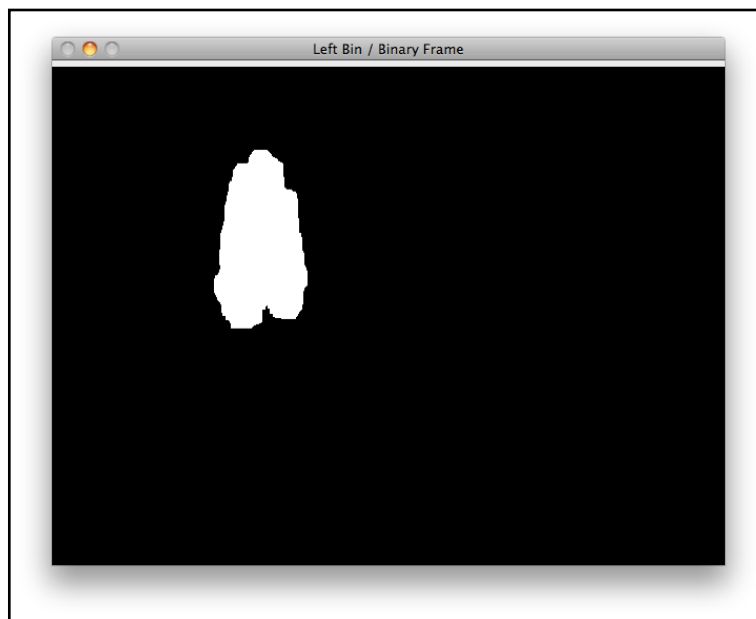


Abbildung 17: Binäres Handbild

6.7 Handdaten

Sobald nun für jede Hand ein Binärbild vorhanden ist, auf dem nur noch diese eine Hand enthalten ist, können verschiedenste Berechnungen durchgeführt werden, um später die Gesten identifizieren zu können. Die Berechnungen und deren Erklärungen finden sich im Anhang unter 11.4.3.

Mit Hilfe der Bibliothek OpenCV werden diese Berechnungen durchgeführt. Für diese Kalkulierungen werden die Handbilder aus dem vorhergehenden Kapitel herangezogen. Um diese Berechnungen durchführen zu können, wird für jedes Handbild erneut eine

Umrisssequenz berechnet. Die Umrisssequenz des vorhergehenden Kapitels, welche mit dem binären Bild aus der Funktionszone erhalten wurde, hat sich ja verändert, da nur noch die Hand ohne Unterarmteile auf dem Handbild ist.

Sobald die neue Umrisssequenz berechnet wurde, werden folgende Daten erhoben:

- Handfläche,
- Handumfang,
- Kompaktheit,
- Schwerpunkt,
- Handumgebendes Rechteck,
- Rechteckmittelpunkt,
- Distanz zwischen Rechteckmittelpunkt und Schwerpunkt,
- Exzentrizität,
- Neigungswinkel

Diese Daten werden für jedes Handbild erhoben und in einem der Hand entsprechenden Objekt abgespeichert, damit in einem nächsten Schritt die Gesten erkannt werden können.

Ausser der Handfläche, dem Handumfang, dem handumgebenden Rechteck und dem Neigungswinkel haben alle berechneten Werte relativen Charakter. Diese Tatsache erhöht die Wiederverwendbarkeit des Programms. Das heisst, dass nicht nur die Charakteristiken einer Hand (vom Programmierer) gebraucht werden kann, um das Programm zu steuern, sondern jede Hand, die die Mindestwerte der absoluten Kriterien erfüllt, kann gebraucht werden um das Programm zu steuern.

Die folgende Abbildung zeigt ein Handbild mit einigen Resultaten dieser Berechnungen.

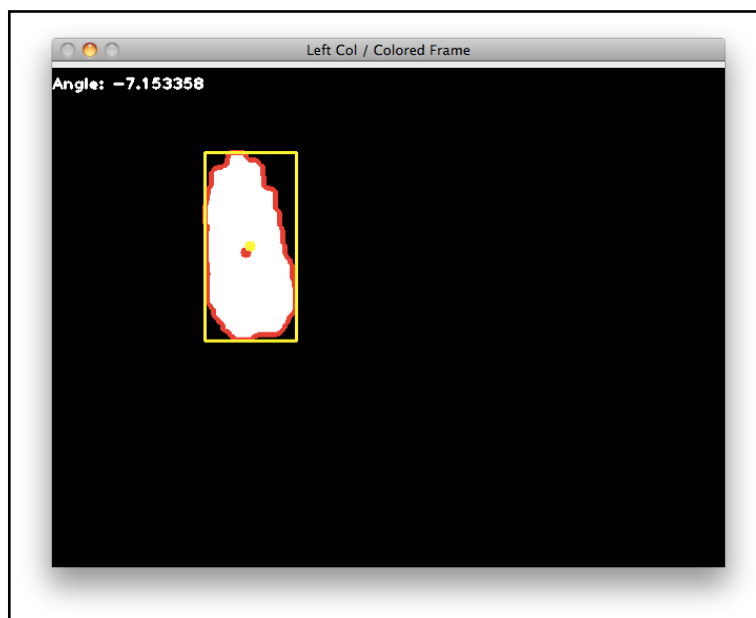


Abbildung 18: Handbild mit Auswertung

In Abbildung 14 sieht man das handumgebende Rechteck mit dessen Mittelpunkt in gelb. Der Handumriss und der Schwerpunkt sind rot gezeichnet.

6.8 Gestenerkennung

Für die Gestenerkennung werden die zuvor erhobenen Daten genutzt um die in Kapitel 5.7 definierten Gesten zu erkennen.

Im Prinzip geht es darum vier verschiedene Gesten festzustellen. Die zu erkennenden Gesten sind die folgenden:

- Faust,
- Hand mit gespreizten Fingern,
- Hand mit zwei gestreckten Fingern,
- Hand mit zusammengelegten und ausgestreckten Fingern

Es werden drei Charakteristiken betrachtet um die Gesten zu unterscheiden.

Um die „Hand mit den gespreizten Fingern“ zu erkennen, wird die Kompaktheit verwendet. Diese Eigenschaft gibt das Verhältnis zwischen Fläche und Umfang der Handform an. Diese Geste hat eine sehr geringe Kompaktheit. Diese Tatsache unterscheidet die „Hand mit gespreizten Fingern“ sehr deutlich von allen anderen Gesten.

Bei diversen Tests um den Grenzwert zu finden, ergaben sich folgende Werte für die Kompaktheit:

	Faust	Hand mit gespreizten Fingern	Hand mit zwei gestreckten Fingern	Hand mit zusammengelegten und gestreckten Fingern
Kompaktheit	0.58 – 0.72	0.18 – 0.21	0.48 – 0.54	0.48 - 0.61

Tabelle 2: Testwerte Kompaktheit

Aus den obigen Werten bietet sich ein oberer Grenzwert von 0.25 für die Handform „Hand mit gespreizten Fingern“ an. Das bedeutet dass alle Gesten, mit einem Wert tiefer als 0.25, werden als Handform „Hand mit gespreizten Fingern“ erkannt.

Als nächstes wird die Exzentrizität verglichen. Diese Charakteristik gibt an, ob eine Form auf dem Bild eine Richtung angeben kann oder nicht. Die Faust hat eine tiefe Exzentrizität, da aus der Form der „Faust“ nicht eindeutig herausgelesen werden kann, in welche Richtung sie orientiert ist. Die noch übrigen Gesten (Hand mit 2 gestreckten Fingern und Hand mit zusammengelegten und ausgestreckten Fingern) haben eine Exzentrizität, welche grösser ist, als die der „Faust“. Damit ist die Geste „Faust“ sehr gut bestimmbar.

Um sicher zu gehen, dass es sich wirklich um die „Faust“ handelt wird zur Kontrolle für diese Geste noch die Kompaktheit eingefügt. Der Grund liegt darin, dass die Geste „Hand mit zwei gestreckten Fingern“ je nach Richtung, in welche die Finger zeigen, dass die Exzentrizität dieser Geste auf fast die gleichen Werte abfallen kann, wie sie für die „Faust“ charakteristisch sind. Um diese Fälle ebenfalls zu behandeln, wird die Kompaktheit kontrolliert. Denn die „Faust“ hat eine höhere Kompaktheit als die „Hand mit zwei gestreckten Fingern“.

Um die Grenze für die Exzentrizität zur Erkennung der Handform „Faust“ herauszufinden, wurden diverse Testläufe durchgeführt und folgende Resultate erhalten.

	Faust	Hand mit zwei gestreckten Fingern	Hand mit zusammengelegten und gestreckten Fingern
Exzentrizität	0.02 – 0.09	0.12 – 0.45	0.28 – 0.56

Tabelle 3: Testwerte Exzentrizität

Bei der Auswertung der Resultate ergab sich ein oberer Grenzwert von 0.1 für die Exzentrizität zur Erkennung der „Faust“. Wie erwähnt treten in sehr seltenen Fällen Probleme auf mit der Unterscheidung von den Handformen „Faust“ und „zwei gestreckte Finger“. Aus diesem Grund wird noch ein unterer Grenzwert von 0.55 für die Kompaktheit festgelegt. Dieser Wert wird aus Tabelle 2 gelesen. Das heisst nun, dass eine Handform mit einer maximalen Exzentrizität von 0.1 und einer minimalen Kompaktheit von 0.55 als „Faust“ identifiziert wird.

Um nun die „Hand mit zusammengelegten und ausgestreckten Fingern“ von der „Hand mit zwei gestreckten Fingern“ zu unterscheiden, wird das berechnete Rechteck mit zum Anzeigefenster parallelen Seiten herangezogen. Der Abstand zwischen dem Mittelpunkt des Rechtecks und dem Schwerpunkt der Hand erlaubt es, diese beiden Gesten zu unterscheiden. Die „Hand mit zusammengelegten und ausgestreckten Fingern“ hat einen kleineren Abstand zwischen diesen Punkten als die „Hand mit zwei gestreckten Fingern“. Diese Tatsache ist in der folgenden Tabelle der Testresultate ersichtlich.

	Hand mit zwei gestreckten Fingern	Hand mit zusammengelegten und gestreckten Fingern
Distanz	15.2 – 21.1	4.0 – 10.8

Tabelle 4: Testwerte Distanz Schwerpunkt zu Rechteckmittelpunkt

Der nun angenommene Grenzwert liegt bei 12.4. Das bedeutet, dass alle Handformen mit einem kleineren Wert als 12.4 für diese Charakteristik die Handform „Hand mit zusammengelegten und gestreckten Fingern“ darstellt. Der Grenzwert wurde so bestimmt, dass für die Handform „Hand mit zusammengelegten und gestreckten Fingern“ eine genügend grosse Marge entsteht, dass Fehldeutungen vermieden werden können.

Diese Tests zur Gestenerkennung werden in genau der Reihenfolge durchgeführt, wie sie hier beschrieben wurden. Zuerst wird getestet ob die Handform „Hand mit gespreizten Fingern“ zutrifft. Als zweites wird getestet ob die Handform „Faust“ möglich ist. In einem dritten Schritt wird getestet, welche der zwei übrigen Handformen am ehesten zutrifft. Es wird angenommen dass die Hand auf jeden Fall eine der vier Handformen ausführt. Der Grund für diese Annahme ist, dass sonst keine Gesten unterstützt werden. Falls keine der unterstützten Handformen ausgeführt wird, wird die ähnlichste der unterstützten Handformen als Resultat zurückgegeben.

Nun wird noch die Orientierung für die Gesten „Hand mit zusammengelegten und ausgestreckten Fingern“ und „Hand mit zwei gespreizten Fingern“ bestimmt. Für die beiden anderen Gesten macht dies keinen Sinn, da die Art der Bestimmung der Orientierung, welche hier verwendet wird, für diese beiden Gesten nicht stabil genug ist. Ein weiterer Punkt, welcher gegen die Orientierungsbestimmung dieser Gesten spricht, sind die unnatürlichen Bewegungen, welche dafür auch ausgeführt werden müssten.

Um die Orientierung zu bestimmen werden drei Charakteristiken verwendet. Eine erste Charakteristik ist der Absolutwert des Winkels der Momente. Falls der Winkel zwischen 0° und 20° gross ist, ist die Hand nach oben oder unten orientiert. Das heisst die Fingerspitzen zeigen nach oben oder unten. Wenn der Mittelpunkt des Rechtecks (weiter oben beschrieben) unterhalb des Handschwerpunkts liegt, zeigen die Fingerspitzen nach unten und die Hand ist somit nach unten orientiert. Falls der Mittelpunkt oberhalb des Handschwerpunkts liegt, zeigen die Fingerspitzen nach oben und die Hand ist somit nach oben orientiert.

Um die Orientierung nach rechts oder links bestimmen zu können, muss der Winkel des Moments zwischen 70° und 90° liegen. Um die Orientierungen rechts und links unterscheiden zu können, werden wiederum der Mittelpunkt des Rechtecks und der Handschwerpunkt hinzugezogen. Wenn der Rechteckmittelpunkt weiter links liegt als der Schwerpunkt, zeigen die Fingerspitzen nach links und die Hand ist somit nach links orientiert. Im anderen Fall, wenn der Mittelpunkt des Rechtecks weiter rechts liegt als der Schwerpunkt, zeigen die Fingerspitzen nach rechts und die Hand ist nach rechts orientiert.

Die Werte des Momentwinkels werden so gewählt, dass die Hand nicht genau vertikal (90°) oder horizontal (0°) orientiert sein muss. Es wird eine Marge von 20° eingebaut, so dass zwar die Hand nicht genau vertikal oder horizontal sein muss, aber doch nicht so, dass für jeden Winkel eine Orientierung bestimmt wird. Das bedeutet, dass nur für klare Winkellagen in horizontaler oder vertikaler Richtung eine Orientierung berechnet wird.

6.9 Bewegungserkennung

Die Bewegungserkennung wird nicht auf jedem aufgenommenen Bild direkt durchgeführt, zuerst wird eine gewisse Zahl an Bilder mit den vorher besprochenen Algorithmen analysiert und die Resultate in einer Liste abgelegt. Experimentell wurde immer abgewartet bis sich 15 Resultate in der Liste befinden. 15 Bilder entsprechen einer halben Sekunde, denn das Kinect Sensorsystem liefert Bilder mit einer Bildrate von 30 Bildern pro Sekunde.

Diese Art der Analyse erlaubt auch, die Bewegungen zu filtern und gegen kleinere Störungen zu stabilisieren.

Sobald die Resultatliste genügend Analysen enthält, wird eine Bewegungserkennung durchgeführt.

Da jedes Resultat in der Liste analysiert wird, kann man mit statistischen Ansätzen Bewegungen erkennen und analysieren.

Es werden die folgenden Zahlen erhoben, um mit statistischen Ansätzen die Bewegungen erkennen zu können:

- Anzahl von „Faust“,
- Anzahl von „Hand mit gespreizten Fingern“,

- Anzahl von „Hand mit zusammengelegten und gestreckten Fingern“,
- Anzahl von „Hand mit zwei Fingern gestreckt“,
- Anzahl von Gesten mit nur einer Hand,
- Anzahl von Gesten mit zwei Händen,
- Anzahl von „Zwei Fäuste“,
- Anzahl von Bildern ohne Hände

Diese Zahlen genügen um zu wissen, welche Handform am meisten vorkommt. Um eine gute Analyse zu erhalten, wird verlangt dass auf mindestens zwei Dritteln der Bilder eine Hand oder zwei Hände gefunden wurde. Falls dies nicht der Fall ist, wird nicht weiter analysiert.

Die Zahl von mindestens zwei Dritteln an Analyseresultate mit einer Hand oder zwei Händen ergibt sich experimentell. Diese Zahl wird so hoch angesetzt, da dadurch die Resultate der Bewegungserkennung stabilisiert werden.

In diesem Schritt der Bildanalyse werden nun auch die endgültigen Gesten erkannt, welche an die Bilder- und Videogalerie als Befehle übermittelt werden. Es werden die folgenden Gesten erkannt und übermittelt:

- Eine Hand:
 - Faust nach links bewegt,
 - Faust nach rechts bewegt,
 - Faust nach oben bewegt,
 - Faust nach unten bewegt,
 - Faust still auf Position,
 - Hand mit gespreizten Fingern still auf Position,
 - Hand mit gespreizten Fingern nach links bewegt,
 - Hand mit gespreizten Fingern nach rechts bewegt,
 - Hand mit gespreizten Fingern nach oben bewegt,
 - Hand mit gespreizten Fingern nach unten bewegt,
 - Hand mit zusammengelegten und gestreckten Fingern still mit der Orientierung,
 - Hand mit zwei Fingern gestreckt still mit der Orientierung
- Zwei Hände:
 - Zwei Fäuste mit Zoomfaktor und Drehwinkel

Es wird die Geste oder Bewegung als Resultat zurückgeliefert, welche am meisten vorkommt, wenn genügend Analyseresultate mit einer oder zwei Händen vorhanden sind.

Eine weitere Spezialität in der Bewegungserkennung ist die Handhabung der Liste der Analyseresultate. Es werden am Ende der Bewegungserkennung nicht alle Resultate aus der Liste gelöscht, sondern ein Drittel (in dem Fall hier 5) werden in der Liste belassen für die nächste Bewegungserkennung. Es wird das zuletzt aufgenommene Drittel in der Liste gelassen. Somit werden auch die Übergänge zwischen den Listen in die Bewegungserkennung miteinbezogen.

Die folgenden Unterkapitel erklären kurz, wie Gesten erkannt werden.

6.9.1 Gesten mit einer Hand

Um eine bewegte Geste durchzuführen, wie dies bei der „Faust“ möglich ist, muss eine Mindestdistanz zurückgelegt werden. Dies liegt daran, dass man, selbst wenn die Hand still ist, ein wenig zittert. Wenn das Zittern direkt dazu führen würde, dass eine bewegte Geste erkannt wird, wäre die Bedienung des Programms sehr mühsam.

Die Mindestdistanz um eine Bewegung zu erkennen, wurde auf experimentelle Weise bei 10 Pixel angelegt. Bei den durchgeführten Tests wurde erkannt, dass dieser Wert eine angenehme Bedienung ermöglicht.

Die Bewegungsrichtung wird dadurch herausgefunden, dass die im ersten Bild, wo die Hand auftaucht und im letzten Bild, in dem sie noch zu finden ist, die Handschwerpunkte gelesen werden. Durch die gegenseitige Lage dieser Punkte kann dann bestimmt werden, in welche Richtung sich die Hand bewegt. Aus diesen beiden Punkten berechnet sich auch die Bewegungsdistanz.

Die Orientierung der Hand wird aus dem Resultat von Kapitel 6.8 erhalten.

6.9.2 Gesten mit zwei Händen

Für die Gesten mit zwei Händen, wird nur eine Geste unterstützt. Dies ist die Geste mit zwei Fäusten. Mit Hilfe dieser Geste werden der Zoomfaktor und der Drehwinkel berechnet. Der Zoomfaktor und der Drehwinkel berechnen sich folgendermassen:

$$z = \frac{d_E}{d_A}$$
$$\alpha = \alpha_E - \alpha_A$$

Wie man sieht, werden jeweils der Anfangs- und Endabstand und ebenfalls der Anfangs- und Endwinkel zwischen den Händen während dieser bewegten Geste zur Berechnung herangezogen.

7 Programm für die Bilder- und Videogalerie

Dieses Kapitel gibt eine Übersicht über die Implementierung des Demonstrationsprogramms. Es werden einige Erklärungen zur Zustandsmaschine und den einzelnen Fenstern gegeben. Die Details jedoch sind im Anhang unter Punkt 11.5 zu finden.

Um den Programmaufbau kurz zu erklären wird zuerst noch das Klassendiagramm gezeigt.

7.1 Klassendiagramm

Dieses Kapitel beschäftigt sich kurz mit dem Aufbau und der Struktur des Programms. Die verwendeten Klassen werden ebenfalls erläutert.

Die folgende Abbildung zeigt anhand des Klassendiagramms, wie das Programm aufgebaut ist.

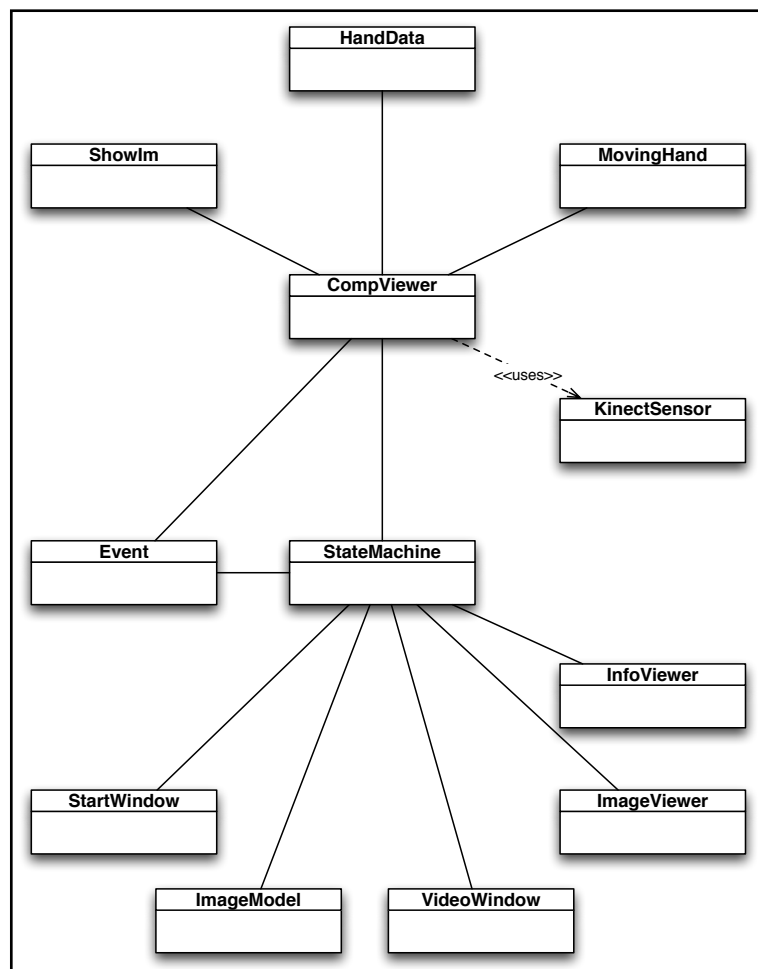


Abbildung 19: Klassendiagramm des Demonstrationsprogramms

Das obige Klassendiagramm zeigt deutlich, dass der Analyseteil des Programms (**CompViewer**, **ShowIm**, **HandData**, **MovingHand** und **KinectSensor**) klar von dem Anzeigeteil getrennt ist. Der für die Anzeige zuständige Programmteil wird von der

Zustandsmaschine (StateMachine) kontrolliert. Die Bildanalyse wird von der Klasse CompViewer kontrolliert und durchgeführt.

7.1.1 Bildanalyse

Dieser Programmteil setzt sich aus den folgenden Klassen und Structs zusammen.

- CompViewer,
- ShowIm,
- HandData,
- MovingHand,
- KinectSensor

Die Klasse CompViewer hat hier eine zentrale Rolle. Sie analysiert die Bilder, die von der Klasse KinectSensor geliefert werden und verarbeitet diese weiter. Dabei entstehen diverse Daten, welche für die verschiedenen Stadien in den Klassen HandData und im Struct MovingHand gespeichert werden. Am Ende der Analyse wird ein Event erzeugt und an das Objekt der Klasse StateMachine gesendet.

Der Struct ShowIm dient dazu, die Anzeige von verschiedenen Bildern in OpenCV zu beschleunigen und zu vereinfachen.

7.1.2 Anzeige der Fenster

Der Programmteil zur Anzeige der Fenster setzt sich aus den folgenden Klassen und Structs zusammen.

- Event,
- StateMachine,
- StartWindow,
- ImageModel,
- VideoWindow,
- ImageViewer,
- InfoViewer

Hierbei kontrolliert die Klasse StateMachine, was angezeigt wird. Die Klasse StateMachine wird durch das Auftreten und Abarbeiten von Events gesteuert. Die Klasse StartWindow beherbergt das Menüfenster. Die Klasse ImageModel wird benötigt, um eine Liste von Bildern zu erstellen, welche an das Galeriefenster übergeben wird. Das Videofenster ist in der Klasse VideoWindow untergebracht. Das Bildfenster wird mit Hilfe der Klasse ImageViewer erstellt. Das Docking Fenster wird in der Klasse InfoViewer erstellt.

7.2 Zustandsmaschine

Das Demonstrationsprogramm, welches eine Bilder- und Videogalerie enthält, beruht hauptsächlich auf einer Zustandsmaschine. Dadurch können die gewünschten Funktionalitäten des Programms auf sehr einfache Art und Weise implementiert werden.

Dieser Ansatz bietet sich an, da aus dem bisherigen Programm zur Analyse von Gesten und Bewegungen sehr einfach Ereignisse ausgelöst werden können. Ereignisse (auch Events) werden benötigt um die Zustandsmaschine zwischen den einzelnen Zuständen umschalten zu lassen.

Die nachfolgende Abbildung zeigt nur die Hauptzustandsmaschine, welche verwendet wird um die richtigen Fenster im richtigen Moment anzuzeigen.

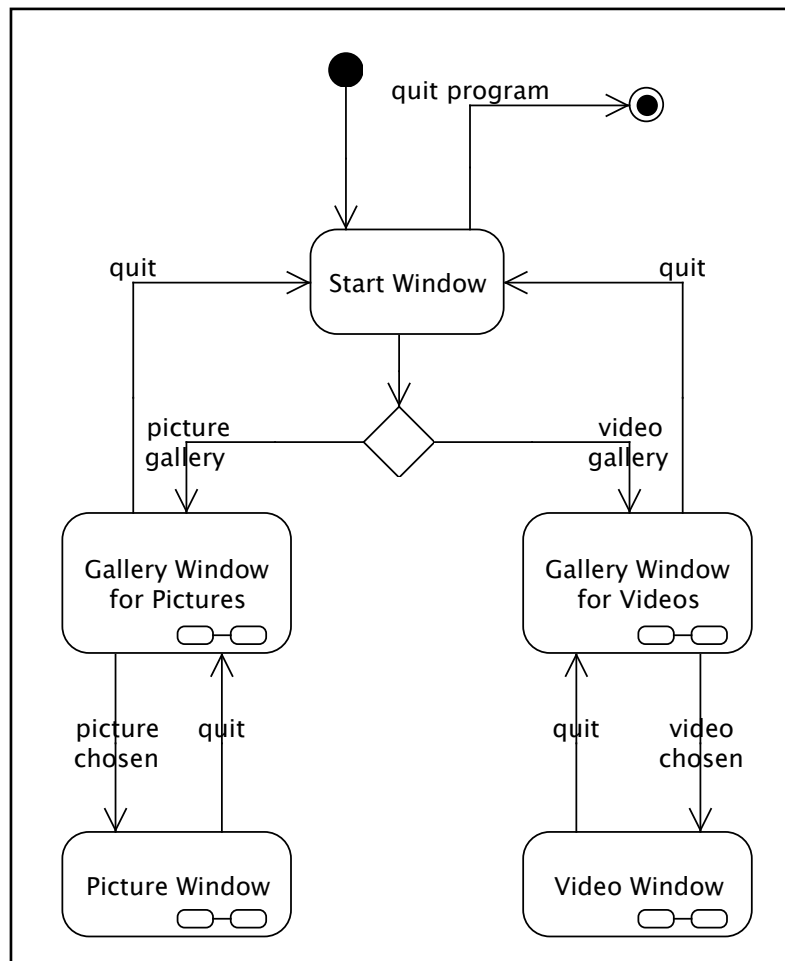


Abbildung 20: Hauptzustandsmaschine des Demonstrationsprogramms

Sobald das Programm gestartet wird, wird das Menüfenster zur Auswahl zwischen der Bilder- und Videogalerie angezeigt. Wenn nun eine Auswahl getroffen wurde, wird in den entsprechenden Zustand gewechselt. Wie man sieht enthalten die Zustände für die Galerien eine eigene angepasste Unterzustandsmaschine. Der Grund hierfür liegt darin, dass in den Galeriefenstern einige Manipulationen möglich sind. Der Benutzer kann hier innerhalb der Galerie scrollen. Diese Funktionalität wird innerhalb der Unterzustandsmaschinen behandelt.

Wählt der Benutzer ein Bild aus, welches er angezeigt haben möchte, wird das Galeriefenster deaktiviert und das Bildfenster aktiviert und im gleichen Zug wird die Zustandsmaschine in den nächsten Zustand bewegt. Wie auch der Galeriezustand enthält der Bildzustand eine Unterzustandsmaschine. Damit werden die entsprechenden Funktionalitäten für das Bildfenster ermöglicht und die zugehörigen Events werden in dieser Unterzustandsmaschine behandelt.

Genau gleich wie beim Galeriefenster für die Bilder funktioniert dies beim Galeriefenster für die Videos. Sobald der Benutzer ein Video ausgewählt hat, wird das Galeriefenster deaktiviert und das Videofenster angezeigt. Wie man in der Abbildung für die Zustandsmaschine weiter oben sieht, enthält der Zustand für die Videoanzeige ebenfalls eine Unterzustandsmaschine. Diese wird verwendet um die entsprechenden Events,

welche die Gestenerkennung generiert, zu verarbeiten, sobald sich die Zustandsmaschine mit der Verarbeitung hier befindet.

7.3 Programmfenster

Da jedes Programmfenster eigene Funktionalitäten aufweist, wird für jeden Programmteil eine Unterzustandsmaschine entworfen und implementiert. Diese Zustandsmaschinen werden in diesem Kapitel kurz erklärt.

7.3.1 Allgemein

Allen Funktionalitäten wird ein eigener Zustand in der entsprechenden Unterzustandsmaschine zugeordnet. Um in diesen Zustand hineinzugelangen muss die zugehörige Geste ausgeführt werden. Diese wurden in Tabelle 1 bestimmt.

Um aus dem Zustand wieder hinauszugehen, muss eine Geste ausgeführt werden, welche nicht dieselbe ist, wie um in den Zustand hineinzugelangen. Dies verbessert die Bedienung und verhindert, dass Gesten doppelt interpretiert werden und so Befehle direkt zweimal hintereinander ausgeführt werden.

Um dies zu untermalen kann das Schliessen des Fensters, beziehungsweise des Programms genannt werden. Wenn der Benutzer die Faust in der linken oberen Ecke still hält, wird in dieser Implementierung immer nur das entsprechende Fenster in das nachfolgende Fenster gewechselt oder eben das Programm geschlossen. Um die Funktionalitäten des neuen Fensters abrufen zu können, muss der Nutzer unbedingt eine andere Geste ausführen als „Faust still auf Position“. Wäre dies nicht der Fall und man würde auf jeden Fall aus dem Funktionalitätszustand direkt wieder auf den Hauptzustand (Show Gallery, Show Video und Show Picture) wechseln, würde das Programm die Befehle so rasch hintereinander ausführen dass der Benutzer kaum mehr reagieren kann. Dies kann man wiederum am Schliessen des Fensters illustrieren. Wenn in diesem Fall die Faust in der linken oberen Ecke still bleibt, schliesst man innerhalb von wenigen Augenblicken alle Fenster und das Programm, wenn man sich beispielsweise im Bildfenster befindet.

7.3.2 Menüfenster

Das Menüfenster besitzt nur eine einzige Funktionalität. Der Benutzer kann in diesem Fenster die Auswahl treffen ob er die Videogalerie oder die Bildergalerie öffnen oder das Programm schliessen möchte. Da in den Fällen des Öffnens der Galerien, die entsprechenden Zustandsmaschinen in Gang gesetzt werden, enthält dieses Fenster, beziehungsweise dieser Zustand, keine Unterzustandsmaschine. Auch das Schliessen des Programms macht keine Unterzustandsmaschine nötig.

7.3.3 Galeriefenster

Das Galeriefenster besitzt neben dem Auswählen des Bildes noch die Funktionalität des Scrollens. Aus diesem Grund wird hier eine Unterzustandsmaschine eingefügt. Damit können die Ereignisse für die Galeriefenster besser behandelt werden.

7.3.4 Bildfenster

Da das Bildfenster einige Funktionalitäten besitzt, wird für dieses Fenster ebenfalls eine Unterzustandsmaschine eingefügt. Diese wird so implementiert, dass jede Funktionalität ihren eigenen Zustand erhält. Dies gestaltet die Implementierung übersichtlicher.

7.3.5 Videofenster

Im Videofenster wird wie im Bildfenster für jede Funktionalität ein Zustand in die Unterzustandsmaschine eingefügt.

8 Schlussfolgerung

8.1 Resultat

Die Diplomarbeit konnte erfolgreich beendet werden. Die gesteckten Ziele wurden alle erreicht. Das Demonstrationsprogramm konnte, wie mit dem Verantwortlichen des Forschungsinstitut Idiap und der HES-SO Valais//Wallis besprochen, implementiert werden. Während der Implementation des Demonstrationsprogramms stellte sich ein Problem. Qt enthält in der Version 4.7.4 keine Multimodulmodule, welche das Vor- und Zurückspulen ermöglichen. Deshalb wurde anstelle dieser Funktionalität, die Lautstärkenregelung eingefügt.

Die definierten Funktionalitäten für das Demonstrationsprogramm sind sehr stark an ähnliche existierende Programme angelehnt. Auch die dazugehörigen Gesten wurden so natürlich wie möglich gewählt. Die Gestenwahl ist sehr stark an die Touchscreen-Displays von Smartphones angelehnt. Grund hierfür ist, dass sehr viele Personen schon ein Smartphone bedient haben und mit grosser Wahrscheinlichkeit auch ähnliche Gesten in diesem Demonstrationsprogramm versuchen werden.

Der Programmteil zur Gestenerkennung zeigt deutlich, wie stark das Kinect Sensorsystem mit Infrarotkamera das Finden von Händen und Erkennen von Gesten vereinfacht. Man kann eine Funktionszone einrichten, welche eine bestimmte Distanz zur Kamera hat. Daraufhin müssen sich die Hände innerhalb dieser Zone bewegen um erkannt zu werden und Befehle an das Programm schicken zu können. Dadurch wird im Prinzip das Finden der Hände an eine Distanz gebunden und nicht an eine Hautfarbe. Um die Gesten zu erkennen, wurden viele numerische Kriterien eingesetzt. Es wurde darauf geachtet, dass möglichst alle Kriterien relativen Charakter haben und nicht absolut sind. Dies erhöht die Brauchbarkeit des Programms. Das einzige ausschliessende Kriterium ist die Handfläche. Diese Charakteristik muss eine Mindestanzahl von 7000 Pixel aufweisen, sonst wird diese Fläche ignoriert. Diese Mindestfläche wurde aber so gewählt, dass nicht nur Personen mit grossen Händen dieses Programm steuern können, sondern möglichst viele die Möglichkeit dazu haben.

8.2 Analyse

Einige Probleme boten sich bei der Gestenerkennung. Beispielsweise wenn sich beide Hände überlagern oder sogar berühren, treten Probleme auf. Die Resultate der Bewegungserkennung sind in diesen Fällen nicht mehr zu gebrauchen.

Eine weitere Problematik, die während der Programmierung auftrat, war die Stabilität der verwendeten Bibliotheken. Dies ist vor allem bei OpenNI ein Problem. Die verwendete Version ist für Mac OS X als instabil gekennzeichnet. Da OpenNI jedoch ständig weiter entwickelt wird, kann damit gerechnet werden, dass eine zukünftige Version stabil sein wird. Bei den durchgeführten Tests, obwohl die Bibliothek als instabil gekennzeichnet war, lief sie sehr zuverlässig über lange Zeit (über eine Stunde).

Auch Microsoft selbst hat Ende Mai eine SDK für die Programmierung mit der Kinect als Betaversion herausgegeben. Diese SDK läuft zwar nur mit Visual Studio und ist noch nicht für andere Plattformen erschienen. Deshalb könnte eine zukünftige Weiterführung

des Projekts mit Hilfe dieser SDK auf einer Windows Plattform in Betracht gezogen werden.

Bei einer Analyse des Demonstrationsprogramms fällt vor allem die hohe CPU-Auslastung auf. Da dieser Wert so hoch ist (über 40 % bei einem 2.13 GHz Core 2 Duo von Intel), kann man daraus schliessen, dass es sich momentan sicher noch nicht anbietet, die Computermouse durch Infrarotkameras zu ersetzen. Hauptsächlich wird der Prozessor durch die Bildanalyse sehr stark belastet. Eine Möglichkeit wäre aber ein Hardwarezwischenstück zu erstellen, welches zwischen Computer und Kinect eingebaut wird. Diese Hardware hätte die Aufgabe die Bildanalyse so weit durchzuführen, dass nur noch die Befehle an den Computer weitergeschickt werden können.

Auch die Bedienung über Gesten ist nicht immer angenehm. Wenn man einen Computer mehrere Stunden mit gestreckten Armen bedienen muss, ist dies sicherlich nicht förderlich für eine einfache und gute Bedienung.

Das erstellte Demonstrationsprogramm läuft sehr gut. Die Bedienung mit den verschiedenen Gesten funktioniert auch ohne Probleme. Da die verwendeten Algorithmen sehr einfach gehalten wurden, ist die Bewegungserkennung und Befehlsübermittlung rasch genug, dass der Benutzer nicht lange warten muss, bis die gewünschte Aktion abgerufen wird. Es werden keine Handmodelle zur Gestenerkennung herangezogen, rein mit Hilfe der Umrisssequenz der Hand werden alle gewünschten Handformen erkannt und die Gesten herausgelesen. Diese Methode ist sehr schnell und für dieses Demonstrationsprogramm sicherlich exakt genug. Sie erlaubt es dreissig Bilder pro Sekunde zu analysieren und daraus die Gesten zu erkennen.

8.3 Vorschläge zur Weiterentwicklung

Falls in einer künftigen Version des Demonstrationsprogramms mehr Handformen unterstützt werden sollen, muss die Methode, welche momentan verwendet wird, verfeinert werden. Mit der aktuellen Methode ist es zwar noch möglich einige Gesten mehr zu erkennen als es momentan getan wird. Vor allem im Bereich für zwei Hände gibt es noch sehr viele Möglichkeiten. Auch Bewegungen von Gesten mit nur einer Hand sind nicht voll ausgeschöpft.

Das Demonstrationsprogramm ist hinsichtlich der Fenster, welche für die Galerien und Anzeige der Bilder und Videos gebraucht werden verbesserbar. Momentan werden die Fenster alle mit einer fixen Grösse von 640 auf 480 Pixel erstellt. Hier könnte ein relatives Koordinatensystem eingeführt werden, um die Übersetzung zwischen den aufgenommenen Punkten von der Kinect (aus einem Bild mit fix 640 auf 480 Pixel) zu den Punkten auf den Qt Fenstern zu ermöglichen.

Um das Programm für den Nutzer angenehmer gestalten zu können, sollte noch versucht werden auf den Qt Fenstern die Position der Hand mit Hilfe von Punkten anzuzeigen. Während der Entwicklung wurden verschiedene Versuche vorgenommen dies zu realisieren, aber es fand sich keine Möglichkeit die Handpositionen auf den Qt Fenstern auf den Vordergrund zu zeichnen. In den meisten Fenstern wurden die gezeichneten Punkte durch den restlichen Inhalt (Buttons, Videowidget, oder ähnlichem) einfach überdeckt. Momentan wird zwar die Position und das Analyseresultat der Gestenerkennung in einem Docking Fenster angezeigt. Dies bietet schon einen guten Komfort, wenn sich aber die Möglichkeit findet die Position und die

Gestenanalyse direkt innerhalb der Anzeige realisieren zu lassen, wäre dies sicher noch besser.

Wie weiter oben schon erwähnt, erlauben die Multimodulmodule von der momentanen Qt Version nicht, in Filmen zu spulen. In einer folgenden Version ist eine Verbesserung hinsichtlich dieser Funktionalitäten geplant und kann somit auch in dem Demonstrationsprogramm eingebaut werden.

8.4 Gedanken zur Diplomarbeit

Das Projekt für diese Diplomarbeit bietet sehr viele Einblicke in die Computer Vision. Es können sehr viele interessante Erfahrungen gesammelt werden, welche nachher nicht nur auf die Themengebiete der Bildanalyse, sondern auch auf die Datenanalyse selbst angewendet werden können. Das Demonstrationsprogramm zeigt auch eindrücklich, was mit der heutigen Technik möglich ist. Mit dem Ansatz der Gestenerkennung für die Bedienung eines Computers werden sehr viele interessante Gedanken für die Möglichkeiten dieses Gebiets in der Zukunft in Gang gesetzt.

Bei einer Recherche vorgängig zu dieser Diplomarbeit wurde deutlich, dass sehr viele Projekte im Zusammenhang mit der Kinect, seit deren Verkaufsstart Ende 2010, realisiert werden.

Beispielsweise befasst sich das Institut für Bildverarbeitung an der ETH Zürich ebenfalls mit diesem Sensorsystem. Dieses Institut versucht 3D-Objekte zu scannen und das Resultat wiederzugeben und sogar an einen anderen Computer zu senden. Auf diesem Weg wird die Kommunikation verändert und man erhält die Möglichkeit auf sehr einfache Weise dem Kommunikationspartner das zur Diskussion stehende Objekt zu zeigen.

Ein weiteres Projekt dieses Instituts ist die Erkennung und Interpretation des Fingeralphabets.

Diese Angaben wurden aus der Sendung „Einstein“ des Schweizer Fernsehen SF1 genommen. Dort werden auch noch diverse andere Projekte vorgestellt. Der Link zu dieser Sendung ist in Punkt 17 der Bibliographie aufgeführt.

9 Datum und Unterschrift

Dieser Bericht und das Projekt wurden realisiert von:

_____Kummer Fernando

Mühlebach

08.07.2011

10 Danksagung

Ich möchte mich hier kurz bei allen Personen bedanken, welche mich in irgendeiner Weise unterstützt haben um diese Arbeit erfolgreich abschliessen zu können.

Persönlich danken möchte ich meinen beiden Betreuern Dr. Pierre Roduit und Flavio Tarsetti. Beide haben mir des Öfteren geholfen meine Ideen richtig zu formen und in Code zu verwandeln. Sie haben mir meine fehlenden Kenntnisse der Computer Vision zugetragen und alles für dieses Projekt nötige (und noch vieles mehr) erklärt.

Weiter hat sich Ralph Martig immer wieder tapfer als Testperson ins Zeug gelegt und mein Programm auf die Probe gestellt. Auch Michael Clausen möchte ich erwähnen. Er hat mir immer wieder mit kleinen Tipps und Tricks für Qt und XCode zur Seite gestanden. Thomas Sterren hat mir ebenfalls bei einigen Fragen zur Programmierung geholfen.

Auch die Development Group am Idiap Research Institute möchte ich nicht vergessen. Sie haben mir immer wieder mit Ideen zu Gesten und deren Erkennung geholfen. Sie halfen mir auch bei Problemen beim Schreiben des Programms.

11 Anhang

11.1 Verzeichnisse für Abbildungen und Tabellen

11.1.1 Abbildungsverzeichnis

ABBILDUNG 1: KINECT-SENSORSYSTEM MIT BESCHRIFTUNG DER SENSORIK [18].....	4
ABBILDUNG 2: KINECT-SENSORSYSTEM INNENANSICHT [19].....	4
ABBILDUNG 3: EIGENHEIT DES KINECT-SENSORSYSTEMS.....	5
ABBILDUNG 4: MENÜFENSTER DES DEMONSTRATIONSPROGRAMMS.....	7
ABBILDUNG 5: GALERIEFENSTER FÜR DIE BILDER- UND VIDEOGALERIE.....	8
ABBILDUNG 6: BILDFENSTER ZUR PRÄSENTATION VON JEDEM BILD EINZELN.....	8
ABBILDUNG 7: VIDEOFENSTER ZUR PRÄSENTATION VON VIDEOS.....	9
ABBILDUNG 8: INFORMATIONSFENSTER MIT HAND INNERHALB DER FUNKTIONSZONE.....	9
ABBILDUNG 9: INFORMATIONSFENSTER MIT HAND AUSSERHALB DER FUNKTIONSZONE.....	10
ABBILDUNG 10: INFORMATIONSFENSTER MIT RECHTECKEN FÜR DAS STARTMENÜ.....	10
ABBILDUNG 11: INFORMATIONSFENSTER MIT RECHTECKEN FÜR DIE GALERIEFENSTER.....	11
ABBILDUNG 12: INFORMATIONSFENSTER MIT RECHTECKEN FÜR BILD- UND VIDEOANZEIGE.....	11
ABBILDUNG 13: DOCKING FENSTER MIT ANALYSERESULTAT UND POSITIONSANZEIGE.....	12
ABBILDUNG 14: FLUSSDIAGRAMM DER GESTENERKENNUNG.....	14
ABBILDUNG 15: 8 BIT VERSION DES ORIGINALBILDS VON DER KINECT.....	15
ABBILDUNG 16: HAND IN DER FUNKTIONSZONE MIT KLEINEM TEIL DES UNTERARMS.....	16
ABBILDUNG 17: BINÄRES HANDBILD.....	18
ABBILDUNG 18: HANDBILD MIT AUSWERTUNG.....	19
ABBILDUNG 19: KLASSENDIAGRAMM DES DEMONSTRATIONSPROGRAMMS.....	25
ABBILDUNG 20: HAUPTZUSTANDSMASCHINE DES DEMONSTRATIONSPROGRAMMS.....	27
ABBILDUNG 21: GESTE ZUM AUSWÄHLEN MIT DER RECHTEN HAND.....	38
ABBILDUNG 22: GESTE ZUM AUSWÄHLEN MIT DER LINKEN HAND.....	38
ABBILDUNG 23: RECHTE FAUST HORIZONTAL BEWEGT.....	38
ABBILDUNG 24: LINKE FAUST HORIZONTAL BEWEGT.....	38
ABBILDUNG 25: RECHTE HAND ZWEI FINGER GESTRECKT NACH LINKS ORIENTIERT.....	39
ABBILDUNG 26: LINKE HAND ZWEI FINGER GESTRECKT NACH LINKS ORIENTIERT.....	39
ABBILDUNG 27: RECHTE HAND ZWEI FINGER GESTRECKT NACH RECHTS ORIENTIERT.....	39
ABBILDUNG 28: LINKE HAND ZWEI FINGER GESTRECKT NACH RECHTS ORIENTIERT.....	39
ABBILDUNG 29: RECHTE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH LINKS ORIENTIERT.....	39
ABBILDUNG 30: LINKE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH LINKS ORIENTIERT.....	39
ABBILDUNG 31: RECHTE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH RECHTS ORIENTIERT.....	40
ABBILDUNG 32: LINKE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH RECHTS ORIENTIERT.....	40
ABBILDUNG 33: BILD MIT MÖGLICHEN DREHBEWEGUNGEN.....	40
ABBILDUNG 34: BILD MIT BEWEGUNGEN ZUM ZOOMEN.....	40
ABBILDUNG 35: RECHTE GEÖFFNETE HAND BEWEGT.....	41
ABBILDUNG 36: LINKE GEÖFFNETE HAND BEWEGT.....	41
ABBILDUNG 37: RECHTE HAND MIT GESPREIZTEN FINGERN.....	41
ABBILDUNG 38: LINKE HAND MIT GESPREIZTEN FINGERN.....	41
ABBILDUNG 39: RECHTE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH OBEN ORIENTIERT.....	42
ABBILDUNG 40: LINKE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH OBEN ORIENTIERT.....	42
ABBILDUNG 41: RECHTE FAUST BEWEGUNGSMÖGLICHKEITEN ZUM EINSTELLEN DER LAUTSTÄRKE.....	42
ABBILDUNG 42: LINKE FAUST BEWEGUNGSMÖGLICHKEITEN ZUM EINSTELLEN DER LAUTSTÄRKE.....	42
ABBILDUNG 43: RECHTE HAND ZWEI FINGER GESTRECKT NACH LINKS ORIENTIERT.....	43
ABBILDUNG 44: LINKE HAND ZWEI FINGER GESTRECKT NACH LINKS ORIENTIERT.....	43
ABBILDUNG 45: RECHTE HAND ZWEI FINGER GESTRECKT NACH RECHTS ORIENTIERT.....	43
ABBILDUNG 46: LINKE HAND ZWEI FINGER GESTRECKT NACH RECHTS ORIENTIERT.....	43
ABBILDUNG 47: RECHTE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH LINKS ORIENTIERT.....	43
ABBILDUNG 48: LINKE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH LINKS ORIENTIERT.....	43
ABBILDUNG 49: RECHTE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH RECHTS ORIENTIERT.....	44
ABBILDUNG 50: LINKE HAND ALLE FINGER ZUSAMMEN GESTRECKT NACH RECHTS ORIENTIERT.....	44
ABBILDUNG 51: HAUPTZUSTANDSMASCHINE.....	53

ABBILDUNG 52: UNTERZUSTANDSMASCHINE BILDERGALERIE	54
ABBILDUNG 53: UNTERZUSTANDSMASCHINE DER VIDEOGALERIE	55
ABBILDUNG 54: UNTERZUSTANDSMASCHINE FÜR DIE BILDANZEIGE	56
ABBILDUNG 55: UNTERZUSTANDSMASCHINE FÜR DIE VIDEOANZEIGE.....	57

11.1.2 Tabellenverzeichnis

TABELLE 1: ZUORDNUNG DER GESTEN ZU DEN FUNKTIONALITÄTEN.....	13
TABELLE 2: TESTWERTE KOMPAKTHEIT	20
TABELLE 3: TESTWERTE EXZENTRIZITÄT.....	21
TABELLE 4: TESTWERTE DISTANZ SCHWERPUNKT ZU RECHTECKMITTELPUNKT	21

11.2 Quellenverzeichnis - Bibliografie

- [1] Dokumentation und Download der Bibliothek OpenNI,
<http://www.openni.org/>, 05. März 2011
- [2] Beim Download mitgelieferte Dokumentation zur Verwendung der
 Bibliothek OpenNI, doxygen Dokumentation von OpenNI, 05. März 2011
- [3] Learning OpenCV, Gary Bradski & Adrian Kaehler, O'Reilly, 2008
- [4] Punkte aus einer Contour herauslesen,
<http://tech.groups.yahoo.com/group/OpenCV/message/2954>, 23. Mai
 2011
- [5] Punkte aus einer Contour herauslesen,
<http://tech.groups.yahoo.com/group/OpenCV/message/28979>, 23. Mai
 2011
- [6] Füllen ausserhalb der gefundenen Contour,
<http://tech.groups.yahoo.com/group/OpenCV/message/33189>, 26. Mai
 2011
- [7] Farbinvertierung eines IplImages,
<http://opencvbasicdemos.blogspot.com/2010/06/image-color-inversion.html>, 26. Mai 2011
- [8] Einsatz der Momente,
<http://public.cranfield.ac.uk/c5354/teaching/dip/opencv/SimpleImageAnalysisbyMoments.pdf>, 27. Mai 2011
- [9] Doxygen Befehle,
<http://www.stack.nl/~dimitri/doxygen/commands.html>, 31. Mai 2011
- [10] C++ GUI-Programming with Qt 4, Jasmin Blanchette & Mark Summerfield,
 2006
- [11] Bibliothek libqxt, <http://dev.libqxt.org/libqxt/wiki/Home>, 07. Juni 2011

- [12] Lösung zum Problem der Kompilierung von libqxt,
<http://dev.libqxt.org/libqxt/issue/50/install-issue-qxt-060-on-mac-os-x-1063>, 07. Juni 2011
- [13] libqxt Dokumentation, <http://libqxt.bitbucket.org/doc/0.6/modules.html>,
07. Juni, 2011
- [14] libqxt Flowview Beispiel,
<http://dev.libqxt.org/libqxt/src/8342fb08fd61/examples/qxtflowview/>,
07. Juni 2011
- [15] Bild rotieren in Qt, [http://wiki.forum.nokia.com/index.php/CS001514 -
Rotate picture in Qt](http://wiki.forum.nokia.com/index.php/CS001514_-_Rotate_picture_in_Qt), 13. Juni 2011
- [16] Video abspielen in Qt, <http://doc.trolltech.com/main-snapshot/demos-qmediaplayer.html>, 14. Juni 2011
- [17] Video zur Präsentation der Kinect im Schweizer Fernsehen in der Sendung Einstein, <http://www.videportal.sf.tv/video?id=dc8bf51-4c7a-4efd-adf5-830e1f6a182d>, 4. März 2011
- [18] Bild der Kinect mit Beschriftungen der Sensoren, <http://mygggo.com/wp-content/uploads/2010/06/kinect12.png>, 05. März 2011
- [19] Bild der Kinect mit Innensicht,
http://www.lolgamesbrasil.com.br/2010_12_01_archive.html, 05. März 2011

11.3 Erklärungen zur Definition der Steuergesten

Dieses Kapitel ist eine Bedienungsanleitung zu dem Demonstrationsprogramm, welches in Kapitel 5 des Berichts entworfen wurde.

11.3.1 Menüfenster

Im Menüfenster werden verschiedene Punkte ausgewählt. Deshalb muss eine Geste definiert werden, welche angibt, dass ein Menüpunkt ausgewählt wurde. Da im Menüfenster nur Menüpunkte ausgewählt werden, darunter auch das Schliessen des Programms, welches in der aktiven Ecke oben links ausgewählt werden kann, wird eine Geste für diese Funktionalität definiert. Die gewählte Geste besteht darin, dass man die offene Hand auf den gewünschten Menüpunkt bewegt und dann die Hand zu einer Faust schliesst.

Die nachstehenden Abbildungen zeigen die Geste für das Auswählen eines Menüpunkts.



Abbildung 21: Geste zum Auswählen mit der rechten Hand



Abbildung 22: Geste zum Auswählen mit der linken Hand

11.3.2 Galeriefenster (Bilder- und Videogalerie)

In den beiden Galeriefenstern werden die möglichen Bilder und Videos angezeigt. Um ein Bild oder ein Video auszuwählen, muss eine Geste definiert werden, welche dazu passt. Da die Geste zum Auswählen eines Menüpunkts schon für das Menüfenster definiert worden ist, wird diese wiederum weitergebraucht. Eine weitere Aufgabe, welche in diesem Fenster über eine Geste verfügen muss, ist das Scrollen. Diese Geste besteht darin, dass die Faust entweder nach rechts oder links bewegt wird.

Die folgenden Abbildungen zeigen die Geste zum Scrollen.



Abbildung 23: Rechte Faust horizontal bewegt



Abbildung 24: Linke Faust horizontal bewegt

11.3.3 Bildfenster

Im Fenster zur Bildanzeige werden mehr Gesten unterstützt als in den Galeriefenstern oder dem Menüfenster. Der Benutzer kann hier beispielsweise zum vorherigen oder zum nächsten Bild gelangen. Dies wird erreicht, wenn man die Finger zusammenlegt und die Hand nach rechts (nächstes) oder links (vorhergehendes) orientiert.

Die nachfolgenden Abbildungen verdeutlichen diese Gesten.



Abbildung 25: Rechte Hand zwei Finger gestreckt nach links orientiert



Abbildung 26: Linke Hand zwei Finger gestreckt nach links orientiert



Abbildung 27: Rechte Hand zwei Finger gestreckt nach rechts orientiert



Abbildung 28: Linke Hand zwei Finger gestreckt nach rechts orientiert



Abbildung 29: Rechte Hand alle Finger zusammen gestreckt nach links orientiert



Abbildung 30: Linke Hand alle Finger zusammen gestreckt nach links orientiert



Abbildung 31: Rechte Hand alle Finger zusammen gestreckt nach rechts orientiert



Abbildung 32: Linke Hand alle Finger zusammen gestreckt nach rechts orientiert

Weiter kann der Benutzer das Bild vergrössern oder verkleinern und gleichzeitig auch drehen. Diese Geste erfordert beide Hände und ist ähnlich wie die entsprechende Fingerbewegung auf den Touchscreens eines Smartphones. Hier wird das Bild mit beiden Händen gepackt (Fäuste irgendwo im Bild), dann werden die Hände mit oder ohne Drehbewegung aufeinander zu- oder voneinander wegbewegt.

Die nachfolgenden Abbildungen veranschaulichen diese Geste.

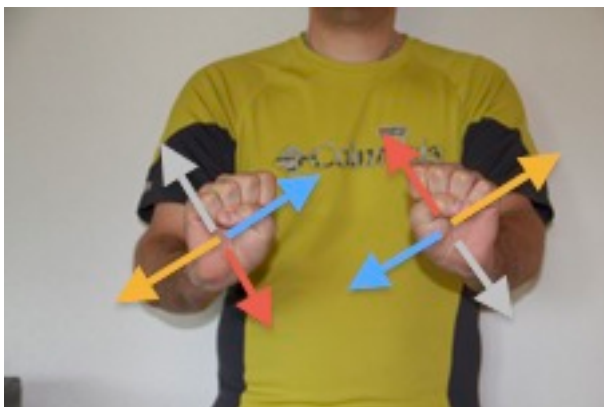


Abbildung 33: Bild mit möglichen Drehbewegungen



Abbildung 34: Bild mit Bewegungen zum zoomen.

Da der Benutzer das Bild vergrössern kann, besteht hier auch die Möglichkeit das Bild zu bewegen. Diese Geste wird nur noch von einer Hand ausgeführt. Sie ähnelt der Scrollbewegung in den Galeriefenstern. Man packt das Bild, das will heissen man legt die offene Hand auf das Bild, und bewegt es in die gewünschte Position. Es wird hier die offene Hand gewählt, da in den Tests gemerkt wurde, dass, wenn beim Zoomen und Drehen eine Hand „verloren geht“, die Geste für die Bewegung des Bildes mit einer Faust sehr viele Unannehmlichkeiten bereitet und das Bild ungewollt bewegt wird.

Die nachstehenden Abbildungen sollen diese Geste verdeutlichen.

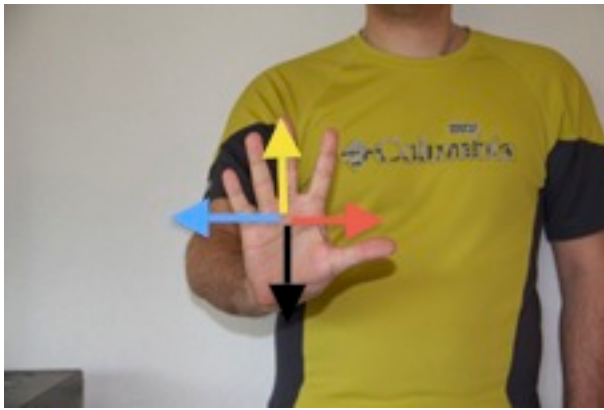


Abbildung 35: Rechte geöffnete Hand bewegt

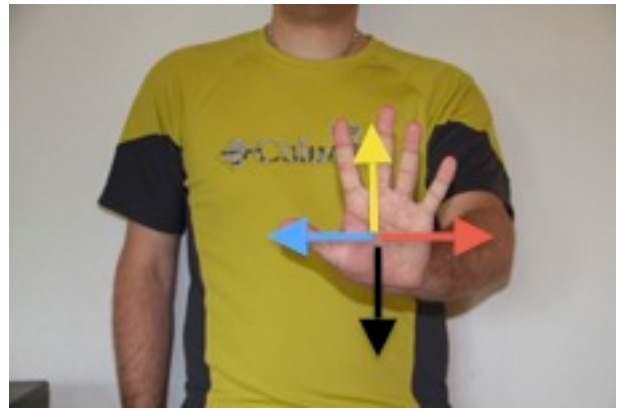


Abbildung 36: Linke geöffnete Hand bewegt

11.3.4 Videofenster

In diesem Fenster werden die Videos in Grossformat abgespielt. Der Benutzer kann das Abspielen jederzeit unterbrechen, beziehungsweise pausieren. Die Geste dazu ist die offene Hand mit gespreizten Fingern.

Die folgenden Abbildungen verdeutlichen diese Geste.



Abbildung 37: Rechte Hand mit gespreizten Fingern



Abbildung 38: Linke Hand mit gespreizten Fingern

Um den Film fortzusetzen kann der Benutzer die Finger der offenen Hand zusammenlegen und die Fingerspitzen nach oben richten. Die nachfolgende Abbildung verdeutlicht diese Geste.



Abbildung 39: Rechte Hand alle Finger zusammen gestreckt nach oben orientiert



Abbildung 40: Linke Hand alle Finger zusammen gestreckt nach oben orientiert

Es wird auch die Möglichkeit geboten die Lautstärke des Films zu regulieren. Um diese Funktionalitäten im Programm abzurufen, wird die Faust entweder nach oben (lauter stellen) oder nach unten (leiser stellen) bewegt.

Die folgenden Abbildungen verdeutlichen diese Geste.



Abbildung 41: Rechte Faust Bewegungsmöglichkeiten zum Einstellen der Lautstärke



Abbildung 42: Linke Faust Bewegungsmöglichkeiten zum Einstellen der Lautstärke

Weiter kann der Benutzer auch zum nächsten oder vorhergehenden Film wechseln. Diese Funktionen des Programms werden abgerufen, wenn die Hand eine Orientierung nach links (vorhergehender) oder nach rechts (nächster) erhält. Die Hand erhält eine Orientierung, wenn entweder zwei Finger zusammen ausgestreckt werden oder wenn alle Finger zusammen ausgestreckt werden und die Finger in die entsprechende Richtung zeigen.

Um diese Gestik zu verdeutlichen, dienen folgende Abbildungen.



Abbildung 43: Rechte Hand zwei Finger gestreckt nach links orientiert



Abbildung 44: Linke Hand zwei Finger gestreckt nach links orientiert



Abbildung 45: Rechte Hand zwei Finger gestreckt nach rechts orientiert



Abbildung 46: Linke Hand zwei Finger gestreckt nach rechts orientiert



Abbildung 47: Rechte Hand alle Finger zusammen gestreckt nach links orientiert



Abbildung 48: Linke Hand alle Finger zusammen gestreckt nach links orientiert



Abbildung 49: Rechte Hand alle Finger zusammen gestreckt nach rechts orientiert



Abbildung 50: Linke Hand alle Finger zusammen gestreckt nach rechts orientiert

11.4 Erklärungen zum Programm zur Gestenerkennung

In diesem Kapitel werden die Details zur Gestenerkennung besprochen. Es wird das Vorgehen gezeigt um die Gesten zu erkennen und auszuwerten. In diesem Sinne ist dieses Kapitel auch so zu interpretieren, dass jedes der folgenden Unterkapitel wie ein Punkt einer Anleitung zu verstehen ist. Dies bedeutet auch, dass jeder Schritt durchgeführt werden muss um eine erfolgreiche Gestenerkennung durchführen zu können. Dieses Kapitel ist eine Ergänzung zum Kapitel 6 des Berichts.

Die gesamte Bildanalyse, welche hier beschrieben wird, bedient sich der Bibliothek OpenCV, welche für solche Zwecke entworfen wurde.

11.4.1 Funktionszone

Da die hier analysierten Bilder nicht RGB Bilder sind sondern Graustufenbilder mit 16 Bit Tiefe, wobei auch jeder Pixelwerte die Distanz zwischen der Kamera und dem realen Punkt darstellt, kann eine sogenannte Funktionszone eingerichtet werden.

Durch eine Funktionszone wird erreicht, dass sich der Benutzer an einem genauen Punkt aufhalten muss. Somit können schon etwaige Störeinflüsse auf ein Minimum reduziert werden.

Wenn sich nun die Hände des Benutzers in dieser Zone bewegen, wird dies von dem Programm wahrgenommen. Dies geschieht folgendermassen.

Jedes Bild, das von der Funktionszone aufgenommen wird, wird analysiert. Es wird zuerst in ein binäres Bild umgewandelt. Vorder- und Hintergrund der Funktionszone erscheinen schwarz und alles innerhalb der Funktionszone erscheint als weisse Fläche.

Im nun folgenden Schritt werden zuerst kleine Bildstörungen, welche als Linie oder kleine Fläche erscheinen, mit Hilfe von mehrmaligem Erweitern der schwarzen Fläche und dem danach folgenden mehrmaligen Verbreitern der weissen Fläche wegradiert. Dieser Vorgang wird „Open“ genannt.

Da dies aber noch nicht reicht, wird nun für jede weisse Fläche die zugehörige Aussenlinie berechnet. Da eine Hand eine grosse weisse Fläche darstellt, wird nun, mit Hilfe der Aussenlinien der gefundenen weissen Flächen, deren Inhalt berechnet. Dies geschieht mit Hilfe der Momente der Bibliothek OpenCV.

Falls eine Fläche eine Mindestgrösse von 7000 Pixel erreicht, wird angenommen, dass dies eine Hand mit einem Teil des Unterarms darstellt. Eine weitere Annahme wird getroffen, in dem immer nur die zwei ersten genügend grossen Flächen behandelt und als Hände mit Unterarmteilen betrachtet werden. Diese Annahme macht jedoch Sinn, da sich die Funktionszone direkt vor der Kamera befindet und man davon ausgehen kann, dass sich immer nur ein Benutzer vor der Kamera aufhält und versucht das Programm zu bedienen. Auch wird sich kaum etwas zwischen der Kamera und dem Benutzer befinden, da dies ja die Bedienung des Programms stören würde. Es wurden kleinere Störungen entfernt, würden sich grössere Objekte zwischen Bediener und Kamera befinden, müsste sich der Benutzer selbst bemühen, diese zu entfernen.

Da nun Flächen vorhanden sein können, welche eine Hand darstellen, wird für jede Hand eine Handzone eingerichtet und ebenfalls für jede Hand werden Bilder erstellt, welche nur noch diese repräsentieren. Dies jedoch ist Inhalt des nächsten Kapitels.

11.4.2 Handzone

Die Handzone wird genutzt um für jede Hand einzeln ein Bild zu erhalten um später darauf Berechnungen durchführen zu können. Ein weiterer Vorteil der Handzone findet sich darin, dass sich die übrigen Unterarmteile entfernen lassen. Um eine Handzone einrichten zu können wird in einem ersten Schritt der Punkt gesucht, welcher die kürzeste Distanz zur Kamera aufweist. Dieser Punkt ist auf dem Umriss der Hand enthalten. Ist dieser Punkt gefunden, können die Grenzen der Handzone bestimmt werden.

Nach dem die Grenzen für die Handzone bekannt sind, wird ein Threshold ausgeführt, um nur noch diese Zone auf den Bildern zu finden. Da sich nun aber immer noch eine zweite unerwünschte Hand auf dem Bild dieser Zone befinden kann, muss diese Hand noch entfernt werden. Wie eingangs des Kapitels erwähnt wurde, werden jedoch nicht nur die unerwünschte Hand, sondern auch die übrigen Unterarmteile mit Hilfe dieser Handzone entfernt.

Um sicherzugehen, dass sich keine zweite Hand auf dem Bild mehr befindet, werden trotzdem noch folgende Schritte durchgeführt. Es kann ja der Fall sein, dass beide Hände den genau gleichen Abstand zur Kamera besitzen.

Die zweite ungewünschte Hand wird entfernt in dem man eine Maske eines Bildes erstellt, dass nur die gewünschte Hand enthält, dies geschieht mit dem Handumriss, welcher in einem vorherigen Schritt gefunden wurde. Da der Handumriss auch noch ungewünschte Teile des Unterarms enthält, kann dieser nicht dazu verwendet werden, direkt das gewünschte Bild der Hand anzufertigen. Nichtsdestotrotz wird dieser Umriss dazu genutzt eine Maske zu erstellen, welche erlaubt eine zweite Hand zu entfernen. Dies wird für beide Hände, falls zwei Hände gefunden wurden, genau gleich durchgeführt.

Mit Hilfe dieser Maske wird sich am Ende nur noch die gewünschte Hand auf dem Bild befinden. Alle restlichen weissen Flächen verschwinden direkt.

Für jede gefundene Hand wird nun ein binäres Bild erstellt, welches erlaubt weitere Berechnungen durchzuführen. Es wird auch ein RGB Bild aus dem binären Bild erstellt, auf welchem diverse Daten angezeigt werden können, um den Benutzer zu informieren.

Sobald nun für jede Hand die nötigen Bilder vorhanden sind, kann zum nächsten Punkt fortgeschritten werden. Hierbei werden die wichtigen Daten jeder Hand berechnet um später die Gesten erkennen zu können.

11.4.3 Handdaten

Hier wird davon ausgegangen, dass für jede gefundene Hand ein binäres Bild und ein RGB Bild vorhanden ist. Das binäre Bild wird dazu verwendet um charakteristische Eigenschaften der Hand, respektive der Geste zu berechnen. Alle in diesem Kapitel durchgeführten Rechnungen stützen sich auf die Formeln aus dem Dokument [8]. Um diese Aufgabe zu erfüllen wurde folgendermassen vorgegangen.

In einem ersten Schritt wird wiederum der Umriss der Hand berechnet, dies geschieht auf der Grundlage des binären Bildes. Mit Hilfe des Handumrisses können die Momente der Hand berechnet werden. Diese Berechnung wird mit Hilfe von OpenCV durchgeführt.

Die wichtigen Punkte der CvMoment Struktur sind hierbei die folgenden:

- m00: Fläche innerhalb des Umrisses
- m10: Wird gebraucht um den Schwerpunkt zu berechnen
- m01: Ebenfalls Teil der Berechnung des Schwerpunkts
- m11: Wird genutzt für die Berechnung der zentralen Momente
- m20: Ist Teil der Berechnung der zentralen Momente
- m02: Gebraucht für die Berechnung der zentralen Momente

Die Fläche und der Schwerpunkt sind schon wichtige Kriterien um einige Eigenschaften der Geste der Hand bestimmen zu können. Um den Schwerpunkt zu bestimmen, wurde die folgende Formel benutzt:

$$x = \frac{m_{10}}{m_{00}}$$

$$y = \frac{m_{01}}{m_{00}}$$

OpenCV bietet noch weitere Möglichkeiten zur Auswertung des Umrisses an. Es kann ein Rechteck bestimmt werden, dass den Umriss vollständig enthält. Mit Hilfe des Mittelpunktes des Rechtecks und dem Schwerpunkt kann auf sehr einfache Art die Orientierung der Hand bestimmt werden, denn der aus den Momenten bestimmbare Winkel alleine reicht nicht aus. Die gegenseitige Lage dieser beiden Punkte ist entscheidend, in welche Richtung die Hand orientiert ist.

Eine weitere Eigenschaft, welche mit Hilfe dieses Rechtecks bestimmt werden kann, ist der Abstand zwischen dem Mittelpunkt des Rechtecks und dem Schwerpunkt. Diese Distanz lässt sich mit folgender Formel ausdrücken.

$$d = \sqrt{(x_s - x_r)^2 + (y_s - y_r)^2}$$

In einem nächsten Schritt wird die Länge des Umrisses bestimmt. Diese wird dazu verwendet um die Kompaktheit der Geste zu bestimmen. Die Kompaktheit stützt sich auf die Annahme, dass ein Kreis die kompakteste aller Formen darstellt. Diese Behauptung führt zu folgender Formel:

$$comp = \frac{4 \cdot \pi \cdot a}{l^2}$$

Diese Formel bedeutet, dass die Fläche mit vier π multipliziert und das Resultat geteilt durch den Umfang im Quadrat die Kompaktheit ergibt. Setzt man die Formeln für den Umfang und die Fläche des Kreises hier in diese Formel ein, erhält man die Kompaktheit gleich eins.

Eine weitere wichtige Charakteristik der Hand ist die Exzentrizität. Diese Eigenschaft gibt an, ob aus einer Form eine Richtung gelesen werden kann. Hat die Exzentrizität einen tiefen Wert (unter 0.1) ist es kaum möglich eine Orientierung der Form zu bestimmen. Um die Exzentrizität berechnen zu können, müssen zuerst die Werte der zentralen Momente berechnet werden. Dies wird mit folgenden Formeln gemacht:

$$\mu_{11} = \frac{m_{11}}{m_{00}} - \left(\frac{m_{10}}{m_{00}} \right) \cdot \left(\frac{m_{01}}{m_{00}} \right)$$

$$\mu_{20} = \frac{m_{20}}{m_{00}} - \left(\frac{m_{10}}{m_{00}} \right)^2$$

$$\mu_{02} = \frac{m_{02}}{m_{00}} - \left(\frac{m_{01}}{m_{00}} \right)^2$$

Mit Hilfe dieser Werte kann nun die Exzentrizität anhand der folgenden Formel bestimmt werden.

$$ecc = \frac{(\mu_{20} - \mu_{02})^2 - 4 \cdot \mu_{11}^2}{(\mu_{20} + \mu_{02})^2}$$

Um die Interpretation dieses Wertes zu vereinfachen, wird immer der Absolutwert gespeichert.

Eine weitere Eigenschaft ist der Neigungswinkel, welcher sich aus den zentralen Momenten berechnen lässt. Die folgende Tabelle gibt Auskunft zur Berechnung des Winkels.

mu₁₁	mu₂₀-mu₀₂	Winkel θ
0	0	0
>0	0	45
<0	0	-45
0	>0	0
0	<0	-90
>0	>0	$\frac{1}{2} \arctan\left(\frac{2 \cdot \mu_{11}}{\mu_{20} - \mu_{02}}\right) - 90^\circ$
<0	>0	$\frac{1}{2} \arctan\left(\frac{2 \cdot \mu_{11}}{\mu_{20} - \mu_{02}}\right) + 90^\circ$
>0	<0	$\frac{1}{2} \arctan\left(\frac{2 \cdot \mu_{11}}{\mu_{20} - \mu_{02}}\right)$
<0	<0	$\frac{1}{2} \arctan\left(\frac{2 \cdot \mu_{11}}{\mu_{20} - \mu_{02}}\right)$

Die Berechnungen für die obigen Winkel werden mit Hilfe der zentralen Momente von OpenCV (aus der CvMoments-Struktur) ausgeführt und nicht mit den zentralen Momenten nach den weiter oben erklärten Formeln für die zentralen Momente.

Diese Daten werden für jedes aufgenommene Bild mit einer Hand oder zwei Händen erhoben und in entsprechenden Objekten zwischengespeichert. Diese Daten werden für die Gestenerkennung weiter genutzt und ausgewertet. Dies ist das Thema des nächsten Kapitels.

11.4.4 Gestenerkennung

Die Gestenerkennung wird für jedes aufgenommene Bild und für jede Hand einzeln ausgewertet. In diesem Abschnitt geht es darum zu erkennen, welche Geste die Hand ausführt. Da ein einzelnes Bild analysiert wird, ist es auch nicht möglich Bewegungen zu erkennen. Die Gestenerkennung wird folgendermassen ausgeführt.

Bei der Gestenerkennung werden die folgenden vier Handformen erkannt:

- Faust
- Hand mit allen Fingern gespreizt
- Hand mit 2 Fingern gestreckt
- Hand mit allen Fingern zusammen ausgestreckt

Mit Hilfe der Kompaktheit kann man direkt die Handform „Hand mit allen Fingern gespreizt“ erkennen. Dies ist dadurch möglich, da der Wert dieser Eigenschaft dieser Handform deutlich unterhalb der Werte der anderen Handformen liegt. Bei den durchgeführten Tests lag dieser Wert unterhalb von 0.25.

Um die Handform „Faust“ von den übrigen zwei Handformen zu unterscheiden werden zwei Eigenschaften benutzt. Es wird die Exzentrizität und die Kompaktheit verwendet. Die Exzentrizität wird benutzt, da die Faust kaum eine Orientierung bestimmt. Die Kompaktheit wird gebraucht, da in einzelnen Fällen die Handform „Hand mit 2 Fingern gestreckt“ ähnliche Werte der Exzentrizität aufweisen kann wie die „Faust“. Da jedoch die Kompaktheit dieser beiden Handformen gut zu unterscheiden ist, wird diese Eigenschaft zusätzlich verwendet. Die Handform „Faust“ weist Werte der Exzentrizität auf die unter 0.1 liegen und die Kompaktheit liegt oberhalb von 0.55. Diese Werte ergaben sich bei zahlreichen Tests.

Um die Handformen „Hand mit 2 Fingern gestreckt“ und „Hand mit allen Fingern zusammen gestreckt“ zu unterscheiden, wird die Distanz zwischen dem Schwerpunkt und dem Mittelpunkt des handumgebenden Rechtecks herangezogen. Diese Distanz liefert ein genügend klares Kriterium. Bei der Handform „Hand mit 2 Fingern“ liegt diese Distanz oberhalb von 12.4 Pixel.

Da für einige Gesten die Orientierung auch eine Rolle spielt, wird diese hier auch berechnet. Dies geschieht mit Hilfe des Winkels und der Position des Schwerpunkts und des Mittelpunkts des handumgebenden Rechtecks.

Um der Hand eine Orientierung zuzuordnen zu können, muss der Winkel der Hand in gewissen Bereichen liegen. Ein erster Bereich ist, wenn der Winkel zwischen 0° und 20° liegt. Wenn dies der Fall ist, sind die Fingerspitzen nach oben oder nach unten orientiert. Um diese zwei Fälle zu unterscheiden werden nun der Rechteckmittelpunkt und Schwerpunkt zu Rate gezogen. Wenn der Rechteckmittelpunkt oberhalb von dem Schwerpunkt liegt, sind die Fingerspitzen nach oben gerichtet. Wenn der Rechteckmittelpunkt unterhalb des Schwerpunkts liegt, sind die Fingerspitzen nach unten gerichtet.

Wenn der Winkel zwischen 70° und 90° liegt, sind die Fingerspitzen nach links oder rechts gerichtet. Falls der Rechteckmittelpunkt weiter links liegt als der Schwerpunkt, sind die Fingerspitzen nach links gerichtet. Im anderen Fall, wenn der Rechteckmittelpunkt weiter rechts liegt als der Schwerpunkt, sind die Fingerspitzen nach rechts gerichtet.

Für den Fall, dass zwei Hände gefunden wurden, werden noch zusätzliche Eigenschaften der Gesten mit zwei Händen berechnet. Für diese Berechnungen werden die Schwerpunkte beider Hände gebraucht. Die berechneten Eigenschaften sind die Distanz und der Winkel zwischen den Händen. Es werden die folgenden Formeln benutzt.

$$d = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$
$$\alpha = \arctan\left(\frac{y_1 - y_2}{x_1 - x_2}\right)$$

Die gewonnenen Erkenntnisse dieser Analyse werden in einem Objekt gespeichert. Dieses Objekt wird in eine Liste abgelegt. Der Grund hierfür liegt darin, dass die Analyse der Gesten später noch weiter geführt wird, sobald sich mehrere solcher Resultatobjekte in der Liste befinden. Ein Vorteil dieser Technik ist die Filterung der Gesten und somit der Stabilisierung des Programms.

Die Gesten werden immer erst analysiert, wenn eine gewisse Anzahl vorhanden ist. Dies ermöglicht auch Bewegungen zu entdecken und analysieren zu können. Das nachfolgende Kapitel beschäftigt sich mit der Bewegungserkennung und der schlussendlichen Gestenerkennung.

11.4.5 Bewegungserkennung

Dieses Kapitel beschäftigt sich mit der Erkennung und Analyse von Bewegungen. Um dies tun zu können, wird nicht mehr jedes Bild einzeln ausgewertet. Sondern eine Bildsequenz.

Der Vorteil der Analyse einer Bildsequenz wurde oben schon genannt. Es wird nämlich das Resultat der Gesten- und Bewegungserkennung stabilisiert. Dies geschieht dadurch, dass nur die Geste übermittelt wird, welche am meisten in der analysierten Bildsequenz auftritt. Zuerst wird der Vorgang für die Aufnahme von Bildern, respektive den Bildauswertungen, in die Bildsequenzen bestimmt.

Es werden nicht Bildsequenzen analysiert, die strikte nacheinander auftreten. Auch die Übergänge zwischen diesen Sequenzen sind wichtig und werden für die Analyse mitbezogen. Dies geschieht dadurch, dass die Liste nicht komplett entleert wird, sobald ein Analysedurchgang getätigt wurde. Es werden die letzten Bildanalyseresultate nicht gelöscht und in der Liste belassen, welche nachher wieder aufgefüllt wird.

Eine Sequenzanalyse ist eher statistischer Natur. Es wird gezählt, welche Geste wie oft vorkommt. Zusätzlich werden noch andere Zahlen erhoben um die Sequenzanalyse stabil zu halten. Es werden die folgenden Zahlen erhoben:

- Anzahl von „Faust“,
- Anzahl von „Hand mit gespreizten Fingern“,
- Anzahl von „Hand mit zusammengelegten und gestreckten Fingern“,
- Anzahl von „Hand mit zwei Fingern gestreckt“,

- Anzahl von Gesten mit nur einer Hand,
- Anzahl von Gesten mit zwei Händen,
- Anzahl von „Zwei Fäuste“,
- Anzahl von Bildern ohne Hände

Diese Zahlen werden erhoben in dem die Liste der Bildanalyseresultate durchlaufen wird und die entsprechenden Variablen inkrementiert werden.

In einem nächsten Schritt werden die erhobenen Zahlen untereinander verglichen und bestimmt um welche Geste es sich handelt. Das Resultat wird in einem Objekt gespeichert, welches im Demonstrationsprogramm als Event weitergebraucht wird.

In einem ersten Schritt wird kontrolliert wie oft Gesten mit einer Hand oder mit zwei Händen ausgeführt wurden. Zusätzlich wird kontrolliert, dass sich nicht zu viele Bilder ohne Hände unter den Resultaten befinden. Wenn genügend Bildanalyseresultate für Gesten mit einer Hand gefunden wurden, wird anhand der Zählresultate nachgeschaut, um welche Geste es sich handeln kann. Als Gesten, welche mit einer Hand ausgeführt werden, werden folgende identifiziert:

- Faust still
- Faust in horizontaler und / oder vertikaler Richtung bewegt
- Hand mit allen Fingern gespreizt still
- Hand mit allen Fingern gespreizt in horizontaler und / oder vertikaler Richtung bewegt
- Hand mit allen Fingern zusammengelegt gestreckt still
- Hand mit zwei Fingern gestreckt still

Um herauszufinden, um welche der vier Handformen es sich handelt, werden die Zählresultate verglichen. Die am meisten vorkommende Handform wird weiter ausgewertet. Es wird kontrolliert, ob sich die Hand bewegt oder ob sie still ist. Um dies zu wissen, wird die Distanz zwischen dem Anfangspunkt und dem Endpunkt der Geste mit folgender Formel berechnet:

$$d = \sqrt{(x_E - x_A)^2 + (y_E - y_A)^2}$$

Da der Benutzer die Hand kaum perfekt still halten kann, müssen sich der Anfangs- und Endpunkt für die Gesten mit Halten der Handposition nicht perfekt am gleichen Ort befinden. Eine Bewegung wird erst dann wahrgenommen, wenn die berechnete Distanz grösser als 50 Pixel wird. Falls sich die Hand bewegt, wird dem Event eine Bewegungsdistanz für die horizontale und die vertikale Achse übermittelt.

Wenn sich unter den Bildanalyseresultaten mehr Gesten mit zwei Händen finden als solche mit nur einer Hand, wird eine Analyse durchgeführt um herauszufinden, welche Geste mit zwei Händen ausgeführt wird. Die einzige unterstützte Geste ist, wenn beide Hände eine Faust bilden. Für den Fall, dass sich zwei Fäuste in der Sequenz bewegen, wird festgestellt, wie sich die Distanz zwischen den Faustschwerpunkten ändert und wie sich der Winkel zwischen diesen Schwerpunkten verhält. Diese zwei Eigenschaften sind wichtig, um bestimmen zu können wie gross der Zoomfaktor und der Drehwinkel sind.

Der Zoomfaktor wird anhand der Distanz am Anfang und am Ende der Sequenz bestimmt. Die folgende Formel dient dazu.

$$zoom = \frac{d_E}{d_A}$$

Der Drehwinkel wird auch mit dem Winkel am Anfang und am Ende der Sequenz bestimmt und zwar mit der folgenden Formel.

$$\alpha = \alpha_E - \alpha_A$$

11.5 Erklärungen zum Programm der Bilder- und Videogalerie

In diesem Kapitel werden die Zustands- und Unterzustandsmaschinen erklärt, welche benötigt werden um das Demonstrationsprogramm laufen zu lassen.

11.5.1 Hauptzustandsmaschine

Die folgende Abbildung zeigt die Hauptzustandsmaschine.

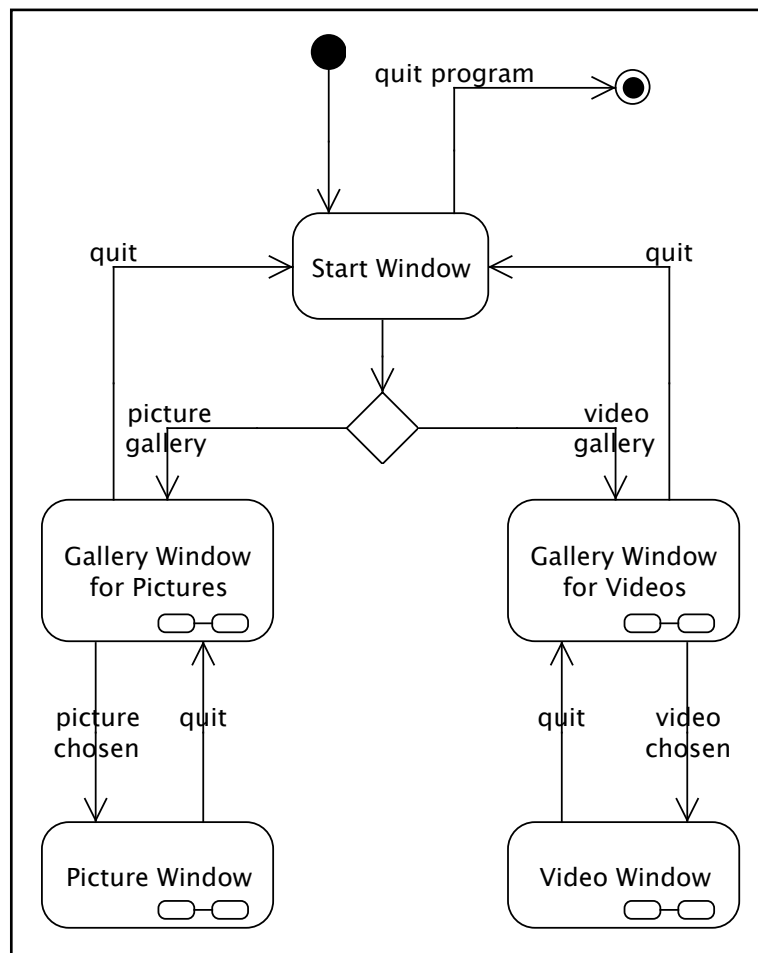


Abbildung 51: Hauptzustandsmaschine

Mit den Events, welche auf den Transitionen definiert sind, kann zwischen den Zuständen gewechselt werden. Die zu den Events passenden Gesten wurden in Anhang 11.3 definiert.

11.5.2 Bildergalerie

Die nachfolgende Abbildung zeigt die Unterzustandsmaschine der Bildergalerie.

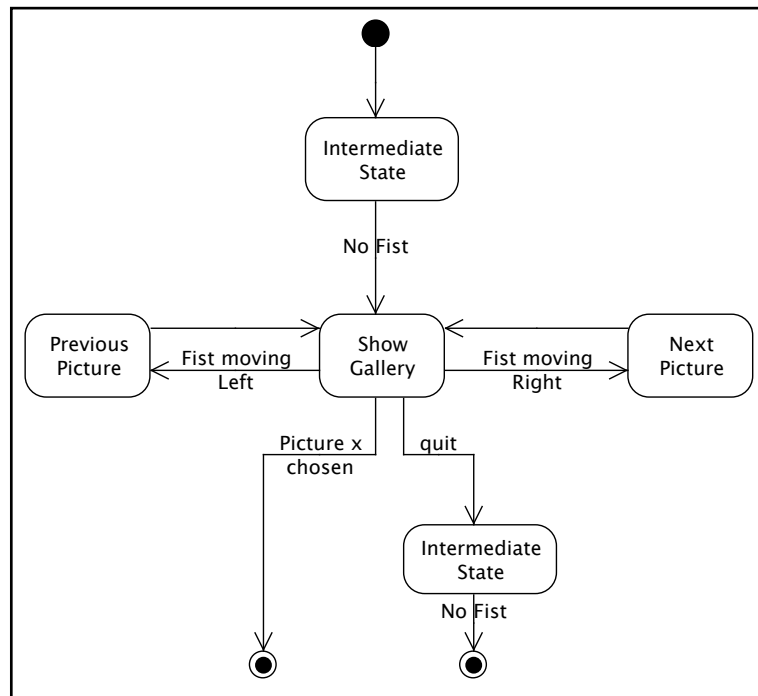


Abbildung 52: Unterzustandsmaschine Bildergalerie

Wie man sieht gibt es eingangs der Unterzustandsmaschine einen Zwischenzustand, hier wird das Hauptfenster ausgeblendet und die Bildergalerie eingeblendet. Das Gegenteil passiert im Zwischenzustand ausgangs der Unterzustandsmaschine.

Es werden zum Wechseln zwischen den Fotos zwei Zustände eingefügt, die die entsprechenden Events behandeln. Um wieder zum Zustand „Show Gallery“ zurückkommen zu können, muss die Hand eine Geste ausführen, die nicht die gleiche ist, wie um in den Zustand zu gelangen.

11.5.3 Videogalerie

Die nachstehende Abbildung zeigt die Unterzustandsmaschine der Videogalerie.

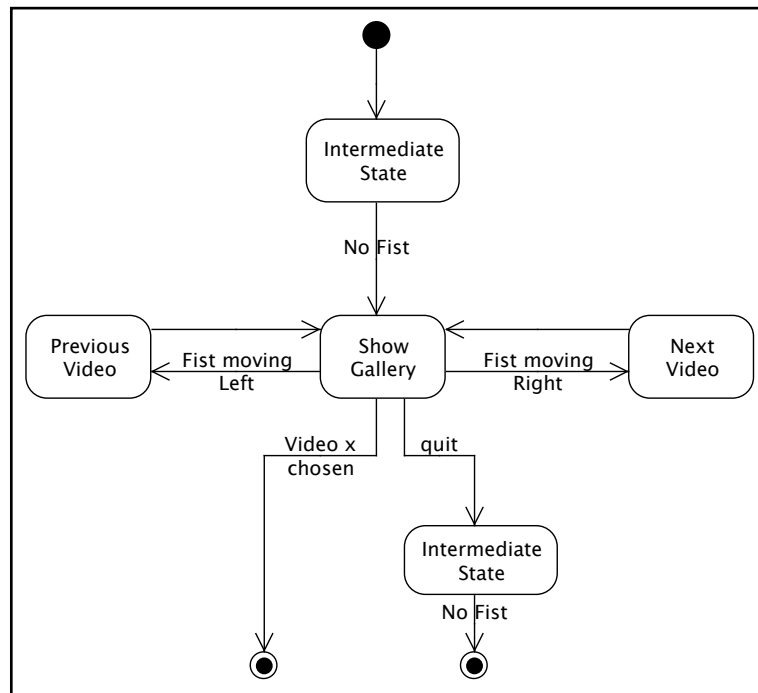


Abbildung 53: Unterzustandsmaschine der Videogalerie

Die beiden Zwischenzustände haben in dieser Unterzustandsmaschine die genau gleiche Funktion wie die Zwischenzustände in der Zustandsmaschine in Kapitel 11.5.2.

Um zwischen den Thumbnails zu wechseln werden zwei Zustände eingefügt. Wenn die entsprechende Geste ausgeführt wird, wird der zugehörige Zustand aufgerufen. . Um aus den Zuständen zum Scrollen herauszukommen, darf die Hand nicht mehr die Geste ausführen, welche gebraucht wird um in den Zustand zu kommen.

11.5.4 Bildanzeige

Die folgende Abbildung zeigt die Unterzustandsmaschine für die Bildanzeige.

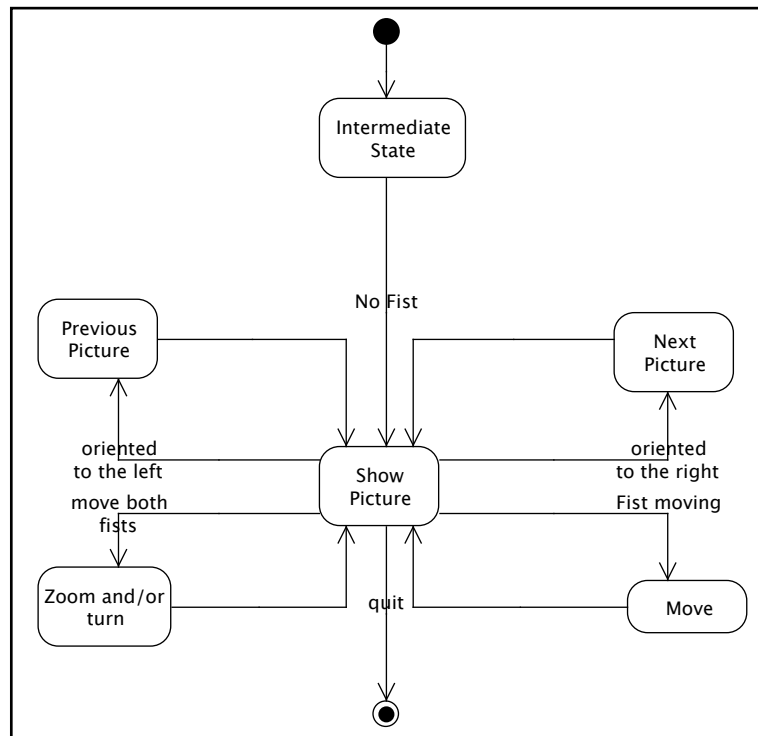


Abbildung 54: Unterzustandsmaschine für die Bildanzeige

Wie man sieht, erhält jede Funktionalität ihren eigenen Zustand. Um in den Zustand hineinzugelangen muss die entsprechende Geste ausgeführt werden. Um den Zustand zu verlassen, muss eine andere Geste ausgeführt werden, als die, die gebraucht wird um in den Zustand hineinzugelangen.

Eingangs der Unterzustandsmaschine ist ein Zwischenzustand. Dieser wird gebraucht um das Galeriefenster auszublenden und das Bildanzeigefenster einzublenden.

11.5.5 Videoanzeige

Die nachstehende Abbildung zeigt die Unterzustandsmaschine für die Videoanzeige.

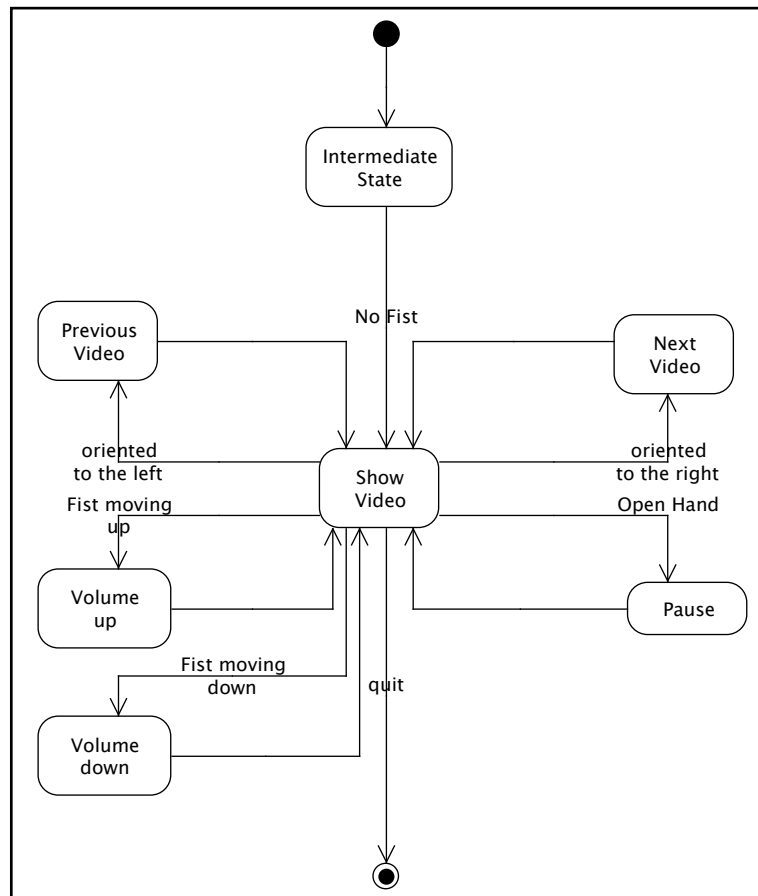


Abbildung 55: Unterzustandsmaschine für die Videoanzeige

Wie man sieht, erhält jede Funktionalität ihren eigenen Zustand. Um in den Zustand hineinzugelangen muss die entsprechende Geste ausgeführt werden. Um den Zustand zu verlassen muss eine andere Geste ausgeführt werden, als die, die gebraucht wird um in den Zustand hineinzugelangen.

Eingangs der Unterzustandsmaschine ist ein Zwischenzustand. Dieser wird gebraucht um das Galeriefenster auszublenden und das Videoanzeigefenster einzublenden.

11.6 Software und elektronische Dokumentation

Die hier angehängte CD enthält die elektronische Dokumentation und das Demonstrationsprogramm.