

Institut für Prozessrechentchnik,  
Automation und Robotik

Prof. Dr.-Ing. R. Dillmann  
Prof. Dr.-Ing. J. Wittenburg  
Universität Karlsruhe

# **Bildbasierte Erkennung statischer Gesten**

Diplomarbeit  
von

**Elena Gimeno Churruca**

Mai 2001

Hauptreferent: Prof. Dr.-Ing. R. Dillmann  
Korreferent: Prof. Dr.-Ing. J. Wittenburg  
Betreuer: Dipl.-Inform. M. Ehrenmann

# Inhaltsverzeichnis

<b>Inhaltsverzeichnis</b>	<b>1</b>
<b>Abbildungsverzeichnis</b>	<b>4</b>
<b>1 Einleitung</b>	<b>5</b>
1.1 Menschen und Roboter . . . . .	5
1.2 Projektrahmen . . . . .	6
1.2.1 Technische Ausstattung . . . . .	6
1.2.2 Ziele der Arbeit . . . . .	7
1.3 Einordnung der Arbeit in den Prozeßder gestenbasierten Steuerung von Robotern . . . . .	8
1.3.1 Vorverarbeitung . . . . .	10
1.3.2 Merkmalsextraktion . . . . .	12
1.3.3 Klassifikation . . . . .	13
1.4 Aufbau der Arbeit . . . . .	13
<b>2 Stand der Technik</b>	<b>15</b>
2.1 Verschiedene Methoden zum Bildbasierte Objekt Erkennung . . . . .	15
2.1.1 A Gesture Interface for Human-Robot-Interaction . . . . .	15
2.1.2 Einfache Handgestikerkennung mit einem zweistufigen Nearest- neighbour Klassifikator . . . . .	16
2.1.3 Objekterkennung basierte auf der Silhouette. . . . .	17
2.1.4 Klassifikation von unregelmäßige gestaltet Mikro-Objekte durch Verwenden Komplex Fourier Deskriptoren . . . . .	17
2.2 Auserwählte Methode. Begründung . . . . .	18
<b>3 Theoretische Grundlagen</b>	<b>19</b>
3.1 Mathematische Analyse mit Hilfe von Fourier Deskriptoren . . . . .	19
3.1.1 Fourieranalyse kontinuierlicher Signale . . . . .	20
3.1.2 Fourieranalyse diskreter Signale . . . . .	20
3.2 Anwendung in der digitalen Bildverarbeitung . . . . .	23
3.2.1 Berechnung der Fourierdeskriptoren . . . . .	23
3.2.2 Eigenschaften der DFT . . . . .	25

<b>4</b>	<b>Ansätze</b>	<b>30</b>
4.1	Vorverarbeitung . . . . .	30
4.1.1	Vorläufiger Algorithmus zur Konturextraktion . . . . .	30
4.1.2	Hautfarbsegmentierung . . . . .	32
4.2	Merkmalsextraktion . . . . .	33
4.2.1	Resampling der Konturkette . . . . .	33
4.2.1.1	Ziel des Resampling . . . . .	34
4.2.1.2	Resampling Methode . . . . .	34
4.2.1.3	Anzahl der abzutasteten Punkte . . . . .	36
4.2.2	Anwendung der Fourier-Transformation . . . . .	41
4.2.2.1	Vornormierung der Konturkette . . . . .	42
4.2.2.2	Berechnung der Deskriptoren . . . . .	44
4.2.2.3	Normierung der Deskriptoren . . . . .	45
4.3	Vergleichs- und Klassifikationsalgorithmus . . . . .	46
4.3.1	Vergleichskriterium . . . . .	47
4.3.2	Entscheidungskriterium . . . . .	48
4.3.3	Modellaufbau . . . . .	49
<b>5</b>	<b>Experimente und Evaluation</b>	<b>51</b>
5.1	Evaluation des Systems als selbständige Anwendung . . . . .	53
5.1.1	Evaluation des Modellaufbaus . . . . .	54
5.1.2	Evaluation der Klassifikationsleistung . . . . .	57
5.2	Evaluation der Implementierung . . . . .	59
<b>6</b>	<b>Zusammenfassung und Ausblick</b>	<b>61</b>
6.1	Erweiterung . . . . .	61
<b>A</b>	<b>Bibliothek Schnittschelle</b>	<b>62</b>
A.1	Notwendigkeit eine angemessene Bibliothek . . . . .	63
A.2	Ablauf der Schnittstelle . . . . .	63
A.2.1	Verarbeitung des Eingabebilds . . . . .	63
A.2.2	Bearbeitung der Bibliothek . . . . .	66
A.2.3	Klassifikation einer Geste . . . . .	69
	<b>Literaturverzeichnis</b>	<b>71</b>

# Abbildungsverzeichnis

1.1	Der Roboter	7
1.2	Der Kamerakopf	7
1.3	Ablauf der Interaktion	8
1.4	Schritte der Bildverarbeitung	10
1.5	Vorverarbeitung des Signales	11
1.6	Merkmalsextraktion	12
3.1	Beispielkontur Dreieck	24
3.2	Gedrehtes Dreieck	26
3.3	Verschobenes Dreieck	27
3.4	Skaliertes Dreieck	28
3.5	Originales Dreieck	29
3.6	Anderer Anfangspunkt	29
4.1	Grauwertbild mit Benutzerhand vor dunklem Hintergrund	31
4.2	Binarisiertesbild der Benutzerhand	31
4.3	Hautfarbsegmentierung	32
4.4	Resampling der Konturkette von Gesten "Stop Open" durch größten Krümmung Kriterium	35
4.5	Beispiel einer Kontur der durch die Resampling geglättet wird.	35
4.6	Resampling der Konturkette von Gesten: "Victory A".	36
4.7	Resampling der Konturkette von Gesten: "Victory B".	36
4.8	Resampling der Konturkette von Gesten "Stop Open" mit 8 Punkte und unterschiedliche Anfangspunkt.	37
4.9	Ergebnisse der Gesten Erkennung von Gesten "Stop Open" mit 8 Punkte. Links mit zufällig gleichen Anfangspunkt und rechts mit unterschiedlichen Anfangspunkt.	38
4.10	Resampling der Konturkette von Gesten "Stop Open" mit 16 Punkte und unterschiedliche Anfangspunkt.	38
4.11	Ergebnisse der Gesten Erkennung von Gesten "Stop Open" mit 16 Punkte und unterschiedliche Anfangspunkt.	39
4.12	Resampling der Konturkette von Gesten "Stop Open" mit 32 Punkte und unterschiedliche Anfangspunkt.	39
4.13	Ergebnisse der Gesten Erkennung von Gesten "Stop Open" mit 32 Punkte und unterschiedliche Anfangspunkt.	40

4.14	Links: Originalkontur; Rechts: Resampling der Konturkette von Gesten "Stop Open" mit 64 Punkte. . . . .	40
4.15	Ergebnisse der Gesten Erkennung von Gesten "Stop Open" mit 64 Punkte und unterschiedliche Anfangspunkt. . . . .	41
4.16	Normierung bezüglich Verschiebung der Geste "Victory A" . . . . .	43
4.17	Normierung bezüglich Verschiebung der Geste "Victory B" . . . . .	43
4.18	Normierung bezüglich Skalierung der Geste "Victory A" . . . . .	44
4.19	Normierung bezüglich Skalierung der Geste "Victory B" . . . . .	44
4.20	Fourier-Deskriptoren von verschiedenen Händen mit der gleichen Geste. . . . .	45
4.21	Abstände von Fourier-Deskriptoren von verschiedenen Hände bei der gleichen Geste. . . . .	46
4.22	Modulen von Fourier-Deskriptoren vom Gestentyp "Stop Open" (links) und "Zeige" (rechts). . . . .	47
4.23	Euklidische Distanz zwischen den Deskriptoren der vorgeführten Gesten und den Modelldeskriptoren. . . . .	48
4.24	Euklidische Distanz für in der Bibliothek nicht enthaltene Gesten. . . . .	49
5.1	In der Experimentsbibliothek verwendete Gestentypen. . . . .	52
5.2	Cambiar las fotos por un futuro diagrama de flujo . . . . .	53
5.3	Euklidische Distanzen mit unterschiedlichem Modellaufbau. . . . .	54
5.4	Euklidische Distanzen mit unterschiedlichem Modellaufbau. . . . .	55
5.5	Euklidische Distanzen mit unterschiedlichem Modellaufbau. . . . .	55
5.6	Euklidische Distanzen mit unterschiedlichem Modellaufbau. . . . .	56
5.7	Euklidische Distanzen mit unterschiedlichem Modellaufbau. . . . .	56
5.8	Euklidische Distanzen mit unterschiedlichem Modellaufbau. . . . .	57
5.9	Benutzte Geste um der Modelldeskriptor "griffe" aufzubauen. . . . .	59
A.1	Schnittstelle für die Verarbeitung der Bibliotheken. . . . .	62
A.2	Graphischer Anzeige des Verarbeitetes Gestes. . . . .	62
A.3	Grauwert Bild. . . . .	64
A.4	Falsch-binarisiertes Bild. . . . .	64
A.5	Richtig-binarisiertes Bild. . . . .	64
A.6	Kontur des falsch-binarisiertes Bild. . . . .	65
A.7	Kontur des richtig-binarisiertes Bild. . . . .	65
A.8	Diagramflußder Bildverarbeitung. . . . .	66
A.9	Diagramflußder Aufbau der Modelle. . . . .	68
A.10	Diagramflußder Klassifikations-Ablauf. . . . .	70

# Kapitel 1

## Einleitung

### 1.1 Menschen und Roboter

Der Titel dieses Kapitels nennt zwei für die vorliegende Arbeit grundlegende Begriffe: “Mensch” und “Roboter”. Während der Begriff “Menschen” klar und einfach zu verstehen ist, wird im Folgenden den Begriff Roboter näher erläutern. Das Wort “Robot” kommt vom Tschechischen “robota”, und die direkte Übersetzung bedeutet “Zwangsarbeit”. Roboter werden heutzutage für viele verschiedene Aktivitäten benutzt. Sie führen Aufgaben durch, die für Menschen gefährlich oder unangenehm sind. Sie ermöglichen die Realisierung von Transaktionen in lebensfeindlichen Umgebungen. Es gibt eine große Anzahl von Definitionen für das Wort “Roboter”, und durch die umfangreiche Forschung in diesem Bereich wird das Anwendungsfeld ständig vergrößert. Damit ändern sich auch ständig die Definitionen. Eine allgemeine Definition ist in der “New Encyclopaedia Britannica” ([1]) zu finden, sie beschreibt einen Roboter als eine automatisch gesteuerte Maschine, die menschliche Arbeit ersetzt. Eine mögliche Klassifikation für Roboter lautet wie folgt:

**Industrie-Roboter.** Diese werden von allen Robotertypen schon am längsten benutzt. Um eine gemeinsame Definition zur Verfügung zu stellen ist der Begriff Industrieroboter nach ISO ([2]) genormt und lautet ”automatisch gesteuertes, wiederprogrammierbares, vielfach einsetzbares Handhabungsgerät mit mehreren Freiheitsgraden, das entweder ortsfest oder beweglich in automatisierten Fertigungssystemen eingesetzt werden kann”.

**Service-Roboter.** Manchmal auch Assistenten genannt. Sie laufen in unstrukturierten Umgebungen und haben eine hohe Flexibilität sowie die Fähigkeit nicht-repetitive Aktivitäten durchzuführen.

**Personal-Roboter.** Diese sind am wenigsten unter diesen drei Robotertypen entwickelt worden. Sie werden manchmal auch Androide genannt, weil sie in Funktionalität und Aussehen dem Menschen ähnlich sind.

Die in dieser Arbeit betrachteten Roboter gehören zum dritten Typ. Viele Aktivitäten werden bereits durch diese Roboter ausgeführt, aber trotzdem sollen sie so

erweitert und optimiert werden, daß sie in das menschliche Alltagsleben integriert werden können. Um dieses Ziel zu erreichen spielt die Kommunikation zwischen Mensch und Roboter eine wesentliche Rolle. Die entsprechende Kommunikation soll so ähnlich wie möglich gestaltet werden wie die Kommunikation zwischen Menschen selbst, weil damit die Anwendung durch nicht spezialisierte Benutzer erleichtert wird.

## 1.2 Projektrahmen

Diese Diplomarbeit findet statt im Rahmen des Leitprojektes MORPHA ([]) des BMBF (Bundesministerium für Bildung und Forschung). Der Schwerpunkt des Projektes MORPHA ist Forschung im Bereich Mensch-Maschine-Interaktion. Mechatronische Systeme sollen mit dem Benutzer kooperieren und ihm assistieren, unter Steuerung durch den Menschen.

Drei wichtige Kanäle in der Interaktion zwischen Mensch und Maschine sind Sprache, Sehen und Fühlen. Das Fühlen soll in Zukunft benutzt werden, damit die Maschine Kräfte bemerkt und auf sie reagieren kann. Sprache und visuelle Kommunikation sind bereits ein wesentlicher Punkt des MORPHA-Projektes. Ein Beispiel für die visuelle Kommunikation stellt die Bildbasierte Gestenerkennung dar. Diese Art Interaktion ermöglicht Kontrolle und Steuerung des Systems sowie Belehrung und Programmieren von Assistenzsystemen.

Die Unternehmen, die am MORPHA-Projekt teilnehmen, legen besonderen Wert auf die Umsetzung der Ergebnisse in industrielle Produkte. Diese neuen Produkte umfassen Programmier- und Bedienkonsolen, Personenerkennungs- und intelligente Raumüberwachungssysteme ebenso wie intelligente Steuerungen für unterschiedlichste mechatronische Systeme. Ein langfristiges Ziel des Vorhabens ist, den Weg für neuartige Assistenzsysteme in der Produktion und im Bereich Haushalt und Pflege vorzubereiten. Beispiele hierfür sind Handhabungs-, Montage-, Mobilitäts- und Reinigungshilfen oder mechatronische Assistenten im Bereich Rehabilitation.

### 1.2.1 Technische Ausstattung

Am Institut IPR der Fakultät für Informatik der Universität Karlsruhe wird ein intelligenter Assistent aufgebaut (vgl. Abb. 1.1). Dieser Roboter besteht aus den folgenden Komponenten:

- Mobile Plattform ODETE (Eigenentwicklung des Instituts, autonomes Fahrzeug mit differenziellem Antrieb, SICK-Laserscanner und Ultraschallsensoren zur Hinderniserkennung)
- Rumpf zur Aufnahme von Rechnermodulen

- Siebenachsiger anthropomorpher Arm der Firma AMTEC mit Kraft-Momenten-Sensor der DLR und Dreifingerhand der Firma Barrett Technologies
- Aktiver Kamerakopf mit parallel kalibrierten Sony XC 777AP Farbkameras (vgl. Abb. 1.2). Der Kamerakopf, mit dem in der Diplomarbeit gearbeitet werden soll, ist dreh- und kippbar und liefert über zwei Matrox Framegrabber ein Paar Farbbilder in der Auflösung 640x400 Pixel. Die Programmierung des bildverarbeitenden Systems erfolgt unter der Verwendung von Tcl/Tk und C. Dabei wird die Matrox Imaging Library (MIL) der Firma Matrox eingesetzt.

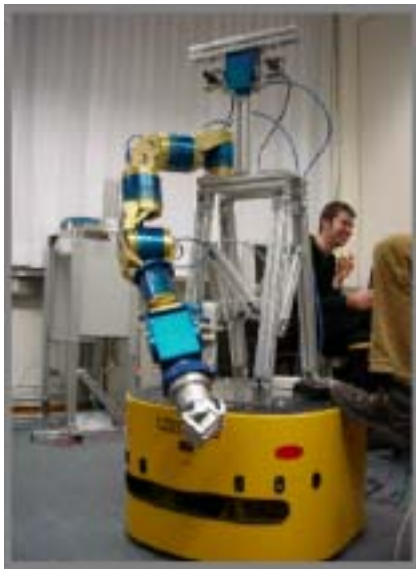


Abbildung 1.1: Der Roboter



Abbildung 1.2: Der Kamerakopf

### 1.2.2 Ziele der Arbeit

Die bildbasierte Erkennung statischer Gesten ist ein Projekt, daß der Arbeitslinie am IPR folgt. Am Institut wurden bereits folgende Vorarbeiten geleistet im Bereich Gestenerkennung:

- Algorithmen zur Hautfarbsegmentierung identifizieren in Bildfolgen den Kopf und die Hände eines Benutzers. Die Hautfarbsegmentierung läuft bei Beleuchtung durch Kunstlicht sehr robust.
- Hidden-Markov-Modelle erkennen nach der Aufnahme einer Bewegung aufgrund der Trajektorie eine vorgegebene Menge dynamischer Gesten (z.B. Zick-Zack-Bewegungen, Kreise, Haken).
- Statische Gestenerkennung per Datenhandschuh.

Die Erkennung dynamischer Gesten soll nun erweitert werden um die Detektion und Klassifikation statischer Gesten. Die bildbasierte statische Gestenerkennung verlangt die folgenden Voraussetzungen:

**Benutzerunabhängigkeit.** Die Gesten sollen erkannt werden unabhängig von der Person, die die Geste vorführt. Die gleiche Geste kann geometrische Unterschiede enthalten, die die Klassifikation der Geste nicht verfälschen sollten.

**Erweiterungsfähigkeit.** Die Erweiterung der Menge der zu erkennenden Gesten muß möglich sein.

**Robustheit** gegen:

- Rauschen in der Vorverarbeitung.
- Variation der Gesten beim gleichen Benutzer.

**Echtzeitfähigkeit.** Da der Roboter vom Benutzer gesteuert werden soll, ist eine echtzeitfähige Verarbeitung notwendig.

**Kompaktheit.** Da das Programm auf einem Roboter implementiert wird, sollte es nicht viel Speicher belegen und wenig Rechenaufwand haben. Es handelt sich um die Steuerung eines mobilen Roboters, d.h. auf dem Rechner werden gleichzeitig mehrere Programme bei begrenztem Platz laufen.

### 1.3 Einordnung der Arbeit in den Prozeß der gestenbasierten Steuerung von Robotern

Das Ziel dieser Arbeit ist es die Kommunikation mit dem Roboter durch Gesten und dadurch eine gestenbasierte Steuerung des Roboters zu ermöglichen. Dafür wird ein Programm in dem Roboter implementiert. Der Ablauf der Steuerung durch die Gestenerkennung verläuft in drei Schritten:

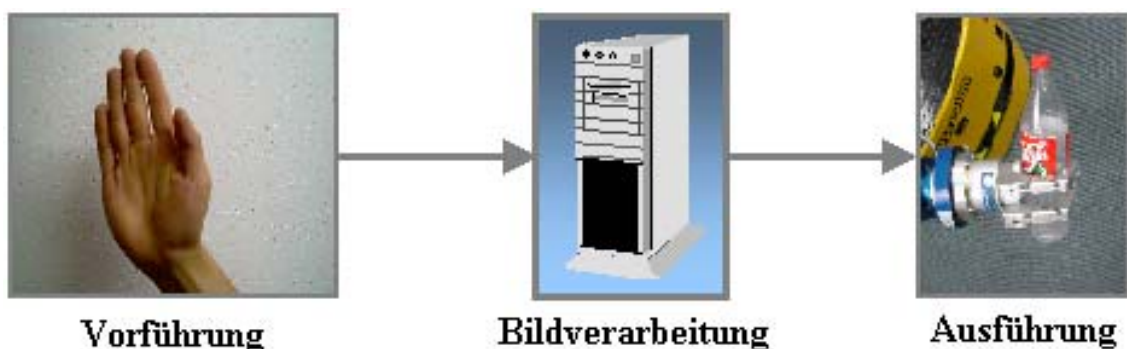


Abbildung 1.3: Ablauf der Interaktion

Eine Geste wird vor den Kameras vorgeführt. Anschließend werden die aufgenommenen Bilder verarbeitet und daraus eine Entscheidung abgeleitet. Je nach Art

der Entscheidung, Klassifizierung oder Nichterkennung der Geste, werden die entsprechende Aktionen ausgeführt (siehe Abb. 1.3).

Die vorliegende Arbeit untersucht einen Teil des Ablaufs der Bildverarbeitung um die erwähnte Entscheidung zu liefern. Die folgenden Ausführungen geben also vertiefende Informationen zur Bildverarbeitung, d.h. dem zweiten Schritt des Ablaufs der Robotersteuerung durch die Gestenerkennung (Abb. 1.3).

Der Prozeß der Bildverarbeitung läßt sich in drei Phasen unterteilen (vgl. Abb. 1.4). In der Vorverarbeitungsphase wird die Kontur der menschlichen Hand ermittelt und als eine Reihenfolge von Punkten ausgegeben. In der zweiten Phase werden die Merkmale des Objektes extrahiert. Als Algorithmus kommt dazu eine Fourier-Transformation zum Einsatz, die auf die Konturenkette angewendet wird. Aufgrund der Eigenschaften dieser Transformation ist die entnommene Information jetzt vergleichbar mit Musterdeskriptoren. Dies ermöglicht im dritten Schritt die Klassifikation, d.h. einen Vergleich des Deskriptors der vorgeführten Geste mit einer Referenzmenge. Anhand dieses Vergleichs kann eine Entscheidung getroffen werden. Es können zwei Fälle auftreten: entweder die Geste existiert nicht in der entsprechenden Bibliothek oder sie stimmt mit einem Element aus den Referenzmenge überein.

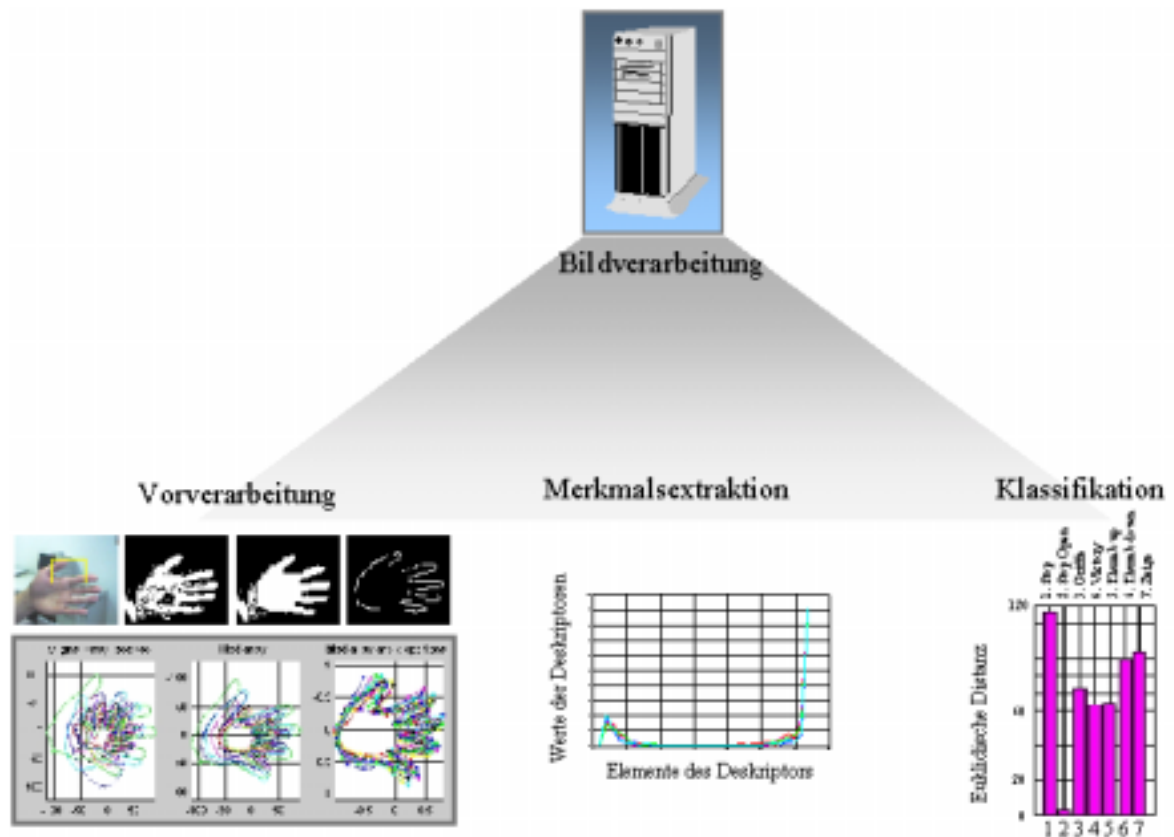


Abbildung 1.4: Schritte der Bildverarbeitung

### 1.3.1 Vorverarbeitung

Die Vorverarbeitung des Bildes hat die Kontur der Benutzerhand zum Resultat und benötigt dazu als Eingabe ein Farbbild. Dieser Schritt kann in drei kleinere aufgeteilt werden, wie in Abb. 1.5 dargestellt:

- **Hautfarbsegmentierung.** Hier werden alle hautfarbenen Pixel in weiß umgesetzt, alle anderen in schwarz.
- **Open-Close-Filtering.** Das erhaltene Bild wird mittels eines Open-Close-Algorithmus gefiltert, um die schwarzen Flecken innerhalb des Objektes und die weißen außerhalb zu entfernen.
- **Extraktion der Kontur.** Aus diesem Bild, das die Objekte als Blobs enthält, wird jetzt ein Bild der Kontur extrahiert.

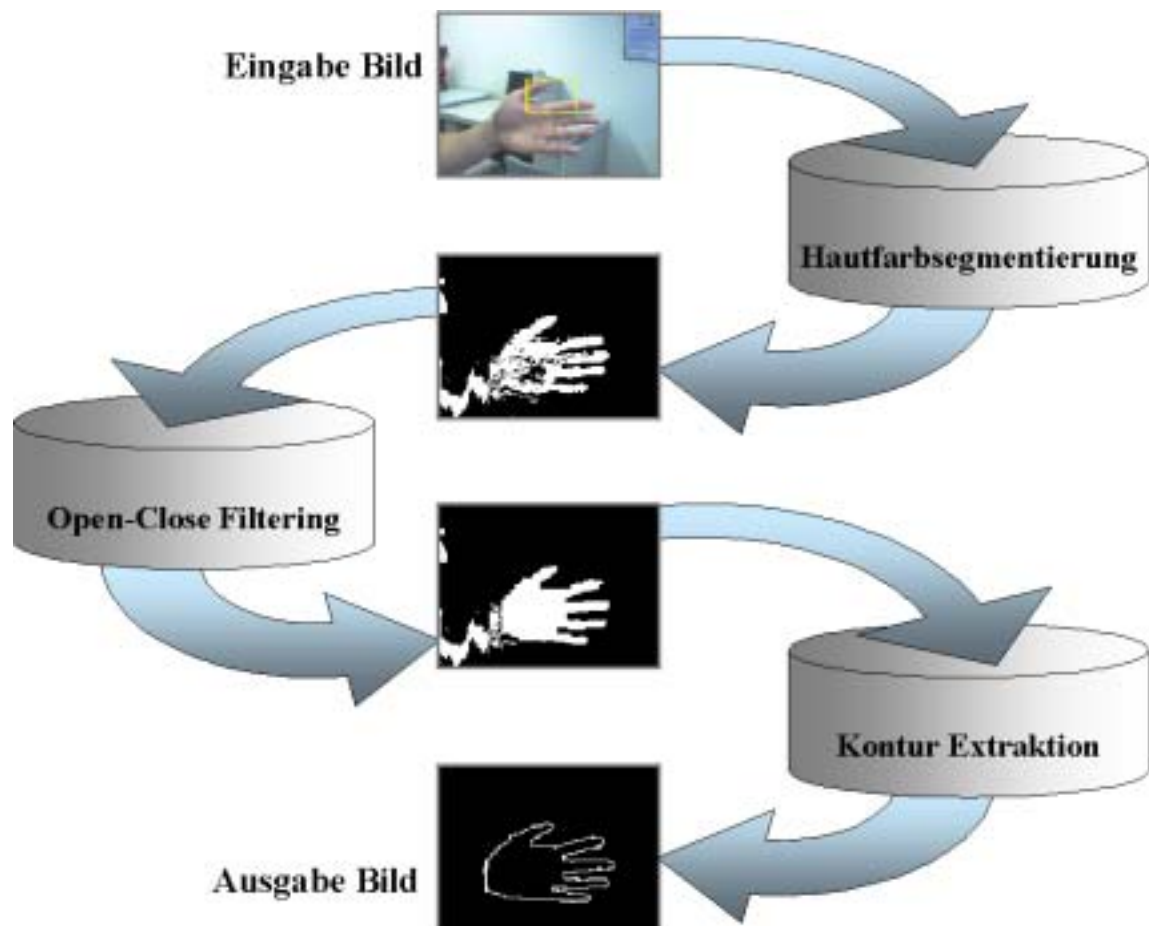


Abbildung 1.5: Vorverarbeitung des Signales

Die ersten beiden Schritte sind in [Ly 01] entwickelt worden. Der Dritte wurde im Rahmen dieser Arbeit mit der Hilfe der Bilverarbeitungsbibliothek Matrox Imaging Library gelöst.

### 1.3.2 Merkmalsextraktion

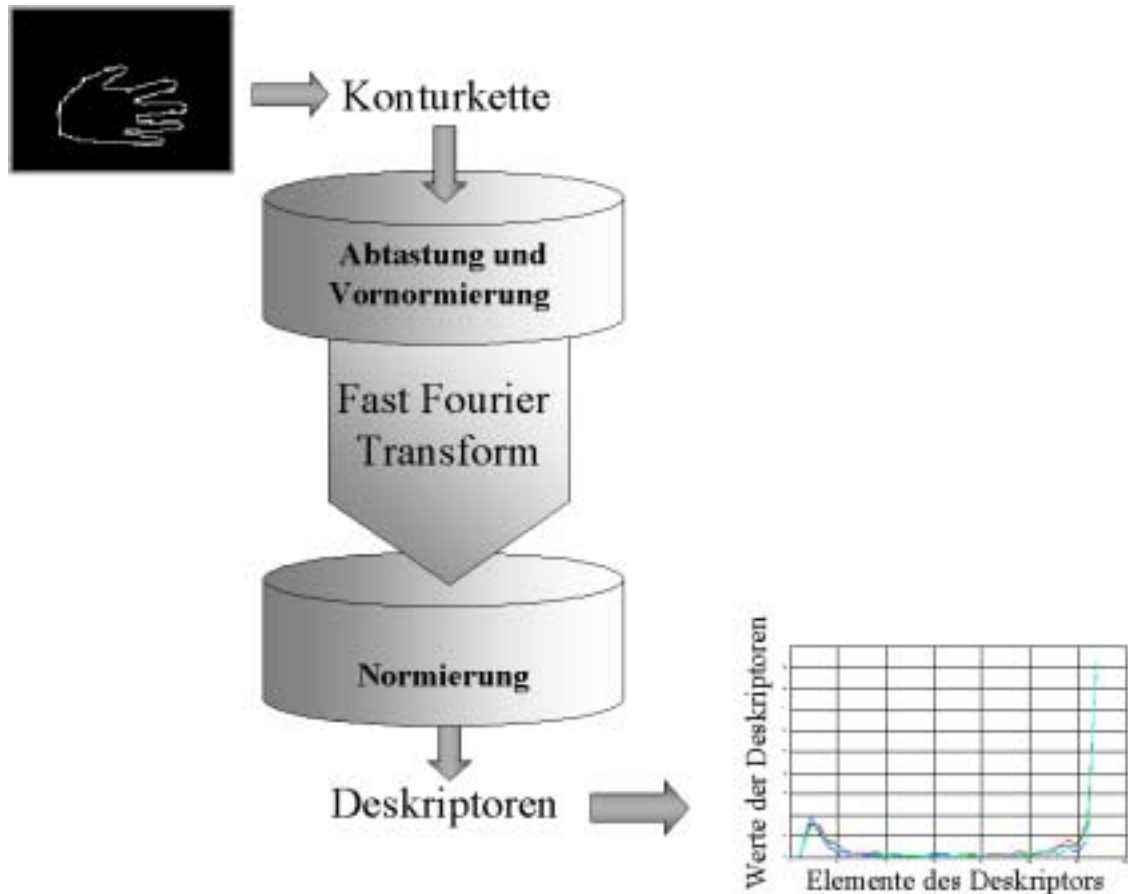


Abbildung 1.6: Merkmalsextraktion

Der Ablauf der Merkmalsextraktion wird in Abb. 1.6 dargestellt. Hier wird aus dem Konturbild die Reihenfolge von Punkten extrahiert, die die entsprechende Kontur darstellen. Die Konturkette wird abgetastet, um die gleiche Vektorlänge von allen Objekten zu haben, damit sie miteinander vergleichbar sind, und um den Rechenaufwand zu verringern. Danach werden durch Anwenden der Fast-Fourier-Transformation auf die Kette von Punkten aus dem vorhergehenden Schritt Deskriptoren berechnet. Diese Deskriptoren können noch nicht verglichen werden. Sie müssen zuerst normiert werden, damit das Ausgabesignal unabhängig von der Größe, der Orientierung, der Position im Bild und dem Anfangspunkt der Reihenfolge ist. Die Normierung wird in zwei Schritten ausgeführt, je nach Einfluß der entsprechenden Eigenschaft in der Fourier-Transformation. Manche Eigenschaften werden direkt auf der Kontur normiert und manche nach der Berechnung der Fourier-Deskriptoren.

### 1.3.3 Klassifikation

Um eine Klassifikation durchzuführen, ist es nötig, die drei folgenden Punkte festzusetzen:

- **Eine Referenzmenge.** Um die vorgeführten Gesten klassifizieren zu können, müssen die Deskriptoren mit Musterdeskriptoren verglichen werden. Diese Referenzmenge ist anpassbar an die Bedürfnisse der Benutzer. Es ist möglich die Bibliothek zu editieren, neue Gesten werden gelernt und nicht mehr gebrauchte können gelöscht werden.
- **Ein Vergleichskriterium.** Zusätzlich muß angegeben werden, wie Deskriptoren verglichen werden können. Dieses Kriterium soll die Unterschiedlichkeit bzw. die Ähnlichkeit zwischen zwei Deskriptoren angeben.
- **Ein Entscheidungskriterium.** Nachdem das Ähnlichkeitsmaß berechnet ist, muß eine Entscheidung getroffen werden. Dieses Kriterium enthält zwei Schritte:
  - Zuerst wird der Deskriptor aus der Referenzmenge gesucht, der mit dem Deskriptor der vorgeführten Geste die größte Ähnlichkeit hat.
  - Danach wird der Grad der Ähnlichkeit mit einem Schwellwert verglichen, um zu entscheiden, ob beide Deskriptoren ähnlich genug sind bzw. ob diese Geste überhaupt in der Referenzbibliothek existiert.

## 1.4 Aufbau der Arbeit

Hier wird die Struktur der vorliegenden Arbeit vorgestellt.

- **Stand der Technik** (Kapitel 2).  
Zuerst wird ein Ausblick auf den jetzigen Stand der Technik und das konkrete Thema und Ziel des Projekts gegeben. Zum Thema Gestenerkennung ist ((schon eine lange Strecke gelaufen worden))?????. Von verschiedenen Möglichkeiten sind für diese Diplomarbeit manche Verfahren implementiert worden, diese wurden angepasst, um unsere Ziele zu erreichen.
- **Theoretische Grundlagen** (Kapitel 3).  
Hier wird das Fundament des Algorithmus für die Vorverarbeitungsphase und die Klassifikation, so wie der für die Merkmalsextraktion erklärt.
- **Ansätze** (Kapitel 4).  
Um das Ziel des Projekts zu erreichen, ist ein Programm implementiert worden. Dieses Kapitel stellt die Implementierung von verschiedene Verfahren vor, wie die vorherige Theoretische Basis mit den verschiedenen Möglichkeiten, die getestet worden sind.

- **Implementation** (Kapitel ??).  
In diesem Kapitel wird die Implementierung des Programm in dem Roboter dargestellt und die Ergebnisse ausgewertet.
- **Experimente und Evaluation** (Kapitel 5).
- **Zusammenfassung und Ausblick** (Kapitel 6).  
Schließlich wird eine Zusammenfassung gegeben und die mögliche Erweiterung des Projekts dargestellt.

# Kapitel 2

## Stand der Technik

Das große Interesse der Forschung am Thema Bildverarbeitung und Objekterkennung hat eine Vielzahl von verschiedenen Methoden entwickeln lassen, um Merkmale aus einem Kamerabild zu extrahieren. Diese sind oft auf bestimmte Anwendungen und Situationen zugeschnitten und daher dort ausreichend.

Die Entscheidung die entwickelte Lösung für den Bildbasierte Statische Gesterkennungsalgorithmus zu implementieren folgt die Berücksichtigung der bisher angewandte Technik. Es wurden verschiedene Objekterkennungsmethode untersucht, manche davon auch für Roboter implementiert sind.

### 2.1 Verschiedene Methoden zum Bildbasierte Objekt Erkennung

#### 2.1.1 A Gesture Interface for Human-Robot-Interaction

Der Arbeit "A Gesture Interface for Human-Robot-Interaction" ([Triesch 98]) handelt sich um eine benutzerunabhängig Geste Schnittstelle an ein Roboter implementiert. Das erlaubt der Benutzer der Roboter zu steuern durch einfache Befehle, zum Beispiel, wie ein Objekt zu greifen und wo dieses Objekt zu legen. Die Analyse des Gestes verlässt auf Echtzeit Verfolgen der Benutzers Hand und ein detailliertes Analyse des Form des Handes bei unkontrollierte reale Welt Umgebungen. Das ergibt die folgende Bedingungen:

- Die visuelles System des Roboters muß mit veränderlichen und komplexe Hintergründe funktionieren. Eine gleichmäßige Hintergrund ist nicht ausreichend für reake Welt Anwendungen.
- Der System soll Benutzer unabhängig sein. Verschiedene Benutzer sollen damit arbeiten ohne wiedertrainieren.
- Die Benutzer braucht keine Handschuhe oder so etwas.

- Die Beleuchtungsbedingungen sind unkontrollierbar.
- Der Roboter soll Echtzeitfähig sein.

Diese Voraussetzungen übereinstimmen mit den Ziele dieser Arbeit in Kapitel 1.2.2 dargestellt. Um der Roboter durch so eine Schnittstelle zu steuern gibt es zwei Schritte, nämlich Verfolgen der Hand mit den Kameras und im Bild, und Hand Geste Klassifikation.

Der **Verfolgen des Handes** wird durch Bewegung und Farbe Queue implementiert. Ein zusätzliches Stereo Queue ausschließt möglichere Ziele die nicht in ((plane of fixation)) sind.

- Color Cue((texto fig3))
- Motion cue((texto fig4))

Attention maps werden extrahiert ((Fig5))

Die **Klassifikation des Gestentyps** erfolgt wenn der Hand zum Stillstand kommt. Die Klassifikation des Gestentyps ist der Teil der für die bilbasierte statische Gesten Erkennung von Interesse ist. Die Erkennung in [Triesch 98] ist an ((Elastic Graph Matching)) basiert, schon erfolgreich beim Objekt und Gesicht Erkennung. Die funktionieren Benutzer unabhängig und beim komplex verändbare Hintergrund. Im ((elastic graph matching)), die Objekte sind als benannte Graphiks representiert, wo die Knoten lokale Bildsinformation enthalten und die Ecken ((edges)) geometrisch Information tragen. (( )) Während die Anpassungsprozeß bestimmte geometrische ((transformations)) sind erlaubt:

- Skalierung bis zu 15% der original Größe.
- Drehung auf der Ebene bis zu 10 Grad.

(( )) Die Transformation des Bildes braucht 2.95 Sekunde wenn es auf eine konventionell Sun UltraSparc Workstation. Die Anpassung des Modellegraphiks fügt 1.88 Sekunden hinzu.

Diese geometrische ((transformations)) sind für die Ziele der jetzige Arbeit zu gering, da es wird hier versucht ein Winkel- und Skalierung-Unabhängige Algorithmus zu entwickeln.

### 2.1.2 Einfache Handgestikerkennung mit einem zweistufigen Nearest-neighbour Klassifikator

((Kestler 99))

### 2.1.3 Objekterkennung basierte auf der Silhouette.

Die Methode auf diesem Fall implementiert identifiziert Objekte basiert auf die Silhouette von Einzelbilder von eine graustufe Kamera gemacht. Die

### 2.1.4 Klassifikation von unregelmäßige gestaltet Mikro-Objekte durch Verwenden Komplex Fourier Deskriptoren

([Kindratenko 96]).

Dieses Methode würde für charakterisieren von komplexe unregelmäßige natürliche Objekte entwickelt. Dieses Methode benutzt Komplex Fourier Analyse anstatt die traditionelle Art von Fourier Analyse. Digitalisierte Bilder von Objekte sind verarbeitet und die Konturen sind extrahiert durch klassische Methode. Eine neue Vorverarbeitung Technik erlaubt die Normalisierung von die Position, Größe und Orientierung des Bildes. Eine neue Kontur Wiederabtastungstechnik ergibt eine genauer Annäherung von der Kontur. Die Wiederabtastener Kontur wird auf eine komplexe Ebene vertreten und die komplexe Fourier Koeffizienten sind berechnet. Diese Technik wird angewendet für die Klassifizierung von einzelne Algenzellen und ihre Konglomerate beim gebrauchten von zwei Methode: Neuronale Netze und Hierarchisches ((clustering))versammelnd.

Hier wird die Form representiert so daß mit geringe Menge von Daten es ist immer noch möglich die original Kontur zu kriegen mit genügend Genauigkeit. Kette Kodierung Representation mit Fourier Analyse zusammengefaßt entspricht eine sehr effektive Methode(( die das Problem mit ((reentrant shapes)) vermeidet)).

Die Fourier Analyse verlangen Ermäßigung der Fourier Koeffizienten Anzahl. Normalerweise sind die erste 10 bis 20 Koeffizienten benutzt, aber die Konturen durch die Rücktransformation dieser Deskriptoren umgebaut sind glatt Versionen von die Ursprüngliche. Das leitet zu Informationsverlust. Es gibt eine zweite Möglichkeit, anstatt die Fourier Deskriptoren zu reduzieren, kann die original Kontur abgetastet werden und dann transformiert. Das ergibt kleinere Deskriptoren und deshalb können alle Koeffizienten betrachtet werden.

Die Klassifikationsalgorithmus hier benutzt ist ((unsupervised)), nämlich Neuronale Netze, in gegeteil zu die üblich benutzt iterativ Methoden die suchen die optimal anpassende Referenz Deskriptor. Diese brauchen eine Bibliothek von Muster Formen, was keine gute Lösung ist für die Klassifikation unregelmäßige gestaltet Mikro-Objekte, weil es gibt viele nicht Klassifizierte.

((en este puedo ampliar explicando los metodos de preprocesamiento y resampling utilizados))

## 2.2 Auserwählte Methode. Begründung

Die in dieser Arbeit implementierte Methode folgt der Linie von den zuletzt dargestellten Methoden. Die Anwendung ist trotzdem ein bisschen unterschiedlich. Hier geht es um Handgesten zu erkennen, von Bildern, die nicht so perfekt sind wie die von der entsprechenden Methode. Deswegen wird nicht die gleiche ((resampling)) Methode benutzt. Hier sollte erklärt werden, warum jeder Vorverarbeitungsschritt gemacht wird (obwohl vielleicht das nach den Methoden erklärt werden sollte).

((A photo should be included to show for example why the contour points are not taken as they are on the algae classifier, it must show how the wrong points which are not significant would be included on the transformation, although there might also be a way to filter them, maybe by eliminating the points that are on ))

# Kapitel 3

## Theoretische Grundlagen

Da ein unverarbeitetes Signal eine Menge Informationen enthält, stellt der Vergleich zweier ursprünglicher Bildsignale ein komplexes Problem dar. Dies ist schon durch die Größe der Datenmenge zu begründen; in diesem Fall Matrizen der Dimension  $m$  mal  $n$ . Aber selbst wenn der Berechnungsaufwand vernachlässigt wird, verursachen stochastische und deterministische Störungen des Meßvorgangs Schwierigkeiten bei der Auswertung. Außerdem ist eigentlich nicht das Bildsignal von Interesse, sondern besondere enthaltene Merkmale, die selten an gleicher Stelle und in gleicher Ausrichtung in Erscheinung treten. Aus diesem Grund werden die Bilder verarbeitet und Methoden angewandt, um einfache Merkmale aus Bildsignalen zu extrahieren.

Deskriptoren sind Merkmalsätze die die Eigenschaften enthalten, die implizit im Bild und eindeutig für jeden Gestentyp sind. Beispiele von Deskriptoren im Bereich Bildverarbeitung sind Momente, Fourier-Transformation, Invarianz, Distanz-Transformation, Hough-Transformation. In dieser Arbeit werden die Fourier-Deskriptoren benutzt aus Gründen die in Kapitel 2 dargestellt wurden. In der hier beschriebenen Lösung zur Gestenerkennung wird die Kontur der Benutzerhand als grundlegendes Klassifikationsmerkmal verwendet. Durch die Anwendung der Fourier-Transformation auf die Kontur werden die entsprechende Deskriptoren erhalten.

### 3.1 Mathematische Analyse mit Hilfe von Fourier Deskriptoren

Fourierdeskriptoren basieren im Rahmen der Bildverarbeitung auf der Anwendung der *diskreten Fourier-Transformation* (DFT). Die diskrete Fourier-Transformation ist eine Methode der näherungsweise Berechnung von Spektren abgetasteter Signale ( $ccc$ ). Die Zeitszusammenhänge bzw. die Ortsbereichsähnlichkeiten sind in diesem Fall durch Fourier-Deskriptoren zu erhalten.

### 3.1.1 Fourieranalyse kontinuierlicher Signale

Im Folgenden wird die Fourieranalyse für die zeitkontinuierlichen Signale dargelegt ebenso wie für Ortsbereichssignale. Innerhalb der kontinuierlichen Signale sind periodische und nicht periodische Signale zu unterscheiden.

Durch die Fouriertransformation wird eine nicht periodische Funktion in ein kontinuierliches Frequenzspektrum umgewandelt. Die Analyse eines periodischen Signals ergibt hingegen eine Folge von Frequenzen oder eine Fourierreihe. Die mathematische Formulierung lautet wie folgt:

#### *Nichtperiodisches Signal* $\longrightarrow$ *kontinuierliches Frequenzspektrum*

Die direkte Transformation lautet:

$$X(f) = \mathcal{F}(x(t)) = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j\omega t} dt = \int_{-\infty}^{+\infty} x(t) \cdot e^{-j2\pi f \cdot t} dt \quad (3.1)$$

Die inverse Transformation hat die Gestalt:

$$x(t) = \mathcal{F}^{-1}(X(f)) = \int_{-\infty}^{+\infty} X(f) \cdot e^{j\omega t} df = \int_{-\infty}^{+\infty} X(f) \cdot e^{j2\pi f \cdot t} df \quad (3.2)$$

#### *Periodisches Signal im Zeitbereich* $\longrightarrow$ *diskretes Fourierspektrum (Fourierreihe)*

Hier lautet die Spektrenzerlegung nun analog:(ccc)

$$c_\omega = \frac{1}{T} \cdot \int_{t_0+T}^{t_0} x(t) \cdot e^{-j2\pi k f_0 t} dt \quad (3.3)$$

Die Darstellung des zerlegten Signales ist gleichzeitig die Rücktransformation (ccc)

$$x(t) = \sum_{k=-\infty}^{+\infty} c_k \cdot e^{+j2\pi k f_0 t} \quad (3.4)$$

### 3.1.2 Fourieranalyse diskreter Signale

Die Analyse diskreter Signale unterscheidet sich von der obenstehenden Darstellung des kontinuierlichen Falls nur in der Eigenschaft, daß die Integration hier durch die Summe über alle einzelnen Elemente ersetzt wird. Auch hier findet die Trennung in periodische und nichtperiodische Signale statt. Das Hauptmerkmal der Abtastung im Zeitbereich ist jedoch die spektrale Periodisierung im Frequenzbereich (ccc).

Die diskreten Signale sind folgendermaßen vertreten:

$$x(n) = \sum_{k=0}^{\infty} x(k) \cdot \delta((n - k) \cdot T_0) \quad (3.5)$$

Damit lautet das kontinuierliche Spektrum der Abtastfolge((ccc)):

$$X(\Omega) = \mathcal{F}(x(n)) = \int_{-\infty}^{+\infty} \sum_{k=0}^{\infty} x(k) \cdot \delta((n - k) \cdot T_0) e^{-j2\pi f t} dt \quad (3.6)$$

Beim Ersetzen von  $\omega = 2\pi f$  und nach Elimination von  $T_0$  (vereinbarungsgemäß setzt man  $T_0 = 1$ ) folgt:

$$X(\Omega) = \sum_{n=0}^{\infty} x(n) \cdot e^{-j\Omega T_0} = \sum_{n=0}^{\infty} x(n) \cdot e^{-j\Omega} \quad (3.7)$$

Auf den selben Ausdruck gelangt man mit Hilfe der  $z$ -Transformation, ausgehend von Gleichung 3.8:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n) \cdot z^{-k} \quad (3.8)$$

Nun beschränkt man den Wertebereich von  $z$ , der eigentlich die gesamte komplexe Ebene mit Ausnahme der Divergenzgebiete der Reihe nach Gleichung 3.8 umfaßt, auf den Einheitskreis.

$$z = e^{j\omega} \quad (3.9)$$

Damit folgt:

$$X(z)|_{z=e^{j\omega}} = \sum_{k=-\infty}^{+\infty} x(k) \cdot e^{-j\omega k} \quad (3.10)$$

Auch hier ist das Spektrum kontinuierlich und periodisch und zwar mit der Periode  $\pi$ .

### ***Abtastung im Zeitbereich $\rightarrow$ Periodisierung im Frequenzbereich***

Die letzte Möglichkeit ist die Abtastung des Frequenzspektrums. Es gibt hier zwei Alternativen:

1. Betrachtung eines periodisch abgetasteten Signales  $x(k)$ .

Die Transformationsvorschrift kann hier einfach aus der Gleichung 3.7 gewonnen werden; dabei wählt man die Zahl  $N$  entsprechend einer Periode von Abtastpunkten. Es gilt dann für die Ermittlung der Fourierkoeffizienten  $X(f)$  die Berechnungsvorschrift:

$$X(\omega) = \sum_{n=0}^{\infty} x(n) \cdot e^{-j\omega n} \quad (3.11)$$

wobei mit der Annahme der Grundschwingung  $\omega_0 = \frac{\omega}{2\pi}$  folgt:

$$\tilde{X}(n) = \frac{1}{N} \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi \frac{kn}{N}} \quad (3.12)$$

Die Berechnungsvorschrift nach Gleichung 3.12 liefert also Fourierkoeffizienten in Analogie zur Fourierreihe eines kontinuierlichen Signales. Im Gegensatz zur zeitkontinuierlichen Fourierreihe, bei der zur vollständigen Wiedergewinnung eines Signals aus dem Spektrum alle, d.h. unendlich viele Koeffizienten benötigt werden, genügen bei der Betrachtung des abgetasteten Signales genau  $N$  Koeffizienten (wobei  $N$  der Länge der Abtastfolge entspricht). Dies ist ohne nähere Erläuterung auch einsichtig, da zur Darstellung von  $n$  diskreten Werten auch  $n$  Exponentialschwingungen ausreichen.

$$x(k) = \sum_{n=0}^{N-1} \tilde{X}(n) \cdot e^{+j2\pi \frac{kn}{N}} \quad (3.13)$$

Das entstehende Spektrum ist aufgrund der Betrachtung eines periodischen Signales natürlich diskret. Die in Gleichung 3.13 gegebene Reihendarstellung bezeichnet man entweder in Anlehnung an die Fourierreihe als DFS (*Discrete Fourier Series*, entspricht der IDFT). Die DFT ist die Berechnungsvorschrift für die Koeffizienten der DFS, wobei sich die zweite Bezeichnung auf die im Punkt 2) gegebene Definition stützt.

## 2. Abtastung eines kontinuierlichen Spektrums

Die äquidistante Abtastung des kontinuierlichen Spektrums

$$X(\omega) = \sum_{k=-\infty}^{+\infty} x(k) \cdot e^{-j\omega k} \quad (3.14)$$

ergibt bei  $\omega = n \cdot \omega_0$

$$X(n) = \sum_{k=-\infty}^{+\infty} x(k) \cdot e^{-jn \cdot \omega_0} \quad (3.15)$$

Da bei der Betrachtung zeitdiskreter Systeme nur jeweils ein eingeschränkter Frequenzbereich betrachtet werden kann (Abtasttheorem), alle Spektren also mit der Periode  $\pi$  periodisch sind, gilt:

$$\omega_0 = \frac{2\pi}{N} \quad (3.16)$$

Dies führt auf die Darstellung

$$X(n) = \sum_{k=-\infty}^{+\infty} x(k) \cdot e^{-j2\pi \frac{n \cdot k}{N}} \quad (3.17)$$

Weil  $X(n)$  durch Diskretisierung der Fouriertransformation  $X(\omega)$  des betrachteten Signales hervorgegangen ist, wird die durch Gleichung gldft2 bezeichnete Transformation auch diskrete Fouriertransformation genannt (DFT).

Die diskrete Fourierreihe (DFS) entspricht also der IDFT, die diskrete Fouriertransformation (DFT) ist die Bildungsvorschrift der Koeffizienten der DFS<sup>1</sup>. Nutzt man die DFT zur Spektrenberechnung nichtperiodischer Signale, so ist zu erwarten, daßes das entsprechende kontinuierliche Spektrum des nichtperiodischen Signals nur näherungsweise wiedergibt. Man bezeichnet dieses Phänomen als *Aliasing* im Zeitbereich (d.h. im Ortsbereich in der digitalen Bildverarbeitung). Wie beim Aliasing, das durch Verletzung des Abtasttheorems entsteht und welches zu Spektrenüberlappungen führt, liefert das Zeitbereichs-Aliasing Signalverfälschungen bei nichtperiodischen Signalen. Dieses Phänomen tritt bei periodischen Signalen nicht auf.

((revisar todas las formulas anteriores comparando con otro libro porque parece que pueden estar confundidas))

## 3.2 Anwendung in der digitalen Bildverarbeitung

Im Fall des vorgeschlagenen Algorithmus zur Gestenerkennung bestehen die abgetasteten Signale aus Handkonfigurationen, die mit den Kameras aufgenommen sind. Genauer gesagt werden die genannten Signale durch eine Reihenfolge von Konturpunkten der Hand gebildet. Es handelt sich also um ein diskretes Signal; hierfür wird der entsprechende Fall betrachtet. In der Bildverarbeitung spricht man vom Ortsbereich anstatt vom Zeitbereich. Um solche zweidimensionale Probleme zu vereinfachen, werden die Bildpunkte in der komplexen Ebene referenziert.

Die Fourierdeskriptoren bestehen nun aus den Koeffizienten, die mit den Methoden der DFT aus den Konturpunkten gewonnen werden können.

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi(\frac{k \cdot n}{N})} \quad (3.18)$$

mit

$$k \rightarrow x(k) | k \in Z, x(k) \in C \quad (3.19)$$

Im Folgenden wird ein Beispiel dargestellt, um die Begriffe ebenso wie die Eigenschaften der Fourierdeskriptoren klar zu machen.

### 3.2.1 Berechnung der Fourierdeskriptoren

Als Beispielkontur soll hier ein Dreieck benutzt werden (siehe Abbildung 3.1). Dadurch ist es möglich, den Einfluß verschiedener geometrischer Änderungen auf die

---

<sup>1</sup>Anmerkung: Die Normierung  $\frac{1}{N}$  kann entweder bei der Hintransformation oder bei der Rücktransformation durchgeführt werden. Bei den folgenden Betrachtungen wird sie bei der Rücktransformation vorgenommen.

Kontur zu betrachten. Zuerst werden die Abtastpunkte ausgewählt, welche die Kontur beschreiben.

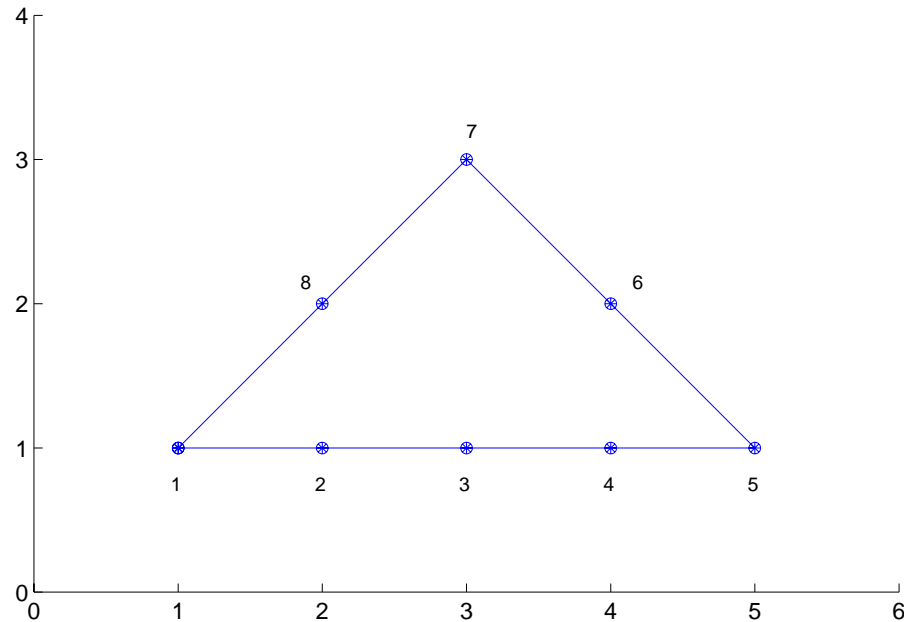


Abbildung 3.1: Beispielkontur Dreieck

Die entsprechenden Vektoren  $\{x(k)\}$ ,  $\{y(k)\}$  enthalten die Folge der Bildpunkte der Kontur:

$$\{x(k)\} = \{1, 2, 3, 4, 5, 4, 3, 2\} \quad (3.20)$$

$$\{y(k)\} = \{1, 1, 1, 1, 1, 2, 3, 2\} \quad (3.21)$$

Beide können im Komplexen zusammengefaßt werden wie folgt:

$$\{s(k)\} = \{x(k)\} + j \cdot \{y(k)\} = \{1+j, 2+j, 3+j, 4+j, 5+j, 4+2j, 3+3j, 2+2j\} \quad (3.22)$$

Wegen Linearität in Bezug auf die Punktvariablen gilt die folgende Beziehung, die den Aufwand für die Berechnung der Deskriptoren verkleinert:

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi \frac{n \cdot k}{N}} \quad (3.23)$$

$$Y(n) = \sum_{k=0}^{N-1} y(k) \cdot e^{-j2\pi \frac{n \cdot k}{N}} \quad (3.24)$$

Wie die folgende Berechnung zeigt, muß die Transformation also nur für eine Folge komplexer Punkte ausgeführt werden anstatt für zwei Folgen von realen Werten:

$$S(n) = \sum_{k=0}^{N-1} s(k) \cdot e^{-j2\pi \frac{n \cdot k}{N}} = \sum_{k=0}^{N-1} (x(k) \cdot e^{-j2\pi \frac{n \cdot k}{N}} + j \cdot y(k) \cdot e^{-j2\pi \frac{n \cdot k}{N}}) \quad (3.25)$$

$$S(n) = \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi \frac{n \cdot k}{N}} + j \cdot \sum_{k=0}^{N-1} (y(k) \cdot e^{-j2\pi \frac{n \cdot k}{N}}) \quad (3.26)$$

$$S(n) = X(n) + jY(n) \quad (3.27)$$

Also  $\{S(n)\}$  ist eine Folge mit  $N$  Elementen  $S(n)|n = 0, 1, \dots, N - 1$  die durch die Anwendung der Fourier-Transformation auf eine Folge von Komplexe Zahlen erhalten wird.

### 3.2.2 Eigenschaften der DFT

Nach der Angabe der Definition der DFT sollen einige Eigenschaften der Transformation näher erläutert werden. Damit wird die Einfluß der Eigenschaften auf die Originalkontur und die Fourier-Transformation dieser Kontur verglichen.

**Identität:** Aus der Kontur  $s(k)$  wird die Deskriptor folge  $S(n)$  gewonnen

**Kontur**  $s(k)$

**Fourier-Deskriptor**  $S(n)$

**Rotation des Objektes:** Der Drehung eines Objektes um den Ursprung des komplexen Koordinatensystems um den Winkel  $\theta$  entspricht die Multiplikation mit dem Faktor  $e^{j\theta}$ . Hierbei ist der Drehwinkel  $\theta$  in Radian angegeben.

Da dieser Faktor alle Elemente der Folge unabhängig von allen anderen Parametern in gleicher Weise beeinflusst, gilt die folgende Beziehung:

$$s_r(k) = s(k) \cdot e^{j\theta} \quad (3.28)$$

$$\{S_{r(\theta)}(n)\} = \mathcal{F}(\{s_r(k)\}) = \mathcal{F}(\{s(k)\} \cdot e^{j\theta}) = e^{j\theta} \mathcal{F}(\{s(k)\}) \quad (3.29)$$

**Kontur**  $s_r(k) = s(k) \cdot e^{j\theta}$

**Fourier-Deskriptor**  $S_{r(\theta)}(n) = S(n) \cdot e^{j\theta}$

Ein numerisches Beispiel soll eine Rotation um 30 Grad des Ausgangsdreiecks veranschaulichen (Abb. 3.2):

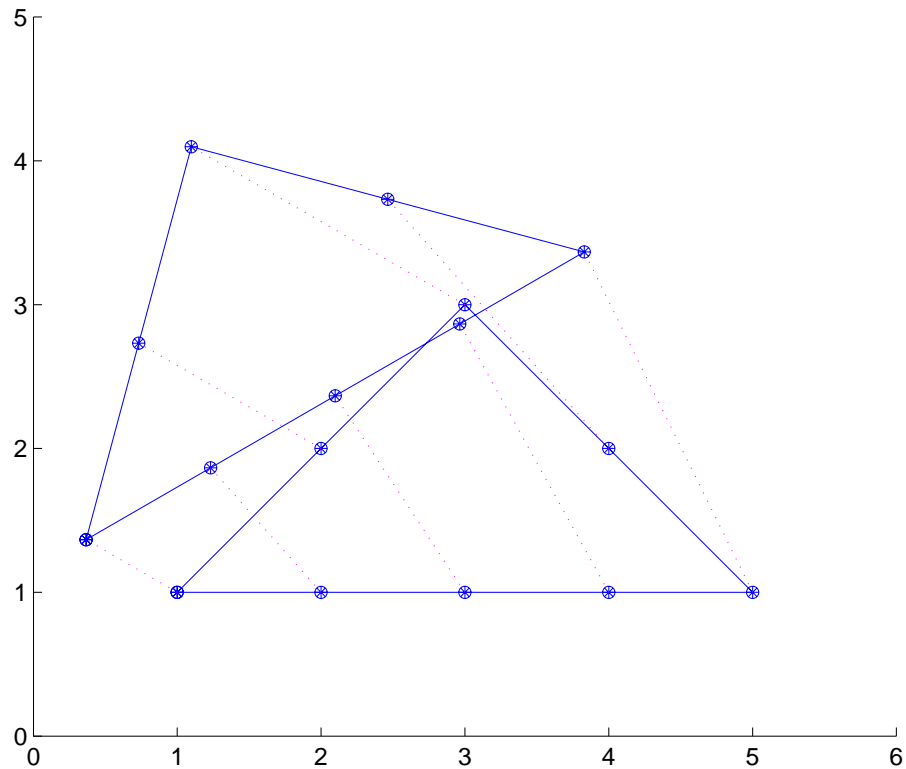


Abbildung 3.2: Gedrehtes Dreieck

((ccc))

**Translation des Objektes:** Die Relation zwischen der Fouriertransform eines Objektes vor und nach Verschiebung ist nicht so direkt darzustellen wie bei der Objektdrehung. Die lässt sich ausdrücken durch die folgende Beziehung zwischen den Konturfolgen:

$$s_t(k) = s(k) + \Delta_{xy} \tag{3.30}$$

$$\begin{aligned} \{S_{t(\Delta)}(n)\} &= \mathcal{F}(\{s_t(k)\}) = \\ &= \mathcal{F}(\{s(k)\} + \Delta_{xy}\{V(k)\}) = \\ &= \mathcal{F}(\{s(k)\}) + \Delta_{xy}\mathcal{F}(\{V(k)\}) \end{aligned} \tag{3.31}$$

$V(k) = 1$   
 $\mathcal{F}(\{V(k)\}) = \{\delta(n)\}$  was ergibt  $\delta(1) = N$  und  $\delta(n) = 0$  für  $n = 1, 2, \dots, N - 1$   
 $\delta(n) = \{N, 0, 0, \dots, 0\}$

**Kontur**  $s_t(k) = s(k) + \Delta_{xy}$

**Fourier-Deskriptor**  $S_{i(\Delta)}(n) = S(n) + \Delta_{xy}\delta(n)$

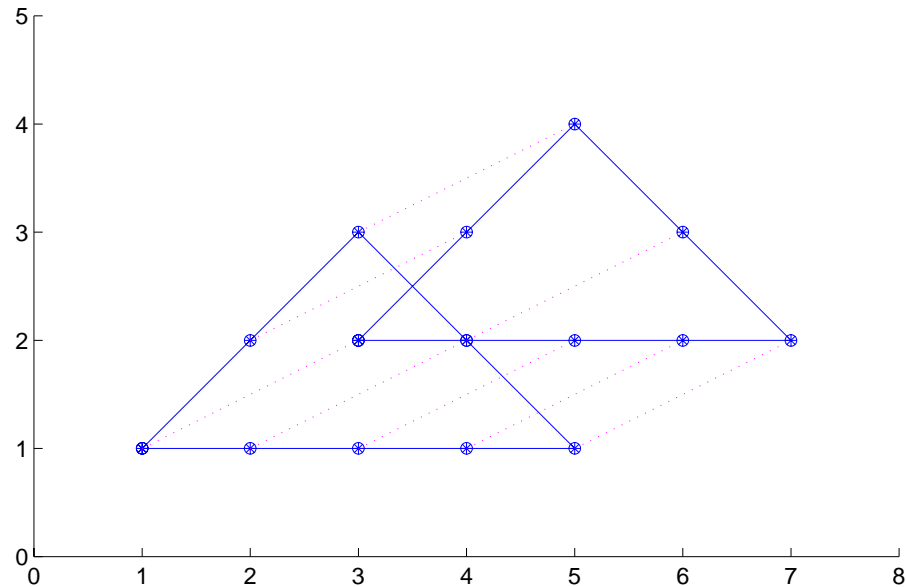


Abbildung 3.3: Verschobenes Dreieck

**Skalierung des Objektes:** Bei der Skalierung eines Objektes ist ein der Rotation sehr ähnlicher Zusammenhang gegeben. In diesem Fall sind alle Komponenten der Folge mit einem bestimmten Wert multipliziert. Die Relation der Deskriptoren ist dann ebenfalls proportional.

$$s_s(k) = \alpha \cdot s(k) \quad (3.32)$$

$$\{S_s(n)\} = \mathcal{F}(\{s_s(k)\}) = \mathcal{F}(\alpha \cdot \{s(k)\}) = \alpha \mathcal{F}(\{s(k)\}) \quad (3.33)$$

**Kontur**  $s_s(k) = \alpha \cdot s(k)$

**Fourier-Deskriptor**  $S_s(n) = \alpha \cdot S(n)$

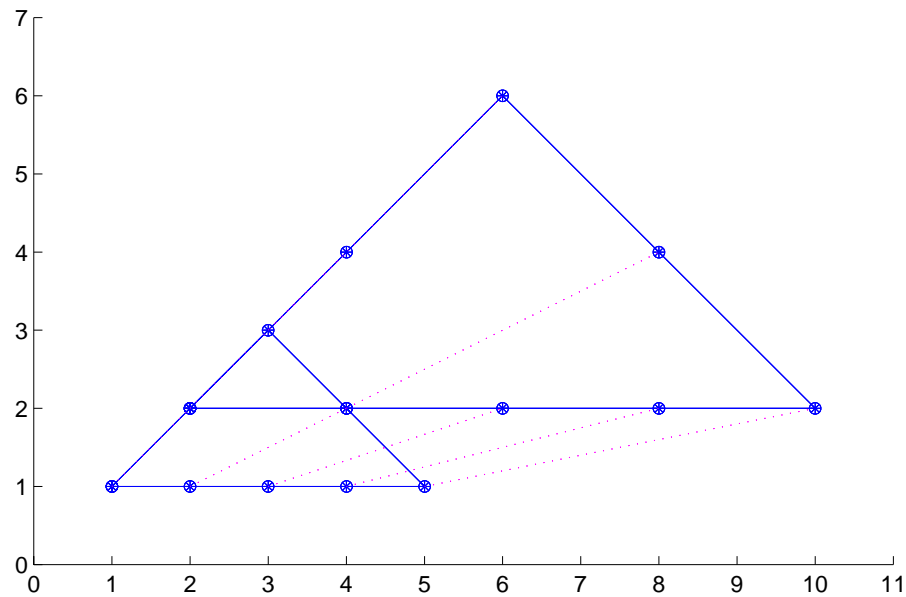


Abbildung 3.4: Skaliertes Dreieck

**Einfluß der Wahl des Anfangspunktes der Konturfolge:** Die Beziehung zwischen dem Einfluß des Anfangspunktes auf der Kontur und auf der Fourier-Deskriptor ist nicht so einfach zu bezeichnen wie bei den anderen Eigenschaften. Um sie festzustellen, benutzen wir das Beispiel mit der Folge von 8 Punkten bei der der Anfangspunkt ist  $k_0 = 3$  verschoben.

$$s_p(k) = s(k - k_0) \quad (3.34)$$

Beim Einsetzen in die Gleichung:

$$S(n) = \sum_{k=0}^{N-1} s(k) \cdot e^{-j2\pi(\frac{k \cdot n}{N})} \quad (3.35)$$

$$\begin{aligned} S(n) &= s(0)e^{-j2\pi(\frac{0 \cdot n}{N})} + s(1)e^{-j2\pi(\frac{1 \cdot n}{N})} + \\ &+ s(2)e^{-j2\pi(\frac{2 \cdot n}{N})} + s(3)e^{-j2\pi(\frac{3 \cdot n}{N})} + s(4)e^{-j2\pi(\frac{4 \cdot n}{N})} + \\ &+ s(5)e^{-j2\pi(\frac{5 \cdot n}{N})} + s(6)e^{-j2\pi(\frac{6 \cdot n}{N})} + s(7)e^{-j2\pi(\frac{7 \cdot n}{N})} \end{aligned} \quad (3.36)$$

$$\begin{aligned} S_p(n) &= s(5)e^{-j2\pi(\frac{5 \cdot n}{N})} + s(6)e^{-j2\pi(\frac{6 \cdot n}{N})} + \\ &+ s(7)e^{-j2\pi(\frac{7 \cdot n}{N})} + s(0)e^{-j2\pi(\frac{0 \cdot n}{N})} + s(1)e^{-j2\pi(\frac{1 \cdot n}{N})} + \\ &+ s(2)e^{-j2\pi(\frac{2 \cdot n}{N})} + s(3)e^{-j2\pi(\frac{3 \cdot n}{N})} + s(4)e^{-j2\pi(\frac{4 \cdot n}{N})} \end{aligned} \quad (3.37)$$

Diese Gleichungen zeigen, daß sich die Elemente des Deskriptors mit einem anderen Anfangspunkt ergeben, wenn die Originaldeskriptorelemente mit  $e^{-j2\pi k_0 n/N}$  multipliziert werden.

**Kontur**  $s_p(k) = s(k - k_0)$

**Fourier-Deskriptor**  $S_p(n) = S(n) \cdot e^{-j2\pi k_0 n/N}$

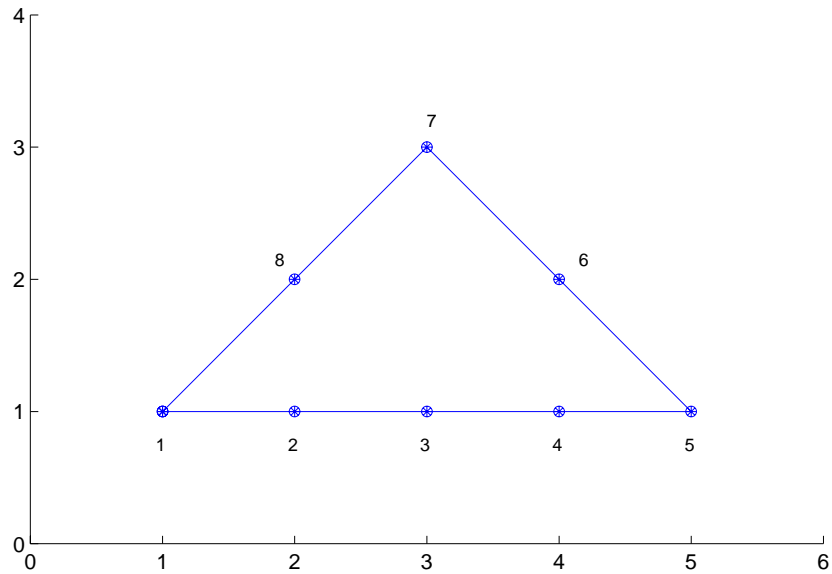


Abbildung 3.5: Originales Dreieck

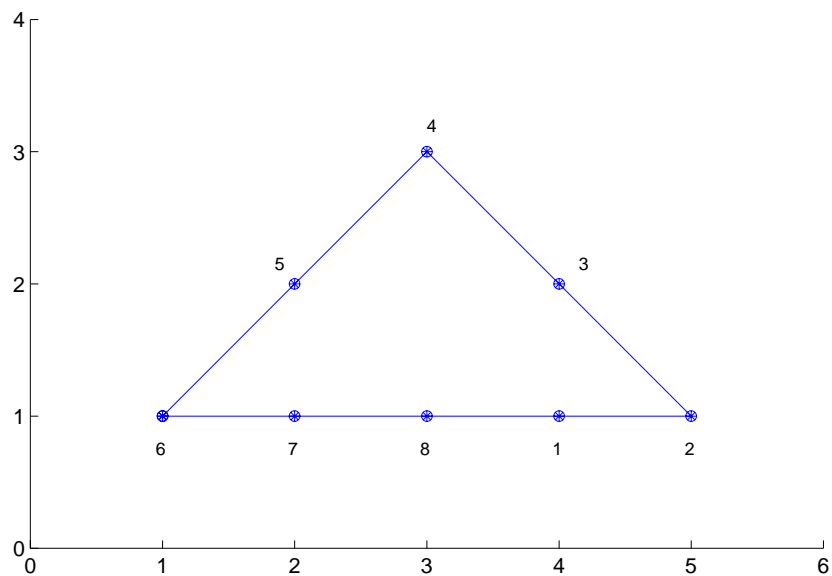


Abbildung 3.6: Anderer Anfangspunkt

# Kapitel 4

## Ansätze

In diesem Kapitel sollen die in Kapitel 3 beschriebenen Grundlagen in die Gestenerkennung mittels Bildverarbeitung übertragen werden. Die in Kapitel 1.3 vorgestellte Vorgehensweise wird hier um die entsprechenden theoretischen Grundlagen erweitert, und die verschiedenen Ansätze und Alternativen, die bei der Durchführung der Diplomarbeit gewählt wurden, werden hier beschrieben und ausgewertet.

Zuerst wird der Vorverarbeitungsprozeß wie in [Ly 01] entwickelt kurz beschrieben. Dieser Algorithmus wird allerdings nur bei der Implementation auf dem Roboter benutzt; zur Entwicklung dieser Diplomarbeit wurde eine einfachere vorläufige Lösung verwendet. Der nächste Schritt ist die Merkmalsextraktion aus der erhaltenen Konturkette. Dafür ist ein Resampling der betreffenden Kette notwendig. Letztendlich wird die Klassifizierungsmethode und der Aufbau der Bibliotheken, die die Modelldeskriptoren enthalten werden, erklärt. Die Algorithmen wurden mit Hilfe von TCL-TK, C und Matlab programmiert.

### 4.1 Vorverarbeitung

Um die Merkmale des Bildsignals zu extrahieren und später zu vergleichen, wird die Fourier-Transformation auf die Kontur der Geste angewendet. Deshalb ist es notwendig, das aufgenommene Rohsignal zu verarbeiten. Die Konturextraktion wird in zwei Arten durchgeführt. Der einer, Hautfarbsegmentierung wird nur bei der Implementation angewendet. Die zweite Methode wird nur vorläufig in der Untersuchungsphase angewendet.

#### 4.1.1 Vorläufiger Algorithmus zur Konturextraktion

Da während der Arbeit an der vorliegenden Diplomarbeit gleichzeitig die Realisierung von [Ly 01] lief, war es nötig, eine Alternative für die Konturextraktion aus dem Bildsignal zu suchen. Dafür ist die Matrox Imaging Library verwendet worden.

- Mit dem Kamerakopf (im Kapitel 1.2.1 beschrieben) werden die Bilder der Hände aufgenommen. Zur Vereinfachung des Algorithmus wird ein schwarzer Hintergrund benutzt.
- Die Farbbilder werden in Grauwertbilder umgewandelt.

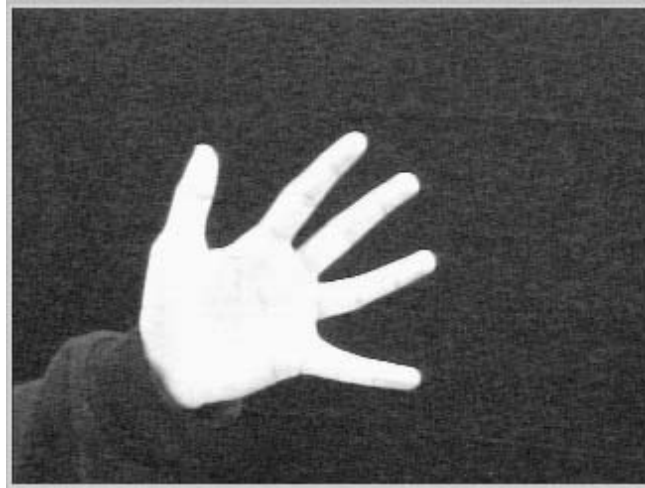


Abbildung 4.1: Grauwertbild mit Benutzerhand vor dunklem Hintergrund

- Diese werden mit einem vorher ausgewählten Schwellenwert binarisiert.

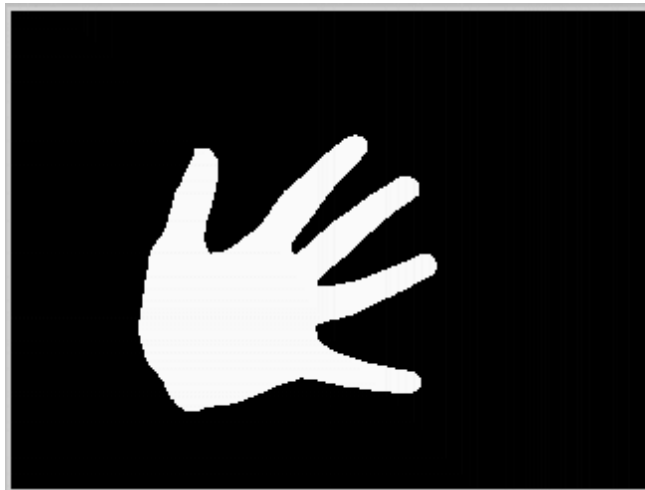


Abbildung 4.2: Binarisiertesbild der Benutzerhand

- Aus dem Bild werden sämtliche Blobs <sup>1</sup> extrahiert.

---

<sup>1</sup>Blobs sind definiert als "zusammenhängende Pixelregionen in einem Bild, welche dieselbe Farbe haben" (nach [MIL99])

- Der Blob der Hand im Bild hat ungefähr zwischen 100 und 40000 Pixel. Die Größe der einzelnen Blobs wird gemessen und es wird festgestellt ob es zwischen den erforderlichen Grenzen liegt. Dies garantiert, dass:
  - die ganze Hand im Bild ist
  - die Hand eine gewisse Mindestgröße aufweist
- Der Blob, der innerhalb der Schwelle liegt, wird als Hand erkannt und seine Kontur wird extrahiert. Die Kontur entspricht die Reihenfolge der Punkten am Grenze des Blobs. Der Anfangspunkt dieser Reihenfolge ist der Punkt weiter links am höchsten Linie des Blobs.

### 4.1.2 Hautfarbesegmentierung

Die adaptive Hautfarbesegmentierung nach [Ly 01] läuft nach dem in Abbildung 4.3 gezeigten Schema ab.

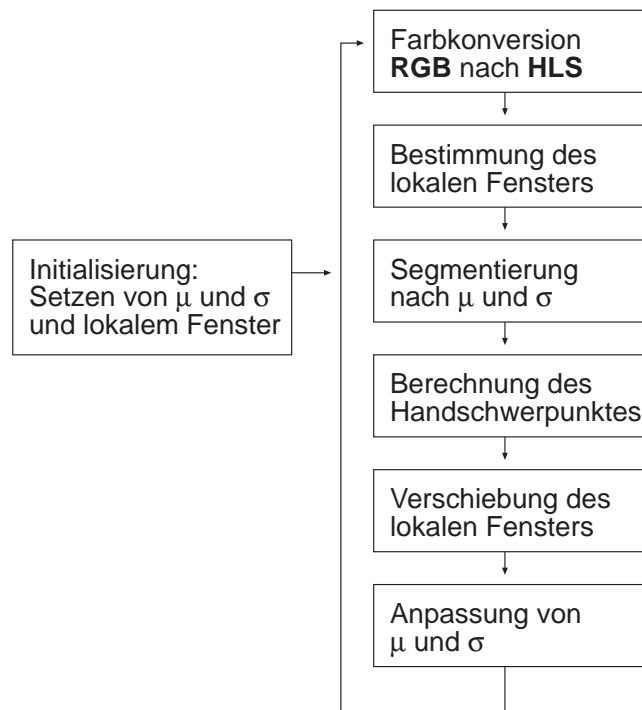


Abbildung 4.3: Hautfarbesegmentierung

Zunächst muss ein Referenzwert für die Segmentierung gesetzt werden. Dies erfolgt durch das Halten der Hand vor die Kamera beim Starten des Algorithmus. Dabei wird ein Mittelwert für den zu findenden Farbwinkel der Haut in einem lokalen Fenster im Kamerabild errechnet.

Dazu wird die von der Kamera gelieferte Darstellung der Farben im RGB-Format<sup>2</sup> in das HLS-Format<sup>3</sup> umgerechnet. Vorteil dieses Formats ist die separate Repräsentation des Farbwerts. Dadurch kann die Segmentierung in Grenzen unabhängig von Helligkeitsschwankungen erfolgen. Außerdem wird hier die Varianz für die Segmentierung gesetzt.

Danach wird der vorhandene Mittelwert in der Schleife immer wieder neu über die gefundenen Farbpunkte berechnet und mit einer Schrittweite  $\Delta$  an das aktuelle Kamerabild adaptiert.

Für die Schrittweite und die Varianz sind empirische Werte gesetzt. Der Algorithmus arbeitet stabil bei sich langsam vollziehenden Schwankungen des Lichtspektrums wie beispielsweise bei Wetterveränderungen. Das Einschalten eines Neonlichtes überschreitet jedoch die Veränderung, die von der Schrittweite  $\Delta$  erfasst werden kann.

Obwohl aufgrund von Rauschen die Hand mit Löchern durchzogen ist, bleibt die Kontur als Erkennungsmerkmal der Geste sehr gut erhalten.

## 4.2 Merkmalsextraktion

Um die Merkmale der Kontur zu extrahieren, werden zuerst die Punkte der Konturkette neu abgetastet. Auf diese neu erhaltene Kette wird daraufhin die Fourier-Transformation angewandt, um vergleichbare Deskriptoren zu erhalten. Ab dieser Schritt werden alle Algorithmen in Matlab untersucht, da Matlab erlaubt einfachere Bearbeitung der Daten.

### 4.2.1 Resampling der Konturkette

Die Konturkette, die direkt aus dem Bild entnommen wird, besteht aus einer veränderlichen Punktmenge. Die Länge der Konturkette hängt von verschiedenen Faktoren ab:

- Die Art der Geste.  
Die Anzahl der Konturpunkte ist z.B. sehr unterschiedlich bei gespreizten und geschlossenen Fingern.
- Die Geometrie der Hand des Benutzers.
- Der Abstand zwischen Hand und Kameras.

---

<sup>2</sup>Jedes Pixel wird dargestellt durch ein Byte-Tripel, dabei steht jedes Byte für den Farbanteil von rot, grün und blau.

<sup>3</sup>Hier wird jedes Bildelement durch die Komponenten Farbwinkel (Hue), Helligkeit (Luminance) und Sättigung (Saturation) dargestellt.

Die Länge der Kette schwankt ungefähr zwischen 200 und 1000 Elementen. Die Abtastung soll eine Anzahl von Punkten extrahieren, die nie größer ist als die Gesamtzahl der Elemente in der Kette.

#### 4.2.1.1 Ziel des Resampling

Das Resampling hat mehrere Ziele, die hier dargestellt sind.

1. Um die Deskriptoren vergleichen zu können, müssen sie die gleiche Anzahl an Elementen haben. Die Anzahl der Elemente eines Fourier-Deskriptors hängt von der Länge des zu transformierenden Vektors ab.
2. Der Rechenaufwand wird in zweierlei Hinsicht reduziert:
  - (a) Je weniger Punkte in der Kontur enthalten sind, umso schneller werden die Transformationen an der Kette ausgeführt.
  - (b) Wenn die Anzahl der Elemente im Vektor eine Potenz von zwei ist, kann der FFT-Algorithmus angewandt werden. Dadurch wird die Anzahl die Operationen von  $N^2$ , wie beim DFT, auf  $N \cdot \ln N$  reduziert.

#### 4.2.1.2 Resampling Methode

Es gibt verschiedene Resampling-Techniken, die auch jeweils unterschiedliche Ergebnisse liefern, je nach implementiertem Kriterium. Im Fall von Formenerkennung basierend auf der Konturkette ist es wichtig, genügend Information zu behalten um die Kontur zu identifizieren. Beim Verwenden von Fourierdeskriptoren im Gestenerkennungsalgorithmus ist es wichtig, wie die Punkte verteilt sind. Es wurden dazu zwei Methoden getestet:

**Kriterium der größten Krümmung** (in [Kindratenko 96] entwickelt). In dieser Methode werden die Punkte nach der Kriterium der größten Krümmung abgetastet. Dadurch werden die Punkte behalten, die mehr Information enthalten. Die Punkte mit größerer Krümmung bezeichnen normalerweise die Fingerspitze. Bei gleichen Gesten kann es vorkommen daß große Unterschiede auftreten in der Menge von Punkten, die an der gleichen Stelle ausgewählt werden. Es kann auch Punkte geben, wo die Kontur Rauschen von der Vorverarbeitung hat (siehe Abb. 4.4). Die Methode von [Kindratenko 96] ist gut geeignet, um durch die Rücktransformation ein neue Kontur ähnlich der Originalkontur zu erhalten, aber dies ist in unserem Fall nicht nötig, da die Klassifizierung erreicht werden soll. Aus diesem Grund ist es besser eine alternative Methode zu benutzen, die nicht kleine Details, sondern das Wesen der Kontur als abgetastete Reihe widerspiegelt.

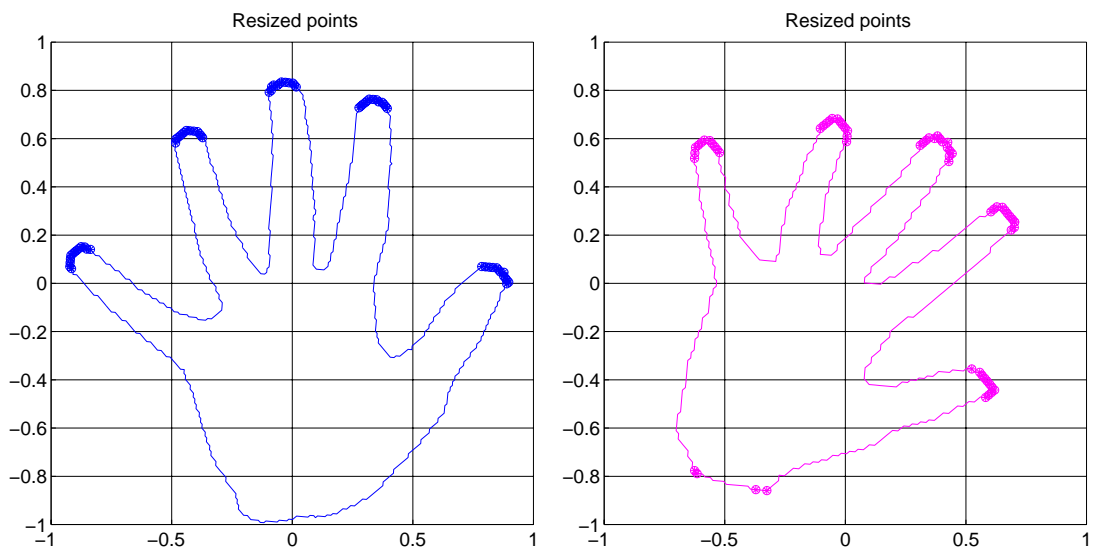


Abbildung 4.4: Resampling der Konturkette von Gesten "Stop Open" durch größten Krümmung Kriterium

**Äquidistanten Abtastung** Bei dem Algorithmus werden Punkte in gleichem Abstand voneinander ausgewählt. Das führt zu unterschiedlichen Abständen zwischen den Punkten je nach der Länge der Kette. Auf diese Weise wird die Kontur geglättet und manche Unregelmäßigkeiten werden dadurch vermieden (Abb. 4.5).

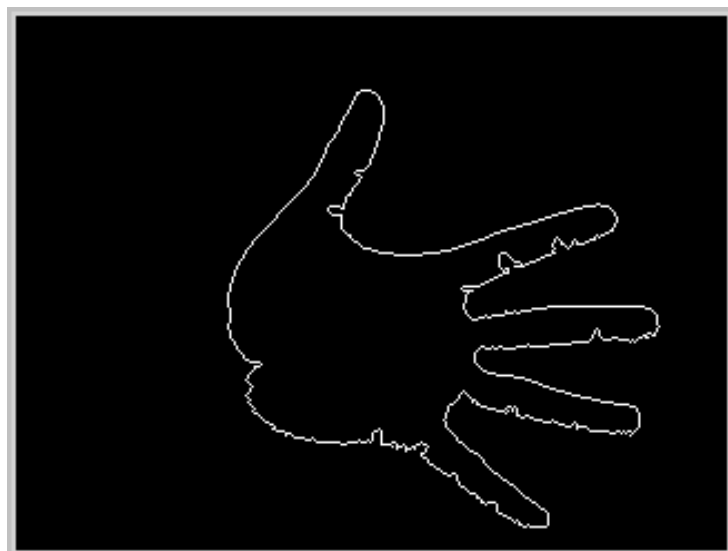


Abbildung 4.5: Beispiel einer Kontur der durch die Resampling geglättet wird.

Da der zweite Algorithmus besser mit Rauschen umzugehen scheint, würde dieser beibehalten. Die Ergebnisse der Abtastung sind in Abb. 4.6 und 4.7 dargestellt.

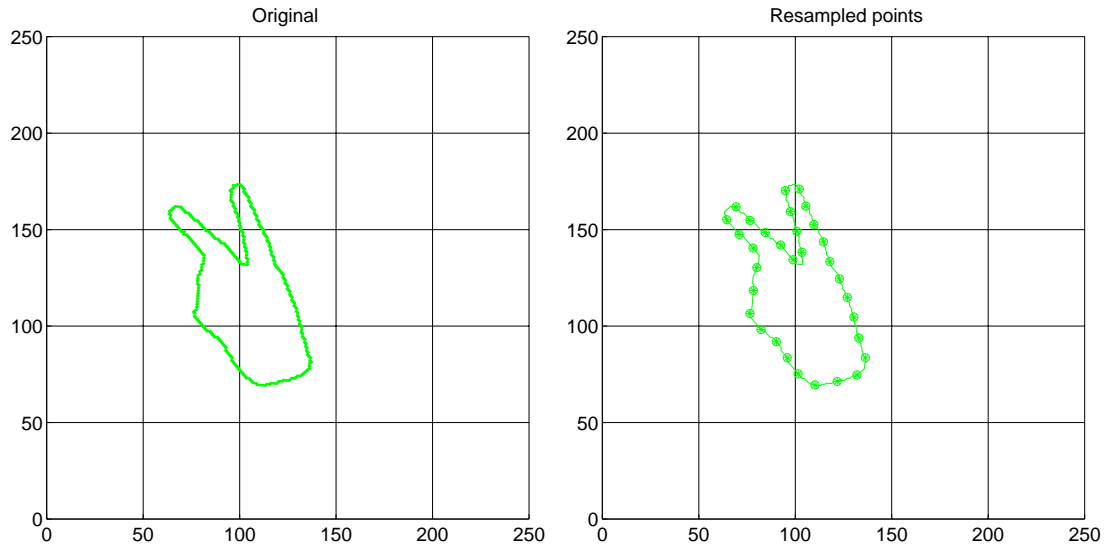


Abbildung 4.6: Resampling der Konturkette von Gesten: “Victory A”.

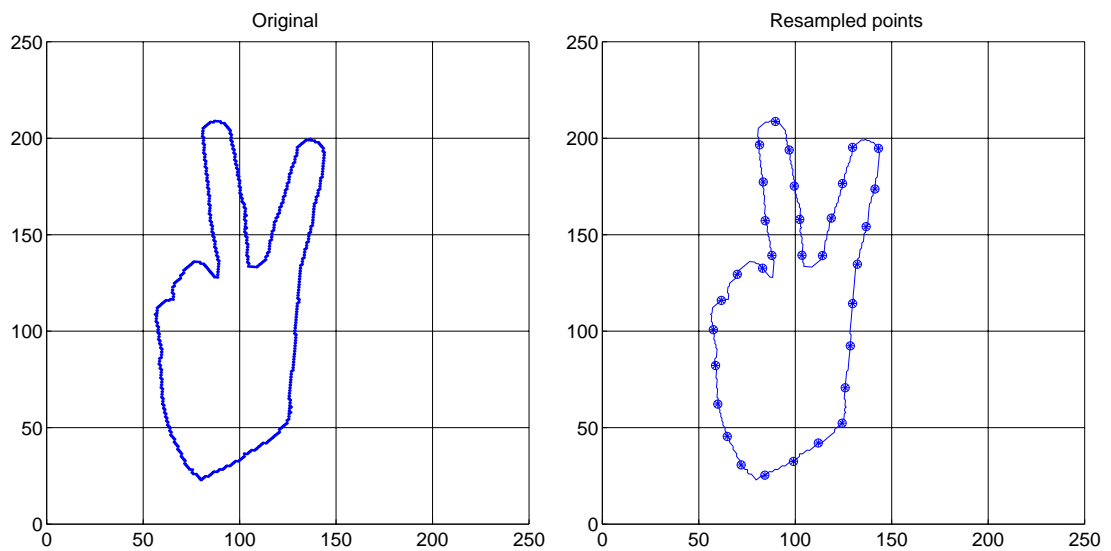


Abbildung 4.7: Resampling der Konturkette von Gesten: “Victory B”.

#### 4.2.1.3 Anzahl der abzutasteten Punkte

Entsprechend den Überlegungen in Kapitel 4.2.1.1, wurden für diesen Programm verschiedene Anzahl von Punkte extrahiert, nämlich 8, 16, 32 und 64, um die beste

zu finden. Der von die benutzte Gestentypen, der mehr Punkte braucht um erkannt zu werden ist der mit den höchsten Proportion Konturlänge / Fläche. Disetwegen wird hier die Geste “Stop Open” als Beispiel benutzt.

Weniger als 8 Punkte beschrenkt die Information von der Kontur erhalten. Mit 8 Punkten wären die Handgesten auch für die menschliche Schwahrnehmung nicht mehr sehr deutlich zu erkennen (siehe Abb. 4.8). Die Resampling Methode hier benutzt fängt immer die Abtastung beim ersten Punkt der Reihenfolge. Deshalb hängt die Lage der abgetastete Punkte von die Anfangspunkt der Kontur ab, wie in den folgenden Abbildungen zu sehen ist. Der Anfangspunkt ist durch die Kontur-extraktion Algorithmus festgestellt (siehe letzte Punkt in Kapitel 4.1.1). Deswegen kann der gleichen Gestentyp verschiedene Anfangspunkte, je nach der Orientierung des Benutzerhands als das Bild aufgenommen war, haben. Die Ergebnisse sind von den Anfangspunkt der Konturkette beeinflußt und diese Einfluß ist je größer desto weniger Punkte abgetastet werden.

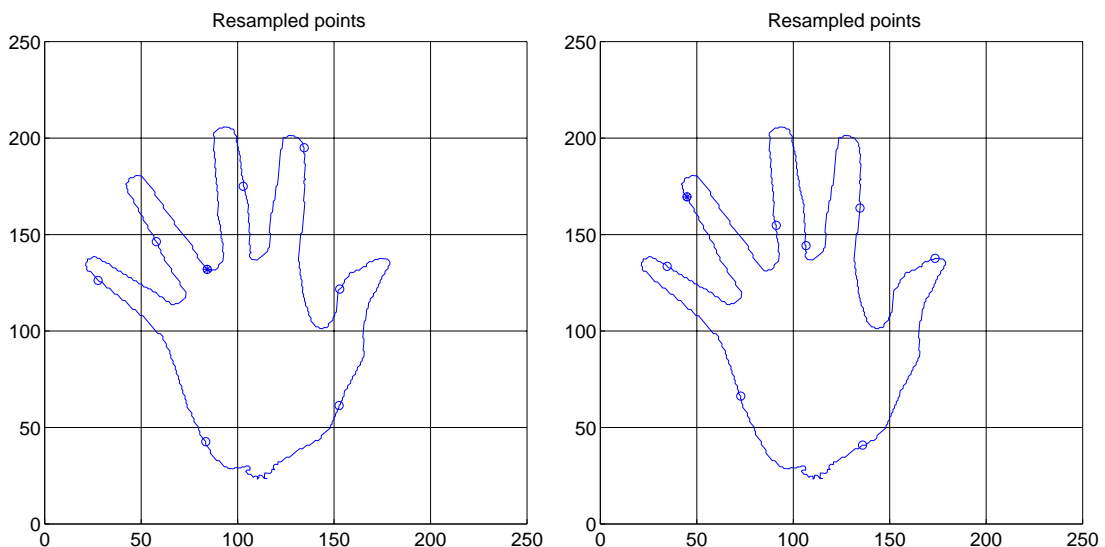


Abbildung 4.8: Resampling der Konturkette von Gesten “Stop Open” mit 8 Punkte und unterschiedliche Anfangspunkt.

Dieses Einfluß der Anfangspunkt ist schon kleiner bei der Abtastung von 16 Punkte (Abb. 4.10). Um die Einfluß die Abtastung an die Ergebnisse zu sehen, werden hier manche Abbildungen davon gezeigt. Hier wurden verschiedene Deskriptoren von Gestentyp “Stop Open” mit alle Modelle im Bibliothek enthalten. Es gibt so viele Balken auf jede Graphik, wie Gestentypen im Modellbibliothek. Ein positives Ergebnis der Vergleich wird erreicht wenn die Balke der richtige Modellgeste klein ist und höchstens unterschiedlich zu den anderen Balken. Die Bedeutung diese Graphiken wird in Kapitel 4.3.1 ausführlich erklärt.

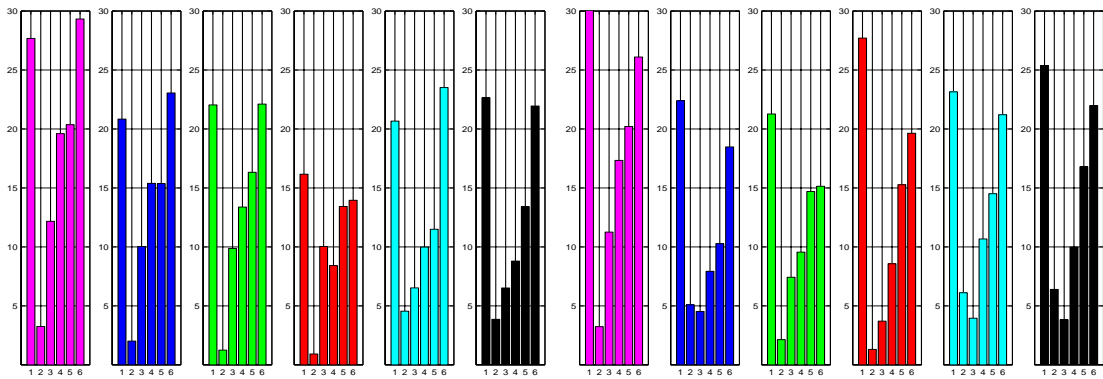


Abbildung 4.9: Ergebnisse der Gesten Erkennung von Gesten “Stop Open” mit 8 Punkte. Links mit zufällig gleichen Anfangspunkt und rechts mit unterschiedlichen Anfangspunkt.

Die Einfluß der Anfangspunkt ist schon kleiner bei der Abtastung von 16 Punkte (Abb. 4.10). In Abb. 4.11 sind die Ergebnisse deutlich besser als die in Abb. 4.9.

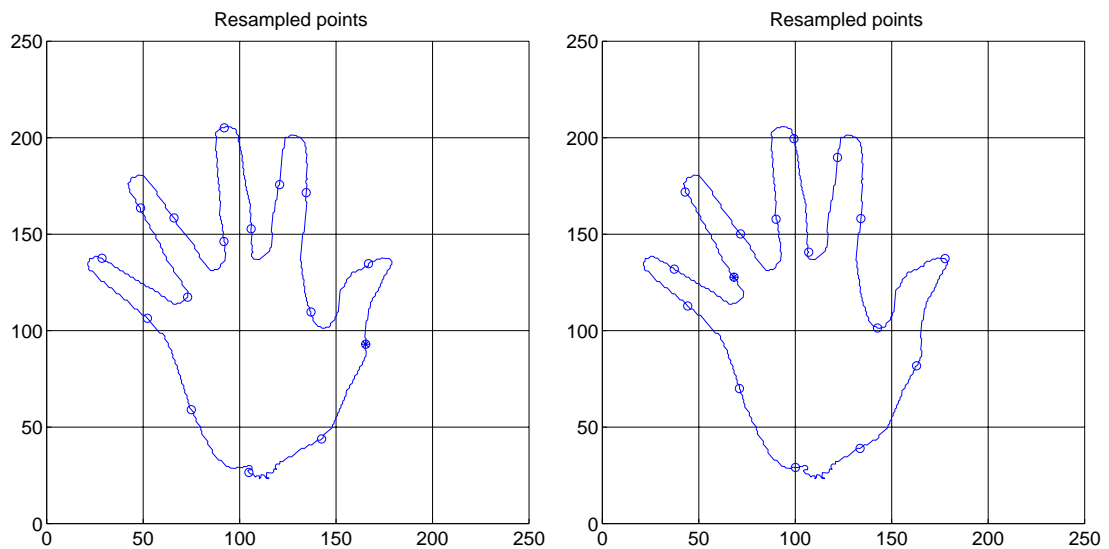


Abbildung 4.10: Resampling der Konturkette von Gesten “Stop Open” mit 16 Punkte und unterschiedliche Anfangspunkt.

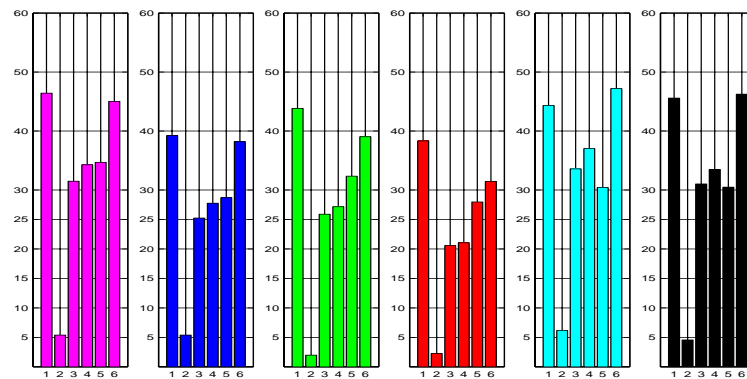


Abbildung 4.11: Ergebnisse der Gesten Erkennung von Gesten “Stop Open” mit 16 Punkte und unterschiedliche Anfangspunkt.

In den folgenden Abbildungen werden die Resampled Konturen und Ergebnisse für 32 abgetastete Punkte dargestellt. Die Unterschied zwischen diese und die mit 16 ist schon gering, aber trotzdem sind die mit 32 ein bißchen besser. Der Anfangspunkt in der Abtastung hat hier überhaupt keine Einfluß.

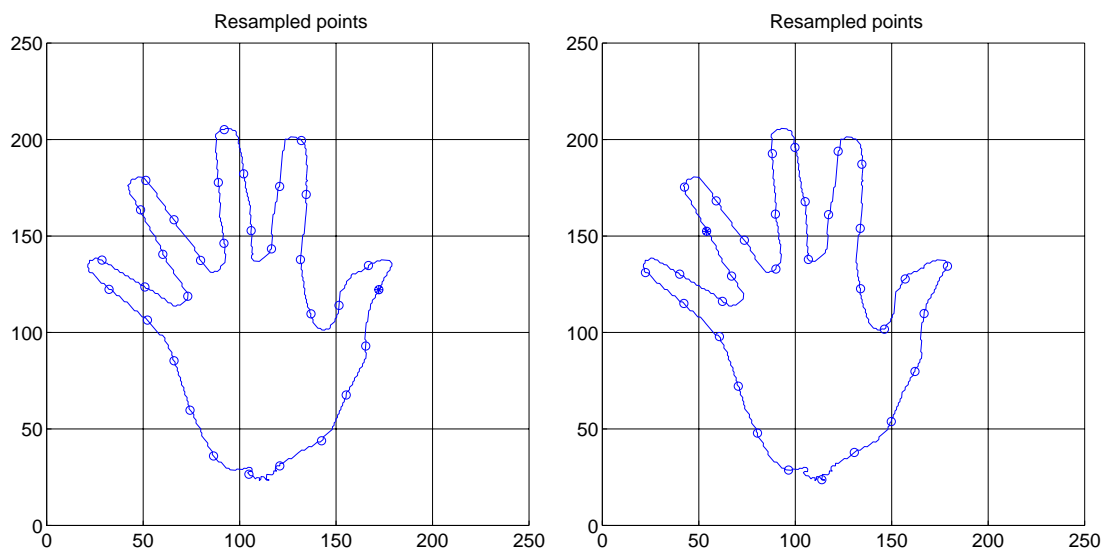


Abbildung 4.12: Resampling der Konturkette von Gesten “Stop Open” mit 32 Punkte und unterschiedliche Anfangspunkt.

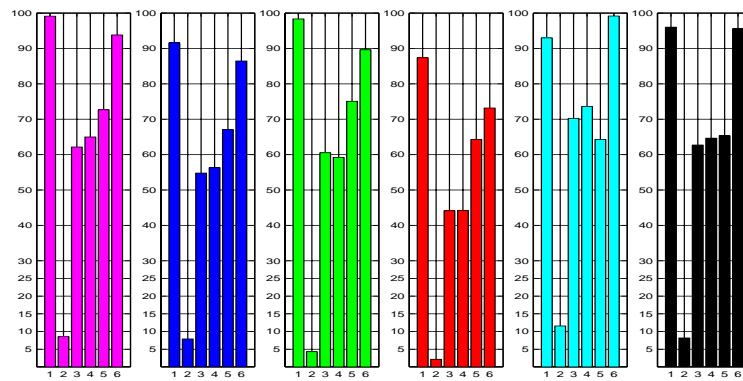


Abbildung 4.13: Ergebnisse der Gesten Erkennung von Gesten “Stop Open” mit 32 Punkte und unterschiedliche Anfangspunkt.

In dem Fall von 64 abgetastete Punkte, gibt es auch keine Einfluß. Wenn die Graphiken in Abb. 4.13 mit 64 Punkte mit den Graphiken in Abb. 4.15 verglichen sind, gibt es, proportionell betrachtet, keine Unterschied zwischen die beide, deswegen ist 32 die ideale Menge von abzutastenden Konturpunkte.

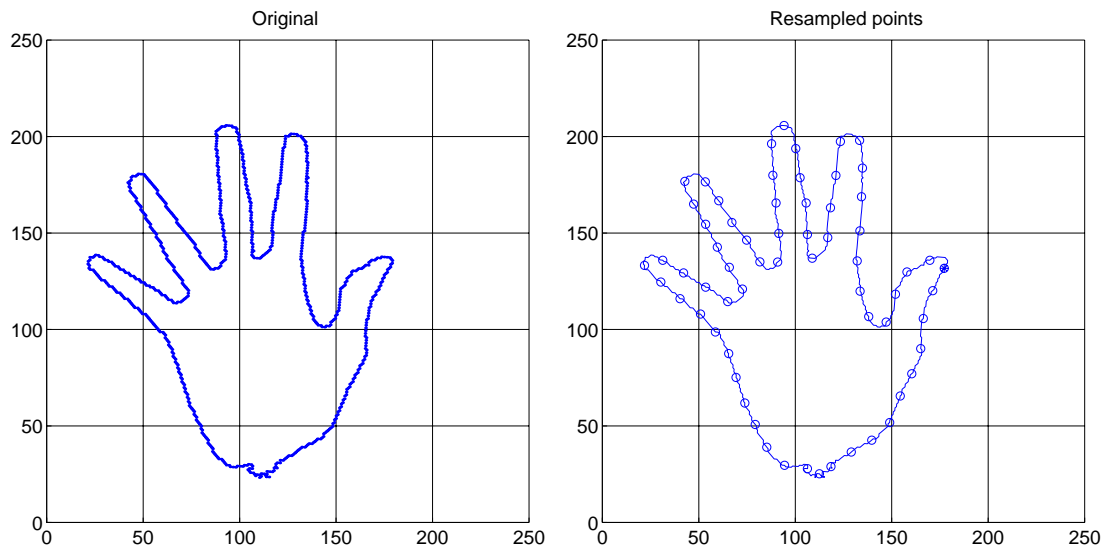


Abbildung 4.14: Links: Originalkontur; Rechts: Resampling der Konturkette von Gesten “Stop Open” mit 64 Punkte.

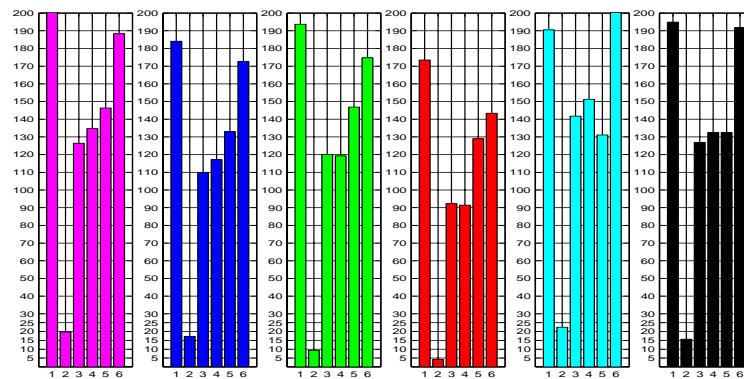


Abbildung 4.15: Ergebnisse der Gesten Erkennung von Gesten “Stop Open” mit 64 Punkte und unterschiedliche Anfangspunkt.

### 4.2.2 Anwendung der Fourier-Transformation

Wenige Fourier-Deskriptoren reichen aus um das Wesentliche einer Kontur zu erfassen. Dies ist eine wertvolle Eigenschaft, da die Fourier-Deskriptoren Formdaten enthalten. Dadurch besteht die Möglichkeit aufgrund der entsprechenden Deskriptoren zwischen ausgeprägten Formen zu unterscheiden.

Nachdem die Konturketten nun alle die gleiche Anzahl von Punkten haben ist es möglich die Fourier-Transformation durchzuführen. Die so enthaltenen Deskriptoren sollten möglichst invariant sein bezüglich der Umsetzung der Geste, Skalierung und Rotation. Eine weitere Bedingung ist der Anfangspunkt der Vektoren. Fourier-Deskriptoren sind nicht unempfindlich gegenüber geometrischen Aspekten, aber es ist möglich die Deskriptoren nach diesen Eigenschaften zu normieren. Allerdings ist es einfacher die Normierung von Skalierung und Verschiebung vor der Fourier-Transformation zu berechnen. Daher werden sie bereits vor der Transformation normiert. Die Normierung wird durchgeführt entsprechend des Einflusses der zugehörigen Eigenschaft auf die Original-Konturkette und ihrer Beziehung mit dem Einfluß der gleichen Eigenschaft auf die resultierenden Deskriptoren. Die Eigenschaften der Fourier-Deskriptoren hängen von den Eigenschaften der Fast-Fourier-Transformation ab, die lautet:

$$X(n) = \sum_{k=0}^{N-1} x(k) \cdot e^{-j2\pi(\frac{k \cdot n}{N})} \tag{4.1}$$

mit

$$k \rightarrow x(k) | k \in Z, x(k) \in \mathbf{C} \tag{4.2}$$

wobei “x” der Vektor ist, dessen Elemente die Konturpunkte sind; “X” ist der Ausgabevektor nach der Fast-Fourier-Transformation von “x”; “N” ist die Anzahl der Punkte in “x”; “k” stellt die einzelnen Elemente von “x” dar; “n” stellt die einzelnen

Elemente von “X” dar.

Bei der Darstellung von Fourier-Deskriptoren haben wir die Eigenschaften die für diesen Algorithmus wichtig sind bereits dargestellt (siehe Kapitel 3). Tabelle 4.1 zeigt diese Eigenschaften noch einmal in einer Übersicht:

Transformation	Kontur	Fourier Deskriptor
Identität	$s(k)$	$S(n)$
Rotation	$s_r(k) = s(k)e^{j\theta}$	$S_r(n) = S(n)e^{j\theta}$
Verschiebung	$s_t(k) = s(k) + \Delta_{xy}$	$S_t(n) = S(n) + \Delta_{xy}\delta(n)$
Skalierung	$s_s(k) = \alpha s(k)$	$S_s(n) = \alpha S(n)$
Anfangspunkt	$s_p(k) = s(k - k_0)$	$S_p(n) = S(n)e^{-j2\pi k_0 n/N}$

Tabelle 4.1: Vergleichstabelle des Einflusses der geometrischen Eigenschaften

Die dargelegten Beziehungen geben an, wie die Fourier-Deskriptoren normalisiert werden müssen um einen Vergleich auszuführen. Die Rotation und Skalierung einer Kontur wirken direkt proportional auf den ganzen Fourier-Deskriptoren-Vektor, wobei eine Verschiebung und eine Änderung im Anfangspunkt unterschiedlichen Einfluß auf jedes Element des Fourier-Deskriptoren-Vektors haben. Das führt zu Schwierigkeiten beim Normalisieren der letzten zwei Parameter.

#### 4.2.2.1 Vornormierung der Konturkette

Die Vornormierung von bestimmten Eigenschaften führt dazu, dass die entsprechende Eigenschaft keinen Einfluß mehr auf die zu berechnenden Deskriptoren haben. Die geometrischen Eigenschaften, die im Verlauf dieses Prozesses betrachtet werden sind Verschiebung und Skalierung. Die Gesten in die Abbildungen die folgen sind die gleichen die in Abb. 4.6 und Abb. 4.7 abgetastet waren.

##### Normierung bezüglich Verschiebung :

Um den Fourier-Deskriptoren im Fall von Verschiebung zu normalisieren, ist es hilfreich die Größe der Verschiebung zu kennen sowie den Vektor  $\delta(u)$  (wie in Tabelle 4.1 angegeben). Dieser Vektor kann unterschiedlich je nach der Größe der Vektoren sein. Es ist einfacher die entsprechende Normierung auf der Kontur zu realisieren. Bei dieser Vorgehensweise werden alle Konturpunkte verschoben, damit der Schwerpunkt der Kette am Koordinatenursprung liegt.

In Abb. 4.16 und 4.17 sind zwei Gesten zu sehen, die von der ursprünglichen Position (im linken Bild) zum Ursprung des Koordinatensysteme (im rechten Bild) verschoben wurden.

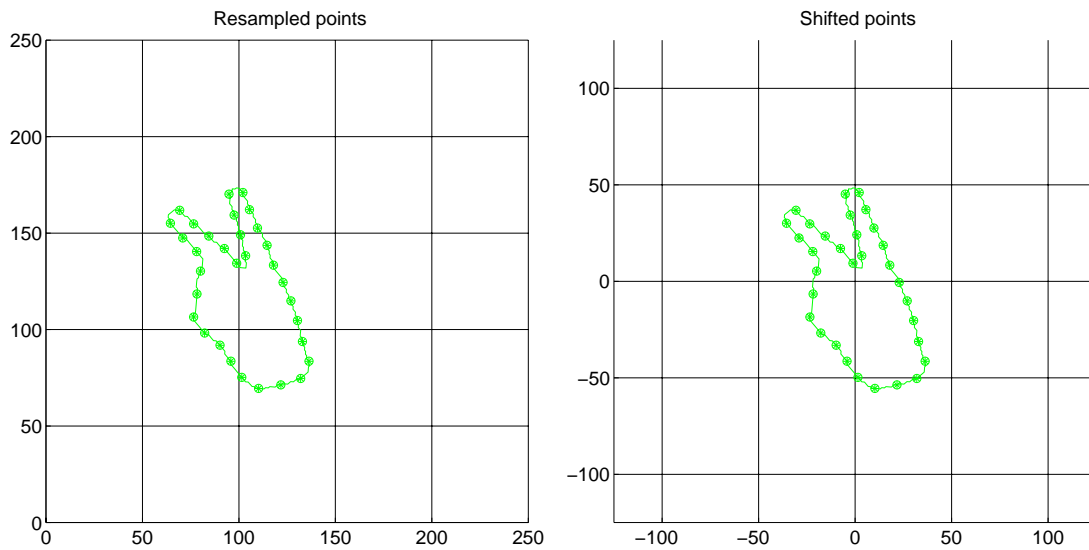


Abbildung 4.16: Normierung bezüglich Verschiebung der Geste "Victory A".

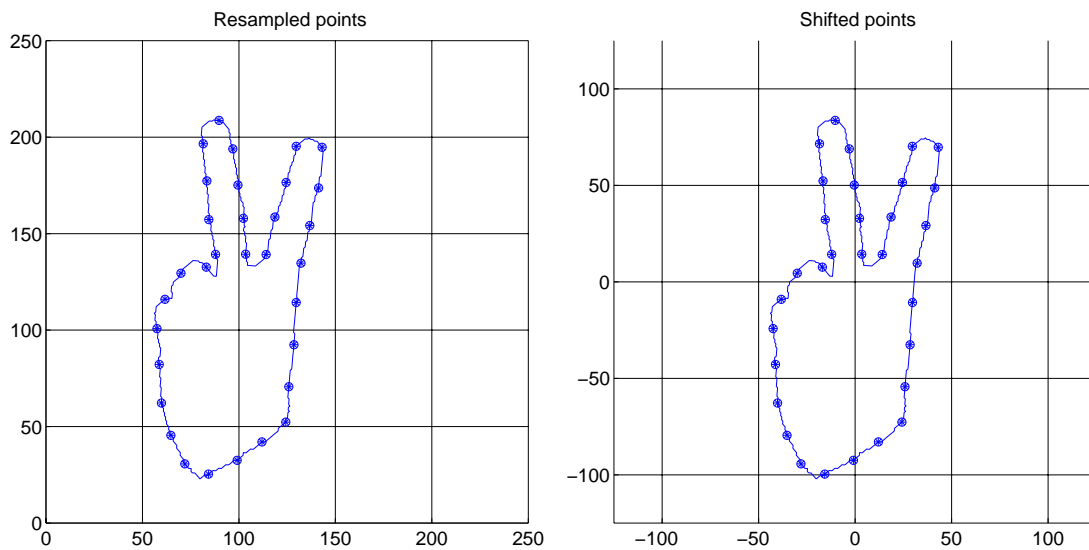


Abbildung 4.17: Normierung bezüglich Verschiebung der Geste "Victory B".

### Normierung bezüglich Skalierung :

Die andere Eigenschaft, die vorzunormieren ist, ist die Skalierung. Deren Einfluß sowohl auf die ursprüngliche Konturkette als auch auf den Deskriptor, besteht jeweils aus dem gleichen Faktor. Es wäre auch möglich die Skalierung nach Berechnung des Deskriptors zu realisieren, aber hier werden vor der Transformation alle Elemente der Konturkette durch die längste Distanz zwischen der Schwerpunktvektor und Kontur geteilt.

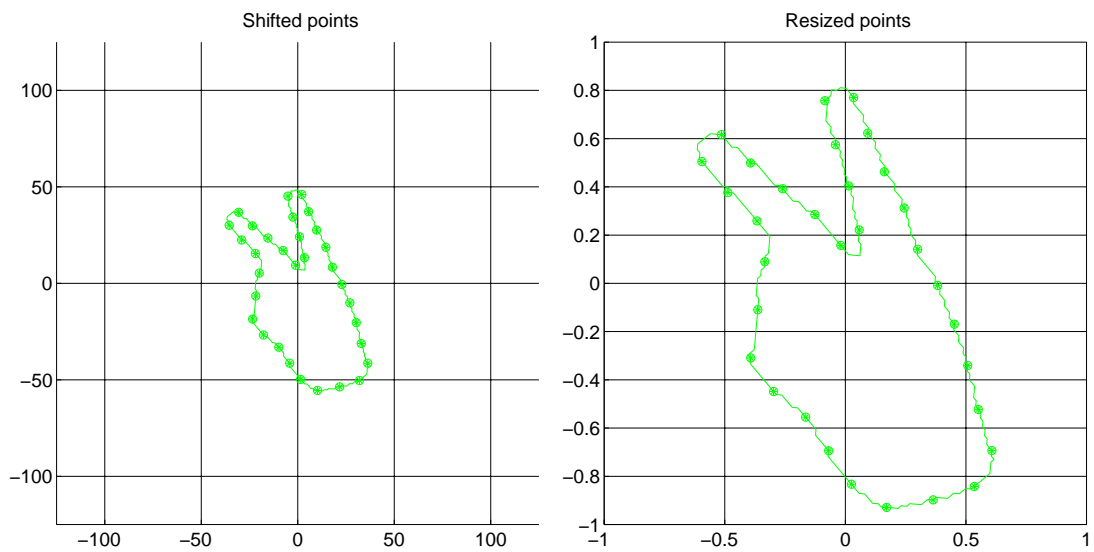


Abbildung 4.18: Normierung bezüglich Skalierung der Geste "Victory A".

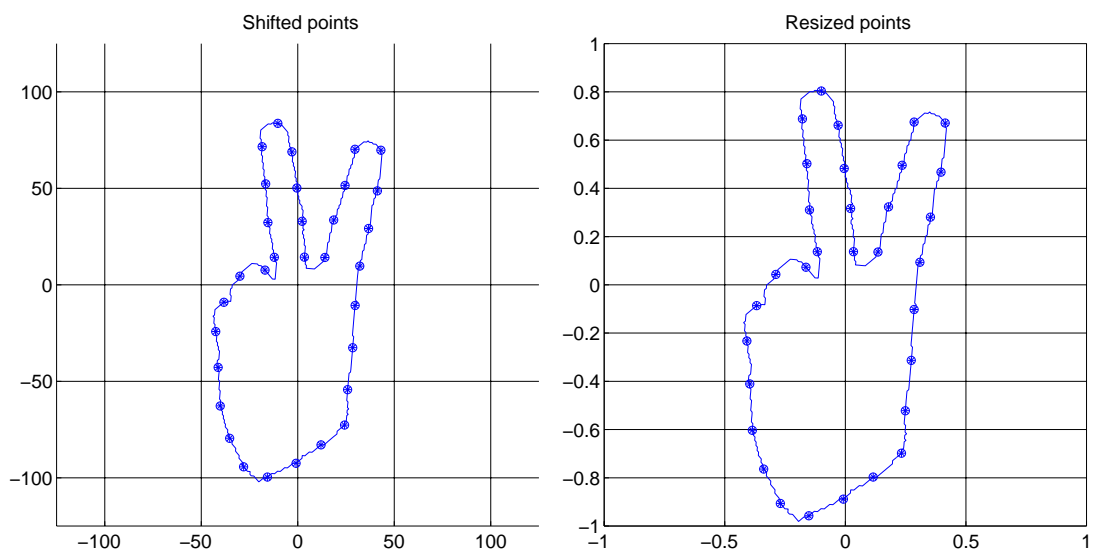


Abbildung 4.19: Normierung bezüglich Skalierung der Geste "Victory B".

Nach diesen beiden Vornormierungs-Operationen haben alle Hände, die die gleiche Geste ausführen, ungefähr die gleiche Größe und sind zentriert. Dadurch fallen die Konturen jetzt bis auf die Orientierung zusammen.

#### 4.2.2.2 Berechnung der Deskriptoren

Jetzt kann auf der aus der Vornormierung enthaltenen Konturkette die FFT ausgeführt werden. Um den Rechenaufwand zu reduzieren, werden die Punkte als

komplexe Zahlen betrachtet. Damit wird die FFT nicht zwei Mal berechnet, einmal für die “x”-Dimension und einmal für die “y”-Dimension, sondern direkt auf der komplexen Reihenfolge 4.3.

$$\{s(k)\} = \{x(k)\} + j \cdot \{y(k)\} \quad (4.3)$$

Das Ergebniss der Anwendung der FFT auf einen komplexen 32-elementigen Vektor ist ein neuer komplexer Vektor der gleichen Dimension, dessen Elemente sind:

$$S(n) = S_x(n) + j \cdot S_y(n) \quad (4.4)$$

In Abb. 4.20 werden die Deskriptoren verschiedener Eingabebilder der gleichen Geste, die von unterschiedlichen Hände vorgeführt wurden dargestellt. Für jedes Muster wird eine andere Farbe verwendet. Die Verteilung der Punkte folgt der gleichen Struktur bis auf den Drehwinkel.

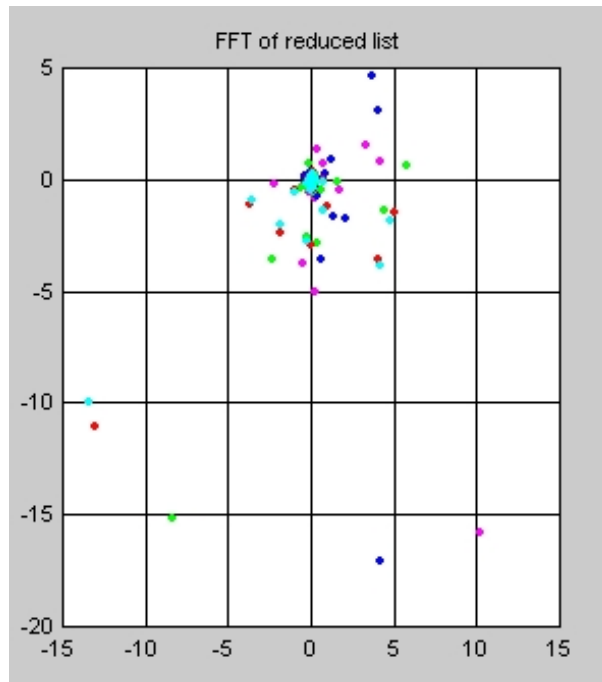


Abbildung 4.20: Fourier-Deskriptoren von verschiedenen Händen mit der gleichen Geste.

### 4.2.2.3 Normierung der Deskriptoren

Bis zu diesem Schritt wurden allerdings noch keine vergleichbare Folgen von Elementen erricht, die bei gleicher Geste genügend Ähnlichkeit aufweisen und bei unterschiedlichen Gesten hinreichend verschieden sind. Hierzu müssen noch die beiden erwähnten geometrischen Eigenschaften normiert werden, nämlich die Rotation des

Bildes und der Anfangspunkt der Konturkette.

Wie in Tabelle 4.1, wo die Eigenschaften der Beziehungen angegeben sind, zu sehen ist, beeinflussen diese beide Eigenschaften nur den Winkel der Fourier-Deskriptoren, aber nicht deren Abstand. Um den entsprechenden Einfluß zum Verschwinden zu bringen werden die Deskriptoren so gewählt, daß die Elemente den Betrag der bisherigen Vektoren darstellen. ((Die Elemente des neuen Deskriptors werden mit “D(n)” bezeichnet.))

$$D(n) = |S(n)| = \sqrt{S_x^2(n) + S_y^2(n)} \quad (4.5)$$

Die Ergebnisse werden in einem Diagramm aufgetragen wobei die “x” Achse die Anzahl der Elemente der Fourier-Deskriptoren angibt (d.h. von 1 bis 32) läuft, und auf der “y” Achse werden die Werte der entsprechenden Elemente aufgetragen. Die in Abb. 4.20 gezeigten Fourier-Deskriptoren besitzen dann die in Abb. 4.21 berechneten Abstände.

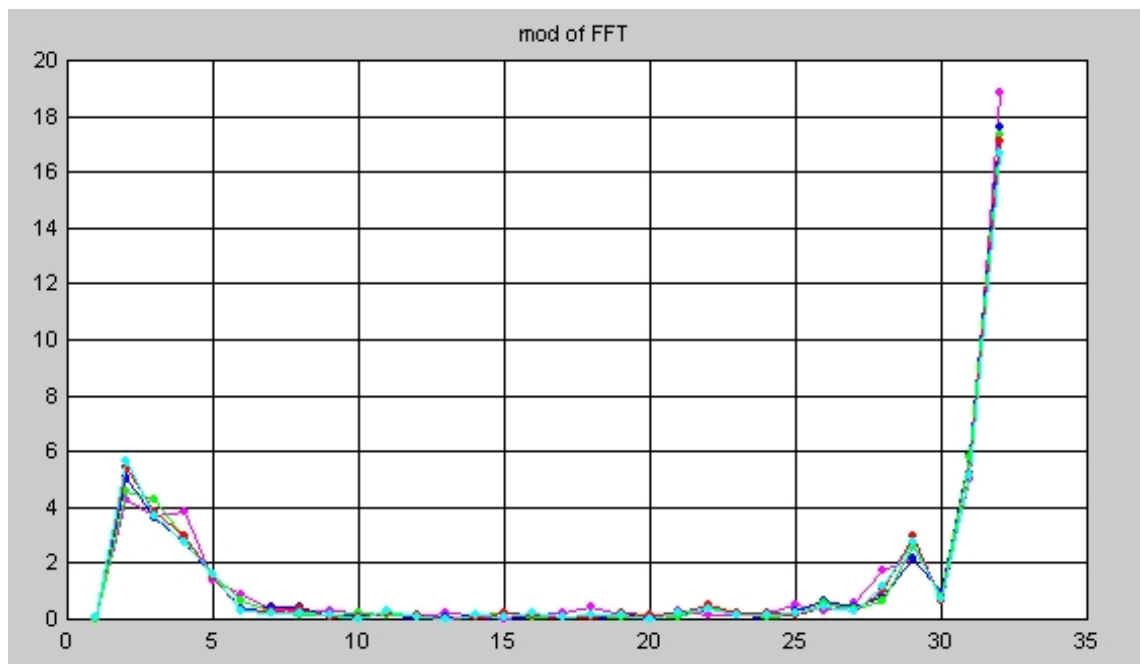


Abbildung 4.21: Abstände von Fourier-Deskriptoren von verschiedenen Hände bei der gleichen Geste.

### 4.3 Vergleichungs- und Klassifikationsalgorithmus

Nach Vorverarbeitung und Merkmalsextraktion wird die Klassifizierungsphase erreicht. Das Klassifikationskriterium besteht aus drei wichtigen Punkten, die im folgenden diskutiert werden. Das erste ist ein Kriterium, um die Deskriptoren mit

den Modelldeskriptoren zu vergleichen, dann ein Kriterium zu entscheiden ob zwei Gesten ähnlich genug sind, und eins wie die Modelldeskriptoren aufgebaut werden.

Hierzu genügt ein einfacher Klassifizierungsalgorithmus, damit das Programm weniger Speicher belegt und schneller laufen kann. Für die vorliegende Arbeit wurde der “Minimum Distance Classifier” (in [González 93] vorgeschlagen) implementiert.

### 4.3.1 Vergleichskriterium

Für die menschliche Wahrnehmung ist es einfach zu unterscheiden, welchem Gestentyp eine zu klassifizierende Geste angehört.

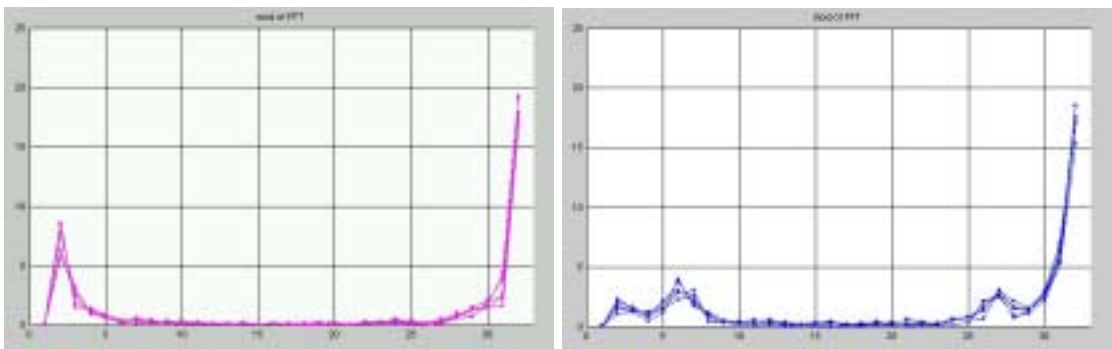


Abbildung 4.22: Modulen von Fourier-Deskriptoren vom Gestentyp “Stop Open” (links) und “Zeige” (rechts).

Da der Roboter nicht wie menschliche Verstand funktioniert, ist die Ähnlichkeit mit einem quantitativen Wert zu bestimmen.

Als Ähnlichkeitsmaß wird hier die Euklidische Distanz berechnet, um den “Minimum Distance Classifier” Algorithmus zu implementieren. Jedes Element ( $D(n)$ ) wird von dem entsprechenden Element des Modelldeskriptors ( $D_M(n)$ ) abgezogen und die erhaltenen Werte werden quadriert um den absoluten Unterschied betrachten zu können. Man erhält so viele Werte wie der Deskriptor hat, die erhöht geben eine Maß ( $\Delta$ ) von die Ähnlichkeit die verglichene Deskriptoren.

$$\Delta = \sum_{n=0}^{N-1} \sqrt{D_M^2(n) - D^2(n)} \quad (4.6)$$

In Abb.4.23 sind 8 Diagramme dargestellt, die man nach dem Vergleich einer vorgeführten Gest mit den Modelldeskriptoren der Bibliothek erhält. Jede Graphik enthält so viele Balken wie Anzahl der Modellgesten im Bibliothek, sieben in diesem Beispiel. Die Balken repräsentieren den analog zum vorherigen Abschnitt berechneten Wert für das Ähnlichkeitsmaß. Je ähnlicher die vorgeführte Geste zum Modell

ist, desto kleiner wird der entsprechende Balken. In diesem Fall gehören die 8 zu klassifizierenden Gesten zum gleichen Gestentyp. Deshalb ist die kleinste Spalte immer beim gleichen Modell (Nummer 1).

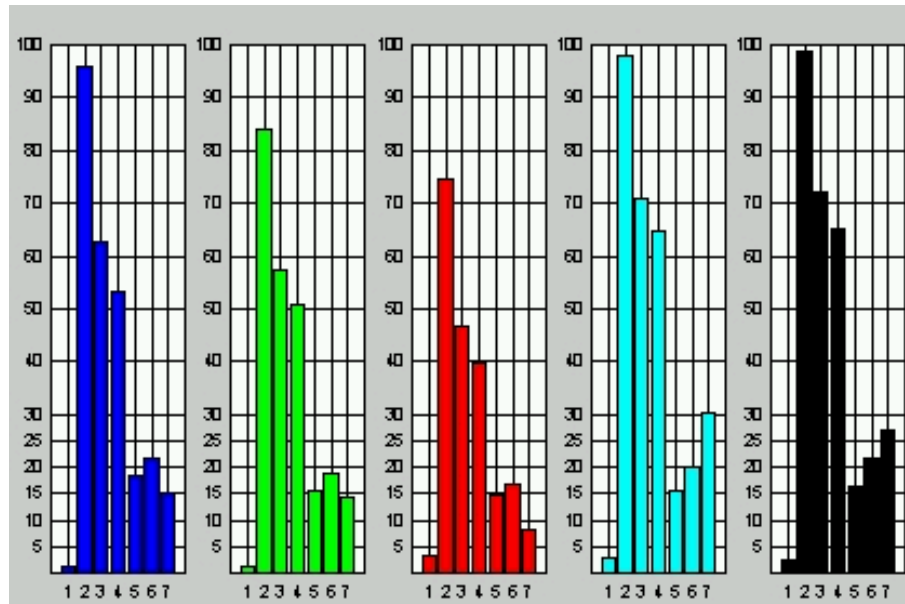


Abbildung 4.23: Euklidische Distanz zwischen den Deskriptoren der vorgeführten Gesten und den Modelldeskriptoren.

### 4.3.2 Entscheidungskriterium

Als erster Schritt in der Klassifikation der Ergebnisse wird der Modelldeskriptor mit der kleinsten Distanz zur vorgeführten Geste als bestmöglicher passender Typ angenommen. Diese Entscheidung gilt allerdings nur als vorläufig, da es auch vorkommen kann, daß die zu klassifizierende Geste in der benutzten Bibliothek gar nicht enthalten ist. In diesem Fall soll die Geste als unerkannt klassifiziert werden.

Daher muß nach dieser ersten provisorischen Klassifikation ein zweiter Schritt durchgeführt werden, um zu bestimmen ob die Euklidische Distanz zwischen den verglichenen Deskriptoren klein genug ist, um die beiden als gleiche Geste zu betrachten. Dafür wird es Schwellenwert durch empirische Beobachtung festgestellt. Dieser Schwellenwert gibt eine Maß des Unterschiedes zwischen dem Deskriptor eine Geste und dem Modell dieses Gestentyps.

Nach der Festsetzung des Grenzwert kann die endgültige Entscheidung getroffen werden. Wenn die berechnete Diferenz den Schwellenwert überschreitet, wird die Geste als unerkannt klassifiziert. Sonst wird die vorläufige Klassifikation als korrekt angenommen.

Abb.4.24 zeigt Diagramme für die Euklidische Diferenz von in der Bibliothek nicht enthaltenen vorgeführten Gesten. Die vorgeführten Handgesten sind links und die entsprechenden Diferenzen rechts dargestellt. Wenn man Abb.4.23 und Abb. 4.24 vergleicht, kann man feststellen daß im zweiten Fall die niedrigsten Balken wesentlich höher sind als die niedrigsten Spalten in Abb. 4.23.

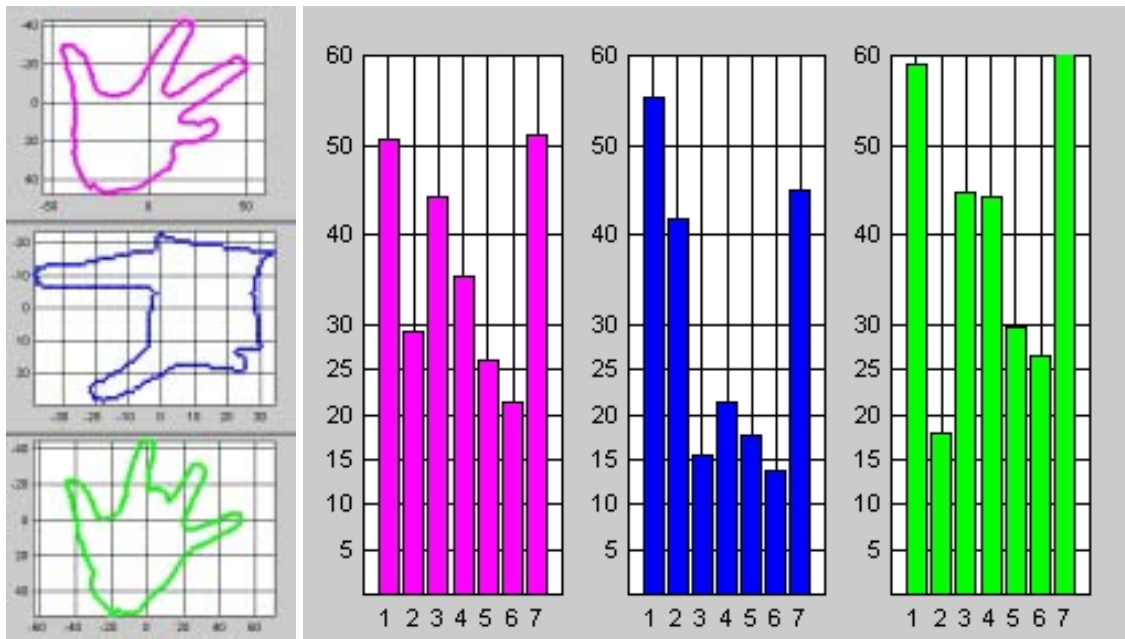


Abbildung 4.24: Euklidische Distanz für in der Bibliothek nicht enthaltene Gesten.

### 4.3.3 Modellaufbau

Um einen guten Ablauf des Programmes zu ermöglichen ist die Verwendung einer adäquaten Bibliothek mit Referenzgesten erforderlich. Dies bedeutet, daß eine Bibliothek für jeden Benutzer des Roboters funktionieren sollte, unabhängig von der Handgeometrie des entsprechenden Benutzers. Sie sollte auch unter den in Kapitel 1 dargestellten funktionalen Bedingungen anwendbar sein. Daher ist die Wahl eines guten Modells sehr wichtig.

In dieser Arbeit wurden zwei Methoden getestet, nämlich “Bestes Muster” und “Mittelwert-Deskriptor”. Für beide Methoden ist es nützlich verschiedene Bilder von verschiedenen Hände und gleichen Gesten abzutasten und die entsprechenden Merkmale in einen Deskriptor zu extrahieren. Mit diesen Deskriptoren wird der Modelldeskriptor gebaut.

**Bestes Muster** Hier werden alle Muster-Deskriptoren miteinander verglichen. Die Euklidischen Distanzen von einen Deskriptor zu allen anderen werden addiert,

was einen Wert ergibt, der den Unterschied zu allen anderen Deskriptoren quantifiziert. Dann wird als Modell diejenige Geste genommen, deren Gesamtunterschied zu allen anderen am kleinsten ist.

**Mittelwert-Deskriptor** Bei der zweiten Methode werden alle Deskriptoren für das Modell benutzt. Der Durchschnittswert von allen Deskriptoren wird berechnet, und dadurch auch der Modell-Deskriptor. Anders gesagt, jedes Element des Modelldeskriptor wird berechnet als Mittelwert der entsprechenden Elemente von allen abgetasteten Gestendeskriptoren.

Bei der ersten Methode die Werte die der Modelldeskriptor enthalten können sich auf die extreme die Kurven finden. Deswegen wenn die Ähnlichkeitsmaß mit diesen Deskriptoren berechnet werden, die werden größer sein als mit der zweiten Methode wenn die mit Deskriptoren deren Werte sich gegenüber in die Kurve finden. Die zweite Methode hat in Tests bessere Ergebnisse gezeigt. Sie behält die Information verschiedener Handgeometrien bei. Dadurch sind die Ähnlichkeitsmasse besser verteilt und das

# Kapitel 5

## Experimente und Evaluation

Nach der Entwicklung des selbständigen Gestenerkennungsprogrammes muß es getestet und in den Roboter implementiert werden. Auch die Ergebnisse des Programmes werden nach der Implementierung evaluiert. Die zur Überprüfung des Programmablaufs durchgeführten Experimente verwenden die in Abb. 5.1 dargestellten Gestetypen, sowohl für die Evaluation als auch für die Implementierung. Die Gesten sollen deutlich verschieden sein. Da die Kontur der Benutzerhand als grundlegendes Klassifikationsmerkmal benutzt wird, können bei ähnlichen Silhouetten schlecht Unterscheidungen getroffen werden. Beispielsweise ist eine Zeigegeste einer Hand mit einem breiten Zeigefinger kaum von einer Zeigegeste mit zwei schmalen Finger zu unterscheiden.

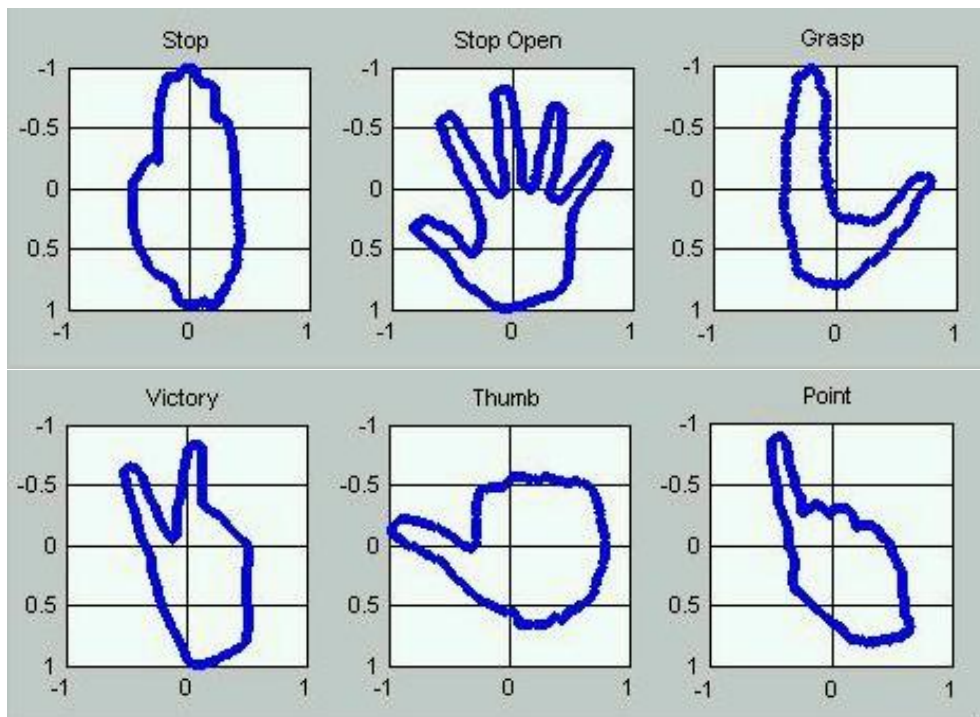


Abbildung 5.1: In der Experimentsbibliothek verwendete Gestentypen.

Zuerst werden verschiedene Bilder mit der Kamera aufgenommen, die von verschiedenen Leute vorgeführt werden, um die Robustheit zu überprüfen. Es werden auch Gesten aufgenommen, die nicht den Typen in Abb. 5.1 angehören. Dadurch können alle Möglichkeiten der im folgenden dargestellten Klassifikation überprüft werden.

- In der Bibliothek vorhandene Gesten:

**Positives Ergebnis :**

- Richtig klassifizierte Geste.

**Negatives Ergebnis :**

- Falsch klassifizierte Geste.
- Unerkannte Geste.

- Nicht zur Bibliothek gehörige Gesten:

**Positives Ergebnis :**

- Unerkannte Geste.

**Negatives Ergebnis :**

- Falsch klassifizierte Geste.

((meter aqui un diagrama)) ((meter aqui un diagrama))

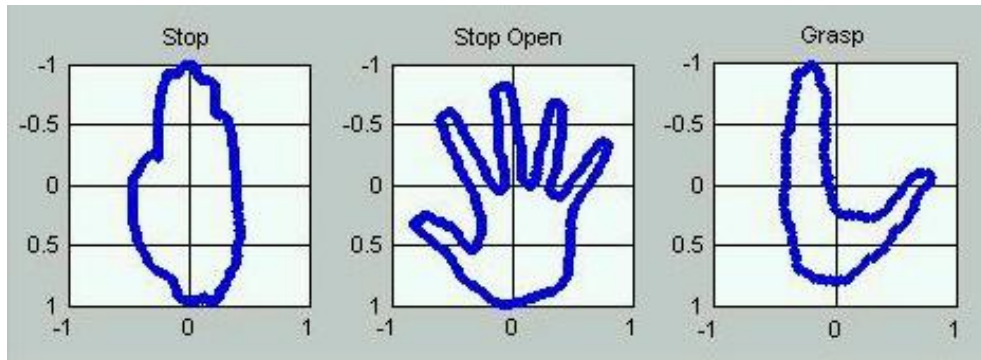


Abbildung 5.2: Cambiar las fotos por un futuro diagrama de flujo

## 5.1 Evaluation des Systems als selbständige Anwendung

Vor der Implementierung des Programmes im Projekt GOA (Kapitel ??) wurde die Leistung des zu implementierenden Algorithmus mit Matlab getestet, da Matlab umfangreiche Funktionen für die Verarbeitung der Daten und ihrer Visualisierung bereit stellt.

Eine Bibliothek wurde aufgebaut wie in Kapitel 4.3.3 beschrieben. Die beiden Algorithmen aus Kapitel 4.3.3 werden hier getestet und verglichen, ebenso wie der Ablauf des Programms mit der ausgewählten Bibliothek. Die Evaluation erfolgt nur bezüglich der Korrektheit des Ergebnisses. Die Laufzeit des in Matlab implementierten Programms ist wenig aussagekräftig, da die eigentliche Implementierung auf dem Roboter in C erfolgen soll. Daher werden hier nur die Ergebnisse der Klassifikation ausgewertet.

Zuerst werden die Bilder mit dem Kamerakopf aufgenommen und durch eine in GOA implementierte Schnittstelle vorverarbeitet und die Konturkette extrahiert. Über die gleiche Schnittstelle werden diese Konturketten in Dateien gespeichert, wie in Anhang A dargestellt. Dadurch können sie später von Matlab importiert werden.

Die entsprechenden Bilder die für diese Experimente aufgenommen werden, werden von zwölf verschiedenen Personen vorgeführt. Die Modelle werden mit sechs Gesten von sechs Personen gebaut. Diese Modelle werden mit anderen Bilder verglichen, ungefähr 30 von jedem Gestentyp, von diesen sechs Personen und weiteren sechs von denen keine Vorführung zum Aufbau des Modells benutzt wurde.

### 5.1.1 Evaluation des Modellaufbaus

Wie in Kapitel 4.3.1 erläutert werden die Ergebnisse der Evaluation in Diagrammen dargestellt (vgl. Abb. 5.3 bis Abb. 5.8, oben nach Durchschnittsmethode, unten nach der Methode mit dem Besten Modell). Die Abbildungen zeigen die Euklidische Distanz für beide vorgestellten Möglichkeiten des Modellaufbaus (vgl. Kapitel 4.3.3), damit diese miteinander verglichen werden können. Jede Abbildung besteht aus zwei Diagrammen, das erste gehört zu den Ergebnissen des Vergleichs mit einer Bibliothek auf Basis der Mittelwert Methode und das zweite zu einer auf Basis der Best Modell Methode. Die entsprechende Bibliothek enthält die in Abb. 5.1 gezeigten Gestentypen, die von 1 bis 6 nummeriert sind. In Abb. 5.3 werden die Ergebnisse des Vergleichs von zum Gestentyp 1 angehörigen Handgesten und den Modell-Deskriptoren gezeigt, in Abb. 5.4 die zum Gestentyp 2 angehörige Handgesten usw.

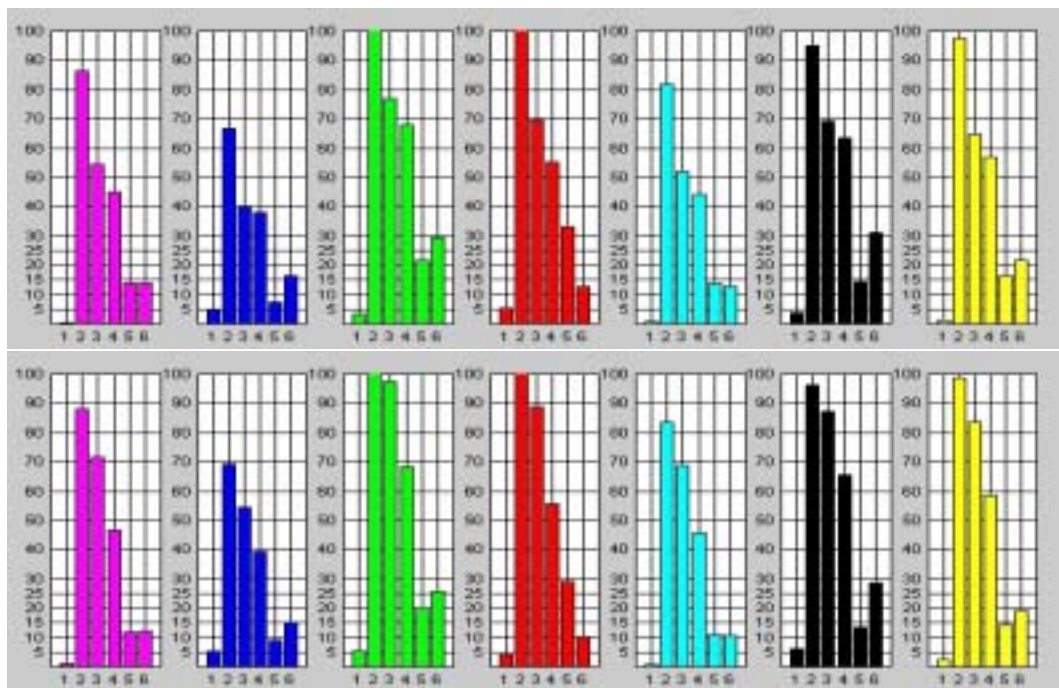


Abbildung 5.3: Euklidische Distanzen mit unterschiedlichem Modellaufbau.

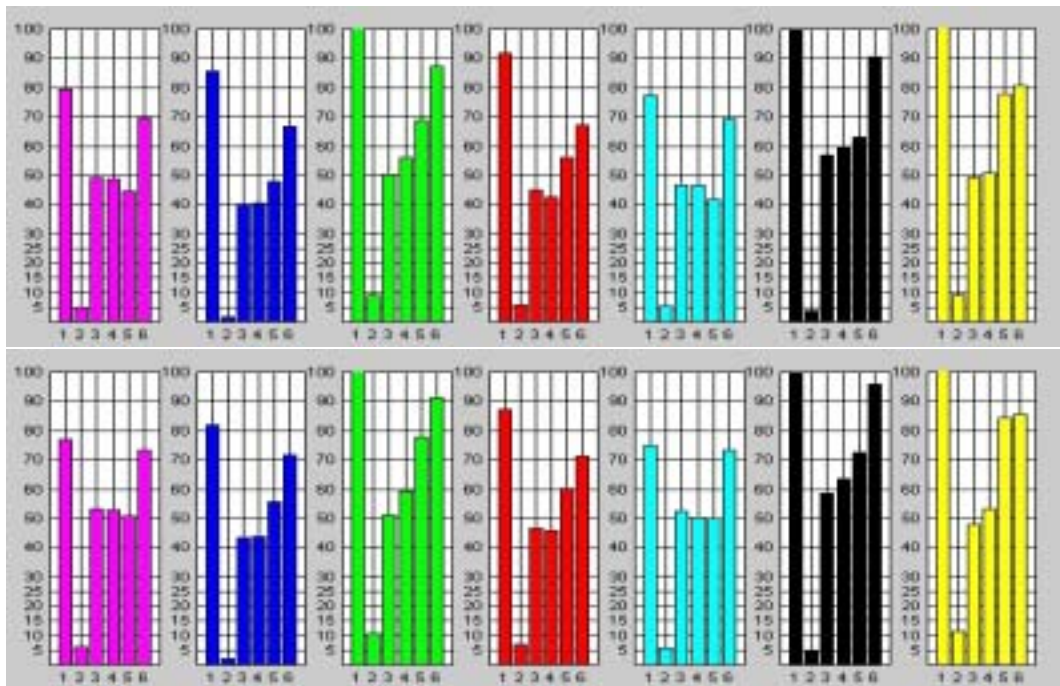


Abbildung 5.4: Euklidische Distanzen mit unterschiedlichem Modellaufbau.

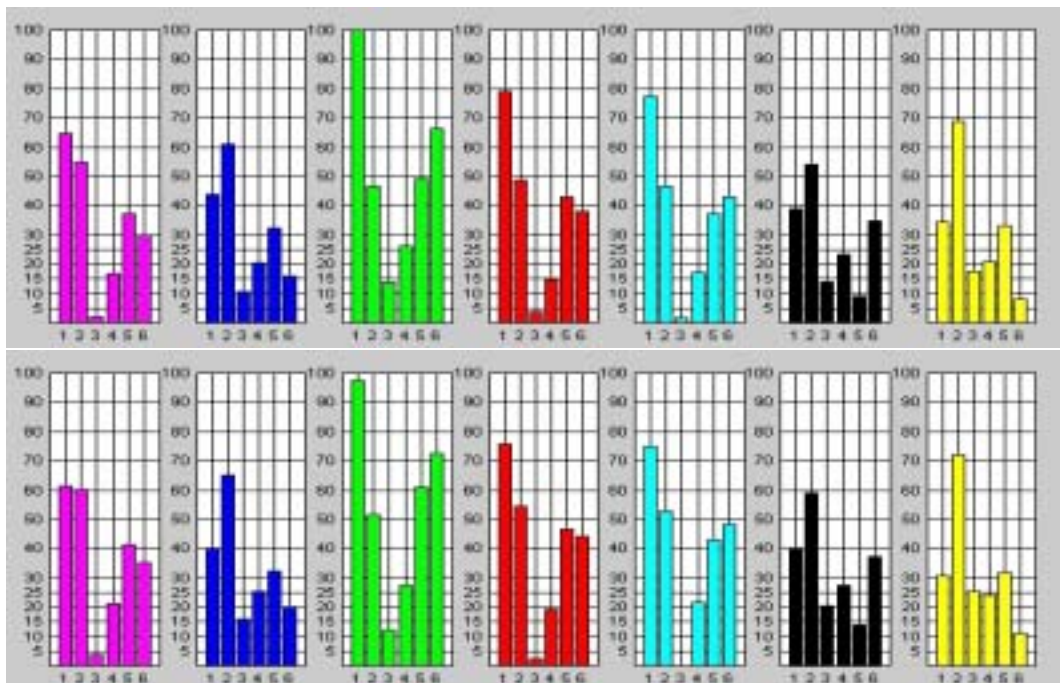


Abbildung 5.5: Euklidische Distanzen mit unterschiedlichem Modellaufbau.

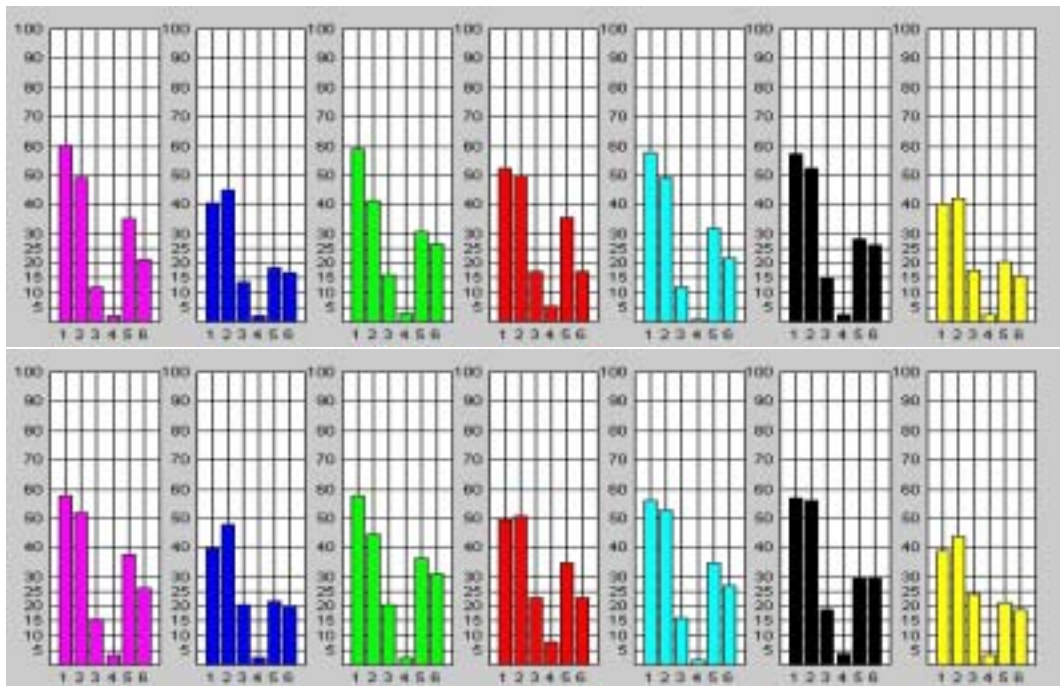


Abbildung 5.6: Euklidische Distanzen mit unterschiedlichem Modellaufbau.

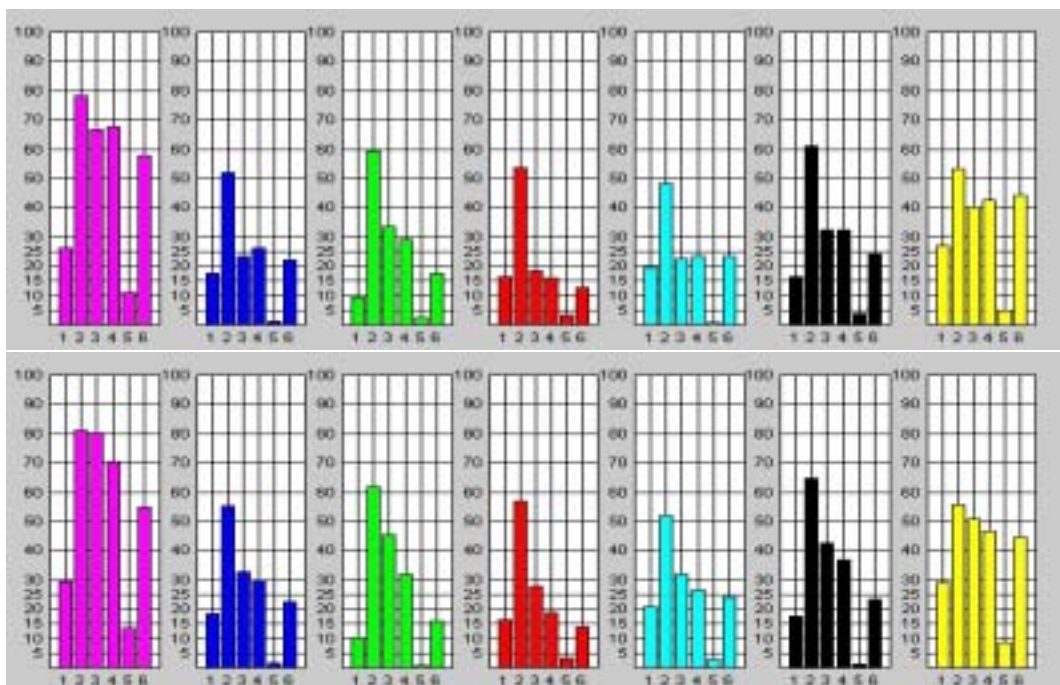


Abbildung 5.7: Euklidische Distanzen mit unterschiedlichem Modellaufbau.

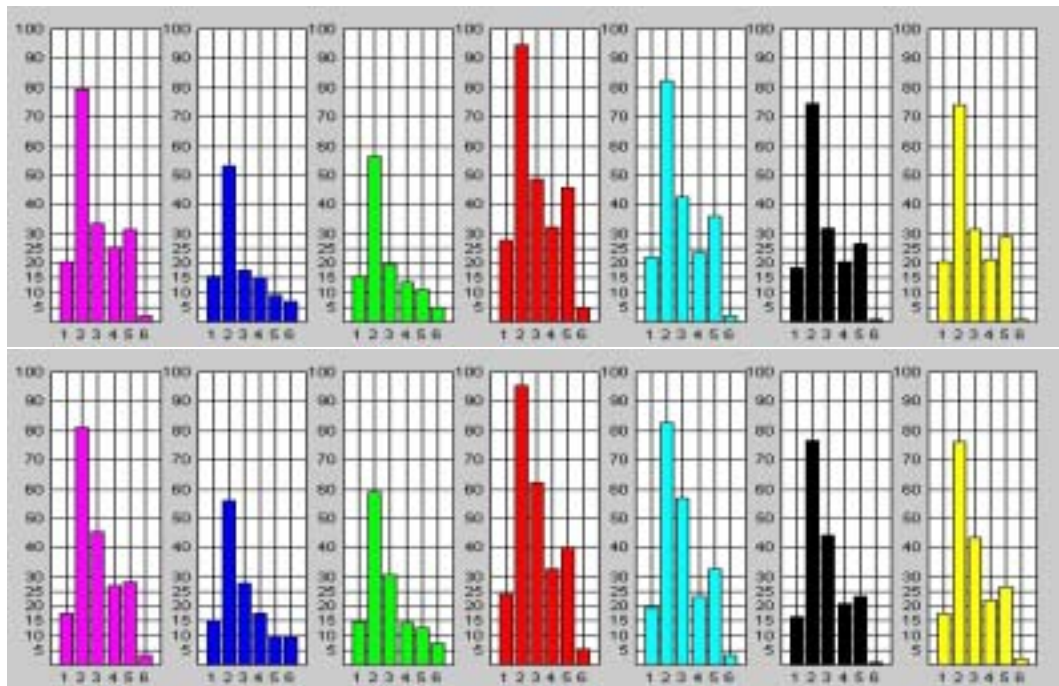


Abbildung 5.8: Euklidische Distanzen mit unterschiedlichem Modellaufbau.

Beide Methoden liefern gute Ergebnisse, obwohl die Ergebnisse bei der durch Mittelung gewonnenen Modellen deutlicher sind als die der anderen. Zwei allgemeine Eigenschaften können beobachtet werden:

- Die Euklidischen Distanzen in den Abbildungen sind beim gleichen Gestentyp kleiner mit Modellen auf Basis des Durchschnitts als mit Modellen auf Basis des Beste-Geste-Algorithmus. Bei unterschiedlichen Gestentypen sind die Distanzen größer. Daher ergeben sich mit dem Durchschnitts-Algorithmus bessere Ergebnisse, wenn die Distanzen verglichen werden.
- In der manchen Fällen liefert die Beste-Modell-Methode die gleichen Unterschiede zwischen den Distanzen wie die Durchschnittsmethode, jedoch bei höheren Werten. Dadurch ist der Unterschied proportional kleiner. Deswegen wird die Beste-Modell-Mehode als schlechter bewertet.

### 5.1.2 Evaluation der Klassifikationsleistung

Die Gesten, die verwendet werden, um den Ablauf des Algorithmus zu testen, wurden von 12 verschiedenen Personen vorgeführt. Die Euklidischen Distanzen werden aus der Bibliothek mit Mittelwert-Deskriptoren berechnet, wie in den oberen Diagrammen in Abb. 5.3 bis Abb. 5.8. Es werden auch Gesten die nicht in der Bibliothek sind (Abb. 5.2) untersucht, um alle Möglichkeiten des Ablaufs des Programmes zu testen. Die Ergebnisse sind in Abb. 5.1 dargestellt. Die Zeilen und Spalten dieser Tabelle haben die folgende Bedeutung:

- In der Bibliothek bestehende Gesten (Stopp, Stopp gespreizt, Griff, Victory, Daumen, Zeigen):

**Positives Ergebnis :**

- Richtig klassifizierte Geste (Reihe 1).

**Negatives Ergebnis :**

- Falsch klassifizierte Geste (Reihe 3).
- Unerkannte Geste (Reihe 2).

- Nicht zur Bibliothek gehörige Gesten (Andere):

**Positives Ergebnis :**

- Unerkannte Geste (Reihe 2).

**Negatives Ergebnis :**

- Falsch klassifizierte Geste (Reihe 3).

	STOP	STOP OPEN	GRASP	VICTORY	THUMB	ZEIGE	OTHER
Richtig klassifiziert	100	96	83,3	96,4	100	100	
Falsch klassifiziert	0	0	8,3	0	0	0	4,3
Unerkannt	0	4	8,3	3,6	0	0	95,7

Tabelle 5.1: Ergebnis der Klassifikation

Die in Tabelle 5.1 ausführlich dargestellten Ergebnisse der Experimente, sind in Tabelle 5.2 kompakter zusammengefaßt. In Tabelle 5.2 läßt sich leicht erkennen, ob das Ergebnis der Klassifikation positiv oder negativ ist.

	STOP	STOP OPEN	GRASP	VICTORY	THUMB	ZEIGE	OTHER
Richtig klassifiziert	100	96	83,3	96,4	100	100	95,7
Unerkannt	0	4	16,3	3,6	0	0	4,3

Tabelle 5.2: Zusammenfassung der Klassifikationsleistung

Bei Betrachtung der Tabellen kann es festgestellt werden, daß der Algorithmus die Klassifikationsfunktion sehr gut erfüllt. Das niedrigste Resultat wird bei der Geste “grasp” erhalten. Der Grund dafür ist jedoch in der Modell-gewinnung zu suchen. Die Geste “grasp” ist sehr unscharf definiert (siehe Abb. 5.9). Es ist eine Geste, die sehr unterschiedliche Abstände zwischen den Fingern haben kann.

Das Klassifikationsergebnis könnte hier wesentlich verbessert werden, wenn man die Geste in mehreren Modellen in die Bibliothek aufnimmt und unterscheiden lässt.

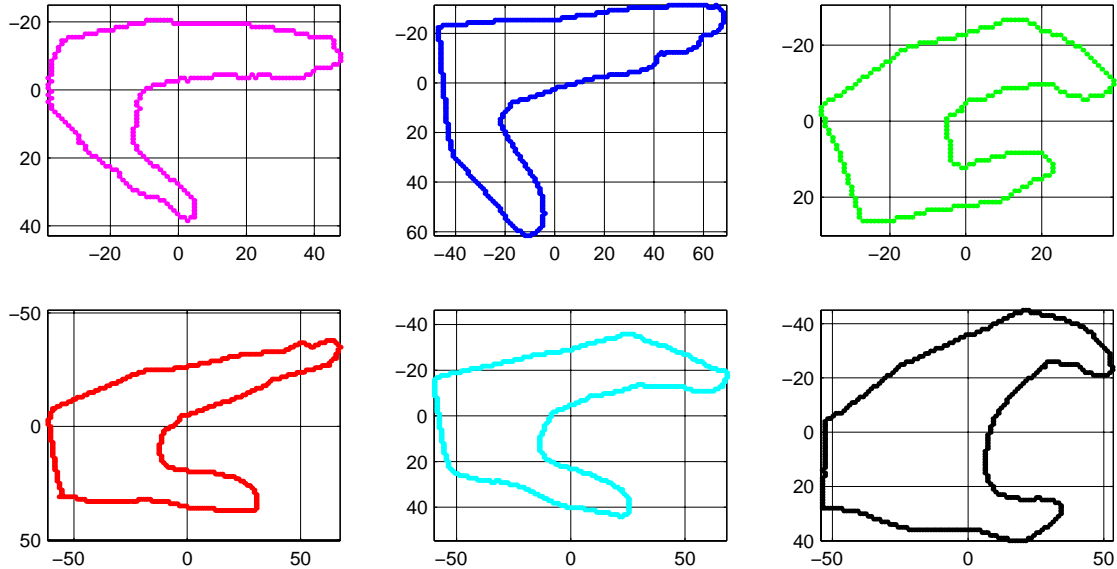


Abbildung 5.9: Benutzte Geste um der Modelldeskriptor “griffe” aufzubauen.

## 5.2 Evaluation der Implementierung

Die Implementierung des Algorithmusses wird überprüft bezüglich dem Ablauf des Programmes selbst, aber auch bezüglich der Beziehung zu anderen Teilen des Projekts GOA.

Der implementierte Algorithmus sollte die in Kapitel 1 dargestellte Voraussetzungen erfüllen:

**Benutzerunabhängigkeit.** Die Benutzerunabhängigkeit wird teilweise erreicht durch den Aufbau der Bibliothek, wie in Kapitel 4.3.3 erläutert. Hierzu werden Gesten verschiedener Personen verwendet. Deshalb ist eine gute Auswahl von Signalen die für die Modelle-Deskriptoren wichtig. Die Benutzerunabhängigkeit wird auch dank der Merkmalsextraktion und der Normierung erreicht.

**Erweiterungsfähigkeit.** Das Programm bietet die Möglichkeit die Bibliotheken an die gegebenen Notwendigkeiten anzupassen. Der geringe Speicherplatz, den die Daten der Bibliotheken brauchen erlaubt die Verwendung unterschiedlicher Bibliotheken und den Wechsel zwischen ihnen.

**Robustheit** gegen:

- Rauschen in der Vorverarbeitung  
Der Einfluß von Rauschen in der Vorverarbeitung wird durch die angewendete Vornormierung minimiert.
- Variation der Gesten beim gleichen Benutzer. Diese Variationen können das Ergebnis verfälschen, haben aber durch die Art des Aufbaus der Bibliotheken nur wenig Bedeutung (vgl. die Diskussion zur Benutzerunabhängigkeit oben). Hier spielt auch die richtige Wahl der Schwellenwerte eine wichtige Rolle (vgl. Kapitel 4.3.2). Eine Variation in der Vorführung der Geste kann eine größere Euklidische Distanz mit sich bringen, selbst wenn die Geste mit dem richtigen Modell-Deskriptor verglichen wird.

**Echtzeitfähigkeit.** Aufgrund der Wahl der Betrachtung von Merkmalen im Frequenzraum und der Existenz schneller Algorithmen zur Abbildung der Ortssignale kann die Klassifikation sehr schnell erfolgen. Die Datenmengen die im Programm verarbeitet werden, sind sehr klein, dadurch erfolgen die Berechnungsprozesse in Echtzeit. Die Laufzeit für die Klassifikation bei einer siebenelementigen Bibliothek beträgt 50 ms und wächst bei mehr Referenzgesten langsam linear.

**Kompaktheit.** Genau wie die Echtzeitfähigkeit wird auch die Kompaktheit dank der Einfachheit der Daten und des implementierten Programmes erreicht.

# Kapitel 6

## Zusammenfassung und Ausblick

In dieser Arbeit wurde ein Algorithmus entwickelt und auf einer Roboter implementiert für die bildbasierte Erkennung statischer Gesten.

Aqui meto las conclusiones del proyecto y como se puede ampliar.

### 6.1 Erweiterung

- ((distincion entre dedo arriba y dedo abajo, asi como la direccion que apunta el indice))
- ((reconocimiento de gestos en 3d))
- ((implementacion con un algorithmo que elimine la parte de brazo))

# Anhang A

## Bibliothek Schnittstelle

Ein wichtiger Punkt dieser Arbeit ist die Flexibilität der Bibliothek. Durch eine Schnittstelle ist es möglich die Elemente in eine Bibliothek zu editieren, und damit die Bibliothek zu der Benutzers Bedurfnisse anpassen. Die Schnittstelle ist mit TCL-TK hergestellt und die Aktionen die durch die Schnittstelle zu führen sind, werden durch C Befehle ausgeführt, mit der Hilfe die MI Library. Die Schnittstelle erlaubt, nicht nur die Bibliotheken zu verarbeiten, sondern auch die Bilder klassifizieren und die Extrahierte Konturpunkte ins ein Datei exportieren.

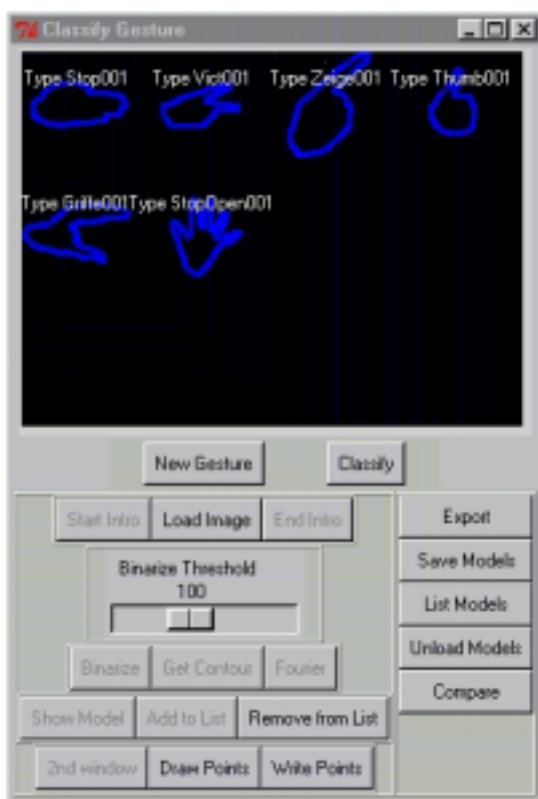


Abbildung A.1: Schnittstelle für die Verarbeitung der Bibliotheken.

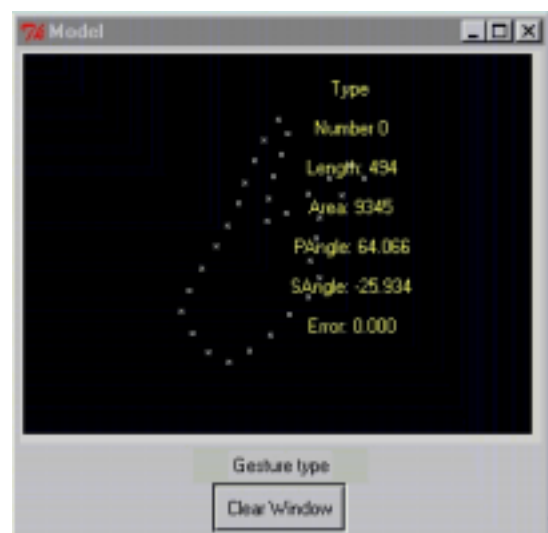


Abbildung A.2: Graphischer Anzeige des Verarbeitetes Gestes.

## A.1 Notwendigkeit eine angemessene Bibliothek

Um die richtige Steuerung des Roboters durch bildbasierte statische Gesten ist es nötig eine gut gebaute Bibliothek zu benutzen, wie in Kapitel 5.1.1 und Kapitel 5.2 erklärt. Deswegen ist es wichtig daß der Benutzer sehen und wählen kann, die abgetastene Gesten, mit den die Modelle aufgebaut werden.

Die Gesten, deren Deskriptoren um den Modell-Deskriptor zu berechnen sind, werden zuerst verarbeitet. Der Benutzer muß entscheiden nach jeder Verarbeitungsschritt ob den Gesten gut ist und weiter verarbeitet sein soll. Das Ergebnis jeder Schritt kann auf die Schnittstellen in Abb. A.1 und A.2 visualisiert werden.

Der Programm funktioniert auch wenn nur eine Geste benutzt wird um das Modell-Deskriptor zu berechnen <sup>1</sup>, aber die Ergebnisse bezüglich Benutzerunabhängigkeit sind besser wenn mehrere vorgeführten Gesten an dieser Stelle benutzt sind. Mit dem gleichen Ziel sollen die Gesten von verschiedene Personen vorgeführt werden. Damit wird der Schwellenwert ((referencialo)) groß genug sein um Hände mit unterschiedlichen Geometrien zu akzeptieren.

## A.2 Ablauf der Schnittstelle

Der Ablauf die Interaktion mit den Bibliotheken wird in der jetzige Sektion beschrieben. Durch die Schnittstelle in Abb. A.1 ist es möglich die bestehende Bibliotheken zu bearbeiten, sowohl als auch neue Bibliothek zu erzeugen. Einzelne Gesten können damit verarbeitet und klassifiziert werden, was ermöglicht die neue Bibliothek auszuprüfen. Die Bilder für den Aufbau der Bibliothek müssen zuerst verarbeitet werden, deswegen wird dieses Punkt zuerst dargestellt, gefolgt von der Bearbeitung der Bibliothek und die Klassifikation eine neue Geste.

### A.2.1 Verarbeitung des Eingabebilds

Die Verarbeitung des Eingabebilds ist auf Diagramm in Abb. A.8 zu sehen. Jeder Schritt entspricht ein Knopf auf der Schnittstelle in Abb. A.1 und ihre Funktionalität wird folgendes dargestellt.

**Load Image** : Ein Dialog Box ((pops up)) und erlaubt der Benutzer ein \*.tif Datei zu wählen und öffnen. Dieser Datei enthält ein Bild von den Roboter aufgenommen. Das Farbbild wird auf der Schnittstelle als eine grauwert image geladet.

---

<sup>1</sup>Das wäre äquivalent zu der Modellaufbau Kriterium “Beste Modell” (in Kapitel 4.3.3 erklärt), wo der Modell-Deskriptor entspricht der Deskriptor einer Geste



Abbildung A.3: Grauwert Bild.

**Set Binarize Threshold** : Hier muß ein Schwellenwert gesetzt werden durch stellen der Bildlaufleiste und der Knopf "Binarize" gedrückt werden. Dieser Wert hängt von das Licht im Originalbild ab. Das Ziel dieser Schritt ist ein Bild mit einem einzigen Blob (die Hand) zu enthalten.

- Wenn ein ungültiger Schwellenwert ausgewählt war, der nächste Schritt kann nicht ausgeführt werden, und wenn der entsprechende Knopf gedrückt wird, ist das Originalbild wieder gezeigt und es wird nach ein neuer Binarisierungs-Schwellenwert gefragt.
- Beim richtigen Schwellenwert wird das binarisiertes Bild gezeigt.



Abbildung A.4: Falsch-binarisiertes Bild.



Abbildung A.5: Richtig-binarisiertes Bild.

**Get Contour** : Dieser Funktion extrahiert die Kontur des in der vorherige Schritte erhaltene Blob.



Abbildung A.6: Kontur des falschbinarisierten Bild.



Abbildung A.7: Kontur des richtigbinarisierten Bild.

**Transformationen** : Die Transformationen, wie in Kapitel ?? dargestellt, werden auf der Kontur durchgeführt beim Drücken des Knopf “Fourier”. Diese Transformationen lauten:

**Vorverarbeitung** : Die Kontur wird normiert.

**Fourier Transformation** : Die Fast-Fourier-Transformation wird auf der vornormierten Konturkette angewendet und dadurch die Fourier-Deskriptoren erhalten.

**Betrag** : Die Betrag der Elemente wird berechnet und der in dieser Arbeit benutzte Deskriptor erhalten.

**Visualisierung** : Die folgende Eigenschaften des Bildes können durch “Write Points” auf der Schnittstelle in Abb. A.2 visualisiert werden:

- der Hauptachse Winkel
- der Querachse Winkel
- die Fläche des Blobs
- die Länge der Originalkontur (ohne Vornormierung)

Der Knopf “Draw Points” zeichnet auf dem gleichen Fenster die Punkte der abgetastete Konturkette.

**Export to File** : Mit dieser Knopf ist es möglich ein Datei zu erzeugen, in dem die Originalpunkte der Konturkette gespeichert werden. Damit ist es nachher möglich diese Daten mit andere Programme zu bearbeiten, wie Matlab in diesem Fall.

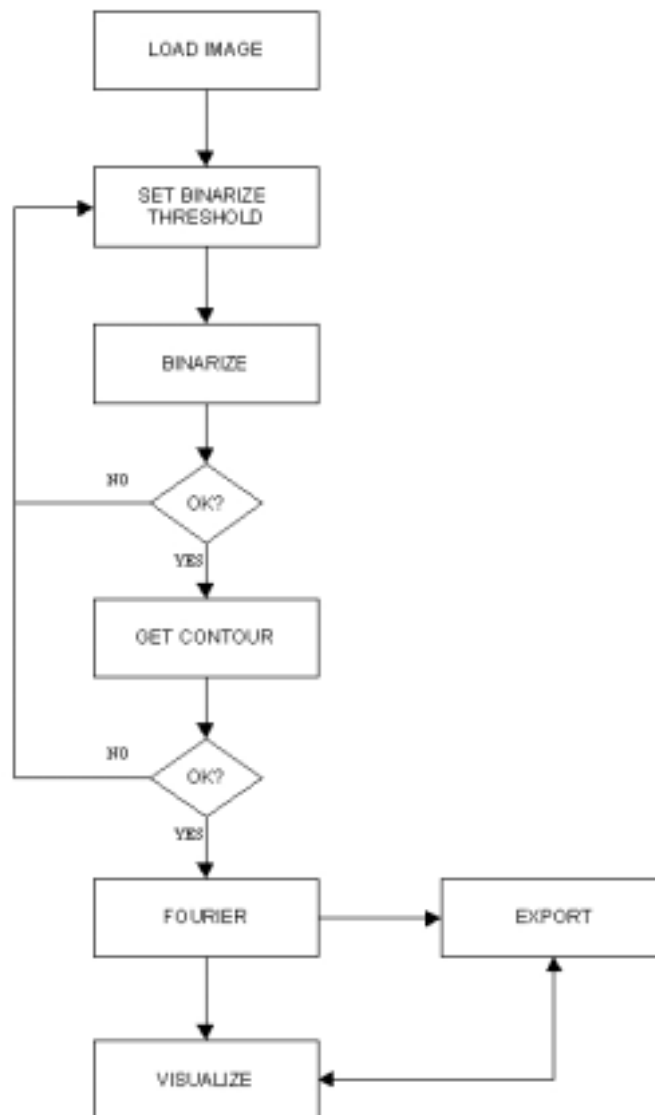


Abbildung A.8: Diagramfluß der Bildverarbeitung.

### A.2.2 Bearbeitung der Bibliothek

Auf Abb. A.9 steht der Ablauf um eine Bibliothek zu bearbeiten. Die Erklärung zu die Einzelschritte wird folgendes dargestellt:

**List Models** Zeigt ein Dialog Box daß erlaubt der Benutzer eine Datei zu wählen, der eine Bibliothek enthält, entweder neue Geste zu testen und klassifizieren oder die Bibliothek zu editieren. Die Geste die in dieser Bibliothek sind werden auf den Bildschirm bezeichnet mit den Namen, die die entsprechende Geste identifiziert.

**Unload Models** Löscht die geladene Bibliothek ohne die Änderungen zu speichern.

Der folgende Ablauf kann auch wenn keine Bibliothek geladet ist ausgeführt werden, daß entspricht eine ganz neue Bibliothek aufzubauen.

**New Gesture** Dieser Knopf stellt, daß die folgende Aktion wird nicht Geste zu klassifizieren sein, sondern eine Bibliothek zu bearbeiten. Ein Fenster “pops up” so der Benutzer ein identifizierende Name für den neuen Gestentyp eingeben kann.

**Start intro** Dieser Knopf wird gedrückt wenn ein neue Gestentyp in der Bibliothek eingefügt werden soll. Danach kann der Benutzer anfangen die Muster Geste, die verarbeitet sein sollen um die Deskriptoren zu erhalten, mit dem die Modelldeskriptoren berechnet werden, zu laden.

**Verarbeitung** Ab dieser Zeitpunkt fängt die Verarbeitung des Bildsignales des zu trainierten Gestentyp an. Der Ablauf ist wie in Kapitel A.2.1 schon dargestellt.

**Load Image**

**Binarize**

**Get Contour**

**Fourier**

**Show Model** Nur wenn genügend von die abgetastete und verarbeitete Bildsignale soll diese Knopf gedrückt werden. Damit wird der Modell-Deskriptor berechnet, und der beste Geste von die abgetastete Hände dieses Gestentyp wird gezeigt .

**Add to List** Um der neue Modell-Deskriptor in der Bibliothek einzufügen.

**Remove from List** Um eine Modell aus der Bibliothek zu entfernen. Der Name der zu entfernen Gestentyp muß eingegeben werden. Diese Knopf kann in jeder Zeit benutzt werden so lange eine Bibliothek geladet ist.

**Save Models** Nach die Bibliothek bearbeitet ist kann es wieder gespeichert werden, entweder im ursprünglichen Datei oder in einem anderen.

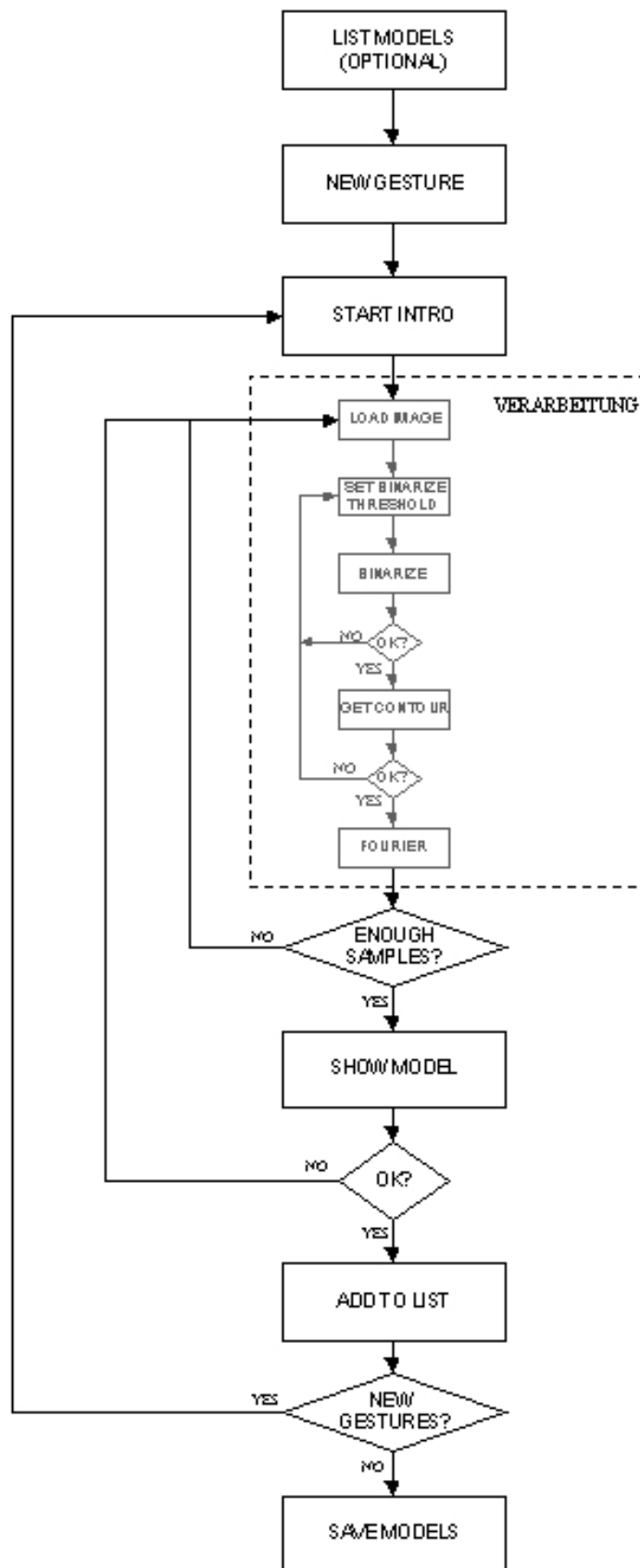


Abbildung A.9: Diagramfluß der Aufbau der Modelle.

### A.2.3 Klassifikation einer Geste

Da das Programm Modell-Deskriptoren um die zu klassifizierende Gesten zu vergleichen braucht, kann dieser Ablauf nur wenn eine Bibliothek geladen ist stattfinden.

**List Models** (Siehe Kapitel A.2.2)

**Classify** Erlaubt die Ablauf der Klassifikation einer Geste.

**Verarbeitung** Die Verarbeitung läuft wie in Kapitel A.2.1 schon beschrieben.

**Load image**

**Binarize**

**Get Contour**

**Fourier**

**Compare** Jetzt wird der berechnete Deskriptor der zu klassifizierende Geste mit der Modell-Deskriptoren verglichen. Das Ergebnis wird an der Schnittstelle gezeigt.

**Visualisierung** Die Visualisierung lautet wie in Kapitel A.2.1 beschrieben, allerdings wird hier zu der Eigenschaften die berechnete Euklidische Distanz zu dem Modell-Deskriptor, zu dem am ähnlichsten ist, geschrieben.

**Export** (siehe A.2.1)

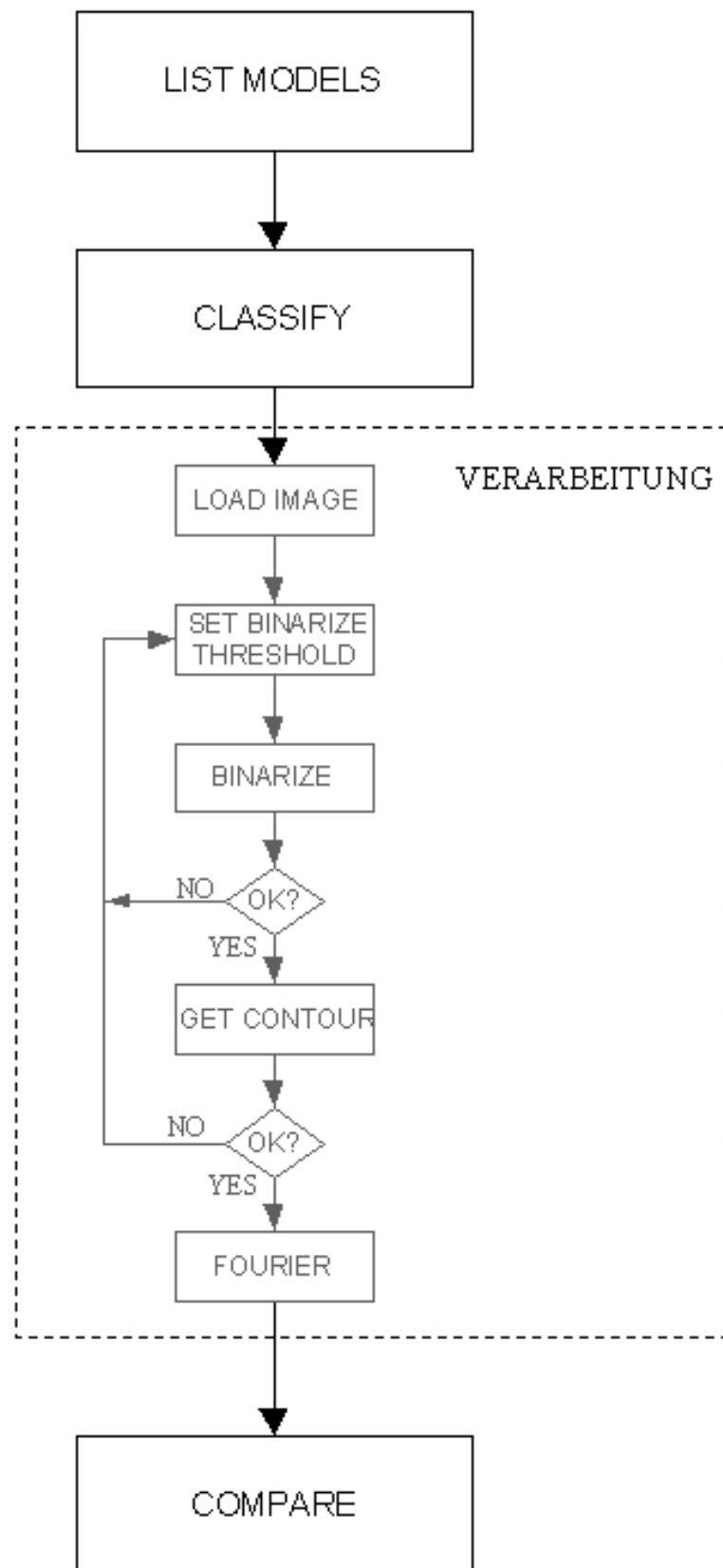


Abbildung A.10: Diagramfluß der Klassifikations-Ablauf.

# Literaturverzeichnis

- [González 93] Rafael C. González, Richard E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1993.
- [Kammeyer 98] K.D. Kammeyer, K. Kroschel. *Digitale Signalverarbeitung*. Teubner Studienbücher, 1998.
- [Kestler 99] Hans A. Kestler, Marcus Borst, Heiko Neumann. Einfache handgesteuerterkennung mit einem zweistufigen nearest-neighbour klassifikator. Technischer Bericht, Universität Ulm, 1999.
- [Kindratenko 96] V. V. Kindratenko, Pierre J. M. Van Espen. Klassifikation von unregelmäßige gestaltet mikro-objekte durch verwenden komplex fourier deskriptoren. Technischer Bericht, University of Antwerp, 1996.
- [Ly 01] Ly. Hautfarbsegmentierung. Diplomarbeit, Universität Karlsruhe, 2001.
- [MIL 99] *Matrox Imaging Library Version 6.1 Command Reference*, 1999.
- [Triesch 98] J. Triesch, C. von der Malsburg. A gesture interface for human-robot-interaction. Tagungsband: *AFGR98*, Seite Gesture Tracking and Recognition and Applications, 1998.