

1. Zusammenfassung (Abstract)

In unserer Gesellschaft spielt die Personenidentifikation eine zunehmend grosse Rolle. Sie wird heute in fast allen Bereichen eingesetzt. Ob beim Kontobezug, beim Einkauf oder beim Grenzübergang. Immer wieder muss man sich ausweisen und seine Identität belegen. Doch Karten und Ausweise können kopiert oder gestohlen, Codes vergessen oder geknackt werden. Es wird darum ständig nach neuen Möglichkeiten der Personenidentifikation gesucht, die genau diese Mängel nicht aufweisen. Hier kommt die Personenidentifikation anhand biometrischer Merkmalen von Händen zum tragen.

In dieser Studienarbeit wurde eine Personenidentifikation über die biometrischen Merkmale der rechten Hand realisiert. Genauer gesagt, findet die Identifikation anhand 16 verschiedener Merkmale statt. Es sind dies Länge, Breite, Umfang und Fläche jedes einzelnen Fingers (ohne Daumen).

Es wurde ein spezieller Tisch für die Erfassung der Hand hergestellt. Mit diesem Tisch können die Lichtverhältnisse und der Abstand zur Kamera kontrolliert werden, um die Qualität des Bildes zu sichern. Im Aufnahmetisch befindet sich eine CCD-Kamera. Diese ist über eine Framegrabber-Karte mit dem PC verbunden.

Für den Computer wurde das Programm „Identifizier“ entwickelt. Mit diesem Programm können zum Einen die Bilder von Händen erfasst und in einem Archiv abgelegt werden. Andererseits kann ein Bild einer Hand ausgemessen und die Messwerte mit den Werten im Archiv verglichen werden. Stimmen die Werte der Hand mit den Werten einer im Archiv abgelegten Hand genügend genau überein, ist die Hand (Person) identifiziert.

Alle Algorithmen zur Berechnung dieser Handdaten wurden selber hergeleitet und implementiert. Es wurde bereits ein Archiv mit über 230 verschiedenen Händen von 80 verschiedenen Personen angelegt die alle mit einer 90% Sicherheit identifiziert werden können.

2. Aufgabenstellung (Original)

**Abt. Elektrotechnik
Bildverarbeitung und Robotertechnik**

SASS2000

**Studienarbeit
für
R. Fuchs und S. Huber**

Personenidentifikation anhand biometrischer Merkmale von Ohren und Händen

1. Einführung

Verschiedene Erfahrungen aus Unglücksfällen oder Kriegen haben gezeigt, dass sich Personen anhand von Körperteilen wie Ohren oder Händen identifizieren lassen. Während Fingerabdrücke oder andere biometrische Merkmale oft nicht rechtzeitig erfasst wurden und Daten damit nicht zur Verfügung stehen, findet man relativ leicht alte Fotos oder Videoaufzeichnungen von Personen, in denen die genannten Merkmale gut sichtbar sind und ausgemessen werden können. Es hat sich gezeigt, dass gerade die Ohren eines Menschen spezifische Merkmale aufweisen, an Hand derer sich eine recht sichere Identifikation vornehmen lässt.

Die meisten Merkmale zur Beschreibung des Erscheinungsbildes des menschlichen Ohres, die heute verwendet werden, gehen zurück auf Arbeiten von Alfred V. Lannarelli [1]. Sein Identifikationssystem beruht auf Merkmalen wie Ohrenform (oval, rund, rechteckig oder dreieckig), Distanzen und Abstandsverhältnissen zwischen charakteristischen Merkmalen, Form der Ohrläppchen usw.

Bei der Charakterisierung der Hände sind Form und Grössenverhältnisse von Hand und Fingern wichtig.

Da diese Merkmale vielfach auch dann noch aus Bildmaterial oder Aufzeichnungen messbar sind, wenn andere biometrische Merkmale nicht bekannt oder nicht mehr zugänglich sind, ist davon auszugehen, dass sie in Zukunft vermehrt Beachtung finden werden.

2. Aufgabenstellung

In dieser Studienarbeit soll eine Personenidentifikation auf Grund von Bildern der Hand oder des Ohres einer Person durchgeführt werden. Dabei kann davon ausgegangen werden, dass die Aufnahmebedingungen wie Kameradistanz und Beleuchtung kontrolliert werden können.

In einer ersten Phase soll eine Archiv mit entsprechenden Bildern von mindestens 12 Personen angelegt werden. Die aufgenommenen Bilddaten sollen entsprechend den gängigen Methoden beschrieben werden.

In einer zweiten Phase sollen Messverfahren erarbeitet und programmiert werden, welche eine Reihe von Merkmalen liefern, die in einem späteren Erkennungsverfahren eingesetzt werden können [2], [3]. Die definierten Merkmale und Masse sollen auch daraufhin untersucht werden, wie gut sich das Messverfahren mit den Mitteln der Bildverarbeitung automatisieren lässt. Die Messwerte sollen in einer Datenbank abgelegt werden und bilden die Basis für eine spätere automatische Erkennung.

In einer dritten Stufe soll ein Klassifikationsverfahren entwickelt werden, welches den von der Kamera unter definierten Aufnahmebedingungen erfassten Körperteilen Ohr bzw. Hand automatisch deren Besitzer zuordnen kann. Das Endziel besteht in der on-line Erkennung von möglichst vielen Personen auf Grund der biometrischen Merkmale, welche ihre Hände oder Ohren liefern.

3. Hilfsmittel

Als Arbeits- und Programmierstation dienen zwei PC (Pentium). Als Programmiersprache wird C++ verwendet mit dem aktuellen Compiler Microsoft Visual C++. Als Betriebssystem dient Windows NT.

Für die Bildverarbeitung steht eine Digitalisierungs- und Bildspeicherkarte Euresys Domino sowie die Module EasyImage und EasyObject der Bildverarbeitungssoftware e-Vision zur Verfügung. Die Domino-Karte ist in den PC zu integrieren.

Für die Bildaufnahme steht eine schwarzweiss CCD Kameras Sony XC 75CE zur Verfügung. Die Beleuchtungseinheiten sind mit vorhandenen Mitteln möglichst einfach aufzubauen.

Literatur, Unterlagen:

[1] Glenn Cheney: The Ear becomes a Quiet Identifier. ID World, Nov./Dec. 1999 Vol.1 No.5.

[2] Zamperoni: Methoden der digitale Bildsignalverarbeitung. Vieweg-Verlag.

[3] Gonzalez, Wintz: Digital Image Processing. Addison Wesley.

4. Bericht

Am Anfang des Berichtes sind aufzuführen:

Aufgabenstellung, Zusammenfassung auf einer A4-Seite, Inhaltsverzeichnis, Einleitung, Zeitplan und Arbeitsteilung.

Der Bericht soll das Lösungskonzept, die eigentliche Lösung und die erreichten Ergebnisse aufzeigen. Die Ergebnisse sind übersichtlich darzustellen und zu interpretieren. Aufgetretene Schwierigkeiten sollen ebenfalls beschrieben werden. Ferner sollen Lösungsvarianten aufgezeigt werden, die bestanden haben, mit deren Vor- und Nachteilen.

Zur Benützung der erstellten Programme soll eine kurze Bedienungsanleitung erstellt werden. Die Programmlistings sind dem Bericht im Anhang beizufügen.

Der Bericht ist in 2 Exemplaren abzugeben. Die Zusammenfassung (A4-Seite gemäss Standardvorlage) ist zusätzlich separat abzugeben. Zusätzlich sollen dem Bericht auf der

hintersten Seite Disketten oder CD-ROM beigelegt werden, die den Text des Berichtes sowie Quellenprogramme enthalten. Die Datenträger sind mit den Namen der Autoren, Titel der Arbeit und Datum zu beschriften.

Aus dem Bericht muss ersichtlich sein, wer für welche Teile der Arbeit und des Berichtes verantwortlich ist.

Der Bericht soll gut verständlich sein. Insbesondere sollen sich künftige Studenten anhand dieses Berichtes in ähnliche Aufgabenstellungen einarbeiten können.

5. Termine

Abgabe der Aufgabenstellung: 10. April 2000.

Abgabe von Bericht und Datenträgern: 11. Juli 2000, 1700.

Präsentation der Arbeit im Hörsaal: 12./13. Juli 2000, gemäss spez. Programm.

6. Organisation

Arbeitsplatz: Labor für Messtechnik.

Betreuung des Labors, Materialbestellungen, Über- und Abgabe von Arbeitsplatz und Laborschlüssel durch Herr Roffler.

Besprechungen gemäss Absprache.

7. Bewertung

Bewertet werden Arbeitsergebnis, Arbeitsweise bezüglich Systematik und Selbständigkeit, Einsatz der Mittel, Qualität der Programme und der Berichterstattung, sowie Teamarbeit.

Nach Abgabe des Berichtes ist die Arbeit zu präsentieren. Diese Präsentation gilt als mündliche Prüfung.

Rapperswil, 10. April 2000

Dr. H. Grossmann

3. Inhaltverzeichnis

1. ZUSAMMENFASSUNG (ABSTRACT)	1
2. AUFGABENSTELLUNG (ORIGINAL)	3
3. INHALTVERZEICHNIS	6
4. EINLEITUNG	10
5. PROJEKTPLANUNG & ARBEITSTEILUNG	11
5.1. Meilensteine	11
5.2. Zeitplan	12
5.2.1. Soll- Zeitplan	12
5.2.2. Ist- Zeitplan	12
5.3. Arbeitsaufteilung	13
6. PHYSIKALISCHE SYSTEMANALYSE	14
6.1. Der Aufnahmetisch	14
6.1.1. Kamera	15
6.1.1.1. Auflösung der Kamera	15
6.1.2. Beleuchtung	16
6.2. Weitere Hilfsmittel	17
6.3. Gesamtsystem	17
6.4. Vorstudie	18
6.4.1. Die Hand als Objekt	18
6.4.2. Ausmessen der Hände	18
6.4.3. Messresultate und Kurven	19
6.4.3.1. Tabelle Fingerlänge	19
6.4.3.2. Tabelle Fingerdicke	19
6.4.3.3. Diagramm Fingerlänge	20
6.4.3.4. Diagramm Fingerdicke	20
6.4.3.5. Hilfe zum Verständnis	21
6.4.3.6. Bemerkungen zu den Messresultaten	21
6.4.4. Fazit	21
7. BILDVERARBEITUNG	22
7.1. Einleitung	22
7.2. Konturcode (Freeman-Code)	23
7.2.1. Beispiel zur Konturcodierung	23
7.3. Nachbarschaft (4-, 8-)	24

7.4.	Schnittpunkt digitaler Kurven.....	24
7.5.	Identifizierer	25
7.5.1.	Benutzte Libraryfunktionen aus eVision.....	25
7.5.1.1.	ImgOper().....	25
7.5.1.2.	ImgContour().....	26
7.5.1.3.	ImgScaleRotate().....	26
7.6.	Bestimmung der Merkmale.....	27
7.6.1.	Bereichseingrenzung der Finger.....	28
7.6.2.	Die Referenzgerade	29
7.6.3.	Die Schnittgerade	29
7.6.4.	Verfahren zur Ermittlung des Umfanges.....	30
7.6.5.	Verfahren zur Ermittlung der Fläche.....	31
7.6.6.	Verfahren zur Ermittlung der Fingerlänge	31
7.6.7.	Verfahren zur Ermittlung der Fingerdicke	32
8.	IDENTIFIKATIONSALGORITHMEN	33
8.1.	Einführung.....	33
8.1.1.	Gauss'sche Glockenkurve	33
8.1.2.	Mittelwert.....	34
8.1.2.1.	Gewichteter Mittelwert.....	34
8.1.2.2.	Mittelwertsbestimmung mit gewichteten Potenzen.....	34
8.1.3.	Auswertung	34
9.	GROBPFLICHTENHEFT	35
9.1.	Einführung.....	35
9.1.1.	Zweck des Dokuments	35
9.2.	Funktionale Anforderungen.....	35
9.2.1.	Das Programm Identifizierer	35
9.2.2.	Archivviewer.....	35
9.2.3.	Datenhaltung	35
9.2.4.	Zusammenhänge (UML- Diagramm).....	36
9.3.	Funktionale Anforderungen im Einzelnen.....	37
9.3.1.	Use Cases	37
9.3.1.1.	Übersicht	37
9.3.1.2.	Use Case Beschreibungen	37
9.4.	Nichtfunktionale Anforderungen.....	39
9.4.1.	Einführung.....	39
9.4.2.	Benutzerschnittstelle	39
9.4.3.	Leistungsanforderungen	39
9.4.4.	Weitere Anforderungen bezüglich Vorgehen und Hilfsmittel.....	39
10.	ANFORDERUNGSSPEZIFIKATION.....	40
10.1.	Klassendiagramm (UML-Notation).....	40
10.2.	Klassenbeschreibung.....	41
10.2.1.	Klasse Hand	41
10.2.2.	Klasse Finger.....	42

10.2.3.	Klasse Identifikation	43
10.2.4.	Klasse Hitliste	43
10.2.5.	Klasse Eintrag	43
10.2.6.	Klasse Handlib1	44
10.2.7.	Klasse Olist	44
10.2.8.	Klasse Pixel	44
10.2.9.	Klasse Line	45
10.2.10.	Klasse konstanten	45
10.2.11.	Struct Elem	45
11.	BEDIENUNGSANLEITUNG	46
11.1.	Installation	46
11.2.	Der Identifier	46
11.2.1.	Die Menüleiste	47
11.2.1.1.	Filemenü	47
11.2.1.2.	Handmenü	48
11.2.1.3.	Settingsmenü	48
11.2.1.4.	Archivemenü	50
11.2.1.5.	Help	50
11.2.2.	Grafikbereich	50
11.2.3.	Ausgabebereich	50
11.2.4.	Identifikationskontrollen	51
11.3.	Der Viewer	52
11.3.1.	Grafikbereich	52
11.3.2.	Anzeige	52
11.3.3.	Kontrollen	52
12.	TESTS UND RESULTATE	54
12.1.	Einführung	54
12.2.	Testpersonen zur Bestimmung der Toleranzbänder	55
12.3.	Messwerte	56
12.3.1.	Meßwerte Fingerdicke	56
12.3.2.	Meßwerte Fingerlänge	57
12.3.3.	Meßwerte Fingerumfang	58
12.3.4.	Meßwerte Fingerfläche	59
12.4.	Diagramme	60
12.4.1.	Diagramm Fingerdicke	60
12.4.2.	Diagramm Fingerlänge	60
12.4.3.	Diagramm Fingerumfang	61
12.4.4.	Diagramm Fingerfläche	61
12.5.	Auswertung	62
12.5.1.	Bestimmung der Toleranzbänder	62
12.5.1.1.	Toleranzbänder	63
12.6.	Systemtest	64
12.6.1.	Einführung	64
12.6.2.	Zutritt erlaubt	65
12.6.2.1.	Erster Durchgang	65

12.6.2.2.	Zweiter Durchgang.....	65
12.6.2.3.	Auswertung	66
12.6.3.	Falscher Zutritt	67
12.6.3.1.	Testdaten	67
12.6.3.2.	Auswertung	67
13.	EPILOG	68
13.1.	Verbesserungsmöglichkeiten und Ideen.....	68
13.2.	Schlusswort	68
14.	LITERATURVERZEICHNIS	69
15.	GLOSSAR	70

4. Einleitung

In unserer Gesellschaft spielt die Personenidentifikation eine zunehmend grosse Rolle. Ob man sich am Bankomat mit Karte und PIN, an der Stempeluhr bei der Arbeit oder an einer Ladenkasse mit einer Cumulus-Karte identifiziert, spielt keine Rolle. Jedes Mal geht es darum erkannt zu werden und eine Verwechslung zu verhindern. Aber schon in dieser beschränkten Auswahl ist zu erkennen, dass die Auswirkungen einer Verwechslung sehr verschiedene Konsequenzen haben können. Karten können kopiert oder gestohlen, Codes vergessen oder geknackt werden, also suchen Konzerne und Regierungen nach Identifikationsmethoden, die genau diese Mängel nicht aufweisen, um z.B. ihre grössten Geheimnisse sicher zu verwahren. Hier kommt die Personenidentifikation anhand biometrischen Merkmalen zum tragen.

In der vorliegenden Studienarbeit werden in einer ersten Phase die Merkmale der Hand erfasst und analysiert. Bereits bekannte Merkmale von Objekten sind z. B. Fläche, Form, Farbe etc. Ob diese Merkmale jedoch für die Erkennung ausreichen werden, oder überhaupt geeignet sind, zeigt erst eine genaue Analyse. Aus dieser Analyse wird dann entschieden welche Merkmale eindeutig und auch messbar sind, um sie für eine erfolgreiche Identifikation einzusetzen. Weiter müssen Algorithmen erarbeiten werden, um diese Merkmale aus den Bildern per Computer auszumessen. Dazu muss die Theorie der Bildverarbeitung aber auch Erkenntnisse aus der Mathematik, Geometrie und vielleicht sogar der Physik zur Hand genommen werden.

Diese Algorithmen werden anschliessend implementiert und in ein benutzerfreundliches Softwareprogramm verpackt. Das Programm soll nicht nur erlauben Hände auszumessen, sondern soll dazu dienen, eine Identifikation durchzuführen, bei der die Hand mit einem erstellten Archiv verglichen werden kann.

Ziel ist es, ein Archiv mit möglichst vielen verschiedenen Händen (Personen) anzulegen, welche dann immer wieder identifiziert werden können. Natürlich wird angestrebt, diese Identifikation On-Line durchzuführen, das heisst, dass die Person welche sich der Kontrolle unterziehen muss, die Hand in ein Aufnahmegerät einlegt und anschliessend sofort die Meldung erhält ob die Erkennung erfolgreich war.

Hinweis: Kursiv geschriebene Wörter (Bsp. *Identifizier*), sind im Glossar aufgeführt. Bei Wörtern, die mehrmals im gleichen Abschnitt auftreten, wird im Allgemeinen nur das erste von diesen kursiv geschrieben.

5. Projektplanung & Arbeitsteilung

5.1. Meilensteine

- Erste Aufnahmen mit der Kamera → Do. 13. April Woche 15
- Fertigstellung des Aufnahmetisches → Mo. 17. April Woche 16
- Erstes Archiv erstellt → Do. 20. April Woche 16
- Merkmale der Hand bestimmt → Do. 27. April Woche 17
- Anforderungsspezifikation Fertig → Mi. 3. Mai Woche 18
- Datenhaltung und GUI funktionsfähig → Mi. 10. Mai Woche 19
- Erkennung der ersten Handkontur → Mo. 15. Mai Woche 20
- Erste Version des Modul "Identifikation" fertig → Sa. 20. Mai Woche 20
- Abtrennung der Finger möglich → Sa. 20. Mai Woche 20
- Erste Messung aller Merkmale der Hand → Fr. 26. Mai Woche 21
- Onlineverfahren funktionsfähig → Di. 30. Mai Woche 22
- Optimierung der Berechnungszeit um Faktor 10 → Mi. 31. Mai Woche 22
- Identifikation von mehr als 20 Testpersonen → Do. 8. Juni Woche 23
- Erste Version des Berichts fertig → Do. 29. Juni Woche 26
- Abgabe → Di. 11. Juli Woche 28

5.2. Zeitplan

5.2.1. Soll- Zeitplan

Arbeiten	Woche	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Vorbereitungsarbeiten															
Software eVision kennenlernen															
Aufbau des Aufnahmetisches															
Erstellen eines Archivs															
Problemanalyse															
Finden von Merkmalen															
Verfahren zur Merkmalsermittlung															
OOA/OOD															
GUI															
Datenhaltung															
Problem Domain															
Implementation															
GUI															
Datenhaltung															
Problem Domain															
Testphase															
Bericht															
Online-Lösung															
OOD/OOA															
Implementation															

5.2.2. Ist- Zeitplan

Arbeiten	Woche	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Vorbereitungsarbeiten															
Software eVision kennenlernen															
Aufbau des Aufnahmetisches															
Erstellen eines Archivs															
Problemanalyse															
Finden von Merkmalen															
Verfahren zur Merkmalsermittlung															
OOA/OOD															
GUI															
Datenhaltung															
Problem Domain															
Implementation															
GUI															
Datenhaltung															
Problem Domain															
Testphase															
Bericht															
Online-Lösung															
OOD/OOA															
Implementation															

5.3. Arbeitsaufteilung

Arbeiten	Woche	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Vorbereitungsarbeiten															
Software eVision kennenlernen		■	■												
Aufbau des Aufnahmetisches		■	■												
Erstellen eines Archivs			■												
Problemanalyse															
Finden von Merkmalen			■	■											
Verfahren zur Merkmalsermittlung			■	■											
OOA/OOD															
GUI				■	■										
Datenhaltung				■	■										
Problem Domain						■									
Verfahren zur Ermittlung einer Ausgleichgerade Untersuchen						■	■								
Verfahren zur Ermittlung der Merkmale Untersuchen						■	■								
Implementation															
GUI					■	■					■	■	■		
Datenhaltung						■	■								
Problem Domain							■								
Berechnungsalgorithmen							■	■	■						
Archivfunktionen							■								
Optimierung der Algorithmen										■					
Allgemeines											■	■			
Testphase															
Testen der Software												■	■	■	■
Testen der Hardware												■	■	■	■
Bericht															
Großpflichtenheft															
Anforderungsspezifikation							■	■							
Sammeln von Testresultaten															
Vorstudie															
Physikalische Systemanalyse															
Bildverarbeitung															
Bedienungsanleitung															
Test/Resultate															
Online-Lösung															
OOD/OOA															
Implementation															

Legende:

■	→	Gesamtlänge Obergebiet
■	→	Gesamtlänge Teilgebiet
■	→	Stephan Huber & Roland Fuchs
■	→	Roland Fuchs
■	→	Stephan Huber

6. Physikalische Systemanalyse

6.1. Der Aufnahmetisch

Damit die Bilder alle eine ausreichende Qualität erreichen, wird ein spezielles Erfassungsgerät benötigt. Deshalb wurde extra ein Aufnahmetisch hergestellt. Damit dieser in kürzester Zeit verfügbar war, wurde auf vorhandene Ressourcen zurückgegriffen. Der eigentliche Tisch entstand aus zwei alten Nachttischen. Im ersten wurde in der Mitte ein Loch für die Kamera gebohrt. Weil die Distanz zwischen Objektiv und Objekt zu klein war, wurde dem zweiten Nachttisch der Boden entfernt und auf den ersten aufgesetzt. Es wurden alle 90° eine Reihe von Löchern gebohrt, um die Aufnahmefläche und die Beleuchtung in Bezug auf das Kameraobjektiv verschieben zu können. Die Aufnahmefläche und die Beleuchtung liegen auf je vier Aluminium Winkeln, die mit dem Tisch verschraubt sind. Als Aufnahmefläche wurde eine Antireflex-Scheibe eingesetzt, um die Reflexionen in Grenzen zu halten. Um den Aufnahmebereich einzuschränken wurde ein Kartonring eingepasst. Zuletzt kam noch ein schwarzer Kartonaufbau dazu, um den Lichteinfall von aussen so stark wie möglich einzuschränken. Der Aufbau wie auch die Kartonabdeckung vorne sind mit Klettband befestigt damit ein schneller Zugriff zur Kamera, der Beleuchtung und der Aufnahmefläche gewährleistet ist. Das Ziel war möglichst konstante Lichtverhältnisse für die Aufnahmen zu schaffen.

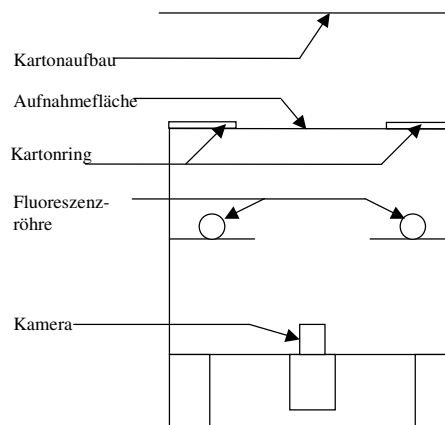
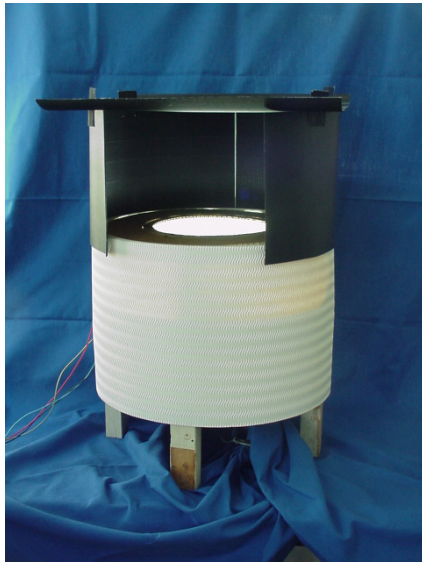


Fig. 6.1 Der Aufnahmetisch

6.1.1. Kamera

Es wird eine CCD monochrom Kamera eingesetzt, die über die Framegrabber-Karte mit dem PC verbunden ist. Die Grösse des Kamerabildes ist 768x576 Bildpunkte. Die Kamera verfügt nicht über ein Zoom, so dass die Grösse der Aufnahme, in unserem Fall einer Hand, nur durch variieren der Distanz zwischen Objektiv und Objekt beeinflusst werden kann. Bei unserem Aufnahmetisch beträgt diese Distanz 24.8 cm. Die Blende ist auf 16 eingestellt und die Softwareeinstellungen bleiben für die Aufnahme auf den Defaultwerten (Gain = 1, Offset = 0).

6.1.1.1. Auflösung der Kamera

Um die Auflösung der Kamera zu messen, wurde eine Schublehre mit der Öffnung von 4mm in den fertig eingerichteten Aufnahmetisch gelegt (siehe Fig. 6.2).

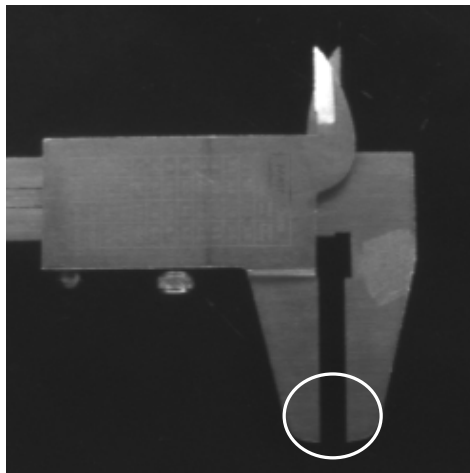


Fig. 6.2 Schublehre im Aufnahmetisch

Der Ausschnitt um die Schnabelspitzen wurde in einem Grafikprogramm vergrössert und die Bildpunkte dazwischen eingefärbt (siehe Fig. 6.3).

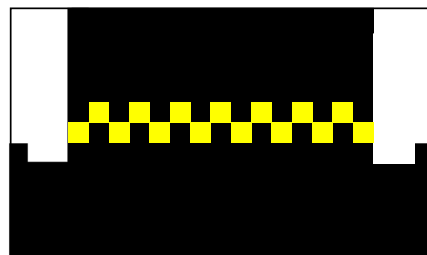


Fig. 6.3 Vergrösserung der Schnabelspitzen

4mm entsprechen also 15 Bildpunkten. Daraus folgt, dass mit dieser Einstellung des Aufnahmetisches, eine Auflösung der Kamera von 0.266mm/Bildpunkt erreicht wird.

6.1.2. Beleuchtung

Der Aufnahmetisch besitzt einen schwarzen Kartonaufbau um den Lichteinfall von aussen einzuschränken. Die Beleuchtung muss für genügend Licht auf der Aufnahme­fläche sorgen und viel stärker als das einfallende Restlicht sein um konstante Lichtverhältnisse herzustellen. Zur Beleuchtung wird eine runde Fluoreszenz­röhre (FL) eingesetzt (siehe Fig. 6.4).



Fig. 6.4 Runde FL im Aufnahmetisch eingebaut

Dadurch wird die Hand auf der Aufnahme­fläche gleichmässig von unten ausgeleuchtet. Die FL liegt auf einem Kartonring. Der Abstand vom Ring zur Aufnahme­fläche wurde auf 12cm eingestellt um zu verhindern, dass auf der Aufnahme eine Reflexion der Lampe zu sehen ist.

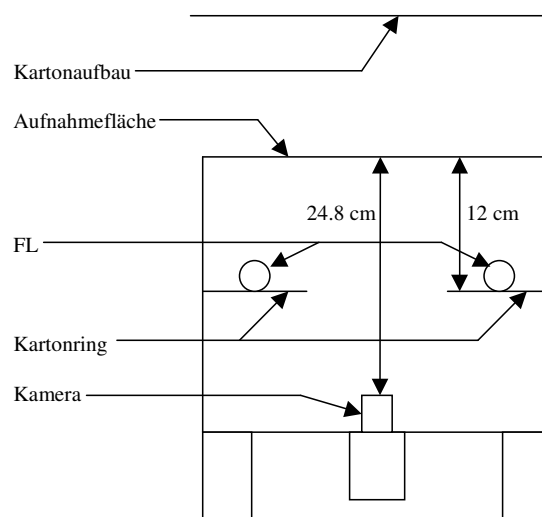


Fig. 6.5 Querschnitt durch den Aufnahmetisch

6.2. Weitere Hilfsmittel

Computer:

- Pentium Prozessor
- Taktfrequenz 90 MHz
- 32MB RAM
- Graphikkarte mit VGA-Modus (640x480 Pixel, 16 Farben)
- Mindestens 1 MB freien Speicherplatz auf der Harddisk
- Peripheriegeräte: Tastatur und Bildschirm

Entwicklungstools:

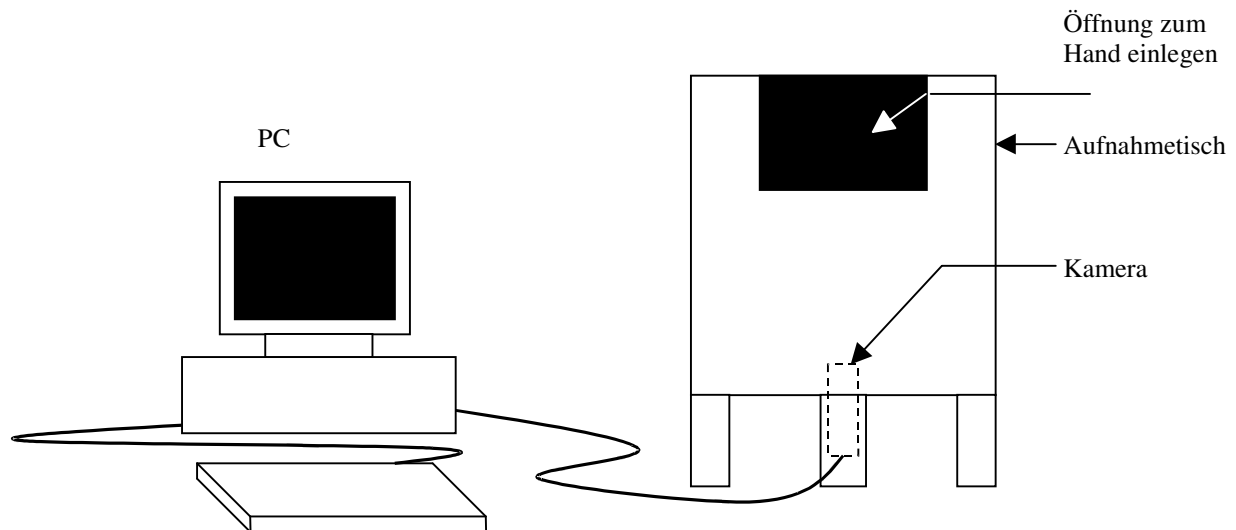
- Microsoft Visual C++ 6.0
- Rational Rose 98 Enterprise Edition
- Easy Access 5.2
- Easy Grab 6.0

Einschubkarten:

- Framegrabber Karte 'EuroCard Domino Mini'

6.3. Gesamtsystem

Über die CCD-Kamera und die Framegrabberkarte empfängt der Computer das Bild der im Aufnahmetisch eingelegten Hand. Im PC kann nun mit der Bildverarbeitung begonnen werden.



6.4. Vorstudie

6.4.1. Die Hand als Objekt

Als Erstes musste die Hand als Objekt studiert werden, um *Merkmale* zu finden, die für eine spätere Identifikation geeignet sind. Schon bald wurde klar, dass man sich nicht auf die ganze Hand konzentrieren kann, sondern nur auf die Finger, da diese bei den Aufnahmen immer gleich bleiben. Der Handballen zum Beispiel ändert seine Form sehr stark bei verschiedenen Aufnahmen (Fig. 6.6).

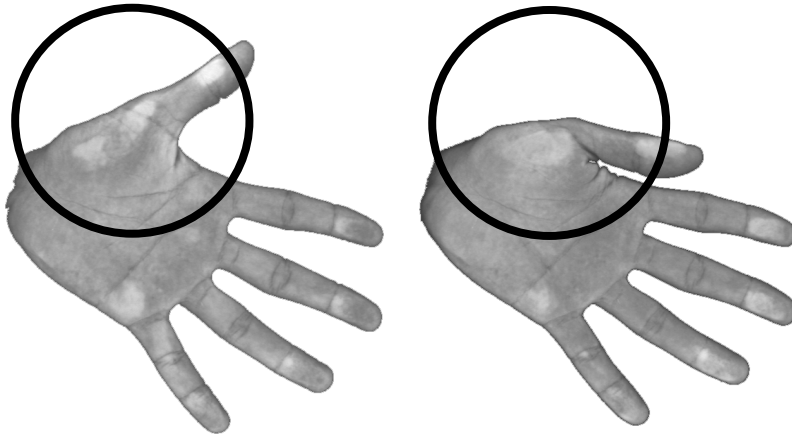


Fig. 6.6

Auch wurde klar, dass die Zeichnungen der Hand, das heisst die Handlinien, und andere optische Merkmale in einer ersten Arbeit nicht in Frage kommen würden, da das Eruiieren dieser sehr schwierig ist. Also wurde entschieden, dass die Merkmale nur von der Form abhängen dürfen. Schnell kamen wir auf die vier Merkmale die uns am eindeutigsten erschienen. Es sind dies der Umfang, die Fläche, die Länge und die breiteste Stelle (Dicke) jedes einzelnen Fingers, ohne den Daumen (Fig. 6.7).

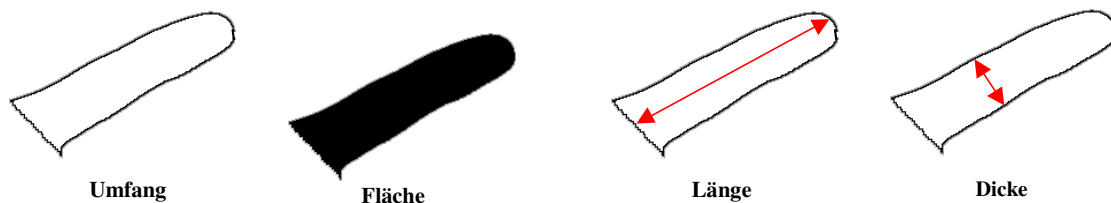


Fig. 6.7

6.4.2. Ausmessen der Hände

Anschließend ging es darum, ein *Archiv* von verschiedenen Händen anzulegen und diese von Hand, das heisst mit Massstab, Zirkel etc. auszumessen. Die Resultate ergaben, ob die gemessenen Merkmale geeignet sind, um später in die *Identifikationsmerkmale* aufgenommen zu werden. Uns war bewusst, dass wir die Fläche und auch den Umfang nur schwer messen können, und haben darum diese beiden Zahlen nicht explizit berechnet, sondern haben aus Überlegungen hergeleitet, dass diese Merkmale auch sehr entscheidend sein müssen. Eine klare Antwort werden wir aber dann erst erfahren, wenn wir die Daten mit dem Computer ausrechnen und vergleichen können.

6.4.3. Messresultate und Kurven

Zu Testzwecken wurden von 10 verschiedenen Personen jeweils 3 Aufnahmen der gleichen (rechten) Hand, aber bei unterschiedlicher Lage, bzw. Spreizung der Finger etc. gemacht. Die so aufgenommenen Hände wurden anschliessend zur Ausmessung ausgedruckt. Alle Messungen wurden auf einem Zeichnungsbrett durchgeführt, wobei mit einem „mm-genauen“ Massstab gemessen wurde. Die Kommastellen resultieren aus nicht eindeutigen Werten, welche auf den nächst folgenden halben mm auf- bzw. abgerundet wurden.

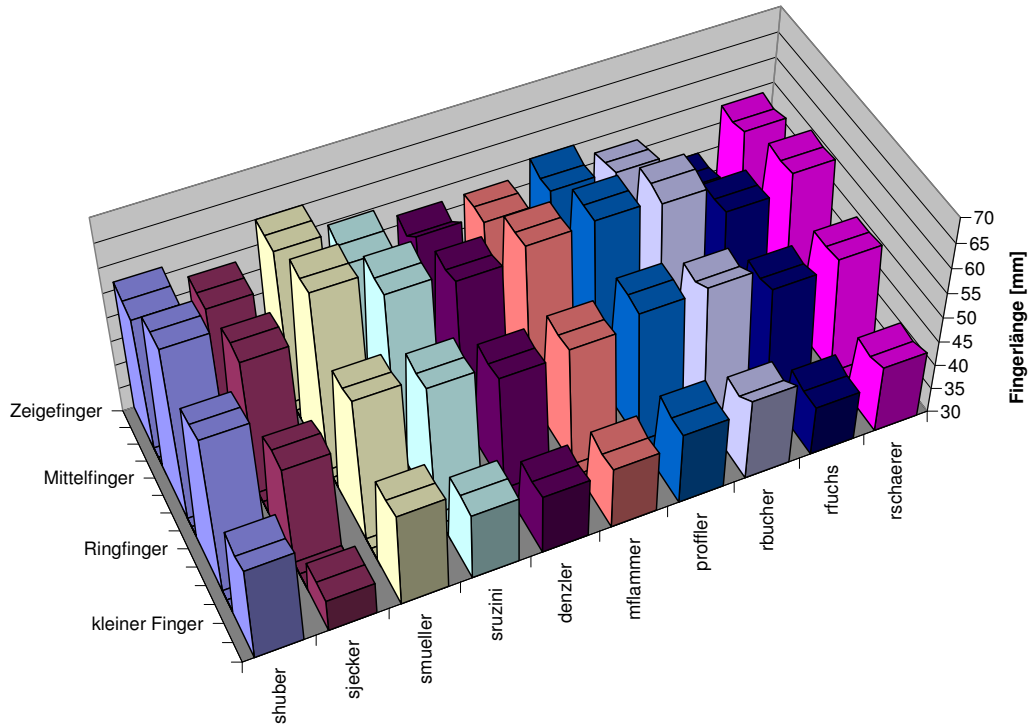
6.4.3.1. Tabelle Fingerlänge

	shuber	sjecker	smueller	sruzzini	denzler	m-flammer	proffler	rbucher	rfuchs	rschaerer
Zeigefinger	56.5	53	60	54.5	52	52	55	52	45	53.5
	56	53	60	53.5	51	52.5	54	53	45	53
	55.5	53	60	53	54	52.5	54.5	54	46	54
Mittelfinger	63	56	65	61	57.5	60	61	61	53.5	57.5
	63.5	56.5	64.5	60	57	60	61	61	53.5	57
	63.5	56	65	60	58	60.5	61	60	54	57.5
Ringfinger	59	48	56.5	55	52	53	56	54.5	50.5	52
	59.5	49	56.5	54.5	51.5	52.5	56	54	50	52
	60	48.5	57.5	55	52	53	55.5	56	51	52.5
Kl. Finger	49	36.5	48.5	44	42	43	45.5	44.5	41	43
	48	36.5	48.5	44	41.5	42	44.5	44	40	42.5
	49	36.5	49	43.5	42	42.5	44.5	46.5	40	43.5

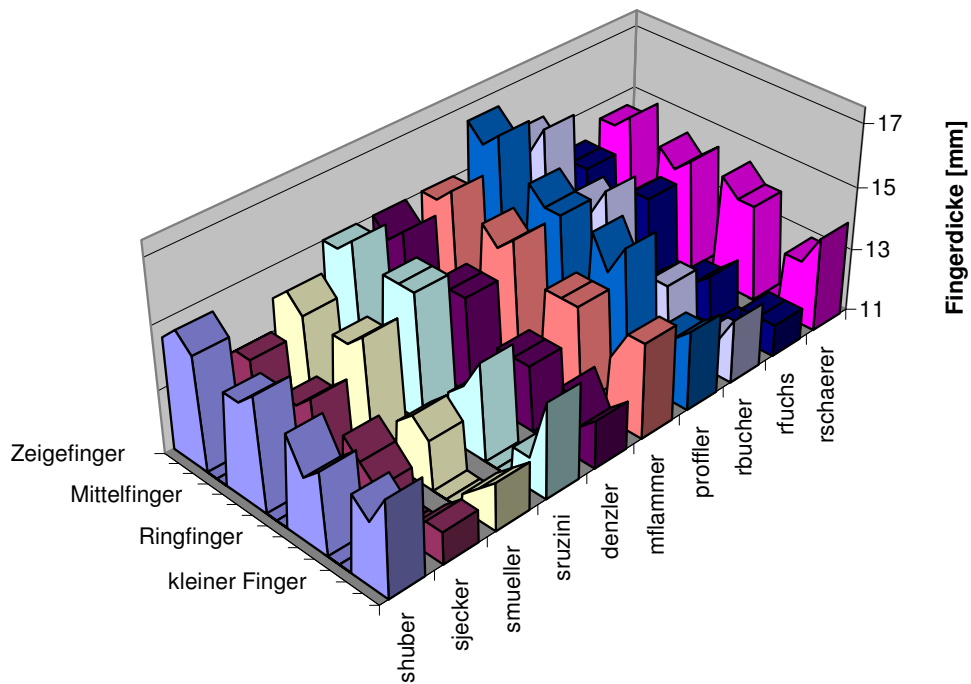
6.4.3.2. Tabelle Fingerdicke

	shuber	sjecker	smueller	sruzzini	denzler	m-flammer	proffler	rbucher	rfuchs	rschaerer
Zeigefinger	14.5	13	14	15	15	15	16.5	15.5	14	14.5
	15	13.5	14.5	15	14.5	15	16	15	14	14.5
	14.5	13.5	14	15.5	15	15.5	16.5	16	14	15
Mittelfinger	14	13	14	15	14	15	15.5	14.5	13.5	14.5
	14	13	14	15	14	14.5	15	14	14	14
	14.5	13.5	14.5	15	14	15	15	15	14	14.5
Ringfinger	14	13	12.5	12.5	13	14	15	12.5	12	14.5
	13	12.5	13	13	13	14	14	13	12	14
	13.5	12	12.5	14	13	14	15	13	12.5	14
Kl. Finger	13.5	11.5	11.5	12	13	13	13	12	12	13
	13	12	12	12	12	14	13	12	12	13
	14	12	12.5	14	12.5	14	13.5	13	12	14

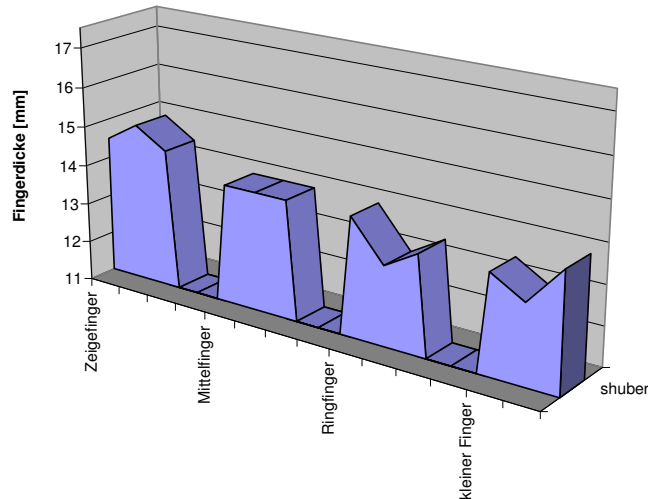
6.4.3.3. Diagramm Fingerlänge



6.4.3.4. Diagramm Fingerdicke



6.4.3.5. Hilfe zum Verständnis



Jede Farbe entspricht einer Person und jeder „Berg“ einem Finger. Bei jedem Finger sind immer drei Werte vorhanden, wie bei den Messdaten (Kap. 6.4.3.x). Im Idealfall besitzen die einzelnen „Berge“ oben eine horizontale Fläche (Mittelfinger). Das bedeutet, dass alle drei Messwerte sehr nahe beieinander liegen. Je grösser die Abweichung desto unebener die „Bergspitze“.

6.4.3.6. Bemerkungen zu den Messresultaten

Wie die Messungen zeigen, variieren die Messresultate von einer Messung zur anderen, das heisst, dass der selbe Finger bei zwei verschiedenen Messungen andere Resultate aufweisen kann. Dies entsteht, weil einerseits die Messgenauigkeit beschränkt ist. Andererseits werden die Messungen immer auf die gleiche Art vorgenommen, jedoch werden die Hände bei unterschiedlicher Position aufgenommen. Dieses Phänomen taucht nicht nur auf, wenn von Hand ausgemessen wird. Auch beim Computer stellen sich die selben Probleme. Darum wird es wahrscheinlich sein, dass bei den endgültigen Resultaten auch gewisse Abweichungen auftauchen. Je kleiner diese Abweichungen sind, um so besser bzw. genauer, wird die Identifikation im Endeffekt erfolgen.

6.4.4. Fazit

Wie aus den Diagrammen zu sehen ist (Kap. 6.4.3.x), können Finger verschiedener Hände gleich lang sein (z.B. Zeigefinger von sjecker und rschearer), jedoch sind die Fingerdicken verschieden. Es treten ab und zu Gemeinsamkeiten von einzelnen Fingern auf, die sich aber durch andere *Merkmale* wieder unterscheiden lassen. Dies ist auch das Prinzip mit welchem unsere Identifikation funktioniert. Wir nehmen an, dass alle Personen, die an unserer Identifikation beteiligt sind, unterschiedliche *Fingermerkmale* aufweisen. Dies heisst natürlich, dass das System auf eine gewisse Anzahl von Personen beschränkt sein wird, die voneinander unterschieden werden können. Vielleicht gibt es einige, die die selbe Fingerdicken haben oder sogar gleich lange Finger, aber dann ist vielleicht die Fläche anders, weil einer knochige Finger und der Andere „fette“ Finger besitzt.

Die Messresultate zeigen, dass diese beiden Merkmale, Dicke und Länge, einfach messbare Grössen sind und sich auch gut eignen, um als Identifikationsmerkmale eingesetzt zu werden. Da die Fläche und der Umfang auch bedingt mit der Dicke bzw. Länge zusammenhängen, sind wir davon überzeugt, dass diese zwei Merkmale sehr aussagekräftig sind und die Resultate nicht stark streuen werden. Wir haben uns deshalb entschlossen, die ganze Identifikation auf diese 4 Merkmale pro Finger (ohne Daumen) zu stützen. Daraus resultiert, dass für jede Person 16 mehr oder weniger unabhängige Attribute existieren werden, die dann ausgewertet werden müssen.

7. Bildverarbeitung

7.1. Einleitung

In der digitalen Bildverarbeitung werden die Originalbilddaten in rechnerkonforme Datenformate transformiert. Sie stehen somit als zwei- oder mehrdimensionale, diskrete Funktionen für die Bearbeitung zur Verfügung. Die Verfahren, die auf die digitalisierten Bilddaten angewendet werden, haben letztlich alle die Zielsetzung, den Bildinhalt für den jeweiligen Gebrauch darzustellen. Das Ergebnis kann, muss aber nicht in bildlicher oder graphischer Form vorliegen. Der Begriff „Bildinhalt“ ist dabei rein subjektiv; dasselbe Bild kann für zwei Anwendungen, mit unterschiedlicher Interessenlage, grundsätzlich verschiedene Informationen mitteilen. Dementsprechend werden auch die Transformationen verschieden sein.

Die Bildverarbeitung wird in vielen Bereichen eingesetzt. Hier einige Beispiele:

- Qualitätskontrolle (Werkstoffprüfung)
- Robotik (industrielle Handhabungsmaschinen)
- Medizin (Tomographie, Radiologie)
- Astronomie (Auswertung von optischen Bilddaten unseres Universums)
- Archäologie (Luftbildauswertungen zur Entdeckung von historischen Baulichkeiten)
- Kriminologie (Fingerabdrücke, Schriftenerkennung, Porträtauswertung)
- Produktion (Bestückung, Sortierung, Fehlersuche)

Wie hier zu sehen, ist die Bildverarbeitung ein Mittel, auf welches wir heute nur schwer verzichten können, womit das folgende chinesische Sprichwort bekräftigt wird:

Ein Bild sagt mehr als tausend Worte

7.2. Konturcode (Freeman-Code)

Die Konturcodierung ist ein Verfahren zur fehlerfreien Darstellung beliebiger Binärobjekte auf einem diskreten Raster, das dank seiner praktischen Vorteile und grosser Ausbaufähigkeit bei den verschiedenen Anwendungen weit verbreitet ist. Der *Konturcode* (Englisch auch Freeman code) ist eine Folge von Ziffern (0..7), die die Form eines Binärobjektes genau darstellt. Diese Ziffern beschreiben den Zusammenhang zweier benachbarter *Bildpunkte*. Das heisst, sie geben die Richtung (siehe Fig. 7.1) des nächstliegenden Bildpunkts an. Es muss auch immer definiert werden, ob der Freeman-Code im Uhrzeiger bzw. Gegenuhrzeigersinn angegeben ist, damit man die Kontur rekonstruieren kann.

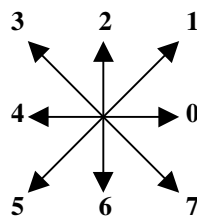


Fig. 7.1 Konturcode

7.2.1. Beispiel zur Konturcodierung

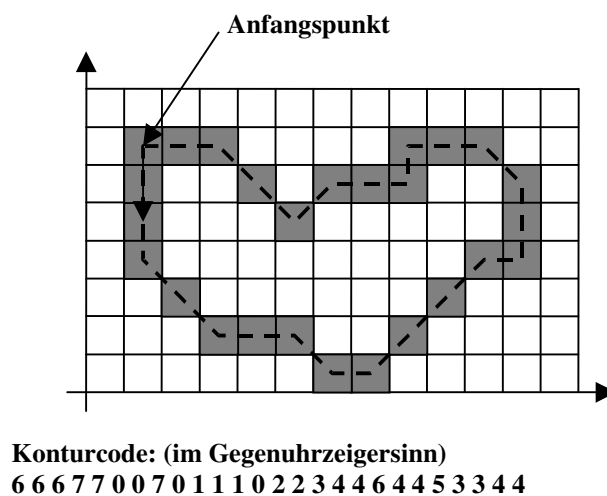


Fig. 7.2 Konturcode eines Objektes

7.3. Nachbarschaft (4-, 8-)

Jeder Bildpunkt besitzt zwei Arten von Nachbarn:

- 4 Nachbarn:** Nachbarn horizontal und vertikal, also links, rechts, oben und unten im Abstand a einer Rastereinheit. Diese Nachbarn werden in der Literatur auch „strenge Nachbarn“ genannt.
- 8 Nachbarn:** Neben den horizontalen bzw. vertikalen Nachbarn, kommen auch noch Nachbarn in der Diagonalen vor. Diese Nachbarpunkte haben aber einen Abstand von $a\sqrt{2}$. Man bezeichnet diese Nachbarn in der Literatur auch als „Nachbarn im weiteren Sinn“. Da in der 8er Nachbarschaft Punkte mit unterschiedlichem Abstand auftauchen, kann die Entwicklung von Algorithmen aufwendiger werden.

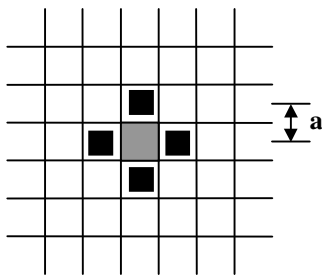


Fig. 7.3 4er Nachbarschaft

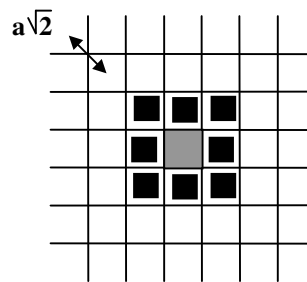


Fig. 7.3 8er Nachbarschaft

7.4. Schnittpunkt digitaler Kurven

Es gibt nichts leichteres als zwei Kurven auf einem Blatt Papier einzuzichnen und den Schnittpunkt mit bloßem Auge zu ermitteln. Doch leider ist dies in der Bildverarbeitung nicht ganz so einfach, da die einzelnen Daten der Bildpunkte auf der Kurve als diskrete Werte vorhanden sind. Auch hier gibt es spezielle Verfahren um Schnittpunkte zu finden. Eine Möglichkeit besteht darin, dass die Koordinaten der einzelnen Bildpunkte der einen Kurve mit den Bildpunkten der anderen verglichen werden. Falls gleiche Werte auftauchen, ist ein Schnittpunkt vorhanden.

Aber Achtung, falls beide Kurven in der 8er Nachbarschaft vorhanden sind, kann es vorkommen, dass der „eigentliche“ Schnittpunkt nicht gefunden wird, obwohl dieser eindeutig vorhanden ist. Dies geschieht dann, wenn der effektive Schnittpunkt nicht auf einem Bildpunkt liegt, sondern in der Mitte von 4 Bildpunkten (Fig. 7.5). Dies kann nicht vorkommen, falls mindestens eine Kurve in der 4er Nachbarschaft vorliegt. Falls dies jedoch aus irgendwelchen Gründen nicht möglich ist, muss mit einem anderen Suchalgorithmus gearbeitet werden. Eine Alternative wäre z. B. das zusätzliche Untersuchen der Nachbarpunkte.

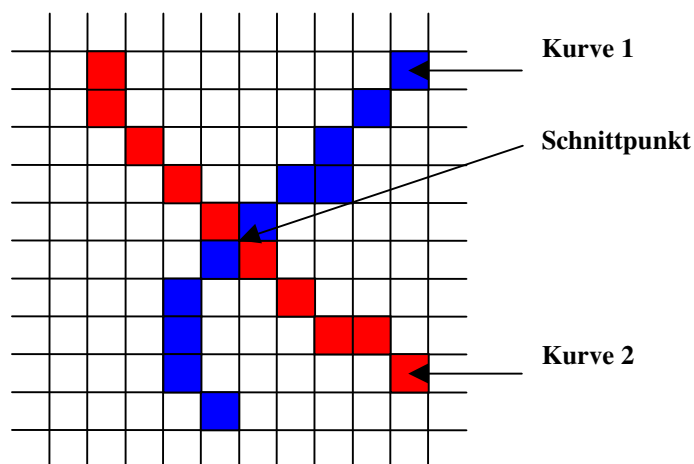


Fig. 7.5 Kein „effektiver“ Schnittpunkt

7.5. *Identifizier*

7.5.1. Benutzte Libraryfunktionen aus eVision

Aus der Bildverarbeitungslibrary EasyLib wurden die drei folgenden Funktionen verwendet:

- ◆ `ImgOper()`
- ◆ `ImgContour()`
- ◆ `ImgScaleRotate()`

Alle anderen Algorithmen der Studienarbeit wurden ohne Hilfe von EasyLib Libraryfunktionen realisiert.

7.5.1.1. *ImgOper()*

Die Funktion `ImgOper()` ermöglicht es Operationen an Bildern vorzunehmen. Z.B. Subtraktion oder Addition von zwei Bildern, das Kopieren von Bildern, das Invertieren usw. Der *Identifizier* benutzt diese Funktion um zwei Bilder voneinander zu subtrahieren.

Funktionsdeklaration:

```
void ImgOper ( enum ARITH_LOGIC_OPERATIONS IMG_SUBTRACT,  
               EROIBW8* &m_EIHandImage, EROIBW8* &m_EITempImage,  
               EROIBW8* &m_EIHandImage);
```

Parameter:

<code>IMG_SUBTRACT</code>	Konstante als Angabe zur Subtraktion
<code>&m_EIHandImage</code>	Zeiger auf das Quellenbild
<code>&m_EITempImage</code>	Zeiger auf das Zielbild

7.5.1.2.ImgContour()

Mit Hilfe der Funktion `ImgContour()`, kann die Kontur eines beliebigen Teiles identifiziert werden. Die Libraryfunktion erzeugt die Kontur der Hand. Es muss ein beliebiger Randpunkt der Hand bekannt sein. Dieser Punkt wird der Funktion als Startwert übergeben.

Weiter benötigt die Funktion eine Grauwertschwelle, sowie die Angabe, ob sich der *Grauwert* der zu findenden *Konturbildpunkt* über oder unter dem eingestellten Schwellwert befindet. Die Funktion folgt nun der Kontur der Hand im Uhr- oder Gegenuhrzeigersinn. Dabei werden nur *Bildpunkt* angefahren, die über resp. unter der eingestellten Grauwertschwelle liegen. Beim Abfahren der Kontur kann zusätzlich zwischen den beiden Verfahren der 8er oder der 4er Nachbarschaft (Kap. 7.3) gewählt werden.

Funktionsdeklaration:

```
void ImgContour (EROIBW8* pSrcImage, enum CONTOUR_MODE n32Mode,  
INT32 n32StartX, INT32 n32StartY, enum CONTOUR_THRESHOLD  
n32ThrMode, UINT32 un32Thr, enum CONNEXITY n32Connexity,  
EVector* pDestVect, BOOL bFreeman = FALSE)
```

Parameter:

<code>pSrcImage</code>	Zeiger auf die Bilddatei
<code>n32Mode</code>	Uhr- oder Gegenuhrzeigersinn
<code>n32StartX</code>	X-Koordinate des Startpunktes
<code>n32StartY</code>	Y-Koordinate des Startpunktes
<code>n32ThrMode</code>	Pixel anfahren mit Grauwerten über oder unter der Grauwertschwelle
<code>un32thr</code>	Grauwertschwelle
<code>n32Connexity</code>	4er oder 8er Nachbarschaft
<code>pDstVector</code>	Resultatvektor der Kontur
<code>bFreeman</code>	Legt entweder den Grauwert des jeweiligen Bildpunktes ab(FALSE) oder den Konturcode (Freeman code) (TRUE).

Resultatvektor:

Der Resultatvektor der Funktion enthält neben den X- und Y- Koordinaten der Konturpixel auch noch einen dritten Wert. Entweder den jeweiligen Grauwert des Bildpunktes oder aber den Konturcode (Kap. 7.2). Dies kann beim Funktionsaufruf gewählt werden (siehe Parameter). Beim Identifier wird immer der Konturcode als dritter Wert gewählt. Dieser kann bei allfälligen Erweiterungen von großer Nützlichkeit sein.

7.5.1.3.ImgScaleRotate()

Rotiert ein Bild um einen bestimmten Winkel.

Funktionsdeklaration:

```
void ImgContour (EROIBW8* tempImage, FLOAT32 x, FLOAT32 y, FLOAT32 x,  
FLOAT32 y, FLOAT32 scale, FLOAT32 scale, FLOAT32 winkel,  
EROIBW8* newImage)
```

Parameter:

<code>tempImage</code>	Quellenbild
<code>x, y</code>	Drehpunkt
<code>scale</code>	Streckfaktor
<code>winkel</code>	Drehwinkel (Linksdrehung)
<code>newImage</code>	Zielbild

7.6. Bestimmung der Merkmale

Es gibt verschiedene Methoden um aus einem Bild Daten heraus zu messen. Wir haben uns auf eine Lösung beschränkt die es erlaubt, alle benötigten *Merkmale* nur aus den Daten der *Handkontur* zu bestimmen. Das heißt soviel, dass für die Messungen der Merkmale nur eine Kontur vorhanden ist und das eigentliche Bild der Hand nur als Vorlage für die Konturbildung und zur Visualisierung aller Messwerte dient.

In einem ersten Schritt ist es nötig, die Finger von der Hand zu trennen (Fig. 7.6) und sogenannte *Fingerkonturen* zu erstellen, da sich alle Merkmale allein auf die Finger beziehen. Diese Abtrennung wird mit Hilfe einer *Referenzgeraden* (Kap. 7.6.2) und einer *Schnittgeraden* (Kap. 7.6.3) realisiert.

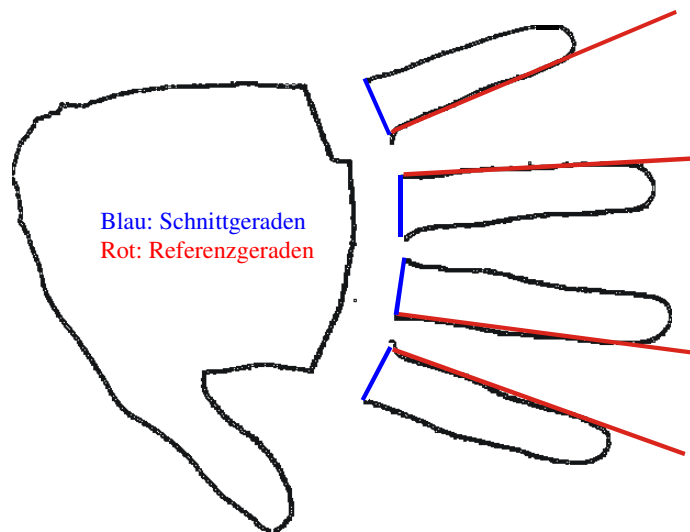


Fig. 7.6 Abgeschnittene Finger

7.6.1. Bereichseingrenzung der Finger

Als erstes müssen die Bereiche eingegrenzt werden, in denen sich die jeweiligen Finger befinden. Dies wurde durch ein sehr einfaches Prinzip realisiert. Es werden alle lokalen *Extremas* (E) der *Handkontur* in X- Richtung ermittelt (Fig. 7.7). Da die Hand immer von der gleichen Seite eingelegt wird, und die Suche nach den Extremas (E) immer am gleichen Ort beginnt, kann nun sehr einfach bestimmt werden, wo sich die einzelnen Finger befinden. Die Extremas E0, E2, E4 und E6 sind nicht *Fingerspitzen* (Kap. 7.6.6).

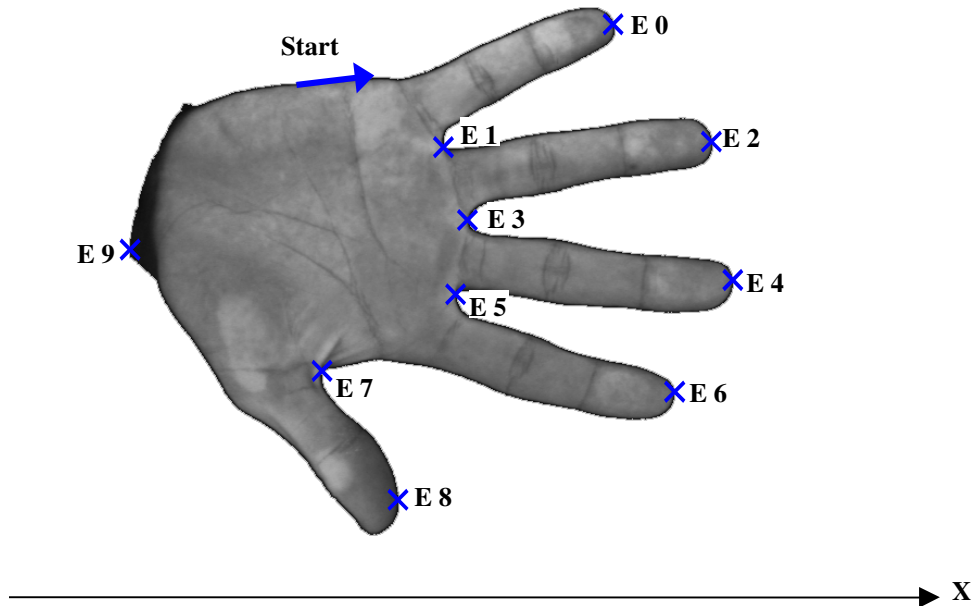


Fig. 7.7

- Im Bereich zwischen E 0 und E 1 liegt die *Referenzgerade* des kleinen Fingers
- Im Bereich zwischen E 1 und E 2 liegt die *Referenzgerade* des Ringfingers
- Im Bereich zwischen E 4 und E 5 liegt die *Referenzgerade* des Mittelfingers
- Im Bereich Zwischen E 5 und E 6 liegt die *Referenzgerade* des Zeigefinger
- E 7, E 8 und E 9 sind Extremas welche nicht von Nutzen sind.

7.6.2. Die Referenzgerade

Zur Messung der *Attribute* wird eine *Referenzgerade* an den Finger gelegt. Eine sogenannte Ausgleichs- oder Regressionsgerade. Von dieser Geraden gehen alle Messungen aus. Es gibt verschiedene Möglichkeiten solch eine Gerade zu ermitteln [Pap. 98]. Da aber die Messresultate nur unmerklich abweichen, wenn die Gerade nicht durch einen komplizierten Algorithmus eingelegt wird, haben wir uns auf ein sehr primitives Verfahren für die Ermittlung der Referenzgeraden geeinigt. Es werden zwei Punkte bestimmt. Der eine Punkt (P1) liegt in einem definierten Abstand zum Fingerspitz auf der Fingerkontur, und der andere (P2) im selben Abstand aber zum Fingertal. Durch diese zwei Punkte ist die Referenzgerade definiert (Fig. 7.8). Ob die Referenzgerade links oder rechts vom jeweiligen Finger zu liegen kommen wurde frei gewählt und hat keinen Einfluss auf die Messung.

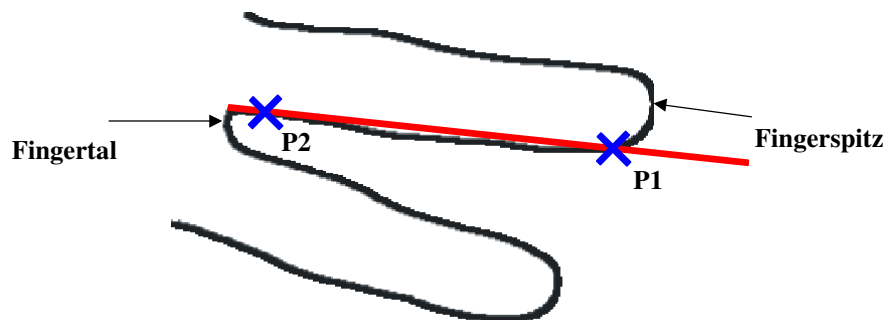


Fig. 7.8 Einlegen der Referenzgeraden

7.6.3. Die Schnittgerade

Um zu definieren wo der Finger beginnt, wird eine *Schnittgerade* eingelegt, die den Finger vollständig von der Hand abtrennt. Diese Abtrennung ist nötig, damit die Messung der Fläche, Länge und des Umfangs immer am gleichen Ort vorgenommen wird. Diese Schnittgerade muss deshalb immer den selben Winkel zur *Referenzgeraden* aufweisen (90°). Dafür wird der äusserste Punkt auf der *Handkontur* (*Fingertal*) bestimmt und eine Gerade erzeugt. Dies im rechten Winkel zur Referenzgeraden (Fig. 7.9).

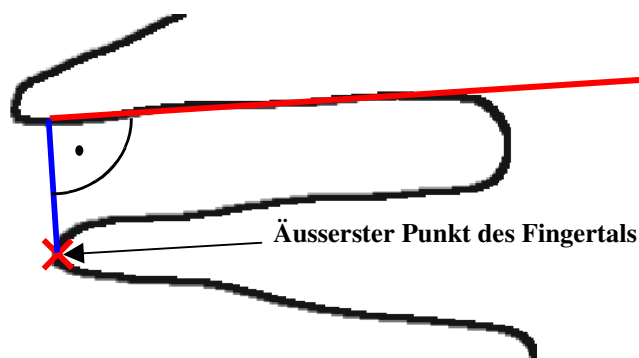


Fig. 7.9 Ermittlung der Schnittgeraden

7.6.4. Verfahren zur Ermittlung des Umfanges

Der Umfang eines beliebigen Objektes kann anhand seiner Kontur sehr einfach ermittelt werden. Der Umfang bedeutet nichts anderes als die Länge der Kontur [Zamp. 91]. Es müssen nur die Anzahl der Bildpunkte in der Kontur gezählt werden, und schon ist der Umfang berechnet.

$$U = (n_g + \sqrt{2} \cdot n_u) \cdot a$$

n_g und n_u sind die Anzahl der geraden, bzw. ungeraden Schritte des Konturcodes und a ist der Abstand zwischen zwei Bildpunkten

Ein ungerader Wert des Konturcodes bedeutet, dass der nächst folgende Bildpunkt diagonal neben dem vorhergehenden liegt. Deshalb muss dieser mit dem Faktor $\sqrt{2}$ versehen werden (Fig. 7.10).

Da bei einem *Konturvektor* in der *eVision*-Funktion der Konturcode optional gewählt werden kann (Kap. 7.5.1.2), ist es sehr einfach, die diagonal liegenden Bildpunkte zu eruieren.

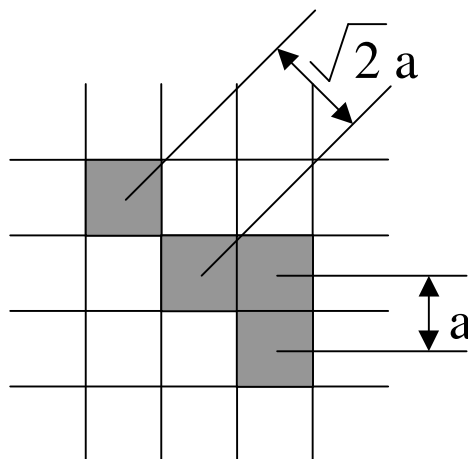


Fig. 7.10 Distanz zwischen zwei benachbarten Bildpunkten

7.6.5. Verfahren zur Ermittlung der Fläche

Die Fläche eines beliebigen Objektes kann mit Hilfe seiner Kontur auf eine graphische Art ermittelt werden. Dazu wird die Kontur in ein leeres Bild eingezeichnet und das Bild von oben nach unten untersucht. Es wird bei jeder Y- Koordinate der erste *Konturbildpunkt* von links und der erste Konturbildpunkt von rechts her bestimmt. Die so eingeschlossenen Bildpunkte ergeben eine Teilfläche ΔF (Fig. 7.11). Nun werden die nächsten zwei Punkte ermittelt und dabei die ΔF fortlaufend addiert. Am Ende resultiert die gesamte Anzahl der Bildpunkte die von der Kontur eingeschlossen sind (inkl. Kontur) d.h. die Fläche.

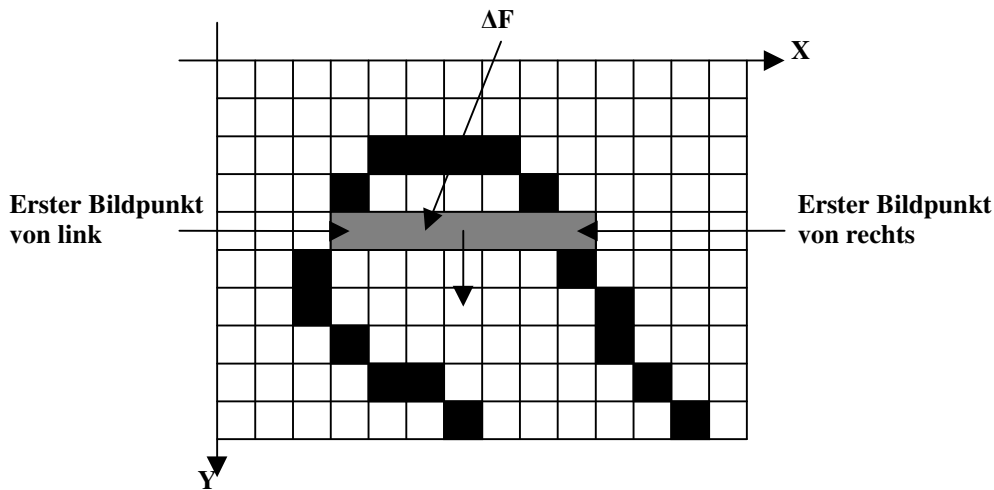


Fig. 7.11 Methode zur Flächenermittlung

7.6.6. Verfahren zur Ermittlung der Fingerlänge

Ein erstes Problem stellt sich gleich an den Anfang. Von wo bis wo ist die Fingerlänge definiert? In unserem Fall haben wir festgelegt, dass die Länge eines Fingers von der Mitte der *Schnittgerade* (Kap. 7.6.3) bis zum Fingerspitz reicht.

Der Fingerspitz wird dadurch ermittelt, dass eine Gerade im rechten Winkel zur *Referenzgeraden* von oben auf den Finger geschoben wird. Dort wo sie den Finger zuerst schneidet, befindet sich der Fingerspitz (Fig. 7.12). Die Bestimmung der Mitte der Schnittgerade ist eine kleine Handrechnung und muss hier nicht näher erläutert werden. Nun muss nur noch die Distanz zwischen diesen zwei Punkten berechnet werden und die Länge des Fingers ist ermittelt.

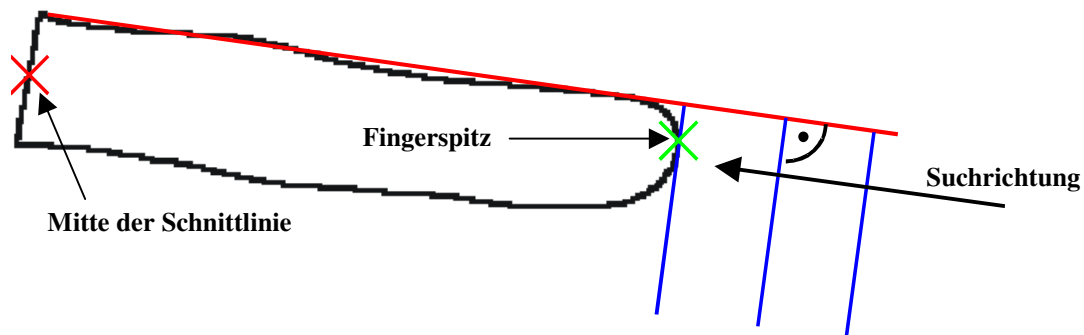


Fig. 7.12 Definition des Fingerspitz

7.6.7. Verfahren zur Ermittlung der Fingerdicke

Ein grosses Problem stellt die Tatsache dar, dass die Finger der selben Hand unterschiedliche Winkel zwischen sich selber und der Y-Achse aufweisen. Dies bedeutet, dass zur Messung immer wieder eine neue Referenz eingeführt werden müsste, um keine grossen Abweichungen zu erhalten. Zur Lösung dieses Problems wurde die Gerade an den Finger gelegt, die als *Referenzgerade* für alle Messungen dienen soll.

Die Ermittlung der Fingerdicke ist auf eine graphische Art gelöst worden. Dieses Verfahren ist zeitsparender als irgendwelche Vergleichsalgorithmen.

Die Fingerkontur wird zuerst in ein leeres Bild eingezeichnet. Nun wird das so entstandene Bild um den Winkel gedreht, der zwischen der y-Achse und der Referenzgerade liegt (Fig. 7.13).

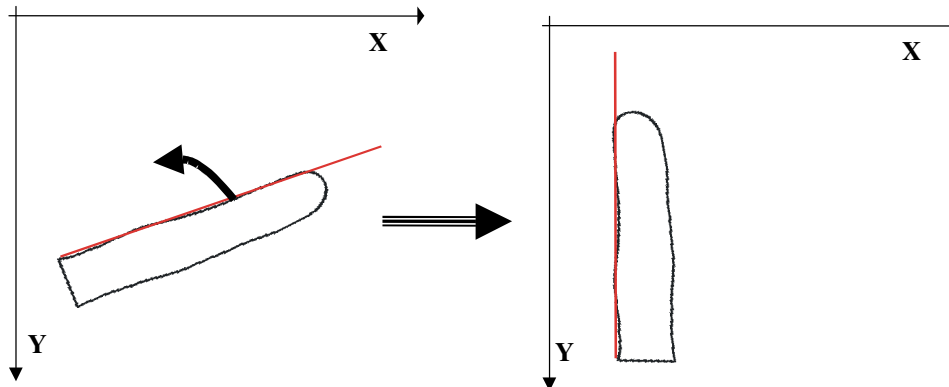


Fig. 7.13 Drehen des Fingers

Durch das Drehen des Fingers kommt er parallel zur Y- Achse zu liegen. Dadurch kann die Ermittlung der Dicke immer mit dem selben Verfahren durchgeführt werden. Es ist nicht mehr abhängig von der Lage des Fingers.

Das Bild wird anschliessend von oben nach Unten untersucht. Bei jeder Y- Koordinate wird der erste *Bildpunkt* der Kontur von links her und der erste Bildpunkt von rechts her gesucht (Fig. 7.14).

Die Distanz zwischen diesen beiden Bildpunkten beschreibt die Dicke des Fingers an der jeweiligen Stelle. Dieser temporäre Wert wird dann zwischengespeichert und es wird der nächste Wert ermittelt. Falls dieser grösser ist, als der bereits vorhandene, wird der alte Wert mit den Neuen überschrieben. So wird immer der grösste Wert abgelegt. Wenn das ganze Bild untersucht wurde, ist die maximale Dicke ermittelt.

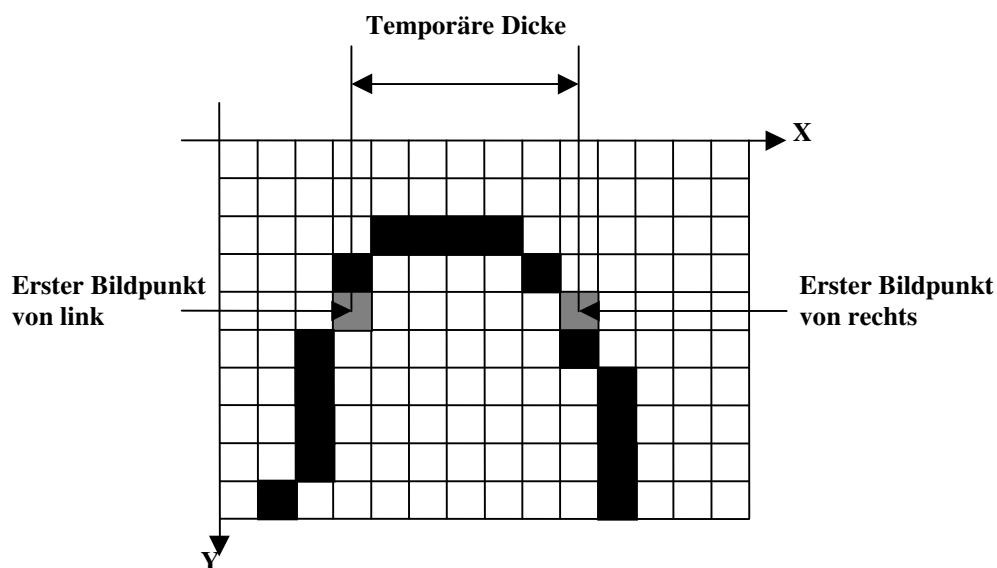


Fig. 7.14 Methode zur Ermittlung der Fingerdicke

8. Identifikationsalgorithmen

8.1. Einführung

Um im Endeffekt eine Personenidentifikation durchführen zu können, genügt es nicht nur die Hand auszumessen. Jetzt muss anhand eines Algorithmus ermittelt werden, ob die eingelegte Hand mit einer im Archiv vorhandenen Hand übereinstimmt, das heisst identifiziert werden kann. Zu diesem Zweck wird für jede Hand im Archiv ein *Ähnlichkeitsmass* berechnet.

8.1.1. Gaussche Glockenkurve

Als erstes wird jedem einzelnen Attribut der zu vergleichenden Hand (Hand aus dem Archiv) ein Wahrscheinlichkeitswert zugewiesen. Daraus resultieren 16 Werte im Bereich von 0 (kleine Wahrscheinlichkeit) und 1 (hohe Wahrscheinlichkeit). Dieses Zuweisen wird mit Hilfe der Gaussverteilung berechnet. Die Gaussverteilung ist die am häufigsten eingesetzte Verteilung bei der Wahrscheinlichkeitsrechnung.

$$w_i = e^{-\frac{1}{2} \left(\frac{w_{akt} - w_{ref}}{k \cdot b} \right)^2}$$

- w_{akt} → Wert der zu identifizierenden Hand (aktuelle Hand)
 w_{ref} → Wert der Hand aus dem Archiv (Referenz Hand)
 $(k \cdot b)$ → Standardabweichung $\sigma = (k \cdot b)$;
Produkt aus dem *Toleranzband* (kap. 12.5.1) und einem Toleranzfaktor (einstellbarer Parameter)

8.1.2. Mittelwert

Als nächster Schritt muss aus den 16 Einzelwerten ein Mittelwert gebildet werden. Für die Mittelwertbildung kennt man viele verschiedene Arten [Pap. 94]. Für die Identifikation kommen aber nur jene in Frage, die mit einer *Gewichtung* arbeiten, da die einzelnen Merkmale nicht alle gleich gut aus dem Bild gemessen werden können und deshalb bei der Verrechnung unterschiedlichen Einfluss haben müssen.

8.1.2.1. Gewichteter Mittelwert

$$\bar{m} = \frac{g_l \sum_4 w_{li} + g_d \sum_4 w_{di} + g_u \sum_4 w_{ui} + g_f \sum_4 w_{fi}}{4[g_l + g_d + g_u + g_f]}$$

w_i → Wahrscheinlichkeitswert (0..1)
 g → Gewichtung für [l]→Länge [d]→Breite [u]→Umfang [f]→Fläche

8.1.2.2. Mittelwertsbestimmung mit gewichteten Potenzen

Dieser Algorithmus kann nur dann eingesetzt werden, wenn die Werte (w) zwischen 0 und 1 liegen.

$$\bar{m} = \frac{\sum_4 w_{li}^{g_l} + \sum_4 w_{di}^{g_d} + \sum_4 w_{ui}^{g_u} + \sum_4 w_{fi}^{g_f}}{16}$$

w_i → Wahrscheinlichkeitswert (0..1)
 g → Gewichtung (Potenz) für [l]→Länge [d]→Breite [u]→Umfang [f]→Fläche

8.1.3. Auswertung

Beim gewichteten Mittelwert werden die einzelnen Werte unterschiedlich gewichtet und haben deshalb unterschiedlich starken Einfluss auf das Resultat. Jedoch ist der Einfluss nicht gross genug. Falls z. B. nur ein Wert stark abweicht, hat dies auch mit einer hohen Gewichtung keinen genügend grossen Einfluss auf das Resultat. Dieses Phänomen ist für die Personenidentifikation nicht geeignet, da dort bei zu grosser Abweichung gar keine Übereinstimmung mehr resultieren darf. Deshalb liegt die Idee mit dem Potenzieren sehr nahe. Bei dieser Methode werden Werte, die stark abweichen, das heisst viel kleiner 1 sind, durch das Potenzieren noch mehr „abgeschwächt“ und haben durch das einen sehr grossen Einfluss auf das Resultat. Diese Formel wurde aus Überlegungen hergeleitet und ist in keinem uns bekannten Lehrbuch zu finden. Hat aber bei den Messungen fast ideales Verhalten gezeigt.

Zum Schluss wird der berechnete Wert mit dem Faktor 100 Multipliziert. Es resultiert ein *Ähnlichkeitsmass* der eingelegten Hand im Bezug zur Hand im Archiv.

Dieser Wert variiert zwischen 100 (Totale Übereinstimmung) und 0 (Keine Ähnlichkeit vorhanden).

9. Grobpflichtenheft

9.1. Einführung

9.1.1. Zweck des Dokuments

Definition der zentralen funktionalen und nichtfunktionalen Anforderungen an das Softwaresystem.

9.2. Funktionale Anforderungen

9.2.1. Das Programm Identifier

Das Programm *Identifier* ist in erster Linie ein Softwareprogramm, mit dem eine Personenidentifikation durchgeführt werden kann. In unserem Fall wird damit eine Zutrittskontrolle realisiert. Mit der dazugehörigen Hardware wird ein Bild der Hand aufgenommen. Diese Hand wird durch das Programm ausgemessen. Anschliessend werden die ausgemessenen Daten mit den im *Archiv* vorhandenen Daten verglichen. Dann entscheidet das Programm, ob die eingelegte Hand identifiziert wurde oder nicht. Die Identifikation muss online durchgeführt werden können. Der Benutzer muss die Parameter zur Messung und Identifikation der Hand einstellen können. Zusätzlich können neue Personen (Hände) dem Archiv zugefügt werden.

9.2.2. Archivviewer

Der Archivviewer erlaubt es dem Benutzer das *Archiv* durchzusehen, und dabei auch Personen (Hände) aus diesem zu entfernen. Das aktuelle Archiv beinhaltet alle Personen (Hände), die identifiziert werden können, also zutrittsberechtigt sind.

9.2.3. Datenhaltung

Die Daten der Hand werden in einem eigenen, noch zu spezifizierenden Datenformat in einer „txt“- Datei abgelegt.

9.2.4. Zusammenhänge (UML- Diagramm)

Das folgende Klassendiagramm zeigt die wichtigsten Zusammenhänge auf. (UML-Notation)

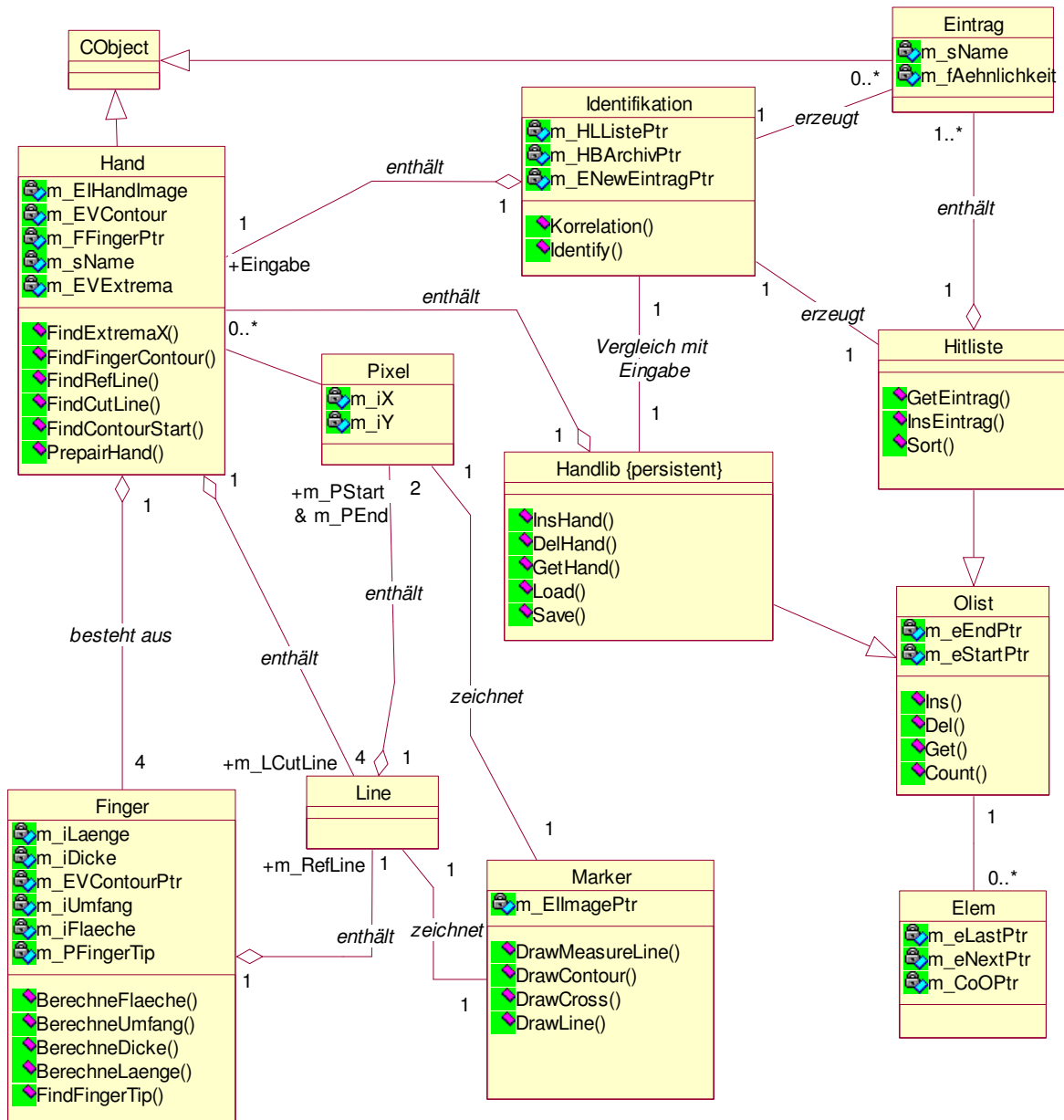


Fig. 9.1 Klassendiagramm OOA

9.3. Funktionale Anforderungen im Einzelnen

9.3.1. Use Cases

9.3.1.1. Übersicht

Vorkommende Aktoren

Benutzer: Person (Programmbenutzer), welcher eine Hand ins Archiv eintragen/ entfernen oder eine Identifikation durchführen will.

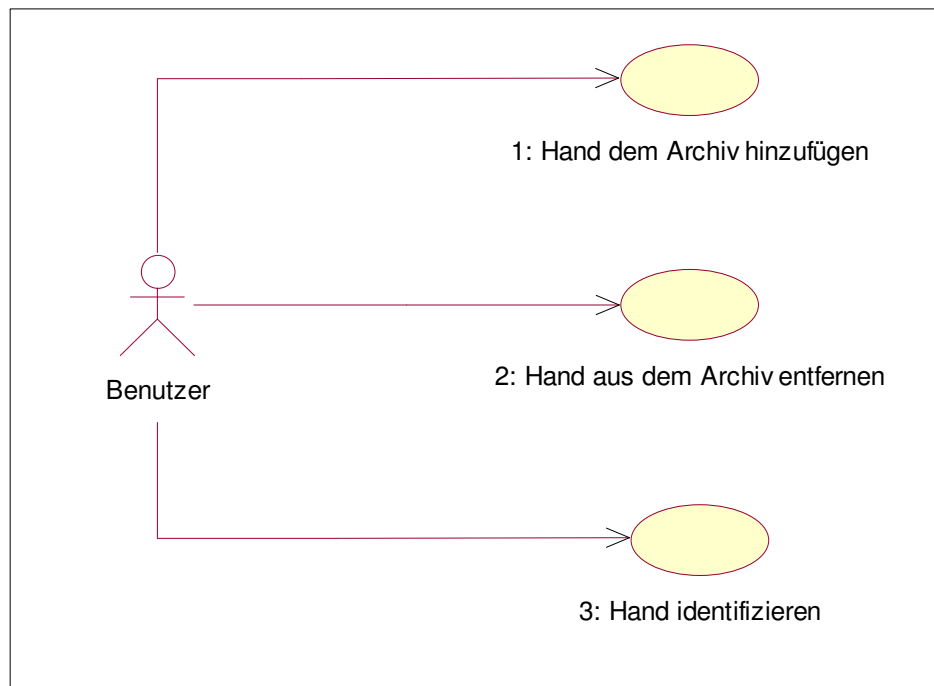


Fig. 9.2 Use Case Diagramm

9.3.1.2. Use Case Beschreibungen

UC 10: Hand dem Archiv hinzufügen

Kurzbeschreibung: Der Benutzer kann Hände erfassen und dem Archiv hinzufügen.

Auslöser (Aktor): Benutzer

Vorbedingung: Es befindet sich eine Hand im Erfassungsgerät

Ablauf:
Bild der Hand einlesen
Bild analysieren
Werte ins Archiv speichern

UC 20: Hand aus dem Archiv entfernen

Kurzbeschreibung:	Der Benutzer kann Hände aus dem Archiv entfernen.
Auslöser (Aktor):	Benutzer
Vorbedingung:	Es befindet sich mindestens eine Hand im Archiv.
Ablauf:	Hand, welche gelöscht werden soll, im Archiv auswählen Hand löschen Sicherheitsabfrage quittieren

UC 30: Hand identifizieren

Kurzbeschreibung:	Der Benutzer kann eine Hand identifizieren lassen.
Auslöser (Aktor):	Benutzer
Vorbedingung:	Es wurde über das Erfassungsgerät ein Bild eingelesen oder es wurde ein Hand geladen
Ablauf:	Bild analysieren Analysedaten mit Archivdaten vergleichen und für jede Hand aus dem Archiv einen Eintrag in der Hitliste erzeugen. Top-Eintrag der Hitliste prüfen, ob die Ähnlichkeit genügend hoch ist. Wenn ja, den Namen der erkannten Hand ausgeben.

9.4. Nichtfunktionale Anforderungen

9.4.1. Einführung

Das Programm ist nur lauffähig mit der dazugehörigen Hardware. (Kamera, Framegrabber Karte, Dongle etc.)

9.4.2. Benutzerschnittstelle

Die Bedienung des Programmes *Identifier* erfolgt mit Hilfe der Maus und Tastatur.

9.4.3. Leistungsanforderungen

Die Berechnung aller *Attribute* soll in einem gewissen Zeitrahmen bleiben, ebenfalls die Identifikation darf eine bestimmte Durchführungszeit nicht überschreiten, da dies für eine evtl. praktische Anwendung nicht tragbar ist.

9.4.4. Weitere Anforderungen bezüglich Vorgehen und Hilfsmittel

Für die Analyse und den Entwurf sind die Methoden der objekt-orientierten SW-Entwicklung anzuwenden. Für die Notation ist UML zu verwenden. Als Programmiersprache ist Visual C++ einzusetzen [Krug, 98] [Chap. 98].

10.2. Klassenbeschreibung

10.2.1. Klasse Hand

Ein Objekt der Klasse Hand entspricht einer ganzen Hand mit allen Daten inklusive des Namens. Die Klasse enthält alle notwendigen Funktionen zur Erzeugung und Benutzung dieser Daten. Eine Hand enthält 4 Finger.

Attribute

m_EIHandImage	Bild der Hand im „tif“ Format
m_EVContour	Kontur der Hand
m_FFinger	Zeigerarray auf die 4 Finger der Hand (0..3, Zeigefinger...kleiner Finger)
m_LCutLine	Array mit den Schnittgeraden welche die Finger von der Hand trennen
m_EVExtrema	Vektor, der alle lokalen Extremas der X-Werte enthält
m_sPfad	Pfad des Bildes der Hand
m_sName	Name der Hand (Besitzer)
m_PCContourStart	Koordinate des Startpunktes zur Konturermittlung

Funktionen

FindExtremaX()	Findet alle lokalen Extremas in X-Richtung
FindContour()	Findet die Kontur der Hand
FindCutLine()	Findet die Gerade welche den jeweiligen Finger von der Hand trennt
FindRefLine()	Approximiert eine Gerade an jeweils eine Seite jedes Fingers
FindContourStart()	Findet den Startpunkt zur Ermittlung der Kontur
GetHandImage()	Liefert einen Zeiger auf das Bild der Hand
GetExtremaPtr()	Liefert den Zeiger auf alle lokalen Extremas der X-Werte
GetFinger()	Liefert den Zeiger auf den gewünschten Finger
GetCutLine()	Liefert den Zeiger auf die gewünschten Schnittgerade
PrepairHand	Wandelt das Originalbild der Hand in ein „brauchbares“ Bild um (Entfernung aller Elemente auf dem Bild ausser der Hand selber)
SetName()	Setzt den Namen
GetName()	Liefert den Namen
SetPfad()	Setzt den Namen des Pfades für das Bild
GetPfad()	Liefert den Pfad des Bildes zurück
SetHandImage()	Übergibt ein Bild an die Hand
Draw()	Visualisiert die Daten aus der Hand (Kontur, Extremas, Länge etc.)
GetExtremaIndex()	Liefert die Position des gewünschten Extremas zurück

10.2.2. Klasse Finger

Ein Objekt der Klasse Finger entspricht einem Finger mit all seinen Daten. Die Klasse enthält alle notwendigen Funktionen zur Erzeugung und Benutzung dieser Daten.

Attribute

m_EVContourPtr	Zeiger auf die Kontur des Fingers
m_fDicke	Dicke des Fingers (an der breitesten Stelle) in mm
m_fLaenge	Länge des Fingers (an der längsten Stelle) in mm
m_fUmfang	Umfang des Fingers (Länge der Kontur) in mm
m_fFlaeche	Fläche des Fingers in mm ²
m_PFingerTip	Koordinate des Fingerspitzes
m_LRefLinePtr	Referenzgerade parallel zum Finger
m_iNummer	Fingernummer (0..3, Zeigefinger...kleiner Finger)
m_LLaegeLine	Linie zur Illustration der Länge
m_LDickeLine	Linie zur Illustration der Dicke

Funktionen

GetDicke()	Liefert die Dicke
GetLaenge()	Liefert die Länge
GetUmfang()	Liefert den Umfang
GetFlaeche()	Liefert die Fläche
GetDickeLine()	Liefert die Linie an der breitesten Stelle des Fingers. Wird zur Illustration der Breite auf dem Bild benötigt
GetLaengeLine()	Liefert die Linie an der längsten Stelle des Fingers. Wird zur Illustration der Länge auf dem Bild benötigt
GetFingerTip()	Liefert die Koordinate des Fingerspitzes
GetNummer()	Liefert die Fingernummer (0..3, Zeigefinger...kleiner Finger)
GetContour()	Liefert den Zeiger auf die Fingerkontur
GetRefLinePtr()	Liefert den Zeiger auf die Referenzlinie
SetRefLinePtr()	Initialisiert die Referenzlinie
SetContour()	Initialisiert die Fingerkontur
SetNummer()	Setzt die Fingernummer (0..3, Zeigefinger...kleiner Finger)
SetFingerTip()	Setzt die Koordinate für den Fingerspitz
BerechneFinger()	Führt alle Berechnungen aus
BerechneUmfang()	Berechnet den Umfang
BerechneFlaeche()	Berechnet die Fläche
BerechneDicke()	Berechnet die Dicke
BerechneLaenge()	Berechnet die Länge
FindFingerTip()	Findet den Fingerspitz. Wird zur Berechnung der Länge benötigt
SetDicke()	Setzt die Dicke
SetLaenge()	Setzt die Länge
SetUmfang()	Setzt den Umfang
SetFlaeche()	Setzt die Fläche

10.2.3. Klasse Identifikation

Die Klasse Identifikation enthält eine „aktuelle Hand“ und vergleicht diese mit den Händen aus dem Archiv. Die Vergleichsergebnisse werden in einer Hitliste mittels Einträge abgelegt. Die Klasse enthält alle notwendigen Funktionen für den Vergleich.

Attribute

m_HLListePtr	Zeiger auf die Hitliste
m_HBArchivptr	Zeiger auf die Handlib (Archiv mit den zu vergleichenden Händen)
m_ENewEintragptr	Zeiger auf einen Eintrag

Funktionen

Korrelation()	Vergleicht die „aktuelle“ Hand mit einer Hand aus dem Archiv und liefert einen Trefferwert (0..100).
Identify()	Liefert zurück, ob der Top-Eintrag der Hitliste über dem Schwellwert liegt (Hand wurde identifiziert) oder unter dem Schwellwert liegt (Hand konnte nicht identifiziert werden). Der Schwellwert kann der Benutzer über das GUI verändern.

10.2.4. Klasse Hitliste

Ein Objekt der Klasse Hitliste enthält Einträge welche den Vergleich der aktuellen Hand mit den Händen aus dem Archiv darstellt. Es wird durch die Identifikation erzeugt. Die Klasse enthält alle notwendigen Funktionen zum Manipulieren der Hitliste.

Funktionen

GetEintrag()	Liefert den Zeiger auf den gewünschten Eintrag in der Liste zurück
InsEintrag()	Fügt einen Eintrag am Anfang der Liste ein
Sort()	Sortiert die Liste nach dem <i>Ähnlichkeitsmass</i> in den Einträgen

10.2.5. Klasse Eintrag

Ein Objekt der Klasse Eintrag modelliert einen Eintrag in der Hitliste. Die Klasse enthält alle notwendigen Funktionen zur Manipulation eines Eintrags.

Attribute

m_sName	Steht für den Namen der Hand aus dem Archiv, mit der verglichen wurde
m_fAenlichkeit	Ist ein Mass für die Ähnlichkeit der Hände (Bsp. 100 → Totale Übereinstimmung, 0 → Keine Übereinstimmung)

Funktionen

SetName()	Setzen des Namens
SetAenlichkeit()	Setzen der Ähnlichkeit
GetName()	Liefert Namen
GetAenlichkeit()	Liefert den Wert für die Ähnlichkeit

10.2.6. Klasse Handlib1

Die Klasse Handlib1 modelliert ein Archiv von Händen und ihrer Attribute. Dies wird mit einer doppelt verketteten Liste realisiert.

Funktionen

InsHand()	Fügt eine Hand am Anfang in die Liste ein
DelHand()	Löscht eine Hand aus der Liste
GetHand()	Liefert den Zeiger auf die gewünschte Hand zurück
Save()	Speichert die ganze Liste ab
Load()	Lädt die ganze Liste
Decode()	Entschlüsselt den Char-Array und schreibt die Daten in die übergebene Hand
CharToInt()	Wandelt eine Ziffer (Char) in den entsprechenden int-Wert um
Potenz()	Berechnet die Potenz anhand der Basis und dem Exponenten

10.2.7. Klasse Olist

Basisklasse für eine doppelt verkettete Liste.

Attribute

m_eEndPtr	Zeiger auf letztes Element in der Liste
m_eStartPtr	Zeiger auf erstes Element in der Liste

Funktionen

Ins()	Fügt ein Element am Anfang in die Liste ein
Del()	Löscht ein Element aus der Liste
Get()	Liefert den Zeiger auf das gewünschte Element zurück
Count()	Liefert Anzahl Element in der Liste

10.2.8. Klasse Pixel

Die Klasse Pixel modelliert einen Bildpunkt. Die Klasse enthält alle notwendigen Funktionen zum Setzen und Abrufen der Attribute.

Attribute

m_iX	X-Koordinate des Bildpunkts
m_iY	Y-Koordinate des Bildpunkts

Funktionen

GetX()	Liefert X-Wert zurück
GetY()	Liefert Y-Wert zurück
SetX()	Setzt X-Wert
SetY()	Setzt Y-Wert
Draw()	Zeichnet ein Kreuz an der Bildpunktcoordinate in der aktuellen Farbe

10.2.9. Klasse Line

Die Klasse Line modelliert eine Linie mit ihrem Startpunkt und ihrem Endpunkt.

Attribute

m_PStart	Startkoordinaten der Linie
m_PEnd	Endkoordinaten der Linie
m_PVLinePtr	Vector mit allen Punkten auf der Linie inkl. Start- und Endpunkt

Funktionen

GetWinkel()	Liefert den Winkel zwischen der Linie und der X-Achse
CreateVector()	Füllt den Vektor mit den Punkten auf der Linie (inkl. Start- und Endkoordinate)
GetLineVector()	Liefert einen Zeiger auf den Vektor, der alle Punkte der Linie enthält (inkl. Start- und Endkoordinate)
GetStart()	Liefert die Startkoordinate der Linie
GetEndt()	Liefert die Endkoordinate der Linie
SetStart()	Setzt die Startkoordinate der Linie
SetEnd()	Setzt die Endkoordinate der Linie
Draw()	Zeichnet eine Linie vom Startpunkt zum Endpunkt in der aktuellen Farbe

10.2.10. Klasse konstanten

Die Klasse konstanten enthält alle notwendigen Konstanten, die immer wieder im Programm gebraucht werden und schnell geändert werden können.

PI	Pi
PIXELDIMENSION	Effektive Auflösung (Pixelgröße in mm)
WIDTH	Breite eines Bildes
HIGHT	Höhe eines Bildes
MAXFINGERDICKE	Maximale Dicke eines Fingers
MAXFINGERABSTAND	Maximaler Abstand (Spreizung) der Finger
MAXFINGERLÄNGE	Maximale Länge eines Fingers
SIGMALAENGE	Parameter zur Einstellung der Gausschen Glockenkurve
SIGMABREITE	Parameter zur Einstellung der Gausschen Glockenkurve
SIGMADICKE	Parameter zur Einstellung der Gausschen Glockenkurve
SIGMAUMFANG	Parameter zur Einstellung der Gausschen Glockenkurve

10.2.11. Struct Elem

Element einer dynamisch doppelt verketteten Liste.

Attribute

m_eLastPtr	Zeiger auf das vorhergehende Element der Liste
m_eNextPtr	Zeiger auf das folgende Element in der Liste
m_CoOPtr	Zeiger auf ein Objekt der Klasse CObject

11. Bedienungsanleitung

11.1. Installation

1. Easy Access 5.2 installieren
Folgende Lizenzen müssen vorhanden sein:
 - EasyImage
 - EasyObject
 - EasyMeasure
2. Identifier.exe in das Zielverzeichnis kopieren
3. Identifier starten

Bemerkungen:

Falls identifier.exe nicht gestartet werden kann, muss Visual Studio 6.0 Enterprise Edition installiert werden um alle nötigen dll- Dateien eizufügen oder aufzufrischen.

11.2. Der Identifier

Nach dem Start von identifier.exe erscheint ein Fenster wie in Fig. 11.1

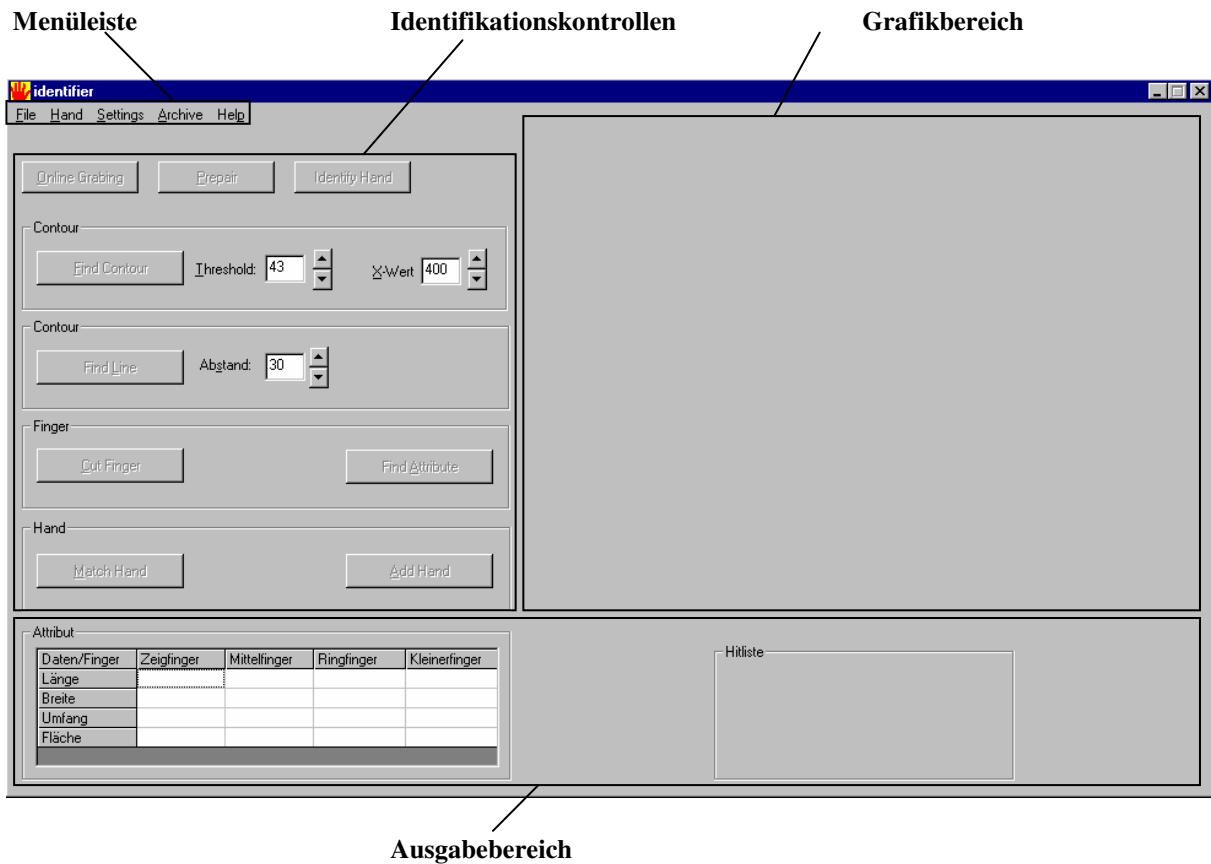
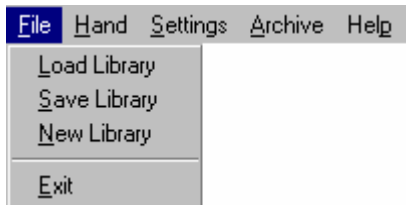


Fig. 11.1 Der Eröffnungsbildschirm des Identifiers

11.2.1. Die Menüleiste

Über die Menüleiste können die Einstellungen des Identifiers verändert werden. Es sind auch einige Befehle darin zu finden. Genauere Informationen darüber sind in den Kapiteln über die einzelnen Menüeinträge zu finden.

11.2.1.1. Filemenü



Load Library: Lädt ein neues Archiv, mit welchem Hände verglichen werden können.

Save Library: Speichert das aktuelle Archiv. Bei Programmstart ist diese Funktion nicht eingeschaltet, denn sie kann erst benutzt werden, wenn einmal ein Archiv geladen oder ein Neues erstellt wurde.

New Library: Legt ein neues leeres Archiv an. Es erscheint ein Dialogfeld in dem der Benutzer aufgefordert wird, einen Namen für das neue Archiv einzugeben (siehe Fig. 11.2). Nach der Eingabe wird ein neuer Ordner im Verzeichnis in dem der Identifier sich befindet angelegt. In diesem neuen Ordner wird dann das neue Archiv mit dem gewählten Namen erstellt.

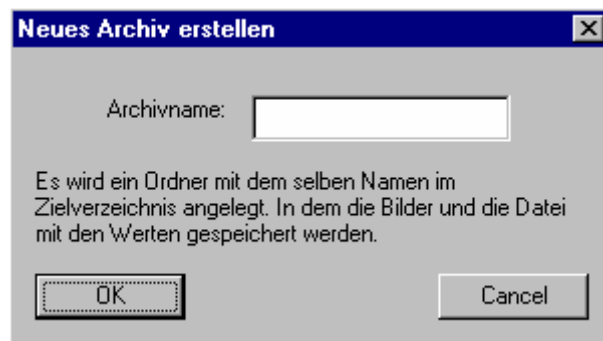


Fig. 11.2 Dialog um ein neues Archiv zu erstellen

Exit: Beendet die Anwendung.

11.2.1.2. Handmenü

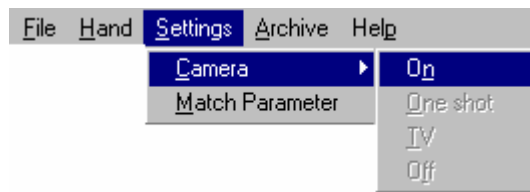


Load Hand: Es erscheint das Open-Standarddialogfeld. Hier kann das .tif-Bild einer erfassten und präparierten Hand geladen werden. Diese sind im Ordner „Hands“ zu finden. Diese Funktion kann erst nach dem Laden oder dem Erstellen eines neuen Archivs benutzt werden.

Add Hand: Wenn eine Hand nicht identifiziert werden konnte, kann mit dieser Funktion eine nicht identifizierte Hand dem aktuellen Archiv hinzugefügt werden.

Abort Identification: Mit dieser Funktion wird die angefangene Identifikation abgebrochen und der Identifier je nachdem ob die Kamera eingeschaltet ist (siehe Kap. 11.1.1.3) in die entsprechende Starteinstellung zurückgesetzt. Mit dieser Funktion kann eine Identifikation, die durch den Identify-Button gestartet wurde nicht abgebrochen werden.

11.2.1.3. Settingsmenü



Camera → On: Schaltet die Kamera ein und zeigt im Grafikbereich das Bild an das beim Einschalten entsteht.

Camera → One shot: Der Identifier funktioniert ähnlich wie ein Fotoapparat, wobei der Online Grabbing-Button dem Auslöser entspricht.

Camera → TV: Der Computer macht einmal pro Sekunde ein Bild und zeigt es im Grafikbereich an. Durch drücken des Online Grabbing-Buttons, wird automatisch das aktuelle Bild gehalten und die Kamera in den One shot-Modus umgeschaltet.

Camera → Off: Schaltet die Kamera ab.

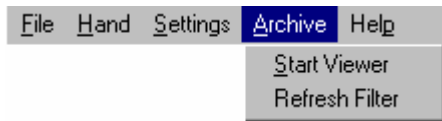
Match Parameter: Es erscheint ein Dialogfeld in dem die Parameter für die Identifikation eingestellt werden können. (siehe Fig. 11.3)



Fig. 11.3 Parametereinstellungen für die Identifikation

- Gewicht:** Mit diesem Parameter wird der Einfluss der Merkmale eingestellt. Hohes Gewicht bedeutet, dass das entsprechende Merkmal mehr Auswirkung auf die Ähnlichkeit hat.
- k:** Wird dieser Parameter erhöht, werden die Teilähnlichkeiten der entsprechenden Merkmale höher. Ist dieser Wert zu gross, ist die Wahrscheinlichkeit dass Unbefugten Eintritt gewährt wird, sehr gross. Ist dieser Wert zu klein, wird es schwierig erkannt und identifiziert zu werden.
- Schwelle:** Ist die Ähnlichkeit des Top-Hitlisten-Eintrages grösser als die Schwelle, gilt die Hand als identifiziert.

11.2.1.4. Archivemenü



Start Viewer: Es wird der Viewer gestartet. Mit dem Viewer kann das aktuelle Archiv durchgesehen und Hände aus diesem gelöscht werden. (siehe Kap. 11.2)

Refresh Filter: Es wird ein Bild aufgenommen und im Ordner des Identifiers abgespeichert unter dem Namen „empty.tif“. Dieses Bild wird zur Aufbereitung der Handaufnahme bei der Identifikation benutzt. Es wird eine einfache Bildsubtraktion durchgeführt. Damit das System richtig funktioniert, muss darauf geachtet werden, dass der Aufnahmetisch leer und die Beleuchtung eingeschaltet ist, bevor die Funktion Refresh Filter ausgeführt wird.

11.2.1.5. Help



About identifier Es erscheint ein Dialogfeld welches Informationen über das Programm Identifier enthält.

11.2.2. Grafikbereich

Im Grafikbereich wird das aktuelle Bild angezeigt.

11.2.3. Ausgabebereich

Im Ausgabebereich werden die Resultate der Identifikation dargestellt. Links im Ausgabebereich sind die Messwerte der ausgemessenen Hand zu sehen, während rechts die Hitliste angezeigt wird.



Fig. 11.4 Der Ausgabebereich

11.2.4. Identifikationskontrollen

Über diese Kontrollen wird die Identifikation einer geladenen oder mit der Kamera aufgenommenen Hand durchgeführt.

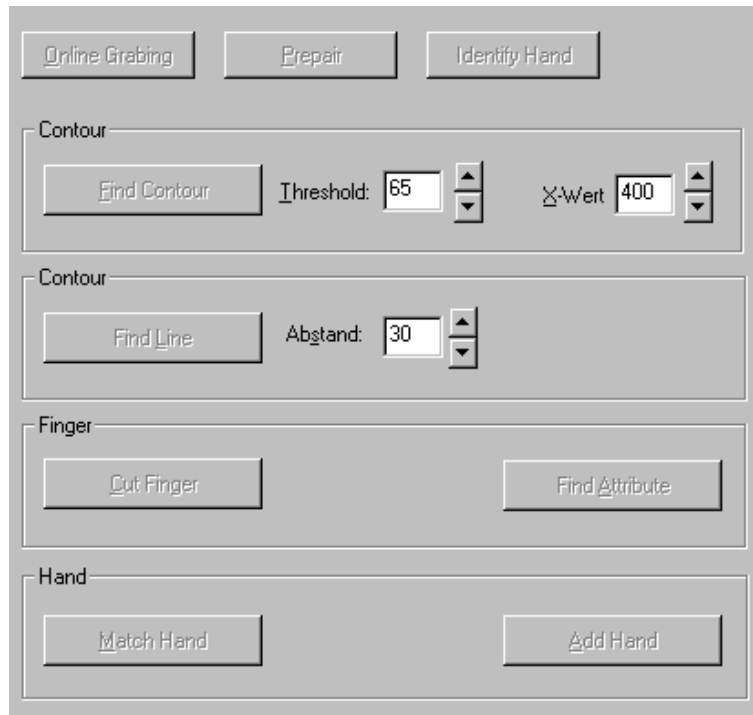


Fig. 11.5 Identifikationskontrollen

- Online Grabbing:** Ein neues wird von der Kamera ein neues Bild eingelesen und für die weitere Bearbeitung zur Verfügung gestellt.
- Prepair:** Durch Drücken des Prepair-Buttons wird der Filter (empty.tif) vom aktuellen Bild subtrahiert, wodurch wirklich nur die eingelegte Hand welche von Interesse ist übrig bleibt.
- Identify Hand:** Es wird eine komplette Identifikation durchgeführt, die nicht unterbrochen werden kann.
- Find Contour:** Es wird die Kontur ermittelt und angezeigt.
- Threshold:** Minimalster Grauwert der Kontur.
- X-Wert:** X-Koordinate des Startwertes für die Suche nach der Kontur.
- Find Line:** Legt an jeden Finger (Ausnahme Daumen) eine Referenzgerade.
- Abstand:** Wert um die Ausgleichsgerade zu beeinflussen.
- Cut Finger:** Generiert pro Finger eine eigene Kontur.
- Find Attribute:** Misst die ganze Hand aus und trägt die Werte im Ausgabebereich in die Tabelle ein.
- Match Hand:** Vergleicht die Werte der ausgemessenen Hand mit den Werten die sich im aktuellen Archiv befinden. Die Hitliste wird im entsprechenden Bereich ausgegeben. Entweder es erscheint eine Meldung, dass die Hand nicht identifiziert wurde, oder der Name der identifizierten Person wird angezeigt.
- Add Hand:** Wurde eine Hand nicht erkannt, kann diese zum aktuellen Archiv hinzugefügt werden.

11.3. Der Viewer

Der Viewer wird über die Menüleiste aufgerufen (Archiv → Start Viewer). Wenn der Viewer gestartet wird erscheint ein Dialogfeld wie es in Fig. 11.6 abgebildet ist.

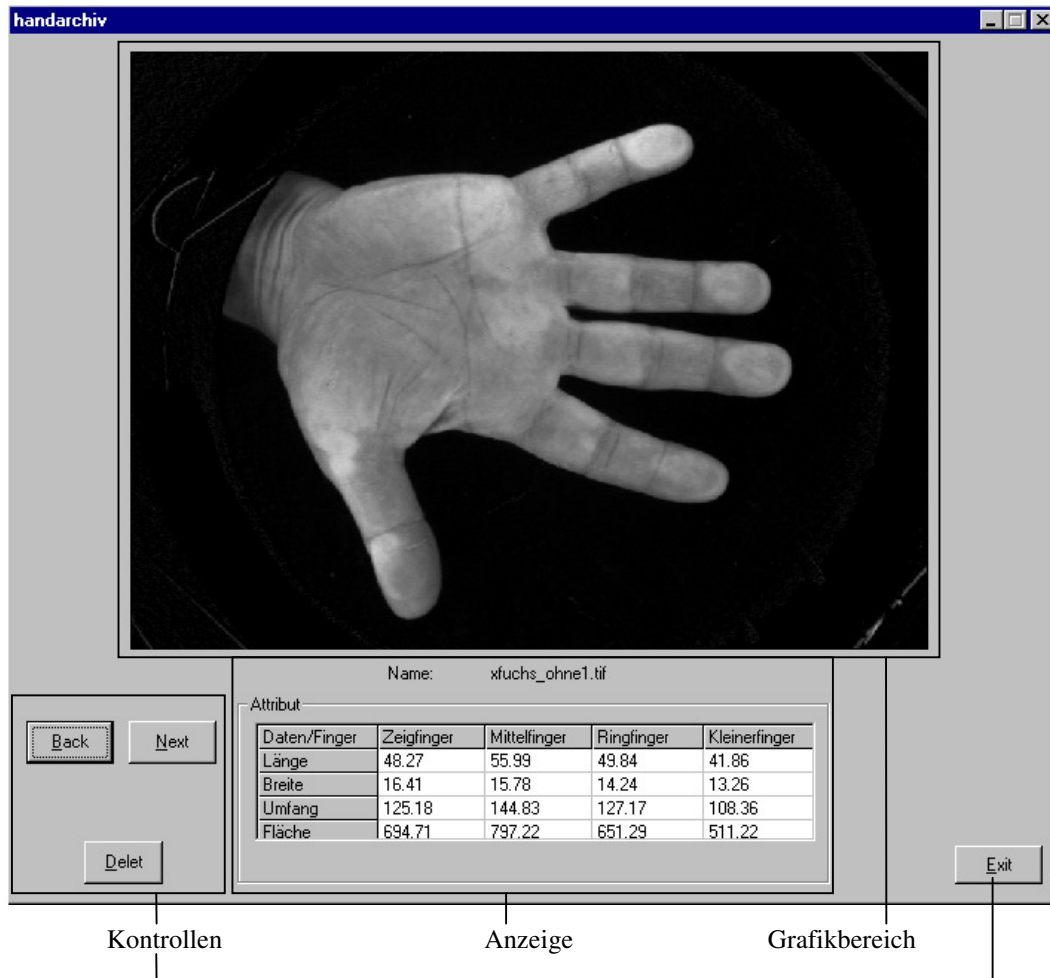


Fig. 11.6 Der Viewer

11.3.1. Grafikbereich

Im Grafikbereich wird das aktuelle Bild angezeigt.

11.3.2. Anzeige

Es wird der Name und alle Attribute der aktuellen Hand angezeigt.

11.3.3. Kontrollen

Next, Back:

Mit diesen Kontrollbuttons kann durch das aktuelle Archiv navigiert werden (Next = Nächste Hand; Back = vorherige Hand). Das Archiv ist wie ein Ring organisiert. Ist man am Ende des Archivs und drückt Next, so beginnt der Durchlauf wieder am Anfang. Analog dazu, wenn am Anfang des Archivs Back gedrückt wird, wird die letzte Hand angezeigt.

Delete:

löscht die angezeigte Hand aus dem aktuellen Archiv. Falls noch eine Hand im Archiv ist, wird die nächste Hand angezeigt. Wird die letzte Hand gelöscht, wird der Grafikbereich grau und die Attribute bleiben leer.

Exit:

Mit Exit wird der Viewer geschlossen und man kehrt zum Identifier zurück.

12. Tests und Resultate

12.1. Einführung

In einer ersten Phase ist es nötig, möglichst viele Hände von verschiedenen Personen mit dem *Identifier* aufzunehmen und auszumessen (Fig. 12.1). Aus diesen Testdaten kann einerseits eine Aussage über die Genauigkeit des Messverfahrens gemacht werden, und andererseits werden anhand dieser Daten die Parameter für die eigentliche Identifikation hergeleitet (Kap. 8). Für diesen Zweck wurden von 20 Testpersonen je 3 verschiedene Aufnahmen der rechten Hand gemacht und anschliessend mit dem Identifier ausgemessen.

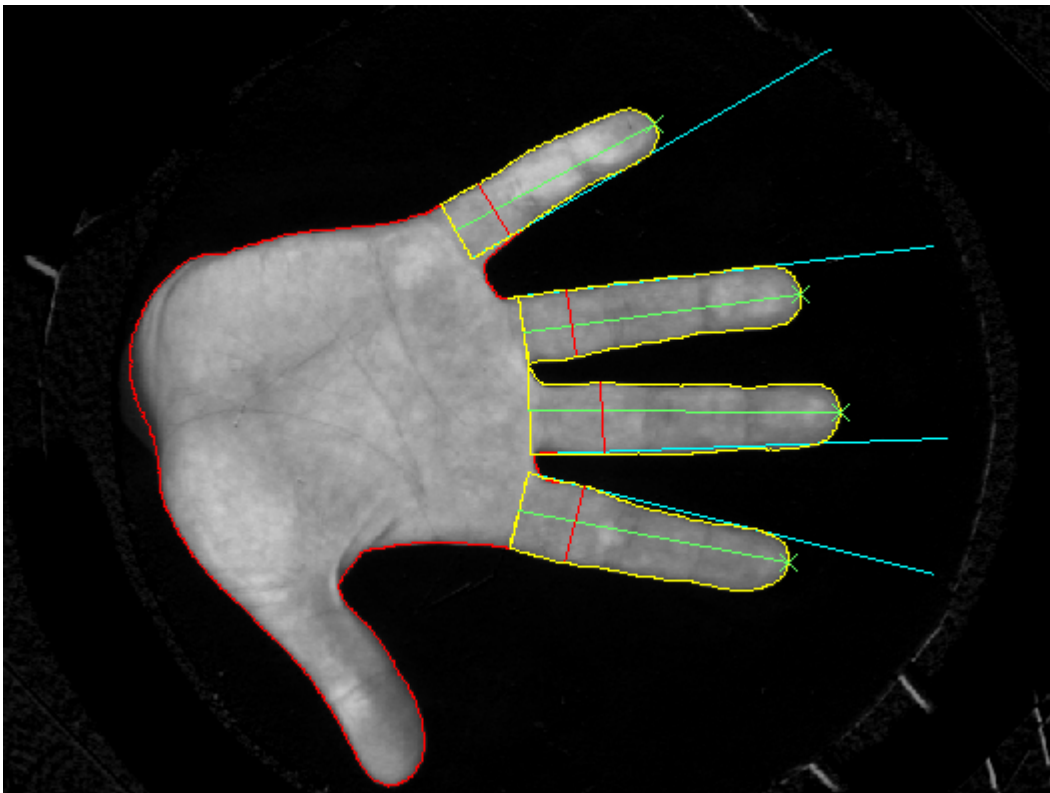
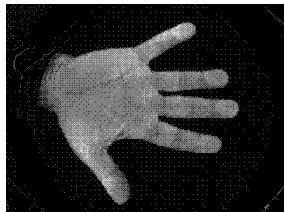


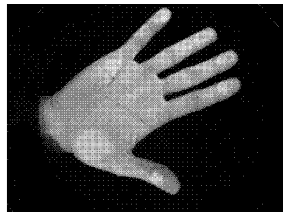
Fig. 12.1 Mit dem Identifier ausgemessene Hand

12.2. Testpersonen zur Bestimmung der Toleranzbänder

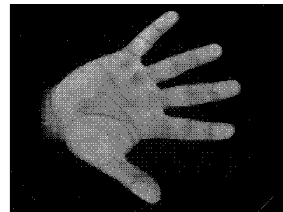
Es wurde einerseits versucht Personen zu finden, die sehr unterschiedliche Hände aufweisen (klein, gross, dick, knochige usw.). Aber andererseits wurde auch darauf geachtet, dass Personen ausgewählt werden, die sehr ähnliche Hände haben, um dem Identifier auch „Knacknüsse“ zu liefern. Die folgenden Testpersonen erfüllen diese Kriterien.



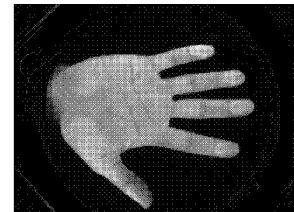
xfuchs



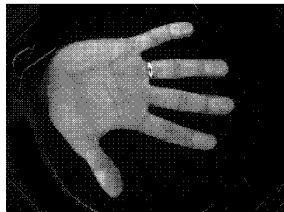
ahuesser



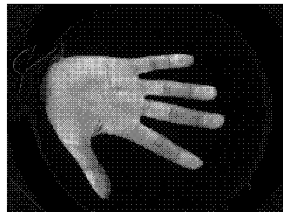
azehnder



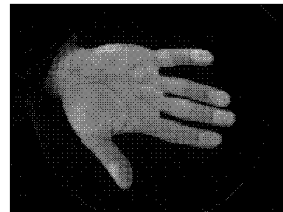
denzler



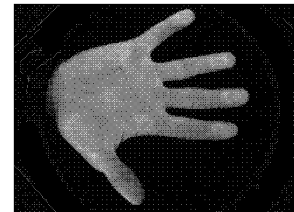
dweber



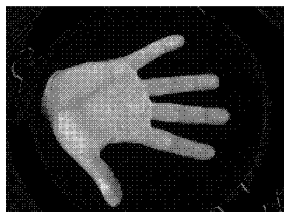
earnold



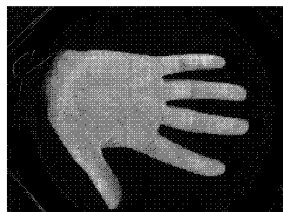
hgrossmann



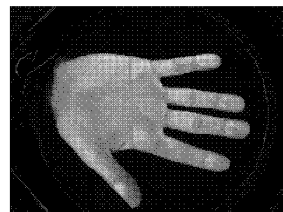
mflammer



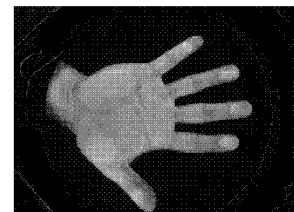
mstella



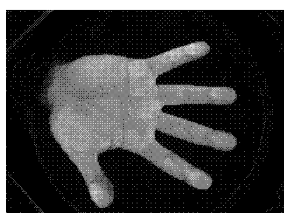
proffler



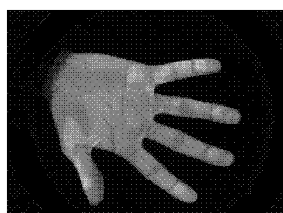
rbucher



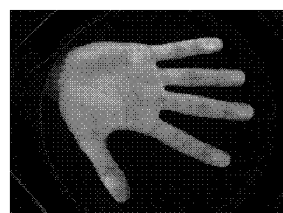
rfuchs



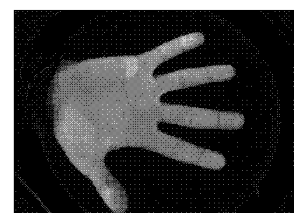
rschaerer



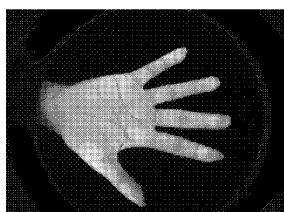
shuber



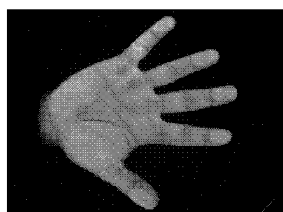
smueller



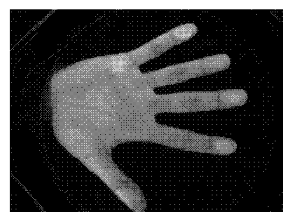
sruzzini



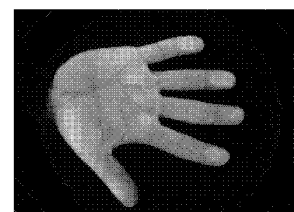
aarnold



aruegg



slang



mburri

12.3. Messwerte

12.3.1. Meßwerte Fingerdicke

	xfuchs	sruzzini	smueller	shuber	rschaerer	rfuchs	aarnold	denzler	ahuesser	earnold
Zeigefinger	16.256	15.24	13.97	13.97	14.7	14.22	13.46	14.73	13.97	14.22
	16.51	15.24	14.47	14.73	14.22	14.22	13.2	15.24	13.72	13.97
	17.01	15.49	14.47	14.47	14.47	13.97	13.2	14.48	13.72	13.97
Mittelfinger	16	14.73	14.22	14.22	14.47	14.22	13.21	14.48	14.22	13.21
	15.74	14.73	14.99	14.48	14.22	14.22	13.21	14.48	13.97	12.95
	16	14.73	14.99	14.48	14.48	14.22	12.95	14.22	13.97	13.21
Ringfinger	14.73	13.46	12.95	13.2	14.73	13.21	12.2	13.72	13.97	11.94
	14.48	13.2	13.2	13.72	13.97	13.46	11.68	13.97	13.46	11.18
	14.7	13.46	13.2	13.46	14.22	12.95	11.68	13.46	13.46	11.43
Kleiner Finger	13.2	11.93	11.94	13.2	13.46	12.2	10.67	11.68	11.94	11.18
	13.46	11.43	12.44	13.2	12.44	12.2	10.67	12.2	11.43	11.18
	13.7	11.93	12.44	12.95	12.7	12.45	10.41	11.94	11.93	10.92

	hgross- mann	m- flammer	mstella	azehnder	proffler	rbucher	mburri	aruegg	dweber	slang
Zeigefinger	14.99	15.24	14.48	15.75	15.75	15.49	13.97	15.24	14.73	15.24
	14.73	15.49	14.48	16	16	14.99	13.97	15.24	14.48	14.73
	15.24	15.49	14.48	16	15.75	16.26	14.22	15.49	14.22	14.73
Mittelfinger	14.22	14.99	13.21	15.24	15.24	14.73	14.47	14.73	14.99	14.99
	14.22	14.99	13.46	15.49	15.75	14.48	14.22	14.73	14.48	14.22
	14.48	15.24	13.21	15.49	14.99	14.48	14.73	14.99	14.41	14.22
Ringfinger	12.7	14.22	12.7	14.48	13.97	13.72	12.7	13.46	14.22	14.22
	12.7	14.48	12.45	14.73	14.22	13.21	12.7	13.46	14.22	13.97
	12.95	14.48	12.45	14.48	13.46	13.21	12.95	13.72	14.05	13.97
Kleiner Finger	12.7	12.95	11.18	13.46	12.95	12.45	11.94	12.45	12.45	12.7
	12.7	13.21	11.18	13.46	12.95	12.19	11.94	12.19	11.94	12.19
	12.95	12.95	11.18	13.2	12.95	12.45	11.94	12.7	11.96	12.95

Die Grau hinterlegten Felder sind Werte die ausserhalb der gewählten Toleranz liegen (Kap. 12.5.1).

Die Messwerte sind in der Fig. 12.2 visualisiert.

12.3.2. Meßwerte Fingerlänge

	xfuchs	sruzzini	smueller	shuber	rschaerer	rfuchs	aarnold	denzler	ahuesser	earnold
Zeigefinger	48.28	53.11	60.8	56.24	53.12	46.33	52.18	52.4	54.86	54.64
	48.22	53.45	58.54	56.01	53.11	46.24	54.46	54.1	55.17	54.62
	48.36	54.38	58.54	56.42	53.56	45.54	52.2	51.37	54.85	54.85
Mittelfinger	56.14	59.5	65.37	62.35	56.66	53.7	53.41	57.16	58.46	57.5
	55.96	59.7	64.1	62.7	56.41	53.95	53.8	57.25	58.85	57.53
	55.48	59.96	64.1	62.72	57.41	53.34	53.66	56.69	58.06	57.79
Ringfinger	49.43	54.35	57.23	59.52	52.33	50.66	48	51.87	51.16	50.38
	49.1	54.7	56.14	59.05	51.12	50.04	48.14	51.66	51.83	49.82
	49.56	54.8	56.14	58.99	51.32	49.34	48.82	51.42	51.28	50.33
Kleiner Finger	41	44.06	49.3	49.78	44.51	38.5	38.89	42.1	42.73	40.36
	42.14	43.62	48.64	48.7	38.69	39.03	39.72	42.39	42.49	40.04
	41.95	43.96	48.64	47.97	43.16	41.03	39.51	42.32	41.69	40.41

	hgross- mann	m- flammer	mstella	azehnder	proffler	rbucher	mburri	aruegg	dweber	slang
Zeigefinger	47.12	52.19	49.9	48.92	54.54	51.5	54.32	55.18	53.77	54.42
	47.12	52.51	49.87	49.31	54.47	53.35	53.67	54.67	54.23	52.43
	47.07	52.7	48.3	51.7	54.22	54.49	53.53	53.91	54.23	53.3
Mittelfinger	52.57	59.94	56.36	56.27	59.67	59.42	57.92	59.34	60.6	61.57
	52.55	59.24	56.39	56.39	59.67	60.51	56.95	58.79	60.84	60.83
	52.43	59.52	55.63	56.8	59.8	59.05	56.9	58.63	61.03	60.75
Ringfinger	46.33	52.66	49.74	50.34	55.91	53.97	55	53.97	55.14	54.88
	46.57	52.52	50.76	51.02	55.4	53.46	53.97	54.62	55.91	55.4
	46.95	53.02	50.07	51.07	55.13	55.79	54.1	53.38	56.23	55.15
Kleiner Finger	39.37	39.9	40.49	41.01	45.51	44.58	45.29	43.48	41.41	46.89
	39.12	42.27	40.23	42.22	44.72	43.69	44.42	43.9	42.01	47.65
	40.65	42.94	40.33	38.29	44.85	46.92	44.3	43.2	42.34	48.43

Die Grau hinterlegten Felder sind Werte die ausserhalb der gewählten Toleranz liegen (Kap. 12.5.1).

Die Messwerte sind in der Fig. 12.3 visualisiert.

12.3.3. Meßwerte Fingerumfang

	xfuchs	sruzzini	smueller	shuber	rschaerer	rfuchs	aarnold	denzler	ahuesser	earnold
Zeigefinger	124.3	134.63	149.64	132.33	133.18	116.44	128.94	132.32	133.4	133.63
	125	133.9	146.88	139.13	130.56	115.98	128.25	135.71	135.57	134.71
	125.52	138.53	146.88	138.8	134.56	115.23	128.04	127.7	135.17	131.63
Mittelfinger	143.77	149.81	161.88	156.47	144.16	138.43	132.07	142.24	144.93	143.41
	143.55	150.41	162.3	158	146.63	138.83	133.71	147.35	147	142.33
	142.87	152.58	162.3	156.26	146.56	138.12	133.67	140.76	145.23	144.49
Ringfinger	130.65	140.42	143.98	151.9	135.18	133.57	121.92	133.74	134.79	124.65
	127.45	141.55	139.28	150.1	133.63	130.77	118.31	137.24	135.12	122.01
	127.15	143.36	139.28	148.95	133.49	134.83	123.62	130.3	135.64	125.64
Kleiner Finger	106.65	110.7	120.47	125.1	114.12	93.57	98.36	105.44	108.45	100.19
	107.4	108.77	119.675	122.36	93.43	100	99.6	107.17	107.08	100.56
	109.17	110.14	119.675	122.86	108.6	96.58	98.79	105.43	106.29	100.09

	hgross- mann	m- flammer	mstella	azehnder	proffler	rbucher	mburri	aruegg	dweber	slang
Zeigefinger	120.7	130.37	126.14	125.1	137.34	133.19	133.94	138.8	135.49	136.84
	120.17	134.15	125.4	125.18	137.79	134.45	131.36	138.39	135.96	134.95
	122.81	133.52	123.29	131.95	136.92	138.14	131.93	137.24	137.34	134.52
Mittelfinger	133.18	151.17	142.64	143.06	151.47	151.11	142.65	149.83	155.26	156.2
	133.52	152.35	141.44	145.27	150.74	154.18	141.8	149.24	154.19	154.82
	135.72	152.48	139.79	146.44	148.78	149.9	141.89	147.93	152.42	153.36
Ringfinger	119.78	136.44	128.4	124.49	138.2	136.48	134.5	134.41	136.4	143.45
	120.58	137.48	129.66	129.81	136.74	135.89	131.01	137.35	143.75	142.44
	121.87	134.44	128.5	130.7	135.79	141.9	131.01	135.22	140.34	138.93
Kleiner Finger	101.32	100.81	101.27	105.3	112.76	111.93	113.08	109.25	105.56	119.04
	100.49	108.12	102.11	105.1	110.92	108.24	113.67	110.17	106.38	119.5
	105.5	108.62	102.48	98.43	112.02	116.98	111.16	108.62	107.45	122.15

Die Grau hinterlegten Felder sind Werte die ausserhalb der gewählten Toleranz liegen (Kap. 12.5.1).

Die Messwerte sind in der Fig. 12.4 visualisiert.

12.3.4. Meßwerte Fingerfläche

	xfuchs	sruzzini	smueller	shuber	rschaerer	rfuchs	aarnold	denzler	ahuesser	earnold
Zeigefinger	688	731.41	766	696.45	735.03	583.1	610.64	697.77	690.84	678.64
	686.25	723.93	782.45	723.3	686.51	574.26	604.77	730.97	696.5	662.45
	700	747.35	782.45	707.9	713.81	562.45	601.1	668.13	676.64	669.42
Mittelfinger	818.39	799.54	946.7	776.8	762.7	753.68	640.13	756.06	785.55	694.64
	801.22	819.6	854.83	814.06	773.87	754.64	626.39	752.06	777.54	680.39
	797	805	854.83	801.9	707.1	684.39	626.13	746.2	745.29	684.06
Ringfinger	651.93	666	675.67	775.4	697.7	587.48	499.35	633.65	628.19	566.9
	647.87	654.26	700	722.2	649.42	578.71	500.45	637.55	618.71	524.71
	659.22	668	700	711.22	652.49	569.03	491.35	622.13	618.71	546
Kleiner Finger	515.6	491.4	541.3	587.3	545.16	408.64	385.68	457.55	471.93	408.4
	514.13	479.35	552.5	588.13	446.84	441.29	379.16	466.7	449.5	397.81
	521.6	487.2	552.5	576.4	514.45	445.03	375.93	461.16	457.4	402.51

	hgross- mann	m- flammer	mstella	azehnder	proffler	rbucher	mburri	aruegg	dweber	slang
Zeigefinger	620.13	706.84	618.26	658	743.93	710.51	678	748.39	712.39	744.64
	610.25	729.29	616.77	676.84	763.22	702.84	654.71	739.68	709.16	717.03
	640.67	723.16	603.74	707.61	743.8	741.87	673.03	745.22	710.34	707.22
Mittelfinger	663.61	830.51	697.42	759.03	796.51	796.13	757.74	787.93	823.48	852.45
	655.22	814.64	704.26	794.26	823.09	788.19	772.26	778.45	810.45	806.9
	680	811.87	700.77	768.13	804.51	795.55	754.9	792	815.56	798
Ringfinger	550.97	668.39	641.74	641.74	741.93	662.45	638	661.1	711.03	704.39
	551.1	664.97	643.35	643.35	726.32	636.77	608.13	648.13	707.61	719.74
	566.13	677.16	634.64	634.64	669.42	664.32	636.9	662.9	706.61	707.68
Kleiner Finger	473.6	466.32	425.55	477.22	505.74	506.97	500.9	496.06	469.42	559.03
	467.68	507.74	436.13	500.77	510.64	492.7	493.48	491.03	470.2	551.55
	485.8	511.6	424.06	433.74	503.81	530.58	500.9	506.6	472.16	560.26

Die Grau hinterlegten Felder sind Werte die ausserhalb der gewählten Toleranz liegen (Kap. 12.5.1).

Die Messwerte sind in der Fig. 12.5 visualisiert.

12.4. Diagramme

In den folgenden Diagrammen sind die Merkmale graphisch dargestellt, und zwar zeigt jedes Diagramm ein Merkmal über alle Testpersonen auf. Pro Farbe (Testperson) ergeben sich 4 Balken (Finger), die jeweils einen einzelnen Finger darstellen. Pro Balken sind jeweils alle 3 Werte (3er-Paket) aufgezeigt. Daraus kann abgeleitet werden, dass falls die Balken spitze Enden aufweisen, die Messwerte pro 3er Paket eine grosse Streuung bzw. eine kleine Streuung aufweisen falls der Balken oben eine horizontale Fläche aufweist.

Weiter kann aus den Diagrammen sehr einfach die Verschiedenheit der Hände (*Merkmale*) herausgelesen werden. Falls alle 16 Balken einer Farbe genau gleich sind wie 16 Balken einer anderen Farbe, liegen zwei Hände von verschiedenen Personen vor, deren Hände aber identisch sind.

12.4.1. Diagramm Fingerdicke

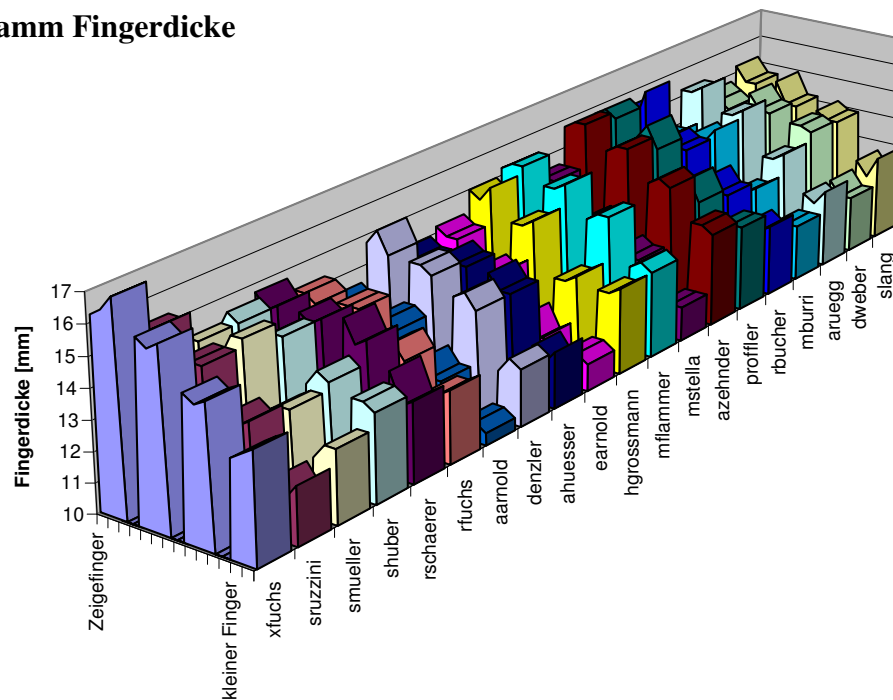


Fig. 12.2

12.4.2. Diagramm Fingerlänge

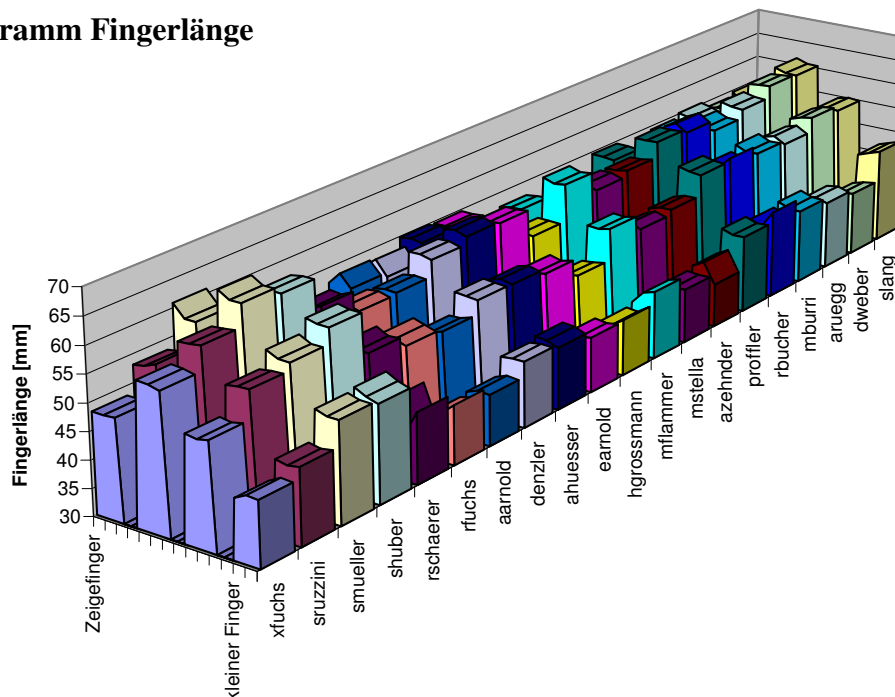


Fig. 12.3

12.4.3. Diagramm Fingerumfang

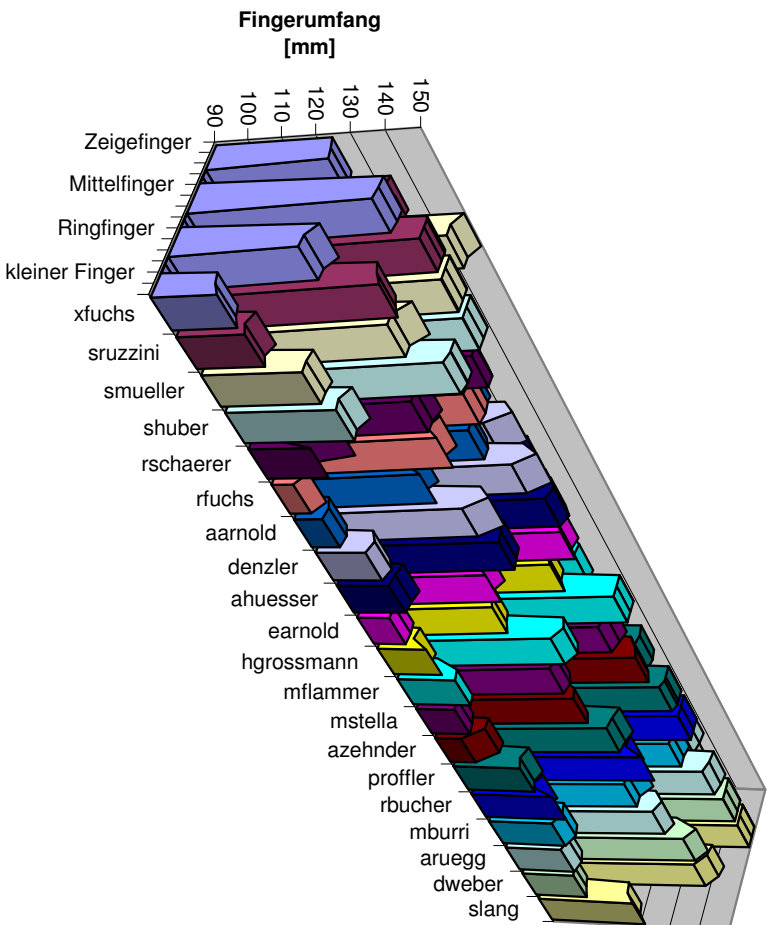


Fig. 12.4

12.4.4. Diagramm Fingerfläche

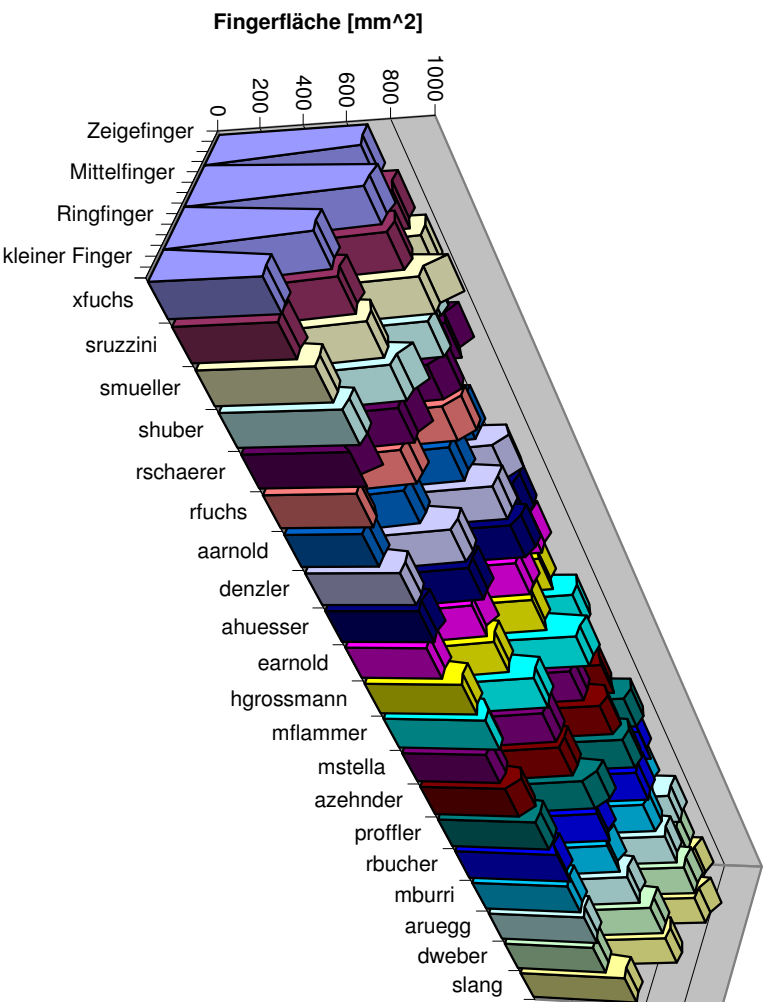


Fig. 12.5

12.5. Auswertung

Wie schon angedeutet liegen die Daten eines Fingers der gleichen Hand aber verschiedener Aufnahmen nicht direkt beieinander, sondern bewegen sich in einem bestimmten Bereich, dem sogenannten *Toleranzband*. Eine erste Aufgabe ist es nun, anhand der Messdaten die Toleranzbänder zu bestimmen. Wir einigten uns darauf, dass wir vier verschiedene Toleranzbänder ermitteln, und zwar für jedes Merkmal eines, das heisst Länge, Dicke, Umfang und Fläche. Es ist nicht sinnvoll für jedes einzelne Attribut ein Toleranzband zu bestimmen, da die einzelnen Toleranzen innerhalb der gleichen *Merkmale* sich nicht stark unterscheiden.

12.5.1. Bestimmung der Toleranzbänder

Ziel des *Toleranzbandes* ist es, dass 88% der Daten im entsprechenden Intervall sein müssen, dieser Wert ist frei gewählt worden. Dies bedeutet, dass 12% der Messwerte ausser Toleranz liegen dürfen. Wir deklarieren diese Messwerte, als Falschmessungen, ungenaue Messungen, und Messfehler. Daraus folgt, dass in jeder Messreihe (Länge, Breite etc.) 10 3er Pakete ausserhalb der Toleranz liegen dürfen (Bsp. 12.1). Die Toleranzen wurden zuerst willkürlich gewählt, und anschliessend die Pakete ermittelt, die nicht in dieser Toleranz liegen. Falls es weniger als 10 waren, wurde die Toleranz verkleinert, und falls es mehr waren, wurde die Toleranz vergrössert. Dies wurde iterativ durchgeführt, und am Schluss resultierten die 4 Toleranzbänder in welchen 88% der jeweiligen Pakete liegen.

Beispiel 12.1 aus der Tabelle „Fingerlänge“

	1	2	3	4	5	6	7	8	9	10
Zeigefinger	47.12	52.19	49.9	48.92	54.54	51.5	54.32	55.18	53.77	54.42
	47.12	52.51	49.87	49.31	54.47	53.35	53.67	54.67	54.23	52.43
	47.07	52.7	48.3	51.7	54.22	54.49	53.53	53.91	54.23	53.3
	54.49									
Mittelfinger	52.57	59.94	56.36	56.27	59.67	59.42	57.92	59.34	60.6	61.57
	52.55	59.24	56.39	56.39	59.67	60.51	56.95	58.79	60.84	60.83
	52.43	59.52	55.63	56.8	59.8	59.05	56.9	58.63	61.03	60.75

- $| \text{Wert 1} - \text{Wert 2} | = \text{Differenz 1} \rightarrow |48.92 - 49.31| = 0.39$
 $| \text{Wert 1} - \text{Wert 3} | = \text{Differenz 2} \rightarrow |48.92 - 51.70| = 2.78$
 $| \text{Wert 2} - \text{Wert 3} | = \text{Differenz 3} \rightarrow |51.70 - 49.31| = 2.39$

Toleranz (Länge) = 2.00 → 3er Paket liegt **ausser** Toleranz wegen Wert 3 (51.70)

- $| \text{Wert 1} - \text{Wert 2} | = \text{Differenz 1} \rightarrow |51.50 - 53.35| = 1.85$
 $| \text{Wert 1} - \text{Wert 3} | = \text{Differenz 2} \rightarrow |51.50 - 54.49| = 2.99$
 $| \text{Wert 2} - \text{Wert 3} | = \text{Differenz 3} \rightarrow |53.35 - 54.49| = 1.14$

Toleranz (Länge) = 2.00 → 3er Paket liegt **ausser** Toleranz wegen Wert 1 (51.50)

- $| \text{Wert 1} - \text{Wert 2} | = \text{Differenz 1} \rightarrow |61.57 - 60.83| = 0.74$
 $| \text{Wert 1} - \text{Wert 3} | = \text{Differenz 2} \rightarrow |61.57 - 60.75| = 0.82$
 $| \text{Wert 2} - \text{Wert 3} | = \text{Differenz 3} \rightarrow |60.83 - 60.75| = 0.08$

Toleranz (Länge) = 2.00 → 3er Paket liegt **in der** Toleranz

Das heisst nun, dass bereits 2 Werte ausser Toleranz liegen, es dürften also nur noch 8 weitere Pakete der insgesamt 80 Pakete ausserhalb liegen, damit dieses Toleranzband (Toleranz Länge = 2.00) als geeignet bezeichnet werden kann.

12.5.1.1. Toleranzbänder

- Länge: Toleranz $\pm 2\text{mm}$
- Breite: Toleranz $\pm 0.6\text{mm}$
- Fläche: Toleranz $\pm 50\text{mm}^2$
- Umfang: Toleranz $\pm 5\text{mm}$

In den Tabellen wurden die Messwerte welche ausser Toleranz liegen grau hinterlegt. Es sind immer 10 pro Tabelle, da 80 3er Pakete pro Merkmal (Tabelle) bestimmt wurden.

12.6. Systemtest

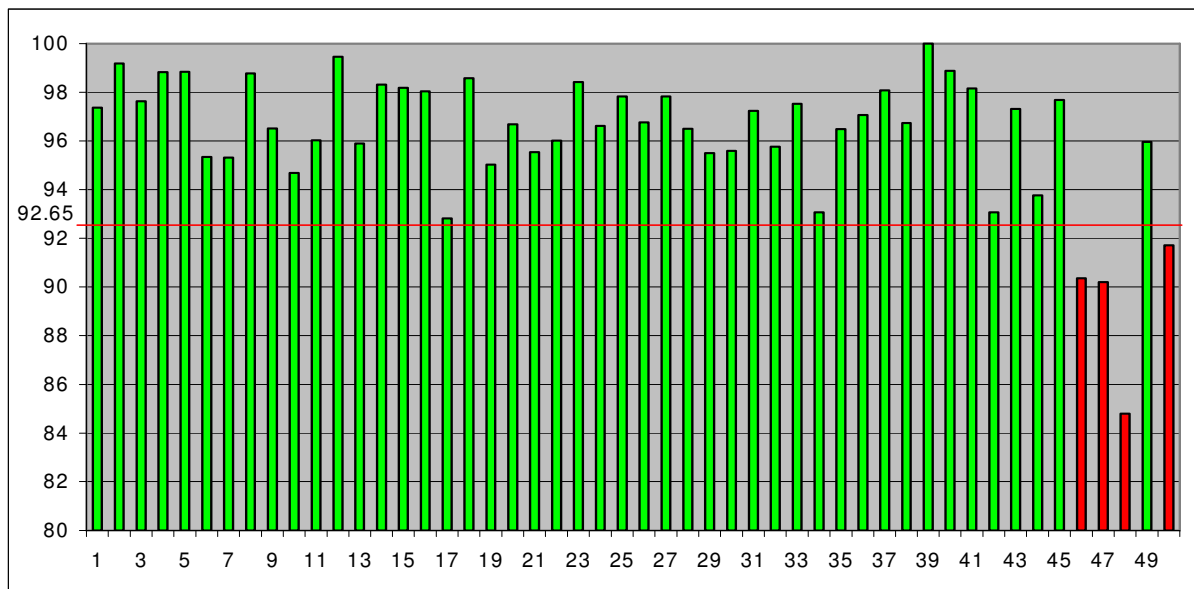
12.6.1. Einführung

Das Programm *Identifier* ist in erster Linie als Zutrittskontrolle einzusetzen. Dementsprechend wurde der Systemtest ausgelegt. Von 80 erfassten Personen wurden 50 zufällig ausgewählt. Im ersten Schritt wurde für jede dieser Personen die erste Hand als Referenz ausgemessen und im Archiv abgelegt. Im zweiten Schritt wurde die zweite und die dritte Hand der Testpersonen identifiziert (ausgemessen und mit dem Archiv verglichen). In diesem Schritt sollte jede Hand ihrem Besitzer beziehungsweise ihrer Referenzhand im Archiv zugeordnet werden können.

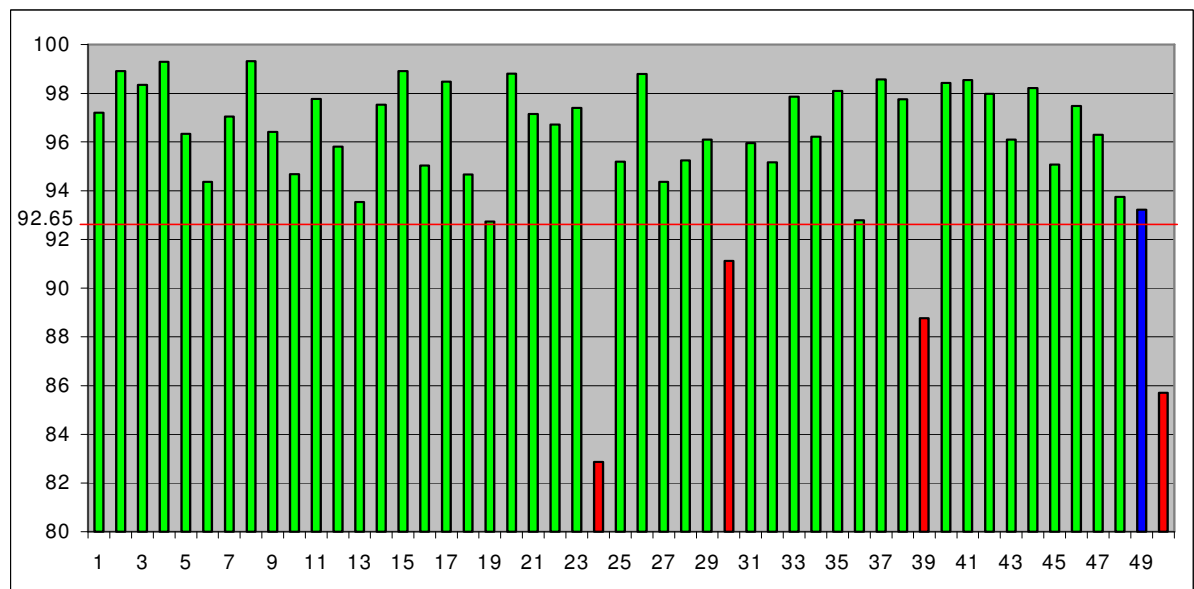
12.6.2. Zutritt erlaubt

Wie schon in der Einführung angesprochen wurde dieser Test zweimal durchgeführt. In den folgenden Diagrammen sind die vom Identifier berechneten Ähnlichkeiten eingetragen. Die genauen Werte und die Namen der Testhände kann dem Anhang entnommen werden.

12.6.2.1. Erster Durchgang



12.6.2.2. Zweiter Durchgang



Legende:

- Erfolgreich identifizierte Hände
- Nicht identifizierte Hände
- Verwechslung
- Schwellwert

12.6.2.3. Auswertung

Der oben beschriebene Test wurde zweimal durchgeführt und erbrachte folgende Resultate:

In 91% aller Messungen wurde die Person richtig identifiziert, 8% wurden nicht erkannt und eine Hand (Person) wurde verwechselt. Das heisst die Hand wurde einer Person zugewiesen zu welcher sie nicht gehört.

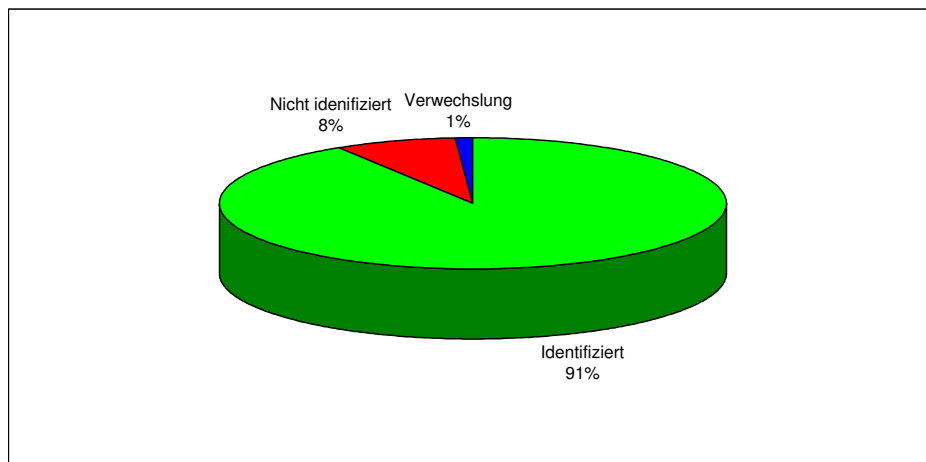


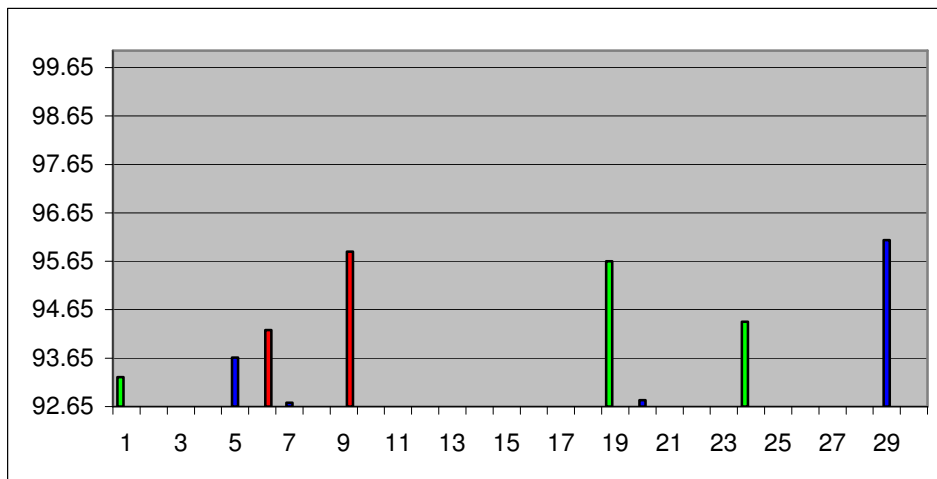
Fig. 12.6 Identifikationsgenauigkeit

12.6.3. Falscher Zutritt

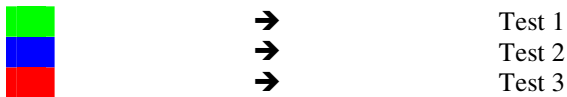
In einer weiteren Testreihe wurde untersucht, wie viele Personen die nicht berechtigt sind, trotzdem Zutritt erhalten, das heisst falsch identifiziert werden. Dafür wurden 30 Personen ausgewählt, die noch nicht im Archiv vorhanden sind. Dieser Test wurde drei Mal durchgeführt, um ein aussagekräftiges Resultat zu erhalten. Die genauen Werte dieses Test sind dem Anhang zu entnehmen.

12.6.3.1. Testdaten

Im folgenden Diagramm sind alle Werte aufgeführt, die über dem Schwellwert (92.65) liegen, das heisst fälschlicherweise identifiziert wurden.



Legende:



12.6.3.2. Auswertung

Das System arbeitet mit 90% Sicherheit. Im Schnitt werden von 30 Testpersonen ohne Zutrittsberechtigung trotzdem drei Personen nicht abgewiesen.

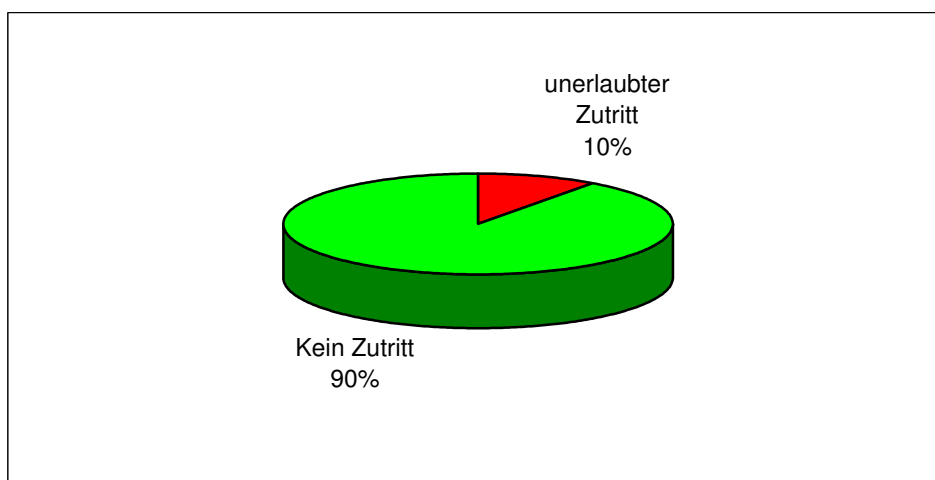


Fig. 12.7 Zutrittsicherheit

13. Epilog

13.1. Verbesserungsmöglichkeiten und Ideen

Zu Beginn der Studienarbeit wurde Zeit investiert um ein möglichst gutes Verfahren zu finden um die *Referenzgerade* zu ermitteln. Es wurden auch gute Ideen gefunden. Um erste Testresultate zu erhalten, wurde aber zuerst nur ein sehr einfacher Algorithmus implementiert. Die Ergebnisse dieses einfachen Algorithmus stellten sich als erstaunlich gut heraus. Nur in Extremfällen werden die Referenzgeraden nicht optimal eingelegt. Schliesslich beschlossen wir den Algorithmus nicht zu wechseln, da das Aufwand/Nutzen-Verhältnis zu gross ist.

Bei der Bestimmung des *Ähnlichkeitsmasses* ist vielleicht eine Verbesserung möglich. Wir haben zu Beginn der Arbeit den Aufwand für diese Bestimmung unterschätzt, was auch in der Planung klar ersichtlich ist. Eine erste Verbesserung wäre, wenn bei der Erfassung einer neuen Person nicht nur eine Hand, sondern mehrere Bilder ausgemessen und die Messwerte gemittelt werden. Um diese Änderung zu realisieren müsste das GUI für die Aufnahme neuer Personen nochmals stark überarbeitet, bzw. ein neues erstellt werden. In der folgenden Diplomarbeit wird dies sicher berücksichtigt werden.

Der ganze Algorithmus zur Bestimmung des *Ähnlichkeitsmasses* kann evtl. auch noch verbessert werden. Es sind Ideen aufgetaucht, die auch mit der Methode von mehreren Aufnahmen beim Eintrag in das *Archiv* arbeiten. Diese Idee wird mit grosser Wahrscheinlichkeit in die folgende Diplomarbeit einfließen.

13.2. Schlusswort

Die Studienarbeit war für uns beide eine sehr interessante und abwechslungsreiche Arbeit. Schon im 2. Studienjahr war uns klar, dass wir uns für eine Studienarbeit in der Bildverarbeitung entscheiden würden. Die Bildverarbeitung war für uns ein Gebiet, das uns sehr faszinierte. Doch leider wird dieses Fach in der Studienrichtung Elektrotechnik nicht angeboten.

Die Arbeit hat uns einen Einblick in die Grundlagen der Bildverarbeitung gegeben, die wir nun in der folgenden Diplomarbeit vertiefen werden. In der Bildverarbeitung kann das Wissen aus vielen verschiedenen Bereichen eingesetzt werden. Im speziellen konnten wir unser Gelerntes aus den vorhergehenden Studienjahren sinnvoll einsetzen. Wie z. B. aus der Mathematik, Informatik, Geometrie und auch der Physik. Es war eine sehr spannende Zeit in der wir viel neues mit auf unseren weiteren Weg mitnehmen konnten.

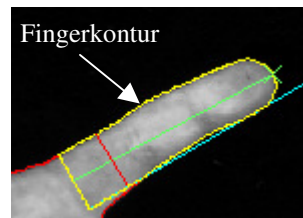
Wir danken Herrn Dr. H. Grossmann für seine Betreuung und Unterstützung bei unserer Arbeit und bei der Behebung von Problemen. Einen ganz speziellen Dank geht aber auch an alle Studenten, Dozenten und anderen Personen, die uns ihre Hand für die Vorstudie und Testphase zur Verfügung gestellt haben.

14. Literaturverzeichnis

- [Gonz. 87] Rafael C. Gonzalez/ Paul Wintz: Digital Image Processing, Addison- Wesley 1987, ISBN 0-201-11026-1
- [Zamp. 91] Piero Zamperoni: Methoden der digitalen Bildsignalverarbeitung, Vieweg 1991, ISBN 3-528-13365-1
- [Pap. 98] Lothar Papula: Mathematische Formelsammlung, Vieweg 1998, ISBN 3-528-44442-8
- [Pap. 94] Lothar Papula: Mathematik für Ingenieure und Naturwissenschaftler, Vieweg 1994, ISBN 3-528-04937-5
- [Krug. 98] David Kruglinski/ Georg Sheperd/ Scot Wingo: Inside Visual C++ 6.0, Microsoft 1998 ISBN 3-86063-461-5
- [Chap. 98] Davis Chapman: Visual C++ in 21 Tagen, Markt & Technik 1998 ISBN 3-8272-2035-1

15. Glossar

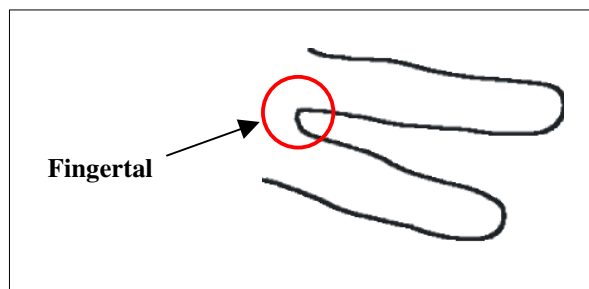
- Archiv:** Ein Archiv besteht aus Handbildern und den dazugehörigen Werten.
- Attribut:** Als Attribute werden die 16 Messwerte der Hand bezeichnet. Es sind dies die Dicke, die Länge, die Fläche und der Umfang jedes einzelnen Fingers.
- Ähnlichkeitsmass:** Das Ähnlichkeitsmass ist ein Wert zwischen 0 und 100. Es ist ein Mass für die Ähnlichkeit der aktuellen Hand zur Hand im Archiv.
- Bildpunkt:** Ein Bildpunkt beschreibt einen Punkt auf dem digitalisierten Bild. Er besitzt eine X und eine Y- Koordinate, die den Standort festhalten.
- Extrema:** Ein Extrema befindet sich dort, wo die erste Ableitung Null wird [Pap. 94].
- Fingerkontur:** Die Kontur eines Fingers.



- Fingermerkmal:** Siehe Identifikationsmerkmal
- Fingernummer:**
- | | | |
|---|---|----------------|
| 0 | → | Zeigefinger |
| 1 | → | Mittelfinger |
| 2 | → | Ringfinger |
| 3 | → | kleiner Finger |

Fingerspitz: (Kap. 7.6.6).

Fingertal: Das Fingertal ist ein kleiner Bereich zwischen zwei Fingern.



- Gewichtung:** Ein Parameter der Softwaremässig verändert werden kann und die Gewichtung der einzelnen Merkmale bestimmt (Kap. 8).
- Grauwert:** Der Grauwert beschreibt die Farbe (Grauwert) eines Bildpunktes. Der Grauwert variiert zwischen 0 (Schwarz) und 255 (Weiss).
- Handkontur:** Die Kontur einer Hand.
- Identifier:** Der Identifier ist der Name des Programmes (Software), das implementiert wurde.

Identifikationsmerkmal:	Als Identifikationsmerkmale werden die Merkmale bezeichnet, nach denen unterschieden wird. Es sind dies die Fläche, die Länge, der Umfang und die Dicke.
Konturbildpunkt:	Bildpunkt auf der Kontur.
Merkmal:	Siehe Identifikationsmerkmal
Referenzgerade:	Ausgleichsgerade an einen Finger (Kap. 7.6.2).
Schnittgerade:	Gerade welche den Finger vollständig von der Hand trennt (Kap. 7.6.3).
Toleranzband:	Dieses Band beschreibt den Bereich in dem sich die Attribute der selben Person aber verschiedener Aufnahmen bewegen darf, damit die Identifikation immer noch erfolgt (Kap. 12.2).
TopEintrag:	Eintrag in der Hitliste, der das grösste Ähnlichkeitsmass aufweist.
Schwellwert:	Falls das Ähnlichkeitsmass der aktuellen Hand über diesem Wert liegt, ist die Hand identifiziert. Der Schwellwert ist defaultmässig auf 92.65 eingestellt, kann aber durch den Benutzer verändert werden.