

1. Zusammenfassung (Abstract)

Wenn Körperteile erwähnt werden, die man zur Erkennung von Personen verwendet, hört man sehr wenig über das Ohr. Fingerabdrücke, die Geometrie der Hand, Gesichtserkennung, Retina Scan und andere biometrischen Merkmale werden heutzutage verwendet, um die Sicherheit und Leistungsfähigkeit der Personenidentifikation zu erhöhen. Dabei ist es belegt, dass auch das Ohr ein Objekt ist, das von Mensch zu Mensch verschieden ist. Ja mehr noch, das Ohr ist eines von den wenigen Körperteilen, welches seine Grundform ein Leben lang behält. Es soll bereits 56 Tage nach der Geburt seine Grundform besitzen, und diese bis zum Tode beibehalten. Bereits im zweiten Weltkrieg, wurden gefallene Marinesoldaten anhand ihrer Ohren identifiziert. Auch in der Personenfahndung der Polizei ist das Ohr kein unbeschriebenes Blatt und wird zur Erstellung von Phantombildern und Personenbeschreibung anhand verschiedener Merkmale klassifiziert.

In dieser Diplomarbeit wurde eine Personenidentifikation anhand der biometrischen Merkmale des linken Ohres realisiert. Als Grundlage dienen verschiedene Merkmale des Ohres. Es sind dies Ohrform, Ohrgrösse, Ohrlage, Form des Ohrläppchens und Breite der Ohrleiste. Diese Daten werden normiert und in einen Vektorraum abgebildet. Eine Verifikation bzw. Identifikation kann nun mittels minimal Distanzklassifikation durchgeführt werden. Ein Vorteil einer Identifikation anhand der Ohren liegt darin, dass sie berührungslos abläuft.

Es wurden verschiedene Aufnahmetechniken untersucht die gezeigt haben, dass die Lichtverhältnisse für ein brauchbares Resultat kontrollierbar sein müssen. Für diesen Zweck wurde speziell eine Aufnahmekabine errichtet, in der die Lichtverhältnisse konstant bleiben. In der Kabine befindet sich eine Schwarzweiss-CCD-Kamera. Diese ist über eine Framegrabber-Karte mit dem PC verbunden.

Es wurde ein Softwareprogramm „Earyfier“ entwickelt, das eingesetzt werden kann, um Bilder von Ohren aufzunehmen. Diese Ohrenbilder können ausgemessen und die Daten in ein Archiv abgelegt werden. Weiter besteht die Möglichkeit, mit dem ausgemessenen Ohr eine Verifikation bzw. Identifikation durchzuführen.

Die Algorithmen zur Extraktion des Ohres und zur Berechnung der Ohrdaten wurden mit Hilfe von geeigneter Literatur selber hergeleitet und implementiert. Es wurde bereits ein Archiv mit über 200 Ohraufnahmen von über 40 verschiedenen Personen angelegt, die alle mit einer Sicherheit von über 90% identifiziert werden können.

2. Aufgabenstellung (Orginal)

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

Personenidentifikation anhand biometrischer Merkmale

1. Einführung

Die Identifikation von Personen anhand von alten Fotos, Filmen oder Videoaufzeichnungen ist in vielen Fällen von grossem Interesse, insbesondere dann, wenn keine anderen Merkmale wie zum Beispiel Fingerabdrücke abgenommen oder überprüft 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. In Zusammenhang mit Unglücksfällen, Kriegen oder Verbrechen kann gerade die Analyse und Auswertung von entsprechendem Bildmaterial von grossem Wert sein. Erschwerend ist sicher der Umstand, dass die Aufnahmebedingungen, d.h. Abstand, Blickwinkel und Beleuchtungsverhältnisse, im Allgemeinen nicht kontrollierbar sind und daher sehr unterschiedlich sein können. Für bestimmte Anwendungen wie etwa der Überprüfung der Identität einer Person bei einer Zutrittskontrolle wäre eine Kontrolle des Umfeldes und damit eine Einschränkung der Variabilität der Bildaufnahmen jedoch denkbar. Praktische Erfahrungen dazu fehlen jedoch bis heute.

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

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 Bildaufnahmen der Ohren von Personen durchgeführt werden. Die Aufgabe ist entsprechend den Komplexitätsgraden gestuft anzugehen.

In einer ersten Teilaufgabe sollen Beschreibungsmöglichkeiten der menschlichen Ohren ausgearbeitet und ein Archiv mit Ohrenbildern angelegt werden. Die Sammlung soll für die vorkommenden Ohrenformen möglichst repräsentativ sein. Die aufgenommenen Bilddaten sollen analysiert und bezüglich charakteristischen Eigenschaften der Ohren wie zum Beispiel Form, Grösse oder anderen geeigneten Merkmalen beschrieben werden.

In einer zweiten Teilaufgabe und der eigentlichen Hauptaufgabe sollen Bildverarbeitungsalgorithmen entwickelt und programmiert werden, mit deren Hilfe die Ohrformen gemessen werden können und die eine Reihe von Merkmalen liefern, die in einem späteren Erkennungsverfahren eingesetzt werden können [2], [3]. Die Messverfahren sollen auch daraufhin untersucht werden, wie robust sie unter verschiedenen Aufnahmebedingungen sind und wieweit sie sich für eine automatische Messung eignen. Die Messwerte sollen in einer Datenbank abgelegt werden und bilden die Basis für eine spätere automatische Personenidentifikation.

Auf diesen Grundlagen soll dann ein Klassifikationsverfahren entwickelt werden, welches die Verifikation im Sinne einer Überprüfung der Identität einer bestimmten bekannten Person erlaubt. Es ist zu prüfen, in welchem Rahmen dabei die Aufnahmebedingungen wie Kameradistanz und Beleuchtung für eine sichere Erkennung variieren dürfen.

In einer weiteren Stufe kann nun das Klassifikationsverfahren in dem Sinne weiterentwickelt werden, dass den von der Kamera (unter praxisbezogenen definierten Aufnahmebedingungen) erfassten Ohren automatisch deren Besitzer zugeordnet werden kann. Das Ziel besteht in der online Erkennung von möglichst vielen Personen auf Grund ihrer Ohren bei gleichzeitig möglichst geringen Anforderungen an die Aufnahmebedingungen.

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, EasyObject und EasyMeasure der Bildverarbeitungssoftware Euresys e-Vision zur Verfügung.

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

Der Bericht soll enthalten: Aufgabenstellung, Zusammenfassung auf einer A4-Seite, Inhaltsverzeichnis, Einleitung, Zeitplan und Arbeitsteilung, Beschreibung der Lösung und allfälliger Lösungsvarianten, der Ergebnisse sowie eine Bedienungsanleitung zu den erstellten Programmen. Die Programm listings sind dem Bericht im Anhang beizufügen. Zusätzlich soll dem Bericht auf der hintersten Seite ein CD-ROM beigelegt werden, das den Text des Berichtes sowie Quellprogramme enthält. Datenträger sind mit den Namen der Autoren, Titel der Arbeit und Datum zu beschriften.

Der Bericht ist in 2 Exemplaren abzugeben.

Zusätzlich ist eine Zusammenfassung der Diplomarbeit an den Abteilungsvorstand gemäss spez. Weisungen abzugeben.

5. Termine

Abgabe der Aufgabenstellung an die Studenten: 23. Oktober 2000.

Abgabe des Berichts der Diplomarbeit und der Datenträger am 14. Dezember 2000, 1600.

Nach Abgabe des Berichtes findet eine Präsentation der Arbeit in Verbindung mit der mündlichen Prüfung statt gemäss Reglement über Prüfungen und Promotionen.

6. Organisation

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

Besprechungen gemäss Absprache.

7. Experte

Herr Dr. Peter Wägli.

8. Bewertung

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

Rapperswil, 22. Oktober 2000

Dr. H. Grossmann

3.	Inhaltsverzeichnis	
1.	ZUSAMMENFASSUNG (ABSTRACT)	1
2.	AUFGABENSTELLUNG (ORIGINAL)	2
3.	INHALTSVERZEICHNIS	5
4.	EINLEITUNG	9
5.	PROJEKTPLANUNG & ARBEITSAUFTEILUNG	10
5.1.	Meilensteine	10
5.2.	Zeitplan	11
5.2.1.	Soll- Zeitplan	11
5.2.2.	Ist- Zeitplan	11
5.3.	Arbeitsaufteilung	12
6.	PHYSIKALISCHE SYSTEMANALYSE	13
6.1.	Aufnahmetechnik	13
6.1.1.	Varinate 1, Rohr:	13
6.1.2.	Variante 2, Anschläge für Kinn und Stirn:	13
6.1.3.	Variante 3, bedingt freistehend, direkt beleuchtet:	13
6.1.4.	Variante 4, bedingt freistehend, indirekt beleuchtet, in einer Kabine/Gang:	13
6.1.5.	Wahl der Aufnahmetechnik	14
6.2.	Die Aufnahmekabine	15
6.2.1.	Kamera	16
6.2.2.	Beleuchtung	16
6.3.	Weitere Hilfsmittel	16
6.4.	Gesamtsystem	17
6.5.	Vorstudie	18
6.5.1.	Einführung	18
6.5.2.	Verschiedene bekannte Merkmale	19
6.5.1.1.	Ohrlage von vorne betrachtet	19
6.5.2.2.	Ohrleiste	19
6.5.2.3.	Ohransatz	19
6.5.2.4.	Darwinsche Erweiterung/Knoten	19
6.5.2.5.	Form des Ohrläppchens	20
6.5.2.6.	Ohrform	20
6.5.3.	Auswahl von geeigneten Merkmalen	21
6.5.3.1.	Kombinationen	21
6.5.4.	Neue Merkmale	22
6.5.4.1.	Winkel zwischen den Punkten Nase-Ohr-Auge	22
6.5.4.2.	Ohrgrösse	22
6.5.5.	Verwendete Merkmale	22
7.	BILDVERARBEITUNG	23
7.1.	Einleitung	23
7.2.	Inhalt	23
7.3.	Konturcode (Freeman-Code)	24
7.3.1.	Beispiel zur Konturcodierung	24
7.4.	Nachbarschaft (4-, 8-)	25
7.5.	CloseContour	26
7.5.1.	Beschreibung der Funktion	26
7.5.2.	Beschreibung des Algorithmus	26
7.6.	Skelettierung	27
7.6.1.	Beschreibung der Funktion	27
7.6.1.1.	Anforderungen an den Skelettierungsalgorithmus	27
7.7.	Beschreibung des Algorithmus	27
7.8.	Kantenextraktion	29
7.9.	Drehsinn	31
7.10.	Bestimmung von Richtung & Betrag eines Konturstückes	32
7.10.1.	Beschreibung der Funktion	32
7.10.2.	Beschreibung des Algorithmus	32

7.10.2.1.	Beispiel	32	
7.11.	Schematischer Ablauf	33	
7.12.	Eingrenzung des Bereiches zur Extraktion des Ohres	34	
7.13.	Vom Grauwertbild zur Ohrkontur	36	
7.13.1.	Einführung	36	
7.13.2.	Ablauf	37	
7.13.3.	Beschreibung	37	
7.14.	Ermittlung der Ohrmaske	38	
7.14.1.	Beschreibung der Funktion	38	
7.14.2.	Beschreibung des Algorithmus	38	
7.14.2.1.	Ermittlung des Endpunktes	38	
7.14.2.2.	Ermittlung des Startpunktes	39	
7.14.2.3.	Erstellung der Maske	39	
8.	MERKMALE DES OHRES	40	
8.1.	Das Ohrläppchen	40	
8.1.1.	Die 4 verschiedenen Formen	40	
8.1.2.	Ermittlung der Form des Ohrläppchen	40	
8.1.2.1.	Problem	40	
8.1.2.2.	Vorgehen	41	
8.1.3.	Effektive Merkmale	41	
8.2.	Die Ohrform	42	
8.2.1.	Die 4 verschiedenen Formen	42	
8.2.2.	Formbeschreibende Grössen	42	
8.2.2.1.	Verhältnis der Breite zur Höhe	43	
8.2.2.2.	Verhältnis der Krümmung am oberen und rechten Rand des Ohres	43	
8.2.2.3.	Verhältnis der Breite oben und unten am Ohr	44	
8.2.2.4.	Steigungswinkel des Ohres	44	
8.3.	Die Ohrleiste (Helix)	45	
8.3.1.	Prinzip der Ohrleistenmessung	46	
8.3.1.1.	Bemerkung	46	
8.4.	Ohransatz	47	
8.4.1.	Prinzip der Messung des Ohransatzes	47	
8.5.	Winkel Eye-Center of Gravity-Nose Tip	48	
8.6.	Earyfier	49	
8.6.1.	Benutzte Libraryfunktionen aus eVision	49	
8.6.1.1.	ImgOper()	49	
8.6.1.2.	ImgContour()	50	
8.6.1.3.	ImgEqualize()	50	
8.6.1.4.	ImgThreshold()	51	
8.6.1.5.	ImgMedian()	51	
8.6.1.6.	ImgConvolRoberts()	51	
8.6.1.7.	BuildObjects ()	51	
8.6.1.8.	MsrFindPointAlongLine()	52	
8.6.1.9.	Match ()	52	
9.	KLASSIFIKATION	53	
9.1.1.	Bestimmung der Varianz	54	
9.1.2.	Raumnormierung	55	
10.	GROBPFLICHTENHEFT	56	
10.1.	Einführung	56	
10.1.1.	Zweck des Dokuments	56	
10.2.	Funktionale Anforderungen	56	
10.2.1.	Das Programm Earyfier	56	
10.2.2.	Datenhaltung	56	
10.2.3.	Zusammenhänge (UML- Dagramm)	57	
10.3.	Funktionale Anforderungen im Einzelnen	58	
10.3.1.	Use Cases	58	
10.3.1.1.	Übersicht	58	

10.3.1.2.	Use Case Beschreibungen	58
10.4.	Nichtfunktionale Anforderungen	60
10.4.1.	Einführung	60
10.4.2.	Benutzerschnittstelle	60
10.4.3.	Leistungsanforderungen	60
10.4.4.	Weitere Anforderungen bezüglich Vorgehen und Hilfsmittel	60
11.	ANFORDERUNGSSPEZIFIKATION	61
11.1.	Klassendiagramm (UML-Notation)	61
11.2.	Klassenbeschreibung	62
11.2.1.	Klasse Ear	62
11.2.2.	Klasse Identifikation	64
11.2.3.	Klasse grabber	64
11.2.4.	Klasse Earlib	64
11.2.5.	Klasse Olist	65
11.2.6.	Struct Elem	65
11.2.7.	Klasse konstanten	65
11.2.8.	Klasse Pixel	66
11.2.9.	Klasse Line	66
12.	BEDIENUNGSANLEITUNG	67
12.1.	Installation	67
12.2.	Der Earyfier	67
12.2.1.	Die Menüleiste	68
12.2.1.1.	Filemenü	68
12.2.1.2.	Archivemenü	68
12.2.1.3.	Cameramenu	69
12.2.1.4.	Statusmenü	70
12.2.2.	Grafikbereich	71
12.2.3.	Identifikationskontrollen	72
13.	TESTS UND RESULTATE	73
13.1.	Einführung	73
13.2.	Testarchiv	73
13.3.	Testauswertung	73
13.3.1.	Für das Verständnis der Testtabellen	73
13.3.2.	Normalfall	74
13.3.3.	Fehlerfälle	75
13.3.3.1.	Zwischen zwei Referenzohren	75
13.3.3.2.	Distanz zu jedem Ohr zu gross	76
13.3.3.3.	Verwechslung	77
13.4.	Auswertung (Übersicht)	78
14.	EPILOG	79
14.1.	Verbesserungsmöglichkeiten und Ideen	79
14.2.	Schlusswort	79
15.	LITERATURVERZEICHNIS	80
16.	GLOSSAR	81

4. Einleitung

Wenn Körperteile erwähnt werden, die man zur Erkennung von Einzelpersonen verwendet, hört man sehr wenig über das Ohr. Fingerabdrücke, die Geometrie der Hand, Gesichtserkennung, Retina Scan und andere biometrischen Merkmale werden heutzutage verwendet, um die Sicherheit und Leistungsfähigkeit der Personenidentifikation zu erhöhen. Dabei ist es belegt, dass auch das Ohr ein Objekt ist, das von Mensch zu Mensch verschieden ist. Ja mehr noch, das Ohr ist eines von wenigen Körperteilen, welches seine Grundform ein Leben lang behält. Es soll bereits 56 Tage nach der Geburt seine Grundform besitzen, und diese bis zum Tode beibehalten. Bereits im zweiten Weltkrieg, wurden gefallene Marinesoldaten anhand ihrer Ohren identifiziert. Auch in der Personenfahndung der Polizei ist das Ohr kein unbeschriebenes Blatt, und wird zur Erstellung von Phantombildern und Personenbeschreibung anhand verschiedener Merkmale klassifiziert.

In der vorliegenden Diplomarbeit wird in einer ersten Phase eine Untersuchung durchgeführt, die das Ziel haben soll, Merkmale zu finden um das Ohr zu beschreiben. Diese Merkmale sollen auch für eine spätere Identifikation in Frage kommen können. Anschliessend müssen Algorithmen hergeleitet werden, die es ermöglichen, aus der Originalaufnahme des Kopfes, das Ohr zu extrahieren und danach vollständig auszumessen. Weiter muss ein geeignetes Klassifikationsverfahren ermittelt werden, das ermöglicht, einen Vergleich der gemessenen Daten durchzuführen.

In einer zweiten Phase geht es darum, alle diese Algorithmen zu implementieren und in ein benutzerfreundliches Softwareprogramm zu packen. Das Programm soll nicht nur erlauben, Bilder von Ohren aufzunehmen und diese anschliessend auszumessen, sondern es soll vielmehr dazu dienen, mit den aufgenommenen Ohren eine Verifikation bzw. Identifikation durchführen zu können. Dies geschieht mittels Vergleich von Ohren in einem Archiv, das selbstverständlich selber angelegt und erweitert werden kann.

Ziel dieser Diplomarbeit ist es in erster Linie eine Verifikation bzw. Identifikation erfolgreich durchführen zu können. Weiter soll eine Aussage darüber gemacht werden, wie gut die Merkmale gewählt wurden, d.h. mit welcher Sicherheit die Erkennung schlussendlich ablaufen kann.

5. Projektplanung & Arbeitsaufteilung

5.1. Meilensteine

- **Start Diplomarbeit** → **Mo. 23.**
Okt. **Woche 43**
- **Erste Aufnahmen des Ohres** → **Mi. 25.**
Okt. **Woche 43**
- **Vorstudie zur Ermittlung der Ohrenmerkmale abgeschlossen** → **Mo. 30.**
Okt. **Woche 44**
- **Definitiver Aufbau der Aufnahmekabine** → **Fr. 3. Nov.**
Woche 44
- **Algorithmus zur Ermittlung des Start- und Endpunktes** → **So. 12. Nov.**
Woche 45
- **Klassifikationsverfahren zur Identifikation abgeschlossen** → **Di. 21.**
Nov. **Woche 47**
- **Implementation des Onlineverfahrens** → **Fr. 24. Nov.**
Woche 47
- **Ausmessung des ersten Ohres per Software** → **Mo. 27.**
Nov. **Woche 48**
- **Erste erfolgreiche Identifikation durchgeführt** → **Fr. 1. Dez.**
Woche 48
- **Erstellung eines Archivs mit über 40 Personen** → **Mo. 4. Dez.**
Woche 49
- **Optimierung des Algorithmus „Grauwertbild zur Ohrkontur“** → **Di. 5. Dez.**
Woche 49
- **Abschluss der Testreihe** → **Do. 7. Dez.**
Woche 49
- **Erste Version des Berichts fertig** → **Mo. 11. Dez.**
Woche 49
- **Abgabe** → **Do. 14. Dez.**
Woche 50

5.2. Zeitplan

5.2.1. Soll- Zeitplan

	W 43	W 44	W 45	W 46	W 47	W 48	W 49	W 50
Vorbereitungsarbeiten								
Bestimmung der Aufnahmeart								
Aufbau der Aufnahmevorrichtung								
Erstellen eines Archivs								
Problemanalyse								
Bestimmung von Merkmalen								
Verfahren zur Merkmalsermittlung								
Extraktion des Ohres								
Verfahren zur Identifikation								
OOA/OOD								
GUI								
Datenhaltung								
Problem Domain								
Implementation								
GUI								
Datenhaltung								
Problem Domain								
Testphase								
Bericht								

5.2.2. Ist- Zeitplan

	W 43	W 44	W 45	W 46	W 47	W 48	W 49	W 50
Vorbereitungsarbeiten								
Bestimmung der Aufnahmeart								
Aufbau der Aufnahmevorrichtung								
Erstellen eines Archivs								
Problemanalyse								
Bestimmung von Merkmalen								
Verfahren zur Merkmalsermittlung								
Extraktion des Ohres								
Verfahren zur Identifikation								
OOA/OOD								
GUI								
Datenhaltung								
Problem Domain								
Implementation								
GUI								
Datenhaltung								
Problem Domain								
Testphase								
Bericht								

5.3. *Arbeitsaufteilung*

	W 43	W 44	W 45	W 46	W 47	W 48	W 49	W 50
Vorbereitungsarbeiten								
Bestimmung der Aufnahmeart								
Aufbau der Aufnahmevorrichtung								
Erstellen eines Archivs								
Problemanalyse								
Bestimmung von Merkmalen								
Verfahren zur Merkmalsermittlung								
Extraktion des Ohres								
Verfahren zur Identifikation								
OOA/OOD								
GUI								
Datenhaltung								
Problem Domain								
Implementation								
GUI								
Datenhaltung								
Problem Domain								
Bilverarbeitungsalgorithmen								
Berechnungsalgorithmen								
Identifikation								
Testphase								
Funktionalität der Software								
Resultate der Identifikation								
Bericht								
Grobpflichtenheft								
Anforderungsspezifikation								
Vorstudie								
Physikalische Systemanalyse								
Bildverarbeitung								
Bedienungsanleitung								
Tests/Resultate								
Allgemeines								

Legende:

- | | | |
|--|---|---|
| | → | Gesamtlänge Obergebiet |
| | → | Gesamtlänge Teilgebiet |
| | → | Stephan Huber & Roland Fuchs |
| | → | Roland Fuchs |
| | → | Stephan Huber |

6. Physikalische Systemanalyse

6.1. Aufnahmetechnik

Bevor mit dem Aufbau für die Aufnahmen begonnen werden konnte, wurden verschiedene Aufnahmevarianten geprüft. Es stehen vier Varianten zur Debatte.

6.1.1. Variante 1, Rohr:

Die Kamera wird in ein Rohr montiert, welches der Benutzer sich für die Aufnahme an das Ohr halten muss.

Vorteile:

- + Konstanter Abstand Ohr – Kamera
- + absolut kontrollierte Lichtverhältnisse

Nachteile:

- Umständlich für den Benutzer
- Problem der Beleuchtung im Rohr
- Evtl. hygienische Bedenken der Benutzer

6.1.2. Variante 2, Anschläge für Kinn und Stirn:

Die Kamera wird fix auf ein Gestell montiert. Vor diesem Gestell befindet sich eine ähnliche Vorrichtung, wie man sie von Sehtestgeräten beim Optiker kennt mit einem Anschlag für Kinn und Stirn. Die Kamera lichtet die Person von der Seite her ab.

Vorteile:

- + Der Kopf ist sehr gut fixiert. Es kann davon ausgegangen werden, dass die Ohrlage und der Abstand Ohr – Kamera nur gering variiert.
- + Mit bedingtem Aufwand lassen sich die Lichtverhältnisse kontrollieren. Zum Beispiel mit einem Vorhang um den Aufnahmebereich.

Nachteile:

- Besser als Variante 1 aber immer noch zu umständlich für den Benutzer
- Evtl. hygienische Bedenken der Benutzer

6.1.3. Variante 3, bedingt freistehend, direkt beleuchtet:

Die Person, die aufgenommen werden soll, steht auf zwei am Boden angezeichneten Füßen und schaut auf einen Fixpunkt an der Wand. Das sind die einzigen Massnahmen um die Person zu fixieren. Die Kamera muss in der Höhe verstellbar sein um kleine wie auch grosse Personen erfassen zu können. Das Kameraobjektiv ist von einer runden Leuchtstoffröhre umgeben.

Vorteile:

- + sehr benutzerfreundlich

Nachteile:

- Der Abstand Ohr – Kamera kann stark variieren.
- Der Kopf ist nur schwach fixiert, das heisst es können keine Winkel auf den Bildrand bezogen werden.
- Schatten und Tageslicht Einflüsse.

6.1.4. Variante 4, bedingt freistehend, indirekt beleuchtet, in einer Kabine/Gang:

Die Person, die aufgenommen werden soll, steht auf zwei am Boden angezeichneten Füßen und schaut auf einen Fixpunkt an der Wand. Das sind die einzigen Massnahmen um die Person zu fixieren. Die Kamera muss verstellbar sein um kleine wie auch grosse Personen erfassen zu können. Die Kabine wird indirekt ausgeleuchtet um starke Schatten auf und um die Ohren herum zu verhindern.

Vorteile:

- + sehr benutzerfreundlich
- + die Lichtverhältnisse lassen sich kontrollieren.

Nachteile:

- Der Abstand Ohr – Kamera kann stark variieren.

Diplomarbeit
Bildverarbeitung
Personenidentifikation mittels biometrischer Merkmale des Ohres

- Der Kopf ist nur schwach fixiert, das heisst es können keine Winkel auf den Bildrand bezogen werden.

6.1.5. Wahl der Aufnahmetechnik

Die Variante 3 schien geeignet zu sein und es wurden schnell schon viele Bilder aufgenommen. Schon bald stellte sich jedoch heraus, dass durch die nicht konstanten Lichtverhältnisse, alle Aufnahmen unbrauchbar waren.

Aus diesem Grund wurde die 3 Variante weiterentwickelt, da das Grundprinzip beibehalten werden sollte. Nur das Problem mit den nicht konstanten Lichtverhältnissen musste eliminiert werden. Dies wurde in der Variante 4 vollständig realisiert. Die Variante 4 scheint sich in vieler Hinsicht als Aufnahmeart zu eignen. Nachteile wie z.B. die Tatsache, dass der Kopf nicht fixiert ist oder dass die Distanz zwischen Objekt und Kamera nicht konstant ist, wurde als eine gewisse Herausforderung angesehen.

6.2. Die Aufnahmekabine

Damit alle Bilder eine ausreichende Qualität erreichen, wurde eine Aufnahmekabine gebaut. Damit diese in kürzester Zeit verfügbar war, wurde auf vorhandene Ressourcen zurückgegriffen. Die Kabine besteht aus fünf Präsentationswandelementen. Als Dach wurden leichte Kunststoffplatten verwendet. Über der Stelle, auf der die Testperson steht, ist ein Leuchtring zwischen zwei Aluminiumprofile montiert, welche auf den Präsentationswandelementen aufliegen. Damit auch hier kein Licht von Aussen einfallen kann, ist dieser Teil mit einem Leinentuch abgedeckt. Damit es in der Kabine genügend hell ist, aber keine störenden Schatten auf dem Ohr und um das Ohr herum auftreten, wird die Kabine zusätzlich mit zwei Scheinwerfern indirekt ausgeleuchtet. Die Kamera ist auf einem Stativ montiert, welches in einer Nische links neben der Testperson steht. Die Kamera kann manuell in der Höhe eingestellt werden, um kleine wie auch grosse Personen zu erfassen. In der Praxis kann dieser Prozess sicher automatisiert werden.

Fig. 6.1 Aufnahmekabine

Fig. 6.2 Kameranische

Fig. 6.3 Kamera mit Stativ

Fig. 6.4 Schematische Darstellung der Aufnahmekabine von oben

6.2.1. 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. Die Blende ist auf 5.6 eingestellt und die Softwareinstellungen bleiben für die Aufnahme auf den Defaultwerten (Gain = 1, Offset = 0).

6.2.2. Beleuchtung

Es wurden einige Experimente betreffend Beleuchtung durchgeführt. Nach einigen Versuchen zeigte sich, dass die Aufnahme in einer Kabine oder etwas ähnlichem stattfinden muss um die Lichteinflüsse von aussen einzuschränken. Das ist der Grund weshalb die Variante 3 zur Variante 4 weiterentwickelt wurde. Durch die Beleuchtung mit dem Leuchtring von oben und der indirekten Beleuchtung der beiden Scheinwerfer wurden die besten Bilder erzielt.

6.3. Weitere Hilfsmittel

Computer:

- Pentium Prozessor
- Taktfrequenz 90 MHz
- 32MB RAM
- Graphikkarte mit SVGA-Modus
- Mindestens 10 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'

6.4. Gesamtsystem

Über die CCD-Kamera und die Framegrabberkarte empfängt der Computer das Bild der in der Aufnahmekabine stehenden Person. Im PC kann nun mit der Bildverarbeitung begonnen werden.

6.5. Vorstudie

6.5.1. Einführung

Ziel der Vorstudie ist es, Merkmale des Ohres zu finden, die für eine spätere Identifikation in Frage kommen. Das heisst Merkmale, die über verschiedene Aufnahmen des selben Ohres konstant bleiben. Da sich in der Vergangenheit schon bereits verschiedene Studien mit diesem Problem auseinandergesetzt haben, ist es unnötig, alle diese Studien nochmals durchzuführen, und alle Merkmale neu zu finden. Aus diesem Grund wurde auf Resultate zurückgegriffen, die bereits im Einsatz sind. In der Personenfahndung der Polizei z.B. ist das Ohr schon lange kein unbeschriebenes Blatt mehr, denn dort wird für die Erstellung von Phantombildern, und zur Personenbeschreibung, das Ohr anhand verschiedener Merkmale unterschieden.

6.5.2. Verschiedene bekannte Merkmale

6.5.2.1. *Ohrlage von vorne betrachtet*

ganz abstehend anliegend	oben abstehend	unten abstehend
-----------------------------	----------------	-----------------

6.5.2.2. *Ohrleiste*

dünne Ohrleiste	dicke Ohrleiste	wulstige Ohrleiste
-----------------	-----------------	--------------------

6.5.2.3. *Ohransatz*

senkrecht	schräg nach hinten	schräg nach vorne
-----------	--------------------	-------------------

6.5.2.4. *Darwinsche Erweiterung/Knoten*

Darwinsche Erweiterung /Knoten

6.5.2.5. *Form des Ohrläppchens*

zwickelförmig rechtwinklig bogenförmig
freihängend

6.5.2.6. *Ohrform*

rund oval dreieckig viereckig

6.5.3. Auswahl von geeigneten Merkmalen

Die Klassifizierung der Ohren in die oben gezeigten Merkmale schien sich als geeignete Lösung anzubieten. Aber ob sich alle bekannten Merkmale eignen, musste zuerst noch untersucht werden. Es stellte sich bald heraus, welche dieser Merkmale für eine spätere Identifikation in Frage kommen würden, und welche nicht. Da die Aufnahme des Kopfes (Ohres) von der Seite her geschehen wird, kommt das Merkmal „Ohrlage von vorne betrachtet“ nicht in Frage, da diese Information durch die seitliche Aufnahme verloren geht. Die „Darwinsche Erweiterung/Knoten“ scheint sich auch nicht zu eignen, da es sehr schwierig werden wird, dieses Merkmal zuverlässig aus den Bilddaten herauszumessen. Aus diesem Grund wird auch dieses Merkmal nicht in die Identifikation aufgenommen. Die anderen 4 Merkmale (Ohrform, Ohrläppchen, Ohrleiste und Ohransatz) jedoch, scheinen sich sehr gut zu eignen, in die Identifikation aufgenommen zu werden. Und auch das Problem der Ermittlung durch Bildverarbeitung scheint bei diesen Merkmalen lösbar zu sein. Jedoch werden diese Merkmale alleine noch nicht genügen, um eine aussagekräftige Identifikation durchzuführen. Das soll hier an einer kleinen Rechnung sichtbar gemacht werden.

6.5.3.1. *Kombinationen*

Ohrleiste:	Unterteilung in <u>3</u> Stufen
Ohrform:	Unterteilung in <u>4</u> Stufen
Ohrläppchen:	Unterteilung in <u>4</u> Stufen
Ohransatz:	Unterteilung in <u>3</u> Stufen

Anzahl verschiedener Kombinationen:

Dies bedeutet, dass anhand dieser 4 Merkmale im Maximum 144 Personen voneinander unterschieden werden können. Es gibt nun verschiedene Möglichkeiten, diese Anzahl zu vergrössern. Eine Möglichkeit ist es, die Unterteilung der einzelnen Merkmale zu vergrössern, oder sogar die Unterteilung ganz zu entfernen, und das Merkmal in eine Zahl zu fassen. D.h. dass eine Ohrleiste z.B. 1.231 Dick, oder eine Ohrform 0.6 bogenförmig ist. Wichtig ist, dass diese neuen Werte dimensionslos sind, und nicht absolut aus dem Bild heraus gemessen werden dürfen, da sich diese bei Veränderung der Distanz Objektiv-Objekt ebenfalls stark ändern würden.

Eine andere Möglichkeit, die Anzahl zu vergrössern, ist es, weitere Merkmale einzuführen. Die Kriterien für diese Merkmale sind sehr einfach zu beschreiben. Grundsätzlich muss dieses Merkmal messbar sein, und darf sich nicht, oder nur schwach, von einer Aufnahme zur anderen der gleichen Person, ändern.

Im Falle des Earyfier wurden beide Möglichkeiten realisiert, das heisst es wurden neue Merkmale eingeführt, und die bekannten Merkmale in eine Zahl gefasst (Kap. 8.).

6.5.4. Neue Merkmale

6.5.4.1. *Winkel zwischen den Punkten Nase-Ohr-Auge*

Bei der Suche nach anderen Merkmalen tauchte die Idee auf, dass es von Vorteil sein könnte, wenn das Ohr auf irgendeine Art in Verbindung mit dem dazugehörigen Kopf gebracht werden könnte. Aus diesem Grund ist ein weiteres Merkmal entstanden, das für eine Identifikation eingesetzt werden kann. Es ist dies der Winkel (α), der entsteht, wenn eine Linie vom Nasenspitze, zum Ohr (Schwerpunkt) und dann zum Auge gezogen wird.

Fig. 6.5 Winkel zwischen Nase- Ohr und Auge

Dieser Winkel stellt einen gewissen Zusammenhang zwischen Gesicht und Ohr dar. Er ist ein Mass dafür, wie weit Auge und Nase auseinander liegen, aber auch wie weit hinten am Kopf, sich das Ohr befindet. Natürlich ist dieses Merkmal nicht alleine abhängig vom Ohr, aber dafür besteht jetzt eine Relation zwischen Kopf und Ohr. Dieser Wert bleibt auch konstant, wenn sich die Entfernung zwischen Objektiv und Objekt ändert, oder wenn sich der Kopf der Person nach hinten bzw. nach vorne neigt. Mit diesem neuen Merkmal sollte es möglich sein die Identifikation genauer durchführen zu können. Eine Aussage darüber zu machen, wie stark sich die Identifikation durch dieses neue Merkmal verbessern wird, ist an dieser Stelle nicht möglich. Das wird erst nach Beendigung von Testreihen der Fall sein. Jedoch scheint es logisch dass sich die Identifikation dadurch verbessern wird.

6.5.4.2. *Ohrgrösse*

Bei den Untersuchungen von Ohraufnahmen stellte sich heraus, dass die Ohrgrösse stark variiert und für ein weiteres Merkmal gut geeignet ist. Zuerst musste jedoch eine geeignete Referenz ermittelt werden, da die Grösse des Ohres nicht absolut gemessen werden darf, aus Gründen, die bereits schon erwähnt wurden. Da der Nasenspitze für die Berechnung eines anderen Merkmals (Kap. 6.5.4.1.) ermittelt wird, schien es naheliegend, die Distanz Ohr- Nase zu verwenden, als Referenz für die Ohrgrösse. Das heisst nun, dass die Ohrgrösse absolut ermittelt wird, und dann durch das Quadrat der Distanz Nase- Ohr geteilt wird.

6.5.5. Verwendete Merkmale

Wie schlussendlich die einzelnen Merkmale ausgemessen und klassifiziert werden, oder wie eine allfällige Unterteilung aussieht, wird sich erst im Laufe der Arbeit zeigen, wenn die Algorithmen zur Merkmalsextrahierung implementiert werden aber auch wenn es um die Identifikationsalgorithmen geht. Klar steht aber bereits, welche Merkmale eingesetzt werden.

- Die Form des Ohres
- Die Form des Ohrfläppchens
- Die Dicke der Ohrleiste
- Die Lage des Ohrensatzes
- Der Winkel zwischen Nase- Ohr und Auge
- Die Grösse des Ohres

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.

7.2. Inhalt

In den folgenden Kapiteln werden Bildverarbeitungsalgorithmen beschrieben, die vom Earyfier eingesetzt werden. Die Algorithmen werden benötigt, um von der Originalaufnahme des Kopfes schliesslich zu den Ohrdaten bzw. Merkmalen zu gelangen. Die beschriebenen Algorithmen wurden alle bis auf den Skelettieralgorithmus (Kap. 7.6.) anhand geeigneter Literatur (Kap. 15.) selber hergeleitet. Weiter finden sich einige Kapitel, die Grundlegendes der Bildverarbeitung aufzeigen, die aber für das Verständnis der Algorithmen unabdingbar sind.

7.3. 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. Gegenuhrzeigersin angegeben ist, damit man die Kontur rekonstruieren kann.

Fig. 7.1 Konturcode

7.3.1. Beispiel zur Konturcodierung

Fig. 7.2 Konturcode eines Objektes

7.4. Nachbarschaft (4-, 8-)

Jeder Bildpunkt besitzt zwei Arten von Nachbarn:

- a.) **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.
- b.) **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. 7.4 8er Nachbarschaft

7.5. *CloseContour*

7.5.1. Beschreibung der Funktion

Die Funktion *CloseContour* versucht unterbrochene Linienelemente zu finden, und diese zu verbinden. Dabei ist voranzusetzen, dass die Linien eine Strichbreite von einem Pixel aufweisen. Falls dies nicht der Fall ist, muss zuerst eine Skelettierung des Bildes durchgeführt werden (Kap. 7.6.). Der maximale Abstand bei dem noch Lücken geschlossen werden, wird als Parameter übergeben.

7.5.2. Beschreibung des Algorithmus

In einem ersten Schritt wird das Bild systematisch nach Linienenden abgesucht (Fig. 7.5). Ein Linienende ist dadurch definiert, dass dieser Endpunkt nur einen Nachbarn aufweist (Kap. 7.4.). Dies ist aber nur bei einer Strichbreite von 1 Pixel der Fall, und muss deshalb vorausgesetzt werden.

Fig. 7.5 Endpunkte rot markiert

Alle Endpunkte von Linien werden in einer ersten Phase gekennzeichnet. In einer zweiten Phase wird verglichen, ob sich zwei oder mehr Endpunkte in einem gewissen definierten Abstand zueinander befinden. Sind dies 2, werden sie direkt verbunden (Fig. 7.6). Sind es aber mehr als zwei, werden alle Punkte miteinander nach einer nicht definierten Reihenfolge miteinander verbunden.

Schliessen mit Abstand 5

Fig. 7.6 Linienende miteinander verbinden

Schliessen mit Abstand 6

7.6. Skelettierung

7.6.1. Beschreibung der Funktion

Die Aufgabe eines Skelettierungsalgorithmus ist, diejenigen Objektpunkte zu tilgen, welche getilgt werden können, ohne dass das Objekt in mehrere nicht zusammenhängende Teile zerfällt (Fig. 7.7)

(die sogenannten „einfachen“ Bildpunkte), mit Ausnahme der Linienendpunkte.

Fig. 7.7 Beispiele zu Nachbarpunkte

7.6.1.1. Anforderungen an den Skelettierungsalgorithmus

- Das Skelett muss aus Linien der Breite eines Bildpunktes bestehen.
- Die topologischen Zusammenhänge des Skeletts müssen denjenigen des Originalbildes entsprechen.
- Die Skelettlinien müssen etwa in der Mitte der Objekte verlaufen.
- Bei dicken oder zerklüfteten Linien, dürfen möglichst wenig kurze Verästelungen des Skeletts entstehen, die meistens irrelevant sind (Rauschunempfindlichkeit).
- Das Ergebnis der Skelettierung darf sich nach einer bestimmten Anzahl Iterationen des Skelettierungsalgorithmus nicht mehr ändern (Stabilität).

7.7. Beschreibung des Algorithmus

Für die praktische Realisierung des Skelettierungsalgorithmus wurde eine spezielle Methode angewandt, bei der explizite Muster lokaler 3x3-Konfigurationen verwendet werden. Falls eines dieser Muster im entsprechenden Originalbild auftritt, ist der aktuelle Bildpunkt P zu tilgen. Ein geeigneter Musteransatz ist im [Zamp. 91] zu finden (Fig 7.8).

Nord	Süd	Ost	West
0 0 0	1 1 X	0 X 1	X X 0
X P X	X P X	0 P 1	1 P 0
X 1 1	0 0 0	0 X X	1 X 0
X 0 0	X 1 X	0 0 X	X 1 X
1 P 0	0 P 1	0 P 1	1 P 0
X 1 X	0 0 X	X 1 X	X 0 0
X = 0 oder 1			

Fig. 7.8 3x3-Konfigurationen zum tilgen von Bildpunkten

Wie bereits erwähnt, ist eine Anforderung an den Algorithmus, dass das Skelett aus Linien der Breite eines Bildpunktes bestehen soll. Aus diesem Grund muss der Algorithmus iterativ, d. h. immer wieder durchgeführt werden, bis kein Bildpunkt mehr getilgt werden kann, bzw. die Strichbreite „1“ erreicht ist. Besonders bei grossen Objekten führt dies zu einer hohen Anzahl von Durchläufen, und einem damit verbundenen grossen Zeitaufwand. Im Falle des Earyfiers, kann davon ausgegangen werden, dass das zu bearbeitende Bild nicht aus grossen Objekten, sondern nur aus Linien besteht (Fig. 7.9). Dies hat zur Folge, dass der Algorithmus im Schnitt 2-3 mal ausgeführt werden muss.

Vor der Skelettierung

Nach einem Durchlauf

Fig. 7.9 Resultat des Skelettierungsalgorithmus

7.8. *Kantenextraktion*

Physiologische Untersuchungen haben gezeigt, dass das menschliche Auge besser auf Diskontinuität als auf Kontinuität geeicht ist. Für die Verarbeitung digitalisierter Bilder heisst das, dass der Versuch, die Segmentierung kanten- und linienorientiert anzugehen, durchaus erfolgversprechend sein kann.

Im Rahmen dieser Arbeit schien es eine gute Möglichkeit zu sein mit Kantenextraktion die Kontur welche das Ohr einschliesst zu finden um daraus eine Maske für das Ohr zu erstellen.

Das Idealergebnis einer Kantenextraktion sollte in den meisten Anwendungen die folgenden Eigenschaften besitzen:

- Binärbild, z.B. weisse Linien und schwarzer Hintergrund
- Linien mit konstanter Breite von 1 oder 2 Bildpunkten
- Lückenlose, möglichst geschlossene Konturlinien
- Keine Störungen, wie kurze Linienäste, Punkte oder Flecken.

In der Praxis ist das Erreichen dieser Ziele mit einer Extraktion oft sehr problematisch und erst durch den Einsatz von verschiedenen Bildverarbeitungsoperatoren zu erreichen. Es ist jedoch nicht ohne qualitative Kompromisse möglich. Eine entscheidende Rolle spielt hierbei der Vorgang Binarisierung mit Hilfe einer Konstanten oder einer lokaladaptiven Schwelle. Der Literatur ist auch nicht eine direkte Lösung dieses Problems zu entnehmen. Piero Zamperoni schreibt in seinem Buch [Zamp. 91] zu diesem Thema, dass in der Praxis für jeden Anwendungsfall speziell eine günstige Kette von Kantenextraktions-, Glättungs-, Binarisierungs- und Bildverbesserungsoperatoren in erster Linie nach empirischen Kriterien zusammengestellt wird.

Es gibt verschiedene Operatoren zur Kantenextraktion. Alle diese Operatoren ausser dem Laplace- und dem Robertsoperator arbeiten mit einer richtungsabhängigen 3x3 Matrix, die in der Literatur auch Maske genannt wird. Der Robertsoperator arbeitet nur mit einer 2x2 Matrix. Er berechnet die Summe der absoluten Differenz der Bildpunktweite in der entsprechenden Diagonalen. Der Laplaceoperator ist der einzige nicht direkt richtungsabhängige Operator aber es wird zwischen vierfach und achtfach Nachbarschaft unterschieden.

Operator	X-Richtung	Y-Richtung
Gradient	$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} 0 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$
Prewitt	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$
Sobel	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$
Robinson	$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$
Kirsch	$\begin{bmatrix} -5 & 3 & 3 \\ -5 & 0 & 3 \\ -5 & 3 & 3 \end{bmatrix}$	$\begin{bmatrix} -5 & -5 & -5 \\ 3 & 0 & 3 \\ 3 & 3 & 3 \end{bmatrix}$

Diplomarbeit
Bildverarbeitung
Personenidentifikation mittels biometrischer Merkmale des Ohres

Operator	Vierfach Nachbarschaft	Achtfach Nachbarschaft
Laplace	$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$	$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix}$

Operator	Von links oben nach rechts unten	Von rechts oben nach links unten
Roberts	$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$

Ein Nachteil des Roberts-Operators ist, dass er besonders empfindlich auf Rauschstörungen reagiert. Der Sobel-Operator hingegen ist bezüglich Rauschstörungen robuster, dies wird jedoch mit etwas verbreiterten Konturen im ungestörten Fall erkauft.

Häufig wird eine Kante gesucht, die X und Y Komponenten beinhaltet. Um in diesem Fall die gesamte Kante zu finden wird der Operator in beide Richtungen angewandt und die Resultate summiert.

7.9. Drehsinn

Der Drehsinn des Konturcodes ist wichtig für die Ermittlung des Start- und Endpunktes der Ohrkontur. Für die Bestimmung des Drehsinns jedes Konturpunktes, werden zwei weitere Punkte benötigt. Der erste Punkt (BACK) liegt eine bestimmte Anzahl von Konturschritten hinter dem aktuellen Punkt (ACTUAL) im Konturvektor und der zweite (FRONT) ist gleich viele Schritte vor dem aktuellen Punkt im Konturvektor. Der Punkt mit dem höheren Index wird FRONT genannt und der Punkt mit dem tieferen Index BACK (Fig. 7.10). Es wird nun der Winkel (hellgrün in Fig. 7.10) ermittelt, welcher zwischen der geraden Fortsetzung der Linie BACK-ACTUAL und der Linie ACTUAL-FRONT entsteht. Gemessen wird immer von der Verlängerung BACK-ACTUAL aus. Der resultierende Winkel liegt im Bereich -180° bis 180° . Ist der Winkel positiv, biegt die Kontur nach links ab (Blauer Fall in Fig. 7.10), biegt sie jedoch nach rechts ab wird der Winkel negativ (roter Fall in Fig. 7.10). Dem Vorzeichen kann also der Drehsinn entnommen werden.

Fig. 7.10 Drehsinn

Der Winkel gibt die Stärke der Biegung an. Ist eine Gerade vorhanden bleibt das Vorzeichen wie bei der letzten Krümmung deren Abweichung nicht 0° war. Es gibt also den Wert 0 wie auch den Wert -0 . Das ist wichtig, damit wenn vor einem geraden Stück der Winkel negativ ist, das Vorzeichen auch während dem geraden Stück negativ bleibt (Fig. 7.11). Für die Ermittlung des Start- und Endpunktes der Ohrkontur wird der Abschnitt der Kontur gesucht, in welchem das Vorzeichen am wenigsten wechselt. Deshalb darf das Vorzeichen nicht wechseln solange keine Biegung in die andere Richtung vorkommt.

Fig. 7.11 Der Drehsinn (Vorzeichen) ist auf Geraden gleich wie der Drehsinn der letzten Biegung. Für beide Konturausschnitte ist der Drehsinn konstant.

7.10. Bestimmung von Richtung & Betrag eines Konturstückes

7.10.1. Beschreibung der Funktion

Anhand dieses Algorithmus ist es möglich die Richtung und den Betrag eines Konturstückes zu berechnen. Der Betrag ist ein Mass dafür, wie stark das Konturstück in die ermittelte Richtung verläuft.

7.10.2. Beschreibung des Algorithmus

Es werden die auftretenden Richtungen im Konturcode aufaddiert und dies ergibt dann bis zu 8 verschiedene Richtungsvektoren (v_0-v_7), mit einer Richtung und einem Betrag. Diese Vektoren werden mittels Vektorgeometrie [Pap. 94] addiert. Daraus resultiert ein Vektor (x) mit einer Richtung und einem Betrag. Der Betrag, wird noch mit der Länge des Konturstückes (m) dividiert, um ein relatives Mass (z) für den Betrag zu erhalten.

7.10.2.1. Beispiel

Das Resultat zeigt, dass der linke Vektor einen viel grösseren relativen Betrag aufweist, da das Konturstück keine Kurven durchläuft. Der rechte Vektor dagegen ist um einiges kleiner im relativen Betrag, da das Konturstück eine gewisse Krümmung aufweist.

7.11. Schematischer Ablauf

Der Ablauf zur Ohrermittlung ist in 4 Schritte unterteilt. In einem ersten Schritt wird der Bereich zur Extraktion des Ohres eingegrenzt, um die Suche nach der Ohrkontur zu vereinfachen. In einem zweiten Schritt, wird eine günstige Kette von Kantenextraktions-, Glättungs-, Binarisierungs- und Bilverbesserungsoperatoren angewandt, die nach empirischen Kriterien zusammengestellt wurde, um aus dem Originalbild ein Kantenbild zu erhalten. Das Resultat aus Schritt III soll eine Maske des vollkommen extrahierten Ohr liefern, um dann mittels dieser Maske in Schritt IV das Kantenbild des Ohres zu erstellen. Das Ohr kann jetzt vollständig ausgemessen werden.

Fig. 7.12

7.12. Eingrenzung des Bereiches zur Extraktion des Ohres

Um die Extraktion des Ohres zu vereinfachen, wird der Bereich eingegrenzt, in dem sich das Ohr befindet.

Es werden mit einem Matching-Tool (Kap. 8.6.1.9.) verschiedene Ohren, sogenannte Referenzohren mit dem Originalbild verglichen. Das Resultat des Matching-Tool liefert die Koordinate des Mittelpunktes des Bereiches, der am besten auf das Bild passt. Durch das Vergleichen einer Vielzahl von verschiedenen Ohrtypen erhält man eine grosse Anzahl von Koordinaten sogenannte „Treffer“. Je mehr Ohren auf diese Weise verglichen werden, umso genauer kann später der Bereich eingegrenzt werden. Ein übliches Trefferbild ist in (Fig. 7.13) dargestellt.

Fig. 7.13 Trefferbild nach dem „Matching“

Bei einer Mittelung über alle Treffer, würden diejenigen, die weit vom effektiven Mittelpunkt weg liegen die ganze Messung zu stark beeinflussen und eine genaue Ortung unmöglich machen. Ideal wäre es, wenn nur die Treffer in die Mittelung eingezogen würden, die nahe beieinander liegen, da dann davon ausgegangen werden kann, dass diese Treffer keine Fehltreffer sind. Aus diesem Grund wird eine Matrix über das Bild gelegt, deren Rastergrösse, einen konstanten Wert hat. Die Treffer werden nun auch in diese Matrix eingetragen. Wenn alle Treffer eingetragen sind, werden die Matricelemente auf ihre Trefferanzahl untersucht. Es werden dann nur diejenigen Werte in die Mittelung einbezogen, die in einem Matricelement auftauchen, welches mehr als einen Treffer enthält (rote Matricelemente in Fig. 7.14).

Fig. 7.14

Diplomarbeit
Bildverarbeitung
Personenidentifikation mittels biometrischer Merkmale des Ohres

Falls keine Matricelemente mit mehr als einem Treffer vorhanden sind, wird automatisch die Anzahl der Matricelemente verkleinert, d.h. das Raster der Matrix vergrössert, und die ganze Messung wiederholt. Beim so ermittelten Wert, kann davon ausgegangen werden, dass er den Mittelpunkt des Ohres repräsentiert. Nun muss nur noch der Bereich um diesen Punkt herum ausgeschnitten werden (Fig. 7.15). Die Grösse dieses Bereiches wird anhand von Messungen ermittelt und muss grösser sein als das grösste zu messende Ohr.

Fig. 7.15 Ausgeschnittenes Ohr

7.13. Vom Grauwertbild zur Ohrkontur

7.13.1. Einführung

Wie oft in der Bildverarbeitung gibt es keine systematische Methode, um „gute“ binäre Kantenbilder zu extrahieren. In der Praxis muss dann für jeden Anwendungsfall speziell eine günstige Kette von Kantenextraktions-, Glättungs-, Binarisierungs- und Bildverbesserungsoperatoren in erster Linie nach empirischen Kriterien zusammengestellt werden [Zamp. 91]. Auch beim Problem der Konturextraktion des Ohres aus dem Grauwertbild, wird so eine „günstige“ Kette zusammengestellt. In erster Linie werden verschiedene Ideen verfolgt, die aber auch mit empirischen Kriterien vermischt werden.

7.13.2. Ablauf

Schritt	Quellenbild 1	Quellenbild 2	Operation	Resultat
I	Grauwertbild (B1)		Invertieren	B2
II	B1		Gleichverteilung des Histogrammes	B3
III	B2		Gleichverteilung des Histogrammes	B4
IV	B4 (Dividend)	B3 (Divisor)	Skalierte Division	B5
V	B5		Schwellenoperator	B6
VI	B6		Medianfilter	B7
VII	B7		Roberts-Operator	B8

Schritt I:

B1
B4

Schritt II+III:

B2

B3

Schritt IV:
VII:B5
B8

Schritt V+VI:

B6

Schritt

B7

7.13.3. Beschreibung

Die Gleichverteilung des Histogrammes (Schritt II+III) bewirkt einerseits, dass bei einem Überschuss an dunklen Stellen, helle Gebiete noch mehr aufgehellt werden (B3). Das umgekehrte ist ebenfalls der Fall, dass bei einem Überschuss an hellen Stellen, die dunklen Gebiete noch dunkler werden (B4). Ein anderer Effekt ist die Kontrasterhöhung in dunklen Bereichen. Aus diesem Grund wurde, an Originalbild und invertierten Bild, diese Gleichverteilung durchgeführt. Als Resultat erhält man nun zwei Bilder, auf denen einmal das Ohr dunkel, und einmal das Ohr hell erscheint (B3/4). Wenn man nun diese beiden Bilder mit einer skalierten Division verrechnet, erhält man ein Bild, in dem das Ohr dunkel ist, und die Bereiche um das Ohr herum sehr hell (B5). Dies geschieht dadurch, dass die Stellen, die beim Dividend (B4) dunkel erscheinen und beim Divisor (B3) hell, im Zielbild (B5) noch dunkler erscheinen. Auf der anderen Seite werden Gebiete, die beim Dividend hell sind, und beim Divisor dunkel, noch mehr aufgehellt. Das so erhaltene Bild kann nun mit dem Schwellwertoperator [Klet. 95] in ein Zweipegelebild gewandelt werden (B6). Zur Elimination von isolierten fehlerhaften Bildpunkten wird noch mit dem Medianfilter [Hab. 91] gefiltert (B7). Zum Schluss wird eine Kantenextraktion mittels des Robertsoperators durchgeführt. Der Nachteil des Roberts-Operators [Zamp. 91], dass er rauschempfindlich ist, wurde durch die Anwendung des Medianfilters stark unterdrückt. Als Resultat erhält man das Kantenextrahierte Bild des Ohres (B8).

7.14. Ermittlung der Ohrmaske

7.14.1. Beschreibung der Funktion

Wie schon vorangehend erklärt, ermöglicht uns eine günstige Folge von Bildverarbeitungsfunktionen, dass aus dem Originalbild des Ohres ein Kantenbild entsteht, welches die Kontur des Ohres enthält. Dieser Algorithmus ist dafür verantwortlich, dass aus einer Kontur, der Start- und Endpunkt des Ohres gefunden wird (Fig. 7.16).

Fig. 7.16 Start- und Endpunkt des Ohres

7.14.2. Beschreibung des Algorithmus

7.14.2.1. Ermittlung des Endpunktes

Es gibt zwei verschiedene Kriterien, die den Endpunkt des Ohres definieren. Bei den einen Konturen ist der Endpunkt so definiert, dass die Kontur den Drehsinn ändert, von Gegenuhrzeiger- nach Uhrzeigersinn. Der andere Typ Endpunkt ist dort zu finden, wo die lokale Krümmung einen bestimmten Wert erreicht (Fig. 7.17). Aus diesem Grund wird über die ganze Kontur der örtliche Drehsinn berechnet (Kap. 7.9.). Diese Daten werden anschliessend gefiltert, damit lokale Unebenheiten in der Kontur geglättet werden können. Zum Zweiten wird ebenfalls auf der ganzen Kontur die örtliche Krümmung berechnet (Kap. 7.9.). Diese Daten werden auch gefiltert. Aus diesen beiden Informationen kann nun der Endpunkt ermittelt werden.

Fig. 7.17 Zwei Typen von Endpunkten

7.14.2.2. *Ermittlung des Startpunktes*

Es werden die gleichen Daten wie für die Endpunktberechnung zur Verfügung gestellt. D.h. Die örtliche Krümmung und der örtliche Drehsinn.
Der Endpunkt ist definiert, als Punkt, wo einerseits die Krümmung gross ist, und andererseits ein Drehsinnwechsel stattfindet. Da sich aber der Startpunkt im Gebiet des Ohrläppchens befindet und diese beiden Kriterien bereits oft schon vor dem Startpunkt eintreffen, muss noch ein weiteres Kriterium eingesetzt werden. Aus diesem Grund wird bei allen möglichen Startpunkten die durchschnittliche Richtung der Kontur in einem Bereich vor dem Punkt und einem Bereich nach dem Punkt berechnet (Kap. 7.10.). Falls diese Richtungen in einem gewissen Bereich liegen, das heisst wenn das Konturstück vorher und nachher in ungefähr die selbe Richtung durchläuft, kann an dieser Stelle kein Startpunkt vorhanden sein (Fig. 7.18).

Fig. 7.18 Startpunktbestimmung

Im Beispiel oben (Fig. 7.18) wird der Algorithmus den ersten Punkt (P1) übergehen, weil die Richtungen der Bereiche vor und nach dem Punkt im selben Bereich liegen. Dagegen wird der zweite Punkt (P2) als Startpunkt erkannt, weil die Kriterien Drehsinnwechsel und grosse örtliche Krümmung erfüllt sind und dazu die Richtung vor und nach dem Punkt stark verschieden sind.

7.14.2.3. *Erstellung der Maske*

Start und Endpunkt werden nun mit einer Linie verbunden und eine Maske aus der so entstandenen Ohrkontur erstellt (Fig. 7.19). Diese wird für die spätere Merkmalsbestimmung benötigt.

Fig. 7.19 Ohrmaske

8. Merkmale des Ohres

8.1. *Das Ohrläppchen*

8.1.1. Die 4 verschiedenen Formen

Die Form des Ohrläppchens wird in zwei Hauptgruppen unterteilt. Zur ersten Gruppe gehören die „angemachten“, und zur zweiten Gruppe die „nicht angemachten“ Ohrläppchen (Fig. 8.1). Diese zwei Gruppen werden je in zwei weitere Formen unterteilt. Daraus ergeben sich 4 verschiedene Formen. Es sind dies zwickelförmig, rechtwinklig, bogenförmig und freihängend (Fig. 8.1), wobei freihängend eine spezielle Form der Bogenform ist.
Fig. 8.1 die 4 verschiedene Ohrläppchentypen

Die Unterscheidung zwischen bogenförmig und freihängend ist so klein, dass sie nur schwer ermittelt werden kann. Aus diesem Grund wurde auf die Unterscheidung zwischen diesen beiden verzichtet. Daraus folgt, dass es für die Unterscheidung der Form des Ohrläppchen nur 3 Typen gibt (zwickelförmig, rechtwinklig und bogenförmig).

8.1.2. Ermittlung der Form des Ohrläppchen

8.1.2.1. *Problem*

Bei der Ermittlung der Form gibt es ein grundsätzliches Problem. Ein Problem, welches in der Praxis immer wieder auftaucht.

Wo liegen die Grenzen zwischen den einzelnen Formen, wo liegt der Unterschied zwischen z.B. einem rechtwinkligen und einem zwickelförmigen Ohrläppchen? Eine Frage, die wir nicht einmal selbst so genau beantworten können, wenn wir eine Vielzahl von Ohren betrachten (Fig. 8.2). Wie soll denn ein Computer erkennen können, welche Form die Ohrläppchen aufweisen.

Fig. 8.2 Verschiedene Ohrläppchen

Eine mögliche Lösung dieses Problems liegt in der Verwendung von Fuzzy-Logik. Dort gibt es als Unterschied, zur binären Logik, nicht nur falsch und richtig, bzw. rechtwinklig oder zwickelförmig, sondern auch Werte dazwischen, d.h. ein Ohr kann ein „bisschen“ zwickelförmig sein, ein „bisschen“ bogenförmig und auch noch ein „bisschen“ rechtwinklig. Im Earyfier wurde das Problem aber von einer anderen Seite angepackt. Ziel ist es, dass Ohrläppchen anhand 3 verschiedenen messbaren Größen zu klassieren.

8.1.2.2. *Vorgehen*

Als erstes muss die Kontur, die das Ohrläppchen beschreibt, in seinen Konturcode (Kap. 7.3.) zerlegt werden. Dieser Code enthält alle essentiellen Informationen zum Verlauf, und somit auch zur Form. Mittels eines Algorithmus zur Bestimmung der lokalen Krümmung [Zamp. 91] wurde ein Wert bestimmt, der ein Mass ist für die Krümmung des ganzen Ohrläppchens. Beim „angemachten“ Ohrläppchen geht dieser Wert gegen Null, da keine Krümmung vorhanden ist. Beim „nicht angemachten“ bzw. in unserem Fall bogenförmigen Ohrläppchen ist dieser Wert grösser Null. Da jedoch in der Realität die Ohrläppchen nicht immer „schön“ rechtwinklig oder bogenförmig sind, kann es jedoch sein, dass auch „angemachte“ Ohren eine gewisse Krümmung aufweisen.

In einem zweiten Schritt, wird mit einem speziell dafür entwickelten Algorithmus (Kap. 7.10.) die absolute Richtung und deren Betrag berechnet, die das Konturstück (Ohrläppchen) beschreibt. Die zwei so ermittelten Werte sagen nun etwas über die Richtung und den Betrag der Richtung des Konturverlaufes des Ohrläppchens aus. Falls ein zwickelförmiges Ohrläppchen vorliegt, ist die Richtung ungleich Null. Bei einem rechtwinkligen, geht dieser Wert gegen Null. Der Betrag dient dazu, eine Aussage darüber zu machen, wie stark die Kontur in dieser Richtung verläuft, denn auch ein bogenförmiges Ohr hat eine bestimmte Richtung (Kap. 7.10), aber da die Richtung innerhalb des Verlaufes sehr stark ändert, wird der Betrag beträchtlich kleiner als beim zwickelförmigen oder rechtwinkligen Ohrläppchen.

8.1.3. **Effektive Merkmale**

Das Ohrläppchen wird also nicht direkt mittels seiner Form unterschieden, sondern anhand von 3 messbaren Grössen, welche die Form beschreiben. Von diesen Grössen kann bei Bedarf auch auf die Form geschlossen werden.

Diese Grössen sind:

- Krümmung des Ohrläppchen
- Richtung des Ohrläppchen
- Betrag der Richtung

8.2. Die Ohrform

8.2.1. Die 4 verschiedenen Formen

Die Form des Ohres wird, wie schon in der Vorstudie aufgezeigt, in 4 Typen eingeteilt. Es sind dies die geometrischen Grundformen Dreieckig, Viereckig, Oval und Rund. Hier taucht das selbe Problem auf, das auch bei der Ermittlung der Form des Ohrläppchen im Vordergrund steht. Wo liegen die Grenzen zwischen den einzelnen Formen. Wer entscheidet, ob ein Ohr nun viereckig oder rund ist? Natürlich, es gibt Ohren, bei denen man sich nicht um die Form streiten muss. Doch im allgemeinen stösst man bei der Bestimmung immer wieder auf Widersprüche (Fig. 8.3). Um dieses Problem zu umgehen, wird der selbe Weg eingeschlagen, der auch beim Ohrläppchen zu einer Lösung führt. Man muss messbare Grössen finden, die abhängig sind von der Form des Ohres.

oval dreieckig? oval? viereckig?

Fig. 8.3 Widersprüche bei Ohrformen

8.2.2. Formbeschreibende Grössen

Wichtig ist es messbare Grössen zu bestimmen, die unabhängig von der Grösse des Ohres sind, da sie konstant bleiben müssen, auch wenn die Distanz zwischen Objekt und Kamera sich verändert. Aus diesem Grund sind die Merkmale entweder Winkel oder Verhältnisse. Als erstes wurden 8 Aussenpunkte (rot) des Ohres bestimmt, um von diesen Punkten aus, die Grössen abzuleiten (Fig. 8.4).

Fig. 8.4 Die 8 Messpunkte zur Formbestimmung

8.2.2.1. *Verhältnis der Breite zur Höhe*

Eine offensichtliche Grösse, die die Form des Ohres beschreibt, ist das Verhältnis der Breite des Ohres, zu seiner Höhe (Fig. 8.5). Dieser Wert ist bei runden Ohren am grössten.

Fig. 8.5 Höhe und Breite des Ohres

8.2.2.2. *Verhältnis der Krümmung am oberen und rechten Rand des Ohres*

Bei den verschiedenen Ohrformtypen ist zu erkennen, dass die Krümmung der Form stark ändert. Aus diesem Grund ist eine weitere Grösse die Krümmung am oberen (grün) und rechten (blau) Rand des Ohres (Fig. 8.6). Dieser Wert ist ein Mass dafür, wie sich die Aussenform vom oberen Teil zum rechten Teil des Ohres verändert. Dieser Wert ist am grössten beim dreieckigen und beim ovalen Ohr. Beim runden und viereckigen Ohr geht dieser Wert gegen null. Es wurde absichtlich nur der obere und rechte Teil des Ohres untersucht, da der linke Teil immer durch eine Gerade ersetzt wird (Kap. 7.14.2.3.) und im unteren Teil des Ohres sich das Ohrfläppchen befindet, welches ein eigenständiges Ohrmerkmal ist.

Fig. 8.6 Obere und rechte Krümmung der Ohrkontur

8.2.2.3. *Verhältnis der Breite oben und unten am Ohr*

Eine weitere messbare Grösse für die Unterscheidung der Form ist das Verhältnis der Breite oben und unten am Ohr (Fig. 8.7). Dieser Wert sagt etwas über die Konkavität des Ohres aus und ist bei dreieckigen Ohren am grössten.

Fig. 8.7 Obere und untere Breite des Ohres

8.2.2.4. *Steigungswinkel des Ohres*

Die letzte Grösse zur Unterscheidung der Form ist der Steigungswinkel (grün), der am unteren Rand des Ohres gemessen wird (Fig. 8.8). Er ist ein Mass für die Steigung der Ohrkontur im unteren rechten Bereich des Ohres. Dieser Wert ist besonders gross bei rechtwinkligen Ohren und besonders klein bei dreieckigen.

Fig. 8.8 Steigungswinkel des Ohres

8.3.1. Prinzip der Ohrleitenmessung

Die Messung der Ohrleiste wird graphisch ermittelt. Dazu wird das Linienbild des Ohres benötigt. Mit dem PointMeasure-Tool (Kap. 8.6.1.8.) wird dann die Dicke der Ohrleiste bestimmt. Um nicht einem Messfehler zum Opfer zu fallen, und allfällige Falschmessungen zu kompensieren, wird nicht nur an einer Stelle gemessen. Sondern es wird ein ganzer Bereich ausgewählt, vermessen und daraus dann der Mittelwert gebildet (Fig. 8.11).

Fig. 8.11 Ohrleitenmessung

8.3.1.1. *Bemerkung*

Der Wert b ist ein Mass für die Breite des oberen Teils der Ohrleiste. Dieser Wert ist abhängig von der Grösse des Originalbildes, d. h. falls die aufgenommene Person weiter weg von der Kamera steht, wird dieser Wert automatisch auch kleiner. Aus diesem Grund muss der Wert b noch mit einem anderen Wert des Ohres in Relation gebracht werden. Der so ermittelte Wert wird deshalb noch durch die Höhe (Länge) des Ohres dividiert. Der daraus entstandene Quotient ist dimensionslos und unabhängig von der Distanz zum Aufnahmeobjekt. Es darf aber nicht vergessen werden, dass der so berechnete Wert nun keine direkte Relation mehr hat zur effektiven Breite der Ohrleiste, denn bei einem „langen“ Ohr, wird dieser Wert automatisch kleiner.

8.4. Ohransatz

Fig. 8.12 senkrecht schräg nach hinten schräg nach vorne

Da der Kopf mit der gewählten Aufnahmetechnik für die Aufnahme nicht fixiert ist, kann dieser Winkel nicht absolut aus der Aufnahme gemessen werden.

Um dieses Merkmal zu messen muss deshalb ein geeigneter Bezug gefunden werden. Aus den verbalen Formulierungen der Literatur geht hervor, dass dieses Merkmal auf den ganzen Kopf bezogen ist. Deshalb wurde der Bezug für diese Messung in der Gesichtskontur gesucht.

8.4.1. Prinzip der Messung des Ohransatzes

Es werden zwei Punkte auf der Kontur des Gesichts bestimmt. Als erster der Nasenspitz (Nosetip) und als zweiter das Tal der Biegung zwischen Nase und Stirn. Dieser Punkt liegt meistens etwa auf Augenhöhe deshalb wird dieser Punkt Eye genannt.

Fig. 8.13 Ansatzwinkel

Legt man eine Gerade durch die Punkte Eye und Nosetip und schneidet diese mit der Geraden durch den Start- und den Endpunkt der Ohrkontur erhält man einen relativen Winkel für den Ohransatz, welcher unabhängig von der Lage des Kopfes ist.

8.5. Winkel Eye-Center of Gravity-Nose Tip

Ein weiteres Merkmal eines Ohres ist der Winkel (Angle), den die Geraden bilden die durch die Punkte Eye und Nosetip gehen und sich im Schwerpunkt (Center of Gravity) des Ohres schneiden (Fig. 8.14).

Fig. 8.14 Nase- Ohr- Auge-Winkel

Es werden die selben Bezugspunkte auf der Gesichtskontur benutzt wie für die Bestimmung des Ohransatzes (Eye, Nosetip → Kap. 8.4.1.). Der Schwerpunkt des Ohres wird auf Grund der ermittelten Ohrfläche bestimmt. Es wird der Schwerpunkt verwendet, weil dieser nicht so stark streut wie etwa der Start- oder Endpunkt.

8.6. *Earyfier*

8.6.1. Benutzte Libraryfunktionen aus eVision

Aus der Bildverarbeitungslibrary EasyLib wurden die folgenden Funktionen verwendet

- ◆ `ImgOper()`
- ◆ `ImgContour()`
- ◆ `ImgThreshold()`
- ◆ `ImgMedian()`
- ◆ `ImgConvolRoberts()`
- ◆ `BuildObjects()`
- ◆ `MsrFindPointAlongLine()`
- ◆ `Match()`

Es werden hier nur die, für das Verständnis wichtigen Funktionen, aufgeführt. Alle anderen Algorithmen der Studienarbeit wurden von Hand, d.h. ohne Hilfe von EasyLib Libraryfunktionen realisiert.

8.6.1.1. *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 Earyfier benutzt diese Funktion hauptsächlich, um Bilder zu kopieren.

Funktionsdeklaration:

```
void ImgOper ( enum ARITH_LOGIC_OPERATIONS IMG_COPY,  
EROIBW8* &m_EISourceImage, EROIBW8* &m_TargetImage);
```

Parameter:

<code>IMG_COPY</code>	Konstante als Angabe zum Kopieren
<code>&m_EIHandImage</code>	Zeiger auf das Quellenbild
<code>&m_EITempImage</code>	Zeiger auf das Zielbild

8.6.1.2. *ImgContour()*

Mit Hilfe der Funktion `ImgContour()`, kann die Kontur eines beliebigen Objektes identifiziert werden.

Es muss ein Randpunkt des Objektes 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 Konturpunkte über oder unter dem eingestellten Schwellwert befinden. Die Funktion folgt

nun der Kontur im Uhr- bzw. Gegenuhrzeigersinn. Beim Abfahren der Kontur kann zusätzlich zwischen den beiden Verfahren der 8er oder der 4er Nachbarschaft (Kap 7.4.) 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 das Bild
<code>n32Mode</code>	Uhr- oder Gegenuhrzeigersinn
<code>n32StartX</code>	X-Koordinate des Startpunktes
<code>n32StartY</code>	Y-Koordinate des Startpunktes
<code>n32ThrMode</code>	Bildpunkte 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 Konturbildpunkte auch noch einen dritten Wert. Dieser kann Optional den jeweiligen Grauwert des Bildpunktes, oder den Konturcode enthalten.

8.6.1.3. *ImgEqualize()*

Bewirkt eine Gleichverteilung der Grauwerte im Histogramm eines Bildes.

Funktionsdeklaration:

```
void mgEqualize(EImageBW8* pSrc, EImageBW8* pDst)
```

Parameter:

<code>pSrc</code>	Zeiger auf das Quellenbild
<code>pDst</code>	Zeiger auf das Zielbild

8.6.1.4. *ImgThreshold()*

Binarisiert ein Grauwert Bild dadurch, dass alle Bildpunkte unterhalb des Schwellwerts auf den Wert 0 und alle Punkte über oder auf dem Schwellwert auf den Wert 255 gesetzt werden. Verschiedene Einstellmodi sind vorhanden:

Funktionsdeklaration:

```
void ImgThreshold (EImageBW8* pSrc, EImageBW8* pDst, int threshold);
```

Parameter:

pSrc	Zeiger auf das Quellenbild
pDst	Zeiger auf das Zielbild
threshold	Schwellwert

8.6.1.5. *ImgMedian()*

Wendet das Median Filter auf ein Bild an (Mittelpunkt der Grauwerte in einer 3x3 Nachbarschaft).

Funktionsdeklaration:

```
void ImgMedian (EImageBW8* pSrc, EImageBW8* pDst);
```

Parameter:

pSrc	Zeiger auf das Quellenbild
pDst	Zeiger auf das Zielbild

8.6.1.6. *ImgConvRoberts()*

Extrahiert die Kanten eines Bildes anhand eines 2x2 Kernels (Kap. 7.8.).

Funktionsdeklaration:

```
void ImgConvRoberts (EImageBW8* pSrc, EImageBW8* pDst);
```

Parameter:

pSrc	Zeiger auf das Quellenbild
pDst	Zeiger auf das Zielbild

8.6.1.7. *BuildObjects ()*

Segmentiert ein Bild in zusammenhängende Objekte, die gleiche Grauwerte enthalten. Diese Objekte können dann anhand verschiedener Attribute geordnet werden, z.B. nach ihrer Grösse, Umfang, Schwerpunkt etc.

Funktionsdeklaration:

```
void BuildObjects (EROIBW8* pSrcImage);
```

Parameter:

pSrcImage	Zeiger auf das Bild, in dem die Objekte gesucht werden sollen
-----------	---

8.6.1.8. *MsrFindPointAlongLine()*

Lokalisiert einen Punkt entlang einer Prüflinie. Die Kriterien für die Lokalisierung des Punktes können mittels Funktionen im Messobjekt eingestellt werden.

Funktionsdeklaration:

int **MsrFindPointAlongLine** (EImageBW8* pSrcImage, EMeasure *pMeasure);

Parameter:

pSrcImage Zeiger auf das Bild, in dem der Punkt gesucht werden soll
pMeasure Zeiger auf ein Messobjekt. Dieses enthält alle Einstellungen zur gewünschten Suche. Das Resultat ist ebenfalls in diesem Objekt enthalten.

8.6.1.9. *Match ()*

Vergleicht ein Bild mit einem im Voraus gelerntem Bild. Es wird der Mittelpunkt vom Gebiet bestimmt, das am besten auf das Bild passt. Die Resultate können über entsprechende Funktionen abgerufen werden.

Funktionsdeklaration:

int **Match** (EImageBW8* pMatchImage)

Parameter:

pMatchImage Zeiger auf das Bild, welches verglichen werden soll

9. Klassifikation

Das Ohr wird mit seinen Attributen als einen mehrdimensionalen Vektor interpretiert. Zwecks besserer Visualisierung wird im folgenden vom dreidimensionalen Raum gesprochen. Alle diese Überlegungen können auch in höherdimensionalen Räumen durchgeführt werden, wie dies beim Ohr der Fall ist.

Die im Archiv vorhandenen Ohren kann man sich als Punkte im Raum vorstellen (P1..P4). Zur Identifikation eines Ohres, werden die Messwerte des Ohres im Raum eingetragen was zu einem weiteren Punkt führt. Es werden nun die Distanzen (d1..d4) von diesem Punkt zu jedem schon vorhandenen Punkt gemessen. Ist die Distanz kleiner als ein definierter Radius und die Differenz zwischen der kleinsten und der zweit kleinsten Distanz genügend gross, wird das gemessene Ohr als identifiziert angesehen.

Fig. 9.1

Die Distanzen sind als die Beträge der Differenzvektoren zwischen dem gemessenen Vektor und den im Archiv vorhandenen Vektoren zu betrachten.

Weil die verschiedenen Parameter verschieden skaliert und verschieden variieren muss eine Normierung durchgeführt werden. Dafür ist die Mahalanobisdistanz r geeignet. Sie lässt sich wie folgt berechnen:

C^{-1}_x ist die Kovarianzmatrix für x . In dieser Matrix sind die Zusammenhänge der Varianzen der einzelnen Merkmale eingetragen. Sind die Merkmale unkorreliert, entsteht eine Diagonalmatrix (Fig.9.2). Auf der Diagonalen sind die Varianzen der einzelnen Merkmale zu finden. In diesem Fall können die einzelnen Merkmale einfach mit der zugehörigen Varianz normiert werden.

Fig. 9.2 Diagonalmatrix mit den Varianzen der einzelnen Attributen
Da die Merkmale unkorreliert sind wurde diese Vereinfachung vorgenommen. Dies ist ein zusätzlicher Vorteil, weil die Kovarianzmatrix schwer zu berechnen ist und viel Zeit und Speicher beansprucht.

9.1.1. Bestimmung der Varianz

Um die Varianz möglichst gut zu bestimmen, ist es nötig entweder eine gute statistische Annahme zu treffen, oder es ist eine grosse Testmenge nötig. Aus Gründen von mangelhafter Theorie in diesem Bereich, ist es nicht möglich eine gute statistische Annahme für dieses Problem zu finden. Deshalb werden die Varianzen mittels einer Testmenge bestimmt. Je grösser diese Testmenge ist, desto besser wird die daraus ermittelte Varianz. Die Varianz für kleine Bibliotheken ist fest vorgegeben. Diese Standardwerte wurden mit einer Bibliothek von 12 Ohren erstellt. Sind mehr Ohren in einer Bibliothek wird die Varianz aus den in der Bibliothek vorhandenen Werten berechnet.

Bevor die Varianz eines Attributes berechnet werden kann muss der Mittelwert m_{Attribut} aller vorhandenen Werte dieses Attributes ermittelt werden.

Die Varianz wird wie folgt berechnet:

9.1.2. Raumnormierung

Wie schon erwähnt, werden alle Attributwerte mit der berechneten Varianz normiert. In dem so normierten Raum wird die Distanz nun nach Euklid bestimmt. Um die Sicherheit der Klassifizierung zu bestimmen kann nun ein Toleranzradius bestimmt werden. An eine Zutrittskontrolle werden zwei gegensätzliche Anforderungen gestellt:

1. Personen mit Zutrittsberechtigung müssen Zutritt erhalten
2. Personen ohne Zutrittsberechtigung muss der Zutritt verwehrt bleiben.

Es kann aber nur eine von beiden Bedingungen optimiert werden. In der Regel muss ein guter Kompromiss gefunden werden. Mit dem oben erwähnten Toleranzradius hat man nun die Möglichkeit die Sicherheit seiner Anwendung anzupassen. Wird der Toleranzradius gross gewählt, wird die erste Anforderung auf Kosten der zweiten besser erfüllt. Wird der Toleranzradius klein gewählt verhält es umgekehrt. (Fig. 9.3)

a) b)

Fig. 9.3 a) Toleranzradius zu klein in diesem Fall keine Identifikation
b) Toleranzradius etwas grösser als in a). Den Messwerten kann ein Referenzpunkt zugewiesen werden (Identifikation).

10. Grobpflichtenheft

10.1. Einführung

10.1.1. Zweck des Dokuments

Definition der zentralen funktionalen und nichtfunktionalen Anforderungen an das Softwaresystem.

10.2. Funktionale Anforderungen

10.2.1. Das Programm Earyfier

Das Programm Earyfier ist ein Softwareprogramm, mit dem in erster Linie eine Verifikation einer Person anhand ihres Ohres durchgeführt werden kann. Das heisst das Programm misst das Bild der Person aus und vergleicht die Messwerte mit den im Archiv (oder Batch) gespeicherten Werten. Da das Programm nicht mit einem Batchreader ausgestattet ist, muss hier symbolisch das Name der Person aus einer Liste gewählt werden.

Im einem zweiten Schritt, soll es möglich sein, nicht nur eine Verifikation, sondern eine Identifikation durchführen zu können. Das bedeutet, dass das Programm anhand der Messwerte, das korrespondierende Ohr im Archiv findet, falls dieses bereits im Archiv vorhanden ist. Die Identifikation muss Online durchgeführt werden können. Zusätzlich, müssen dem Archiv neue Ohren hinzugefügt, sowie schon vorhandene gelöscht werden können.

10.2.2. Datenhaltung

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

10.2.3. Zusammenhänge (UML- Diagramm)

Das folgende Klassendiagramm zeigt die wichtigsten Zusammenhänge auf. (UML-Notation)
Fig. 10.1 Klassendiagramm OOA

10.3. Funktionale Anforderungen im Einzelnen

10.3.1. Use Cases

10.3.1.1. Übersicht

Vorkommende Aktoren

Benutzer: Person (Programmbenutzer), welche ein neues Ohr ins Archiv eintragen/entfernen oder eine Verifikation/Identifikation durchführen will.

Fig. 10.2 Use Case Diagramm

10.3.1.2. Use Case Beschreibungen

UC 10: Ohr dem Archiv hinzufügen

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

Auslöser (Aktor): Benutzer

Vorbedingung: Es befindet sich eine Person im Aufnahmeraum oder es wurde ein Bild geladen.

Ablauf:

- Bild des Ohres einlesen
- Bild analysieren
- Werte im Archiv speichern

UC 20: Ohr aus dem Archiv entfernen

Kurzbeschreibung:	Der Benutzer kann Ohren aus dem Archiv entfernen.
Auslöser (Aktor):	Benutzer
Vorbedingung:	Es befindet sich mindestens ein Ohr im Archiv.
Ablauf:	Ohr, welches gelöscht werden soll, aus dem Archiv auswählen Sicherheitsabfrage quittieren Ohr löschen

UC 30: Ohr verifizieren/identifizieren

Kurzbeschreibung:	Der Benutzer kann ein Ohr verifizieren/identifizieren lassen.
Auslöser (Aktor):	Benutzer
Vorbedingung:	Es wurde in der Aufnahmekabine ein Bild eingelesen oder es wurde ein Bild eines Ohres geladen
Ablauf:	Bild analysieren Analysedaten mit Archivdaten vergleichen und falls eine Übereinstimmung stattgefunden hat, den Namen der erkannten Person ausgeben

10.4. Nichtfunktionale Anforderungen

10.4.1. Einführung

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

10.4.2. Benutzerschnittstelle

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

10.4.3. 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 wäre.

10.4.4. 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].

11. Anforderungsspezifikation

11.1. Klassendiagramm (UML-Notation)

Im Folgenden Klassendiagramm werden nur die Verbindungen zwischen den einzelnen Klassen aufgezeigt. Ein ausführliches Klassendiagramm mit allen Kardinalitäten, Funktionen, Attributen etc. ist im Anhang zu finden.

Fig. 11.1 Klassendiagramm Übersicht

11.2. Klassenbeschreibung

11.2.1. Klasse Ear

Ein Objekt der Klasse Ohr entspricht einem Ohr mit all seinen Attributen, inklusive dem Namen der Person, zu der das Ohr gehört. Die Klasse enthält alle notwendigen Funktionen zur Erzeugung und Benutzung dieser Daten.

Attribute

m_EIEarTruePtr	Zeiger auf das Originalbild des Ohres
m_EIEarLinePtr	Zeiger auf das Linienbild des Ohres
m_EIEarContourPtr	Zeiger auf das Bild mit der Extrahierten Ohrkante des Ohres
m_EIEarOnlyPtr	Zeiger auf das Bild mit der Ohrmaske
m_EIEarIsolatedPtr	Zeiger auf das Bild mit den extrahierten Kanten der Ohrleiste
m_EVContour	Vektor der die Ohrkontur enthält
m_LCutLine	Linie, welche die Ohrkontur schliesst, vom Start- zum Endpunkt des Ohres
m_sName	Name des Ohres (Besitzer)
m_PEarBottom	Koordinate des äussersten Punktes unten am Ohr
m_PEarTop	Koordinate des äussersten Punktes oben am Ohr
m_PCofgravity	Koordinaten des Schwerpunktes des Ohres
m_PNoseRef	Koordinaten des Nasenspitzes
m_PEyeRef	Koordinaten des Talpunktes zwischen Nase und Stirn
m_fAngleEar	Lage (Winkel) des Ohres absolut gemessen
m_fAngleEyeNose	Winkel zwischen den Punkten Nase- Ohr (Schwertpunkt) und Auge
m_fProportion	Beschreibt das Verhältnis der Höhe zur Breite des Ohres (Proportionen)
m_fOhrleiste	Ein Mass für die Dicke der Ohrleiste
m_fKrueimmungEarlobe	Ein Mass für die Krümmung des Ohrläppchens
m_fAbsAngleEarlobe	Ein Mass für die Richtung, die die Kontur Ohrläppchen durchläuft
m_fAmountEarlobe	Der Betrag des Richtungsvektors des Ohrläppchens
m_fCurvProportion	Das Verhältnis der Krümmung rechts und oben am Ohr
m_fWidthProportion	Ein Mass für die Konkavität des der Ohrform
m_fAngleuplift	Winkel vom untersten Punkt des Ohres zum äussersten Punkt auf der rechten Seite
m_fAngleAnsatz	Lage (Winkel) des Ohres bezogen auf die Gerade zwischen Nosetip und EyeRef
m_fArea	Ein Mass für die Fläche des Ohres

Funktionen

Alle Get- und Set- Funktionen der einzelnen Attribute sind an dieser Stelle nicht aufgeführt.

GetKruemmung() Liefert die durchschnittliche Krümmung über ein beliebiges Konturstück

GetDirectionVector() Liefert die Richtung und den Betrag eines Konturstückes

CalcAngle() Berechnet den Winkel zwischen 2 Punkten bezogen auf die Horizontale

GetEarImage() Liefert ein Bild des Ohres zurück. Es kann mittels Parameter das gewünschte Bild gewählt werden.
 0 → Liefert das Originalbild,
 1 → Liefert das Linarisierte Bild,
 2 → Liefert das Bild der Extrahierten Ohrkante
 3 → Liefert das Bild der Ohrmaske

FindContours() Erzeugt aus dem Originalbild ein Linienbild

FindEarContour() Extrahiert aus dem vorhandenen Linienbild des Ohres diejenige Linie, welche die Ohrkontur enthält

CloseContour() Schliesst Linien, deren Enden nahe bei einander liegen. Der Abstand kann über Parameter eingestellt werden

Skeleton() Skelettiert ein Bild nach der 8-Metrik Methode von Zamperoni [Zamp. 91]

CreateEar() Ermittelt Start- und Endpunkt des Ohres und erzeugt daraus eine Maske

IsolateEar() Erzeugt ein Linienbild des Ohres, zur Messung der Ohrleiste

CalcForm() Ermittelt die Ohrform

CalcCofgravity() Berechnet den Schwerpunkt des Ohres

CalcEarAngle() Ermittelt den absoluten Winkel des Ohres (Lage)

CalcAngleNoseEye() Ermittelt den Winkel zwischen Nase Ohr und Auge

CalcEarlobe() Berechnet die drei Attribute, welche die Ohrläppchenform beschreiben

CalcEarLeiste() Ermittelt die Dicke der Ohrleiste

CalcProportion() Berechnet das Verhältnis von Höhe zu Breite des Ohres

CalcAngeAnsatz() Ermittelt den relativen Winkel (Lage) des Ohres

CalcArea() Berechnet die Fläche des Ohres bezogen auf das Quadrat der Strecke zwischen Nase und Ohr (Schwerpunkt)

11.2.2. Klasse Identifikation

Die Klasse Identifikation enthält ein „aktuelles“ Archiv

Attribute

m_fSigmaAngleWidth	
m_fSigmaWidthProportion	
m_fSigmaAmountEarlobe	
m_fSigmaCurvProportion	
m_fSigmaAbsAngleEarlobe	
m_fSigmaEarLeiste	Standardabweichung jedes Attribute des Ohres
m_fSigmaKruemmungEarlobe	
m_fSigmaAngleEyeNose	
m_fSigmaAngleEar	
m_fSigmaProportion	
m_fSigmaArea	
m_ELArchivePtr	Zeiger auf das verwendete Archiv

Funktionen

CalcSigma()	Berechnet alle Standardabweichungen
Identify()	Identifiziert das Ohr, das der Funktion übergeben wird. Bei erfolgreicher Identifikation wird die Position im Archiv zurückgegeben.
Verify()	Verifiziert das Ohr, das der Funktion übergeben wird. Liefert TRUE falls die Verifikation erfolgreich ausgeführt wurden sonst FALSE.

11.2.3. Klasse grabber

Die Klasse stellt die Verbindung zwischen Kamera und Programm her, und stellt alle Funktionen zur Verfügung, um Bilder von der Kamera einzulesen.

Attribute

m_pImage	Zeiger auf das eingelesene Bild von der Kamera
----------	--

Funktionen

grabNewImage()	Liefert ein neues Bild von der Kamera und legt Speicher dafür an
reGrabImage()	Liefert ein neues Bild von der Kamera, nach dem grabNewImage() bereits ausgeführt wurde

11.2.4. Klasse Earlib

Die Klasse Earlib modelliert ein Archiv von Ohren und ihrer Attribute. Dies wird mit einer doppelt verketteten Liste realisiert. Die Klasse Earlib ist von der Klasse Olist abgeleitet.

Funktionen

InsEar()	Fügt ein Ohr am Anfang in die Liste ein
DelEar()	Löscht ein Ohr aus der Liste
GetEar()	Liefert den Zeiger auf das gewünschte Ohr zurück
Save()	Speichert die ganze Liste ab

2. Studienarbeit
Bildverarbeitung
Personenidentifikation mittels biometrischer Merkmale des Ohres

Load()	Lädt die ganze Liste
Decode() Ohr ab	Enschlüsselt den Char-Array aus dem Archiv und legt die Daten in ein Ohr ab

11.2.5. 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

11.2.6. 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

11.2.7. Klasse konstanten

Die Klasse konstanten enthält alle notwendigen Konstanten, die Programm verwendet werden, und schnell geändert werden können.

PI	Pi
WIDTH	Breite eines Bildes des gesamten Kopfes
HIGHT	Höhe eines Bildes des gesamten Kopfes
AREAWIDTH	Breite eines Ohrenbildes
AREAHIGHT	Höhe eines Ohrenbildes
SCHWARZ	Steht für den COLORREF RGB(0,0,0) →schwarz
WEISS	Steht für den COLORREF RGB(255,255,255)
→weiss	
DREHWINKEL_LINKS_OBEN	Schwellwert zur Findung des Endpunktes
DREHWINKEL_RECHTS_OBEN	Schwellwert zur Findung des Endpunktes
DREHWINKEL_LINKS_UNTEN	Schwellwert zur Findung des Startpunktes
WINDOW_DREHSINN	Grösse des Fenster zur Drehsinnermittlung
REFCOUNT	Anzahl Aufnahmen(Bilder) für die Aufnahme ins
Archiv	
SLIDER_MULTIPLIFIER	Skalierfaktor für die Sicherheitseinstellung
MAX_REF_OHREN	Anzahl Referenzohren

11.2.8. Klasse Pixel

Die Klasse Pixel modelliert einen Bildpunkt. Die Klasse enthält alle notwendigen Funktionen zum Setzen und Holen 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

11.2.9. 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() Endcoordinate)	Füllt den Vektor mit den Punkten auf der Linie (inkl. Start- und
GetLineVector() enthält (inkl. Start- und Endcoordinate)	Liefert einen Zeiger auf den Vektor, der alle Punkte der Linie
GetStart()	Liefert die Startcoordinate der Linie
GetEndt()	Liefert die Endcoordinate der Linie
SetStart()	Setzt die Startcoordinate der Linie
SetEnd()	Setzt die Endcoordinate der Linie
Draw() Farbe	Zeichnet eine Linie vom Startpunkt zum Endpunkt in der aktuellen

12. Bedienungsanleitung

12.1. Installation

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

Bemerkungen:

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

12.2. Der Earyfier

Nach dem Start von Earyfier.exe erscheint ein Fenster wie in Fig. 12.1

Fig. 12.1 Der Eröffnungsbildschirm des Earyfier

12.2.1. 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.

12.2.1.1. Filemenü

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

Quit: Beendet die Anwendung.

12.2.1.2. Archivemenü

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

Save: 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: Legt ein neues leeres Archiv an. Es erscheint ein Dialogfeld in dem der Benutzer aufgefordert wird, einen Namen für das neue Archiv einzugeben (Fig. 12.2). Nach der Eingabe wird ein neuer Ordner im Verzeichnis in dem der Earyfier sich befindet angelegt. In diesem neuen Ordner wird dann das neue Archiv mit dem gewählten Namen erstellt.

Fig. 12.2 Dialog um ein neues Archiv zu erstellen

Add Ear: Mit dieser Funktion kann ein neues Ohr dem aktuellen Archiv hinzugefügt werden.

Del Ear: Mit dieser Funktion kann ein Ohr (Person) aus dem Archiv entfernt werden. Es erscheint ein Dialogfeld in dem der Benutzer aufgefordert wird, den Namen der Person, die aus dem Archiv gelöscht werden soll, auszuwählen (Fig. 12.3).

Fig. 12.3 Personenauswahl, löschen einer Person

12.2.1.3. Cameramenü

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

One shot: Der Earyfier funktioniert ähnlich wie ein Fotoapparat, wobei der „Image from camera“-Button dem Auslöser entspricht.

TV: Der Computer macht einmal pro Sekunde ein Bild und zeigt es im Grafikbereich an. Durch drücken des „Image from camera“-Button, wird automatisch das aktuelle Bild gehalten und die Kamera in den One shot-Modus umgeschaltet.

Off: Schaltet die Kamera ab.

12.2.1.4. *Statusmenü*

Verification: Im Modus Verification muss der Benutzer seinen Namen aus dem Archiv wählen (Fig 12.4). Darauf wird geprüft ob das aktuelle Ohr demjenigen aus dem Archiv genügend ähnlich ist. Ist die Ähnlichkeit ausreichend, wird das Ohr (Person) Verifiziert, sonst wird angegeben, dass das Ohr nicht verifiziert werden konnte.

Fig. 12.4 Personenauswahl, Verifikation.

In der Realität kann man sich an Stelle der Auswahl des Namens vorstellen, dass der Benutzer zum Beispiel eine Magnetkarte einführt, auf welcher sein Name oder gar die kompletten Daten seines Ohres gespeichert sind.

Identification: Im Modus Identification wird das Archiv nach dem ähnlichsten Ohr zum aktuellen Ohr durchsucht. Darauf wird die Ähnlichkeit mit dem zweit ähnlichsten Ohr verglichen. Ist der Unterschied genügend gross, wird das Ohr identifiziert und der entsprechende Name ausgegeben. Ansonsten wird bekannt gegeben weshalb keine Identifikation zustande kam.

12.2.2. Grafikbereich

Fig. 12.5 Grafikbereich

Aufnahme der Kamera /

Geladenes Bild: In diesem Bild ist das von der Kamera aufgenommene, bzw. das aus einem File geladene Bild zu sehen.

Ausschnitt

mit dem Ohr: Nach dem drücken des Buttons [Find Area] wird im Originalbild das Ohr gesucht. Die Region um das Ohr wird in diesem Bild angezeigt.

Zweipegelbild: Hier wird das Zweipegelbild des Ohres angezeigt.

Kantenbild: Hier wird das Resultat der Konturextraktion angezeigt, die auf das Zweipegelbild angewandt wird.

Extrahiertes Ohr: Ausgeschnittenes Ohr bereit zur Messung der Merkmale.

12.2.3. Identifikationskontrollen

Über diese Kontrollen wird die Identifikation eines geladenen oder mit der Kamera aufgenommenen Ohres durchgeführt.

Fig. 12.6 Identifikationskontrollen

Image from

camera: Ein neues Bild wird von der Kamera eingelesen und zur weiteren Bearbeitung bereitgestellt.

Find Area: Es wird das Ohr im Originalbild gesucht und angezeigt.

Match Nr: Bestimmt die Anzahl Referenzohren, die benutzt werden um das Ohr im Originalbild zu finden. Dieser Wert kann über den entsprechenden Spinbutton oder durch direktes eintragen in der Editbox verändert werden.

Threshold: Dieser Editbox wird der Schwellwert für das Zweipegelebild eingetragen. Ist der Schwellwert schon richtig eingestellt, muss nur der [Threshold] Button gerückt werden. Der Wert kann über den entsprechenden Spinbutton oder durch direktes eintragen in der Editbox verändert werden.

Find Contours: Es wird die Kontur des Ohres ermittelt und angezeigt.

Extract Ear: Es wird die Kontur des Ohres verarbeitet und das extrahierte Ohr angezeigt.

Calculate

Ear Attribute: Misst das ganze Ohr aus und hält die Messwerte fest.

Identify: Vergleicht die Werte des ausgemessenen Ohres mit den Werten die sich im aktuellen Archiv befinden. Wird das Ohr erkannt erscheint ein Fenster dem der Name der Person zu entnehmen ist.

Sicherheit: Bestimmt die Grösse des virtuellen Toleranzradius. Je höher die Sicherheit, desto besser muss die Übereinstimmung zwischen den Archivwerten und den Messwerten sein damit die Person erkannt wird. Zugleich nimmt jedoch die Wahrscheinlichkeit einer Verwechslung ab.

13. Tests und Resultate

13.1. Einführung

Die ersten Tests haben gezeigt, dass die Qualität der Aufnahmen sehr entscheidend ist. Die komplette Testreihe ist dem Anhang zu entnehmen. Im folgenden Kapitel werden verschiedene Auszüge aus dieser Testreihe für Erklärungen aufgeführt. Da die Identifikation auf einer minimal Distanzklassifizierung basiert sind die einzelnen Distanzen zu untersuchen.

13.2. Testarchiv

Um eine Person in ein Archiv aufzunehmen wird von der Person das Ohr fünf mal erfasst und ausgemessen. Das Programm mittelt die Messwerte der fünf Aufnahmen und legt die so berechneten Werte in ein Archiv ab. Das verwendete Testarchiv hat den Namen 06_12_5Ref.txt.

13.3. Testauswertung

In diesem Test wurden die Ohren von 40 Personen verglichen. Von jeder Person wurden fünf Aufnahmen geprüft. Bei jeder Aufnahme wurde das Ohr ausgemessen und die

Distanzen zu jedem Ohr des Archivs ermittelt. Für eine erfolgreiche Identifikation muss die Distanz zum eigenen, im Archiv gespeicherten Ohr am kleinsten sein. Zudem muss auch der Unterschied zur zweit kleinsten Distanz genügend gross sein.

13.3.1. Für das Verständnis der Testtabellen

Die Distanz einer Aufnahme zum Archivwert der Person ist im Normalfall gelb und im Fehlerfall rot unterlegt. Tritt ein Fehler auf ist auch der Wert der zum Fehler führt rot unterlegt. Die Numerierung der Spalten ist wie folgt organisiert. In der ersten Spalte sind die Personennummern eingetragen. In der zweiten Spalte sind die Namen der Testpersonen zu finden. In den weiteren Spalten sind die Distanzen der jeweiligen Aufnahme zu den Archivwerten eingetragen. Die erste Zahl im Spaltentitel ist die Personnummer der Person, die auf dieser Aufnahme abgebildet ist. Die zweite Zahl ist die Nummer der Aufnahme, unter welcher sie im Bildarchiv zu finden ist.

Fig. 13.1 Hilfe für das Verständnis der Testtabellen; Beispieltabelle

13.3.2. Normalfall

Als Normalfall wird der Fall betrachtet bei dem die unter 13.3. genannten Bedingungen erfüllt sind. Das heisst eine Person wird richtig erkannt und identifiziert. Im Idealfall wären alle Testresultate hier einzuordnen. Dieser Fall soll hier mit einem Ausschnitt aus der Testreihe verdeutlicht werden.

Fig. 13.2 Ausschnitt aus der Testreihe: Normalfall

Aus der Tabelle ist erkennbar, dass die Distanz vom Archivwert zur entsprechenden Aufnahme (gleiche Personnummer) in der ganzen Spalte klar am tiefsten ist.

Die Distanz zum eigenen Ohr ist in der Regel viel kleiner, als die Distanzen zu den anderen Ohren. Daraus kann abgeleitet werden, dass die Merkmale, nach welchen die Ohren klassifiziert werden für eine gute Identifikation geeignet sind.

13.3.3. Fehlerfälle

13.3.3.1. *Zwischen zwei Referenzohren*

Dieser Fehler ist das Resultat einer Sicherheitsvorkehrung. Für eine erfolgreiche Identifikation muss, wie erwähnt der Unterschied zwischen der kleinsten und der nächst grösseren Distanz genug gross sein. Diese Massnahme wird am besten anhand eines zweidimensionalen Beispiels erklärt.

Fig. 13.3 Zweidimensionales Beispiel für die Bufferzone

Ohne die eingeführte Bufferzone (Fig. 13.3 a)) ist die Wahrscheinlichkeit für Verwechslungen viel höher als mit einer Bufferzone (Fig. 13.3 b)). Im Beispiel von oben gehört M1 zu P1 und M2 zu P2. Im Fall a) findet eine Verwechslung statt. Im Fall b) hingegen, wird niemand erkannt. Ob der daraus entstandene Schaden nun bei a) oder bei b) grösser ist, lässt sich nicht verallgemeinern. Es kommt vielmehr auf die jeweilige Anwendung an. Am Beispiel einer Präsenzerfassung, wo beim Fall a) auf ein falsches Konto zugegriffen wird, ist der Schaden wohl grösser.

In unserem Test trat dieser Fehler bei fünf Messungen auf. In zwei von diesen fünf Fällen (Test: 35.4, 36.2) ist die Situation ähnlich der, die im Beispiel erwähnt wird. In den anderen drei Messungen hätte eine Verwechslung stattgefunden, da zwei andere Werte miteinander konkurrieren, die tiefer sind als der Wert der Person, die erkannt werden sollte (Test: 34.4, 32.4, 20.1). Zu beachten ist aber, dass alle Distanzen in diesen Messungen relativ gross sind (Werte > 2). Das heisst entsprechend stark vom Archivwert abweichen. Der Grund dafür ist meist in der Qualität der Aufnahme zu suchen.

Fig. 13.4 Ausschnitt aus der Testreihe: Zwischen zwei Referenzohren

13.3.3.2. Distanz zu jedem Ohr zu gross

Das ist der Fall wenn die kürzeste Distanz grösser ist, als die maximal erlaubte Distanz. Dieser Fehler hat sehr viel mit der Sicherheit des Systems zu tun. Für den Test war die Sicherheit minimal eingestellt, was bedeutet, dass die maximal erlaubte Distanz 5 ist. Im Test trat nur einmal ein solcher Fall auf (Fig. 13.5). Hier ist die Qualität der Aufnahme sehr schlecht.

Fig. 13.5 Ausschnitt aus der Testreihe: Distanz zu jedem Ohr zu gross

13.3.3.3. Verwechslung

Bei diesem Fehler ist die Distanz zu einem fremden Ohr im Archiv kleiner als die Distanz zum eigenen Ohr. Dies ist für eine Identifikation wohl der schlimmste Fehler. Im Test kommen 13 Verwechslungen vor. Dabei muss berücksichtigt werden, dass der Sicherheitsregler während dem Test auf der Position 0 stand. Wird die Sicherheit erhöht, kommen automatisch weniger Verwechslungen vor. Dafür taucht der unter Kap. 13.3.3.2. besprochene Fehler häufiger auf.

Fig. 13.6 Ausschnitt aus der Testreihe: Verwechslung

13.4. Auswertung (Übersicht)

Fig. 13.7 Sicherheitsregler auf der Stufe 0.

Fig. 13.8 Sicherheitsregler auf der Stufe 3.

Fig. 13.9 Sicherheitsregler auf der Stufe 3,75.

Aus den Diagrammen soll der Zusammenhang des Sicherheitsschiebers und den Resultaten die das System liefert klar werden. Ist die Sicherheit gering (Fig. 13.7) können mehr Personen richtig identifiziert werden. Es kommen aber auch mehr Verwechslungen vor. Ist die Sicherheit höher (Fig. 13.8 und 13.9) werden zwar weniger Personen verwechselt, doch diesen Vorteil erkaufte man sich teuer, denn es können bedeutend weniger Personen identifiziert werden. Bei einer Sicherheit von 3,75 (Fig. 13.9) ist die Wahrscheinlichkeit, dass eine Verwechslung stattfindet, bei 0%. Jedoch wird nur in 6 von 10 Fällen die Personen erfolgreich identifiziert.

14. Epilog

14.1. Verbesserungsmöglichkeiten und Ideen

Ein Grundproblem dieser Diplomarbeit ist es, das Ohr vollständig aus der Originalaufnahme zu extrahieren. Dieses Problem hat uns auch am meisten Arbeit bereitet und Zeit gekostet. Und obschon wir eine recht gute Lösung für dieses Problems anbieten, wäre die Extraktion noch zu verbessern. Mit dieser Verbesserung des Algorithmus würde auch die Identifikation automatisch besser ausfallen, denn die Tests haben bewiesen, dass die ausgewählten Merkmale des Ohres ausreichen, um eine gute Identifikation zu realisieren. Ein anderes Problem ist die Aufnahme selber. Es hat sich gezeigt, dass bei den Aufnahmen immer wieder Bilder entstehen, die für eine weitere Verarbeitung unbrauchbar sind. Wenn man nun die Qualität der Aufnahmen steigern könnte, würde dies auch zu besseren Resultaten führen.

Eine Idee, die wir aus zeitlichen Gründen nicht umsetzen konnten, ist die, dass das Programm nicht nur Liveaufnahmen verarbeiten kann, sondern dies auch mit anderen Bilddokumenten möglich ist. Das würde bedeuten, dass z.B. anhand von Videoaufnahmen, oder Fotografien Personen identifizieren werden können.

14.2. Schlusswort

Uns war klar, dass die Grundlagen der Bildverarbeitung, die wir uns in der 2. Studienarbeit angeeignet hatten hier nicht ausreichen werden. Dies hat sich dann auch schnell bestätigt, und wir mussten uns mit tiefgründigen Problemen auseinandersetzen, die auch einen grossen Zeitaufwand zur Folge hatten. Dieser grosse Zeitrückstand konnte nie richtig aufgeholt werden und forderte am Ende dann doch einige Überstunden. Wir sind aber schlussendlich sehr zufrieden mit unserer Arbeit und haben das Gefühl, etwas erreicht zu haben. Die Arbeit ist in unseren Augen auch eine gute Vorbereitung auf das Berufsleben, in welches wir beide anschliessend eintreten werden.

An dieser Stelle danken wir Herrn Dr. H. Grossmann für seine Unterstützung und Betreuung. Er hat uns immer wieder bei der Behebung von Problemen geholfen und wir konnten von seinem grossen Wissen im Bereich der Bildverarbeitung profitieren. Im weiteren danken wir auch Herrn Schuster, der uns bei Fragen betreffend der Klassifikation immer hilfreich mit Rat zur Seite stand. Ein ganz besonderer Dank geht an alle Studenten, Dozenten und anderen Personen, die sich für Testaufnahmen immer wieder zur Verfügung stellten. Ohne sie wäre ein Test unseres Programmes in keiner Weise möglich gewesen.

15. 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
- [Klet. 95] Reinhard Klette/ Piero Zamperoni: Handbuch der Operatoren für die Bildbearbeitung, Vieweg 1995, ISBN 3-528-16431-X
- [Wahl 84] Friedrich M. Wahl: Digitale bildsignalverarbeitung, Springer-Verlag 1984, ISBN 0-387-13586-3
- [Hab. 91] Peter Haberäcker: Digitale Bildverarbeitung, Carl Hanser Verlag München Wien 1991, ISBN 3-446-16339-5
- [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

16. Glossar

Archiv: Ein Archiv besteht aus den Messwerten der zutrittberechtigten Personen.

Attribut: Als Attribute werden die 11 Messwerte des Ohrs bezeichnet. Es sind dies:

- ◆ Steigungswinkel des Ohres
- ◆ Verhältnis der Breite oben und unten am Ohr
- ◆ Betrag des Richtungsvektors des Ohrläppchens
- ◆ Verhältnis der Krümmung oben und rechts des Ohres
- ◆ Durchschnittliche Richtung(Winkel) der Ohrläppchenkontur
- ◆ Dicke der Ohrleiste(relativ)
- ◆ Durchschnittliche Krümmung des Ohrläppchens
- ◆ Winkel zwischen Nase und Augen. (Gemessen vom Ohr aus)
- ◆ Lage des Ohres(relativ)
- ◆ Verhältnis der Höhe zur Breite des Ohres (Proportionen)
- ◆ Fläche des Ohres(relativ)

Aufnahmekabine: Kap. 6.2.

Bildpunkt: Ein Bildpunkt beschreibt einen Punkt auf dem digitalisierten Bild. Er besitzt eine X und eine Y- Koordinate, die den Standort festhalten.

Bildsegment: Ein Bildsegment ist eine zusammenhängende Teilmenge des Bildes.

Bildsegmentierung: Ein Bild kann in disjunkte Bildsegmente zerlegt werden, indem Ränder von Bildsegmenten bestimmt werden oder Bildsegmente nach flächenorientierten Kriterien berechnet werden.

CCD-Kamera: (charge coupled device camera) Für Bildverarbeitungssysteme werden bevorzugt CCD-Kameras für die Bildeingabe verwendet. In der CCD-Kamera werden die durch das Vorsatz-Objekt erhaltenen analogen Bilder auf ein diskretes Sensorfeld abgebildet, welches im einfachen Fall nur eine Zeile ist (Zeilen.Kamera), aber i.a. als Matrix angeordnet ist (Matrix-Kamera).

Drehsinn: Kap. 7.9.

Earifyer: Earifyer ist der Name des Programmes (Software), das implementiert wurde.

Endpunkt: Punkt am Anfangsteil der Ohrleiste (Kap. 7.14.1.).

Euklidischer Abstand: Der euklidische Abstand zweier Punkte $p_1 = (x_1, y_1)$ und $p_2 = (x_2, y_2)$ der Ebene wird mit $d_2(p_1, p_2)$ bezeichnet und ist durch definiert.

Glättung: Durch eine Angleichung der Grauwerte (Glättung) eines Bildes werden „Ausreisser“ unterdrückt rauschbedingte zufällige Grauwertschwankungen eingeebnet und Texturmuster abgeschwächt.

Grauwert: Der Grauwert beschreibt die Farbe (Grauwert) eines Bildpunktes. Der Grauwert variiert zwischen 0 (Schwarz) und 255 (Weiss).

Kante: Im Grauwertbild ist eine Kante allgemein gesprochen durch eine starke Grauwertänderung zwischen zwei dicht beieinander liegenden Bildsegmenten charakterisierbar. Die durch die Kante getrennten Bildsegmente sind ebenfalls i.a. nicht durch gleichförmige Grauwerte beschreibbar bzw. können gestört sein.

Kantenbild: Ein Grauwertbild, in dem das Ergebnis eines Kantenoperators im entsprechenden Bildpunkt als Grauwert dargestellt ist.

Kantenextraktion: Mittels Kantenextraktion werden im digitalen Bild Bildmerkmale mit der zugeordneten Eigenschaft „Kantenwert“ oder „Kantenstärke“ berechnet. Dies kann als Bildtransformation erfolgen, indem die berechneten Bildmerkmale z. B. durch einen fixierten Grauwert markiert werden. Resultat einer Kantenextraktion kann ein Kantenbild sein.
Kap. 7.8.

Kantenoperator: Eine Bildtransformation zur Kantenextraktion wird Kantenoperator genannt.

Konturcode: Kap. 7.3.

Min. Distanz

Klassifikation: Klassifikation anhand des euklidischen Abstandes. Es wird der kleinste Abstand vom Messwert zu einer Referenz gesucht.

Ohrmerkmale: Kap. 8.

Ohrmerkmale: Originalaufnahmen von Ohren, die zur Eingrenzung des Ohrbereiches benutzt werden (Kap. 7.12.).

Schwellwertoperator: Ein Schwellwertoperator transformiert Grauwertbilder in Zweipegelbilder; er basiert auf einer Schwelle S mit $0 \leq S \leq 255$. Grauwerte u im Eingabebild werden auf 255 abgebildet, falls $u > S$ gilt. Anderfalls wird ihnen der Wert 0 zugeordnet.

Schwellenwert: Durch einen Schwellenwert wird ein Wertebereich in zwei Teilmengen zerlegt, z.B. jene Menge aller Werte, die kleiner oder gleich dem Schwellenwert sind, und in jene Menge aller Werte, die grösser als dieser Schwellenwert sind.

Skelettierung: Kap. 7.6.

Startpunkt: Punkt beim Übergang vom Kopf zum Ohrläppchen (Kap. 7.14.1.).

Zweipegelbild: In einem Zweipegelbild sind für die Bildwerte nur zwei Werte, z.B. 0 und 255 zugelassen.