Bestimmung der Gesichtspose unter Anwendung von Gaborwaveletnetzen

Diplomarbeit
am Lehrstuhl für Kognitive Systeme
des Instituts für Informatik und Praktische Mathematik
der Christian-Albrechts-Universität zu Kiel
Prof. Dr. Gerald Sommer

vorgelegt von
Sven Bruns

Betreuer: Dipl. Inform. Volker Krüger

Oktober 2000

Zusammenfassung

Posenschätzung ist eine Fähigkeit, die Menschen selbst unter schwierigen Bedingungen, wie schlechter Beleuchtung oder teilweiser Verdeckung, mit einer hohen Genauigkeit beherrschen. In dieser Arbeit wird ein erscheinungsbasiertes Verfahren zur nichtbelastenden Bestimmung der Gesichtspose in monokularen unkalibrierten Videosequenzen vorgestellt.
Es wird ein auf *Gaborwaveletnetzen* basierender Ansatz verwendet, der es ermöglicht, Gesichter anhand ihrer typischen Struktur zu lokalisieren. Dabei werden nicht einzelne Gesichtsmerkmale isoliert betrachtet, sondern die Menge aller Gesichtsmerkmale als Ganzes.
Durch ein auf Gaborwaveletnetzen basierendes Trackingverfahren wird ein Gesicht unter Verwendung eines *Gaborsuperwavelets* auf einem Standard-PC (450 MHz Pentium II) mit 20 Hz verfolgt und subpixelgenau lokalisiert.
Unter Verwendung künstlicher neuronaler Netze konnte die Gesichtspose eines Puppenkopfes mit einem mittleren quadratischen Fehler von weniger als $0.25°$ bestimmt werden. Dazu wurden als Netzeingabe die Gewichte des Gaborsuperwaveletnetzes sowie die Filterantworten der ungeraden Gaborfunktionen des Superwavelets untersucht. Das Generalisierungsverhalten der Waveletnetze hat sich beim Tracking und bei der Merkmalsextraktion als außerordentlich gut erwiesen.
Ferner wird die Eignung der Gaborwaveletnetze zur *progressiven Attention*, d.h. zu einer aufgabenorientiert angepaßten Aufmerksamkeit, in Experimenten untersucht und bestätigt.

Inhaltsverzeichnis

1 Einleitung

1.1 Motivation

Die Entwicklung der Schnittstelle zwischen Mensch und Maschine steckt im Umbruch. Die allgemeine Verfügbarkeit von Rechenleistung, wie sie noch vor wenigen Jahren teuren Groß- und Hochleistungsrechnern vorbehalten war, ermöglicht es, einen Teil dieser Leistung für die Realisierung einer benutzerfreundlichen Schnittstelle zwischen Mensch und Maschine zu verwenden. Besonders deutlich wird dies in der zunehmenden Verbreitung sprachgesteuerter Systeme. Diktiersysteme werden kommerziell vermarktet und funktionieren auf Standard-PCs. Diese Entwicklung stellt einen großen Schritt in der Entwicklung des Human-Computer-Interfaces (HCI) dar.
Eine der natürlichsten Arten der Interaktion mit der Umwelt stellt die Blickrichtung dar. Der Mensch schaut an, was ihn interessiert. Die Beobachtung der Blickrichtung erlaubt daher einen Rückschluß auf die Attention einer Person. Ein menschlicher Beobachter beherrscht diese Kunst mit einer hohen Genauigkeit.
Bisher existieren noch keine frei erhältlichen, auf jedem Standard-PC lauffähigen Systeme, die eine Steuerung des Rechners über den Blick ermöglichen. Es existieren am Markt bisher nur einfache Ansätze, wie z.B. die Videokamera Sony EVI-D31, welche eine autonome Gesichtsverfolgung bereitstellt. Die Anwendungen solcher einfachen Funktionen sind aber eher im Bereich der Videokonferenzen zu finden als im HCI.
Von besonderer Wichtigkeit für eine natürliche Schnittstelle zwischen Mensch und Maschine ist, daß die Bestimmung der Blickrichtung *nichtbelastend (non-intrusive)* ist, d.h. der Anwender keine aufwendige Ausrüstung tragen muß oder in seiner Bewegungsfreiheit eingeschränkt wird. Eine Sonderstellung nehmen dabei sehr leichte und kaum störende Geräte ein, wie z.B. Headsets für die Spracheingabe, welche allgemein akzeptiert werden.
Da in den letzten Jahren Farbvideokameras zur preiswerten Massenware geworden sind, rückt die Verwendung der Blickrichtung als Eingabemethode in den Mittelpunkt des Interesses. Die Verwendung der Blickrichtung hat gegenüber der Maus offensichtliche Vorteile, z.B. bleiben die Hände für andere Tätigkeiten frei. Dies ist insbesondere wegen der zunehmenden Zahl von Erkrankungen, die durch die unergonomische Arbeit am Computer auftreten, ein sinnvolles Anwendungsgebiet.

1.1.1 Die Gesichtspose als Näherung der Blickrichtung

Es muß zwischen der *Blickrichtung* und der *Gesichtspose* unterschieden werden. Die Blickrichtung ergibt sich aus der Stellung der Augen relativ zur Kamera, während wir unter der Gesichtspose die Normale der Gesichtsebene verstehen.
Es ist offensichtlich, daß die Blickrichtung der Gesichtspose in Bezug auf Genauigkeit und Universalität vorzuziehen ist, da die Blickrichtung die präziseste Möglichkeit darstellt, die Attention zu bestimmen. Allerdings stellt die Verwendung der Blickrichtung hohe Anforderungen:

- Die Augen müssen im Kamerabild als Merkmale detektiert werden. Durch das Auftreten von Lidschlag, Lichtreflexen sowie Brillen ist eine robuste Lösung schwierig.

- Für eine hohe Genauigkeit werden hochaufgelöste Bilder der Augenregion benötigt.

Diese Anforderungen lassen sich in der Praxis schwer erfüllen. Die direkte Bestimmung der Blickrichtung aus der *Stellung der Augäpfel*, d.h. der Verformung der Iris, kann nur unter Verwendung hochaufgelöster Bilder der Augenregion durchgeführt werden. Wenn sich eine Person frei im Raum bewegen darf, stehen solche Bilder in der Regel nicht zur Verfügung[1]. In diesem Fall kann die Blickrichtung anhand der *Stellung der Pupillen relativ zur Gesichtspose* bestimmt werden.

Demgegenüber bietet die Bestimmung der Gesichtspose einige Vorteile:

- Die Gesichtspose ist eine gute Näherung der Blickrichtung und kann als Vorstufe zur Augendetektion verwendet werden.

- Es ist eine erscheinungsbasierte Bestimmung möglich, was der Robustheit zugute kommt, da eine Fehldetektion einzelner Merkmale eine deutlich geringere Rolle spielt.

- Der Bewegungsspielraum für den Kopf ist größer, da das ganze Gesicht betrachtet wird.

Ein offensichtlicher Nachteil der Verwendung der Gesichtspose ist die Tatsache, daß die Pose in manchen Situationen stark von der Blickrichtung abweichen kann, wie es in Abbildung 1 illustriert ist. Da der Kopf dem Blick nachgeführt wird, treten solche großen Abweichungen in der Praxis selten auf. Deshalb erscheint eine Verwendung der Gesichtspose anstelle der Blickrichtung in vielen Fällen vertretbar. Vielmehr sind große Abweichungen zwischen Blickrichtung und Pose ein Zeichen für eine kurzfristige Ablenkung der Aufmerksamkeit, wie sie z.B. durch ein klingelndes Telefon verursacht werden kann. Die Gesichtspose kann in dieser Situation zur Validierung der Blickrichtung verwendet werden.
Ferner ist zu bedenken, daß im Fall von verdeckten oder durch Lidschlag geschlossenen Augen die Pose die *einzige* Möglichkeit darstellt, Rückschlüsse auf die Attention zu ziehen.

1.1.2 Anwendungsmöglichkeiten

Eine funktionierende Blickrichtungsdetektion eröffnet ein breites Spektrum an Anwendungsmöglichkeiten:

[1] Manche Arbeiten (z.B. [12]) behelfen sich damit, den Kopf der Testperson zu fixieren, was der Idee der nichtbelastenden Bestimmung entgegensteht.

(a) (b) (c) (d)

Abbildung 1: *Dargestellt sind vier Einzelbilder einer Sequenz. Die Testperson fixiert in allen Bildern den gleichen Punkt auf dem Monitor. Es wird deutlich, daß die Gesichtspose allein nicht für eine eindeutige Bestimmung der Blickrichtung ausreicht.*

- Die wahrscheinlich wichtigste Anwendung ist die Navigation am Bildschirm als Ersatz der Maus. Im Zusammenhang mit Spracherkennung ist damit eine Bedienung des Computers ohne Verwendung der Hände möglich. Auch für körperlich behinderte Menschen wäre die normale Arbeit am Computer - und damit die Ausübung eines Berufes - wieder möglich.
- In der Wahrnehmungsforschung ist man an der Wanderung des Blicks über eine Szene interessiert. Eine nichtbelastende Bestimmung der Blickrichtung führt zu unverfälschteren Ergebnissen als Systeme, die sich permanent durch ihre Beschränkungen (Gewicht, Bewegungsspielraum) in Erinnerung rufen.
- Virtuelle Realität: Fernsteuerung von Aktionen an entfernten oder unzugänglichen Orten.
- Blickrichtungsdetektion ermöglicht *kontextsensitive Systeme*, in denen der Benutzer Bezug auf die Umwelt nehmen kann. So kann der Benutzer Fragen wie *Was ist das?* stellen. Das System kann dann anhand der Blickrichtung des Anwenders das gemeinte Objekt lokalisieren und ggf. identifizieren.
- Ein System kann erkennen, ob es der Empfänger eines gesprochenen Kommandos ist oder ob sich mehrere Personen unterhalten.
- Ein System kann einem zögernden Benutzer die zum Objekt in Blickrichtung passende Hilfestellung geben.
- Im detektierten Gesicht können die Lippenbewegungen zur Unterstützung einer Spracherkennung herangezogen werden.

Ein oft vernachlässigter Punkt bei der geplanten Verwendung der Blickrichtung zur Bildschirmnavigation ist die Bereitstellung einer geeigneten Benutzeroberfläche. Diese muß sich in zwei wesentlichen Punkten von heute üblichen Oberflächen unterscheiden: Zum einen beträgt die maximal mögliche Genauigkeit der Blickrichtungsbestimmung etwa 1° (siehe Abschnitt 6.3). Das entspricht bei einem Betrachtungsabstand von

50 cm einer Unsicherheit von 9 mm bzw. über 30 Pixeln. Zum anderen soll darauf hingewiesen werden, daß man die Blickrichtung im Gegensatz zur Maus nicht “loslassen” kann, d.h. es muß sichergestellt sein, daß nicht unbeabsichtigt eine Aktion ausgelöst wird.

1.2 Arbeiten zur Posenbestimmung und Blickrichtungsdetektion

Die Forschung auf dem Gebiet der Blickrichtungs- und Posenbestimmung in monokularen Bildern teilt sich in zwei Hauptfelder auf: Auf der einen Seite stehen die *merkmalsbasierten Ansätze*, die versuchen, die Pose bzw. Blickrichtung anhand von Gesichtsmerkmalen geometrisch zu berechnen. Auf der anderen Seite sind die *erscheinungsbasierten Ansätze* zu nennen, bei denen die Auswertung von Farbinformationen bzw. Filterantworten vorgenommen wird.

Als Vertreter der merkmalsbasierten Ansätze ist die Arbeit von P. Ballard und G.C. Stockman [3] zu nennen, in der ein System vorgestellt wird, welches in der Lage ist, die Augen und Nase einer Person in einer monokularen Bildsequenz[2] zu verfolgen und daraus die Gesichtspose zu berechnen. Es wird ein natürliches Gesicht verwendet, ohne das Anbringen von Markierungen oder die Betonung bestimmter Gesichtsmerkmale durch Schminken. Die Autoren setzten sich als Ziel, über die detektierte Gesichtspose ein bestimmtes Feld innerhalb eines 8×8 Felder großen Monitorbildes zu selektieren. Dazu wurden zwei Vorgehensweisen untersucht: Zum einen wurden die Augen und der Mund durch Ellipsen modelliert und in jedem Bild gematcht. Es wurde angenommen, daß das Gesicht symmetrisch ist, daß beide Augen und der Mund vollständig sichtbar sind und daß der Kopf nicht nach links oder rechts geneigt ist. Ferner wurde vorausgesetzt, daß *die 3D-Abstände zwischen Mund und den beiden Augen bekannt sind*. Nur unter diesen Voraussetzungen ist das sogenannte *Perspective 3 Point-Problem (P3P-Problem)* lösbar. Damit ist das Verfahren für die allgemeine Verwendung nicht geeignet, da eine aufwendige Kalibrierung nötig ist. Die notwendige Ellipsenanpassung war 1995 mit 80 Sekunden/Bild viel zu langsam. Die erreichte Genauigkeit lag bei etwa 3°. Dabei ist kritisch anzumerken, daß der Mund ein nichtstationäres Gesichtsmerkmal ist, da seine Position und Form stark vom Gesichtsausdruck abhängt. Die relativ guten Ergebnisse beziehen sich daher mit Sicherheit auf einen konstanten Gesichtsausdruck.

Der zweite von Ballard und Stockman untersuchte Ansatz beruht auf der Verwendung von *Glanzlichtern (twinkles)*. Bei diesem Verfahren wurde das Gesicht der Testperson mit einer fest plazierten, punktförmigen, weißen Lichtquelle bestrahlt. Die Glanzlichter auf den Augen und der Nase wurden extrahiert. Für die Extraktion des “nose twinkle” war die Entwicklung eines aufwendigen Schwellwertverfahrens nötig. Es wurde angenommen, daß die Abstände zwischen beiden Augen und der Nasenspitze bekannt sind, daß die Augen kugelförmig sind und daß sich nur ein Gesicht im Bild befindet, da sonst die Glanzlichter nicht eindeutig zuzuordnen sind. Ferner müssen die

[2] allerdings nicht in Echtzeit

Augen sichtbar sein, d.h. die Testperson darf keine Brille tragen, da sonst Glanzlichter auf den Brillengläsern auftreten würden. Es kam zu Fehldetektionen durch Tränenbildung, hervorgerufen durch die direkte Beleuchtung des Gesichts. Der Zeitbedarf lag bei unter drei Sekunden, der mittlere Fehler betrug $5° - 10°$ pro Richtung. Damit ist das Verfahren für die praktische Anwendung nahezu unbrauchbar.

Das Problem der unbekannten Abstände zwischen den Gesichtsmerkmalen lösen A. Gee und R. Cipolla in [10] durch einen Ansatz, der fünf markante Punkte im Gesicht verwendet. Unter der Annahme einer *schwachen Perspektive (weak perspective)*, d.h. daß die Tiefenunterschiede im Gesicht, verglichen mit dem Abstand zur Kamera, klein sind, gelingt es den Autoren, ein Gesichtsmodell zu entwickeln, welches für alle Gesichter als wahr angenommen wird. Damit kommen Gee und Cipolla bei der Posenbestimmung ohne die Kenntnis personenspezifischer Abstände aus. Die Autoren gehen von einem symmetrischen Gesicht aus, in dem die Nasenspitze auf der Symmetrieachse liegt. Ferner werden zwei Längenverhältnisse im Gesicht als für alle Gesichter konstant angenommen[3]. Es wird von einer Sichtbarkeit der Augen- und Mundwinkel ausgegangen, die nach Ansicht der Autoren über einen weiten Bereich gegeben ist. Die Arbeit von Gee und Cipolla hat ihren Schwerpunkt auf den geometrischen Überlegungen und nicht auf der tatsächlichen Lokalisierung der Merkmale. Diese wurde nur am Rande unter Benutzung eines einfachen, von Hand initialisierten Tracking-Verfahrens durchgeführt. Dabei wurden die Pupillen und die Lippenlinie als dunkelste Regionen innerhalb einer kleinen Umgebung verfolgt. Aufgrund dieser Vereinfachungen betrug die erreichte Geschwindigkeit etwa 100 Hz. Der Fehler lag bei $3° - 6°$.

Wie bereits erwähnt, ist die Bestimmung der Gesichtsnormalen nur der erste Schritt zur Bestimmung der Blickrichtung. Der zweite Schritt, nämlich die Ermittlung der Blickrichtung der Augen, wurde von H. Klingspohr, T. Block und R.R. Grigat in [12] untersucht. Dazu wurde der Kopf einer Versuchsperson im Abstand von 50 cm vor einem Monitor fixiert, so daß Lage und Entfernung zu einer, unter dem Monitor montierten, Kamera konstant waren. Nach einer manuellen Vorgabe wurden die Augen anhand des hohen Kontrastes zwischen Iris und Augapfel detektiert. Mit Hilfe der Hough-Transformation wurde der Umriß der Iris extrahiert und eine Ellipse darauf angepaßt. Anhand der Verformung der Ellipse konnte auf die Blickrichtung der Augen geschlossen werden. Als Praxistest wurde ein Cursor mit den Augen über den Bildschirm bewegt. Die dazu erforderliche Kalibrierung bestand darin, daß die Versuchsperson nacheinander in die vier Monitorecken schauen mußte. Die Zykluszeit betrug eine Sekunde, die erreichte Genauigkeit wird mit $2.3°$ angegeben. Die Robustheit des Verfahrens hängt stark von der korrekten Segmentierung der Iris ab. Auch in diesem Versuchsaufbau ist das Tragen einer Brille nicht zulässig.

[3] Es handelt sich dabei um das Verhältnis des Abstandes von Mundlinie und Nasenbasis L_m zu Mundlinie und Augenlinie L_f sowie das Verhältnis der Verbindungslinie von Mundlinie und Augenlinie L_f zu der senkrecht dazu stehenden Verbindung zu Nasenspitze L_n. Es wird $L_m : L_f = 0.4$ und $L_n : L_f = 0.6$ angenommen.

In [4] stellen S. Baluja und D. Pomerleau ein Blickverfolgungssystem vor, welches ein Grauwertbild des Auges als Eingabe für ein künstliches neuronales Netz verwendet. Das rechte Auge wird dabei anhand des Glanzlichts gesucht. Die Blickrichtung des Auges wird mit einer Geschwindigkeit von 15 Hz und einer Genauigkeit von etwa $1.7°$ geschätzt. Die Autoren weisen explizit darauf hin, daß bei ihrem Verfahren ein Fixieren des Kopfes nicht notwendig sei. Die vorgestellte Implementierung des Verfahrens deutet darauf hin, daß die Autoren davon ausgehen, daß das Gesicht auf den Bildschirm gerichtet bleibt, d.h. das Auge vollständig sichtbar ist. Für das Training des neuronalen Netzes werden 2000 Sätze von Trainingsdaten verwendet, die durch Verfolgung des Cursors auf dem Bildschirm gesammelt werden. Das Training ist personenspezifisch, d.h. es muß für jeden neuen Benutzer wiederholt werden.

In [23] stellen A.C. Varchmin, R. Rae und H. Ritter einen merkmalsbasierten Ansatz vor, bei dem das Gesicht in einem ersten Schritt durch ein Farbhistogramm detektiert wird. Innerhalb der Gesichtsregion werden dann die Augen, die Nasenlöcher und der Mund durch Farbverarbeitung und Verwendung von “eigeneyes” und “eigennoses” gesucht. Die Mundwinkel werden anhand ihrer Form erkannt. Die gefundenen Merkmale werden einem Verifikationsschritt unterzogen und als Eingabevektoren für “Local Linear Map”-Netzwerke (LLM-Netze) verwendet. Die Netzausgaben werden von einem weiteren LLM-Netz zur Gesichtspose kombiniert. Die Kamera war in den Experimenten unter dem Monitor plaziert, so daß Augen und Nasenlöcher immer sichtbar waren. Ferner wurde eine Lampe zur Erzeugung von Glanzlichtern auf den Augen verwendet. Das Training für die Bestimmung der Blickrichtung war sowohl vom Benutzer als auch von der Beleuchtung abhängig. Ein Trainingssatz umfaßte 625 Bilder. Der Fehler lag bei $1.5°$ horizontal und $2.5°$ vertikal. Die Zykluszeit betrug eine Sekunde.

S. McKennan, S. Gong und J.J. Collins verfolgen in [17] einen erscheinungsbasierten Ansatz. Dabei verwenden sie eine Bewegungsdetektion zusammen mit Gaborwavelets, um Sequenzen von normalisierten Gesichtern aus einer Bildsequenz mit komplexem Hintergrund zu extrahieren. Das Ziel der Arbeit war es, aus einer Videosequenz, wie sie z.B. von einer Überwachungskamera aufgenommen wird, die Gesichter für eine automatische Gesichtserkennung aufzubereiten. Dabei können auch mehrere Gesichter gleichzeitig extrahiert werden. Ferner wurde die Rotation der Gesichter um die Hochachse mit Hilfe eines neuronalen Netzes ermittelt.

In der Diplomarbeit von E. Ábráham-Mumm [1] ist es gelungen, die Gesichtspose eines Puppenkopfes mit Hilfe eines künstlichen neuronalen Netzes auf $0.5°$ genau zu bestimmen. Dazu wurde der Puppenkopf im Kamerabild extrahiert und mit einem äquidistanten Raster überzogen. An den Kreuzungspunkten des Rasters wurden je vier Gaborfilter mit den Orientierungen $0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}$ plaziert. Die Verteilung der Filter erfolgte äquidistant auf der detektierten Gesichtsfläche. Die Antworten dieser Filter wurden als Eingabe für ein künstliches neuronales Netz verwendet. Die Gesichtsfläche wurde in jedem Bild durch Segmentierung mit einen fest vorgegebenen Farbschwellwert detektiert, was sich in sehr hohen Anforderungen an die Beleuchtung der Szene äußerte.

Abschließend bleiben zwei rein farbbasierte Arbeiten zu nennen: Ein auf Geschwindigkeit optimiertes Verfahren zur Gesichtsverfolgung wurde von G.R. Bradski in [5] vorgestellt. Zugunsten der Echtzeitfähigkeit wurde auf eine Bestimmung der Gesichtspose verzichtet, dafür wurde die Translation, d.h. die x-, y- und z-Koordinaten des Betrachters relativ zur Kamera sowie die Neigung des Kopfes ("roll") bestimmt. Der Benutzer durfte sich frei vor der Kamera bewegen, ein Training auf eine bestimmte Person war nicht notwendig. Die Praxistauglichkeit des Verfahrens wurde anhand von multimedialen Anwendungen demonstriert[4]. Diese Art der Steuerung ist allerdings körperlich belastend und wenig intuitiv. Die Zykluszeit betrug weniger als 1/30 Sekunde, d.h. die Daten konnten in Echtzeit extrahiert werden. Dabei betrug die Standardabweichung des Rollwinkels $2.4°$. Die Position des Benutzers konnte mit einer Standardabweichung von 0.27 cm (x), 0.58 cm (y) und 3.4 cm (z) bestimmt werden.

Q. Chen et al. ist es in [9] durch einen rein farbbasierten Ansatz gelungen, die Gesichtspose zu approximieren. Dabei wurde angenommen, daß sich ein menschlicher Kopf in eine Haar- und eine Gesichtsregion einteilen läßt. Anhand der Lage dieser Regionen zueinander gelingt es den Autoren, Rückschlüsse auf die Kopfpose zu ziehen. Eine große Einschränkung des Verfahrens ist, daß das Vorhandensein von Haaren vorausgesetzt wird. Existiert keine Haarregion, so wird das Verfahren singulär. Ebenso wird die Annahme einer "normalen" Kurzhaarfrisur gemacht. Insgesamt ist das Verfahren zwar einerseits durch die ausschließliche Verwendung von Farbinformationen sehr robust, aber andererseits nur für eine sehr beschränkte Klasse von Personen anwendbar. Die erreichte Genauigkeit lag bei $6.8°$ horizontaler Drehung, $5.7°$ vertikaler Neigung und $2.9°$ beim Rollwinkel. Diese Ergebnisse wurden unter Verwendung von Modellen ermittelt, die durch einen 3D-Scanner von den Testpersonen erstellt wurden.

Vor- und Nachteile der Verfahren

Aus den unterschiedlichen Ansätzen ergeben sich typische Eigenschaften. Alle geometrischen Ansätze arbeiten merkmalsbasiert, sind also auf die exakte Lokalisierung spezieller Gesichtsmerkmale (z.B. Augenwinkel, Pupille, Nasenspitze, Mundwinkel) angewiesen. Schon eine geringe Fehldetektion spiegelt sich in einem großen Fehler bei der Posenbestimmung wider. Ein Vorteil der geometrischen Verfahren ist die effiziente Berechnung der Pose, *nachdem* die Merkmale detektiert worden sind. Viele Verfahren erreichen dadurch Geschwindigkeiten von über 50 Hz.
Ferner ist eine Konsequenz aus dem vielen Arbeiten zugrunde liegenden geometrischen Modell, daß aufgrund der Verwendung des *Arcus Tangens* der Fehler gerade bei Frontalansichten maximal wird. Damit sind diese Ansätze für Arbeiten am Bildschirm nur sehr eingeschränkt nutzbar, da dort gerade der Bereich um die Frontalansicht von Interesse ist.

[4]Bradski et al. demonstrierten die Steuerung eines virtuellen Flugs und die Hindernisvermeidung in einem virtuellen Gebäude.

Die merkmalsbasierten Ansätze, welche die Pose bzw. Blickrichtung durch Verwendung künstlicher neuronaler Netze bestimmen, sind zwar wesentlich langsamer als die geometrischen Verfahren, dafür sind sie aber erheblich genauer (etwa $2°$). Durch Verwendung von Hierarchien neuronaler Netze kann die Stabilität auf Kosten der Geschwindigkeit erhöht werden. Beispielsweise verwenden Varchmin, Rae und Ritter in [23] zehn neuronale Netze, um Merkmale zu extrahieren und auf Plausibilität zu prüfen.

Der größte Vorteil der erscheinungsbasierten Ansätze ist, daß sie nicht auf die Detektion einzelner Merkmale angewiesen sind. Dadurch sind sie im allgemeinen stabiler als merkmalsbasierte Ansätze.
Bei den erscheinungsbasierten Verfahren ergibt sich ein uneinheitliches Bild: Die rein farbbasierten Ansätze [9; 5] sind aufgrund ihrer geringen Genauigkeit und spezieller Annahmen (z.B. über die Frisur bei [9]) für praktische Anwendungen unbrauchbar, dafür aber sehr schnell.
Andere Verfahren, wie z.B. [1], sind langsamer und erfordern das Training neuronaler Netze, erreichen dafür aber eine hohe Genauigkeit (etwa $0.5°$).

Die Ergebnisse der wichtigsten Arbeiten sind in Tabelle 1 zusammengefaßt.

	mittlerer Fehler	Zeitbedarf	Verfahren
Ballard & Stockman	3°	80s	geometrisch, Ellipsen
Ballard & Stockman	5°–10°/Richtung	3s	geometrisch, Twinkle
Gee & Cipolla	3°–6°	100Hz (Tracker)	geometrisch, konstante Längenverhältnisse
Chen et al.	7°/Richtung	–	farbbasiertes System, Segmentierung Gesicht/Haare
Ábráham–Mumm	0.5°	–	Gaborfilter, NN
Klingspohr, Block & Grigat	2.3°	1Hz	Kopf fixiert, Twinkle
Baluja & Pomerleau	1.7°	15Hz	Twinkle, NN
Varchmin, Rae & Ritter	1.5° Pan, 2.5° Tilt	1Hz	Eigeneyes, Eigennoses, Twinkle, ~10 NNs
Bradski	–	> 30Hz	farbbasiertes System, nur x–,y–,z–Position, Rollwinkel

Legende:
Bestimmung der Gesichtspose
Bestimmung der Blickrichtung
Bestimmung von Position und Rollwinkel

Tabelle 1: *Überblick über Arbeiten zur Bestimmung von Gesichtspose und Blickrichtung.*

1.3 Zielstellung der Arbeit

In dieser Arbeit soll ein erscheinungsbasierter Ansatz zur Bestimmung der Gesichtspose präsentiert werden, der die Robustheit der farbbasierten Ansätze mit einer gegenüber [1] noch verbesserten Genauigkeit kombiniert.
Der Schwerpunkt in [1] lag auf dem Vergleich unterschiedlicher Typen künstlicher neuronaler Netze und weniger auf der Entwicklung einer optimalen Filterbank. Es liegt die Vermutung nahe, daß mit einer geschickteren Plazierung der Filter ein noch geringerer Fehler zu erwarten ist.
Für diesen Zweck bietet sich die Verwendung von *Waveletnetzen* an. Bei einem Waveletnetz handelt es sich im wesentlichen um eine Linearkombination von zweidimensionalen Waveletfunktionen, die jeweils die Freiheitsgrade Translation, Dilatation und Rotation haben. Durch geeignete Parametrisierung kann ein Waveletnetz ähnlich einem RBF-Netz zur Approximation beliebiger Funktionen verwendet werden.
Dabei ist die Wahl der Waveletfunktion von großer Bedeutung für die Güte der Approximation.
Dazu werden in dieser Arbeit *ungerade Gaborwavelets*[5] verwendet. Insbesondere sind Gaborfilter geeignet, lokalspektrale Signalkomponenten zu extrahieren (vgl. [21]).
In [14] werden Gaborwaveletnetze erfolgreich zur ausdrucksunabhängigen Identifikation von Gesichtern eingesetzt. Ein Gaborwaveletnetz wird dabei zur Modellierung eines Grundgesichts (*"Template"*) verwendet. Dieses Template wird durch einen Optimierungsschritt auf anderen Gesichtern repositioniert und in Größe und Orientierung angepaßt. Anhand der Ähnlichkeit zwischen dem reparametrisierten Template und dem untersuchten Grauwertbild konnte nahezu eindeutig auf die Identität der Testperson geschlossen werden.
Die Möglichkeit der subpixelgenauen Reparametrisierung des Waveletnetz-Templates schafft die Voraussetzung dafür, Merkmale an einer festen Gesichtsposition, etwa der Nase, zu extrahieren. Damit ist ein Weg gefunden, um Filter unabhängig von der Gesichtspose an Gesichtsmerkmalen zu plazieren.
Diese Arbeit wird zeigen, daß mit diesem *erscheinungsbasierten Ansatz* tatsächlich eine deutliche Verbesserung der Genauigkeit, verbunden mit einer höheren Robustheit im Vergleich zu [1], erzielt werden kann. In [15] wurden unter anderem erste Ergebnisse dieser Diplomarbeit präsentiert.

1.4 Aufbau der Arbeit

Die folgende Arbeit ist in fünf Abschnitte unterteilt. In Kapitel 2 werden die *Gaborwaveletnetze* eingeführt, welche in dieser Arbeit zur Gesichtsrepräsentation eingesetzt werden. Anschließend wird das Matching eines Gaborwaveletnetzes auf anderen Ansichten des repräsentierten Gesichtes unter Verwendung des *Gaborsuperwavelets* realisiert. Mit der *progressiven Attention* wird ein Konzept vorgestellt, welches einen op-

[5] Mit "ungeraden Gaborwavelets" bezeichnen wir Waveletfunktionen, deren Mutterwavelet eine ungerade Gaborfunktion ist.

timalen Kompromiß zwischen Geschwindigkeit und Genauigkeit bei Echtzeitanwendungen ermöglicht.

Da für eine Posenbestimmung die Position des Gesichtes bekannt sein muß, wird in Kapitel 3 ein Algorithmus zur Gesichtsverfolgung in monokularen, unkalibrierten Bildfolgen präsentiert. Dabei wird über mehrere Stufen die Genauigkeit sukzessive verbessert, bis durch Matching des Gaborsuperwavelets schließlich eine Genauigkeit im Subpixelbereich erreicht wird.

Damit sind die Grundlagen für die Bestimmung der Gesichtspose gelegt. In Kapitel 4 wird die Schätzung der Gesichtspose mit künstlichen neuronalen Netzen betrachtet. Zur Merkmalsextraktion werden Gaborwaveletnetze verwendet. Es werden unterschiedliche Merkmalsvektoren auf ihre Eignung zur Posenbestimmung untersucht. Abschließend werden die verwendeten künstlichen neuronalen Netze vorgestellt.

In Kapitel 5 werden der Aufbau und Ablauf sowie die Ergebnisse der Experimente vorgestellt. In der ersten Hälfte des Kapitels wird die prinzipielle Eignung der Gaborwaveletnetze zur Posenbestimmung untersucht, während im zweiten Teil des Kapitels das Generalisierungsvermögen und die erreichbare Genauigkeit bei Verwendung progressiver Attention untersucht werden. Die Ergebnisse der Posenschätzung mit den vorgeschlagenen Verfahren werden ausführlich vorgestellt und diskutiert.

In Kapitel 6 werden die Ergebnisse der Arbeit zusammengefaßt und mit denen anderer Arbeiten, insbesondere denen aus der Arbeit von E. Ábráham-Mumm, verglichen. Ein besonderes Augenmerk liegt ferner auf Betrachtungen zur Übertragung des Verfahrens auf praktische Anwendungen. Abschließend werden Schwachpunkte und zukünftige Erweiterungsmöglichkeiten diskutiert.

2 Gaborwaveletnetze zur Gesichtsrepräsentation

Um Gesichter maschinell verarbeiten zu können, wird eine geeignete Gesichtsrepräsentation benötigt. Dazu sind Repräsentationen gesucht, welche die Eingangsdaten möglichst sparsam kodieren, gleichzeitig aber die Merkmale erhalten.
Nicht jede Repräsentation eignet sich gleich gut für alle Anwendungen. So ist für eine einfache Gesichtsverfolgung eine Repräsentation durch die Gesichtsfarbe hinreichend [5; 13], während zur Identifikation einer Person ein genauere Repräsentation nötig ist, etwa durch Bunch-Graphen [24] oder durch Waveletnetze [11; 14].
In [1] ist es gelungen, die Gesichtspose eines Puppenkopfes mit einer hohen Genauigkeit zu schätzen. Dabei wurde die Gesichtsregion mit 64 Gaborfilterpaaren abgetastet, die in einem gleichabständigen 4×4-Raster in jeweils 4 Orientierungen ($0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4}$) plaziert wurden. Da die Gesichtsregion durch Farbsegmentierung detektiert wurde, haben die Filterantworten keinen direkten Bezug zu Gesichtsmerkmalen. Es liegt die Vermutung nahe, daß sich die Genauigkeit der Posenschätzung durch eine geschicktere Plazierung der Filter verbessern läßt.
Zu diesem Zweck bieten sich die von Zhang und Beneviste in [25] eingeführten Waveletnetze an. Bei den Waveletnetzen handelt es sich im wesentlichen um RBF-Netze, bei denen Waveletfunktionen anstelle von radialen Basisfunktionen verwendet werden. Ein Vorteil der Verwendung von Waveletnetzen gegenüber der starren Verteilung der Filter wie in [1] liegt darin, daß die Wavelets durch ein Optimierungsverfahren an Merkmalen plaziert werden können. Dadurch entsteht ein direkter Zusammenhang zwischen den Wavelets und dem zugrunde liegenden Grauwertbild. Es sollte darauf hingewiesen werden, daß *keine Identifikation der Merkmale* vorgenommen werden muß. Die Gesichtsmerkmale werden nur als Landmarken benutzt.
Im Hinblick auf praktische Anwendungen ist ein echtzeitfähiges System erstrebenswert. Durch ihre Struktur, als Linearkombination von Wavelets, sind Waveletnetze für *progressive Attention*, d.h. für eine aufgabenorientiert steuerbare Genauigkeit, verwendbar.

2.1 Wavelets

In den letzten Jahren sind Wavelets in vielen Anwendungsgebieten zu einem verbreiteten Werkzeug geworden. Wavelets werden beispielsweise in der Medizin zur Analyse von EKG-Daten verwendet [16] oder in der Qualitätskontrolle eingesetzt. Ein zunehmend wichtigeres Anwendungsgebiet ist die Kompression von Bildern und Bildfolgen. Die Kompression von Grauwertbildern unter Verwendung von Waveletnetzen wird z.B. in [19] untersucht.

2.1.1 Was sind Wavelets?

Allgemein bezeichnet man mit dem Begriff *Wavelet* solche Funktionen, die durch Translation und Dilatation aus einer Grundfunktion, dem sogenannten *Mutterwavelet*,

hervorgehen. Das Mutterwavelet (und damit auch seine verschobenen und skalierten "Kinder") muß dabei gewisse Anforderungen erfüllen:
Eine Funktion $\psi \in \mathbb{L}^2(\mathbb{R}^2)$ heißt ψ *Wavelet*, falls ein $C_\psi \in \mathbb{R}$ existiert, so daß ψ die folgende *Zulässigkeitsbedingung* erfüllt:

$$0 < C_\psi := 4\pi^2 \int\limits_{\mathbb{R}^2} \frac{\|\hat{\psi}(\omega)\|}{\|\omega\|} d\omega < \infty \tag{1}$$

Jedes Wavelet kann wieder als Mutterwavelet verwendet werden. Der Name Wavelet beschreibt die Eigenschaft, daß der Funktionswert um die x-Achse pendelt, der Mittelwert eines Wavelets ψ ist Null[6]:

$$0 = \hat{\psi}(0) = \frac{1}{\sqrt{2\pi}} \int\limits_{\mathbb{R}} \psi(t) dt\,. \tag{2}$$

Verbreitete Waveletfunktionen sind das Haarwavelet, welches aufgrund seiner Einfachheit einen großen theoretischen Wert hat, das Meyerwavelet und die in dieser Arbeit verwendeten *Gaborwavelets*, welche für ihre günstigen Eigenschaften als Kantendetektoren bekannt sind.

2.1.2 Gaborwavelets

In dieser Arbeit wird der ungerade Imaginärteil der Gaborfunktion als Mutterwavelet verwendet:

$$\psi(x, y) = \mathrm{e}^{-\frac{1}{2}(x^2+y^2)} \sin(x) \tag{3}$$

Die ungerade Gaborfunktion ist in Abbildung 2 dargestellt.

Aus Gleichung (3) werden Wavelets durch Translation c_x, c_y, Dilatation s_x, s_y und Rotation α gewonnen:

$$\psi_{\mathbf{n}}(\mathbf{x}) = \psi\left(\mathbf{SR}\left(\mathbf{x} - \mathbf{t}\right)\right) \tag{4}$$

mit

$$\mathbf{S} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} c_x \\ c_y \end{pmatrix}.$$

Die Freiheitsgrade Translation, Dilatation und Rotation werden im Parametervektor $\mathbf{n} = (c_x, c_y, s_x, s_y, \alpha)^T$ zusammengefaßt.

[6] Der Beweis folgt aus der Zulässigkeitsbedingung (1) durch den Satz von Riemann-Lebesgue, siehe [16].

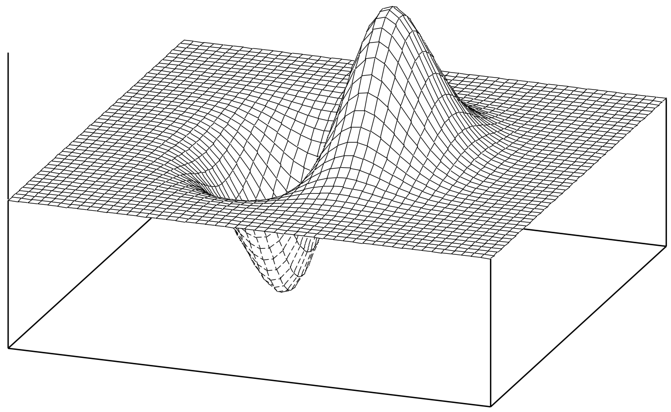


Abbildung 2: *Die ungerade Gaborfunktion* $\psi(x,y) = \mathrm{e}^{-\frac{1}{2}(x^2+y^2)}\sin(x)$.

2.2 Waveletnetze

Die durch die Wavelets kodierten *lokalen Merkmale* werden durch Linearkombination in einem Waveletnetz zu einer *globalen Objektrepräsentation* zusammengeführt, ähnlich wie in einem RBF-Netz.
Jede *Repräsentation* impliziert auch eine *Rekonstruktion*. Die Waveletnetze rekonstruieren ein zugrundeliegendes Grauwertbild. Dabei hängt die Qualität der Rekonstruktion von der Anzahl der verwendeten Wavelets ab.
Sei $\mathbf{g} : [0,\ldots,K-1] \times [0,\ldots,L-1] \longrightarrow [0,\ldots,255]$ ein Grauwertbild der Größe $K \times L$ Pixel mit dem Mittelwert $\overline{\mathbf{g}}$. Durch ein Optimierungsverfahren sollen N Wavelets so parametrisiert werden, daß ihre Linearkombination g optimal approximiert:

$$\mathbf{g}(x,y) \approx \sum_{i=1}^{N} \omega_i \psi_{\mathbf{n}_i}(x,y) + \overline{\mathbf{g}}\,. \tag{5}$$

Die Struktur eines Waveletnetzes ist in Abbildung 3 schematisch dargestellt.

Definition

Sei $\mathbf{\Psi} = \{\psi_{\mathbf{n_1}},\ldots,\psi_{\mathbf{n_N}}\}$ eine Familie von Wavelets. Sei ferner g ein Grauwertbild mit dem Mittelwert $\overline{\mathbf{g}}$.
Wähle nun $\omega_i \in \mathbb{R}$ und $\mathbf{n}_i = (c_{x_i}, c_{y_i}, s_{x_i}, s_{y_i}, \alpha_i)^T \in \mathbb{R}^5$ für $i = 1,\ldots,N$ derart, daß

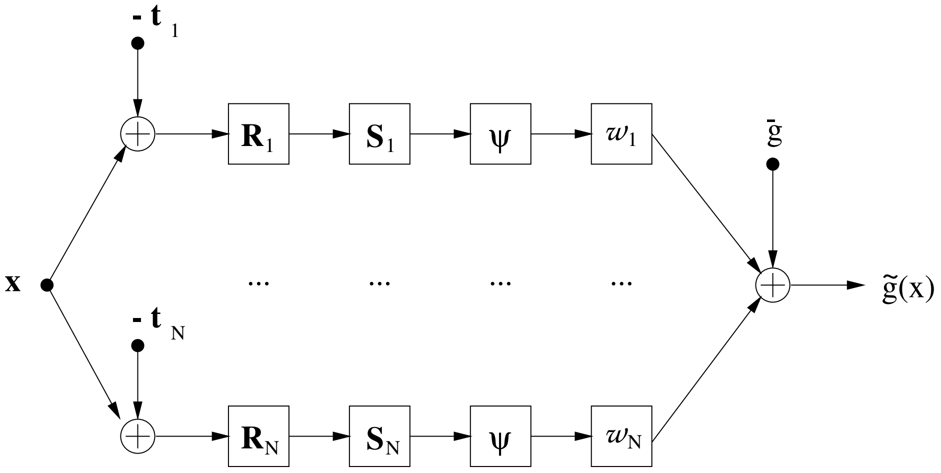


Abbildung 3: *Schematischer Aufbau eines Waveletnetzes: Die Approximation des Funktionswertes* $\mathbf{g}(\mathbf{x})$ *ergibt sich aus Linearkombination der Wavelets* $\psi_{\mathbf{n}_i}$ *und dem Mittelwert* $\overline{\mathbf{g}}$ *von* $\mathbf{g}$.

die Energiefunktion

$$E = \|I - \sum_i \omega_i \psi_{\mathbf{n_i}} - \overline{\mathbf{g}}\|_2^2 \tag{6}$$

auf dem Auswertungsbereich minimal wird.

Durch

$$\begin{array}{rcl} \mathbf{\Psi} & = & (\psi_{\mathbf{n_1}}, \ldots, \psi_{\mathbf{n_N}})^T \quad \text{und} \\ \mathbf{w} & = & (\omega_1, \ldots, \omega_N)^T \end{array}$$

wird das *Waveletnetz* $(\mathbf{\Psi}, \mathbf{w})$ zum Grauwertbild $\mathbf{g}$ definiert.

Falls $\mathbf{\Psi}$ eine Familie von ungeraden Gaborwavelet der in Gleichung (4) angegebenen Form ist, so sprechen wir von einem *Gaborwaveletnetz*. Der Begriff "Gaborwaveletnetz" wird im folgenden mit *GWN* abgekürzt.

Zur Lösung der Minimierungsaufgabe (6) wird das Levenberg-Marquardt-Verfahren verwendet, welches eine effiziente Kombination aus Gradientenabstiegsverfahren und Newtonverfahren darstellt [20].
Es wird darauf hingewiesen, daß die Parameter kontinuierlich gewählt werden können, d.h. $\mathbf{n} = (c_x, c_y, s_x, s_y, \alpha)^T \in \mathbb{R}^5$. Die Wavelets können dadurch subpixelgenau an die örtliche Bildstruktur angepaßt werden[7].

[7] Bei der Wahl des Parametervektors $\mathbf{n}$ als Element des $\mathbb{R}^5$ wurde bereits an die hier verwendeten Gaborwavelets gedacht, die als nicht rotationssymmetrische Wavelets über 5 Freiheitsgrade verfügen. Bei der Verwendung rotationssymmetrischer Wavelets reduziert sich $\mathbf{n}$ auf 4 Parameter.

2.3 Gesichtsrepräsentation

Da wir bei der Posenbestimmung nur an der Gesichtsregion interessiert sind, muß in dieser Anwendung nicht das gesamte Bild, sondern nur ein Ausschnitt repräsentiert werden. Der einzige Unterschied zu dem oben vorgestellten globalen Vorgehen besteht darin, daß die Energiefunktion (6) nur auf der Gesichtsregion minimiert wird. Die Gesichtsregion wird in dieser Arbeit durch einen rechteckigen Bildausschnitt festgelegt. In praktischen Anwendungen (siehe Abschnitt 3) läßt sich diese Region z.B. durch die Hautfarbe identifizieren.

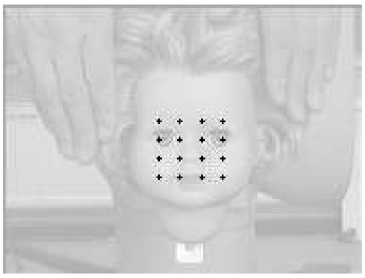
2.3.1 Grob-Fein-Strategie

Die in der Bildverarbeitung verbreitete Verwendung von Auflösungshierarchien läßt sich auf Waveletnetze übertragen. Dazu werden in jeder Stufe die Anzahl der Wavelets horizontal und vertikal verdoppelt sowie deren Skala halbiert. In [19; 11] wurden komplette Auflösungspyramiden untersucht.
Für die spezielle Anwendung der Gesichtsrepräsentation hat sich eine zweistufige Pyramide bewährt. Der Grund für diese Wahl liegt in dem beschränkten Skalenbereich, in dem sich die relevanten Gesichtsmerkmale befinden. Zudem wird die Vorinitialisierung der Skala der zweiten Netzebene nicht halbiert, sondern nur um den Faktor 1.5 reduziert.
Die Entscheidung, trotzdem zwei Stufen zu verwenden, hat den Grund, daß die grobere Ebene sehr bequem für die Vorpositionierung beim Matching (siehe Abschnitt 2.4.1) und für Echtzeit-Gesichtsverfolgung (siehe Abschnitt 3) verwendet werden kann. Die zweite Ebene kann bei Bedarf zur Erhöhung der Genauigkeit herangezogen werden. In Abschnitt 2.6 wird mit der *progressiven Attention* eine Methode vorgestellt, die eine feiner abgestufte Adaption der Genauigkeit ermöglicht.

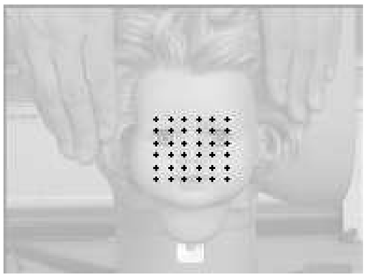
2.3.2 Initialisierung

Zur Gesichtsrepräsentation wird in dieser Arbeit eine grob-fein-Strategie (*coarse to fine*) verwendet. Auf eine gegebene Normalansicht eines Gesichtes wird ein aus zwei Ebenen bestehendes Waveletnetz antrainiert. Zur Initialisierung werden die Wavelets der ersten Ebene in einem gleichabständigen 4×4-Raster auf der inneren Gesichtsregion verteilt. Diese Ebene dient zur groben Repräsentation der Gesichtsmerkmale, Zusätzlich zu der groben Ebene wird noch eine zweite, feinere Ebene hinzugefügt, um die Repräsentation weiter zu verfeinern. Die zweite Ebene besteht aus 36 Wavelets, die zur Initialisierung in einen ebenfalls gleichabständigen 6×6-Raster verteilt werden. Die initiale Verteilung der Wavelets auf die beiden Ebenen ist in Abbildung 4 dargestellt.

Bei der Initialisierung werden die Dilatationsparameter der Wavelets so gewählt, daß die Amplitude einer einhüllenden Gaußfunktion beim halben Abstand zwischen den



(a)



(b)

Abbildung 4: *Darstellung der Waveletpositionen bei der Initialisierung der beiden Ebenen des Gaborwaveletnetzes. In (a) ist die grobe 4×4-Ebene und in (b) die feinere 6×6-Ebene dargestellt.*

Wavelets auf die Hälfte abgefallen ist. Die anfängliche Orientierung eines jeden Wavelets wird so gewählt, daß der Betrag der Projektion $< \psi_{\mathbf{n}_i}, \mathbf{g} >$ maximal wird. In [11] hat sich diese Heuristik als vorteilhaft für die Minimierung der Energiefunktion (6) herausgestellt.
Die Startpositionierung der Wavelets in einem rechteckigen Gitter stellt noch kein Optimum dar. In [11] wird die Positionierung in einer ovalen Region über dem inneren Gesicht vorgeschlagen.

2.3.3 Optimierung

Die Optimierung der Waveletparameter der beiden Ebenen wird in zwei getrennten Schritten durchgeführt. Im ersten Schritt werden die 16 Wavelets der groben Ebene gemäß Gleichung (6) reparametrisiert. Dazu wird der Levenberg-Marquardt-Algorithmus verwendet.

Der Optimierungsvorgang wird in Abbildung 5 an einem Beispiel verdeutlicht. In (a) sind die Positionen der optimal positionierten Wavelets der ersten Netzebene dargestellt, darunter in (b) das Rekonstruktionsergebnis.
In (c) sind die Positionen der Wavelets der zweiten Ebene eingezeichnet, in (d) ist der Rekonstruktionsbeitrag der zweiten Ebene dargestellt.
Die Waveletpositionen des gesamten Netzes sind in (e) dargestellt, (f) zeigt schließlich die Rekonstruktion von $\mathbf{g}$ aus der Summe der beiden Netzebenen.
Die Markierungen in den Abbildungen (a), (c), (e) zeigen nur die Position der Wavelets an und haben keinen Zusammenhang mit deren Ausdehnung.
Ein Großteil der Wavelets wurde an markanten Positionen im Gesicht plaziert.
Es soll festgehalten werden, daß die hier vorgestellten Verfahren zur Initialisierung eines Waveletnetzes und dessen Optimierung bis auf die Markierung der Gesichtsregion ohne manuelle Eingriffe auskommen. Damit ist mit geringem Aufwand eine Automatisierung des gesamten Vorganges möglich. Das Auffinden der Gesichtsregion kann z.B. wie in Abschnitt 3 anhand der Hautfarbe geschehen.

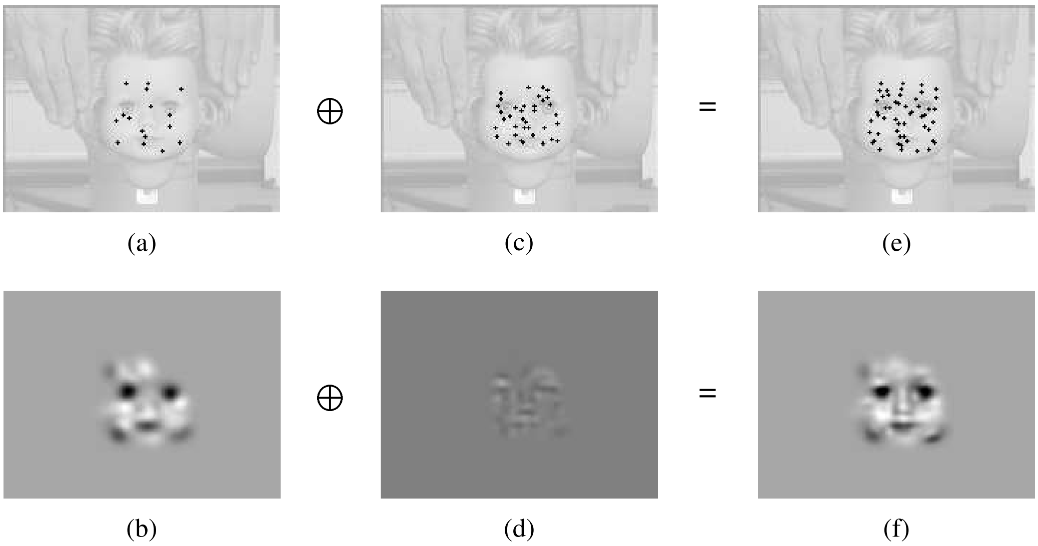


Abbildung 5: *Die obere Zeile zeigt die Waveletpositionen nach der Reparametrisierung der beiden Ebenen des Gaborwaveletnetzes. In (a) sind die Positionen der ersten 16 Wavelets dargestellt, (c) zeigt die Waveletpositionen der zweiten, aus 36 Wavelets bestehenden, Netzebene, (e) die Waveletpositionen des gesamten Netzes. In der unteren Zeile ist die Rekonstruktion aus den im Bild darüber markierten Wavelets dargestellt. (b) zeigt die Rekonstruktion mit der ersten Netzebene, (d) zeigt den Beitrag der zweiten Ebene. In (f) ist die Gesamtrekonstruktion als Summe von (b) und (d) dargestellt.*

2.4 Exakte Gesichtslokalisierung durch Reparametrisierung

Die oben vorgestellte Repräsentation von $\mathbf{g}$ durch das Gaborwaveletnetz $(\mathbf{\Psi}, \mathbf{w})$ ist spezifisch für ein Bild $\mathbf{g}_{\mathrm{train}}$. Für die praktische Anwendung ist eine Anpassung des Waveletnetzes $(\mathbf{\Psi}, \mathbf{w})$ auf beliebige Ansichten des repräsentierten Gesichtes notwendig, da die Filterantworten des Waveletnetzes nur Sinn machen, wenn die Wavelets wieder auf den von ihnen repräsentierten Merkmalen zu liegen kommen.
Dieses Problem stellt sich zum Beispiel bei der Gesichtsverfolgung des auf den Bildschirm blickenden Benutzers. Der Benutzer schaut dabei in der Regel aus einem Abstand von 50-60 cm frontal auf den Bildschirm. Um auf den linken bzw. rechten Rand des Bildschirms zu sehen, muß der Kopf um etwa $\pm 20^\circ$ horizontal gedreht werden. Um den oberen bzw. unteren Rand zu fixieren, muß der Kopf um etwa $\pm 15^\circ$ vertikal geneigt werden. All diesen Gesichtsposen ist gemein, daß die Gesichtsmerkmale wie Augen, Nase und Mund vollständig sichtbar bleiben.
Es stellt sich die Aufgabe, das GWN $(\mathbf{\Psi}, \mathbf{w})$ affin so zu deformieren, daß die Wavelets wieder "an ihren Platz" kommen.
Unter der Annahme einer schwachen Perspektive ("weak perspective") kann das Gesicht, bei ausreichend großem Abstand zur Kamera, als eine Ebene im Raum betrachtet

werden. Zur Beschreibung der Projektion der Ebene im Kamerabild benötigt man eine Translation c_x, c_y, eine Dilatation s_x, s_y und eine Scherung s_{xy} sowie eine Rotation α. Auf die Verwendung der Scherung s_{xy} wird in dieser Arbeit aus Stabilitätsgründen verzichtet.
Es ist unser Ziel, einen Parametervektor $\mathbf{n} = (c_x, c_y, s_x, s_y, \alpha)$ so zu bestimmen, daß die Wavelets des GWN in Bild $\mathbf{I}_t$ wieder auf den Gesichtsmerkmalen zu liegen kommen, auf denen sie im Trainingsbild $\mathbf{g}_{\text{train}}$ plaziert worden waren.
Wir setzen in dieser Arbeit ein aktives System voraus, d.h. wir können davon ausgehen, daß sich die Gesichtsposition und Orientierung zwischen zwei zeitlich eng aufeinanderfolgenden Bildern nur geringfügig ändert. Das bedeutet, daß die ungefähre Position, Skala und Orientierung des Gesichtes in Bild $\mathbf{I}_t$ aus dem vorhergehenden Bild $\mathbf{I}_{t-1}$ bekannt sind. Es wird sich zeigen, daß dieses Wissen genügt, um das Gaborwaveletnetz erfolgreich neu zu positionieren. Diese Reparametrisierung kann sehr elegant durch die Konstruktion eines *Gaborsuperwavelets* vorgenommen werden.

2.4.1 Gaborsuperwavelet

Aufgrund der Linearität der Summe ist eine endliche Linearkombination von Wavelets selbst wieder ein Wavelet. Somit kann das Gaborwaveletnetz $(\mathbf{\Psi}, \mathbf{w})$ als Mutterwavelet aufgefaßt werden. Durch Hinzufügen der Freiheitsgrade Translation, Dilatation und Rotation erhält man ein sog. Superwavelet (eine Kurzform für *superposition-wavelet*), welches mit dem Levenberg-Marquardt-Algorithmus auf einem Grauwertbild optimal reparametrisiert werden kann.

Definition:

Sei $(\mathbf{\Psi}, \mathbf{w})$ ein GWN mit $\mathbf{\Psi} = (\psi_{\mathbf{n_1}}, \dots, \psi_{\mathbf{n_N}})$ und $\mathbf{w} = (\omega_1, \dots, \omega_N)$.
Das *Gaborsuperwavelet (GSW)* $\mathbf{\Psi}_\mathbf{n}$ ist als Linearkombination der Gaborwavelets $\psi_{\mathbf{n_i}}$ des GWN $(\mathbf{\Psi}, \mathbf{w})$ definiert:

$$\mathbf{\Psi}_\mathbf{n}(\mathbf{x}) = \sum_{i=1}^{N} \omega_i \psi_{\mathbf{n_i}} \left(\mathbf{SR}(x - t)\right) \ . \tag{7}$$

Dabei ergeben sich die Dilatationsmatrix $\mathbf{S}$, die Rotationsmatrix $\mathbf{R}$ und der Translationsvektor $\mathbf{t}$ aus dem Parametervektor $\mathbf{n}$ des GSW $\mathbf{\Psi}_\mathbf{n}$:

$$\mathbf{S} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{pmatrix}, \quad \mathbf{t} = \begin{pmatrix} c_x \\ c_y \end{pmatrix}.$$

Reparametrisierung des GSW

Sei $\mathbf{I}$ ein Grauwertbild mit dem Mittelwert $\bar{\mathbf{I}}$, auf dem das GSW $\mathbf{\Psi}_\mathbf{n}$ reparametrisiert werden soll. Ein Superwavelet erfüllt als Linearkombination von Wavelets die Voraussetzungen an ein Wavelet und hat die typischen Wavelet-Parameter Translation c_x, c_y,

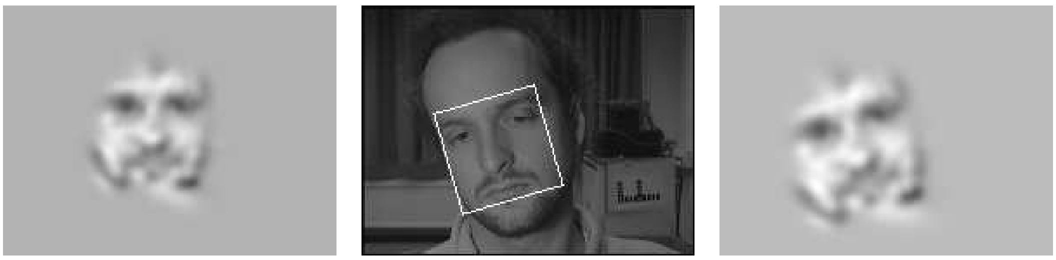
Dilatation s_x, s_y und Rotation α. Dadurch ist es möglich, wie bei der Optimierung des Gaborwaveletnetzes in Abschnitt 2.3.3, vorzugehen.
Das Gaborsuperwavelet $\mathbf{\Psi_n}$ kann auf sehr elegante Art und Weise durch Minimierung einer Energiefunktion E bezüglich seines Parametervektors $\mathbf{n}$ reparametrisiert werden:

$$E = \min_{\mathbf{n}} \|\mathbf{I} - \mathbf{\Psi_n} - \bar{\mathbf{I}}\|_2^2 . \tag{8}$$

Zur Minimierung der Energiefunktion (8) wird der Levenberg-Marquardt-Algorithmus verwendet.
Aufgrund seiner Konstruktion erbt das Superwavelet die Eigenschaften der "normalen" Wavelets und läßt sich mit Subpixelgenauigkeit positionieren und skalieren. Dabei bleibt die Struktur des Waveletnetzes unverändert. Es wird ausdrücklich darauf hingewiesen, daß *ausschließlich* die fünf Freiheitsgrade c_x, c_y, s_x, s_y und α des Superwavelets $\mathbf{\Psi_n}$ angepaßt werden. Die $6N$ Freiheitsgrade ($5N$ Parameter und N Gewichte) des Waveletnetzes $(\mathbf{\Psi}, \mathbf{w})$ bleiben von der Reparametrisierung unberührt.

Die Reparametrisierung des GSW $\mathbf{\Psi_n}$ entspricht dem Matching des durch das zugehörige Waveletnetz repräsentierten Gesichtes auf einem Grauwertbild $\mathbf{I}$. Der Vorteil gegenüber dem konventionellen Matching eines Grauwert-Templates liegt in der Möglichkeit, die Genauigkeit jederzeit durch Anpassung der Waveletanzahl zu steuern, ohne daß dazu verschiedene Templates gespeichert werden müßten.
Im weiteren werden wir die Begriffe *Reparametrisierung* und *Matching* synonym verwenden.

Ein Beispiel für ein reparametrisiertes Superwavelet ist in Abbildung 6 dargestellt.



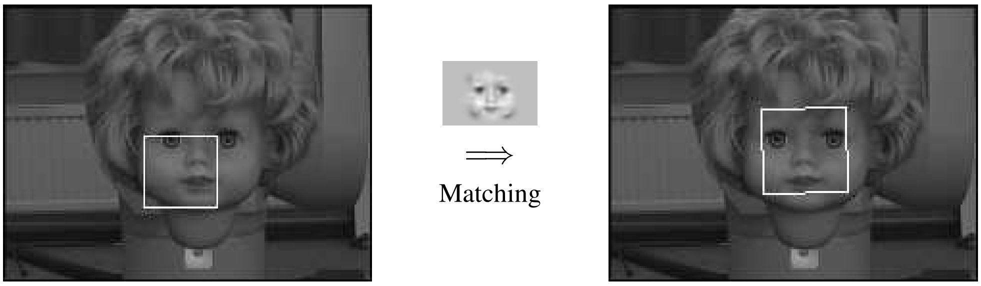
(a) (b) (c)

Abbildung 6: *In (a) ist das Superwavelet in seiner Grundform dargestellt, (b) zeigt das Bild, auf dem das GSW reparametrisiert wurde. Der weiße Rahmen markiert die Position des reparametrisierten GSW. Das reparametrisierte Superwavelet ist in (c) abgebildet. Aufgrund der nicht adaptierten Gewichte wirkt die Rekonstruktion "falsch".*

2.4.2 Konvergenzverhalten der Reparametrisierung

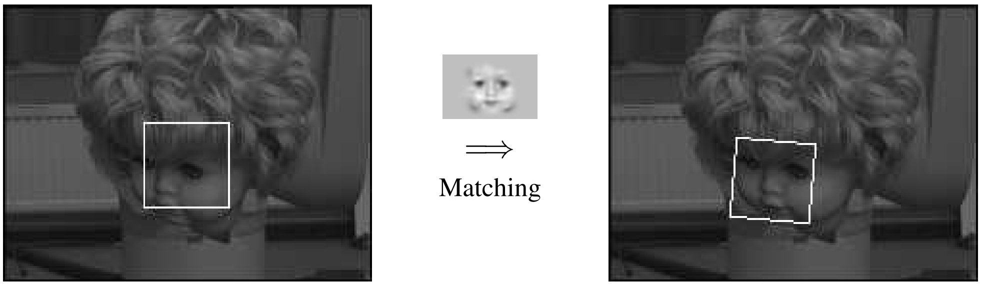
Da das Levenberg-Marquardt-Verfahren in der Regel nur in ein lokales Minimum gelangt, muß eine geeignete Vorinitialisierung der Parameter des Superwavelets gefun-

den werden.
In Experimenten hat sich gezeigt, daß die Reparametrisierung eines Gaborsuperwavelet aus 16 Wavelets bei einer Abweichung von ± 10 Pixeln in x- und y-Richtung, von $\pm 10\%$ in der Skala und von $\pm 10°$ bei der Rotation noch korrekt konvergiert (siehe dazu auch [14]). Abbildung 7 zeigt Beispiele für das Matching bei schlechter Initialisierung. 10 Pixel entsprechen etwa einem Drittel der Breite des eingezeichneten weißen Rechtecks.



(a) $c_x = 92.0, c_y = 87.0,$ $s_x = s_y = 1.1, \alpha = 0°$

(b) $c_x = 102.0, c_y = 76.4,$ $s_x = s_y = 0.98, \alpha = 0.6°$



(c) $c_x = 96.0, c_y = 82.0,$ $s_x = s_y = 0.95, \alpha = 0°$

(d) $c_x = 85.5, c_y = 91.5,$ $s_x = s_y = 1.05, \alpha = -5.2°$

Abbildung 7: *Beispiele für das Matching bei schlechter Initialisierung der Startwerte. In (a) und (c) sind Abweichungen um* ± 10 *Pixel im Aufpunkt und* $\pm 10\%$ *in der Skala dargestellt. Der Startwinkel wurde mit* $0°$ *initialisiert. In (b) und (d) sind die Ergebnisse der Reparametrisierung dargestellt.*

Der Konvergenzbereich hängt insbesondere auch von der Sichtbarkeit der Gesichtsmerkmale ab. Die angegebenen Werte beziehen sich dabei auf eine Frontalansicht. Ist das Gesicht im Kamerabild aufgrund starker Neigung extrem deformiert, wird der Konvergenzbereich immer schmaler, bis schließlich kein Matching mehr möglich ist. Diese Problematik ist typisch für merkmalsbasierte Ansätze und betrifft auch jene erscheinungsbasierten Ansätze, die Gesichtsmerkmale als Landmarken benötigen.
In Abbildung 7 (d) ist zu erkennen, daß aufgrund der fehlenden Scherung kein perfek-

tes Matching mehr möglich ist. Das Hinzufügen der Scherung s_{xy} zu den Freiheitsgraden des Superwavelets würde die volle Breite der affinen Transformationen ermöglichen. Andererseits wird das Matching mit zunehmender Anzahl an Freiheitsgraden instabiler. Der Verzicht auf die Scherung hat sich als guter Kompromiß zwischen Stabilität und Flexibilität erwiesen.

2.5 Anpassung der Gewichte

Wie bereits erwähnt, wird bei der Reparametrisierung des GSW nur eine Anpassung der Parameter des Superwavelets vorgenommen. Dadurch wird das Waveletnetz als Ganzes verformt. Die Gewichte der Wavelets bleiben dabei unberührt. Im Idealfall ist mit der Reparametrisierung des GSW das Matching erfolgreich beendet und die Gewichte der Wavelets haben bereits ihre optimalen Werte. Homogene Beleuchtungsänderungen, d.h. Veränderungen aller Grauwerte um einen konstanten Wert, haben Aufgrund der gleichanteilsfreien Wavelets keinen Einfluß auf den Matchingprozeß. Allerdings haben auch homogene Änderungen der Beleuchtungsverhältnisse einen Einfluß auf die Qualität der Rekonstruktion. Für eine optimale Rekonstruktion genügt es aber, den Mittelwert entsprechend anzupassen. Eine Neuberechnung der Gewichte ist in diesem Fall nicht notwendig.

In der praktischen Anwendung kann es Gründe geben, die eine Neuberechnung der Gewichte $(\omega_1, \ldots, \omega_N)$ erforderlich machen:

- Die Beleuchtungsverhältnisse haben sich inhomogen verändert: In der realen Welt sind homogene Beleuchtungsänderungen nur selten anzutreffen. Häufiger sind ungleichmäßige Beleuchtungsänderungen, z.B. durch das Einschalten einer Lichtquelle. In vielen Fällen erschwert Schattenwurf die Rekonstruktion.

- Durch den Verzicht auf Scherung kann das Waveletnetz nicht an alle auftretenden affinen Verformungen optimal angepaßt werden. Die zwangsläufig auftretenden Abweichungen in der Rekonstruktion können durch Anpassung der Gewichte verringert werden. Es wird sich später zeigen, daß diese Einschränkung einen Weg zur Bestimmung der Gesichtspose eröffnet.

Ein Beispiel dazu ist in Abbildung 8 dargestellt. Aufgrund der fehlenden Scherung wirkt die Rekonstruktion des nach links schauenden Puppenkopfes in (d) unnatürlich. Es wird sich aber zeigen, daß die Rekonstruktion für die Bestimmung der Gesichtspose hinreichend gut ist. In Experimenten hat sich gezeigt, daß die Freigabe der Scherung das Gesamtsystem instabil werden läßt. Die Bereitstellung einer geeigneten Regularisierung ist Aufgabe zukünftiger Arbeiten. Ferner erhöht sich der Zeitbedarf der Reparametrisierung mit der Anzahl der Freiheitsgrade, was insbesondere bei der Gesichtsverfolgung zu Zeitproblemen führen kann. Auf diese Problematik wird in Abschnitt 3 näher eingegangen.

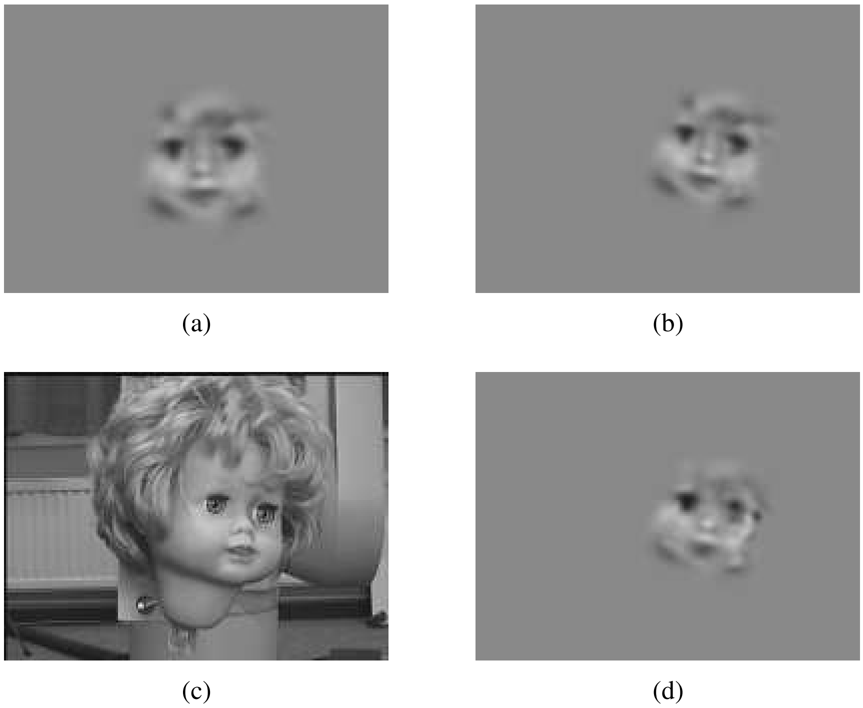


Abbildung 8: *In (a) ist das GSW dargestellt, (c) zeigt das Quellbild. Das reparametrisierte GSW ist in (b) zu sehen. Da sich das Kamerabild (c) und das GSW (a) in der Helligkeit unterscheiden, ist das Ergebnis unbefriedigend. In (d) zeigt sich die Verbesserung der Darstellung durch die Optimierung der Waveletgewichte. Aufgrund der fehlenden Scherung wirkt die Rekonstruktion des nach links schauenden Puppenkopfes unnatürlich.*

Theoretisch wäre eine Anpassung der Gewichte im Rahmen der Reparametrisierung denkbar. Obwohl die eigentliche Strukturinformation des Gesichtes durch die fest gewählten Paramtervektoren der Wavelets $\psi_1, \ldots, \psi_N$ getragen wird, ermöglicht die Freigabe der Gewichte eine weitgehende Anpassung des Waveletnetzes an beliebige Strukturen bezüglich der Energiefunktion (8).
Daher müssen die Gewichte in einem separaten Schritt, unter Fixierung der Positionierungsparameter des GSW, angepaßt werden.
Zur Anpassung der Gewichte stehen die Optimierung mit dem Levenberg-Marquardt-Algorithmus sowie die direkte Berechnung der Gewichte über die *dualen Wavelets* zur Wahl. Beide Verfahren sollen im folgenden vorgestellt werden:

2.5.1 Optimierung der Gewichte durch Minimierung der Energiefunktion

Die naheliegendste Methode zur Anpassung des Gewichtsvektors $\mathbf{w}$ ist die Verwendung des Levenberg-Marquardt-Verfahrens, welches in Abschnitt 2.3.3 zur Optimierung des gesamten Waveletnetzes $(\mathbf{\Psi}, \mathbf{w})$ verwendet wurde.
Dazu werden die Waveletparameter $\mathbf{n}_1, \ldots, \mathbf{n}_N$ festgehalten und die Energiefunk-

tion (6) bezüglich der Gewichte $(\omega_1, \ldots, \omega_N)$ minimiert.

Die Nachteile dieses Verfahrens sind zum einen die teure Berechnung und zum anderen die Gefahr, nur in ein lokales Minimun zu gelangen.

2.5.2 Direkte Berechnung der optimalen Gewichte

Die optimalen Gewichte, die den Approximationsfehler in Gleichung (8) global minimieren, können direkt berechnet werden.
Da die in dieser Arbeit verwendeten Gaborwavelets nichtorthogonal sind, läßt sich die in [19] vorgeschlagene Methode, die Projektionen $< \psi_{\mathbf{n}_i}, \mathbf{g} >$ der Wavelets $\psi_{\mathbf{n}_i}$ auf das zu approximierende Bild g als Gewichte zu verwenden, nicht direkt anwenden.
In [11; 14] wird als Lösung für dieses Problem die Verwendung der zu $\psi_{\mathbf{n}_1}, \ldots, \psi_{\mathbf{n}_N}$ dualen Wavelets vorgeschlagen.
Das Wavelet $\tilde{\psi}_{\mathbf{n}_i}$ heißt *duales Wavelet zu* $\psi_{\mathbf{n}_i}$, genau dann, wenn:

$$< \tilde{\psi}_{\mathbf{n}_i}, \psi_{\mathbf{n}_j} >= \delta_{i,j} \; .$$

Sei $\tilde{\mathbf{\Psi}} = \left(\tilde{\psi}_{\mathbf{n}_1}, \ldots, \tilde{\psi}_{\mathbf{n}_N}\right)^T$ der Vektor der zu $\mathbf{\Psi}$ dualen Wavelets. Offensichtlich gilt

$$\mathbf{\Psi}\tilde{\mathbf{\Psi}}^T = \mathrm{Id}, .$$

Dann ist

$$\tilde{\psi}_{\mathbf{n}_i} = \sum_{j=1}^{N} (\mathbf{\Psi}^{-1})_{i,j} \psi_{\mathbf{n}_j} \tag{9}$$

mit

$$\mathbf{\Psi}_{i,j} =< \psi_{\mathbf{n}_i}, \psi_{\mathbf{n}_j} > \; .$$

Wegen $\mathbf{w} =< \tilde{\mathbf{\Psi}}, \mathbf{g} >$ folgt unter Verwendung von Gleichung (9):

$$\begin{aligned} \omega_i &= < \tilde{\psi}_{\mathbf{n}_i}, \mathbf{g} > \\ &= < \sum_{j=1}^{N} (\mathbf{\Psi}^{-1})_{i,j} \psi_{\mathbf{n}_j}, \mathbf{g} > \\ &= \sum_{j=1}^{N} (\mathbf{\Psi}^{-1})_{i,j} < \psi_{\mathbf{n}_j}, \mathbf{g} > \end{aligned} \tag{10}$$

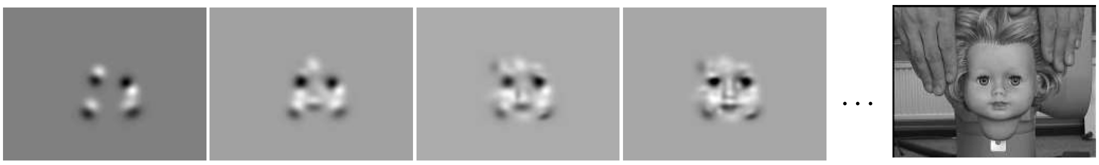
Ein Blick auf (10) zeigt, daß die Matrix $\mathbf{\Psi}^{-1}$ konstant ist, d.h. die Berechnung kann einmalig nach Abschluß der Trainingsphase durchgeführt werden.
Damit ist eine effiziente Methode zu direkten Berechnung der optimalen Gewichte gefunden.

2.6 Progressive Attention

Um in einer Echtzeitumgebung bestehen zu können, muß ein aktives System eine maximale Reaktionszeit einhalten (siehe Abschnitt 3). Die Anforderung, eine hohe Genauigkeit bei einer bestimmten Geschwindigkeit zu erreichen, kann von Waveletnetzen in optimaler Weise erfüllt werden. Durch die Anzahl M der verwendeten Wavelets kann die Genauigkeit der Rekonstruktion von sehr grob bis nahezu photorealistisch gezielt gesteuert werden.

$$\mathbf{g}_{(M)}(\mathbf{x}) = \sum_{i=1}^{M} \omega_i \psi_{\mathbf{n}_i}(\mathbf{x})$$

Diese Fähigkeit wird in Abbildung 9 demonstriert.
Bei einem geeignet konstruierten Netz kann die Waveletanzahl dynamisch an die aktuellen Erfordernisse angepaßt werden. Somit genügt es, ein einziges Waveletnetz mit der maximal erforderlichen Genauigkeit anzutrainieren. Dies kommt den Anforderungen von Echtzeitanwendungen entgegen, da auf diese Art und Weise - durch vollständige Ausschöpfung des zur Verfügung stehenden Zeitintervalles - eine maximale Menge an Information extrahiert werden kann. Eine Zeitbetrachtung zum Einfluß der Waveletanzahl auf den Matchingprozeß wird in Abschnitt 5.5.3 durchgeführt.



$N = 4$ $N = 8$ $N = 16$ $N = 52$ ... Original

Abbildung 9: *Waveletnetze eignen sich für die Verwendung im Zusammenhang mit progressiver Attention: Die Genauigkeit der Repräsentation kann je nach benötigter Präzision durch dynamische Anpassung der Zahl M der verwendeten Wavelets gesteuert werden: $\mathbf{g}_{(M)}(\mathbf{x}) = \sum_{i=1}^{M} \omega_i \psi_{\mathbf{n}_i}(\mathbf{x})$. Dazu genügt es, bei geeigneter Sortierung der Wavelets, ein einziges Waveletnetz mit der maximal benötigten Anzahl von Wavelets zu trainieren. Die dargestellten Abbildungen sind aus einem Netz mit 52 Gaborwavelets erstellt worden.*

2.7 Konstruktion geeigneter Waveletnetze für progressive Attention

An dieser Stelle soll ein Konstruktionsverfahren vorgestellt werden, welches beliebige Waveletnetze für progressive Attention anpaßt.
Um ein Maximum an Information bezüglich der Waveletanzahl zu extrahieren, ist es notwendig, die Wavelets nach ihrer "Wichtigkeit" zu sortieren. Dabei stellt sich die Frage, wie man die "Wichtigkeit" eines Wavelets bestimmen kann. Im folgenden werden drei Heuristiken vorgeschlagen und in Abschnitt 5 verglichen.

2.7.1 Verwendung mehrerer Teilnetze

Die alleinige Verwendung der groben Netzebene ist eine naheliegende Idee. Es wäre auch denkbar, ein Waveletnetz aus mehr als zwei Ebenen zu konstruieren, um eine feinere Abstufung zu erreichen. Dieses Vorgehen hat zwei Nachteile: Die gewünschte Struktur muß schon bei der Konstruktion des Netzes festgelegt werden. Ferner hängt die "Wichtigkeit" eines Wavelets auch vom Eingabebild ab. Hat sich z.B. die Beleuchtung geändert, oder wird sogar ein anderes Gesicht verwendet, so können nach Reparametrisierung und Anpassung der Gewichte durchaus Wavelets aus der feinsten Netzebene wichtige Bildinhalte repräsentieren. Daher wird ein flexibleres Verfahren benötigt.

2.7.2 Sortierung der Wavelets nach dem Betrag der Gewichte

Das Gaborsuperwavelet $\mathbf{\Psi_n}$ wird auf dem Bild der Grundstellung des Puppenkopfes reparametrisiert. Die Gewichte $(\omega_1, \dots, \omega_N)$ für eine optimale Rekonstruktion werden berechnet.

Definiere eine Permutation

$$\pi : \{1, \dots, N\} \longrightarrow \{1, \dots, N\} \text{ mit}$$

$$\pi(j) = \arg\max_{i \in \Omega_j} |w_i| \quad \text{für } j = 1, \dots, N \tag{11}$$

wobei

$$\Omega_j = \{1, \dots, N\} \setminus \left(\bigcup_{k=1}^{j-1} \pi(k) \right) .$$

Dann gilt $|\omega_{\pi(1)}| \geq \ldots \geq |\omega_{\pi(N)}|$.

Mit Hilfe der Permutation π können nun die Wavelets innerhalb des Waveletnetzes $(\mathbf{\Psi}, \mathbf{w})$ zu einem neuen Waveletnetz $(\pi\mathbf{\Psi}, \pi\mathbf{w})$ umgeordnet werden. Die Aufteilung in eine grobe und eine feine Netzebene wird durch die Umordnung überflüssig. Aus dem neuen Waveletnetz $(\pi\mathbf{\Psi}, \pi\mathbf{w})$ läßt sich das repräsentierte Gesicht in nahezu beliebigen Genauigkeitsabstufungen rekonstruieren.

Um eine Konvergenz des Matchingprozesses sicherzustellen, kann es Sinn machen, die Ebenenstruktur beizubehalten und eine Permutation π_{fein} nur auf die feinere Ebene anzuwenden.

Der Gewichtsvektor wird dazu in die Gewichte der ersten (groben) Ebene des Gaborwaveletnetzes $\omega_1, \dots, \omega_{N_1}$ und die Gewichte der zweiten (feinen) Ebene des Gaborwaveletnetzes $\omega_{N_1+1}, \dots, \omega_{N_1+N_2}$ unterteilt:

$$\mathbf{w}_{\text{fein}} = (\omega_1, \dots, \omega_N) = (\omega_1, \dots, \omega_{N_1}, \omega_{N_1+1}, \dots, \omega_{N_1+N_2}) \tag{12}$$

Definiere eine Permutation

$$\pi_{\text{fein}} : \{1, \ldots, N\} \longrightarrow \{1, \ldots, N\} \text{ mit}$$

$$\pi_{\text{fein}}(j) = \arg \max_{i \in \Omega_{\text{fein}_j}} |w_i| \quad \text{für } j = N_1 + 1, \ldots, N_1 + N_2 \tag{13}$$

wobei

$$\Omega_{\text{fein}_j} = \{N_1 + 1, \ldots, N_1 + N_2\} \setminus \left(\bigcup_{k=1}^{j-1} \pi_{\text{fein}}(k) \right) .$$

Durch die unveränderte grobe Ebene wird das Konvergenzverhalten der Reparametrisierung verbessert. In praktischen Experimenten war für eine zuverlässige Konvergenz eine grobe Ebene von 8-12 Wavelets notwendig.

2.7.3 Sortierung der Wavelets nach dem Betrag der normierten Gewichte

Die alleinige Betrachtung der Gewichte hat den Nachteil, daß kleine Wavelets mit großer Amplitude großflächigen Wavelets mit kleinerer Gewichtung vorgezogen werden, obwohl das großflächige Wavelet möglicherweise einen größeren Beitrag zur Minimierung der Energiefunktion (6) liefert.
Als Ausweg aus dieser Situation bietet es sich an, die Beträge der Gewichte mit der Skala der Wavelets zu normieren.

Definiere eine Permutation

$$\pi_{\text{norm}} : \{1, \ldots, N\} \longrightarrow \{1, \ldots, N\} \text{ mit}$$

$$\pi_{\text{norm}}(j) = \arg \max_{i \in \Omega_j} \left|\frac{w_i}{s_{x_i} s_{y_i}}\right| \quad \text{für } j = 1, \ldots, N \tag{14}$$

wobei

$$\Omega_j = \{1, \ldots, N\} \setminus \left(\bigcup_{k=1}^{j-1} \pi_{\text{norm}}(k) \right) .$$

Der Einfluß der hier vorgeschlagenen Verfahren zur Reduktion der Waveletanzahl wird in Abbildung 10 und 11 am Beispiel eines Puppenkopfes illustriert. Die visuelle Qualität der progressiven Rekonstruktion ist bei gleicher Waveletanzahl bei den mit π und π_{norm} sortierten Waveletnetzen sichtlich höher als beim unsortierten Waveletnetz. Ab 36 Wavelets ist bei den sortierten Netzen nahezu kein Unterschied zum Endzustand mehr erkennbar.
Die Ergebnisse der Posenschätzung unter Verwendung der verschiedenen Methoden werden in in Abschnitt 5 vorgestellt und diskutiert.

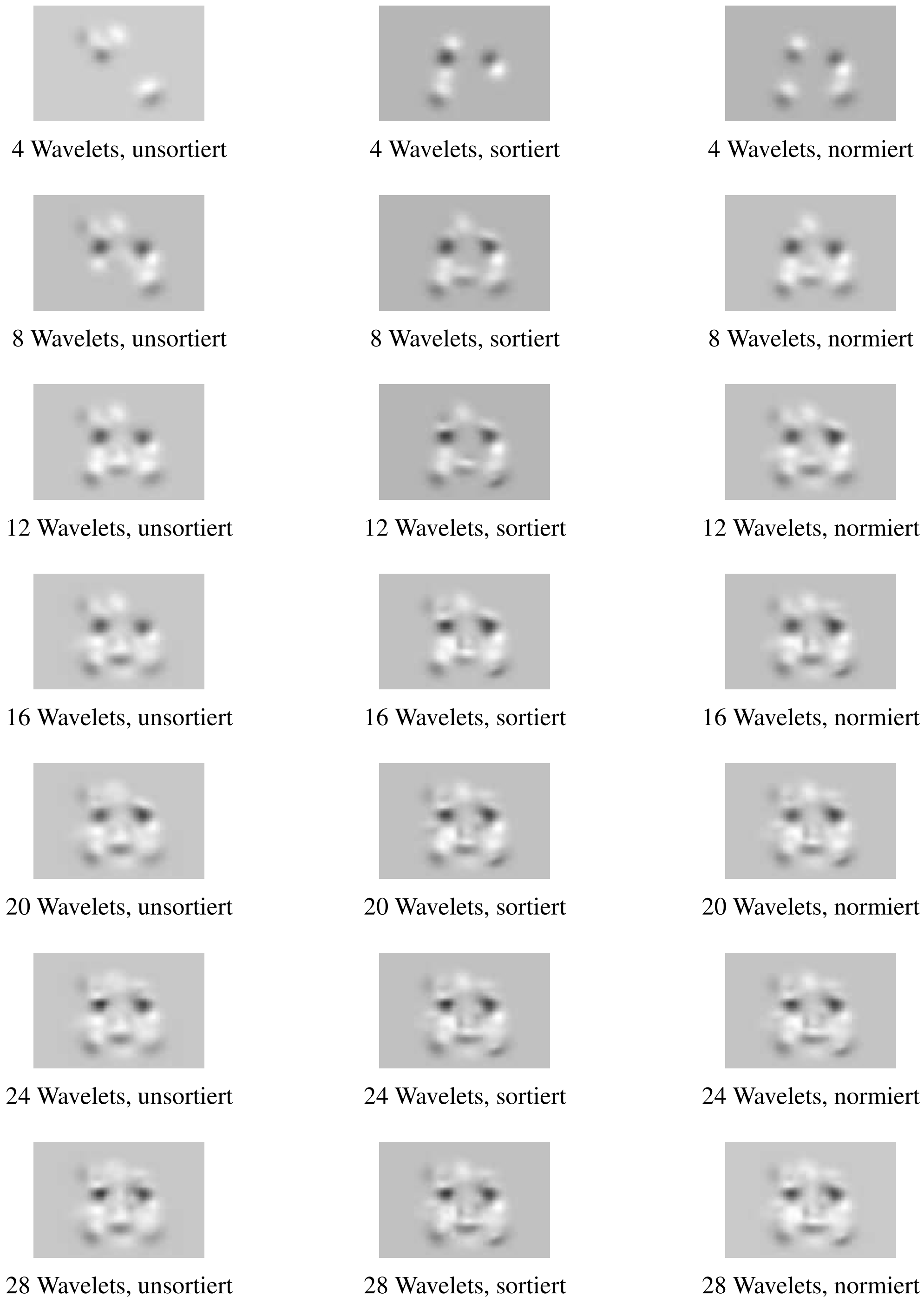
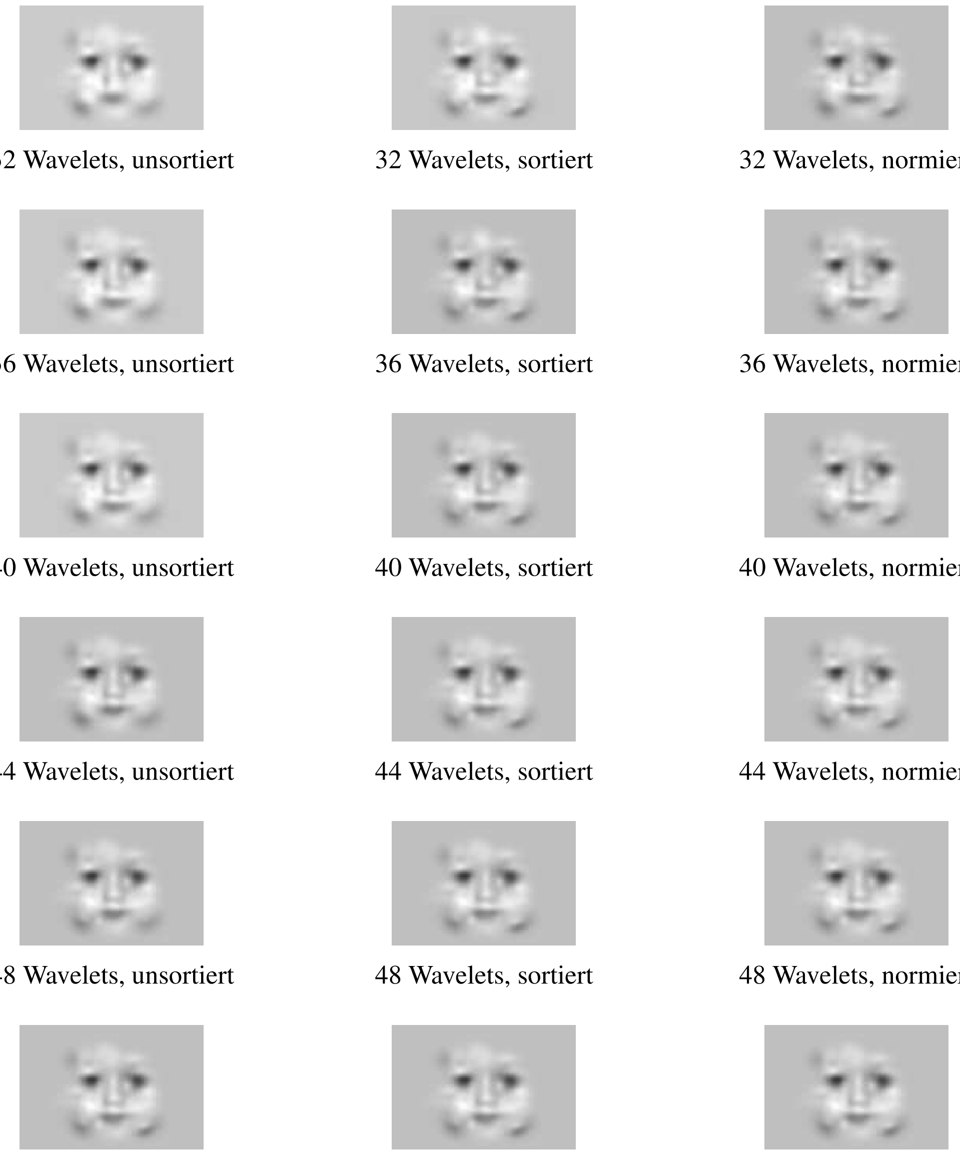


Abbildung 10: *Visueller Vergleich der drei Verfahren zur Reduktion der Waveletanzahl. Die linke Spalte zeigt das Ergebnis der Rekonstruktion bei Verwendung der ersten M Wavelets. In der mittleren Spalte wurden die Wavelets mit der Permutation π nach ihren Beträgen sortiert und die M ersten Wavelets verwendet. Die rechte Spalte zeigt das Rekonstruktionsergebnis bei Verwendung der Permutation π_{norm}.*
Ab etwa 16 Wavelets wirkt die Rekonstruktion aus den mit π und π_{norm} umgeordneten Waveletnetzen schärfer als aus dem unsortierten Netz in der linken Spalte.



32 Wavelets, unsortiert | 32 Wavelets, sortiert | 32 Wavelets, normiert

36 Wavelets, unsortiert | 36 Wavelets, sortiert | 36 Wavelets, normiert

40 Wavelets, unsortiert | 40 Wavelets, sortiert | 40 Wavelets, normiert

44 Wavelets, unsortiert | 44 Wavelets, sortiert | 44 Wavelets, normiert

48 Wavelets, unsortiert | 48 Wavelets, sortiert | 48 Wavelets, normiert

52 Wavelets | 52 Wavelets | 52 Wavelets

Abbildung 11: *Visueller Vergleich der drei Verfahren zur Reduktion der Waveletanzahl. Die linke Spalte zeigt das Ergebnis der Rekonstruktion bei Verwendung der ersten M Wavelets. In der mittleren Spalte wurden die Wavelets mit der Permutation π nach ihren Beträgen sortiert und die M ersten Wavelets verwendet. Die rechte Spalte zeigt das Rekonstruktionsergebnis bei Verwendung der Permutation π_{norm}.*
Während die Rekonstruktion aus den mit π und π_{norm} umgeordneten Waveletnetzen bereits mit 32 Wavelets vom Endergebnis kaum mehr unterscheidbar ist, verbessert sich die Darstellung aus dem unsortierten Netz in der linken Spalte mit weiteren Wavelets.

3 Gesichtslokalisierung durch Verfolgung

Das in Abschnitt 2 vorgestellte Verfahren zur Reparametrisierung des Superwavelets $\mathbf{\Psi_n}$ benötigt für die erfolgreiche Konvergenz eine Initialisierung der Startparameter in einer Umgebung der Lösung (siehe Abschnitt 2.4.2).
Daher muß die Gesichtsregion im Kamerabild vor der Anwendung der Reparametrisierung grob lokalisiert werden.
Die Gesichtslokalisierung in einem Kamerabild ist eine nicht triviale Aufgabe und Gegenstand vieler Arbeiten. In [1] wird das Problem durch eine Farbsegmentierung im RGB-Raum gelöst. Dabei wird angenommen, daß das Gesicht das einzige Objekt einer speziellen Farbregion im Kamerabild ist. Dem Vorteil der einfachen Realisierung steht als Nachteil eine geringe Stabilität gegenüber. Für eine erfolgreiche Lokalisierung müssen wohldefinierte Beleuchtungsverhältnisse eingehalten werden.
Ferner stellt sich das Lokalisierungsproblem in jedem Bild erneut, womit die Wahrscheinlichkeit einer Fehllokalisierung steigt.

In dieser Arbeit wird ein anderer Weg beschritten. Um das *Initialisierungsproblem* bei jedem neuen Bild zu vermeiden, wird ein *aktiver Ansatz* zur Gesichtslokalisierung verwendet.

3.1 Aktives Sehen

Das Vorgehen eines aktiven visuellen Systems ist auf eine oder mehrere spezielle Aufgaben ausgerichtet (*purposive vision*). Im Gegensatz zu passiven Systemen, welche die gesamte Szene analysieren müssen, um benötigte Informationen zu extrahieren, kann ein aktives visuelles System seine Fähigkeiten auf die Bildregion konzentrieren, die für die Lösung der Aufgabe relevant ist [2].
Die Konzentration der Attention auf einen Bildausschnitt wird durch die Annahme einer kontinuierlichen Welt ermöglicht. Unter der Voraussetzung, daß das interessierende Objekt eine bestimmte Geschwindigkeit nicht überschreitet, kann davon ausgegangen werden, daß das Objekt seine Position, Größe und Orientierung vom Zeitpunkt t zum Zeitpunkt $t + 1$ nur um einen kleinen Betrag verändert. Mit dieser Annahme genügt es zum Zeitpunkt $t + 1$ in einer kleinen Umgebung der Position im Bild t nach dem Objekt zu suchen. Dabei könnte auch eine durch Analyse der bisherigen Bildsequenz gewonnene Bewegungsschätzung verwendet werden.
Es stellt sich nur noch ein *Update-Problem*.
Die Abfolge von Bildaufnahme, Verarbeitung, Interpretation und Handlung bildet den sog. *Wahrnehmungs-Handlungs-Zyklus*. Der Zyklus ist in Abbildung 12 schematisch dargestellt.

3.1.1 Anforderungen an ein aktives visuelles System

Um in einer Echtzeitumgebung bestehen zu können, muß ein aktives System eine maximale Reaktionszeit einhalten, wobei die Definition des Begriffes *Echtzeit* von der

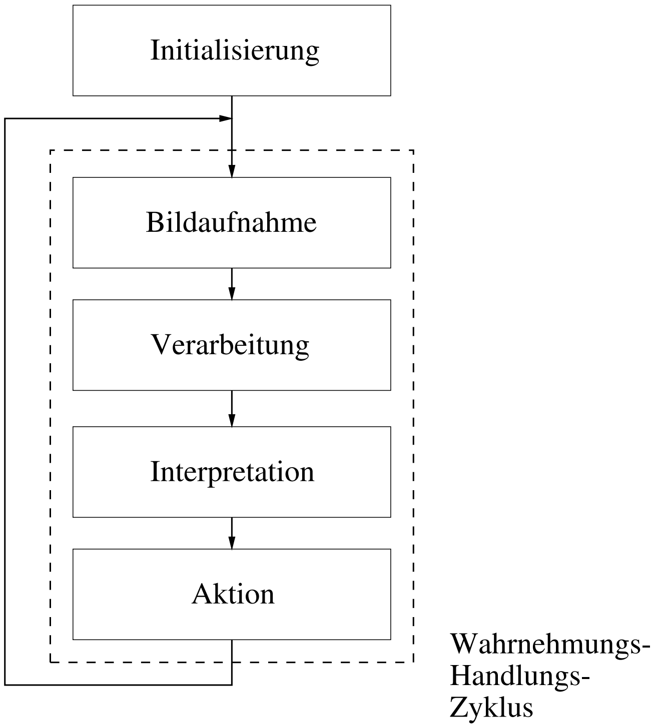


Abbildung 12: *Schematische Darstellung eines Wahrnehmungs-Handlung-Zyklus (Perception-Action-Cycle). Für die erfolgreiche Anwendung muß der gesamte Zyklus innerhalb einer maximalen Zeitspanne durchgeführt werden.*

konkreten Anwendung abhängt[8].
Für die in dieser Arbeit untersuchte Gesichtsverfolgung haben Experimente eine Geschwindigkeit von 20 Hz als brauchbaren Wert für ein stabiles Tracking ergeben. In Extremsituationen, wie z.B. plötzlichen Richtungswechseln, kam es bei dieser Geschwindigkeit in Experimenten dennoch gelegentlich zum Verlust des Objektes. Daher ist es von großer Wichtigkeit, daß eine möglichst hohe Geschwindigkeit eingehalten werden kann, d.h der *Wahrnehmungs-Handlungs-Zyklus* muß innerhalb einer maximalen Zeitspanne vollständig durchlaufen werden.
Im Gegensatz dazu muß für die Bestimmung der Gesichtspose eine möglichst große Menge an Informationen extrahiert werden. Dieses Dilemma kann durch Verwendung von *progressiver Attention* (siehe Abschnitt 2.6) in befriedigender Weise gelöst werden.

3.1.2 Vorteile aktiven Sehens

Im Gegensatz zu [1] werden in dieser Arbeit Bildfolgen $\mathbf{I}_0, \ldots, \mathbf{I}_T$ verwendet. Durch die Verwendung von zeitlich aufeinanderfolgenden Bildern ist es möglich, gezielt nur

[8]Wird z.B. die Drift der Kontinente betrachtet, genügt eine Messung pro Tag, soll der Flügelschlag eines Kolibris verfolgt werden, sind bis zu 200 Messungen pro Sekunde notwendig.

die interessierende Bildregion zu betrachten und über mehrere Bilder zu verfolgen. Dadurch entfällt eine permanente Neulokalisierung. Das *Initialisierungsproblem*, welches in [1] bei jedem Bild wieder auftritt, stellt sich nur einmalig zu Beginn des Versuchs.
Die Anforderungen an die Initialisierungsprozedur verringern sich bei einem aktiven visuellen System drastisch:

- Die Initialisierung kann unter genau definierten Bedingungen stattfinden, z.B. kann davon ausgegangen werden, daß die Testperson frontal in die Kamera blickt.
- Die Prozedur kann bei Versagen wiederholt werden, d.h. ein Fehler bei der Initialisierung gefährdet nicht die Stabilität des Systems.
- Die zur Initialisierung benötigte Zeit ist von untergeordneter Bedeutung. Dies ist besonders bei zeitkritischen Anwendungen ein nicht zu unterschätzender Vorteil. Es ist für einen Benutzer akzeptabel, einmalig für ein oder zwei Sekunden zu warten, wenn er sich danach vor dem Bildschirm frei und in normalem Tempo bewegen darf. Dagegen ist es nicht praktikabel, wenn sich ein Benutzer nur in Zeitlupe bewegen darf. Aus demselben Grund kann der Aufwand für die Initialisierungsprozedur größer sein, als bei einem passiven System.
- Eine manuelle Initialisierung oder deren Bestätigung ist einfach zu realisieren und für den Anwender nicht übermäßig belastend. Ferner ist die Qualität einer benutzergestützen Initialisierung der einer automatischen Initialisierung meist überlegen.

3.2 Gesichtsverfolgung

In vielen Situationen ist es wünschenswert, wenn ein Algorithmus ohne spezielles Wissen auskommt. Diesem Wunsch steht jedoch der Nachteil gegenüber, daß keinerlei Wissen über die Struktur des verfolgten Objektes benutzt werden kann. Daraus resultiert ein Verlust an Genauigkeit, da nur sehr allgemeine Annahmen (z.B. die Hautfarbe) benutzt werden können. So ist es zwar mit relativ geringem Aufwand möglich, aus der Farbinformation die Position des Kopfes im Bild sowie dessen Rotation bezüglich der Bildebene zu berechnen. In [5] werden beispielsweise die Position im Raum und der Rollwinkel des Kopfes extrahiert. Die für uns interessanten Freiheitsgrade der vertikalen Drehung und der horizontalen Neigung können durch diesen einfachen Ansatz hingegen nicht erhalten werden. Daher wird in dieser Arbeit der Farbverfolgungsansatz um eine erscheinungsbasierte Stufe erweitert. Für diesen Zweck wird ein Gaborwaveletnetz offline auf ein bekanntes Gesicht angelernt und dann zur Gesichtsverfolgung verwendet.

3.3 Versuchsaufbau der ursprünglichen Tracking-Software

Die ursprüngliche Tracking-Software [13] war dazu gedacht, das Gesicht eines Benutzers anhand der Gesichtsfarbe zu verfolgen. Dazu war auf dem Monitor eine aktive Kamera montiert (Freiheitsgrade: Zoom, Pan, Tilt). Die Kamera wurde so geregelt, daß sich das Gesicht des Benutzers immer in der Mitte des Kamerabildes befindet. Die Regelung erfolgt dabei nicht durch Angabe absoluter Kamera-Parameter, sondern durch Vorgabe der gewünschten Bewegungsrichtung. Die Regelung wird beendet, wenn das Objekt sich an der gewünschten Stelle im Kamerabild befindet. Dieser Ansatz hat den Vorteil, daß sich der Benutzer in einem relativ großen Bereich vor der Kamera frei bewegen und sogar im Raum umhergehen kann.
Der Nachteil der aktiven Verfolgung durch eine bewegte Kamera ist jedoch, daß sich die Orientierung des Objektes zur Kamera ändert. Dadurch ist die Grundwahrheit (d.h. relative Position der Kamera zum Objekt) nur näherungsweise bekannt, da ein Auslesen der Kamera-Parameter sehr langsam ist. Zweitens ist es nicht möglich, eine zeitliche Korrespondenz zwischen Bildaufnahme und Kamera-Parametern herzustellen, da die Kamera das Erreichen einer Position nicht signalisieren kann.
Aus diesem Grund wird in dieser Arbeit mit einer *passiven* Kamera gearbeitet. Dadurch ist es möglich, zuverlässige Aussagen über die Gesichtspose zu treffen.

3.4 Der Tracking-Algorithmus

Der verwendete Tracking-Algorithmus besteht aus vier Stufen: Bewegungsdetektion, Farbdetektion, Farbverfolgung und Gaborwaveletnetz-Verfolgung. Von Schritt zu Schritt verringert sich der betrachtete Bildausschnitt, so daß mit jedem Schritt mehr Rechenleistung pro Bildpunkt zur Verfügung steht.
Dieser Zuwachs an freier Rechenkapazität schafft Raum für immer genauere, immer kostenintensivere Algorithmen und gipfelt in der merkmalsbasierten Gesichtsverfolgung mit Gaborwaveletnetzen, welche mit Subpixelgenauigkeit erfolgt. Der schematische Ablauf ist in Abbildung 13 dargestellt. Dabei steigen die Genauigkeit und der Rechenaufwand mit jeder Stufe der Pyramide, während gleichzeitig die Robustheit abnimmt. Um dem entgegenzuwirken, wird beim Verlust des Objektes um eine oder mehrere Stufen in der Pyramide abgestiegen, bis das Objekt wiedergefunden wird.

3.4.1 Vorgabe einer Hautfarbe

Die Trackingsoftware ist als endlicher Automat implementiert. Der Programmablauf ist in Abbildung 14 vereinfacht dargestellt. In Schritt 1 wird während der Initialisierung einmalig eine Hautfarbe interaktiv vorgegeben. Dadurch wird, gegenüber einer fest kodierten Hautfarbe, eine höhere Flexibilität bei verschiedenen Hauttypen und Beleuchtungsverhältnissen erreicht. In Schritt 2 wird die Kamera initialisiert, d.h. Zoom, Pan und Tilt werden auf die jeweiligen Grundwerte zurückgesetzt.

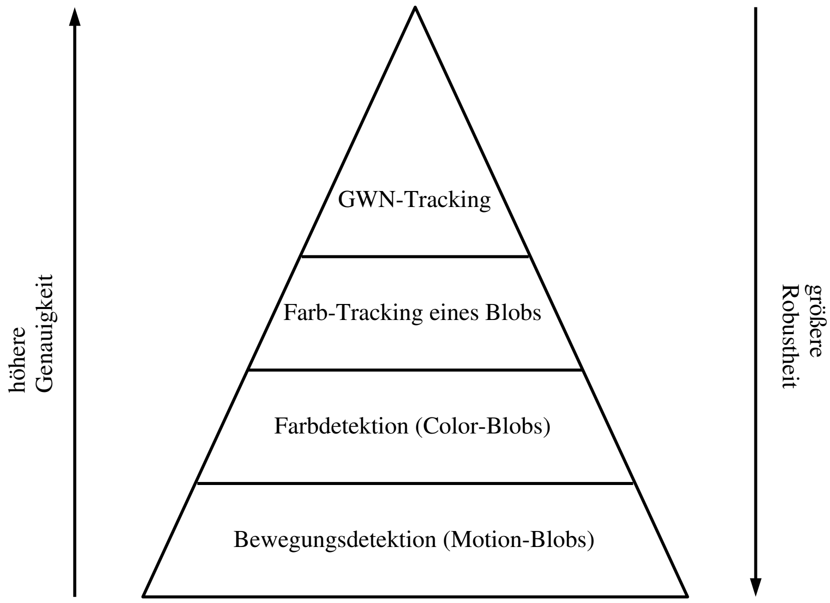


Abbildung 13: *Schematischer Ablauf des Tracking-Algorithmus: Bewegungsdetektion, Farbdetektion, Farbverfolgung und Gaborwaveletnetz-Verfolgung. Die Breite der Pyramide gibt die Größe des betrachteten Bildausschnittes wieder.*

3.4.2 Bewegungsverfolgung

Der Algorithmus beginnt damit, daß im gesamten Kamerabild nach Bewegungen gesucht wird. Diesem Ansatz liegt die Annahme zu Grunde, daß sich Personen die ganze Zeit in Bewegung befinden. Dieser Schritt ist trotz der Vorgabe der Hautfarbe sinnvoll, da nur durch dieses Vorgehen verhindert werden kann, daß unbelebte Objekte oder Objekte im Bildhintergrund, etwa Werbeplakate, fälschlicherweise selektiert werden. Ferner muß die detektierte Bewegung einen festen Schwellwert überschreiten, um eine Fehldetektion durch Bildrauschen zu verhindern. Diese Stufe entspricht Schritt 3 in Abbildung 14.

3.4.3 Farbdetektion

Die gefundenen Bereiche, die Bewegung enthalten, werden in Schritt 4 auf ihre Ähnlichkeit mit der vorgegebenen Hautfarbe untersucht. Es wird angenommen, daß der größte hautähnliche Bereich das der Kamera am nächsten gelegene Objekt repräsentiert. Dieses Objekt wird als “aktuelles Objekt” ausgewählt. Dann wird zur Farbverfolgung (Schritt 5) gewechselt.

3.4.4 Farbverfolgung

Bei der Farbverfolgung (Schritt 5) wird nur noch die direkte Umgebung des aktuellen Objektes betrachtet. Die Verfolgung geschieht mit Hilfe des CAMSHIFT[9]-Algorithmus, wie er in [5] vorgestellt wurde.
Wenn das Objekt über 10 Iterationen stabil geblieben ist, wird versucht, zur Gaborwaveletnetz-basierten Gesichtsverfolgung zu wechseln.

3.4.5 Gaborwaveletnetz-Verfolgung

Im Gegensatz zur farbbasierten Gesichtsverfolgung wird in der Gaborwaveletnetz-Verfolgung (Schritt 6) das Gesicht anhand seiner charakteristischen Merkmale verfolgt. Dazu wird versucht ein Gaborsuperwavelet des Gesichtes auf dem Kamerabild zu reparametrisieren.
Die initiale Position und Skala für die Reparametrisierung werden aus den Parametern der farbbasierten Objektverfolgung abgeleitet. Der Einfachheit halber wird die initiale Position als Schwerpunkt des aktuellen Objektes angenommen. Die initiale Skala wird als Quotient vom Durchmesser des Objektes und der Größe des Gaborsuperwavelets bestimmt. Unter diesen Voraussetzungen klappt der Wechsel meistens, ist das Gesicht jedoch weit von einer Frontalansicht entfernt, schlägt der Wechsel gelegentlich fehl und es wird in die Farbverfolgung (Schritt 5) zurückgewechselt.

Alternative Möglichkeiten für den Wechsel zur GWN-Verfolgung

Eine stabilere Lösung für dieses Problem bietet Raum für weitere Untersuchungen. Im speziellen Fall des Puppengesichtes wäre es beispielsweise möglich, die blauen Augen als Maxima im UV-Bild zu lokalisieren. Da in dieser Arbeit nur möglichst allgemeine Annahmen gemacht werden sollen, scheidet diese Möglichkeit aus.
Eine andere Möglichkeit wäre es auszunutzen, daß die Reparametrisierung in einer Umgebung der Lösung eindeutig konvergiert (siehe Abschnitt 2.4.2). Dazu wird die detektierte Gesichtsregion mit einem Raster überzogen. Auf diesem Raster versuchen wir - von innen nach außen - Gaborsuperwavelets anzupassen. Abbildung 15 illustriert diesen Algorithmus. Dabei wäre es sinnvoll, in der Mitte der vermuteten Gesichtsregion zu beginnen und den Suchbereich nach jeder Iteration zu vergrößern, bis das gesamte Raster abgedeckt ist. Das Verfahren ist beendet, wenn ein Erfolg eintritt oder eine maximale Zahl von Iterationen erreicht ist. Ein erfolgreiches Matching muß anhand der Parameter des GSW auf Plausibilität geprüft und ggf. verworfen werden. Kriterien für ein korrektes Matching sind eine weitgehende Übereinstimmung zwischen Skala und Position des Superwavelets mit der detektierten Fläche sowie ein kleiner Rotationswinkel[10]. Der Rechenaufwand für dieses Suchverfahren ist in ungünstigen Fällen

[9] Continuously Adaptive Mean Shift

[10] In unseren Experimenten betrug der Rotationswinkel selbst in extremen Posen immer weniger als 25°. Bei mißlungenen Reparametrisierungen traten häufig Rotationswinkel größer als 30° auf.

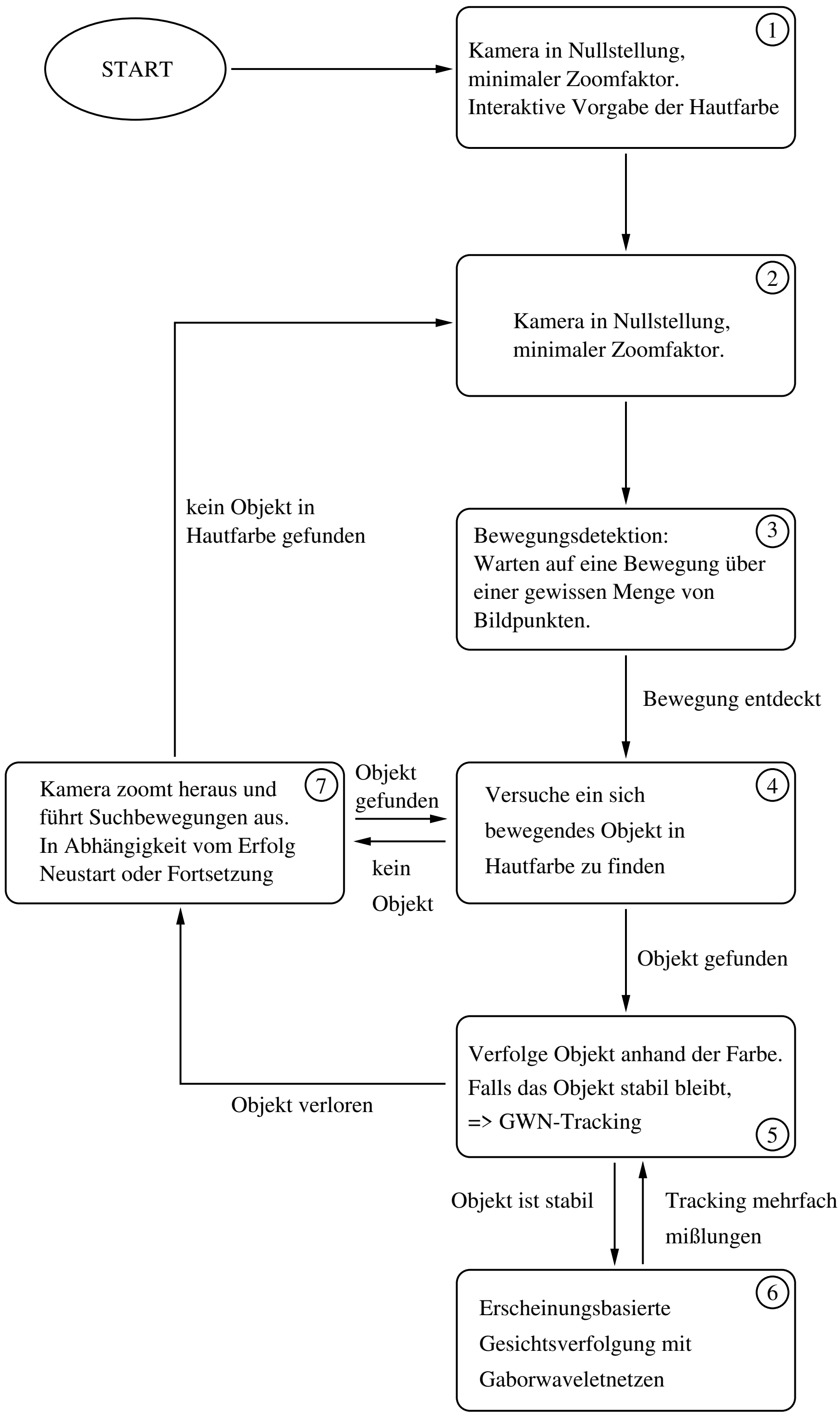


Abbildung 14: *Ein grober Überblick über den Programmablauf der Tracking-Software. Der Algorithmus wurde als endlicher Automat implementiert.*

(starke Neigung des Gesichts) sehr hoch. Aufgrund der Unabhängigkeit der einzelnen Reparametrisierungen kann das Verfahren auf Mehrprozessorsystemen zumindest parallelisiert werden.

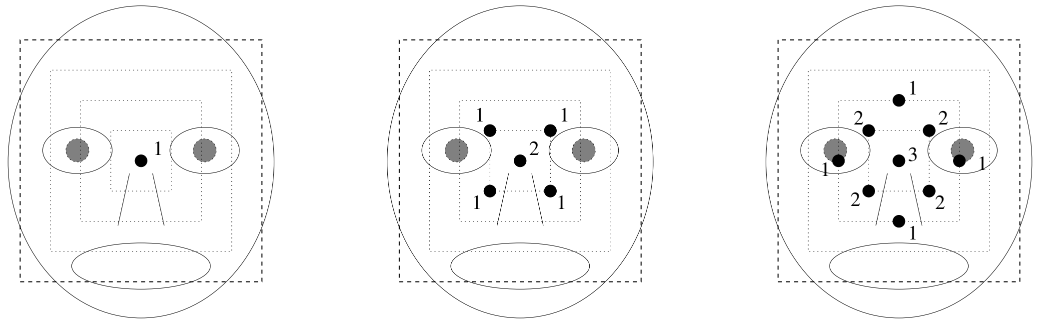


Abbildung 15: *Vorschlag eines Suchschemas für die automatische Suche der initialen Parameter des GSW. Der gestrichelte Rahmen umschreibt die durch Farbverfolgung gefundene Gesichtsfläche. Der Algorithmus beginnt in (a) mit einer Iteration im innersten Suchbereich. In (b) ist der zweite Schritt dargestellt, in dem innen die zweite Iteration durchgeführt wird und im nächsten Bereich die erste Iteration. In (c) ist der dritte Schritt illustriert. Die Iterationen werden diesem Schema folgend solange fortgesetzt, bis ein GSW erfolgreich reparametrisiert wurde oder eine maximale Zahl von Iterationen erreicht ist.*

3.5 Implementierung

Im Rahmen dieser Diplomarbeit wurde die Tracking-Software aus [13] von der Programmiersprache C nach C++ portiert, neu strukturiert und erweitert. Ferner wurden Anpassungen an die neue Hardware-Plattform (Intel-PC unter dem Betriebssystem Linux) vorgenommen. Die ursprüngliche Version der Software war an eine SUN-Workstation unter dem Betriebssystem Solaris angepaßt.

3.5.1 Anpassungen für den Echtzeitbetrieb

Für den Echtzeiteinsatz auf Standard-Hardware müssen einige Einschränkungen gemacht werden. Die horizontale und vertikale Bildgröße von 384×288 Pixeln wird durch Weglassen jeder zweiten Zeile und Spalte auf 192×144 Pixel reduziert, so daß sich die Menge der Bilddaten auf ein Viertel verringert[11].
Ferner wird ein relativ grobes GSW, bestehend aus 9–16 Wavelets, verwendet.
Mit diesen Einschränkungen ist eine Gesichtsverfolgung mit etwa 20 bis 28 Bildern/Sekunde möglich[12].

[11] Als beabsichtigter Nebeneffekt wird damit auch der "Lattenzauneffekt" unterdrückt.

[12] Der Wert von 28 Bildern pro Sekunde gibt einen Spitzenwert an. Da die Bildextraktion asyn-

Dies genügt in vielen Fällen für eine stabile Gesichtsverfolgung. Das Verfahren bricht ab, wenn in drei aufeinanderfolgenden Versuchen der Matching-Algorithmus nicht konvergiert. In diesem Fall wird zur Farbverfolgung zurückgewechselt. Eine mögliche Erweiterung für den Fall eines Fehlmatchings wäre eine Bewegungsvorhersage, die aus der Bewegung des Gesichtes in den letzten Bildern die Position im neuen Bild schätzt. Die Eignung dieser Erweiterung muß aber noch genauer untersucht werden, da sich in den Versuchen der plötzliche Wechsel der Bewegungsrichtung, wie er bei der Arbeit am Bildschirm häufig auftritt, als besonders problematisch erwiesen hat.
In diesem Fall würde eine Bewegungsvorhersage prinzipbedingt eine falsche Position vorhersagen. Es erscheint erfolgversprechender, den Matching-Algorithmus weiter zu beschleunigen, oder auf schnellere Hardware auszuweichen.
Beispielsweise wäre es möglich, die beiden Halbbilder eines PAL-Signals getrennt auszuwerten und somit einen Videostrom von 50 echten Bildern pro Sekunde zu erhalten. Unter der Voraussetzung, daß der Matching-Algorithmus diese Bildrate bewältigen könnte, ist eine erhebliche Steigerung der Stabilität in kritischen Situationen zu erwarten. Dieser Gedanke wird in Abschnitt 6.2.3 im Zusammenhang mit der Posenbestimmung näher betrachtet.
Weiterhin könnte es sinnvoll sein, zusätzlich zum Waveletnetz-Tracking die Farbverfolgung fortzuführen und ggf. die Position des Waveletnetzes über die Farbverfolgung direkt zu korrigieren, ohne die merkmalsbasierte Verfolgung verlassen zu müssen.

3.5.2 Beschränkungen der Hardware

Im Laufe der Implementierungsarbeiten mußten aufgrund von Beschränkungen der Hardware, bzw. der zugehörigen Software, folgende Kompromisse eingegangen werden:

Asynchrone Bildextraktion

Die Bildextraktion erfolgt in dieser Arbeit asynchron zur Bildaufnahme. Dies ist eine Einschränkung der verwendeten Schnittstellensoftware zur Framegrabberkarte. Das analoge Kamerabild wird von links oben nach rechts unten digitalisiert und in eben dieser Reihenfolge im Speicher abgelegt. Da die Digitalisierung nicht in einem Stück, sondern zeilenweise durchgeführt wird, kann es vorkommen, daß das aktuelle Kamerabild erst zu einem Teil digitalisiert ist, wenn das Programm auf das digitalisierte Bild zugreift. In diesem Fall wird eine Mischung aus einem Teil des neuen und des alten Bildes ausgelesen. Dies kann insbesondere bei schnellen Bewegungen zu dem Effekt führen, daß die obere Hälfte eines Objektes sich an einer anderen räumlichen Position befindet, als die untere Hälfte[13]. Der Matching-Algorithmus schlägt auf solchen

chron zur Bildaufnahme erfolgt, kann dieser Wert über den tatsächlichen 25 Bildern/Sekunde des PAL-Formats liegen. In diesem Fall werden manche Bilder mehrfach ausgewertet.

[13]Dieser Effekt hat nichts mit dem bekannten “Lattenzauneffekt” zu tun. Der Lattenzauneffekt wird durch die Aufteilung des Bildes in zwei Halbbilder aus geraden und ungeraden Zeilen verursacht. Der

Bildern in der Regel fehl.

Geringe Farbauflösung

Die in dieser Arbeit verwendete Framegrabberkarte liefert im YUV-Modus eine Luminanzkomponente Y mit 8 Bit Auflösung, d.h. 256 verschiedenen Helligkeitswerten. Die Chrominanzkomponenten U und V werden jedoch nur mit 4 Bit, also 16 Werten, aufgelöst. Dadurch ist die Gesichtsverfolgung über die Hautfarbe weniger stabil als auf den SUN-Rechnern, deren Framegrabberkarten auch in die U- und V-Komponenten mit 8 Bit auflösen. Für die praktische Anwendung ist die 4 Bit-Auflösung aber noch hinreichend gut.

3.6 Experimentelle Ergebnisse

Die Grundvoraussetzung für die kontinuierliche Bestimmung der Gesichtspose ist es, das Gesicht zuverlässig zu detektieren. Um diese Voraussetzung zu erfüllen, ist als Zwischenetappe die Tracking-Software um eine merkmalsbasierte Gesichtsverfolgung erweitert worden. Es war das Ziel, einen schnellen Algorithmus zu entwickeln, der aber hinreichend robust ist, um in der Praxis von Nutzen zu sein.
In Abbildung 16 sind vier Bilder aus einer Tracking-Sequenz dargestellt. Das weiße Rechteck umschreibt die Fläche des reparametrisierten Gaborsuperwavelets, d.h. die detektierte Gesichtsfläche. Das Tracking lief stabil mit etwa 20 Hz, während sich die Testperson normal vor dem Bildschirm bewegt hat.

Gaborsuperwavelet aus 16 Wavelets

(a) (b) (c) (d)

Abbildung 16: *Einzelbilder aus einer Tracking-Sequenz: (a) Bild 3, (b) Bild 10, (c) Bild 26, (d) Bild 43. Das weiße Rechteck umschreibt das durch Matching mit dem oben stehenden GSW detektierte Gesicht.*

hier genannte Effekt verursacht Verschiebungen *innerhalb* eines Halbbildes.

4 Posenbestimmung mit Gaborwaveletnetzen

In dieser Arbeit wird die Gesichtspose anhand von Gaborwaveletnetzen bestimmt. Dazu werden unter Verwendung des Gaborsuperwavelets Merkmale aus Grauwertbildern extrahiert und als Eingabevektoren für künstliche neuronale Netze verwendet.
Die erste Hälfte dieses Kapitels stellt den Versuchsaufbau (4.1) und ein Verfahren zur Posenbestimmung vor. Die Bestimmung der Gesichtspose unterteilt sich in drei Schritte: In (4.2) wird das Gesicht subpixelgenau lokalisiert. Dann werden in (4.3) unterschiedliche Merkmale sowie Verfahren zu deren Extraktion vorgestellt. Schließlich wird in (4.4) ein automatisiertes Verfahren zur Erzeugung einer Trainings- und Testdatenmenge angegeben.
Im der zweiten Hälfte dieses Kapitels werden ab (4.5) die verwendeten neuronalen Netze in kurzer Form eingeführt.
Die Ergebnisse der Posenschätzung werden in Kapitel 5 ausführlich vorgestellt und diskutiert.

4.1 Versuchsaufbau

Um die Ergebnisse dieser Arbeit mit denen aus [1] vergleichbar zu halten, wurde der Versuchsaufbau weitestgehend unverändert übernommen. Insbesondere wurde für die Experimente derselbe Puppenkopf benutzt (siehe Abbildung 17).
Der etwa 20 cm durchmessende Puppenkopf wurde mittels einer speziellen Halterung senkrecht unter dem Handgelenk des Roboterarms RX-90 der Firma Stäubli befestigt. Der Roboterarm verfügt über 6 Gelenke und ist in der Lage, eine vorgegebene Position innerhalb seines Arbeitsbereiches mit einer hohen Genauigkeit anzufahren.
Um Aussagen über die Genauigkeit der Posenschätzung treffen zu können, muß sichergestellt sein, daß die Grundwahrheit bekannt ist. Dies ist eine prinzipielle Schwierigkeit aller Verfahren, die Gesichtspose einer Person zu bestimmen und die Voraussetzung für die Verwendung eines überwachten Lernverfahrens. Eine Betrachtung der Vorgehensweise in anderen Arbeiten zeigt eine große Vielfalt von Methoden, an die Grundwahrzeit zu gelangen: Chen et al. beispielsweise arbeiten mit computererzeugten 3D-Modellen von Testpersonen [9], während bei Klingspohr, Block und Grigat der Kopf der Versuchsperson kurzerhand fixiert wurde [12].
Durch Verwendung des Roboterarmes ist die Grundwahrheit, also die Pose des Puppenkopfes, bekannt. Eine Verfälschung der Ergebnisse durch einen Fehler in der Grundwahrheit kann daher ausgeschlossen werden.
Für die Durchführung der Versuche arbeitete der Roboter im Werkzeugkoordinatensystem. In diesem Betriebszustand werden alle Koordinaten relativ zur Grundposition des Roboterhandgelenkes angegeben. Der Ursprung des Werkzeugkoordinatensystems wurde in die Mitte des Puppenkopfes verschoben. Dadurch war es möglich, den Rotations- und Neigungswinkel direkt vorzugeben.
Der Kopf wurde mit einer handelsüblichen Farbvideokamera vom Typ Sony EVI-D31 aufgenommen, die auf einem Tisch direkt vor der Basisplatte des Roboters positioniert

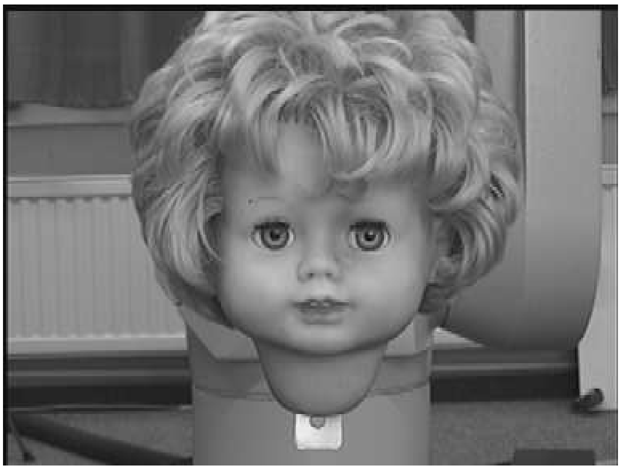


Abbildung 17: *Der in den Experimenten verwendete Puppenkopf hat blonde Haare, helle Haut, ein relativ rundes Gesicht und ist etwa 20 cm groß. Derselbe Kopf wurde bereits in der Arbeit von E. Ábráham-Mumm benutzt.*

war.
Die Höhe des Tisches wurde so eingestellt, daß sich der Kopf ungefähr in der Mitte des Kamerabildes befand. Ferner wurde der Zoom der Kamera so eingestellt, daß das Puppengesicht in seiner Grundstellung, wie in Abbildung 17, etwa 50% der Höhe des Kamerabildes ausfüllte.
Um die Szene auszuleuchten, wurde hinter der Kamera eine weiße Wand plaziert, die von Halogenstrahlern beleuchtet wurde. Diese Anordnung bewirkt eine gleichmäßige, indirekte Beleuchtung des Puppenkopfes. In Abbildung 18 ist der Versuchsaufbau schematisch dargestellt.

4.2 Gesichtslokalisierung

Unter Verwendung der Ergebnisse der Gesichtsverfolgung wird je nach Bedarf entweder in regelmäßigen Abständen oder beim Eintritt eines Ereignisses - wie einem Mausklick oder einem gesprochenen Kommando - eine exakte Reparametrisierung des Gaborsuperwavelets $\mathbf{\Psi_n}$ auf dem aktuellen Grauwertbild $\mathbf{I}_t$ durchgeführt. Die Filter werden in diesem Schritt wieder optimal - in bezug auf die zur Verfügung stehenden Freiheitsgrade - über den durch sie repräsentierten Merkmalen plaziert. Ein Beispiel für die Plazierung der Filter durch Reparametrisierung ist in Abbildung 19 dargestellt.

4.3 Merkmalsextraktion

In dieser Arbeit werden künstliche neuronale Netze zur Schätzung der Gesichtspose verwendet. Die verwendeten Netztypen werden am Ende dieses Kapitels in Abschnitt 4.5 vorgestellt.

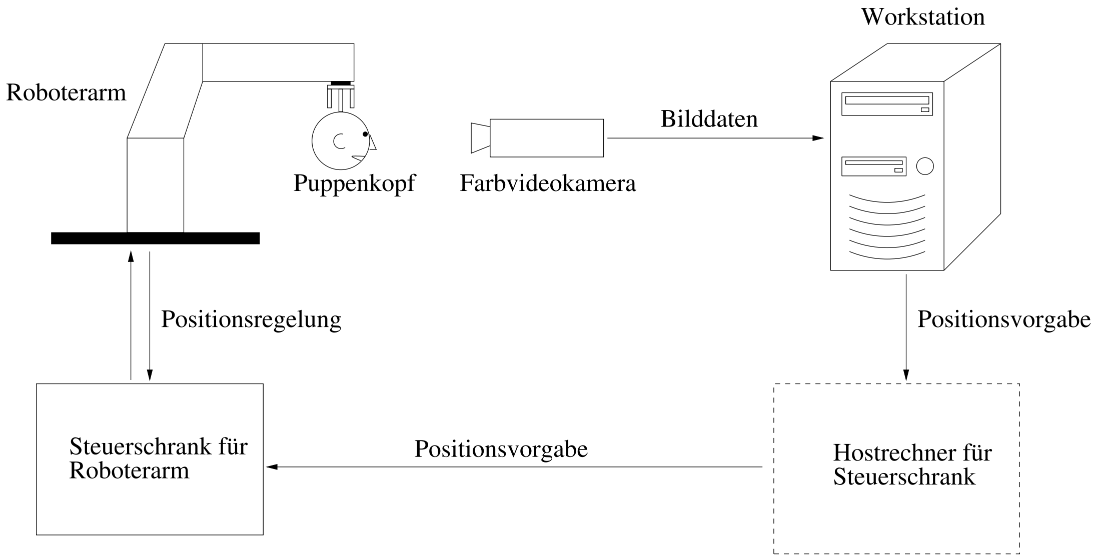
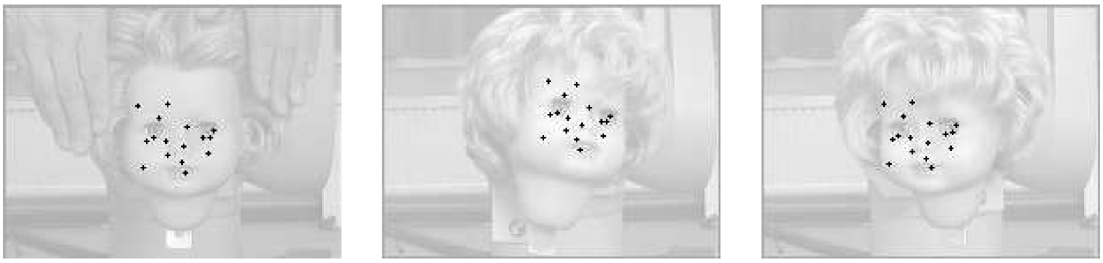


Abbildung 18: *Der Puppenkopf ist am Handgelenk des Roboterarmes befestigt und wird von einer Farbvideokamera frontal aufgenommen. Die Gesichtspose kann über die Positionierung des Handgelenkes von der Workstation präzise vorgegeben werden - die Grundwahrheit für die Experimente zur Posenschätzung ist damit exakt bekannt.*



(a) (b) (c)

Abbildung 19: *In (a) sind die ursprünglichen Waveletposition auf dem Trainingsbild dargestellt. In (b) und (c) sind die Waveletpositionen nach der Reparametrisierung des GSW Ψ_n auf zwei zufällig gewählten Bildern einer Testsequenz markiert. Der Übersichtlichkeit halber sind nur die ersten 16 Wavelets des gesamten Netzes eingezeichnet.*

Die Merkmalsextraktion stellt die Projektion der Eingabedaten auf einen niederdimensionalen Unterraum dar, trennt also insbesondere problemrelevante von problemirrelevanter Information.
Für eine gute Posenschätzung ist eine geeignete Merkmalsextraktion notwendig. Dabei sollte die Anzahl der extrahierten Merkmale zugunsten der Effizienz möglichst klein sein, gleichzeitig aber genug Information kodieren, so daß eine gute Genauigkeit erreicht wird.
Mit den Gaborwaveletnetzen haben wir in Abschnitt 2 eine kompakte Gesichtsre-

präsentation vorgestellt, die mit wenigen Wavelets eine große Menge an Information kodiert. Im folgenden wollen wir das GWN $(\mathbf{\Psi}, \mathbf{w})$ auf seine Eignung als Eingabe für ein künstliches neuronales Netz untersuchen.

4.3.1 Waveletgewichte der Rekonstruktion

Durch seine eingeschränkten Freiheitsgrade kann das Gaborsuperwavelet $\mathbf{\Psi_n}$ beim Matching auf Bildern $\mathbf{I}$, die von der Trainingsansicht $\mathbf{g}_{\mathrm{train}}$ abweichen, nicht perspektivisch korrekt plaziert werden. Die dadurch auftretenden Fehler können zum Teil durch Anpassung der Waveletgewichte korrigiert werden. Ebenso werden Änderungen in der Beleuchtung durch Anpassung der Gewichte zum großen Teil ausgeglichen.
Durch die perspektivisch bedingte, deterministische Fehlplazierung haben die angepaßten Gewichte einen Zusammenhang mit der Gesichtspose im Testbild. Es liegt daher nahe, die Gewichte $(\omega_1, \ldots, \omega_N)$ als Merkmalsvektor zu verwenden.
Die Ergebnisse der Posenbestimmung mit den angepaßten Gewichten werden in Abschnitt 5 vorgestellt.

4.3.2 Normierte Gewichte

Bei genauerer Betrachtung des Rekonstruktionsvorgangs fällt auf, daß nicht alle Wavelets einen gleich großen Beitrag zum Rekonstruktionsergebnis leisten. Der Beitrag eines Wavelets ψ_i hängt von seinem Gewicht ω_i und seiner Ausdehnung - d.h. seinen Dilatationsparametern s_x, s_y - ab.
Die Normierung der Merkmale kann insbesondere bei der progressiven Attention ausgenutzt werden, um die Wavelets eines Netzes nach ihrer “Wichtigkeit” zu sortieren.
In Abschnitt 5 werden die Resultate unter Verwendung normierter Merkmalsvektoren vorgestellt.

4.3.3 Filterantworten der reparametrisierten Wavelets

Alternativ zur Verwendung der Waveletgewichte $(\omega_1, \ldots, \omega_N)$ können stattdessen die Filterantworten an den Aufpunkten der reparametrisierten Gaborwavelets verwendet werden. Da die Reparametrisierung anhand der Grauwertkanten im Bild vorgenommen wird, besteht ein Zusammenhang zwischen den Antworten der Gaborfilter und den optimierten Waveletgewichten, welche die Grauwerte des GSW dem Quellbild annähern.
In [1] werden die Filterantworten von 64 Gaborfilterpaaren verwendet, die in einem äquidistanten 4×4-Raster in den 4 Orientierungen $(0, \frac{\pi}{4}, \frac{\pi}{2}, \frac{3\pi}{4})$ über der Gesichtsregion verteilt werden. Motiviert durch dieses Vorgehen und die bei den Gewichten gemachten Betrachtungen zur verschobenen Plazierung der Wavelets, erscheinen die Filterantworten der ungeraden Gaborwavelets als geeignete Merkmale.

4.3.4 Reparametrisierungsparameter des Superwavelets

Die Parameter des reparametrisierten Superwavelets c_x, c_y, $\frac{s_x}{s_y}$, θ scheinen sich als Netzeingabe anzubieten. In ersten Versuchen waren die Ergebnisse hervorragend - leider stellte sich bei genauerer Betrachtung heraus, daß schon der Aufpunkt des reparametrisierten Superwavelets die beiden gesuchten Winkel eindeutig festlegt. Dieser Zusammenhang ist in Abbildung 20 illustriert. Der Aufpunkt des Gaborsuperwavelets, welcher konstruktionsbedingt etwa der Nasenspitze entspricht, bewegt sich durch die feste Rotation des Puppenkopfes um einen festen Punkt im Raum auf einem Kugelsegment. Dadurch ergibt das Auftragen der erkannten 500 Positionen des abgefahrenen Winkelbereiches ($\pm 20°$ horizontal und $\pm 15°$ vertikal) ein bauchiges Rechteck. Es ist offensichtlich, daß sich beliebige Positionen durch einfache Interpolation eindeutig Neigungswinkeln zuordnen lassen. Die Verwendung der Translationsparameter c_x, c_y des Superwavelets zur Posenschätzung scheidet damit aus.

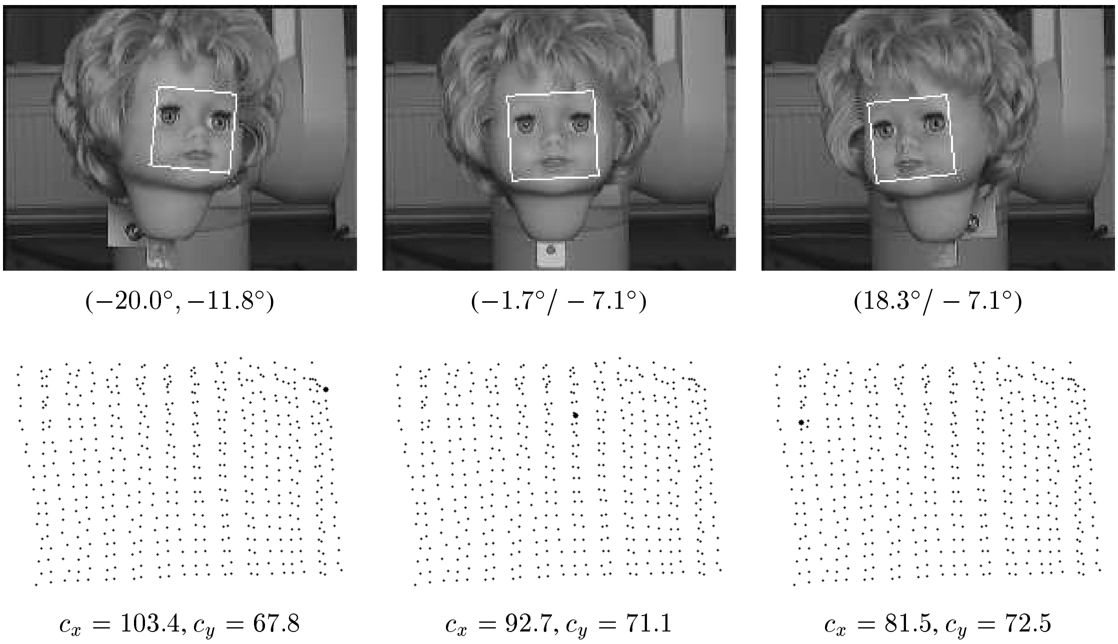


Abbildung 20: *Wenn der Puppenkopf sich um einen festen Punkt im Raum dreht, bestimmen die Positionierungsparameter c_x und c_y des reparametrisierten Gaborsuperwavelets die Gesichtspose in eindeutiger Weise. In der oberen Zeile sind die detektierten Gesichtsflächen durch ein weißes Rechteck markiert. Die in Klammern angegebenen Gradwerte beschreiben die horizontale Drehung und die vertikale Neigung. In der unteren Zeile sind die 500 während des mäanderförmigen Abfahrens des gesamten Winkelbereiches ermittelten Pixelpositionen der Aufpunkte des reparametrisierten Superwavelets aufgetragen. Die zum Bild in der oberen Zeile gehörende Position ist in dem vergrößerten Bildausschnitt darunter markiert.*

Bemerkung

Bei der in [1] verwendeten Merkmalsextraktion sind unter idealen Beleuchtungsverhältnissen auch die besten Ergebnisse zu erwarten.
Bei der hier vorgeschlagenen Methode der Reparametrisierung würden im Falle einer vollständigen affinen Transformation (mit den Freiheitsgraden Translation c_x, c_y, Dilatation s_x, s_y, Rotation α und Scherung s_{xy}) und bei konstanter diffuser Beleuchtung alle Wavelets exakt über den trainierten Merkmalen positioniert werden. Damit würden alle Filterantworten und die daran gekoppelten Gewichte konstant bleiben.
Dieses Szenario kann in dieser Arbeit aufgrund der fehlenden Scherung nicht auftreten.
Für den Fall, daß die Scherung zugelassen werden soll, müssen andere Merkmale verwendet werden. Aufgrund der angenommenen exakten Reparametrisierung, kann die Pose anhand der Scherung s_{xy} und des Rotationswinkels α in eindeutiger Weise direkt berechnet werden. Durch die direkte Berechnung würde weiterhin die zeitaufwendige Merkmalsextraktion durch Filterung und Optimierung entfallen.
Diese eigentlich elegante Methode der Posenbestimmung wird hier nicht weiter verfolgt, da wir Zugunsten der Stabilität auf die Verwendung der Scherung verzichtet haben.

4.4 Posenschätzung durch künstliche neuronale Netze

Zur Erzeugung der Trainingsdaten werden die Merkmale an äquidistant verteilten Positionen in dem Winkelbereich extrahiert, auf dem später die Pose bestimmt werden soll. Die Gesichtsposition wird einmalig zu Beginn der Merkmalsextraktion vorgegeben, danach wird das Gesicht durch Gaborwaveletnetz-Tracking (siehe Abschnitt 3.4.5) verfolgt. Die extrahierten Merkmale werden zum Offline-Training des neuronalen Netzes verwendet.
Die Testdatenmenge wird durch die Merkmalsextraktion in zufälligen Gesichtsposen erzeugt. Wie bei der Erzeugung der Trainingsdaten wird das Gesicht zu Beginn einmalig manuell lokalisiert und dann verfolgt.
Aus den extrahierten Merkmalen wird unter Verwendung von LDCS- und LSDCS-Netzen die Gesichtspose geschätzt.
Die Ergebnisse der Posenschätzung werden in Kapitel 5 ausführlich vorgestellt.

4.5 Die verwendeten künstlichen neuronalen Netze

In vielen praktischen Anwendungen stellt sich das Problem, daß die zu einem Vorgang gehörige Funktion $f : \mathbb{R}^m \longrightarrow \mathbb{R}^n$ nicht explizit bekannt ist.
Künstliche neuronale Netze liefern eine Approximationsmöglichkeit für solche - nur teilweise bekannten - mathematischen Funktionen. Voraussetzung für die Approximation dieser Funktionen ist eine hinreichend große Menge bekannter Stützstellen

$Train := \{(\mathbf{x}, \mathbf{y}) \mid \mathbf{y} = f(\mathbf{x})\}$. Die Qualität der Approximation hängt dabei von mehreren Faktoren ab, wie z.B. dem Netztyp, der Anzahl der verwendeten Neuronen und der Verteilung und Anzahl der Stützstellen.
Im folgenden werden die in dieser Arbeit verwendeten *LDCS*- und *LSDCS-Netze* in kurzer Form vorgestellt. Weitergehende Informationen finden sich in [6; 22; 1].

4.5.1 DCS-Netz

Das *DCS-Netz (Dynamic Cell Structures)* [6] ist ein dynamisches neuronales Netz, welches das gute Approximationsverhalten der RBF-Netze mit der Repräsentationskraft optimal Topologie-erhaltender Karten (*OTPMs*) kombiniert [7; 6]. Sein besonderes Merkmal ist die Fähigkeit, die Topologie des Eingaberaumes zu erlernen, was es zu einer effizienteren Ausgabeberechnung und Adaption gegenüber den RBF-Netzen befähigt.

Das Netz läßt sich als Graph-Struktur - bestehend aus einer Menge $\mathbf{V}$ von *Neuronen* und einer symmetrischen Kantenrelation $\mathbf{E}$ - darstellen, die beschreibt, welche Neuronen benachbart sind:

$$\mathbf{V} = \{1, \ldots, N\} \text{ und}$$

$$\mathbf{E} = \{(i,j) \mid \text{Neuron } i \text{ und Neuron } j \text{ sind benachbart}\}$$

Auf dieser Graph-Struktur läßt sich eine Nachbarschaft für ein Neuron $i \in \mathbf{V}$ definieren:

$$\mathrm{Nh}(i) = \{j \in \mathbf{V} \mid (i,j) \in \mathbf{E}\} \tag{15}$$

Das Neuron, welches dem Eingabevektor $\mathbf{x} \in \mathbb{R}^{\mathrm{m}}$ am nächsten liegt, wird mit bmu($\mathbf{x}$) (*best matching unit*) und das zweitnächste Neuron mit second($\mathbf{x}$) bezeichnet. Es gilt:

$$\forall i \in \mathbf{V} : \|\mathbf{w}_i - \mathbf{x}\|_2 \geq \|\mathbf{w}_{\mathrm{bmu}(\mathbf{x})} - \mathbf{x}\|_2$$

und

$$\forall i \in \mathbf{V} \setminus \{\mathrm{bmu}(\mathbf{x})\} : \|\mathbf{w}_i - \mathbf{x}\|_2 \geq \|\mathbf{w}_{\mathrm{second}(\mathbf{x})} - \mathbf{x}\|_2 \,.$$

Ein Neuron wird durch das Zentrum seines *rezeptiven Feldes* $\mathbf{w}_i \in \mathbb{R}^{\mathrm{m}}$, d.h. durch seine Position im Eingaberaum und seine *Aktivierungsfunktion*

$$a_i(\mathbf{x}) = \exp\left(-\frac{\|\mathbf{w}_i - \mathbf{x}\|_2^2}{\sigma_i^2}\right). \tag{16}$$

beschrieben. Sinnvollerweise soll ein Neuron nur auf Stimuli reagieren, die in seiner Nähe liegen. Als Aktivierungsfunktion wird eine Gaußfunktion verwendet, d.h. die Aktivierung eines Neurons nimmt mit der Entfernung des Stimulus vom Zentrum des rezeptiven Feldes ab. Bei der Gaußfunktion ist die Höhe der Aktivierung symmetrisch und hängt nur vom Abstand des Eingabevektors $\mathbf{x} \in \mathbb{R}^{\mathrm{m}}$ zum Zentrum des rezeptiven Feldes $\mathbf{w}_i \in \mathbb{R}^{\mathrm{m}}$ des Neurons ab.

Der Durchmesser des rezeptiven Feldes wird für jedes Neuron getrennt durch $\sigma_i \in \mathbb{R}$ bestimmt. Es ist zweckmäßig, σ_i so zu wählen, daß sich die Einzugsbereiche benachbarter Neuronen zwar überlappen, die Aktivierung des Neurons i aber für einen Eingabevektor $\mathbf{x}$ in der Nähe des Zentrums des rezeptiven Feldes von Neuron $j \in \mathbf{V}$ nur noch sehr gering ist (für $i \neq j$). Die Überlappung darf aber nicht zu klein gewählt werden, da sonst die Aktivierung für Eingabewerte in der Mitte zweier Neuronen zu gering wird und zu fehlerhaften Netzantworten führt.
Damit die Aktivierungsfunktionen während des Lernvorgangs optimal bleiben, müssen sie sich jeder Änderung der Struktur anpassen, d.h. ebenfalls dynamisch sein. Dazu wird Gleichung (16) folgendermaßen erweitert:

$$a_i(\mathbf{x}) = \begin{cases} \exp\left(-\frac{\|\mathbf{w}_i - \mathbf{x}\|_2^2}{\sigma_i^2} \cdot (-\ln p)\right) \cdot \frac{1}{\mathrm{sumAct}(\mathbf{x})} & : \ i \in (\{\mathrm{bmu}(\mathbf{x})\} \cup \mathrm{Nh}(\mathrm{bmu}(\mathbf{x}))) \\ 0 & : \ \mathrm{sonst} \end{cases} \tag{17}$$

und für die Normierung der Gesamtaktivierung auf 1:

$$\mathrm{sumAct}(\mathbf{x}) = \sum_{j \in \{\mathrm{bmu}(\mathbf{x})\} \cup \mathrm{Nh}(\mathrm{bmu}(\mathbf{x}))} \exp\left(-\frac{\|\mathbf{w}_j - \mathbf{x}\|_2^2}{\sigma_j^2} \cdot (-\ln p)\right) .$$

Der konstante Faktor $p \in [0,1]$ bestimmt, wie stark sich die Aktivierungsfunktionen benachbarter Neuronen überlappen, wenn die Neuronen gleichmäßig verteilt sind. Die Fallunterscheidung in (17) hat zur Folge, daß für den Eingabevektor $\mathbf{x} \in \mathbb{R}^m$ nur $\mathrm{bmu}(\mathbf{x})$ und die Nachbarn von $\mathrm{bmu}(\mathbf{x})$ aktiviert werden, d.h. die Approximation findet lokal statt.

Jedem Neuron ist ein Ausgabevektor $\mathbf{o}_i \in \mathbb{R}^n$ zugeordnet. Die Ausgabe $\mathbf{y}_i \in \mathbb{R}^n$ des Neurons i auf den Eingabevektor $\mathbf{x} \in \mathbb{R}^m$ ergibt sich zu

$$\mathbf{y}_i(\mathbf{x}) = a_i(\mathbf{x})\mathbf{o}_i . \tag{18}$$

Die Netzaktivierung ist die Summe der Einzelaktivierungen:

$$\mathbf{y}_{Netz}(\mathbf{x}) = \sum_{i \in \mathbf{V}} a_i(\mathbf{x})\mathbf{o}_i \tag{19}$$

Training des DCS-Netzes

Das Offline-Training des Netzes unterteilt sich in zwei Schritte:

Lernen der Topologie

Im ersten Schritt wird die Topologie des Eingaberaumes erlernt. Es wird dort ein Neuron im Eingaberaum eingefügt, wo der Approximationsfehler maximal ist. Danach werden die Zentren $\mathbf{w}_i$ aller Neuronen durch den *k-means*-Algorithmus adaptiert. Abschließend werden die Verbindungen zwischen den Neuronen neu erzeugt, indem zu jedem Trainingspaar $(\mathbf{x}_t, \mathbf{y}_t) \in Train$ die Neuronen bmu($\mathbf{x}_t$) und second($\mathbf{x}_t$) verbunden werden.

Lernen der Ausgaben

Im zweiten Schritt werden die Netzausgaben anhand der Trainingsmenge gelernt. Idealerweise sollte die Netzausgabe zu einem $\mathbf{x}_t \in \mathbb{R}^m$ gerade $f(\mathbf{x}_t) = \mathbf{y}_t$ sein. Dieser Umstand läßt sich als lineares Ausgleichsproblem formulieren und lösen:

$$\begin{pmatrix} a_1(\mathbf{x}_1) & \dots & a_N(\mathbf{x}_1) \\ \vdots & & \vdots \\ a_1(\mathbf{x}_{|Train|}) & \dots & a_N(\mathbf{x}_{|Train|}) \end{pmatrix} \begin{pmatrix} \mathbf{o}_1^T \\ \vdots \\ \mathbf{o}_{|Train|}^T \end{pmatrix} = \begin{pmatrix} \mathbf{y}_1^T \\ \vdots \\ \mathbf{y}_{|Train|}^T \end{pmatrix} \tag{20}$$

Falls die Anzahl der Trainingsdaten $|Train|$ kleiner als die Anzahl der Neuronen N ist, ist das Gleichungssystem unterbestimmt und es existiert keine eindeutige Lösung, sondern nur ein Lösungsraum. In diesem Fall werden die Trainingsdaten “auswendig gelernt” und das Approximationsverhalten ist i.a. schlecht.

4.5.2 DCS-Netz mit linearer Erweiterung (LDCS)

Unter der Annahme, daß die zu approximierende Funktion $f : \mathbb{R}^m \longrightarrow \mathbb{R}^n$ zwischen einem Neuronenzentrum und dem am nächsten liegenden Eingabedatum linear approximierbar ist, können die partiellen Ableitungen von f in den Zentren der rezeptiven Felder berechnet und zur Verbesserung der Approximation verwendet werden.
Jedem Neuron wird zusätzlich zum Ausgabevektor $\mathbf{o}_i$ die $n \times m$-Jacobi-Matrix $\mathbf{J}_i$ zu f an der Stelle $\mathbf{w}_i$ zugeordnet.
Die Berechnung der Netzausgabe wird zu

$$\mathbf{y}_{Netz}(\mathbf{x}) = \sum_{i \in \mathbf{V}} a_i(\mathbf{x}) \left(\mathbf{o}_i + \mathbf{J}_i(\mathbf{x} - \mathbf{w}_i) \right) \tag{21}$$

erweitert. Dieser Netztyp wird als *LDCS-Netz* bezeichnet.

Das Lernen der Topologie und der Netzausgaben verläuft analog zum DCS-Netz.

4.5.3 SDCS-Netz

Das *SDCS-Netz (Subspace Dynamic Cell Structures)* ist eine Verbesserung des DCS-Netzes. Der Unterschied zu den DCS-Netzen besteht darin, daß bei den SDCS-Netzen die Approximation nicht im Eingaberaum, sondern in einem niederdimensionalen Unterraum stattfindet.
Die Grundidee dabei ist, daß in fast allen Anwendungen die Dimension des Eingaberaumes von der Art der Datenextraktion abhängt - in dieser Arbeit z.B. von der Anzahl

der verwendeten Wavelets. Die eigentliche intrinsische Dimension des Eingaberaumes ist aber unabhängig von der Datenextraktion. Meistens ist sie deutlich kleiner als die Dimension des Eingaberaumes[14].
Durch Verlagerung der Berechnungen in lokale intrinsische Unterräume wird bei hochdimensionalen Eingaberäumen eine deutliche Effizienzverbesserung erreicht.
Insbesondere wird durch die Projektion der Eingabedaten auf intrinsische Unterräume der Rauschanteil unterdrückt.

Aufbau des SDCS-Netzes

Der Aufbau des SDCS-Netzes ähnelt dem des DCS-Netzes. In einem ersten Schritt wird die Topologie des Eingaberaumes durch ein DCS-Netz gelernt. Dann wird jedem Neuron zusätzlich eine Matrix $\mathbf{E}_i$ der normierten Eigenvektoren und ein Vektor $\mathbf{e}_i$ der zugehörigen Eigenwerte zugeordnet. Diese erhält man durch Hauptkomponentenanalyse der Korrelationsmatrix $\mathbf{C}_i$ des jeweiligen Neurons.
Die Korrelationsmatrix zum Neuron i wird definiert durch

$$\mathbf{C}_i = \frac{1}{\mid \mathrm{Nh}(i) \mid} \mathbf{W}^T \mathbf{W} \tag{22}$$

mit

$$\mathbf{W} = \begin{pmatrix} (\mathbf{w}_{j_1} - \mathbf{w}_i)^T \\ \vdots \\ (\mathbf{w}_{j_k} - \mathbf{w}_i)^T \end{pmatrix} \quad \text{wobei } \mathrm{Nh}(i) = \{j_1, \ldots, j_k\} = \mathrm{Nh}(i). \tag{23}$$

Die Eigenvektoren von $\mathbf{C}_i$ spannen in der Umgebung des zugehörigen Neurons die gesuchte Mannigfaltigkeit auf. Für die Berechnung der Matrix der Eigenvektoren $\mathbf{E}_i$ und des Vektors der Eigenwerte $\mathbf{e}_i$ haben Murase und Nayar in [18] ein effizientes Verfahren vorgestellt.
Die Größe der Eigenwerte erlaubt Rückschlüsse auf die Varianz der Eingabedaten in Richtung der zugehörigen Eigenvektoren. Man geht davon aus, daß alle unverrauschten Eingabevektoren im Einzugsbereich des Neurons innerhalb der Mannigfaltigkeit liegen. Kleine Eigenwerte deuten auf eine geringe Varianz hin. Es wird angenommen, daß diese nur durch Rauschen verursacht wird. Daher werden alle Eigenvektoren E_{i_j} aus $\mathbf{E}_i$ gestrichen, deren zugehöriger Eigenwert e_{i_j} unterhalb einer Schwelle ε_i liegt. Der Schwellwert wird üblicherweise als ein Prozent des maximalen Eigenwertes gewählt. Man erhält eine neue Matrix $\tilde{\mathbf{E}}_i$, welche die Basis des lokalen intrinsischen Unterraumes in der Umgebung des Neurons i darstellt.
Die Aktivierung wird nicht mehr anhand der Entfernungen im $\mathrm{I\!R}^{\mathrm{m}}$, sondern im durch $\tilde{\mathbf{E}}_i$ aufgespannten Unterraum durchgeführt. Die Aktivierungsfunktion wird zu

[14] Die *intrinsische Dimensionalität* einer Mannigfaltigkeit ist in [7] als die minimale Anzahl von Freiheitsgraden definiert, die nötig ist, diese zu generieren.

$$a_i(\mathbf{x}) = \begin{cases} \exp\left(-\frac{\|\tilde{\mathbf{E}}_i\mathbf{w}_i - \tilde{\mathbf{E}}_i\mathbf{x}\|_2^2}{\sigma_i^2} \cdot (-\ln p)\right) \cdot \frac{1}{\mathrm{sumAct}(\mathbf{x})} & : \; i \in (\{\mathrm{bmu}(\mathbf{x})\} \cup \mathrm{Nh}(\mathrm{bmu}(\mathbf{x}))) \\ 0 & : \; \mathrm{sonst} \end{cases} \tag{24}$$

mit

$$\sigma_i^2 = \frac{\sum\limits_{j \in \mathrm{Nh}(i)} \|\tilde{\mathbf{E}}_i\mathbf{w}_i - \tilde{\mathbf{E}}_i\mathbf{w}_j\|_2^2}{\mid \mathrm{Nh}(i) \mid} \tag{25}$$

und

$$\mathrm{sumAct}(\mathbf{x}) = \sum_{j \in \{\mathrm{bmu}(\mathbf{x})\} \cup \mathrm{Nh}(\mathrm{bmu}(\mathbf{x}))} \exp\left(-\frac{\|\tilde{\mathbf{E}}_i\mathbf{w}_j - \tilde{\mathbf{E}}_i\mathbf{x}\|_2^2}{\sigma_j^2} \cdot (-\ln p)\right) .$$

Das Lernen der Netzausgaben geschieht analog zum DCS-Netz mit dem Unterschied, daß die Entfernungen zwischen den Eingabevektoren und den Neuronenzentren in den lokalen intrinsischen Unterräumen berechnet werden.

4.5.4 SDCS-Netz mit linearer Erweiterung (LSDCS)

Wie beim DCS-Netz kann das SDCS-Netz unter der Annahme, daß die zu approximierende Funktion in den Zentren der rezeptiven Felder linear approximierbar ist, um die Verwendung der partiellen Ableitungen in den Neuronenzentren erweitert werden. Im Gegensatz zum LDCS-Netz findet die Berechnung der Jacobi-Matrizen nicht im $\mathbb{R}^m$, sondern in den lokalen intrinsischen Unterräumen statt. Daraus resultiert eine große Effizienzsteigerung, da die Dimension der intrinsischen Unterräume i.a. deutlich kleiner als die Dimension des Eingaberaumes ist.
Die Netzausgabe ergibt sich zu:

$$\mathbf{y}_{Netz}(\mathbf{x}) = \sum_{i \in \mathbf{V}} a_i(\mathbf{x}) \left(\mathbf{o}_i + \mathbf{J}_i(\tilde{\mathbf{E}}_i\mathbf{x} - \tilde{\mathbf{E}}_i\mathbf{w}_i)\right) \tag{26}$$

Diese Erweiterung des SDCS-Netzes wird als *LSDCS-Netz* bezeichnet.
Das Lernen der Netzausgaben verläuft analog zum SDCS-Netz.

Verwendete Netze zur Posenbestimmung

In [22; 1] wurde übereinstimmend festgestellt, daß die Verwendung partieller Ableitungen die Approximationsgüte wesentlich verbessert, d.h. daß LDCS- und LSDCS-Netze den Netzen ohne lineare Erweiterung deutlich überlegen sind. Daher haben wir in dieser Arbeit die Posenschätzung nur mit LDCS- und LSDCS-Netzen untersucht.

5 Ergebnisse

In diesem Kapitel werden die experimentellen Ergebnisse der Posenbestimmung unter Verwendung von Gaborwaveletnetzen präsentiert. Der erste Teil dieses Kapitels beschäftigt sich der Erläuterung der Versuchsdurchführung (5.1). Dann wird die Vorgehensweise in [1] mit dem hier verwendeten Verfahren verglichen (5.2). Es folgen erste Untersuchungen über die Eignung der vorgestellten Methode mit manueller Vorplazierung des Superwavelets (5.3). Im weiteren werden die Ergebnisse der Online-Versuche vorgestellt und diskutiert (5.4). Im letzten Teil des Kapitels werden verschiedene Eigenschaften anhand von Offline-Experimenten näher untersucht (5.5). Da diese Arbeit durch die guten Ergebnisse in [1] motiviert ist, liegt ein besonderes Augenmerk auf dem Vergleich der Ergebnisse.

5.1 Versuchsablauf

Die Versuchsdurchführung ist zweigeteilt. Der erste Teil des Versuchs dient der Erzeugung der Trainingsdaten, der zweite Teil dient der Gewinnung der Testdaten. Falls nicht anders angegeben, umfaßt die Trainingsmenge 500 Datensätze und die Testmenge 100 Datensätze. Der Umfang der Datensätze entspricht dem aus [1]. Die große Zahl an Trainingsdaten hat mehrere Gründe. Zum einen wird durch die große Zahl der Trainingsdaten ein Auswendiglernen vermieden[15] und zum anderen profitiert die Qualität der Posenschätzung von der Anzahl der Beispiele.
Bis auf den ersten Versuch erfolgt die Erzeugung der Trainings- und Testdaten nach manueller Initialisierung automatisch.

Erzeugung der Trainingsdaten

Bei den Trainingsdaten fährt der Roboterarm den verwendeten Winkelbereich von $\pm 20°$ horizontal und $\pm 15°$ vertikal in 25 Schritten horizontal sowie 20 Schritten vertikal mäanderförmig ab. Die Schrittweite beträgt somit $1.64°$ horizontal und $1.55°$ vertikal. An den Meßpunkten werden der Roboterarm angehalten und die Merkmale extrahiert.
Um die Gefahr von Instabilitäten zu senken, wurden die Dilatationsparameter des Superwavelets gekoppelt, d.h. es gilt $s_x = s_y$. Ferner wurde keine Scherung zugelassen.
Der Zeitbedarf für die Merkmalsextraktion hängt von den extrahierten Merkmalen ab. Bei der Verwendung von 16 Wavelets dauert die Extraktion der optimierten Gewichte unter einer Sekunde.

[15]Durch die Verwendung der partiellen Ableitungen in den Zentrumsvektoren der Neuronen ist bei 500 Trainingsbeispielen und einem 52-dimensionalen Eingabevektor das Gleichungssystem beim LDCS-Netz bereits bei 10 Neuronen unterbestimmt.

Erzeugung der Testdaten

Die Testdaten werden wie die Trainingsdaten erzeugt, mit dem Unterschied, daß zufällige Positionen im erlaubten Winkelbereich angefahren werden.

Netztraining und Auswertung

Für die Simulation der LDCS- und LSDCS-Netze wird die gleiche Simulationssoftware wie in [1; 6; 22] verwendet. Das Training der neuronalen Netze erfolgt offline nach der Erzeugung der Test- und Trainingsdaten.
Das Training beginnt mit einem Neuron. Schrittweise werden solange Neuronen in das Netz eingefügt, bis entweder das Gleichungssystem (20) unterbestimmt wird, oder der Fehler über einen längeren Zeitraum stagniert oder ansteigt.
Als Aktivierungsfunktion wurde für beide Netztypen die in Gleichung (17) vorgestellte Variante verwendet, d.h. es werden die Best Matching Unit und ihre Nachbarn aktiviert. Diese Wahl hat sich in [1] als vorteilhaft erwiesen und sich in unseren Experimenten bestätigt.

5.2 Vergleich der Vorgehensweise

In [1] werden die Gesichter durch eine aufwendige und anfällige Farbsegmentierung lokalisiert und auf eine konstante Größe skaliert. Aus den skalierten Bildern werden an rasterförmig angeordneten Aufpunkten durch Faltung mit Gaborfiltern Merkmalsvektoren extrahiert, die als Eingabe für ein künstliches neuronales Netz dienen. Die Aufpunkte der Gaborfilter sind dabei fest verteilt, d.h. es findet keine Anpassung an besondere Merkmale des Eingabebildes statt.
Im Gegensatz zur Vorgehensweise in [1] wird in dieser Arbeit nicht das Kamerabild skaliert sondern das Gaborsuperwavelet $\mathbf{\Psi_n}$. Daher gibt es keinen Informationsverlust durch Skalierung, d.h. es kann die gesamte verfügbare Bildinformation genutzt werden. Ferner wird das Superwavelet mit Subpixelgenauigkeit reparametrisiert, wie in Abschnitt 2.4.1 beschrieben. Dadurch kommen die Filter an den trainierten Merkmalen im Gesicht zu liegen. Dies ist ein wichtiger Vorteil im Vergleich zur starren Verteilung der Gaborfilter.
Ein weiterer Vorteil der hier benutzten Vorgehensweise liegt in der Robustheit des Versuchsaufbaus. Während in [1] die Farbsegmentierung extrem kritisch war, ist der aktive Tracking-Algorithmus (siehe Abschnitt 3) durch Verwendung des YUV-Farbmodells robust gegenüber homogenen Helligkeitsänderungen.
Ferner wird das Gaborsuperwavelet durch den Matching-Algorithmus auch bei einer Anfangsabweichung noch optimal reparametrisiert (siehe Abschnitt 2.4.1). Diese Verbesserungen in der Merkmalsextraktion geben Anlaß zur Hoffnung, die Ergebnisse aus [1] noch zu verbessern.
In dieser Arbeit wurde der erlaubte Winkelbereich auf $\pm 20^\circ$ horizontale Drehung und $\pm 15^\circ$ vertikale Neigung eingeschränkt, da ansonsten durch Verdeckung von Merk-

malen das Matching erschwert ist. Dieser Bereich genügt für die in der Einleitung genannten Anwendungen aber völlig.

5.3 Ergebnisse mit manueller Positionsvorgabe

In einem ersten Schritt soll die prinzipielle Eignung der Gaborwaveletnetze zur Bestimmung der Gesichtspose verifiziert werden.
Da für ein erfolgreiches Matching eine gute Initialisierung notwendig ist, wurde in jedem Bild die Startposition für den Matching-Algorithmus manuell vorgegeben. Dazu wurde in jedem Bild die Nasenspitze markiert, welche ungefähr mit dem Aufpunkt des Gaborsuperwavelets übereinstimmt. Die Orientierung wurde mit $0°$ und die Skalierungen s_x, s_y mit einem festen Wert initialisiert. Dieses ungewöhnliche Vorgehen war sinnvoll, da es an diesem Punkt nur um die prinzipielle Eignung ging. Wir weisen darauf hin, daß *nicht die exakte Lösung vorgegeben wurde*, sondern nur eine Umgebung, in der der Algorithmus konvergiert (siehe Abschnitt 2.4.1).
Die genaue Position des Superwavelets wird durch die Reparametrisierung subpixelgenau gefunden, ebenso wie die Skala und die Orientierung.
Die manuelle Vorgabe ist an dieser Stelle vertretbar, da sich die Konvergenz-Umgebung durch aktive Gesichtsverfolgung gewinnen läßt, wie in Abschnitt 3 beschrieben.
Zur Posenschätzung werden nur die Gewichte $(\omega_1, \ldots, \omega_N)$ des reparametrisierten Gaborsuperwavelets $\mathbf{\Psi_n}$ benutzt.
Es ist zu erwarten, daß die Ergebnisse durch die manuelle Vorpositionierung möglicherweise besser ausfallen als bei einer autonomen Initialisierung. Die Resultate dieser Versuchsreihe sind somit als Referenzwerte zu verstehen.
Die Ergebnisse der Versuchsreihe sind in Tabelle 2 dargestellt. Mit diesen Ergebnissen ist die in der Einleitung vermutete Eignung der Gaborwaveletnetze zur Posenschätzung bestätigt. Die Ergebnisse sind noch deutlich besser als die ohnehin schon guten Resultate in [1], die zum Vergleich im unteren Teil der Tabelle aufgeführt sind.
Wie intuitiv erwartet, wird die Posenschätzung umso genauer, je mehr Wavelets verwendet werden. Es fällt auf, daß das Offline-LDCS-Netz selbst mit nur 16 Eingabewerten eine erheblich bessere Posenschätzung liefert als das LSDCS-Netz mit 52 Eingabewerten, wobei das LSDCS-Netz nur einen kleinen Gewinn aus dem größeren Eingaberaum ziehen kann.
Nachdem an der prinzipiellen Eignung der Methode kein Zweifel besteht, bleibt zu zeigen, daß sich ähnlich gute Ergebnisse auch unter praxisorientierteren Bedingungen, d.h. ohne die manuelle Vorgabe der Startposition in jedem Bild, erzielen lassen.

5.4 Ergebnisse der Online-Versuche

In diesem Abschnitt wird die manuelle Vorgabe der Startparameter für den Matchingprozeß in jedem Bild durch aktive Gesichtsverfolgung ersetzt (siehe Abschnitt 3).

Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	0.30	0.81
	LSDCS	0.70	1.60
52 Waveletgewichte	LDCS	0.19	0.49
	LSDCS	0.55	1.90
Ergebnisse von E. Ábráham-Mumm			
Filterantworten von 64 Gaborfilterpaaren	LDCS	0.50	1.24
	LSDCS	0.64	1.88

Tabelle 2: *Testergebnisse der Posenschätzung mit manueller Initialisierung für jedes Bild. In der Tabelle sind jeweils der mittlere und der maximale Testfehler in Grad/Drehrichtung angegeben. Im unteren Teil der Tabelle sind zum Vergleich die Ergebnisse aus der Arbeit von E. Ábráham-Mumm aufgeführt.*

In den folgenden Experimenten werden die Auswirkungen verschiedener Varianten der automatischen Wahl der Startparameter für den Matchingprozeß untersucht. Die Gesichtsverfolgung findet dabei unter Echtzeitbedingungen statt.

5.4.1 Matching mit festem Startwert für Skala und Rotationswinkel

Die Merkmalsextraktion wird mit einem Rotationswinkel von 0° und einem konstanten Startwert für die Skalierung gestartet, d.h. nur der Startwert für die Translation wird aus der Gesichtsverfolgung übernommen. Mit dieser Wahl soll ein möglicher Einfluß der Reihenfolge, in der die Positionen angefahren werden, ausgeschlossen werden. Die Ergebnisse sind in Tabelle 3 dargestellt. Wie oben bereits vermutet, fallen die Ergebnisse etwas schlechter aus als bei der manuellen Initialisierung, sind aber immer noch deutlich besser als in [1].

Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	0.39	0.80
	LSDCS	0.68	2.25
52 Waveletgewichte	LDCS	0.28	0.55
	LSDCS	0.59	1.49

Tabelle 3: *Testergebnisse der Posenschätzung mit automatischer Initialisierung für jedes Bild. In der Tabelle sind jeweils der mittlere und der maximale Testfehler in Grad/Drehrichtung angegeben. Der Matchingalgorithmus wurde mit* $\alpha = 0^\circ$ *und* $s_x = s_y = \mathrm{const}$ *initialisiert.*

5.4.2 Matching mit Skala und Rotationswinkel des vorherigen Bildes

In diesem Versuch wurde auf die Initialisierung mit konstantem Winkel und konstanter Skala verzichtet. Stattdessen wurde der Winkel α_{t-1} des letzten Bildes $\mathbf{I}_{t-1}$ verwendet.

Die Skalen s_x und s_y wurden mit den um 10% vergrößerten Werten des letzten Bildes initialisiert. Die Modifikation um 10% hat den Sinn, daß das Matchingverfahren erfahrungsgemäß bei einem zu kleinen Suchbereich eher falsch konvergiert, als bei einem zu großen Bereich. Wie aus Tabelle 4 zu entnehmen, sind Ergebnisse im Rahmen der Meßgenauigkeit mit denen aus (5.4.1) identisch. Da die hier gewählte Methode durch den Verzicht auf eine feste Skala gegenüber der Initialisierung mit konstanten Werten unabhängig von der Entfernung des Objekts ist, wird sie in den weiteren Versuchen verwendet. Die Verwendung des Rotationswinkels α_{t-1} erhöht die Stabilität bei extremer Rotation und beschleunigt den Matchingprozeß.

Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	0.39	0.87
	LSDCS	0.69	2.28
52 Waveletgewichte	LDCS	0.29	0.69
	LSDCS	0.58	1.35

Tabelle 4: *Der Matchingalgorithmus wurde mit* $\alpha_t = \alpha_{t-1}$ *und* $s_{x_t} = 0.9 \cdot s_{x_{t-1}}$, $s_{y_t} = 0.9 \cdot s_{y_{t-1}}$ *initialisiert. In der Tabelle sind jeweils der mittlere und der maximale Testfehler in Grad/Drehrichtung angegeben.*

5.4.3 Matching mit Adaption des Mittelwertes

In den bisherigen Versuchen wurde der Mittelwert des Gaborsuperwavelets unter der Annahme einer konstanten Beleuchtung *nicht* auf das zu matchende Bild angepaßt. Weicht aber die Beleuchtung während der Versuchsdurchführung von der bei der Erzeugung des GSW ab, wird der Gleichanteil nicht korrekt kompensiert. Daher wird in diesem Versuch der Gleichanteil vor jedem Matching neu bestimmt. Die Ergebnisse in Tabelle 5 zeigen eine wahrnehmbare Verbesserung des mittleren Testfehlers. Wie schon in den bisherigen Versuchen ist auch hier der Testfehler des LDCS-Netzes sogar mit 16 Gewichten noch signifikant kleiner, als der des LSDCS-Netzes mit 52 Gewichten.

Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	0.36	0.78
	LSDCS	0.71	1.51
52 Waveletgewichte	LDCS	0.23	0.53
	LSDCS	0.54	1.51

Tabelle 5: *Der Mittelwert wurde vor jedem Matching neu angepaßt. In der Tabelle sind jeweils der mittlere und der maximale Testfehler in Grad/Drehrichtung angegeben.*

5.4.4 Posenbestimmung mit Filterantworten

Wie in Abschnitt 4 erwähnt, existieren noch weitere geeignete Merkmale, nämlich die Filterantworten der ungeraden Gaborwavelets des reparametrisierten Superwavelets. Daher werden in einer Versuchsreihe sowohl die Gewichte als auch die Filterantworten als Merkmale extrahiert. Die verblüffenden Ergebnisse sind in Tabelle 6 dargestellt. Während das LDCS-Netz offenbar besser mit den Gewichten umgehen kann, erzielt das LSDCS-Netz mit den Filterantworten bessere Ergebnisse als mit den Waveletgewichten.
Die Ursache dafür liegt darin, daß sich die Gewichte aus dem Zusammenspiel vieler benachbarter Wavelets ergeben, d.h. eine globalere Struktur widerspiegeln. Die Filterantworten sind hingegen vollkommen unabhängig voneinander, kodieren also nur lokale Informationen. Dies kommt der Approximation in lokalen intrinsischen Unterräumen entgegen.

Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	0.37	0.84
	LSDCS	0.69	1.89
52 Waveletgewichte	LDCS	0.26	0.65
	LSDCS	0.54	1.51
16 Filterantworten	LDCS	0.50	1.01
	LSDCS	0.52	0.98
52 Filterantworten	LDCS	0.41	0.92
	LSDCS	0.51	0.93

Tabelle 6: *Direkter Vergleich unter identischen Bedingungen von Gaborwaveletgewichten und Filterantworten als Netzeingaben.*

5.4.5 Verringerung der Anzahl der Trainingsdaten

Da die meiste Zeit des Trainings für die Aufnahme der Trainingsbilder verbraucht wird, wird die Zahl der Trainingsbilder testweise von 500 auf 150 gesenkt. Damit beträgt die Schrittweite $2.73°$ horizontal und $3.1°$ vertikal. Es ist zu erwarten, daß die Ergebnisse schlechter ausfallen als mit der größeren Trainingsmenge, da die neuronalen Netze die Daten möglicherweise schneller auswendig lernen und damit schlecht generalisieren werden. Die Experimente konnten aufgrund der kleinen Trainingsmenge nur mit 16-dimensionalen Eingabevektoren durchgeführt werden, da sonst das Gleichungssystem (20) unterbestimmt war. Die in Tabelle 7 aufgeführten Ergebnisse bestätigen diese Vermutung. Allerdings fällt auf, daß das LSDCS-Netz mit den 16 Filterantworten als Eingabe nahezu die gleichen Ergebnisse erzielt wie schon im vorherigen Versuch (vgl. Tabelle 6). Die Ursache für dieses bemerkenswerte Ergebnis dürfte an der Projektion der Eingabedaten auf die lokalen intrinsischen Unterräume liegen.

Verkleinerte Trainingsmenge			
Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	0.70	1.63
	LSDCS	1.23	3.33
16 Filterantworten	LDCS	0.61	1.30
	LSDCS	0.54	1.16

Tabelle 7: *Ergebnisse mit 16 Gewichten bzw. 16 Filterantworten auf einer auf 150 Bilder verkleinerten Trainingsmenge.*

5.4.6 Erhöhung der Anzahl der Trainingsdaten

Es stellt sich die Frage, ob durch eine Vergrößerung der Trainingsmenge eine Verbesserung der Genauigkeit erreicht werden kann. Zu diesem Zweck wurde eine Trainingsmenge von 1500 Bildern aufgenommen. Aufgrund der großen Datenmenge war das Training extrem langwierig, so daß in Tabelle 8 nur die Ergebnisse unter Verwendung der Gewichte angegeben werden. Die Ergebnisse des LSDCS-Netzes ändern sich nur marginal im Vergleich zu 500 Trainingsbildern. Beim LDCS-Netz zeigt sich ein Overfitting. Die Trainingsdaten werden auswendig gelernt und der mittlere quadratische Fehler verdoppelt sich im Vergleich zu einer Trainingsmenge von 500 Bildern.
Die weiteren Versuche werden wieder mit 500 Trainingsbildern durchgeführt, um die erreichbare Genauigkeit nicht unnötig durch Overfitting zu verringern.

Vergrößerte Trainingsmenge			
Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
52 Waveletgewichte	LDCS	0.40	1.01
	LSDCS	0.52	1.21

Tabelle 8: *Ergebnisse mit 52 Gewichten auf einer auf 1500 Bilder vergrößerten Trainingsmenge.*

5.4.7 Erweiterter Winkelbereich

Da der für die Experimente gewählte Winkelbereich von $\pm 20°$ horizontal und $\pm 15°$ vertikal für einige Anwendungen zu knapp ist, wurde ein erweiterter Winkelbereich von $\pm 30°$ horizontal und $\pm 25°$ vertikal untersucht. Es wurden weiterhin 500 Trainingsbilder verwendet. Somit betrug die Schrittweite $2.44°$ horizontal und $2.55°$ vertikal.
In Extremposen (z.B. Gesicht nach ganz rechts unten geneigt) erscheinen die Gesichtsmerkmale bereits deutlich deformiert, wie in Abbildung 21 dargestellt. Die Ergebnisse sind in Tabelle 9 zusammengefaßt.
Die Waveletgewichte sind im erweiterten Winkelbereich aufgrund der deutlichen Deformation des Gesicht im Kamerabild nicht mehr für die Posenschätzung geeignet. Der

Abbildung 21: *Das Abbild des 30° nach rechts und 25° nach unten blickenden Gesichtes ist deutlich deformiert.*

Fehler bei Verwendung des LDCS-Netzes steigt bei 16 Gewichten um den Faktor vier und bei 52 Gewichten sogar um den Faktor sechs an.
Die Verwendung der Filterantworten liefert hingegen weiterhin brauchbare Ergebnisse. Besonders hervorzuheben ist das Ergebnis des LSDCS-Netzes mit 52 Filterantworten, welches den gleichen mittleren quadratischen Fehler wie auf dem kleineren Winkelbereich erzielt (vgl. Tabelle 6). Die Überlegenheit der Filterantworten gegenüber den Waveletgewichten in dieser Anwendung resultiert aus der vergleichsweise schlechten Positionierung der Wavelets, die in den Extremposen weit von den durch sie repräsentierten Merkmalen zu liegen kommen. Dies wird durch schlechte Sichtbarkeit der Merkmale und die fehlende Scherung verursacht. Die Gewichte werden bei der Optimierung so angepaßt, daß sie den Fehler der Rekonstruktion minimieren. Da die Wavelets weit von den kodierten Merkmalen entfernt liegen, geht der gelernte Bezug zwischen den einzelnen Wavelets (d.h. die Gesichtsstruktur) verloren. Das LSDCS-Netz hat aufgrund seiner lokalen Arbeitsweise weniger Probleme mit Strukturverlust und verschlechtert sich auf den Gewichten nur geringfügig (vgl. Tabelle 6).

Winkelbereich $(\pm 30^\circ / \pm 25^\circ)$			
Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	1.34	3.60
	LSDCS	0.95	2.98
52 Waveletgewichte	LDCS	1.41	3.33
	LSDCS	0.87	2.41
16 Filterantworten	LDCS	0.77	2.16
	LSDCS	0.62	1.58
52 Filterantworten	LDCS	0.76	1.73
	LSDCS	0.48	1.11

Tabelle 9: *Ergebnisse auf einem erweiterten Winkelbereich von* $\pm 30^\circ$ *horizontal und* $\pm 25^\circ$ *vertikal.*

5.5 Ergebnisse der Offline-Versuche

Die Verwendung von Online-Experimenten erlaubt es, Aussagen über die Realisierbarkeit, Stabilität und Praxistauglichkeit zu machen.
Manche Aussagen - etwa über die genauen Auswirkungen der Variation einzelner Parameter - lassen sich hingegen nur schwer treffen, da die Testbedingungen von Experiment zu Experiment leicht variieren. Insbesondere bei den in Abschnitt 5.5.2 vorgestellten Experimenten zur progressiven Attention läßt sich unter wechselnden Bedingungen keine verläßliche Aussage machen.
Um Einflüsse durch Änderungen in der Umgebung auszuschließen, finden die folgenden Experimente mit einer aufgezeichneten Bildfolge statt. Das Tracking wird durch eine Vorparametrisierung aus dem zur Bildfolge gehörenden Online-Durchlauf ersetzt.

5.5.1 Generalisierungsvermögen

Für den praktischen Einsatz wäre es vorteilhaft, wenn nicht für jede Person ein eigenes Gaborwaveletnetz erzeugt werden muß.
Daher soll in diesem Versuch die Generalisierungsfähigkeit der Gaborwaveletnetze untersucht werden. Der naheliegendste Ansatz wäre es, *ein festes Waveletnetz* zur Merkmalsextraktion bei *mehreren Personen* zu verwenden. Aus Gründen der Vergleichbarkeit der Umgebungsbedingungen schlagen wir den umgekehrten Weg ein: Es werden mit *verschiedenen Waveletnetzen* die Merkmale aus der Bildfolge einer *festen Puppe* extrahiert. Als Waveletnetze wurden

- das zur Puppe gehörende Waveletnetz “Puppe I”,
- das zu einer anderen Puppe gehörende Waveletnetz “Puppe II” und
- das zu einer menschlichen Testperson gehörende Waveletnetz

verwendet. Die zugehörigen Gaborwaveletnetze sind in Abbildung 22 dargestellt.
Die Ergebnisse der Posenschätzung unter Verwendung der verschiedenen Waveletnetze sind in Tabelle 10 aufgeführt.
Intuitiv war zu erwarten, daß bei Verwendung von “fremden” Waveletnetzen die Posenschätzung deutlich schlechter ausfallen würde.
Diese Erwartung erfüllt sich nur teilweise. Bei Verwendung von optimierten Waveletgewichten als Eingabe für das LSDCS-Netz ist der mittlere quadratische Fehler in der Posenschätzung bei Puppe II sehr viel größer als beim Original-Waveletnetz oder dem der Testperson. Der mittlere quadratische Fehler ist in Abbildung 23 als Säulendiagramm dargestellt.
Bei Verwendung der Filterantworten hat das zur Merkmalsextraktion verwendete Waveletnetz keine erkennbaren Auswirkungen auf die Genauigkeit der Posenschätzung. Die Ergebnisse sind bei allen drei Waveletnetzen nahezu identisch und in Abbildung 24 graphisch dargestellt.

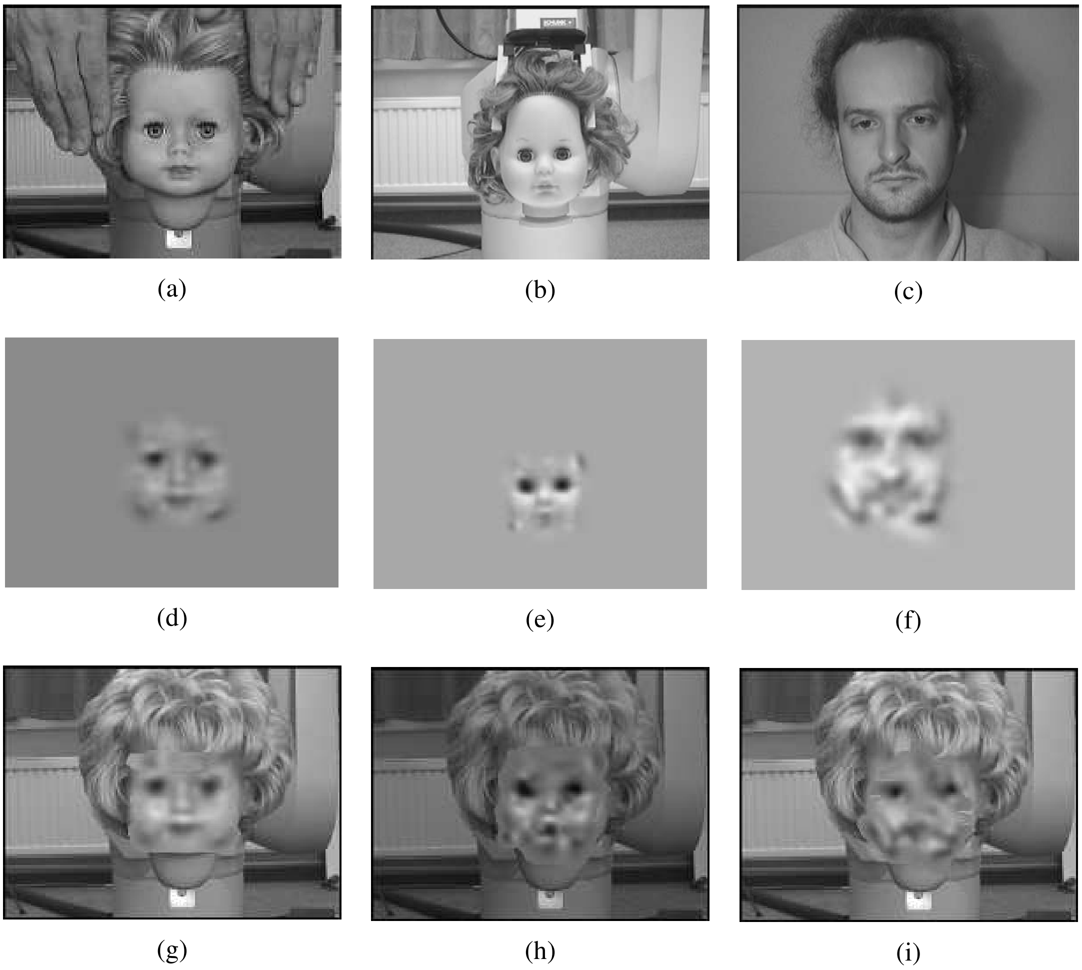


Abbildung 22: *In der oberen Zeile (a)-(c) sind jeweils die zum Training verwendeten Gesichter abgebildet. Darunter (d)-(f) sind die konstruierten Waveletnetze dargestellt. In der untersten Zeile (g)-(i) ist ein Testbild jeweils mit seiner Rekonstruktion aus dem darüber stehenden Waveletnetz überlagert. Von links nach rechts sind die Original-Puppe (a,d,g), ein weiterer Puppenkopf (b,e,h) und eine menschliche Testperson (c,f,i) dargestellt. Bei allen Waveletnetzen wurden 52 Wavelets für die Rekonstruktion verwendet.*

Die Ursache für dieses überraschende Ergebnis liegt darin, daß jedes Waveletnetz optimal reparametrisiert wurde, d.h. die Filter immer exakt über der gleichen Gesichtsregion plaziert wurden. Dieser Effekt wurde auch in [1] genutzt, wo die Filter gleichabständig über die Gesichtsregion verteilt wurden.
Es scheint, als ob das verwendete Waveletnetz für die Posenbestimmung nur insoweit von Bedeutung ist, daß die kodierten Merkmale für eine exakte Reparametrisierung ausreichen müssen.

Original-Waveletnetz (Puppe I)			
Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	0.35	0.75
	LSDCS	0.68	1.69
52 Waveletgewichte	LDCS	0.23	0.53
	LSDCS	0.59	1.74
16 Filterantworten	LDCS	0.47	0.99
	LSDCS	0.50	1.09
52 Filterantworten	LDCS	0.46	0.97
	LSDCS	0.51	1.12
Waveletnetz für Puppe II			
Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	0.59	1.53
	LSDCS	1.48	3.98
52 Waveletgewichte	LDCS	0.32	1.13
	LSDCS	1.23	3.10
16 Filterantworten	LDCS	0.42	1.02
	LSDCS	0.44	0.90
52 Filterantworten	LDCS	0.41	1.02
	LSDCS	0.50	1.08
Waveletnetz für Testperson			
Netzeingabe	Netztyp	mittlerer Testfehler	maximaler Testfehler
16 Waveletgewichte	LDCS	0.38	1.20
	LSDCS	0.73	1.63
52 Waveletgewichte	LDCS	0.27	0.51
	LSDCS	0.65	1.51
16 Filterantworten	LDCS	0.46	1.14
	LSDCS	0.54	1.08
52 Filterantworten	LDCS	0.42	1.16
	LSDCS	0.53	1.08

Tabelle 10: *Untersuchung der Generalisierungsfähigkeit von Gaborwaveletnetzen. Bei Verwendung der Waveletgewichte fallen die Ergebnisse bei Merkmalsextraktion mit "fremden" Waveletnetzen teilweise schlechter aus. Werden hingegen die Filterantworten verwendet, so ergibt sich ein überraschendes Ergebnis: Alle Waveletnetze eignen sich gleich gut. Die Ursache für diese erstaunliche Tatsache dürfte darin liegen, daß die Filter immer zuverlässig über der gleichen Region plaziert werden.*

5.5.2 Progressive Attention

Im weiteren soll der Einfluß der Anzahl der verwendeten Wavelets auf die erreichbare Genauigkeit näher untersucht werden. In den bisherigen Experimenten hat sich

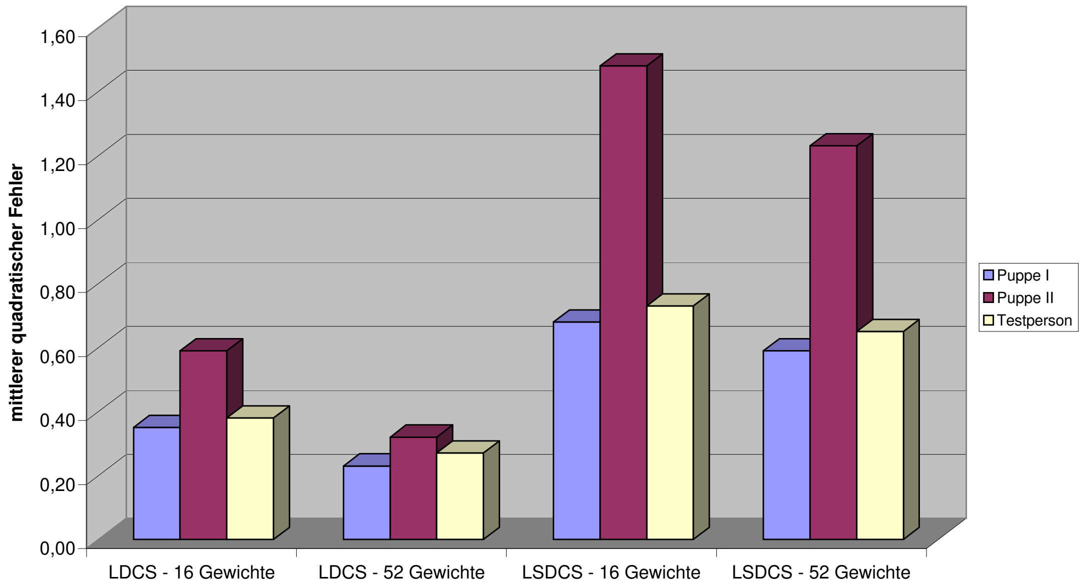


Abbildung 23: *Bei Verwendung der Waveletgewichte nimmt die Genauigkeit bei "fremden" Waveletnetzen ab.*

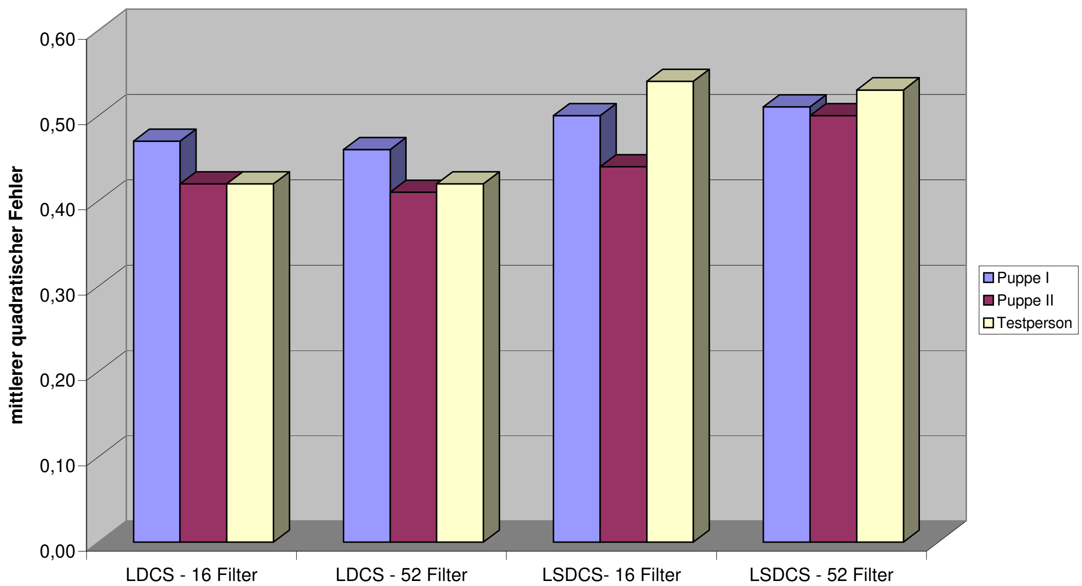


Abbildung 24: *Die Genauigkeit der Posenschätzung ist bei Verwendung der Filterantworten nahezu unabhängig vom verwendeten Waveletnetz.*

gezeigt, daß die Verwendung von 52 statt 16 Wavelets zu einer höheren Genauigkeit der Posenschätzung führt. Besonders deutlich hat sich dies bei den LDCS-Netzen gezeigt, während die LSDCS-Netze nur einen geringen Gewinn aus der Erhöhung der Waveletanzahl ziehen konnten.

Insbesondere ist von Interesse, ob sich die Genauigkeit durch das Hinzufügen von Wavelets beliebig erhöhen läßt.
Um diese Fragestellungen zu untersuchen, stehen zwei Möglichkeiten zur Auswahl. Die naheliegendste Methode wäre es sicherlich, für die verschiedenen Anzahlen von Wavelets jeweils ein neues Waveletnetz zu trainieren. Dabei stellt sich aber das Problem, wie die Waveletnetze initialisiert werden sollen - wie sollen beispielsweise die Wavelets bei der Initialisierung verteilt werden?

Aus Effizienzgründen wird in dieser Arbeit ein einfacherer Ansatz gewählt. In Abschnitt 2.6 wurden im Zusammenhang mit der *progressiven Attention* drei Methoden vorgestellt, welche es ermöglichen, mit *einem Waveletnetz nahezu beliebige Genauigkeitsabstufungen* zu erzielen.
Für die folgenden Experimente wurden aus einer Bildserie mit den drei vorgestellten Methoden Merkmale extrahiert. Es war die Erwartung, daß die Sortierung und Normierung der Wavelets zu einer merklichen Verbesserung der Ergebnisse im Vergleich zur Verwendung der unsortierten Wavelets führen würden.

Posenschätzung aus optimierten Gewichten

Die Ergebnisse der Posenschätzung unter Verwendung der drei vorgeschlagenen Methoden zur Auswahl der Gewichte für progressive Attention sind in Tabelle 11 aufgelistet.
Die Ergebnisse für die LDCS- und LSDCS-Netze sind in den Abbildungen 25 und 26 graphisch dargestellt.
Es zeigt sich, daß sowohl beim LDCS- als auch beim LSDCS-Netz die Genauigkeit bei allen drei Verfahren praktisch identisch ist.
Dieses Ergebnis entspricht nicht den geäußerten Erwartungen. Die Erklärung liegt in der Konstruktion des Waveletnetzes. Wie aus den Abbildungen 10 und 11 ersichtlich wird, sind die Wavelets schon im unsortierten Waveletnetz gleichmäßig über die Gesichtsregion verteilt. Dadurch kodieren die unsortierten Wavelets bereits genug Information, um Rückschlüsse auf die Gesichtspose zu ermöglichen. Da die gleichmäßige Verteilung - bezüglich der Indizierung der Wavelets - über die Gesichtsfläche nicht als garantiert angenommen werden kann, erscheint die explizite Sortierung der Wavelets nach dem Betrag der Gewichte als grundsätzlich sinnvoll[16].

Es bestätigt sich die schon in den Online-Experimenten beobachtete Überlegenheit des LDCS-Netzes bei der Verwendung von Gewichten als Netzeingabe. Ab etwa 12 Wavelets ist der mittlere Fehler beim LSDCS-Netz doppelt so hoch wie beim LDCS-Netz. Dieses Verhalten erklärt sich dadurch, daß sich in den Gewichten die Überlappung der Wavelets widerspiegelt, d.h. der Grauwert eines Pixels wird durch mehrere Wave-

[16]Auf die Konstruktion eines Waveletnetzes, welches im unsortierten Zustand schlechte Ergebnisse erzielt, soll hier verzichtet werden. Einen Problemfall stellt beispielsweise die Verwendung eines Waveletnetzes auf einem “fremden” Gesicht dar, da sich die ursprüngliche Bedeutung der Gewichte durch die Anpassung auf das fremde Gesicht grundlegend verändern kann.

lets repräsentiert. Das Zusammenspiel mehrerer Wavelets fügt sich nur schlecht in das Konzept der bei den LSDCS-Netzen verwendeten lokalen intrinsischen Unterräume.

Beide Netztypen zeigen einen asymptotischen Abfall des mittleren quadratischen Fehlers, wobei die Kurve beim LDCS-Netz glatter ist. Der abgeflachte Verlauf der Fehlerentwicklung läßt den Schluß zu, daß eine Erhöhung der Waveletanzahl keine signifikante Verbesserung mehr erwarten läßt.

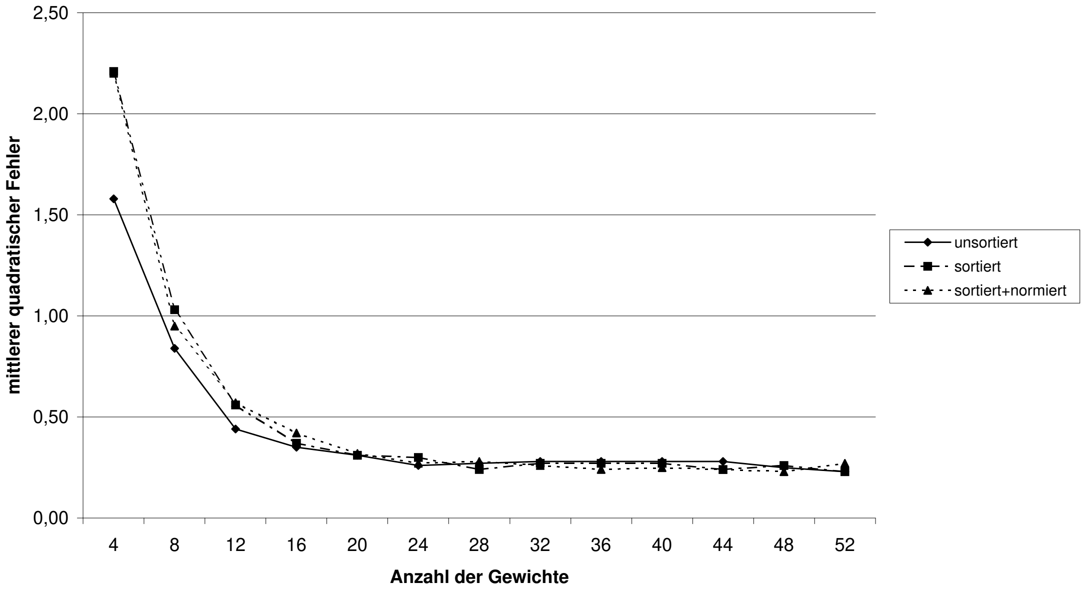


Abbildung 25: *Genauigkeit der Posenschätzung mit LDCS-Netzen aus Waveletgewichten in Abhängigkeit von der Waveletanzahl.*

Posenschätzung aus Filterantworten

Die Ergebnisse der Posenschätzung aus den sortierten[17] und unsortierten Filterantworten sind in Tabelle 12 zusammengefaßt und in den Abbildungen 27 und 28 graphisch dargestellt.

Der Verlauf des mittleren quadratischen Fehlers ähnelt dem bei Verwendung der Waveletgewichte. Bei beiden Verfahren zur progressiven Attention liegen die Testfehler eng zusammen. Beim LSDCS-Netz kommt der mittlere quadratische Fehler bereits bei 8 Wavelets mit $0.59°$ nahe an sein Minimum heran, während das LDCS-Netz dazu 28 Wavelets benötigt. Der minimale mittlere Testfehler liegt beim LSDCS-Netz mit $0.50°$ allerdings auf deutlich höherem Niveau als beim LDCS-Netz mit $0.39°$.

Bereits ab 16 Wavelets ist beim LSDCS-Netz unter Verwendung des sortierten Waveletnetzes keine signifikante Verbesserung der Posenschätzung mehr feststellbar. Beim LDCS-Netz steigt der Fehler ab 48 Wavelets sogar wieder an, da Rauschen auf den

[17]Die Filterantworten sind anhand der Beträge der Waveletgewichte sortiert worden.

Progressive Attention - LDCS						
Netzeingabe	unsortiert		sortiert		sortiert und normiert	
Gewichte	mittlerer Fehler	maximaler Fehler	mittlerer Fehler	maximaler Fehler	mittlerer Fehler	maximaler Fehler
4	1.58	4.81	2.21	7.70	2.20	5.03
8	0.84	2.39	1.03	2.99	0.96	2.64
12	0.44	1.42	0.56	1.56	0.57	1.79
16	0.35	0.75	0.37	0.91	0.42	1.11
20	0.31	0.71	0.31	0.75	0.32	0.71
24	0.26	0.66	0.30	0.98	0.27	0.63
28	0.27	0.65	0.24	0.58	0.28	0.78
32	0.28	0.74	0.27	0.64	0.26	0.61
36	0.28	0.70	0.27	0.58	0.24	0.79
40	0.28	1.01	0.27	0.92	0.25	0.50
44	0.28	0.66	0.24	0.59	0.24	0.68
48	0.25	0.55	0.26	0.72	0.23	0.67
52	0.23	0.53	0.23	0.53	0.27	0.76

Progressive Attention - LSDCS						
Netzeingabe	unsortiert		sortiert		sortiert und normiert	
Gewichte	mittlerer Fehler	maximaler Fehler	mittlerer Fehler	maximaler Fehler	mittlerer Fehler	maximaler Fehler
4	1.79	5.71	2.16	7.12	2.23	5.08
8	1.13	4.21	1.33	4.76	1.13	4.02
12	0.79	2.20	0.89	3.05	0.93	2.41
16	0.68	1.69	0.82	2.40	0.76	1.76
20	0.76	2.15	0.79	2.48	0.83	3.21
24	0.76	2.82	0.77	1.95	0.59	1.72
28	0.72	1.87	0.72	2.47	0.68	1.71
32	0.62	1.59	0.64	1.88	0.59	1.65
36	0.76	2.03	0.69	2.29	0.64	1.74
40	0.70	1.74	0.71	2.22	0.66	1.50
44	0.72	2.59	0.60	2.03	0.72	2.25
48	0.67	2.36	0.70	2.42	0.71	2.00
52	0.59	1.74	0.59	1.74	0.69	1.95

Tabelle 11: *Vergleich der Genauigkeit der Posenschätzung unter Verwendung verschiedener Methoden zur progressiven Attention. In der Tabelle sind die Ergebnisse für das LDCS- und LSDCS-Netz aufgeführt. Als Netzeingabe wurden die Waveletgewichte der Rekonstruktion verwendet.*

Trainingsdaten mitgelernt wird. Daher läßt eine Erhöhung der Waveletanzahl keine Verbesserung bei der Posenschätzung erwarten.

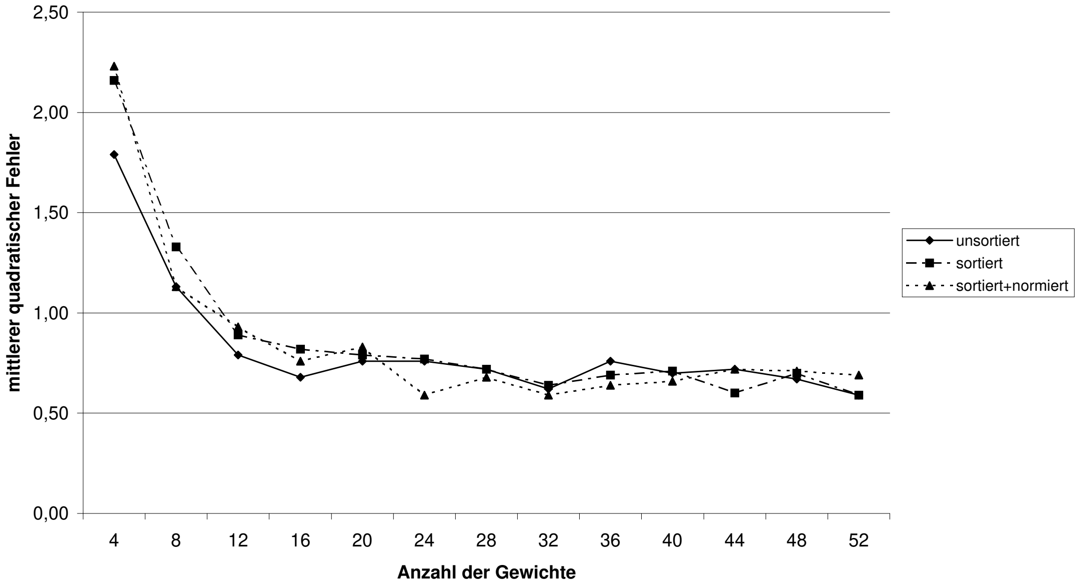


Abbildung 26: *Genauigkeit der Posenschätzung mit LSDCS-Netzen aus Waveletgewichten in Abhängigkeit von der Waveletanzahl.*

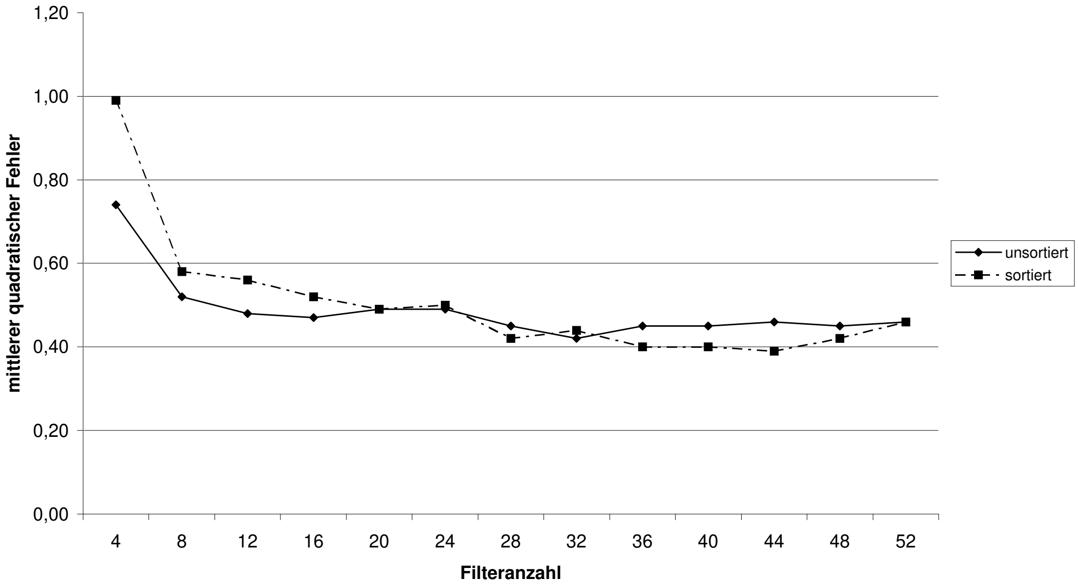


Abbildung 27: *Genauigkeit der Posenschätzung mit LDCS-Netzen aus den Filterantworten in Abhängigkeit von der Waveletanzahl.*

Vergleich der Genauigkeit von LDCS- und LSDCS-Netzen

Es hat sich in den Experimenten herauskristallisiert, daß im Bereich von $\pm 20°$ horizontal und $\pm 15°$ vertikal die Waveletgewichte besser zur Posenschätzung geeignet

Progressive Attention - LDCS				
Netzeingabe	unsortiert		sortiert	
Filterantworten	mittlerer Fehler	maximaler Fehler	mittlerer Fehler	maximaler Fehler
4	0.74	3.15	0.99	3.35
8	0.52	1.22	0.58	1.28
12	0.48	1.19	0.56	1.56
16	0.47	0.99	0.52	1.09
20	0.49	1.04	0.49	1.04
24	0.49	1.15	0.50	1.02
28	0.45	0.95	0.42	0.82
32	0.42	1.02	0.44	0.95
36	0.45	0.92	0.40	1.27
40	0.45	1.08	0.40	1.05
44	0.46	0.99	0.39	1.03
48	0.45	1.00	0.42	0.93
52	0.46	0.97	0.46	0.97

Progressive Attention - LSDCS				
Netzeingabe	unsortiert		sortiert	
Filterantworten	mittlerer Fehler	maximaler Fehler	mittlerer Fehler	maximaler Fehler
4	0.90	4.10	0.96	2.18
8	0.51	1.15	0.59	1.28
12	0.50	1.12	0.56	1.14
16	0.50	1.09	0.52	1.16
20	0.50	1.11	0.54	1.22
24	0.51	1.13	0.52	1.09
28	0.50	1.06	0.52	1.18
32	0.51	1.10	0.51	1.16
36	0.49	1.15	0.51	1.10
40	0.51	1.05	0.52	1.16
44	0.50	1.09	0.51	1.13
48	0.51	1.06	0.50	1.11
52	0.51	1.12	0.51	1.12

Tabelle 12: *Vergleich der Genauigkeit der Posenschätzung unter Verwendung von verschiedenen Methoden zur progressiven Attention. In der Tabelle sind die Ergebnisse für das LDCS- und LSDCS-Netz aufgeführt. Als Netzeingabe wurden die Filterantworten des reparametrisierten Gaborsuperwavelets verwendet.*

sind als die Filterantworten. Die Ursache hierfür liegt darin, daß die Filterantworten lokal begrenzt sind, während sich die Gewichte aus der Überlagerung vieler Wavelets

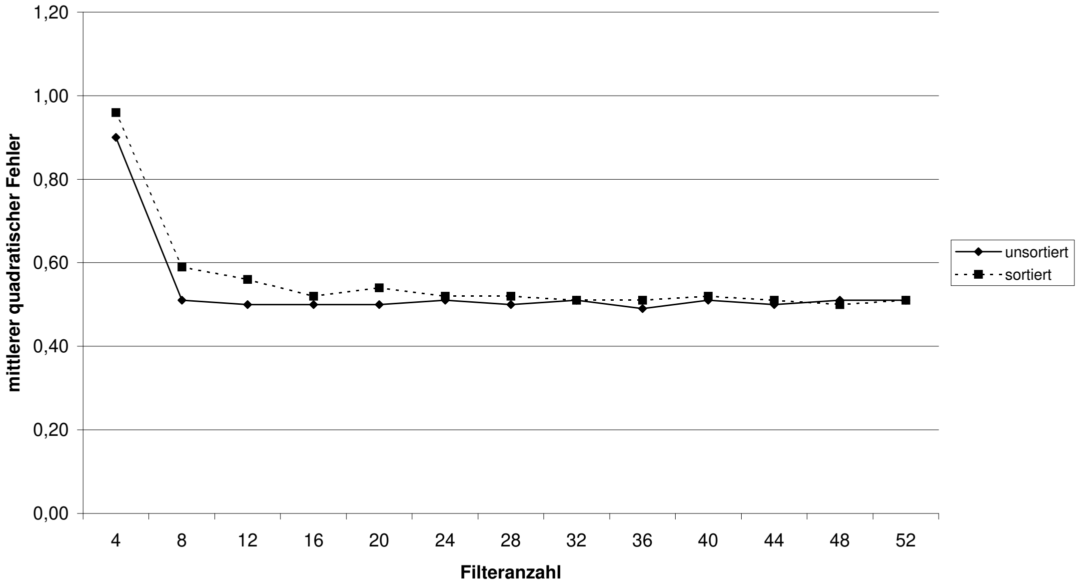


Abbildung 28: *Genauigkeit der Posenschätzung mit LSDCS-Netzen aus den Filterantworten in Abhängigkeit von der Waveletanzahl.*

bestimmen und die Gesichtsstruktur globaler beschreiben.
Ferner hat sich gezeigt, daß die LDCS-Netze den LSDCS-Netzen zum Zwecke der Posenbestimmung überlegen sind. In Abbildung 29 werden die Ergebnisse der beiden Netztypen unter Verwendung der Gewichte eines nach den Beträgen der Waveletgewichte sortierten Waveletnetzes dargestellt. Der mittlere quadratische Fehler liegt beim LDCS-Netz ab 16 Wavelets um mehr als 50% unter dem des LSDCS-Netzes.

Auf dem in Abschnitt 5.4.7 untersuchten erweiterten Winkelbereich von $\pm 30°$ horizontal und $\pm 25°$ vertikal haben sich die Waveletgewichte aufgrund der perspektivischen Verzerrungen als unbrauchbar erwiesen, so daß auf die Filterantworten zurückgegriffen werden muß. Für die Posenbestimmung aus Filterantworten auf dem erweiterten Winkelbereich haben sich die LSDCS-Netze als geeigneter erwiesen.

5.5.3 Matchingzeit in Abhängigkeit von der Waveletanzahl

Die Bedeutung der progressiven Attention für Echtzeitanwendungen liegt in der Skalierbarkeit. Der Einfluß der Waveletanzahl auf die Genauigkeit ist ausgiebig untersucht worden. Um den Zusammenhang zwischen Dauer der Reparametrisierung und der Größe eines Waveletnetzes zu untersuchen, wurden nach den Gewichten sortierte Waveletnetze der Größen $4, 8, \ldots, 48, 52$ Wavelets mit einer Abweichung von 6 Pixeln in x- und y-Richtung sowie 10% in der Skala von der Idealposition auf einem Testbild reparametrisiert. In den Experimenten zum Zeitbedarf wurden die benötigte Gesamtzeit sowie die Anzahl der Iterationen aufgezeichnet. Ferner wurden die Matchingergebnisse visuell auf Korrektheit überprüft. Bei der verwendeten Frontalansicht

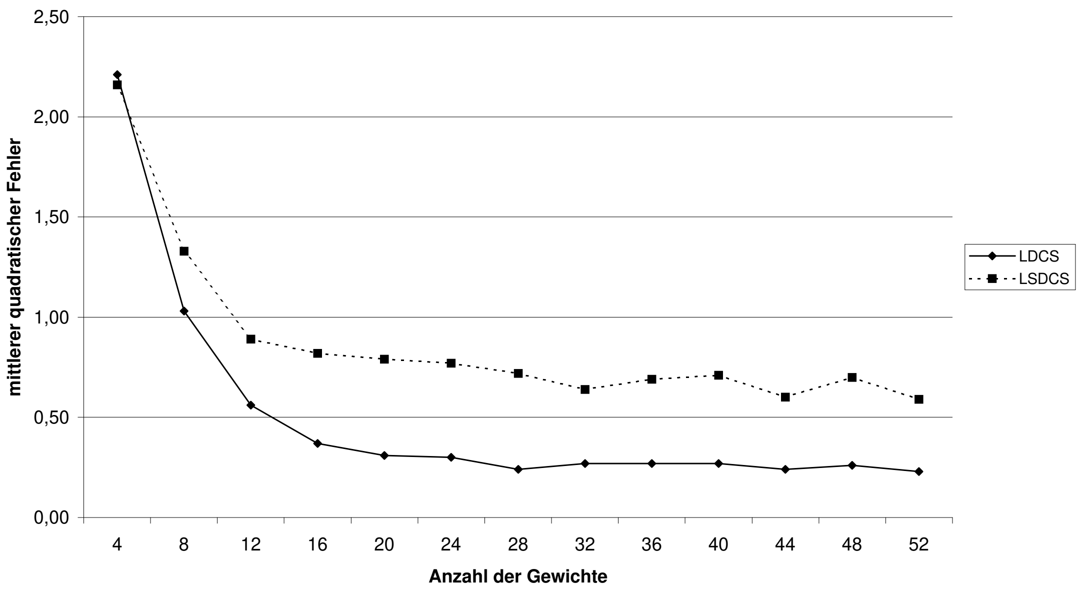
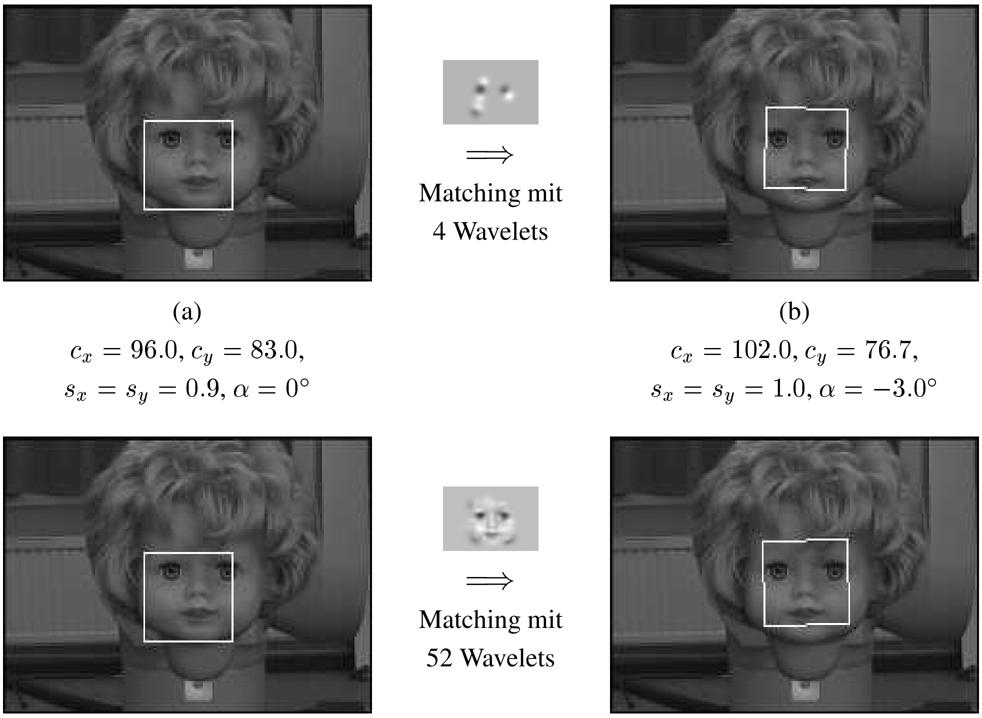


Abbildung 29: *Vergleich der Genauigkeit von LDCS- und LSDCS-Netzen zur Posenschätzung. Dargestellt ist der mittlere quadratische Fehler bei Verwendung von sortierten Waveletgewichten.*

gelang sogar noch mit 2 Wavelets ein Matching, was die Robustheit des Verfahrens unterstreicht. Die Ergebnisse für 4 und 52 Wavelets sind in Abbildung 30 dargestellt. Es fällt auf, daß sich die Ergebnisse des Matchings in Abhängigkeit von der Waveletanzahl geringfügig unterscheiden. Die durch Verringerung der Waveletanzahl reduzierte Strukturinformation führt dazu, daß die Energiefunktion (8) durch die abweichende Plazierung besser minimiert wird.
Die Ergebnisse der Zeitmessungen sind in Tabelle 13 und in Abbildung 31 dargestellt. Die Zeit für eine Einzeliteration hängt nahezu linear von der Zahl der Wavelets ab. Bei der Zeit für das gesamte Matching bietet sich ein ähnliches Bild. Die Stufen in der Kurve resultieren daraus, daß die für das Erreichen der Abbruchbedingung - d.h. für die Minimierung der Energiefunktion (8) - notwendige Anzahl von Iterationen in Abhängigkeit von der Waveletanzahl schwankt.

Bemerkung zum Tracking in Echtzeit

Die in Tabelle 13 angegebene Zeit von 300 ms für das Matching mit 16 Wavelets ist bedeutend größer als die in Kapitel 3 angegebenen 50 ms (20 Hz). Der Unterschied liegt zum einen in der relativ großen Distanz von 6 Pixeln in x- und y-Richtung begründet, welche beim Tracking selten auftritt. Zum anderen wird bei der Auswertung der Energiefunktion (8) das zum Echtzeit-Tracking verwendete Gaborsuperwavelet mit dem Faktor 3 in x- und y-Richtungen unterabgetastet. Der Rechenaufwand pro Iterationsschritt sinkt dadurch um den Faktor 9. Damit konnte das Tracking bei hinreichender



(a)
$c_x = 96.0, c_y = 83.0,$
$s_x = s_y = 0.9, \alpha = 0°$

(b)
$c_x = 102.0, c_y = 76.7,$
$s_x = s_y = 1.0, \alpha = -3.0°$

(c)
$c_x = 96.0, c_y = 83.0,$
$s_x = s_y = 0.9, \alpha = 0°$

(d)
$c_x = 102.0, c_y = 76.4,$
$s_x = s_y = 0.98, \alpha = 0.6°$

Abbildung 30: *Für die Experimente zur Matchinggeschwindigkeit wurden die Startparameter wie in der linken Spalte (a,c) gewählt. Das Matchingergebnis mit 4 Wavelets ist in (b) und mit 52 Wavelets in (d) dargestellt. Beide Matchingvorgänge konvergieren nahezu gleich. Lediglich der Rotationswinkel unterscheidet sich aufgrund der stark unterschiedlichen Waveletverteilungen.*

Genauigkeit auf unter 50 ms beschleunigt werden.

5.5.4 Trainingsdauer

Tabelle 14 zeigt typische Trainingszeiten für die verwendeten neuronalen Netze in Abhängigkeit von der Dimension des Eingabevektors, wie sie während des Trainings gemessen wurden. Angegeben sind die Zeiten, die für das Training von 500 Datensätzen und bei einer Netzgröße von 7 Neuronen beim LDCS-Netz und von 27 Neuronen beim LSDCS-Netz benötigt wurden.
Die Schwankungen in der Trainingsdauer beim LSDCS-Netz entstehen durch die Verwendung lokaler intrinsischer Unterräume, deren Dimension nicht konstant ist, sondern in Abhängigkeit von den Eingabedaten variiert. Dadurch ist die Trainingsdauer eines Netzes mit mehr Neuronen manchmal kürzer als die eines Netzes mit weniger

Wavelets	Matching [ms]	Einzeliteration [ms]	Iterationen
4	130	22	6
8	300	30	10
12	300	38	8
16	300	44	7
20	420	53	8
24	590	59	10
28	580	64	9
32	730	73	10
36	680	76	9
40	770	86	9
44	870	87	10
48	870	97	9
52	960	107	9

Tabelle 13: *Die Zeit für einen Iterationsschritt hängt nahezu linear mit der Waveletanzahl zusammen. Die Stufen in der Kurve resultieren daraus, daß die für das Erreichen der Abbruchbedingung notwendige Anzahl von Iterationen in Abhängigkeit von der Waveletanzahl schwankt.*

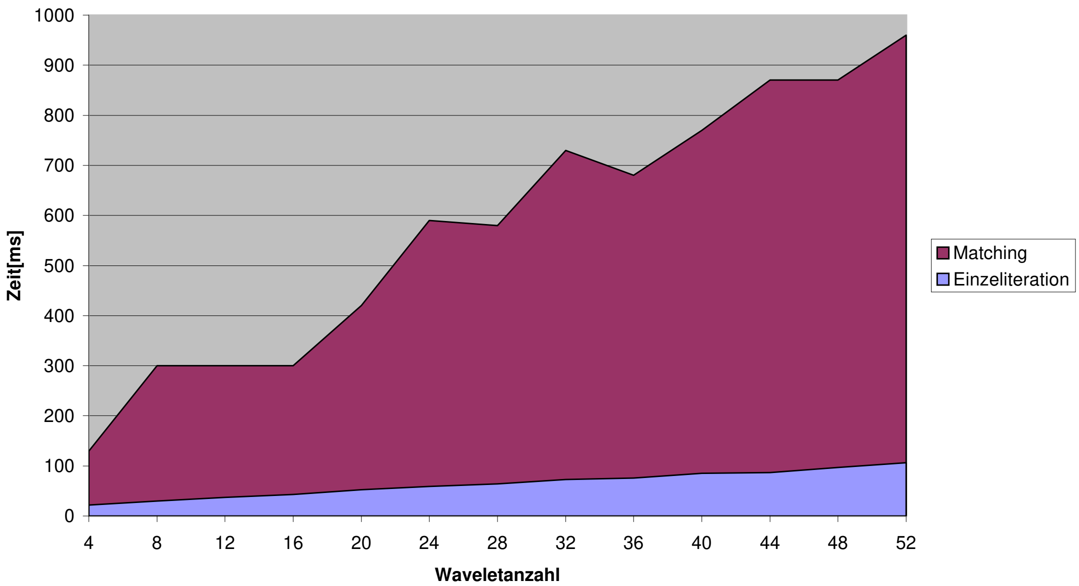


Abbildung 31: *Die Matchingdauer hängt nahezu linear von der Waveletanzahl ab.*

Neuronen. Es sollte aber erwähnt werden, daß die Dimensionen der Unterräume in einer vergleichbaren Größenordnung lagen.
Durch verbesserte Ausnutzung der Compiler-Optimierungen ließen sich die angegebenen Zeiten noch um den Faktor 5 verringern. Eine Beschleunigung um den Faktor 10

gegenüber den angegebenen Werten erscheint in Anbetracht der momentan verfügbaren Hardware realistisch.

Dimension des Eingabevektors	LDCS Trainingszeit [s]	LSDCS Trainingszeit [s]
16	38	215
20	60	217
24	85	213
28	114	227
32	153	251
36	194	285
40	253	290
44	287	286
48	400	292
52	446	321

Tabelle 14: *Typische Trainingszeiten für die verwendeten neuronalen Netze in Abhängigkeit von der Dimension des Eingabevektors wie sie während des Trainings gemessen wurden. Angegeben sind die Zeiten, die für das Training von 500 Datensätzen bei einer Netzgröße von 7 Neuronen beim LDCS-Netz und von 27 Neuronen beim LSDCS-Netz benötigt wurden. Die Schwankungen in der Trainingsdauer beim LSDCS-Netz entstehen durch die Projektion auf lokale intrinsische Unterräume. Deren Dimension ist nicht konstant, sondern variiert in Abhängigkeit von den Eingabedaten.*

6 Diskussion und Ausblick

In der vorliegenden Arbeit wurde ein auf Gaborwaveletnetzen basierendes Verfahren zur nichtbelastenden Bestimmung der Gesichtspose in monokularen unkalibrierten Videosequenzen vorgestellt. Die Gesichtspose eines Puppenkopfes ist mit einem mittleren quadratischen Fehler von weniger als $0.25°$ bestimmt worden. Damit hat sich die eingangs geäußerte Vermutung, daß eine geschicktere Plazierung der Filter mit einer Verbesserung der Posenschätzung einhergeht, bestätigt.
Dazu wurde ein auf Gaborwaveletnetzen basierendes Tracking implementiert, welches ein Gesicht subpixelgenau mit 20 Hz verfolgt.
Das Generalisierungsverhalten der Waveletnetze hat sich als außerordentlich gut erwiesen. Dieser Umstand liegt in der großen Anzahl von Freiheitsgraden begründet, die durch Verwendung des Gaborsuperwavelets und die Anpassung der Waveletgewichte verfügbar sind.
Das vorgestellte Verfahren zeichnet sich gegenüber anderen Arbeiten durch die Möglichkeit aus, mittels progressiver Attention einen optimalen Kompromiß zwischen Geschwindigkeit und Genauigkeit zu erreichen.
Ferner wurde ein Verfahren vorgestellt, das Waveletnetze durch Sortierung der Wavelets für die Verwendung im Zusammenhang mit progressiver Attention vorbereitet.

6.1 Vergleich mit der Diplomarbeit von E. Ábráham-Mumm

Diese Arbeit wurde durch die guten Ergebnisse von E. Ábráham-Mumm [1] motiviert. Mit dem hier vorgestellten Verfahren ist es gelungen, die wesentlichen dort auftretenden Einschränkungen zu beheben.
An erster Stelle ist die Gesichtsdetektion zu nennen, die in [1] durch eine beleuchtungsabhängige Farbsegmentierung auf Standbildern erfolgte. Dazu mußte vorausgesetzt werden, daß sich genau ein hautfarbenes Objekt im Kamerabild befindet. In dieser Arbeit wird stattdessen ein auf Gaborwaveletnetzen basierendes Trackingverfahren verwendet, welches robust gegenüber homogenen Beleuchtungsänderungen ist. Weiterhin wird ein spezielles Objekt verfolgt. Das Vorhandensein weiterer hautfarbener Objekte (Gesichter, Hände) ist ohne Auswirkungen auf die Gesichtsverfolgung.
In [1] wurde die detektierte Gesichtsfläche auf eine konstante Größe skaliert, während das hier verwendete Superwavelet durch Reparametrisierung optimal an das Gesicht angepaßt werden kann. Damit kann die gesamte verfügbare Bildinformation genutzt werden.
Der Schlüssel zur Verbesserung der Ergebnisse aus [1] liegt schließlich in der Verwendung der Waveletgewichte, die - im Gegensatz zu den lokalen Filterantworten - durch ihre Überlagerung die *Struktur des Gesichts* globaler beschreiben. Die Verwendung von 52 Gaborwavelets ermöglichte eine Erhöhung der Genauigkeit auf bis zu $0.21°$.
Bei alleiniger Verwendung der Gaborfilter konnten die Ergebnisse aus [1] nur geringfügig verbessert werden. Als Fortschritt gegenüber [1] verbleibt die auf $\frac{1}{10}$ redu-

zierte Filteranzahl zu nennen, welche für eine Genauigkeit von $0.5°$ notwendig war[18]. Als Vorteil der farbbasierten Plazierung der Gaborfilter über der Gesichtsregion in [1] bleibt der sehr große Winkelbereich von $\pm 75°$ horizontal und $\pm 30°$ vertikal zu nennen. Dagegen betrug der größte in dieser Arbeit untersuchte Winkelbereich lediglich $\pm 30°$ horizontal und $\pm 25°$ vertikal. Es soll allerdings darauf hingewiesen werden, daß der hier untersuchte Winkelbereich für die Arbeit am Bildschirm völlig ausreichend ist. Ferner wurden in [1] keine Untersuchungen zur Generalisierungsfähigkeit gemacht.

6.2 Verbesserungsmöglichkeiten und Übertragung auf praktische Anwendungen

In dieser Arbeit wurde die Gesichtspose eines Puppenkopfes unter Laborbedingungen mit hoher Genauigkeit bestimmt. Damit ist ein wichtiger Schritt getan. Der nächste große Schritt ist die Übertragung des Verfahrens auf praktische Anwendungen.

6.2.1 Erlangung der Grundwahrheit

Bei einem menschlichen Benutzer ist die Erlangung der Grundwahrheit ein aufwendiger Prozeß, für den viele verschiedene Lösungsansätze untersucht wurden. Bei der geplanten Verwendung der Gesichtspose zur Steuerung von Aktionen auf dem Bildschirm vereinfacht sich die Problemstellung allerdings wesentlich, da man nicht an der absoluten Pose (z.B. in Grad), sondern nur an Korrespondenzen von Pose und Bildschirmposition interessiert ist. Diese Korrespondenzen lassen sich durch die Verfolgung eines sich über den Bildschirm bewegenden Objektes durch den Benutzer leicht gewinnen [23; 4; 12]. Das Training kann online geschehen, während der Benutzer normal mit der Maus am Bildschirm arbeitet. Unter der Annahme, daß der Benutzer zum Zeitpunkt eines Mausklicks die Mausposition am Bildschirm fixiert, erhält man auf nichtbelastende Art und Weise die benötigten Korrespondenzen.
Für diese Anwendung bietet sich die Verwendung progressiver Attention an. Während der normalen Arbeit wird das Gesicht des Benutzers mit verminderter Genauigkeit verfolgt. Wird dann ein Mausklick ausgeführt, extrahiert das System die Merkmale mit maximaler Genauigkeit.
Das Training kann beendet werden, wenn der Fehler eine gewisse Schranke unterschreitet.

6.2.2 Auswahl eines geeigneten Waveletnetzes

Für eine stabile Gesichtsverfolgung muß ein zum Benutzer passendes Waveletnetz gewählt werden. Wie in Abschnitt 5.5.1 gezeigt, ist es dazu nicht zwingend notwendig, ein Waveletnetz für jeden Benutzer zu erzeugen. Vielmehr genügt es, ein hin-

[18] In [1] wurden 64 Gaborfilterpaare verwendet, während in dieser Arbeit ein Waveletnetz aus 12 ungeraden Gaborwavelets zum Tracking und zur Posenbestimmung mit einer Genauigkeit von $0.5°$ ausgereicht hat.

reichend ähnliches Waveletnetz zu verwenden. In [11; 14] sind Verfahren vorgestellt, welche die Identität einer Person anhand eines auf Gaborwaveletnetzen basierenden Ähnlichkeitsmaßes bestimmen. Dieses Ähnlichkeitsmaß kann benutzt werden, um in einer vorhandenen Datenbank ein zum Benutzer passendes Waveletnetz automatisch auszuwählen.

6.2.3 Stabilität

Die Stabilität der Gesichtsverfolgung - und damit auch der Posenbestimmung - hängt im wesentlichen von zwei Faktoren ab. Zum einen muß die Bildverarbeitung eine Mindestgeschwindigkeit einhalten und zum anderen muß der Benutzer homogen beleuchtet sein.
Die Geschwindigkeit könnte durch progressive Attention und ein zweistufiges Verfahren der Bildaufnahme optimiert werden: Selbst preiswerte Kameras und Framegrabber sind in der Lage Bildfolgen im PAL-Format, d.h. mit 50 Halbbildern in der Sekunde, aufzunehmen. Da die Position eines ruhenden Objektes sich zwischen zwei Halbbildern um maximal ein Pixel verschiebt, können die geraden und ungeraden Zeilen eines Vollbildes für den Zweck der Gesichtsverfolgung als voneinander unabhängige Bilder halbierter Auflösung verwendet werden. Der somit zur Verfügung stehende Videostrom von 50 Hz kann von einem Prozeß zur groben Gesichtsverfolgung genutzt werden. Da ein genauer Zeitablauf einzuhalten ist, sollte dieser Prozeß auf Systemebene implementiert werden.
Bei einer Aktion des Benutzers, z.B. einem Mausklick, kann ein weiterer, parallel laufender Prozeß gestartet werden. Dieser Prozeß kann anhand der groben Gesichtsposition eine exakte Reparametrisierung des Superwavelets vornehmen, die notwendigen Merkmale extrahieren und ggf. die Pose bestimmen. Aufgrund des *im Hintergrund weiterlaufenden Trackingprozesses* geht - selbst bei einer aufwendigen Merkmalsextraktion und Posenbestimmung - die *Gesichtsposition nicht verloren*. Damit ist eine zeitliche Entkoppelung zwischen Tracking und Posenbestimmung erreicht. Das vorgeschlagene Verfahren ist in Abbildung 32 skizziert.
Da die Posenbestimmung nichtbelastend sein soll, verbietet sich die Beleuchtung des Benutzers mit einer leistungsstarken Lichtquelle. Weil eine homogene Beleuchtung für das vorgestellte Verfahren notwendig ist, wäre als Ausweg die Verwendung einer Infrarotlichtquelle und Infrarotkamera zu untersuchen.

6.2.4 Verwendung von Scherung und Rotationswinkel

Die Verwendung von Scherung und Rotationswinkel des reparametrisierten Superwavelets wurde bereits in Abschnitt 4.3 vorgeschlagen. Unter der Voraussetzung, daß das Tracking stabil funktioniert, wäre es möglich, die Pose direkt und ohne eine aufwendige Merkmalsextraktion zu bestimmen.

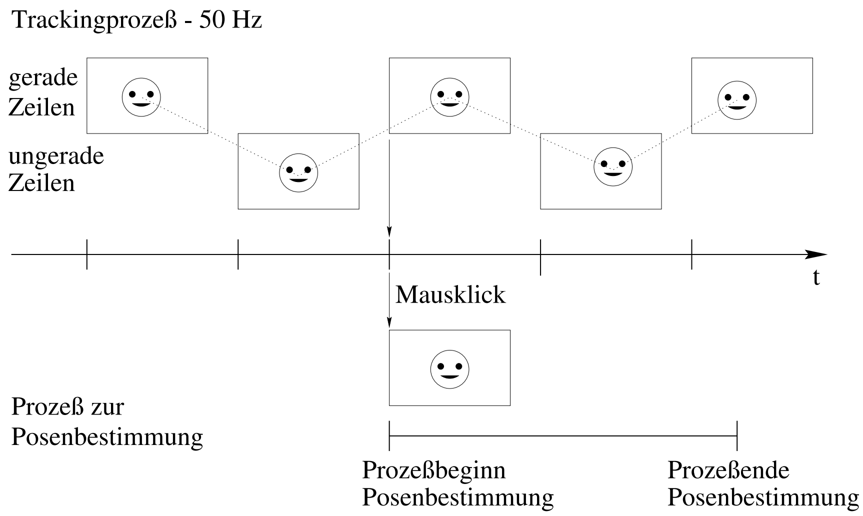


Abbildung 32: *Um eine stabile Gesichtsverfolgung sicherzustellen, könnten Tracking und Posenbestimmung in getrennte Prozesse verlagert und somit entkoppelt werden.*

6.2.5 Erweiterung auf Blickrichtungsdetektion

Wie in Abschnitt 1.1.1 festgestellt, kann die Gesichtspose in vielen Fällen als Näherung der Blickrichtung verwendet werden. Es muß allerdings zugegeben werden, daß die Verwendung der Pose nicht die gleiche Flexibilität wie die Verwendung der Blickrichtung ermöglicht.
Um ausgehend von der Gesichtspose die Blickrichtung zu bestimmen, könnte das verwendete Waveletnetz um zwei Teilnetze in der Augenregion erweitert werden. Durch die Reparametrisierung des Superwavelets mit Freigabe der Scherung ist die relative Position der Augen bekannt. Die auf den Augen plazierten Teilnetze können dann zusammen mit der Pose zur Bestimmung der Blickrichtung verwendet werden.

6.2.6 Unterstützung von Spracherkennung durch Lippenlesen

Eine weiteres mögliches Anwendungsgebiet für Gaborwaveletnetze ist die Spracherkennung. Wie bereits in Abschnitt 1.1.2 vorgeschlagen, kann die Bestimmung der Gesichtspose genutzt werden, um die Lippen zu lokalisieren und zur Verbesserung der Erkennungsrate heranzuziehen.
Es bleibt zu untersuchen, ob ein Waveletnetz auf die Lippenregion trainiert werden kann, welches eine Silbenerkennung realisieren könnte. Aufbauend auf dem Lippenlesen, das in gewisser Weise eine Lautschätzung darstellt, könnte dann eine Spracherkennung - d.h. die Interpretationsschicht - aufbauen.

6.3 Natürliche Grenzen bei der Bestimmung der Blickrichtung

In [2] stellt Aloimonos fest, daß alle in der Natur vorkommenden visuellen Systeme aktiv sind. Ein aktiver Betrachter kann seine Aufmerksamkeit auf die Region lenken, die ihn am meisten interessiert, d.h. er kann seine visuellen Fähigkeiten ("Rechenleistung") optimal einsetzen. Diese Lokalität spiegelt sich im Aufbau des menschlichen Auges wider. Die höchste Auflösung steht nur in der *fovea centralis* zur Verfügung. Die Auflösung des menschlichen Auges nimmt zu den Rändern stark ab. Nur mit dieser Spezialisierung ist das Erreichen eines großen Blickwinkels in Kombination mit einer hohen Auflösung möglich. Würde das Auflösungsvermögen der Netzhaut überall so hoch wie in der Fovea sein, so würde das menschliche Gehirn mehrere Tonnen wiegen [19]. Die Einschränkung des Auflösungsvermögens hat die erfreuliche Konsequenz, daß die Bestimmung der Blickrichtung überhaupt mit einer hohen Genauigkeit möglich ist. Durch die sich zum Rand hin verringernde Auflösung ist der Mensch gezwungen, das Auge direkt auf das ihn interessierende Objekt zu richten, da er nur in der Fovea die maximale Auflösung erreicht. Daraus folgt als Konsequenz, daß sich die Attention eines Menschen anhand der Stellung seiner Augen schätzen läßt. Wäre die Auflösung über den ganzen Blickwinkel konstant - wie etwa bei den Facettenaugen von Insekten - so könnte die Blickrichtung nur anhand der Gesichtspose geschätzt werden.

Üblicherweise wird die Augenstellung beim Menschen anhand der Verformung der Iris oder der Pupillenstellung bestimmt. Da die Fokussiergenauigkeit des menschlichen Auges nur etwa 1 Grad beträgt [8], kann die Blickrichtung ebenfalls nicht genauer bestimmt werden. Daraus ergibt sich bei einem Betrachtungsabstand von etwa 50 cm eine Unsicherheit von rund 9 mm. Bei einer Workstation-typischen Bildschirmauflösung von 96 dpi entspricht das mehr als 30 Pixeln. Diese Betrachtung zeigt die Grenzen der erreichbaren Genauigkeit. Ungeachtet des betriebenen Aufwands wird es niemals möglich sein, durch Blickrichtungsdetektion etwa einen Mauszeiger pixelgenau zu positionieren.

Als Konsequenz daraus folgt, daß für eine bequem bedienbare Mensch-Maschine-Schnittstelle ein erheblicher Forschungsbedarf für die Entwicklung von geeigneten Konzepten für Bedienoberflächen besteht.

[19] Aloimonos nennt in [2] einen Wert von 30 000 Pfund.

Symbolverzeichnis

Verzeichnis der verwendeten Symbole und Bezeichnungen.

Waveletnetze

$\mathbf{A}^T$	transponierte Matrix $\mathbf{A}$
$\mathbf{v}^T$	transponierter Vektor $\mathbf{v}$
$< \cdot, \cdot >$	Skalarprodukt
$\mathbb{L}^2(\mathbb{R}^2)$	Raum der quadratintegrierbaren Funktionen über $\mathbb{R}^2$
ψ	Waveletfunktion, Mutterwavelet
$\psi_{\mathbf{n}}$	Wavelet mit Parametervektor $\mathbf{n} = (c_x, c_y, s_x, s_y, \alpha)$
$\hat{\psi}$	Fouriertransformierte von ψ
$\mathbf{\Psi}$	Familie von Waveletfunktionen
$\mathbf{\Psi}_{\mathbf{n}}$	Superwavelet mit Parametervektor $\mathbf{n} = (c_x, c_y, s_x, s_y, \alpha)$
$\tilde{\mathbf{\Psi}}$	Vektor von dualen Wavelets
c_x, c_y	Translationsparameter eines (Super-)Wavelets
s_x, s_y	Dilatationsparameter eines (Super-)Wavelets
s_{xy}	Scherungsparameter eines Superwavelets
α	Rotationsparameter eines (Super-)Wavelets
$\mathbf{S}$	Dilatationsmatrix
$\mathbf{R}$	Rotationsmatrix
$\mathbf{t}$	Translationsvektor
N	Anzahl der Wavelets in einem Waveletnetz
$\mathbf{g}$	Grauwertbild
$\overline{\mathbf{g}}$	Gleichanteil des Grauwertbildes $\mathbf{g}$
$\mathbf{I}, \mathbf{I}_t$	Grauwertbild aus einer Bildsequenz (zum Zeitpunkt t)
π	Permutation

Neuronale Netze

$\mathbf{E}$	Kantenmenge eines Graphen
$\mathbf{V}$	Knotenmenge eines Graphen
$\mathbf{C}_i$	Korrelationsmatrix zum Neuron i
$\mathbf{E}_i$	Matrix der Eigenvektoren zum Neuron i
$\mathbf{e}_i$	Vektor der Eigenwerte zur Matrix $\mathbf{E}_i$
Nh(i)	Menge der zum Neuron i benachbarten Neuronen
Test	Menge von Testdaten
Train	Menge von Trainingsdaten
bmu$(\mathbf{x})$	das im Eingaberaum zu $\mathbf{x}$ am nächsten liegende Neuron (*best matching unit*)
second$(\mathbf{x})$	das im Eingaberaum zu $\mathbf{x}$ am zweitnächsten liegende Neuron
$\mathbf{y}_{Netz}(\mathbf{x})$	Netzantwort auf den Eingabevektor $\mathbf{x}$
$\mathbf{y}_i(\mathbf{x})$	Antwort des Neurons i auf den Eingabevektor $\mathbf{x}$
$\mathbf{J}_i$	Jacobi-Matrix zum Neuron i

Abbildungsverzeichnis

Tabellenverzeichnis

Literatur

[1] Erika Ábráham-Mumm. *Bestimmung der Gesichtspose mit künstlichen neuronalen Netzen*, 1998. Diplomarbeit am Lehrstuhl für Kognitive Systeme des Instituts für Informatik und Praktische Mathematik der Christian-Albrechts-Universität zu Kiel.

[2] Y. Aloimonos. Active vision revisited. In Yannis Aloimonos, editor, *Active Perception*, pages 1–18. Lawrence Erlbaum, Hillsdale, 1993.

[3] P. Ballard and G.C. Stockman. Controlling a Computer via Facial Aspect. In *IEEE Trans. on Systems, Man, and Cybernetics, Vol. 25, No. 4*, pages 669–677, 1995.

[4] S. Baluja and D. Pomerleau. Non-intrusive gaze tracking using artificial neural networks. In J. Cowan et.al., editor, *Advances in Neural Information Processing Systems 6*, pages 753–760. Morgan Kaufmann Publisher, 1994.

[5] Gary R. Bradski. Computer Vision Face Tracking For Use in a Perceptual User Interface. *Microcomputer Research Lab, Santa Clara, CA, Intel Corporation, Q2 '98*, 1998.

[6] J. Bruske. Dynamische Zellstrukturen - Theorie und Anwendung eines KNN-Modells. Technical Report 9809, Inst. f. Inf. u. Prakt. Math., Christian-Albrechts-Universität zu Kiel, Juni 1998.

[7] J. Bruske and G. Sommer. Dynamic cell structure learns perfectly topology preserving maps. *Neural Computation*, 7(4):845–865, 1995.

[8] R.H.S. Carpenter. *Movements of the eyes*. Pion Press, London, 1988.

[9] Qian Chen, Haiyuan Wu, Tadayosi Shioyama, Tetsuo Shimada, and Kunihiro Chihara. A Robust Algorith for 3D Head Pose Estimation. In *ICPR98*, pages 1356–1359, 1998.

[10] A. Gee and R. Cipolla. Determining the gaze of faces in images. *Image and Vision Computing*, 12(10):639–647, 1994.

[11] Alexander Happe. *Auf dem Weg zur Gesichtserkennung unter Verwendung von Gaborwaveletnetzen*, 2000. Diplomarbeit am Lehrstuhl für Kognitive Systeme des Instituts für Informatik und Praktische Mathematik der Christian-Albrechts-Universität zu Kiel.

[12] H. Klingspohr, T. Block, and R.-R. Grigat. A passive real-time gaze estimation system for human-machine interfaces. In *Int. Conf. on Computer Analysis of Images and Patterns, CAIP*, Kiel, Germany, Sept. 10-12, 1997.

[13] V. Krüger, R. Herpers, K. Daniilidis, and G. Sommer. Teleconferencing using an attentive camera system. In *Int. Conf. on Audio- and Video-based Biometric Person Authentication*, pages 142–147, 1999.

[14] V. Krüger and G. Sommer. Gabor Wavelet Networks for Object Representation. Technical Report Number 2002, Christian-Albrechts-Universität zu Kiel, Institut für Informatik und Praktische Mathematik, Februar 2000.

[15] Volker Küger, Sven Bruns, and Gerald Sommer. Efficient Head Pose Estimation with Gabor Wavelet Networks. In *Proc. Conf. BMVC*, Bristol, Great Britain, 11-14 September, 2000.

[16] Louis, Maaß, and Rieder. *Wavelets*. Teubner Studienbücher Mathematik, B.G. Teubner Stuttgart, 1994.

[17] S. McKenna, S. Gong, and J. Collins. Face tracking and pose representation. In *Edinburgh*. Proc. British Machine Vision Conference, 1996.

[18] H. Murase and S.K.Nayar. Visual Learning and Recognition of 3-D Objects from Appearance. *International Journal of Computer Vision*, 14(1):5–24, 1995.

[19] Alexander Pelc. *Bildkompression durch gesteuerte Wavelet-Netze*, 1997. Diplomarbeit am Lehrstuhl für Kognitive Systeme des Instituts für Informatik und Praktische Mathematik der Christian-Albrechts-Universität zu Kiel.

[20] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes, The Art of Scientific Computing*. Cambridge University Press, Cambridge, UK, 1986.

[21] Gerald Sommer, Kostas Daniilidis, Udo Mahlmeister, and Michael Hansen. Skript zur Vorlesung Computer Vision I, 1994. Lehrstuhl für Kognitive Systeme, CAU zu Kiel.

[22] Jan Töster. *Visuell basiertes Greifen gestützt auf neuronale Netze*, 1997. Diplomarbeit am Lehrstuhl für Kognitive Systeme des Instituts für Informatik und Praktische Mathematik der Christian-Albrechts-Universität zu Kiel.

[23] A.C. Varchmin, R. Rae, and H. Ritter. Image based recognition of gaze direction using adaptive methods. In I. Wachsmuth, editor, *Proceedings of the International Gesture Workshop*, lncs, pages 245–257. Springer, 1997.

[24] L. Wiskott, J. M. Fellous, N. Krüger, and C. v. d. Malsburg. Face recognition by elastic bunch graph matching. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 19(7):775–779, July 1997.

[25] Q. Zhang and A. Benveniste. Wavelet Networks. *IEEE Trans. on Neural Networks*, 3(6):889–898, Nov. 1992.

Danksagung

An dieser Stelle möchte ich all den Personen danken, die zum Gelingen dieser Arbeit beigetragen haben.

Ich danke Volker Krüger für seine intensive Betreuung dieser Diplomarbeit. Durch seine konstruktive Kritik und Geduld hat er viel zum Gelingen dieser Arbeit beigetragen. Ferner danke ich dem Lehrstuhl für kognitive Systeme für hilfreiche Diskussionen und eine angenehme Atmosphäre. Insbesondere gilt mein Dank auch den technischen Mitarbeitern des Lehrstuhls, die bei technischen Problemen umgehend geholfen und meine unzähligen Sonderwünsche mit viel Geduld ertragen haben. Mein Dank gilt desweiteren Erika Ábráham-Mumm, Sven Buchholz und Norbert Krüger für anregende Diskussionen.

Weiterhin gilt mein Dank meiner Verlobten Petra Leunig, die mir durch ihre unendliche Geduld und Zuwendung die Zeit während des Schreibens dieser Arbeit sehr erleichtert hat.

Für die Korrektur meiner zahllosen Rechtschreib- und Grammatikfehler sowie für konstruktive Kritik bedanke ich mich bei Petra, Andreas Bunten, Stefan Ludwig und Birgit Schimkus.

Abschließend möchte ich meinen Eltern für ihre andauernde Unterstützung vor und während meines Studiums danken. Ohne sie wäre diese Arbeit nicht in dieser Form möglich gewesen. Besonderer Dank gilt meinem Onkel Rüdiger Bruns, der mein Interesse an den Naturwissenschaften geweckt und stets gefördert hat.

Erklärung

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Kiel, den 15. Oktober 2000

- -

Sven Bruns