Attentionsgesteuerte progressive Übertragung von Bildern

vorgelegt von
Michael J. Sandgaard
Kiel, im Juli 1997

Diplomarbeit
am Lehrstuhl für Kognitive Systeme
des Instituts für Informatik und Praktische Mathematik
der Christian–Albrechts–Universität zu Kiel
Prof. Dr. G. Sommer
betreut durch Dipl.-Ing. Udo Mahlmeister

Zusammenfassung

Moderne Kompressionsverfahren bedienen sich oft verlustbehafteter Algorithmen, um eine hohe Datenkompression zu erreichen. Betrachtet man das Szenario einer Bilddatenbankabfrage und sucht nach einem bestimmten Bild, so kann die effektiv zu übertragende Datenmenge weiter reduziert werden, indem man die Übertragung von uninteressanten Bildern möglichst früh abbricht. Um diesen Entscheidungszeitpunkt in möglichst kurzer Zeit zu erreichen, bedienen wir uns eines aufgabenbezogenen progressiven Übertragungsschemas, das im Gegensatz zu zeilenbasierten progressiven Übertragungsstrategien schon nach kurzer Zeit die Darstellung der aufgabenrelevanten Bildanteile ermöglicht. Ist man nur an bestimmten Bildinformationen interessiert, kann dieser Entscheidungszeitpunkt deutlich früher erreicht werden, indem zuerst die gesuchten Strukturen übertragen werden.

Wir präsentieren ein attentionsgesteuertes progressives Kompressionsverfahren, das auf einer *steuerbaren Pyramide* basiert, die in Ort und Orientierung steuerbar ist. Aus den Transformationskoeffizienten errechnen wir die *lokale Amplitude und Orientierung*. Dieser Ansatz ermöglicht zum einen eine Reduktion der Koeffizienten und zum anderen, auf den Orientierungsbildern nach einem vorgegebenen Modellobjekt zu suchen. Für den Vergleich von Modell- und Originalbild benutzen wir vier Verfahren, die auf *Histogrammen* von Vektoren arbeiten, die über den drei obersten Pyramidenstufen der Orientierungsbilderpyramide gebildet werden. Das Ergebnis dieser Ähnlichkeitsmessung ist eine Auffälligkeitskarte, die die Übertragungsreihenfolge der Koeffizienten festlegt, um zu einer aufgabenbezogenen progressiven Übertragung zu gelangen.

In dieser Arbeit werden schrittweise die einzelnen Teile des attentionsgesteuerten progressiven Kompressionsverfahrens erläutert und die Ergebnisse ausführlicher Testreihen vorgestellt, um die Leistungsfähigkeit der Repräsentation nach lokaler Amplitude und Orientierung und der verwendeten Verfahren darzustellen.

Inhaltsverzeichnis

Kapitel 1

Einführung

In den vergangenen Jahren haben lokale (LAN) und globale (WAN) Netzwerke, wie es z.B. das Internet darstellt, eine immer größere Bedeutung gewonnen. Jedes größere Unternehmen tauscht Daten über Computer aus, die in einem Netzwerk verbunden sind. Gleichzeitig sind eine immer größere Anzahl von Unternehmen, aber auch von Privatanwendern, im Internet bzw. World Wide Web (WWW) präsent. Universitäten nutzen diese Form des globalen Informationsaustauschs schon seit Jahren. Mit der steigenden Kommerzialisierung zeigt sich jedoch eine Verschiebung in der Art der ausgetauschten Daten: Wurden früher fast nur Texte über ein Netzwerk ausgetauscht, steigt seit einigen Jahren die Anzahl der umfangreicheren Daten, wie es z.B. Bilder darstellen. So ist es heutzutage z.B. schon möglich, sich medizinische Bilder, Bilder von Personen, Landschaften oder Galerien usw. im WWW anzusehen.

Diese sich immer weiter steigernden Möglichkeiten lassen für die Zukunft auch ganze Bilddatenbanken erwarten, aus denen jedes beliebige Bild abgefragt werden kann. So wäre es z.B. denkbar, daß Ärzte über Computernetzwerke Röntgenbilder oder Computertomographiebilder austauschen und mit in Datenbanken gespeicherten Daten vergleichen. Diese steigende Anzahl von zu übertragenden Bildern erfordert die Enwicklung immer besserer Algorithmen zur Bildkompression, um die sehr großen Datenmengen über begrenzte Leitungskapazitäten übertragen zu können. Insbesondere haben hierbei Subbandkompressionsverfahren wie z.B. die Waveletkompression eine immer größere Bedeutung gewonnen. Zahlreiche Autoren haben sich mit den Bildkompressionseigenschaften der Wavelettransformation beschäftigt (siehe [23], [25], [24] oder [14] als Beispiele).

Die steigende Menge an digital vorliegenden Bildern stellt jedoch auch große Anforderungen an Datenbanken, in denen diese Bilder abgelegt sind. Bildinhalte können nicht mit wenigen Worten zusammengefaßt werden wie dies bei Texten der Fall ist. Schlüsselworte (oder Suchbegriffe) für Bilder zu vergeben ist hochgradig subjektiv, teuer und sichert in keinster Weise zu, daß bei einer Anfrage an die Datenbank auch Bilder übertragen werden, die durch Synonyme charakterisiert sind. Statt dessen bietet es sich an, der Bilddatenbank einen kleinen Bildausschnitt zu senden, auf dem ein Beispiel für das gesuchte Objekt dargestellt ist. Ansätze dieser Art werden in verschiedenen Artikeln zu Visual Information Management Systems (VIMS) propagiert [8], [9], [15]. Das VIMS wählt anhand des Beispielbildes (und evtl. weiter eingeschränkt durch die Auswahl von groben Schlüsselworten oder Kategorien) eine vorgegebene Anzahl von Bildern aus, die dem Beispielbild am ähnlichsten sind. Diesen Vorgang bezeichnet man auch als *Indexing*. Die ausgewählten Bilder werden dann übertragen, woraufhin evtl. eine neue verfeinerte Anfrage gestartet wird.

In vielen Anwendungen werden Bilder zeilenweise übertragen. Trotz wachsender Bandbreiten können diese Bildübertragungen nicht augenblicklich geschehen, da gleichzeitig eine überproportionale Steigerung der Bildvolumen zu verzeichnen ist. Es ist also realistisch anzunehmen, daß die Übertragung eines Bildes zumindest wenige Sekunden benötigt. Durch eine zeilenweise Übertragungsstrategie kann der Bildinhalt jedoch erst sehr spät erkannt werden, d.h. oftmals erst, wenn mehr als die Hälfte des Bildes übertragen ist. Da es bei einer Anfrage an eine Bilddatenbank oft vorkommen kann, daß uninteressante Bilder übertragen werden (insbesondere wenn nicht genau bekannt ist wonach gesucht wird), bedeutet dies, daß sehr viele überflüssige Daten übertragen werden. Um dieses Problem zu beseitigen, ist es angebracht, eine progressive Übertragungsstrategie zu verfolgen. Bei der progressiven Übertragung wird eine Auflösungspyramide erzeugt, von der zunächst nur die Koeffizienten der obersten (also kleinsten) Pyramidenstufe übertragen werden. Wird dieses Bild auf die Originalgröße ausgedehnt, ergibt sich durch sehr wenige Koeffizienten ein unscharfes Bild, das jedoch schon einen ersten Eindruck des Bildinhalts vermittelt. Das Bild wird durch die Übertragung von weiteren Koeffizienten immer schärfer bis im Endeffekt das komplette Bild übertragen wurde. Diese progressive Übertragungsstrategie, die zuerst in [2] dargestellt wurde, ermöglicht ein deutlich früheres Erreichen des Zeitpunkts, zu dem der Bildinhalt erkannt wird. Die Übertragung von uninteressanten Bildern kann also sehr schnell abgebrochen werden, was Übertragungskapazitäten spart und somit effektiv eine Datenkompression darstellt. Ein Ansatz, der diese Strategie verfolgt, wurde bereits 1983 in [3] vorgestellt. Die dort verwendete Struktur ermöglicht es auch, daß Stellen angegeben werden, an denen das Bild feiner aufgelöst werden soll. Es ist bei diesem System also zum Zweck einer aufgabenbezogenen Codierung schon eine gewisse Interaktion möglich.

Mehrere Autoren haben die ursprüngliche Darstellung als Laplace-Pyramide, wie sie in [2] vorgestellt wurde, verfeinert. Wir werden in dieser Arbeit auf einer Pyramidendarstellung aufsetzen, die einige weitere nützliche Eigenschaften aufweist und das erste Mal in [21] beschrieben wurde.

Der oben angesprochene Entscheidungszeitpunkt, an dem über den Abbruch der Bildübertragung entschieden wird, kann durch eine weitere Verfeinerung der Übertragungsstrategie noch früher erreicht werden. Überträgt man nicht alle Koeffizienten gleichberechtigt, sondern bevorzugt Koeffizienten, die "interessante" Bildstellen codieren, wird der Bildinhalt sehr schnell beurteilbar. Um diese "interessanten" Bildstellen zu finden, können verschiedene Ansätze verfolgt werden. Zabrodsky und Peleg benutzen in [26] ein sehr einfaches

Modell für die menschliche visuelle Aufmerksamkeit, um eine Übertragungsreihenfolge der Koeffizienten festzulegen. Dabei benutzen sie Erkenntnisse aus der Psychologie und übertragen z.B. Kanten vor Grauwertplateus.

Wir verwenden dagegen einen Interaktionmechanismus, bei dem dem System ein Beispielbild vorgegeben wird, nach dem gesucht werden soll (query by example). Die Bildstrukturen, die diesem Beispielbild am ähnlichsten sind, werden zuerst übertragen. Für den Vergleich des Modell- und des Originalbildes kommen vier Verfahren zur Anwendung, die auf einem Histogrammvergleich über lokalen Orientierungsbildern basieren. In [13] und [11] wurde ein Verfahren zum Auffinden von Objekten mit Hilfe von Orientierungshistogrammen über Farbbildern vorgestellt. Da wir im Vergleich zu den dort untersuchten RGB-Bildern weniger Information zur Verfügung haben, bilden wir in dieser Arbeit Histogramme über mehrere Pyramidenstufen.

In Kapitel 2 stellen wir das Konzept der Steuerbarkeit vor und erläutern die Konzeption der steuerbaren Pyramide, die es ermöglicht, beliebige Orts- und Orientierungsinformationen aus wenigen Stützstellen zu interpolieren. Dabei werden wir auch auf die Optimierung der geeigneten Filter eingehen.

Kapitel 3 behandelt die Berechnung der lokalen Amplitude und Orientierung aus den Filterantworten der steuerbaren Pyramide, während Kapitel 4 einen auf diesen beiden Konzepten basierenden Kompressionsalgorithmus beschreibt. Wir werden dabei auch auf die Leistungsfähigkeit des vorgestellten Verfahrens eingehen.

Kapitel 5 erläutert dann die vier Vergleichsverfahren, die eine Auffälligkeitskarte erzeugen, nach der die Koeffizienten übertragen werden. Dabei wird die gleiche Repräsentation verwendet, die auch für die Codierung eingesetzt wird. Wir gehen auf zwei unterschiedliche Konzepte ein: gleitende und nicht gleitende Vergleichsverfahren.

Die Leistungsfähigkeit des vorgestellten attentionsgesteuerten progressiven Bildkompressionsverfahrens wird anschließend durch ausführliche Versuchsreihen in Kapitel 6 evaluiert. Wir führen dabei Untersuchungen durch, die sowohl auf Augen- , Personen- und Objektlokalisierung abzielen.

Kapitel 2

Steuerbare Pyramiden

In diesem Kapitel geht es um eine Transformation auf der Basis einer *steuerbaren Pyramide*, ein Begriff, der als erstes von E. Simoncelli et al. geprägt wurde [21]. Er entwickelt eine Bildpyramide, die auf *steuerbaren Filtern* aufbaut [4]. Zunächst entwickeln wir allgemein die Theorie der Steuerbarkeit einer Transformation, für die die steuerbare Pyramide ein Beispiel ist und gehen dann auf die Implementierung und Probleme beim Filterdesign ein.

2.1 Steuerbarkeit

2.1.1 Einführung

In vielen Anwendungen in der Signalverarbeitung werden Signale in einzelne Subbänder zerlegt, die dann getrennt voneinander weiterverarbeitet werden. Dabei werden die Subbänder normalerweise über der Nyquist-Rate abgetastet, um *Aliasfehler* zu vermeiden. Moderne Verfahren, die auf *Quadrature Mirror Filtern (QMF's)* oder *Wavelets* basieren, halten sich nicht an dieses Theorem, sondern tasten ein Signal kritisch ab, d.h. daß die Anzahl der Abtastpunkte in der Darstellung genauso groß ist wie im Signal. Sie verhindern das Auftreten von Aliasfehlern, indem sie sicherstellen, daß sich diese Effekte beim Rekombinieren der Subbänder aufheben.

Eine der wichtigen Eigenschaften der *Wavelettransformation* ist, daß sich die einzelnen Basisfunktionen nur um Translationen und Dilatationen unterscheiden. Man könnte also erwarten, daß sich bei einer Dilatation oder Translation des Eingangssignals die Transformationskoeffizienten ähnlich verhalten. Dies ist jedoch nicht der Fall. Selbst bei einer Translation um nur einen Abtastpunkt verändern sich Koeffizienten auf allen Subbändern. Man kann jedoch eine etwas schwächere Eigenschaft formulieren und zwar in dem Sinn, daß sämtliche Information, die durch ein Subband repräsentiert wird, auch nach einer

Translation in diesem Subband zu finden ist. Hierfür stellt das Nyquist-Kriterium eine notwendige und hinreichende Bedingung dar, denn da kritisch abgetastete Subbandtransformationen normalerweise das Nyquist-Kriterium verletzen, verschiebt sich die Information durch eine Translation von einem Subband auf ein anderes.

Die oben angesprochene abgeschwächte Eigenschaft nennen wir *Steuerbarkeit*, ein Begriff, der auch auf andere Deformationen des Eingangssignals ausgedehnt werden kann. Eine zweidimensionale Wavelettransformation kann berechnet werden, indem man Basisfunktionen benutzt, die sich in Position, Skalierung und Orientierung unterscheiden. Man kann Steuerbarkeit also auch für Dilatationen und Rotationen des Eingangssignals definieren, denn auch bei diesen zwei Deformationen verhält sich die Wavelettransformation wie bei einer Translation, d.h. daß sich auch bei diesen Deformationen Information von einem Subband auf andere verschiebt.

Konkret wollen wir also ein beliebig um θ verschobenes, rotiertes oder skaliertes Signal $f^{\theta}(x)$ aus einer endlichen Menge von N Basisfunktionen $f^{\theta_n}(x)$ interpolieren. Es soll also die folgende Gleichung gelten:

$$f^{\theta}(x) = \sum_{n=0}^{N-1} k_n(\theta) f^{\theta_n}(x),$$

wobei $k_n(\theta)$ die entsprechenden Interpolationfunktionen darstellen.

In den nächsten Abschnitten betrachten wir zunächst getrennt die Steuerbarkeit der einzelnen Deformationen und gehen dann auf die Möglichkeiten ein, die Steuerbarkeiten miteinander zu kombinieren.

2.1.2 Allgemeine Vorbemerkungen

Für die weiteren Abschnitte nehmen wir an, daß die Basisfunktionen einer Subbandtransformation aus Translationen, Rotationen und Dilatationen einer gewöhnlichen Funktion (Kern) hervorgehen. Nehmen wir an, daß dieser Kern gleichzeitig in Ortsraum, Orientierung und Skalierung lokalisiert ist, kann man jede Basisfunktion als einen Punkt (x, y, r, θ) im Parameterraum ansehen. Dabei beschreiben x und y die Lokalisierung im Ortsraum und r und θ die log-polaren Koordinaten im Frequenzraum, beziehen sich also auf die Skalierung und Orientierung. Obwohl die Parameter aus einem kontinuierlichen Wertebereich kommen können, beschränken wir uns auf eine endliche Menge von Basisfunktionen und *interpolieren* eine Filterantwort aus den Transformationskoeffizienten einer endlichen Menge von Abtastpunkten. Eine solche Interpolation ist immer möglich, wenn die Transformation invertierbar ist. Man invertiert einfach die Transformation, rekonstruiert das Signal und faltet es mit dem entsprechenden Filterkern.

Darüberhinaus möchte man einen Parameter kontinuierlich verändern können, während alle anderen fest bleiben. Zum Beispiel soll es an einem festen Ort und bei fester Skalierung möglich sein, die Filterantwort jeder beliebigen Orientierung zu interpolieren. Wir lassen in den nächsten Abschnitten also alle Parameter bis auf den zu untersuchenden fest. Bei der Steuerbarkeit im Ortsraum betrachten wir auch nur den eindimensionalen Fall; den zweidimensionalen Fall behandeln wir separat im Abschnitt über kombinierte Steuerbarkeit.

2.1.3 Steuerbarkeit im Ortsraum

Um Steuerbarkeit im Ortsraum etwas konkreter zu definieren, betrachten wir sie in einem Rahmen, der den meisten Lesern bekannt ist. Sei $f(x)$ ein kontinuierliches Signal, daß durch Faltung mit einem Kern $h(x)$ analysiert werden soll. Das Ergebnis wird anschließend abgetastet, d.h. das Ergebnis liegt nur noch an einer diskreten Anzahl von Stellen vor. Damit das Faltungsergebnis an jeder Stelle rekonstruiert werden kann, muß das Nyquist-Kriterium erfüllt sein. Die Abtastrate muß also mindestens so groß sein wie die mathematische Frequenzbandbreite des Kerns $h(x)$. Steuerbarkeit im Ortsraum korrelliert also mit dem Nicht-Vorhandensein von Aliasfehlern.

Wir leiten also eine Variante des Abtasttheorems her, bei der eine Menge von Interpolationsfunktionen benutzt werden kann, um das Steuern durchzuführen. Wir beschränken uns dabei auf kontinuierliche Eingangssignale $f(x)$, die in einer Fourierreihe entwickelt werden können, da sich dies leicht auf diskret abgetastete Signale übertragen läßt. Weiterhin nehmen wir als notationelle Vereinfachung an, daß die Signalperiode 2π beträgt. Wir berechnen eine Menge von N Transformationskoeffizienten $y[n]$, indem das Eingangssignal mit einem Kern $h(x)$ gefaltet wird und das Ergebnis anschließend äquidistant mit einer Schrittweite von $\Delta_x = \frac{2\pi}{N}$ abgetastet wird:

$$y[n] = \int_0^{2\pi} h(n\Delta_x - x) f(x) dx, \qquad n \in \{0, 1, \ldots, N-1\}. \tag{2.1}$$

Die in Gleichung 2.1 beschriebene Transformation ist steuerbar, wenn eine Menge von Interpolationsfunktionen $b_n(x_0)$ existiert, die, gewichtet mit den Transformationskoeffizienten $y[n]$, das innere Produkt des Signals mit einer beliebig verschobenen Version des Kerns $h(x)$ ergeben:

$$\int_0^{2\pi} h(x_0 - x) f(x) dx = \sum_{n=0}^{N-1} b_n(x_0) y[n], \tag{2.2}$$

wobei x_0 eine beliebige Verschiebungsgröße darstellt. Der folgende Satz (angelehnt an [4]) beschreibt eine Bedingung, unter der die Transformation steuerbar ist, und mit dessen Hilfe die Interpolationsfunktionen berechnet werden können.

Satz 1 *Die Transformation, die in Gleichung 2.1 definiert ist, ist genau dann steuerbar, wenn eine Menge von Interpolationsfunktionen $b_n(x_0)$ existiert, die die Gleichung*

$$\begin{pmatrix} e^{jx_0k_0} \\ e^{jx_0k_1} \\ \vdots \\ e^{jx_0k_{M-1}} \end{pmatrix} = \begin{pmatrix} 1 & e^{j\Delta_x k_0} & e^{j2\Delta_x k_0} & \dots & e^{j(N-1)\Delta_x k_0} \\ 1 & e^{j\Delta_x k_1} & e^{j2\Delta_x k_1} & \dots & e^{j(N-1)\Delta_x k_1} \\ & \vdots & \vdots & & \vdots \\ 1 & e^{j\Delta_x k_{M-1}} & e^{j2\Delta_x k_{M-1}} & \dots & e^{j(N-1)\Delta_x k_{M-1}} \end{pmatrix} \begin{pmatrix} b_0(x_0) \\ b_1(x_0) \\ \vdots \\ b_{N-1}(x_0) \end{pmatrix} \tag{2.3}$$

für alle x_0 erfüllt, wobei $\{k_0, k_1, \dots, k_{M-1}\}$ die Menge der Frequenzen ist, für die die Fourierreihe H(k) von $h(x)$ ungleich null ist.

Beweis siehe Anhang.

Die Verwandschaft der Bedingung in Gleichung 2.3 mit dem Nyquist-Kriterium ist offensichtlich. Die Invertierbarkeit der Matrix impliziert eine untere Schranke für die Abtastrate N und somit der Steuerbarkeit. Die Anzahl der Abtastwerte muß größer oder gleich der Anzahl der Fourierkomponenten sein, die ungleich null sind (siehe [4]).

Es ist wichtig anzumerken, daß außer der Implikation welche Fourierkoeffizienten ungleich null sind, die Interpolationsfunktionen unabhängig vom Kern $h(x)$ sind. Haben also zwei Kerne die gleichen Fourierkoeffizienten ungleich null, sind ihre Interpolationsfunktionen gleich.

Ist die Matrix in Gleichung 2.3 nicht invertierbar, kann man eine Lösung mit kleinstem quadratischen Fehler bestimmen, indem als Annäherung die Pseudo-Inverse benutzt wird. Sollte die Matrix dagegen unterbestimmt sein, findet man keine eindeutigen Interpolationsfunktionen. In diesem Fall kann entweder wiederum die Pseudo-Inverse benutzt werden, um eine Menge von Interpolationsfunktionen mit Hilfe der Minimum-Norm zu erzeugen. Alternativ dazu können auch Interpolationsfunktionen bestimmt werden, die einer weiteren Eigenschaft genügen (z.B. minimaler Einzugsbereich).

Verschiebungsinvariante Energieantwort

In einer Transformation, die Aliasfehler erzeugt, hängt die Energieantwort von der Signalposition ab, d.h. daß eine Translation des Eingangssignals eine Umverteilung der Energie auf die Subbänder bewirkt, auch wenn keine Änderung im Frequenz-Spektrum des Eingangssignals stattgefunden hat.

Die Steuerbarkeitsbedingung ist also äquivalent zu der Bedingung, daß die Energie der Subbandkoeffizienten in einem Subband erhalten bleibt, wenn das Eingangssignal verschoben wird. Wir formulieren dies in dem folgenden Satz:

Satz 2 *Sei eine Menge von Transformationskoeffizienten gegeben, die mit Gleichung 2.1 berechnet wurden. Die Transformation ist genau dann steuerbar im Ort, wenn für jedes Eingangssignal $f(x)$ die Energie der Koeffizienten $\sum_{n=0}^{N-1} |y[n]|^2$ invariant unter Translationen von $f(x)$ ist.*

Beweis Wir zeigen die Behauptung durch eine Betrachtung der diskreten Fourier-Transformation (DFT) $Y(k)$ von $y[n]$. Benutzt man übliche Erkenntnisse der Abtastung, so kann man schreiben:

$$Y(k) = \sum_l H(k + lN) \cdot F(k + lN), \tag{2.4}$$

wobei $F(k)$ die Fourierreihe des Eingangssignals und $H(k)$ die Fourierreihe des Faltungskerns ist. Der Index l läuft über die Menge der ganzen Zahlen. Die Summe korrespondiert mit der Wiederholung des Spektrums, welche bei der Abtastung des Signals auftritt.

Nehmen wir an, daß die Transformation steuerbar ist. Wendet man Parseval's Theorem an, so können wir die Energie des Ausgangssignals mit Hilfe der DFT wie folgt ausdrücken:

$$\begin{aligned}
\sum_{n=0}^{N-1} |y[n]|^2 &= \sum_{k=0}^{N-1} |Y(k)|^2 \\
&= \sum_{k=0}^{N-1} \left| \sum_l H(k + lN) \cdot F(k + lN) \right|^2 \\
&= \sum_{k=0}^{N-1} \sum_l |H(k + lN) \cdot F(k + lN)|^2 + \\
&\quad \sum_{k=0}^{N-1} \sum_{l \neq m} H(k + lN) \cdot F(k + lN) \cdot H^*(k + mN) \cdot F^*(k + mN),
\end{aligned} \tag{2.5}$$

wobei * für komplexe Konjugation steht und die Indizes l und m über die ganzen Zahlen laufen. Da wir Steuerbarkeit annehmen, es also keine Aliasfehler gibt, ist der zweite Term null. Der verbleibende Teil ist invariant unter Translationen des Eingangssignals:

$$\sum_{n=0}^{N-1} |y[n]|^2 = \sum_{k=0}^{N-1} \sum_l |H(k + lN)|^2 |F(k + lN)|^2. \tag{2.6}$$

Das heißt, daß das Ersetzen von $F(k)$ durch $e^{jkx_0}F(k)$ für beliebiges x_0 auf der rechten Seite der Gleichung den Ausdruck nicht verändert, die Energieantwort also invariant unter Translation ist.

Die Umkehrung ist ebenfalls richtig. Nehmen wir an, daß das System nicht steuerbar ist. Dann erzeugt die Abtastung Aliasfehler. Also tritt im zweiten Term von Gleichung 2.5 ein sinusförmiges Signal an einer Stelle auf, an der ein Aliasfehler zu sehen ist (d.h. es existieren k, l und m, für die das Produkt im zweiten Term nicht null ist). Wird das Eingangssignal verschoben, wird dieser zweite Term (ungleich null) durch den Phasenfaktor moduliert. □

Sollte es unmöglich oder unpraktikabel sein, Steuerbarkeit in einem Parameter zu erreichen, kann eine einfachere Bedingung benutzt werden. Wir nennen diese Eigenschaft *glatte Basisenergie* in Bezug auf einen Parameter. Betrachtet man z.B. den Ortsparameter, bedeutet dies, daß die Summe der Energie der Basisfunktionen für ein Subband konstant als Funktion über dem Ort ist. Diese Eigenschaft ist eine direkte Folgerung aus der Steuerbarkeit, wohingegen die Rückrichtung im allgemeinen nicht gilt.

Korollar 3 *Wenn die Transformation, die durch Gleichung 2.1 gegeben ist, steuerbar ist, ist die Quadratsumme der Basisfunktionen $\sum_n h_n^2(x)$ konstant für alle x.*

Beweis Wir zeigen die Behauptung, indem wir das Impuls-Eingangssignal $F(k) = e^{jkx_0}$ betrachten. Dann ist $y[n]$ der Wert der n-ten Basisfunktion am Ort des Impulses. In diesem Fall ist die linke Seite der Gleichung 2.6 gleich der Summe der Quadrate einer Untermenge der Kern-Abtastpunkte mit Abstand der Abtastbreite:

$$\sum_{n=0}^{N-1} |y[n]|^2 = \sum_n |h(x_0 - n\Delta_x)|^2.$$

Die rechte Seite von Gleichung 2.6 ist die Summe über alle Frequenzen des Kern-Energiespektrums, d.h. eine Konstante. Also muß die Summe der Quadrate der Basisfunktionen an jedem Ort die gleiche Konstante ergeben:

$$\sum_n |h(x - n\Delta_x)|^2 = c, \qquad \text{für alle } x.$$

□

Die glatte Basisenergie einer Transformation ist folglich eine notwendige, aber keine hinreichende Bedingung für Steuerbarkeit im Ortsraum. Als Beispiel betrachten wir die Faltung mit dem Kern

$$h(x) = \begin{cases} 1, & x \in [0,1] \\ 0, & \text{sonst} \end{cases}$$

und anschließender äquidistanter Abtastung mit $\Delta x = 1$. Das System hat eindeutig glatte Basisenergie, da die Energie jeder Basisfunktion über dem entsprechenden Intervall eins ist und die Basisfunktionen sich nicht überlappen. Trotzdem ist diese Menge nicht steuerbar, denn das Kernspektrum ist unendlich in Ausdehnung, woraus folgt, daß das Abtasten der Basisfunktionen Aliasfehler erzeugt.

Wir benutzen diese Eigenschaft, um in Abschnitt 2.2 eine zweidimensionale Transformation zu entwickeln.

2.1.4 Steuerbarkeit der Orientierung

Das im letzten Abschnitt beschriebene Prinzip der Steuerbarkeit kann auch auf Orientierungen angewendet werden. Ist eine Menge von Basisfunktionen gegeben, die den gleichen Ort und die gleiche Skalierung, aber endlich viele Orientierungen aufweisen, möchten wir jede beliebige Orientierung interpolieren können. Diese Idee wurde ausführlich von Freeman und Adelson [4] behandelt, die auch darauf eingehen, wieviele rotierte Basisfunktionen benötigt werden, um die Filterantwort einer beliebigen Orientierung zu berechnen. Des weiteren leiten sie auch die Interpolationsformel 2.3 her, die wir im vorherigen Abschnitt schon benutzt haben.

Wir müssen die rotierten Basisfunktionen also nur als Abtastpunkte im Orientierungsraum betrachten und können ganz analog zum vorherigen Abschnitt die gesuchten Basisfunktionen berechnen. Ebenso gilt auch die Bedingung, daß die Summe der Quadrate der Transformationskoeffizienten an einem festen Ort und einer festen Skalierung invariant ist, wenn die Orientierung des Eingangssignals verändert wird.

Die einfachste Möglichkeit, Basisfunktionen zu berechnen, stellen $N{+}1$ N-te Richtungsableitungen einer polar-separablen Funktion dar. Die Richtungsableitungen sind also rotierte Kopien derselben Funktion. Die Angularkomponente der Richtungsableitung besteht aus $\frac{N+1}{2}$ verschiedenen Harmonischen, wenn N ungerade ist, und aus $\frac{N}{2}$ Harmonischen, wenn N gerade ist. Nach dem Abtasttheorem genügen also $N+1$ Abtastpunkte des Orientierungsparameters, um die Phasenkomponente vollständig zu beschreiben.

In dieser Arbeit beschränken wir uns auf Basisfunktionen, die mit Hilfe der N-ten Richtungsableitung berechnet werden. Die Theorie der Steuerbarkeit ist jedoch nicht auf solche Funktionen begrenzt. Sie kann auf beliebige polar-separable Funktionen ausgedehnt werden, die eine bandbegrenzte Angularkomponente haben. Diese Filter, die *steuerbare Wedgefilter* genannt werden, wurden ursprünglich entwickelt, um die Symmetrie der orientierten Energie bei den auf Richtungsableitungen basierenden steuerbaren Filtern zu entfernen. Diese Symmetrie führt z.B. am Ende einer Linie zu einer bimodalen Filterantwort. Auch an Ecken oder T-Kreuzungen werden von den Filtern, die auf N-ten Richtungsableitungen basieren, fehlerhafte Filterantworten erzeugt. Steuerbare Wedgefilter antworten auch an diesen Stellen korrekt, sind jedoch deutlich aufwendiger. Weitere Details zu diesen Filtern können in [17] und [18] nachgelesen werden.

2.1.5 Steuerbarkeit der Skala

Die Anwendung des Konzeptes der Steuerbarkeit auf Skalierungen ist schwieriger als die Anwendung auf Translationen und Rotationen, da wir dabei eine Symmetriebedingung haben und wir den Skalenparameter auf einer logarithmischen Achse abbilden wollen. Um das Problem ein wenig zu vereinfachen, nehmen wir an, daß die Funktionen im Frequenzraum symmetrisch oder antisymmetrisch zum Ursprung sind. Um diese Symmetriebedingung fassen zu können, konstruieren wir wie folgt eine *reflektiert steuerbare* Funktion.

Satz 4 *Es sei eine steuerbare Repräsentation gegeben, die auf der Funktion $H(r)$ mit den Interpolationsfunktionen $b_n(r_0)$ basiert. Dann können wir die Funktion*

$$U_{\pm}(r, r_0) = H(r - r_0) \pm H(-r - r_0)$$

aus der Menge der Funktionen

$$U_{\pm}(r, n\Delta_r) = H(r - n\Delta_r) \pm H(-r - n\Delta_r)$$

mit $n \in \{0, 1, \ldots, N-1\}$ berechnen, wobei wir die gleichen Interpolationsfunktionen benutzen. Wir sagen, daß $U_{\pm}(r, r_0)$ reflektiert steuerbar *ist.*

Beweis siehe Anhang.

Normalerweise sind die Signale, die wir hier betrachten wollen, bandbegrenzt, d.h. wir können den Wertebereich für r auf ein bestimmtes Intervall $r \in [-\omega_{max}, \omega_{max}]$ begrenzen. Da wir Steuerbarkeit für periodische Signale betrachtet haben, können wir das Intervall ebenfalls periodisch machen, indem wir die Intervallgrenzen entsprechend setzen.

Ein weiteres Problem stellt dar, daß die oben beschriebenen reflektiert steuerbaren Kerne mit gleich breiten Frequenzbändern korrespondieren. Betrachten wir aber eine Menge von Basisfunktionen, die durch Dilatationen aus einem Kern entstehen, so sind die Fouriertransformierten der Basisfunktionen verschobene Kopien des transformierten Kerns mit logarithmischem Abstand. Um also auf einer logarithmischen Achse gleich breite Frequenzbänder zu erhalten, müssen wir den Wertebereich verzerren. Haben wir also z.B.

$$U(r, r_0) = \sum_n b_n(r_0) U(r, n\Delta_r),$$

so gilt trivialerweise für jede Verzerrungsfunktion $\rho(r)$:

$$U(\rho(r), r_0) = \sum_n b_n(r_0) U(\rho(r), n\Delta_r).$$

Dabei ist anzumerken, daß durch das Verzerren weder die Steuerbarkeit noch die Eigenschaft der glatten Basisenergie (siehe Korollar 3) beeinträchtigt wird. Da wir an auf Skalierung abgestimmten Funktionen interessiert sind, ist $\rho(r) = \log(r)$ die richtige Verzerrungsfunktion, was allerdings eine Singularität im Ursprung bewirkt. Dies macht es notwendig, die Funktion um den Ursprung herum abzuändern, wodurch die Steuerbarkeit jedoch nicht beeinträchtigt wird. Die Änderung bewirkt nur, daß die Basisfunktionen nicht fehlerfrei durch Dilatationen des Kerns erzeugt werden können.

2.1.6 Kombinierte Steuerbarkeit

In den vorherigen Abschnitten haben wir die Steuerbarkeit einzelner Parameter untersucht. Für einige Anwendungen ist es allerdings wünschenswert, exakte Steuerbarkeit in mehreren Parametern gleichzeitig zu haben. Dies ist jedoch nur möglich, wenn zwei Parameter, die beide gesteuert werden sollen, unabhängig voneinander sind. Wir müssen also überprüfen, ob es Abhängigkeiten zwischen den einzelnen Parametern gibt.

Dabei ist festzustellen, daß es möglich ist, eine zweidimensionale Transformation gleichzeitig in x, y und der Orientierung exakt zu steuern. Probleme treten nur auf, wenn man Parameter betrachtet, die im Fourierraum komplementär sind, wie Orts- und Skalierungsparameter. Als Beispiel hierfür betrachten wir eine eindimensionale Transformation mit Basisfunktionen für Ort und Skalierung. Örtliche Steuerbarkeit impliziert, daß die Basisfunktionen im Fourierraum eine begrenzte Bandbreite (Wirkungsbereich) haben. Das bedeutet also, daß die Basisfunktionen im Ortsraum einen unendlichen Wirkungsbereich haben. Andererseits müssen sie wegen der Steuerbarkeit der Skalierung im Ortsraum einen kompakten Wirkungsbereich haben. Es ist also unmöglich, eine Transformation zu entwickeln, die gleichzeitig exakt steuerbar in diesen zwei komplementären Parametern ist.

Um einen Ausweg aus diesem Dilemma zu finden, gibt es mehrere Möglichkeiten. Die einfachste besteht darin, den kontinuierlichen Wertebereich eines Parameters beizubehalten. Diese Lösung ist aber für viele Anwendungen ineffizient und somit nicht akzeptabel.

Eine andere Möglichkeit ist, eine approximiert steuerbare Darstellung zu finden. Dazu muß man ein Maß der kombinierten Nicht-Steuerbarkeit einführen, mit dem dann bei fester Abtastung Basisfunktionen gefunden werden können, die dieses Maß minimieren. Die *Singulärwertzerlegung* (SVD) ist ein Ansatz, mit dem eine approximiert steuerbare Darstellung berechnet werden kann. Eine ausführlichere Darstellung diese Ansatzes kann in [12] nachgelesen werden.

Wir werden im folgenden als weitere Möglichkeit versuchen, die Steuerbarkeitsbedingung abzuschwächen. Dafür ersetzt man die Steuerbarkeitsbedingung in einem Parameter durch die weniger restriktive Bedingung der glatten Basisenergie (siehe Korollar 3). Wir werden im nächsten Abschnitt feststellen, daß eine solche Ersetzung zu einer involutorischen Transformation führt.

2.1.7 Involutionseigenschaften

Für eine Transformation in der Bildverarbeitung und insbesondere in der Bildcodierung ist eine wichtige Eigenschaft, daß sie involutorisch ist. Jeder Transformationskoeffizient wird mit Hilfe einer Projektionsfunktion berechnet, die jeweils eine feste Position und Form sowohl im Orts- als auch im Frequenzraum hat. Ist die Transformation involutorisch, sind die Basisfunktionen für die Rücktransformation die gleichen wie bei der Transformation. Ist diese Eigenschaft nicht gegeben, entstehen durch die nicht lineare Berechnung der Koeffizienten bei der Rücktransformation Fehler in Orts- und Frequenzparametern.

Jede orthonormale Transformation ist involutorisch. Damit jedoch eine involutorische Transformation orthonormal ist, muß sie zusätzlich linear unabhängig sein. Man kann also eine Transformation konstruieren, die involutorisch, aber überbestimmt ist.

Ein einfaches Beispiel für eine solche Transformation ist eine orthogonale Transformation, die, durch Duplizieren jeder Projektionsfunktion und dividieren der Kern-Gewichte durch $\sqrt{2}$, nicht orthogonal und involutorisch gemacht wird. Die so entstandene Transformation ist eine überbestimmte Darstellung, bei der die zwei identischen Basisfunktionen die gleiche Information beinhalten. Die Transformation ist involutorisch, da Basis- und Projektionsfunktionen identisch sind. Ein weiteres komplizierteres Beispiel ist die steuerbare Pyramide, die wir in Abschnitt 2.2 beschreiben.

In den vorherigen Abschnitten haben wir die Transformationsparameter getrennt voneinander betrachtet. Nun müssen wir die Transformation als Ganzes betrachten. Wir beschränken uns dabei auf Transformationen, die mit Hilfe von Analyse-/Synthese-Bänken dargestellt werden können. Analyse-/Synthese-Bänke sind allgemein genug, um beliebige lineare Transformationen darzustellen, allerdings wollen wir hier nur Subbandtransformationen wie die Wavelettransformation betrachten.

Wie in den vorherigen Abschnitten nehmen wir auch hier an, daß unser Eingangssignal 2π-periodisch ist. Zusätzlich benötigen wir die Annahme, daß das Eingangssignal auf eine endliche Menge von Frequenzen $k \in [-k_{max}, k_{max}]$ bandbegrenzt ist. Das Eingangssignal wird mit einer Menge von M Kernen $h_m(x)$ gefaltet, wobei der Index m das Frequenzband des Kerns bezeichnet. Die Ergebnisse dieser Faltungen wird an N_m Punkten mit Abstand $\frac{2\pi}{N_m}$ abgetastet, wodurch wir die Menge der Sequenzen von Transformationskoeffizienten $y_m[n]$ erhalten.

Im Synthese-Teil werden die Koeffizienten benutzt, um ein kontinuierliches Signal zu rekonstruieren, indem sie mit Dirac-Delta-Funktionen an den entsprechenden Abtastpunkten gewichtet werden, d.h. wir berechnen aus dem diskreten Signal $y_m[n]$ das kontinuierliche Signal $\gamma_m(x) = \sum_{n=0}^{N_m-1} y_m[n] \cdot \delta(x - n\Delta_x)$. Diese Signale werden dann mit den gespiegelten Kernen $g_m(-x)$ gefaltet. Die Negation erlaubt eine konsistente Definition der Basis- und Projektionsfunktionen.

Ist diese Struktur gegeben, kann der folgende Satz gezeigt werden:

Satz 5 *Ist eine Subbandtransformation in jedem Subband örtlich steuerbar und hat diese im komplementären Fourierraum glatte Basisenergie über alle Frequenzen, so ist sie involutorisch.*

Beweis Wir beweisen den Satz, indem wir die Konstruktion für den eindimensionalen Fall angeben. Die Erweiterung auf höhere Dimensionen ist dann einfach. Wir nehmen an, daß die Voraussetzungen gelten und daß Basis- und Projektionsfunktionen gleich sind, d.h. $g_m(x) = h_m(x)$. Dann ist zu zeigen, daß die beschriebene Transformation die Identität beschreibt. Benutzen wir Gleichung 2.4, können wir das Ergebnis der ganzen Transformation wie folgt beschreiben:

$$\begin{aligned}
\hat{F}(k) &= \sum_{m=0}^{M-1} \left[\sum_{l=0}^{N_m-1} H_m(k + lN_m) \cdot F(k + lN_m) \right] H_m(-k) \\
&= \sum_{m=0}^{M-1} H_m(k) H_m(-k) \cdot F(k) + \\
&\quad \sum_{m=0}^{M-1} \sum_{l=1}^{N_m-1} H_m(k + lN_m) H_m(-k) F(k + lN_m).
\end{aligned}$$

Die zweite Summe in dieser Gleichung beschreibt den Aliasteil. Da wir aber annehmen, daß das System steuerbar im Ortsraum ist, ist dieser Teil null. Die verbleibende Systemausgabe ist dann

$$\hat{F}(k) = \left[\sum_{m=0}^{M-1} |H_m(k)|^2 \right] \cdot F(k).$$

Ist die Menge der $H_m(k)$ so entworfen, daß sie flache Energie hat, gilt

$$\sum_{m=0}^{M-1} |H_m(k)|^2 = c$$

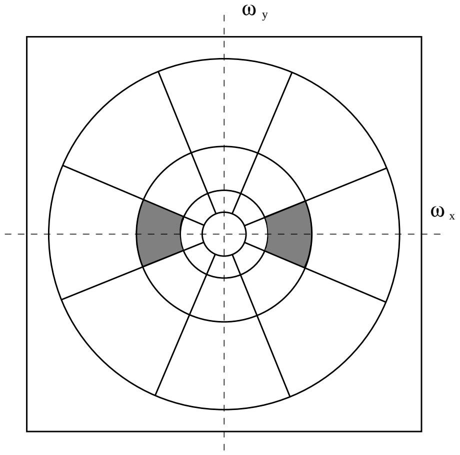


Abbildung 2.1: Idealisierte Darstellung der spektralen Aufteilung, die eine steuerbare Pyramide mit $k = 4$ Orientierungen bewirkt. Die Basisfunktionen sind durch Translationen, Rotationen und Dilatationen ähnlich (außer dem anfänglichen Hochpaß- und dem abschließenden Tiefpaßanteil). Die graue Region entspricht der spektralen Antwort eines vertikal orientierten Subbands.

für alle k.

Nehmen wir an, daß c eine Einheit ist, was durch eine einfache Normierung der Basisfunktionen auch leicht erreicht werden kann, so berechnet die Transformation die Identität und ist somit involutorisch. □

Mit Hilfe dieses Satzes werden wir nun im nächsten Abschnitt eine involutorische Pyramiden-Transformation konstruieren, die in Ort und Orientierung steuerbar ist.

2.2 Steuerbare Pyramiden

In diesem Abschnitt wird eine Transformation vorgestellt, die die Vorteile der Steuerbarkeit in verschiedenen Parametern ausnutzt. Die steuerbare Pyramide ist eine zweidimensionale Transformation, die kombiniert steuerbar in Orts- und Orientierungsparametern ist. Zusätzlich weist sie im Skalierungsparameter eine glatte Basisenergie auf, wodurch impliziert wird, daß die Transformation involutorisch ist.

2.2.1 Grundlagen

Wir teilen ein Bild mit Hilfe eines linearen Verfahrens in Subbänder auf, die in Skala und Orientierung lokalisiert sind. Dabei ist die Berechnung der einzelnen Skalen durch ein rekursives System beschränkt. Als Basisfilter für die Orientierung werden $N + 1$

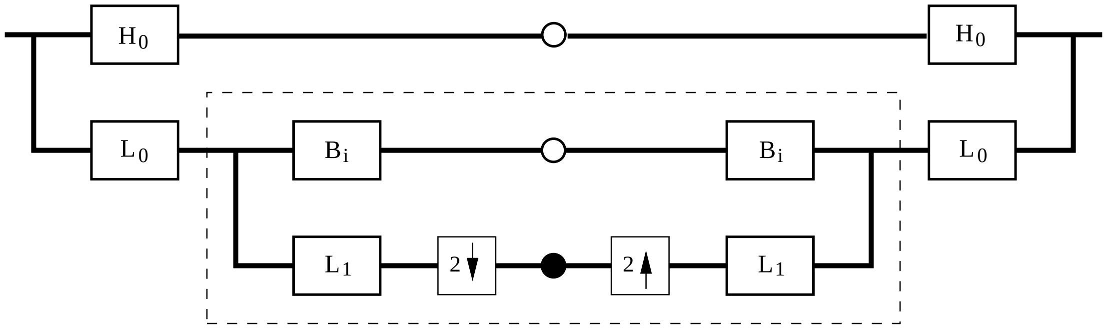


Abbildung 2.2: System-Diagramm für den radialen Teil der steuerbaren Pyramide. Die beiden Kästen rechts und links vom ausgefüllten Kreis beschreiben Unter- bzw. Überabtastung um Faktor 2. Alle anderen Kästen beschreiben 2D-Faltungen. Für einen rekursiven Schritt wird der gestrichelte Kasten an der Stelle des ausgefüllten Kreises eingesetzt.

N-te Richtungsableitungen benutzt. Dies stellt die einfachste Möglichkeit dar, einen steuerbaren Basisfiltersatz zu wählen [4]. Die Transformation ist so konstruiert, daß man eine beliebige Anzahl k von Orientierungsbändern berechnen kann, was jedoch auch bedingt, daß die entstehende Transformation um den Faktor $\frac{4k}{3}$ überbestimmt ist. Diese Überbestimmtheit begrenzt die Effizienz der Berechnung, ein Nachteil, der jedoch durch die große Flexibilität der steuerbaren Pyramide wieder ausgeglichen wird.

Die konstruierte Transformation ist im Fourierraum (ideal) polar-separabel. Eine Aufteilung der Frequenzbänder am Beispiel von vier Orientierungen und drei Pyramidenstufen ist in Abbildung 2.1 dargestellt.

Zunächst wird das Bild also mit einem nicht orientierten Hochpaßfilter $H_0(\vec{\omega})$ und einem nicht orientierten Tiefpaßfilter $L_0(\vec{\omega})$ für die Rekursion vorbereitet (siehe Abbildung 2.2). Der Tiefpaßanteil wird dann mit k orientierten Bandpaßfiltern $B_k(\vec{\omega})$ gefiltert, wodurch wir die Basisfilterantworten erhalten. Der Tiefpaßanteil wird ebenfalls mit einem schmalen Tiefpaßfilter $L_1(\vec{\omega})$ gefiltert und dann um den Faktor zwei in x- und y-Richtung unterabgetastet. Für einen Rekursionsschritt wird dann der gestrichelte Kasten an der Stelle mit dem ausgefüllten Kreis eingesetzt.

Betrachtet man das Diagramm in Abbildung 2.2 näher, so stellt man fest, daß für das rekonstruierte Bild im Frequenzraum die folgende Gleichung gilt:

$$\hat{X}(\vec{\omega}) = \left(|H_0(\vec{\omega})|^2 + |L_0(\vec{\omega})|^2 (|L_1(\vec{\omega})|^2 + \sum_{k=0}^{n} |B_k(\vec{\omega})|^2) \right) X(\vec{\omega}) + a.t.,$$

wobei a.t. den Aliasterm beschreibt. Um zu garantieren, daß keine Aliasfehler auftreten, muß der Filter $L_1(\vec{\omega})$ so beschränkt sein, daß seine Übertragungsfunktion für Radialfrequenzen größer als $\frac{\pi}{2}$ null ist. Weiterhin sollte die Transferfunktion des ganzen Systems eins sein, damit keine Amplitudenverzerrungen auftreten.

Im folgenden beschreiben wir die Bedingungen, die an die Filter gestellt werden, genauer. Dabei gehen wir auf Bedingungen sowohl für die Steuerbarkeit in verschiedenen Parametern als auch für die rekursive Struktur ein.

2.2.2 Bedingungen für die perfekte Rekonstruktion

Um eine perfekte Rekonstruktion garantieren zu können, müssen, wie oben angesprochen, einige Bedingungen gelten. Wir beschreiben diese in Anlehnung an das Systemdiagramm in Abbildung 2.2:

1. Zunächst beschreiben wir die Bedingung, daß es keine Amplituden-Verzerrungen geben darf:

$$|L_0(\vec{\omega})|^2 \left[|L_1(\vec{\omega})|^2 + \sum_{k=0}^{n} |B_k(\vec{\omega})|^2 \right] + |H_0(\vec{\omega})|^2 = 1. \tag{2.7}$$

2. Der Tiefpaßteil in der Abbildung darf durch das Einfügen des Rekursionsblocks des Systems nicht verändert werden:

$$|L_1\left(\frac{\vec{\omega}}{2}\right)|^2 \left[|L_1(\vec{\omega})|^2 + \sum_{k=0}^{n} |B_k(\vec{\omega})|^2 \right] = |L_1\left(\frac{\vec{\omega}}{2}\right)|^2. \tag{2.8}$$

3. Als letztes muß noch eine Bedingung definiert werden, die das Entstehen von Aliasfehlern beschränkt:

$$L_1(\vec{\omega}) = 0, \qquad \text{für } \ |\vec{\omega}| > \frac{\pi}{2}. \tag{2.9}$$

2.2.3 Angulare Bedingungen

Die Bedingungen, die an die orientierten Bandpaßfilter gestellt werden, leiten sich aus der Forderung der Steuerbarkeit im Orientierungsparameter ab. Wir benutzen $N+1$ N-te Richtungsableitungen. Die Bedingung kann wie folgt ausgedrückt werden:

$$B_k(\vec{\omega}) = B(\vec{\omega})[-j\cos(\theta - \theta_k)]^N, \tag{2.10}$$

wobei $\theta = \arg(\vec{\omega})$, $\theta_k = \frac{\pi k}{N+1}$ für $k \in \{0, 1, \ldots, N\}$ und

$$B(\vec{\omega}) = \sqrt{\sum_{k=0}^{N} |B_k(\vec{\omega})|^2}.$$

Die Gleichung 2.10 beschreibt also, daß $B_k(\vec{\omega})$ die N-te Richtungsableitung der Funktion $\frac{B(\vec{\omega})}{|\vec{\omega}|^N}$ in Richtung θ_k ist.

2.2.4 Filterdesign

In diesem Abschnitt werden wir einen Algorithmus beschreiben, mit dem es möglich ist, Filter für eine steuerbare Pyramide mit den oben genannten Bedingungen zu entwickeln. In [19] wurden Filter mit Hilfe von gewichteten least-square Methoden im Frequenzraum berechnet. Dieser Ansatz kann hier jedoch nicht benutzt werden, da die Gleichungen 2.7 und 2.8 zu einer Fehlerfunktion führen, die nicht quadratisch in den Filterkoeffizienten ist. Der direkte Versuch, eine gewichtete Kombination der Bedingungen aus Gleichung 2.7 bis 2.10 zu minimieren, konvergiert nach [10] nicht.

Wir treffen also weitere Annahmen, um den Entwicklungsprozeß zu vereinfachen. Zunächst kann man feststellen, daß die Gleichungen 2.7 und 2.8 sehr ähnlich sind. In einem Rekursionsschritt übernimmt $L_1\left(\frac{\vec{\omega}}{2}\right)$ die Rolle des vorverarbeitenden Filters $L_0(\vec{\omega})$, weswegen wir im Entwicklungs-Algorithmus auch $L_0(\vec{\omega}) = L_1\left(\frac{\vec{\omega}}{2}\right)$ setzen.

Diese Beschränkung impliziert einen rekursiven Entwicklungsalgorithmus, wie er in Abbildung 2.3 dargestellt ist. Dabei wird zunächst die Anzahl der Orientierungen $N+1$, die Größe der Filtermasken und der initiale Tiefpaßfilter $L_0(\vec{\omega})$ gewählt. Dann werden die Filter $L_1(\vec{\omega})$ und $B_k(\vec{\omega})$ entworfen, indem man als Fehlermaß eine gewichtete Summe der maximalen absoluten Fehler der Bedingungen 2.8, 2.9 und 2.10 benutzt. Der resultierende Filter $L_1(\vec{\omega})$ wird verkleinert, um einen neuen Filter $L_0(\vec{\omega})$ zu erhalten, worauf der beschriebene Prozeß wiederholt wird. In jedem Iterationsschritt wird der Filter $L_0(\vec{\omega}) = L_1\left(\frac{\vec{\omega}}{2}\right)$ als eine Frequenz-Gewichtungsfunktion für die Berechnung der Filter $L_1(\vec{\omega})$ und $B_k(\vec{\omega})$ benutzt (siehe Gleichung 2.8). Konvergiert die Iteration, kann der Filter $H_0(\vec{\omega})$ mit Hilfe von Gleichung 2.7 leicht berechnet werden. In [10] wurde bemerkt, daß die Iteration schnell, d.h. typischerweise nach drei Schritten, konvergiert.

Die Iteration wird initialisiert, indem Filterkerne bestimmt werden, die die gewichtete Summe der durchschnittlichen Quadratfehler der Bedingungen aus den Gleichungen 2.8, 2.9 und 2.10 minimieren. Innerhalb der Rekursion wird eine minmax-Fehlerfunktion benutzt, d.h. eine gewichtete Summe der maximalen absoluten Fehler der Bedingungen. Dabei wird der Rekursions-Bedingung die doppelte Gewichtung gegenüber den beiden anderen Bedingungen gegeben.

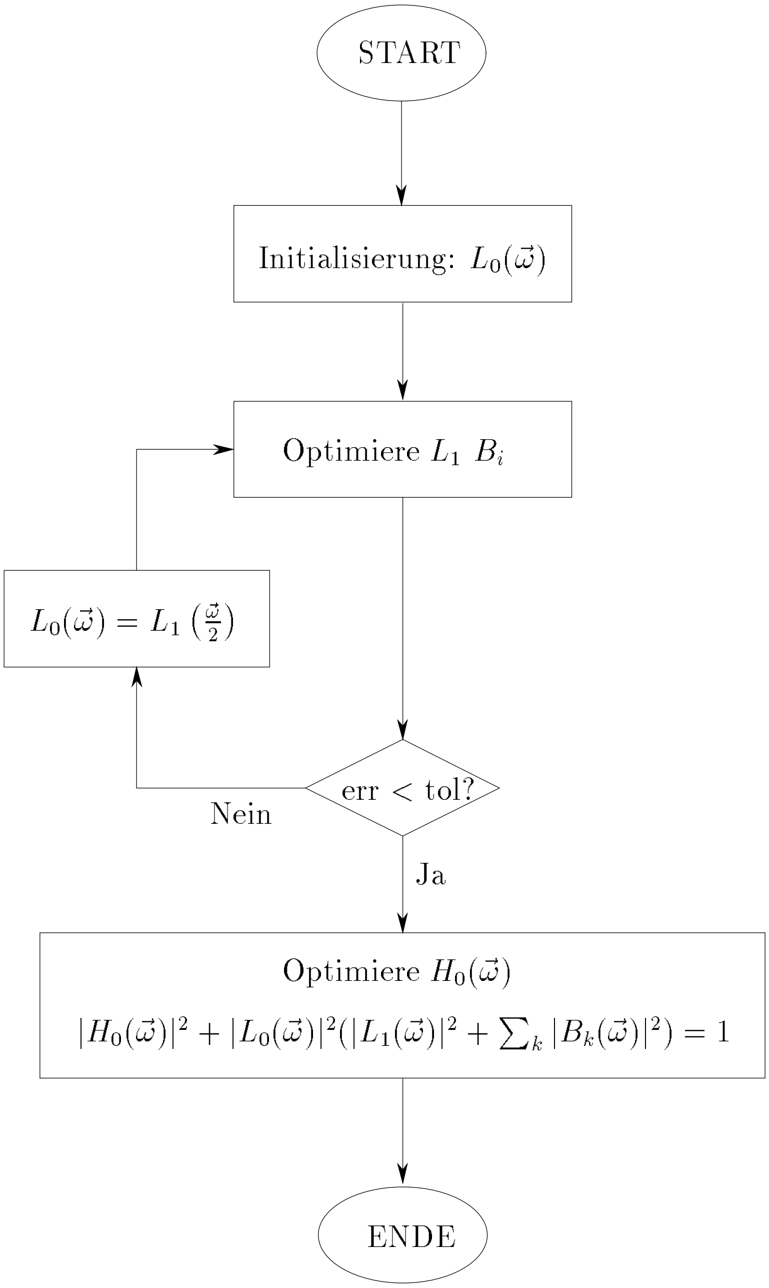


Abbildung 2.3: Flußdiagramm für den Design-Algorithmus der Filter der steuerbaren Pyramiden

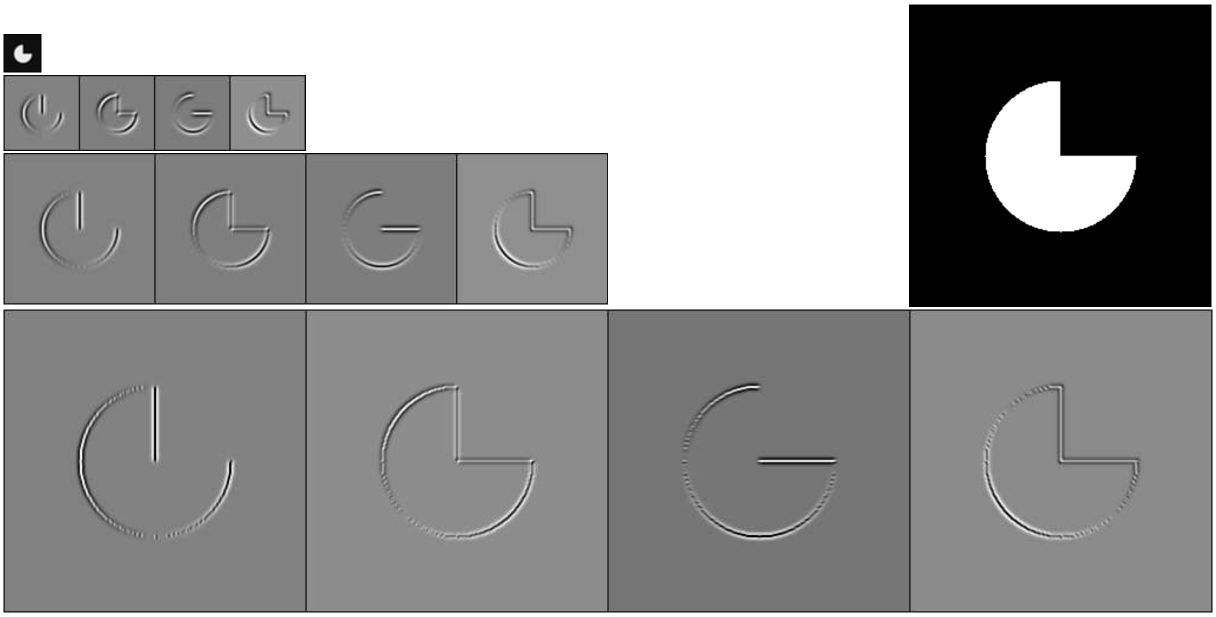


Abbildung 2.4: Beispiel für die Struktur einer steuerbaren Pyramide mit drei Stufen und vier Orientierungen. Rechts oben ist das Original abgebildet. Links oben: Koeffizientenbild des verbleibenden Tiefpaßes. Darunter: Bandpaß-Koeffizientenbilder der Pyramide.

In jedem Rekursionsschritt (siehe Abbildung 2.3) wird für die numerische Optimierung der Filter $L_1(\vec{\omega})$ und $B_k(\vec{\omega})$ ein Breadth First Search Algorithmus verwendet (BFGS-Methode mit kubischer und quadratischer Reihensuche). Diese Methode ist in hochdimensionalen Räumen sehr effizient. Im Fall der ersten Ableitung, ist die Dimension für einen 17×17-Filter $L_1(\vec{\omega})$ und 9×9-Filter $B_k(\vec{\omega})$ bereits 60. Dabei wird angenommen, daß die Filter lineare Phasenfilter sind. Ebenso werden soweit es möglich ist Symmetrieeigenschaften ausgenutzt, d.h. daß die zwei Bandpaßfilter rotierte Kopien voneinander sind. Somit muß nur über die unabhängigen Koeffizienten von einem Bandpaßfilter optimiert werden.

Entwicklung von steuerbaren Pyramidenfiltern höherer Ordnung

Der oben beschriebene Algorithmus iteriert Filter für eine steuerbare Pyramide, die auf der ersten Richtungsableitung basiert. Möchte man eine höhere Winkelauflösung erreichen, ist es nötig, höhere Richtungsableitungen zu benutzen. In diesem Fall beschäftigen wir uns mit der dritten Richtungsableitung, durch die vier Bandpaßfilter impliziert werden. Die Symmetrieachsen der Filter $B_k(\vec{\omega})$ sind bei einer Orientierung von $45k$ Grad angeordnet. Da wir hier jedoch ein rechtwinkliges Abtastgitter benutzen, tritt durch eine solche Anordnung der Symmetrieachsen ein Problem auf: Der Filter $B_1(\vec{\omega})$ kann nicht durch Rotation um 45 Grad aus dem Filter $B_0(\vec{\omega})$ gewonnen werden. Dagegen erhält man den Filter $B_2(\vec{\omega})$ aus $B_0(\vec{\omega})$ durch Rotation um 90 Grad, ebenso wie $B_3(\vec{\omega})$ aus $B_1(\vec{\omega})$. Aus diesem Grund müssen wir in der Optimierungsfunktion unabhängige Koeffizienten für zwei der vier Bandpaßfilter berücksichtigen.

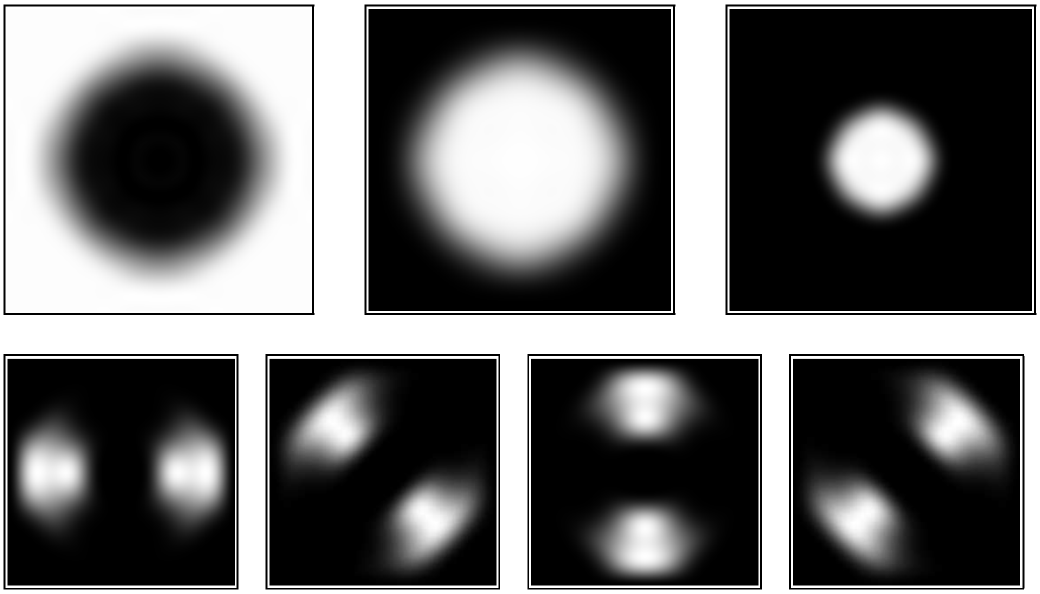


Abbildung 2.5: Fourier-Spektren von steuerbaren Pyramidenfiltern der dritten Richtungsableitung. Obere Reihe: Hochpaßfilter $H_0(\vec{\omega})$, Tiefpaßfilter $L_0(\vec{\omega})$, Tiefpaßfilter $L_1(\vec{\omega})$. Untere Reihe Bandpaßfilter $B_k(\vec{\omega})$.

Durch diese Vermehrung der Koeffizienten steigt die Dimension des Optimierungsraumes im Vergleich zum Fall der ersten Richtungsableitung sprunghaft an. Deshalb ist der Design-Algorithmus der ersten Richtungsableitung sowohl in Effizienz als auch in der Verläßlichkeit der Lösung unpraktikabel. Um dieses Problem zu lösen, wird der Filter $L_1(\vec{\omega})$ (und damit auch $L_0(\vec{\omega})$ als unterabgetastete Version von $L_1(\vec{\omega})$) aus dem Fall der ersten Richtungsableitung genommen. Die Optimierung erfolgt hier also nur für die Filter $B_0(\vec{\omega})$ und $B_1(\vec{\omega})$.

Beispiel-Filter

In [10] werden steuerbare Pyramidenfilter für $N = 0$, $N = 1$ und $N = 3$ angegeben. Wir verwenden hier nur Filter mit vier Orientierungen, also mit $N = 3$. In Abbildung 2.5 sind die Fourierspektren der Filter $H_0(\vec{\omega})$, $L_0(\vec{\omega})$, $L_1(\vec{\omega})$ und $B_k(\vec{\omega})$ (mit $k = 0, \ldots, 3$) abgebildet. Die Filter $B_k(\vec{\omega})$ und $L_0(\vec{\omega})$ haben eine Größe von 9×9 und $L_1(\vec{\omega})$ von 17×17.

Eine Anwendung dieser Filter ist in Abbildung 2.4 dargestellt. Das Originalbild (rechts oben) wird in eine Pyramidenrepräsentation transformiert. Bei dem dargestellten Beispiel fehlt das Hochpaß-Koeffizientenbild. Der Rekonstruktionsfehler liegt bei den oben dargestellten Filtern bei ca. 45dB unter dem Signal.

Der hier vorgestellte Filtersatz bildet die Grundlage für die vorliegende Arbeit. Wir benutzen ihn in den folgenden Abschnitten, um die lokale Orientierung und Amplitude zu berechnen und darauf einen Histogramm-Vergleich durchzuführen.

Kapitel 3

Lokale Orientierung

In diesem Kapitel betrachten wir die Bedeutung der lokalen Orientierung für die Repräsentation von Grauwerten. Wir geben zunächst eine Einführung in dieses Gebiet und beschreiben dann einen Ansatz, der auf Quadraturfiltern basiert. Danach gehen wir darauf ein, wie die lokale Orientierung mit den von uns benutzten Filtersätzen bestimmt wird. Dabei erläutern wir die Berechnung der lokalen Amplitude und Orientierung und die Rekonstruktion der ursprünglichen Filterantworten. Der letzte Abschnitt dieses Kapitels behandelt die Signifikanz von einfach orientierten Punkten im Vergleich zu mehrfach orientierten Punkten.

3.1 Einführung

Neben der Steilheit einer Kante ist ihre Orientierung eine wichtige Information. Wir müssen uns also mit der Orientierung in einer kleinen Bildumgebung beschäftigen (*lokale Orientierung*). Liegt in dieser kleinen Umgebung eine Kante oder ein gerichtetes Muster vor, ist die Orientierung ein wichtiges Merkmal. Ist dagegen ein Grauwertplateau oder ein irreguläres Muster zu beschreiben, liefert uns die Orientierung keine wichtige Information.

Wie wir wissen, ist es ohne große Probleme möglich, orientierungsselektive Filter zu konstruieren, die Strukturen einer bestimmten Richtung extrahieren. Für eine Struktur, die eine bestimmte Orientierung aufweist, erhalten wir also Filterantworten, die von dem Winkel zwischen den Orientierungen der Filter und der Struktur abhängen. Die Filterantwort ist maximal, wenn beide Orientierungen übereinstimmen. Dieses Maximum soll mit einer minimalen Anzahl von Filtern interpoliert werden, um so die lokale Orientierung an einem Ort zu bestimmen. Für eine solche Interpolation werden in der Orientierung steuerbare Filter wie z.B. die Filter aus Kapitel 2 oder Quadraturfilter benötigt.

Im nächsten Abschnitt werden wir zunächst darauf eingehen, wie die lokale Orientierung mit Quadraturfiltern bestimmt werden kann. Quadraturfilter detektieren die lokale Orientierung auf einer Skala von 0 bis π, so daß die Orientierung für z.B. Indexing-Probleme benutzt werden kann. Danach beschreiben wir, wie die lokale Orientierung mit Hilfe der Filter aus Kapitel 2 bestimmt werden kann.

3.2 Lokale Orientierung: Quadraturfilter

In der Literatur wird die lokale Orientierung (oder auch die lokale Energie) oft durch die Anwendung von orientierungsselektiven Quadraturfiltern berechnet (siehe [6] oder [1]). Aus diesem Grund werden wir in diesem Abschnitt auf dieses Konzept näher eingehen. Wir stellen zunächst Quadraturfilter allgemein vor und geben ein Beispiel für einen solchen Filter an. Anschließend erläutern wir, wie die lokale Orientierung mit den Quadraturfiltern berechnet wird.

3.2.1 Quadraturfilter

Bei der genaueren Analyse von Bildern interessiert oftmals, welche Ausrichtungen Strukturen aufweisen und wie signifikant diese sind. Es sollen also Kanten einer festen Orientierung detektiert werden. Dabei ist man jedoch nicht an der Symmetrie bzw. Phase einer Kante interessiert, so daß weiß-schwarz Kanten die gleiche lokale Orientierung wie schwarz-weiß Kanten erzeugen.

Wir suchen also nach Filtern, die die lokale Amplitude und die lokale Phase optimal erfassen. Betrachten wir im Ortsraum ein harmonisches Signal, können wir durch eine Quadrierung zwar ein Signal erzeugen, das in der Amplitude quadratisch ist, jedoch mit doppelter Wellenzahl variiert. Um dieses Problem zu lösen, bedient man sich des trigonometrischen Additionstheorems $\cos^2\phi + \sin^2\phi = 1$. Wir müssen das Signal also um $\frac{\pi}{2}$ verschieben, was der Transformation einer Sinus- in eine Kosinusfunktion entspricht. Die Summe der Quadrate ergeben eine phasenunabhängige Funktion, deren Wert dem Amplitudenquadrat des Signals entspricht.

Die Verschiebung eines Signals um $\frac{\pi}{2}$ wird durch die *Hilberttransformation* realisiert. Sie erzeugt also aus dem Realteil $f(x)$ den negativen Imaginärteil $\hat{f}(x)$ eines analytischen Signals $z(x) = f(x) - j\hat{f}(x)$. Die Hilberttransformation ist im Ortsraum durch

$$\hat{f}(x) = \mathcal{H}\{f(x)\} = f(x) * \frac{-1}{\pi x}$$

und im Frequenzraum durch

$$\hat{F}(u) = F(u) j \operatorname{sgn}(u)$$

definiert.

Da in der Literatur die Hilberttransformation nur auf gerade und ungerade Funktionen angewendet wird, beschränken wir uns im folgenden ebenfalls auf diese Funktionen. Die Hilberttransformation invertiert die Symmetrie von Funktionen, d.h. es gilt

$$\mathcal{H}f^e = -f^o \qquad \mathcal{H}f^o = f^e,$$

wobei e für eine gerade (even) und o für eine ungerade (odd) Funktion stehen. Eine Funktion und ihre Hilberttransformierte wird *Quadraturpaar* genannt. Die Hilberttransformation bildet die Grundlage für eine Familie von Filtern, die *Quadraturfilterpaare* heißen. Beide Filter gehen also durch die Hilberttransformation ineinander über. Die Quadraturfilterpaare haben die Eigenschaft, daß beide Filter die gleiche Amplitudenfunktion haben und sich in der Phasenlage unterscheiden. Der gerade Filter ist rein reell, erzeugt also keine Phasenverschiebung, während der ungerade Filter rein imaginär ist und eine Phasenverschiebung um $\pm\frac{\pi}{2}$ erzeugt.

In der Literatur kann man viele Quadraturfilter finden. Die bekanntesten, aber nicht exakten Quadraturfilterpaare sind die *Gaborfilter*. Wir stellen hier jedoch einen nicht so bekannten Quadraturfilter vor: den Anderssonfilter.

Der Anderssonfilter

Der Anderssonfilter, der von M.T. Andersson in [1] vorgestellt wird, ist ein Quadraturfilter, der steuerbar in Orientierung und Skala ist. Der Filter ist polar separabel, d.h. seine Übertragungsfunktion kann in einen radialen und einen angularen Teil zerlegt werden:

$$H_{\sigma,\theta,B}(\rho,\vartheta) = Q_{\sigma,B}(\rho)\cdot A_B(\vartheta), \tag{3.1}$$

wobei ρ die Radial- und ϑ die Angularfrequenz bezeichnen und $Q(\rho)$ die Eigenschaften eines Bandpasses aufweist. Der Anderssonfilter läßt sich zudem in den folgenden Parametern variieren:

- Mittenfrequenz q, die umgekehrt proportional zur Skala σ ist.
- Logarithmischer Radialbandbreite B.
- Orientierung θ.

Da der Filter zur Familie der Quadraturfilter gehört, setzt er sich aus einem geraden und ungeraden Anteil zusammen. Für den angularen Anteil muß also zwischen geradem und hilbertransformierten ungeradem Anteil unterschieden werden. Die Amplitudenfunktionen der geraden und ungeraden Filteranteile sind dagegen identisch, wie in den allgemeinen Bemerkungen zu Quadraturfiltern bereits geschrieben wurde.

Der radiale und angulare Filteranteil sind definiert durch

$$Q_{\sigma,B}(\rho) = \begin{cases} \cos^2\left(\frac{\pi \ln(\sigma\rho)}{2B\ln 2}\right) & , \text{ falls } \frac{1}{2^B\sigma} \leq \rho < \frac{2^B}{\sigma} \\ 0 & , \text{ sonst} \end{cases} \tag{3.2}$$

und

$$A^e_\theta(\vartheta) = \cos^2(\vartheta - \theta) \tag{3.3}$$

$$A^o_\theta(\vartheta) = jA^e_\theta(\vartheta)\,\mathrm{sgn}(\cos(\vartheta - \theta)). \tag{3.4}$$

Der Anderssonfilter wurde in [13] eingesetzt, um Objekte mit Hilfe von lokalchromatischen Orientierungshäufigkeiten zu lokalisieren.

Ein Nachteil der Quadraturfilter ist, daß diese im Fourierraum durch ihre Transferfunktion vorgegeben sind, d.h. bei der Berechnung von Filtermasken praktikabler Größe werden Koeffizienten ungleich null abgeschnitten. Aus diesem Grund muß ein Optimierungsprozeß durchgeführt werden, der die Koeffizienten auf der kleinen Maskengröße so verändert, daß sie der idealen Transferfunktion möglichst nahe kommen. Das Bestreben, einen maximalen Fehler mit dieser Approximation zu unterbieten, impliziert also eine Mindestgröße für die berechneten Filtermasken.

3.2.2 Lokale Orientierung

Wie wir schon in der Einleitung beschrieben haben, möchte man eine minimale Anzahl von Quadraturfiltern dazu benutzen, die lokale Orientierung an jedem Ort zu bestimmen. In [6] wird gezeigt, daß man mindestens drei Richtungsfilter benötigt, die auf einem Intervall der Breite π gleichverteilt sind. Allgemein benötigt man also Richtungsfilter θ_k mit

$$\theta_k = -\frac{\pi k}{K} \quad k = 0, 1, \ldots, K-1. \tag{3.5}$$

Da wir in dieser Arbeit nur mit vier Richtungsfiltern arbeiten, wollen wir auch hier diesen Fall näher betrachten. Wir betrachten orientierungsselektive Filter mit den Orientierungen 0, $-\frac{\pi}{4}$, $-\frac{\pi}{2}$ und $-\frac{3\pi}{4}$.

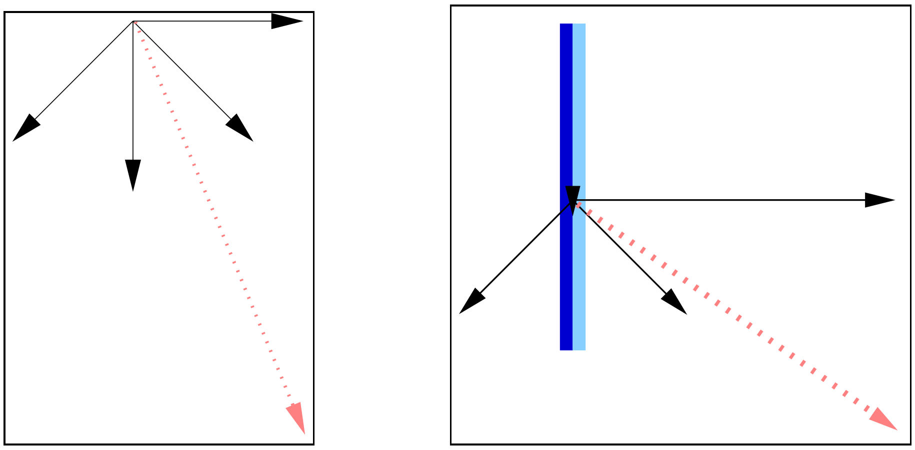


Abbildung 3.1: Ergebnisse der Vektoraddition ohne Verdoppelung der Winkel. Links: Ergebnis bei vorhandener Mehrfachorientierung. Rechts: Ergebnis bei eindeutiger Orientierung.

Wir benutzen für die Berechnung der lokalen Orientierung ein Verfahren von Jähne, das aus Überlegungen der Physik zu Trägheitsmomenten entstanden ist. Die Berechnungen werden dabei im Ortsraum durchgeführt, d.h. die Filterantworten h_k der K Richtungsfilter (in unserem Fall vier) werden als Orientierungsenergien interpretiert und in der komplexen Ebene gewichtet aufsummiert. Diese Berechnung kann man sich als eine gewichtete Vektoraddition veranschaulichen. Es gilt also

$$\tilde{o} = \sum_{k=0}^{K-1} h_k e^{-i\theta_k}. \tag{3.6}$$

Die Vektorsumme, die in Gleichung 3.6 berechnet wird, ergibt allerdings die falsche lokale Orientierung. Dies wollen wir uns anhand von zwei Beispielen klar machen, die in Abbildung 3.1 dargestellt sind. Die dünnen, durchgezogenen Vektoren entsprechen den Filterantworten der vier Richtungsfilter, während der dickere, gepunktete Vektor jeweils das Ergebnis der Vektoraddition symbolisiert. In dem links dargestellten Beispiel ist eine Situation gegeben, in der alle Filter gleich stark antworten, also eine Mehrfachorientierung vorherrscht. Da es nicht möglich ist, eine eindeutige Orientierung zuzuweisen, sollte das korrekte Ergebnis der Nullvektor sein. Auch in dem zweiten Beispiel, in dem eine eindeutige Orientierung vorliegt, kommt man durch die Vektoraddition auf einen falsch orientierten Vektor.

Die Lösung dieses Problems liegt in der Verdopplung der Winkel der Filterorientierungen. Wir modifizieren also Gleichung 3.6 und gehen zu der korrekten Vektorsumme

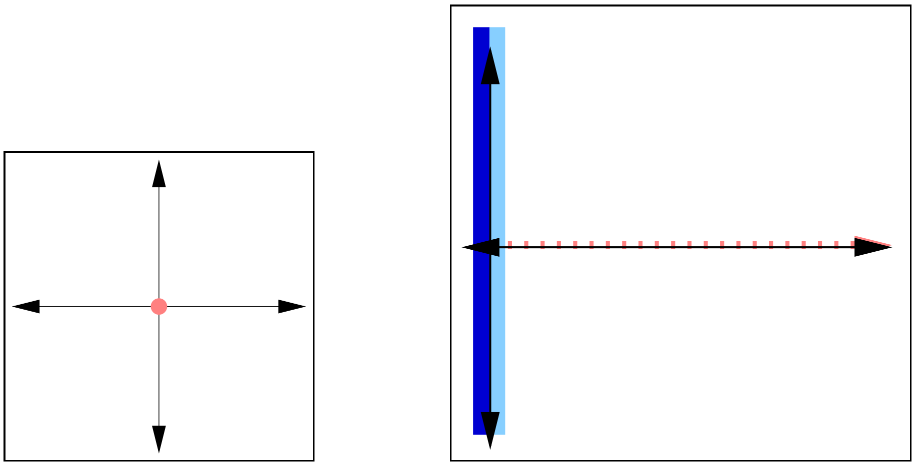


Abbildung 3.2: Ergebnisse der Vektoraddition mit Verdopplung der Winkel. Links: Ergebnis bei vorhandener Mehrfachorientierung. Rechts: Ergebnis bei eindeutiger Orientierung.

$$o = \sum_{k=0}^{K-1} h_k e^{-i2\theta_k} \tag{3.7}$$

über und erhalten somit ein komplexes Ergebnis, aus dem Amplitude $|o|$ und Orientierung $arg(o)$ berechnet werden können. Da wir jedoch die Winkel der Filterorientierungen verdoppeln, ist es notwendig, den Winkel der errechneten Orientierung zu halbieren. Die lokale Orientierung umfaßt also, ebenso wie die Orientierungen der Quadraturfilterantworten, einen Bereich von 180 Grad. In Abbildung 3.2 sind die Ergebnisse der Vektoraddition bei Verdopplung der Winkel dargestellt. Man sieht, daß durch die Verdopplung die korrekten Orientierungen berechnet werden.

3.3 Lokale Orientierung: Simoncellifilter

Die Berechnung der lokalen Orientierung, wie wir sie im vorherigen Abschnitt vorgestellt haben, ist für die Zwecke der Bildcodierung unbrauchbar. Sollen Bilder aus lokalen Amplituden und Orientierungen rekonstruiert werden, hat es fatale Folgen, wenn schwarz-weiß Kanten die gleiche lokale Orientierung ergeben wie weiß-schwarz Kanten. Deshalb ist es nicht möglich, Quadraturfilter einzusetzen ohne die Phase der Grauwertverläufe mitzucodieren. In Abbildung 3.3 ist die lokale Orientierung ϑ dargestellt, wie sie aus Quadraturfilterantworten berechnet werden kann. Man sieht, daß nur lokale Orientierungen von 0° bis 180° berechnet werden, z.B. die lokale Phase $\frac{\pi}{2}$ also die gleiche lokale Orientierung erzeugt wie die lokale Phase $\frac{3\pi}{2}$. Aus diesem Grund verwenden wir die in

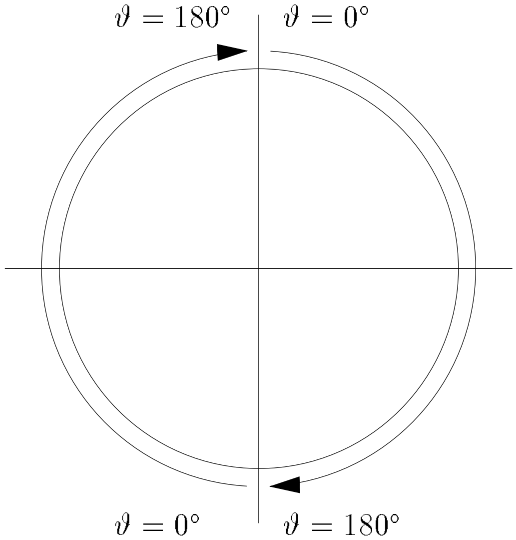


Abbildung 3.3: Zusammenhang: Lokale Phase und lokale Orientierung bei Quadraturfiltern

Kapitel 2 beschriebenen Filter von E. Simoncelli, die wir im folgenden *Simoncellifilter* nennen werden.

Im folgenden Abschnitt beschreiben wir, wie die lokale Orientierung mit Hilfe der Simoncellifilter berechnet wird. Da wir die lokalen Orientierungsbilder für die Bildkompression nutzen wollen, gehen wir dann darauf ein, wie die Filterantworten aus der lokalen Amplitude und Orientierung rekonstruiert werden können.

3.3.1 Lokale Orientierung

Die von uns benutzten Simoncellifilter sind vier ungerade Richtungsfilter, die in den gleichen Richtungen antworten, wie die oben beschriebenen Quadraturfilter. Da diese auch negative Vektoren als Filterantwort erzeugen können, ergeben sich bei der Berechnung der Vektorsumme Orientierungen von 0° bis 360°. Wir führen bei der Vektoraddition also keine Verdopplung der Winkel der Filterorientierungen durch, so daß sich für die Berechnung der lokalen Orientierung wiederum die Gleichung 3.6 ergibt.

Aus dem komplexen Ergebnis läßt sich die korrekte Orientierung $arg(\tilde{o})$ bestimmen. Daß die berechnete Orientierung tatsächlich richtig ist, veranschaulichen wir in Abbildung 3.4. Es ist zu sehen, daß der Orientierungsvektor in die richtige Richtung zeigt, die Orientierung also korrekt berechnet wird. Die Amplitude wird durch dieses Verfahren jedoch verstärkt. Es wäre wünschenswert, daß die lokale Amplitude den gleichen Wert aufweist wie ein senkrecht zur Kante ausgerichteter Richtungsfilter. In Abbildung 3.4 würde dies also der Länge des nach rechts zeigenden Vektors entsprechen. Würde die, durch die Vektoraddition berechnete Amplitude verwendet werden, ergäbe sich eine Verstärkung

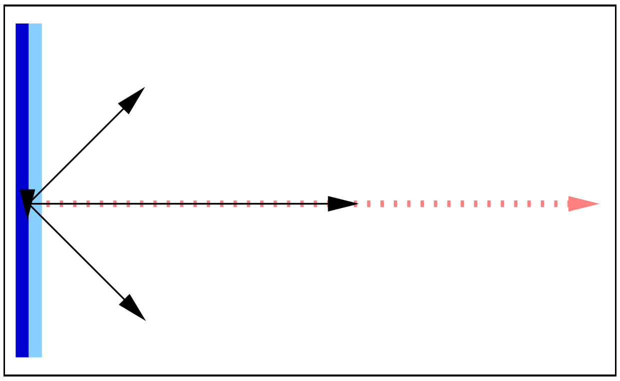


Abbildung 3.4: Vektoraddition der Antworten der Simoncellifilter. Der gepunktete, helle Vektor stellt das Ergebnis der Vektoraddition dar.

sämtlicher Amplitudenwerte, was in dem rekonstruierten Bild zu einer Kontrasterhöhung führen würde.

Statt des so berechneten Amplitudenwerts verwenden wir eine andere Methode, um den gewünschten Wert zu bestimmen. Dabei greifen wir auf die Winkelverdopplung zurück, die bei den Quadraturfiltern verwendet wurde. Um diese korrekt einsetzen zu können, müssen wir den Orientierungsbereich der Filterantworten halbieren, indem wir den Betrag der Filterantworten bilden. Es ist also die folgende Vektoraddition zu berechnen:

$$a = \sum_{k=0}^{K-1} |h_k| e^{-i2\theta_k}. \tag{3.8}$$

Aus diesem komplexen Ergebnis kann dann die lokale Amplitude $|a|$ berechnet werden. Betrachten wir diese Berechnung an Hand der Filterantworten in Abbildung 3.4, so ist festzustellen, daß sich die beiden schwächeren Filterantworten aufheben und nur die maximale Filterantwort bestehen bleibt.

3.3.2 Rekonstruktion der Filterantworten

Da wir die lokale Amplitude und Orientierung in einem Kompressionsverfahren benutzen wollen, ist es nötig, daß wir aus einem Amplituden- und Orientierungsbild die ursprünglichen Filterantworten der Simoncellifilter rekonstruieren. Diese Rekonstruktion kann nicht verlustfrei sein, da wir durch die Vektoradditionen jeder Kombination von vier Filterkoeffizienten eine lokale Amplitude und eine lokale Orientierung zuweisen. Auf diesen Informationsverlust gehen wir näher im nächsten Abschnitt ein.

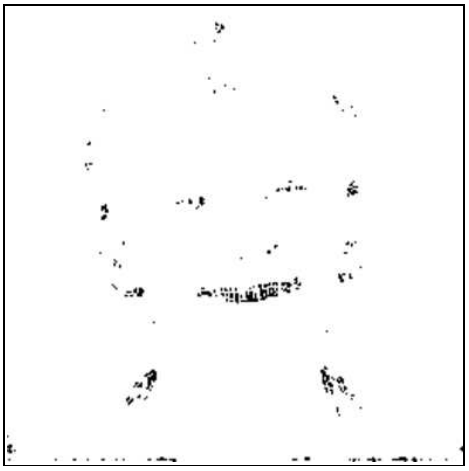


Abbildung 3.5: Vergleich der Häufigkeit von Einfach- (links) gegenüber Mehrfachorientierungen (rechts).

Die Rekonstruktion der Filterantworten geschieht durch eine einfache Projektion der lokalen Amplitude und Orientierung auf die Richtungen der ursprünglichen Simoncellifilter. Die rekonstruierten Filterantworten berechnen sich also wie folgt:

$$h_k = A \cdot (\cos\theta_k \cdot \cos\vartheta + \sin\theta_k \cdot \sin\vartheta), \tag{3.9}$$

wobei A die lokale Amplitude, ϑ die lokale Orientierung bezeichnet und k von 0 bis $K-1$ läuft. Man erhält also für jeden Filter einen Rekonstruktionswert, der bis auf die oben angesprochenen Ausnahmen mit den ursprünglichen Filterantworten übereinstimmt.

3.4 Einfach- und Mehrfachorientierungen

Wie in den vorherigen Abschnitten erläutert wurde, wird einem Ort durch die Vektoraddition von K (in unserem Fall vier) Richtungsfiltern eine eindeutige Orientierung zugewiesen. Antworten alle Filter gleich stark, da an dieser Stelle keine eindeutige Orientierung vorliegt, wie dies bei einem Punkt der Fall ist, wird eine Amplitude von null errechnet. Die Filterkoeffizienten erhalten durch die Rekonstruktion ebenfalls den Wert null, was einen Informationsverlust an dieser Stelle bedeutet. Ein Informationsverlust wird also immer dann verursacht, wenn mehr als eine Orientierung an einem Ort vorliegt. Es ist uns also nicht möglich, Strukturen wie z.B. Ecken oder Punkte zu erfassen.

Effektiv verlieren wir durch die von uns durchgeführte Berechnung der lokalen Amplitude und Orientierung jedoch nur wenig Information. In natürlichen Bildern sind Stellen mit einfachen Orientierungen deutlich häufiger zu finden als Stellen mit mehreren Orientierungen. Um diese Tatsache zu veranschaulichen, haben wir in Abbildung 3.5 aus einem Beispielbild alle Punkte mit einfachen und mehrfachen Orientierungen extrahiert. Die linke Abbildung zeigt sämtliche Einfachorientierungen, die im Bild vorkommen. Man kann das dargestellte Bild relativ gut erkennen, was die Häufigkeit der Einfachorientierungen veranschaulicht.

Auf der rechten Seite sind dagegen alle Stellen dargestellt, die mehr als eine Orientierung aufweisen. Man sieht, daß die Anzahl der Stellen mit Mehrfachorientierungen sehr gering ist. Durch die von uns durchgeführten Vektoradditionen werden also an sehr wenigen Stellen Fehler verursacht, die sich nicht auf den Gesamteindruck des rekonstruierten Bildes auswirken.

Diese Aussage gilt jedoch nur für natürliche Bilder. Es ist ohne weiteres möglich , künstliche Bilder zu erzeugen, die an sehr vielen Stellen mehr als eine Orientierung aufweisen und somit durch die Rekonstruktion sehr fehlerhaft dargestellt werden.

Die oben beschriebenen Nachteile für die Rekonstruktion, die durch die fehlerhafte Behandlung der mehrfach orientierten Punkte entsteht, haben jedoch keine Auswirkungen auf die Lokalisierung. Wie wir in Kapitel 5 erläutern werden, verwenden wir für die Lokalisierung Histogramme von Vektoren, deren Komponenten aus den berechneten lokalen Orientierungen mehrerer Skalen bestehen. Durch die Struktur der Orientierungshistogramme werden mehrfach orientierte Punkte ausreichend repräsentiert, auch wenn diese bei der Berechnung auf einer Skala vernachlässigt werden.

Kapitel 4

Codierung auf Basis einer steuerbaren Pyramide

Kompressionsverfahren reduzieren die für die Repräsentation eines Bildes benötigte Datenmenge. Sie können in *verlustfreie* und *verlustbehaftete* Verfahren eingeteilt werden. Verlustfreie Verfahren entfernen überflüssige Information (Redundanz) aus einem Bild, erzielen allerdings keine hohen Kompressionsraten. Verlustbehaftete Verfahren reduzieren nicht wahrnehmbare (Irrelevanz) oder sogar für die Wahrnehmung wichtige (Relevanz) Information. Soll eine hohe Kompressionsrate erzielt werden, so ist es zwingend erforderlich, daß eine verlustbehaftete Kompression wie z.B. die Quantisierung verwendet wird. Entfernt man allerdings durch diese Kompression relevante Information aus einem Bild, entstehen sichtbare Rekonstruktionsfehler.

In diesem Kapitel beschreiben wir ein Kompressionsverfahren, das auf einer steuerbaren Pyramide (siehe Kapitel 2) und einer Repräsentation der Koeffizienten mit Hilfe der lokalen Orientierung (siehe Kapitel 3) basiert. Zunächst werden wir etwas näher auf Quantisierung im allgemeinen eingehen und dann beschreiben, wie wir die Quantisierung in unserem Algorithmus durchführen. Danach erläutern wir den Kompressionsalgorithmus und untersuchen die Güte des beschriebenen Verfahrens.

4.1 Quantisierung

Der Schritt der Quantisierung folgt bei der Digitalisierung eines Bildes der Abtastung (sampling). Bei der Abtastung wird konform mit dem Abtasttheorem ein kontinuierliches Signal auf ein diskretes Gitter projiziert. Die Quantisierung bildet dann den kontinuierlichen Wertebereich der Abtastwerte auf eine endliche Menge ab; es geht also effektiv Information verloren. Wie diese Abbildung erfolgt, hängt häufig von dem gewünschten Ergebnis ab. Stellt man keine großen Ansprüche an die Leistung des Quantisierers, wird

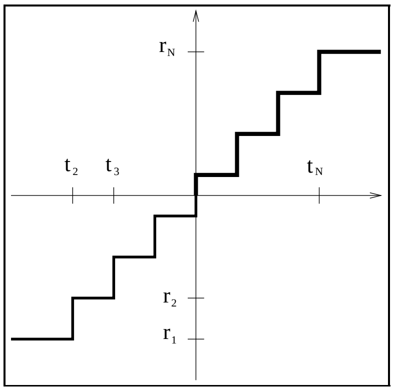


Abbildung 4.1: Ein gleichförmiger achtstufiger Quantisierer.

man eine einfache *skalare Quantisierung* anwenden. Betrachtet man z.B. die Leistung eines Quantisierers bzgl. Kompression und Rekonstruktion, so wird man vor allem auf minimale Rekonstruktionsfehler und eine geringe Coderate Wert legen. Sind die Anforderungen an Quantisierungsfehler und Anzahl der benutzten Werte dagegen höher, ist es nötig auf ausgefeiltere Algorithmen zurückzugreifen (z.B. *Vektorquantisierung*). Die Vektorquantisierung geht dabei auf statistische Abhängigkeiten zwischen einzelnen Pixelwerten ein, während bei der skalaren Quantisierung jeder Wert für sich betrachtet wird, d.h. nur auf die Statistik nullter Ordnung eingegangen wird.

Durch die Quantisierung entsteht gewöhnlich immer ein Verlust an Information, da eine größere Werteanzahl nicht aus einer geringeren nach der Quantisierung rekonstruiert werden kann. Wird dagegen ein bereits quantisiertes Signal noch einmal quantisiert, kann es sein, daß sich die Anzahl der ursprünglichen Werte nicht verändert.

Wir gehen zunächst auf die einfache skalare Quantisierung ein und werden anschließend kurz das sehr viel größere und kompliziertere Gebiet der Vektorquantisierung anschneiden.

4.1.1 Skalare Quantisierung

Bei einer skalaren Quantisierung wird der Wertebereich in N Intervalle zerlegt. Ist u der Wert an einem Abtastpunkt, so liegt u in genau einem Intervall $[t_k, t_{k+1})$, wobei $k \in \{1, \dots, N-1\}$. Der Wert u wird somit durch den Repräsentativwert r_k des Intervalls ersetzt. Ein skalarer Quantisierer hat also eine typische Treppenfunktion (siehe Abbildung 4.1). Die oben genannte Aufteilung läßt jedoch die Fragen offen, wie die Intervalle einzuteilen und die Repräsentativwerte anzuordnen sind. Die einfachste Möglichkeit ist, allen Intervallen die gleiche Größe zuzuweisen und die Repräsentativwerte in die Mitte jedes Intervalls zu legen. Man kommt somit zu der Klasse der *gleichförmigen Quantisierer*. Häufig sind das erste Intervall nach unten und das letzte Intervall nach oben offen. In verschiedenen Anwendungen werden die Intervallbreiten auch auf eine andere Weise gewählt. Es ist z.B. manchmal sinnvoll, für die Breite eine logarithmische Skala zu wählen.

Soll ein Quantisierer bzgl. Rekonstruktion im L_2-Sinne optimiert sein, kann der *Lloyd-Max Quantisierer* benutzt werden, der den durchschnittlichen Quadratfehler minimiert. Dabei wird angenommen, daß u eine reelle skalare Zufallsvariable mit kontinuierlicher Dichtefunktion $p_u(x)$ ist. Man sucht also die Koeffizienten t_k und r_k ($k = 1, \ldots, N$) eines N-stufigen Quantisierers, der den durchschnittlichen Quadratfehler

$$\mathcal{E} = E[(u - q(u))^2] = \int_{t_1}^{t_{N+1}} (x - q(x))^2 p_u(x) dx$$

minimiert. Dabei ist $q : \mathbb{R} \to (t_1, t_{N+1})$ eine Funktion, die einer reellen Zahl den entsprechenden Repräsentativwert zuordnet. Nimmt man nun an, daß $t_{k-1} \leq t_k$ ist und die Grenzen t_1 und t_{N+1} bekannt sind, ergibt sich das folgende nichtlineare Gleichungssystem:

$$\begin{aligned} t_k &= \frac{(r_k - r_{k-1})}{2} \\ r_k &= \frac{\int_{t_k}^{t_{k+1}} x p_u(x) dx}{\int_{t_k}^{t_{k+1}} p_u(x) dx}. \end{aligned}$$

Die Lösung des Gleichungssystems kann z.B. iterativ mit dem Newtonverfahren bestimmt werden.

4.1.2 Vektorquantisierung

Vektorquantisierung ist eine Verallgemeinerung der skalaren Quantisierung auf einen Vektor. Während die skalare Quantisierung in erster Linie bei der Digitalisierung eines Signals benutzt wird, findet die Vektorquantisierung oft bei der Kompression bereits digitalisierter Signale eine Anwendung. Durch die Verallgemeinerung auf einen Vektor kann annähernd jede beliebige Struktur dargestellt werden: z.B. ein Segment eines Bildes, die spektrale Hüllenkurve eines Sprachklangs oder die Pyramidenrepräsentation eines Bildes. Man kann Vektorquantisierung als eine Form von Pattern Matching sehen, bei dem ein Eingangsmuster durch ein Element einer Menge von vordefinierten Vorgabemustern approximiert wird.

Definition 6 *Ein Vektorquantisierer Q der Dimension k und Größe N ist eine Abbildung eines Vektors aus dem k-dimensionalen Euklidischen Raum $\mathbb{R}^k$ in eine endliche Menge $\mathcal{C}$ von N Ausgabevektoren, die* Codevektoren *oder* Codeworte *genannt werden.*

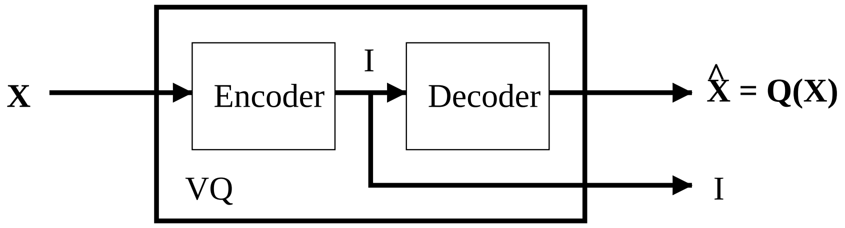


Abbildung 4.2: Ein Vektorquantisierer als Kaskade von Encoder und Decoder.

Ein Vektorquantisierer kann in zwei Komponenten aufgespalten werden: einen *Encoder*, der für einen Eingabevektor X den Index I des Codevektors ausgibt und einen *Decoder*, der über den Index den entsprechenden Ausgabevektor rekonstruiert (siehe Abbildung 4.2).

Die einfachste Möglichkeit ein Codebuch, d.h. Abbildungsvorschrift und Menge der Codevektoren zu konstruieren, besteht darin, einen separablen Vektorquantisierer zu verwenden. Dabei wird jede Komponente des Vektors mit einem skalaren Quantisierer quantisiert. Diese Strategie führt immer zu rechtwinkligen Quantisierungszellen und ist im allgemeinen nicht sehr leistungsfähig.

Für eine optimiertere Konstruktion bzw. Verfeinerung eines Codebuchs gibt es eine große Anzahl von Algorithmen. Zunächst muß man jedoch entscheiden, welche Philosophie mit dem Vektorquantisierer verfolgt werden soll. Bei einer großen Gruppe von Vektorquantisierern wird das Codebuch adaptiv optimiert, so daß der Quantisierungsfehler minimal wird. Um dies zu erreichen werden im allgemeinen iterative Trainingsverfahren benutzt, die eine schrittweise Verbesserung durchführen (z.B. Lloyd- bzw. LBG-Alorithmus). Diese Algorithmen verkomplizieren die Konstruktion eines Vektorquantisierers und gehen nur auf die Statistik der vorgegebenen Vektoren ein. Möchte man jedoch Wissen über die statistische Verteilung der Vektoren oder über das perzeptionelle Auflösungsvermögen des Betrachters in die Konstruktion einfließen lassen, sind die adaptiven Algorithmen ungeeignet. Man nutzt statt dessen heuristische Verfahren zur Konstruktion. Diese fixen Vektorquantisierer bedienen sich einer fest vorgegebenen Strategie, wobei keine Anpassung an die gegebenen Vektoren durchgeführt wird.

Eine ausführliche Behandlung dieses Themas findet man in [5].

4.2 Eigenschaften der Kompression

In diesem Abschnitt führen wir Maße für Coderate und Signalqualität eines Bildes ein. Anschließend erläutern wir den Aufbau des von uns verwendeten Kompressionsverfahrens und gehen mit Hilfe einiger Experimente auf dessen Leistungsfähigkeit ein.

4.2.1 Entropie und SNR

Um unser Kompressionsverfahren bewerten zu können, benötigen wir Maße sowohl für die Coderate als auch für die Signalqualität des codierten Bildes. Diese Maße sind in der gesamten Bildverarbeitung gebräuchlich, so daß es möglich ist, unser Verfahren mit anderen zu vergleichen.

Entropie

Ein sehr wichtiges Maß für den Informationsgehalt einer Sequenz ist die *Entropie*. Sie stellt den eigentlichen Informationsgehalt dar, der unabhängig von der Wahl der Repräsentation ist. Nach einem Satz von Shannon ist die Entropie n-ter Ordnung die theoretische untere Schranke für verlustfreie Kompressionsalgorithmen, die sich der Statistik n-ter Ordnung bedienen (z.B. Entropiecodierer). Bei der Entropie n-ter Ordnung werden die Auftrittswahrscheinlichkeiten von Folgen aus $n+1$ Symbolen betrachtet. In der Bildverarbeitung wird üblicherweise die Entropie nullter Ordnung angewandt, d.h. jedes Symbol einer Sequenz wird für sich betrachtet. Die Entropie nullter Ordnung ist definiert durch

$$H_0 = -\sum_{k=0}^{K-1} p_k \log p_k,$$

wobei $p_k \in [0,1]$ die Auftrittswahrscheinlichkeiten der einzelnen Symbole sind. Wie für Wahrscheinlichkeiten üblich, gilt $\sum_{k=0}^{K-1} p_k = 1$. Enthält die Sequenz nur ein Symbol b_k, so wird die Entropie minimal und es gilt

$$H_0 = -1 \log 1 = 0,$$

da dann $p_k = 1$ ist. Sind die Auftrittswahrscheinlichkeiten dagegen gleichverteilt, wird die Entropie maximal. Es gilt

$$H_0 = -\sum_{k=0}^{K-1} \frac{1}{K} \log \frac{1}{K} = \log K,$$

wobei $p_k = \frac{1}{K}$ für alle k. Nimmt man den Logarithmus zur Basis zwei, wird die Entropie in *Bit pro Symbol* angegeben, d.h. bei Bildern in Bit/Pixel.

Maße für die Signalqualität

Alle Kompressionsalgorithmen, die mit Hilfe von Quantisierung oder Transformation hohe Kompressionsraten erreichen, sind verlustbehaftet. Dieser Verlust bedingt eine Verschlechterung der Signalqualität, was sich mit einem adäquaten Maß quantifizieren läßt.

Um die Maße für die Signalqualität herzuleiten, werden wir zunächst zwei Maße einführen, die auf Signalen gebräuchlich sind: den Mittelwert und die Varianz. Der Mittelwert μ_x eines Signals berechnet sich wie folgt:

$$\mu_x = \frac{1}{N} \sum_{i=0}^{N-1} x(i).$$

Die Varianz σ_x^2 eines Signals ist gegeben durch

$$\sigma_x^2 = \frac{1}{N} \sum_{i=0}^{N-1} (x(i) - \mu)^2.$$

Für den Vergleich zwischen zwei digitalen Repräsentationen x_i (Original) und $\hat{x}_i$ (Rekonstruktion) eines Signals mit endlicher Ausdehnung N kann der *Mean Square Error* (MSE) benutzt werden:

$$\mathrm{MSE}_{x,\hat{x}} = \frac{1}{N} \sum_{i=0}^{N-1} (x(i) - \hat{x}(i))^2.$$

Der MSE wird größer, je mehr die beiden Signale voneinander abweichen. Trotzdem ist die Aussagekraft des MSE begrenzt, da er von der Skalierung der Abtastwerte abhängt. Aus diesem Grund normalisiert man den MSE, indem die Varianz des Originalsignals in die Berechnung einbezogen wird. Man erhält den *varianznormalisierten Signal to Noise Ratio* (vvSNR), der wie folgt definiert ist:

$$\mathrm{vvSNR}_{x,\hat{x}} = 10 \log_{10} \frac{\sigma_x^2}{\mathrm{MSE}_{x,\hat{x}}}.$$

Der vvSNR wird in Dezibel (dB) angegeben und ist ein logarithmisches Maß für das Verhältnis zwischen dem Originalsignal und der Abweichung in der Rekonstruktion. Je höher der vvSNR ist, desto besser approximiert die Rekonstruktion das Original.

Ein alternatives Maß der Signalqualität, für das keine Berechnung der Varianz erforderlich ist, stellt der *peak-to-peak Signal to Noise Ratio* (ppSNR) dar:

$$\text{ppSNR}_{x,\hat{x}} = 10 \log_{10} \frac{(K_{max} - K_{min})^2}{\text{MSE}_{x,\hat{x}}},$$

wobei K_{min} und K_{max} die in diesem Signal minimal und maximal möglichen Abtastwerte angeben. Bei 8 Bit pro Abtastwert ergibt sich also für $(K_{max} - K_{min})^2 = ((2^8 - 1) - 0)^2 = 65025$. Da der ppSNR in der Literatur gebräuchlicher ist, werden wir diesen im weiteren benutzen und einfach mit "SNR" abkürzen.

Der SNR ist ein objektives und mathematisch handhabbares Maß für die Qualität eines Signals. Da der Mensch jedoch z.B. bei der Betrachtung eines Bildes subjektiv vorgeht, ist es wichtig, daß Unterschiede zwischen Original und Rekonstruktion zusätzlich von Menschen beurteilt werden, also die subjektive Bildqualität bestimmt wird. Normalerweise wird ein Bild mit niedrigerem SNR-Wert auch vom Menschen schlechter eingeschätzt. Es gibt jedoch Fälle, bei denen sich objektive und subjektive Bewertung deutlich voneinander unterscheiden. Als Beispiel für einen solchen Fall betrachten wir ein Portraitbild. Im ersten Fall wird jeder zweite Bildpunkt und im zweiten Fall die gesamte untere Bildhälfte auf den Mittelwert der Grauwerte gesetzt. Während der SNR in beiden Bildern annähernd gleich ist, wird der Mensch das erste Bild deutlich besser bewerten, da es zwar Streifen aufweist, aber immer noch erkennbar ist. Beim zweiten Bild fehlt dagegen eine Hälfte der Bildinformation völlig.

Im Bereich hoher Werte korelliert der SNR im allgemeinen mit der subjektiven Bildqualität, während bei niedrigen SNR-Werten eine subjektive Beurteilung zu anderen Ergebnissen kommen kann. Der SNR ist also eine nützliche Bewertung, die aber teilweise mit Vorsicht behandelt werden muß.

4.2.2 Implementierung

Das von uns entwickelte Kompressionsverfahren basiert auf der Implementierung einer steuerbaren Pyramide[1], die von E. Simoncelli stammt (siehe [20]). Der Programmcode wurde so adaptiert, daß dieser in der Bildverarbeitungssoftware *Khoros 2.0* genutzt werden konnte.

In Abbildung 4.3 wird der gesamte Kompressionsalgorithmus veranschaulicht. Zunächst wird das Originalbild in eine steuerbare Pyramide transformiert (siehe Kapitel 2). Normalerweise haben wir hierfür vier Pyramidenstufen benutzt. Es ist jedoch auch möglich drei oder fünf Stufen zu verwenden. Die vier Bandpaßfilter sind rotierte Kopien voneinander

[1] Source code und Filterkern der Implementierung können über anonymous ftp unter der Adresse ftp.cis.upenn.edu:pub/eero/steerpyr.tar.Z heruntergeladen werden.

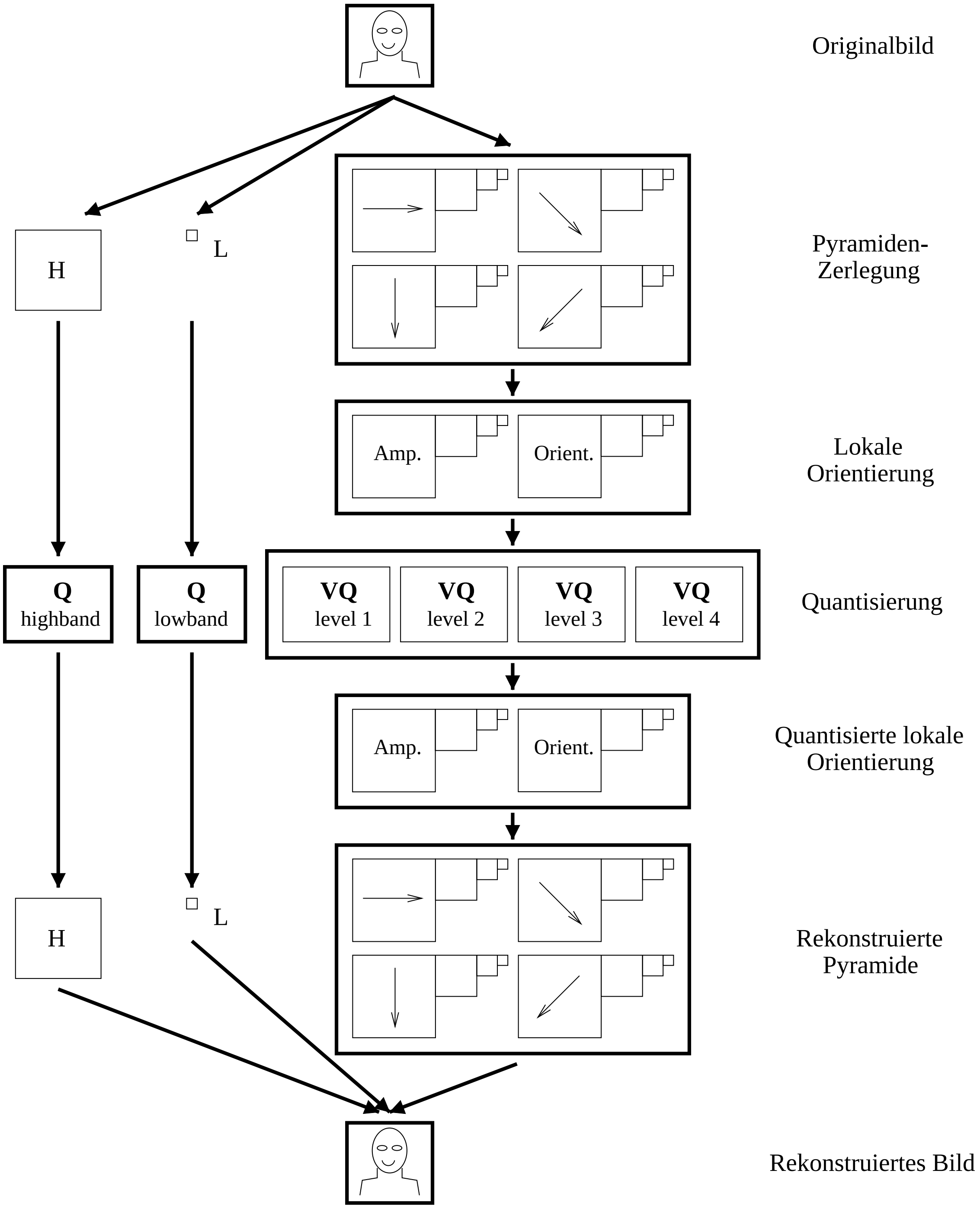


Abbildung 4.3: Kompressionsalgorithmus auf der Basis einer steuerbaren Pyramide. Das Originalbild wird zunächst in eine steuerbare Pyramide transformiert. Dabei bezeichnet H den Hochpaßanteil und L den Tiefpaßanteil der Pyramidenzerlegung. Aus den orientierten Richtungsbändern wird die lokale Orientierung berechnet. Anschließend folgt eine Quantisierungstufe, wobei **Q** skalare und **VQ** Vektorquantisierer bezeichnen. Zur Rekonstruktion des Bildes werden dann die inversen Transformationen angewendet.

und reagieren auf Strukturen mit einer Orientierung von 0, $-\frac{\pi}{4}$, $-\frac{\pi}{2}$ und $-\frac{3\pi}{4}$. Im nächsten Schritt werden die vier Orientierungspyramiden verwendet, um die lokale Orientierung auf den verschiedenen Pyramidenstufen zu errechnen (siehe Kapitel 3). Man erhält also je eine Pyramide für die lokale Orientierung und die lokale Amplitude. Danach folgt eine Quantisierungsstufe. Die Quantisierungsstufe wird einzeln für jedes zu quantisierende Frequenzband angepaßt. Dadurch ist es möglich, Frequenzen, die das Menschliche Visuelle System (MVS) nicht so stark wahrnimmt, gröber zu quantisieren. Um den Hochpaß- und Tiefpaßanteil zu quantisieren, benutzen wir einen einfachen gleichförmigen skalaren Quantisierer, der um den Nullpunkt eine Totzone hat, d.h. einen Bereich aufweist, in dem alle Werte auf null quantisiert werden. Der Hochpaßanteil, der genauso groß ist wie das Originalbild, wird relativ grob quantisiert (ca. 80%-90% der Werte werden zu null quantisiert), da es kaum Informationen aufweist, die für die Erkennung des Bildes bedeutend sind. Der Tiefpaßanteil, der das Restbild nach der rekursiven Bandpaßfilterung darstellt, wird dagegen sehr fein quantisiert.

Bei den übrigen Frequenzbändern wendet man für jede Pyramidenstufe eine Vektorquantisierung von lokaler Amplitude und Orientierung an (siehe Abbildung 4.4 für ein Beispiel). Dabei werden die Amplitudenbilder analog zu Hoch- und Tiefpaßanteil skalar mit einer Totzone quantisiert. Die Orientierungsbilder werden abhängig von der Amplitudenquantisierung bearbeitet. Ist die quantisierte Amplitude null, so wird die Orientierung auf einen ausgezeichneten Wert gesetzt, der anzeigt, daß an dieser Stelle keine eindeutige Orientierung vorliegt. Die Totzone ist aus zwei Gründen wichtig. Zum einen werden geringe Amplitudenwerte vom MVS sehr schlecht wahrgenommen, so daß die Quantisierung zu null Code spart, was sich nur wenig auf die subjektive Bildqualität auswirkt. Zum anderen hängt die Orientierungsquantisierung von der lokalen Amplitude ab. Eine zugewiesene Nicht-Orientierung spart gleichfalls Code, da z.B. bei einem Grauwertplateau durch eine Lauflängencodierung viele Werte zusammengefaßt werden können.

Die Orientierung wird bei niedrigen Amplituden mit wenigen Orientierungswerten und bei steigenden Amplitudenwerten mit immer mehr Orientierungen quantisiert. Dies entspricht dem Phänomen, daß Orientierungsfehler mit steigendem Kontrast stärker wahrgenommen werden. Wie groß die Totzone und die Anzahl der Orientierungswerte direkt nach der Totzone ist und wie schnell die Werteanzahl steigt, kann, genau wie die Anzahl der Amplitudenwerte, je nach gewünschter Bildqualität direkt eingestellt werden. Damit die Menge der Eingabeparameter nicht unüberschaubar wird, legen wir nur fest, wieviele mögliche Amplitudenwerte es auf der obersten Pyramidenstufe geben soll und in wieviele Orientierungsintervalle die äußerste Amplitudenstufe zerlegt wird. Die Steigerung der Orientierungsanzahl sowie die Verringerung der Amplitudenstufenanzahl für höhere Frequenzen (d.h. absteigende Pyramidenstufen) hängen von diesen Werten ab. Es ist also auf einfache Weise möglich, sich für ein optimales Verhältnis von hoher Kompressionsrate, aber schlechterer Qualität und niedrigerer Kompressionsrate, aber guter Qualität zu entscheiden.

Für eine Übertragung oder Speicherung eines Bildes genügt es, die Indizes von jedem

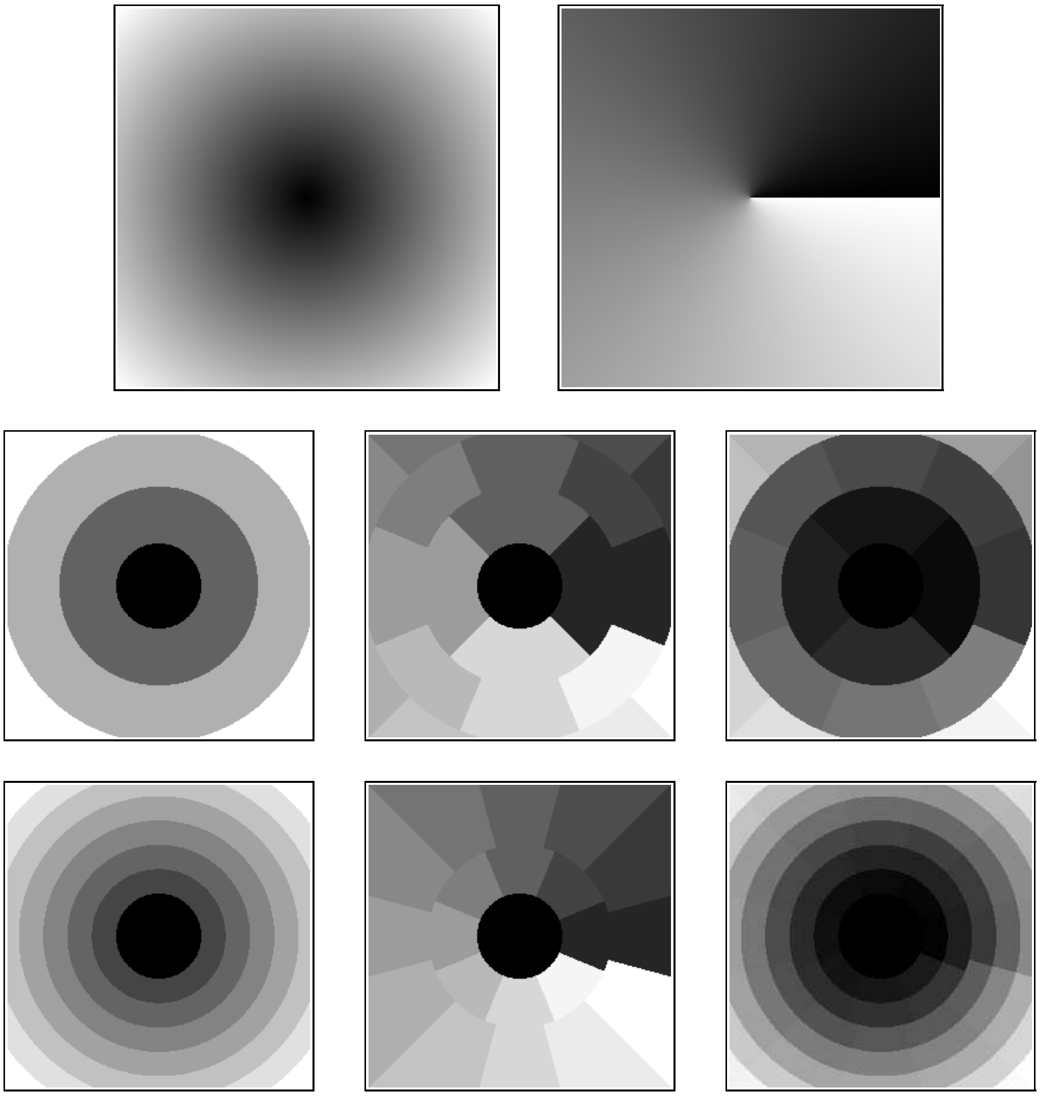


Abbildung 4.4: Beispiel für Vektorquantisierungen: Obere Zeile: Testbilder mit Amplitude von 0–1 (links) und Orientierung von 0–2π (rechts). Mitte: Quantisierte Amplitude, quantisierte Orientierung und Statistik bei vier Amplitudenwerten und 4–20 Orientierungen (hier nur zwölf erreicht). Untere Zeile: Quantisierte Amplitude, quantisierte Orientierung und Statistik bei acht Amplitudenwerten und 8–12 Orientierungswerten (alle zwei Stufen gesteigert).

Abbildung 4.5: Beispielkompression: Links das Originalbild mit 8 Bit/Pixel, in der Mitte eine Rekonstruktion mit 0.58 Bit/Pixel, rechts eine Rekonstruktion mit 2.05 Bit/Pixel.

Vektor über ein Indexbild zu übertragen und zusammen mit dem Codebuch für die Rekonstruktion zu benutzen. Wir verzichten an dieser Stelle jedoch auf die konkrete Durchführung der weiteren Kompression und berechnen auf der Basis der Indexbilder die Entropie, um die minimal zu erwartende Coderate eines Entropiecodierers nullter Ordnung anzunähern.

Aus den quantisierten Frequenzbändern werden anschließend die Filterantwortpyramiden der vier ursprünglichen Orientierungen rekonstruiert. Im letzten Schritt wird das Bild aus der so entstandenen steuerbaren Pyramide rekonstruiert. Da die steuerbare Pyramide, wie in Kapitel 2 gezeigt wurde, involutorisch ist, erfordert dieser Schritt einfach eine erneute Anwendung der Pyramidenfilter.

4.2.3 Testreihen

Für unsere Testreihe zu dem oben beschriebenen Kompressionsverfahren haben wir das Standardbild "Lena" in einer Größe von 256×256 Pixeln mit 256 Graustufen benutzt. In Abbildung 4.5 ist ein Vergleich von Original und rekonstruierten Bildern dargestellt. Das mittlere Bild wurde mit 0.58 Bit/Pixel codiert, während das rechte Bild mit 2.05 Bit/Pixel codiert wurde. Wie auch der subjektive Eindruck bestätigt, ist die Bildqualität im mittleren Bild (SNR 25.00 dB) niedriger als im rechten Bild (SNR 29.01 dB). In Abbildung 4.6 ist der Zusammenhang von SNR zu Coderate aufgeführt. Man sieht, daß die Kurve einen wurzelförmigen Verlauf hat, d.h. die Qualität nimmt schon bei niedrigen Coderaten einen akzeptablen Wert an.

Um zu überprüfen, wieviel Information die lokale Amplitude bzw. die lokale Orientierung trägt, haben wir in zwei weiteren Versuchen vor der Rekonstruktion zum einen die lokale Orientierung durch weißes und zum anderen die lokale Amplitude durch exponentielles Rauschen ersetzt. Die Versuche wurden ohne Quantisierungsstufe durchgeführt, um bei diesen zwei Versuchen einen Nebeneffekt der Quantisierung auszuschließen. Während

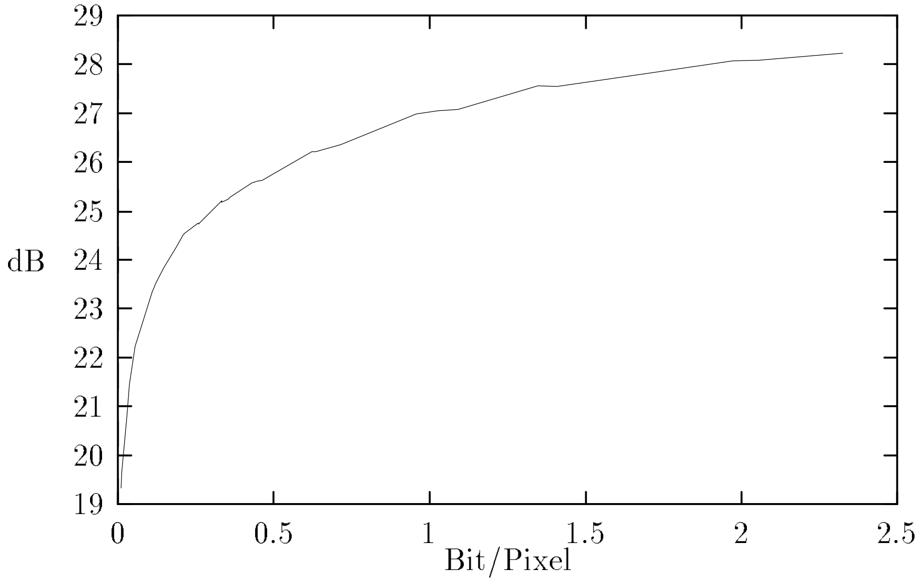


Abbildung 4.6: Zusammenhang von SNR in dB und Coderate in Bit/Pixel. Die Entropie ohne Quantisierung beträgt 28.85 dB.

nach dem Ersetzen der Amplitude das Bild noch zu erkennen ist (Abbildung 4.7, links), kann nach dem Ersetzen der Orientierung das Bild nur noch erahnt werden (Abbildung 4.7, rechts). Die lokale Orientierung trägt also deutlich mehr Information als die lokale Amplitude.

Um festzustellen wie fein die lokale Amplitude und Orientierung quantisiert werden müssen, damit die Bildqualität bei niedriger Coderate möglichst gut ist, haben wir eine weitere Testreihe durchgeführt. Wir messen den SNR bei einer festgelegten Anzahl von Amplituden- und Orientierungswerten (siehe Abbildung 4.8). Die Amplitudenanzahl auf der obersten Pyramidenstufe nimmt 3, 23, 43, 63, 83 und 103 Werte an, während für die Orientierungen 1, 4, 8, 12, 16, 20 und 80 Werte maximal angenommen werden. Auf der x-Achse ist die Steigerung der Coderate in Bit/Pixel aufgetragen, die durch die Erhöhung der maximal möglichen Orientierungswerte entstanden ist. Als Basiscoderate zur Bildung der Differenz dient die erste Messung mit drei Amplitudenwerten und einem Orientierungswert. Ebenso ist auf der y-Achse die Steigerung der Coderate in Bit/Pixel aufgetragen, die durch die Erhöhung der maximal möglichen Amplitudenwerte entstanden ist.

Es ist festzustellen, daß bei maximal 3 Amplitudenwerten eine Steigerung der Orientierungswerte nur geringfügige Auswirkungen hat, da viele Werte zu null quantisiert werden, d.h. in die Totzone fallen, die in den meisten Frequenzbändern doppelt so breit wie eines der übrigen Intervalle ist. Erhöht man die Anzahl der Amplitudenwerte auf 23, gibt es einen steilen Anstieg des SNR-Wertes. Steigert man die Anzahl noch mehr, bewirkt dies nur noch einen geringfügigen Anstieg des SNR-Wertes. Bei der Steigerung der Anzahl der Orientierungen ergibt sich ein anderes Bild: Erhöht man die Orientierungsanzahl von 1 auf 4, entsteht ein starker Anstieg der Coderate. Die weitere Steigerung der Orientie-

Abbildung 4.7: Versuchsreihe: Links wurde vor der Rekonstruktion die lokale Amplitude durch exponentielles Rauschen ersetzt. Rechts wurde vor der Rekonstruktion die lokale Orientierung durch weißes Rauschen ersetzt.

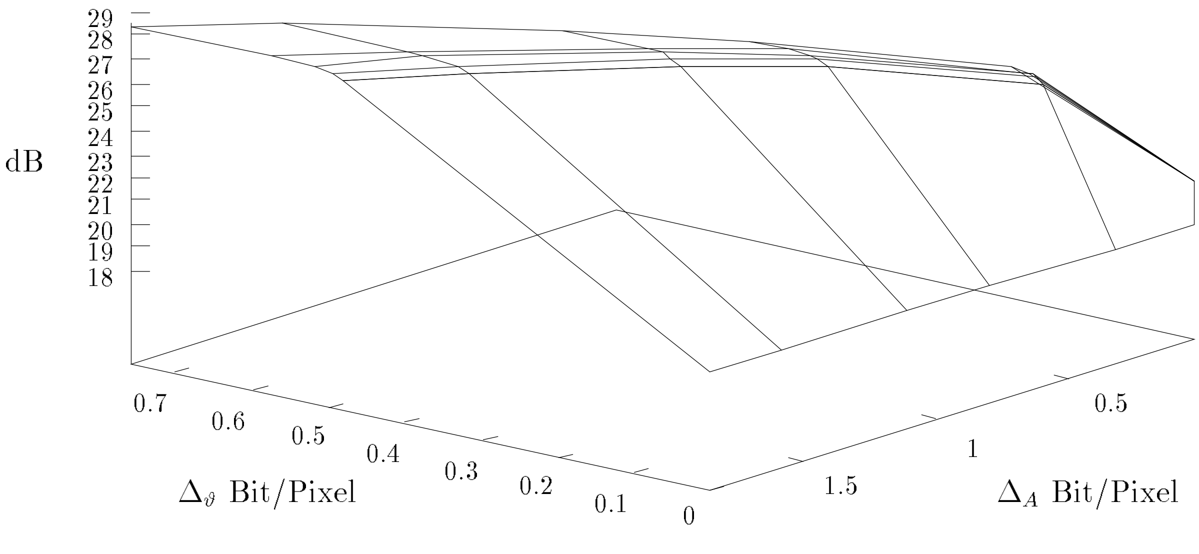


Abbildung 4.8: Entwicklung des SNR bei Erhöhung der möglichen Amplituden- und Orientierungswerte. Dabei stellt Δ_ϑ die durch die Erhöhung der Orientierungswerte und Δ_A die durch die Erhöhung der Amplitudenwerte verursachte Steigerung der Coderate dar. Die Basis für die Differenzbildung bildet eine Messung mit 3 Amplitudenwerten und einem Orientierungswert. Es werden die folgenden Werte angenommen: Amplitude: 3, 23, 43, 63, 83, 103. Orientierung: 1, 4, 8, 12, 16, 20, 80.

rungswerte bewirkt einen weiteren geringfügigen Anstieg des SNR-Wertes.

Zusammenfassend kann man sagen, daß eine gewisse Anzahl von Amplitudenwerten für eine gute Qualität zwingend notwendig ist, während eine hohe Werteanzahl keinen signifikanten Effekt erzielt. Die Orientierungsanzahl sollte 4 Werte nicht unterschreiten. Durch eine Erhöhung dieses Wertes kann auf Kosten der Coderate eine weitere Verbesserung der Bildqualität erzielt werden.

In den folgenden Kapiteln verwenden wir eine Quantisierung von 80 Amplituden- und 20 Orientierungswerten, da wir als vollständige Rekonstruktion ein Bild mit sehr guter Qualität erhalten wollen (siehe Abbildung 4.5 rechts). Wir gehen also davon aus, daß das Bild nicht nur angefordert wird, um einen sehr groben Überblick über den Bildinhalt zu gewinnen. Die relativ schlechte Coderate (ca. 2 Bit/Pixel) könnte durch eine weitere Optimierung des Kompressionsverfahrens deutlich verbessert werden. Das Hauptaugenmerk dieser Arbeit liegt jedoch auf der Lokalisierung und der attentionsgesteuerten progressiven Übertragung, die auf diesem Kompressionsverfahren basieren.

Kapitel 5

Attentive progressive Bildübertragung

In diesem Kapitel gehen wir auf die attentionsgesteuerte progressive Übertragung der berechneten Transformationskoeffizienten ein. Wie wir schon in der Einführung erwähnt haben, wollen wir ein Bild nicht zeilenweise übertragen, sondern zunächst die Regionen rekonstruieren, die den Anwender tatsächlich interessieren. Im folgenden geben wir als erstes eine Einführung, wie wir dieses Problem angehen wollen. Dann erläutern wir die Berechnung der Auffälligkeitsbilder und beschreiben die progressive Übertragung der Koeffizienten.

5.1 Einführung

In den meisten Anwendungen erfolgt die Übertragung von Bildern zeilenweise, d.h. jede Zeile wird komplett rekonstruiert übertragen. Der Nachteil von diesem Ansatz ist, daß der Anwender den Inhalt eines Bildes erst sehr spät (oftmals erst nachdem mehr als 50% des Bildes zu sehen ist) erkennen kann. Sind zudem sehr viele Bilder zu übertragen, aus denen eine Auswahl zu treffen ist (wie z.B. bei einer Anfrage an eine Bilddatenbank), ist die Menge der übertragenen Daten sehr groß. Die progressive Übertragung beseitigt dieses Problem, indem zunächst ein sehr unscharfes Bild übertragen wird, das nach und nach immer schärfer wird. Um ein unscharfes Bild in Originalgröße darzustellen, müssen nur wenige Koeffizienten übertragen werden. Trotzdem gewinnt der Anwender, während sich das Bild immer mehr verbessert, schon sehr früh einen erster Eindruck, was in dem Bild dargestllt ist. Er kann also relativ schnell entscheiden, ob er die Übertragung des kompletten Bildes wünscht oder nicht. Dieser Entscheidungszeitpunkt kann noch früher erreicht werden, wenn die Strukturen, die den Anwender hauptsächlich interessieren, als erstes fein aufgelöst werden. Wir orientieren uns bei unserem Verfahren also an einer Attentionsstragie, bei der der Anwender einen Bildausschnitt vorgibt und das System als

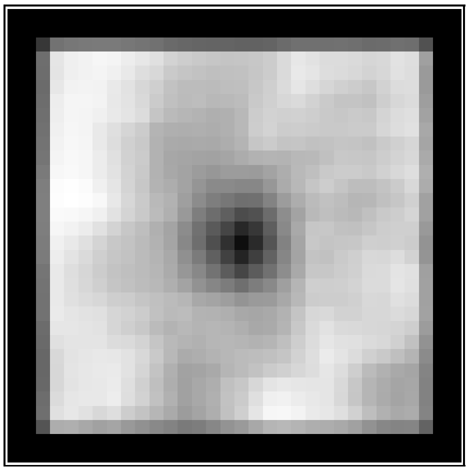



Abbildung 5.1: Beispiel für eine Auffälligkeitskarte: Im Originalbild links wurde nach dem linken Auge gesucht. Rechts: Mit χ^2-Test berechnete Auffälligkeitskarte.

erstes alle Strukturen überträgt, die diesem Bildausschnitt am ähnlichsten sind. Das kann bei Portraitbildern z.B. ein Gesicht oder sogar ein Auge sein, während man bei der Suche nach Bildern mit einem bestimmten Objekt ein Beispiel für dieses vorgeben wird.

Wir beschränken uns in unserem Verfahren auf die Übertragung von einem einzelnen Bild und gehen nicht auf die Suche in einer Bilddatenbank ein. Um unser Verfahren in einer Bilddatenbank anzuwenden, müßte zusätzlich eine Optimierung in Bezug auf Identifikation und nicht nur auf Lokalisierung durchgeführt werden. Dies ist jedoch nicht das Ziel dieser Arbeit. Es wird ein Bildausschnitt gegeben, mit dessen Hilfe von dem zu übertragenden Bild eine Auffälligkeitskarte gebildet wird, die die Übertragungsreihenfolge der Koeffizienten festlegt. Wir benutzen dazu das Kompressionsverfahren aus Kapitel 4 und wenden vier Vergleichsverfahren an, die auf Histogrammen arbeiten.

5.2 Aufgabenbezogene progressive Übertragung

Möchte man eine Strategie zur aufgabenbezogenen progressiven Übertragung entwickeln, muß man sich überlegen, in welcher Reihenfolge die Koeffizienten eines Kompressionsalgorithmus übertragen werden sollen. Betrachtet man die steuerbare Pyramide mit den lokalen Orientierungen, so stellt man fest, daß in den unteren Pyramidenstufen auf Grund der steigenden Bildgröße deutlich mehr Koeffizienten zu finden sind. Auch beim Hochpaßanteil, der genauso groß wie das Originalbild ist, sind viele Koeffizienten zu finden. Der Hochpaßanteil trägt jedoch kaum Informationen, die zum frühen Erkennen des Bildes beitragen, so daß wir diese Koeffizienten zuletzt übertragen. Dagegen benötigen wir alle Koeffizienten des Tiefpaßanteils, um überhaupt einen ersten Eindruck des Bildes geben zu können. Das erste rekonstruierte Bild besteht also nur aus dem zurücktransformierten Tiefpaßanteil.

Für die Übertragung der Bandpaßbilder in den Pyramidenstufen benutzen wir die oben angesprochene Attentionsstrategie. Wir bilden eine Auffälligkeitskarte, die durch einen Vergleich von Modell- und Originalbild entsteht (siehe Abschnitt 5.3). Diese Auffälligkeitskarte (siehe Abbildung 5.1 für ein Beispiel) hat die gleiche Größe wie die oberste Pyramidenstufe, damit möglichst wenig Ortsinformationen übertragen werden müssen. Als oberste Pyramidenstufe bezeichnen wir die gröbste Skala, also das kleinste Koeffizientenbild. Es wird folglich die Übertragungsreihenfolge der Koeffizienten festgelegt. Für jeden Auffälligkeitswert, der sich auf einen Punkt auf der obersten Pyramidenstufe bezieht, wird die gesamte darunterliegende Pyramide übertragen. Es werden also in einem Übertragungsschritt bei einer vierstufigen Pyramide 170 Koeffizienten übertragen: jeweils für Amplitude und Orientierung 1 Koeffizient auf der obersten Stufe, 4 auf der zweiten, 16 auf der dritten und 64 auf der vierten. Ist zum Beispiel ein 256×256 Bild in vier Pyramidenstufen transformiert, erhält man $32 \cdot 32 = 1024$ Übertragungsschritte. Die Auffälligkeitswerte werden sortiert, so daß die Koeffizienten entsprechend ihrer Ähnlichkeit zu dem gesuchten Modellbild übertragen werden können.

Möchte man dieses Verfahren noch etwas verfeinern, kann man eine relative Anzahl von Auffälligkeitswerten festlegen, bis zu der die oben beschriebene Strategie angewendet wird (z.B. 20 % der Werte). Ab diesem Zeitpunkt ist es realistisch anzunehmen, daß das gesuchte Objekt vollständig übertragen wurde. Deshalb wird die Auffälligkeitskarte im weiteren benutzt, um die verbleibenden Koeffizienten pyramidenstufenweise zu übertragen.

Im folgenden Abschnitt werden wir beschreiben, mit welchen Verfahren eine Auffälligkeitskarte berechnet werden kann.

5.3 Verfahren zum Vergleich von Histogrammen

Histogramme stellen die Häufigkeit von Merkmalen in einem Signal dar. Bei Bildern werden oft die Anzahl von Grauwerten oder bei Farbbildern von RGB-Vektoren bestimmt. Wie bereits in [13] gezeigt wurde, ist Farbe ein sehr geeignetes Merkmal für die Identifikation von Objekten. Grauwertbilder haben dagegen nur ein Drittel der Merkmale (Grauwert zu RGB-Vektor), was die Lokalisierung eines Objektes mit Indexing-Verfahren fast unmöglich macht. Um dieses Problem zu lösen, bilden wir Vektoren über die drei obersten Pyramidenstufen der lokalen Orientierungsbilder. Die lokale Amplitude hängt stark von den Beleuchtungsverhältnissen in einer Szene ab. Aus diesem Grund vernachlässigen wir die lokale Amplitude und betrachten im weiteren nur Histogramme über Orientierungsbildern.

Da die Orientierungsbilder der beiden obersten Pyramidenstufen jedoch kleiner als das Orientierungsbild der dritte Pyramidenstufe sind, vergrößern wir beide Bilder auf die Größe der dritten Pyramidenstufe. Um die korrekten Orientierungswerte an den neu hinzugekommenen Pixeln zu berechnen, nutzen wir eine Eigenschaft der steuerbare Pyramide und steuern im Ort, interpolieren die neuen Werte also aus den bekannten Orientierungswerten. Diese Interpolation muß auf den Filterantworten der steuerbaren Pyramide durchgeführt werden. Wir führen eine Überabtastung um den Faktor zwei in x- und y-Richtung der zu vergrößernden Koeffizientenbilder der entsprechenden Pyramidenstufe durch und falten diese Bilder anschließend mit dem Filter $L_1(\vec{\omega})$. Die lokale Orientierung wird dann ganz analog mit der Formel aus Kapitel 3 berechnet. Für das Orientierungsbild der obersten Pyramidenstufe müssen diese Schritte (Überabtasten um Faktor zwei in x- und y-Richtung und Faltung mit $L_1(\vec{\omega})$) zweimal durchgeführt werden, um auf die gleiche Größe wie das Orientierungsbild der dritten Pyramidenstufe zu kommen.

Über diese drei gleich großen Orientierungsbilder werden dann Vektoren gebildet. Auf der Basis dieser Vektoren werden die Histogramme für den Vergleich von Modell- und Originalbild berechnet. Der Histogrammeintrag $\mathcal{H}_V$ für einen Vektor $V = (v_1, v_2, v_3)$ über den Orientierungsbildern der drei obersten Pyramidenstufen O_1, O_2 und O_3 ist wie folgt definiert:

$$\mathcal{H}_V(O_1, O_2, O_3) = \left| \left\{ (x,y) | v_1 = \hat{O}_1(x,y) \wedge v_2 = \hat{O}_2(x,y) \wedge v_3 = O_3(x,y) \right\} \right|,$$

wobei "ˆ" das Steuern im Ort symbolisiert.

In der Literatur gibt es eine Vielzahl von Algorithmen, die speziell die Ähnlichkeit zweier Merkmalshistogramme bestimmen und sich hinsichtlich Komplexität und Leistungsfähigkeit unterscheiden [16]. Wir haben insgesamt vier Verfahren implementiert, um somit genauere Aussagen über die Leistungsfähigkeit der jeweiligen Verfahren treffen zu können. In den folgenden Abschnitten werden wir die Algorithmen näher erläutern und darauf eingehen, welche Voraussetzungen die einzelnen Verfahren benötigen. Die Einschätzung der Leistungsfähigkeit der verschiedenen Algorithmen findet man in Kapitel 6, in dem die Ergebnisse verschiedener Versuchsreihen dargestellt werden.

5.3.1 Rückprojektion

Die Rückprojektion ist, genau wie die Intersektion (siehe Abschnitt 5.3.5), ein Verfahren, das von Swain und Ballard in einem Artikel über "Color-Indexing" [22] vorgestellt wurde. Der Algorithmus wurde entwickelt, um in Farbbildern Objekte zu lokalisieren. Es ist also für uns ohne weiteres möglich, die Rückprojektion zu benutzen, indem die Farbhistogramme durch mehrskalige Orientierungshistogramme ausgetauscht werden. Diese Erweiterung auf Orientierungshistogramme ist zuerst in [13] und [11] vorgestellt worden.

Die Rückprojektion wird in den folgenden Schritten durchgeführt:

1. Berechnung der Pyramidenhistogramme von Original- und Modellbild.

2. Berechnung des Verhältnis-Histogramms aus den beiden Histogrammen.

3. Rückprojektion des Verhältnis-Histogramms auf die Pyramide des Originalbildes.

4. Glättung der Rückprojektion mit einem Boxfilter in der Größe des Modellbildes.

Zunächst werden die Pyramidenhistogramme, genau wie wir es im vorherigen Abschnitt beschrieben haben, erzeugt. Das Verhältnis-Histogramm R wird dann aus den beiden Histogrammen M des Modellbildes und I des Originalbildes wie folgt berechnet:

$$R_i = \begin{cases} \min\left(1, \frac{M_i}{I_i}\right) & , \text{ falls } I_i \neq 0 \\ 0 & , \text{ sonst} \end{cases} \qquad i \in \{1, \ldots, n\}, \tag{5.1}$$

wobei n die Anzahl der maximal möglichen unterschiedlichen Vektoren beschreibt. Diese Anzahl hängt nur von den Quantisierungsschritten ab, also von den Werten, die die jeweiligen Vektorkomponenten annehmen können. Das Verhältnis-Histogramm stellt das Verhältnis der Vorkommen von Vektoren in Original- und Modellbild dar. Es enthält Werte zwischen null und eins, wobei ein Wert nahe eins bedeutet, daß dieser Vektor einen starken Hinweis auf das Objekt darstellt, während ein Wert nahe null genau das Gegenteil impliziert. Tritt ein Vektor des Objektes auch oft im Hintergrund des Originalbildes auf, so ist dieser kein charakteristisches Merkmal für das Vorhandensein des Objektes an einem Ort. Kommt dieser Vektor dagegen nur im Objekt selbst vor, wird der Wert bei eins liegen und somit ein gutes Merkmal darstellen.

Die Rückprojektion projiziert das Verhältnis-Histogramm auf die dritte Stufe der Bildpyramide. Dies ist nötig, da wir nur auf dieser Pyramidenstufe jedem Vektor genau einen Ort zuweisen können. Ist h eine Funktion, die einem Vektor einen Eintrag im Histogramm zuordnet und $v_{x,y}$ der Vektor an der Stelle (x, y), so ist die Rückprojektion definiert durch

$$b_{x,y} = R_{h(v_{x,y})} \qquad \in [0, 1]. \tag{5.2}$$

Wir haben somit eine Auffälligkeitskarte berechnet, die genauso groß wie die dritte Pyramidenstufe ist. Da wir für die progressive Übertragung jedoch eine Auffälligkeitskarte benötigen, die so groß ist wie die erste Pyramidenstufe, filtern wir die berechnete Karte mit einem 4×4 Binomialfilter und nehmen eine Unterabtastung um den Faktor vier in x- und y-Richtung vor.

Im nächsten Schritt wird ermittelt, wo in der rückprojizierten Karte im lokalen Mittel die häufigsten Werte auftreten. Dazu wird die Rückprojektion mit einem Boxfilter B geglättet, der ebenso groß wie die oberste Pyramidenstufe des Modellbildes ist. Stellt $*$ die Faltung dar, ergibt sich die geglättete Rückprojektion durch

$$b = B * b. \tag{5.3}$$

Die Glättung mit dem Boxfilter bewirkt ein Aufsummieren der rückprojizierten Werte in einer Umgebung um den Aufpunkt. Sind in einem Bereich viele hohe Werte, kommt es bei der Aufsummierung ebenfalls zu einem hohen Wert. Dieser Wert wird im Idealfall, d.h. wenn das gesuchte Objekt tatsächlich im Bild zu finden ist, maximal, wenn sich der Boxfilter genau über dem gesuchten Objekt befindet.

Die Rückprojektion, für die nur das Histogramm des Modellbildes übertragen werden muß, ist eine zur Effizienzsteigerung vereinfachte Variante der gleitenden Intersektion (siehe Abschnitt 5.3.5). Die Werte des Verhältnishistogramms lassen sich ebenso wie die Auffälligkeitswerte parallel berechnen, weswegen sich eine Echtzeitimplementierung auf einer Spezialarchitektur anbietet (z.B. MV200 der Firma Datacube). Eine solche Implementierung ist in [13] zu finden.

5.3.2 Bayessche Formel

Dieser Ansatz zur Objektlokalisierung basiert auf der Entscheidungstheorie von Bayes. Dabei wird angenommen, daß das Entscheidungsproblem in wahrscheinlichkeitstheoretischer Form vorliegt und alle nötigen Wahrscheinlichkeiten bekannt sind. Wir beobachten auf der dritten Pyramidenstufe einen Merkmalsvektor M_k und berechnen für diesen die Wahrscheinlichkeit, daß dieser zu einem gesuchten Objekt O_n gehört. Es wird also für die Lokalisierung die Maximum a posteriori Wahrscheinlichkeit berechnet (MAP-Strategie). Diese a posteriori Wahrscheinlichkeit ergibt sich nach der Bayesschen Formel wie folgt:

$$p(O_n|M_k) = \frac{p(M_k|O_n)p(O_n)}{\sum_n p(M_k|O_n)p(O_n)}. \tag{5.4}$$

Dabei ist $p(O_n)$ die a priori Wahrscheinlichkeit des Objektes O_n und $p(M_k|O_n)$ die Wahrscheinlichkeits-Dichtefunktion des Objektes O_n, die sich von der absoluten Häufigkeit des Merkmals im Histogramm von O_n nur durch einen Normalisierungsfaktor unterscheidet.

Die einzugebende a priori Wahrscheinlichkeit ist die erwartete relative Fläche des gesuchten Objektes in Bezug auf die Gesamtfläche. Diese wird sich von Bildklasse zu Bildklasse teilweise stark unterscheiden. Betrachten wir zum Beispiel die Suche nach einem Auge, so wird die a priori Wahrscheinlichkeit bei Portraitbildern ca. 5% bis 8% groß sein, während bei einer Totalen mit nur einem Menschen vielleicht nur 0.5% bis 1% erreicht werden.

Die obige Formel wird benutzt, um aus n Objekten ein ausgezeichnetes herauszufinden. Wir wollen jedoch nur Objekt von Nicht-Objekt bzw. Hintergrund unterscheiden. Bezeichnen wir mit O_O das gesuchte Objekt und mit O_B das Hintergrundobjekt, so vereinfacht sich die Formel zu

$$p(O_O|M_k) = \frac{p(M_k|O_O)p(O_O)}{p(M_k|O_O)p(O_O) + p(M_k|O_B)p(O_B)}, \tag{5.5}$$

wobei $p(O_B)$ die a priori Wahrscheinlichkeit für das Hintergrundobjekt O_B ist, die sich durch

$$p(O_B) = 1 - p(O_O)$$

ergibt. Wir benötigen zur Berechnung der Wahrscheinlichkeit also sowohl das Histogramm des Objektes als auch das des Hintergrundobjektes. Für jeden Vektor $v_{x,y}$ der dritten Pyramidenstufe, berechnen wir mit Hilfe von Gleichung 5.5 die Wahrscheinlichkeit, daß dieser Vektor zu dem Objekt gehört und erhalten somit eine Auffälligkeitskarte, die ebenso groß wie die dritte Pyramidenstufe ist und auf der wir gemäß MAP nach dem Maximum suchen, um das gesuchte Objekt zu lokalisieren.

Da wir jedoch eine Auffälligkeitskarte in der Größe der obersten Pyramidenstufe erhalten wollen, glätten wir das Ergebnis mit einem 4×4 Binomialfilter. Danach wird das geglättete Bild um den Faktor vier in x- und y-Richtung unterabgetastet.

Bayes ist ebenfalls ein effizientes Verfahren. Es benötigt allerdings sowohl das Histogramm des gesuchten Objektes also auch des Hintergrundes, was den Nutzen für einige Anwendungen erheblich einschränkt. Die Schätzung der a priori Wahrscheinlichkeit über die relative Fläche wirkt zudem einer Robustheit gegenüber Skalierung entgegen.

5.3.3 Gleitende Histogramme

Die bisherigen Verfahren arbeiten alle punktweise auf dem gesamten Originalbild, wobei jedem Punkt ein Aufmerksamkeitswert zugeordnet wird, der nur von dem an diesem Punkt gefundenen Vektor abhängt. Eine allgemeinere Strategie ist es, um jeden Aufpunkt ein Histogramm zu bilden, das ein Gebiet in der Größe des Modellbildes einbezieht. Dazu wird über das Originalbild ein Fenster geschoben, das der Größe des Modellbildes entspricht (siehe Abbildung 5.2). Berechnet man diese gleitenden Histogramme auf einem sequentiellen Computer, bietet es sich aus Effizienzgründen an, die Verschiebung des Fensters mäanderförmig durchzuführen. Dazu fügt man in ein Histogramm die durch die Verschiebung neu hinzukommenden Vektoren ein und entfernt gleichzeitig die aus dem Fenster gerückten Vektoren. Auf einem parallelen Computer bringt diese Strategie allerdings keinen Effizienzvorteil. Statt dessen werden die gesamten Histogramme für jeden Aufpunkt parallel berechnet.

Die so berechneten Histogramme werden mit dem Histogramm des Modellbildes verglichen, um zu einer Auffälligkeitskarte zu gelangen. Da wir für diesen Vergleich das

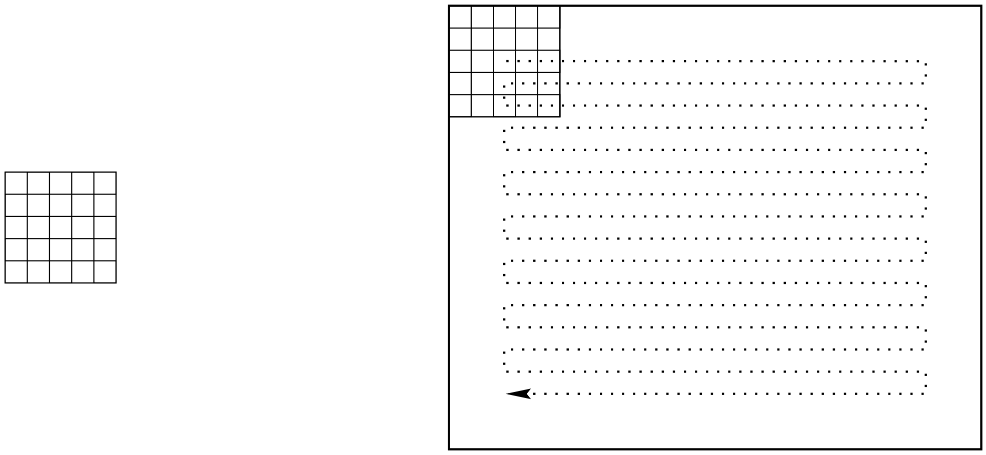


Abbildung 5.2: Gleitendes Histogramm: Links ist ein Modellbild abgebildet, über dem ein Histogramm berechnet werden soll. Rechts: Mäanderförmiges Verschieben des Histogrammfensters über das Originalbild.

Auftreten von Randeffekten vermeiden wollen, verschieben wir das Histogrammfenster nicht über den Rand des Originalbildes heraus. Dadurch wird im Auffälligkeitsbild ein Randstreifen erzeugt, für den keine Attentionswerte berechnet werden, was für unser Verfahren jedoch keine Verzerrung darstellt, da die gesuchten Objekte nicht direkt am Rand lokalisiert sind. Möchte man auch Objekte am Rand finden, muß der Randbereich mit einbezogen werden, indem z.B. Fensterteile, die aus dem Originalbild herausragen, nicht in das Histogramm mit einbezogen werden, so daß die Histogramme immer dünner besetzt sind.

Um die Verfahren, die die gleitenden Histogramme nutzen, mit der Rückprojektion und Bayes vergleichen zu können, werden die Histogramme auf der dritten Pyramidenstufe gebildet und verglichen. Die so errechnete Auffälligkeitskarte wird dann, genau wie bei den anderen Verfahren auch, mit einem 4×4 Binomialfilter gefiltert und um den Faktor vier unterabgetastet.

In den folgenden Abschnitten betrachten wir zwei Verfahren, mit denen für einen Aufpunkt ein Attentionswert berechnet werden kann.

5.3.4 χ^2-Test

Der χ^2-Test ist ein statistisches Verfahren, um festzustellen, ob zwei Verteilungen von der gleichen oder von zwei verschiedenen Verteilungsfunktionen stammen. Wir berechnen also den "Abstand" zwischen dem gleitenden Histogramm und dem Histogramm des Modellbildes, um bis zu einem gewissen Grad der Sicherheit annehmen zu können, ob an der untersuchten Stelle, das gesuchte Objekt zu finden ist. Der χ^2-Test von zwei Histogrammen M des Modellbildes und I des Ausschnittes aus dem Originalbild ist definiert durch

$$\chi^2(I, M) = \sum_i \frac{(I_i - M_i)^2}{I_i + M_i}. \tag{5.6}$$

Man kann sehen, daß Vergleichswerte von null – wenn die Histogramme genau übereinstimmen – bis zu der Summe aller Histogrammeinträge der beiden Histogramme – wenn kein Histogramm-Merkmal in beiden Histogrammen zu finden ist – angenommen werden können.

Der χ^2-Test ist ein sehr aufwendiges Verfahren, da Gleichung 5.6 für jeden Punkt der Auffälligkeitskarte berechnet werden muß.

5.3.5 Intersektion

Die Intersektion ist ein Verfahren, das von Swain und Ballard in einem Artikel über "Color-Indexing" [22] zum Vergleich von einem Objekthistogramm mit Histogrammen einer Datenbank entwickelt wurde. Wir ersetzen die RGB-Vektoren durch Pyramidenvektoren und vergleichen mit dem Verfahren ein Objekthistogramm mit den gleitenden Histogrammen. Das Verfahren setzt die gleiche Größe der zu vergleichenden Histogramme voraus. Diese Voraussetzung, die bei Swain und Ballard nicht gegeben war, ist in unserem Fall erfüllt, so daß wir keine weitere Normierung durchführen müssen. Ist M das Histogramm des Modellbildes und I des Originalbildes, so ist die Intersektion definiert durch

$$\mathcal{I}(I, M) = \sum_i \min(I_i, M_i). \tag{5.7}$$

Das Ergebnis der Intersektion ist die Anzahl der Vektoren, die in Modell- und Originalbild (bzw. in dem Ausschnitt des Originalbildes) miteinander korrespondieren. Dieses Ergebnis wird bei Swain und Ballard auf Werte zwischen null und eins normiert, was jedoch nichts an der Reihenfolge der Werte ändern würde, die wir für die aufgabenbezogene progressive Übertragung benötigen. Aus diesem Grund verzichten wir auf die Normierung.

Die Intersektion ist fast genauso aufwendig wie der χ^2-Test was die Berechnung der Histogramme betrifft, hat jedoch einen kleinen Effizienzvorteil durch die gegenüber Gleichung 5.6 einfachere Minimumbildung.

Kapitel 6

Versuchsreihen

Um die Leistungsfähigkeit der von uns entwickelten progressiven Übertragungsstrategie genau zu untersuchen, haben wir eine Vielzahl von Versuchsreihen durchgeführt. Im nächsten Abschnitt werden wir erläutern, welche Versuche wir durchgeführt haben. Dabei beschreiben wir die Rahmenbedingungen und stellen die von uns gewählten Testbilder vor. Der darauf folgende Abschnitt beschreibt die Versuchsergebnisse und erläutert dabei auftretende Phänomene.

6.1 Versuchsreihen

In diesem Abschnitt werden wir alle durchgeführten Versuche beschreiben. Einige der Versuche sind hinzugekommen, nachdem vorherige Ergebnisse gedeutet worden sind. Wir haben dabei versucht, uns systematisch an die Grenzen der Leistungsfähigkeit des Verfahrens heranzutasten.

6.1.1 Rahmenbedingungen

Um alle Versuchsreihen miteinander vergleichen zu können, haben wir diese möglichst unter den gleichen Parametrisierungen durchgeführt. Nur bei einer Versuchsreihe zur Augenlokalisierung variieren wir bewußt die Quantisierungsparameter, um die Auswirkung auf die Lokalisierungseigenschaften feststellen zu können. Bis auf diese Ausnahme haben wir alle Versuchsreihen mit den gleichen Quantisierungsstufeneinteilungen getestet, die auch für die Kompression benutzt wurde. Dabei ist es wichtig, anzumerken, daß der Schwerpunkt dieser Arbeit auf der Lokalisierung und der anschließenden Übertragung liegt. Die Quantisierungsparameter sind also in Bezug auf diese Zielstellung gewählt, was auch die relativ schlechten Kompressionsergebnisse erklärt.

Abbildung 6.1: Originalbild (links) und ausgeschnittenes Auge als Modellbild (rechts).

Innerhalb einer Versuchsreihe haben alle Testbilder die gleiche Größe, um Verfälschungen der Ergebnisse zu vermeiden. Sollen Ergebnisse aus unterschiedlichen Versuchsreihen verglichen werden, haben wir die Parameter gleichfalls übereinstimmend gewählt.

Für eine qualitative Einschätzung der Versuchsergebnisse benötigen wir ein Qualitätsmaß, das einen Vergleich der verschiedenen Verfahren ermöglicht. Da es durch die subjektive Aufgabenstellung (z.B. wann beide Augen zu sehen sind) unmöglich ist, ein objektives Verfahren zu entwickeln, das entscheidet, ob ein gesuchter Bildausschnitt vollständig decodiert ist, weichen wir auf ein subjektives Vergleichsverfahren aus. Die progressive Decodierung eines Bildes wird mehreren Testpersonen vorgeführt. Diese haben dann im Vergleich mit den progressiven Decodierungen der anderen Verfahren den Decodierungsschritt bestimmt, bei dem das gesuchte Objekt vollständig decodiert ist.

6.1.2 Versuchsreihe: Augenlokalisierung in Portraitbildern

In dieser Versuchsreihe wollen wir die Bildklasse der Portraitbilder untersuchen. Möchte man ein Gesicht erkennen, so ist es sehr wichtig, daß die Augen zu sehen sind, da diese eine große Menge Information tragen. Deshalb geben wir als Modellbild ein Auge vor. Das Auge wird für den ersten Versuch direkt aus dem Originalbild (siehe Abbildung 6.1) geschnitten, um zu verifizieren, daß die Lokalisierung eines im Bild vorkommenden Ausschnitts in wenigen Schritten möglich ist. Da dieses Originalbild das Referenzbild zu den weiteren Versuchen darstellen soll, haben wir eine en face Darstellung gewählt. Die Quantisierung der Orientierungsbilder nimmt vier Orientierungen bei kleinen Amplituden an und steigert sich bis zu 20 Orientierungen bei hohen Amplituden, d.h. es wird eine Quantisierung gewählt, die durch die Vektorquantisierung von den Amplitudenwerten abhängt.

Eine Untersuchung, bei der die Auswirkungen einer Quantisierungsparameteränderung betrachtet werden, ist in Abschnitt 6.1.3 zu finden.

Referenzversuch

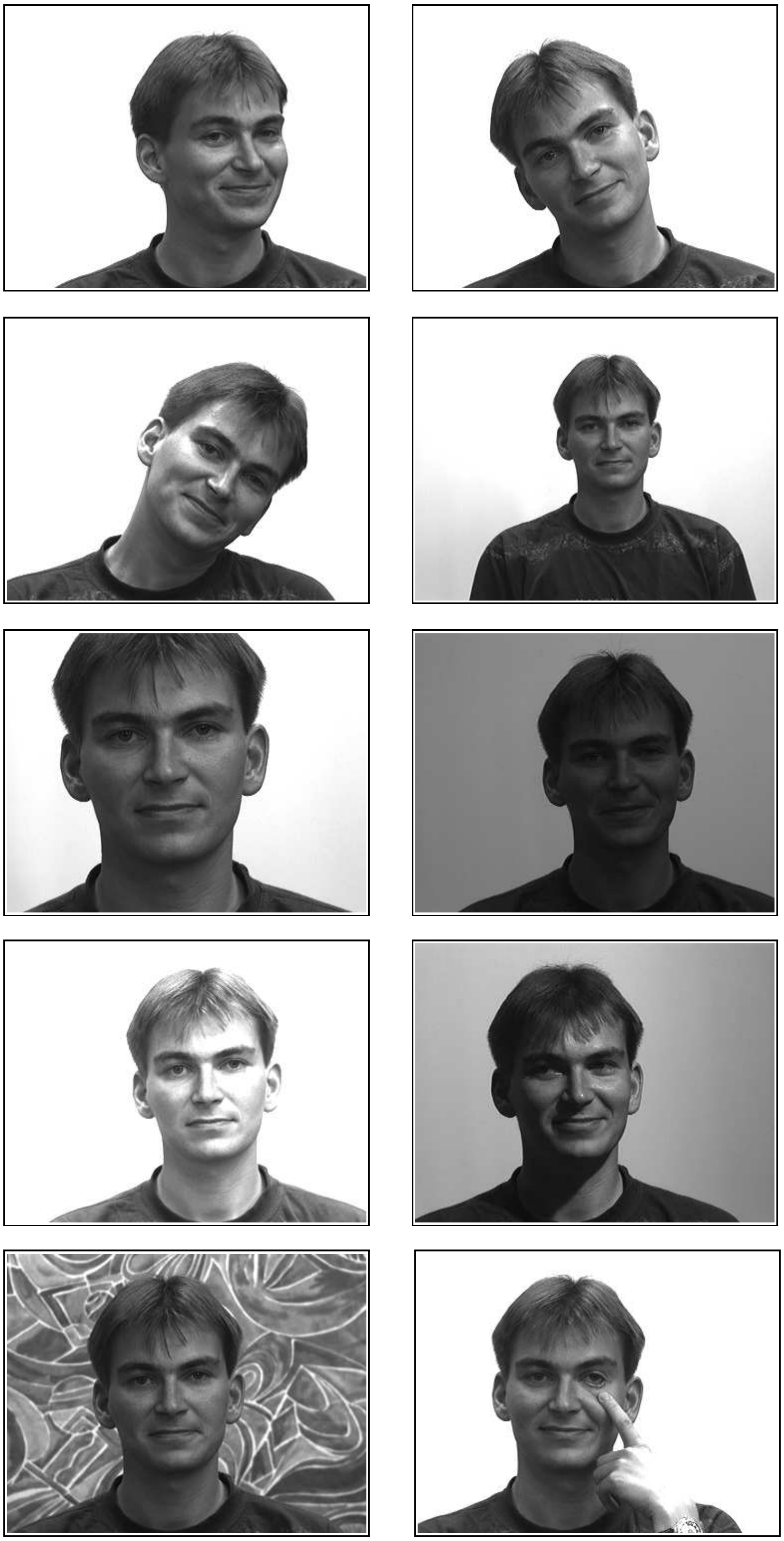
1.	Lokalisierung des Referenzauges im Referenzbild a. Lokalisierung des linken Auges

Wie wir schon in Kapitel 5 erläutert haben, werden die beiden obersten Pyramidenstufen im Ort gesteuert, um diese auf die Größe der dritten Pyramidenstufe auszudehnen. So können wir die Komponenten eines Vektors auf jeder Ebene exakt lokalisieren. Um zu zeigen wie es sich auf die Lokalisierungseigenschaften auswirkt, wenn nicht im Ort gesteuert wird, haben wir den Referenzversuch wiederholt, wobei wir jedoch nicht im Ort gesteuert haben. Alle anderen Parameter haben wir für diesen Versuch konstant gelassen.

Referenzversuch ohne örtliches Steuern

2.	Lokalisierung des Referenzauges im Referenzbild ohne örtliches Steuern a. Lokalisierung des linken Auges

Da es a priori nicht bekannt ist, wie das zu übertragende Bild aussieht, d.h. wie die Person auf dem Portrait dargestellt ist, müssen wir untersuchen wie empfindlich unser Algorithmus reagiert, wenn man ein en face Modellbild (also das Auge des Referenzbildes) vorgibt und das zu übertragende Bild zwar die gleiche Person enthält, die Darstellung jedoch abweicht. Deshalb haben wir in zehn weiteren Versuchsreihen die Robustheit der Verfahren gegenüber einer Variation des Bildinhalts untersucht. Da wir uns auf Portraitbilder beschränken, müssen nur Parameter variiert werden, die in dieser Bildklasse vorkommen können. Dazu gehören Änderung der Blickrichtung (Halbprofil), Rotation um die optische Achse, Änderung der Skala, Änderung des Kontrastes, Änderung des Hintergrundes und Verzerrung des gesuchten Objektes. Die von uns ausgewählten Testbilder sind in Abbildung 6.2 dargestellt. In der folgenden Tabelle werden die beschriebenen Versuche noch einmal aufgelistet.



Rotation
Neigung 1
Neigung 2
Entfernt
Dicht
Niedriger Kontrast
Hoher Kontrast
Natürlich
Hintergrund
Verzerrung

Abbildung 6.2: Testbilder der Robustheitsuntersuchung.

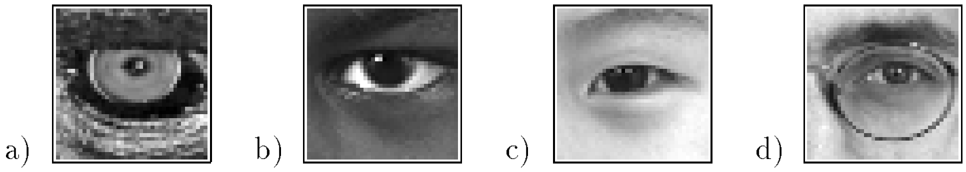


Abbildung 6.3: Modellbilder: Augen mit vom Referenzobjekt abweichender Orientierungsverteilung. a) Affenauge, b) Auge eines schwarzen Jungen, c) Auge eines asiatischen Mädchens, d) Auge mit Brille.

Robustheit

3.	Robustheit gegenüber Änderungen der Blickrichtung a. Drehung des Kopfes um ca. 22.5°
4.	Robustheit gegenüber Rotationen um die optische Achse a. Neigung des Kopfes um ca. 22.5° b. Neigung des Kopfes um ca. 45°
5.	Robustheit gegenüber Skalenänderungen a. Größere Entfernung zur Kamera b. Geringere Entfernung zur Kamera
6.	Robustheit gegenüber Kontraständerung a. Niedriger Kontrast b. Hoher Kontrast
7.	Robustheit gegenüber Beleuchtungsqualität a. Natürliche Beleuchtung (Schlagschatten)
8.	Robustheit gegenüber Hintergrundstrukturen a. En face Darstellung mit texturiertem Hintergrund
9.	Robustheit gegenüber Verzerrung a. Verzerrung des linken Auges

Bei den oben beschriebenen Versuchen wurde als Modellbild das Auge der im Referenzbild dargestellten Person benutzt, d.h. die Augenform des gesuchten Auges stimmt in etwa mit denen der Testbilder überein. Um festzustellen wie die Verfahren auf andere Augenformen reagieren, haben wir eine weitere Versuchsreihe durchgeführt. Dazu haben wir vier Augen ausgewählt, die eine andere Orientierungsverteilung erzeugen als das Referenzauge (siehe Abbildung 6.3).

Das erste Auge ist ein Affenauge, das sehr rund ist und somit alle Orientierungen annähernd gleichverteilt erzeugt. Rechts daneben ist ein Auge eines schwarzen Jungen dargestellt. Auch hier werden die Orientierungen anders sein, da sich der Kontrast stark von dem Kontrast des Referenzauges unterscheidet und die Grauwertübergänge teilweise umgekehrt sind. Das dritte Modellbild ist das Auge eines asiatischen Mädchens, dessen Form mandelförmig ist, während die Orientierungsverteilung des vierten Auges durch eine Brille verändert wird. In der nachfolgenden Tabelle sind die durchgeführten Versuche noch einmal zusammengefaßt.

Fremdaugen

10.	Lokalisierung eines Fremdauges im Referenzbild a. Lokalisierung des Affenauges b. Lokalisierung des Auges eines schwarzen Jungen c. Lokalisierung des Auges eines asiatischen Mädchens d. Lokalisierung des Auges mit Brille

6.1.3 Versuchsreihe: Quantisierungsvariation

Diese Versuchsreihe untersucht, wie stark die Ergebnisse der vier Vergleichsverfahren von der Wahl der Quantisierungsparameter abhängen.

Die einfachste Möglichkeit, eine Quantisierungsstufe für die Lokalisierung zu wählen, besteht darin, die gleiche Quantisierungsstufe zu benutzen, die auch bei der Kompression zur Anwendung kommt. Für die durchgeführten Versuche verwenden wir auf den verschiedenen Pyramidenstufen unterschiedliche Amplitudenfeinheiten (30–80 mögliche Werte). Die Orientierungsquantisierung hängt von der Amplitude ab. Ist die Amplitude klein, werden nur vier verschiedene Werte angenommen, während höhere Amplitudenwerte bis zu 20 Orientierungswerte implizieren. In den quantisierten Orientierungswerten ist also Amplitudeninformation enthalten. Dieser Versuch wurde bereits als Referenzversuch (Versuch 1) durchgeführt.

Möchte man eine Orientierungsquantisierung erhalten, die nicht von den Amplitudenwerten abhängt (bis auf die Totzone, die eine "Nicht-Orientierung" erzeugt), muß die mögliche Orientierungswerteanzahl fix gewählt werden. Für diese Versuchsreihe überprüfen wir, wie der Algorithmus auf vier bis 20 fixe Orientierungswerte reagiert. Die Versuche der Versuchsreihe sind noch einmal in der folgenden Tabelle dargestellt.

Quantisierung der Orientierungswerte

11.	Quantisierung mit fixer Anzahl von Orientierungswerten a. Quantisierung mit 4 Orientierungswerten b. Quantisierung mit 8 Orientierungswerten c. Quantisierung mit 12 Orientierungswerten d. Quantisierung mit 16 Orientierungswerten e. Quantisierung mit 20 Orientierungswerten

Wie sich später herausstellen wird, ist die Leistungsfähigkeit der Quantisierung des Referenzversuches und des Versuches 11a. in Bezug auf die Lokalisierung vergleichbar. Zudem liefert die gleitende Intersektion die besten Ergebnisse. Aus diesem Grund haben wir die Versuche 4 und 6 mit einer fixen Orientierungsquantisierung von vier Orientierungswerten wiederholt, um die Leistungsfähigkeit dieser Quantisierung mit der Leistungsfähigkeit der gewählten Quantisierung bei der Augenlokalisierung zu vergleichen. Die Versuche sind in der folgenden Tabelle dargestellt:

Augenlokalisierung bei vier Orientierungswerten

12.	Lokalisierung bei vier Orientierungswerten a. Neigung des Kopfes um ca. 22.5° b. Neigung des Kopfes um ca. 45° c. Niedriger Kontrast d. Hoher Kontrast

6.1.4 Versuchsreihe: Lokalisierung von Personen

Diese Versuchsreihe soll untersuchen, wie sicher bezüglich der Orientierungsverteilung komplexere Objekte wie z.B. Gesichter in einem Bild gefunden werden können. Gleichzeitig ist auch für die Erkennbarkeit von Personen besonders wichtig, daß Gesichter früh zu erkennen sind. Wir geben also als Modellbild einen Bildausschnitt vor, in dem ein Gesicht zu sehen ist. Als zu übertragendes Bild haben wir die in Abbildung 6.4 dargestellte Szene gewählt.

Ein Großteil des gewählten Bildes besteht aus Hintergrund und Körpern der dargestellten Personen, so daß die Algorithmen von den zu lokalisierenden Gesichtern abgelenkt werden könnten. Zudem unterscheiden sich die Kopfformen und Frisuren der dargestellten Personen erheblich, d.h. daß ein vorgegebenes Modellbild nicht mit allen Gesichtern gleich gut übereinstimmen kann.

In Abbildung 6.5 sind die von uns gewählten Modellbilder zu sehen, mit denen wir im Testbild die Personensuche durchgeführt haben. Der linke Kopf ist direkt aus der Szene ausgeschnitten worden, weist jedoch eine ungewöhnliche Kopfform auf. Der zweite Kopf ist dagegen nicht in der Szene vorhanden. Dieser zweite Fall wird der von uns angestrebten Anwendung näherkommen, da in den meisten Fällen der Bildinhalt des zu übertragenden Bildes unbekannt sein wird.

Diese Versuchsreihe hat auch eine große praktische Bedeutung. Es gibt viele Anwendungen, bei denen nur das Gesicht einer Person übertragen werden muß. So ist zum Beispiel bei Videokonferenzen oder beim Bildtelefon weder der umgebende Hintergrund noch der Körper einer Person von Interesse.

Die durchgeführten Versuche sind noch einmal in der folgenden Tabelle aufgelistet:

Abbildung 6.4: Testbild für die Lokalisierung von Personen.

Abbildung 6.5: Modellbilder für die Lokalisierung von Personen. Links: Kopf einer im Bild dargestellten Person. Rechts: Kopf einer nicht im Bild vorhandenen Person.

Abbildung 6.6: Suche nach dem Hintergrund in einer Szene mit mehreren Personen. Umrandet: Ausgeschnittener Hintergrundbereich.

Gesichtslokalisierung

13.	Lokalisierung von Gesichtern in einer Szene mit mehreren Personen a. Lokalisierung eines in der Szene vorkommenden Gesichtes b. Lokalisierung eines nicht in der Szene vorkommenden Gesichtes

Für die in diesem Abschnitt durchgeführten Versuche haben wir immer ein sehr spezifischen Modellbild verwendet. Um zu zeigen, daß die Lokalisierungseigenschaften nicht nur auf einer geschickten Auswahl der Modellbilder beruhen, haben wir einen Versuch durchgeführt, bei dem nach einem sehr unspezifischen Modellbild gesucht wird. Wir haben, wie in Abbildung 6.6 dargestellt ist, als Modellbild einen Teil aus dem Hintergrund geschnitten und diesen für die Suche benutzt. Der ausgewählte Bereich ist leicht texturiert, so daß das Modellbild auch Orientierungen enthält, die in anderen Teilen des Testbildes ebenfalls zu finden sind.

Hintergrundlokalisierung

14.	Lokalisierung eines unspezifischen Objektes in einer Szene mit mehreren Personen a. Lokalisierung des Hintergrundes

Abbildung 6.7: Szene mit ähnlichen Gegenständen (links) und ausgeschnittener Gegenstand als Modellbild (rechts).

Arbeitet der Algorithmus wie erwartet, sollte zunächst der Hintergrund, dann die relativ einfarbigen Körperpartien und zum Schluß die Gesichter decodiert werden.

6.1.5 Versuchsreihe: Lokalisierung von Gegenständen

In den bisherigen Versuchen haben wir zumeist Teile von Personen lokalisiert. Oftmals sollen jedoch auch Bilder übertragen werden, auf denen nur Gegenstände dargestellt sind. Wir wollen also in einer Versuchsreihe feststellen, wie genau ein bekannter Gegenstand in einem Bild lokalisiert werden kann. Dieser Versuch läßt sich mit den von H. Pahl durchgeführten Versuchen zur farbkonstanten Objektlokalisierung aus lokalchromatischen Orientierungshäufigkeiten vergleichen (siehe [13]). Im Gegensatz zu den dort untersuchten Farbbildern stellen Grauwertbilder eine höhere Anforderung an die Lokalisierungseigenschaften der verwendeten Repräsentation, da die bedeutende Merkmalsdimension der Farbe fehlt.

Wir haben als Versuchsbild ein Bild benutzt, das in den Untersuchungen von H. Pahl verwendet wurde (siehe Abbildung 6.7). Das Bild zeichnet sich durch zwei sehr ähnliche Gegenstände aus, bei deren Lokalisierung auf Farbbildern der in [13] beschriebene Algorithmus Schwierigkeiten hatte.

Objektlokalisierung

15.	Lokalisierung eines Objektes in einer Szene mit ähnlichen Gegenständen a. Lokalisierung einer im Bild dargestellten Dose

6.2 Versuchsergebnisse

In diesem Abschnitt stellen wir die Ergebnisse der im vorherigen Abschnitt beschriebenen Versuchsreihen vor und erläutern diese. Aus Platzgründen werden wir nicht für jeden Versuch eine Bildfolge abbilden, sondern dies auf ausgewählte Versuche beschränken. Um die Leistung der Algorithmen bei den einzelnen Versuchen miteinander vergleichen zu können, haben wir bei allen Portraitbildversuchen die Anzahl der Decodierungsschritte gezählt, die benötigt werden, bis das gesuchte Objekt vollständig decodiert ist. Da wir immer zehn Decodierungsschritte auf einmal betrachten, werden bei einer Bildgröße des Auffälligkeitsbildes von 45×35 Pixeln insgesamt 158 Schritte bis zur vollständigen Decodierung des Bildes durchgeführt ($45 \cdot 35 = 1575$ DIV $10 = 157$ Schritte, der Hochpaßanteil wird zuletzt hinzugefügt, so daß man insgesamt auf 158 Schritte kommt). Wir veranschaulichen diese Vergleichsdaten mit Hilfe von Balkendiagrammen. Dabei sind wir in Einzelfällen von dem gewählten Maßstab (50 Decodierungsschritte) abgewichen, wenn die Daten eine solche Änderung erforderlich gemacht haben.

Bei den Versuchen zur Personenlokalisierung in einer Szene mit mehreren Personen geben wir dieses subjektive Qualitätsmaß nicht an, da schon eine nicht vollständig decodierte Person, den gesamten Erfolg bei der Erkennung der anderen Personen zu nichte machen würde. Aus diesem Grund werden wir diese Versuchsergebnisse jeweils anhand von Bildfolgen der Decodierungssequenz erläutern.

Auf Seite 103 befindet sich ein Faltblatt, auf dem eine Übersicht über die durchgeführten Versuche zu finden ist.

6.2.1 Versuchsreihe: Augenlokalisierung in Portraitbildern

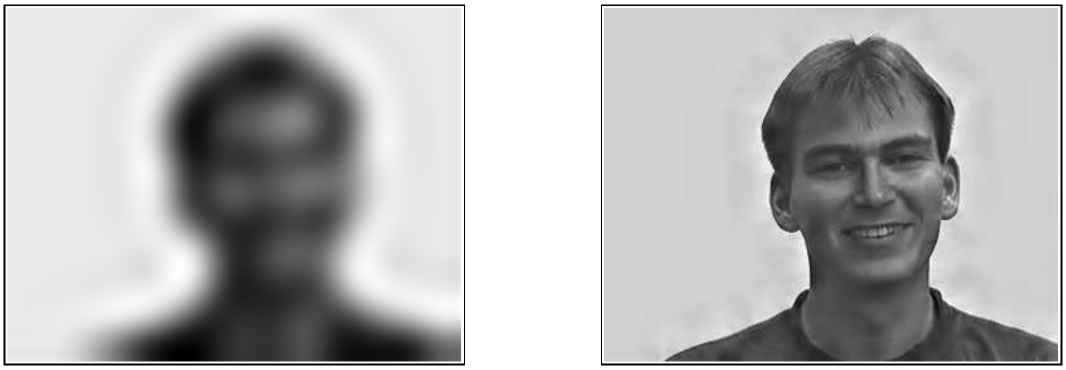


Abbildung 6.8: Erstes (links) und letztes Bild (rechts) der Decodierungssequenz des Referenzversuchs (Versuch 1).

Für alle weiteren Versuche zur Augenlokalisierung wird der Versuch 1 als Referenzversuch dienen. Aus diesem Grund stellen wir alle Vergleichsverfahren (gleitender χ^2-Test, glei-

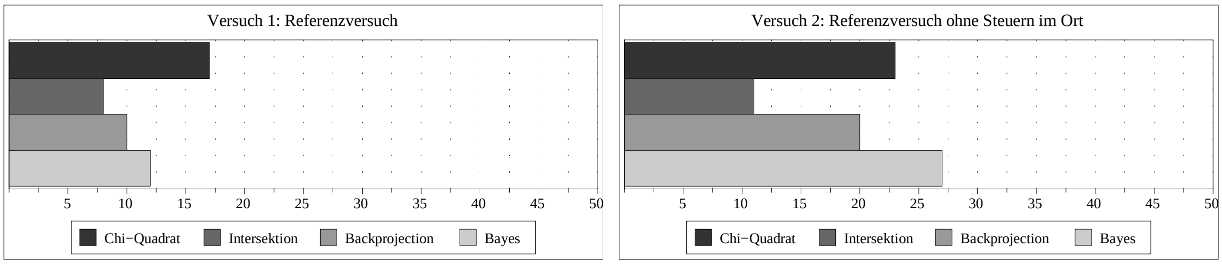


Abbildung 6.9: Balkendiagramm der für die Augenrekonstruktion benötigten Decodierungsschritte – Versuch 1 (Referenzversuch) und 2 (Referenzversuch ohne örtliches Steuern).

tende Intersektion, Rückprojektion und Bayes) in jeweils fünf Bildern dar. In Abbildung 6.8 ist sowohl das Anfangsbild, d.h. das Bild, das von jedem Verfahren zuerst übertragen wird, als auch das letzte vollständig decodierte Bild zu sehen. Bei den abgebildeten Bildfolgen stellen wir für jedes Vergleichsverfahren jeweils fünf Bilder der Decodierungssequenz dar. Dabei beschränken wir uns in Viererschritten auf die ersten 20 Decodierungsschritte, da bei den meisten Versuchen nach dieser Anzahl von Schritten das Objekt vollständig decodiert ist.

In Abbildung 6.9 sind die Ergebnisse der Versuche 1 (Referenzversuch) und 2 (Referenzversuch ohne Steuern im Ort) graphisch dargestellt. Man sieht, daß sich die Lokalisierungseigenschaften deutlich verschlechtern, wenn nicht im Ort gesteuert wird. Bei den gleitenden Verfahren ist die Verbesserung nicht so groß ist (χ^2-Test: 17 zu 23, Intersektion: 8 zu 11) wie bei den beiden anderen Verfahren (Rückprojektion: 10 zu 20, Bayes: 12 zu 27). Des weiteren ist festzustellen, daß die gleitende Intersektion das Verfahren mit den besten Lokalisierungseigenschaften ist. Dabei kommt es nicht nur auf das exakte Lokalisieren des Punktes an, an dem die Histogramme eine maximale Ähnlichkeit aufweisen. Die diesen Punkt umgebenden Punkte müssen ebenfalls einen hohen Auffälligkeitswert besitzen, damit das Objekt sehr früh vollständig decodiert wird.

Die guten Lokalisierungseigenschaften sind auch in den dargestellten Bildfolgen (siehe Abbildung 6.10 und 6.11) festzustellen. Während beim χ^2-Test erst beim fünften Bild beide Augen zu sehen sind, sind diese bei der Intersektion schon nach zwei Bildern vollständig decodiert.

Die Versuche, die wir zur Untersuchung der Robustheit durchgeführt haben, bestätigen bis auf zwei Ausnahmen dieses erste Ergebnis. In Abbildung 6.12 ist das Balkendiagramm einer solchen Ausnahme zu sehen. Während die gleitende Intersektion im Vergleich zu dem gleitenden χ^2-Test und zu Bayes ein gutes Ergebnis zeigt (15 zu 26 und 20), sind bei der Rückprojektion schon nach fünf Schritten beide Augen vollständig decodiert.

Bei den Versuchen 4 und 5 zeigt die gleitende Intersektion dagegen die deutlich besten Lokalisierungseigenschaften (siehe Abbildung 6.13). Dabei fällt auf, daß die gleitenden

Versuch 1 – χ^2-Test **Versuch 1 – Intersektion**

4 Schritte 4 Schritte

8 Schritte 8 Schritte

12 Schritte 12 Schritte

16 Schritte 16 Schritte

20 Schritte 20 Schritte

Abbildung 6.10: Bildfolgen des Referenzversuchs (Versuch 1): χ^2-Test (links), Intersektion (rechts).

Versuch 1 – Rückprojektion


4 Schritte


8 Schritte


12 Schritte

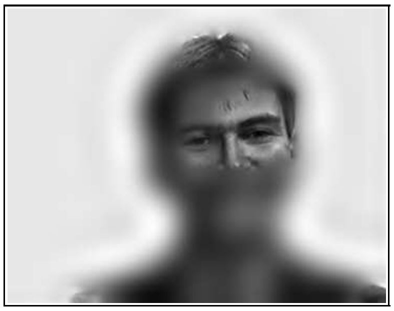

16 Schritte


20 Schritte

Versuch 1 – Bayes


4 Schritte


8 Schritte


12 Schritte


16 Schritte


20 Schritte

Abbildung 6.11: Bildfolgen des Referenzversuchs (Versuch 1): Rückprojektion (links), Bayes (rechts).

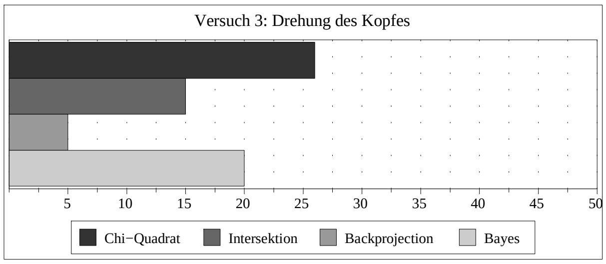


Abbildung 6.12: Balkendiagramm der für die Augenrekonstruktion benötigten Decodierungsschritte – Versuch 3 (Drehung des Kopfes).

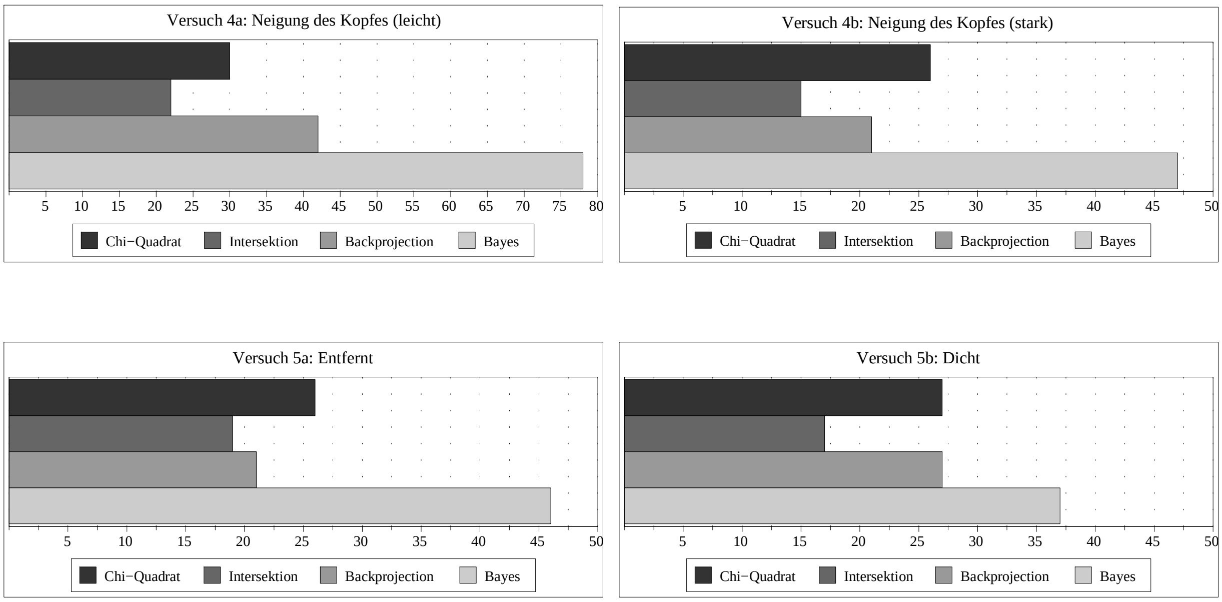
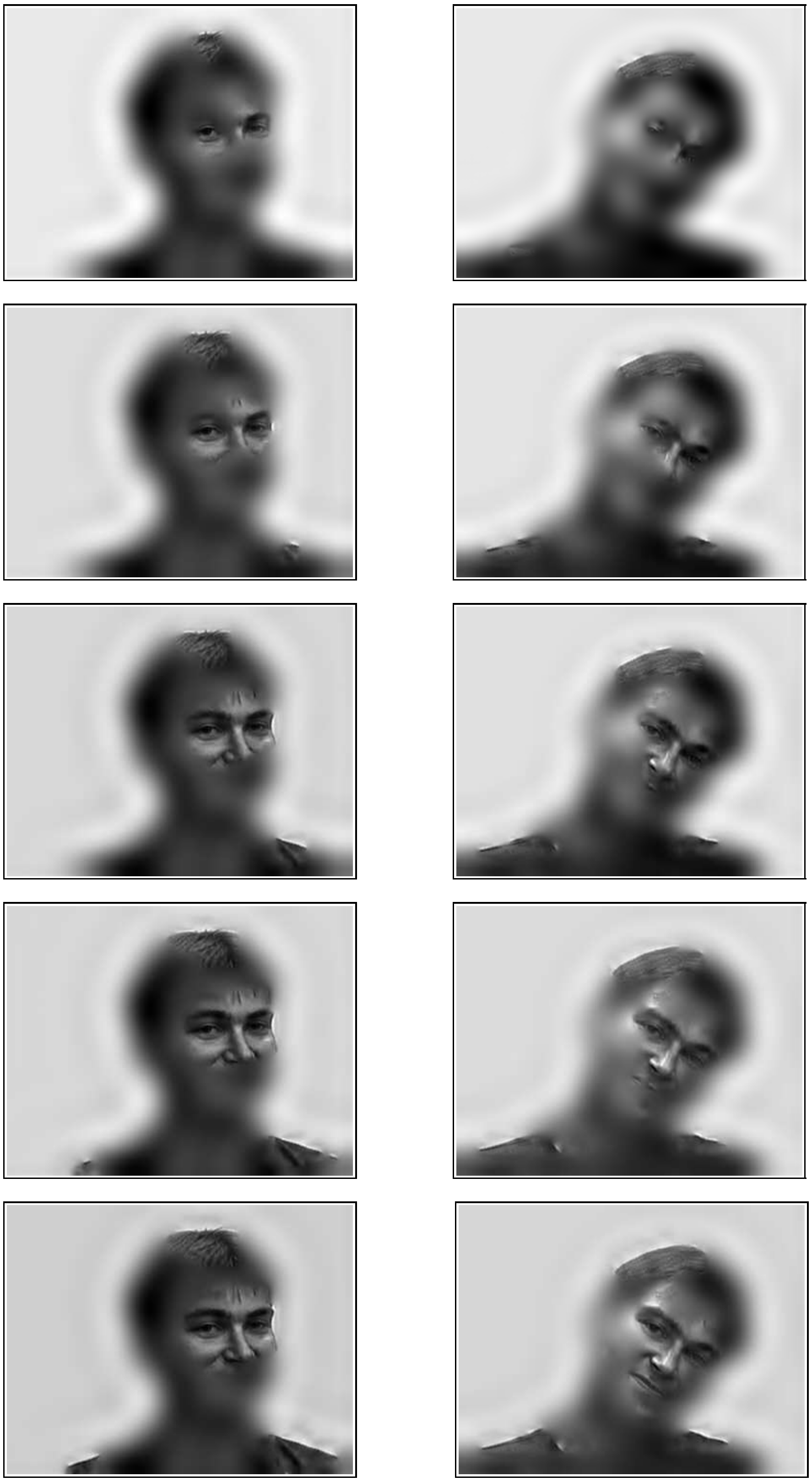


Abbildung 6.13: Balkendiagramm der für die Augenrekonstruktion benötigten Decodierungsschritte – Versuch 4 (Kopfneigung) und 5 (Skalenänderung).

Verfahren bei jedem Versuch eine ähnliche Anzahl von Decodierungsschritten benötigen, wohingegen die Rückprojektion und Bayes stark schwankende Resultate zeigen. Die gleitenden Verfahren benötigen ca. 26 Schritte für den χ^2-Test und ca. 17 Schritte für die Intersektion.

Bei den Versuchen 4a und 4b fällt zudem auf, daß die Lokalisierungeigenschaften bei der leichten Kopfneigung (Versuch 4a) schlechter sind als bei der starken Kopfneigung (Versuch 4b). Dieses Phänomen läßt sich damit erklären, daß die gewählten Versuchsbilder

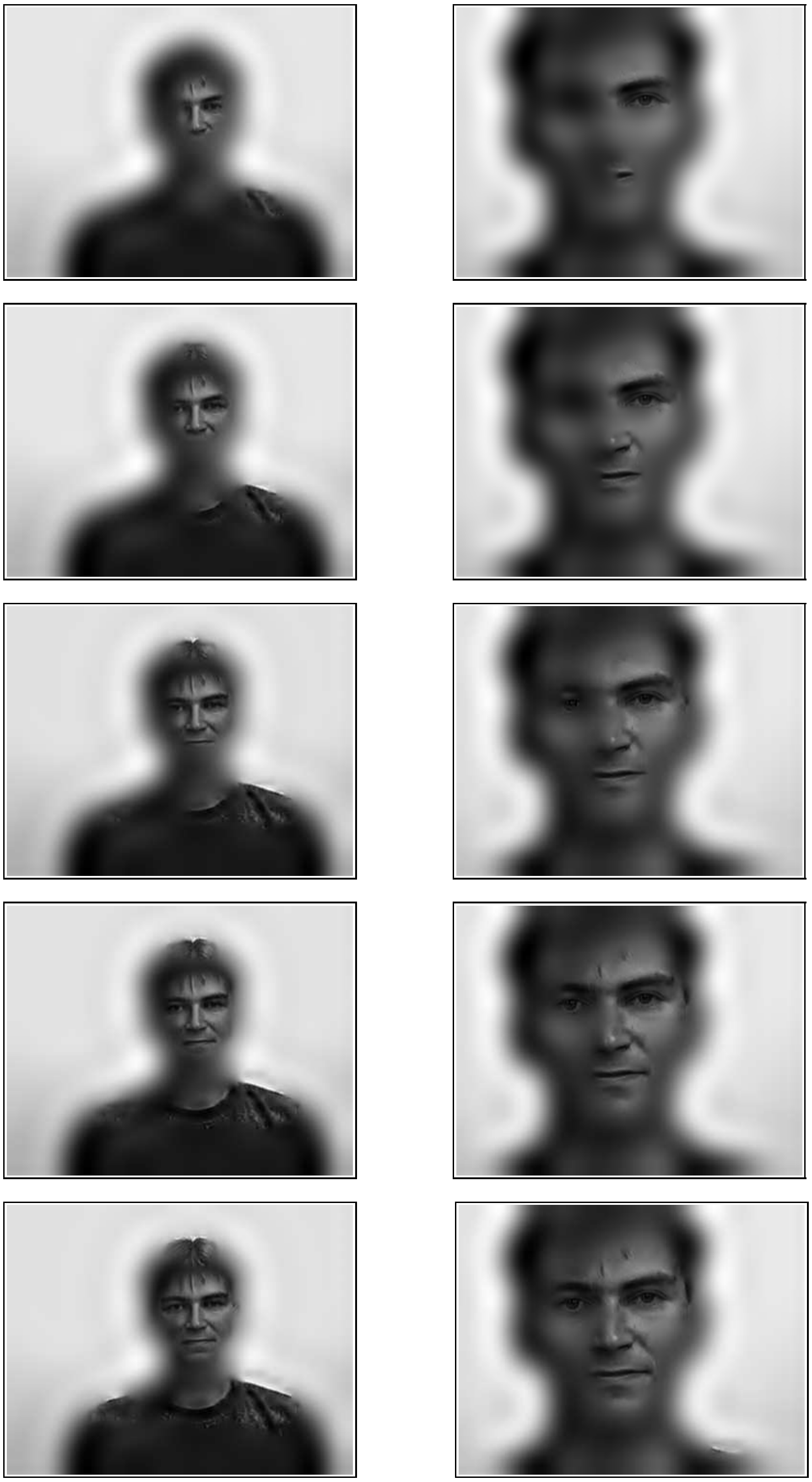
Versuch 3 – Rückprojektion **Versuch 4b – Intersektion**



4 Schritte | 4 Schritte
8 Schritte | 8 Schritte
12 Schritte | 12 Schritte
16 Schritte | 16 Schritte
20 Schritte | 20 Schritte

Abbildung 6.14: Bildfolge von Versuch 3 (Kopfdrehung): Rückprojektion (links), Bildfolge von Versuch 4b (starke Kopfneigung): Intersektion (rechts).

Versuch 5a – Intersektion **Versuch 5b – Intersektion**



4 Schritte 4 Schritte

8 Schritte 8 Schritte

12 Schritte 12 Schritte

16 Schritte 16 Schritte

20 Schritte 20 Schritte

Abbildung 6.15: Bildfolge von Versuch 5a (größere Entfernung zur Kamera): Intersektion (links), Bildfolge von Versuch 5b (geringere Entfernung zur Kamera): Intersektion (rechts).

keine perfekte Ausleuchtung aufweisen. Bei Versuch 4a liegt das rechte Auge in einem Schatten und ist somit schwerer zu lokalisieren. Diese Feststellung werden wir auch bei Versuch 6a (schlechter Kontrast) wiederholen können. In Abbildung 6.14 und 6.15 sind für die Versuche 3, 4b, 5a und 5b die Bildfolgen des jeweils besten Verfahrens abgebildet. Man sieht, daß trotz der Unterschiede zu dem gesuchten Auge bei allen Versuchen beide Augen nach wenigen Schritten vollständig decodiert sind. Vor allem die Intersektion wird nicht durch Änderungen der Augenausrichtung und der Skala beeinträchtigt, arbeitet also sehr robust. Durch einen schlechten Kontrast erhöht sich jedoch bei allen Verfahren die Anzahl der benötigten Decodierungsschritte. In Abbildung 6.16 ist das Balkendiagramm des Versuchs 6a, also einem Bild mit sehr schlechtem Kontrast, dargestellt.

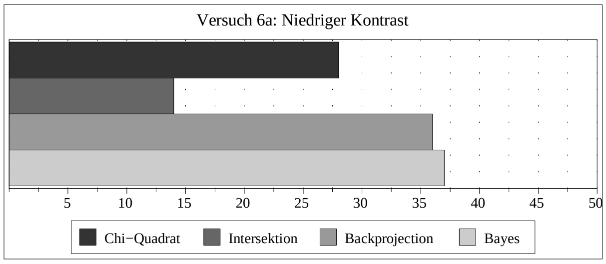


Abbildung 6.16: Balkendiagramm der für die Augenrekonstruktion benötigten Decodierungsschritte – Versuch 6a (niedriger Kontrast).

Es ist deutlich zu sehen, daß alle Verfahren relativ viele Schritte zur Decodierung der Augen benötigen. Obwohl die gewählte Darstellung en face und in der gleichen Größe wie das Referenzbild gewählt wurde, bewegen sich die Ergebnisse in ähnlichem Rahmen wie bei den Rotations- oder Skalenversuchen. Ein guter Kontrast erhöht also die Lokalisierungsleistung der Verfahren. Dies liegt daran, daß die von uns eingesetzte Quantisierung der Orientierungen nicht unabhängig von der Amplitude ist. Ist die Amplitude klein, fällt sie in die Totzone und impliziert im Orientierungsbild die Nicht-Orientierung. Zudem werden höhere Amplituden mit mehr Orientierungswerten quantisiert. Versuch 11 untersucht die Abhängigkeit der Lokalisierungseigenschaften von der Quantisierung genauer.

In Abbildung 6.17 sind die Ergebnisse der Versuche 6b, 7, 8 und 9 als Balkendiagramm dargestellt. Man sieht, daß bei Versuch 6b (überbelichtetes Bild) von allen Verfahren sehr gute Ergebnisse erzielt werden. Dies untermauert die Vermutung, daß die angewendeten Lokalisierungsverfahren einen hohen Kontrast benötigen, um die besten Leistungen zu erbringen.

Die Ergebnisse von Versuch 7 dagegen zeigen trotz der starken Schlagschatten gute Ergebnisse, die ungefähr in der gleichen Größenordnung wie die Ergebnisse des Referenzversuchs liegen. Die Verfahren arbeiten also robust gegenüber Schwankungen der Beleuchtungsqualität. Ähnliche Ergebnisse liefert auch der Versuch 9, bei dem ein Auge leicht verzerrt ist.

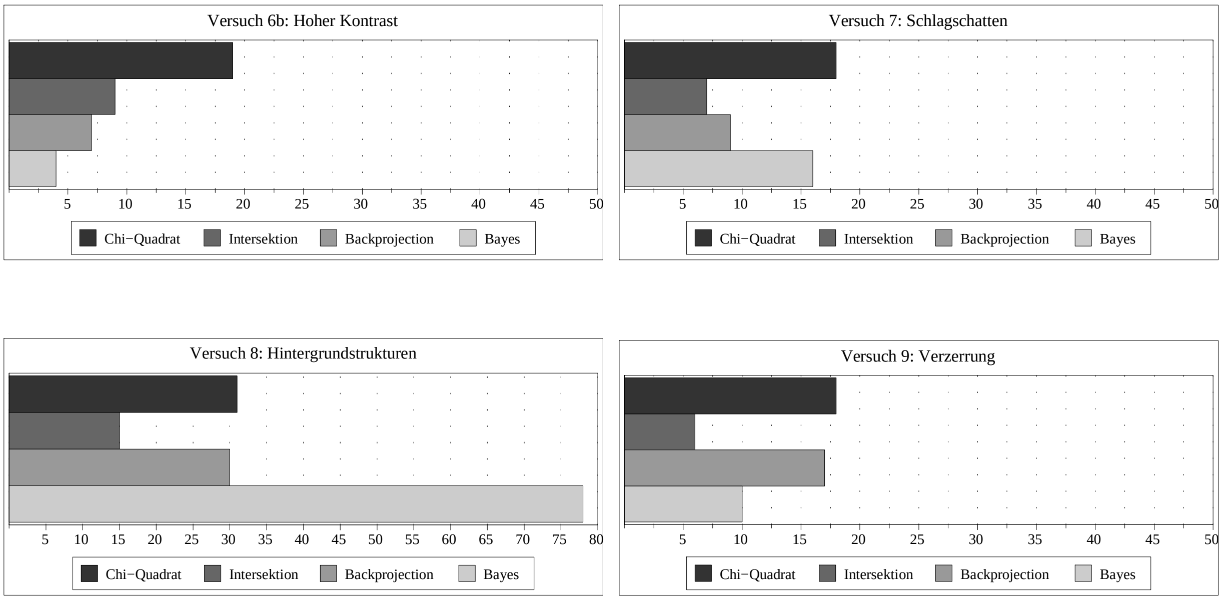


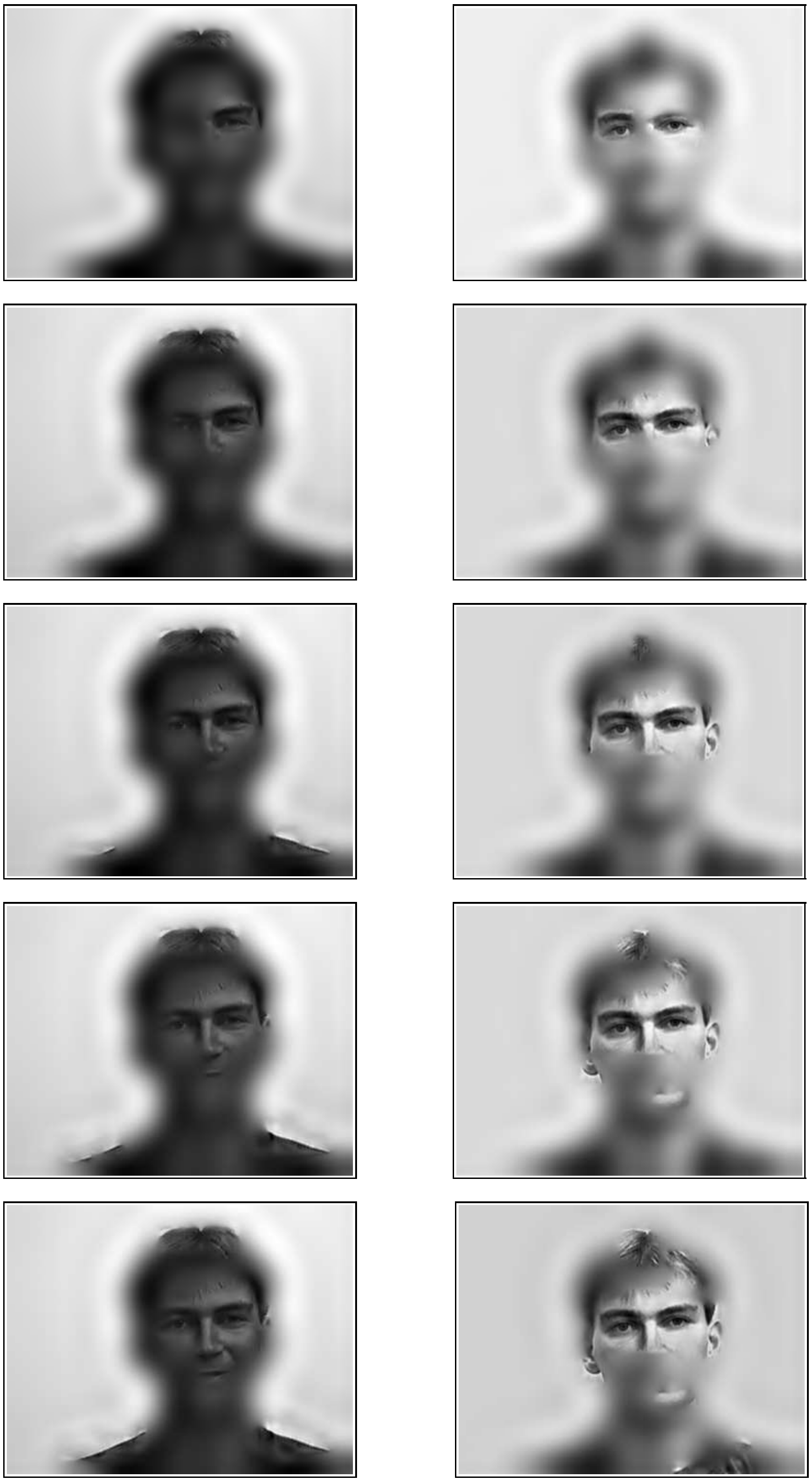
Abbildung 6.17: Balkendiagramm der für die Augenrekonstruktion benötigten Decodierungsschritte – Versuch 6b (hoher Kontrast), 7 (Schlagschatten), 8 (Hintergrundstrukturen) und 9 (Verzerrung).

Eine leichte Verformung des gesuchten Objekts führt also nicht zu einer Verschlechterung der Lokalisierungseigenschaften.

In Versuch 8 dagegen verschlechtern sich die Ergebnisse durch den grob texturierten Hintergrund deutlich. Während die gleitende Intersektion nur 15 Decodierungsschritte benötigt, also genauso viele wie bei einigen anderen Versuchen, zeigen die anderen Verfahren deutlich schlechtere Ergebnisse. Bei Bayes sind sogar nach 78 Schritten beide Augen nicht vollständig decodiert. Dieses schlechte Ergebnis läßt sich durch die Wahl des Hintergrundobjektes erklären. Bei allen Versuchen dieser Versuchsreihe haben wir das linke Auge des Referenzbildes als Modellbild benutzt. Ebenso wird für Bayes der gleichförmige Hintergrund des Referenzbildes verwendet. Da in diesem Hintergrundobjekt sehr wenige Orientierungsvektoren ungleich der Nicht-Orientierung vorhanden sind, werden die Orientierungvektoren des Modellbildes stark gewichtet. Sie stellen also ein gutes Indiz für das Vorhandensein des gesuchten Objektes dar. Da sich der Hintergund in diesem Versuch jedoch stark vom Hintergrundobjekt unterscheidet und somit im Hintergrund vorhandene Orientierungsvektoren als Merkmale für das Objekt angenommen werden, erzielen viele Punkte des Hintergrundes einen hohen Auffälligkeitswert, so daß diese vor einem Großteil der Augenpunkte übertragen werden.

Ein texturierter Hintergrund kann also, genau wie einige andere Veränderungen des Referenzbildes, die Lokalisierungsleistung verschlechtern. Während die gleitende Intersektion in diesen Fällen ca. 15 Schritte (von 158) benötigt, um das gesuchte Objekt vollständig

Versuch 6a – Intersektion **Versuch 6b – Bayes**



4 Schritte 4 Schritte

8 Schritte 8 Schritte

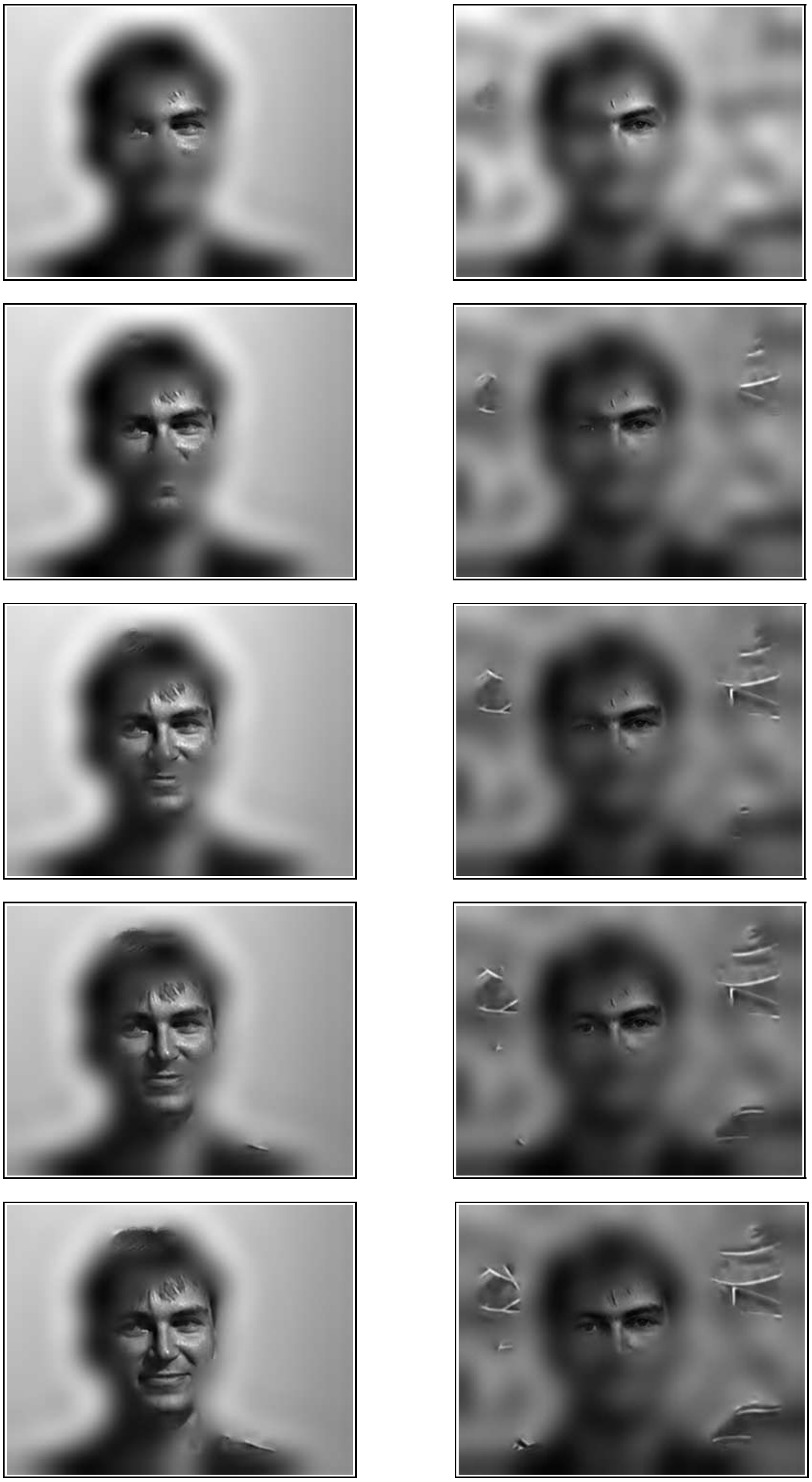
12 Schritte 12 Schritte

16 Schritte 16 Schritte

20 Schritte 20 Schritte

Abbildung 6.18: Bildfolge von Versuch 6a: Intersektion (links), Bildfolge von Versuch 6b: Bayes (rechts).

Versuch 7 – Intersektion **Versuch 8 – Intersektion**



4 Schritte 4 Schritte

8 Schritte 8 Schritte

12 Schritte 12 Schritte

16 Schritte 16 Schritte

20 Schritte 20 Schritte

Abbildung 6.19: Bildfolge von Versuch 7 (Schlagschatten): Intersektion (links), Bildfolge von Versuch 8 (Hintergrundstrukturen): Intersektion (rechts).

zu decodieren, benötigen die anderen Verfahren zum Teil deutlich länger.

In Abbildung 6.18 und 6.19 sind noch einmal die Bildfolgen der besten Verfahren von den Versuchen 6a, 6b, 7 und 8 dargestellt. Man sieht auch hier, daß die Augen in den meisten Fällen sehr schnell gefunden werden. Wie man aus den Balkendiagrammen ablesen kann, sind in einigen Versuchen andere Vergleichsverfahren fast genauso gut. Wir haben uns jedoch entschlossen, immer das beste Verfahren darzustellen, auch wenn das Verfahren in anderen Versuchen sehr schlechte Leistungen zeigt.

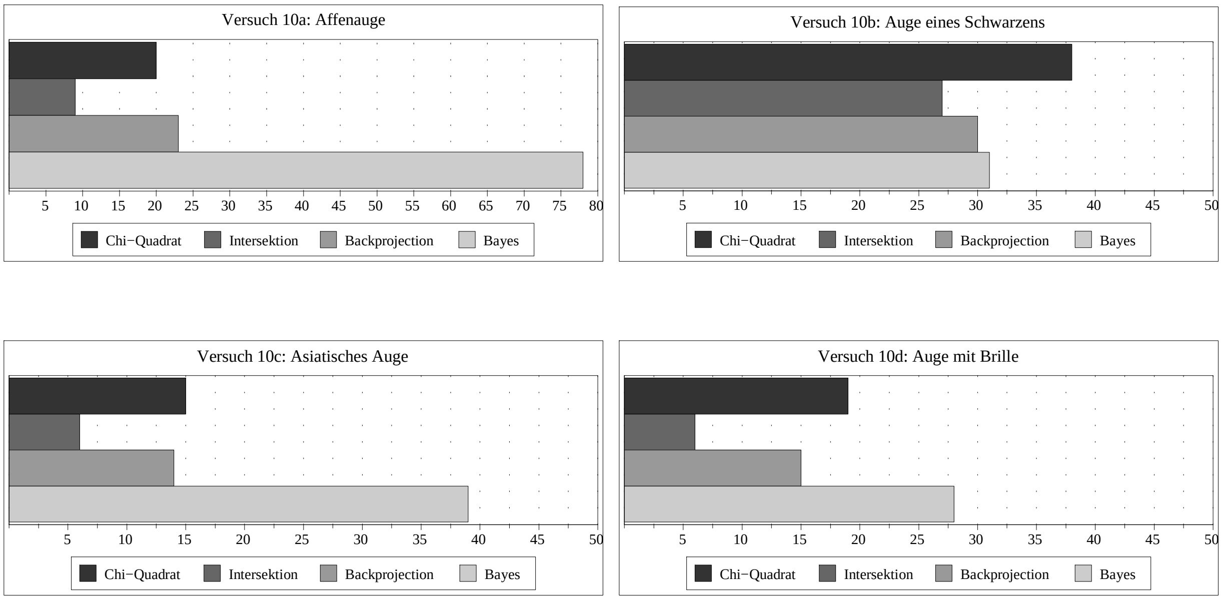


Abbildung 6.20: Balkendiagramm der für die Augenrekonstruktion benötigten Decodierungsschritte – Versuch 10a (Affenauge), 10b (Auge eines Schwarzen), 10c (Auge einer Asiatin) und 10d (Auge mit Brille).

In Abbildung 6.20 sind die Ergebnisse des Versuchs 10, also des Fremdaugenversuchs, dargestellt. Obwohl die Orientierungsverteilung in den gewählten Modellbildern von der Verteilung im Testbild abweicht, bewegen sich die Ergebnisse der gleitenden Verfahren (χ^2-Test, Intersektion) bei drei Versuchen (10a: Affenauge, 10c: Auge einer Asiatin, 10d: Auge mit Brille) in der gleichen Größenordnung wie beim Referenzversuch. Die beiden Verfahren sind also robust gegenüber Abweichungen in den Orientierungshistogrammen. Bei Rückprojektion und Bayes ist bei diesen Versuchen ein deutlicher Leistungsabfall festzustellen. Das Bayes-Verfahren zeigt bei Versuch 10a (Affenauge) die gleichen Schwächen wie bei Versuch 8 (Hintergrundstrukturen). Diese lassen sich auch hier mit dem übertragenen Hintergrundobjekt erklären. Es besteht zum Großteil aus dem Affenkopf, der fast das gesamte Bild ausfüllt. Folglich werden Merkmalsvektoren falsch gewichtet, die eigentlich zu dem gesuchten Objekt gehören.

Die Ergebnisse des Versuchs 10b (Auge eines Schwarzen) zeigen bei allen Verfahren einen starken Leistungsabfall. Dieser läßt sich damit erklären, daß die Orientierungsverteilung in diesem Modellbild extrem von der Verteilung abweicht, die durch ein mit heller Haut umgebenes Auge erzeugt wird. Trotzdem ist die Lokalisierungsleistung der gleitenden Intersektion noch akzeptabel (27 von 158 Decodierungsschritten). Bei starken Abweichungen der verglichenen Orientierungshistogramme, verschlechtern sich also auch die Ergebnisse der gleitenden Verfahren.

6.2.2 Versuchsreihe: Quantisierungvariation

In den vorherigen Versuchen hat man gesehen, daß die Lokalisierungseigenschaften der Algorithmen schlechter werden, wenn der Kontrast gering ist. Da wir bei diesen Versuchen eine Quantisierung verwendet haben, die nicht unabhängig von den Amplitudenwerten ist, könnte man hier den Grund für diesen Effekt vermuten. Deshalb haben wir fünf Versuche mit fixer, d.h. mit bis auf die Totzone von den Amplitudenwerten unabhängiger, Quantisierung durchgeführt.

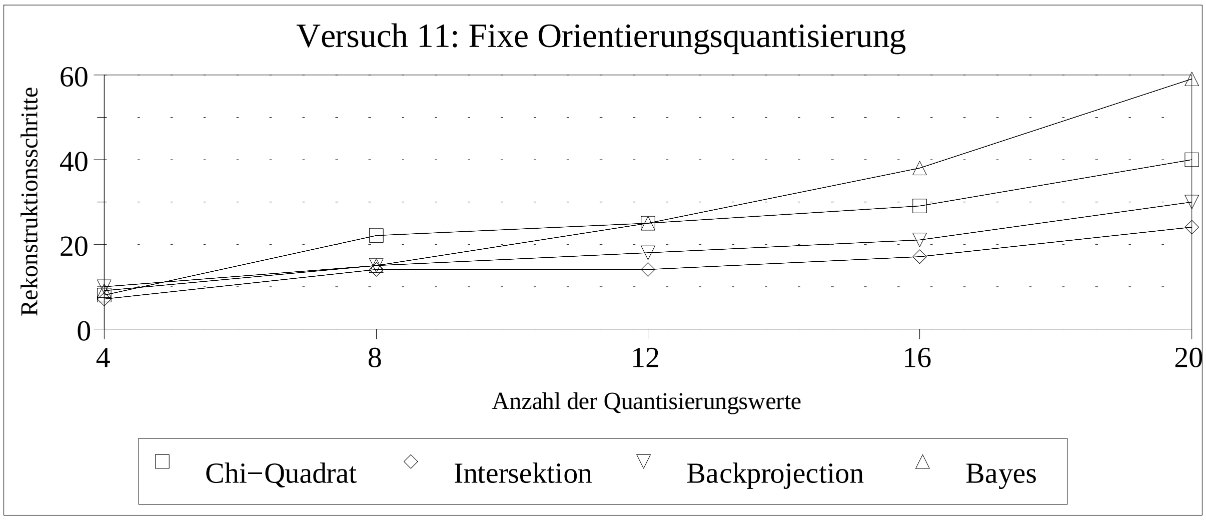


Abbildung 6.21: Kurven der benötigten Decodierungsschritte – Versuch 11 (Quantisierungsvariation). Auf der x-Achse sind die Anzahl der verwendeten Orientierungswerte und auf der y-Achse die benötigten Rekonstruktionsschritte aufgetragen.

In Abbildung 6.21 sind in einer Grafik die Kurven der Versuchsergebnisse dargestellt. Es ist festzustellen, daß bei allen Verfahren die Lokalisierungseigenschaften mit steigender Anzahl der Orientierungswerte schlechter werden. Bei vier Orientierungswerten liefert die gleitende Intersektion ähnliche Resultate wie bei der im Referenzversuch benutzten Quantisierung (sieben gegenüber acht Schritten). Die anderen drei Verfahren zeigen dagegen deutlich bessere Resultate, bleiben jedoch immer noch hinter der Intersektion zurück.

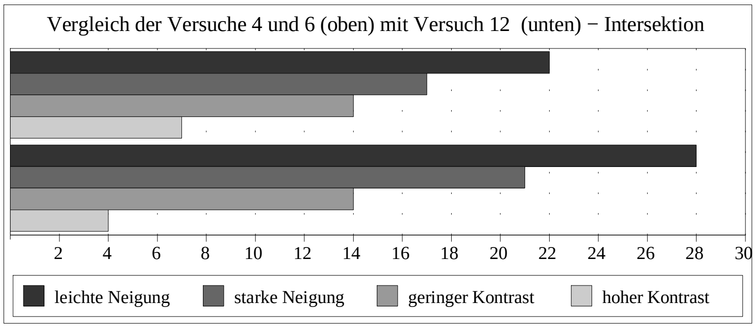


Abbildung 6.22: Balkendiagramm: Vergleich der Versuche 4 und 6 (oben) mit Versuch 12 (unten).

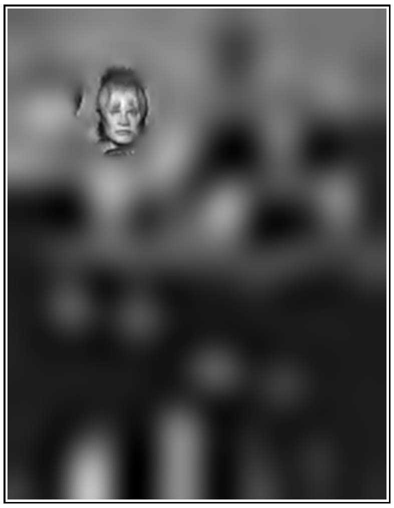
Versuche mit einigen Bildern des Robustheitsversuchs bestätigen dieses Ergebnis, so daß wir für das beste Verfahren, also die gleitende Intersektion bei fixer Quantisierung mit vier Orientierungswerten, einige Versuche des Robustheitsversuchs (Versuch 4: Kopfneigung, Versuch 6: Kontraständerung) wiederholen. Die Ergebnisse sind in Abbildung 6.22 dargestellt. Bei einem Vergleich stellt man fest, daß sich alle Ergebnisse bis auf das Ergebnis des Versuchs 6b (hoher Kontrast), das auch im ersten Versuch schon sehr gut war, verschlechtern. Auch bei Versuch 12c ist keine Verbesserung eingetreten. Die Verschlechterung der Lokalisierungseigenschaften liegt also nicht an der variablen Quantisierung, sondern eher daran, daß in diesen Fällen deutlich mehr Werte in die Totzone fallen und als Nicht-Orientierung eingestuft werden.

Die bisher angewendete Quantisierung, die auch bei der Kompression eingesetzt wird, liefert folglich die besten Ergebnisse. Bei variabler Quantisierung stellen die mit höheren Amplituden hinzukommenden Orientierungswerte also wichtige zusätzliche Merkmale dar. Bei einer Steigerung von acht auf zwölf Orientierungswerte kommen Werte hinzu, die z.B. bei einer fixen Quantisierung mit 20 Orientierungswerten nicht vorhanden sind (siehe Abschnitt 4.2.2 für zwei Quantisierungsbeispiele).

6.2.3 Versuchsreihe: Lokalisierung von Personen

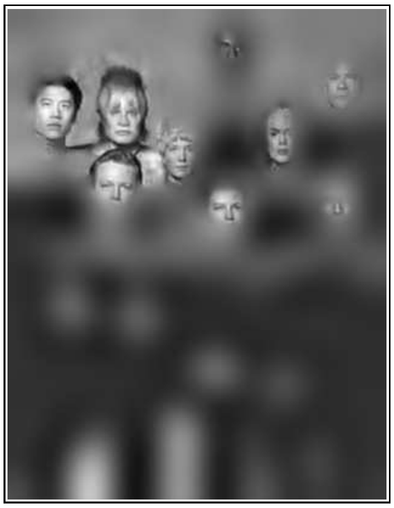
In dieser Versuchsreihe untersuchen wir die Leistungsfähigkeit der Algorithmen auf einer anderen Bildklasse: Szenen mit mehreren Personen. Wir suchen in diesen nach Gesichtern und geben als Modellbild in Versuch 13a ein Gesicht aus der Szene und in Versuch 13b ein nicht vorkommendes Gesicht ähnlicher Größe vor. In Abbildung 6.23 ist die Bildfolge

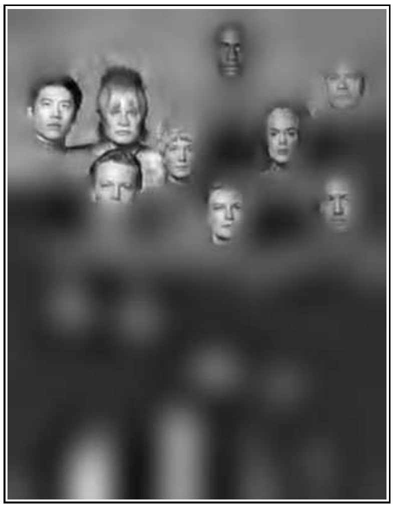
Versuch 13a – Rückprojektion

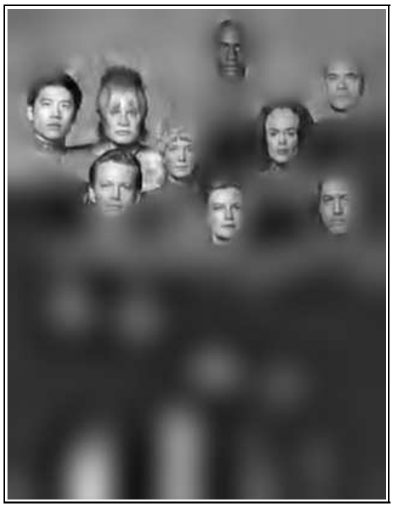

5 Schritte


10 Schritte


15 Schritte


20 Schritte


25 Schritte


30 Schritte

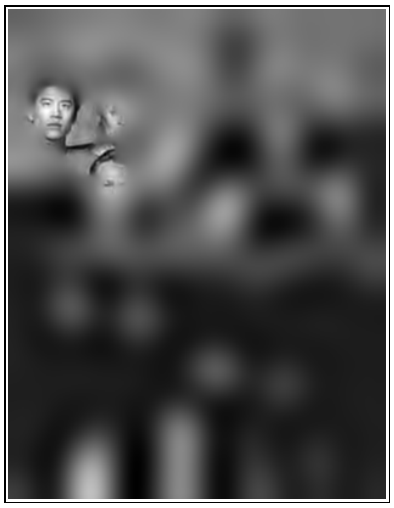

35 Schritte


40 Schritte


45 Schritte

Abbildung 6.23: Bildfolge von Versuch 13a (Lokalisierung einer im Bild vorkommenden Person): Rückprojektion.

Versuch 13b – Rückprojektion

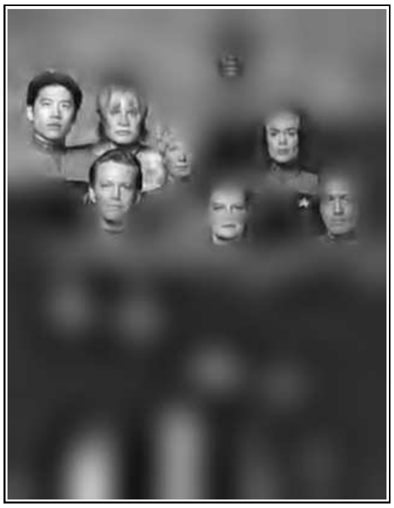


5 Schritte

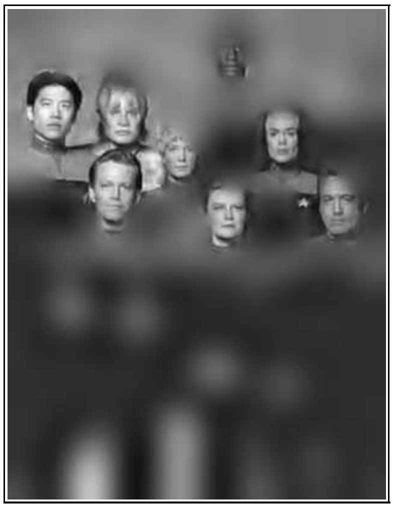
10 Schritte

15 Schritte

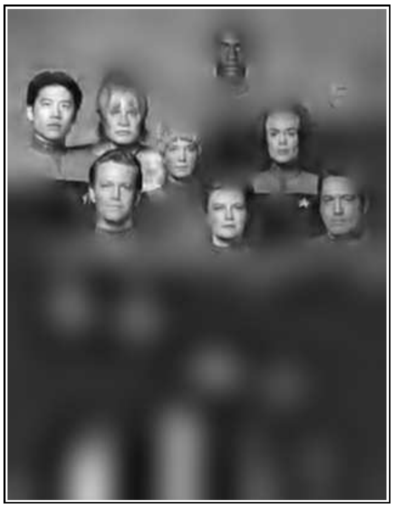
20 Schritte



25 Schritte



30 Schritte



35 Schritte

40 Schritte

45 Schritte

Abbildung 6.24: Bildfolge von Versuch 13b (Lokalisierung einer nicht im Bild vorkommenden Person): Rückprojektion.

Versuch 14 – Rückprojektion


10 Schritte


20 Schritte


30 Schritte


40 Schritte


50 Schritte


60 Schritte


70 Schritte


80 Schritte


90 Schritte

Abbildung 6.25: Bildfolge von Versuch 14 (Hintergrundlokalisierung): Rückprojektion.

der Rückprojektion dargestellt. Zwar liefert die gleitende Intersektion auch auf diesen Bildern die besten Ergebnisse, läßt die Rückprojektion jedoch nicht weit hinter sich. Da sich in dem Testbild eine Person dicht am Rand befindet, haben wir für die Darstellung die Rückprojektion gewählt, um auch diese lokalisieren zu können. Da das gewählte Testbild größer ist (312×403 Pixel), ist auch das Auffälligkeitsbild größer (39×50 Pixel), was mehr Rekonstruktionsschritte impliziert (196). Aus diesem Grund stellen wir neun Bilder mit einem Abstand von jeweils fünf Schritten dar. Man sieht, daß im Versuch 13a schon nach fünf Schritten das Modellgesicht decodiert ist. Mit steigender Schrittzahl erscheinen immer mehr, zum Teil auch sehr unterschiedliche Gesichter. Nach den dargestellten 45 Schritten sind sowohl bei Versuch 13a als auch bei Versuch 13b (siehe Abbildung 6.24) fast alle Gesichter vollständig decodiert.

Der Einsatzbereich der Verfahren ist also nicht an geometrisch einfachere Modell wie Augen gebunden. Um diese These zu untermauern, haben wir in einem weiteren Versuch nach einem Hintergrundausschnitt des Testbildes gesucht (Versuch 14). Dieses Modellbild weicht stark von den bisher benutzten ab. In Abbildung 6.25 ist eine Bildfolge aus neun Bildern zu sehen, die jeweils zehn Schritte auseinander liegen. Die höhere Schrittanzahl resultiert aus der Tatsache, daß in dem Bild deutlich mehr Hintergrund zu finden ist, als Gesichter.

Auch hier zeigt sich, daß nach 90 Schritten ein Großteil des Hintergrundes – insbesondere die gesuchte Stelle, die als erstes übertragen wird – decodiert ist, während von den Gesichtern fast keine Teile rekonstruiert wurden.

Wir können also feststellen, daß der Algorithmus auch für sehr unspezifische Modellbilder gute Ergebnisse liefert.

6.2.4 Versuchsreihe: Lokalisierung von Gegenständen

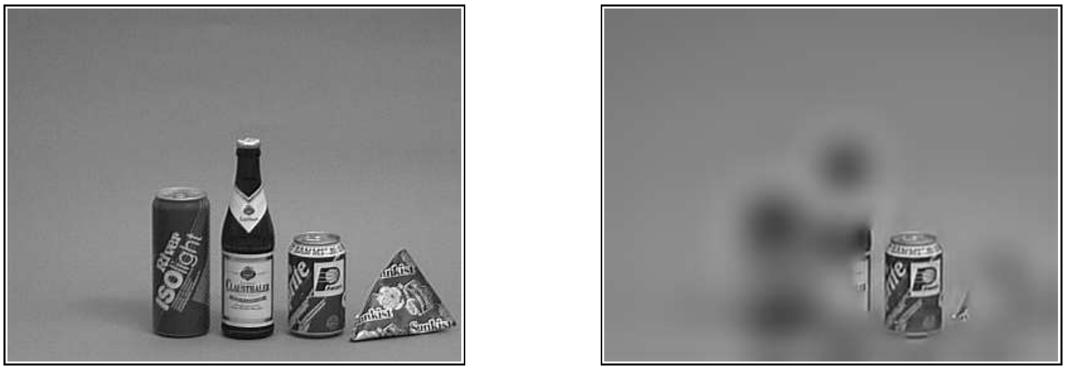


Abbildung 6.26: Versuch 15: Originalbild (links) und Rekonstruktion nach zehn Rekonstruktionsschritten (rechts).

Dieser Versuch stellt noch einmal eine andere Anwendungsmöglichkeit des Algorithmus dar. Dabei ist aus zwei sehr ähnlichen Objekten ein bekanntes Modellbild zu finden und zu rekonstruieren.

Abbildung 6.26 zeigt das Ergebnis der Übertragung nach zehn Schritten. Man sieht, daß das gesuchte Objekt fehlerfrei und sehr schnell übertragen wird, während die sehr ähnliche andere Getränkedose noch nicht zu sehen ist.

6.3 Fazit

Aus den Versuchsreihen läßt sich also schließen, daß der attentionsgesteuerte progressive Kompressionsalgorithmus gute Lokalisierungsergebnisse liefert und für fast alle Probleme die gesuchten Modellbilder zuerst übertragen hat. Bei einigen Versuchen wie z.B. dem Portraitbild mit texturiertem Hintergrund (Versuch 8) werden auch andere Strukturen als das gesuchte Objekt decodiert. Dennoch werden gleichermaßen Teile des gesuchten Objekts übertragen, die für die visuelle Erkennung des Bildinhalts ausreichen.

In annähernd allen Versuchen hat sich die gleitende Intersektion als das beste Verfahren herausgestellt. Dies bestätigt die Resultate von Schiele und Crowley [16], die auch herausfanden, daß die Intersektion für dünn besetzte Histogramme gute Ergebnisse liefert, während der χ^2-Test eher für stark besetzte Histogramme vorzuziehen ist. Des weiteren kann man sagen, daß die gleitenden Verfahren sehr robuste und stabile Ergebnisse liefern, ganz im Gegensatz zu Rückprojektion und Bayes. Diese beiden Verfahren sind zwar in zwei Versuchen besser als die Intersektion, benötigen in anderen Versuchen jedoch deutlich mehr Decodierungsschritte, um die gestellte Aufgabe zu lösen.

Alle Verfahren zeigen, auch bei einem Abweichen von der variablen Quantisierung leichte Schwächen bei schlechtem Kontrast. Der Leistungsabfall bewegt sich allerdings im Rahmen der übrigen Robustheitsuntersuchungen wie z.B. der Skalenänderung.

Der Leistungsabfall bei geringer werdender Ähnlichkeit ist jedoch ein großer Vorteil, wenn ein bekanntes Objekt in einem Bild lokalisiert und übertragen werden soll. Auch wenn sehr ähnliche Objekte im Testbild vorhanden sind, wird zuerst ein Bereich um den Punkt mit dem maximalen Auffälligkeitswert decodiert. Dies zeigt sich z.B. in den Versuchen 1 (das linke Auge wird zuerst rekonstruiert), 13a (der gesuchte Kopf wird zuerst rekonstruiert und 15 (schnelle Rekonstruktion der Dose).

Wird dagegen nur ein Modellbild vorgegeben, das einem Objekt im Bild ähnlich ist, benötigt die gleitende Intersektion etwas länger, um das Objekt vollständig zu rekonstruieren. Im allgemeinen ist die Rekonstruktion jedoch schon nach einer Übertragung von einem Bruchteil der Koeffizienten abgeschlossen.

Kapitel 7

Diskussion und Ausblick

In dieser Arbeit haben wir ein attentionsgesteuertes progressives Kompressionsverfahren vorgestellt. Wir haben die einzelnen Komponenten des Kompressionsalgorithmus erläutert und sind zunächst auf die steuerbare Pyramidentransformation eingegangen. Dabei haben wir auch einen Algorithmus zur Optimierung der benötigten orientierungsselektiven Filter angegeben. Anschließend haben wir erläutert wie die lokale Amplitude und Orientierung aus den Filterantworten der steuerbaren Pyramide berechnet werden kann. In diesem Zuge haben wir auch gezeigt, daß es legitim ist, ein Modell zu benutzen, das nur Einfachorientierungen korrekt wiedergibt. Wir haben eine Quantisierungsstufe angegeben, die den Hoch- und Tiefpaßanteil skalar und die lokalen Amplituden- und Orientierungsbilder mit einem Vektorquantisierer quantisiert. Der von uns implementierte fixe Vektorquantisierer weist eine Totzone auf und steigert die Anzahl der Orientierungswerte mit zunehmender Amplitude.

Wir beschreiben die Rücktransformation eines so zerlegten und quantisierten Bildes und untersuchen die Güte des entwickelten Kompressionsalgorithmus. Dabei haben wir die Quantisierungsparameter variiert und festgestellt, daß die lokale Orientierung deutlich mehr Information trägt als die lokale Amplitude.

In dem darauffolgenden Kapitel haben wir eine attentionsgesteuerte progressive Übertragungsstrategie vorgestellt, die anhand eines vorgegebenen Modellbildes zuerst alle Stellen des Bildes überträgt, die dem Modellbild ähnlich sind. Für diesen Vergleich werden Vektoren über den drei Orientierungsbildern der obersten Pyramidenstufen gebildet. Um für die Lokalisierung der Vektorkomponenten gleich große Orientierungsbilder zu erhalten, werden die beiden obersten Orientierungspyramidenstufen im Ort gesteuert, die fehlenden Werte also interpoliert. Die so gebildeten Vektoren werden von vier Histogrammvergleichsverfahren verwendet, um eine Auffälligkeitskarte zu erstellen. Wir haben zwei "gleitende" Verfahren vorgestellt, d.h. Verfahren, bei denen ein Fenster der Größe des Modellbildes über das zu untersuchende Bild geschoben wird und nur innerhalb dieses Fensters die Histogramme verglichen werden (χ^2-Test, Intersektion). Zum Vergleich werden zwei Verfahren untersucht, die einen Histogrammvergleich über das ganze Bild durchführen (Rückprojektion, Bayes).

In ausführlichen Versuchsreihen haben wir abschließend die Leistungsfähigkeit des von uns vorgestellten attentionsgesteuerten progressiven Übertragungsverfahrens untersucht. Dabei haben wir Versuche zur Augen-, Personen- (bzw. Gesichter-) und Objektlokalisierung durchgeführt. Es hat sich gezeigt, daß alle Verfahren sehr gute Lokalisierungseigenschaften zeigen, wenn das gesuchte Modellbild direkt aus dem zu untersuchenden Bild genommen wird. Bei Modellbildern, die von dem tatsächlichen Bildinhalt abweichen, hat sich gezeigt, daß die gleitende Intersektion die besten Leistungen liefert. Insgesamt kann man sagen, daß die Leistungen der gleitenden Verfahren sehr robust und stabil sind, während die Lokalisierungsleistung der nicht gleitenden Verfahren bei verschiedenen Versuchen stark schwankt.

Ein weiterer Vorteil der vorgestellten Coderarchitektur besteht darin, daß sowohl für die Codierung als auch für die Lokalisierung die gleiche Struktur verwendet wird.

Ausblick

Das von uns vorgestellte attentionsgesteuerte progressive Kompressionsverfahren muß an einigen Stellen noch verbessert werden. Wir geben im fogenden einen Überblick über die Probleme, die noch gelöst werden sollten:

Verbesserung der Effizienz

Die Implementierung des vorgestellten Algorithmus benötigt auf einer SPARCstation 4 der Firma SUN einige Minuten (je nach Bildgröße zwischen drei und fünf Minuten), um eine Auffälligkeitskarte zu berechnen. Diese Zeitspanne ist für eine Anwendung in einer Bilddatenbank deutlich zu lang und verhindert die angestrebte Interaktionsstrategie. Eine Verbesserung könnte durch eine Implementierung auf einem Parallelrechner oder durch eine Optimierung der verwendeten Datenstruktur erreicht werden. Die Implementierung auf einem Parallelrechner ist, angesichts der Vorarbeit, die in [13] auf dem Bildverarbeitungsrechner MV200 der Firma Datacube geleistet wurde, ein realistisches Ziel.

Verbesserung der Kompressionsleistung

Das von uns entwickelte Kompressionsverfahren erzeugt im Vergleich zu gebräuchlichen Standardverfahren eine ca. vier Mal so große Datenmenge. Die Verbesserung der Kompressionsleistung, die nicht der Schwerpunkt dieser Arbeit war, kann durch eine Kombination von weiteren Kompressionalgorithmen erzielt werden. Eine Unterabtastung der

Koeffizientenbilder um den Faktor zwei in x- und y-Richtung würde eine vierfache Reduktion der Datenmenge bedeuten, ohne, unserer Meinung nach, eine rapide Verschlechterung der Qualität zu verursachen. Weitere Algorithmen wie z.B. die prädiktive Codierung sind gut geeignet, um eine weitere Verbesserung zu erzielen.

Integration in eine Bilddatenbank

Um das vorgestellte Verfahren in einer Bilddatenbank nutzen zu können, müßte eine geeignete Stategie zur Identifikation gesuchter Bilder entwickelt werden. Es wäre denkbar, einen Ähnlichkeitsschwellwert einzuführen, ab dem ein Bild als identifiziert eingestuft wird. Dieser Schwellwert müßte sich jedoch dynamisch an die Aufgabenstellung anpassen.

Eine weitere zu untersuchende Strategie stellt die Möglichkeit dar, eine verbal gestellte Anfrage in ein Modellhistogramm umzusetzen. Dieser Ansatz birgt durch die Subjektivität der Einschätzung von Bildinhalten allerdings einige große Probleme, so daß diese Idee nur als ein Fernziel angesehen werden kann.

Anhang A

Beweise einiger Sätze

Beweis von Satz 1

Wir beginnen mit der Definition der Steuerbarkeit aus Gleichung 2.2:

$$
\begin{aligned}
\int_0^{2\pi} h(x_0 - x) f(x) dx &= \sum_{n=0}^{N-1} b_n(x_0) y[n] \\
&= \sum_{n=0}^{N-1} b_n(x_0) \int_0^{2\pi} f(x) h(n\Delta_x - x) dx \\
&= \int_0^{2\pi} f(x) \sum_{n=0}^{2\pi} b_n(x_0) h(n\Delta_x - x) dx.
\end{aligned}
$$

Damit diese Gleichung für jedes Signal $f(x)$ gilt, muß gelten:

$$
h(x_0 - x) = \sum_{n=0}^{N-1} b_n(x_0) h(n\Delta_x - x). \tag{A.1}
$$

Wir können den beliebig verschobenen Kernel $h(x - x_0)$ also als eine Linearkombination der Basis-Funktionen $h(x - n\Delta_x)$ schreiben, die jeweils einen Abtastabstand von Δ_x voneinander haben. Das bedeutet, daß die abgetastete Basismenge den gesamten Teilraum der Translationen des Kerns aufspannt. Der lineare Teilraum ist also invariant gegenüber Translationen.

Da wir annehmen, daß $f(x)$ periodisch ist, können wir o.B.d.A. annehmen, daß $h(x)$ auch periodisch ist. Sei $H(k)$ die (komplexe) Fourier-Reihe von $h(x)$. Ändert man das

Vorzeichen von x und berechnet auf beiden Seiten von Gleichung A.1 die Fourier-Reihe, erhält man:

$$H(k)e^{jkx_0} = H(k)\sum_{n=0}^{N-1} b_n(x_0)e^{jkn\Delta_x} \qquad \text{für alle } x_0, k. \tag{A.2}$$

Durch diese Gleichung ist eine Menge von linearen Bedingungen an die Interpolationsfunktionen $b_n(x_0)$ gegeben, die die Phasenveränderung jeder einzelnen Fourierkomponente für die Translation beschreibt. Existiert eine Lösung für die Gleichung, ist die Transformation steuerbar.

Die durch Gleichung A.2 gegebenen Bedingungen für die Interpolations-Funktionen gelten natürlich nur für die Frequenzen k, für die $H(k)$ ungleich null ist. Ist $H(k)$ ungleich null für eine endliche Menge $\{k_0, k_1, \ldots, k_{M-1}\}$ von Frequenzen, können die Bedingungen in Matrixschreibweise formuliert werden. Dann kann Gleichung A.2 geschrieben werden wie in Gleichung 2.3. Ist die Matrix invertierbar, können wir die Bedingung an die Interpolations-Funktionen $b_n(x_0)$ erfüllen und die Transformation ist steuerbar. □

Beweis von Satz 4

Angenommen $H(r)$ ist steuerbar. Dann gilt:

$$H(r - r_0) = \sum_n b_n(r_0)H(r - n\Delta_r).$$

Daraus folgt:

$$\begin{aligned} U_\pm(r, r_0) &= H(r - r_0) \pm H(-r - r_0) \\ &= \sum_n b_n(r_0)H(r - n\Delta_r) \pm \sum_n b_n(r_0)H(-r - n\Delta_r) \\ &= \sum_n b_n(r_0)[H(r - n\Delta_r) \pm H(-r - n\Delta_r)] \\ &= \sum_n (r_0)U_\pm(r, n\Delta_r). \end{aligned}$$

Somit ist die Behauptung bewiesen[1]. □

[1] Dabei ist zu beachten, daß Korollar 3 nur zutrifft, wenn man die symmetrischen und antisymmetrischen Basisfunktionen zum Berechen der Summe benutzt, Das heißt also, die Summe $\sum_n [U_+(r, n\Delta_r)^2 + U_-(r, n\Delta_r)^2] = 2\sum_n [H(r - n\Delta_r)^2 + H(-r - n\Delta_r)^2]$ ist konstant, wenn $H(r)$ steuerbar ist.

Literaturverzeichnis

[1] M. T. Andersson. *Controllable Multidimensional Filters and Models in Low Level Computer Vision.* PhD thesis, Linköping University, 1992.

[2] P. J. Burt and E. H. Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on Communications*, 31(4):532 – 540, 1983.

[3] Jr. F. S. Hill, Jr. S. Walker, and F. Gao. Interactive image query system using progressive transmission. In *Computer Graphics (SIGGRAPH '83 Proceedings)*, pages 323 – 330, 1983.

[4] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 13, pages 891 – 906, 1991.

[5] A. Gersho and R. M. Gray. *Vector Quantization and Signal Compression.* Kluwer Academic Publishers, 1992.

[6] B. Jähne. *Digitale Bildverarbeitung.* Springer Verlag, 1989.

[7] A. K. Jain. *Fundamentals of Digital Image Processing.* Prentice Hall, 1989.

[8] R. Jain. NSF-ARPA workshop on visual information management systems. Workshop Report, 1993.

[9] R. Jain, A. P. Pentland, and D. Petkovic. NFS-ARPA worshop on visual information management systems. Workshop Report, 1995.

[10] A. Karasaridis and E. P. Simoncelli. A filter design technique for steerable pyramid image transforms. In *ICASSP*, Atlanta, GA, May 1996.

[11] U. Mahlmeister, H. Pahl, and G. Sommer. Color-orientation indexing. In *Konferenz*, 1997.

[12] M. Michaelis. *Low Level Image Processing Using Steerable Filters.* PhD thesis, Christian-Albrechts-Universität zu Kiel, 1995.

[13] H. Pahl. Farbkonstante Objektlokalisierung und -identifikation aus lokalchromatischen Orientierungshäufigkeiten. Diplomarbeit, Christian-Albrechts-Universität zu Kiel, 1997.

[14] M. Pall. Einsatz von Subbandkodierung und Wavelets bei der Kompression von Bilddaten. Diplomarbeit, Universität Karlsruhe, 1993.

[15] R. W. Picard. Light-years from Lena: Video and image libraries of the future. Technical Report 339, MIT Media Laboratory, 1995.

[16] B. Schiele and J. L. Crowley. Object recognition using multidimensional receptive field histograms. In *Fourth European Conference on Computer Vision*, pages 610–619, April 1996.

[17] E. Simoncelli and H. Farid. Steerable wedge filters. In *Proc Int'l Conf Computer Vision*, Boston, MA, June 1995.

[18] E. Simoncelli and H. Farid. Steerable wedge filters for local orientation analysis. *IEEE Trans Image Proc*, 5(9):1377–1382, September 1996.

[19] E. P. Simoncelli. Design of multi-dimensional derivative filters. In *First Int'l Conf on Image Processing*, Austin, Texas, November 1994.

[20] E. P. Simoncelli and W. T. Freeman. The steerable pyramid: A flexible architecture for multi-scale derivative computation. In *Second Int'l Conf on Image Processing*, Washington, DC, October 1995.

[21] E. P. Simoncelli, W. T. Freeman, E. H. Adelson, and D. J. Heeger. Shiftable multiscale transforms. *IEEE Trans. Information Theory*, 38(2):587–607, March 1992. Special Issue on Wavelets.

[22] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7(1):11 – 32, 1991.

[23] A. B. Watson. Efficiency of a model human image code. *Journal of the Optical Society of America*, 4(12):2401–2417, December 1987.

[24] P. H. Westerink, D. E. Boekee, J. Biemond, and J. W. Woods. Subband coding of images using vector quantization. *IEEE Transactions on Comunication*, 36(6):713–719, 1988.

[25] J. W. Woods and S. D. O'Neil. Subband coding of images. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 34(5):1278–1288, October 1986.

[26] H. Zabrodsky and S. Peleg. Attentive transmission. *Journal of Visual Communication and Image Represantation*, 1(2):189 – 198, 1990.

Übersicht über die Versuchsreihen

1.	Lokalisierung des Referenzauges im Referenzbild a. Lokalisierung des linken Auges
2.	Lokalisierung des Referenzauges im Referenzbild ohne örtliches Steuern a. Lokalisierung des linken Auges
3.	Robustheit gegenüber Änderungen der Blickrichtung a. Drehung des Kopfes um ca. 22.5°
4.	Robustheit gegenüber Rotationen um die optische Achse a. Neigung des Kopfes um ca. 22.5° b. Neigung des Kopfes um ca. 45°
5.	Robustheit gegenüber Skalenänderungen a. Größere Entfernung zur Kamera b. Geringere Entfernung zur Kamera
6.	Robustheit gegenüber Kontraständerung a. Niedriger Kontrast b. Hoher Kontrast
7.	Robustheit gegenüber Beleuchtungsqualität a. Natürliche Beleuchtung (Schlagschatten)
8.	Robustheit gegenüber Hintergrundstrukturen a. En face Darstellung mit texturiertem Hintergrund
9.	Robustheit gegenüber Verzerrung a. Verzerrung des linken Auges
10.	Lokalisierung eines Fremdauges im Referenzbild a. Lokalisierung des Affenauges b. Lokalisierung des Auges eines schwarzen Jungen c. Lokalisierung des Auges eines asiatischen Mädchens d. Lokalisierung des Auges mit Brille
11.	Quantisierung mit fixer Anzahl von Orientierungswerten a. Quantisierung mit 4 Orientierungswerten b. Quantisierung mit 8 Orientierungswerten c. Quantisierung mit 12 Orientierungswerten d. Quantisierung mit 16 Orientierungswerten e. Quantisierung mit 20 Orientierungswerten
12.	Lokalisierung bei vier Orientierungswerten a. Neigung des Kopfes um ca. 22.5° b. Neigung des Kopfes um ca. 45° c. Niedriger Kontrast d. Hoher Kontrast
13.	Lokalisierung von Gesichtern in einer Szene mit mehreren Personen a. Lokalisierung eines in der Szene vorkommenden Gesichtes b. Lokalisierung eines nicht in der Szene vorkommenden Gesichtes
14.	Lokalisierung eines unspezifischen Objektes in einer Szene mit mehreren Personen a. Lokalisierung des Hintergrundes
15.	Lokalisierung eines Objektes in einer Szene mit ähnlichen Gegenständen a. Lokalisierung einer im Bild dargestellten Dose

Danksagung

Hiermit möchte ich Prof. Dr. Sommer für das Ermöglichen dieser Arbeit an seinem Lehrstuhl danken. Meinem Betreuer Udo Mahlmeister danke ich für seine zahlreichen Anregungen und Diskussionen, die sehr zum Gelingen der Arbeit beigetragen haben. Auch meinen Korrekturlesern Sönke, Volker und Alex möchte ich für Ihre Mühe danken. Sollten sich jetzt noch Fehler in der Arbeit befinden, so sind diese auf jeden Fall durch eilige abschließende Korrekturen entstanden.

Ganz besonders möchte ich auch allen Leuten danken, die in den Zimmern 304-306 mit Ihrer guten Laune und Fröhlichkeit eine sehr angenehme Arbeitsatmosphäre geschaffen haben.

Erklärung:

Hiermit versichere ich, daß ich die vorliegende Arbeit selbständig verfaßt und ausschließlich die angegebenen Hilfsmittel und Quellen verwendet habe.

Kiel, den