Architektur eines multimodalen Forschungssystems zur iterativen inhaltsbasierten Bildsuche

Michael Pfeiffer

Dipl.-Ing. Michael Pfeiffer
AG Angewandte Informatik
Technische Fakultät
Universität Bielefeld
E-Mail: pfeiffer@techfak.uni-bielefeld.de

Abdruck der genehmigten Dissertation zur Erlangung
des akademischen Grades Doktor-Ingenieur (Dr.-Ing.).
Der Technischen Fakultät der Universität Bielefeld
am 23. Mai 2006 vorgelegt von Michael Pfeiffer,
am 29. September 2006 verteidigt und genehmigt.

Gutachter:
Prof. Dr. Franz Kummert, Universität Bielefeld
Juniorprof. Dr. Tim Wilhelm Natkemper, Universität Bielefeld

Prüfungsausschuss:
Prof. Dr. Jürgen Lehmann, Universität Bielefeld
Prof. Dr. Franz Kummert, Universität Bielefeld
Juniorprof. Dr. Tim Wilhelm Natkemper, Universität Bielefeld
Dr. Stefan Kopp, Universität Bielefeld

Gedruckt auf alterungsbeständigem Papier °° ISO 9706

Architektur eines multimodalen Forschungssystems zur iterativen inhaltsbasierten Bildsuche

Dissertation
zur Erlangung des akademischen Grades
Doktor der Ingenieurswissenschaften (Dr.-Ing.)

vorgelegt von

Michael Pfeiffer

an der Technischen Fakultät
der Universität Bielefeld

Mai 2006

Danksagung

Einen herzlichen Dank an alle, die zum Gelingen dieser Dissertation beigetragen haben.

Besonders möchte ich hier die Leitung der Arbeitsgruppe Angewandte Informatik der Technischen Fakultät der Universität Bielefeld, Gerhard Sagerer und Franz Kummert, erwähnen, die neben der fachlichen Betreuung für ein hervorragendes persönliches Arbeitsumfeld gesorgt hat. Es liegt in diesem Verhältnis begründet, dass die vorliegende Arbeit vollendet wurde.

Die Atmosphäre, die in der Arbeitsgruppe vorherrscht, ließ ein produktives und angenehmes Arbeiten zu, dabei bleibt mir die Zeit auf V9 als besonders positiv in Erinnerung. Namentlich möchte ich hier Thomas Käster erwähnen, mit dem ich gemeinsam die Büros in der Wissensfabrik in Bielefeld gewechselt habe. Auch hier ging das Verhältnis weit über das hinaus, was man bei Bürokollegen erhoffen kann. Ich hoffe, es bleibt noch lange bestehen. Danken möchte ich auch Christian Bauckhage, Silke Fischer und Fabio Magnifico, die an der Planung, Vorbereitung, Durchführung und der Auswertung des Akzeptanztests beteiligt waren.

Nach Braunschweig geht ein Dank an meine Schwester für die Text Revision, dessen Ergebnisse in vielen Stunden am Telefon in die Niederschrift eingearbeitet wurden.

Ein besonderer Dank aber gilt meinen vier Frauen, die so viel Geduld und Enthaltsamkeit beweisen mussten, bis das Ende des Tunnels erreicht war.

Inhaltsverzeichnis

Kapitel 1

Hilfe, wir Versinken im Bildermeer!

Der kleine Eisenbahnfreund hat im Laufe der Jahre eine beachtliche Sammlung von Bildern angelegt. Die liebevoll eingerahmten DIA-Positiv-Aufnahmen befinden sich nun Film für Film in je einem staubundurchlässigen Kasten. Alle Kästen sind sorgsam gestapelt und füllen bereits den größten Teil des Regals, der ursprünglich als Platz für ganz andere Dinge geplant war.

Um den Überblick über die archivierten Aufnahmen nicht zu verlieren, hat sich unser Eisenbahnfreund teure Kontaktabzüge anfertigen lassen, die in Alben, wiederum nach Filmen sortiert, abgelegt sind. Obwohl ihm ein gutes Gedächtnis dabei hilft, viele Bilder, die zum Beispiel auf Fotopapier abgezogen werden sollen, recht schnell wiederzufinden, gibt es sicher viel mehr Bilder, von denen er nicht mehr weiß, dass diese zu seiner Sammlung gehören.

Manche mögen sagen: „Gut, aber das ist doch ein Sonderfall. Nur Leute, die so fanatisch ihr Hobby ausüben, produzieren so viele Bilder, dass es schwierig wird, diese zu organisieren!“, aber diese Aussage ist falsch.

Durch Einführung der digitalen Fotografie explodierte die Menge der privat produzierten Aufnahmen, da ein Foto augenscheinlich nichts kostet, und lediglich die Anschaffungskosten für die Kamera als Faktor wahrgenommen werden. Und so häufen sich in den Privathaushalten immer mehr Fotos, die sehr oft in Ordnern auf den Festplatten bzw. auf Archiv-Medien wie CD oder DVD organisiert werden.

Im professionellen Bereich ist die Problematik naturgemäß deutlich länger bekannt. Die Werbebranche beispielsweise benötigt Bildmaterial aus allen vorstellbaren Bereichen. Erschwerend für das notwendige Bildvolumen kommt hier hinzu, dass viele Bilder auch nicht zu alt sein dürfen, weil Szenen aus der aktuellen Zeit benötigt werden.

Medizinische Bildarchive geben ein weiteres Beispiel. Hier werden Bilder, oft Röntgenaufnahmen, Aufnahmen der Comutertomografie (CT), oder der Magnetresonanztomografie (MRT), unterschiedlichster Befunde gesammelt. Durch die Analyse der Bilder des Archivs entwickeln Forscher Verfahren, bei denen die Verwendung solcher Aufnahmen den Prozess der Diagnostik bzw. der Früherkennung von Krankheiten unterstützt.

Bilder stellen noch in vielen anderen Bereichen die Grundlage eines Arbeitsprozesses dar, bei dem für das Auffinden geeigneter Bilder große Bilddatenbestände nach unterschiedlichen Kriterien untersucht werden müssen. In solchen Fällen ist eine leistungsfähige technische Un-

terstützung dringend erforderlich, es bietet sich der Einsatz einer Bilddatenbank, die genau die geforderten Möglichkeit bietet, an.

Bilddatenbanksysteme der ersten Generation basierten auf textueller Annotation des Bildinhalts, die manuell durchgeführt werden musste. Bei diesen Systemen konnte die Technik traditioneller textbasierter Datenbankmanagementsysteme eingesetzt werden. Jedoch bildet der Vorgang der Annotation bei diesen Systemen den Schwachpunkt, wie folgende Punkte zeigen:

Es gibt keine „richtige" Menge von Annotationen für ein Bild. „Ein Bild sagt mehr als tausend Worte" heißt es, und das verdeutlicht das Problem. Die Annotation der Bilder wird im Allgemeinen problembezogen durchgeführt. Das führt jedoch zu extremer Unflexibilität, die keinerlei Änderung der Aufgabenstellung zulässt, ohne die Annotation entsprechend der neuen Anforderung zu wiederholen.

Die Annotation ist stark subjektiv geprägt. Es gibt immer Stichworte, über die gestritten werden kann, ob sie einem Bild zugeordnet werden sollen oder nicht. Damit ist es fragwürdig, ob dieselbe Person dieselbe Annotation der Bilder erstellen würde, wenn sie diese ein zweites Mal durchführen müsste. Umso mehr variieren die Annotationen unterschiedlicher Personen, da Interessen und Wissensstand verschieden sind und so unter Umständen andere Schwerpunkte im Bild gesehen werden.

Die manuelle Annotation ist sehr zeitaufwendig. Die mit diesem manuellen Vorgang verbundenen Kosten sind sehr hoch und mit wachsender Bildmenge ebenfalls stetig steigend.

Die aufgeführten Aspekte führten Anfang der neunziger Jahre zu dem Bestreben, Suchverfahren zu entwickeln, die auf den Bilddaten selber basieren, also ohne weitere manuell hinzugefügte Daten betrieben werden können. Diese inhaltsbasierte Bildsuche (engl.: *Content-Based Image Retrieval, CBIR*) setzt auf einer kompakten formalen Repräsentation des Bildinhalts auf, die automatisch extrahiert wird. Farb-, Textur- und Formmerkmale werden häufig für diese Repräsentation benutzt. Mittels Ähnlichkeitsbestimmungen und einer entsprechend angepassten Form der Datenbankanfrage, zum Beispiel durch Angabe eines Beispielbildes, konnten erste Datenbanksysteme aufgebaut werden.

Obwohl mit solchen Systemen durchaus beachtliche Suchergebnisse erlangt werden können, versagen sie genau dann, wenn sich der semantische Inhalt eines Bildes nicht durch die formale Repräsentation des Systems darstellen lässt. Man spricht hier von der semantischen Lücke (engl.: *Semantic Gap*)[Sme00]. Zum Schließen dieser Lücke kann der Datenbankbenutzer in den Suchprozess integriert werden (engl.: *User In The Loop*)[Hua02, Zho03]. In einem solchen interaktiven iterativen Suchvorgang wird vom Benutzer eine geeignete Beurteilung des Suchergebnisses verlangt.

Die Interaktion mit dem System kann in diesem Fall unterschiedlich gestaltet werden. Es existieren Systeme, bei denen die Gewichtungen der verwendeten Merkmale verändert werden können bis hin zu Systemen, bei denen eine einfache Bewertung der Bilder des Suchergebnisses vorgenommen werden muss. Auf der Basis der Bewertung, wie gut bzw. schlecht ein Bild die Suchintention widerspiegelt, wird eine Trainingsmenge erstellt, die als Grundlage eines Lernmechanismus dient, durch den das System in die Lage versetzt wird, die formale

Repräsentation der Suchanfrage der Suchintention des Benutzers anzupassen und damit das Suchergebnis entscheidend zu verbessern.

Das Ziel dieser Arbeit, die im Rahmen des LOKI[1]-Teilprojekts „Techniken zur intelligenten Navigation in digitalen Bilddatenbanken“, INDI, entstanden ist, ist der Entwurf und die globale Entwicklung des Bilddatenbanksystems als flexibles Forschungssystem. Dabei soll das System folgende Kerneigenschaften besitzen: Zum einen soll eine intelligente Navigation in den zugrunde liegenden Bilddaten möglich sein, indem das System die Suchintention des Bedieners adaptiert, zum anderen soll das System über natürliche, menschliche Kommunikationskanäle, wie Sprache und Gestik, bedienbar sein.

Wie gut die Umsetzung der natürlichen Interaktion gelungen ist und ob diese Art der Interaktion gut dafür geeignet ist, ein inhaltsbasiertes Bilddatenbanksystem zu bedienen, soll evaluiert werden.

Generell ist die Flexibilität bezüglich der Austauschbarkeit von Systemkomponenten, wie beispielsweise bildverarbeitenden Modulen, bei einer Bilddatenbank anzustreben. Das ermöglicht zum Beispiel den Einsatz des Systems für außergewöhnliches Bildmaterial. Bei der Entwicklung des hier beschriebenen Forschungprototyps gilt dies jedoch im Besonderen. Bei Teilen des Systems, die für sich einen Forschungsschwerpunkt bilden, können unterschiedliche Ansätze einfach ausgetauscht und gegenübergestellt werden.

Die vorliegende Arbeit ist wie folgt strukturiert:

In Kapitel 2 wird in die Thematik der inhaltsbasierten Bilddatenbanken anhand existierender Systeme eingeführt. Ebenso bildet die Beleuchtung unterschiedlich gearteter Systeminteraktionen einen Schwerpunkt dieses Kapitels.

In Kapitel 3 wird der Systementwurf vorgestellt. Es werden Aspekte der iterativen inhaltsbasierten Bildsuche beleuchtet, die sich auf die Architektur des Systems ausgewirkt haben. Außerdem wird die Prozessarchitektur und die zugrunde liegende Kommunikation begründet. Schließlich werden besondere Anforderungen, die durch die Integration von Sprache und Gestik auftreten, bezüglich des Einflusses auf die Systemarchitektur diskutiert.

Kapitel 4 beinhaltet die Spezifikation des Bilddatenbank-Servers. Neben der internen Prozessstruktur, die in der Lage ist, unterschiedlich geartete Clients zu bedienen, wird die Umsetzung der Modularität, die die Flexibilität des Systems ausmacht, dargestellt. Ebenso wird die integrierte adaptierfähige Sucheinheit vorgestellt. Schließlich werden besondere Dienste präsentiert, die zum Beispiel für die Verbesserung der Natürlichkeit der Systeminteraktion benötigt werden.

Das fünfte Kapitel stellt den multimodalen Datenbank-Client vor. Es werden die unterschiedlichen Interaktionskanäle, wie Mausbedienung, sprachliche Äußerungen und bildbezogene Gesten, die am Touchscreen-Monitor durchgeführt werden, aufgezeigt. Das Kapitel schließt mit der gewählten Prozessstruktur und der dafür notwendigen Ablaufsteuerung, die auf einer besonderen Kommunikationseinheit basiert. Die Ablaufsteuerung synchronisiert und fusioniert die asynchron auftretenden Interaktionen und löst entsprechende Aktionen aus.

[1] „Lernen zur Organisation komplexer Systeme der Informationsverarbeitung“, LOKI, ist ein BMB+F Verbundprojekt, mit einer dreijährigen Laufzeit.

Die für die Entwicklung benötigten Werkzeuge werden in Kapitel 6 vorgestellt. Dabei handelt es sich zum einen um einen Pre-Compiler, der die für die Kommunikation zwischen heterogenen Systemplattformen benötigte Datenrepräsentation erzeugt, und zum anderen um einen Parsergenerator für die effiziente Anbindung des verwendeten Spracherkenners.

Eine Evaluierung mittels eines Akzeptanztests schließt sich in Kapitel 7 an. Dieser Test soll zeigen, wie effektiv eine multimodale Interaktion am Beispiel einer inhaltsbasierten Bilddatenbank einzusetzen ist. Die Arbeit schließt mit einer Zusammenfassung.

Kapitel 2

Interaktion bei inhaltsbasierten Bilddatenbanksystemen

In diesem Kapitel werden die gängigen Techniken, die bei inhaltsbasierten Bilddatenbanksystemen Anwendung finden, und mögliche Formen der Systeminteraktion aufgezeigt. Der hier gegebene Überblick bildet die Grundlage für die weitere Konzipierung des Systems, bei dem die einzigartige Kombination der natürlichen Interaktion und intelligenter Bildsuche umgesetzt wurde.

2.1 Inhaltsbasierte Bildsuche

Inhaltsbasierte Bilddatenbanksysteme bestechen dadurch, dass sie ohne manuelles Zutun initialisiert und betrieben werden können. Die Basis der Suche, die Repräsentation des visuellen Inhalts, kann automatisch durch die Extraktion von Merkmalen erstellt werden.

Basierend auf einer kompakten Repräsentation können durch Anwendung mathematischer Formeln Distanzen berechnet werden, die eine Aussage über die Ähnlichkeit zweier Bilder zulässt. Es ist stark von den verwendeten Merkmalen abhängig, wie gut Bilder mit unterschiedlichem visuellen Inhalt durch die angesprochene Distanzbildung separiert bzw. ähnliche Bilder zusammen gehalten werden können. Die interne Repräsentation der Bilder wird daher meist durch die Verwendung mehrerer Merkmale verbessert.

In den meisten Fällen findet die systeminterne Darstellung der Merkmale in der mathematischen Vektorrepräsentation statt. Man spricht hier von Merkmalsvektoren (engl.: *Feature Vectors*). Die Dimension der verwendeten Merkmale ist nicht festgelegt, und deshalb können so genannte Merkmalsräume unterschiedlicher Dimension aufgespannt werden. Mit der vektoriellen Darstellung der Merkmale lässt sich zum Beispiel ein Ähnlichkeitswert aus der räumlichen Distanz, beispielsweise dem euklidischen Abstand, zweier Merkmalsvektoren herleiten. Die Menge aller für ein Bild in den verwendeten Merkmalsräumen berechneten Vektoren wird im Folgenden als Merkmalsvektorsatz bezeichnet.

Die Verwendung von Merkmalen hat direkten Einfluss auf die Konzeption der Anfrageerstellung. Wo für eine Datenbankanfrage bei einem textuell annotierten System lediglich eine An-

zahl von Bildinhaltsattributen angegeben werden muss, gilt es bei einem bildinhaltsbasierten System, einen Merkmalsvektorsatz zu finden, der dem gesuchten Bild ähnlich ist. Ein Ähnlichkeitsvergleich des gesuchten Bildes mit allen Bildern der Datenbank muss somit relativ zu den anderen Bildern einen sehr hohen Ähnlichkeitswert ergeben. Ein unerfahrener Benutzer ist nicht in der Lage, selbst einen Merkmalsvektorsatz zu erstellen und selbst für den Fachmann ist diese Aufgabe in vielen Fällen nur sehr unbefriedigend lösbar. Diese Aufgabe muss das System erledigen, indem der Merkmalsvektorsatz aus einer für den Benutzer geeigneten Anfrage ermittelt wird.

Suchanfragen

QBIC (Query By Image Content) [Fli95], eine Entwicklung von IBM, war das erste kommerziell angebotene inhaltsbasierte Bildsuchsystem und beinhaltet exakt die angeführten Techniken. QBIC unterstützt gleich mehrere unterschiedliche Formen der Datenbankanfrage. In Abbildung 2.1 werden unterschiedliche Anfrageformen dargestellt.

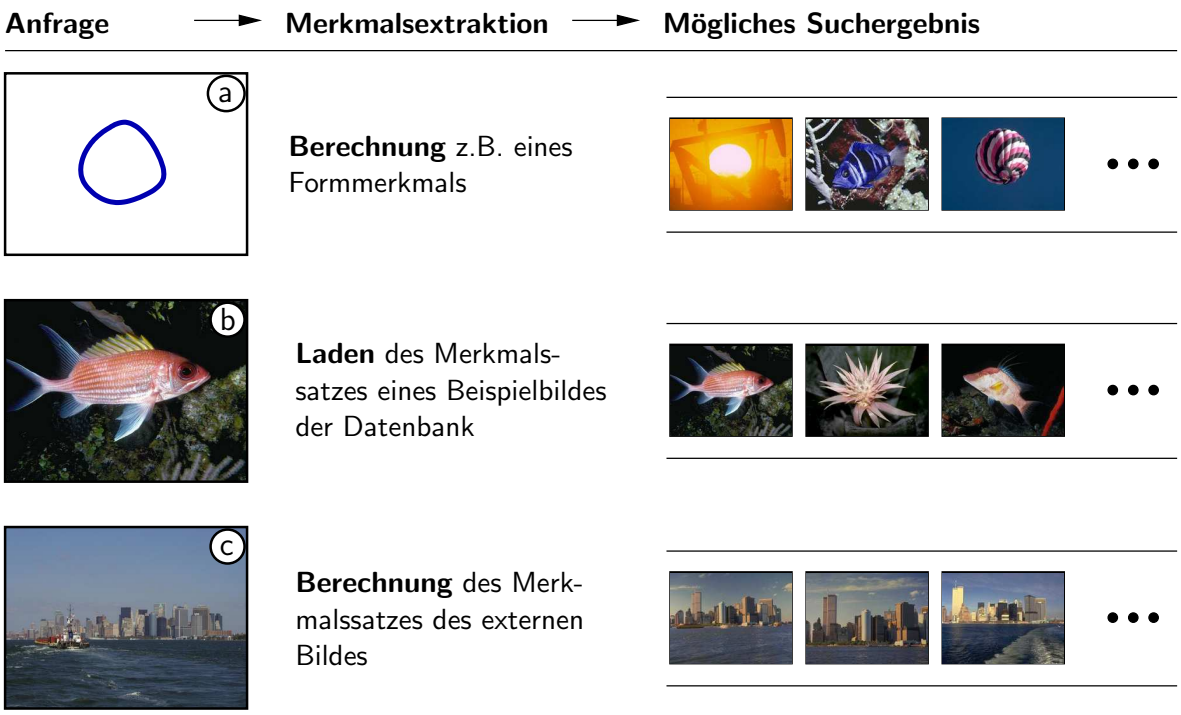


Abb. 2.1: **Formen der Anfrage bei inhaltsbasierten Systemen:** Teil (a) der Abbildung veranschaulicht den Prozess einer Anfrage mittels einer Skizze. Aus dieser werden beispielsweise Formmerkmale extrahiert, mit denen anschließend eine Suche durchgeführt wird. Suche (b) und Suche (c) werden beide auf Basis eines gewählten Beispielbildes durchgeführt. Die Suchen unterscheiden sich dahin gehend, dass im Fall (b) die Merkmalsvektoren bereits in der Datenbank gespeichert vorliegen wohingegen im Fall (c) die Berechnung der Merkmalsvektoren zu Beginn der Suche durchgeführt werden muss.

Anfragen können so geartet sein, dass der Benutzer angehalten wird, eine Skizze anzufertigen. Dabei kann die Auswertung der Skizze unterschiedlich angelegt sein. Beinhaltet das Datenbanksystem eine Objekterkennung, können wie im Fall (a) der Abbildung Formmerkmale (Abschnitt 3.2.2) zur Suche eingesetzt werden. Aus Skizzen können aber auch Layout-Informationen extrahiert werden. So kann beispielsweise ein geometrischer Aufbau mit Farbverteilung relativ einfach durch eine Skizze spezifiziert werden (siehe Abschnitt 3.2.1). Die Merkmalsberechnung muss in jedem Fall zur Laufzeit an der vorliegenden Skizze durchgeführt werden.

Augenscheinlich einfacher ist es, die Suche durch ein gewähltes Beispielbild zu spezifizieren, das dem gesuchten Bild bzw. der Suchintention recht ähnlich sein soll. Jedoch gestaltet sich die Auswahl des Beispiels nicht unproblematisch. Wenn, wie im Fall (b) der Abbildung 2.1, ein Beispiel aus der Datenbank genommen wird, muss der Benutzer zunächst einen Teil der Datenbankbilder durchsehen, um zu einem annehmbaren Bild zu gelangen. Meist wird dem Benutzer in diesem Fall eine Zufallsmenge von Bildern präsentiert. Kann wie im Fall (c) der Abbildung ein externes Beispielbild verwendet werden, setzt diese Anfrage das Vorhandensein eines solchen Bildes voraus. In diesem zweiten Fall müssen die der Suche zugrunde liegenden Merkmalsvektoren zur Laufzeit für das externe Bild berechnet werden.

Lineare Suchsysteme

Bei linearen Suchsystemen, zu denen auch QBIC zählt, gestaltet sich ein Suchvorgang wie in Abbildung 2.2 dargestellt.

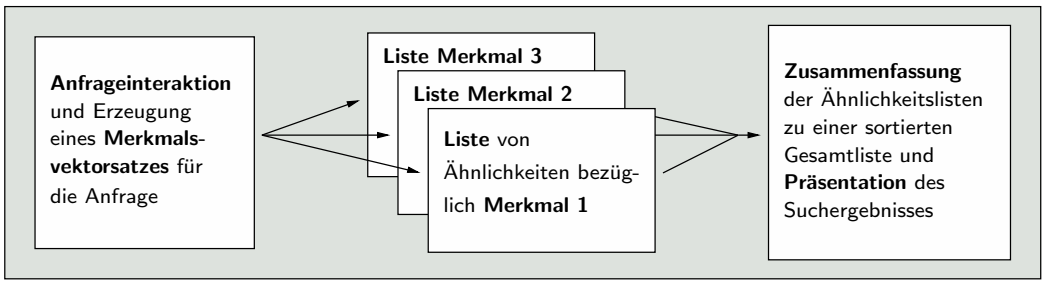


Abb. 2.2: **Ablauf der linearen Suche am Beispiel QBIC:** Zunächst wird der für die Suche notwendige Merkmalsvektorsatz durch Benutzerinteraktion erzeugt. Auf dessen Basis wird für jedes beteiligte Merkmal ein Ähnlichkeitswert für jedes Bild der Datenbank erzeugt. In der finalen Phase werden die Merkmalslisten zu einer sortierten Gesamtliste zusammengefasst und dem Benutzer in geeigneter Weise präsentiert.

Die Suche wird in drei Phasen aufgeteilt. Die erste Phase ist die der bereits angeführten Anfrageerstellung, die als Resultat einen Merkmalsvektorsatz für den Vergleich liefert. In der zweiten Phase wird für jedes Merkmal des Anfragevektorsatzes eine Ähnlichkeitsbestimmung mit dem entsprechenden Merkmalsvektor aller Datenbankbilder durchgeführt, so dass für jedes verwendete Merkmal jedem Bild der Datenbank ein Ähnlichkeitswert zugewiesen werden kann. In der anschließenden Phase werden die Ähnlichkeitswerte aller Merkmale eines Bildes in geeigneter Weise zu einem Gesamtähnlichkeitswert zusammengefasst, wobei hier die unterschiedliche

Dynamik der verwendeten Merkmale berücksichtigt werden muss (siehe Abschnitt 3.2.3, Distanzbildung und Kombination). Für die Präsentation wird in den meisten Fällen eine nach dem Gesamtähnlichkeitswert sortierte Liste gebildet. Auf der Basis dieser Liste können dann die n ähnlichsten Bilder geladen und dem Benutzer präsentiert werden.

Die Suchergebnisse sind in vielen Fällen durchaus gut. Ein Nachteil der linearen Suche ist jedoch die Tatsache, dass alle Merkmale gleichgewichtig an der Bildung des Suchergebnisses beteiligt sind. Viel sinnvoller ist es, die Merkmale unterschiedlich stark zu berücksichtigen, so dass Merkmale, die besser für die aktuelle Suche geeignet sind, entsprechend gestärkt und andere wiederum geschwächt werden. Dem System fehlen jedoch die Daten, die Gewichtung automatisch optimal festzulegen. Aus diesem Grund wird der Benutzer in den Suchvorgang eingebunden, wie im Folgenden gezeigt wird.

Iterative Suchsysteme

Iterative Suchsysteme binden den Benutzer mit in den Suchablauf ein, so dass die Suche durch entsprechende Interaktionen verfeinert und dadurch das Suchergebnis verbessert werden kann. Bei einfach gehaltenen Systemen werden beispielsweise lediglich Merkmalsgewichtungen manuell durch den Benutzer gesetzt. Um gewünschte Ergebnisse zu erhalten, setzt diese Art der Interaktion jedoch voraus, dass der Benutzer genau über die verwendeten Merkmale informiert ist und genügend Erfahrung mit dem Einfluss der unterschiedlichen Merkmale auf die Suche gesammelt hat.

Ein sehr bekanntes interaktives System ist MARS (Multimedia Analysis and Retrieval System) [Hua96], das an den Universitäten von Illinois (Urbana-Campaign) und Kalifornien (Irvine) entwickelt wurde. Dabei handelt es sich um ein sehr leistungsfähiges Query-By-Example-System, bei dem der Benutzer angehalten ist, die Bilder des Suchergebnisses entsprechend der Relevanz bezüglich der Suche zu bewerten. Mit dieser Bewertung ist das System in der Lage, seine internen Parameter selbständig so zu adaptieren, dass das Suchergebnis besser der Suchintention des Benutzers entspricht.

Abbildung 2.3 zeigt den iterativen Suchablauf, wie er im MARS-System integriert ist. Die vom Benutzer abgegebenen Relevanzbewertungen werden auf verschiedene Weise vom System verwertet. Hier ist zunächst die Bildung des Anfragevektorsatzes zu erwähnen. Im initialen Suchschritt entspricht der Anfragesatz dem Merkmalsvektorsatz des gewählten Anfragebildes. In den folgenden Schritten werden die Vektoren des Anfragesatzes durch die bewerteten Bilder zum Beispiel zu den Zentrumsvektoren der positiv bewerteten Bilder verschoben (siehe Abschnitt 4.6.1). Weiteren Einfluss nehmen die Bewertungen bei dem Ähnlichkeitsvergleich der Vektoren. Hier kann beispielsweise eine Analyse der Komponenten der Vektoren aller positiv bewerteten Bilder vorgenommen werden. Komponenten, die eine niedrige Varianz aufweisen, werden hervorgehoben, weil sie offenbar eine gesuchte Gemeinsamkeit repräsentieren. Komponenten mit hoher Varianz werden entsprechend abgeschwächt. Schließlich werden die Bewertungen bei der Zusammenfassung des Gesamtähnlichkeitswertes eingesetzt. Ganz generell sollen die Merkmale verstärkt werden, bei denen sich die Abstandswerte der bewerteten Bilder gut mit den abgegebenen Bewertungen vereinbaren lassen.

Ein weiteres iteratives System mit einem völlig anderen Ansatz ist das PicSOM-System [Laa00, Laa01], das an der Universität Helsinki entwickelt wurde. Dieses System basiert auf selbst

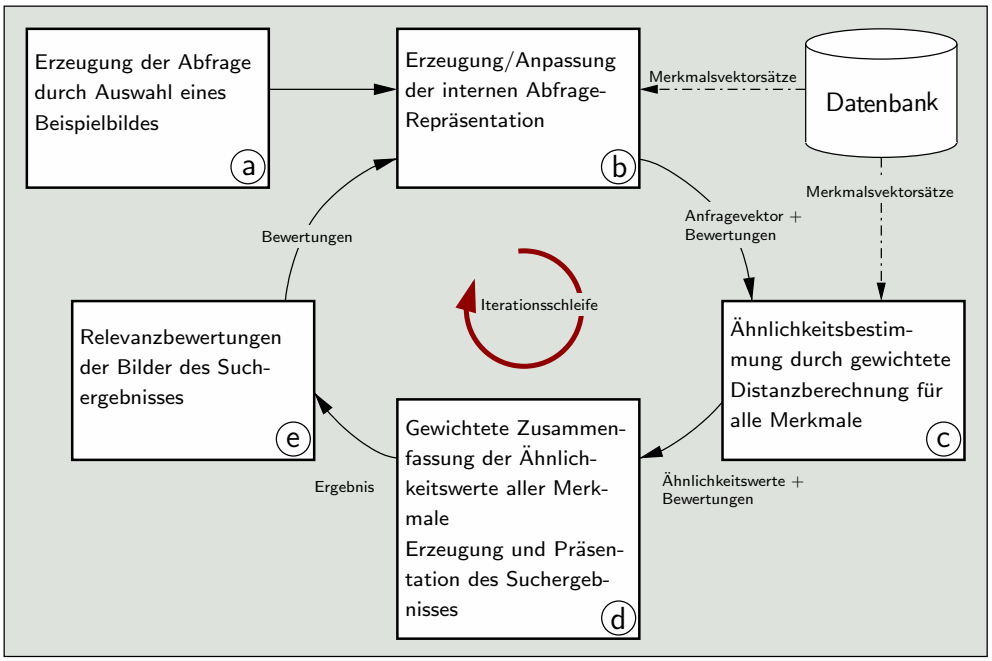


Abb. 2.3: **Ablauf der iterativen Suche am Beispiel MARS:** Gestartet wird die Suche durch Auswahl eines Anfragebildes (a). Aus dem Anfragebild bzw. einigen bewerteten Bildern wird ein Merkmalsvektorsatz für den Vergleich gebildet (b), wobei eine Datenbank die zugrunde liegenden gespeicherten Merkmalsvektorsätze liefert. Im Teil (c) der Abbildung wird ein Ähnlichkeitsvergleich mit allen in der Datenbank gespeicherten Merkmalsvektorsätzen durchgeführt, wobei eine Gewichtung, berechnet aus den abgegebenen Bewertungen (ab Iteration 2), in den Vergleich einfließt. Anschließend (d) werden die Ergebnisse des Ähnlichkeitsvergleichs der unterschiedlichen Merkmale für jedes Bild zu einem Gesamtwert zusammengefasst, wobei auch hier die zuvor durchgeführten Bewertungen des Benutzers in eine Gewichtung der verwendeten Merkmale einfließen. Es resultiert eine sortierte Liste der Bilder der Datenbank, die dem Benutzer präsentiert wird. Der Benutzer kann anschließend im Teil (e) eine Relevanzbewertung durchführen.

organisierenden Karten (engl.: *Self Organizing Maps, SOM*), die sich bereits beim Einsatz in Textsuchsystemen bewährt hatten. Eine SOM organisiert die beinhalteten Elemente als n-dimensionales Gitter, wobei ähnliche Elemente, die die Knoten des Gitters bilden, benachbart angelegt werden. PicSOM benutzt zweidimensionale Gitter und hält für jedes verwendete Merkmal eine Kaskade von SOMs (engl.: *Tree Structured SOM*, TS-SOM). Alle verwendeten TS-SOMs müssen vor der Inbetriebnahme der Datenbank erstellt werden.

Die Interaktion in PicSOM erfolgt ähnlich wie im MARS-System mittels Vergabe einer Relevanz-Bewertung durch den Benutzer. Dabei können zunächst positive Relevanzen in einer initial präsentierten Bildmenge zugewiesen werden. Diese Bewertungen werden als positive Impulse auf die Knoten in den Karten der verwendeten TS-SOMs eingetragen. Alle nicht positiv bewerteten präsentierten Bilder werden mit einem negativen Impuls versehen. Durch Anwendung eines Tiefpassfilters werden Kartenbereiche verstärkt, in denen eine Häufung von positiv bewerteten Knoten zu verzeichnen ist, und entsprechend die Bereiche geschwächt, in denen viele negative Beispiele liegen. In den einzelnen TS-SOMs werden nun Kandidatenlisten für

die Benutzerpräsentation erstellt. Ein Bild ist dann ein Kandidat, wenn es in einem positiven Bereich liegt und dem Benutzer bisher nicht präsentiert wurde. Die Kandidatenlisten der beteiligten TS-SOMs werden schließlich zu einer einzigen Liste zusammengefasst und dem Benutzer vorgestellt. In der jetzt neu beginnenden Iteration können wiederum positive Relevanzen vergeben bzw. entzogen werden.

2.2 Systeminteraktion

Die voranschreitende Leistungsfähigkeit und Miniaturisierung moderner Rechnersysteme ermöglichen einen immer weiter reichenden Einsatz, und es erschließen sich daraus fortwährend neue Anwendungsgebiete. Ein Nebeneffekt dieser Entwicklung, bezogen auf interaktive Systeme, ist: „Es gibt immer mehr unbedarfte Benutzer, die mit Systemen dieser Art in Kontakt treten!“ Die Entwicklung einer interaktiven Applikation birgt damit auch neben den bekannten technischen zu Hürden die Herausforderung, die Interaktion mit dem zukünftigen Benutzer so zu gestalten, dass die Applikation einfach, natürlich und dadurch intuitiv zu bedienen ist.

Die Forschung mit dem Ziel, die Interaktionen der Systeme optimal zu gestalten, die Mensch-Maschine-Kommunikation (engl.: *Human Computer Interaction, HCI*), wird bereits seit Jahrzehnten betrieben, sie tritt aber im Zuge der Eroberung des Massenartikelmarkts immer mehr in den Vordergrund. So ist es beispielsweise für Hersteller oben angegebener Artikel nicht tragbar, dass sich nach der Markteinführung herausstellt, dass aufgrund von schlecht gearteter Interaktion ein Artikel vom Kunden nicht wie erwartet angenommen wird.

Für die Interaktion werden zur Zeit unter anderem folgende Kommunikationskanäle verwendet:

Standardkanäle: Als Kanäle des Informationsaustauschs werden im Allgemeinen die konventionellen Eingabegeräte, Tastatur und Maus, und als Ausgabegerät ein Monitor benutzt. Bei solchen Systemen beschränkt sich die Gestaltung der Interaktion auf die Konstruktion einer so genannten Windows-Icons-Menues-Pointers-Oberfläche (WIMP)[Ovi99], die also aus Standardelementen einer aktuellen Bedienoberfläche besteht. Einzig die Anordnung und die Benutzung der verschiedenen Element-Typen wie beispielsweise Schaltflächen und Textfelder ist hier der Bestandteil der Gestaltung.

Spezielle Hardware: Handelt es sich bei dem System nicht um ein Rechnersystem im klassischen Sinne sondern um ein eingebettetes System (engl.: *Embedded System*), beispielsweise um ein Mobiltelefon, dann besitzt das Gerät meist eine Tastatur, die der Geräteform angepasst ist. Mit dieser Tastatur besteht die Selektionsmöglichkeit von Funktionen aus Menüs, die auf einer kleinen Anzeige dargestellt werden. Zu dieser Kategorie gehören auch moderne digitale Fotoapparate. In beiden Kategorien etablieren sich immer wieder Trends wie die besondere Anordnung von Tasten und neuartige Interaktionsmöglichkeiten wie Drehräder (engl.: *Jog Dial*). Diese Neuheiten sollen eine einfache und intuitive Arbeit mit dem Gerät/System ermöglichen, wie beispielsweise das Drehrad, das für das sich schnell und häufig wiederholende Ausführen ein und derselben Funktion, zum Beispiel das Blättern, eingesetzt wird.

Sprachausgabe: Sprache wird für die Systeminteraktion in beide Richtungen zwischen den Kommunikationspartnern benutzt. Die Sprachausgabe kann sowohl durch eine Sprachsynthese als auch durch das Wiedergeben aufgezeichneter Äußerungen durchgeführt werden. Die uneingeschränkte Menge der Äußerungen bei der Sprachsynthese stehen dem natürlichen Klang der aufgezeichneten Sprachausgabe gegenüber.

Sprachausgabe lässt sich vor allem dann sinnvoll einsetzen, wenn der Benutzer nicht gezwungen werden soll, den Blick zu verändern, um eine Systemantwort aufzunehmen. Ein Beispiel für diesen Einsatz ist die Navigationshilfe im Fahrzeug. Alle Informationen werden nicht allein auf einem Display ausgegeben, sondern zusätzlich durch eine Sprachausgabe dem Fahrer mitgeteilt.

Dieses Beispiel veranschaulicht den weiteren Vorteil der Sprachausgabe, nämlich dass durch die Übermittlung der Informationen durch ein akustisches Signal die Aufmerksamkeit des Bedieners, hier also die des Fahrers, sicher und zum rechten Zeitpunkt wieder auf das System gelenkt wird. Ein akustisches Signal ist in diesem Fall besonders gut geeignet, denn es wird unabhängig von der Kopfstellung und Blickrichtung immer gut aufgenommen.

Generell ist die Sprache ein natürlicher Kommunikationskanal, der sich dann besonders gut einsetzen lässt, wenn dem Benutzer zum Beispiel ein interner Systemzustand mitgeteilt werden soll. Etwas umfangreichere Informationen lassen sich in solchen Fällen ohne Verwendung eines entsprechenden Displays optisch zum Beispiel durch Verwendung eines Piktogramms nur umständlich darstellen. Gleichzeitig wirft diese Art der Interaktion aber die Forderung auf, Nachfragemöglichkeiten des Benutzers zuzulassen, die bei Vergessen oder bei akustischem Nichtverstehen angewendet werden können.

Spracheingabe: Die Spracheingabe, bei der also die Interaktion aus gesprochenen Äußerungen bzw. Anweisungen des Benutzers besteht, ist eine technisch sehr anspruchsvolle Anforderung an das verarbeitende System. Ein Spracherkennungssystem wird grundsätzlich auf das zu lösende Problem zugeschnitten. Dabei wird der Wortschatz, das Lexikon, möglichst klein gehalten, um mit vertretbaren Mengen von Trainingsmaterial hohe Erkennungsraten zu erzielen. Aus diesem Grund ist die sprachliche Interaktion meist deutlich eingeengt, und sprachliche Äußerungen können somit nicht frei, also natürlich, durchgeführt werden. Trotz der angesprochenen Nachteile ist die Spracheingabe aber dann ideal einzusetzen, wenn der Benutzer ohne besondere technische Hilfsmittel und ohne an eine bestimme Position gebunden zu sein interagieren möchte [Coe98]. In solchen intelligenten Umgebungen (engl.: *Intelligent Environments*) können mehrere Benutzer gleichberechtigt und auf natürliche Art und Weise mit dem System interagieren, ohne dass aktiv ein Eingabegerät weitergereicht werden muss.

Gesten: Gesten können wie Sprache sowohl zur Ein- als auch zur Systemausgabe benutzt werden. Da abgesehen von humanoiden Robotern von den wenigsten System Gesten zur Ausgabe erwartet werden, wird hier die weitere Betrachtung lediglich auf Gesten zur Systemeingabe beschränkt.

In der visuellen Gestenerkennung ist im Allgemeinen die Erkennung von Handgesten, zum Beispiel einer Zeigegeste, gemeint. Zur Detektion solcher Gesten werden Kameras

eingesetzt, deren Datenströme in Echtzeit ausgewertet werden müssen. Für die Mensch-Maschine-Interaktion sind oft nur Zeigegesten von Bedeutung, aber gerade im Hinblick auf eine Interaktion zur Anfrageerstellung einer Bilddatenbank wären zum Beispiel formbeschreibende Gesten eine sehr wünschenswerte Interaktionsmöglichkeit. Ähnlich wie bei der Spracherkennung sind jedoch die Hürden, die für eine robuste Gestenerkennung genommen werden müssen, sehr hoch. Hier sind beispielsweise unterschiedliche und dynamische Beleuchtungsverhältnisse als Grund zu nennen. Sehr vielversprechend sind Ansätze, bei denen neben den reinen Videodaten Informationen weiterer Eingabemodalitäten für die Erkennung herangezogen werden. So entsteht ein erweiterter Kontext, der eine robustere Erkennung zulässt, weil viele Hypothesen ausschließbar werden.

Der Einsatz unkonventioneller Kanäle zur Systeminteraktion, wie die Benutzung von Sprache oder Gestik, muss sehr gut geplant werden [Coe98]. Spezialisten sind sich den Anforderungen, die solche Techniken an das System stellen, durchaus bewusst und verzeihen Erkennungsfehler. Anders geartet ist dies bei unbedarften Benutzern, sie verlieren schnell die Geduld, zum Beispiel Anweisungen zu wiederholen oder falsch interpretierte Anweisungen rückgängig zu machen. Auf der anderen Seite bieten, wie Oviatt in [Ovi99] feststellt, multimodale Systeme durch die Kombination der Modalitäten bei bestimmten Interaktionen eine deutliche Vereinfachung.

Systeme, bei denen sich negative Erfahrungen häufen, werden schnell als unbrauchbar abgetan und nicht mehr benutzt. Daher sollten bei der Systemkonzeption und der späteren Entwicklung unkonventionelle Kanäle sehr bedacht eingesetzt werden und die Interaktionsmöglichkeiten durch Benutzbarkeitstests mit Versuchspersonen überprüft werden.

Die in diesem Kapitel vorgestellten Techniken werden im folgenden Kapitel aufgegriffen und bezüglich deren Realisierung und Kombination analysiert.

Kapitel 3

Konzeption des Bilddatenbanksystems INDI

In diesem Kapitel wird anhand der grob definierten Systemanforderungen ein Konzept erarbeitet, das als Grundlage für die in den folgenden Kapiteln umgesetzte Implementierung dient. Dabei wird der Einsatz bereits angeführter Techniken hinsichtlich der gegebenen Anforderung abgewogen.

3.1 Systemanforderungen

Das in dieser Arbeit vorgestellte Bildsuchsystem INDI kombiniert moderne Techniken intelligenter inhaltsbasierter Bildsuche mit einer natürlich gestalteten Interaktion mittels Sprache und Gesten, die an einem Touchscreen-Display durchgeführt werden (siehe Abbildung 3.1). Durch diese Kombination unterscheidet sich das System von allen anderen zur Zeit existierenden Bilddatenbanksystemen.

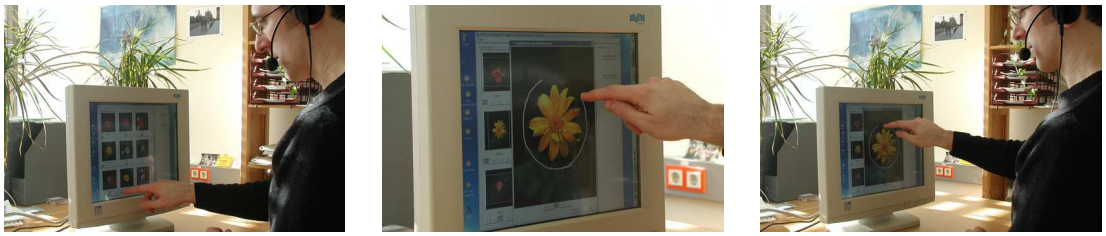


Abb. 3.1: **Natürliche Interaktion mit INDI:** Der Benutzer hat die Möglichkeit, das Bildsuchsystem mittels Gesten am Touchscreen und Sprache zu bedienen.

Das hier vorgestellte System ist im Rahmen des Teilprojekts „Techniken zur intelligenten Navigation in digitalen Bilddatenbanken“, das dem Verbundprojekt des BMB+F „Lernen zur Organisation komplexer Systeme der Informationsverarbeitung“ untergeordnet war, entstanden. Aus dem Titel des Teilprojekts wurde das Akronym INDI gebildet, mit dem im Folgenden der Arbeit das Bilddatenbanksystem bezeichnet wird.

Die Motivation für das Projekt war, einen natürlichen Zugang zu einem Bildsuchsystem zu schaffen, indem unterschiedliche Lerntechniken in einer natürlich gearteten Interaktion und in intelligenten Suchtechniken eingesetzt werden.

Im Bereich der Bildsucheinheit soll durch eingesetztes Systemlernen in einem iterativen Vorgang eine Adaption der internen Parameter erfolgen, so dass die Suchintention des Benutzers besser getroffen wird und damit eine Verbesserung des Suchergebnisses zu verzeichnen ist. Ebenso sollen lernende Techniken bei der Benutzerinteraktion eingesetzt werden. Hier zu erwähnen sind sowohl Erkenner, die die unterschiedlichen Interaktionsmodalitäten verarbeiten, als auch Techniken zum Referenzieren von Bildregionen. Neben der Verwendung globaler Bildinformationen, die sich auf das Gesamtbild beziehen, sollen ebenso lokale Informationen gewonnen werden. Das motiviert sich aus der Tatsache, dass sich eine Suche oft auf ein im Bild enthaltenes Objekt bezieht. Durch die Bestimmung von inhaltlich zusammenhängenden Bildregionen soll diesem Benutzerverhalten entgegengekommen werden.

Durch die Kombination der angegebenen Punkte soll auch solchen Benutzern der Zugang zum System ermöglicht werden, die wenig Vorkenntnisse mit Rechensystemen und deren Benutzung besitzen. Ebenso sollen Benutzer zum Sucherfolg gelangen, die keine Erfahrung mit Bildverarbeitung gesammelt haben, so dass technische Details nie Gegenstand der Interaktion werden dürfen. Durch die natürliche Interaktion soll auch ein gemeinschaftliches Arbeiten mehrerer Benutzer mit der Datenbank unterstützt werden, ohne dass dabei besondere technische Hilfsmittel zur Interaktion übergeben werden müssen.

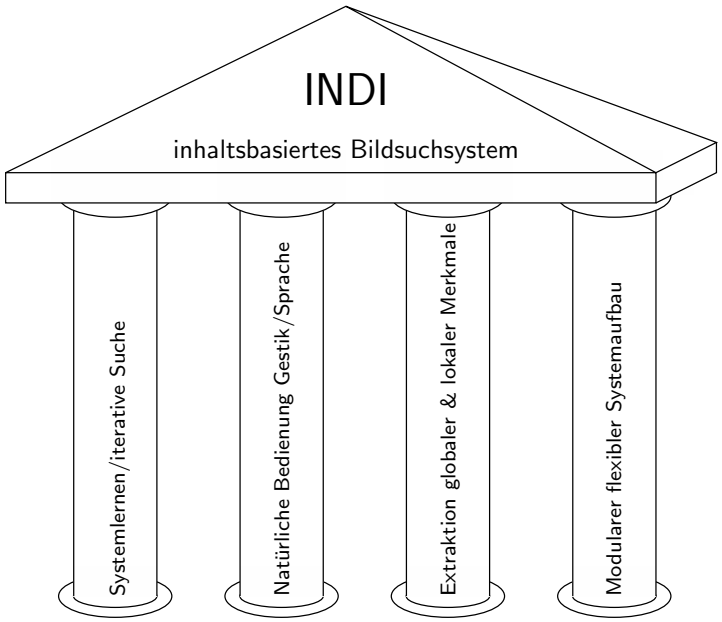


Abb. 3.2: **Anforderungen an das inhaltsbasierte Bildsuchsystem INDI**

Da es bei diesem Datenbanksystem, wie wir im Folgenden sehen werden, einige Verarbeitungseinheiten gibt, die an spezielles Bildmaterial angepasst werden können, und andere Teile existieren, die für sich einen Forschungsschwerpunkt bilden, ist es daher wünschenswert, solche Teile einfach austauschbar zu halten.

Abbildung 3.2 fasst die Hauptanforderungen, die an das Bildsuchsystem gestellt werden, zusammen. Diese Anforderungen werden in den folgenden Abschnitten bezüglich ihrer Umsetzungsmöglichkeiten diskutiert.

3.2 Der Suchprozess

Der Suchprozess bildet die Grundlage des Systems, auf die die anderen Systemteile, wie zum Beispiel die Interaktion aufsetzen. Er ist damit der wichtigste zu spezifizierende Systemteil. Einige der nun folgenden Aufbaudetails betreffen lediglich Interna des Suchablaufs und wirken sich nicht auf die Benutzung der Datenbank aus. Andere, wie beispielsweise die Auswahl des Suchverfahrens, haben maßgeblichen Einfluss auf das Gesamtsystem.

3.2.1 Formulierung der Suchanfrage

Das Umsetzen der semantischen Beschreibung einer Suchintention in eine formale Beschreibung, die vom System verwendet wird, ist der Vorgang, der hier als „Formulierung der Suchanfrage“ bezeichnet wird. Entsprechend der Systemanforderung sollte die Gestaltung der Anfrage so geschaffen sein, dass vom Bediener der Datenbank keinerlei Fachkenntnisse im Bereich der Bildverarbeitung bzw. der Datenbanken vorausgesetzt werden können. Vielmehr sollte die Anfrage einfach und intuitiv durchgeführt werden können und zu nachvollziehbaren Ergebnissen führen.

Suchverfahren

Zu der Klärung der Frage, wie eine Suchanfrage formuliert werden soll, wird zunächst untersucht, mit welchen möglichen Suchintentionen ein Benutzer ein Bildsuchsystem gebrauchen möchte. Entsprechend der Suchintention werden folgende Suchverfahren unterschieden (vergleiche [Sme00, Cox00]:

Zielsuche: Bei der Zielsuche soll ein konkretes Bild aus einer Bildermenge gefunden werden. In diesem Fall ist der Benutzer meist mit der Bildermenge vertraut und hat eine exakte Vorstellung vom Suchbild vor Augen.

Dieses Szenario ist bei einem privaten Fotoarchiv denkbar, aus dem ein Bild für einen Vergrößerungsabzug zu suchen ist.

Kategoriensuche: Ist nicht der Inhalt auf konkrete Individuen bzw. Objekte festgelegt, dann werden Bilder eines bestimmten Szenarios gesucht. Diese Kategoriensuche ist meist nicht auf das Finden eines einzelnen Bildes beschränkt, sondern es wird so lange gesucht, bis eine Menge von Bildern einer Kategorie gefunden wurde. Es ist möglich, dass die gesamte Menge das Ziel der Suche sein kann, ebenso ist vorstellbar, dass dann ein einzelnes Bild dieser Menge als das Zielbild definiert wird.

Eine solche Suche ist beispielsweise bei dem Layout einer Internet-Seite vorstellbar, auf der zum Beispiel ein schönes Segelschiff auf hoher See bei herrlichem Wetter abgebildet

werden soll. Es kommt nicht darauf an, ein ganz bestimmtes Schiff zu finden, lediglich der Gesamteindruck des gefundenen Bildes bestimmt den Sucherfolg.

Durchblättern: Bei diesem Suchverfahren ist der Bildinhalt nur vage oder gar nicht festgelegt. Der Benutzer hat bei diesem Verfahren keine eindeutige Suchintention, sondern diese kann während der Suche stark variieren. Eine solche Suche kann vom System nicht weitergehend unterstützt werden, da der Benutzer nicht in der Lage ist, eine eindeutige Suchanfrage zu formulieren. Das System kann hier lediglich für eine gute Übersicht und für komfortable Möglichkeiten der Navigation sorgen. So könnte es beispielsweise die Möglichkeit geben, Bilder, die vom Benutzer als potentielle Kandidaten markiert werden, in einer zusätzlichen Galerie zu präsentieren.

Klassifikation von Suchanfragen

Smeulders et al. stellen in [Sme00] Klassen von Anfragen vor, mit denen die besprochenen Suchverfahren durchgeführt werden können. Die Klassifizierung wird anhand des semantischen Inhalts in exakte und ungefähre Anfragen vorgenommen.

Exakte Anfrage: Als exakte Anfragen werden solche bezeichnet, die semantisches Wissen referenzieren. Je nachdem, worauf sich die Semantik bezieht, werden die Anfragen gruppiert. Anfragen wie: „Bilder mit einem Tier vor einer Wiese“, die sich auf die geometrische Anordnung von Objekten im Bild beziehen, bilden hier eine Gruppe. Anfragen, die sich auf das Vorkommen von Bildinhalten beziehen, stellen eine weitere Gruppe dar. Eine solche Anfrage könnte folgendes Aussehen haben: „Bild mit mindestens 20% Himmel, 30 %Wald und 20% Wiese“. Die letzte Gruppe dieser Klasse bilden Anfragen, die sich nicht direkt auf den Bildinhalt beziehen, sondern weitergehende semantische Zusammenhänge der Bilder referenzieren. Hier kann „Bilder, die in der Schweiz aufgenommen wurden“ als Beispiel genannt werden.

Das Resultat einer Suche mit einer exakten Anfrage ist eine Menge von Bildern, auf die die Attribute der Anfrage zutreffen.

Ungefähre Anfrage: Im Gegensatz zu den exakten Anfragen wird in dieser Klasse mit Beispielen gearbeitet. Anfragen bedienen sich hier also nicht der semantischen Information. Eine ungefähre Anfrage würde wahrscheinlich von jedem Menschen durchgeführt werden, wenn diese Person weder mündlich noch schriftlich den Bildinhalt beschreiben dürfte.

Anfragen dieser Klasse können durch die Auswahl eines einzelnen ähnlichen Bildes oder durch eine Zusammenstellung je einer Menge von ähnlichen und unähnlichen Bildern getätigt werden. Die hier genannte Ähnlichkeit könnte sich beispielsweise auf den farblichen oder geometrischen Aufbau der Bilder beziehen.

Anfragen mittels eines Beispiels

Exakte Anfragen lassen sich nicht von einem rein inhaltsbasierten System bearbeiten, denn die semantische Analyse der Bilder ist zur Zeit nicht automatisch von einem System durchführbar. Aus diesem Grund sollen die nicht exakten Anfragen im Folgenden näher untersucht werden.

Abbildung 2.1 des vorherigen Kapitels veranschaulicht die unterschiedlichen Anfragegruppen in der Klasse. Generell muss das System in der Lage sein, aus dem gegebenen Beispiel eine formale Darstellung der Anfrage zu erstellen, die als Grundlage für eine folgende Suche dient. Das Suchergebnis ist bei dieser Klasse im Allgemeinen eine nach Ähnlichkeit geordnete Liste aller Bilder der Datenbank.

Geometrischer Aufbau: Ein Beispiel des geometrischen Aufbaus eines Bildes kann vom Benutzer zum Beispiel durch Anfertigen einer Skizze gegeben werden (engl.: *Query By Sketch*). Eine solche Skizze kann vom System nicht direkt für den Vergleich mit den in der Datenbank gespeicherten Bildern benutzt werden. Deshalb muss sowohl die Skizze als auch die Bilder der Datenbank entsprechend analysiert werden.

Die Segmentierung des Bilddatenbestandes ist eine Möglichkeit, mittels Form- oder Layout-Merkmalen einen Vergleich mit der gezeichneten Skizze durchzuführen. Wird die Segmentierung automatisch durchgeführt, dann sind die Ergebnisse nur dann sinnvoll für den Betrieb dieser Anfrageart geeignet, wenn der Bilddatenbestand bezüglich seiner Domäne eingeschränkt ist.

Gelänge es, geeignete Symmetrieeigenschaften der Bilder einer Datenbank zu extrahieren, so wäre ein Vergleich der Skizze auf Basis dieser Eigenschaften denkbar, ohne Bilder segmentieren zu müssen.

Auch das Farb-Layout eines Bildes kann mittels einer Skizze der Anfrage hinzugefügt werden. Die gegebene Farbinformation darf jedoch nur als Anhaltspunkt aufgefasst werden, denn die subjektive Farbwahrnehmung weicht oft erheblich von der tatsächlichen Farbverteilung ab. Auch sollte die Menge der zu benutzenden Farben entsprechend eingeschränkt sein, um dem Benutzer nicht eine zu hohe Wertigkeit des Farbtons zu suggerieren. Eine vorherige Segmentierung bietet auch hier den entscheidenden Vorteil, dass Farbinformationen auf semantisch zusammenhängenden und nicht etwa generischen Bildbereichen ermittelt werden können.

Bei einem relevanzbasierten iterativen Bildsuchsystem bietet es sich an, neben den Bildcharakteristika, die für den Ähnlichkeitsvergleich bei diesem Anfragetyp benutzt werden, nach dem initialen Suchschritt weitere Charakteristika von bewerteten Bildobjekten für den Ähnlichkeitsvergleich heranzuziehen. Dadurch entsteht eine verbesserte Beschreibung des Gesuchten.

Beispielbild: Die Anfrage mittels eines Beispielbildes bildet die zweite Gruppe dieser Klasse (engl.: *Query By Example*). Denkbar einfach und intuitiv gestaltet sich hier das Erzeugen der Anfrage. Durch die Auswahl eines Bildes oder einer Region eines Bildes der Datenbank bzw. durch das Einbringen eines sich nicht in der Datenbank befindlichen Bildes kann eine Anfrage formuliert werden.

Diese Art der Anfrage ist sehr gut bei der Suche in einem Papierfotoarchiv vorstellbar, bei der mehrere Personen beteiligt sind. Durch ein ähnliches Bild kann den Suchpartnern vermittelt werden, wie das gesuchte Bild ungefähr aussieht.

Ebenso einfach ist das Umsetzen der Anfrage in eine formale Darstellung, denn für jedes in der Datenbank befindliche Bild existiert bereits eine solche Darstellung. Für

externe Bilder muss diese Darstellung mit vorhandenen Algorithmen berechnet werden, wobei lediglich sichergestellt werden muss, dass diese Berechnung innerhalb eines vom Benutzer akzeptierbaren Zeitraums durchgeführt werden kann.

Gruppierung von Beispielbildern: Oft reicht die Angabe eines einzelnen Bildes für eine Anfrage nicht aus, weil das Bild unterschiedliche Teilinhalte umfasst und nur einer dieser Teile die eigentliche Suchanfrage repräsentiert. In diesem Fall bietet es sich an, mehrere Bilder anzugeben, die den gesuchten Inhalt gemeinsam aufweisen. Auch könnten negative Bildbeispiele angegeben werden, bei denen der Bildinhalt nicht die Suchintention widerspiegelt.

In dieser Gruppe von Anfragen werden also vom Benutzer Bildgruppierungen gebildet, wobei mindestens eine Gruppe mit positiven Beispielen für die Suchintention spezifiziert werden muss. Zusätzlich können auch negative Beispielbilder genannt werden. Basierend auf den Gruppierungen werden statistische Analysen durchgeführt, um daraus zu schließen, welche Gemeinsamkeiten aus formaler Sicht positive sowie negative Bilder aufweisen.

Des Weiteren kann aus der Menge der positiv klassifizierten Bilder eine neue formale Darstellung eines Suchbeispiels gewonnen werden, die für die sich anschließende Suche verwendet wird.

Fazit für den Systementwurf

Die Suche mittels exakter Anfragen, die auf semantischem Wissen beruht, kommt dem Benutzer sicherlich entgegen, denn es ist natürlich, die Dinge beim Namen zu nennen. Gegen diese Klasse spricht lediglich, dass die benötigte semantische Information nicht immer automatisch vom System ermittelt werden kann. Besteht in solchen Fällen die Notwendigkeit, eine exakte Anfrage umzusetzen, müssen die fehlenden Informationen manuell erzeugt und hinzugefügt werden.

Ein inhaltsbasiertes System extrahiert die der Suche zugrunde liegende Information automatisch. Nicht exakte Anfragen können von solchen Systemen verarbeitet werden. Anfragen, die durch Anfertigen einer Bildskizze gestellt werden, wirken sehr vielversprechend. Hier herrscht jedoch die Gefahr, dass die Skizzen immer mit Objektwissen des Benutzers gezeichnet werden und nur gute Suchergebnisse zu erwarten sind, wenn die Segmentierung der Bilder von ähnlich guter Qualität ist. Gerade das automatische Segmentieren ist zur Zeit nur hinreichend gut gelöst, wenn die Bilddomäne stark eingeschränkt ist, was unter Umständen nicht erwünscht ist. Daher muss der Segmentiervorgang ohne Einschränkungen bei dem verwendeten Bildmaterial von Hand erfolgen. Des Weiteren öffnet sich diese Art der Anfrage nur solchen Benutzern, die entsprechende Erfahrung mit dem Erzeugen geeigneter Skizzen aufweisen.

Anfragen, die mit einem Beispielbild oder durch Gruppierung mehrerer Bilder gestellt werden, stellen sich dem Benutzer einfach, intuitiv und unmissverständlich dar. Lediglich die Auswahl eines oder mehrerer Beispielbilder ist ein Schritt der Anfrageformulierung, der zufriedenstellend gelöst werden muss. Hier gilt es, dem Benutzer eine Auswahl der Datenbankbilder zu geben, die entsprechend weit gestreut ist, so dass ein ähnliches Bild gefunden werden kann. Diese als *page zero*-Problem bezeichnete Aufgabe [Cas98] kann zum Beispiel durch Clustern

der Bilddatenmenge gelöst werden, wobei bei der zufälligen Auswahl die Clusterzugehörigkeit entsprechend berücksichtigt wird. Des Weiteren muss der Benutzer die Möglichkeit haben, aus der Ergebnismenge der Suche ein neues bzw. die neuen Beispielbilder auszuwählen. Eine iterative Suche ist ideal für die Erstellung der Gruppen positiver und negativer Beispielbilder. Durch Bewertungen der Suchergebnisse können genau diese Attribute den Bildern zugeordnet werden.

3.2.2 Merkmale zur formalen Beschreibung von Bildinhalten

Für ein inhaltsbasiertes Bildsuchsystem gilt es, eine formale Beschreibung von Bildern zu finden, die automatisch berechnet und auf deren Basis eine Ähnlichkeitsbestimmung zwischen zwei Bildern durchgeführt werden kann.

Entsprechend einer menschlichen Beschreibung eines Bildes, bei der die wichtigsten Inhalte, Attribute und der geometrische Aufbau des Bildes charakterisiert wird, sollen bedeutsame beschreibende Eigenschaften eines Bildes durch die bereits erwähnte Merkmalsextraktion gewonnen werden.

Im Allgemeinen wird für die interne Repräsentation der gewonnenen Merkmale eine vektorielle Darstellung verwendet. Diese mathematische Repräsentation als Merkmalsvektor hat viele verarbeitungstechnische Vorteile und ist ideal für eine nachfolgende Ähnlichkeitsbestimmung der

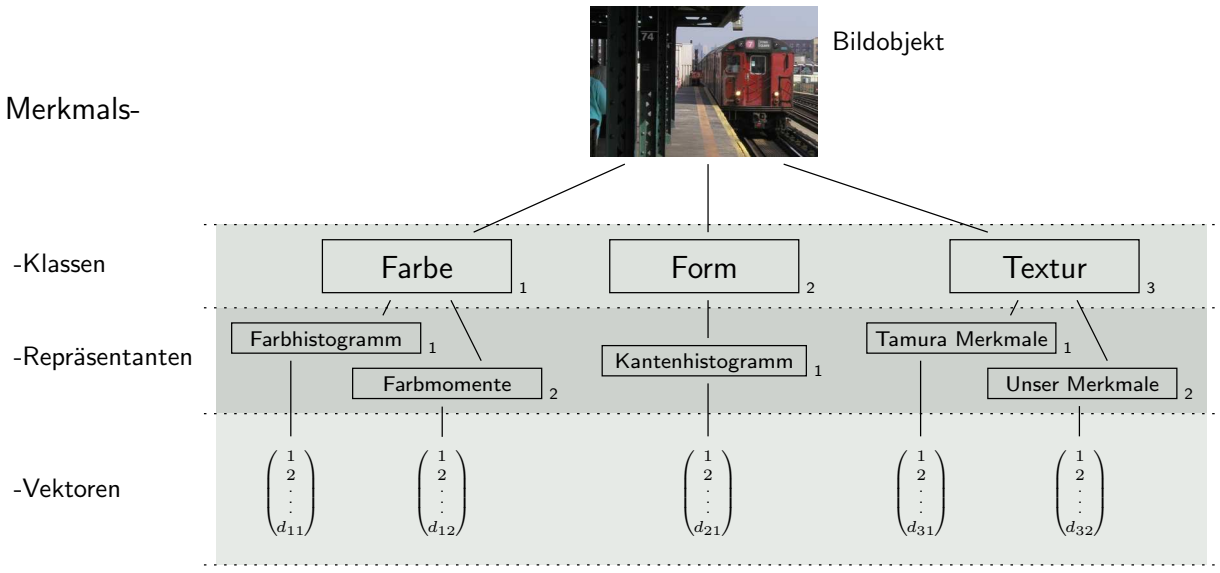


Abb. 3.3: **Vom Bildobjekt zum Merkmalsvektor:** Das hier gezeigte Systembeispiel verwendet Merkmale aus den drei Merkmalsklassen Farbe, Form und Textur. Die Besetzung der Klassen ist nicht gleichgewichtet. Die Klassen Farbe und Textur beinhalten je zwei Merkmalsrepräsentanten, die Klasse Form nur einen. Für jedes Bildobjekt wird für jeden Repräsentanten ein Vektor extrahiert, wobei die Vektoren unterschiedliche Dimensionen d_{ij} aufweisen können.

Merkmale zweier Bilder verwendbar. Andere Merkmale lassen sich nicht durch einen Vektor repräsentieren. Vorstellbar ist hier ein Merkmal, das „Hauptfarben“ genannt werden könnte. Die Anzahl der Farben ist hier variabel und die Beschreibung der Farbe selbst ist unter Umständen schlecht durch einen einzigen skalaren Wert zu beschreiben. Ergebnisse solcher Merkmalsextraktionen werden als Signaturen bezeichnet.

Um eine gute Beschreibung der Bilder zu erhalten, wird nicht allein ein einziges, sondern ein ganzer Satz dieser Merkmale verwendet. Dabei wird die inhaltliche Bedeutung der verwendeten Merkmale gestreut, so dass alle gewünschten Attribute eines Bildes möglichst gut unterschieden werden können. Die Summe aller dem Bild zugehörigen Merkmalsvektoren bzw. Signaturen stellt die formale Repräsentation eines Bildes, also dessen Inhalt, im System dar.

Die in der Abbildung 3.3 gezeigte Begrifflichkeit wird im Folgenden hinsichtlich der Merkmalsextraktion verwendet. Diese hier aufgegriffene hierarchische Ordnung, die von Rui et al. [Rui97] vorgestellt wurde, wird bei der Vergabe von Gewichten für das Systemlernen besonders interessant.

Die gängigsten in den zahlreichen Bildsuchsystemen eingesetzten Merkmalsklassen basieren auf Farb-, Textur- und Forminformationen, wie in Abbildung 3.3 gezeigt. Dabei handelt es sich um generelle Merkmale, die nicht auf eine spezielle Bilddomäne angepasst sind.

Farbinformation: Die Farbinformation wird häufig durch ein Histogramm repräsentiert. Mit einem solchen Histogramm wird ausgedrückt, wie viele Pixel des zu untersuchenden Bildes einen bestimmten Farbwert aufweisen. Je nach Intention werden bei der Histogrammerzeugung unterschiedliche Farbräume zugrunde gelegt. Histogramme haben den Vorteil, dass bezüglich der Bildgröße und Rotation invariant sind und damit zu guten Suchergebnissen führen können [Swa91]. Das durch die Diskretisierung bedingte Farbrauschen, kann durch kumulative Histogramme [Str95] kompensiert werden. Histogramme haben, wenn sie bezüglich der Ähnlichkeit genügend aussagekräftig bleiben sollen, den Nachteil, dass sie im Allgemeinen sehr hochdimensional und spärlich besetzt sind. Eine vereinfachte und kompakte Repräsentation von Farbhistogrammen wird durch das von Stricker und Orengo vorgestellte Verfahren *Color Moments* (siehe auch [Str95]) erzeugt. In diesem Fall wird die Farbverteilung jedes Farbkanals durch die ersten drei Momente, den Mittelwert, die Varianz und die Schiefe, repräsentiert. Damit wird die Dimension des Merkmalsvektors drastisch reduziert.

Texturinformation: Eine Texturinformation gibt Aufschluss über den visuellen Aufbau einer homogen erscheinenden Fläche eines Bildes. Hierbei kann es sich um Flächen homogener Farbe aber auch um gleichmäßig gemusterte Flächen handeln. Im Gegensatz zu den Farbinformationen werden Texturinformation also aus den nachbarschaftlichen Beziehungen mehrerer Pixel bestimmt. Aus der Tatsache, dass ein Muster wiederum ein Muster beinhalten kann, so wie es beispielsweise bei einem mit einem Muster (Makro Textur) bedruckten Stoff, der an sich bereits eine Textur (Mikro Textur) aufweist, der Fall ist, stellt sich die Frage, in welcher Skalierung die Texturinformation bestimmt werden soll. Sebe und Lew stellen in [Seb01] eine Übersicht über gängige Texturverfahren vor.

Forminformation: Ebenso wie bei den Texturinformationen werden bei Forminformationen nicht einzelne Pixel betrachtet, sondern besondere nachbarschaftliche Beziehungen aus-

gewertet. Ein Repräsentant dieser Gruppe ist das Kantenhistogramm, das die Anzahl der Kantenpixel von Kanten unterschiedlicher Richtungen beinhaltet (siehe [Bra00]). Eine zweite Klasse von Formmerkmalen zeichnet sich dadurch aus, dass die Berechnung auf zuvor detektierten Bildregionen oder Objekten basiert. Der Einsatz solcher Merkmale bleibt jedoch meist spezialisierten Systemen vorbehalten, da die Bildsegmentierung oder die Detektion von interessanten Bildregionen (engl.: *Regions Of Interest*) nur auf speziellem Bildmaterial zu befriedigenden Ergebnissen führt. Die Merkmalsrepräsentanten wie beispielsweise Fläche, Zirkularität oder Exzentrizität basieren nicht wie die bereits genannten auf den Farbinformationen der Pixel, sondern hier wird für die Pixel lediglich die binäre Zugehörigkeit zu einer Region ausgewertet.

Die hier vorgestellten Farb- und Texturmethoden, wie beispielsweise ein Histogramm, extrahieren statistische Informationen, zum Beispiel das Vorkommen einer bestimmten Farbe oder bestimmter Grauwertübergänge. Die räumliche Zuordnung bezogen auf das Gesamtbild wird dabei gänzlich verworfen. Oft ist aber genau die räumliche Verteilung der Information von Bedeutung, hier mag ein klassisches Strandbild mit hellem Sand und blauem Himmel oder ein Bild, was die untergehende Sonne zentriert zeigt, als Beispiel genannt werden.

Layout-Merkmale beinhalten diese räumliche Information. Ähnlich wie bei der Anwendung von Bildsegmentierern werden bei diesen Merkmalen die oben angeführten Methoden der Verktorbestimmung nicht allein auf dem gesamten Bild, sondern auch auf lokalen Bereichen angewendet. Im Gegensatz zu den segmentierten Bildern werden allerdings die Bildbereiche meist durch ein oder mehrere sich überlagernde gleichförmige Raster festgelegt.

Fazit für den Systementwurf

Für das Erreichen guter Suchergebnisse ist es unumgänglich, das System mit einer Menge von unterschiedlich gearteten Merkmalsrepräsentanten aus verschiedenen Klassen auszustatten. Für die Verbesserung der Suchergebnisse bei der Verwendung von speziellem Bildmaterial ist es denkbar, spezialisierte Merkmale einzusetzen. Um diese Flexibilität zu erhalten, ist lediglich die Repräsentation der extrahierten Bildinformation festzulegen. Die darauf basierende Weiterverarbeitung kann dann einheitlich durchgeführt werden.

Die Struktur von Signaturen kann schlecht generalisiert werden. Es ist daher nicht sinnvoll, jene Repräsentation zu vereinheitlichen. Signaturen müssen in jeder Hinsicht einer besonderen Verarbeitung unterzogen werden. Es ist also lediglich dafür Sorge zu tragen, dass das System in der Lage ist, zwischen Signaturen und Merkmalsvektoren zu unterscheiden.

3.2.3 Distanzbildung und Kombination

Das Kernstück der inhaltsbasierten Bildsuche bildet die Idee, die Merkmale als als Vektoren in einem Raum zu interpretieren, auf dessen Basis die Möglichkeit besteht, einen räumlichen Abstand zwischen den Vektoren zweier Bildobjekte zu berechnen. Dieser Abstand bildet dann wiederum die Grundlage für die Generierung eines Ähnlichkeitswertes. Unter der Verwendung mehrerer Merkmale für die formale Repräsentation der Bildobjekte müssen die erzeugten Distanzen bzw. Ähnlichkeitswerte zu einem einzigen Wert zusammengefasst werden.

Können die erzeugten Merkmale wirklich als Vektor interpretiert werden, bei denen die Komponenten voneinander unabhängig sind jedoch ähnliche Charakteristika aufweisen, dann können Abstandsmaße wie zum Beispiel der euklidische Abstand, wie in Gleichung (3.1) gezeigt, für die Distanzbildung verwendet werden. Die genannten Voraussetzungen sind jedoch nicht immer erfüllt.

$$d(\vec{r}, \vec{q}) = \sqrt{(\vec{r} - \vec{q})^T (\vec{r} - \vec{q})} \tag{3.1}$$

Oftmals weisen die einzelnen Komponenten der Vektoren unterschiedliche Wertebereiche oder unterschiedliche Dynamik auf. Damit auch diese Vektoren den oben beschriebenen Voraussetzungen entsprechen, werden Normierungen durchgeführt. Die Unterschiedlichen Normierungsverfahren sollen dafür sorgen, dass die Differenzen, die sich für die unterschiedlichen Komponenten zwischen den Vektoren ergeben, gleiche oder ähnliche Wertebereiche aufweisen. Mit diesem Vorgehen wird vermieden, dass Differenzen einer Komponente grundsätzlich von Differenzen anderer Komponenten überschattet werden.

Die Normierung findet im Allgemeinen direkt im Anschluss an die Merkmalsberechnung statt. Die Normierung der Wertebereiche der unterschiedlichen Komponenten kann direkt durchgeführt werden, denn die Wertebereiche sind durch die Berechnungsvorschriften der Merkmale klar gegeben. Die Anwendung von Normierungen, die die unterschiedliche Dynamik der Vektoren berücksichtigen, verlangt jedoch die Analyse der bereits gebildeten Vektoren und muss dadurch in einem separaten Schritt nach der Merkmalsberechnung durchgeführt werden. Sie hat damit den Nachteil, dass sie von dem Inhalt der Bilder der Datenbank abhängt, was bei dem Einfügen neuer Bilder in die Datenbank berücksichtigt werden muss.

Bei einer anderen Gruppe von Merkmalen sind die Komponenten der vektoriellen Darstellung nicht wie eben betrachtet unabhängig voneinander. So sind beispielsweise die Komponenten eines Farbhistogramms durch die eingesetzte Quantisierung stark mit den jeweiligen Nachbarn verbunden. Eine Distanzberechnung durch den von Swain et. al vorgestellten Histogrammschnitt [Swa91] berücksichtigt diese Abhängigkeit nicht. Ein Abstandsmaß, was die Abhängigkeiten der Komponenten untereinander ganz allgemein unterstützt, ist der in Abschnitt 3.2.4 vorgestellte generalisierte euklidische Abstand, der bei symmetrischen Matrizen auch den Gesetzmäßigkeiten von metrischen Räumen unterstützt (vergleiche Zeidler [Zei96]).

Werden die oben angeführten Distanzberechnungen richtig auf die verwendeten Merkmale angepasst, resultiert für den Vergleich zweier Bildobjekte eine der Menge der verwendeten Merkmale entsprechende Anzahl von Distanzwerten. Dabei können auch Distanzberechnungen verwendet werden, die nur für einen bestimmten Merkmalsrepräsentanten sinnvoll einsetzbar sind. Die gebildeten Distanzwerte weisen wiederum unterschiedliche Wertebereiche und Dynamik auf, was für die sich anschließende Zusammenfassen berücksichtigt werden muss.

Es werden zwei grundsätzliche Methoden, die Distanzwerte der zu benutzenden Merkmalsräume zu kombinieren, unterschieden. Das ist zum einen die Linearkombination der Einzeldistanzen, also eine hierarchisch gebildete Distanz, und im anderen Fall handelt es sich um die die Zusammenlegung der Merkmalsräume zu einem gemeinsamen Raum, in dem nur eine einzige Distanz gebildet werden muss, der flache Ansatz. Beide Ansätze werde im Folgenden gegenübergestellt.

Vereinigung der Merkmalsräume

Unter der Vereinigung der Merkmalsräume ist zunächst eine Vergrößerung der Raumes zu verstehen, bei der die Merkmalsvektoren konkateniert werden, so dass sich die Dimension des vereinigten Raums aus der Summe aller Repräsentantendimensionen ergibt. Durch diese Maßnahme muss bei einer Distanzbestimmung nur ein Wert berechnet werden, was die Algorithmik sehr einfach hält. Der gravierende Vorteil dieser Handhabung von Merkmalsräumen ist jedoch der, dass auf dem vereinigten Raum eine Hauptkomponentenanalyse durchgeführt werden kann. Diese Analyse erlaubt es, eine Dimensionsreduktion aufgrund von redundanter Information der Vektoren durchzuführen, so dass sich sowohl bezüglich der Speicherung als auch der Verarbeitung der Vektoren ein erheblich kleinerer Verbrauch von Ressourcen ergibt. Diese Vorgehensweise kann auch in jedem Raum separat durchgeführt werden, jedoch werden Redundanzen, die vektorübergreifend sind, nicht erfasst.

Schwierig gestaltet sich hier die Normierung des konkatenierten Vektors, bei der die Dimension der zugrunde liegenden Vektoren mit berücksichtigt werden muss. Wird dieser Verhältnismäßigkeit keine Rechnung getragen, dann dominieren die Merkmale mit höherdimensionalen Vektoren die sich anschließende Distanzbildung.

Obwohl die Vereinigung der Merkmalsräume zunächst augenscheinlich sehr vielversprechend ist, birgt sie den Nachteil, dass die Wertung der beteiligten Merkmalsrepräsentanten fest zu gleichen Teilen in die Distanzberechnung eingeht. Damit ist eine späteres Modifizieren der Gewichtung durch den Einsatz eines Lernverfahrens nicht mehr möglich.

Distanzbasierte Verknüpfung der Distanzen

Bei dieser Art der Distanzbildung werden die Distanzen zunächst für jeden beteiligten Merkmalsrepräsentanten gebildet. Die Gesamtdistanz bildet sich aus der gewichteten Summe der Einzeldistanzen, wobei in die Gewichtung Normierungsgrößen eingehen können.

Bei dem realen Einsatz einer Bilddatenbank ist davon auszugehen, dass die Repräsentanten der unterschiedlichen Merkmalsklassen ungleich stark vertreten sind. Daher muss dafür Sorge getragen werden, dass die Merkmalsklassen, die nur wenige Repräsentanten aufweisen können nicht durch andere Klassen überschattet werden. Auch hier können Gewichtungen der Einzeldistanzen Abhilfe schaffen.

Rangbasierte Verknüpfung der Distanzen

Gänzlich kann auf eine Normierung verzichtet werden, wenn die ermittelten Distanzen nur für das Bilden einer Abfolge der Objekte in jedem Merkmalsraum benutzt werden. Mit dem Einsatz dieser rangbasierten Verknüpfung werden also die Objekte in eine äquidistante Abfolge gebracht. Diese Vereinfachung bringt es naturgemäß mit sich, dass jegliche Relationen der Distanzen nicht mehr in die Gesamtdistanz eingehen können. Der Rang, den ein Objekt in der Abfolge einnimmt muss wiederum auf einen Distanz- oder Ähnlichkeitswert abgebildet werden. Diese Werte können dann wie bei der distanzbasierten Verknüpfung aufsummiert werden. Die Repräsentanz der unterschiedlichen Merkmalsklassen ist hier ebenso zu berücksichtigen.

Kombination von rang- und distanzbasierter Verknüpfung

Schließlich ist eine Kombination von rang- und distanzbasierter Verknüpfung denkbar, die ebenfalls nur dann einsetzbar ist, wenn die Suche in allen Merkmalsräumen getrennt durchgeführt wird. So kann beispielsweise die Distanz eines Bildobjekts eines bestimmten Ranges in jedem Repräsentantenraum als Normierungsgrundlage dienen.

Fazit für den Systementwurf

Aus den vorangehenden Abschnitten geht hervor, dass der hierarchische Ansatz zwar zur Laufzeit rechenintensiv ist, jedoch eine erhebliche Flexibilität beinhaltet, die für den Forschungsprototyp wünschenswert ist.

Eine besondere Flexibilität bietet der hierarchische Ansatz in der Möglichkeit, für die unterschiedlichen Merkmalsräume angepasste, also unterschiedliche Distanzfunktionen zu wählen. Das ist sogar zwingend erforderlich, wenn Signaturen mit variablen Vektorlängen verglichen werden sollen.

Eine Vereinigung der Merkmalsräume kann als ein Sonderfall der hierarchischen Distanzberechnung behandelt werden, bei der durch das Zusammenlegen der Einzelräume die Suche faktisch nur auf einem Merkmal beruht. Daher wird die Flexibilität des Systems bei der Umsetzung der hierarchischen Suche maximal garantiert.

3.2.4 Iterative Suche / Systemlernen

Im vorangegangenen Kapitel wurden bereits unterschiedliche Formen iterativer Bildsuche vorgestellt (siehe Abschnitt 2.1). Durch die Einbindung des Benutzers in eine iterative Suche wird das System in die Lage versetzt, die Suchintention des Benutzers zu adaptieren und damit das Suchergebnis entscheidend zu verbessern.

Bei der Einbeziehung des Benutzers in den Suchvorgang gilt es, zwei Fragestellungen zu beantworten:

Interaktion: Wie soll die Interaktion gestaltet sein, so dass der Benutzer einen klar nachvollziehbaren Einfluss auf die Erzeugung des Suchergebnisses haben kann?

Einflussnahme: Wie soll das Ergebnis der Interaktion für das Systemlernen zur Verbesserung des Suchergebnisses herangezogen werden?

Das Ziel der Interaktion, die die Basis des Systemlernens bildet, ist es, den Mangel des semantischen Wissens, den das Systems aufweist, zu kompensieren. Hintergründe der technischen Realisierung sollen hier vom Benutzer ferngehalten werden. Deshalb dürfen Interaktionen nur allgemein verständliche Anforderungen an den Benutzer stellen. Weit verbreitet ist eine Interaktion, bei der der Benutzer die Bilder des präsentierten Suchergebnisses bezüglich der Relevanz der Suche bewerten muss. Bewertungen können diskret in wenigen Stufen oder auch

prozentual bezüglich der Relevanz abgegeben werden. Diese Art der Interaktion erfüllt alle genannten Anforderungen ist vollkommen losgelöst von den verwendeten Merkmalen bzw. deren Repräsentanten.

Die abgegebenen Bewertungen lassen sich auf mehreren Wegen zur Adaption der Suche verwenden. Ein sehr vielversprechender Ansatz ist der, der im MARS-System verwirklicht wurde [Rui97, Por99]. Bei diesem Ansatz wird die Relevanzbewertung des Benutzer dazu verwendet, die zugrunde liegenden Bildcharakteristika, die Merkmalsrepräsentanten, zu gewichten und interne Systemparameter anzupassen, um das semantische Konzept des Benutzers zu beschreiben. Dabei handelt es sich also im Allgemeinen eher um eine Kategorien- als um eine Zielsuche (vergleiche Abschnitt 3.2.1).

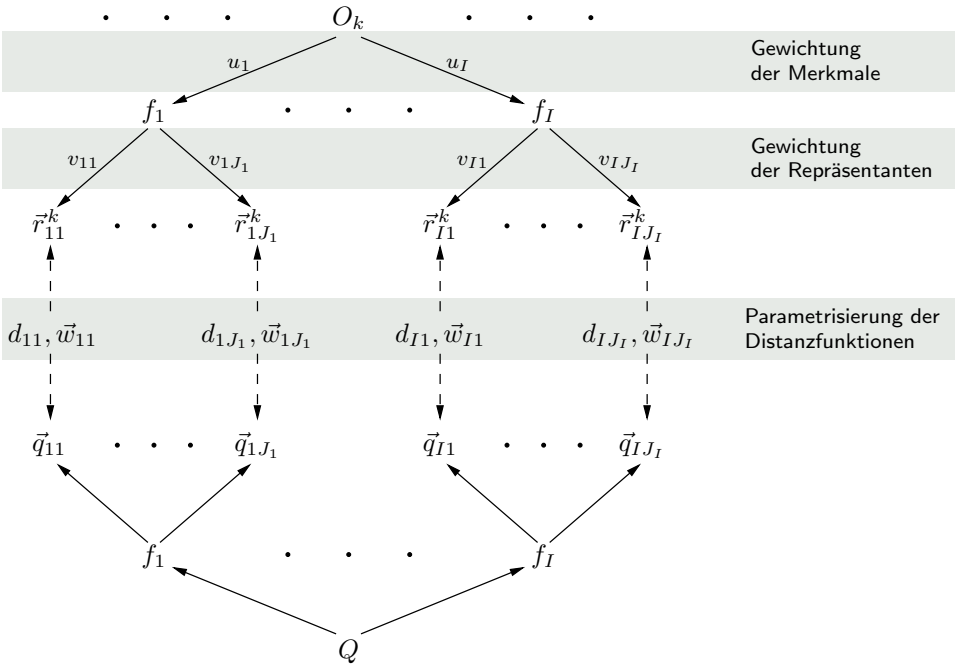


Abb. 3.4: **Gewichtete hierarchische Distanzbildung bei MARS [Rui98]:** Die Distanzberechnung eines Bildobjekts O_k und des Beispielobjekts Q wird stufenweise durchgeführt. Die Berechnung basiert auf den entsprechenden Vektoren $\vec{r}_{ij}$ und $\vec{q}_{ij}$ der Repräsentantenebene, die unter Zuhilfenahme der zugewiesenen Distanzfunktionen d_{ij} und den ermittelten Gewichtsvektoren $\vec{w}_{ij}$ durchgeführt wird. Die ermittelten Distanzen werden auf Merkmalsebene mit v_{ij} gewichtet zusammengefasst. Die so berechneten Merkmalsdistanzen werden unter Berücksichtigung der Merkmalsgewichte u_i zu dem Gesamtdistanzwert verknüpft.

Abbildung 3.4 präsentiert die beim MARS-System verwendete Distanzberechnung, wobei die Gesamtdistanz folgendermaßen gebildet wird:

$$D(O_k, Q) = \sum_{i=1}^{I} u_i D_i \tag{3.2}$$

$$D_i(O_k, Q) = \sum_{j=1}^{J_i} v_{ij} D_{ij} \tag{3.3}$$

$$D_{ij}(O_k, Q) = d_{ij}(\vec{r}_{ij}^{\,k}, \vec{q}_{ij}, \vec{w}_{ij}) \tag{3.4}$$

Die Einflussnahme der Relevanzbewertung findet auf zwei der drei Ebenen statt. Folgende Bewertungen mit den zugehörigen Wertigkeiten wurden eingesetzt:

$$\pi = \begin{cases} +3 & \hat{=} \text{ sehr relevant} \\ +1 & \hat{=} \text{ relevant} \\ 0 & \hat{=} \text{ kein Urteil} \\ -1 & \hat{=} \text{ nicht relevant} \\ -3 & \hat{=} \text{ gar nicht relevant} \end{cases}$$

Die Menge der positiv bewerteten Bildobjekte bildet eine Sonderposition. Hier wird angenommen, dass die zugehörigen Bildobjekte Gemeinsamkeiten aufweisen, die bei allen gesuchten Bildobjekten ebenfalls zu verzeichnen sind. Um diese Gemeinsamkeiten zu extrahieren, wird eine statistische Analyse der Repräsentantenvektoren der positiv bewerteten Bilder durchgeführt. Die Komponenten der Vektoren, die eine hohe Streuung aufweisen, werden geschwächt, Komponenten, bei denen eine kleine Standardabweichung zu verzeichnen ist, werden verstärkt. Es resultiert daraus für jeden Raum eines Merkmalsrepräsentanten ein Gewichtungsvektor, der in Abbildung 3.4 als $\vec{w}_{ij}$ bezeichnet und in der gewichteten Distanz in Gleichung (3.4) verwendet wird. Für die Skalare v_{ij} kam ein heuristischer Ansatz zum Einsatz. Dazu wurden für jeden beteiligten Repräsentanten die Bildobjekte entsprechend der ermittelten Distanz der letzten Suchiteration sortiert. Die ähnlichsten N Bildobjekte werden bezüglich der abgegebenen Relevanzen untersucht. Alle für diese Bildobjekte abgegebenen Relevanzen werden aufsummiert und vergrößern oder verkleinern damit das Gewicht des betrachteten Repräsentanten. Die Gewichte der Merkmale werden nicht adaptiert. Durch diese Ebene besteht jedoch eine Trennung, so dass alle beteiligten Merkmale unabhängig von der zugehörigen Anzahl von Repräsentanten gleich gewichtet werden.

Indirekt in die Distanzberechnung geht die Adaption der Relevanzbewertung in die Verschiebung der Anfragevektoren q_{ij} ein (engl.: *Query vector movement*). Motiviert ist dieses Verfahren nach [Roc71] dadurch, dass durch die Auswahl eines Beispielobjekts ein guter, aber nicht unbedingt ein perfekter Satz von Anfragevektoren für einen Ähnlichkeitsvergleich zur Verfügung steht. Ausgehend von dem Originalvektor wird der Anfragevektor ein Stück weit zu den Vektoren der positiv bewerteten Bildobjekte hin- bzw. von den Vektoren der negativ bewerteten Objekte weg geschoben [Rui99a].

Fazit für den Systementwurf

Die einzusetzende Interaktion sollten auch im Hinblick auf die Verwendung von Modalitäten wie Sprache und Gestik sehr einfach gehalten werden. Die hier vorgestellte Abgabe von Relevanzen erfüllt genau diese Anforderung und bietet, wie aufgezeigt, eine ideale Grundlage für ein Systemlernen. Zu untersuchen bleibt, ob anders geartete Interaktionen mit relativen Bewertungen, beispielsweise die Abgabe eines Urteils über zwei Bilder, welches der beiden die Suchintention besser widerspiegelt, ebenso leistungsfähig und effizient einzusetzen sind.

Bezüglich des Systemlernens soll das System möglichst offen bleiben. Der hier vorgestellte Systemansatz, der erstmalig im MARS-System eingesetzt wurde, bildet mit dem der hierarchischen Distanzberechnung zugrunde liegenden Objektmodell eine hervorragende Grundlage für Eigen- und Weiterentwicklungen von Adaptionsmechanismen. Vielversprechend, aber mit hohen Ansprüchen an das verarbeitende System ist vor allem die parametrisierbare Distanzberechnung, die anschaulich zu einer Deformierung der Vektorräume führt. Die Erweiterung der allgemeinen Distanzberechnung auf den generalisierten euklidischen Abstand, wie in Gleichung (3.5) dargestellt, schafft weitere Flexibilität für das Systemlernen. Die Erstellung eines Index zur schnelleren Distanzberechnung wird mit dem Einsatz der parametrisierten Distanzberechnung jedoch äußerst fragwürdig.

$$d_{\vec{r},\vec{q}} = \sqrt{(\vec{r} - \vec{q})^T \, \underline{W} (\vec{r} - \vec{q})} \tag{3.5}$$

3.3 Daten-, Last- und Funktionalitätsverteilung

Generell ist es bei einem Datenbanksystem wünschenswert, dies als datenverteiltes System zu realisieren, da es sich im Regelfall um große Datenmengen handelt, die vom System verwaltet werden sollen. Würde dem nicht entsprochen, liefe dies auf eine redundante Datenhaltung hinaus, sofern unterschiedliche bzw. gleichzeitig laufende Anwendungen auf die Daten zugreifen sollen. Datenbanksysteme werden in erster Linie als Client-Server-Systeme ausgelegt, bei denen ein Server existiert, der die Daten direkt verwaltet. Der Server ermöglicht es dann einem oder mehreren Clients, geordnet auf die verwalteten Daten zuzugreifen.

Je nach Komplexität des Datenzugriffs entsteht dadurch auf Seiten des Servers eine nicht unerhebliche Rechenlast, deren Größe mit der Anzahl gleichzeitiger Zugriffe der angemeldeten Clients steigt. Des Weiteren muss der Server dafür Sorge tragen, den Datenzugriff so zu ordnen, dass keinerlei inkonsistente Zustände durch datenverändernde Zugriffe der unterschiedlichen Clients entstehen können. Sind alle gleichzeitigen Zugriffe nur in Ausnahmefällen datenverändernd, dann kann die Systemperformanz durch den Einsatz paralleler Programmiertechniken auf Client-Ebene verbessert werden. Die Parallelisierung ist in diesem Fall sehr einfach durchzuführen, da es dann keinerlei Datenabhängigkeiten zwischen den Aufrufen unterschiedlicher Clients geben kann.

Auch bei der Realisierung eines Forschungssystems bietet es sich an, die oben angeführte Architektur zu verwirklichen. Je nach Komplexität der Rechenlast erzeugenden Algorithmen kann es sinnvoll sein, zumindest die Client- und Server-Funktionalität auf unterschiedlichen Rechenknoten auszuführen, um System-Antwortzeiten in annehmbaren Größen zu halten. Durch die

Trennung in Client-Server-Architektur wird es möglich, Clients mit unterschiedlichem funktionellen Fokus zu erstellen. Die klar spezifizierte Schnittstelle zwischen Client und Server begünstigt nicht allein eine bezüglich der Fehleranfälligkeit reduzierte Client-Entwicklung, sondern ermöglicht den Einsatz unterschiedlicher Programmiersprachen für Client und Server und damit ebenso die Nutzung des Internets als Verbindungskanal zwischen Client und Server. Damit entstehen beispielsweise Einsatzmöglichkeiten für Datenbank-Clients, die weltweit benutzbar sind.

3.3.1 Client-Server-System

Setzt man die Realisierung eines Client-Server-Systems voraus, dann stellt sich primär die Frage, wie die Verteilung der Funktionalität und der damit fest gekoppelten Daten zwischen Client und Server vorgenommen wird. Soll der Datenbank-Server eine iterative Suche zur Verfügung stellen, dann müssen neben den eigentlichen Bilddaten noch Daten, die die aktuelle Suche betreffen, verwaltet werden. Bei der Entscheidung bezüglich der Funktionalitätsverteilung sind mehrere Aspekte zu berücksichtigen.

Server-Integrität: In keinem Fall darf der Client in der Lage sein, den Server durch ungültige Parameterdaten so zu beeinflussen, dass es zu einem Systemabsturz kommen kann, denn ein Systemabsturz würde alle an dem Server angemeldeten Clients beeinflussen. Sicherheit kann in diesem Fall geschaffen werden, indem der Server alle ihm übertragenen Daten sowohl auf Konsistenz als auch auf Datengültigkeit überprüft. Der dazu notwendige Aufwand ist umso höher, je komplexer die Berechnung, die zur Gewinnung der Daten durchgeführt werden muss, ist. Unter Umständen ist eine Überprüfung nicht durchführbar.

Gemeinsame Funktionalität: Besteht die Anforderung, unterschiedlich geartete Datenbank-Clients zu betreiben, dann gilt es, ein Optimum zwischen gemeinsam benötigter Funktionalität, die in der Server-Applikation plaziert ist, und der Komplexität der zwischen Client und Server ausgetauschten Daten zu finden.

Übertragene Datenmenge: Generell ist die zwischen Client und Server übertragene Datenmenge so klein wie möglich zu halten, um Antwortzeiten nicht unnötig zu erhöhen.

Fazit für den Systementwurf

Die besondere Funktionalität Bildsuche, die der Datenbank-Server zur Verfügung stellen soll, ist eine interaktive und iterative Suche, wobei eine interne Gewichtung gelernt werden soll, um sukzessive das Suchergebnis zu verbessern. Durch diese Anforderung besteht der innere Zustand einer Suche, repräsentiert durch entsprechende Datenstrukturen, nicht allein aus einer geordneten Liste von Referenzen auf die Bilder des Suchergebnisses, sondern umfasst alle nötigen Daten, die für die Adaption bzw. das Lernen notwendig sind. Die hierfür benötigten Datenstrukturen hängen stark von den zu benutzenden Adaptionen bzw. Lernverfahren ab. Um die Flexibilität in einem der wichtigsten Forschungsschwerpunkte des Systems zu erhalten,

sollten bezüglich der eingesetzten Verfahren zum Zeitpunkt des Systementwurfs keinerlei Festlegungen gemacht werden, und die benötigten Datenstrukturen werden als beliebig komplex angesetzt.

Aus den oben angeführten Aspekten ergibt sich daher, dass der Server in der Lage sein muss, für jeden angemeldeten Client alle zur Suche zugehörigen Daten vorzuhalten. Diese Vorgehensweise hält die Menge der auszutauschenden Daten gering und ermöglicht erst die vollständige Überprüfbarkeit der Parameterdaten, die im anderen Fall alle Daten, die den aktuellen Zustand der Suche beschreiben, beinhalten würden. Diese Vorgehensweise verwirklicht eine Kapselung, so dass nur die Daten nach außen sichtbar und freigegeben werden, deren Interpretation und Veränderung sinnvoll durchgeführt werden kann. Als Parameterdaten sollen daher lediglich Referenzen der Bildobjekte und andere, die iterative Suche betreffende Daten, wie beispielsweise Bildobjektbewertungen, ausgetauscht werden. Das Übertragen von Vektoren in binärer oder textueller Form soll nur vom Server zum Client zur Überprüfung und nicht für den normalen Betrieb erlaubt sein.

3.3.2 Kommunikationssystem

Die Aufteilung des Systems in eine Client- und eine Server-Applikation erfordert neben einer eindeutigen Spezifikation der Server-Dienste auch den Einsatz eines Kommunikationskanals bzw. Kommunikationssystems. Für die Realisierung der benötigten Kommunikation existieren unterschiedlich komplexe Implementierungen, auf die bei der Entwicklung des Datenbanksystems zurückgegriffen werden kann. Man unterscheidet hier relativ simpel gehaltene Programmbibliotheken, die lediglich das Erzeugen und die Bedienung eines Kommunikationskanals zur Verfügung stellen und komplexe Systeme, die die Kommunikation so weit abstrahieren, dass es für den Entwickler nicht von Bedeutung ist, wo die unterschiedlichen Funktionen ausgeführt und über welche Kanäle Parameter und Ergebnisdaten an die und von der ausführenden Stelle übermittelt werden.

Im Folgenden werden drei Konzepte vorgestellt, die in die nähere Wahl als Kommunikationssystem zwischen Client und Server gekommen sind.

Kommunikation basierend auf TCP/IP

Ohne Einschränkung der Benutzung im Internet (IP-Protokoll; Schicht 3 ISO/OSI Referenzmodell) sollte eines der Standard-Netz-Protokolle der Kommunikation zwischen Client und Server zugrunde liegen. Welches der auf IP basierenden Protokolle zur Verwendung kommt, hängt von der Art des zu erwartenden Netzverkehrs zwischen diesen beiden Applikationen ab. Die geplante Interaktivität des Datenbanksystems ist dafür verantwortlich, dass der Server über einen gewissen Zeitraum hinweg Anfragen von einem Client beantworten muss. Dieser Zeitraum ist durch die Zeit, die benötigt wird, um zum Sucherfolg zu gelangen, bestimmt. Das beschriebene Verhalten wird durch eine verbindungsorientierte Kommunikation, bei der zu Anfang ein Kommunikationskanal aufgebaut und dieser bis zur Beendigung des Clients aufrecht erhalten wird, begünstigt.

Ein verbindungsorientierter Kanal basierend auf dem IP-Protokoll wird durch das TCP-Protokoll (Schicht 4 ISO/OSI Referenzmodell) realisiert. Im Gegensatz zu dem verbindungslosen Protokoll UDP stellt TCP eine gesicherte Verbindung zur Verfügung, bei der sichergestellt wird, dass gesendete Pakete den Adressat erreichen. Tritt ein schwerwiegender Fehler auf, der nicht bzw. nicht in einem gewissen Zeitrahmen vom System selbständig behoben werden kann, zum Beispiel der Fall, dass die Netzverbindung unterbrochen wurde, so wird der Sender des Pakets davon sofort in Kenntnis gesetzt. Eine so beendete Verbindung sollte die Applikation entsprechend behandeln können und in einen sicheren Zustand zurückkehren.

Abbildung 3.5 zeigt den Ablauf der notwendigen Systemaufrufe für den Auf- und Abbau der Verbindung und die eigentliche Kommunikation. Die Koordination der Benutzung des Kom-

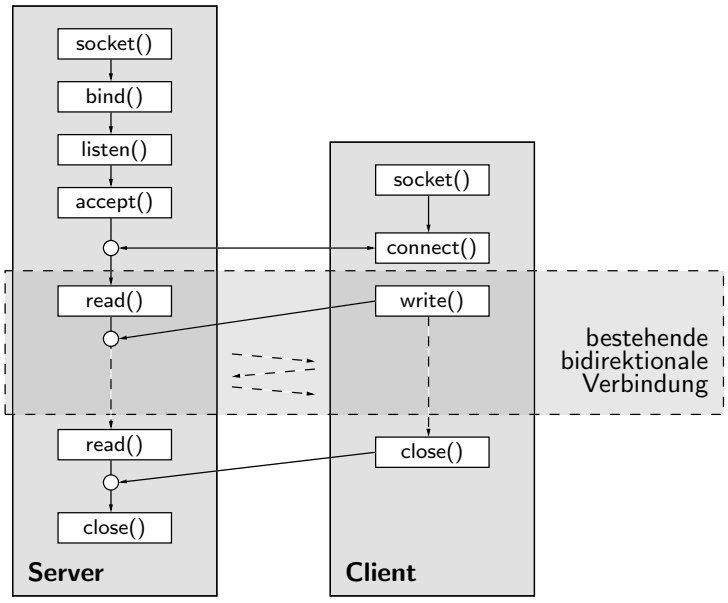
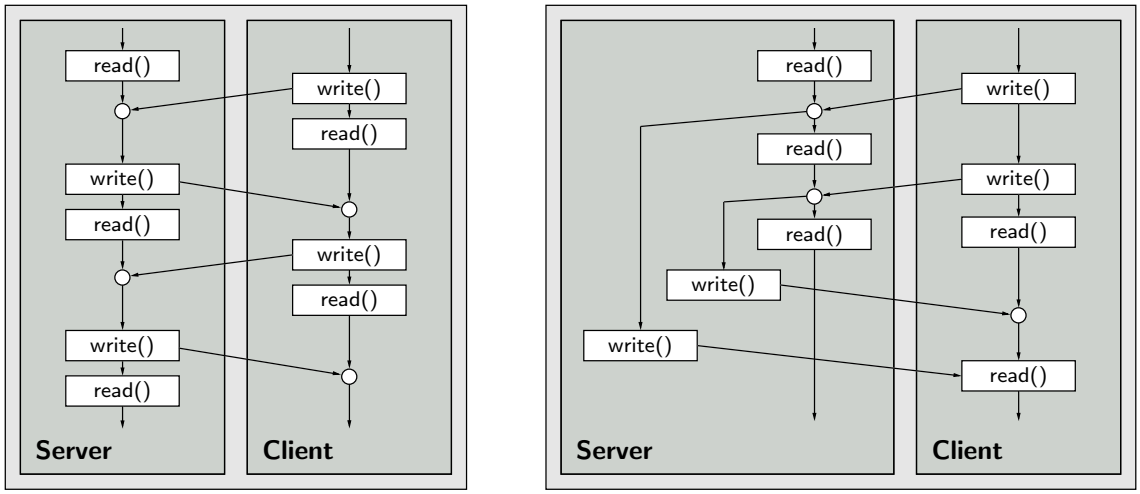


Abb. 3.5: **TCP/IP Socket Client-Server-Verbindung (nach [Ste98]):** Dargestellt ist ein vollständiger Zyklus einer TCP/IP-Verbindung auf Systemaufrufebene. Nach dem Verbindungsaufbau, für den der Client IP-Adresse und Port für das Adressieren des Servers angeben muss, besteht eine bidirektionale Verbindung, bei der beide Seiten einen Datenaustausch initiieren können. Beendet wird die Verbindung, indem einer der Kommunikationspartner den Kanal schließt.

munikationskanals ist ein weiterer wichtiger festzulegender Punkt. Das Client-Server-Konzept legt die Arbeitsweise des Datenbank-Servers fest. Diese Server-Applikation nimmt eine Anforderung eines Clients entgegen, bearbeitet diese und sendet eine Antwort an die Adresse des Clients zurück. Man unterscheidet hier zwei mögliche Betriebsmodi abhängig davon, ob der Server in der Lage ist, während der Bearbeitung einer Anforderung weitere Anforderungen entgegen zu nehmen und zu bearbeiten. Abbildung 3.6 zeigt eine synchrone und einen asynchrone Koppelung von Server und Client in je einer Beispielsequenz.

Der synchrone Betriebsmodus ist einfach zu realisieren und sicher, da auf der Seite des Servers, bedingt durch einen aktiven Client, keinerlei Datenabhängigkeiten zwischen mehreren Anforderungen bestehen. Der asynchrone Betriebsmodus hat jedoch den Vorteil, dass der Client



(a) Synchrone Auftragsbearbeitung

(b) Asynchrone Auftragsbearbeitung

Abb. 3.6: **Koppelung der Client- und Server-Prozesse:** In dem synchronen Betriebsmodus bearbeitet der Server anfallende Anfragen sequentiell. Dies ist eine sichere Arbeitsweise, die nachvollziehbar ist. In dem asynchronen Betriebsmodus führt der Server eine parallele Bearbeitung der Anforderungen durch (dargestellt durch parallele senkrechte Pfade). Durch geschickte Organisation der Kommunikation bzw. durch Nebenläufigkeit (hier nicht dargestellt) auf Seiten des Clients kann die Zeit, die der Server zur Erstellung der Antworten benötigt, als Rechenzeit genutzt werden. Im Fall der synchronen Kommunikation wird diese Zeit durch einheitliche Verarbeitung der Kommunikationsaufträge meist nicht genutzt.

nicht grundsätzlich auf die Antwort des Servers warten muss, sondern gleich nach Absenden eines Auftrags mit weiteren Berechnungen fortfahren kann. In diesem Modus gibt es zwei grundsätzliche Gegebenheiten, die bei der Implementierung zu berücksichtigen sind:

1. Die Nebenläufigkeit der Auftragsbearbeitung ermöglicht es, dass Antworten des Servers nicht in derselben Reihenfolge wie die der erteilten Aufträge zurückkommen. Das heißt für die Implementierung des Clients, dass die Möglichkeit bestehen muss, die Antworten des Servers nachträglich zu den Aufträgen an den Server zuzuordnen zu können, um die Ergebnisdaten richtig interpretieren zu können. Ebenso ist der Zugriff auf den Kommunikationskanal für das Übermitteln der Antworten zu synchronisieren.

2. Durch die Nebenläufigkeit der Aufträge muss Server seitig eine Synchronisation bezüglich des Datenzugriffs erfolgen. Datenverändernde Aufträge müssen exklusiv bearbeitet werden, um inkonsistente Datenstände zu vermeiden. Genauer heißt dies, dass datenverändernde Aufträge nur dann gestartet werden dürfen, wenn kein anderer Auftrag noch aktiv ist, und lesende Aufträge nur gestartet werden dürfen, wenn kein datenverändernder Auftrag in Bearbeitung ist.

Remote Procedure Calls

Remote Procedure Calls (RPC) wurden in den achtziger Jahren erstmalig vorgestellt. Birrell und Nelson hatten Anstoß daran genommen, dass die Struktur von Client und Server eines verteilten Systems immer wieder um die zentrale Kommunikationskomponente aufgebaut wurde. Dem Entwickler sollte mit diesem System ein Werkzeug in die Hand gegeben werden, mit dem Aufrufe von Prozeduren auf entfernten Servern denen von lokalen Prozeduren gleichen, so dass ein eventuell zugrunde liegender Netzzugriff vollkommen transparent bleibt.

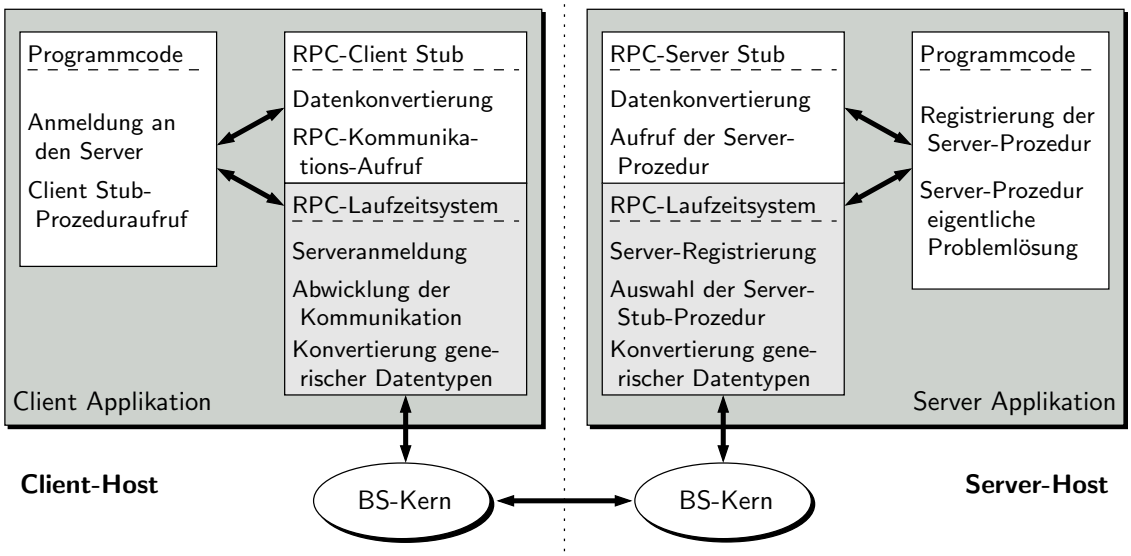


Abb. 3.7: **Kommunikation mittels RPC:** Zur Initialisierung eines RPC-Systems registriert zunächst der Server eine Prozedur bei dem RPC-System. Dann kann sich der Client auf die entsprechende Prozedur anmelden. Der eigentliche Prozeduraufruf über den neu geschaffenen Kommunikationskanal wird durch den Aufruf der lokalen Client-Stub-Prozedur initiiert (engl.: *Stub* = Stummel, Stumpf). Die Stub-Prozedur führt eine Datenkonvertierung durch und veranlasst das Versenden des Anfragepakets vom RPC-Laufzeitsystem. Die Kommunikation auf Basis des TCP/IP- oder UDP/IP-Protokolls wird von den beteiligten Betriebssystemen durchgeführt. Auf der Seite des Servers wird das Paket von dem RPC-Laufzeitsystem aufgenommen und von dem Dispatcher ermittelt, welche angemeldete Server-Stub-Prozedur für das Ausführen der Anfrage aufgerufen werden muss. Die Stub-Prozedur führt wiederum eine Datenkonvertierung durch und ruft die eigentliche Server-Prozedur auf. Die generierte Antwort wird zurückgeschickt, wobei die Datenkonvertierung in umgekehrter Reihenfolge stattfindet.

Im Laufe der Zeit wurden mehrere Implementierungen von RPC vorgestellt. Die Implementierung vom Sun Microsystems, ONC-RPC (Open Network Computing - Remote Procedure Calling), oft auch als SUN-RPC bezeichnet [Ste99], hat sich weitestgehend durchgesetzt und ist unter den meisten Betriebssystemen verfügbar. Die Arbeitsweise eines solchen Client-Server-Systems ist in Abbildung 3.7 dargestellt.

Der Server, der zu Beginn gestartet wird, registriert alle integrierten Prozeduren bei dem RPC-System auf dem Server-Host. Meldet sich nun ein Client auf eine der Server-Prozeduren an, so wird vom RPC-System eine Verbindung aufgebaut, dessen Typ, UDP oder TCP, vom Client vorgegeben wird. Steht die Verbindung, dann kann über diese Verbindung die zuvor angegebene Prozedur aufgerufen werden, so wie es in der Abbildung 3.7 aufgezeigt ist.

Die in der Abbildung dargestellte Datenkonvertierung ist in Schicht 6, der Darstellungsschicht, des ISO/OSI Referenzmodells anzusiedeln. Hier wird dafür gesorgt, dass binär übertragene Daten in die richtige Datenrepräsentation der unterschiedlichen Rechnerarchitekturen gewandelt werden (siehe auch Abschnitt 6.1). Jede Parameterstruktur, die als Übergabe- bzw. Rückgabestruktur benutzt wird, benötigt eine eigene Funktion, die die Datenkonvertierung in eine externe und systemübergreifend einheitliche Repräsentation durchführt. Da das Erstellen solcher Funktionen von Hand fehleranfällig ist, existiert dafür im RPC-Paket ein Interface-Pre-Compiler. Dieser Pre-Compiler erhält als Eingabe eine Spezifikations-Datei (engl.: *RPC Specification File*). Diese Datei enthält neben dem Namen und der Beschreibung der Parameterstrukturen auch Informationen, die die konkrete RPC-Schnittstelle des Servers eindeutig identifizierbar machen. Als Ergebnis liefert der Pre-Compiler die benötigten Funktionen für die Datenkonvertierung in einer C-Datei und eine C-Header-Datei mit der Definition der Strukturen und allen Funktionsprototypen. Des Weiteren werden Client- und Server-Stub-Funktionen ebenfalls in eigenen C-Dateien erzeugt, die beim Erzeugen der entsprechenden Applikationen mit eingebunden werden müssen.

Auch wenn die Client-Server-Kommunikation mittels Prozeduraufruf einen nicht verteilten Charakter aufweist und der prozeduralen Programmierung entspricht, kann ein RPC-System nicht ganz verbergen, dass es sich um besondere Prozeduren handelt, die dabei aufgerufen werden. Das Zusammenstellen der Parameter einer RPC-Prozedur, die für die Ein- und Ausgabe je in einer Struktur gekapselt werden müssen, stellt eine Besonderheit dar. Zeiger können nicht als Parameter übergeben oder zurückgegeben werden, weil sie in dem Prozessraum des anderen Prozesses keinerlei Sinn ergeben, und globale Variablen existieren ebenfalls nicht über Prozessgrenzen hinweg, so dass deren Verwendung ebenfalls ausfällt. Der Prozedurale Charakter der RPCs wird klar durch einen synchronen Betrieb des Client-Server-Systems umgesetzt (siehe Abbildung 3.6), indem der Client-Prozess blockiert, bis er die Antwort auf die Anfrage vom Server erhalten hat.

Das Prinzip des RPC-Systems wurde bisher anhand einer Server-Prozedur gezeigt. Wie verhält sich das System jedoch, wenn ein Client sich auf mehrere Prozeduren eines Servers anmeldet? Das RPC-System baut für jede Prozedur, auf die sich ein Client anmeldet, eine eigene Verbindung auf. Damit kann die Anzahl gleichzeitig bestehender Verbindungen zu einem Server ungewünscht hoch werden, wenn der Server entweder viele Prozeduren anbietet, die von einem Client alle benötigt werden, oder wenn viele Clients gleichzeitig vom Server bedient werden müssen. Die Anzahl der gleichzeitig bestehenden Verbindungen kann nur so reduziert werden, dass der Client die Verbindung nach jedem Aufruf sofort wieder abbaut bzw. das verbindungslos Protokoll UDP für die Aufrufe verwendet.

Ein entscheidender Nachteil von ONC-PRC, der zur Zeit nur in den Implementierungen vom SUN-Solaris-Betriebssystem beseitigt wurde, ist der, dass der Server nicht in der Lage ist, mehrere Anforderungen unterschiedlicher Clients gleichzeitig zu bearbeiten. Alle Anfragen, Prozeduraufrufe, an einen Server werden in der Reihenfolge des Eintreffens sequentiell abgear-

beitet. Damit beeinflussen sich die beim Server angemeldeten Clients unmittelbar. Ein Client muss also auch dann auf die Abarbeitung eines gerade aktiven Prozeduraufrufs warten, wenn die Prozedur gerade auf ein Betriebsmittel vom Betriebssystem wartet, denn der gegenseitige Ausschluss besteht auf Aufrufebene.

CORBA

CORBA (Common Object Request Broker Architecture) [Sie99] geht in seinen Anforderungen viel weiter als die bisher vorgestellten Ansätze. Ende der achtziger Jahre bildete sich die OMG (Object Management Group) deren Ziel es war, Standards für verteilte Anwendungen in heterogenen Umgebungen zu entwickeln und zu publizieren. Die Grundlage für dieses Bestreben bildete die Tatsache, dass aus technischen und wirtschaftlichen Gründen unterschiedlichste Rechnerarchitekturen (von Mainframe bis zu eingebetteten Systemen zur Steuerung von Produktionsanlagen) zusammenarbeiten und unter Verwendung aller erdenklichen Protokolle miteinander kommunizieren müssen. Dabei sollte vor allem auch die Möglichkeit geschaffen werden, bereits bestehende Systeme mit einfachen Mitteln zusammenarbeiten zu lassen. Ein solches Kommunikationssystem wird im Allgemeinen als Verteilungsplattform bzw. als Middleware bezeichnet und kann aufgrund der Art, wie es sich dem Entwickler einer Applikation präsentiert, als Erweiterung der Betriebssystem-Funktionalität aufgefasst werden.

Wie der Name CORBA bereits andeutet, handelt es sich um ein objektorientiertes Konzept der Kommunikation, bei dem grundsätzlich ein Objekt referenziert wird, das dann entsprechende vom Client gewünschte Funktionalität ausführt und Ergebnisse zurückliefert, also ebenfalls wie RPC nach dem Request/Reply-Prinzip funktioniert. Für die Kommunikation des Clients mit dem referenzierten Objekt ist der Object Request Broker (ORB) zuständig. Er nimmt die Aufträge vom Client entgegen und leitet sie an einen Server weiter, wobei der ORB über Techniken verfügt, den passenden Server aufzufinden. Die vom Server gelieferten Ergebnisse werden dann dem Client zurückgeliefert. Da Server und Client auf unterschiedlichsten Betriebssystemen laufen können, unternimmt der ORB auch die Datenkonvertierung in die externe bzw. in die maschinenspezifische Datenrepräsentation. Ebenso werden unterschiedliche Transportprotokolle (ISO/OSI-Schicht 4) unterstützt.

Da die OMG lediglich Standards festlegt, entstanden eine Reihe verschiedener CORBA-Implementierungen. Damit die ORBs verschiedener Implementierungen miteinander kommunizieren können, müssen sie in der Lage sein, ORB-spezifische Daten untereinander auszutauschen. Hierfür beinhaltet der CORBA-Standard ab Version 2.0 das General Inter-ORB Protocol (GIOP) das Formate und Transfer-Syntax für den Austausch festlegt. Dieses Protokoll ist nicht vom benutzten Transportprotkoll abhängig. Für eine gemeinsame Basis legt der CORBA-Standard jedoch fest, dass jede CORBA-Implementierung die Inter-ORB-Kommunikation über TCP/IP, das Internet-ORB-Protocol (IIOP), unterstützen muss.

CORBA ist nicht wie RPC an eine Programmiersprache gebunden. Um das zu erreichen, wurde eine im Standard definierte Sprach-Umsetzung (engl.: *Language Mapping*) festgelegt. Für die Definition der Schnittstellen der Objekte unabhängig von Programmiersprachen wurde eine Schnittstellen-Beschreibungssprache (engl.: *Interface Description Language, IDL*) spezifiziert, mit der im Wesentlichen die Objekte durch Angabe der Attribute und der Operationen mit deren Parameter spezifiziert werden.

Der Precompiler erzeugt, wie beim RPC-System auch, eine ganze Reihe von Dateien, die zur Verwendung bzw. zur Weiterentwicklung anstehen. Abbildung 3.8 zeigt den Vorgang der Entwicklung eines CORBA-Client-Server-Systems. Neben der Möglichkeit, so genannte stati-

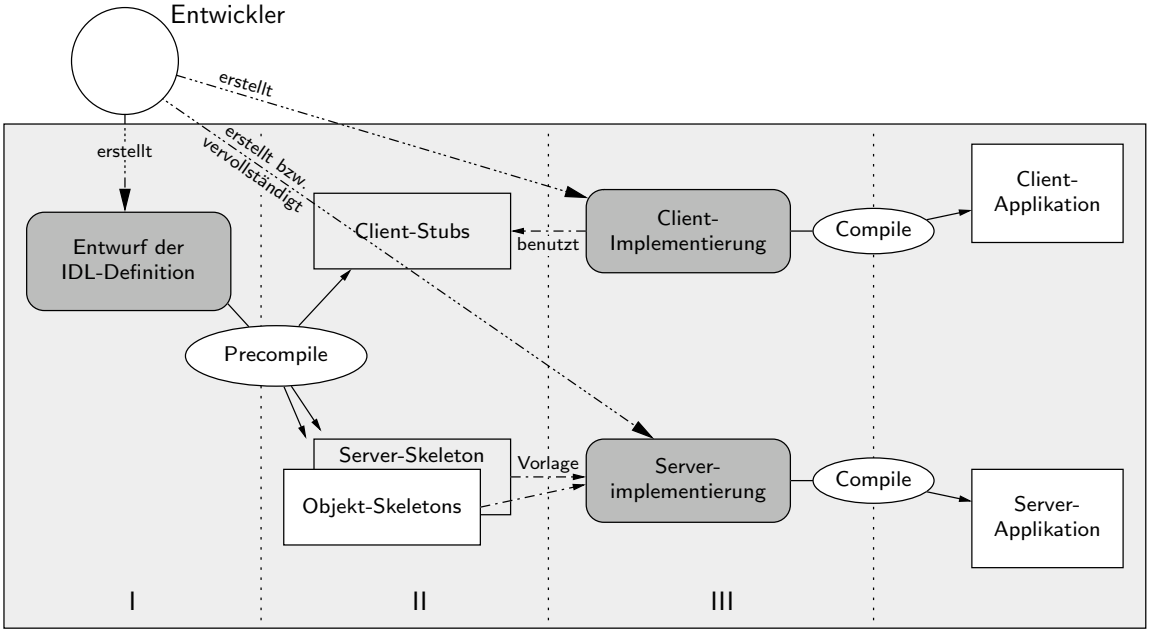


Abb. 3.8: **Entwicklung eines Client/Server-Systems mit CORBA (nach [Lan03]):** Die Entwicklung unterteilt sich in drei Phasen. Zunächst werden die Objekt-Schnittstellen in der IDL spezifiziert. Durch einen anschließenden Precompiler-Aufruf für die entsprechende Programmiersprache werden auf der Seite des Clients Stubs, über die der Zugriff auf die CORBA-Objekte erfolgt, zur Verfügung gestellt. Server-seitig werden zwei Dateitypen erzeugt. Zum einen wird ein Beispiel-Server erstellt, der als Entwicklungsgrundlage dienen kann, und zum anderen werden Dateien mit je einem Gerüst (engl.: *Skeleton* = Skelett) aller von den Objekten zur Verfügung gestellten Funktionen erzeugt, die entsprechend der Funktionalität zu vervollständigen sind. Sind in Phase III alle Objektfunktionen sowie der Code für die Server- und Client-Applikation erstellt, dann können anschließend die Applikationen erzeugt werden.

sche Aufrufe von Objektfunktionen zu implementieren, bei denen die Objekt-Schnittstellen zur Entwicklungszeit zu spezifizieren sind, damit sowohl Client als auch Server Kenntnis darüber haben, wie die übermittelten Pakete zu erstellen bzw. zu interpretieren sind, können auch dynamische Aufrufe durchgeführt werden. Die Funktionsparameter werden bei dieser Art des Aufrufs durch einen Namen referenziert.

Um diese Zuweisungen von Werten zu Parameternamen (engl.: *Named Values*) durchführen zu können, führt der CORBA-Server ein Interface Repository, das die Schnittstellen-Definitionen aller angemeldeten Objektfunktionen beinhaltet. Die Schnittstellen bei einem dynamischen Funktionsaufruf (auch statische Funktionen können dynamisch aufgerufen werden) sind also unabhängig von der aufgerufenen Funktion immer identisch.

Die Leistungsfähigkeit von CORBA geht weit über ein einfaches Client-Server-System hinaus. Die Stärken von CORBA kommen dann zum Tragen, wenn es sich um ein komplexes System handelt, bei dem die unterschiedlichen Applikationen aufgrund der Verteilung der Objekte bzw. der Funktionalität sowohl Client- als auch Server-Charakter aufweisen. Heterogene Systeme, die mit unterschiedlichen Programmiersprachen erstellt wurden, können durch den Einsatz von CORBA gemeinsam eine komplexe verteilte Anwendung realisieren.

Fazit für den Systementwurf

In Anbetracht der Tatsache, dass die Kommunikation zwischen den verteilten Applikationen nicht im Mittelpunkt der Systementwicklung stehen sollte, bieten die hier vorgestellten Kommunikationssysteme RPC und CORBA eine sehr interessante Lösung. Die Basis-Codestruktur wird bei beiden Systemen automatisch erzeugt, und der Entwickler kann sich auf das Implementieren der eigentlich wichtigen Funktionalität beschränken. Mit einer solchen Wahl wird die Entwicklung jedoch stark von dem benutzten Kommunikationssystem abhängig. Bei der Wahl von CORBA beispielsweise sollte die gesamte Softwareentwicklung in einer objektorientierten Programmiersprache erfolgen, da Abweichungen vom eigentlichen Konzept, hier dem Arbeiten mit Objekten, schnell zu einer unübersichtlichen bzw. unverständlichen Lösung führen können. Wird der Betrieb eines Servers mit mehreren gleichzeitig angemeldeten Clients gewünscht, so ist der Einsatz des RPC-Systems nur bedingt möglich. Da es bei einem solchen Betrieb unbedingt erforderlich ist, dass die Abarbeitung der Clients entkoppelt durchgeführt wird, zum Beispiel durch Benutzung nebenläufiger Threads auf dem Server, wäre man hier mit der Entwicklung auf ein SUN-Solaris-Betriebssystem einer neueren Generation festgelegt.

Wird gänzlich auf ein Kommunikationssystem verzichtet, so kann die Entwicklung ohne Einschränkungen durchgeführt werden. Jedoch bekommt in diesem Fall die Entwicklung der Kommunikationsschnittstelle ein erheblich höheres Gewicht. So muss hier die Funktionalität der Darstellungsschicht (Schicht fünf des ISO/OSI Referenzmodells) mit in den Code aufgenommen werden. Ebenso gehören der Verbindungsauf- und -abbau zur Entwicklung der Applikation.

3.3.3 Datenhaltung

Unabhängig davon, ob die Zielapplikation eine monolithische Gestalt annehmen oder ein Client-Server-System werden soll, muss das System die zum Betrieb der Datenbank notwendigen Daten verwalten. Dabei handelt es sich hauptsächlich um die binären Bilddaten und die berechneten Merkmalsvektoren der Merkmalsrepräsentanten.

Für einen schnellen Betrieb mit großen Datenmengen sind effiziente Zugriffsverfahren zwingend notwendig. Des Weiteren soll das System, wie bereits angedeutet, bezüglich der verwendeten Merkmalsrepräsentanten und Bildsegmentierer einfach projektierbar sein. Das alles sind Gründe, die für den Einsatz eines bestehenden Datenbanksystems, das für die Organisation und den Zugriff auf die Betriebsdaten verwendet wird, sprechen, denn eine Eigenentwicklung steht thematisch nicht im Vordergrund des Forschungsprojekts.

Entsprechend Abschnitt 3.2.4 ist eine Hauptaufgabe der Bilddatenbank, in jedem Suchschritt parametrisierte Distanzen von den Anfragevektoren zu allen gespeicherten Vektoren eines jeden

Repräsentanten zu bilden. Um das zu bewerkstelligen, ist ein Zugriff auf alle gespeicherten Vektoren erforderlich. Diese Berechnung ist an sich bereits sehr aufwendig und sollte nicht durch unnötigen Datentransport erschwert werden. Daher bietet sich an, bei der Wahl der zu verwendenden Datenbanksoftware auf die Möglichkeit Wert zu legen, dass die Distanzberechnung vom Datenbank-Server selbst durchgeführt werden kann [Käs01].

Ideal für den allgemeinen Einsatz eines Bilddatenbank-Servers wäre die Möglichkeit, vom Client durch eine Erweiterung einer bestehenden Abfragesprache, beispielsweise SQL, die Ähnlichkeitssuche durchführen zu können, wobei die Suche nicht prinzipiell auf Bilddaten beschränkt wäre. Dann bestünde ebenfalls die Möglichkeit, Meta-Daten zu den sich in der Datenbank befindlichen Bildern zu speichern. Aber auch dieser Punkt ist kein Schwerpunkt des Projekts und die Funktionalität wäre lediglich bei einem breiten Einsatz des Datenbanksystems notwendig.

3.4 Modularität und Flexibilität

Eine der Haupt-Systemanforderungen ist die Modularität bzw. der flexible Aufbau des Systems. Die Erfüllung dieser Anforderung ist nicht allein mit einem Konzept umzusetzen.

An dieser Stelle sollen lediglich die Hauptaspekte herausgestellt werden. Punkte, die weiter ins Detail gehen, sind nicht allein in diesem Kapitel sondern vor allem auch in den Implementierungskapiteln 4 und 5 zu finden, da es sich oft um Implementierungsdetails handelt.

Merkmale/Repräsentanten: Die Ergebnisse der Bildsuche sind maßgeblich von der Wahl der Merkmale bzw. deren Repräsentanten abhängig. Je nach Einsatzgebiet der Datenbank, also dem zu verwaltenden Bildmaterial, sind unterschiedliche Repräsentanten besser bzw. schlechter geeignet. Damit direkt verbunden sind die verwendeten Bildsegmentierer und speziellen Distanzfunktionen. Die Implementierung dieser Funktionalitäten ist ideal von der eigentlichen Datenbank separierbar, da sich Parameter und Rückgabe sehr einfach spezifizieren lassen und lediglich einfache zugrunde liegende Datentypen festgelegt werden müssen. Gerade im Hinblick auf die gemeinsame Entwicklung und den Einsatz des Forschungsprototyps ist die Benutzung dieser Funktionalität projektierbar zu halten.

Mit diesem Konzept stellt sich das System als ideale Testplattform für die Entwicklung neuer allgemeiner bzw. spezieller Merkmale, Segmentierer und Distanzfunktionen dar.

Suchsystem: Das Suchsystem selber beinhaltet mit dem Thema „Systemlernen“ einen weiteren Haupt-Systemschwerpunkt. Da es durchaus im Bereich des Möglichen liegt, unterschiedliche Suchsysteme zu entwickeln bzw. unterschiedliche Lösungsansätze mit identischer Umgebung einander gegenüber zu stellen, gilt es, diesen Systemteil durch einfach gehaltene Schnittstellen und hierarchische Datenhaltung gut zu kapseln.

Datenbank-Client: Setzt man die Entwicklung eines Client-Server-Systems voraus, wie in Abschnitt 3.3 diskutiert, dann gilt es, die vom Server zur Verfügung gestellten Dienste einfach zu halten, um diesen von Clients mit unterschiedlichem Schwerpunkt benutzbar zu machen.

3.5 Multimodale und natürliche Interaktion

Die natürliche und intuitive Bedienung eines Bilddatenbanksystems ist eines der wichtigsten Ziele, das im INDI-Projekt verwirklicht werden soll. Auch hier ist dieses Ziel durch die Kombination einiger Teilziele zu erreichen. Aspekte wie die Erstellung der Suchanfrage und die Interaktion bei dem iterativen Suchprozess wurden bereits im Abschnitt 3.2 angesprochen. In diesen Fällen wird durch eine einfache und durchschaubare Interaktion versucht, technische Details zu kapseln und diese damit vom Benutzer fern zu halten, um Verwirrungen oder eine Überforderung zu vermeiden.

Die Verwendung der Modalitäten, die bei einer natürlichen zwischenmenschlichen Kommunikation eingesetzt werden, ist die Idee, um die Interaktion mit dem System zu vereinfachen, sie leichter durchschaubar zu gestalten und damit die Menge potentieller Systembenutzer zu vergrößern. Für die Realisierung dieses Ziels sollen Gestik und Sprache als Interaktionskanäle für die Steuerung des Systems verwendet werden. Die Systemreaktionen sollen auf dem Standardkanal Monitor dem Benutzer präsentiert werden.

Obwohl multimodale Systeme im Allgemeinen durch die Möglichkeit, Mehrdeutigkeiten besser auflösen zu können, robuster arbeiten [Ovi99], können falsch eingesetzte Modalitäten auch zu Verwirrungen und damit Ablehnung führen. Mit der Kombination der inhaltsbasierten iterativen Bildsuche und der multimodalen Bedienung des Systems ist die hier vorgestellte Entwicklung zur Zeit einzigartig, so dass bezüglich des genauen Einsatzes der unterschiedlichen Modalitäten auf keinerlei Erfahrung zurückgegriffen werden kann. Die bewusst einfach gehaltene Interaktion mit dem System verlangt, dass die folgenden Aufgaben unter Verwendung der genannten Kanäle möglichst einfach und intuitiv durchgeführt werden können.

1. Navigation durch die Bildmenge des Suchergebnisses bzw. der initial präsentierten Bilder und deren Bildregionen
2. Bewertung dedizierter Bildobjekte
3. Auslösen einer Suchiteration
4. Administrative Aufgaben, wie beispielsweise das Beginnen einer anderen Bildsuche

Mit der Erstellung der Suchanfrage, die jedoch bei manchen Systemen entfällt, weil sie implizit durch die Abgabe von Bewertungen durchgeführt wird (siehe zum Beispiel PicSOM auf Seite 8), sind es fünf Aktionsgruppen, die zu berücksichtigen sind.

Im Gegensatz zum Spracherkenner, der lediglich mit einem Lexikon auf den speziellen Einsatz in einer Bilddatenbank vorbereitet werden muss, ist die Gestalt der Gestenerkennung nicht so einfach festzulegen. Im Hinblick auf die oben aufgezählten Aufgaben können lediglich zwei Tätigkeiten, die mit den Händen bzw. Armen ausgeführt und mit einer oder mehreren Kameras beobachtet werden, als natürlich bezeichnet werden. Hierbei handelt es sich hauptsächlich um Zeigegesten zum Referenzieren von angezeigten Bildern und des Weiteren um formbeschreibende Gesten. Das in Abbildung 3.9 vorgestellte Szenario zeigt ein System, das die genannten Interaktionen erlauben würde. Der Benutzer der Datenbank steht vor einer Projektionswand,

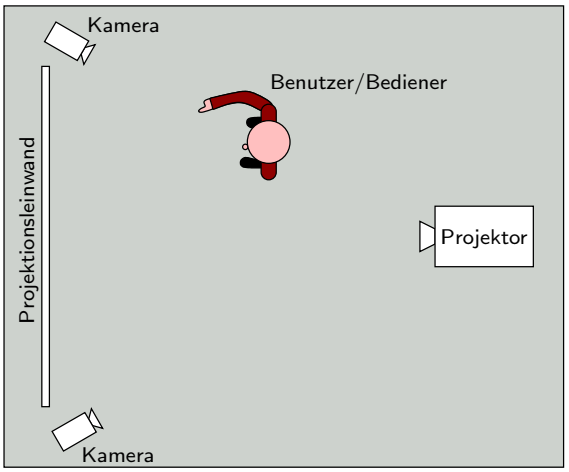


Abb. 3.9: **Erkennung natürlicher Gesten:** In diesem Szenario kann der Benutzer frei und ohne Tragen von technischen Hilfsmitteln mittels Gestik, die mit den Händen ausgeführt wird, mit dem System interagieren. Die Bedienungsoberfläche wird vom Projektor auf die Wand im linken Teil der Abbildung geworfen. Die neben der Projektionswand angebrachten Kameras liefern den zur Gestenerkennung notwendigen Datenstrom.

auf der die Datenbankoberfläche dargestellt wird. Die Tätigkeiten werden von zwei Videokameras beobachtet, die neben der Projektionswand in unterschiedlicher Höhe befestigt sind. Die Auswertung muss mit mindestens zwei Kameras erfolgen, um Ungenauigkeiten bei der Bestimmung des Zeigepunkts zu verkleinern. Die Gestenerkennung basiert im Allgemeinen auf der Erkennung von hautfarbenen Regionen und ist damit stark beleuchtungsabhängig. Das hier gezeigte System ist davon insbesondere betroffen, da der Benutzer im Schein des Lichts des Projektors steht. Dem kann durch den Einsatz einer Rückwandprojektionstechnik oder eines entsprechend großen Displays entgegengewirkt werden. In einem solchen Szenario kann das System nach entsprechender Kalibrierung freie Zeige-, Form- oder größenbeschreibende Gesten erkennen, ohne dass der Benutzer weitere technische Gegenstände am Körper anlegen muss.

Die Erstellung eines solchen Systems wird mit einem hohen Preis erkauft. Zum einen handelt es sich um die finanziellen Kosten der Anschaffung der zur Realisierung notwendigen technischen Geräte, zum anderen aber ist der Aufwand sehr hoch, die Robustheit der Gestenerkennung zu erhalten. Sollen mehrere Benutzer gemeinsam das System bedienen, ist der technische Aufwand wiederum erheblich höher.

Viel einfacher und mit wenigen Einschränkungen können die oben genannten Tätigkeiten durch den Einsatz eines Touchscreen-Displays erkannt werden. Eine Zeigegeste wird hier durch das Tippen auf die Display-Fläche realisiert, formbeschreibende Gesten können durch Anfertigen von Skizzen durchgeführt werden, wobei ein Finger des Benutzer hier als virtueller Stift fungiert. Die gleichzeitige Bedienung des Systems von mehreren Benutzern ist abgesehen von den meist deutlich kleineren Abmaßen eines Touchscreen-Displays ohne weiteren Aufwand möglich. Ein Touchscreen-Display ist bereits in ein gängiges Computer-System integriert und kann als Maus-Ersatz dienen. Durch diesen Aspekt motiviert, erscheint es durchaus sinnvoll, die Bedienoberfläche mit Standardbedienelementen an bestimmten Stellen zu ergänzen.

Einsatz der Modalitäten

Die eingesetzten Modalitäten sind bezüglich der einzelnen Interaktionen nicht gleichwertig. Bestimmte Aspekte lassen sich sprachlich einfach formulieren aber nicht mit einer Geste ausdrücken. Bei anderen Aspekten ist dies nahezu umgekehrt. Vor allem räumliche Gegebenheiten sind durch Zeigegesten besonders einfach auszudrücken wohingegen der Einsatz der Sprache hier erheblich umständlicher werden kann.

Die Erstellung der Suchanfrage ist an die Wahl des Suchsystems anzupassen. Bei einer Beispielsuche muss lediglich ein Beispielbild ausgewählt werden. Zur Lösung dieser Aufgabe bietet sich der klassische komplementäre Einsatz der Modalitäten Sprache und Gestik an. Während auf das gewünschte Bildobjekt gezeigt wird, erklärt der Benutzer, welche Aktion mit dem Bild durchgeführt werden soll. In diesem Fall ist das die Auswahl des Bildes als Beispielbild. Dementsprechend ist bei der Bewertung von Bildern zu verfahren.

Handelt es sich hier um ein Suchsystem, das mit Objekten, die zum Beispiel von einem Objekterkenner automatisch detektiert wurden, arbeiten kann, dann ist es möglich, die Suchanfrage durch die Erstellung einer Skizze durchzuführen, wobei diese Aktion durch eine entsprechende sprachliche Instruktion eingeleitet werden kann.

Die administrativen Aufgaben, das Durchführen einer Suchiteration und die Navigation in dem Suchergebnis, können nur durch eine Sprachinteraktion ausgelöst werden. Alternativ bietet es sich bei dem Einsatz eines Touchscreen-Displays an, für diese Aktionen Schaltflächen und andere Dialogelemente zur Verfügung zu stellen. Wenn zur Darstellung des Suchergebnisses keine Listenform, sondern beispielsweise eine zweidimensionale distanzgetreue Darstellung gewählt wird, könnte die Navigation durch einen virtuellen Flug durch den sich ergebenen Raum angeboten werden. Hierbei kann die Bewegungsrichtung durch Zeigegesten sowie Geschwindigkeit und Zoom durch sprachliche Eingaben festgelegt werden. Entgegen der ansonsten einzuhaltenden Einschränkung der dem Benutzer zuzumutenden Bildmenge können hier alle Bilder der Datenbank präsentiert werden.

Stärkung der Bedienung durch Gesten

Als alternative Interaktionen bieten sich bei dem Einsatz eines Touchscreen-Displays so genannte Touchscreen-Gesten zur Integration in das System an. Solche Gesten sind an das Ausfüllen von Formularen angelehnt. Bei der Auswahl der Lottozahlen beispielsweise wird die Selektion der Zahlen durch das Zeichnen eines Kreuzes auf der entsprechenden Zahl kenntlich gemacht. Diese Technik kann bei der Auswahl bestimmter Aktionen, die die Bilder betreffen, eingesetzt werden.

Dieses Angebot des alternativen Auslösens bestimmter Aktionen steigert die Flexibilität bei der Benutzung und kommt Vorlieben der Benutzer entgegen. Die Interaktion am Touchscreen gewinnt aber vor allem durch den Einsatz dieser Technik, denn für die in Kapitel 7 beschriebene Evaluation, sollte das System unter anderem ausschließlich in dieser Modalität betrieben werden.

Referenzieren von Regionen

Da das Datenbanksystem nicht allein mit ganzen Bildern, sondern auch mit Bildobjekten arbeiten soll, muss die Möglichkeit geschaffen werden, sowohl Bewertungen als auch das Selektieren des Beispielobjekts auf der Menge der Teilbilder durchführen zu können. Bezüglich der Bildobjekte soll keinerlei Einschränkung gemacht werden, so dass Objekte existieren dürfen, die sich überschneiden. Mit dieser Festlegung ist das Referenzieren eines Bildobjekts nicht durch eine einfache Zeigegeste zu realisieren. Vielmehr soll das System in der Lage sein, durch die Angabe von bestimmten wenigen Attributen einer Bildregion diese zu referenzieren. Das entspricht dem natürlichen Vorgehen, wenn zur Beschreibung keinerlei Objektwissen eingesetzt werden darf. Als Attribute können zum Beispiel Form, Größe oder Farbe eingesetzt werden, die dem System sprachlich aufgezählt werden.

3.6 Gesamtsystem

Die Überlegungen und Gegenüberstellungen der vorherigen Abschnitte dieses Kapitels führen zu der Gesamtarchitektur, wie sie in Abbildung 3.10 gezeigt ist. Aufgrund der idealen

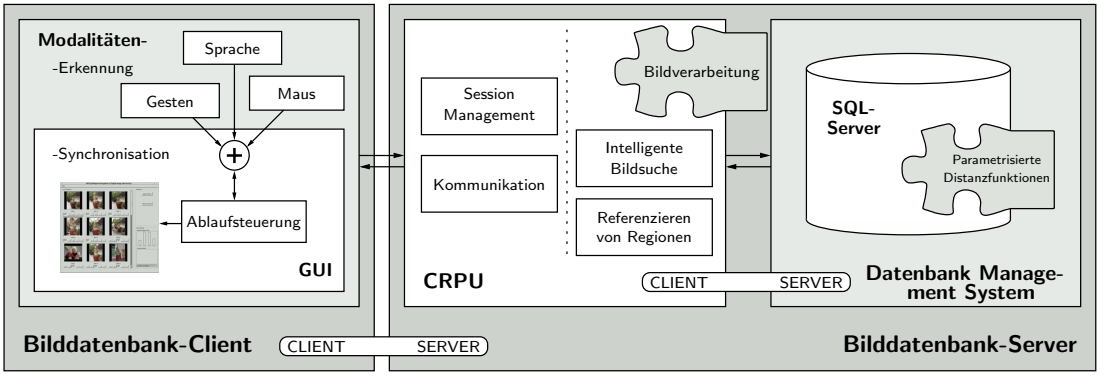


Abb. 3.10: **Gesamtarchitektur des Bildsuchsystems INDI:** Das System teilt sich in drei Teile auf. Auf der linken Seite ist die Benutzerschnittstelle mit der Verarbeitung der Modalitäten und dem zentralen Steuerelement zu finden. Dieser Teil, der Bilddatenbank-Client, kommuniziert mit der eigentlichen Bildsucheinheit in der Mitte. Zusätzlich zu den administrativen Server-Eigenschaften weist diese Applikation eine Bildverstehenskomponente auf, die zur Referenzierung von Bildregionen herangezogen wird. Die Bildverarbeitungsmodule zur Segmentierung und Merkmalsbildung sind ebenfalls in diesen Teil des Systems integriert. Die angeschlossene SQL-Datenbank soll die Datenverwaltung durchführen. Bezüglich der reinen Datensicherung ist sie um die parametrisierte Distanzberechnung erweitert. Die beiden Applikationen im rechten Teil der Abbildung bilden zusammen den Bilddatenbank-Server

Möglichkeiten des Systemlernens mit der damit verbundenen natürlichen Art der Interaktion fiel die Entscheidung, wie die Suche durchgeführt werden soll, auf eine Beispielsuche mit der Möglichkeit, iterativ eine Relevanzbewertung der präsentierten ähnlichsten Bildobjekte durchführen zu können. Nicht allein die Option, anders geartete Clients zu entwerfen, die sich

über das Internet an den Bilddatenbank-Server anmelden können, führten zu der Entscheidung ein Client-Server-System zu entwerfen. Implizit ist durch diese Maßnahme eine klare Trennung von Suchsystem und natürlicher Bedienung durch multimodale Interaktionskanäle gegeben. Da der Bilddatenbank-Server zweigeteilt ist, entsteht die abgebildete dreischichtige Gesamtsystemstruktur.

Multimodaler Bilddatenbank-Client

Der Benutzer wird lediglich mit dem Datenbank-Client, der auf der linken Seite der Abbildung plaziert ist, konfrontiert. Die Ablaufsteuerung bildet hier das zentrale Element. Die Steuerung der Bedienoberfläche mit der Präsentation der Suchergebnisse gehört ebenso zur Aufgabe dieses Systemteils wie der Verbindungsauf- und -abbau zum Datenbank-Server. Die Ablaufsteuerung koordiniert die asynchron eintreffenden Ereignisse der angeschlossenen Modalitätenerkenner. Dabei werden zwei Arten der Synchronisation unterschieden. Zum einen müssen Ereignisse unterschiedlicher Erkenner, die in einem Kontext stehen, als solche detektiert und als gemeinsames Ereignis weiterverarbeitet werden. Zum anderen müssen aktionsauslösende Ereignisse, die im aktuellen Systemzustand aber unsinnig oder verboten sind, verworfen werden. Die akzeptierten Ereignisse werden auf Aktionen abgebildet und in Aufrufe von Datenbankdiensten und Oberflächenaktionen umgesetzt. Diese Dienste sind oft datenabhängig, so dass hier eine entsprechende Koordination vorzunehmen ist. Das vom Server gelieferte Suchergebnis erfordert beispielsweise das anschließende Laden der im Suchergebnis angegebenen Bildobjekte für die Anzeige in der Bedienoberfläche.

An dieser Stelle soll nicht näher spezifiziert werden, ob die eingesetzten Erkenner als eigenständige Applikation mit den dann notwendigen Schnittstellen fungieren oder ob die Erkenner als Ganzes in die Client-Applikation eingebunden werden. Als Interaktionsmodalitäten für den natürlichen Zugang sollen Gestik und Sprache eingesetzt werden. Da die Benutzung einer grafischen Bedienoberfläche die Modalität Maus als standardmäßig anbietet und dieser als bereits bestehender Interaktionskanal für die Entwicklung des Systems enorm wertvoll ist, soll die Maus ebenso zur Interaktion einsetzbar sein.

CRPU

Die ausgewählte Inter-Prozess-Kommunikation, basierend auf TCP/IP, prägt die Struktur des Konfigurations- und Suchmoduls (engl.: *Configuration and Retrieval Processing Unit, CRPU*). Der administrative Teil erledigt den Verbindungsauf- und -abbau sowie die Entgegennahme von Anfragen, die Ausführung der entsprechenden Dienste und das Generieren einer Antwort an den Client. Neben den Diensten, wie beispielsweise dem Entgegennehmen von Bildobjektbewertungen oder dem Anfordern eines Bildes zur Anzeige, bildet die intelligente Suche den Hauptteil dieser Applikation, der aber aus Architektursicht lediglich einen von vielen Diensten darstellt. Die Referenzierung von Regionen ist ebenfalls ein besonderer Dienst, der das wahrscheinlichste Bildobjekt eines Bildes entsprechend der angegebenen Attribute liefert.

Die bildverarbeitenden Teile dieser Applikation, wie Segmentierer und Merkmalsberechnung, sind kein fester Bestandteil der Applikation sondern je nach Wunsch ladbar. Diese Module

werden sowohl zur Initialisierung des Datenbestandes als auch zur Laufzeit für das Hinzufügen neuer Bildregionen oder zur Suche mit einem extern eingebrachten Beispielbild benötigt.

Lediglich diese Applikation hält eine Verbindung zu dem angeschlossenen Datenbank-Management-System.

SQL-Datenbank-Server

Die Verwaltung und das Ablegen der zum Betrieb der Bilddatenbank notwendigen Daten wird einem angeschlossenen SQL-Datenbank-Management-System überlassen [Lan95]. Da die Distanzbildung zweier Repräsentantenvektoren dann sehr teuer ist, wenn sie außerhalb des Datenbank-Management-Systems stattfindet, soll diese Funktionalität in diesen Systemteil integriert werden. Hierfür werden nachladbare Module, so genannte Plugins, mit der gewünschten Funktionalität erstellt, die beim Starten des SQL-Datenbank-Servers geladen werden. Die Funktionalität wird dem SQL-Client durch eine Erweiterung des SQL-Funktionsumfangs bereitgestellt.

Die Realisation der in diesem Abschnitt vorgestellten Konzepte wird in den nachfolgenden zwei Kapiteln nach Bilddatenbank-Server und -Client getrennt erläutert.

Kapitel 4

Datenbank-Server

Der in diesem Kapitel näher vorgestellte Bilddatenbank-Server bildet das Rückgrat des Bilddatenbanksystems. Während der Datenbank-Client die Interaktion mit dem Benutzer durchführt, um eine Suchanfrage zu formulieren und Relevanzbewertungen entgegenzunehmen, und Suchergebnisse anzeigt, wird die eigentliche Bildsuche im Server des Datenbanksystems durchgeführt. Die zur Suche gehörenden Daten werden vom Server gehalten und iterativ modifiziert.

4.1 Datenhaltung

Die Datengrundlage eines Bilddatenbanksystems stellen die Bilder dar, die für eine Suche herangezogen werden sollen. Bei dem hier dargestellten inhaltsbasierten Suchsystem stützt sich jede Suchanfrage auf die aus den Bildern extrahierten Vektoren der unterschiedlichen Merkmalsrepräsentanten. Die Dimension der erzeugten Vektoren variiert von Repräsentant zu Repräsentant (siehe Abschnitt 4.2.2). In diesem Abschnitt wird zunächst beschrieben, welche Daten für den Betrieb der Datenbank notwendig sind und wie auf diese zugegriffen werden muss. Daraus ergibt sich die notwendige Datenhaltung bzw. -generierung.

4.1.1 Bildobjekt

Da ein Bild im Allgemeinen nicht als Ganzes wahrgenommen wird, sondern sofort Objekte erkannt und in einen semantischen Zusammenhang gebracht werden, ist es sinnvoll, diese Unterteilung im System ebenfalls vorzunehmen. Das Bilddatenbanksystem muss mit Bildregionen arbeiten können, die im Idealfall einzelne oder gruppierte Objekte beinhalten.

Für die Generierung solcher Bildregionen, die von Segmentierern durchgeführt wird, gibt es unterschiedliche Ansätze, die je nach Anwendung unterschiedlich gut geeignet sind. Die automatisch bestimmten, aber auch die manuell markierten Regionen eines Bildes werden im Folgenden als Bildobjekte bezeichnet.

4.1.2 Bilddatenhierarchie

Abbildung 4.1 veranschaulicht den Verarbeitungsprozess, der für den Betrieb der Bilddatenbank für jedes Bild durchgeführt werden muss. Das Bild wird im ersten Schritt durch den Einsatz mindestens eines Segmentierers in Bildobjekte zerlegt. Der sich anschließende Verarbeitungs-

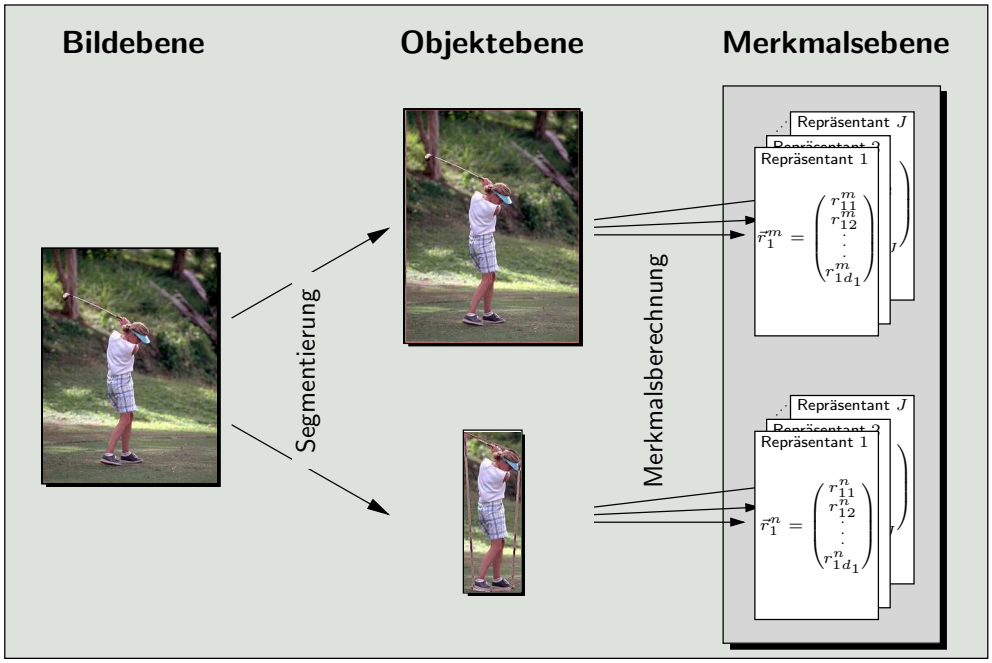


Abb. 4.1: **Abhängigkeiten der Bilddaten:** Jedes Bild der Datenbank wird durch Segmentierer in unterschiedlich viele Bildobjekte zerlegt. Das ganze Bild stellt ebenfalls ein Bildobjekt dar, wobei für jedes Bildobjekt die Vektoren aller Merkmalsrepräsentanten berechnet werden.

schritt besteht in der Berechnung der Merkmalsvektoren der einzelnen Repräsentanten. Auch hier ist für den korrekten Betrieb der Datenbank gefordert, dass mindestens ein Repräsentant verwendet wird, da dies die Grundlage für die Ähnlichkeitsbestimmung von Bildobjekten ist.

Die berechneten Merkmalsvektoren sind für den Suchbetrieb zwingend erforderlich und müssen daher in geeigneter Art und Weise gespeichert werden. Gleiches gilt für die Bilddaten und die Ergebnisse der Segmentierer, die Bildregionen, die zwar nicht für die inhaltsbasierte Suche, jedoch für die Interaktion mit dem Benutzer auf der Ebene des Bilddatenbank-Clients erforderlich sind.

4.1.3 Speichern der Daten

Die im vorherigen Abschnitt aufgezeigten, automatisch generierten Daten bilden den Hauptteil der zu speichernden Daten. Zusätzlich müssen Konfigurationsdaten über die zu benutzenden Segmentier- und Merkmalsmodule gespeichert werden, was für den Erhalt der Flexibilität des Systems erfordert ist.

Generell stellt sich die Frage, wie der Zugriff auf diese Daten erfolgen soll. Zum einen ist zu bedenken, dass es sich um eine Datenbank mit nicht herkömmlichen Daten wie zum Beispiel Bildern handelt, die nicht in traditionellen Datentypen gehalten werden können. Zum anderen ist die hier bestehende Datenabhängigkeit sehr unkompliziert und flach.

Auf eine nähere Untersuchung, welche Vorteile durch den Einsatz eigener Datenhaltungsmechanismen zu erzielen sind, wurde im Rahmen dieser Arbeit verzichtet. Der Grund liegt in dem sehr hoch anzusetzenden Aufwand, eine robuste Datenhaltung mit effizientem Datenzugriff sowie Sortierungsmöglichkeiten zu entwickeln.

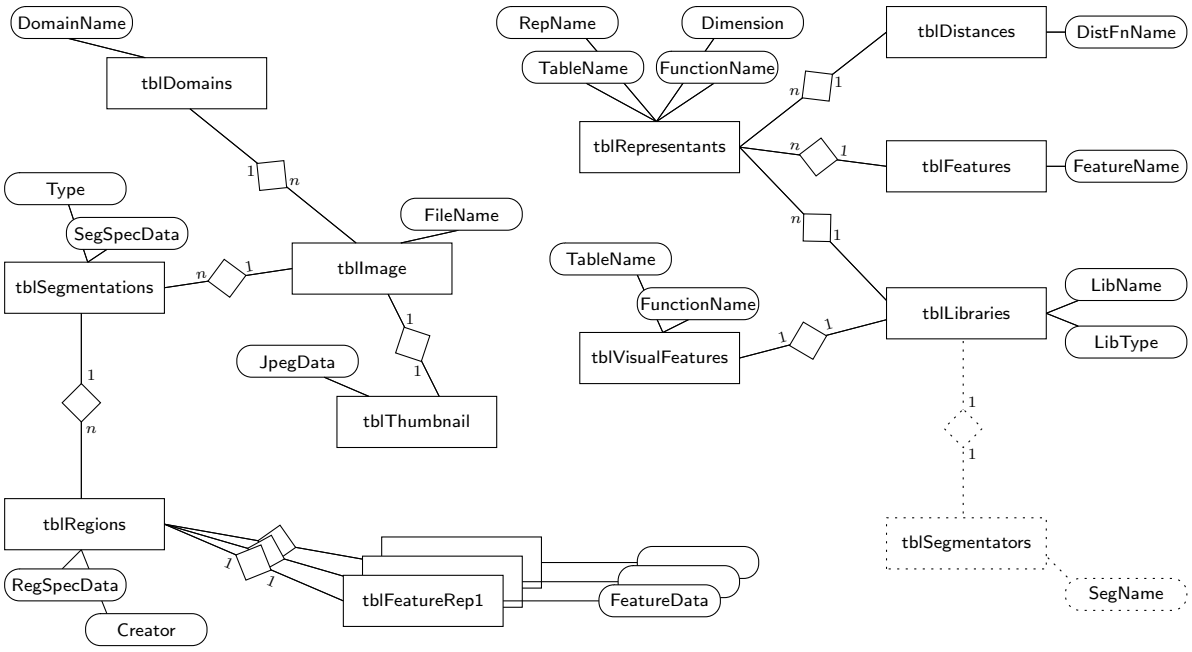


Abb. 4.2: **Entity Relationship Diagram:** Datenorganisation in der verwendeten Datenbank. Zu beobachten ist die Zweiteilung der Daten entsprechend der Betriebsart in Online (links) und Offline (rechts). Der Übersicht halber hier sind nur die wichtigsten Attribute aufgezeigt

In Vorarbeiten wurden verschiedene Alternativen von verwendbaren Datenbankprodukten diskutiert [Käs01]. Im Gegensatz zu der dort vorgestellten Lösung beschränkt sich der Einsatz der selbst implementierten Funktionen zur Erweiterung des SQL-Sprachumfangs bei dem hier vorgestellten System auf die Berechnung des Abstands von Merkmalsvektoren (siehe 4.2.3, Distanzfunktionen). Die Implementierung der Datenbankanbindung ist so ausgelegt, dass sie weitestgehend unabhängig von dem benutzten Datenbankprodukt ist. Um das zu erreichen, ist das Abspeichern von binären Daten zwingend notwendig. Da binäre Daten von der verwendeten Rechnerarchitektur abhängig sind, muss hier eine architekturunabhängige Datenrepräsentation eingesetzt werden (siehe Abschnitt 6.1.1). Auch hier wird die NDR-Datenrepräsentation, die bereits bei der Kommunikation zwischen Client und Server verwendet wird, eingesetzt. Es fiel wie in [Käs01] die Wahl auf das Datenbankprodukt MySQL, das sich als frei verfügbare Applikation durch eine hervorragende Performanz auszeichnet.

Alle Daten werden in der Datenbank entsprechend der Abbildung 4.2 gespeichert. Allein die zugrunde liegenden Bilder werden wie in [Käs01] im Dateisystem gehalten. Der Grund hierfür besteht darin, dass eine SQL-Datenbank primär nicht auf das Speichern von binären Daten ausgelegt ist und mit Einbußen der Performanz beim Zugriff zu rechnen ist, wenn eine Tabelle in großem Maße Binärdaten enthält. Aus diesem Grund wurde auf referentielle Integrität in diesem Punkt verzichtet.

4.2 Modularität

Ein besonderer Aspekt bei der Konzeption des Bilddatenbanksystems, das als Forschungssystem verwendet werden sollte, war die Garantie einer extremen Flexibilität bei der Auswahl der Segmentierungs-, Merkmals- und Distanzberechnung. Dies unterscheidet die Entwicklung von der eines kommerziellen Systems, bei der unter Umständen aus Kostengründen auf diese Flexibilität verzichtet werden muss. Gerade die hier genannten Komponenten des Systems sind bezüglich ihrer Güte extrem von dem verwendeten Bildmaterial abhängig. Die Entwicklung optimaler Komponenten war jedoch nicht der Schwerpunkt des Projektes INDI.

Der Schwerpunkt der Entwicklung eines natürlich zu bedienenden intelligenten Bildsuchsystems liegt diesbezüglich in der Adaption an die Suchintention des Benutzers. Ein solches System lernt, welche Bildmerkmale besonders gut für eine Suche geeignet sind und welche nicht. Es bietet sich daher an, die Menge der benutzten Bildmerkmale sowohl in der Anzahl als auch von der Zusammenstellung nicht a priori festzulegen, sondern im Gegenteil die Flexibilität zu wahren.

Im Gegenzug bietet die nun fertige Implementierung des Bilddatenbanksystems dank der enthaltenen Flexibilität eine ideale Plattform für die Entwicklung und den Test von Bildmerkmalen aller Art.

Das Konzept der angeführten Flexibilität wird in dem modularen Aufbau aller angeführten Berechnungen umgesetzt. Da die enthaltenen Algorithmen eher als Konfiguration denn als dedizierter Bestandteil des Bildsuchsystems zu verstehen sind, werden alle diese Berechnungen in dynamische Programmbibliotheken ausgelagert. Damit kann das Suchsystem bezüglich der bildverarbeitenden Algorithmen erweitert bzw. verändert werden, ohne neu übersetzt werden zu müssen. Bezüglich der Entwicklung der Programmbibliotheken gilt neben der vereinheitlichten Schnittstelle die Anforderung, dass alle beinhalteten Funktionen thread-sicher, das heißt reentrant fähig, sein müssen.

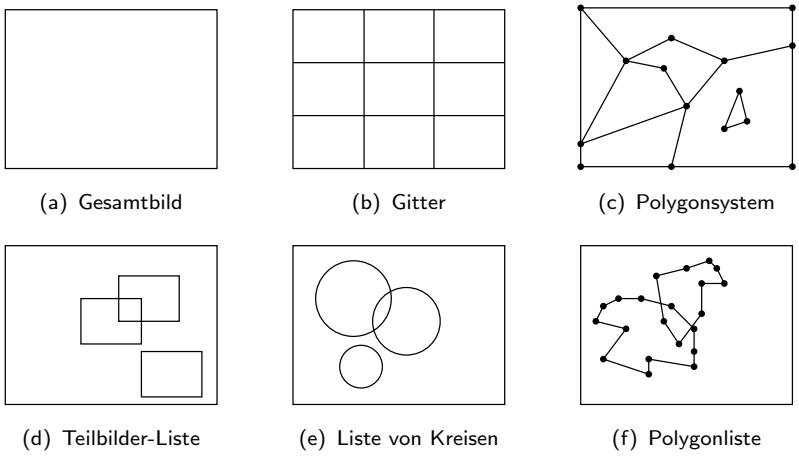
Die folgenden Abschnitte, Segmentierung und Merkmalsberechnung, beschreiben die Schnittstellen, der entsprechenden Bibliotheken. Ein Modul, das diesen Schnittstellen entspricht, kann vom INDI-System ohne weiteres Hinzutun erkannt und verwendet werden.

4.2.1 Segmentierung

Als Ergebnis bildet eine Segmentierung eine beliebige Menge von Segmenten. Bezüglich der Form und Darstellung des Segmentierergebisses soll an dieser Stelle keine Einschränkung gelten, so dass die Datenstruktur, die das Ergebnis beinhaltet, von dem verwendeten Segmentierer

abhängt. Hier lässt sich also keine einheitliche Struktur für die Verwaltung aller Segmentierergebnisse definieren. Für die einheitliche Weiterverarbeitung jedoch muss das Segmentierergebnis in eine Regionendarstellung überführt werden, die im weiteren Verlauf dieses Abschnitts vorgestellt wird.

Abbildung 4.3 zeigt exemplarisch, wie Ergebnisse unterschiedlicher Segmentierer aussehen können. Es ergeben sich verschiedene Klassen von Segmentierergebnissen. Zum einen sind das Ergebnisse, bei denen das gesamte Bild in Regionen aufgeteilt wird (dargestellt in 4.3(a)–(c)), zum anderen können in den Darstellungen 4.3(d)–(f) Überlappungen der Regionen auftreten.



(a) Gesamtbild (b) Gitter (c) Polygonsystem

(d) Teilbilder-Liste (e) Liste von Kreisen (f) Polygonliste

Abb. 4.3: **Repräsentationen von Segmentierergebnissen:** (a)–(c) sind Ergebnisse, bei denen die Menge aller Regionen das Gesamtbild ergibt. In (d) und (e) wird die Region durch geometrische Formen repräsentiert, diese lassen sich durch wenige Größen beschreiben. (f) zeigt ein Segmentierergebnis, in dem jede Region durch einen geschlossenen Polygonzug repräsentiert wird.

Folgende Aufstellung zeigt, durch welche Datenstrukturen solche Segmentierergebnisse dargestellt werden können:

(a), (b) **Gitter/Gesamtbild**: In diesem Fall sind die entstehenden Regionen nur von der Bildgeometrie, nicht aber vom Bildinhalt abhängig. Damit kann das Segmentierergebnis auf die Gittergeometrie beschränkt werden, die vollkommen unabhängig vom betrachteten Bild ist. Repräsentation (a) bildet lediglich einen Sonderfall von (b).

(c) **Polygonsystem**: Bei diesem Segmentierergebnis bieten sich drei Arten der Repräsentation an:

1) Label-Map: Je eine Speichereinheit pro Pixel hält die Regionenzugehörigkeit. Diese Art ist sehr speicherineffizient, jedoch sehr laufzeiteffizient bei der Klassifikation von Pixeln einzusetzen

2), 3) Je eine Liste von Punkten und Kanten bzw. eine Liste von geschlossenen Polygonzügen (siehe auch (f)): Diese Art der Repräsentation ist speichereffizient. Der Aufwand für die Klassifikation eines Pixels bezüglich der Regionzugehörigkeit ist jedoch erheblich größer als in 1).

(d) **Liste von Teilbildern**: Hier werden die Translationsinformationen und die Box-Geometrie in einer Liste gespeichert.

(e) **Liste von Kreisen**: Entsprechend der Teilbildliste werden hier Kreismittelpunkte und -radien in einer Liste für die Darstellung benutzt.

(f) **Liste von geschlossenen Polygonzügen**: Jede Region wird als Liste von Punkten repräsentiert, die einen geschlossenen Polygonzug darstellen.

Die Bildsegmentierung wurde einleitend durch das Vorhandensein verschiedener Objekte in einem Bild motiviert (siehe 4.1.1, Bildobjekt). Da es im Allgemeinen nicht sinnvoll ist, ein Bild vollständig in Objekte zu zerlegen, bietet es sich nicht an, solche Segmentierer einzusetzen, die Ergebnisse vom Typ 4.3(b) und (c) liefern. Objekte innerhalb von Bildern lassen sich ganz allgemein gültig im diskreten Fall eines digitalen Bildes durch Segmentierergebnisse der Form 4.3(f) darstellen.

Interne Repräsentationen der Bildregionen

Unabhängig von den durch Segmentierer erzeugten Datenstrukturen muss eine systemweit eindeutige Repräsentation von Bildobjekten definiert sein. Auch hier gilt es einen Mittelweg, zwischen Systemperformanz und Benutzung von Systemressourcen zu finden, wie folgend erläutert wird.

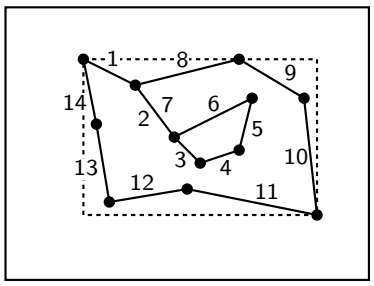


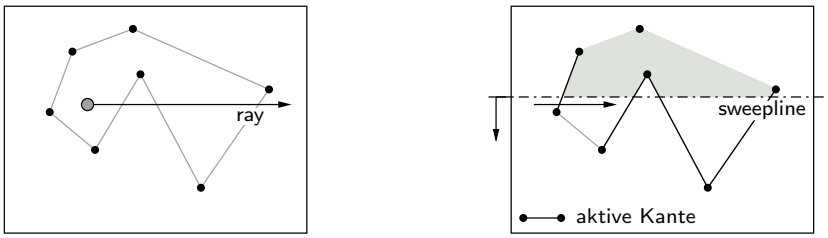
Abb. 4.4: **Interne Regionenrepräsentation**

Die Überlegungen des vorherigen Abschnitts führten zu der Festlegung der im Folgenden beschriebenen interne Repräsentation eines Bildobjekts. Zunächst wird eine das Bildobjekt umgebende Box (engl.: *bounding box*), die der Darstellung eines Teilbildes entspricht, berechnet. Diese Information ist die Grundlage eines Bildobjekts und ausnahmslos definiert. Sofern das Objekt nur einen Teil dieser Box umfasst, wird der Repräsentation ein geschlossener Kantenzug hinzugefügt, der in Koordinaten relativ zum Ursprung der umgebenden Box definiert ist.

Abbildung 4.4 zeigt eine solche Bildregion innerhalb eines Bildes. Die umgebende Box, die durch die Maxima und Minima der Kantenzug-Punktkoordinaten festgelegt ist, ist gepunktet dargestellt. Das Besondere an gerade diesem Objekt ist, dass der innere Teil, der von den Kanten 3, 4, 5 und 6 umgeben ist, nicht zum Objekt gehört. Kanten 2 und 7 sind bei dieser Konstruktion deckungsgleich.

Die Berechnung der Zugehörigkeit eines einzelnen Pixels zu einem Bildobjekt ist in der Applikation des Datenbank-Clients bei der Selektion von Regionen durch Klicken einer bestimmten

Stelle im Bild erforderlich. Die Berechnung ist bei der Verwendung eines Kantenzuges als Berechnungsgrundlage abhängig von der Anzahl der enthaltenen Kanten entsprechend hoch. Denn dabei muss nach dem Strahlenschnittverfahren [Hai94] (engl.: *ray crossing*) festgestellt werden, wie viele der Kanten des Zuges einen Strahl ausgehend vom Testpunkt schneiden. Basierend auf diesem Ergebnis kann eine Aussage darüber getroffen werden, ob ein Punkt zu der Region des Kantenzuges, also zum Bildobjekt, gehört oder nicht.



(a) Klassifikation einzelner Pixel (b) Klassifikation aller Pixel

Abb. 4.5: **Klassifikation der Pixel bezüglich der Bildobjektzugehörigkeit auf Basis eines Kantenzuges:** Klassifikation eines einzelnen Pixels durch Betrachtung aller Kanten, die die Zeile des betrachteten Pixels schneiden (a). Klassifikation aller Pixel durch das Führen einer Liste von aktiven Kanten, deren Schnittpunkte mit der betrachteten Zeile (sweepline) berechnet werden (b).

Muss eine solche Klassifikation für alle Pixel der umgebenden Box durchgeführt werden, beispielsweise bei der Berechnung der Objektmerkmale, so ist eine andere Repräsentation des Objekts zu bevorzugen. In diesem Fall wird die Klassifikation bezüglich der Objektzugehörigkeit für jedes Pixel der umgebenden Box mit einem Sweepline-Algorithmus [Ben79] durchgeführt. Bei diesem Verfahren gibt es eine aktive Zeile, die das Bild von oben nach unten durchläuft. Die Punkte des Kantenzuges werden bezüglich ihrer Zeilenkoordinate sortiert. Damit lässt sich effizient eine Liste aktiver Kanten, die die aktive Zeile schneiden, bilden und pflegen. Die Pixel werden so zeilenweise klassifiziert. Zählt man die Kanten, die linksseitig des betrachteten Pixels liegen, dann ist die Zugehörigkeit gegeben, wenn es sich um eine ungerade Anzahl von Kanten handelt und umgekehrt.

Es bietet sich an, das Ergebnis einer solchen Berechnung in einer Bitmap zu speichern. Eine solche Datenstruktur, die den indizierten Zugriff auf das Klassifikationsergebnis erlaubt, ist ebenfalls in der Repräsentation des Bildobjekts vorhanden. Da der Speicheraufwand dafür jedoch entsprechend groß ist, wird die Berechnung nur auf Anfrage durchgeführt und diese Bitmap gefüllt (bei dem von INDI verwendeten Bilddatentyp JPEG, bei dem 24 Bit für jedes Pixel benutzt werden, ist der Speicheraufwand für die benötigten Bitmaps für n Regionen gleich $n/24$ der Bildgröße).

Aufbau einer Segmentiererprogrammbibliothek

Die hier aufgezeigten Programmbibliotheken sind wie erwähnt kein fester Bestandteil der Applikation. Sie werden vielmehr zur Laufzeit nachgeladen. Der Hauptbestandteil einer solchen

Programmbibliothek ist neben der eigentlichen Berechnung der Segmentierung eine Umwandlung der Segmentierergebnisse in die interne Repräsentation von Bildobjekten.

```
typedef struct t_SegInfo
{
   char *                szName;              /* Name der Segmentierung */
   int                   iSegmentID;          /* Identifikationsnummer der Segmentierung */
   fnSegment *           ptSegment;           /* Segmentierungsfunktionszeiger */
   fnSegToDB *           sConvertSetToDB;     /* Funktionen zur Konvertierung in und aus einer ... */
   fnDBToSeg *           ptConvertDBToSeg;    /* ... plattformunabhängigen Repräsentation */
   fnGetRegions *        tGetRegFromSeg;      /* Extrahieren von Regionen aus der Segmentierung */
   fnFreeSegmentation *  vFreeSegmentation;   /* Freigabe der Segmentierung */
   fnGetSegLibVersion *  iGetVersion;         /* Version der Programmbibliothek */
   fnSegDetatch *        vDetatch;            /* Freigabe der Bibliothek */
} T_SegInfo;
```

Abb. 4.6: **Administrative Struktur eines Segmentierers:** Jede Segmentiererprogrammbibliothek stellt eine Struktur des hier gezeigten Aufbaus zur Verfügung. Neben dem Namen enthält die Struktur die Identifikationsnummer des Segmentierers, die im System eindeutig ist, sowie eine Reihe von Zeigern auf Funktionen, über die die eigentliche Funktionalität der Bibliothek angesprochen werden kann.

Aufgrund der Tatsache, dass eine solche Bibliothek erst zur Laufzeit geladen wird, ist der Datenbank-Server-Applikation zunächst nicht bekannt, wo sich die gewünschten Informationen in der Programmbibliothek befinden. Deshalb muss eine Gemeinsamkeit definiert werden, der alle Segmentierungsbibliotheken entsprechen. Bei den Segmentiererprogrammbibliotheken existiert eine Funktion unter einem festen Namen, die eine Informationsstruktur zurückliefert, wie sie in Abbildung 4.6 dargestellt ist.

Ein Segmentierergebnis wird immer mit der systemweit eindeutigen `iSegmentID` abgespeichert. Dadurch wird festgelegt, welche Programmbibliothek für die Verarbeitung des Ergebnisses herangezogen werden muss. Folgende vier Funktionen werden für die Erzeugung bzw. die Verarbeitung der Segmentierungsergebnisse benötigt:

`ptSegment`: Diese Funktion führt die Segmentierung auf einem übergebenen Bild durch. Das Ergebnis ist eine spezielle C-Struktur, die das Segmentierungsergebnis beinhaltet. Alle diese speziellen Strukturen beinhalten am Anfang einen allgemeinen Strukturkopf, so dass eine gemeinsame Verwaltung dieser Strukturen möglich ist.

`sConvertSetToDB`: Die binären Daten eines Segmentierungsergebnisses sollen in der Datenbank gespeichert werden können. Die Konvertierung in eine von der Rechenplattform unabhängigen Repräsentation dieser Daten ist notwendig und wird von dieser Funktion übernommen. Auch hier bietet es sich an, die NDR-Repräsentation zu wählen.

`ptConvertDBToSeg`: Eine Rücktransformation aus der plattformunabhängigen Repräsentation übernimmt diese Funktion.

`tGetRegFromSeg`: Das Extrahieren aller Regionen aus dem Segmentierergebnis führt diese Funktion aus. Das Ergebnis besteht aus einer Liste von Regionen in der internen Regionenrepräsentation.

4.2.2 Merkmalsberechnung

Die Merkmalsberechnung wird hauptsächlich bei der Initialisierung der Datenbank durchgeführt. Dies ist ein langwieriger Prozess, und deshalb ist bei der Entwicklung dieser Module auf Laufzeiteffizienz Wert zu legen. Das ist ein Grund, warum die Struktur einer Merkmalsprogrammbibliothek anders gehalten ist als die eines Segmentierers. In einigen Fällen basieren die unterschiedlichen Merkmalsräume auf denselben grundsätzlichen Daten, die aus dem Bild berechnet werden müssen, wie zum Beispiel die Konvertierung in einen bestimmten Farbraum. Es bietet sich in diesen Fällen an, solche grundsätzlichen Berechnungen für alle darauf basierenden Merkmale zu benutzten, also nur einmalig durchzuführen. Um das zu erreichen, muss eine Merkmalsprogrammbibliothek die Funktionen für die Berechnung mehrerer Repräsentanten von Merkmalen verwalten können.

Aufbau einer Merkmalsprogrammbibliothek

Die Programmbibliothek eines Merkmals muss wie die eines Segmentierers eine Funktion zum Auslesen des funktionalen Inhalts zur Verfügung stellen. Des Weiteren werden administrative Funktionen, zum Beispiel zum An- und Abmelden der Bibliothek, verlangt. `T_RepInfo` ist die Struktur, die für die Beschreibung eines Repräsentanten benutzt wird, sie ist in Abbildung 4.7 dargestellt.

```
typedef struct t_RepInfo
{
   int      iDimension;       /* Dimension des Merkmalsvektors*/
   char *   szFunctionName;   /* Funktionsname der Berechnungsfunktion*/
   char *   szName;           /* Name des Merkmals */
   int      iFeatureID;       /* Klasse des Merkmals */
   int      iRepID;           /* Systemweite eindeutige Identifikationsnummer */
   int      iDistanceID;      /* Identifikationsnummer der zu verwendenden Distanzfunktion */
} T_RepInfo;
```

Abb. 4.7: **Administrative Struktur eines Repräsentanten:** Jede Merkmalsprogrammbibliothek stellt ein Array der hier dargestellten Struktur zur Verfügung

Der logische Aufbau von Merkmalen und Merkmalsrepräsentanten wurde in Abbildung 3.3 vorgestellt. Diese Hierarchie ist in dem Eintrag `iFeatureID` fixiert. Alle dem System bekannten Merkmalsklassen wie Farbe, Textur, Struktur und Form haben eine systemweit eindeutige Identifikationsnummer, die in diesem Eintrag referenziert wird. Eine Zuordnung der zu benutzenden Distanzfunktion, die für den Vergleich zweier Vektoren bzw. Signaturen für den Repräsentanten der Dimension `iDimension` benutzt werden soll, wird durch den Eintrag `iDistanceID` hergestellt.

Der Name des Repräsentanten im Eintrag `szName` wird maßgeblich in der Datenbankorganisation verwendet (siehe dazu Abschnitt 4.3, Initialisierung der Datenbank). Der Funktionsname, der die Berechnung des Merkmalsvektors, bzw. der Signatur durchführt, wird in `szFunctionName` übergeben.

Berechnungsfunktion

Die Berechnung eines Merkmalsvektors findet auf der Basis einer übergebenen Region statt. Wie einleitend erwähnt, gilt es, bei der Entwicklung dieser Berechnungsfunktionen kurze Laufzeiten zu erzielen. Die Repräsentation von Regionen eines Bildes, wie sie in Abschnitt 4.2.1 vorgestellt wurde, verfolgt jedoch die genau entgegengesetzte Anforderung, den Speicherbedarf so gering wie möglich zu halten. Aus diesem Grund wurde die Arbeitsstruktur einer Region um eine Bitmap der Größe der umgebenden Box erweitert. Die Bitmap enthält für jedes Pixel der Region die Information der Zugehörigkeit des Pixels zu der Region. Die Erstellung der Bitmap ist vor der Berechnung der Merkmale einer Region, die durch einen Kantenzug und nicht allein durch die umgebende Box definiert ist, durchzuführen. Fehlt die Bitmap gänzlich, dann bezieht sich die Merkmalsberechnung auf den vollständigen Inhalt der umgebenden Box.

Bereitstellen mehrerer Repräsentanten auf der Basis einer Berechnungsgrundlage

Wie bereits erwähnt, kann eine Merkmalsbibliothek eine Reihe von Repräsentanten beinhalten, die alle auf derselben Datenbasis arbeiten. Eine solche Datenbasis soll für jede Region nur einmalig berechnet werden müssen.

Das stellt an die Bibliothek die Anforderung, die Informationen für aufeinander folgende Aufrufe der Berechnungsfunktionen zu erhalten. Da nicht jede Bibliothek eine solche Sammlung von Repräsentanten beinhaltet, sollen die Daten nach außen gekapselt werden. Das ist nur unter Verwendung von globalen Variablen möglich. Bei der Entwicklung einer solchen Bibliothek sind damit folgende Punkte zu beachten:

1. Die Bibliothek muss selbständig erkennen, wann die einmalige Berechnung auf der Basis der Region durchgeführt werden muss bzw. wann diese Daten freigegeben werden dürfen. Für diese Erkennung kann die eindeutige Identifikationsnummer der Region herangezogen werden. Vom System wird garantiert, dass die Merkmalsvektoren einer Region nacheinander berechnet werden und dass erst, nachdem der letzte Vektor bestimmt ist, mit einer anderen Region fortgefahren wird.

2. Da die Berechnungsfunktionen nicht allein für die Initialisierung der Datenbank benutzt werden, sondern auch von unbestimmt vielen Server-Threads, die je einen Datenbank-Client bedienen und parallel existieren, muss jeder Thread seine eigenen globalen Variablen benutzen.

4.2.3 Distanzberechnung

Die Gruppe der Distanzfunktionen stellt eine Besonderheit im System dar, denn diese Funktionen werden nicht vom Bilddatenbanksystem selbst sondern von der verwendeten MySQL-Datenbank ausgeführt. Sie bilden eine Erweiterung des SQL-Funktionsumfangs.

Generell wird eine Distanzfunktion zum Vergleich zweier Vektoren eines Merkmalsraums herangezogen (siehe 4.2.3). Der Name Distanzfunktion stammt aus der Vektorinterpretation der

Merkmale. In diesem Fall wird die Distanz zwischen zwei Punkten, die durch die Vektoren gegeben sind, als Grundlage für die Aussage von Ähnlichkeit der Vektoren bestimmt. Haben die Merkmalsrepräsentanten jedoch eine andere Bedeutung, wie es zum Beispiel bei Histogrammen der Fall ist, dann wäre eine einfache Distanzberechnung unter Umständen nicht aussagekräftig. Solche Merkmale erfordern dann häufig eine speziell angepaßte Distanzfunktion.

Die Repräsentationen von Merkmalsvektoren sind, unabhängig von ihrer Bedeutung im System, identisch und können damit von Distanzfunktionen desselben Prototyps verarbeitet werden. Alle Distanzfunktionen müssen als Resultat einen Wert zurückliefern, den man als „Maß der Unähnlichkeit“ definieren könnte.

Die Distanzberechnung wird, wie in Abschnitt 3.2.3 beschrieben, parametrisiert durchgeführt. Somit besteht die Parameterliste einer solchen Funktion aus den beiden zu vergleichenden Vektoren mit der dazugehörigen Dimension und einem Gewichtsparameter, der als Vektor oder als quadratische Matrix ausgelegt sein kann. Damit ist die Berechnung eines erweiterten bzw. generalisierten euklidischen Abstands möglich.

```
SELECT DistanceFunction('<VectorX>', '<VectorY>', <Dim>,
                        '<Weights>', <WeightDim>)
```

Abb. 4.8: **SQL-Aufruf einer Distanzfunktion:** Der Aufruf berechnet den Distanzwert der Vektoren `VectorX` und `VectorY` unter Verwendung des Gewichtsvektors bzw. der Gewichtsmatrix `Weights`

Ein prinzipieller Aufruf in SQL ist in der Abbildung 4.8 aufgezeigt. Welche der beiden Varianten bei einem Aufruf gemeint ist, entscheidet sich an dem Parameter `WeightDim` bezüglich der Vektordimension `Dim`. Entspricht dieser dem Quadrat des Eintrags `Dim`, dann wird der generalisierte euklidische Abstand berechnet. In dem Sonderfall des erweiterten euklidischen Abstands wird ein Vektor $\vec{w}$, der die Spur der Gewichtsmatrix $\underline{W}$ enthält, übergeben. In diesem Fall sind `Dim` und `WeightDim` identisch. Die für die Berechnung der Abstände benutzte Matrix ergibt sich in diesem Fall zu:

$$\underline{W} = \begin{pmatrix} w_1 & 0 & \cdots & 0 \\ 0 & w_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & w_{Dim} \end{pmatrix}$$

Die hier vorgenommene Unterscheidung liegt in der Performanz der Berechnung begründet. Benutzte man in jedem der beiden Fälle eine Matrix zur Berechnung des Abstands, dann würden im dem Fall, dass lediglich die Diagonalmatrix besetzt ist, $Dim(Dim-1)$ Multiplikationen mit NULL und entsprechend viele Summationen von NULL unnötig berechnet werden.

4.3 Initialisierung und Inbetriebnahme einer Datenbank

Die Initialisierung der Datenbank wird vor der eigentlichen Inbetriebnahme der Bilddatenbank durchgeführt. Das Ziel dieses Vorgangs ist, eine MySQL-Datenbank anzulegen und diese mit allen notwendigen Daten, die für den Betrieb der Bilddatenbank notwendig sind, zu füllen.

Konfiguration und Initialisierung

Die initiale Konfiguration besteht aus folgenden Teilen, die in Textdateien einer Initialisierungsapplikation übergeben werden.

Eine Menge von Segmentiererprogrammbibliotheken: Jede verwendete Segmentiererbibliothek stellt einen Segmentierer zur Verfügung, der bei dem Vorgang der Initialisierung benutzt werden soll.

Eine Menge von Merkmalsprogrammbibliotheken: Alle in den angegebenen Merkmalsprogrammbibliotheken enthaltenen Repräsentanten werden bei der Initialisierung der Datenbank verwendet.

Eine Menge von Bildern: Die zu verwendenden Bilder müssen in Verzeichnisse gelegt werden. Der Initialisierung wird eine Liste von Verzeichnissen übergeben. Alle in den Verzeichnissen enthaltenen Bilder werden in die Datenbank eingefügt.

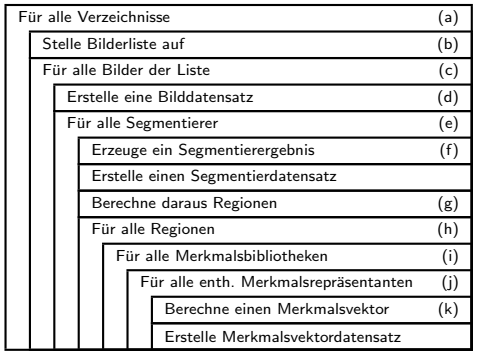


Abb. 4.9: **Berechnung aller Datenbankeinträge**

Abbildung 4.9 stellt den Algorithmus zum Füllen der Datenbank dar. Ausgehend von der Liste der zu verwendenden Verzeichnisse (a) wird für jedes Verzeichnis die Menge der einzufügenden Bilder ermittelt (b). Für jedes gefundene Bild (c) wird folgende Prozedur durchgeführt: Zunächst wird ein Datenbankeintrag für das Bild vorgenommen (d). Jetzt wird die projektierte Liste von Segmentierern abgearbeitet (e). Das Ergebnis jedes Segmentierers wird in der Datenbank gespeichert (f) und daraus je eine Liste von Regionen erstellt (g). Mit jeder Region wird die folgende Verarbeitung durchgeführt (h): Aus der Liste der projektierten Merkmalsbibliotheken (i) werden alle jeweils beinhalteten Merkmalsrepräsentanten (j) angewendet und je ein Merkmalsvektor berechnet, der dann in der Datenbank abgelegt wird (k).

Inbetriebnahme

Nach Ablauf des Initialisierungsvorgangs steht die neu erzeugte Datenbank dem Bilddatenbank-Server zur Verfügung. Da die gesamte Konfiguration der Datenbank in der Datenbank selbst

abgelegt ist, ist es ausreichend, dem Bilddatenbank-Server lediglich Ort und Namen der Datenbank mitzuteilen, was durch einen Eintrag in der Systemkonfigurationsdatei erledigt wird.

4.4 Struktur des Bilddatenbank-Servers

In Abschnitt 3.3.2 wurde die zugrunde liegende Applikation einer TCP-Client- und Server-Applikation vorgestellt. Der hier vorliegende Abschnitt geht auf die technische Umsetzung dieser Struktur ein.

Wird ein Server so konzipiert, dass eine gleichzeitige Anmeldung mehrerer Clients erlaubt ist, und das ist hier der Fall, dann muss die Server-Applikation so ausgelegt werden, dass für jeden Client eine nebenläufige Server-Komponente erzeugt wird. Es gibt zwei gängige Möglichkeiten, dieses Verhalten umzusetzen, wobei der hauptsächliche Unterschied im Zugriff auf gemeinsame Daten besteht. Server, bei denen keinerlei globale Daten existieren oder bei denen sogar eine gemeinsame Datenhaltung unerwünscht ist, wie es beispielsweise bei einem *telnet*-Server der Fall ist, erzeugen für jeden sich anmeldenden Client einen eigenständigen Prozess durch einen `fork`-Aufruf.

Besteht hingegen die Anforderung, gemeinsame Daten zu benutzen, sei es aus Gründen der Performanz, weil zum Beispiel die Daten sehr aufwendig zu berechnen sind oder aufgrund der Datenmenge, die bei Mehrfachinstanziierung eine erhebliche Menge an Systemspeicher verbraucht, dann bietet es sich an, die Nebenläufigkeit durch Threads zu realisieren. Da sich Threads einen gemeinsamen Adressraum teilen, wird kein zusätzlicher Aufwand für die Nutzung gemeinsamer Daten notwendig.

Abbildung 4.10 zeigt den gewählten Aufbau des Bilddatenbank-Servers. Die administrativen Aufgaben, die maßgeblich aus dem Verbindungsaufbau und der weiteren Verwaltung von Such-Sessions bestehen, werden von den zwei Threads in der Abbildung links durchgeführt. Nach dem Verbindungsaufbau wird je ein ausführender Thread kreiert. Diese Threads sind auf der rechten Seite der Abbildung zu sehen.

4.4.1 Single-/ Multi-Client-Session, Datenhaltung

Wie bereits in der Abbildung 4.10 angedeutet, gibt es zwei unterschiedliche Arten von Clients, die vom Server bedient werden können. Abhängig davon, wie die Anmeldung des Clients geartet ist, wird eine der beiden folgenden Betriebsarten angewendet (siehe Abbildung 4.12).

1. **Single-Client:** In dieser Betriebsart werden alle Datenbankanfragen, die von einem Datenbank-Client gestellt werden, über eine einzige Verbindung zum Datenbank-Server sequentiell bearbeitet. Diese Betriebsart bildet den Standardfall.

2. **Multi-Client:** Eine solche Betriebsart wird von einem Client gefordert, der nicht in der Lage ist, die Verbindungen zum Datenbank-Server zu halten. Web-Browser gehören beispielsweise zu dieser Gattung von Clients. Ein Web-Browser baut für jedes zu ladende Objekt einer Seite eine eigene Verbindung zum entsprechenden Server auf, denn dadurch

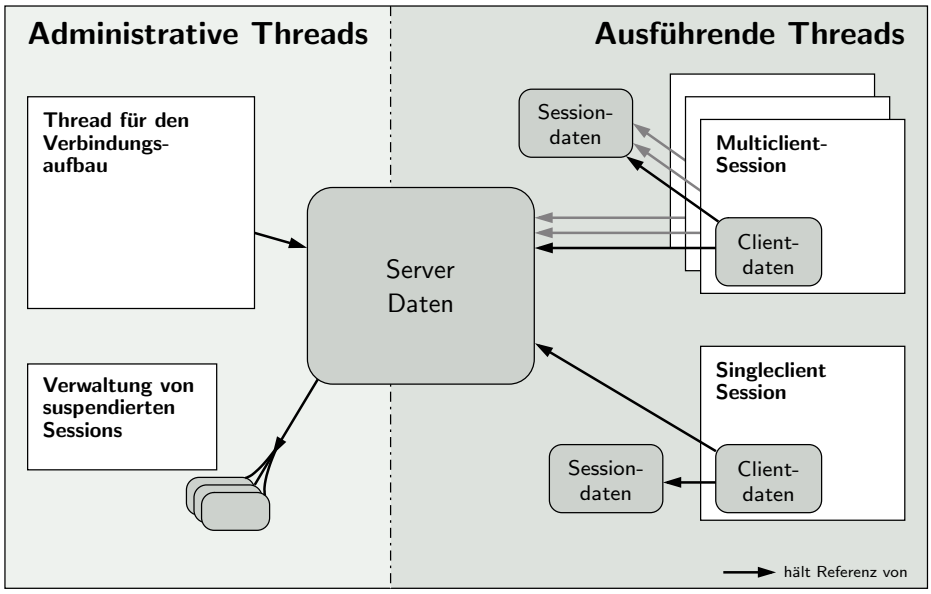


Abb. 4.10: **Server-Thread- und Datenstruktur:** Die Applikation teilt sich auf in administrative Threads und solche, die die eigentlichen Suchanfragen bearbeiten. Die Daten sind hierarchisch geordnet. Unterschieden werden Server-Daten, die applikationsweit verfügbar sind, in Session-Daten, die die Informationen einer aktiven Suche beinhalten, und Client-Daten, die Informationen über die bestehende Verbindung halten.

kann ein einheitlicher Zugriff auf die in der Seite enthaltenen Referenzen auf Objekte von unterschiedlichen Servern erfolgen.

Nicht nur die Anforderungen der Multi-Client-Betriebsart sprechen für den Einsatz von Threads und für den sich daraus ergebenden gemeinsamen Speicher. Auch andere Betriebsmittel können zentral verwaltet von allen beteiligten Threads gemeinsam genutzt werden. Wie in Abbildung 4.10 zu sehen, wird in dem Server hierarchisch zwischen drei Datenstrukturen unterschieden:

- **Server-Daten:** Ein Sammlung globaler Daten. Sie umfasst unter anderem Informationen über geladene Programmbibliotheken, eine Liste von suspendierten Sessions, einen Synchronisationsmechanismus zum Aufwecken schlafender Threads und Synchronisationsstrukturen für den geregelten Zugriff auf die Server-Daten. Diese globale Struktur ist allen Threads zugänglich.
- **Session-Daten:** Alle Daten, die den aktuellen Zustand der Suche beschreiben, sind in dieser Struktur enthalten. Außerdem beinhaltet die Struktur eine Synchronisationsdatenstruktur, mit der der Zugriff von zustandsverändernden Anfragen geregelt wird.
- **Client-Daten:** Die Client-Datenstruktur hält neben den Referenzen auf die beiden oben genannten Strukturen im wesentlichen nur die Daten, die zur Realisierung bzw. der Aufrechterhaltung der TCP/IP-Verbindung notwendig sind.

Besondere Anforderungen der Multi-Client-Betriebsart

Wenn ein Web-Client, der in der Multi-Client Betriebsart arbeitet, beispielsweise eine Seite mit einem Suchergebnis aufbauen will, werden entsprechend der Anzahl der darzustellenden Bilder Anfragen an den Server initiiert (siehe 5.5, Ablauf einer Suchiteration), die je von einem eigenen Thread bearbeitet werden. Für die Erzeugung von Threads ist ein erheblicher Arbeitsaufwand einzuräumen. Da zu erwarten ist, dass entsprechend viele Threads bei einer folgenden Anfrage wieder benötigt werden, bietet es sich an, Threads, die die Suchanfrage beendet haben und deren Verbindung beendet werden soll, für spätere Bearbeitungen zur Verfügung zu halten, sie also lediglich zu suspendieren. Dafür wird vom Server eine spezielle Verwaltungsanfrage „SuspendSession“ zur Verfügung gestellt, die in diesen Fällen anzuwenden ist.

4.4.2 Verbindungsaufbau und Aufbau eines ausführenden Threads

Der Verbindungsaufbau muss bei allen Betriebsarten identisch sein, denn zum Zeitpunkt des Aufbaus ist dem Server noch nichts über die Beschaffenheit des sich anmeldenden Clients bekannt.

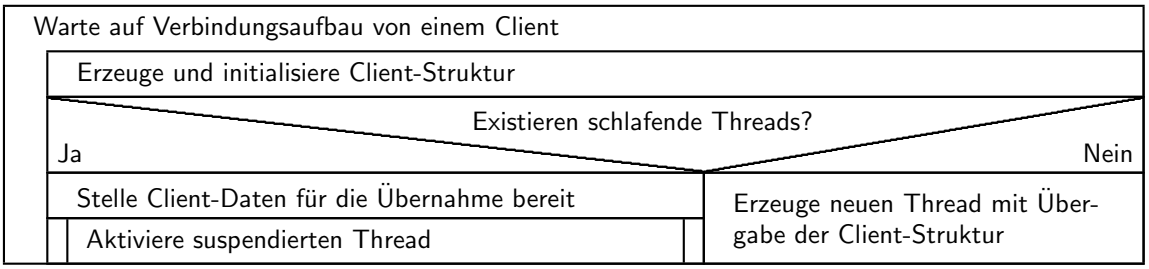


Abb. 4.11: **Verbindungsaufbau Client/Server:** Die hier dargestellte Aufgabe erstreckt sich lediglich auf das Anlegen einer Datenstruktur und das Bereitstellen eines ausführenden Threads

Sofern der Server aktuell eine Multi-Client-Session aufrechterhält, existieren suspendierte Threads im System. Bei dem Verbindungsaufbau werden diese Threads wieder als ausführende Threads eingesetzt, wobei es in diesem Fall unwichtig ist, ob der aktuell anstehende Thread zu einer Single- oder einer Multi-Client-Suche gehört. Diese Strategie begünstigt das Abbauen von suspendierten Threads, wenn keine weiteren Multi-Client-Suchen bestehen.

Der Verbindungsaufbau gestaltet sich wie in Abbildung 4.11 dargestellt. Zunächst wird eine Client-Struktur erzeugt. Falls suspendierte Threads existieren, wird die Client-Struktur zur Übergabe vorbereitet und ein suspendierter Thread aktiviert. Im anderen Fall wird ein neuer Thread erzeugt, dem die Client-Struktur direkt übergeben wird.

Abbildung 4.12 stellt den Ablauf eines ausführenden Threads dar. Nach der Verbindungsaufnahme wird die Arbeit an einer der markierten Stellen aufgenommen. Ein neu erzeugter Thread beginnt die Arbeit oben durch das Eintreten in die globale Schleife (⊛). Ein suspendierter Thread nimmt die Arbeit an der durch ⊤ markierten Stelle wieder auf. Nachdem in diesem

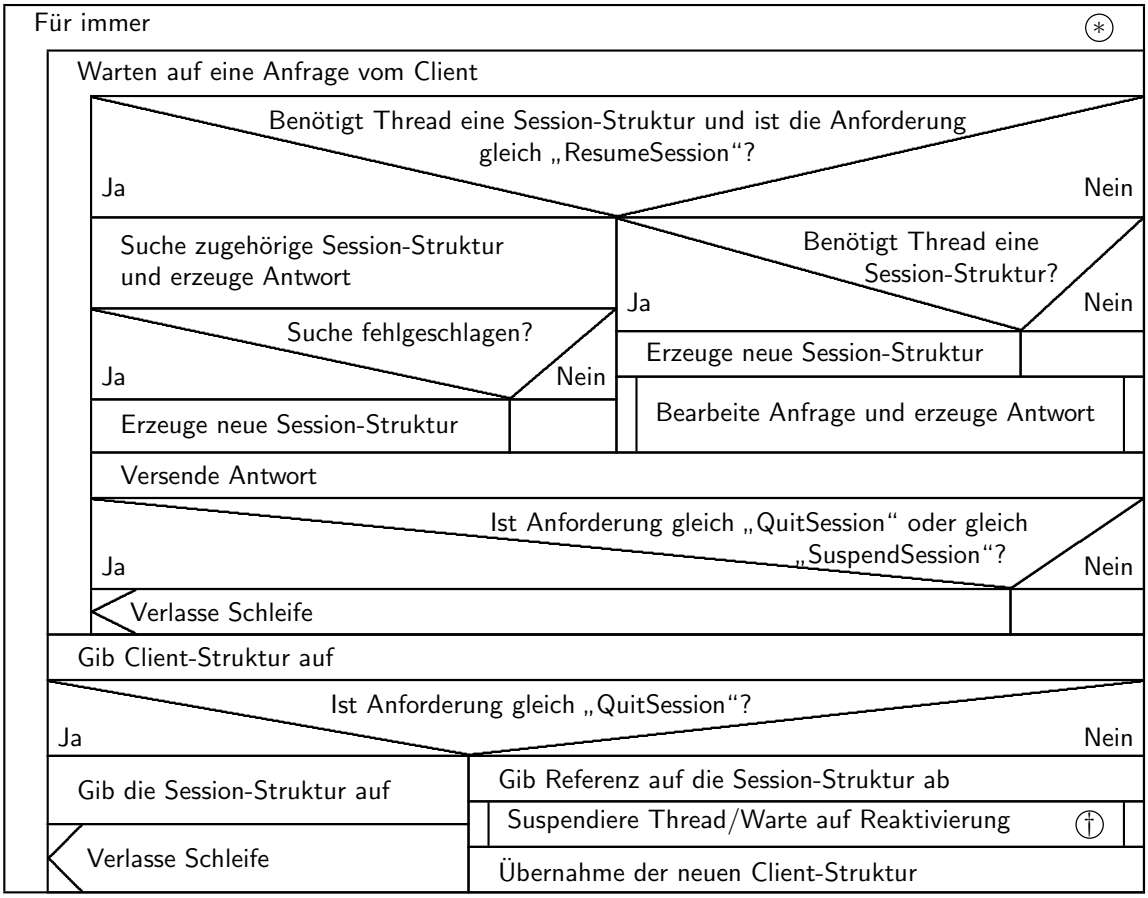


Abb. 4.12: **Ablauf ausführender Thread:** Dieses Struktogramm stellt den globalen Ablauf eines ausführenden Client-Threads dar. Je nachdem, ob ein neuer Thread erzeugt wurde oder ein suspendierter Thread reaktiviert wurde, beginnt die Arbeit an den markierten Stellen ⊛ und †.

Fall die Client-Struktur übernommen wurde, ist der Zustand der Threads beider Varianten bei dem Eintritt in die Anfrageschleife identisch.

Die erste Anfrage des Clients entscheidet, ob es sich bei dem neuen Client um einen Teil einer Multi-Client-Session handelt oder ob der Client eine eigene Such-Session durchführt. Im ersten Fall handelt es sich um eine „ResumeSession“-Anfrage. Hier wird die referenzierte Session-Datenstruktur der Multi-Client-Session gesucht und dem Thread zugewiesen. Bei allen anderen Anfragen wird für den Thread eine neue Session-Struktur erzeugt.

In der Anfrageschleife wird der Hauptteil der Arbeit erledigt. Es werden so lange Anfragen bearbeitet, bis sie durch entsprechende Verwaltungsanfragen verlassen wird. Handelt es sich hierbei um eine „SuspendSession“-Anfrage, dann entledigt sich der Thread aller Daten bzw. der Referenzen und suspendiert sich selbst.

4.5 Schnittstelle zur Außenwelt

Jeder Client, der Dienste von dem Datenbank-Server in Anspruch nehmen will, muss den Vorgaben der Schnittstelle entsprechen. Die Kommunikation ist paketorientiert und grundsätzlich so ausgelegt, dass auf jede Anfrage eines Clients eine Antwort kreiert wird (Request/Reply-Prinzip). Mit diesem Verhalten wird eine Synchronisation mit dem Client erreicht, so dass der Client über den Zustand des Servers informiert ist und auf weitere Synchronisationsmechanismen verzichtet werden kann.

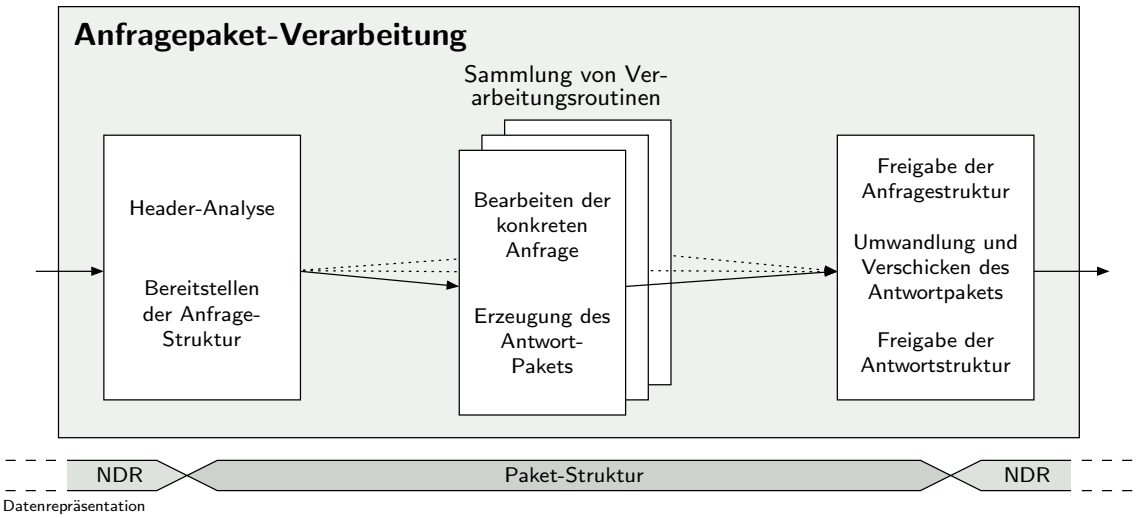


Abb. 4.13: **Vereinheitlichte Bearbeitung einer Anfrage vom Client mit zugrunde liegender Datenrepräsentation:** Die Daten des einheitlichen Kopfes einer eingehenden Anfrage werden untersucht und entsprechend der vorhandenen Identifikationsnummer wird entschieden, welche NDR-Konvertierungs- und welche Verarbeitungsfunktion zur Anwendung kommen. Das von der Verarbeitungsfunktion erzeugte Antwortpaket wird entsprechend dem Typ durch Ausführen der entsprechenden NDR-Funktion für das Senden vorbereitet.

Der Server hat bezüglich der Kommunikation keinerlei inneren Zustand, und das erlaubt es, alle Pakete gleichberechtigt zu behandeln, sie also ohne eine bestimmte Reihenfolge oder Prioritäten abzuarbeiten. Abhängigkeiten auf höheren Ebenen existieren jedoch (siehe Seite 63, Anfragen, die die Suche betreffen).

Eine TCP/IP-Verbindung, wie sie zwischen Client und Server aufgebaut wurde, stellt zwei Datenkanäle (engl.: *Streams*) zur Verfügung, wobei für jede Datenflussrichtung ein eigener Kanal benutzt wird. Zur Detektion, um welchen Typ es sich bei der Anfrage vom Client handelt, müssen alle Pakete ein gemeinsames Grundgerüst besitzen. Intern ist die Repräsentation eines Pakets eine entsprechende C-Struktur, wobei jedem Pakettyp eine eindeutige Struktur zugewiesen ist. Die Gemeinsamkeit aller Pakete ist eine Sub-Struktur vom Typ `T_CommHeader`. Diese Struktur bildet den Kopf (engl.: *Header*) aller Paketstrukturen. Alle dem Paketkopf folgenden Informationen sind bezüglich Typ, Größe, Reihenfolge und Länge paketabhängig und als Parameter des Pakets zu sehen.

Abbildung 4.13 zeigt die Abfertigung eines Anfragepakets. Zunächst wird der Paketkopf analysiert und der Pakettyp extrahiert. Der Paktetyp legt zum einen fest, welches Unterprogramm zur Konvertierung des empfangenen Pakets aus der NDR-Repräsentation in die interne C-Struktur benutzt wird und zum anderen, welches Unterprogramm für die nun folgende Bearbeitung des Pakets aufgerufen werden soll. Aufgrund der Tatsache, dass der Pakettyp abgezählt ist, lässt sich die Selektion der aufzurufenden Unterprogramme durch die Verwendung von Arrays erreichen. Das hat den Vorteil, dass die Hinzunahme weiterer Server-Dienste auf dieser Ebene lediglich das Verändern mehrerer Tabellen zur Folge hat. Die Bearbeitungsunterprogramme müssen Antwortpakete erzeugen, die wiederum mit dem dem Antwortpakettyp entsprechenden Umwandlungsunterprogramm in die NDR-Repräsentation umgesetzt und an den Client zurückgeschickt werden.

Im Folgenden werden alle implementierten Server-Dienste aufgeführt sowie deren Arbeitsweise und Ergebnisse erläutert. Die Dienste sind in zwei Klassen eingeteilt. Dies ist zum einen die Klasse der Dienste, die dem Akquirieren von Informationen vom Server dienen oder zur Verwaltung von Verbindungen zum Server benutzt werden, und zum anderen Dienste, die sich auf eine Such-Session beziehen, also Daten lesen bzw. den Suchzustand verändern.

Akquisition von Informationen

`CRPUGetImage`: Mit dieser Anfrage werden die Bilddaten des durch eine Identifikationsnummer angegebenen Bildes angefordert. Als Antwort schickt der Server ein Paket vom Typ `GUISetImage`, in dem die Identifikationsnummer, der Name und die JPEG-Daten des Bildes enthalten sind.

`CRPUGetThumbNail`: Analog zu `CRPUGetImage` können mit diesem Paket die JPEG-Daten eines verkleinerten Bildes (engl.: *Thumbnail*), das in einer Übersicht angezeigt werden kann, angefordert werden. Auch hier wird als Antwort ein Paket vom Typ `GUISetImage` gesendet. Ein Marker in diesem Paket gibt den Typ des Bildes (Original/Verkleinert) an.

`CRPUGetRepNames`: Zu Analysezwecken kann mit diesem Paket eine Liste aller vom System verwendeten Merkmalsrepräsentanten geladen werden. Das zurückgelieferte `GUISetRepNames`-Paket beinhaltet ein Array von Namen.

`CRPUJoinVisionSpeech`: Eine Besonderheit im Server stellt diese Anfrage dar. Mit ihr ist das Referenzieren eines Bildobjekts innerhalb eines Bildes durch die Angabe einer oder mehrerer Attribute möglich. Das System entscheidet anhand vorher ermittelter visueller Attribute, welches Bildobjekt den genannten Attributen am ehesten entspricht (siehe Seite 69, Referenzieren von Regionen).

`CRPUGetImageID`: Für den speziellen Fall, dass das System durch den Recorder/Player gesteuert wird (siehe Abschnitt 5.6), müssen Bild-IDs und Domänenzugehörigkeit zum gegebenen Bildobjekt gefunden werden. Dazu dient die hier beschriebene Anfrage. Auf diese wird ein `GUISetImageID`-Paket mit den genannten Parametern als Antwort erzeugt.

Verbindungs Auf-/Abbau

`CRPUSuspendSession`: Wird ein Client einer Multi-Client-Session beendet, dann muss er vor dem Beenden diese Anfrage senden. Sie Anfrage veranlasst ein Suspendieren des zugehörigen Threads wie in Abbildung 4.12 dargestellt. Der initiale Client einer Multi-Client-Session muss zunächst dieses Paket absetzen. Er bekommt in der Antwort vom Typ `GUISessionAcknowledge` eine systemweit eindeutige Identifikationsnummer der neu kreierten Session, mit der alle Clients der Session die Verbindung aufnehmen können.

`CRPUResumeSession`: Mit diesem Paket nimmt ein neuer Client einer Multi-Client-Session die Verbindung zum Server auf. Dem Server wird in diesem Paket die zuvor zugewiesene Session-Identifikationsnummer übergeben. Als Antwort wird dem Client ein `GUINOP`-Paket ohne jeglichen Parameter gesendet. Handelte es sich bei der übergebenen ID um eine ungültige Nummer, dann wird eine neue Session kreiert.

`CRPUQuitSession`: Durch die Anfrage mit diesem Pakettyp wird eine bestehende Such-Session beendet. Dieser Dienst darf bei einer Multi-Client-Session ebenfalls eingesetzt werden, jedoch muss sichergestellt werden, dass kein weiterer Client dieser Session aktiv ist. Da diese Einschränkung unter Umständen schwierig zu gewährleisten ist, sollte bei Multi-Client-Sessions auf das Absetzen dieser Anfrage verzichtet werden. Eine Session, für die sich eine bestimmte Zeit kein Client angemeldet hat, wird automatisch beendet. Diese Haltezeit ist projektierbar.

Anfragen, die die Suche betreffen

`CRPUNewSession`: Diese Anfrage wird im Allgemeinen initial vom Client durchgeführt. Damit wird die Session-Struktur initialisiert und alle eventuell bisher abgegebenen Bewertungen verworfen. Nun wird aus der Menge aller Bilder der Datenbank zufällig ein Satz von Bildern einer festen Größe ermittelt. Das Antwortpaket vom Typ `GUISetImageSet` enthält einen Vektor mit Identifikationsnummern (ID) der so ausgewählten Bilder.

`CRPUStartSearch`: Mit dem Aufruf dieses Dienstes startet der Client eine Suchiteration, die auf dem aktuellen Suchzustand und den zuvor übermittelten Bewertungen basiert. Dem Aufruf ist die Identifikationsnummer eines Bildobjekts als Parameter beigefügt. Mithilfe dieser ID kann der Server entscheiden, ob eine aktuelle Suche fortgesetzt wird oder ob durch Setzen eines neuen Bildobjekts eine neue Suche gestartet wird. Eine ausführlichere Beschreibung einer Suchiteration wird in Abschnitt 4.6.1 gegeben. Als Antwort wird ein Paket vom Typ `GUISetImageSet` (siehe CRPUNewSession) generiert, das einen Vektor von sortierten Identifikationsnummern der Bilder enthält. Die Sortierung wird anhand einer nach Ähnlichkeit geordneten Liste von Bildobjekten durchgeführt, wobei vom ähnlichsten Objekt beginnend das zugehörige Bild in dem Vektor eingetragen wird, ohne dass ein Bild mehrfach in dem Vektor aufgenommen wird.

`CRPUSetSampleJpeg`: Eine außergewöhnliche Art, eine neue Suche zu beginnen, ist das Einbringen eines nicht in der Datenbank befindlichen Bildes. Dazu überträgt der Client

mit diesem Anfragepaket ein JPEG-komprimiertes Bild zum Server. Das Bild wird dekomprimiert, dann werden alle aktivierten Merkmale des Bildes berechnet und die so gewonnenen Vektoren als Vergleichsvektoren gespeichert. Anschließend wird direkt eine Suchiteration gestartet, und damit ergibt sich ein `GUISetImageSet`-Paket als Antwort.

`CRPUGetBestRegion`: Die Liste der angezeigten Bilder wird, wie in Abschnitt 5.5 (Ablauf einer Suchiteration, Seite 86) erläutert, aus der sortierten Liste von Bildobjekten bestimmt. Mit dem Erhalt des Suchergebnises, das aus einer sortierte Liste von Bildern besteht, fehlt dem Client noch die Information, welche zu dem Bild gehörende Region für den Rang in der Bildliste ausschlaggebend war, um eine entsprechende Markierung in dem Bild erzeugen zu können. Als Ergebnis liefert die Anfrage ein Paket vom Typ `GUISetBestRegion` zurück. Hier wird neben der Bild-ID auch die Größe des Bildes übertragen, um die Relationen zu dem ebenso enthaltenen Polygonzug zu erhalten. Zu den Daten des Bildobjekts gehören neben dem Polygonzug noch eine Objekt-Identifikationsnummer, eine aktuelle Bewertung und die Referenz einer Domänenzugehörigkeit (siehe 5.6).

`CRPUGetAllRegions`: Eine vollständige Liste aller Objekte eines Bildes wird mithilfe dieser Anfrage erzeugt. Neben der Bild-ID gehört eine Bildobjekt-ID zur Anfrage. Dieser Parameter legt fest, welches Bildobjekt in der Antwortliste an erster Stelle erscheint. Hier gibt es drei verschiedene Modi: 1) das Objekt, dessen ID übergeben wurde, 2) das Objekt, das am ähnlichsten war (siehe CRPUGetBestRegion) und 3) ein beliebiges Bildobjekt soll das erste Objekt der Antwortliste sein. Als Antwort wird ein `GUISetAllRegions`-Paket zurückgeschickt. Neben der erwähnten Liste der Bildobjekte mit Polygonzug, ID, Bewertung und Domänenzugehörigkeit wird die Bildobjekt-ID, die der Anfrage als Parameter übergeben wurde, und die Objekt-ID, die das gesamte Bild repräsentiert, zurückgeliefert.

`CRPUInitWeights`: Ein Dienst, der eher zur Kontrolle bei der Systementwicklung eingesetzt werden kann, wird mit diesem Paket angestoßen. Mit dieser Anfrage wird die Suche mit dem aktuell ausgewählten Bildobjekt neu gestartet. So kann kontrolliert werden, wie eine andere Bewertung das Suchergebnis beeinflusst. Da direkt eine Suchiteration gestartet wird, ist das Ergebnis dieser Anfrage ebenfalls ein Antwortpaket vom Typ `GUISetImageSet`.

`CRPUInitRelevances`: Mithilfe dieses Dienstes werden alle abgegebenen Bewertungen, die in der Session-Struktur gespeichert werden, zurückgesetzt. Dieser Dienst wird im Allgemeinen vor dem Starten einer Suchiteration mit einem neuen Vergleichsobjekt aufgerufen. Die Antwort, die in diesem Fall generiert wird, ist ein `GUINOP`-Paket. Dieses Paket enthält weder Daten, noch soll es eine Aktion auf Seiten des Clients auslösen. Da der Client und der Server nach dem Request/Reply-Prinzip arbeiten, ist diese Antwort notwendig.

`CRPUSetRelevance`: Das Setzen der Bewertungen für entsprechende Bildobjekte wird mit diesem Paket durchgeführt. Dabei werden in einem Vektor die Wertepaare von Objekt ID und gesetzter Bewertung an den Server übertragen. Die so gesetzten Bewertungen der Bildobjekte werden in der Such-Session gespeichert, so dass eine einmal gesetzte Bewertung auch über eine Suchiteration hinaus gültig bleibt. Soll eine Bewertung zurückgenommen werden, so muss also eine Neutral-Bewertung für das entsprechende

Bildobjekt übertragen werden. Dieses Paket wird dem Client mit einer GUINOP-Antwort bestätigt.

CRPUInsertRegion: Der Benutzer der Datenbank hat die Möglichkeit, eigene Bildobjekte zu definieren. Dazu muss neben der Bild-ID ein geschlossener Kantenzug, durch den das Bildobjekt definiert ist, an den Server übertragen werden. Der Server lädt dann das referenzierte Bild und berechnet für den spezifizierten Kantenzug für alle aktivierten Merkmale die Vektoren (siehe Abschnitt 4.6.3). Für das Bildobjekt wird ein Datensatz in der MySQL-Datenbank angelegt, der sowohl den übergebenen Kantenzug als auch die erzeugten Referenzierungsmerkmale enthält. Des Weiteren werden die berechneten Merkmalsvektoren in den entsprechenden Repräsentantentabellen der Datenbank gespeichert. Als Antwort erhält der Client neben der Bild-ID auch die Identifikationsnummer des neu kreierten Bildobjekts.

CRPUGetRepWeights/CRPUGetRepCompWeights: Mit diesen Diensten ist es zu Kontrollzwecken möglich, die aktuelle Belegung der Gewichte der aktuellen Suche zu bestimmen. Dabei wird mit CRPUGetRepWeights die Belegung der Gewichte auf Merkmalsebene angefordert. Mit CRPUGetRepCompWeights kann für den übergebenen Merkmalsrepräsentanten die Parametrisierung der Distanzfunktion als Spur der Gewichtsmatrix $\underline{W}$ angefordert werden (siehe Gleichungen 3.5 und 4.2). Als Antwort erhält der Client das entsprechende Paket GUISetRepWeights bzw. GUISetRepCompWeights mit dem angegebenen Gewichtsvektor in der entsprechenden Dimension.

Paketabhängigkeiten und Synchronisation

Die erwähnten Paketabhängigkeiten beschränken sich auf die CRPUGetBestRegion-Anfrage. Naturgemäß kann nur dann das ähnlichste Bildobjekt eines Bildes ermittelt werden, wenn bereits eine Ähnlichkeitsbestimmung durchgeführt, also eine Suchiteration gestartet wurde. Der Datenbank-Client muss in dieser Anfrage dafür sorgen, dass sie erstmalig nach einem initialen Suchschritt durchgeführt wird.

Der Zugriff auf die Session-Daten, die durch Anfragen vom Client (unter „Anfragen, die die Suche betreffen“ aufgelistet) verändert werden können, muss synchronisiert werden, um eine Datenintegrität zu gewährleisten. Davon ist lediglich der Multi-Client-Betriebsmodus betroffen, denn im Single-Client-Modus ist die Abarbeitung der Anfragen durch die Request/Reply-Struktur serialisiert und damit eine Synchronisation implizit gegeben. Unter Datenintegrität wird in diesem Fall die Konsistenzhaltung der Daten verstanden, so dass ein Systemabsturz oder das Übermitteln inkorrekter Daten ausgeschlossen wird.

4.6 Besondere Server-Dienste

In diesem Abschnitt werden drei Dienste aufgrund ihrer herausstechenden Funktionalität vorgestellt. Dabei handelt es sich um den Kern der Bildsuche, das Referenzieren von Bildregionen anhand einer Liste von Regionenattributen als Unterstützung der natürlichen Interaktion und schließlich das Ergänzen des Bilddatenbestands durch Hinzufügen eines Bildobjekts.

4.6.1 Aufbau einer Suchiteration

Eine Suchiteration wird im Regelfall als Dienst `CRPUStartSearch` durchgeführt (siehe Abschnitt 4.5). Anhand der übergebenen Identifikationsnummer (ID) der Beispielregion wird

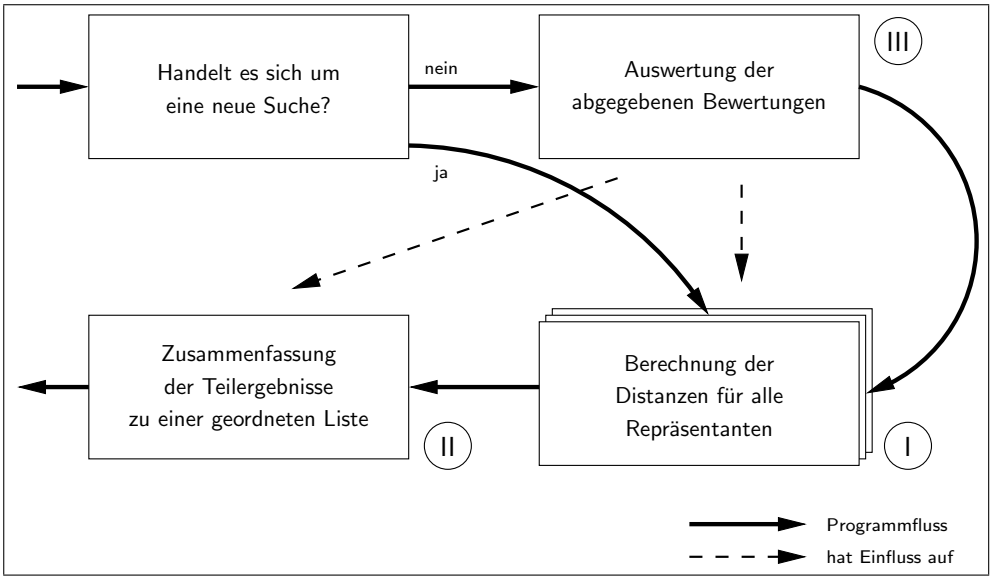


Abb. 4.14: **Ablauf einer Suchiteration:** Die Suchiteration wird mit der Angabe eines Suchobjekts angestoßen. Wenn die aktuelle Suche fortgesetzt wird, werden die zuvor abgegebenen Bewertungen dazu benutzt, die Gewichte im System neu anzupassen und ein neues Vergleichsobjekt zu bestimmen. Die bezüglich der unterschiedlichen Repräsentanten ermittelten Distanzen werden in geeigneter Weise zu einer nach Ähnlichkeit geordneten Liste zusammengefasst.

entschieden, ob es sich um eine bereits bestehende oder eine neue Suche handelt. Abbildung 4.14 zeigt den Aufbau des Suchiterationsdienstes, der einem Teil der in Abbildung 2.3 gezeigten Iterationsschleife des Mars-Systems [Hua96] entspricht. Die Umsetzung der drei in der Abbildung aufgeführten Aktionen unterscheidet sich jedoch von der ursprünglichen MARS-Implementierung. Alle im Folgenden aufgeführten Techniken sind ausführlich in der Arbeit von Käster [Käs05] erläutert und hier nur in einer Übersicht angeführt.

Hierarchische Distanzberechnung

Ausgehend von der in Abbildung 3.4 gezeigten Distanzbildung des MARS-Systems wird die Hierarchie um eine Ebene vereinfacht. Dabei werden die Gewichtswerte der Merkmale, u_i, und

die der Merkmalsrepräsentanten, v_{ij}, aus den Gleichungen 3.2 und 3.3 in einem neuen Wert der Merkmalsrepräsentanten zusammengefasst.

$$\begin{aligned} D(O_k, Q) &= \sum_{i=1}^{I} u_i \sum_{j=1}^{J_i} v_{ij} D_{ij}(O_k, Q) = \sum_{i=1}^{I} \sum_{j=1}^{J_i} u_i v_{ij} D_{ij}(O_k, Q) \\ &= \sum_{i=1}^{I} \sum_{j=1}^{J_i} v'_{ij} D_{ij}(O_k, Q) = \sum_{j'=1}^{J'} v'_{j'} D_{j'}(O_k, Q), \;\; J' = \sum_{i=1}^{I} J_i \end{aligned} \tag{4.1}$$

Die Berechnung der Distanzen in den Repräsentantenräumen wurde, wie bereits angedeutet, durch die Verwendung einer Gewichtsmatrix erweitert. Damit werden Adaptionen, wie sie im Folgenden angeführt werden, möglich. Die Distanzfunktion $D_{j'}$ ist also in der allgemeinen, funktionsunabhängigen Form folgendermaßen definiert:

$$D_{j'}(O_k, Q) = d_{j'}(\vec{r}^{\,k}_{j'}, \vec{q}_{j'}, \underline{W}_{j'}) \tag{4.2}$$

Adaption des Anfragevektors Q und der Komponenten-Gewichtsmatritzen $\underline{W}$

Die Evaluierungen von Käster basieren auf den Adaptionen, die in den Arbeiten von Rui und Huang [Rui00, Rui01] vorgestellt wurden. Dabei wird aus der Menge der positiv bewerteten Bildobjekte in jedem Merkmalsraum zunächst ein idealer Anfragevektor erzeugt und anschließend eine Distanzminimierung der Beispielsvektoren zu dem Anfragevektor durchgeführt. Hinter dieser Technik verbirgt sich das Modell, dass die für die Suche relevanten Bildobjekte Häufungspunkte in den beteiligten Räumen bilden sollen. Das entstehende Abstandsmaß entspricht einem Mahalanobis-Abstand, der einen Sonderfall des generalisierten euklidischen Abstands darstellt. Die Basis für diese Adaptionen bildet die Menge der H positiv bewerteten Bildobjekte $X = \{O_1, O_2, \ldots, O_H\}$.

Es ergibt sich als optimale Lösung:

$$\vec{q}^{\,*}_{j'} = \frac{\sum_{h=1}^{H} \pi_h \vec{r}^{\,h}_{j'}}{\sum_{h=1}^{H} \pi_h} \tag{4.3}$$

$$\underline{W}^{*}_{j'} = (\det(\underline{C}_{j'}))^{\frac{1}{N_{j'}}} \, \underline{C}^{-1}_{j'} \tag{4.4}$$

mit

$$c_{j'_{mn}} = \frac{\sum_{h=1}^{H} \pi_h (\vec{r}^{\,h}_{j'_m} - \vec{q}^{\,*}_{j'_m})(\vec{r}^{\,h}_{j'_n} - \vec{q}^{\,*}_{j'_n})}{\sum_{h=1}^{H} \pi_h} \tag{4.5}$$

Da oft aufgrund der kleinen Menge von positiven Bewertungen die Inverse der Kovarianzmatrix $\underline{C}_{j'}$ nicht berechenbar ist (vergleiche Gleichung 4.4), wurde eine Rückfalltechnik eingebaut, die bei solchen Problemen zum Zuge kommt und eine benutzbare Gewichtsmatrix $\underline{W}_{j'}$ erzeugt. Dabei kommen Techniken zum Einsatz wie die Beschränkung der Kovarianzmatrix auf eine Diagonalmatrix, bei der lediglich die Hauptdiagonale nach der Gleichung 4.5 besetzt wird, oder die Regularisierung der Kovarianzmatrix, bei der die Menge der zugrunde liegenden Vektoren $\vec{r}_{j'}$ bzw. die Elemente $c_{j'_{mn}}$ der Kovarianzmatrix verrauscht werden.

Die hier präsentierten Mechanismen zeigen in [Käs05] die für die iterativen Bildsuchsysteme elementar wichtigen Adaptionsfähigkeiten. Diese Adaption setzt die Verwendung von Gewichtsmatritzen bei der Parametrisierung der Distanzberechnung voraus (siehe Abschnitt 4.2.3).

Lernen der Repräsentantengewichte

Die Gewichtung der Merkmalsrepräsentanten, also die Bestimmung der $v'_{j'}$ aus Gleichung 4.1, wird nach einem heuristischen Verfahren durchgeführt. Grundlage für die Berechnung dieser Gewichte bildet die Menge aller H bewerteten also auch negativ bewerteten Bildobjekte. Entscheidend an der Adaption, die in Gleichung 4.6 dargestellt ist, ist der Änderungsterm, der durch die Lernrate $\alpha \in [0, 1]$ begrenzt ist.

$$v'^{+1}_{j'} = v'_{j'} + \alpha \sum_{h=1}^{H} \pi_h \Psi(R_{j'}(O_h, Q)) \tag{4.6}$$

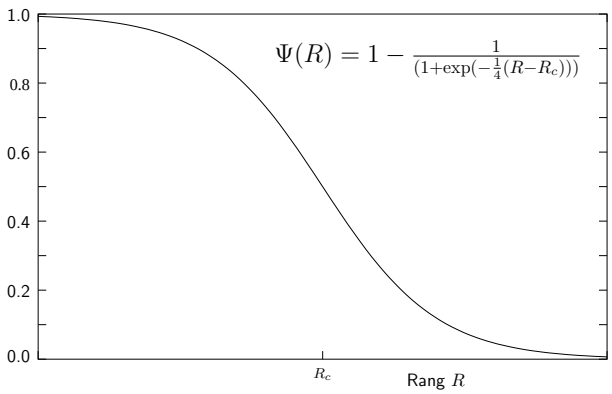


Abb. 4.15: **Rang-Gewichtung-Abbildung**

Die Änderung ist abhängig davon, welchen Rang $R_{j'}(O_h, Q)$ das bewertete Bildobjekt O_h in der nach Distanz sortierten Liste aller Bildobjekte in dem betrachteten Repräsentantenraum j' einnimmt. Die hier eingesetzte sigmoide Funktion Ψ bildet die ermittelten Ränge so ab, dass nur solche Objekte maßgeblich in die Gewichtsbildung eingehen, die einen niedrigen Rang erhalten haben, also vom System in dem betrachteten Raum als recht ähnlich eingestuft worden sind. Abbildung 4.15 zeigt den Verlauf der Funktion in Abhängigkeit des Ranges. Der Rang R_c, der als Parameter der Funktion zu verstehen ist, legt den Wendepunkt der Kurve fest. Die Gewichtung ist bei diesem Rang auf 50% herabgesetzt. Der so über den Rang ermittelte Wert wird mit der vom Benutzer abgegebenen Relevanzbewertung $\pi_h \in \{2, 1, 0, -1, -2\}$ gewichtet. Schließlich bildet die Summe über alle bewerteten Bildobjekte die Änderung des Relevanzgewichtes.

Implementierungsdetails

Der in Abbildung 4.14 abgebildete Dienst basiert auf wenigen Parametern. Lediglich das ausgewählte Beispielobjekt und eventuell abgegebene Relevanzbewertungen von Bildern des im vorherigen Suchschritt ermittelten Ergebnisses beeinflussen das neu zu berechnende Suchergebnis. Alle weiteren Daten, wie beispielsweise aktuelle Gewichtungen der Repräsentanten werden von diesem Dienst selbst verwaltet und in einer eigenen Datenstruktur, die den Session-Daten zugeordnet ist (vergleiche Abbildung 4.10), gespeichert.

4.6.2 Referenzieren von Regionen

Der hier vorgestellte Dienst stellt eine Besonderheit in der Welt der Bilddatenbanken dar. Mithilfe dieses Dienstes ist es möglich, das Objekt eines Bildes zu finden, das den übergebenen beschreibenden Attributen am ehesten entspricht. Dieser Dienst stellt eine Unterstützung der natürlichen Bedienbarkeit des Systems dar, da mit Hilfe dieser Funktionalität ein Bildobjekt, dessen Region im Bild durch einen Kantenzug gekennzeichnet ist, direkt durch die sprachliche Angabe von zutreffenden Attributen referenzierbar ist.

Die Plazierung dieser Funktionalität in den Bilddatenbank-Server liegt darin begründet, dass für die Verarbeitung der Bildattribute entsprechend zugeschnittene Merkmale der existierenden Bildobjekte erzeugt werden müssen. Die Erzeugung dieser Merkmale ist zur Laufzeit aufgrund der dazu benötigten Zeit nahezu unmöglich. Daher werden diese Merkmale wie die Merkmalsrepräsentanten auch beim Einfügen eines Bildobjekts berechnet. Des Weiteren steht die Funktionalität durch die gewählte Plazierung neuen Typen von Bilddatenbank-Clients zur Verfügung.

Die in diesem System eingesetzten Attribute beschreiben Charakteristika wie Farbe, Intensität, Größe und räumliche Anordnung, die durch horizontale und vertikale Positionsangaben realisiert ist. Alle diese Angaben sind sehr unscharf und stark von weiteren im Bild enthaltenen Regionen, dem Regionenkontext, abhängig. Das begründet den Einsatz eines probabilistischen Verfahrens zur Lösung des Referenzproblems.

Das von Wachsmuth und Sagerer [Wac02] vorgestellte Verfahren modelliert eine ähnlich geartete Problematik, die Integration von sprachlicher und visueller Beschreibung von Objekten, als prohabilistischen Prozess in Form eines Bayes-Netzes.

Abbildung 4.16 zeigt das im INDI-System realisierte Bayes-Netzwerk. Auf der visuellen Ebene, in der Abbildung oben plaziert, wird für jede in dem Bild enthaltene Region ein Satz von Zustandsvariablen erzeugt. Dabei enthält ein solcher Satz für jedes mögliche beschreibende Attribut eine Zustandsvariable. Im unteren Teil der Abbildung ist die sprachliche Modellierung einer Region durch einen weiteren Satz von Zustandsvariablen abgebildet. Die Knoten der visuellen und die der sprachlichen Ebene stehen je Attribut in einer $1:1$-Beziehung. Die Diskretisierung der Zustandsvariablen ist zwischen visueller und sprachlicher Modellierung unterschiedlich gewählt. Die Anzahl der möglichen Zustände dieser Knoten ist jedoch unterschiedlich ausgelegt. Bezüglich der Regionen ist eine $1:N$-Beziehung modelliert, bei der die sprachliche Äußerung einer der N möglichen visuell definierten Regionen zugeordnet werden kann.

Die Tabellen der bedingten Wahrscheinlichkeiten (engl.: *Conditional Probability Tables, CPT*), die für die Auswertung des Bayes-Netzes notwendig sind, wurden von Hand erstellt.

$$r* = \underset{r \in \{1,2,\cdots N\}}{\operatorname{argmax}} P(S = r|e)$$

Das Resultat des durchgeführten Inferenzvorgangs ist, nachdem die Verbundwahrscheinlichkeit des Netzes maximiert wurde, in der Auswahlvariablen S zu finden. Der wahrscheinlichste Index $r*$ ist also der, bei dem sich die maximale Wahrscheinlichkeit bei der angenommenen Referenz $(S = r)$ unter den beobachteten Evidenzen e ergibt.

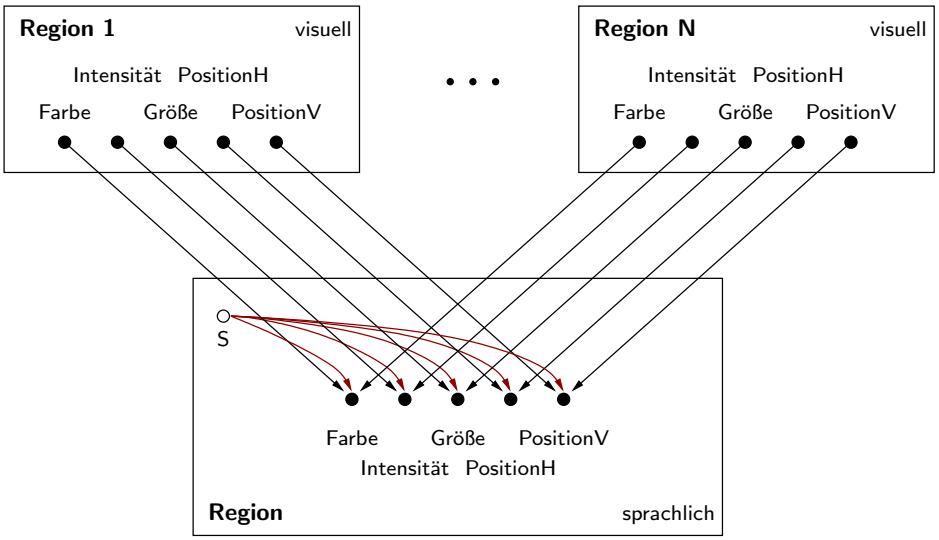


Abb. 4.16: **Bayes-Netz zur Beschreibung von Regionen:** Im oberen Teil der Abbildung sind die Instanzen der im Bild enthaltenen Regionen zu sehen, für die je ein Satz von Zustandsvariablen erzeugt wird. Denen gegenüber steht die sprachlich beschriebene Region im unteren Teil der Abbildung. Die Auswahlvariable S$\in \{1, \cdots, N\}$ wird durch den Inferenzvorgang auf den Index der Region gesetzt, die am wahrscheinlichsten referenziert wurde. Dieser Vorgang beruht auf den beobachteten Evidenzen sowohl auf visueller Ebene durch die Merkmalsextraktion ermittelt als auch auf sprachlicher Ebene aus den angegebenen Attributen extrahiert.

4.6.3 Einfügen eines neuen Bildobjekts

Der Datenbank-Client erlaubt es, neue Bildregionen durch Erzeugen von Kantenzügen zu erstellen. Diese Aktion erfolgt zur Laufzeit, und da das Systemkonzept es erfordert, dass alle Bildobjekte, die an der Suche beteiligt sein sollen, in der Datenbank eingetragen sind, ist die Berechnung der Repräsentantenvektoren, mit dem dazugehörenden Einfügen der erzeugten Datensätze, ebenfalls zur Laufzeit erforderlich. Die Tatsache, dass der Server nicht einem Client allein vorbehalten ist, erfordert es, entsprechende Synchronisations oder Ausschlussmechanismen bei diesem Dienst umzusetzen.

Abbildung 4.17 zeigt schematisch den Ablauf des Vorgangs. Um die Berechnung der Repräsentanten durchführen zu können, muss zunächst dafür gesorgt werden, dass alle zur Berechnung notwendigen Programmbibliotheken geladen werden. Hierbei handelt es sich um eine Aktion mit einer erhöhten Tragweite, denn die hier geladenen Bibliotheken werden server-weit benutzt. Nach der Berechnung aller notwendigen Vektoren werden diese in die Tabellen der Datenbank eingetragen.

Dem Suchkonzept entsprechend werden grundsätzlich alle in der Datenbank befindlichen Bildobjekte in die Suche einbezogen. Wenn sich während der Ausführung einer Suchiteration die Anzahl der Objekte, beispielsweise durch das Hinzufügen eines Objekts von einem unabhängigen Datenbank-Client ändert, führt das zu einem nicht tolerierbaren Fehler. Hier gibt es zwei Ansätze zur Lösung des Problems. Das Hinzufügen eines Bildobjekts kann durch einen Mechanismus des gegenseitigen Ausschlusses geschützt werden, so dass ein weiterer Client

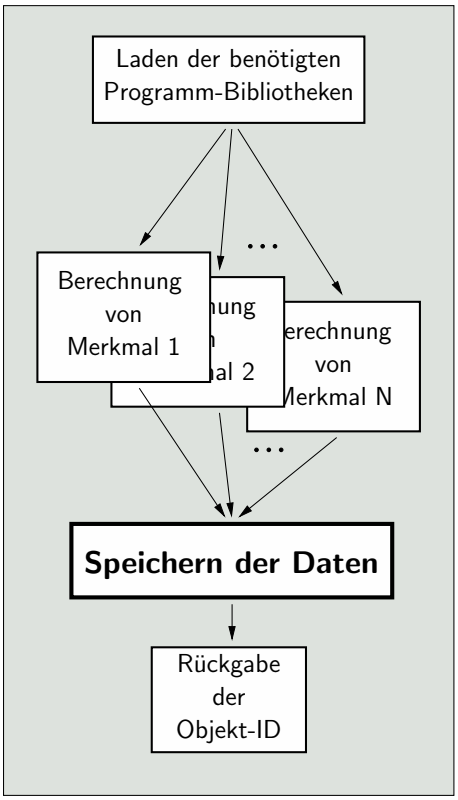


Abb. 4.17: **Einfügen eines neuen Bildobjekts:** Die Abbildung zeigt den linearen Ablauf beim Hinzufügens eines Bildobjekts. Das hervorgehobene Speichern der Daten erfordert die besondere Behandlung.

keinen undefinierten Datenzustand vorfinden kann kann. Ein anderer Ansatz, ist, den hinzugefügten Bildobjekten eine Kennung zuzuordnen, so dass diese Bildobjekte nur für den Client sichtbar sind, der diese auch in die Datenbank eingefügt hat. Der Hauptgrund für die Umsetzung des zweiten Punkts ist, dass das Verhalten der Datenbank nicht durch die Benutzung anderer beeinflusst werden soll. Die damit verbundenen administrativen Aufgaben, wie das Löschen der Bildobjekte, sobald der erzeugende Client abgemeldet wurde, wurden dabei in Kauf genommen.

Die Aktion schließt mit der Rückgabe der Identifikationsnummer des neu angelegten Objekts, die zur Weiterverarbeitung im Client notwendig ist.

Kapitel 5

Datenbank-Client

Die im letzten Kapitel vorgestellte Server-Komponente des Bilddatenbanksystems stellt die Funktionalität, die zur eigentlichen Bildsuche notwendig ist, zur Verfügung. Die Ablaufsteuerung, die den Datenfluss und damit auch die Zugriffe auf den Server koordiniert, wird von dem in diesem Kapitel vorgestellten multimodalen Bilddatenbank-Client durchgeführt (siehe Abbildung 5.1). Neben der Anforderung, Ergebnisse in geeigneter Weise dem Benutzer zu

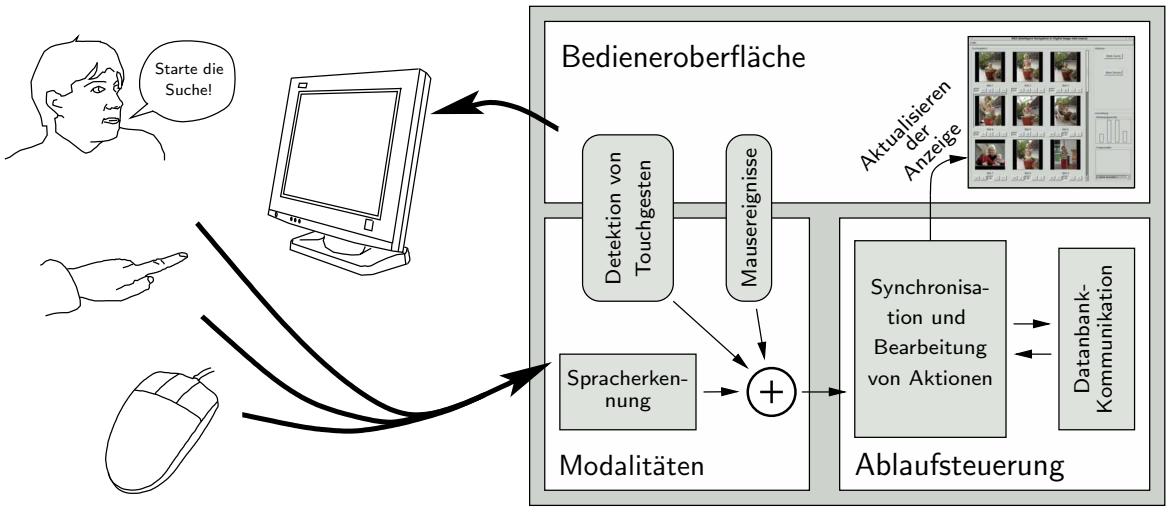


Abb. 5.1: **Bilddatenbank-Client zwischen Benutzer und Server:** Hier dargestellt ist der Bilddatenbank-Client, der das Bindeglied zwischen Benutzer und Datenbank-Server bildet. Der Client ist in der Lage, die multimodalen Interaktionen vom Benutzer zu erkennen, diese zu synchronisieren und zu fusionieren und entsprechend resultierende Aktionen auszuführen. Ergebnisse werden dem Benutzer in einer Bedienoberfläche präsentiert.

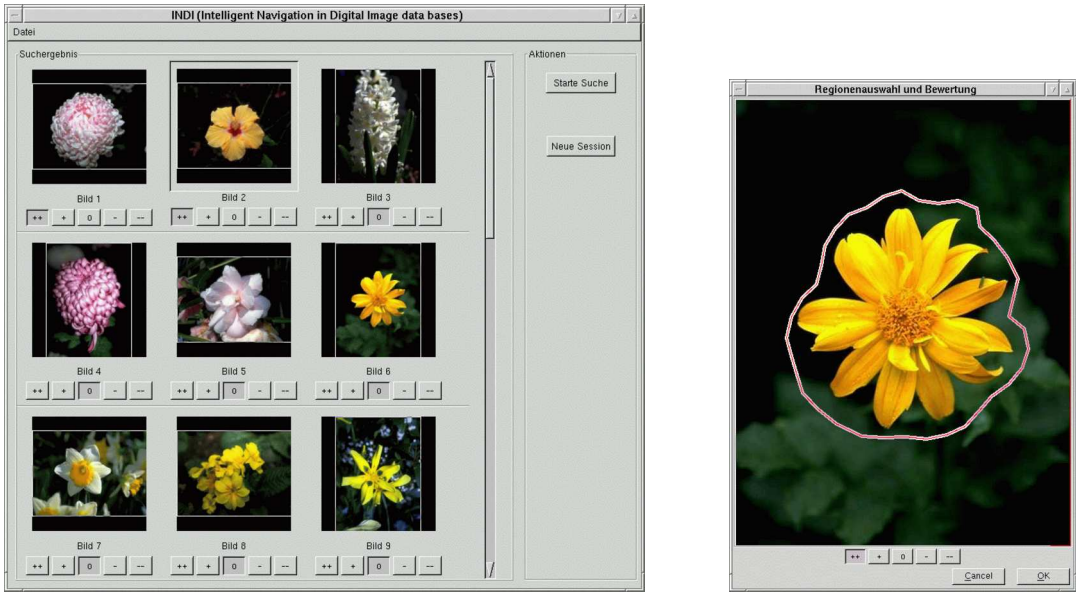
präsentieren, kommt dem Client die besondere Aufgabe zu, verschiedene asynchrone Eingabemodalitäten vorzuverarbeiten, diese dann zu fusionieren und koordiniert weiterzuverarbeiten.

Aufgrund der Tatsache, dass die unterschiedlichen Modalitäten asynchrone Ereignisse auslösen, bietet es sich an, die Datenquellen je von eigenen Threads bearbeiten zu lassen. Damit erschließt sich die Möglichkeit, alle dann bereits vorverarbeiteten Ereignisse an einer zentralen Stelle der Applikation zu synchronisieren bzw. weiterverarbeiten zu können.

5.1 Bedienoberfläche

Im Gegensatz zu vielen Bilddatenbanken, die sich dem Benutzer als Web-Anwendung präsentieren, also mithilfe eines Browsers bedienbar sind, muss der INDI-Datenbank-Client aufgrund der besonderen Art der Interaktion als eigenständige Applikation ausgelegt werden.

Die unabdingbare Anforderung, das Suchresultat als Liste von Bildern darzustellen, und die aus Sicht der Entwicklung und Evaluierung resultierende Anforderung, eine Bedienungsmöglichkeit mit der Maus zu gewährleisten, legt es nahe, zur Realisierung ein Toolkit, mit der die Entwicklung einer grafischen Benutzerschnittstelle durchgeführt werden kann, zu benutzen. Neben



(a) Hauptfenster

(b) Selektion und Bewertung der Bildregionen

Abb. 5.2: **Grafische Bedienoberfläche**

dem erheblich geringeren Aufwand für die Realisierung spielt der Wiedererkennungwert, den die Benutzer bereits erfahren, wenn das Hauptfenster der Applikation erscheint, eine wichtige Rolle für deren Sicherheit bei der Bedienung. Abbildung 5.2 zeigt die Bedienoberfläche, die für die Interaktion mit dem Bilddatenbank-Client geschaffen wurde. Teil (a) der Abbildung zeigt das Hauptfenster, in dem das Suchergebnis bzw. die initiale Auswahl von Bildern gezeigt wird. Das Bild, das die gewählte Beispielregion enthält, wird durch einen hinterlegten Rahmen gekennzeichnet. Unter jedem Bild der Liste besteht die Möglichkeit, eine Relevanzbewertung durchzuführen, ebenso wird hier die aktuelle Bewertung angezeigt. Abbildung 5.2(b) zeigt die

Großansicht eines Bildes des Hauptfensters. Hier können alle Bildregionen eingesehen und bewertet werden. Des Weiteren können in diesem Fenster durch Umranden neue Bildregionen erzeugt werden.

Bei der Wahl des Toolkits waren Aspekte wie Betriebs-Plattform, Programmiersprache, Parallelisierung und freie Verfügbarkeit von Bedeutung. Ein wichtiger Punkt, der ebenfalls für Toolkit GTK+ sprach, ist die weite Verbreitung und die damit vorhandene Erfahrung, so dass auch komplexe Probleme schnell lösbar sind.

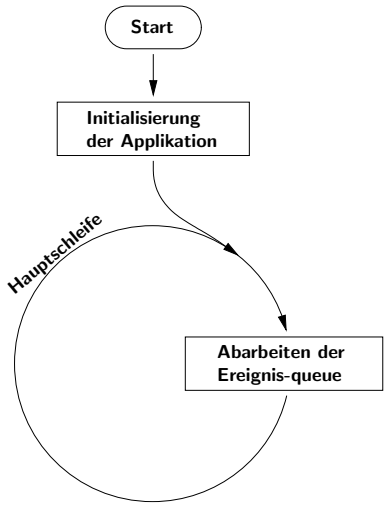


Abb. 5.3: **GTK-Prozessstruktur**

Die Prozess-Grundstruktur bei Applikationen, die für die Benutzung unter einer grafischen Oberfläche erstellt werden, ist grundsätzlich von der Form, die in Abbildung 5.3 gezeigt ist. Nach einem Programmteil, der die Initialisierung der Applikation und Erstellung des Hauptfensters zur Aufgabe hat, wird die Hauptschleife betreten, die lediglich beim Beenden der Applikation verlassen wird. Die Hauptschleife mit der darin enthaltenen Verwaltung von eintreffenden Ereignissen wird von dem verwendeten Toolkit durchgeführt. Vom Entwickler sind so genannte callback-Funktionen bereitzustellen, die bei Eintreffen eines bestimmten Ereignises vom Toolkit aufgerufen werden. Alle vom Toolkit verwalteten Ereignisse werden in einer Warteschlange (engl.: *queue*) organisiert und entsprechend des FIFO-Prinzips abgearbeitet.

Die Verarbeitung von anderen Modalitäten, wie beispielsweise der Sprache, muss der Eigenart entsprechend asynchron durchgeführt werden. Entscheidend hierbei ist, dass der Zeitpunkt des Auftretens eines solchen asynchronen Ereignisses festgehalten werden kann.

5.2 Spracherkennung

Die Spracherkennung erfolgt in dem INDI-System mit dem von Fink vorgestellten Spracherkenner [Fin99]. Dabei handelt es sich um ein sprecherunabhängiges System auf der Basis von Hidden-Markov-Modellen. Eine besondere Fähigkeit des Systems, die in dem INDI-System zur vollen Geltung kommt, ist eine Sprachverstehens-Komponente [Wac98]. In den meisten Systemen, die mit einer Sprachinteraktion ausgestattet sind, gehorchen gültige Äußerungen einer Grammatik. Dieses zusätzliche, domänenabhängige Wissen wird dem System zugänglich gemacht, um die Erkennungsleistung und damit die Robustheit der Spracherkennung zu verstärken.

Einsatz des Spracherkenners

Der Spracherkenner ist als eigenständige Applikation ausgelegt. Als Schnittstellen bietet dieser Erkenner zum einen DACS, ein Kommunikationssystem für verteilte Anwendungen [Jun98],

an, und zum anderen können die Ergebnisse auf den Standard-Ausgabekanal der Applikation gelegt werden.

Mit der Teilung des Bilddatenbanksystems in eine Client-/Server-Architektur besteht keinerlei Notwendigkeit, das Spracherkennungssystem auf einer anderen Plattform als der des Clients selber zu aktivieren. Daher wurde als Applikationsschnittstelle zwischen Spracherkenner und Bilddatenbank-Client die zweite Variante, der Standard-Ausgabekanal, gewählt.

Die technische Anbindung des Erkenners an die Bilddatenbank soll nun so erfolgen, dass das Betreiben des Bilddatenbank-Clients vollkommen unabhängig davon ist, ob der Spracherkenner bereits gestartet wurde oder nicht. Dieses Vorgehen hat den Vorteil, dass keinerlei Konfigurationen bezüglich der Benutzung des Spracherkenners vorzunehmen sind, wenn der Bilddatenbank-Client mittels veränderter Eingabemodalitäten benutzt werden soll.

Für die Interprozess-Kommunikation, die für den Datenaustausch der eigenständigen Prozesse notwendig ist, wird eine so genannte *named pipe*, auch *FIFO* genannt (engl.: *First In First Out*), verwendet. Dieses Betriebsmittel stellt einen bidirektionalen Kommunikationskanal bereit. Die *named pipe* wird über das Filesystem verwaltet und kann von jeder Applikation, der Name und Lokalität des Eintrags bekannt ist, geöffnet und damit benutzt werden.

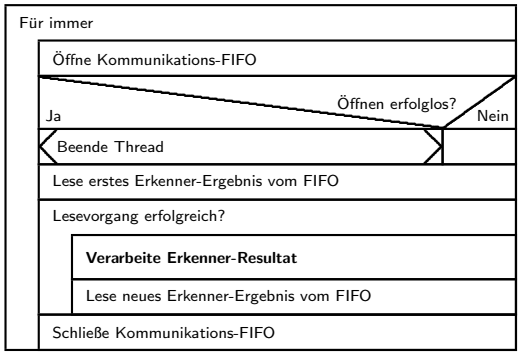


Abb. 5.4: **Spracherkennungs-Thread**

Die Anbindung an die Spracherkenner-Applikation wird in Abbildung 5.4 gezeigt. Auf der Seite des Bilddatenbanksystem kommt ein eigenständiger Thread zum Einsatz, der in dem Fall, dass die *named pipe* existiert und zu öffnen ist, eine Schleife betritt und hier auf Ergebnisse vom Spracherkenner wartet. Sobald etwas eingegangen ist, wird dies semantisch analysiert. In dem Fall, dass es sich um eine korrekte Anweisung handelt, wird ein entsprechendes Paket generiert und dies mittels der in Abschnitt 5.5 angeführten Kommunikationsschnittstelle zur Weiterverarbeitung geleitet.

Semantische Analyse des Ergebnisses der Spracherkennung

Wenn eine projektierbare Zeit lang das aufgenommene Audiosignal einen Ruhewert unterschritten hat, wird die zu dem Zeitpunkt wahrscheinlichste Interpretation des Audiosignals in Form von Symbolen aus dem zugrunde liegenden Lexikon und den angewendeten Grammatikregeln vom Spracherkenner ausgegeben. Nicht immer kann der Erkenner den gesprochenen Satz wiedergeben, auch dann nicht, wenn dieser ein gültiges Kommando darstellt. Das liegt beispielsweise an Störungen des Audiosignals, oder daran, dass Worte nicht klar genug gesprochen wurden. Da die der Erkennung zugrunde liegende Grammatik nicht strikt angewendet wird, sondern lediglich Wahrscheinlichkeiten beim Erkennungsvorgang beeinflusst, können auch Sätze ausgegeben werden, die der Grammatik nicht entsprechen.

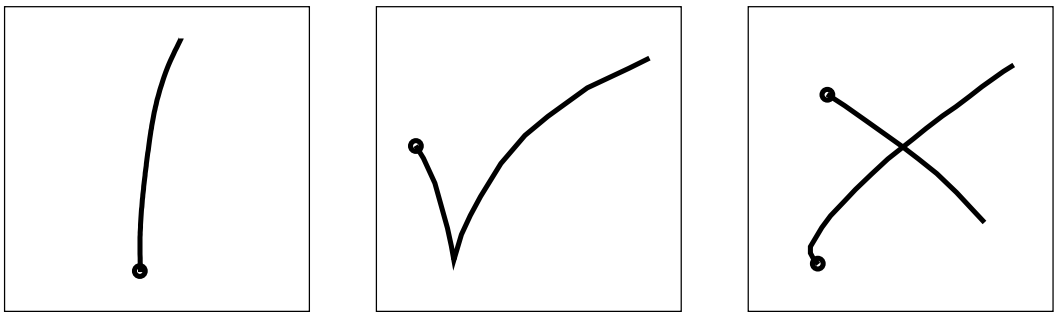
Ebenso wie bei den im folgenden Abschnitt beschriebenen Touchscreen-Gesten ist es auch hier nicht wünschenswert, dass aufgrund von Erkennungsfehlern nicht gewollte Aktionen ausgeführt werden. Daher ist die Sprachweiterverarbeitung so ausgelegt, dass alle Ergebnisse des Erkenners, die nicht der Grammatik entsprechen, verworfen werden.

Mit dem im Abschnitt 6.2 vorgestellten Parsergenerator wird die Ausgabe des Spracherkenners untersucht und in einfach weiterzuverarbeitende Daten umgesetzt.

5.3 Gesten am Touchscreen-Display

Bei der Verwendung eines Touchscreen-Displays für die Bedienung der Bilddatenbank können viele Standardaktionen von der Mausbedienung übernommen werden. So können beispielsweise die Bewertungen der Bildobjekte durch Berühren eines der Bewertungsschalter unter dem entsprechenden Bild durchgeführt werden. Ebenso wird ein Scroll-Balken durch Berühren und Schieben dazu verwendet, zu weiteren Bildern des Suchergebnisses zu gelangen. Es gibt bei der INDI-Client-Applikation jedoch Mausaktionen, die nicht direkt für die Bedienung mittels Touchscreen-Displays ausgelegt sind. Hierbei handelt es sich um bildbezogene Aktionen, die bei der Mausbedienung durch Doppelklick bzw. durch Klicken mit der rechten Maustaste ausgelöst werden.

Einige bildbezogene Aktionen können in dem INDI-System durch das Ausführen einer Touchscreen-Geste auf dem entsprechenden Bild, ausgelöst werden. Die zugrunde liegende



(a) Bild-Großansicht (b) Auswahl Beispielobjekt (c) Auswahl Beispielobjekt

Abb. 5.5: **Bildbezogene Aktionen mittels Touchscreen-Gesten**

Idee hat ihren Ursprung in dem schriftlichen Ausfüllen von Formularen, wobei an bestimmten Stellen des Formulars zum Beispiel ein Kreuz für eine 1:N-Auswahl gemacht werden soll. Abbildung 5.5 zeigt die realisierten Touchscreen-Gesten.

Da das Basissystem, der Window-Manager bzw. das Betriebssystem, die Aktionen, die mithilfe des Touchscreens durchgeführt werden, direkt auf Mausaktionen abbildet, kann die Gestenerkennung nicht als eigenständige Applikation ausgeführt werden. Vielmehr muss der Applikationsteil, der die Bedienoberfläche realisiert, die Erkennung der Touch-Gesten übernehmen.

Gewinnung der Rohdaten

Die Grundlage für die sich anschließende Weiterverarbeitung bilden Kantenzüge als Beschreibung der Touch-Geste. Eine Touchscreen-Geste beginnt mit dem Aufsetzen des Fingers auf das Display und endet entsprechend dann, wenn der Finger vom Display entfernt wird. Die so festgelegten Start- und Endpunkte der Geste werden vom System der Applikation als Drücken und Loslassen der Maustaste übermittelt. Die sich dazwischen befindlichen Punkte werden durch periodisches Abfragen der aktuellen Mausposition ermittelt.

Für die Bilddarstellung in der Client-Applikation wurde eigens ein GTK-Widget erstellt. Mithilfe dieses Oberflächen-Bausteins ist es einfach möglich, Bilder zum Beispiel skaliert in festen Größen mit eventuell enthaltenen Regionen darstellen. Sowohl für die Gewinnung von Touchscreen-Gesten als auch für die benutzerdefinierten Bildregionen war es sinnvoll, eine Mausverfolgungseinheit (engl.: *Mouse-Tracker*) an das beschriebene GTK-Widget anzuschließen. Der Baustein ist nun in der Lage, bei der Fertigstellung eines Kantenzuges ein entsprechendes Ereignis an die Applikation zu melden. Dort wird im Anschluss der Kantenzug ausgelesen und analysiert.

5.3.1 Merkmalsberechnung

Bei der Abbildung der Kantenzüge auf eine Menge gültiger Touchscreen-Gesten handelt es sich um ein klassisches Musterklassifikationsproblem. Zur Lösung des Problems müssen die eingehenden Kantenzüge zunächst einer Vorverarbeitung unterzogen werden, um resultierend einheitlich weiterverarbeitet werden zu können [Nie83].

Das Ziel bei der Beschreibung von Mustern durch Merkmalsvektoren ist in diesem Fall, eine gute Separierbarkeit der zu erkennenden Klassen zu erhalten. Dabei ist zu beachten, dass die Geste *X*, die der Auswahl des Bildobjekts dient (siehe Abbildung 5.5(c)), die aus zwei Kantenzügen besteht, die separat erkannt werden müssen. Da die Merkmalsberechnung jedoch auch noch bei Erweiterungen von Gesten benutzbar sein sollte, wurde versucht, eine möglichst allgemein gültige Beschreibung von Kantenzügen, die bei solchen Gesten erzeugt werden, zu entwerfen.

Die Abbildung 5.6 zeigt das initiale Vorgehen bei der hier durchgeführten Merkmalsberechnung. Der Startpunkt des Kantenzuges, $\vec{p}_1$ bildet den Ursprung des Koordinatensystems. Der Rotationswinkel α ist die erste Merkmalsgröße. Durch die Drehung um diesen Winkel wird der Endpunkt $\vec{p}_N$ des Kantenzuges auf die positive X-Achse gelegt. Diese Merkmalskomponente ist dann besonders aussagekräftig, wenn Start- und Endpunkt des Kantenzuges nicht zu dicht zusammen liegen. Alle weiteren Größen, die in das Merkmal eingehen, sollten bezüglich der Rotation invariant sein.

Es ergibt sich:

$$\alpha = \operatorname{atan2}(p_{Ny} - p_{1y}, p_{Nx} - p_{1x})$$

Die Lageverteilung bezüglich der positiven und negativen Halbebene, die durch die X-Achse getrennt sind, wird durch die folgende Merkmalskomponente beschrieben. Diese Komponente differenziert symmetrische, also S-förmige, links- und rechtsseitige Kantenzüge (siehe Abbildung 5.7(a) und 5.5(b)). Die Lageverteilung ergibt sich zu:

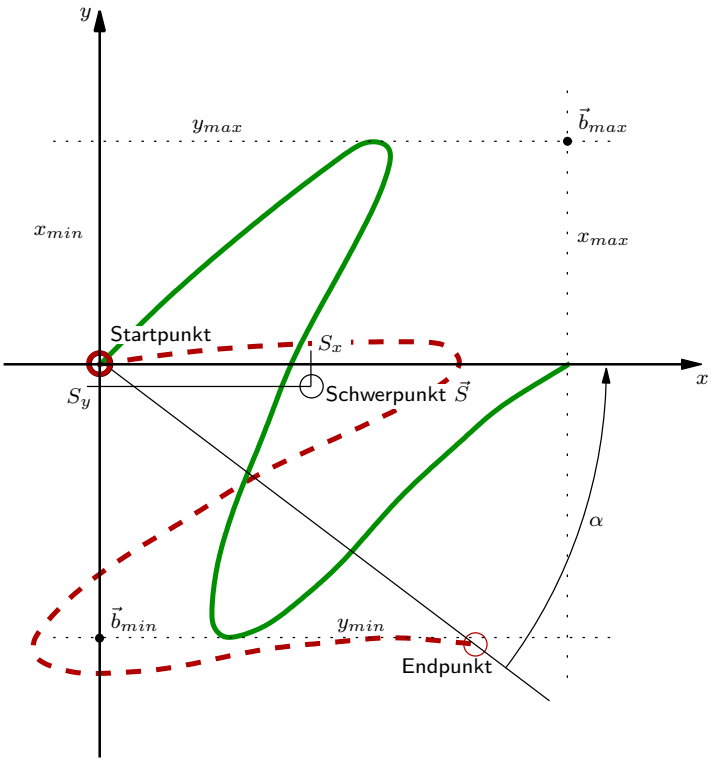


Abb. 5.6: **Merkmale der Kantenzüge:** Zur Bestimmung der Kantenzug-Merkmale wird der originale Kantenzug (rot, gestrichelt) so translatiert und rotiert, dass der Startpunkt im Ursprung und der Endpunkt des Kantenzuges auf der positiven X-Achse zu liegen kommt. Die Box, die den abgebildeten Kantenzug (grün) umfasst und durch $\vec{b}_{max} = (x_{max}, y_{max})^T$ und $\vec{b}_{min} = (x_{min}, y_{min})^T$ beschrieben ist, bildet die Grundlage für die Berechnung aller Merkmale.

$$v_y = \frac{y_{max}}{y_{max} - y_{min}}$$

Benutzer sollten nicht gezwungen werden, die Geste in einer vorgeschriebenen Größe durchzuführen. Daher müssen die weiteren Größen invariant bezüglich der absoluten Gestengröße gehalten werden. Dafür werden zum einen die Diagonale d_b der den Kantenzug umgebenden Box (engl.: *bounding box*), beschrieben durch $\vec{b}_{max}$ und $\vec{b}_{min}$, und die absolute Länge des Kantenzuges l_k als Normierungsgrößen ermittelt:

$$\begin{aligned} d_b &= \left| \vec{b}_{max} - \vec{b}_{min} \right| \\ l_k &= \sum_{n=1}^{N-1} |\vec{p}_{n+1} - \vec{p}_n| \end{aligned}$$

Eine Beziehung zwischen der Länge des Kantenzuges und der umschließenden Box wird ebenfalls als Merkmalskomponente eingebracht. Anschaulich ist dies ein Maß dafür, wie weit der Kantenzug von einer Geraden abweicht.

$$g = \frac{d_b}{l_k}$$

Zwei weitere Komponenten werden von den auf die Boxen-Diagonale normierten Koordinaten des Kantenzug-Schwerpunktes gebildet. Die sich daraus ergebende Größe beschreibt die Verteilung des Kantenzuges. Der Schwerpunkt berechnet sich wie folgt:

$$\vec{S} = \frac{\sum_{n=1}^{N-1} \left(|\vec{p}_{n+1} - \vec{p}_n| \left(\vec{p}_{n+1} + \vec{p}_n \right) \right)}{2 l_k}$$

Der fünf-dimensionale Vektor zur Beschreibung eines Kantenzuges ist damit folgendermaßen definiert:

$$\vec{c} = \left(\alpha, v_y, g, \frac{S_x}{d_b}, \frac{S_y}{d_b} \right)^T \tag{5.1}$$

5.3.2 Der Klassifikator

Die Aufgabe, einen detektierten Kantenzug, der in Form eines Merkmalsvektors vorliegt, einer der hier möglichen Touchscreen-Gesten zuzuordnen, wird allgemein von einem numerischen Klassifikator durchgeführt. Gemeinsam ist solchen Klassifikatoren, dass grundsätzlich eine, die wahrscheinlichste Klasse als Ergebnis geliefert wird. Dabei ist zunächst unwichtig, wie sicher die Richtigkeit dieses Ergebnisses einzustufen ist.

Bei einer Applikation wie der des hier beschriebenen Bilddatenbank-Clients ist es jedoch enorm wichtig, falsche Klassifikationsergebnisse zu vermeiden, da dem Benutzer nicht zugemutet werden soll, eine somit falsch durchgeführte Aktion wieder rückgängig zu machen. Aus diesem Grund muss der Klassifikator mit einem Rückweisungs-Mechanismus versehen werden, so dass unsichere Klassifikationsergebnisse erkannt und die Ausführung einer Aktion vermieden werden kann.

Mithilfe eines Polynomklassifikators kann die Rückweisung effektiv bewerkstelligt werden (siehe Abschnitt 5.3.3). Bei einem Polynomklassifikator, der K Klassen unterscheiden soll, werden entsprechend viele linear unabhängige Polynomfunktionen vom Grad G, die Trennfunktionen $d_k(\vec{c})$, erstellt. Alle diese Funktionen erhalten den Merkmalsvektor $\vec{c}$ als Parameter. Die Entscheidung, welcher Klasse ein Merkmalsvektor zugeordnet wird, fällt durch eine Maximums-Bildung über die Menge der Trennfunktionen. Kompakt lässt sich ein solcher Klassifikator in der folgenden Form darstellen:

$$\vec{d}(\vec{c}) = \underline{A}^T \vec{\varphi}(\vec{c}) \tag{5.2}$$

Diese Darstellung unterstreicht die Zweischichtigkeit des Polynomklassifikators. $\vec{\varphi}(\vec{c})$ wird auch als erweiterter Merkmalsvektor bezeichnet und stellt die polynomiale Multiplikation der Merkmalskomponenten dar. Die Dimension dieses Vektors ist von dem Grad G der verwendeten Polynome und der Dimension M des Merkmalsvektors abhängig, sie ergibt sich zu $\binom{M+G}{G}$.

Die Parametermatrix $\underline{A}$ wird mithilfe einer klassifizierten Stichprobe erzeugt. Hierbei wird der Erwartungswert des mittleren quadratischen Fehlers minimiert, der folgendermaßen definiert ist:

$$\epsilon(\underline{A}) = E\left\{(\vec{\delta}(\vec{c}) - \underline{A}^T\vec{\varphi}(\vec{c}))^2\right\}$$

Dabei stellt $\vec{\delta}(\vec{c})$ die idealen Trennfunktionen dar, bei denen der Wert der k-ten Funktionskomponente eine Eins als Ergebnis liefert, wenn $\vec{c}$ der k-ten Klasse angehört. Alle anderen Funktionen liefern in diesem Fall eine Null.

Der hier angewendete Polynomklassifikator wurde mit Polynomen vom Grad 4 erzeugt. Die klassifizierte Stichprobe enthielt 1015 Vektorelemente entsprechend Gleichung 5.1. Ein klassifizierter Testdatensatz von 93 Elementen führte zu einem durchaus akzeptablen Klassifikationsfehler von 2.2%. Für den Erhalt der klassifizierten Gesten wurde eine eigens zu diesem Zweck entworfene Testapplikation benutzt, die alle sechs Klassenelemente in unterschiedlicher Reihenfolge, jedoch gleich oft dem Benutzer präsentierte. Dieser wurde angehalten, eine entsprechende Geste durchzuführen. Besonders das Kreuz (Abbildung 5.5(c)) stellt eine Ausnahme dar, denn für die Ausführung dieser Geste gab es Varianten sowohl in der Reihenfolge der gezeichneten Kantenzüge als auch bei deren Orientierung.

5.3.3 Realisierung der Rückweisung

Wie im vorangegangenen Abschnitt erwähnt, ist es dem Benutzer der Datenbank nicht zumutbar, eine Aktion, die aufgrund einer falsch klassifizierten Geste ausgeführt wurde, wieder rückgängig zu machen. Daher wurde Wert auf eine robuste Rückweisung von nicht klar erkannten Gesten gelegt.

Es wurden drei Aspekte, die zu einer Rückweisung führen können, umgesetzt:

1. Zusätzlich zu den erlaubten Touchscreen-Gesten wurden zwei weitere trainiert, die einen unterschiedlichen Aufbau vorweisen (vergleiche Abbildung 5.7 und 5.5). Damit wird der durch den allgemein gehaltenen Merkmalsvektor aufgespannte Raum besser durch vorhandene Klassen abgedeckt. Gesten, die diesen beiden Klassen zugeordnet wurden, werden verworfen.

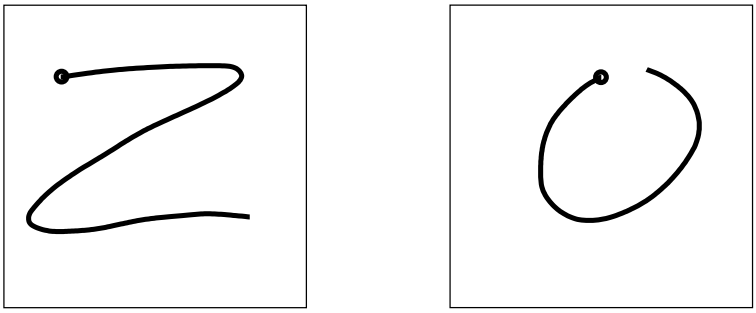
2. Grundsätzlich werden nur Kantenzüge zu der Gestenerkennung zugelassen, die eine Mindestlänge von 75 Pixeln vorweisen. Damit wird zum einen verhindert, dass Zeigegesten, die durch das Berühren des Bildes durchgeführt werden, automatisch als Touchscreen-Geste ausgewertet werden, und zum anderen wird verhindert, dass Abtast-Ungenauigkeiten eingehen, die entstehen, wenn ein Finger für die Positionierung bei einem Touchscreen benutzt wird. Diese Ungenauigkeiten wirken sich bei kurzen Kantenzügen mit schnell folgenden Richtungsänderungen durch Rollen der Fingerkuppe besonders stark aus.

3. Das dritte Kriterium wertet die Sicherheit des Klassifikationsergebnises aus. Dazu werden die Ergebnisse der Trennfunktionen $d_k(\vec{c})$ verwendet (siehe Gleichung 5.2).

$$e_q = \sum_{k=1}^{K} d_k(\vec{c}) - d_{min}(\vec{c}),\ d_{min}(\vec{c}) = \min_{k \in \{1,2,\cdots,K\}} d_k(\vec{c})$$

e_q ist ein Maß dafür, wie sich die Funktionsergebnisse ähneln. Im Idealfall, bei dem lediglich ein Funktionsergebnis gleich Eins ist, ergibt sich für e_q ebenfalls eine Eins. Je größer dieser Wert wird, desto schlechter ist das Klassifikationsergebnis zu bewerten. Empirisch wurde der Schwellwert $e_{qs} = 2.8$ festgelegt.

Abbildung 5.7 zeigt die trainierten Gesten, die zu einer Rückweisung führen. Diese beiden



(a) Symmetrische Geste (b) Lange rechtsseitige Geste

Abb. 5.7: **Touchscreen-Gesten der Rückweisungsklassen**

Gesten unterscheiden sich besonders durch die y-Verteilungs-Symmetrie, v_y, die bei dem Z-Symbol gänzlich anders ist, und bei beiden hier gezeigten Gesten durch erheblich längere Kantenzüge bezüglich der umschließenden Box, g.

5.4 Prozessstruktur

In den vorangegangenen Abschnitten wurden die unterschiedlichen Applikationsteile, die sowohl für die Verarbeitung der unterschiedlichen Eingabemodalitäten als auch für die Präsentation des Suchergebnisses verantwortlich sind, vorgestellt. Aufgrund des asynchronen Charakters der Ereignisse, die durch die unterschiedlichen Modalitäten erzeugt werden, wurde die Client-Applikation in mehrere Threads aufgeteilt. Die dadurch entstehende Struktur ist logisch getrennt und lässt sich daher einfacherer und robuster entwickeln.

Durch die in Abbildung 5.8 dargestellte Trennung wird eine komplexe Ablaufsteuerung notwendig. Abbildung 5.8 zeigt im linken Teil die beiden Threads, die die eingehenden Ereignisse, die durch die Interaktion mit dem Benutzer entstehen, vorverarbeiten und dann an die Ablaufsteuerung unter Benutzung ihrer Kommunikationsschnittstelle weiterleiten. Durch diese Verlagerung der Ausführung wird vermieden, dass einer der erzeugenden Threads zum Beispiel durch die Anforderung eines Server-Dienstes unnötig lange blockiert. Ein solches Verhalten wäre vor allem bei dem Thread der Bedienoberfläche nicht zu akzeptieren, da es in diesem Fall zu Verzögerungen des Bildaufbaus kommen könnte.

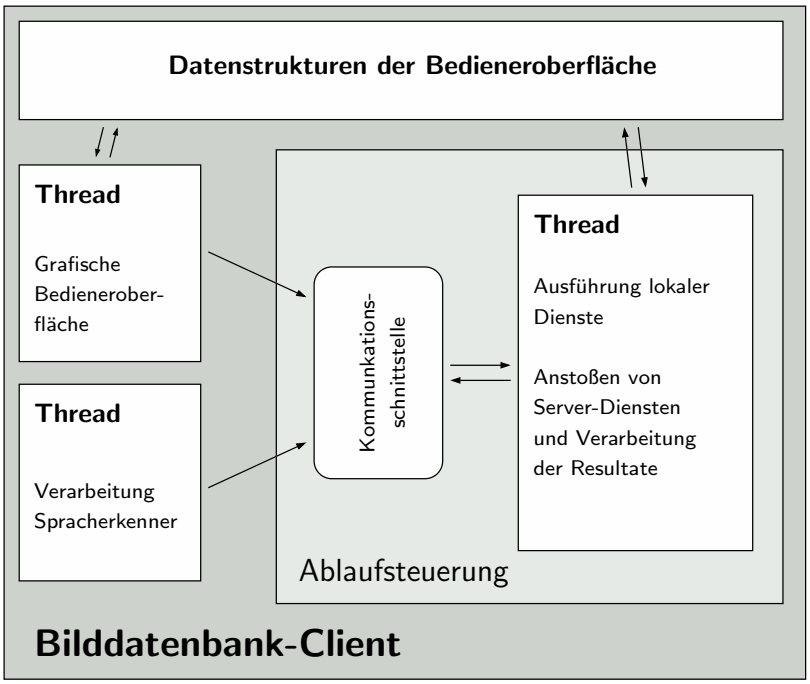


Abb. 5.8: **Thread-Struktur des Clients:** Die Ablaufsteuerung stellt das zentrale Element dieser Drei-Thread-Struktur dar. Sie beinhaltet die Kommunikationsschnittstelle, die eine Multi-Producer/Single-Consumer-Struktur aufweist und lokale Dienste ausführt sowie Server-Dienste anstößt. Ergebnisse können direkt in die Bedienoberfläche eingetragen werden.

Die Ablaufsteuerung wird immer dann aktiv, wenn Anforderungen resultierend aus den Interaktionen vorliegen. Die Ablaufsteuerung kann als *lokaler Server* bezeichnet werden. Die Steuerung unterscheidet zwischen lokalen Diensten, die von der Client-Applikation selbst ausgeführt werden, und solchen, bei denen ein Dienst des Datenbank-Servers in Anspruch genommen werden muss. Beide Arten von Diensten werden von der Kommunikationsschnittstelle entgegen genommen. Bei der Anforderung eines Datenbank-Server-Dienstes wird das Paket an den Server weitergeleitet. Dieser erzeugt als Antwort wiederum ein Paket mit einer lokalen Dienstanforderung, die sofort nach Erhalt von der Ablaufsteuerung ausgeführt wird. Sämtliche hier erwähnten Dienste werden im Kontext der Ablaufsteuerung ausgeführt, die Zugriff auf die Daten der Bedienoberfläche hat, um beispielsweise die Bilddaten in dem Fenster der Bedienoberfläche zu setzen. Aufgrund des Eingriffs dieser Dienste in die Bedienoberfläche ist es so wichtig, dass alle Dienste von einem unabhängigen Thread ausgeführt werden, denn mit dieser strikten Trennung werden Verklemmungen beim Zugriff auf die Daten der Bedienoberfläche ausgeschlossen.

5.5 Ablaufsteuerung

Da es nicht vorhersagbar ist, wann Sendeanforderungen der unterschiedlichen Kommunikationskanäle der zentralen Kommunikationsschnittstelle, in der die anstehenden Anforderungen zwischengespeichert werden, zugeführt werden, muss der Zugriff synchronisiert werden. Hierbei

handelt es sich jedoch nicht um einen einfachen gegenseitigen Ausschluss, sondern aufgrund der besonderen Datenabhängigkeiten ist eine komplexe Synchronisation entworfen worden.

Die Ablaufsteuerung unterscheidet drei unterschiedliche Betriebsmodi:

1. Warten auf Sendeanforderungen
2. Ausführen eines einzelnen Dienstes
3. Sequentielles Ausführen einer dynamischen Gruppe von Diensten

Alle vom Client angebotenen Dienste verändern seinen internen Zustand. Bei den meisten dieser Dienste, die aufgrund einer Benutzerinteraktion ausgeführt werden, bestehen Datenabhängigkeiten, die nur durch Ausführen anderer lokaler Dienste oder Datenbank-Server-Dienste aufgelöst werden können. Auch bei einfachen Diensten wie dem, der den Ausschnitt der angezeigten Bilder in der Ergebnismenge verändert (Herunter- oder nach oben Scrollen), kann die Aufgabe nicht unbedingt von einem einzelnen Dienst bewerkstelligt werden. Sollten die Übersichtsbilder des aktuellen Bildergebnises, die durch die Aktion sichtbar werden, noch nicht im Client vorliegen, so müssen diese zunächst vom Datenbank-Server geladen werden. Dasselbe gilt für die am ähnlichsten bewerteten Regionen der Bilder, die auch in der Übersicht

Funktion	**bes. Parameter**	**Bedeutung**
`vCommInit`	Größe	Definition einer Kommunikations-Instanz
`iCommPutMsg`	Blockieren	Einfügen eines zu sendenden Pakets. Mit dem Parameter Blockieren (Shared) wird gekennzeichnet, ob es sich um eine Mehrfachanfrage handelt oder nicht.
`iCommResetPrivate`	Finales Paket	Beenden des Aufbauens einer Mehrfachanfrage. Ein optionales finales Paket, das als letztes Paket auch der evtl. verlängerten Mehrfachanfrage gesendet wird, kann übergeben werden.
`pvCommGetMsgBlocked`	Status	Blockierendes Empfangen eines Pakets. In Status wird der Status (normal, Start MfA, Ende MfA) des Pakets zurückgeliefert.

Tab. 5.1: **API der Kommunikationsschnittstelle:** Auf Erzeugerseite werden die Funktionen `iCommPutMsg` und `iCommResetPrivate` benutzt, um Dienste anzufordern und anzuzeigen, dass die Sequenz von Diensten, die eine private Benutzung erfordert, beendet ist. Auf der Konsumentenseite wird mit `pvCommGetMsgBlocked` der nächste zu bearbeitende Dienst abgeholt. `vCommInit` dient der einmaligen Instanziierung der Kommunikationsschnittstelle.

angezeigt werden müssen.

Aufgrund der Datenabhängigkeiten musste der Betriebsmodus 3 implementiert werden. Ist dieser Betriebsmodus aktiv, werden die in der Kommunikationsschnittstelle eingetragenen Dienste sequentiell abgearbeitet und aufgrund des dann herrschenden nicht konsistenten internen

Zustands alle anderen Anforderungen für die Dienstannahme verworfen. Oftmals fehlen zum Zeitpunkt der Interaktion noch die nötigen Daten, um bereits dann alle notwendigen Dienste anzufordern. Das ist beispielsweise der Fall beim Start einer Suchiteration, wenn die zu präsentierenden Bilder erst durch das Suchergebnis, als Ergebnis des ersten Dienstes, bekannt werden. Aus diesem Grund kann die Sequenz von Diensten in der Kommunikationsschnittstelle bei der Bearbeitung des letzten Dienstes der Sequenz verlängert werden. Abbildung 5.9 veranschaulicht in einem Zustandsdiagramm das Verhalten der Kommunikationsschnittstelle, die durch die in Tabelle 5.1 dargestellte Programmierschnittstelle (engl.: *Application Programming Interface, API*) bedient wird.

Abbildung 5.9 zeigt die möglichen Zustände der Kommunikations-Schnittstelle und deren be-

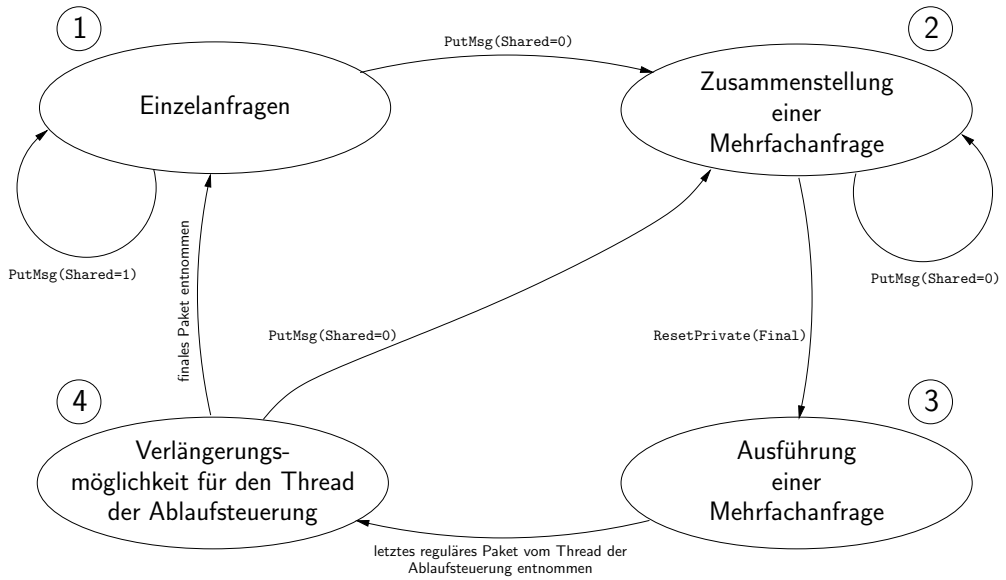


Abb. 5.9: **Zustände der Kommunikationsschnittstelle:** Ausgehend vom Zustand 1, in dem Einzelanfragen durchgeführt werden, wird bei einer Mehrfachanfrage eine Sequenz von Zuständen durchlaufen. Die Zustandsübergänge werden durch die Funktionen `ResetPrivate()` und `PutMsg()` bzw. durch die Entnahme von Paketen für die Abarbeitung ausgelöst.

dingte Übergänge. Ausgehend von Zustand 1 werden sowohl Einzel- als auch Mehrfachanfragen entgegengenommen. Einzelanfragen (`Shared=1`) werden beliebig angenommen und ausgeführt. Handelt es sich bei der Anfrage um die erste einer Sequenz (`Shared=0`), wechselt die Kommunikationsschnittstelle in den Zustand 2.

In diesem Zustand wird die Sequenz von Diensten durch das Akzeptieren beliebig vieler weiterer Anfragen aufgebaut. Ist die Sequenz zunächst vollständig, dann wird dies durch den Funktionsaufruf von `iCommResetPrivate` angezeigt. Dieser Funktion kann optional eine weitere Anforderung übergeben werden. Dieser besondere Dienst wird definitiv als letzter der Sequenz, unabhängig davon wie oft oder lang diese verlängert wurde, abgearbeitet. Mit dieser Funktionalität ist es möglich, Besonderheiten, zum Beispiel die Benutzerschnittstelle betreffend, am Ende der Sequenz auszulösen. Eine solche Anforderung wird jedoch nur dann akzeptiert, wenn

nicht bereits in einem vorherigen Durchlauf dieses Zustands ein solcher finaler Dienst gesetzt wurde.

In dem nun erreichten Zustand 3 werden alle angenommenen Dienste in der Reihenfolge, in der deren Anfragen eingingen, abgearbeitet. Die Annahme von Anfragen ist in diesem Zustand nicht erlaubt. Eventuelle Anfragen würden verworfen werden.

Mit der Bearbeitung des letzten regulären Dienstes, einem, der also mit `iCommPutMsg()` angefordert wurde, wird der Zustand 4 angenommen. Hier besteht die Möglichkeit, die Sequenz zu verlängern. Wird in der Bearbeitung des Dienstes erneut eine Anforderung mit `iCommPutMsg()` durchgeführt, wechselt die Kommunikationsschnittstelle wiederum in den Zustand 2, um noch weitere Anforderungen entgegenzunehmen. Wird die Bearbeitung des Dienstes jedoch, ohne die Sequenz zu verlängern, verlassen, wird der eventuell angeforderte finale Dienst ausgeführt und damit der Zustand 1 angenommen. Bei der Ausführung des finalen Dienstes ist die Kommunikationsschnittstelle leer und für alle Typen von Anfragen freigeschaltet.

Ablauf einer Suchiteration

Die Benutzung der Kommunikationsschnittstelle soll hier exemplarisch anhand einer Suchiteration gezeigt werden. In diesem Beispiel, das die komplexeste Abfolge von Informationsaustausch darstellt, werden alle Mechanismen, die die Kommunikationsschnittstelle bereitstellt, ausgenutzt.

Abbildung 5.10 stellt den Paketlaufplan grafisch dar. Die Suchiteration wird durch den Aufruf der Funktion `vStartSearch` gestartet. Im Teil I werden eventuell neu generierte Regionen abgespeichert und neu zugewiesene Identifikationsnummern empfangen. Dieser Schritt ist zu Anfang notwendig, da die Identifikationsnummern für das Referenzieren der neuen Regionen als Beispielregion und als bewertete Region notwendig sind. Im sich anschließenden Teil II werden die aktuellen Bewertungen für alle Regionen gesetzt. Handelt es sich um eine neue Suche, müssen zuvor alle bisherigen Bewertungen zurückgenommen werden (*CRPUInitRelevances*). Im Teil III wird die eigentliche Suche mit der aktuell selektierten Region gestartet. Die Bilddatenbank liefert eine sortierte Liste von Bildern entsprechend dem aktuellen Suchzustand an den Client zurück. Diese Liste wird im Teil IV analysiert und noch nicht vorhandene Übersichtsbilder in der Funktion `vLoadThumbs` beim Durchlaufen einer Schleife nachgeladen. Da das eigentliche Suchergebnis nicht aus den bisher verarbeiteten Bildern, sondern aus den Regionen der Bilder besteht, werden nun die Identifikationsnummern und die Kantenzüge der jeweils ähnlichsten Regionen der Bilder angefordert. Das wird in einer Schleife im Teil V durchgeführt. Schließlich wird im letzten Teil die grafische Oberfläche aktualisiert und ein Analyseschritt in der Auswertung von *SearchFinished* durchgeführt.

Der Paketlaufplan der Suchiteration verdeutlicht die Datenabhängigkeit einer Kommunikationssequenz. Wir haben es hier mit einer zweimaligen Verlängerung der Kommunikations-Sequenz zu tun. Im ersten Fall müssen Identifikationsnummern der neu kreierten Regionen ermittelt werden. Durch das finale Paket *GUIInsRegsFinished*, das sich der Client selber sendet, wird eine Wiederaufnahme der Kommunikationssequenz mit den zuvor nicht vorhandenen Informationen möglich. Die Teile II und III können nun ohne Verzögerung nacheinander durchgeführt werden. In dieser Phase wird das finale Paket *GUISSFinished*, das als letztes einer

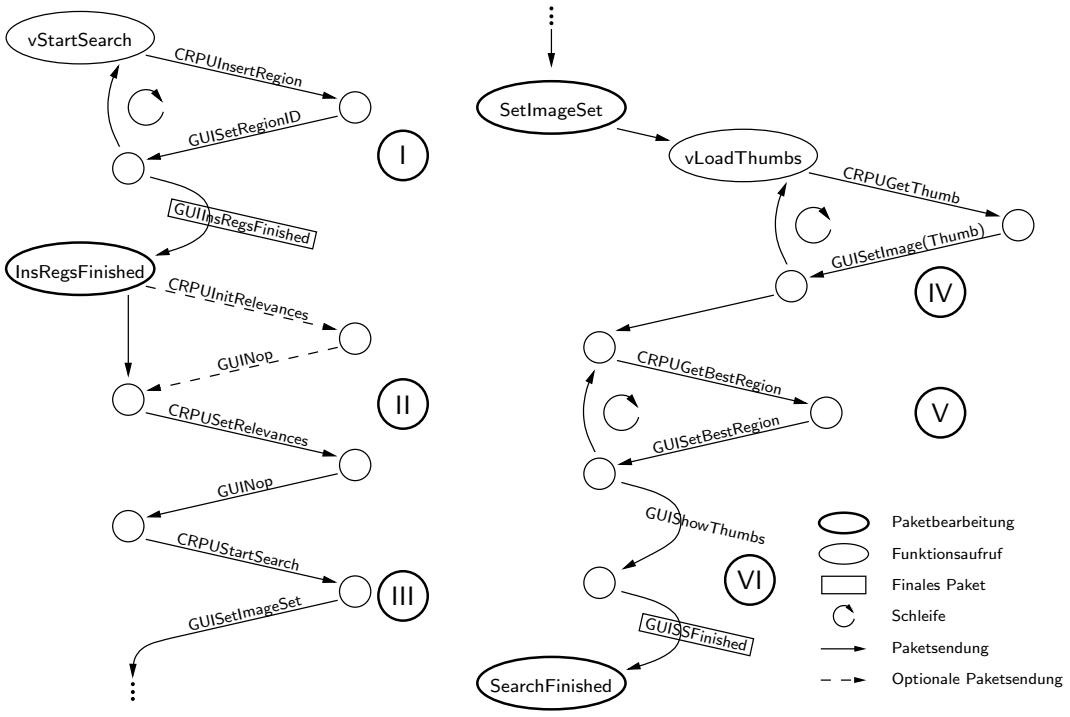


Abb. 5.10: **Paketlaufplan einer Suchiteration:** Die zeitliche Abfolge der Pakete, die bei der Durchführung einer Suchiteration von Client (jeweils links) und Server (jeweils rechts) ausgetauscht werden, ist in dieser Abbildung zu sehen. Die Kommunikation ist in drei Phasen aufgeteilt, die durch die Dienste *vStartSearch*, *InsRegFinished* und *SetImageSet* begonnen werden. Nur bei der Ausführung dieser Dienste werden neue Anforderungen an die Kommunikationsschnittstelle gestellt.

Suchiteration ausgewertet wird, erzeugt. Die zweite Verlängerung wird in der Verarbeitung des Pakets *SetImageSet* durchgeführt. Die nun vorhandenen Informationen der Ähnlichkeitssuche, die von dem Bilddatenbank-Server generiert wurden, können zum Anfordern von fehlenden Daten und für die Aktualisierung der Oberfläche des Clients in den verbleibenden Teilen IV, V und VI benutzt werden.

Fusion von Sprach- und Gesten-Ereignissen

Einige Aktionen des Clients können durch die Kombination von Interaktionen der Modalitäten Gestik und Sprache ausgelöst werden. Es handelt sich hierbei um bildbezogene Aktionen, bei denen die Art der Aktion durch die sprachliche Äußerung und das Bildobjekt durch Antippen des gewünschten Bildes festgelegt werden. Die kausale Verbindung der beiden Interaktionen wird durch Verwendung eines der Demonstrativpronomen „dieses“ oder „das“ hergestellt.

Äußerungen wie zum Beispiel: „Das Bild gefällt mir gut“, „Zeige dieses Bild“ oder „Nimm dieses Bild hier als Beispielbild“ sind daher möglich, um eine Bewertung abzugeben, ein Bild zur Großansicht zu bringen oder ein Bild als Beispielbild zu markieren.

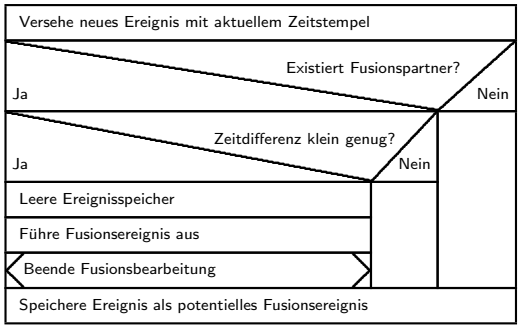


Abb. 5.11: **Fusion der Ereignisse**

Technisch wird eine solche Kombinationsgeste durch zwei Ereignisse, die aus Interaktionen der beteiligten Modalitäten resultieren, ausgelöst. Die Fusion der beiden Ereignisse erfolgt auf Basis der Zeit. Die Ereignisse, die durch die Kommunikationsschnittstelle der einheitlichen Weiterverarbeitung zugeführt wurden, werden, wie in Abbildung 5.11 zu sehen, mit einem Zeitstempel versehen und im Kontext der Bedienoberfläche gespeichert. Sobald ein korrespondierendes Ereignis eintrifft, werden die Zeitstempel beider Ereignisse verglichen und bei genügend kleiner Zeitdifferenz das resultierende Fusionsereignis ausgelöst.

5.6 Client als Testwerkzeug

Bei der Entwicklung der Suchkomponente des Datenbanksystems, die sich im Datenbank-Server befindet, ergibt sich immer wieder die Schwierigkeit, unterschiedliche Algorithmen oder Parametrierungen so gegenüber zu stellen, dass Rückschlüsse über die Suchqualität möglich werden. Die Komponente, die diese Tests so schwer durchführbar macht, ist die, dass es sich hierbei um ein iteratives System handelt, bei dem die Benutzerinteraktion eine so entscheidende Rolle spielt.

Für die Durchführung eines Test der Suchkomponente ergeben sich insbesondere die zwei folgenden Anforderungen:

Wiederholung einer bereits durchgeführten Suche: Im Falle eines Fehlverhaltens der Suchkomponente, das nur unter bestimmten Umständen auftritt, ist es wünschenswert, eine bereits durchgeführten Suche beliebig oft und fehlerlos wiederholen zu können. Ebenso ist es denkbar, eine Suche in verschieden Parametrierungen direkt gegenüber zu stellen. Auch in diesem Fällen ist die Exaktheit der Wiederholung zwingend erforderlich.

Abgesehen von der Tatsache, dass es sich bei solchen Test um eine fehleranfällige und sehr zeitraubende Tätigkeit handelt, ist eine identische Suche auch aus technischen Gründen nicht generell interaktiv wiederholbar. Ein solcher Fall tritt ein, wenn ein Bildobjekt, das in der zu wiederholenden Suche bewertet wurde, in der wiederholten Suche aufgrund der unterschiedlichen Parametrierung jedoch nicht angezeigt wird. Da interaktiv nur die Bildobjekte bewertet werden können, die angezeigt werden, kann in dem Fall die Suche nicht unter normaler Benutzung des Bilddatenbank-Clients wiederholt werden.

Durchführung automatischer Kategorien-Bewertungen: Es bedarf technischen Unterstützung, wenn umfangreichen Tests durchgeführt werden sollen, die eine statistisch fundierte Auswertung der Ergebnisse zulassen sollen. Hier ist die Fehleranfälligkeit und der enorme zeitliche Aufwand als Grund für die Notwendigkeit einer Automatisierung von der Abgabe von Bewertungen und des Startens von Suchiterationen zu nennen.

Werkzeug für die automatische Bedienung des Clients

Für die Erfüllung der genannten Forderungen war sowohl das Aufzeichnen aller Interaktionen mit entsprechenden Parametern als auch die Auswertung dieser Aufzeichnungen mit den daraus resultierenden Aktionen auf Seite des Datenbank-Clients nötig.

Abb. 5.12: **Automatische Bedienung**

Die generell vom Client durchgeführten Aufzeichnungen wurden so ausgelegt, dass sie sowohl von dem in Abbildung 5.12 gezeigten Werkzeug ausgeführt werden können als auch alle notwendigen Daten enthalten, die die Grundlage für die im Kapitel 7 ausgeführte Evaluierung bilden.

Das gezeigte Werkzeug erlaubt es, die Aufzeichnung ohne Pause und Ausgaben auf der Bedienoberfläche auszuführen. Nach Ausführung der letzten durchzuführenden Aktion wird die Bedienoberfläche aktualisiert. Diese Vorgehensweise hat den Vorteil, das langwierige, unbeobachtete Tests nicht unnötig durch den zeitraubenden Aufbau der Bedienoberfläche verlängert werden. Die Ausführung kann jedoch jederzeit auf Granularität von Suchiterationen angehalten und auch schrittweise fortgeführt werden. Bei der schrittweisen Ausführung wird die Bedienoberfläche nach jeder Aktion aktualisiert. Dabei besteht zwischen jedem Schritt die Möglichkeit, interaktiv tätig zu werden, also von der aufgezeichneten Suche abzuweichen.

Die aufgeführte Funktionalität ließ sich nur durch einen erheblichen Eingriff in den in Abbildung 5.10 gezeigten Suchablauf realisieren, auf dessen geänderte Darstellung hier jedoch verzichtet werden soll.

Automatische Bewertung

Der Punkt zwei der oben angegebenen Anforderungen, eine automatische Bewertung auszuführen, wurde ebenfalls realisiert. Dazu wurde das Bildobjekt in der Datenbank um das Attribut „Kategorie“ erweitert. Das Attribut wurde automatisch aus der Organisation der ArtExplosion-Bildsammlung[1], die die Grundlage für alle automatischen Auswertungen bildete (vergleiche [Käs05]), gewonnen. Die Kategorie, die dem Beispielobjekt zugeordnet wurde, bildet auch das Auswahlkriterium für die automatische Bewertung. Alle Bildobjekte, die dem Client als Ergebnis der Suchiteration übermittelt werden und die Zugehörigkeit zu dieser Kategorie aufweisen, werden als positiv relevant bewertet.

Die automatische Bewertung ist jedoch nur dann wirklich sinnvoll einzusetzen, wenn das Testsystem in der Lage ist, in einer Stapelverarbeitung engl.: *Batch processing* mehrere Suchen mit unterschiedlichen Beispielobjekten und einer projektierbaren Anzahl von Suchiterationen durchzuführen. Als Quelle für die Stapelverarbeitung dient der Rahmen der Aufzeichnungsdatei. Ein entsprechender Eintrag versehen mit der Identifikationsnummer des Bildobjekts, der

[1] Die Bildsammlung „ArtExplosion 600000 Images“ ist in unterschiedliche semantische Kategorien wie beispielsweise Tiere, Ballett, Golf, Stockholm usw. eingeteilt. Da die Zuordnung im Dateisystem realisiert ist, kann jedes Bild nur einer Kategorie angehören. Die Sammlung stammt von der der Nova Development Corporation (http://www.novadevelopment.com).

Anzahl der durchzuführenden Suchiterationen und des Ergebnisranges, bis zu dem automatische Bewertungen durchzuführen sind, ist in der Datei einzufügen.

Kapitel 6

Entwicklungswerkzeuge

In diesem Kapitel werden zwei Werkzeuge vorgestellt, die im Rahmen von INDI entstanden sind. Diese Werkzeuge ermöglichen durch automatisch erzeugte Code-Module eine rasche und sichere Änderung von Kommunikations-Schnittstellen und der Verarbeitung der Ergebnisse des eingesetzten Spracherkenners.

6.1 Datenrepräsentationen und deren Generierung

Verteilte Anwendungen tauschen naturgemäß Daten zwischen den einzelnen Teilen der Applikation aus, die auf unterschiedlichen Rechenknoten ausgeführt werden. Da die interne Datenrepräsentation auf verschiedenen Plattformen architekturabhängig ist und damit unterschiedlich sein kann, muss dafür Sorge getragen werden, dass die Daten, soweit es notwendig ist, konvertiert werden. Hierbei handelt es sich zum einen um die Reihenfolge im Speicher und damit die Wertigkeit der Bytes (engl.: *Byte order*), die für die Repräsentation bestimmter Datentypen verwendet werden, und zum anderen um die Menge der Bytes für die Repräsentation eines bestimmten Datentyps. Für eine einheitliche Verarbeitung der zu übertragenden Daten ist es daher notwendig, eine definierte und einheitliche Datenrepräsentation beim Datenaustausch zweier Kommunikationspartner zu verwenden.

Eine einfache jedoch, sehr ineffiziente Methode, die beispielsweise von der MySQL-Datenbank verwendet wird, ist die Übertragung der Daten in Zeichenketten. Hier wird vor allem bei der Dekodierung viel Zeit aufgewendet. Effizienter ist die Übertragung der Daten in binärer Form, was die Konvertierung in die so genannte Network Byte Order, die allgemein anerkannte einheitliche Reihenfolge der Byte-Wertigkeiten, und das Anpassen der Anzahl der für die Datentypen zu verwendenden Bytes erfordert.

6.1.1 Network Data Repräsentation (NDR)

Handelt es sich bei der Entwicklung um die eines komplexen verteilten Systems, dann ist die Überwachung des Datenaustauschs der Komponenten eine sehr wichtige Möglichkeit der Kontrolle bzw. eine Informationsquelle bei einer Fehlersuche.

Damit die übertragenen Daten von einer dritten, unabhängigen Applikation interpretiert und angezeigt werden können, müssen neben den binär kodierten Nutzdaten auch die diesen Daten zugrunde liegenden Datentypen und -strukturinformationen dem Datenstrom beigefügt werden.

DACS (Distributed Application's Communication System), ein an der Universität Bielefeld entwickeltes Kommunikationssystem für verteilte Anwendungen [Jun98], beinhaltet aus der oben angeführten Motivation heraus eine solche Datenrepräsentation, die NDR genannt wird. Diese Datenrepräsentation ist typisiert und strukturiert.

Für die Übertragung von skalaren Typen stellt NDR entsprechende Funktionen zur Verfügung. Nicht allein skalare Typen können transferiert werden, auch Vektoren skalarer Typen können durch einen einfachen Funktionsaufruf übertragen werden. Auf der Empfängerseite wird für den empfangenen Vektor der benötigte Speicher alloziert und mit den empfangenen Daten gefüllt. Da im Allgemeinen eine Übertragung nicht allein aus einem Skalar oder einem oben beschriebenen Vektor besteht, ist die Hauptanwendung bei der Benutzung von NDR die Übertragung von Strukturen. Die Programmierschnittstelle legt dabei fest, dass für die Übertragung von Strukturen eigene Funktionen erstellt werden müssen, die bezüglich des Aufbaus bestimmten Konventionen entsprechen müssen. Der Inhalt einer Struktur wird hierbei durch Aufruf der Funktionen für skalare Typen bzw. Vektoren oder andere Übertragungsfunktionen für Strukturen in der entsprechenden Reihenfolge festgelegt. Auch hier sorgt die Empfängerseite für das Allozieren des benötigten Speichers.

Die NDR-Funktionalität wurde aus dem DACS-Kern für das INDI-Projekt extrahiert und in einer Programmbibliothek zusammengefasst.

Die Funktionalität der bestehenden NDR-Implementierung reichte aber für die Verwendung in INDI nicht aus, so dass einige Erweiterungen vorgenommen werden mussten. Tabelle 6.1 zeigt den erweiterten Funktionsumfang, dessen Realisierung in den folgenden Unterabschnitten näher erläutert wird.

Ergänzung der Ganzzahldatentypen

Die mangelnde Spezifikation von Datentypen in C bezüglich ihrer Größe hielt die Entwickler von NDR ab, Konvertierungen für alle in C möglichen Datentypen bereitzustellen.

Gerade jedoch 64-Bit breite Ganzzahltypen sind ein zentraler Bestandteil von INDI, da die Datenbankschlüssel der verwendeten MySQL-Datenbank genau einen solchen Datentyp fordern.

Die Hinzunahme des 64-Bit-Datentyps konnte nur unter bestimmten Konventionen zugelassen werden. Da keine Spezifikation der Programmiersprache C die Datenbreite der Typen `long` bzw. `long long` genau festlegt, muss der Benutzer in geeigneter Form exakt festlegen, wie viele Bits (32 oder 64) in der entsprechenden Variablen dieser Typen gültig sein sollen.

Unter Zuhilfenahme dieser Information und dem Wissen über die Breite des Datentyps `long` bzw. `long long` des Compilers und der Byteorder der Architektur lassen sich die entsprechenden Abbildungen zur architekturunabhängigen Datenrepräsentation durchführen. Abbildung 6.1 veranschaulicht diesen Vorgang anhand eines Strukturbeispiels.

Die hier beschriebene Technik ermöglicht die Verwendung von Datentypen, deren interne Repräsentation architekturabhängig unterschiedlich sein kann. Neben der Angabe des verwen-

Kategorie	NDR-Funktion	C-Datentyp	Neu	Breite/Funktion
Skalar	ndr_int	int		32 Bit
	ndr_short	short		16 Bit
	ndr_char	char		8 Bit
	ndr_byte	unsigned char		8 Bit
	ndr_long32	long/long long	*	32 Bit
	ndr_long64	long/long long	*	64 Bit
	ndr_float	float		32 Bit
Vektor	ndr_string	char *		nullterminierte Zeichenkette
	ndr_vector			Vektor von skalaren Grundtypen
	ndr_long_vector		*	Vektor der long-Datentypen
	ndr_hyper			mehrdimensionaler Vektor von skalaren Grundtypen
	ndr_xvector		*	Vektor von komplexen Strukturen
	ndr_list			Liste von komplexen Strukturen
Struktur	ndr_pair			Paar-Operator; Strukturerweiterung/-verzweigung
	ndr_nil			Strukturende

Tab. 6.1: **Erweiterter Satz von NDR-Funktionen zur Datenübertragung**

deten NDR-Datentyps muss der Konvertierungsfunktion bei dem long-Datentyp eine Angabe über die Breite der verwendeten Variablen und die Verschiebung innerhalb dieser Variablen gemacht werden. Mit dieser zusätzlichen Information variiert jedoch der Code der Übertragungsfunktion auf den unterschiedlichen Architekturen!

Speicherverwaltung/-anforderung

Dem DACS-Paket wurde eine eigene Speicherverwaltung gegeben. Mit dieser Speicherverwaltung ist es für jede Speicheradresse möglich, festzustellen, ob sie einen angeforderten Speicherbereich repräsentiert und wie dessen Größe ist. Diese Technik wurde aufgrund verschachtelter Strukturen, deren Speicherbereich nur auf oberster Ebene angefordert bzw. aufgegeben werden durfte, nötig. Diese Speichertechnik birgt den entscheidenden Nachteil, dass ein von DACS angeforderter Speicherblock auch mit den von DACS bereitgestellten Funktionen verwaltet werden muss. Da INDI jedoch unter Verwendung von verschiedenen Programmpaketen implementiert werden musste, hätte die Verwendung der DACS-Speicherverwaltung dazu geführt, dass zwei unterschiedliche Arten der Verwaltung von Speicher parallel im System existiert hätten. Diese Parallelverwaltung wäre sehr aufwendig, ineffizient und vor allem fehleranfällig gewesen.

Um die Benutzung des speziellen DACS-Speichers zu vermeiden, werden nun die NDR-Funktionen, die für die Übertragung von Strukturen erstellt werden müssen, zweistufig aufgebaut. Die äußere Stufe beinhaltet ausschließlich die Speicherverwaltung, die innere Stufe die eigentliche Datenkonvertierung, so dass bei einem verschachtelten Strukturaufbau ledig-

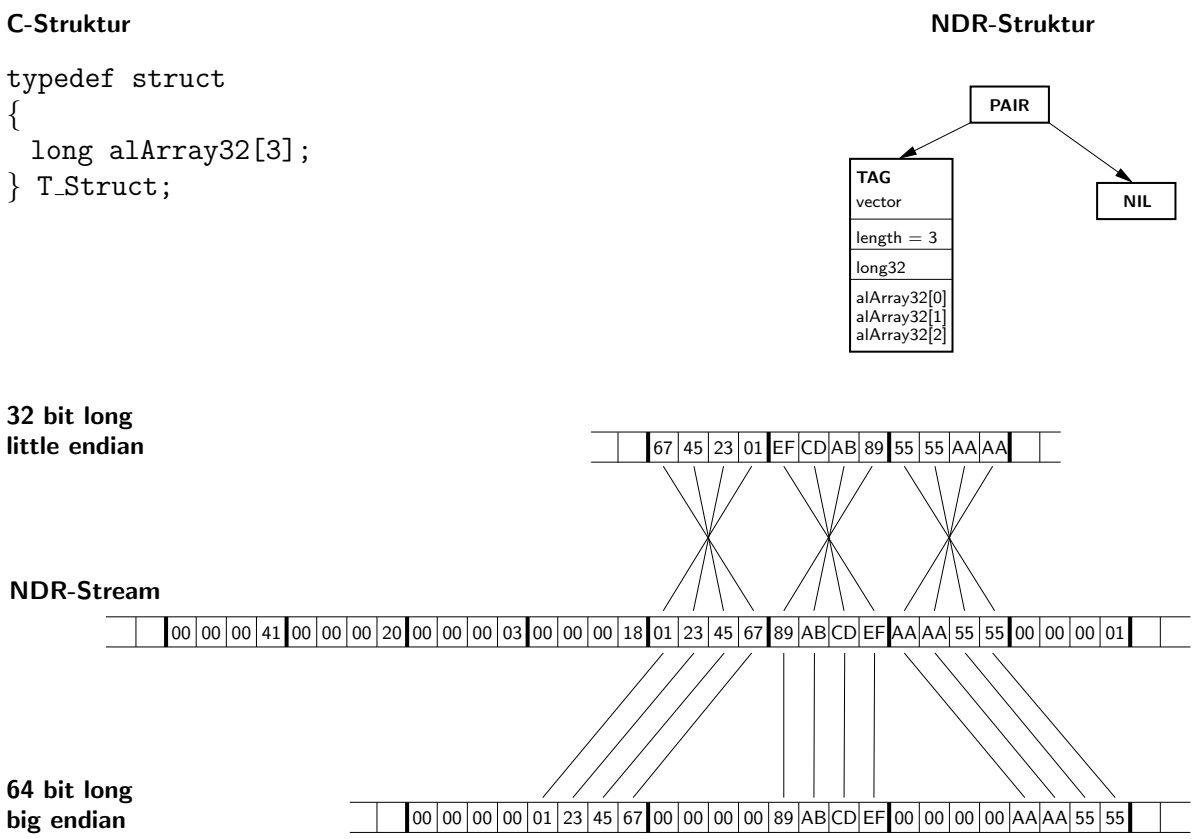


Abb. 6.1: **Repräsentation einer Struktur unter Verwendung von NDR:** Veranschaulichung der unterschiedlichen Repräsentation der C-Struktur `T_Struct`. Dargestellt sind die Speicherauszüge auf unterschiedlichen Systemarchitekturen und die Repräsentation der NDR-Struktur in einem entsprechendem NDR-Stream.

lich einmal die Speicherverwaltung, die äußere Stufe, ausgeführt wird. Für jede eingebettete Struktur darf nur die Datenverarbeitungsstufe aufgerufen werden.

Vektoren von komplexen Strukturen

Arrays von komplexen Strukturen ließen sich bisher nur durch die NDR-Funktion `ndr_list` in NDR-Strukturen umwandeln. Die Verwendung von Listen hat den Nachteil, dass die Speicheranforderung zweistufig erfolgt, denn die Liste wird im Allgemeinen als Vektor von Zeigern auf die eigentlichen Elemente implementiert. Unter bestimmten Umständen ist es aber für die nachfolgende Verarbeitung der Daten wünschenswert, eine einstufige Speicheranforderung, wie sie bei Vektoren durchgeführt wird, zu verwenden.

Die einstufige Speicheranforderung bei der Übertragung von Arrays als Vektoren von Strukturen setzt voraus, dass die Größe eines einzelnen Elements des Vektors ermittelbar ist. Mit der NDR-Programmierschnittstelle ist dies aber nicht ohne weiteres möglich, denn die NDR-Funktion, die einen Vektor verarbeitet, erhält lediglich den Zeiger auf die innere NDR-Funktion, die für die Verwaltung der Elementstruktur zuständig ist. Daher wurde die folgende Konvention festgelegt: Wird die innere NDR-Funktion einer Struktur mit einem ungültigen Stream-Parameter (`NDRS`

*) aufgerufen, dann muss die negative Größe eines Strukturelements zurückgegeben werden. Hier wird ein negativer Wert verwendet, um den Rückgabewert von den ansonsten positiven Statuswerten klar zu trennen.

6.1.2 Der NDR-Pre-Compiler

Die im vorherigen Abschnitt erläuterten Erweiterungen und Konventionen, vor allem aber die Tatsache, dass aufgrund der neuen Datentypen der Programmcode für die NDR-Funktionen architekturabhängig geworden ist, führten zu der Anforderung, die NDR-Funktionen automatisch erstellen zu lassen. Der hierzu erstellte NDR-Pre-Compiler verwendet C-Header-Dateien als Eingabe. Die in diesen Dateien beinhalteten Strukturen werden erkannt und verarbeitet. Die erstellten NDR-Funktionen werden in einem Modul bestehend aus einer C- und einer H-Datei ausgegeben. Das Modul ist direkt kompilierbar und kann dem Projekt hinzugefügt werden.

Das Pre-Compiler-Projekt ist so gehalten, dass die Code-Erzeugung auch leicht auf andere Programmiersprachen wie beispielsweise JAVA erweitert werden kann.

Abbildung 6.2 verdeutlicht an einem Beispiel alle hier im Folgenden vorgestellten Konventionen, die bei der Benutzung des NDR-Pre-Compilers *h2ndr* zur automatischen Generierung von NDR-Funktionen einzuhalten sind.

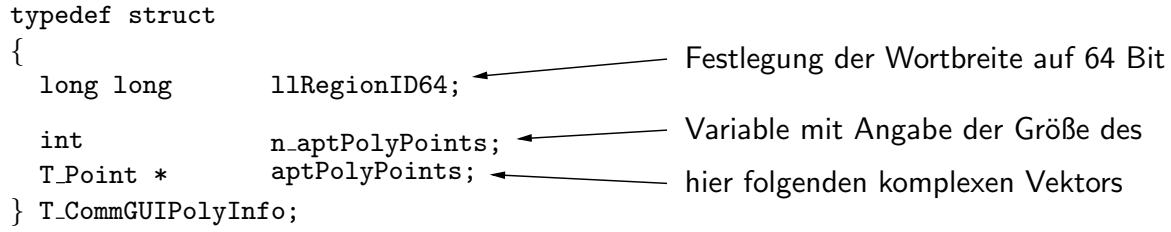


Abb. 6.2: **Konventionen des NDR-Pre-Compilers am Beispiel**

Konventionen bei dynamischen Arrays

Eine Variablendeklaration von Arrays in der Programmiersprache C beschränkt sich meist nur auf die verwendeten Datentypen und deren Dimensionen, denn lediglich bei statischen Arrays werden Angaben über die Größe gemacht. Dynamische Vektoren, Listen und Matrizen benötigen zur Vervollständigung der Beschreibung eine bzw. mehrere Größenangaben, die den Gültigkeitsbereich der Indizierung des Datentyps festlegen.

Da nun auch dynamisch allozierte Vektoren und Listen korrekt von NDR verarbeitet werden sollen, müssen die fehlenden Informationen durch geeignete Konventionen bei der Kodierung der Strukturen eingebracht werden. Daher müssen die Größenangaben grundsätzlich in Variablen vom Typ `int` vor der eigentlichen Liste, dem Vektor bzw. der Matrix (`ndr_hyper`) in der der Dimension entsprechenden Anzahl gemacht werden. Nur so kann die Zuordnung vom Pre-Compiler richtig durchgeführt werden. Zur einfacheren Lesbarkeit und zur Verdeutlichung

der Zusammengehörigkeit bietet es sich an, die Größenvariablen entsprechend der Vektor-/Listenvariable mit entsprechendem Prä- und Suffix zu benennen. Auch auf diese Konvention zur Vermeidung von Fehlern weist der Compiler hin, wenn dagegen verstoßen wird.

Konventionen bei der Verwendung von long-Datentypen

Die Verwendung des `long`-Datentyps erfordert aufgrund der unterschiedlichen Repräsentation der verschiedenen Architekturen die Angabe über die tatsächlich von der Applikation verwendete Wortbreite (32 oder 64 Bit). Diese Angabe muss in dem Variablennamen kodiert werden. Die Variablen vom `long`-Datentyp müssen eine der Zeichenketten „`32`“ oder „`64`“ im Namen beinhalten. Ist dies nicht der Fall, wird ein Übersetzungsvorgang mit einer Fehlermeldung abgebrochen.

6.2 Parsergenerierung für die Sprachverarbeitung

Die Integration des Spracherkennungssystems ISR [Fin99] war ein Meilenstein bei der Entwicklung des Bilddatenbanksystems INDI. Wie bereits in Abschnitt 5.2 erläutert, wurde hier eine sehr restriktive Auswertung der Ergebnisse des Spracherkenners benötigt. Der Erkenner wird, wie dort beschrieben, so eingesetzt, dass die erkannten Äußerungen als ASCII-Datenstroms emittiert werden. Dieser Datenstrom muss nun nachfolgend semantisch analysiert und weitergehend ausgewertet werden.

Die Auswertung, die eine strikte Einhaltung der zugrunde liegenden ISR-Grammatik verlangt, lässt sich naturgemäß automatisch aus dieser Grammatik erstellen. Der Aufwand für die Entwicklung einer solchen Automatisierung ist recht hoch und muss sich daher rechtfertigen. Die Motivation für die Entwicklung einer Automation ist zweigeteilt. Zum einen ist zu erwarten, dass der verwendete Spracherkenner auch noch in anderen Anwendungen mit ähnlichen Anforderungen zum Einsatz kommt, so dass sich auch dort die Entwicklungszeit verkürzen lässt. Der viel wichtigere Grund ist die einfache und fehlerunanfällige Wartung und Änderung des Systems.

Wird auf eine Automatisierung verzichtet, dann bietet es sich an, einen Standardparser wie *yacc* oder *bison* einzusetzen. In diesem Fall muss der Entwickler die ISR-Grammatik von Hand in eine *yacc*-Grammatik übersetzen. Die resultierende Datei wird mit Programmsegmenten für die Auswertung ergänzt. Stehen jedoch Änderungen an dem Lexikon oder der Grammatik an, dann muss neben den Dateien, die dem Spracherkenner zur Konfiguration dienen, auch die Quelldatei des Parsers angepasst werden. Dabei entstehende Inkonsistenzen werden leicht übersehen, da diese nicht automatisch erkannt werden können.

Anders stellt sich das bei dem entwickelten Werkzeug `grmscan` zum Kreieren von Parsern für die Auswertung der Ergebnisse des Spracherkenners auf der Basis der dem Erkenner zugrunde liegenden Grammatik dar. Abbildung 6.3 zeigt den Datenfluss des hier vorgestellten Werkzeugs. Zusammen mit der ISR-Grammatikdatei und einer Konfigurationsdatei, die initial vom `grmscan` erstellt und weiterführend gepflegt wird, kann eine C-Datei generiert werden, die direkt kompilierbar ist und dem Projekt hinzugefügt werden kann. Die Konfigurationsdatei muss vom Entwickler vollständig ausgefüllt werden (siehe Abbildung 6.5). Erst jetzt kann die

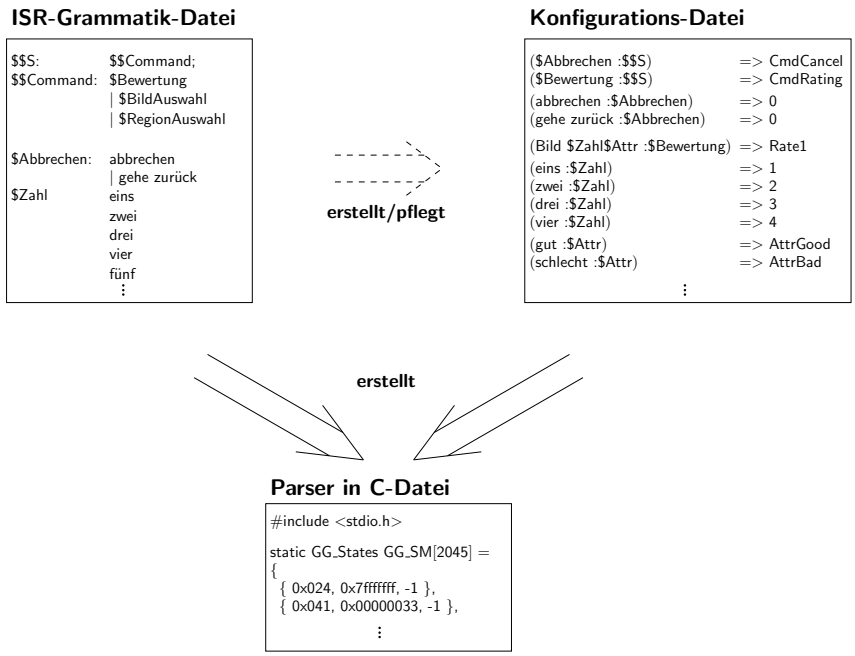


Abb. 6.3: **Automatische Generierung des Parsers:** Auf der Basis der Grammatikdatei wird eine Konfigurationsdatei erstellt bzw. gepflegt. Die Zuweisungen der gewünschten Nonterminale der Grammatik werden in der Konfigurationsdatei eingetragen. Basierend auf den beiden oberen Dateien wird ein Parser erstellt, der eine Funktion exportiert, die für jede Äußerung aufgerufen werden muss.

resultierende C-Projektdatei erstellt werden. Bei einer Änderung der Grammatik wird durch `grmscan` auch die Konfigurationsdatei verändert bzw. erweitert und ein Fehler generiert, wenn neue Sektionen der Konfigurationsdatei entstanden sind, die wiederum vom Entwickler zu behandeln sind.

6.2.1 Aufbau der ISR-Grammatik

Das hier vorgestellte Werkzeug baut auf einem speziellen Modus des Spracherkenners auf, in dem erkannte Nonterminale dem Ausgabedatenstrom hinzugefügt werden. Durch die Auswertung dieser Nonterminale kann auf eine erneute vollständige syntaktische und semantische Analyse, die bereits im ISR durchgeführt wurde, verzichtet werden. Vielmehr muss eine syntaktische und semantische Analyse des veränderten Datenstroms durchgeführt werden.

Die dem Spracherkenner zugrunde liegende Grammatik ist eine aus der Familie der LR-Grammatiken [Aho85]. Semantisch weisen die ISR-Grammatiken jedoch eine Besonderheit auf. Nonterminale, die immer mit einem vorangestellten Dollar-Zeichen gekennzeichnet sind, sind in zwei Klassen eingeteilt. Es existieren interne Nonterminale, die zwar gleichwertige Bestandteile der Grammatik sind, jedoch nach außen hin unsichtbar bleiben. Diese Nonterminale werden durch zwei führende Dollar-Zeichen gekennzeichnet. Die Nonterminale der anderen Klasse werden dem Ausgabedatenstrom wie in Abbildung 6.4 zu sehen beigefügt.

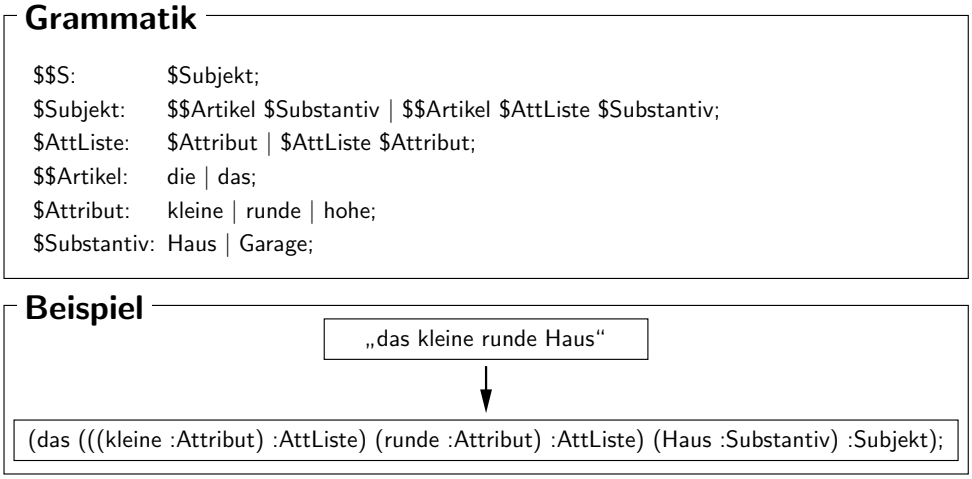


Abb. 6.4: **Beispiel ISR-Grammatik:** Mit der oben dargestellten Grammatik wird der Spracherkenner konfiguriert. Ein der Grammatik entsprechender Satz wird wie unten zu sehen ausgegeben. Nonterminale, deren Namen mit mit der $$-Zeichenkette beginnen, erscheinen nicht in der Ausgabe.

6.2.2 Konfiguration

Die ISR-Grammatik wird von `grmscan` analysiert und alle möglichen Kombinationen von Terminalen, die ganzen Worten entsprechen, und externen Nonterminalen für die angegebenen Regeln ermittelt, denn das Ziel ist es, der nachverarbeitenden Applikation nur die benötigten Informationen kompakt zur Verfügung zu stellen. Unwichtige Informationen sollen jedoch, soweit es möglich ist, verworfen werden. Die extrahierten Kombinationen werden, wie im weitergeführten Beispiel in Abbildung 6.5 zu sehen, in der Konfigurationsdatei eingetragen. Im Ge-

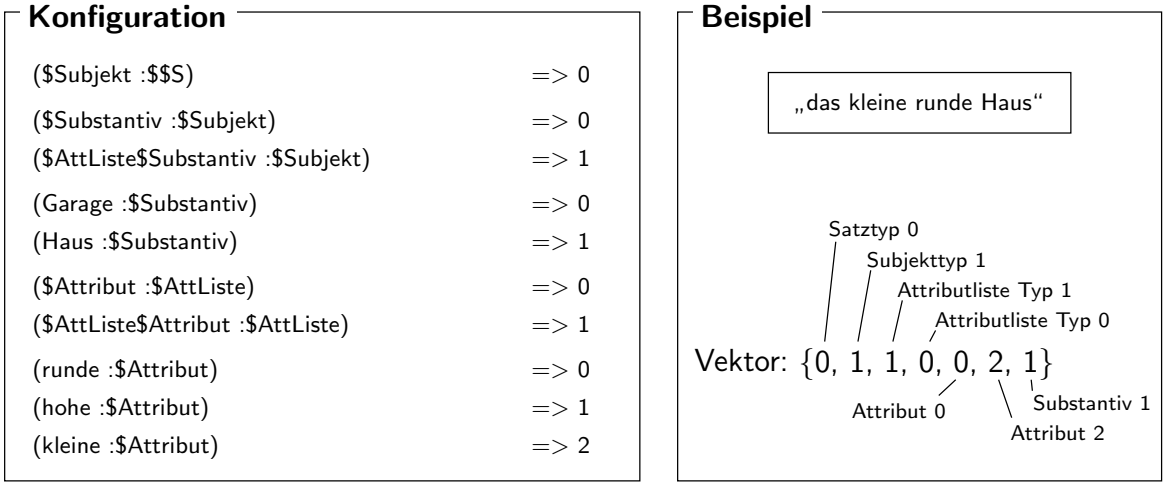


Abb. 6.5: **Beispiel Konfigurationsteil:** Die Basis dieser Konfigurationsdatei ist die in Abbildung 6.4 gezeigte Grammatik. Im linken Teil sind die extrahierten Regeln mit den vom Entwickler zugewiesenen Konstanten dargestellt. Der rechts dargestellte, der Applikation übergebene Vektor repräsentiert die bereits benutzte Beispieläußerung.

gensatz zum Spracherkenner werden Regeln für interne Nonterminale hier nicht berücksichtigt. Der Entwickler ist jetzt angehalten, jeder Kombination eine Konstante zuzuweisen, die dann in der nachgeschalteten Applikation ausgewertet werden kann. Dabei wird die erkannte Äußerung in Form eines Vektors, der mit den den Regelkombinationen zugeordneten Konstanten gefüllt wird, übergeben.

Um keinerlei Inkonsistenzen zwischen Konfiguration und Auswertung bezüglich der verwendeten Konstanten zu erhalten, ist es sinnvoll, diesen Konstanten unter Verwendung von C-Makros Namen zu geben. Damit diese Definitionen auch bei der Kompilierung vorhanden sind, gibt es weitere Sektionen in der Konfigurationsdatei, die das Einbinden von C-Quellcode erlauben. Der Gesamtaufbau der Konfigurationsdatei ist in Abbildung 6.6 gezeigt. Dieser ist dem eines

```
%
%{
  C-Code, der am Anfang der Zieldatei plaziert wird
%}
%%
  Konfigurationsteil (automatisch extrahierte Regeln mit
                      manuell hinzugefügten Zuweisungen)
%%
  C-Code, der nach den automatisch generierten Tabellen plaziert wird
```

Abb. 6.6: **Aufbau der Konfigurationsdatei:** Neben dem oben beschriebenen Konfigurationsteil besteht die Möglichkeit, C-Code im Kopf als auch am Ende der Datei zu plazieren.

lex-Files nachempfunden. Sowohl im Kopf als auch am Ende der Datei besteht die Möglichkeit, C-Code einzufügen, der ohne Änderungen in die Ausgabedatei kopiert wird.

6.2.3 Parsertabellen

`grmscan` generiert aus dem Konfigurationsteil der Konfigurationsdatei drei Tabellen, die in

Tabelle	Bedeutung
GG_SM	Eigentliche Zustandsmaschinen. Tripel enthalten das zu untersuchende Zeichen, den Index eines eventuellen Alternativzustands und einen eventuellen Endzustandsindex
GG_SSM	Startindizes der Zustandsmaschinen. Index 0 beinhaltet die Namen der Nonterminale. Alle weiteren Zustandsmaschinen stehen für je eine oder mehrere Regeln eines Nonterminals
GG_Result	Tabelle der zugewiesenen Rückgabewerte der erkannten Regeln für alle Endzustände

Tab. 6.2: **Übersicht der automatisch generierten Parsertabellen**

Tabelle 6.2 angeführt sind. Mithilfe dieser drei Tabellen ist es möglich, die vom Spracherkenner gelieferten Sätze zu analysieren.

Kapitel 7

Evaluierung

Die interhumane Kommunikation wird meist unbewusst auf vielen Kommunikationskanälen gleichzeitig durchgeführt. Neben den offensichtlichen Modalitäten wie Gestik und gesprochenen Worten ist Mimik, aber auch die Betonung beim Sprechen zu nennen. Je mehr Kanäle an einer multimodalen Kommunikation in konstruktiver Art und Weise beteiligt sind, desto sicherer bzw. erfolgreicher ist die durchgeführte Kommunikation [Ovi99].

Der Einsatz mehrerer Modalitäten bei einem technischen System ist aber nicht immer sinnvoll, auch wenn man entgegnen mag, dass die zwischenmenschliche Interaktion grundsätzlich multimodal geartet ist. Kann der Benutzer zum Beispiel die Modalitäten nicht wie gewohnt einsetzen oder verlangt das System eine Bedienung, die der Benutzer nicht gewohnt ist, dann führt dies schnell zu Verwirrung und Unmut. In solchen Fällen wird die gewünschte Funktionalität unter Umständen ganz abgelehnt oder über andere Kanäle durchgeführt. Dabei wird auch eine nicht vorgesehene und womöglich komplizierte Art der Interaktion akzeptiert, wenn dieser als natürlicher empfunden wird.

Die Frage, wie gut der Einsatz der multimodalen Bedienung gelungen ist, stellt sich generell bei der Entwicklung eines multimodalen Systems und sollte durch eine entsprechende Evaluierung beantwortet werden. Die von uns durchgeführte Evaluierung basiert auf den Daten eines Akzeptanztests, der hier im Folgenden vorgestellt wird.

7.1 Akzeptanztest

Im Gegensatz zu anderen Evaluierungen des Bilddatenbanksystems INDI [Käm02, Käs05] soll der hier vorgestellte Test lediglich die multimodale Bedienung des Systems beinhalten und die Antwort auf die oben aufgeworfene Frage geben, ob das System gut unter Verwendung der Modalitäten Gestik und Sprache zu bedienen ist. Die Evaluierung besteht aus zwei Akzeptanztests, wobei der hier dokumentierte direkt auf den ersten [Bau03, Käs03] mit den dort gewonnenen Erfahrungen aufbaut.

Mit den Evaluierungen einzelner Komponenten des Systems können Aussagen über deren Robustheit und Erkennungsraten getroffen, jedoch kein Urteil über die multimodale Benutzbarkeit gefällt werden. Hierzu ist es notwendig, das System als Ganzes dem Test zu unterziehen.

Die Qualität der Einzelkomponenten, wie beispielsweise die des eingesetzten Spracherkenners [Wac98, Fin99, Plö02], ist jedoch bei der Beurteilung des gesamten Systems durchaus einflussreich. Sollten vom System sprachliche Äußerungen oft nicht oder falsch verstanden werden, so fällt das Gesamturteil eher schlecht aus. Funktioniert die Komponente jedoch sehr gut, dann wird sie nicht als eigenständige Komponente wahrgenommen und damit die Beurteilung der multimodalen Bedienbarkeit eher nicht beeinflusst.

Die genannte Problematik besteht insbesondere bei dem Suchsystem. Die Tatsache, dass die zu lösende Aufgabe eine Bildsuche ist, jedoch die Bedienung des Bildsuchsystems im Fokus des Tests steht, wirft hier die Gefahr von Fehlbewertungen auf. Daher wurde das Konzept umgesetzt, die Bedienung des Systems in unterschiedlich gearteten Konstellationen bezüglich der zu verwendenden Modalitäten durchführen zu lassen. Die Ergebnisse in den unterschiedlichen Gruppen sind damit alle weitestgehend identisch beeinflusst und lassen sich dadurch direkt vergleichen.

Für die Bewertung der Bedienbarkeit des Systems sollten die von Preece [Pre02] vorgestellten Kriterien angewendet werden. Es handelt sich hierbei um die folgenden Grundgrößen, die Aussagen über die Bedienbarkeit zulassen:

Geschwindigkeit: Wie schnell kann die gestellte Aufgabe erfüllt werden?

Komplexität: Wie viele unterschiedliche Aufgaben können mit dem System erfüllt werden?

Qualität: Wie gut können die unterschiedlichen Aufgaben gelöst werden?

Lernerfolg: Wie schnell lernt der Benutzer, die unterschiedlichen Aufgaben mit dem System zu erfüllen?

Konzentration: Muss der Benutzer sehr bedacht arbeiten, um keine Bedienungsfehler zu machen?

Zufriedenheit: Macht es dem Benutzer Spaß, mit dem System zu arbeiten?

Nicht alle der hier aufgeführten Größen sind für die Bewertung von INDI bzw. für die Gegenüberstellung der unterschiedlichen Modalitäten bezüglich ihrer Benutzbarkeit sinnvoll. Die benötigten Daten sind jedoch zum Teil nicht messbar. Aus diesem Grund wurde ein Fragebogen entworfen, um aus dem persönlichen Eindruck der Testpersonen auf die fehlenden Daten schließen zu können. Ein solcher Fragebogen muss von den Testpersonen im Anschluss an die Arbeit mit dem System ausgefüllt werden.

Testablauf

Eine beliebige Testperson in die Lage zu versetzen ein komplexes System wie das Bildsuchsystem INDI, einsetzen zu können, erfordert, obwohl bei der Entwicklung Wert auf eine natürliche und intuitive Bedienung gelegt wurde, eine entsprechende Einführung. Sowohl die als unbekannt anzusetzende Art der Bildsuche als auch die Möglichkeiten, die die Bedienung mit den in diesem Bereich noch unkonventionellen Modalitäten Gestik und Sprache bietet, müssen den Probanden nahe gebracht werden.

Zur Einführung wurde daher ein Video erstellt, das den Testpersonen zu Beginn des Tests vorgeführt werden konnte. Dadurch wurde gewährleistet, dass jede Testperson dieselben, notwendigen Informationen erhält, was ansonsten bei der Fülle der Informationen nicht zu garantieren gewesen wäre. Das Video, dessen Drehbuch im Anhang B zu finden ist, weist die folgende Struktur auf:

Begrüßung und Einleitung:

Bildsuche: Es erfolgt eine Einweisung in die Bildsuche und die Bedienung der Applikation mit der Maus. Das Video zeigt sowohl das Interaktions-Szenario als auch Details der Bedienoberfläche. Von einem Sprecher werden die gezeigten Interaktionen erläutert.

Bedienung in den möglichen Modalitäten: Der Reihe nach wird die Bedienung in der Modalität Sprache, die besonderen Möglichkeiten, die bei der Benutzung des Touchscreens gegeben sind, und die Interaktion in der Kombination Maus/Touchscreen und Sprache vorgestellt.

Experiment: Schließlich wird erläutert, welche Aufgabe in dem bevorstehenden Experiment auf die Testperson zukommt.

Zur Verfestigung des gerade Aufgenommenen konnte das System anschließend ausprobiert werden. Dazu wurden die Testpersonen mündlich aufgefordert, eine Bildsuche durchzuführen und dabei die zugewiesenen Modalitäten zu benutzen. Während dieser Phase wurde die Person zum Klären von Nichtverstandenem ermutigt, da die sich anschließenden Testphase nicht von solchen Unterbrechungen gestört werden sollte.

Nach der Durchführung des Experiments mussten die Teilnehmer in dem Fragebogen den Eindruck, den das System hinterlassen hatte, dokumentieren. Um eine eventuelle Beeinflussung des Testergebnisses zu vermeiden, wurden weiterführende Fragen der interessierten Testpersonen erst nach der Abgabe des Fragebogens beantwortet.

Das Experiment

Für das Experiment wurde eine Datenbank mit 1250 Bilder der ArtExplosion-Bildsammlung (siehe Fußnote auf Seite 89) erstellt. Die Bilder wurden zu gleichen Teilen aus zehn verschiedenen Kategorien wie beispielsweise Golf, Ballon oder Blumen ausgewählt. Die Aufgabe, die die Testpersonen zu bewältigen hatten, bestand aus drei Suchdurchläufen vom Typ Zielsuche (siehe Abschnitt 3.2.1). Abbildung 7.1 zeigt die drei Bilder, die in der Datenbank gefunden werden sollten. Nacheinander wurde je eines dieser Bilder den Testpersonen vor dem Suchdurchlauf ausgehändigt. Dabei wurde eine feste Reihenfolge eingehalten. Für jeden Suchdurchlauf bekam die Testperson drei Minuten Zeit. Konnte das Bild in dieser Zeit gefunden werden, so wurde diese Suche als erfolgreich gewertet.

Sieben weibliche und 33 männliche Testpersonen im Alter von 19 bis 37 Jahren wurden auf dem Campus der Universität Bielefeld aus unterschiedlichen akademischen Richtungen, wie beispielsweise Lehramt, Psychologie, Rechtswissenschaften aber auch Informatik, rekrutiert.

Abb. 7.1: **Bilder der Zielsuche:** Für jedes der Bilder wurde eine Zielsuche durchgeführt. Dabei wurde die hier abgebildete Reihenfolge von links beginnend eingehalten.

Alle Testpersonen gaben an, keine besondere Erfahrung bezüglich Bilddatenbanken gesammelt zu haben.

Wie einleitend erwähnt, wurden die Testpersonen in Gruppen, die sich bezüglich der zu verwendenden Modalitäten unterscheiden, eingeteilt. Die folgenden vier Gruppen wurden für den Test ausgewählt:

Maus: Die erste Gruppe sollte die Applikation mit der als bekannt anzusetzenden Modalität Maus bedienen. Diese Gruppe (bzw. synonym Modalität) wird im Folgenden mit *M* abgekürzt.

Touchscreen: Die zweite Gruppe sollte zur Bedienung lediglich den Touchscreen-Monitor benutzen. Hier sollten sowohl Standard-Interaktionen wie das Drücken der Schaltflächen durch einfaches Antippen der entsprechenden Fläche als auch die in Kapitel 5.3 besprochenen Touchscreen-Gesten eingesetzt werden. Diese Gruppe erhält als Kürzel ein *T*.

Maus - Sprache: Diese Gruppe (*MS*) erhielt die Möglichkeit, neben den Aktionen mit der Maus sowohl sprachliche Äußerungen als auch kombinierte Interaktionen für die Steuerung der Applikation einzusetzen.

Touchscreen - Sprache: Die Teilnehmer der Gruppe *TS* sollten die Interaktionen in der Form durchführen, die als am wenigsten bekannt anzusetzen ist. Auch hier konnten und sollten Interaktionen in einzelnen Modalitäten wie auch in deren Kombination durchgeführt werden.

Die nicht angegebenen drei Gruppierungen bieten keine sinnvolle Möglichkeit der Interaktion. Obwohl die gesamte Funktionalität auch unter ausschließlichem Einsatz von Sprache ausgeschöpft werden kann, wäre ein so konfiguriertes System keines, das als benutzbar zu bezeichnen wäre. Das Aneinanderhängen von zum Teil längeren sprachlich verfassten Kommandos würde schnell als zu kompliziert und damit als störend und unnatürlich empfunden werden.

Die Kombination von Maus und Touchscreen wurde deshalb als unnatürlich eingestuft, weil für beide technischen Geräte die Hand als Aktor einzusetzen ist und damit eine Konkurrenzsituation auftritt.

7.2 Auswertung der Ergebnisse

Die zur Auswertung notwendigen Daten wurden, wie erwähnt, zweigleisig gesammelt. Systemseitig besteht die Möglichkeit, einen exakten Mitschnitt der durchgeführten Aktionen zu führen. Dieser Mitschnitt wird in einer Datei mit einem festgesetzten Namen fixiert. Es galt sicherzustellen, dass eine eindeutige Zuordnung der Daten des Fragebogens zu der vom System erstellten Datei stattfand.

Da die Betreuung des Tests von unterschiedlichen Personen durchgeführt werden sollte, wurde die in Abbildung 7.2 gezeigte Testapplikation erstellt. Diese Applikation vergabt eine eindeutige

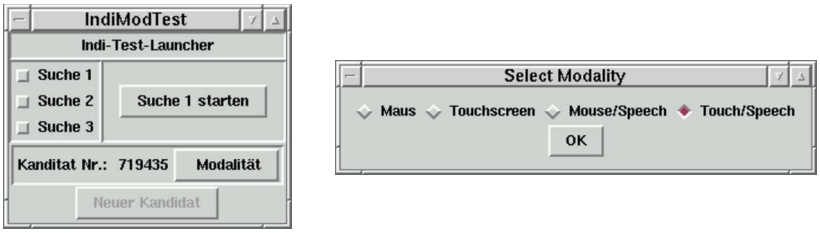


Abb. 7.2: **Test Manager:** Mithilfe dieser Applikation wurde die technische Komponente bei der Testdurchführung gesteuert. Die Datenbankapplikation wurde vor jedem Suchdurchlauf initialisiert und neu gestartet. Außerdem wurde der Testperson eine eindeutige Nummer zugewiesen, die auf dem zugehörigen Fragebogen notiert werden musste.

Experimentnummer, die auf dem Fragebogen zu notieren waren. Unter dieser Nummer wurde der Mitschnitt der Datenbankapplikation abgelegt. Des Weiteren wurde durch den Einsatz der Testapplikation sichergestellt, dass das System für jede Testperson in identischer Art und Weise zur Verfügung stand. Um beispielsweise den Sucherfolg vergleichen zu können, war es wichtig, den Zufallszahlen-Generator, der für die Zusammenstellung der Menge der initial präsentierten Bilder benötigt wird, immer identisch zu initialisieren.

Systemdaten

Die Daten, die von dem System bei jeder Benutzung mitgeschnitten werden, sind in drei Klassen eingeteilt. Es handelt sich zum einen um Daten, die für die Rekonstruktion einer Suche notwendig sind und zum anderen um Datensätze, die lediglich dokumentieren, dass eine Aktion wie beispielsweise das Öffnen der Großansicht eines Bildes durchgeführt wurde. Schließlich wird das vom Datenbank-Server gelieferte Suchergebnis in der Aufzeichnungsdatei abgelegt (vergleiche Abschnitt 5.6, Client als Testwerkzeug). In allen drei Fällen wird der Zeitpunkt der Aktion und die beteiligte Modalität abgespeichert.

Mit dem Zeitpunkt der ersten Aktion begann für das System das Experiment. Durch die statische Datenbank waren die Identifikationsnummern der gesuchten Bilder bzw. deren Bildregionen konstant. Deshalb ließen sich aus den Daten des Mitschnitts folgende Messgrößen für die Auswertung automatisch ermitteln:

Erfolg: Konnte das gesuchte Bild innerhalb der zur Verfügung gestellten 180 Sekunden gefunden werden?

Suchdauer: Die Zeit, die für eine Suche benutzt wurde. Das entspricht im Fall von Sucherfolg, der Zeit die zum Finden des Bildes benötigt wurde, und im anderen Fall die zur Verfügung gestellte Zeit von 180 Sekunden.

Aktionen: Die Gesamtanzahl aller durchgeführten Aktionen.

Bewertungen: Die Gesamtanzahl aller Bewertungsaktionen. Hier gehen auch Bewertungen ein, die korrigiert wurden.

Suchiterationen: Die Anzahl der durchgeführten Suchiterationen.

Dauer einer Iteration: Die durchschnittliche Zeit, die für die Durchführung einer Suchiteration benötigt wurde.

Aktionen einer Iteration: Die durchschnittliche Anzahl von durchgeführten Aktionen innerhalb einer Iteration.

Bewertungen einer Iteration: Die durchschnittliche Anzahl von Bewertungen innerhalb einer Suchiteration.

Der Abbruch einer nicht erfolgreichen Suche nach der vorgegebenen Zeit führte zu der besonderen Behandlung der Durchschnittswerte von Zeit, Aktionen und Bewertungen einer Suchiteration (Abbildung 7.3 verdeutlicht die unterschiedliche Behandlung). Die Experimentphase nach

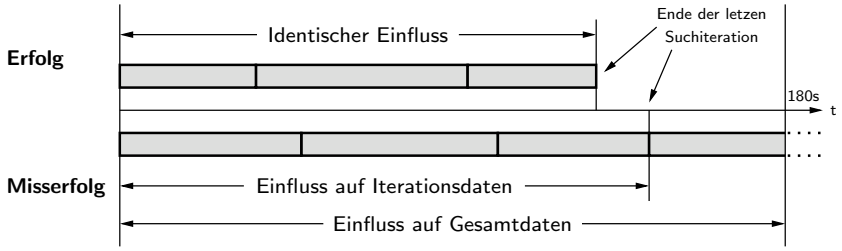


Abb. 7.3: **Datengrundlage bei der Ermittlung der Iterationswerte:** Im oberen Fall, bei dem das Experiment erfolgreich nach drei Suchiterationen mit dem Finden des gesuchten Bildes abgeschlossen werden konnte, bilden alle Daten die Grundlage für die Bestimmung der Iterationswerte. Wurde eine nicht erfolgreiche Suche nach 180 Sekunden abgebrochen, wie im unteren Fall zu sehen, gehen nur die Daten bis zur letzten abgeschlossenen Suchiteration in die Iterationswerte ein.

der letzten abgeschlossenen Suchiteration ging nicht in die Bestimmung der Iterationswerte ein. Im Falle eines Sucherfolges endete die Suche mit der Präsentation des Suchergebnisses, so dass hier alle Aktionen auf die Iterationswerte Einfluss nahmen.

Wie einleitend erwähnt, soll die Auswertung nach den von Preece [Pre02] vorgestellten und an das System angepassten Kriterien erfolgen. Die aufgeführten Kriterien, die über Messgrößen

ermittelt werden, können oft nicht direkt gewonnen werden. Vielmehr können die Messgrößen lediglich Indizien liefern, die Aussagen über die entsprechenden Kriterien erlauben. Die oben aufgeführten Messgrößen sollen Aufschluss über die Attribute Geschwindigkeit, Qualität und Lernerfolg geben.

Die Aufzeichnung der vom Benutzer durchgeführten Aktionen wurde bei diesem Experiment sehr viel detaillierter durchgeführt, auch um Gewissheit darüber zu bekommen, ob die Anzahl der Bewertungen ein Maß für die Geschwindigkeit ist, mit der der Benutzer die Applikation bedient.

Fragebogen

Zur Ermittlung der Daten, die nicht durch Auswertung von Messdaten gewonnen werden können, wurde, wie einleitend erwähnt, in Zusammenarbeit mit Kollegen aus der Psychologie ein Fragebogen entworfen.

In dem Fragebogen, der im Anhang A abgebildet ist, können die interessierenden Fragen direkt verfasst werden. Es gibt jedoch einige Richtlinien, die beachtet wurden:

Stärkung des seriösen Eindrucks: Durch für den Test eher unwichtige Fragen, wie die über Alter oder Geschlecht, gewinnt der Fragebogen an Seriosität und wird von den Testpersonen ernster genommen.

Konträre Fragestellungen: Durch konträre Fragestellungen, die mit einem Eintrag aus einer Bewertungs-Skala beantwortet werden, wird der Benutzer gezwungen, unterschiedliche Bewertungen durchzuführen, das heißt er kann sich nicht auf eine einheitliche Bewertung festlegen. So können Testpersonen ausfindig gemacht werden, deren Testergebnisse keine hohe Aussagekraft haben. Des Weiteren wird die Testperson dazu gezwungen, öfter über die Bedeutung der Bewertungen nachzudenken

Formulierung: Die Fragen müssen einfach und möglichst kurz formuliert werden, um schnell und sicher verstanden zu werden. Durch die persönliche Ansprache mittels „Du“ wurde versucht, ein möglichst angenehmes persönliches Verhältnis zu schaffen, so dass die Fragen ohne Stress der Testsituation beantwortet werden konnten.

Ergebnisse und Diskussion

Die Auswertung der gemessenen Werte und die Antworten des Fragebogens haben ergeben, dass auch die hier vorgestellte zweite Evaluierung, bei der die Aufzeichnung der gemessen Werte verfeinert und vor allem die Menge der Testpersonen vergrößert wurde (vergleiche [Bau03, Käs03]), keine gesicherte statistische Aussage zulässt. Wenige der Größen zeigen eine Normalverteilung, Signifikanzen lassen sich nicht feststellen. Die Daten des Tests, die in den folgenden Tabellen und Abbildungen aufgeführt sind, lassen jedoch Trendaussagen zu.

Die Auswertung der gemessenen Werte erfolgt auf zwei unterschiedlichen Bezugssystemen. Tabelle 7.1 zeigt die Gegenüberstellung der Messwerte bezüglich der gesuchten Bilder. Alle hier

(a) Mittelwerte aller Bildsuchen

Bild	Sucherfolg	Zeit / Experiment	Aktionen / Experiment	Bewertungen / Experiment	Suchschritte / Experiment	Zeit / Suchschritt	Aktionen / Suchschritt	Bewertungen / Suchschritt
Autorennen	60.0%	125.35	33.70	10.38	3.33	37.15	10.11	2.88
Ballon	45.0%	131.62	45.20	12.72	4.28	31.44	10.97	3.11
Blume	60.0%	126.75	53.48	11.57	4.60	28.77	11.72	2.60

(b) nur erfolgreiche Bildsuchen

Bild	Zeit / Experiment	Aktionen / Experiment	Bewertungen / Experiment	Suchschritte / Experiment	Zeit / Suchschritt	Aktionen / Suchschritt	Bewertungen / Suchschritt
Autorennen	88.92	26.71	9.67	2.96	30.81	9.02	2.97
Ballon	72.50	29.89	9.22	3.11	25.44	10.28	3.21
Blume	91.25	38.17	10.46	3.62	25.59	10.26	2.66

(c) nur Bildsuchen ohne Erfolg

Bild	Aktionen / Experiment	Bewertungen / Experiment	Suchschritte / Experiment	Zeit / Suchschritt	Aktionen / Suchschritt	Bewertungen / Suchschritt
Autorennen	44.19	11.44	3.88	46.65	11.74	2.74
Ballon	57.73	15.59	5.23	36.34	11.54	3.02
Blume	76.44	13.25	6.06	33.54	13.90	2.51

Tab. 7.1: **Gegenüberstellung der Resultate bezüglich der Bilder:** Die oben angegebenen Messwerte wurden bezüglich der gesuchten Bilder in einer Mittelwertbildung ausgewertet. Die Zeitwerte sind in Sekunden angegeben. Durch die separaten Tabellen für erfolgreiche Suchen bzw. solchen ohne Erfolg lassen sich Korrelationen der Werte zum Sucherfolg leicht nachvollziehen. Die Bilder wurden in der hier angegebenen Reihenfolge gesucht.

und im Folgenden präsentierten Messwerte sind Mittelwerte bezüglich der Bewertungsgrundlage. Um Effekte, die durch die unterschiedliche Bearbeitungszeit von erfolgreichen Suchen und solchen, bei denen kein Erfolg zu verzeichnen war, sichtbar zu machen, ist auch die Auswertung in Teil (b) und (c) der Abbildung getrennt dargestellt. Die Auswertungen bezüglich der Suchbilder erlauben zwei Trendaussagen:

1. Die Aktionen der Testpersonen, die für die unterschiedlichen Suchen angewendet wurden, steigt von Suche zu Suche an. Dieses Verhalten ist sowohl bei erfolgreichen als auch nicht erfolgreichen Suchen zu verzeichnen. Daraus lässt sich ableiten, dass die Testpersonen einen hohen Lernerfolg hatten und sich schnell in der Bedienung sicher fühlten.

2. Die Bildsuchen waren von unterschiedlichem Schwierigkeitsgrad. Die Suche nach dem Ballon-Bild wurde am wenigsten von Erfolg gekrönt.

   Interessant ist hierbei, dass die durchschnittliche Zeit bei erfolgreichen Suchen dieses Bildes deutlich niedriger ist als bei den anderen Bildern. Das könnte darauf zurückzuführen sein, dass die Anfragen bzw. Bewertungen auf Basis einer unterschiedlichen semantischen Bildbetrachtung geschehen ist. So könnten beispielsweise einige Testpersonen beliebige Bilder, die einen oder mehrere Ballons zeigen, positiv bewertet haben, wohingegen sich

andere Benutzer dazu entschieden haben, eher den Gesamteindruck des Bildes zu bewerten. Bei diesem Vorgehen würden beispielsweise nur Ballon-Bilder mit ähnlichem Hintergrund als positive relevant klassifiziert werden (siehe Abbildung 7.1). Die hier vorliegenden Werte spiegeln also klar den Einfluss, den das inhaltsbasierte Suchsystem auf die Ergebnisse ausübt, wider.

(a) Mittelwerte aller Bildsuchen

Modalität	Sucherfolg	Zeit / Experiment	Aktionen / Experiment	Bewertungen / Experiment	Suchschritte / Experiment	Zeit / Suchschritt	Aktionen / Suchschritt	Bewertungen / Suchschritt
M	55.6%	129.96	58.81	11.44	4.67	32.10	13.39	2.90
T	54.5%	128.94	50.24	17.03	4.45	29.58	11.41	3.93
MS	43.3%	131.87	39.30	9.97	3.73	33.13	10.92	2.44
TS	66.7%	120.97	29.00	7.23	3.43	35.24	8.20	2.07

(b) nur erfolgreiche Bildsuchen

Modalität	Zeit / Experiment	Aktionen / Experiment	Bewertungen / Experiment	Suchschritte / Experiment	Zeit / Suchschritt	Aktionen / Suchschritt	Bewertungen / Suchschritt
M	89.93	38.60	10.00	3.13	31.31	12.58	3.01
T	86.39	39.44	13.83	3.67	24.67	11.64	3.87
MS	68.92	27.00	9.69	3.08	21.94	7.95	2.78
TS	91.45	22.75	6.20	3.05	30.64	7.31	2.09

(c) nur Bildsuchen ohne Erfolg

Modalität	Aktionen / Experiment	Bewertungen / Experiment	Suchschritte / Experiment	Zeit / Suchschritt	Aktionen / Suchschritt	Bewertungen / Suchschritt
M	84.08	13.25	6.58	33.09	14.40	2.76
T	63.20	20.87	5.40	35.48	11.14	4.00
MS	48.71	10.18	4.24	41.69	13.20	2.19
TS	41.50	9.30	4.20	44.45	9.98	2.02

Tab. 7.2: **Gegenüberstellung der gemessenen Werte bezüglich der benutzten Modalitäten:** Die oben angegebenen Mittelwerte der Messungen wurden hier bezüglich der benutzten Modalität ermittelt. Die verwendeten Kürzel *M*, *MS*, *T* und *TS* entsprechen den auf Seite 104 aufgeführten Modalitäten.

Tabelle 7.2 beinhaltet die Messwerte aufgeteilt nach der verwendeten Modalität. Auch hier lassen sich Zusammenhänge erkennen:

1. Betrachtet man die Anzahl von Aktionen, die in einer bestimmten Zeit durchgeführt wurden, dann schlagen die multimodalen Suchen mit einer etwas geringeren Rate zu Buche. Das bestätigt die Tatsache, dass Aktionen, die durch Sprachäußerungen ausgelöst wurden, etwas länger dauern. Eventuell werden die Aktionen aber auch aufgrund ihrer Neuartigkeit mit mehr Bedacht eingesetzt.

2. Der erzielte Sucherfolg scheint wenig von dem Benutzermodus abzuhängen. Die Gruppe MS erzielte einen deutlich geringeren Erfolg als die andere multimodale Gruppe TS. Die unimodalen Gruppen M und T liegen bezüglich des Erfolges im Mittelfeld.

   Auch hier ist zu vermuten, dass das inhaltsbasierte Suchsystem einen erheblichen Einfluss auf die gemessenen Ergebnisse hat (vergleiche Sucherfolg und Zeit pro Experiment der Gruppe MS mit den entsprechenden Werten der Ballon-Suche aus Tabelle 7.1). Da genau dieser Einfluss durch das Gegenüberstellen der Bedienung in unterschiedlichen Modalitäten vermieden werden sollte, ist daraus zu schließen, dass die Menge der Testpersonen für die Auswertung dieser Messwerte nicht ausreicht.

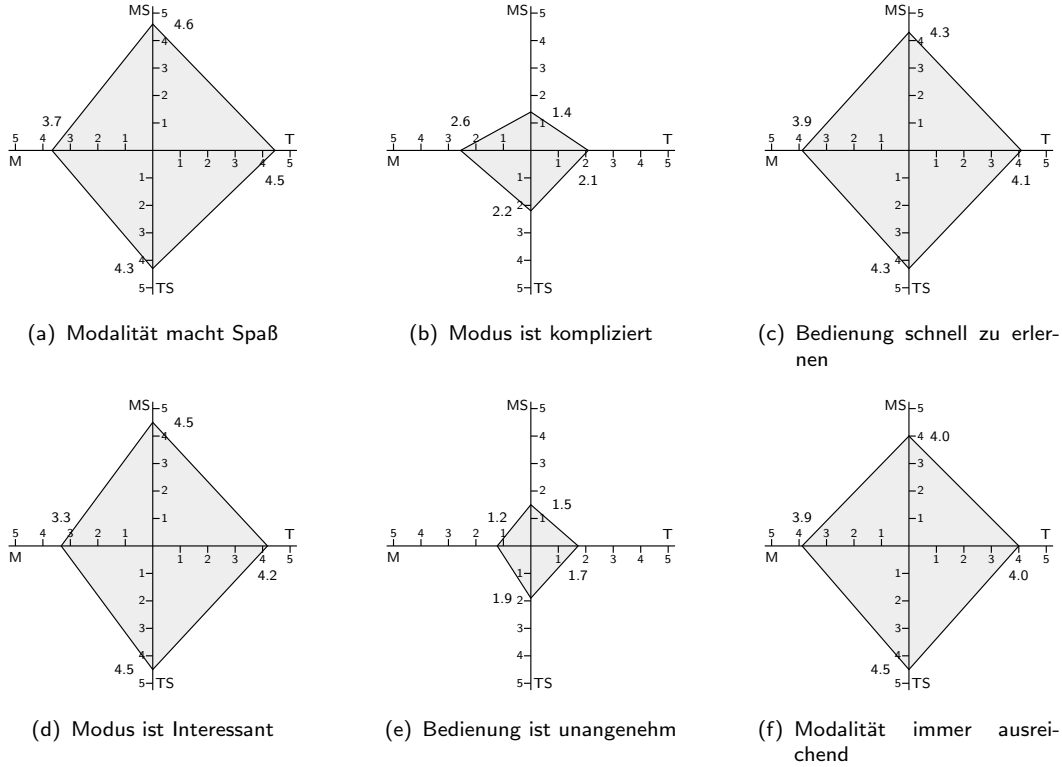
Deutlichere Tendenzen lassen sich aus den Ergebnissen des Fragebogens, die in Tabelle 7.3 dargestellt sind, ermitteln. Die Fragen lassen sich auf acht eigentliche Antworten zusammen-

Frage	**M**	**T**	**MS**	**TS**
Die Arbeit mit dieser Modalität kommt mir entgegen	3.2	3.2	3.9	3.3
Der Modus ist kompliziert zu bedienen	2.6	2.1	1.4	2.2
Korrekturen sind schnell durchzuführen	3.2	3.2	3.7	3.4
Die Bedienung erfordert viel Geduld	2.9	3.0	2.5	2.7
Die Anfrage konnte über diesen Modus effektiv bewältigt werden	3.6	4.1	3.8	4.2
Die Modalität war für mich immer ausreichend	3.9	4.0	4.0	4.5
Der Benutzermodus ist einfach zu handhaben	3.7	3.5	4.6	4.0
Die Bedienung war mir unangenehm	1.2	1.7	1.5	1.9
Es macht Spaß, über diese Modalität im Bildregister zu arbeiten	3.7	4.5	4.6	4.3
Ich habe mich über die Bedienung geärgert	2.0	1.7	1.5	1.6
Es ist interessant, mit diesem Modus zu arbeiten	3.3	4.2	4.5	4.5
Die Modalität erfordert überflüssige Anweisungen	3.2	2.6	2.1	2.8
Die Bedienung ist schnell zu lernen	3.9	4.1	4.3	4.3

Tab. 7.3: **Auswertung des Fragebogens bezüglich der Modalitäten:** Hier abgebildet ist eine vollständige Gegenüberstellung der Antworten auf die Fragen des Fragebogens. Gleich- bzw. gegenläufige Zusammenfassungen sind durch die Stärke der Trennlinien gekennzeichnet. Für die hier angegebenen Werte wurden die Antworten aus jeder Gruppe zu Mittelwerten zusammengefasst. Die Bewertungs-Skala umfasst Antworten von nein (1) bis ja (5) (siehe hierzu auch Anhang A).

fassen, wenn Aussagen, wie beispielsweise „Die Arbeit mit dieser Modalität kommt mir entgegen“ und „Der Modus ist kompliziert zu bedienen“, aufgrund von Gegen- oder Gleichläufigkeit vereint werden. Abbildung 7.4 zeigt eine Auswahl der Fragen, die hier zur Interpretation herangezogen werden sollen. Folgende Interpretationen liegen auf der Hand:

1. Die direkte Frage nach dem Spaß (Abbildung 7.4(a)), den die Bedienung des Systems in der entsprechenden Modalität erzeugt, bringt hier die Vermutung nahe, dass alles, was



(a) Modalität macht Spaß

(b) Modus ist kompliziert

(c) Bedienung schnell zu erlernen

(d) Modus ist Interessant

(e) Bedienung ist unangenehm

(f) Modalität immer ausreichend

Abb. 7.4: **Ausgewählte Ergebnisse des Fragebogens:** Sechs der acht Einschätzungen sind hier als Diagramm aufgetragen. Ungleiche Ausprägung der Fläche kennzeichnet die Unterschiede der Messwerte.

nicht konventionell ist und als interessant empfunden wird (vergleiche Teil (d)), auch Spaß macht.

Dieses Urteil könnte auch Einfluss auf die Beantwortung der Frage, ob die Modalität für die Bedienung immer ausreichend ist (siehe Teil (f)), haben. In Anbetracht der Tatsache, dass der Funktionsumfang in allen Modalitäten identisch ist, stellt die unterschiedliche Beantwortung einen Widerspruch dar. Hier ist festzustellen, dass je außergewöhnlicher sich die Bedienung gestaltet, desto höher die Betonung ist, dass die Modalität einen genügend großen Funktionsumfang aufweist.

2. Wenig Varianz bezüglich der unterschiedlichen Modalitäten zeigt die Beurteilung, ob die Bedienung des Systems schnell zu erlernen ist (Teil (c)). Dabei ist zu vermuten, dass die höheren Bewertungen der multimodalen Bediener darauf zurückzuführen ist, dass ein Erstaunen darüber herrschte, wie gut und einfach sich ein System mittels sprachlicher Äußerungen steuern lässt.

3. Keine der Bedienungen in den unterschiedlichen Modalitäten wurde als sonderlich unangenehm ausgewiesen (vergleiche Teil (e)).

   Die Benutzung der Sprache bei der Bedienung einer Maschine wird eher als unangenehmer Faktor eingeschätzt. Umso erstaunlicher ist es, dass die Benutzung des Touchscreen-Displays unangenehmer empfunden wurde als die der Kombination Maus und Sprache.

Dieser Beobachtung könnte darauf zurückzuführen sein, dass die Bedienung von Dialogen einer grafischen Oberfläche eher für die Maus ausgelegt ist. So ist beispielsweise das Scrollen der Bildauswahl mit den Touchscreen-Display aufgrund der etwas kleinen Schaltflächen nicht so gut bedienbar, wie mit der Maus bzw. durch den Einsatz von sprachlichen Äußerungen.

4. Interessant ist das Ergebnis der Frage, ob der Bedienmodus kompliziert sei (siehe Teil (b)). Nach diesem Ergebnis ist das System mit Maus allein deutlich komplizierter zu bedienen als in der Kombination Maus und Sprache. Eine Erklärung hierfür ist die unterschiedliche Funktionsbelegung der Maustaste, die zu Verwirrungen führen kann. Die Modi, die das Touchscreen-Display beinhalten, liegen bei dieser Beurteilung im Mittelfeld. Hier kann die oben angesprochene Funktionalität auch durch Einsatz der Touch-Gesten erreicht werden.

Zusammenfassend kann hier festgehalten werden, dass der zweite Akzeptanztest die Hauptaussage des ersten Tests vollkommen bestätigt.

Eignung und Spaß: Die multimodale Bedienung des hier vorgestellten interaktiven intelligenten Bildsuchsystems steht den unimodalen Bedienungsformen bezüglich der zu lösenden Aufgabe in nichts nach. Vielmehr ist bei der multimodalen Arbeit weniger Ärger und deutlich mehr Spaß vorhanden.

Lernerfolg: Bezüglich des Lernerfolges kann gesagt werden, dass auch hier die multimodalen Bedienungsformen ähnlich gute Erfolgsraten vorweisen können, was sich in der zügig schneller werdenden Arbeit widerspiegelt.

Diese Ergebnisse lassen die Behauptung zu, dass die natürliche Interaktion durch Benutzung von Sprache und Touch-Gesten ideal bei der Bedienung eines iterativen Bildsuchsystem eingesetzt werden können.

Kapitel 8

Zusammenfassung und Ausblick

Durch den Einzug digitaler Kameras in die privaten Haushalte ist auch hier ein enormer Zuwachs der digitalen Bilddatenbestände zu verzeichnen. Diese Entwicklung wurde im kommerziellen Bereich bereits länger beobachtet und führte zu dem Bestreben, entsprechende Techniken für eine einfache Verwaltung der Bilddatenbestände zu entwickeln.

Subjektivität und der enorme manuelle Arbeitsaufwand, der bei den zunächst eingesetzten textbasierten Bildsuchsystemen zu verzeichnen war, führten zu der Forderung, die formale Repräsentation der Bilder ausschließlich aus dem visuellen Bildinhalt zu extrahieren. Diese so genannten inhaltsbasierten Bilddatenbanksysteme sind in der Lage, die Extraktion der für die Suche notwendigen Daten automatisch durchzuführen. In diesem Bereich der Forschung ist ein enormer Anstieg von Aktivitäten festzustellen [Rui99b, Sme00, Dat05].

Datta et. al [Dat05] stellen unter anderem fest, dass es nun an der Zeit sei, im Forschungsbereich inhaltsbasierter Bildsuche mehr Gewicht auf die Entwicklung von Endanwendungen zu legen, um den Anforderungen, die ein gemeiner Anwender an ein solches System stellt, gerecht zu werden. Ein Teil davon konnte durch die hier vorgestellte Arbeit, deren Ziel es war, intelligente Techniken der inhaltsbasierten Bildsuche mit natürlich gearteter Interaktion zu kombinieren, geleistet werden. Die Bedienung des zu entwickelnden Systems sollte so geartet sein, dass es auch ohne Vorwissen in den Bereichen Bildverarbeitung und Bildsuche erfolgreich bedient werden kann. Ein solches System öffnet sich einer entsprechend großen Menge von Anwendern.

Zusammenfassung

Zu Beginn dieser Arbeit wurden zunächst grundlegende Prinzipien der Funktionsweise inhaltsbasierter Bildsuche vorgestellt. Anhand von Systembeispielen wurden gängige Bildsuchtechniken angeführt. Iterative Bildsuchsysteme, die den Benutzer in den Suchprozess integrieren, heben sich hervor, da durch den Einsatz eines geeigneten Lernprozesses die semantische Lücke, die zwischen der formalen Bilddarstellung des Systems und der subjektiven Beschreibung des Anwenders besteht, verkleinert werden kann. Unterschiedliche Techniken, die zur Interaktion mit Systemen verwendet werden, wurden im Anschluß angeführt. Ein Schwerpunkt wurde hier

auf die natürlichen Kanäle der Interaktion gelegt und die technischen Herausforderungen, die bei der Realisation von Erkennern natürlicher Gesten bestehen, wurden herausgestellt.

Aus den Anforderungen, die an das zu entwickelnde System gestellt wurden, wurde ein Systemkonzept erstellt. Dabei wurden die wichtigsten Systemkomponenten wie die Gesamtarchitektur mit den dafür notwendigen Kommunikationsmechanismen, die iterative inhaltsbasierte Bildsuche und die Gestaltung der Interaktionsmöglichkeiten sowie die Umsetzung des modularen Systemaufbaus bezüglich alternativer Realisierungsmöglichkeiten analysiert. Die Analysen führten zu dem im Anschluss vorgestellten Gesamtkonzept, einem hierarchischen iterativen inhaltsbasierten Bildsuchsystem, das sowohl unter Verwendung der Maus als auch durch den Einsatz von Sprache und Gesten am Touchscreen-Display bedient werden kann.

Die Dokumentation der Umsetzung des Systemkonzepts erfolgte dann entsprechend dem Aufbau des Systems in den Teilen Datenbank-Server und -Client. Auf der Seite des Servers wurde zunächst die Datenorganisation mit der dazu notwendigen Zugriffstechnik vorgestellt. Anschließend wurde der Aufbau der Server-Applikation präsentiert, der durch die Anforderung, diese auch von einem Web-Client benutzbar zu machen, maßgeblich geprägt ist. Aus der Liste der verfügbaren Server-Dienste wurden drei herausgestellt, die die maßgebliche Server-Funktionalität ausmachen. Dabei handelte es sich um die Dienste eines iterativen Suchvorgangs, des Referenzierens von Bildregionen anhand von Regionenattributen und des Hinzufügens von Bildobjekten zu dem Bilddatenbestand.

Die Erkennung und die sich anschließende Weiterverarbeitung der natürlichen Interaktionen, wie sprachlicher Äußerungen und Gesten am Touchscreen-Display, wurden ebenso erläutert wie der sich daraus ergebende Aufbau der Client-Applikation. Hier wurde ein Schwerpunkt auf die Erkennung der Touchscreen-Gesten gelegt, da der dazu entwickelte Erkenner in den Systemteil der Bedienoberfläche integriert werden musste. Im Anschluß wurde eine Ablaufsteuerung, die das Herzstück der Applikation bildet, mit der dazu gehörenden Kommunikationsstruktur vorgestellt. Die Ablaufsteuerung sorgt für die entkoppelte Verarbeitung von Ereignissen der Erkenner und der Bedienoberfläche.

Die Erstellung von Entwicklungswerkzeugen für eine sichere Systempflege wurde in dieser Arbeit motiviert und die Funktion zweier erstellter Werkzeuge erläutert. Hierbei handelte es sich zum einen um das Konvertieren von internen und externen Datenrepräsentationen und zum anderen um die Vorverarbeitung der vom Spracherkenner erkannten Äußerungen für eine einfache Weiterverarbeitung. In beiden Fällen handelte es sich dabei um codeerzeugende Applikationen.

Für die Erläuterung der Frage, wie gut sich die natürlich geartete Interaktion für die Bedienung eines iterativen Bilddatenbanksystems eignet, wurde eine Systemevaluierung durchgeführt und im Rahmen dieser Arbeit erläutert. Die Ausführung umfasst sowohl die Konzeption des durchgeführten Akzeptanztests als auch die erzielten Ergebnisse. Bei der Evaluierung konnte gezeigt werden, dass sich die natürliche Interaktion ebenso gut wie die Standardbedienung mit der Maus für die Steuerung des entwickelten iterativen Bildsuchsystems eignet. Die ermutigende Tatsache, dass den Probanden die Interaktion mittels Sprache und Gestik deutlich mehr Spaß machte als die konventionelle Bedienung, wurde herausgestellt.

Abschließend kann festgestellt werden, dass die Umsetzung des hier vorgestellten Bildsuchsystems hervorragend gelungen ist. Die Abstimmung der natürlichen Interaktion mit den lernenden Suchtechniken, die bei der inhaltsbasierten Bildsuche eingesetzt wurden, ist gut gelungen,

so dass die natürliche, aber zur Zeit noch unkonventionelle Interaktion mittels Sprache und Gesten am Touchscreen-Display sich in vollem Maße als leistungsfähig herausgestellt hat und den Anwendern sehr gut gefällt.

Ausblick

Aufgrund der Reichhaltigkeit der in dieser Arbeit berücksichtigten Systemkomponenten sind die Themenbereiche, die im Folgenden für eine Anschlußarbeit vorgeschlagen werden, weit gestreut.

Ein Thema, das die Systemtechnik zum Inhalt hat, betrifft die Schnittstelle, die der Datenbank-Server dem Client zur Verfügung stellt. Diese Schnittstelle ist mit dem Einsatz von NDR ausschließlich für die Sicherheit des Betriebs ausgelegt. Für den Einsatz eines Bilddatenbank-Servers als Endanwendung fehlt jedoch die entscheidende Funktionalität, Meta-Daten, die den Bildern der Datenbank zugeordnet werden sollen, zu verwalten. Es wäre daher sinnvoll, die vorhandenen Datenbankdienste durch die Benutzung einer Anfragesprache zur Verfügung zu stellen. Hier können Multimedia-Anfragesprachen wie „Multimedia Retrieval Markup Language, MRML“ [Mül03] und „SQL Multimedia and Application Packages, SQL/MM“ [Mel01] für den Einsatz untersucht werden.

Die sprachliche Interaktion, die das INDI-System anbietet, ist zur Zeit nur unidirektional als Eingabekanal des Systems eingesetzt. Eine natürliche Kommunikation findet jedoch in einem Dialog statt. Durch den Einsatz eines Dialogs könnte die restriktive Auswertung der Äußerungen, die der Spracherkenner an die Applikation weiterleitet, gelockert werden. Durch entsprechende Nachfragen könnten Uneindeutigkeiten, die durch Fehler bei der Spracherkennung entstehen, geklärt werden. Wie hoch jedoch ein negativer Einfluss durch Sprachausgaben und einen weiterführenden Dialog wäre, weil beispielsweise die Wiederholung einer Äußerung weniger Zeit erforderte, als einen klärenden Dialog zu führen, müsste durch eine entsprechende Evaluierung beleuchtet werden.

Da sich die einfach gestaltete Interaktion bei der iterativen inhaltsbasierten Bildsuche durch Abgabe von Relevanzbewertungen als sehr leistungsfähiges Instrument herausgestellt hat, wäre es interessant, zu untersuchen, ob ähnlich einfach gehaltene Interaktionen durch Anwendung geeigneter Lernverfahren ebenfalls zu guten Suchergebnissen führen oder eventuell für die Erweiterung der Trainingsmenge der bereits eingesetzten Verfahren dienen können. Eine relative Bewertung zweier dem Anwender präsentierten Bilder, bei der der Benutzer beispielsweise auf die Frage: „Welches Bild gefällt Ihnen besser?“ antworten soll, könnten eine solche einfache Interaktion sein. Diese könnte sowohl durch den Einsatz von Gesten als auch mit einer sprachlichen Äußerung durchgeführt werden.

Anhang A

Evaluierung - Fragebogen

Der in Kapitel 7 vorgestellte Fragebogen ist auf den folgenden zwei Seiten abgebildet. Der Fragebogen ist in drei Teile aufgegliedert:

1. Allgemeine Fragen zur Person und deren eigene Einschätzung ihrer Vorkenntnisse
2. Der Fragenteil bestehend aus dreizehn Fragen, die per Kreuz auf einer Skala von Eins bis Fünf beantwortet werden sollten
3. Fragen über den persönlichen Eindruck des Systems

Liebe Teilnehmerin, lieber Teilnehmer,

zum Schluß der Sitzung habe ich noch einige Fragen an Dich, sie betreffen Deine Zufriedenheit mit dem Benutzermodus, den Du gerade getestet hast.
„Benutzermodus“ und „Modalität“ werden übrigens im gleichen Sinn verwandt und bezeichnen die jeweilige Bedienungsart (also Eingabe per Maus, Touchscreen oder Sprache).
Das Suchergebnis und die Bilddatenbank selbst sollten dabei möglichst wenig Deine Bewertung der Bedienung beeinflussen!

Zunächst einige Angaben zu Deiner Person:

Alter: _____ Geschlecht: m w

Beruf/Studiengang: _____

Wie viele Stunden in der Woche nutzt Du ungefähr den Computer? _____ Std.

Mit welchem Benutzermodus hast Du bereits **vor diesem Experiment** gearbeitet?

Maus □
Touchscreen □
Sprache □

Hast Du Erfahrung mit der Navigation in Bilddatenbanken? Ja □ Nein □

Bitte beantworte nun die folgenden Fragen, durch Ankreuzen der Skala:

	nein	eher nein		eher ja	ja
Der Benutzermodus ist einfach zu handhaben.	1 □	2 □	3 □	4 □	5 □
Es macht Spaß, über diese Modalität im Bildregister zu arbeiten.	1 □	2 □	3 □	4 □	5 □
Die Modalität war für mich immer ausreichend.	1 □	2 □	3 □	4 □	5 □
Die Bedienung erfordert viel Geduld.	1 □	2 □	3 □	4 □	5 □
Die Bedienung war mir unangenehm.	1 □	2 □	3 □	4 □	5 □
Ich habe mich über die Bedienung geärgert.	1 □	2 □	3 □	4 □	5 □

	nein	eher nein		eher ja	ja
Korrekturen sind schnell durchzuführen.	1 □	2 □	3 □	4 □	5 □
Die Bedienung ist schnell zu lernen.	1 □	2 □	3 □	4 □	5 □
Der Modus ist kompliziert zu bedienen.	1 □	2 □	3 □	4 □	5 □
Es ist interessant, mit diesem Modus zu arbeiten.	1 □	2 □	3 □	4 □	5 □
Die Anfrage konnte über diesen Modus effektiv bewältigt werden.	1 □	2 □	3 □	4 □	5 □
Die Modalität erfordert überflüssige Anweisungen.	1 □	2 □	3 □	4 □	5 □
Die Arbeit mit dieser Modalität kommt mir entgegen.	1 □	2 □	3 □	4 □	5 □

Ich hätte gerne eine andere bzw. weitere Modalität benutzt. Ja □ Nein □
Wenn ja, welche? Maus □
Touchscreen □
Sprache □

Eine Frage nur für Testpersonen in den Gruppen Maus/Sprache oder Touchscreen/Sprache:

Wie hoch schätzt Du subjektiv Deine anteilige Nutzung der beiden Modalitäten ein?

Ausschließlich Maus bzw. Touchscreen genutzt	1	2	3	4	5	6	7	Ausschließlich Sprache genutzt

Kritik, allgemeine Anmerkungen, was auch immer für uns interessant sein könnte:

Vielen Dank für Deine Antworten und die Teilnahme an diesem Experiment!

Anhang B

Evaluierung - Videodrehbuch

Für die Einführung der Testpersonen, die an dem in Abschnitt 7.1 beschriebenen Akzeptanztest teilnahmen, wurde ein Video produziert. Ziel dieses Videos war es, allen Testpersonen eine identischen Grundlagen für den sich anschließenden Test zu übermitteln.

Das auf den folgenden Seiten abgedruckte Drehbuch bildete die Grundlage für die Produktion des Videos und wurde den an der Produktion beteiligten Personen in dieser Form übergeben. Das Video sollte einen professionellen Eindruck erwecken, um die Testpersonen von der Ernsthaftigkeit des Experiments zu überzeugen. Das Video wurde im Medienlabor an der Universität Bielefeld professionell erstellt und geschnitten.

Inhaltsverzeichnis Drehbuch

Einleitende Worte für die Produktion

Dieses Video soll Testpersonen an eine Rechnerapplikation heranführen, so dass sie anschließend in der Lage sind, mit dieser Applikation zu arbeiten, genauer: damit einen Test durchzuführen. Da es sich nicht um ein Standardsystem mit einer Standardeingabe handelt, sind einige erklärende Worte und Beispiele nötig, obwohl die Applikation intuitiv bedienbar sein soll und natürliche Interaktionen zulässt.

Das Szenario des Videos ist starr. Es agieren ein Sprecher und ein Bediener. Das System wird der Kamera nur durch einen TFT-Bildschirm und ein Headset-Mikrofon, das auf der rechten Kopfhälfte ‚montiert' wird, präsentiert. Da es sich um ein System mit natürlicher Interaktion handelt, ist vor allem auch die Interaktion Inhalt dieses Videos. Die natürliche Interaktion wird in Form von Sprache und durch die Benutzung des Touchscreen-Displays durchgeführt. Um gleichzeitig sowohl Bediener als auch Display (einschl. Berührungen des Displays) zu sehen, ist ein Blick von hinten über den linken Arm des Benutzers auf das Display zu wählen. Da in Teilen des Videos nur die Bedienung mit der Maus gezeigt wird, ist der Display-Inhalt wichtig. Hier ist zu überlegen, ob das Display direkt aufgezeichnet wird (nicht mit der Kamera). Es sollte jedoch immer der richtige Ton darunter liegen. Da in bestimmten Fällen Einzelheiten auf dem Bildschirm wichtig sind, muss der Display-Inhalt wirklich groß sein. Es bietet sich an, hier mit einer zweiten Kamera zu arbeiten, wobei die erste auf den Bediener bzw. Bediener mit Monitor, die andere jedoch nur auf den Bildschirm gehalten wird.

Der Sprecher sollte frontal aufgenommen werden (wie in der Tagesschau).

Einleitung

Drehanweisungen	Sprechertext
Einleitend wird ein Sprecher gezeigt, der nett in die Kamera lächelt und vorträgt.	Hallo, ich begrüße Sie zu einem Akzeptanztest, der von der Arbeitsgruppe Angewandte Informatik der Technischen Fakultät durchgeführt wird. Sie werden sich vielleicht wundern, warum hier einleitende Worte mittels eines solch unpersönlichen Videos präsentiert werden. Wir wollen, dass alle Testpersonen exakt dieselben vorbereitenden Informationen für den anstehenden Test bekommen sollen, und das ist nur möglich, wenn das aus der Konserve vorgetragen wird. Ich möchte Ihnen jetzt das System INDI vorstellen, das zur Bildsuche in einem festen Bildbestand dient. Dieses System funktioniert nach dem Prinzip der Ähnlichkeitssuche und nicht etwa nach Stichworten, wie man das von anderen Systemen kennen mag. Für die Formulierung einer Suche mit INDI wird eines der Bilder der Datenbank ausgewählt, das eine gewisse Ähnlichkeit zu dem gesuchten Bild aufweist. Um an ein solches Bild zu gelangen, werden dem Benutzer eine Reihe zufällig bestimmter Bilder präsentiert. Das System hat neben der Hauptfunktion der Bildsuche aber noch andere Besonderheiten aufzuweisen. Es lässt sich nämlich nicht allein, wie gewohnt, mit der Maus bedienen. INDI versteht gesprochen Sprache und erkennt auch Gesten, die am Touchscreen durchgeführt werden, es ist also ein multimodales System. Die multimodale Bedienung steht im Mittelpunkt des hier durchgeführten Tests, wie später näher erläutert wird.

Bildsuche

Der Sprecher leitet dieses Kapitel ein. Zu diesem Zeitpunkt darf man neben dem Sprecher auch den Bediener sehen!	Zunächst möchte ich Sie jedoch mit der Bedienoberfläche des Systems und der Art der Suche vertraut machen. In dieser Phase möchte ich Ihnen die Bedienung mit der Maus allein demonstrieren, da diese Art der Bedienung jedem sofort geläufig ist.

Die Kamera schwenkt auf das Touchscreen-Display, auf dem die Applikation läuft. Davor sitzend ist ein Bediener zu sehen.	Hier sehen wir jetzt die Bedienoberfläche des Bildsuchsystems. Im linken Teil sehen Sie neun Bilder. Unter Zuhilfenahme des Sliders kann zu den weiteren Bildern gescrollt werden. Insgesamt beinhaltet die Oberfläche siebenundzwanzig Bilder.
Die entsprechenden Aktionen, die der Sprecher erwähnt, werden synchron vom Bediener durchgeführt. Hier sind zum Teil Details auf dem Screen wichtig so zum Beispiel die Rahmen der Regionen, auf die im Text Bezug genommen wird.	Wie bereits erwähnt, wird die Suche durch das Auswählen eines Beispielobjekts gestartet. Dies wird durch Doppelklicken mit der linken Maustaste auf das entsprechende Bild, das das Beispielobjekt beinhaltet, eingeleitet. Die Auswahl des Beispielbildes wird durch einen markierten Rahmen gekennzeichnet. Starten wir jetzt die Suche durch das Klicken des entsprechenden Buttons auf der rechten Seite, dann wird das ganze Bild als Suchobjekt verwendet. Wie Teile eines Bildes, Regionen, ausgewählt werden können, sehen wir gleich. Zunächst wollen wir uns jedoch das Suchergebnis betrachten. Das System hat nun alle Regionen der Datenbank entsprechend der Ähnlichkeit zu der gewählten Beispielregion geordnet. Die Bilder werden entsprechend der ähnlichsten enthaltenen Region bewertet. Die 27 ähnlichsten Bilder werden präsentiert. Oben links, im Bild 1, wird das beste unten rechts, Nummer 27, das schlechteste Bild plaziert. Welches die beste Region in den jeweiligen Bildern ist, wird durch eine Umrahmung angezeigt.
Es wird immer noch der Bediener gezeigt, der Sprecher erzählt im Hintergrund.	Anders als bei vielen anderen Systemen geht jetzt die Suche weiter. Das System verlangt jetzt vom Benutzer eine Bewertung der Regionen. Und das führen Sie, wie hier zu sehen, durch Klicken eines der Buttons unter dem Bild durch. Doppelplus wird gewählt, wenn es sich um eine sehr relevante Region handelt, Doppelminus, bei einer sehr irrelevanten Region. Möchte man eine Region nicht bewerten lässt man die initiale Stellung Null aktiviert. Wichtig sind hier zweierlei Dinge: (1) Durch die so abgegebene Bewertung wird die markierte Region bewertet und (2) die Suche mit positiven Bewertungen kann eher zum Erfolg geführt werden. Das entspricht den menschlichen Beschreibungen, bei denen eher charakteristische Gegebenheiten genannt werden als nicht zutreffende Attribute. Nach Abgabe der gewünschten Bewertungen wird ein neuer Suchschritt gestartet.

... Fortsetzung

Um alle Regionen bewerten oder eine Suche mit einer bestimmten Region eines Bildes starten zu können, wechselt man durch Klicken der rechten Maustaste auf dem entsprechenden Bild in die Großansicht eines Bildes. Es werden jetzt alle Regionen des Bildes angezeigt. Durch Klicken in die entsprechende Region des Bildes lässt sich eine Bildregion selektieren. Befinden sich an der Mausposition mehrere Regionen, muss evtl. mehrfach geklickt werden.

Für alle Regionen eines Bildes kann so eine Bewertung abgegeben werden. Eine besondere Stellung nimmt in diesem Zusammenhang die Beispielregion ein. Die Region, die beim Verlassen der Großansicht mit dem OK-Button selektiert ist, wird zur neuen Beispielregion. Damit beginnt also eine neue Suche.

Interaktion in den Modalitäten

Einleitend wird wieder der Sprecher gezeigt.

Die Mensch-Maschine-Kommunikation bietet ein großes Forschungsspektrum und hat schon sehr viele Forscher beschäftigt. Die Standard-Interaktionen mittels Tastatur und Maus sind aus technischen Gründen entstanden und werden inzwischen auch akzeptiert. Es ist jedoch generell angenehmer, ohne technische Hilfsmittel zu kommunizieren. Deshalb wurde in das System, wie einleitend bereits erwähnt, Sprach- und Gestikerkennung zur Interaktion integriert. Man darf diese Fähigkeiten jedoch nicht mit denen eines Menschen gleichsetzen. Das System versteht nur das, worauf es programmiert ist. Das sind bestimmte Wörter und Sätze und ein Repertoire von Gesten.

Wird also beispielsweise ein Satz korrekt verstanden, so kann das bei einem Satz, bei dem nur ein Wort ausgetauscht wurde oder die Satzstellung unter Benutzung der selben Wörter verändert wurde, nicht mehr der Fall sein. Wird man vom System nicht verstanden, obwohl der richtige Satz gesagt wurde, dann muss er **normalgesprochen** wiederholt werden.

Sprache: Gezeigt wird jetzt wieder das System mit dem Benutzer. Bedient wird ausschließlich mit Sprache. Wichtig ist hier, dass vom Sprecher jeweils nur kurz erläutert wird (auch nur wenn nötig), was der Benutzer mit sprachlichen Instruktion erreichen will.	S: Wir führen jetzt eine Suche aus und benutzen hierbei lediglich die Sprachsteuerung. Zunächst wird ein Beispielbild ausgewählt und der erste Suchschritt durchgeführt. B: Zeige weitere Bilder B: Scrolle nach unten B: Scrolle nach oben B: Nimm das Bild in der Mitte B: Starte die Suche S: Jetzt werden Bewertungen durchgeführt B: Bild zwei ist sehr gut B: Die Region des Bildes in der Mitte ist gut B: Das Bild oben rechts ist gut B: Scrolle nach unten B: Das rechte Bild in der unteren Reihe ist sehr gut
Die Dialoge des Benutzers sind Beispiele und müssen auf die Applikation angepasst sein!	S: Jetzt wird das Bild vergrößert, um alle anderen interessanten Bildregionen zu sehen. B: Zeige das Bild unten rechts B: Nimm die helle Region B: Die Region ist gut B: Die große, grüne Region ist sehr gut B: Schließe das Fenster B: Starte die Suche
Touchscreen: Zunächst ist der Sprecher im Bild, der ziemlich gestikulierend klarmacht, dass Gesten ein wirklich natürlicher Weg zur Kommunikation ist.	Zusätzlich zur gewohnten Bedienung eines Touchscreens ist das System in der Lage, Gesten, die auf der Display-Fläche durchgeführt werden, zu erkennen. Das sind Gesten, wie sie vom Ausfüllen von Formularen wie zum Beispiel von Lottoscheinen bekannt sind.

Jetzt überblenden auf Schirm mit Bediener. Die Gesten werden synchron zu dem Kommentar des Sprechers durchgeführt.	Schauen wir uns die Bedienung, die ausschließlich mithilfe des Touchscreens durchgeführt wird, an. Zur Auswahl des Beispielbildes wird ein 'X' oder ein Haken in das entsprechende Bild gezeichnet. Durch das Ziehen des Sliders wird nach unten bzw. nach oben gescrollt. Jetzt wird das Beispielbild vergrößert, um eine bestimmte Region des Bildes auszuwählen. Dazu zeichnet der Benutzer einen senkrechten Strich nach oben in dem entsprechenden Bild. Durch Antippen der gewünschten Region wird diese selektiert. Das Fenster wird geschlossen und die Suche gestartet. Jetzt führt der Benutzer Bewertungen durch das Antippen der Bewertungsbuttons unter den Bildern aus und startet eine weitere Suche.
Kombination Touch/Speech Der Sprecher ist wieder im Bild und führt vor, was er sagt.	Der Mensch verwendet die unterschiedlichen Modalitäten dort, wo sie am besten einzusetzen und unkompliziert ist. So sagt man: „Nimm **den** blauen Eimer dort mit!"Diese Aufforderung ließe sich auch ausschließlich per Sprache geben, das wäre jedoch deutlich komplizierter! Es lassen sich alle bildbezogene Aktionen mittels Touchscreen und Sprachanweisung durchführen. Dabei wird die Aktion durch die Sprache festgelegt und das Bild durch Berühren des Displays an der entsprechenden Stelle referenziert.
Die Kamera zeigt wieder den Bediener und sein Touchscreen.	S: Schauen wir uns das im Beispiel an. B: Nimm dieses Bild. B: Starte die Suche B: Dieses Bild ist sehr gut B: Dieses Bild ist gut B: Zeige dieses Bild B: die kleine Region ist sehr gut B: Schließe das Fenster B: Starte die Suche

Der Test

Hier wird hauptsächlich der Sprecher gezeigt. Unterstützend wird ein-, zweimal der Bildschirm gezeigt.	Sie haben jetzt die unterschiedlichen Bedienungsmöglichkeiten des Bildsuchsystems INDI kennengelernt. In dem Test, den wir unter anderem mit Ihnen durchführen möchten, wollen wir die Bedienung und Benutzbarkeit des Systems in den unterschiedlichen Interaktionsmodalitäten bzw. deren Kombinationen gegenüberstellen. Wir werden hier eine sog. Zielsuche durchführen, bei der Sie ein in der Datenbank gespeichertes Bild gezeigt bekommen, das Sie mit dem Bildsuchsystem wiederfinden sollen. Für eine solche Suchaufgabe bekommen Sie drei Minuten Zeit. Wurde das Bild innerhalb der drei Minuten gefunden, d.h. das Bild befindet sich unter den 27 angezeigten Bildern, dann ist die Suchaufgabe beendet. Insgesamt sollen drei solcher Zielsuchaufgaben durchgeführt werden. Das Zielbild können Sie während der Suche, wie hier gezeigt, die ganze Zeit über betrachten. Es ist nicht entscheidend bzw. schlimm, wenn Sie das Bild nicht innerhalb der zugestandenen Zeit finden, denn bei diesem Test geht es nur darum, die Bedienung in den unterschiedlichen Modalitäten zu testen. Die Suche sollte jedoch zielgerecht durchgeführt werden und möglichst auch zum Erfolg führen!
Der Sprecher wird gezeigt, der einen Fragebogen in die Hand genommen hat	Im Anschluss an den praktischen Test, möchten wir Sie bitten, einen Fragebogen auszufüllen, der Fragen über das gerade durchgeführte Arbeiten mit dem System enthält. Es sei noch einmal betont, dass nicht das Suchergebnis des Systems bewertet werden soll, sondern die Bedienung eines solchen Systems in unterschiedlichen Modalitäten. Das Team der Arbeitsgruppe möchte sich an dieser Stelle schon einmal für die Zusammenarbeit mit Ihnen bedanken und wünscht Ihnen guten Sucherfolg!

Literatur

[Aho85] A. V. Aho, R. Sethi, J. D. Ullman: *Compilers: Principles, Techniques, and Tools*, Addison/ Wesley, 1985.

[Bau03] C. Bauckhage, T. Käster, M. Pfeiffer, G. Sagerer: *Content-Based Image Retrieval by Multimodal Interaction*, in *Proc. of the 29th Annual Conference of the IEEE Industrial Electronics Society*, 2003, S. 1865–1870.

[Ben79] J. L. Bentley, T. Ottmann: *Algorithms for Reporting and Counting Geometric Intersections.*, *IEEE Trans. Computers*, Bd. 28, Nr. 9, 1979, S. 643–647.

[Bra00] S. Brandt, J. Laaksonen, E. Oja: *Statistical Shape Features in Content-Based Image Retrieval*, in *Proc. of IEEE Internatinal Conference on Pattern Recognition*, Bd. 2, Barcelona, Spain, Sep. 2000, S. 1062–1065.

[Cas98] M. L. Cascia, S. Sethi, S. Sclaroff: *Combining Textual and Visual Cues for Content-Based Image Retrieval on the World Wide Web*, in *Proc. of IEEE Workshop on Content-based Access of Image and Video Libraries*, Santa Barbara, CA, Juni 1998, S. 24–28.

[Coe98] M. H. Coen: *Design principles for intelligent environments*, in *Proc. of the fifteenth national/tenth conference on Artificial intelligence/Innovative applications of artificial intelligence*, American Association for Artificial Intelligence, Menlo Park, CA, USA, 1998, S. 547–554.

[Cox00] I. J. Cox, M. L. Miller, T. P. Minka, T. Papathomas, P. N. Yianilos: *The Bayesian Image Retrieval System, PicHunter: Theory, Implementation and Psychophysical Experiments*, *IEEE Transactions on Image Processing*, Bd. 9, Nr. 1, 2000.

[Dat05] R. Datta, J. Li, J. Z. Wang: *Content-based image retrieval: approaches and trends of the new age*, in *Proc. of the 7th ACM SIGMM international workshop on Multimedia information retrieval*, ACM Press, 2005, S. 253–262.

[Fin99] G. A. Fink: *Developing HMM-based Recognizers with ESMERALDA*, in V. Matoušek, P. Mautner, J. Ocelíková, P. Sojka (Hrsg.): *Lecture Notes in Artificial Intelligence*, Bd. 1692, Springer, Berlin Heidelberg, 1999, S. 229–234.

[Fli95] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele, P. Yanker: *Query by Image and Video Content: The QBIC System*, *Computer*, Bd. 28, Nr. 9, 1995, S. 23–32.

[Hai94] E. Haines: *Point in polygon strategies*, in *Graphics gems IV*, Academic Press Professional, Inc., 1994, S. 24–46.

[Hua96] T. S. Huang, S. Mehrotra, K. Ramchandran: *Multimedia Analysis and Retrieval System (MARS) Project*, in *Proc. of the 33rd Annual Clinic on Library Application of Data Processing - Digital Image Access and Retrieval*, März 1996, S. 0–.

[Hua02] T. Huang, X. Zhou, M. Nakazato, I. Cohen, Y. Wu: *Learning in Content-Based Image Retrieval*, in *Proc. of 2nd International Conference on Development and Learning*, Cambridge, MA, Juni 2002, S. 155–164.

[Jun98] N. Jungclaus: *Integration verteilter Systeme zur Mensch-Maschine-Kommunikation*, Dissertation, Universität Bielefeld, Technische Fakultät, 1998.

[Käm02] T. Kämpfe, T. Käster, M. Pfeiffer, H. Ritter, G. Sagerer: *INDI – Intelligent Database Navigation by Interactive and Intuitive Content-Based Image Retrieval*, in *Proc. IEEE International Conference on Image Processing*, Bd. III, 2002, S. 921–924.

[Käs01] T. Käster: *Konzeption und Implementierung eines SQL-Datenbank-Backends zur Speicherung von Multimediadaten*, Diplomarbeit, Universität Bielefeld, Technische Fakultät, 2001.

[Käs03] T. Käster, M. Pfeiffer, C. Bauckhage, G. Sagerer: *Combining Speech and Haptics for Intuitive and Efficient Navigation through Image Databases*, in *Proc. International Conference on Multimodal Interfaces*, 2003, S. 180–187.

[Käs05] T. Käster: *Intelligente Bildersuche durch den Einsatz inhaltsbasierter Techniken*, Dissertation, Universität Bielefeld, Technische Fakultät, 2005.

[Laa00] J. Laaksonen, M. Koskela, S. Laakso, E. Oja: *PicSOM – Content-Based Image Retrieval with Self-Organizing Maps*, *Pattern Recognition Letters*, Bd. 21, Nr. 13-14, Dez. 2000, S. 1199–1207.

[Laa01] J. Laaksonen, M. Koskela, S. Laakso, E. Oja: *Self-Organising Maps as a Relevance Feedback Technique in Content-Based Image Retrieval*, *Pattern Analysis and Applications*, Bd. 4, Nr. 2-3, Juni 2001, S. 140–152.

[Lan95] S. Lang, P. Lockemann: *Datenbankeinsatz*, Springer-Verlag, Berlin Heidelberg New York, 1995.

[Lan03] S. Lankes: *Konzeption und Umsetzung einer echtzeitfähigen Verteilungsplattform für eingebettete Systeme*, Shaker Verlag Aachen, 2003.

[Mel01] J. Melton, A. Eisenberg: *SQL multimedia and application packages (SQL/MM)*, *SIGMOD Rec.*, Bd. 30, Nr. 4, 2001, S. 97–102.

[Mül03] H. Müller, A. Geissbuhler, S. Marchand-Maillet: *Extension to the Multimedia Retrieval Markup Language: A communication protocol for content-based image retrieval*, September 2003.

[Nie83] H. Niemann: *Klassifikation von Mustern*, Springer-Verlag, Berlin, 1983.

[Ovi99] S. Oviatt: *Ten myths of multimodal interaction*, *Communications of the ACM*, Bd. 41, Nr. 11, 1999, S. 77–81.

[Plö02] T. Plötz, G. Fink: *Robust Time-Synchronous Environmental Adaptation for Continuous Speech Recognition Systems*, in *Proc. ICSLP*, Bd. II, 2002, S. 1409–1412.

[Por99] K. Porkaew, M. Ortega, S. Mehrotra: *Query Reformulation for Content Based Multimedia Retrieval in MARS.*, in *International Conference on Multimedia Computing and Systems*, Bd. II, IEEE, 1999, S. 747–751.

[Pre02] J. J. Preece, Y. Rogers, H. Sharp: *Interaction Design: beyond human-computer interaction*, John Wiley & Sons, New York, 2002.

[Roc71] J. Rocchio: *The SMART retrieval system*, Prentice-Hall, 1971.

[Rui97] Y. Rui, T. Huang, S. Mehrotra: *Content-Based Image Retrieval With Relevance Feedback in MARS*, in *IEEE Intl. Conf. on ICIP'97, Santa Barbara, CA*, 1997.

[Rui98] Y. Rui, T. Huang: *Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval*, *IEEE Trans. on Circuits and Video Tech., Special Issue on Segmentation Description, and Retrieval of Video Content*, Bd. 8, Nr. 5, 1998.

[Rui99a] Y. Rui, T. Huang: *A Novel Relevance Feedback Technique in Image Retrieval*, ACM Multimedia, 1999.

[Rui99b] Y. Rui, T. Huang, S. Chang: *Image retrieval: current techniques, promising directions and open issues*, *Journal of Visual Communication and Image Representation*, Bd. 10, Nr. 4, April 1999, S. 39–62.

[Rui00] Y. Rui, T. Huang: *Optimizing Learning in Image Retrieval*, in *IEEE Int'l. Conf. on Computer Vision and Pattern Recognition*, Hilton Head, SC, USA, June 2000.

[Rui01] Y. Rui, T. Huang: *Relevance Feedback Techniques in Image Retrieval*, in M. Lew (Hrsg.): *Principles of Visual Information Retrieval*, Kap. 9, Springer-Verlag, 2001, S. 221–258.

[Seb01] N. Sebe, M. S. Lew: *Texture Features for Content-Based Retrieval*, in M. S. Lew (Hrsg.): *Principles of Visual Information Retrieval*, Kap. 3, Springer-Verlag, London, 2001, S. 51–85.

[Sie99] J. Siegel: *CORBA 3 Fundamentals and Programming*, John Wiley & Sons, Inc., New York, NY, USA, 1999.

[Sme00] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain: *Content based image retrieval at the end of the early years*, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Bd. 22, Nr. 12, 2000, S. 1349–1380.

[Ste98] W. R. Stevens: *UNIX Network Programming Volume 1 Networking APIs: Sockets and XTI*, Prentice-Hall, 2. Ausg., 1998.

[Ste99] W. R. Stevens: *UNIX Network Programming Volume 2 Interprocess Communications*, Prentice-Hall, 2. Ausg., 1999.

[Str95] M. Stricker, M. Orengo: *Similarity of Color Images*, in *Proc. of Storage and Retrieval for Image and Video Databases (SPIE)*, 1995, S. 381–392.

[Swa91] M. Swain, D. Ballard: *Color indexing*, *International Journal of Computer Vision*, Bd. 7, Nr. 1, 1991, S. 11–32.

[Wac98] S. Wachsmuth, G. A. Fink, G. Sagerer: *Integration of Parsing and Incremental Speech Recognition*, in *Proceedings of the European Signal Processing Conference*, Bd. 1, Rhodes, Sep. 1998, S. 371–375.

[Wac02] S. Wachsmuth, G. Sagerer: *Bayesian Networks for Speech and Image Integration*, in *Proc. of 18th National Conference on Artificial Intelligence*, 2002, S. 300–306.

[Zei96] E. Zeidler (Hrsg.): *Teubner – Taschenbuch der Mathematik*, B.G. Teubner, 1996.

[Zho03] X. Zhou, T. Huang: *Relevance Feedback in Image Retrieval: A Comprehensive Review*, *Multimedia Systems*, Bd. 8, Nr. 6, April 2003, S. 536–544.