

Auswertung von 2D und 3D unstrukturierten Daten für die Objekt- und Lageerkennung

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

der Fakultät für Informatik
der Universität Fridericiana zu Karlsruhe (TH)

genehmigte

D i s s e r t a t i o n

von

Dipl.-Ing. Kirill Safronov

aus Ufa, Russland

Tag der mündlichen Prüfung: 8. Juli 2009

Erster Gutachter: Prof. Dr.-Ing. Heinz Wörn

Zweiter Gutachter: Prof. Dr.-Ing. Jürgen Beyerer

Danksagung

Mein besonderer Dank gilt Herrn Prof. Dr.-Ing. Heinz Wörn für die Betreuung und stetige Unterstützung meiner wissenschaftlichen Tätigkeit am Institut für Prozeßrechentechnik, Automation und Robotik (IPR) der Universität Karlsruhe (TH). Ebenso danke ich Herrn Prof. Dr.-Ing. Jürgen Beyerer für die Übernahme des Korreferats und die sachliche und konstruktive Kritik an meiner Arbeit.

Weiterhin danke ich Herrn Dr.-Ing. Igor Tchouchenkov, Leiter der Gruppe „Steuerung und Regelung von Robotersystemen“ am IPR für zahlreiche wertvolle Hinweise und anregende Diskussionen.

Herrn Prof. Dr.-Ing. Vladimir Vasilyev von der Staatlichen Technischen Luftfahrtuniversität Ufa (Russland) möchte ich besonders für die hilfreiche Anmerkungen und Hinweise zu meiner Dissertation danken.

Allen Kollegen des Institutes bedanke ich mich für die gute Zusammenarbeit. Besonders danken möchte ich Daniel Braun und Stephan Irgenfried.

Schließlich danke ich meiner Frau Nadia, meiner Tochter Ksenia und meiner ganzen Familie für das Verständnis und die Unterstützung während meiner Zeit als Doktorand.

Karlsruhe, im Juli 2009

Kirill Safronov

Inhaltsverzeichnis

1	Einleitung	9
1.1	Motivation und Problembeschreibung	9
1.2	Zielsetzung	10
1.3	Ergebnisse	11
1.4	Gliederung der Arbeit	12
2	Stand der Forschung	13
2.1	Problemstellung der Objekt- und Lageerkennung	13
2.2	Methoden zur Objekt- und Lageerkennung	16
2.2.1	Klassifikation der Methoden zur Objekt- und Lageerkennung	16
2.2.2	Musterbasierte Methoden	16
2.2.3	Objektbasierte Methoden	23
2.2.4	Synthetische Methoden	28
2.2.5	Vergleich der Methoden für die Objekt- und Lageerkennung	29
2.3	Fazit und Zielsetzung	34
3	Hierarchisch-iterative Methode für die Objekt- und Lageerkennung	35
3.1	Entwicklung der hierarchisch-iterativen Methode für die Objekt- und Lageerkennung	35
3.1.1	Aufgabenstellung der Objekt- und Lageerkennung	35
3.1.2	Prinzipien der hierarchischen iterativen Methode für die Objekt- und Lageerkennung	35
3.2	Algorithmus der hierarchischen iterativen Methode für die Objekt- und Lageerkennung	36
3.3	Algorithmen der hierarchischen iterativen Methode für die Objekt- und Lageerkennung	43
3.3.1	Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen	44

3.3.2	Hierarchisches Matching	52
3.3.3	Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten	54
3.3.4	Algorithmus der Deformation des Objektprototyps	58
3.3.5	Algorithmus des iterativen Matchings	61
3.3.6	Algorithmus der Hierarchisierung von Objektprototypen	63
3.3.7	Analyse der entwickelten Algorithmen	66
3.3.8	Fazit	67
4	Systeme für die Objekt- und Lageerkennung	69
4.1	Entwicklungsverfahren von Systemen für die Objekt- und Lageerkennung .	69
4.2	Entwicklung des Lageerkennungssystems für die Registrierung medizinischer Bilddatensätze	72
4.2.1	Problemstellung	72
4.2.2	Architektur des Lageerkennungssystems	73
4.2.3	Experimentelle Ergebnisse	81
4.3	Entwicklung des Objekterkennungssystems für die Schrifterkennung	85
4.3.1	Problemstellung	85
4.3.2	Architektur des Lageerkennungssystems	86
4.3.3	Experimentelle Ergebnisse	93
4.4	Entwicklung des Objekterkennungssystems für das Griff-in-die-Kiste-Problem	95
4.4.1	Problemstellung	95
4.4.2	Architektur des Objekt- und Lageerkennungssystems	97
4.4.3	Zusammenfassung	106
5	Entwicklung und Untersuchung von HIMOLE-basierten Software-Systemen	109
5.1	Softwaresystem für die automatisierte OP-Planung	111
5.2	Softwaresystem für die Qualitätssicherung bei der Herstellung von Piezo-Motoren	111
5.3	Softwaresystem für das Griff-in-die-Kiste-Problem	112
5.4	Zusammenfassung	113
6	Zusammenfassung und Ausblick	115
A	Experimentelle Ergebnisse	119
B	Abkürzungsverzeichnis	125

C Symbolverzeichnis	127
Abbildungsverzeichnis	130
Tabellenverzeichnis	135
Literaturverzeichnis	136

Kapitel 1

Einleitung

1.1 Motivation und Problembeschreibung

Eine der wichtigsten Aufgaben, die der Mensch im Alltag ständig löst, ist die Erkennung und Klassifikation von verschiedenen Objekten. Diese Aufgabe wird vom Menschen sehr effizient sowohl im Alltagsleben (beim Lesen, während der Unterhaltung) als auch am Arbeitsplatz (bei der visuellen Qualitätssicherung, bei der Suche und Identifikation von Objekten in den Bildern) gelöst. Dabei ist der Mensch in der Lage, nicht nur konkrete, sondern auch abstrakte Muster zu erkennen (z.B. logische Zusammenhänge). Deshalb ist die Entwicklung von effizienten, automatischen Systemen zur Erkennung von Mustern und Objekten, die an die menschlichen Fähigkeiten herankommen und diese Fähigkeiten übertreffen können, die Hauptaufgabe im Bereich des maschinellen Sehens.

Die Objekterkennung ist seit langem Gegenstand der Forschung. Schon in den 60er Jahren wurden erste Ansätze zur Lösung dieses Problems diskutiert. Viele Bücher und Veröffentlichungen, die die Thematik der Objekterkennung im Vordergrund haben, wurden publiziert ([30, 31, 28, 27, 1]). Eine Reihe von Methoden und Systemen zur Objekterkennung wurden entwickelt und in verschiedenen Bereichen eingesetzt.

Trotz des hohen Niveaus der Forschung im Bereich der Objekterkennung, bleibt die Aufgabe der Entwicklung von effizienten Systemen zur Objekterkennung kompliziert. Es gibt eine Reihe von praktischen Aufgaben, die mit den existierenden Verfahren und Methoden nicht effizient gelöst werden können. Heutzutage ist das Ziel der Forschung im Bereich der Objekterkennung die Lösung von konkreten Aufgaben. Dies führt jedoch zu folgenden Problemen: die existierenden Systeme sind von den Daten und zu erkennenden Objekten stark abhängig. Nach der Veränderung der Daten oder Objektklassen ist es notwendig, das Erkennungssystem, soweit möglich, den geänderten Bedingungen anzupassen, falls dies möglich ist, sowie das Lernen des Systems mit neuen Lernstichproben durchzuführen.

Eine weitere Schwachstelle der existierenden Methoden und Systeme zur Objekterkennung ist die eingeschränkte Fähigkeit der gleichzeitigen Objekt- und Lageerkennung (Orientierung und Position in der Szene).

Ein Ansatz, die oben genannten Probleme zu lösen, ist die Entwicklung einer neuen Methode zur Objekt- und Lageerkennung, die auf der Kombination eines Matching-Verfahrens

und einer Reihe von Optimierungsmethoden basiert. Das Prinzip dieses Verfahrens besteht in der Vereinigung von verschiedenen Matching-Methoden unter Verwendung eines allgemeinen, mathematischen Modells der Objekterkennung aufgrund der Auswertung von unstrukturierten 2D und 3D Daten, das von der Natur der Daten und zu erkennenden Objekten unabhängig ist.

Dieses Verfahren erfordert die Lösung folgender Probleme: Optimierung des Lernprozesses (Dauer und Komplexität), Vereinfachung der Merkmalsextraktion, Beschleunigung des Matching-Vorgangs und Erkennung von Objekten in den verrauschten und verzerrten Eingabedaten. Gegenstand dieser Dissertation ist die Entwicklung einer solchen Methode zur Objekt- und Lageerkennung.

1.2 Zielsetzung

Die ausgeführte Analyse von verschiedenen Methoden hat gezeigt, dass heutzutage kein universelles Verfahren für die Lösung der Objekt- und Lageerkennung existiert, das eine beliebige Aufgabe unter Verwendung von gleichbleibenden Klassifikatoren lösen kann. Außerdem wurde es festgestellt, dass jede Methode eine Reihe von Nachteilen hat, die ihre Anwendungsbereiche beschränkt, z.B. eingeschränkte Fähigkeit zur gleichzeitigen Objekt- und Lagererkennung in 2D- oder 3D-Daten.

Das Ziel der hier vorgestellten Arbeit ist die Erhöhung von Geschwindigkeit, Genauigkeit und Zuverlässigkeit der Objekt- und Lagererkennung durch Auswertung von 2D und 3D unstrukturierten Daten mit der Hilfe einer neuen, hierarchisch-iterativen Methode. Im Kontext dieser Dissertation lässt sich der Begriff der „unstrukturierten Daten“ erklären als die digitale Repräsentation einer realen Szene, z.B. als Folge von Bilderfassung und anschließender Abtastung. Im einfachsten Falle handelt es sich um eine Menge von Binär- oder Grauwerten. Der Zusatz „unstrukturiert“ unterstreicht, dass diese Daten keine zusätzlichen Informationen (Metadaten), über die Semantik der Szene (z.B. Objekte und deren Lage oder den Kontext) enthalten, die zur Unterstützung der Objekterkennung verwendet werden könnten.

Um dieses Ziel zu erreichen, sind folgende Aufgaben zu lösen:

1. Entwicklung einer Methode, die auf der Auswertung von 2D und 3D unstrukturierten Daten basiert und gleichzeitige Objekt- und Lageerkennung durchführen kann
2. Entwicklung von Algorithmen und Prozeduren, welche die vorgeschlagene Methode realisieren
3. Entwurf der Architektur eines universellen Systems zur Objekt- und Lageerkennung, die auf der vorgeschlagenen Methode basiert, und Erarbeiten eines Realisierungsvorschlags
4. Entwicklung und Implementierung der Software, die die vorgeschlagene Methode realisiert
5. Bewertung der entwickelten Systeme bei der Lösung praktischer Aufgaben durchführen

1.3 Ergebnisse

Im Rahmen der hier vorgestellten Arbeit wurden folgende Ergebnisse erzielt:

1. Eine Methode zur Auswertung von 2D und 3D unstrukturierten Daten für die Objekt- und Lageerkennung wird vorgeschlagen. Die Neuheit dieser Methode besteht in einer Kombination von Matching-Verfahren mit Optimierungsalgorithmen, sowie in der Interpretation des Matching-Verfahrens, die im Auffinden und Lokalisieren eines vorgegebenen Objektes in 2D und 3D Daten besteht. Die entwickelte Methode besitzt eine hohe Rauschunempfindlichkeit und hängt nicht von Datenarten und den zu erkennenden Objekten ab. Sie erhöht nicht nur wesentlich die Wahrscheinlichkeit und Geschwindigkeit der korrekten Objekterkennung, sondern auch die Genauigkeit der Lage- und Positionsbestimmung des Objektes. Das ermöglicht die Entwicklung von Objekterkennungssystemen, die in der Lage sind, viele praktische Aufgaben, wie Registrierung medizinischer Bilddatensätze, industrielle Schrifterkennung oder Objekt- und Lagererkennung für die Griff-in-die-Kiste-Anwendung, zu lösen.
2. Ein System von Algorithmen und Prozeduren, die die Bestandteile der vorgeschlagenen Methode sind und die Erhöhung der Genauigkeit und Geschwindigkeit bei der Erkennung von Objekten und ihrer Lage ermöglichen, wurde entwickelt. Dieses System enthält folgende Algorithmen: Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen, Algorithmus des hierarchischen Matchings, Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten, Algorithmus der Deformation des Objektprototyps, Algorithmus des iterativen Matchings, Algorithmus der Hierarchisierung von Objektprototypen
3. Die Architektur des universellen Systems zur Objekterkennung, die auf der vorgeschlagenen hierarchisch-iterativen Methode basiert, sowie die Methodik der Verwendung dieses Systems bei der Lösung verschiedener Aufgaben wurden entwickelt.
4. Die Software, welche die vorgeschlagene hierarchisch-iterative Methode realisiert, wurde in C++ unter Verwendung der Bildverarbeitungsbibliotheken OpenCV und VTK implementiert. Mit Hilfe dieser Implementierung wurde eine Reihe von praktischen Aufgaben im Rahmen von Automatisierungssystemen gelöst.
5. Auf Basis der entwickelten Architekturen wurden Systeme zur Objekt- und Lageerkennung entwickelt, mit denen viele Experimente durchgeführt wurden, die die Effizienz der Verwendung der hierarchisch-iterativen Methode gezeigt haben:
 - Registrierung medizinischer Bilddatensätze: ein typisches herkömmliches Matching-Verfahren, probabilistisches Matching und hierarchisch-iterative Methode wurden verglichen. Die Experimente haben gezeigt, dass der Einsatz der hierarchisch-iterativen Methode im Vergleich zum klassischen Matching-Verfahren und probabilistischen Matching ermöglicht hat, die Erkennungszeit um den Faktor 7,2 bzw. 1,15, den Fehler der Positionsbestimmung um den Faktor 6,4 bzw. 2,6, den Fehler der Orientierungsbestimmung um den Faktor 3,6 bzw. 1,4 zu reduzieren, sowie die Anzahl der korrekten Erkennungen um den Faktor 2,3 bzw. 1,3 zu erhöhen.

- **Schrifterkennung:** aktuelle Software-Systeme für die Schrifterkennung (TOCR, SimpleOCR, etc.) und das hier vorgeschlagene System wurden verglichen. Die Experimente haben gezeigt, dass ein Einsatz des Systems, das auf der hierarchisch-iterativen Methode basiert, ermöglicht, die Anzahl der korrekten Erkennungen um den Faktor 1,8, was 99,7% aller bearbeiteten Bilder entspricht, im Vergleich zu den *TOCR*-Ergebnissen zu erhöhen.
- **Griff-in-die-Kiste-System:** ein typisches herkömmliches Matching-Verfahren und das vorgeschlagene System wurden verglichen. Die Experimente haben gezeigt, dass der Einsatz von der hierarchischen iterativen Methode hat im Vergleich zu dem klassischen Matching-Verfahren ermöglicht, die Erkennungszeit um den Faktor 7,2 zu reduzieren, den Fehler der Positions- und Orientierungsbestimmung um den Faktor 4,4 bzw. 3,8, sowie die Anzahl der korrekten Erkennungen um den Faktor 2,2 zu erhöhen.

1.4 Gliederung der Arbeit

Die Ausarbeitung ist entsprechend der Aufgabenstellung strukturiert und gliedert sich in insgesamt sechs Kapitel. Während im einleitenden ersten Kapitel das Promotionsthema und die verfolgten Ziele dargestellt werden, wird im Kapitel zwei auf den Stand der Forschung näher eingegangen. In den Kapiteln drei bis fünf wird die eigentliche Arbeit behandelt. In Kapitel drei werden die einzelnen Prinzipien und Algorithmen der entwickelten hierarchisch-iterativen Methode für die Objekt- und Lageerkennung detailliert erläutert. In Kapitel vier werden die Architektur und die Methodik der Verwendung eines Systems zur Objekt- und Lageerkennung, das auf der vorgeschlagenen Methode basiert, vorgestellt und die Architektur der Systeme, zur Lösung folgender Aufgaben: Registrierung medizinischer Bilddatensätze, Schrifterkennung und Griff-in-die-Kiste. Im Kapitel fünf wird die Software, die die vorgeschlagene hierarchische iterative Methode realisiert, vorgestellt und Kapitel sechs gibt schließlich eine Zusammenfassung der Arbeit und einen Ausblick.

Kapitel 2

Stand der Forschung

2.1 Problemstellung der Objekt- und Lageerkennung

In allgemeinen Fall besteht das Problem der Objekterkennung in der Klassifizierung eines Objektes unter Verwendung von verschiedenen Merkmalen, die bei der Analyse von Daten extrahiert werden, d.h. die Zuordnung von Objekten, die einen hohen Ähnlichkeitsgrad haben, zu einer Klasse.

Das Problem der Objekterkennung kann in zwei Fälle unterteilt werden ([97]):

- Erkennung eines einzelnen Objektes, d.h. die Zuordnung dieses Objektes zu einer Klasse.
- Erkennung mehrerer Objekte, d.h. die Lokalisierung von Objekten in der Szene und die Klassifizierung dieser Objekte. Diesen Prozess nennt man auch Szeneanalyse.

Die Aufgaben des zweiten Falls sind mathematisch komplizierter als die Aufgaben der ersten Klasse. Um diese Aufgabe zu lösen, werden rechenintensive Algorithmen benötigt.

Die Vorgehensweise eines Algorithmus für die Aufgaben der ersten Klassen ist in der Abbildung 2.1 dargestellt.

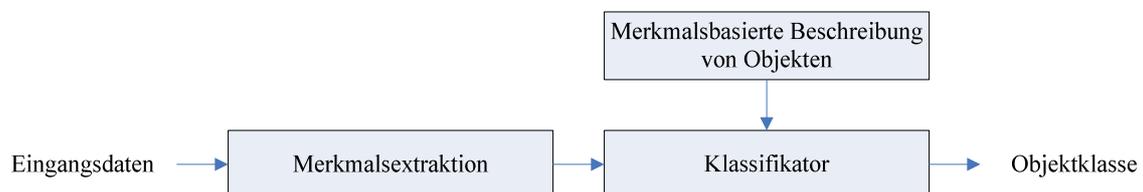


Abbildung 2.1: Algorithmus zur Erkennung eines Objektes

Die Aufgaben des zweiten Falls werden normalerweise folgendermaßen gelöst: die Szene wird segmentiert und jedes Segment wird klassifiziert (s. Abb. 2.2).

Allerdings hat dieses Verfahren folgende Nachteile [24]:

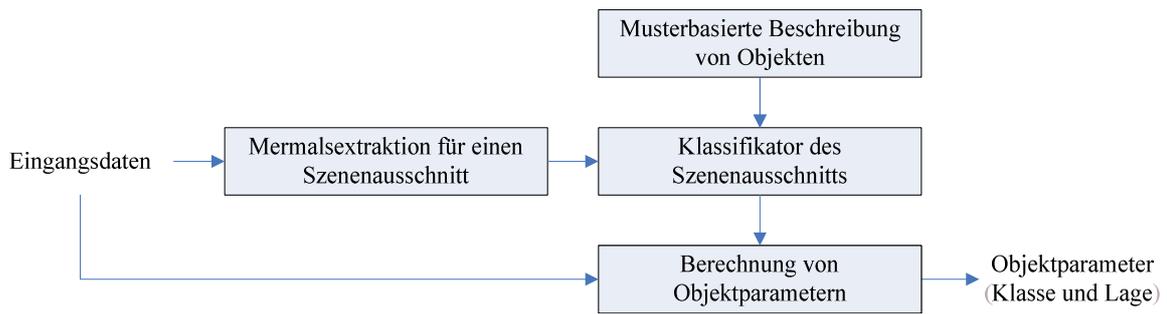


Abbildung 2.2: Algorithmus zur Suche und Erkennung mehrerer Objekte in der Szene

- Ein Segment der Szene, das von einem Klassifikator segmentiert wurde und zu einem Objekt zugeordnet wurde, bestimmt keine Koordinaten des Objektes.
- In manchen Fällen enthält die Szene verschiedene Segmente, die keinen räumlichen Zusammenhang haben, aber ein gemeinsames Objekt beschreiben. Dies führt zur falschen Erkennung von Objekten.
- In manchen Fällen können Segmente, die verschiedene Objekte beschreiben, einen Zusammenhang haben. Dies führt zum Verlust eines Objektes.
- Die Segmente, die verschiedene Objekte beschreiben, können in räumlicher Nähe liegen. Dies führt zum Verlust eines Objektes oder zur Fehlerkennung von Objekten.

Diese Nachteile führen dazu, dass die Geschwindigkeit, die Genauigkeit und die Zuverlässigkeit der Objekterkennung für die Lösung einer Reihe von praktischen Aufgaben nicht ausreichend sind. Zu solchen Aufgaben gehören:

1. Registrierung medizinischer Bilddatensätze. Diese Aufgabe besteht in der Analyse von Tomogrammen eines Organs oder Körperteils und in der Suche nach ähnlichen Tomogrammen in einer Datenbank zur Diagnosefeststellung oder für die Fusion mit einem Tomogramm anderen Typs (z.B. CT- und MRT-Tomogramme) des selben Organs.
2. Schrifterkennung in der Industrie, die durch Rauschen, Verzerrungen und Deformation des Textes erschwert wird.
3. Erkennung mehrerer Objekte verschiedener Klassen, die überlappt sein können, und ihrer Lage für das Griff-in-die-Kiste-System.

Bei der Lösung dieser Aufgaben müssen folgende Besonderheiten berücksichtigt werden:

- Wesentliche Modifikationen oder Anpassungen bei der Veränderung der Datensätze sind nicht erwünscht
- Gleichzeitige Objekt- und Lageerkennung (Position und Orientierung)

- Hohe Geschwindigkeit der Objekt- und Lageerkennung (z.B. für die Schrifterkennung und das Griff-in-die-Kiste-System)

Bei der Entwicklung von Erkennungssystemen müssen auch die Besonderheiten der Daten und zu erkennenden Objekte berücksichtigt werden. Nach einer Veränderung der Daten oder zu erkennenden Objekten müssen die Parameter der Erkennungsfunktionen neu eingestellt werden, was zum unerwünschten Aufwand führt. Außerdem müssen auch die Bedingungen an den Einsatz von Methoden und Algorithmen berücksichtigt werden, z.B. im industriellen Umfeld.

Nach der Analyse der oben genannten Besonderheiten wurden folgende Anforderungen zu einer Methode für die Objekt- und Lageerkennung definiert (s. Tab. 2.1).

Tabelle 2.1: Anforderungen an Methoden für die Objekt- und Lageerkennung

N.	Anforderung
Anforderungen an die Ergebnisse der Erkennung	
1	Klassifizierung aller Objekte in der Szene
2	Bestimmung der Lage (Position und Orientierung) aller Objekte in der Szene
Anforderungen an das Lernen	
3	Möglichkeit des automatischen Lernens
4	Möglichkeit des Selbstlernens während der Erkennung
5	Keine wesentliche Modifikation oder Anpassung bei der Veränderung der Datensätze
Anforderungen an die Geschwindigkeit	
6	Objekt- und Lageerkennung in Echtzeit
7	Gleichzeitige Objekt- und Lageerkennung
Anforderungen an die Merkmalsextraktion	
8	Recheneffizienz des Algorithmus
9	Merkmalsextraktion in Echtzeit
10	Minimaler Einfluss auf die Effizienz des Erkennungsalgorithmus
11	Objekte verschiedener Klassen müssen Merkmale besitzen, die sich sehr stark voneinander unterscheiden
12	Kompaktheit des Merkmalssatzes
Anforderungen an die Zuverlässigkeit	
13	Unempfindlichkeit gegenüber Rauschen und Verzerrungen
14	Unempfindlichkeit gegenüber Helligkeit und Kontrast
15	Unempfindlichkeit gegenüber affinen Transformationen von Objekten (Position und Orientierung)
16	Invarianz gegenüber Aufnahmeposition der Szene
Anforderungen an die Genauigkeit	
17	Hohe Genauigkeit der Objekt- und Lageerkennung

Um die Methode, die alle diese Anforderungen erfüllt, zu entwickeln, müssen existierende Methoden zur Objekt- und Lageerkennung analysiert werden.

2.2 Methoden zur Objekt- und Lageerkennung

2.2.1 Klassifikation der Methoden zur Objekt- und Lageerkennung

In der Literatur existieren verschiedene Verfahren zur Einteilung von Objekterkennungsmethoden in Gruppen. So kann die Klassifikation beispielsweise anhand von Merkmalen der Objekte durchgeführt werden ([25]). Folgende Fälle sind zu unterscheiden:

- Methoden zur Objekt- und Lageerkennung, die auf der Entscheidungstheorie basieren
- Strukturelle Methoden, die die strukturellen Beziehungen von Objekten bei der Erkennung berücksichtigen

In [56, 15] wird eine andere Klassifikation durchgeführt:

- Template matching. Solche Methoden erkennen das Objekt durch Vergleich mit der Schablone aus der Datenbank ([93])
- Statistische Methoden. Solche Methoden verwenden statistische Eigenschaften von Objekten (z.B. Verteilungsdichtefunktionen) ([39, 11])
- Neuronale Netze
- Strukturelle und syntaktische Methoden. Solche Methoden verwenden logische Regeln und Grammatiken für die Objekterkennung

In [73, 19] wird die Klassifikation folgendermaßen durchgeführt:

- Musterbasierte Methoden: Erkennung basiert auf Operationen mit den Merkmalen von Objekten
- Objektbasierte Methoden: Erkennung basiert auf Operationen mit den ganzen Objekten

2.2.2 Musterbasierte Methoden

Musterbasierte Methoden verwenden verschiedene Eigenschaften und Zusammenhänge von Objektmerkmalen, z.B. Merkmalswerte, Merkmalsverteilung. Diese Methoden führen die Erkennung nur anhand von Merkmalen ohne Berücksichtigung des Gesamtobjektes durch.

Fast jede musterbasierte Methode kann man folgendermaßen darstellen: Nehmen wir an, dass es W Klassen von Objekten $\omega_1, \omega_2, \dots, \omega_W$ gibt. Dann müssen W Entscheidungsfunktionen $d_1(x), d_2(x), \dots, d_W(x)$ bestimmt werden, die ein Muster x zu der Klasse ω_i zuordnen.

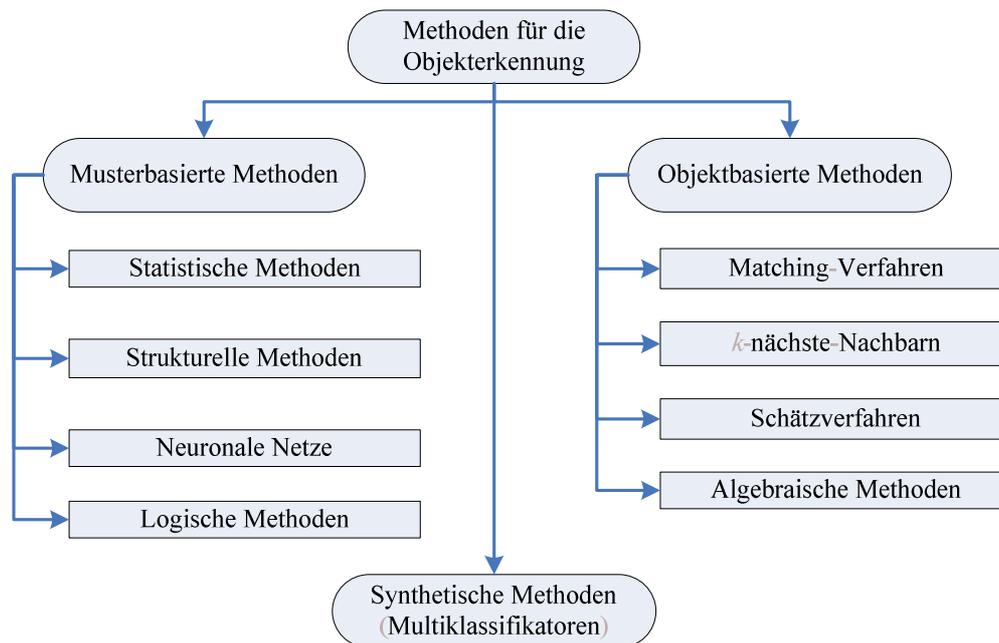


Abbildung 2.3: Methode für die Objekterkennung

Ein unbekanntes Muster x wird der Klasse i zugeordnet, wenn die Entscheidungsfunktion $d_i(x)$ den höchsten Wert hat. Bei der Unbestimmtheit müssen zusätzliche Bedingungen analysiert werden:

$$d_i(x) > d_j(x) \quad j = 1, 2, \dots, W; j \neq i \quad (2.1)$$

Die Entscheidungsgrenze zwischen Klassen ω_i und ω_j wird durch die Menge x bestimmt, für die gilt

$$d_i(x) = d_j(x) \quad (2.2)$$

oder durch die Funktion d_{ij} , für die gilt

$$d_{ij}(x) = d_i(x) - d_j(x) = 0, \quad i, j = 1, \dots, W, i \neq j \quad (2.3)$$

Optimale statistische Klassifikatoren

Die Methoden dieses Ansatzes betrachten die Merkmalsvektoren als Realisierung eines mehrdimensionalen Wertes, der im Merkmalsraum nach einer Wahrscheinlichkeitsdichtefunktion verteilt ist.

Die Methoden dieses Ansatzes können folgendermaßen klassifiziert werden:

- Parametrische Methoden. Der analytische Form der Verteilungsdichtefunktion ist bekannt.
- Nichtparametrische Methoden. Die Verteilungsdichtefunktion ist nicht bekannt.

Der bekannteste Klassifikator aus der Klasse von parametrischen Methoden ist der Bayes-Klassifikator: ein Muster x ist zur Klasse ω_i zugeordnet, wenn $r_i(x) < r_j(x)$ für $j = 1, 2, \dots, W; j \neq i$ (mit r_j - mittlerer Verlust durch Zuteilung von x zur Klasse ω_j). Der Bayes-Klassifikator minimiert die Summe von mittleren Verlusten:

$$\sum_{k=1}^W L_{ki} p(x|\omega_k) P(\omega_k) < \sum_{q=1}^W L_{qj} p(x|\omega_q) P(\omega_q) \quad (2.4)$$

mit $p(x|\omega_k)$ - Wahrscheinlichkeitsverteilung des Musters x , wenn die Klasse ω_k vorgegeben ist; $P(\omega_k)$ - die Wahrscheinlichkeit, dass Klasse ω_k auftritt; L_{ki}/L_{qj} - Verlust durch Zuteilung von Muster x , das zur Klasse $\omega_{k/q}$ gehört, zur Klasse $\omega_{i/j}$.

Um diese Methode für die Objekterkennung zu verwenden, muss die Wahrscheinlichkeitsverteilung jeder Klasse und die Wahrscheinlichkeit des Auftretens jeder Klasse bekannt sein. In der Regel wird angenommen, dass alle Klassen gleichwahrscheinlich sind: $P(\omega_j) = \frac{1}{W}$. Alternativ kann die Wahrscheinlichkeitsverteilung mit Hilfe von apriori-Wissen geschätzt werden.

Wenn die Wahrscheinlichkeitsverteilung $p(x|\omega_k)$ unbekannt ist, müssen die Methoden aus der Wahrscheinlichkeitstheorie eingesetzt werden, die mehrdimensionale Zufallsgrößen beschreiben. Wenn der Merkmalsvektor x ist n -dimensional, dann ist $p(x|\omega_k)$ eine n -dimensionale Funktion. Allerdings ist es kompliziert, diese Methoden praktisch zu verwenden, wenn die Stichproben jeder Klasse nicht repräsentativ sind oder Wahrscheinlichkeitsverteilungen kaum zu definieren sind.

Nichtparametrische Methoden verwenden verschiedene Distanzfunktionen zwischen den Objekten aus den Lernstichproben und den zu erkennenden Objekten. Deshalb können diese Methoden den objektbasierten Methoden zugeordnet werden.

Die Schwierigkeit bei der Verwendung dieser Methoden ist die Unzuverlässigkeit bei Verwendung nichtrepräsentativer Lernstichproben [89].

Andere Klassifikatoren dieses Ansatzes sind:

- Minimax-Algorithmus
- Neyman-Pearson Klassifikator

Logische Methoden

Logische Methoden basieren auf der booleschen Logik und werten die Information aus, die sich sowohl in einzelnen Merkmalen als auch in den Kombinationen dieser Merkmale befindet.

Für jede Klasse ω_i wird ein System der logischen Entscheidungsregeln definiert, die diese Klasse beschreiben [100]:

$$R_i = \{ \varphi_i^t : X \rightarrow 0, 1 | t = 1, \dots, T_i \}, \quad i = 1, 2, \dots, W \quad (2.5)$$

mit T_i - Anzahl an logischen Regeln; W - Anzahl an zu erkennenden Klassen.

Wenn die Anwendung der Regel φ_i^t 1 ergibt, dann wird das Muster x der Klasse ω_i zugeordnet. Wenn $\varphi_i^t = 0$, dann macht die Regel φ_i^t keine Aussage.

Im einfachsten Fall wird die gewichtete Summe der Regeln, die das Muster x der Klasse ω_i zuordnen, wie folgt berechnet:

$$\Gamma_i(x) = \frac{1}{T_i} \sum_{t=1}^{T_i} \varphi_i^t(x) \quad (2.6)$$

T_i wird hier als Normierungsfaktor benutzt.

Das Muster x wird der Klasse zugeordnet, für die $\Gamma_i(x)$ maximal ist:

$$a(x) = \max(\Gamma_i(x)), \quad i = 1, \dots, W \quad (2.7)$$

Unter Verwendung von Stimmengewichtung wird für jede Regel φ_i^t ein Gewicht $\alpha_i^t \geq 0$ definiert:

$$\Gamma_i(x) = \sum_{t=1}^{T_i} \alpha_i^t \varphi_i^t(x), \quad \alpha_i^t \geq 0 \quad (2.8)$$

Die Gewichte für die Regeln jeder Klasse sind normiert:

$$\sum_{t=1}^{T_i} \alpha_i^t = 1, \quad i = 1, 2, \dots, W \quad (2.9)$$

Im allgemeinen Fall lässt sich sagen, dass die logischen Methoden Zusammenhänge in der Lernstichprobe finden und ein System von logischen Entscheidungsregeln, die gewichtet sind, definieren. Diese Gruppe ist vielfältig und enthält Methoden verschiedener Komplexität. Für die booleschen Merkmale werden Baum-Klassifikatoren, Deadlock-Algorithmen und Kora-Algorithmus [12] eingesetzt.

Kora-Algorithmus und anderer logische Methoden sind rechenintensiv, weil sie eine Brute-Force-Methode für die Konjunktionsanalyse verwenden, deshalb sind diese Methoden effizient nur für die Erkennung in kleindimensionalen Merkmalsräumen [89].

Strukturelle Methoden

Methoden dieser Klasse berücksichtigen die strukturelle Zusammenhänge von zu erkennenden Objekten. Zu dieser Klasse gehören folgende Methoden:

- **Matching Shape Numbers.** Diese Methode basiert auf dem Vergleich von Konturen, die als Zeichenkette beschrieben sind ([6]). Die Methode ist dem Template Matching ähnlich, verwendet jedoch ein anderes Distanzkriterium: Der Ähnlichkeitsgrad k von zwei Konturen ist die Baumtiefe, bei der die Objekte noch übereinstimmen.

Der Distanz zwischen zwei Formen a und b ist umgekehrt proportional zu R :

$$D(a, b) = \frac{1}{R} \quad (2.10)$$

Je größer R ist, desto ähnlicher sind zwei Objekte/Formen/Konturen. (s. Abb. 2.4).

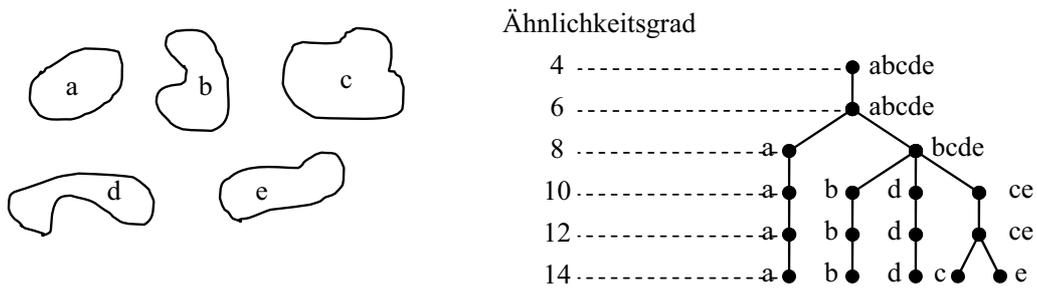


Abbildung 2.4: Hypothetischer Ähnlichkeitsbaum

- **Matching von Zeichenfolgen.** Die Konturen von zwei Objekten a und b sind als folgende Zeichenfolgen kodiert: $a_1a_2 \dots a_n$ und $b_1b_2 \dots b_m$. Es stimmen die Zeichenfolgen bei der Position k überein, wenn $a_k = b_k$. α ist die Anzahl von Übereinstimmungen von Zeichenfolgen. In diesem Fall ist die Anzahl von Zeichen, die nicht übereinstimmen:

$$\beta = \max(|a|, |b|) - \alpha \quad (2.11)$$

mit $|arg|$ - Anzahl von Zeichen. $\beta = 0$, wenn a und b identisch sind [25].

Der Ähnlichkeitsgrad von Objekten a und b kann folgendermaßen berechnet werden:

$$R = \frac{\alpha}{\beta} = \frac{\alpha}{\max(|a|, |b|) - \alpha} \quad (2.12)$$

R ist unendlich, wenn zwei Objekte identisch sind, und ist gleich 0, wenn a und b verschieden sind (in diesem Fall $\alpha = 0$).

- **Syntaktische Erkennung von Zeichenfolgen.** Der Algorithmus dieser Methode besteht aus den folgenden Schritten:
 1. Folgende Elemente werden festgelegt (s. Abb. 2.5):
 - Strukturelemente, die ein Objekte beschreiben können
 - Eine Grammatik
 2. Ein Automat für die Erkennung wird erzeugt. Die Struktur dieses Automaten wird durch Grammatik bestimmt.

Die Grammatik wird mit Hilfe von vier Mengen definiert:

$$G = (N, T, P, S) \quad (2.13)$$

mit N - Nichtterminalsymbolen; T - Terminalsymbolen; P - Produktionsmenge; S - Startsymbol.

Der Prozess der Erkennung läuft wie folgt ab: Es wird erkannt, ob ein Objekt zu einer Sprache $L(G)$ gehört, die von Grammatik G definiert ist. Mit Hilfe von syntaktischen Regeln können verschiedene Zusammenhänge festgelegt werden.

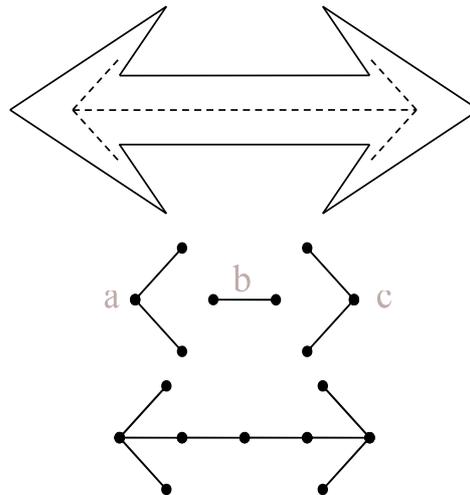


Abbildung 2.5: Objekt. Strukturelemente. Struktur, die von der Grammatik definiert wird.

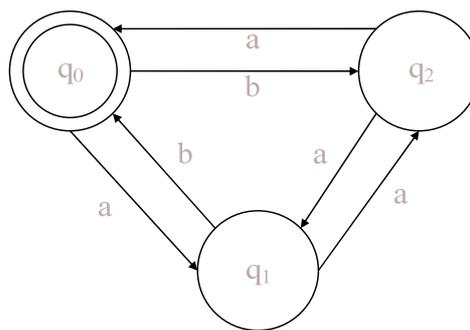


Abbildung 2.6: Endlicher Zustandsautomat für die syntaktische Erkennung von Symbolfolgen

Der Automat, der in der Abbildung 2.6 dargestellt ist, erkennt die Zeichenfolge $w = abbaaa$ aber erkennt die Zeichenfolge $w = aabb$ nicht. Eine syntaktische Methode zur Erkennung verwendet einen endlichen Automaten, der in der Lage ist, ein Objekt, das als Zeichenfolge dargestellt ist, zu erkennen.

- **Syntaktische Erkennung von Bäumen.** Objekte, die erkannt werden müssen, können auch als Bäume dargestellt werden. In diesem Fall wird eine Grammatik von Bäumen festgelegt. Der Automat, der solche Bäume annimmt, fängt im Gegensatz zum Automaten für die Zeichenfolgeerkennung mit den Baumblättern an.

Syntaktische Methoden verwenden für die Erkennung jeder Klasse von Objekten einen Automaten. Diese Automaten werden nach der Analyse des Problems oder im Rahmen eines Lernvorganges definiert.

Neuronale Netze

Die obengenannten Methoden verwenden Lernstichproben, jede von denen mehrere Objekte einer Klasse enthält, um die Parameter jeder Klasse von Objekten zu berechnen. Der Bayes-Klassifikator beispielsweise verwendet als Parameter Vektoren der Erwartungswerte von Merkmalen und Kovarianzmatrizen von Merkmalen.

In diesem Fall wird das Lernen folgendermaßen durchgeführt: Objekte aus einer Stichprobe werden verwendet, um die Parameter der Entscheidungsfunktion dieser Klasse zu berechnen. Nach der Berechnung von Parametern wird die Struktur des Klassifikators fixiert. Deshalb ist die Qualität der Erkennung von den Lernstichproben abhängig.

Bei der Lösung realer Aufgaben sind die statistischen Parameter von Klassen meistens unbekannt und können nicht berechnet werden. Es bietet sich an, andere Erkennungsmethoden zu benutzen, die Entscheidungsfunktionen auch beim Lernen definieren können. Eine dieser Methoden sind neuronale Netze ([26]).

In der Abbildung 2.7 wird ein Perzeptron, das Objekte von zwei Klassen erkennen kann, dargestellt.

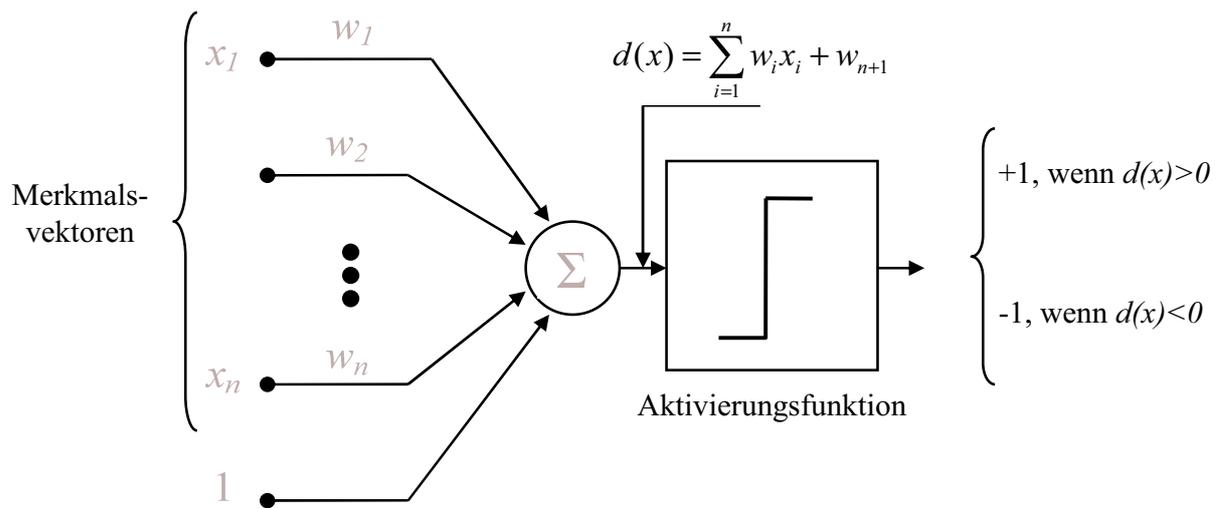


Abbildung 2.7: Neuronales Netz, das zwei Klassen erkennt

Die Entscheidungsfunktion eines solchen Perzeptrons basiert auf der gewichteten Summe von Eingangssignalen (Merkmale) und realisiert die Zuordnung der zu erkennenden Objekte zu verschiedenen Klassen ([47, 87]):

$$d(x) = \sum_{i=1}^n w_i x_i + w_{n+1} \tag{2.14}$$

mit $w_i, i = 1 \dots n, n + 1$ - Gewichte.

Ein mehrschichtiges neuronales Netz, das Objekte mehrerer Klassen erkennen kann, ist in der Abbildung 2.8 dargestellt.

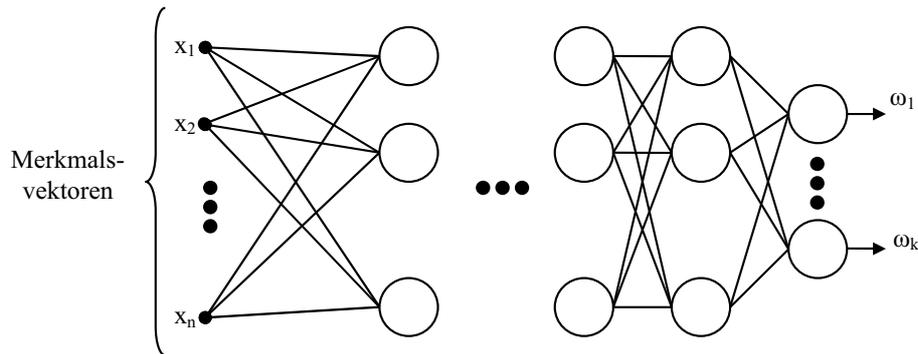


Abbildung 2.8: Mehrschichtiges neuronales Netz, das mehrere Klassen erkennen kann

2.2.3 Objektbasierte Methoden

Die objektbasierten Methoden betrachten die Objekte als Ganzes im Sinne eine bestimmten Kombination ihrer Merkmale. Die Hauptoperationen, die die Methodendieser Gruppe realisieren, sind die Bestimmung der Ähnlichkeit und des Unterschieds zwischen Objekten ([72, 71, 9]). In diesem Zusammenhang werden hier die Begriffe „Objekt“ und „Muster“ gleichgesetzt.

Template matching

Die Matching-Methoden betrachten jede Klasse als Vektor von Merkmalen eines Objektes, das als Prototyp (Schablone) dieser Klasse gilt ([4]). Wenn ein unbekanntes Objekt erkannt wird, wird dieses Objekt zu der Klasse zugeordnet, deren Prototyp am besten dieses Objekt beschreibt. Solche Prototypen können auch die realen Objekte sein. Der Ähnlichkeitsgrad des zu erkennenden Objekts und des Prototyps einer Klasse kann auf verschiedene Weise berechnet werden. Die bekanntesten Vertreter dieser Methoden sind der Minimum-Distanz-Klassifikator und Template Matching mittels Korrelation (Maximum-Korrelation-Klassifikator).

Der **Minimum-Distanz-Klassifikator** berechnet der euklidische Distanz zwischen dem Merkmalsvektor des zu erkennenden Objektes und dem Merkmalsvektor des Prototyps einer Klasse. Die Entscheidung für eine Klasse wird auf Basis der minimalen Distanz getroffen.

Prototyp einer Klasse ist der Erwartungswertvektor von Merkmalen dieser Klasse:

$$m_j = \frac{1}{N_j} \sum_{x \in \omega_j} x_j, \quad j = 1, 2, \dots, W \quad (2.15)$$

mit N_j - Anzahl an Merkmalsvektoren der Klasse ω_j .

Die Distanz wird folgendermaßen berechnet:

$$D(x, m_j) = \|x - m_j\|, \quad j = 1, 2, \dots, W \quad (2.16)$$

mit $\|m\|$ - Euklidische oder Mahalanobis Distanz. Als $\|m\|$ kann auch Tangendistanz verwendet werden [11].

Ein Beispiel der Berechnung des Tangentenabstands ist in der Abbildung 2.9 dargestellt.

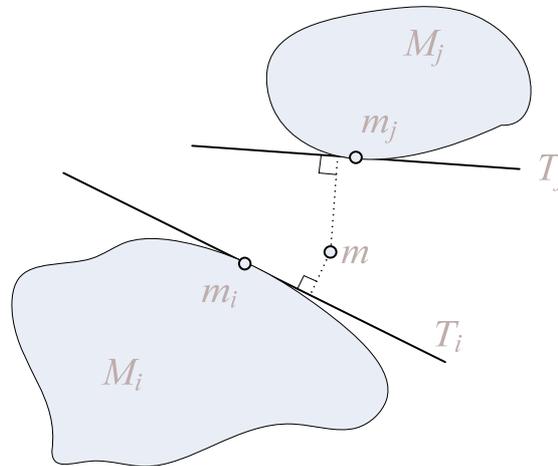


Abbildung 2.9: Berechnung der Tangendistanz

Der Abstand zwischen Merkmalsvektoren m und m_i wird nicht als Punktabstand sondern als Abstand $D(m, m_i)$ von m_i zur Tangente T_i der Mannigfaltigkeit M_i berechnet. m gehört zur Klasse ω_i , wenn $D(m, m_i) < D(m, m_j)$.

Die Tangendistanz ist robust bezüglich kleiner, systematischer Variationen.

Der Minimum-Distanz-Klassifikator wird sinnvollerweise eingesetzt, wenn die Distanz zwischen den Erwartungswerten von zwei Klassen größer als die Distanz zwischen zwei Objekten einer Klasse ist.

Bei der Lösung von realen Aufgaben kann die Distanz zwischen den Erwartungswerten von Klassen genauso groß wie von Objekten einer Klasse sein. In diesem Fall müssen die Daten sorgfältig vorverarbeitet werden.

Der **Kreuzkorrelationsklassifikator** verwendet die Korrelation zwischen Objekten. Nehmen wir an, es gibt zwei Bilder $f(x, y)$ und $w(x, y)$. Dann wird die Korrelation zwischen diesen Bildern folgendermaßen berechnet:

$$c(x, y) = \sum_s \sum_t f(s, t)w(x + s, y + t) \quad (2.17)$$

Eine solche Kreuzkorrelationsfunktion ist sehr empfindlich gegenüber Veränderung der Amplituden von f und w . Deshalb wird ein Korrelationskoeffizient wie folgt berechnet:

$$\gamma(x, y) = \frac{\sum_s \sum_t [f(s, t) - \bar{f}(s, t)] [w(x + s, y + t) - \bar{w}]}{\sqrt{\sum_s \sum_t [f(s, t) - \bar{f}(s, t)]^2 \sum_s \sum_t [w(x + s, y + t) - \bar{w}]^2}} \quad (2.18)$$

mit \bar{w} - Durchschnittsgrauwert in der Schablone w ; \bar{f} - Durchschnittsgrauwert des Bildes f in der Region, welcher der Position von w entspricht. Die Summe wird entlang aller

Koordinaten berechnet. Der Wertbereich des Korrelationskoeffizienten liegt zwischen -1 und 1. Er ist von der Veränderung von Amplituden f und w unabhängig.

Die Normierung der Größe benötigt eine räumliche Skalierung von Schablonen. Dieser Vorgang ist sehr rechenintensiv. Eine Rotationsnormierung erfordert zusätzliche Berechnungen. Um die beste Überlagerung von zwei Bildern zu finden, müssen alle möglichen Rotationen von $w(x, y)$ geprüft werden. Das führt dazu, dass diese Methode bei der Erkennung von rotierten Objekten selten zu gebrauchen ist.

Die Matching-Methoden zeigen gute Ergebnisse, aber sie sind ohne Modifikationen zu rechenintensiv.

Unter Verwendung von invarianten Merkmalen kann den Rechenaufwand minimiert werden. Die Methoden, die invariante Merkmale berechnen, können in drei Kategorien eingeteilt werden [11, 83]:

- Normalisierungsmethoden: Die Idee ist einen Repräsentanten aus jeder Äquivalenzklasse zu finden. Beispielsweise werden Objektkonturen extrahiert und in eine kanonische Form, die gegenüber Rotation, Translation und Skalierung invariant ist, transformiert, z.B. Fourier-Deskriptoren für Konturen.
- Differentialmethoden: Der Ansatz basiert auf der Voraussetzung, dass eine infinitesimale Transformation die Muster nur infinitesimal verändert. In diesem Fall ändert sich die Ableitung des invarianten Merkmals nicht, z.B. Integral über ein Kreissegment.
- Integralmethoden: Ein invariantes Merkmal wird durch die Integration über die Transformationsgruppe konstruiert.

Es gibt eine Reihe von Matching-Methoden, die auf der Registrierung von Daten und Modellen der zu erkennenden Objekte basieren (**Modellbasierte Objekterkennung**):

1. In [104] wird es vorgeschlagen, die 2D-zu-3D-Registrierung durchzuführen. Das Modell eines zu erkennenden Objektes wird als 3D-CAD-Modell definiert. Anschließend werden die Merkmale aus dem 2D-Bild extrahiert und mit dem Modell mittels ICP-Algorithmus (Iterative Closest Point) registriert. Als Merkmale werden Kanten von Objekten verwendet. In einem beispielhaften Experiment [104] werden die folgenden Ergebnisse erzielt: die Erkennungszeit beträgt 5-6 s, die Genauigkeit der Lageerkennung ist 0.5mm und 1° . Die Rotationen von Objekten, die bestimmt werden können, liegen im Intervall $[-50^\circ \dots 50^\circ]$. Dabei ist die Anzahl an Merkmalen mit 150 vergleichsweise klein.

In [57, 4] wird die 3D-zu-2D-Registrierung beschrieben. Das 3D-Modell eines Objektes wird auf das Bild projiziert und die aus dem Bild extrahierte Merkmale werden mit den Merkmalen des Modells registriert.

Ein ähnliches Verfahren wurde in [52] entwickelt: zunächst wird aus einer Menge an rigiden Modellen ein deformierbares Modell eines Objektes erzeugt, das eine Reihe von geometrisch ähnlichen Objekten repräsentiert, d.h. ein Prototyp, das eine Klasse von Objekten beschreibt. Dann werden ein Bild, das das zu erkennende Objekt enthält, und das deformierbare Modell mit Hilfe einer Gütefunktion, die als Summe

der Abstände zwischen Bild- und Modellkantensegmenten definiert ist, registriert. Die Modellanpassung wird auf Gradientenbildern durchgeführt. Bei der Optimierung der Gütefunktion werden Grob-zu-Fein-Strategie (für eine sortierte Menge von Skalierungsparametern des Modells wird jeweils eine Optimierung durchgeführt) und iterative Optimierung (die Optimierung wird als iterativer Prozess über eine bestimmte Anzahl von Modellparametern realisiert) eingesetzt. Nach der Lageschätzung werden die Merkmale des zu erkennenden Objekts aus dem Bild extrahiert, z.B. bei der Erkennung von Flugzeugen werden folgende Merkmale verwendet: Winkel zwischen der Längsachse und der Tragfläche, Länge und Spannweite des vorderen und hinteren Flugzeugteils. Abschließend wird das Objekt mittels dieser Merkmale mit einem Klassifikator klassifiziert.

2. Um die Objekterkennung in komplexen 3D-Szenen durchzuführen, wird in [40] vorgeschlagen, eine 3D-Szene in einer szenenbezogenen Form zu kodieren. Diese Idee wird auch in [37], [32] verwendet. Diese Methode extrahiert aus der 3D-Szene sogenannte *spin-images* (orientierte Punkte), die mit dem Modell des zu erkennenden Objekts registriert werden. Die Erkennung von mehreren Objekten wurde nicht realisiert. Ein ähnliches Prinzip wird auch in [106] verwendet.

In [6] werden die Konturen von Objekten unter Berücksichtigung des *shape contexts* registriert.

3. In [44, 53] wird folgendes Prinzip verwendet: aus der 2D- oder 3D-Daten wird eine Reihe von unterschiedlichen Merkmalen extrahiert, die mit den Merkmalen von Modellen der zu erkennenden Objekte registriert werden.

Eine ähnliche Methode wird in [71] vorgeschlagen: Das Modell des zu erkennenden Objekts wird als einer Satz von *patches* definiert. Anschließend werden diese *patches* in 2D-Bild gesucht und mit dem Modell gematcht.

In [49, 50, 48] wird Visual-Servoing behandelt, wobei die Erkennung von getrackten Objekten basiert auf dem Matching von Merkmalen, die aus den Daten (2D-Bild) und aus dem Modell des Objektes extrahiert werden.

In [66, 67] wird vorgeschlagen, die Eigenvektoren und Eigenwerte aus den Bildern zu berechnen und diese Information für die Objekt- und Lageerkennung zu verwenden: die Eigenpunkte werden gematcht.

4. In [5] werden ein 2D-Bild und mehrere 2D-Ansichten des 3D-Modells eines Objektes unter Verwendung von genetischen Algorithmen registriert. Das Verfahren hat das Ziel, die Lage eines Objektes in der Szene zu bestimmen. Die Erkennung von mehreren Objekten wird nicht behandelt.

Die oben beschriebenen Methoden haben folgende Probleme:

- Diese Methoden sind meistens sehr rechenintensiv.
- Diese Methoden wurden für bestimmte Datentypen entwickelt und sind daher nicht ohne weiteres für andere Datentypen geeignet.
- Viele Methoden sind nicht in der Lage, mehrere Objekte verschiedener Klassen zu erkennen.

- Einige Methoden benötigen langwierige Vorverarbeitung der Daten.

Eine Übersicht über die modellbasierte Objekterkennung kann man in [13] und in [72] finden. Die meisten Methoden des modellbasierten Matchings wurden im Rahmen von Visual-Servoing-Systemen entwickelt ([77, 2, 54, 95, 43]).

K-Nächste-Nachbarn

Bei der Klassifizierung eines Objektes mit der K-Nächste-Nachbarn-Methode werden k nächstliegende, schon klassifizierte Objekte im Merkmalsraum gesucht. Als Abstandsmaße kann euklidischer Abstand verwendet werden. Dem zu erkennenden Objekt wird die Klasse zugeordnet, die die größte Anzahl bereits klassifizierter Objekte in der k -Nachbarschaft hat (Auszählung der Stimmen). Eine Klasse wird nicht von einem Prototyp, sondern von mehreren Objekten, deren Klasse schon bekannt ist, beschrieben.

Um ein Objekt eindeutig zu klassifizieren, braucht man viele Trainingsdaten, was zur Erhöhung des Rechenaufwands führen kann. Bei der Reduzierung der Anzahl an Trainingsbeispielen wird auch die Repräsentativität der Lernstichproben minimiert.

Schätzverfahren

Schätzverfahren basieren auf der Berechnung von Ähnlichkeitsschätzungen, die den Ähnlichkeitsgrad des zu erkennenden Objekts und der schon klassifizierten Objekte in verschiedenen Merkmalsräumen beschreiben (zu erkennende Objekte sind gleichzeitig in mehreren Merkmalsräumen abgebildet). Der Gesamtähnlichkeitsgrad ist eine Kombination der Ähnlichkeitsgrade in allen Merkmalsräumen. Somit ermöglicht dieses Verfahren den verschiedenen Informationsgehalt von Merkmalen zu berücksichtigen.

Für die Schätzung der Ähnlichkeit von zwei Objekten verwendet der Klassifikator folgende Komponente:

- Funktionen zur Berechnung des Ähnlichkeitsgrades einzelner Merkmale von zwei Objekten
- Funktionen zur Berechnung des Ähnlichkeitsgrades von Merkmalskombinationen in Merkmalsräumen
- Gewichtung von einzelnen Merkmalen und Merkmalskombinationen in der Schätzung
- Gewichtung aller klassifizierten Objekte (Prototypen)

Dieses Verfahren ist eine Erweiterung des K-Nächste-Nachbarn-Verfahrens, deshalb hat das Schätzverfahren ähnliche Probleme:

- Verschlechterung der Repräsentativität der Lernstichproben bei der Reduzierung von Trainingsdaten
- Hoher Rechenaufwand

Die Erkennung wird zweistufig durchgeführt: zuerst werden die Schätzungen gemacht und dann wird aufgrund dieser Schätzungen eine Entscheidung getroffen [107].

2.2.4 Synthetische Methoden

Um die Effizienz der Objekt- und Lageerkennung zu erhöhen, wurden verschiedene Methoden zur Entwicklung von den Systemen, die auf den Ergebnissen mehrerer Klassifikatoren basieren, vorgeschlagen. Einen solchen kombinierten Klassifikator nennt man auch Multiklassifikator.

Das Prinzip basiert auf der Annahme, dass ein Objekt von verschiedenen Klassifikatoren mit unterschiedlicher Wahrscheinlichkeit erkannt wird. Die Entwicklung einer synthetischen Methode (Multiklassifikator) kann alle Vorteile der Methoden, die in diesem Klassifikator verwendet werden können, zusammenbringen.

Ein solcher Multiklassifikator kann nach dem Prinzip der sequentiellen oder der parallelen Klassifizierung aufgebaut sein.

1. Bei der sequentiellen Klassifizierung werden die Merkmalsvektoren von allen Klassifikatoren, die der Multiklassifikator in sich vereinigt, sequentiell analysiert (s. Abb. 2.10). Wenn ein Klassifikator in der Lage ist, eindeutig die Klasse eines Objektes zu erkennen, wird die Entscheidung über die Zuordnung sofort getroffen. Ansonst wird die Klassifizierung von dem nächsten Klassifikator durchgeführt. Um einen solchen Klassifikator zu entwickeln, ist es nötig, die Parameter aller Komponenten und ihre Zusammenhänge zu definieren. Der Nachteil dieser Methode besteht darin, dass das Ergebnis der Klassifizierung nur von einem Klassifikator bestimmt ist.

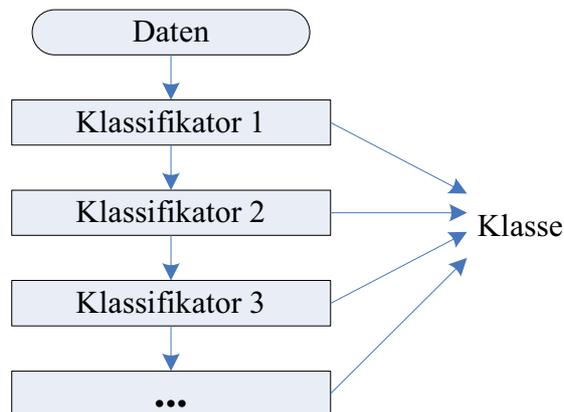


Abbildung 2.10: Sequentieller Multiklassifikator

2. Bei der parallelen Klassifizierung werden die Merkmalsvektoren gleichzeitig analysiert (s. Abb. 2.11). Jeder Klassifikator klassifiziert dasselbe Objekt. In diesem Fall ist es nötig, die Aggregatfunktion, und nicht nur die optimalen Parameter jedes Klassifikators zu definieren.

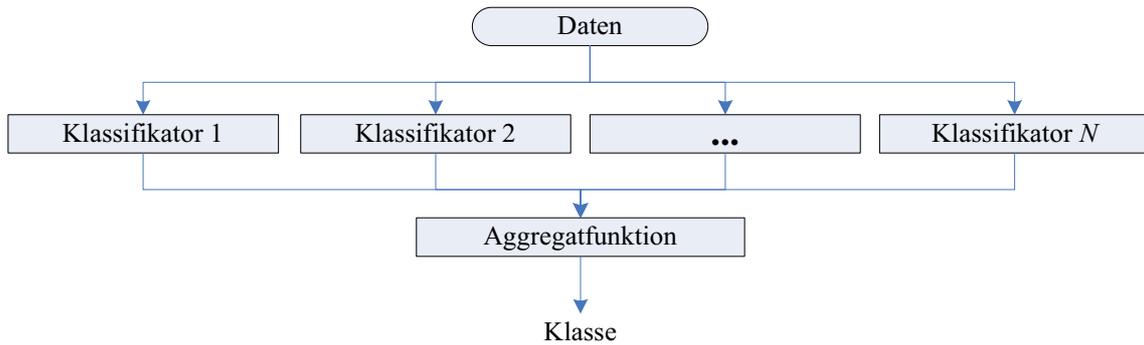


Abbildung 2.11: Paralleler Multiklassifikator

Weiterer Ansatz zur Konstruktion eines synthetischen Multiklassifikators ist das Boosting-Verfahren [20, 84]. Dieses Verfahren besteht in der Kombination von mehreren schwachen Klassifikatoren in einem einzigen starken Klassifikator. Die schwachen Klassifikatoren berücksichtigen meistens nur ein Merkmal des zu erkennenden Objekts. Aus diesem Grund sind sie in der Lage die Klassifizierung ziemlich schnell durchzuführen. Die Klassifikatoren werden so gewichtet, dass die Besten mehr Gewicht erhalten werden. Ein erzeugter Klassifikator sieht folgendermaßen aus:

$$F(x) = \operatorname{sgn}\left(\sum_{i=1}^W w_i d_i(x)\right) \quad (2.19)$$

mit w_i - Gewicht des Klassifikators i ; m - Anzahl an schwachen Klassifikatoren; d_i - schwacher Klassifikator:

$$d_i(x) = \begin{cases} +1 & \text{falls } x \in \omega_i \\ -1 & \text{falls } x \notin \omega_i \end{cases} \quad (2.20)$$

Boosting wird für die Aufgaben verwendet, in denen es notwendig ist, nur zwei Klassen von Objekten zu unterscheiden.

2.2.5 Vergleich der Methoden für die Objekt- und Lageerkennung

Der Überblick über den Stand der Forschung hat gezeigt, dass es eine Reihe von Methoden für die effiziente Objekt- und Lageerkennung gibt. Allerdings ist nicht jede Methode als Software-System realisierbar. Es gibt zwei Gründe dafür:

1. Das mathematische Modell der Daten, die zu erkennende Objekte enthalten, ist nicht trivial zu definieren. Dies führt zur Verwendung von vereinfachten Modellen und Methoden bei der Lösung von praktischen Aufgaben.
2. Jede Methode hat Nachteile, die ihren Anwendungsbereich stark reduzieren. z.B. der Nachteil aller musterbasierten Methoden ist das Problem der Merkmalsgewinnung. Die Unterscheidbarkeit zweier Klassen hängt von den verwendeten Merkmalen ab.

2. Stand der Forschung

In der Tabelle 2.2 sind die Anwendungsbedingungen und die Nachteile jeder Gruppe von Methoden dargestellt.

Tabelle 2.2: Anwendungsbedingungen und Nachteile von existierenden Methoden für die Objekt- und Lageerkennung

Methode	Anwendungsbedingungen	Nachteile
Musterbasierte Methoden Statistische Methoden	<ol style="list-style-type: none"> 1. Bekannte Verteilungen 2. Deutlich getrennte Klassen 	<ol style="list-style-type: none"> 1. Notwendigkeit wesentliche Modifikation bzw. Anpassung bei der Veränderung der Datensätze durchzuführen (Anforderung 5) 2. Unzuverlässigkeit bei Verwendung nichtrepräsentativer Lernstichproben 3. Starke Empfindlichkeit gegenüber Datenverzerrungen (Anforderungen 13, 14) 4. Hoher Schwierigkeitsgrad der Erstellung von Verteilungsdichtefunktionen (Anforderung 8) 5. Kein automatisches Lernen und Selbstlernen (Anforderungen 3, 4) 6. Wesentliche Modifikation der Methode bei der Veränderung von Daten (Anforderung 5)
Logische Methoden	Kleine Dimension des Merkmalsraums	<ol style="list-style-type: none"> 1. Notwendigkeit wesentliche Modifikation bzw. Anpassung bei der Veränderung der Datensätze durchzuführen (Anforderung 5) 2. Notwendigkeit alle logischen Regeln durchzusuchen (Anforderung 8) 3. Rechenintensive Methode (Anforderung 6) 4. Kein automatisches Lernen und Selbstlernen (Anforderungen 3, 4)

Strukturelle Methoden	Kleine Dimension des Merkmalsraums	<ol style="list-style-type: none"> 1. Notwendigkeit wesentliche Modifikation bzw. Anpassung bei der Veränderung der Datensätze durchzuführen (Anforderung 5) 2. Starke Empfindlichkeit gegenüber Verzerrungen (Anforderungen 13, 14) 3. Die Beschreibung von Objekten mittels Grammatik ist nicht trivial (Anforderungen 3, 4)
Neuronale Netze	Beliebige Dimension des Merkmalsraums, aber kleine Anzahl von Klassen	<ol style="list-style-type: none"> 1. Blackbox-Problem 2. Große Anzahl von Lernstichproben und lange Dauer des Lernens (Anforderungen 8, 10)
Objektbasierte Methoden		
Matching-Verfahren	<ol style="list-style-type: none"> 1. Beliebige Dimension des Merkmalsraums 2. Große Anzahl von Klassen 	<ol style="list-style-type: none"> 1. Notwendigkeit wesentliche Modifikation bzw. Anpassung bei der Veränderung der Datensätze durchzuführen (Anforderung 5) 2. Starke Abhängigkeit der Ergebnisse von der Gütefunktion (Anforderungen 3, 4) 3. Starke Empfindlichkeit gegenüber Verzerrungen von Objekten (Anforderungen 13, 14) 4. Rechenintensive Methode (Anforderung 6)
K -Nächste-Nachbarn	<ol style="list-style-type: none"> 1. Kleine Dimension des Merkmalsraums 2. Kleine Anzahl von Klassen 	<ol style="list-style-type: none"> 1. Notwendigkeit wesentliche Modifikation bzw. Anpassung bei der Veränderung der Datensätze durchzuführen (Anforderung 5) 2. Starke Abhängigkeit der Ergebnisse von der Gütefunktion (Anforderungen 3, 4, 5) 3. Rechenintensive Methode: Alle Prototypen der Klassen müssen geprüft werden (Anforderung 6)

Schätzverfahren	<ol style="list-style-type: none"> 1. Kleine Dimension des Merkmalsraums 2. Kleine Anzahl von Klassen 	<ol style="list-style-type: none"> 1. Notwendigkeit wesentliche Modifikation bzw. Anpassung bei der Veränderung der Datensätze durchzuführen (Anforderung 5) 2. Starke Abhängigkeit der Ergebnisse von der Gütefunktion (Anforderung 5) 3. Rechenintensive Methode: Alle Prototypen von Klassen müssen geprüft werden (Anforderung 6) 4. Keine automatisches Lernen und Selbstlernen (Anforderungen 3, 4)
Synthetische Methoden	<ol style="list-style-type: none"> 1. Kleine Dimension des Merkmalsraums 2. Kleine Anzahl von Klassen 	<ol style="list-style-type: none"> 1. Notwendigkeit wesentliche Modifikation bzw. Anpassung bei der Veränderung der Datensätze durchzuführen (Anforderung 5) 2. Alle Nachteile der Methoden, die zu den Multiklassifikator gehören

Aus der Analyse, die in der Tabelle 2.2 zusammengefasst ist, lässt sich ersehen, dass keine Methode alle Anforderungen aus Tabelle 2.1 erfüllt. Einer der weitverbreiteten Nachteile ist, wesentliche Modifikation bzw. Anpassung bei der Veränderung der Datensätze durchzuführen, d.h. keine Methode ist eine universelle Methode im Sinne der obengenannten Anforderungen.

Die meisten Methoden haben einen weiteren Nachteil: sie können keine gleichzeitige Objekt- und Lageerkennung durchführen. Viele Methoden können ein Objekt lokalisieren, aber keine genaue Lage (Position und Orientierung) in der Szene bestimmen.

Als ein allgemeiner Nachteil kann die folgende Einschränkung genannt werden: der Merkmalsraum muss möglichst kleine Dimension haben. Viele Methoden haben auch keine Invarianz gegenüber von verschiedenen Transformationen der Objekte (z.B. Rotation, Skalierung, Deformation)

2.3 Fazit und Zielsetzung

Die Analyse von verschiedenen Methoden zur Objekt- und Lageerkennung, die in diesem Kapitel durchgeführt wurde, hat gezeigt, dass es viele ungelöste Probleme im Bereich der kombinierten Objekt- und Lageerkennung gibt. Die Forschung in diesem Bereich konzentriert sich meistens auf die Lösung von konkreten, spezialisierten Aufgabenstellungen. Der Grund dafür ist das Fehlen eines universellen Verfahrens, das die Erkennung von Objekten verschiedener Klassen ermöglicht. Hieraus folgt der wesentliche Nachteil bestehender Erkennungssysteme: Solche Systeme sind von den verarbeiteten Daten und Objekten stark abhängig. Bei der Veränderung der Daten oder von zu erkennenden Objekten ist es notwendig, das System zu modifizieren oder anzupassen. Außerdem muss man auch den Lernprozess erneut durchzuführen.

Ein weiterer Nachteil von existierenden Systemen ist die nicht vorhandene oder eingeschränkte Fähigkeit der gleichzeitigen Objekt- und Lageerkennung im 2D- oder 3D-Raum.

Um die oben genannten Nachteile zu beseitigen, ist die Entwicklung einer Methode für die kombinierte Objekt- und Lageerkennung erforderlich, die auf der Kombination eines Matching-Verfahrens und Optimierungsmethoden basiert. Das Prinzip dieses Verfahrens besteht in der Vereinigung von verschiedenen Matching-Methoden unter Verwendung eines allgemeinen, mathematischen Modells der Objekterkennung auf Basis der Auswertung von unstrukturierten 2D und 3D Daten, welches von den Datentypen und den zu erkennenden Objekten unabhängig ist.

Das Ziel der hier vorgestellten Arbeit ist die Erhöhung von Geschwindigkeit, Genauigkeit und Zuverlässigkeit der Objekt- und Lageerkennung durch Auswertung von 2D und 3D unstrukturierten Daten mit der Hilfe einer neuen hierarchisch-iterativen Methode.

Kapitel 3

Hierarchisch-iterative Methode für die Objekt- und Lageerkennung

3.1 Entwicklung der hierarchisch-iterativen Methode für die Objekt- und Lageerkennung

3.1.1 Aufgabenstellung der Objekt- und Lageerkennung

Nach der Analyse der den praktischen Anforderungen an die Objekt- und Lageerkennung, die im Kapitel 2.1 genannt wurden, wird im Folgenden die Aufgabenstellung der kombinierten Objekt- und Lageerkennung formal definiert.

Gegeben:

- W Prototypen M_1, M_2, \dots, M_W , die W Klassen beschreiben
- R^K - Daten (vorverarbeitete Sensordaten), die eine Anzahl von Objekten X_1, X_2, \dots, X_N enthalten (K - Anzahl der Dimensionen)

Gesucht:

für jedes Objekt X_j ist ein Paar $\{M_i, P_i\}$ zu finden, welches die Klasse und die Lage des Objekts in der Szene beschreibt:

$j = 1 \dots W$, $i = 1 \dots N$, und P_i - Lage des Objekts i :

$$P_i = \{p_{i1}, p_{i2}, \dots, p_{iL}\} \quad (3.1)$$

$p_{i1}, p_{i2}, \dots, p_{iL}$ - Lageparameter des Objekts i ; L - Anzahl der Lageparameter.

3.1.2 Prinzipien der hierarchischen iterativen Methode für die Objekt- und Lageerkennung

Die Aufgabe der Objekt- und Lageerkennung kann zwar mit fast jeder aktuellen Methode gelöst werden, allerdings hat die Analyse von Erkennungsmethoden, die im Kapitel 2

durchgeführt wurde, gezeigt, dass keine Methode alle Anforderungen erfüllt, die in Tabelle 2.1 angegeben sind. Somit ist es nötig, eine Methode zu entwickeln, die alle Anforderungen aus der Liste erfüllt.

Die in dieser Arbeit entwickelte hierarchische iterative Methode basiert auf dem klassischen Matching-Verfahren, was die Verwendung aller Merkmale eines Objektes ermöglicht und die Stabilität der Erkennung beim Rauschen und Verzerrungen sichert. Das Hauptprinzip der vorgeschlagenen Methode ist die Suche in 2D und 3D unstrukturierten Daten nach bestimmten Objekten, die durch spezielle Modell-Prototypen gegeben sind. Die Kombination des Matching-Verfahrens mit Optimierungsalgorithmen ermöglicht ihrerseits die Erhöhung der Geschwindigkeit der Objekt- und Lageerkennung.

Das Matching-Verfahren wurde als Basismethode ausgewählt, weil dieses Verfahren folgende Anforderungen erfüllt:

- Anforderungen 1 und 2 an die Ergebnisse der Erkennung: als Ergebnisse werden nicht nur Klasse des Objekts sondern auch die Lage dieses Objekts in der Szene ausgegeben.
- Anforderung 7 an die Geschwindigkeit der Erkennung: das Matching-Verfahren ist in der Lage, das Objekt und seine Lage in der Szene gleichzeitig zu erkennen.
- Anforderungen 8-11 an den Algorithmus zur Merkmalsextraktion. Diese Anforderungen können nur mit dem objektbasierten Matching-Verfahren erfüllt werden. Das merkmalsbasierte Matching-Verfahren erfüllt diese Anforderungen nicht.
- Anforderungen 15-16 an die Zuverlässigkeit: Invarianz gegen Position und Orientierung des zu erkennenden Objekts, sowie gegen affine Transformationen.

Der Algorithmus des klassischen Matching-Verfahrens ist in Abbildung 3.1 dargestellt.

Obwohl das klassische Matching-Verfahren wesentliche Anforderungen an die Erkennungsmethode erfüllt, hat er noch eine Reihe von Nachteilen:

- rechenintensive Methode
- starke Empfindlichkeit gegenüber Verzerrungen der zu erkennenden Objekte

Darüber hinaus werden auch die folgenden Anforderungen nach Tabelle 2.1 nicht erfüllt: 3-6, 12-14 und 17.

Die Beseitigung der Nachteile, sowie die Erfüllung aller Anforderungen an die Erkennungsmethode wurden durch die Modifikationen dieses Verfahrens, die in den nächsten Kapiteln beschrieben sind, erreicht.

3.2 Algorithmus der hierarchischen iterativen Methode für die Objekt- und Lageerkennung

Die im Kapitel 3.1.1 beschriebene Aufgabe kann mit Hilfe der hierarchisch-iterativen Methode folgendermaßen gelöst werden. Die Daten (R^K) und der Prototyp (M_i) der Klasse ω_i

3.2 Algorithmus der hierarchischen iterativen Methode für die Objekt- und Lageerkennung

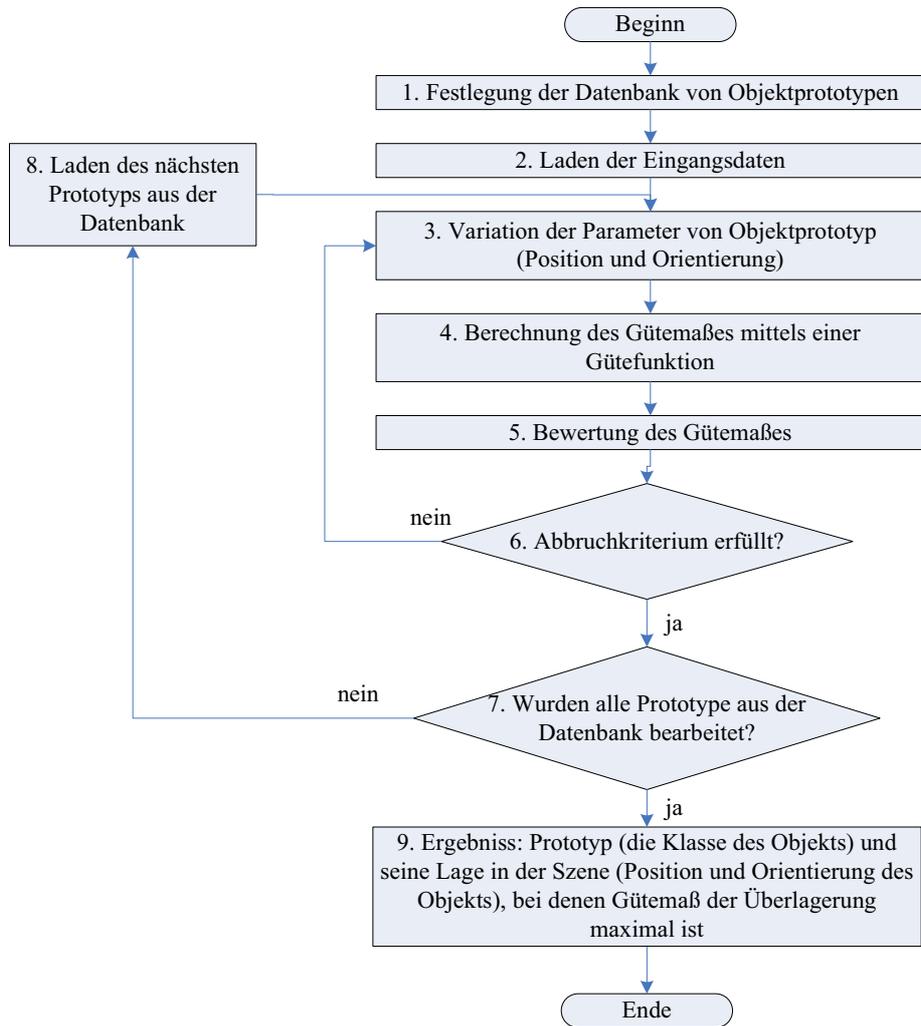


Abbildung 3.1: Algorithmus des klassischen Matching-Verfahrens

können durch hierarchische Modelle gegeben werden. D.h. das Modell ist die hierarchische Anordnung von Datendarstellungen (z.B., Voxelmmodell A , Polygonmodell B , etc.):

$$\begin{aligned}
 M_i &= \{\{A_1^{M_i}, B_1^{M_i}, \dots\}, \dots, \{A_{N_P}^{M_i}, B_{N_P}^{M_i}, \dots\}\} \\
 R^K &= \{\{A_1^{R^K}, B_1^{R^K}, \dots\}, \dots, \{A_{N_P}^{R^K}, B_{N_P}^{R^K}, \dots\}\}
 \end{aligned}
 \tag{3.2}$$

mit $\{A_k^i, B_k^i, \dots\}$ - Stufe der Pyramidendarstellung k ; N_P - Anzahl der Stufen von Pyramidendarstellung.

Eine Pyramide ist die hierarchische Darstellung der Daten, welche auf der Basis (Stufe N_P) die Originaldaten, auf anderen Stufen Daten mit niedrigerer Auflösung und Detaillierung und auf der Spitze die Daten mit minimaler Auflösung und Detaillierung (Stufe 1) enthält.

3. Hierarchisch-iterative Methode für die Objekt- und Lageerkennung

Ein Beispiel einer solchen Pyramide kann die Gauss-Pyramide für 2D-Bilder sein. Jede Stufe dieser Pyramide wird mittels Gauss-Filterung berechnet und deren Datenmenge ohne den Verlust von wesentlicher Information um den Faktor 2 reduziert.

Um die Güte der Überlagerung des Prototyps mit dem zu erkennenden Objekt zu bewerten, wird ein Gütekriterium $F_{i,m}$, das durch die Gütefunktion $f(X_i, R^K, P_m)$ berechnet wird, verwendet (P_m hier ist die aktuelle Position und Orientierung des Prototyps in der Szene).

Die Gütefunktion kann man folgendermaßen definieren:

$$f(M_i, R^K, P_m) = \check{f}(\bar{f}_1(M_i, R^K, P_m), \bar{f}_2(M_i, R^K, P_m), \dots) \quad (3.3)$$

mit $\bar{f}_1, \bar{f}_2, \dots$ - Gütefunktionen niedriger Ordnung, die einzelne Komponenten des Datenmodells verwenden (z.B. nur Voxellmodell oder Polygonmodell).

In der Tabelle 3.1 sind alle Eingangsparameter des Algorithmus der hierarchisch-iterativen Methode dargestellt.

Tabelle 3.1: Eingangsparameter des Algorithmus der hierarchisch-iterativen Methode für die Objekt- und Lageerkennung

Parameter	Beschreibung
$\{G_1, G_2, \dots, G_{N_G}\}$	Parametersätze der Filter (z.B. Glättungsfilter, Gradientenfilter) für die Vorverarbeitung der Daten
N_G	Anzahl an Parametersätze der Filter für die Vorverarbeitung
N_P	Anzahl der Pyramidenstufen
$\{M_1, M_2, \dots, M_W\}$	Prototypen der zu erkennenden Klassen
W	Anzahl an Prototypen der zu erkennenden Klassen
F_{min}	Minimaler Wert des Gütekriteriums. Das Objekt ist als nicht erkannt bestimmt, wenn die Güte der Überlagerung niedriger als dieser Wert ist.

Der Algorithmus der entwickelten Methode ist in Abbildung 3.2 schematisch dargestellt.

In der Tabelle 3.2 sind alle Schritte des Algorithmus detailliert erklärt.

3.2 Algorithmus der hierarchischen iterativen Methode für die Objekt- und Lageerkennung

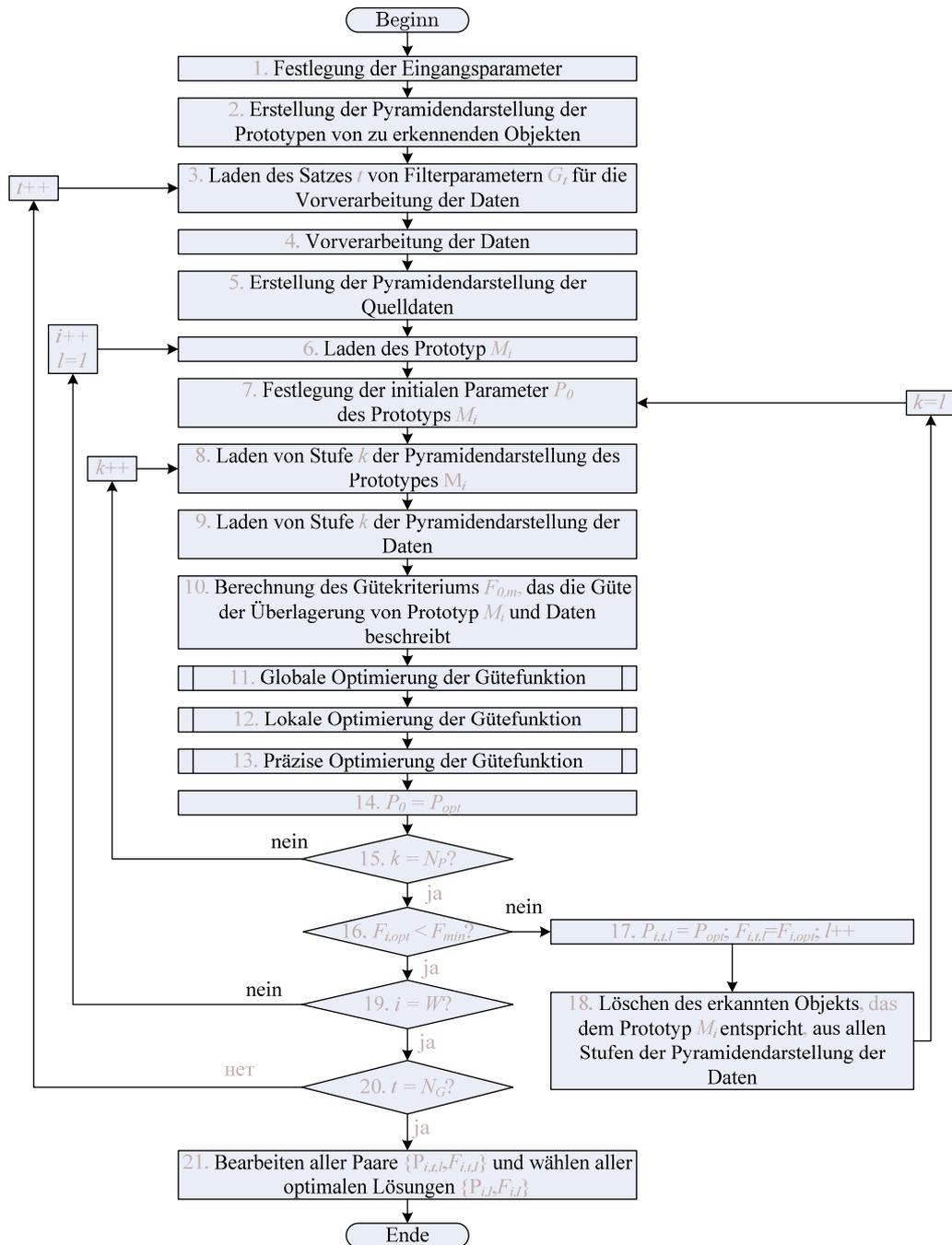


Abbildung 3.2: Algorithmus der hierarchisch-iterativen Methode für die Objekt- und Lageerkennung

Tabelle 3.2: Schritte des Algorithmus der hierarchischen iterativen Methode für die Objekt- und Lageerkennung

N.	Schritt	Beschreibung
1	Die Eingangsparameter $\{G_1, G_2, \dots, G_{N_G}\}$, N_G , N_P , $\{M_1, M_2, \dots, M_W\}$, F_{min} und Zwischenvariablen $i = t = k = l = 1$ werden festgelegt.	Die Festlegung wird manuell durchgeführt. Die Prototypen können als CAD-Daten beschrieben werden. Die Filter für die Vorverarbeitung der Daten hängen von den zu bearbeitenden Daten ab
2	Die Pyramidendarstellung für jeden Prototyp M_i mit N_P Stufen wird erstellt	Eine Pyramidendarstellung der Eingangsdaten oder des Objektprototyps ist eine hierarchische Struktur der Darstellungen mit verschiedener Auflösung und Detaillierung
3	Der Satz t von Filterparametern wird geladen (G_t)	
4	Die Eingangsdaten werden vorverarbeitet	Die Vorverarbeitung wird mit den Filtern und Parametern, die im Schritt 3 geladen wurden, durchgeführt
5	Pyramidendarstellung der Eingangsdaten wird erstellt	siehe Schritt 2
6	Der Prototyp M_i wird aus der Datenbank geladen	Es ist anzunehmen, dass Objekte der Klasse i in den Daten vorhanden sind
7	Initiale Parameter P_0 des Prototyps M_i werden festgelegt	Als Parameter werden räumlich-lineare Parameter (Translationen entlang x -, y - und z -Achse), räumlich-axiale (Rotationen um x -, y - und z -Achse) und nicht-räumliche (Verzerrungsfaktoren, etc) verwendet
8	Die Stufe k der Pyramidendarstellung des Prototyps M_i wird geladen	Die Stufe N_P ist die Basis der Pyramide, d.h. originale Daten. Die Stufe 0 ist die größte Darstellung der Daten
9	Die Stufe k der Pyramidendarstellung der Eingangsdaten wird geladen	siehe Schritt 8
10	Das Gütekriterium $F_{i,0}$ wird berechnet	Die Gütefunktion nimmt als die Argumente die Parameter des Prototyps (P_0) an. Der Wert dieser Funktion (Gütekriterium) beschreibt die Güte der Überlagerung

11	Globale Optimierung der Gütefunktion	Es wird nach einem Parametersatz des Prototyps gesucht, bei dem das Gütemaß maximal (minimal) ist siehe Schritt 11
12	Lokale Optimierung der Gütefunktion	siehe Schritt 11
13	Präzise Optimierung der Gütefunktion	Der gefundene Parametersatz wird gespeichert
14	$P_0 = P_{opt}$	Wenn alle Stufen der Pyramidendarstellungen des Objektprototyps M_i und der Daten bearbeitet wurden, dann wird die Variable k um 1 erhöht und der Übergang zum Schritt 8 wird durchgeführt. Sonst zum Schritt 11. Am Ende müssen alle Stufen der Pyramidendarstellungen bearbeitet worden sein
15	Wenn $k = N_P$, dann zum Schritt 8 übergehen. Sonst zum Schritt 11	Wenn das Gütekriterium kleiner als den minimalen Wert des Gütekriteriums ist, wird das Objekt als nicht erkannt bewertet
16	Wenn $F_{i,opt}$ kleiner als F_{min} ist, wird der Übergang zum Schritt 19 durchgeführt. Sonst zum Schritt 17	Der gefundenen Parametersatz des Prototyps und der aktuelle Wert des Gütekriteriums werden gespeichert. l - Anzahl an die Objekte von der Klasse j , die schon erkannt wurden
17	Werden folgende Operationen durchgeführt: $P_{i,t,l} = P_{opt}$ und $F_{i,t,l} = F_{i,opt}$, sowie l wird um 1 erhöht	Dies vermeidet die Wiedererkennung von bereits erkannten Objekten
18	Das erkannte Objekt, das dem Prototyp M_j entspricht, wird aus allen Stufen der Pyramidendarstellung der Eingangsdaten entfernt. k wird um 1 erhöht. Der Übergang zum Schritt 7 wird durchgeführt	Wenn alle Prototypen W verwendet wurden, wird der Übergang zum Schritt 20 durchgeführt. Sonst wird j um 1 erhöht und der Übergang zum Schritt 6 durchgeführt
19	Wenn $i = W$, dann zum Schritt 20 übergehen. Sonst zum Schritt 6	Wenn alle Parametersätze von Filtern G verwendet wurden, wird der Übergang zum Schritt 21 durchgeführt. Sonst wird i um 1 erhöht, Operation $l = 1$ und der Übergang zum Schritt 3 werden durchgeführt
20	Wenn $t = N_G$, dann zum Schritt 21 übergehen. Sonst zum Schritt 3	

21	Alle Paare $P_{i,t,l}, F_{i,t,l}$ werden bearbeitet und die optimale Lösung $P_{j,t}, F_{j,t}$ werden ausgewählt	Um nicht eindeutige Situationen zu lösen, werden alle erkannte Objekte nochmal bewertet. Die Auswahl wird durch den Vergleich von den Werten des Gütekriteriums durchgeführt
----	--	--

Die Vorteile des vorgeschlagenen Verfahrens sind die folgenden:

- Vereinfachung und Beschleunigung des Lernens: es reicht aus, nur die Gütefunktionen auszuwählen und ihre Zusammenhänge zu definieren (Anforderung 3 aus der Tabelle 2.1)
- Möglichkeit der Adaption zu realisieren: Anpassung der Gütefunktionparameter im Laufe des Erkennungsprozesses (Anforderung 4)
- Bei der Veränderung von Daten und zu erkennenden Objekten reicht es aus, die zusätzlichen Gütekriterien zu definieren, die Parameter von Gütefunktionen anzupassen und den Prototyp von neuen Objektklassen in die Datenbank zu legen (Anforderung 5)
- Objekterkennung in Echtzeit: z.B. im System für die Schrifterkennung und für das Griff-in-die-Kiste-System (Anforderung 6)
- Erkennung möglich in den verrauschten Daten deformierter und verzerrter Objekte (Anforderung 13 und 14)
- Vereinfachung der Merkmalsextraktion: als der Prototyp einer Objektklasse wird ein reales Objekt dieser Klasse benutzt (Anforderung 12)
- Die durchgeführten Experimente, die im Folgenden beschrieben werden, haben gezeigt, dass die Systeme, die auf der hierarchischen iterativen Methode basieren, alle Anforderungen an die Genauigkeit der Lageerkennung erfüllen (Anforderung 17)

3.3 Algorithmen der hierarchischen iterativen Methode für die Objekt- und Lageerkennung

Um die Genauigkeit und die Geschwindigkeit der Objekterkennung zu erhöhen, wurden folgende Algorithmen und Prozeduren entwickelt, die Bestandteil der hierarchischen iterativen Methode sind (s. Abb. 3.3):

- Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen. Dieser Algorithmus realisiert die gleichzeitige Optimierung einer oder mehrerer Gütefunktionen durch die Suche nach einem globalem Optimum in drei Schritten (globale, lokale und präzise Suche) bei der Verwendung von Kombination der räumlich-linearen, räumlich-axialen und parametrischen Argumenten der Funktion.
- Algorithmus des hierarchischen Matchings. Dieser Algorithmus realisiert das Matching des Prototyps eines Objektes mit den Daten, die als Pyramide dargestellt sind. Jede Stufe dieser Pyramide enthält diese Daten in verschiedener Auflösung und Detaillierung ohne Verlust wesentlicher Information.
- Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten. Dieser Algorithmus realisiert die Verwendung von Distanz- oder Zeigermatrizen bei der Berechnung von Gütekriterien, die die Suche des nächsten Elementes der Daten zu jedem Element des Objektprototyps fördern.

3. Hierarchisch-iterative Methode für die Objekt- und Lageerkennung

- Algorithmus der Deformation des Objektprototyps. Dieser Algorithmus realisiert die Deformation der Prototypen von Objekten bei dem Matching dieser Prototypen mit den Daten, die verrauscht und verzerrt sein können.
- Algorithmus des iterativen Matchings. Dieser Algorithmus realisiert den iterativen Prozess des Matchings von Prototypen des zuerkennenden Objekts mit den Daten bei der Änderung der Filterparameter, die für die Vorverarbeitung der Daten verwendet sind, während jeder Iteration.
- Algorithmus der Hierarchisierung von Objektprototypen. Dieser Algorithmus realisiert die Berücksichtigung der Gewichte von Bestandteilen des Objektprototyps, sowie die Zerlegung dieses Prototyps in die Komponenten bei der Berechnung der Gütekriterien.

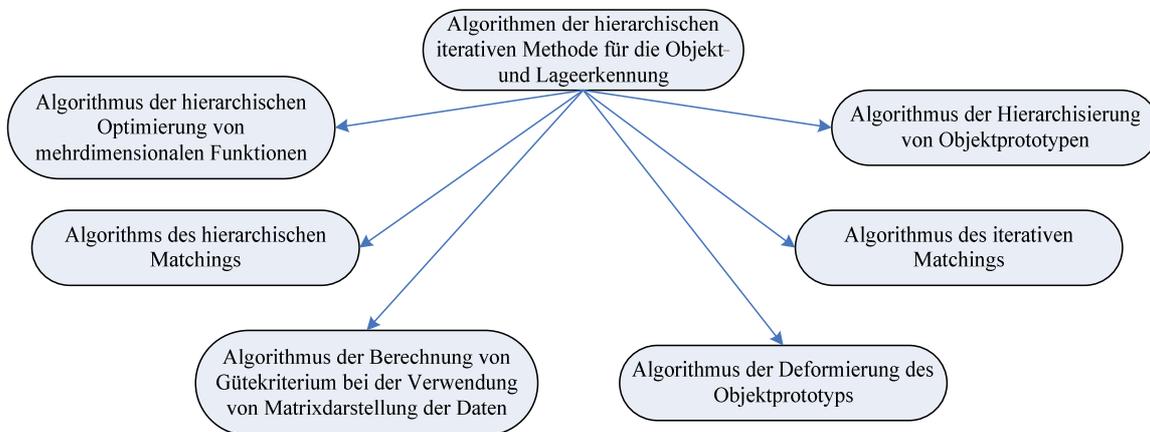


Abbildung 3.3: Algorithmen der hierarchischen iterativen Methode für die Objekt- und Lageerkennung

3.3.1 Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen

Die hierarchisch-iterative Methode für die Objekt- und Lageerkennung basiert auf Matching-Verfahren, d.h. die Objekterkennung besteht im Vergleich des Objektprototyps, der vorgegeben ist, mit dem zu erkennenden Objekt. Die Güte der Überlagerung beschreibt ein Gütekriterium, das mit der Hilfe einer mehrdimensionalen Gütefunktion berechnet wird, die als Argumente verschiedene Parameter des Objektprototyps annimmt, z.B. die Lage des Prototyps im Raum durch x -, y - und z -Translationen und Rotationen, Verzerrungsfaktoren, usw.).

Bei der Erkennung eines Objektes in mehrdimensionalen Daten, das durch einen Prototyp vorgegeben ist, ist es notwendig, die Werte der Gütefunktionen bei allen möglichen Kombinationen der Prototypparameter zu berechnen. Dieser rechenintensive Prozess kann durch der Verwendung von Optimierungsmethoden vereinfacht werden (z.B. Powell-Methode, Downhill-Simplex-Methode, Newton-Methode, u.a.). Eine solche Methode optimiert eine

mehrdimensionale Gütefunktion, die die Güte der Überlagerung beschreibt, durch die Prüfung ihrer Werte mit der beschränkten Menge der Prototypparameter, so dass der Prozess des Vergleichs wesentlich beschleunigt wird.

Es gibt eine Reihe von Optimierungsverfahren. Die verbreiteten Methoden lassen sich in zwei Klassen ([91, 3, 74, 18]) einteilen:

1. Deterministische Methoden: Newton-Methode, konjugierte Gradienten, Brent-Methode, Downhill-Simplex-Verfahren, usw.
2. Stochastische Methoden: simulierte Abkühlung, Bergsteigeralgorithmen, evolutionäre Algorithmen, neuronale Netze, usw.

Jede Klasse von Optimierungsmethoden hat sowohl Vorteile als auch Nachteile. Als Optimierungsmethode, auf deren Basis der Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen entwickelt wurde, wurde die Powell-Methode gewählt. Die Vorteile dieses Algorithmus sind folgende:

- Die Powell-Methode gehört zu der Klasse deterministischer Methoden, d.h. in dieser Methode werden keine Zufallsvariablen verwendet. Dies führt zur genauen Kontrolle über den Optimierungsablauf.
- Es ist keine Berechnung von Ableitungen notwendig.
- Die Powell-Methode ist sehr effizient für die Suche nach einem Minimum/Maximum.

Der Algorithmus der Suche nach einem Minimum/Maximum von mehrdimensionalen Funktionen mit der Hilfe des Powell-Algorithmus ist in der Abbildung 3.4 dargestellt.

Der Powell-Algorithmus optimiert eine mehrdimensionale Funktion folgendermaßen:

1. Der initiale Satz der Gütefunktionsargumente, d.h. die initialen Parameter eines Objektprototyps, wird ausgewählt.
2. Eine Optimierung entlang einer einzelnen Koordinate mit Hilfe der Algorithmen für eindimensionale Optimierung (z.B. Brent-Methode, Methode der goldenen Schnitt) wird durchgeführt.
3. Wenn die Position, in der keine weitere Optimierung entlang der ausgewählten Koordinate möglich ist oder die Anzahl von möglichen Iterationen überschritten wurde, erreicht wurde, wird die Optimierung entlang der nächsten Koordinate durchgeführt.
4. Die Suche nach einem Extremum ist abgeschlossen, wenn keine weitere Optimierung entlang jeder Koordinate möglich ist.

Bei der Entwicklung dieses Algorithmus wurden folgende Anforderungen berücksichtigt:

- Möglichkeit der Optimierung einer diskreten Gütefunktion. Normalerweise werden die Optimierungsmethoden für die Suche eines Extremums von kontinuierlichen Funktionen verwendet.

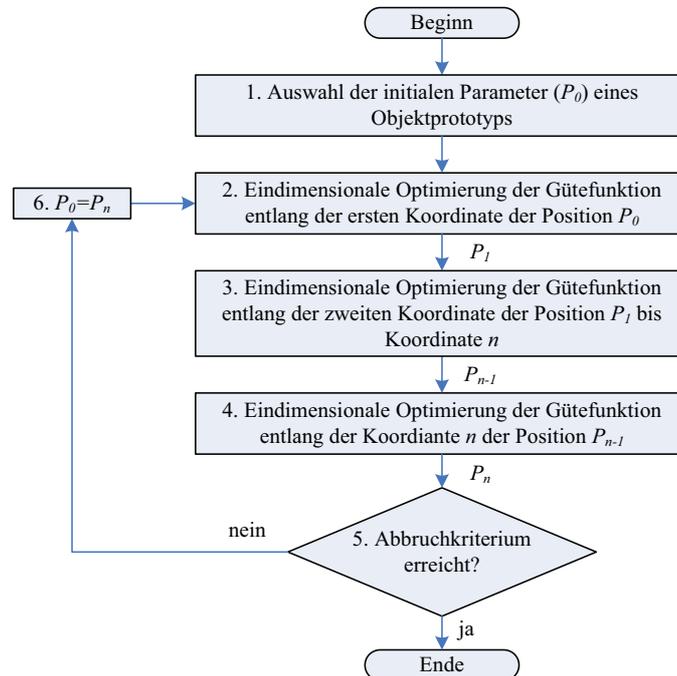


Abbildung 3.4: Powell-Algorithmus zur Optimierung von mehrdimensionalen Funktionen

- Erhöhung der Wahrscheinlichkeit des Auffindens eines globalen Extremums statt eines lokalen.

Nach der Analyse der oben genannten Anforderungen wurden folgende Algorithmen-Erweiterungen entwickelt:

1. Hierarchische Optimierung der Gütefunktion.
2. Verwendung von Kombination der räumlich-linearen, räumlich-axialen und parametrischen Argumenten der Gütefunktion.
3. Verwendung von parametrischen Argumenten (keine räumlichen Koordinaten).
4. Multikriterielle Optimierung.

Hierarchische Optimierung der Gütefunktion

Dieser Algorithmus basiert auf dem Prinzip der Optimierung der Gütefunktion in einigen Schritten, d.h. die Suche des Extremums wird iterativ durchgeführt. Die Optimierung beginnt mit dem groben Auffinden der möglichen Extrema der Funktion (sowohl globale als auch lokale) und beendet nach der Bestimmung des globalen Extremums der Gütefunktion.

Die hierarchische Optimierung von Gütefunktion enthält folgende Stufen:

- globale Suche

- lokale Suche
- präzise Suche

In Abbildung 3.5 sind Komponenten jeder Stufe der hierarchischen Optimierung dargestellt.

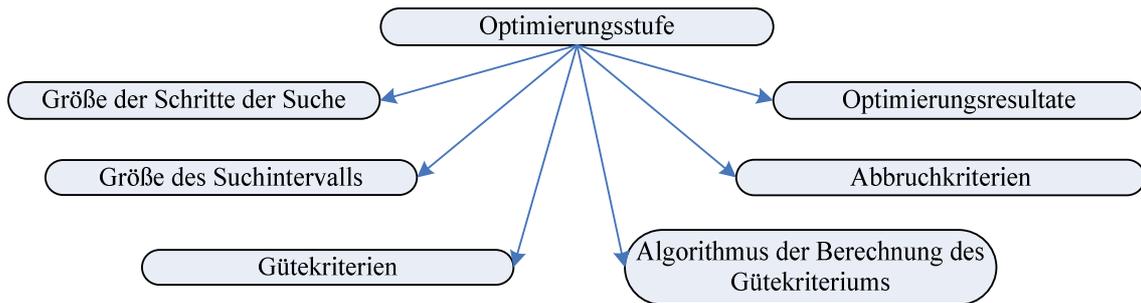


Abbildung 3.5: Komponenten einer Stufe der hierarchischen Optimierung

Drei Stufen der Optimierung unterscheiden sich voneinander durch folgende Komponenten:

- **Schrittweite der Suche und Größe des Suchintervalls.** Bei der globalen Suche ist der Abstand zwischen zwei Suchpunkte entlang einer Koordinate größer als bei der lokalen Suche. Bei der präzisen Suche wird ein Extremum nur in einer kleinen Umgebung des Punktes eines möglichen Extremums, das bei der lokalen Suche gefunden wurde, gesucht.
- **Gütekriterien.** Bei einer Stufe der Optimierung ist es möglich, zusätzliche Gütekriterien zu verwenden.
- **Abbruchkriterien.** Unter Abbruchkriterien des Optimierungsvorganges versteht man folgende Werte:
 - maximale Anzahl der Iterationen der eindimensionalen Optimierung
 - maximale Anzahl der Iterationen der mehrdimensionalen Optimierung
 - Differenz zwischen den Gütemaß bei der aktuellen und der letzten Optimierungssiterationen
- **Algorithmus der Berechnung von Gütekriterien.** z.B. wird für die Beschleunigung der Güteberechnung bei der globalen Suche eine Distanzmatrix verwendet. Und bei der präzisen Suche wird eine Zeigermatrix verwendet (siehe 3.3.3).
- **Optimierungsergebnisse.** Resultat der globalen Suche ist ein Satz von Punkten, die möglichen Extrema der Gütefunktion darstellen. Resultat der präzisen Suche ist ein Punkt, d.h. globales Extremum.

3. Hierarchisch-iterative Methode für die Objekt- und Lageerkennung

In der Abbildung 3.6 ist den Algorithmus der hierarchischen Optimierung einer modellierten Gütefunktion dargestellt.

Die Vorgehensweise der hierarchischen Optimierung sieht folgendermaßen aus:

1. Die erste Stufe der Optimierung ist die globale Suche. Resultat dieser Suche ist der Satz von Punkten, in denen die Wahrscheinlichkeit des Auffindens eines globalen Extremums relativ hoch ist.
2. Die zweite Stufe ist die lokale Suche, die mit kleineren Schritten in den Umgebungen von Punkten, welche bei der globalen Suche gefunden wurden, durchgeführt wird. Resultat dieser Suche ist der Punkt, in dem das globale Extremum mit hoher Wahrscheinlichkeit zu finden ist.
3. Die letzte Stufe ist die präzise Suche, die Suchintervalle bei diesem Schritt sind sehr klein. Die Suche wird nur in der Umgebung des Punktes durchgeführt, der bei der lokalen Suche gefunden wurde. Resultat ist der präzisere Punkt des globalen Extremums.

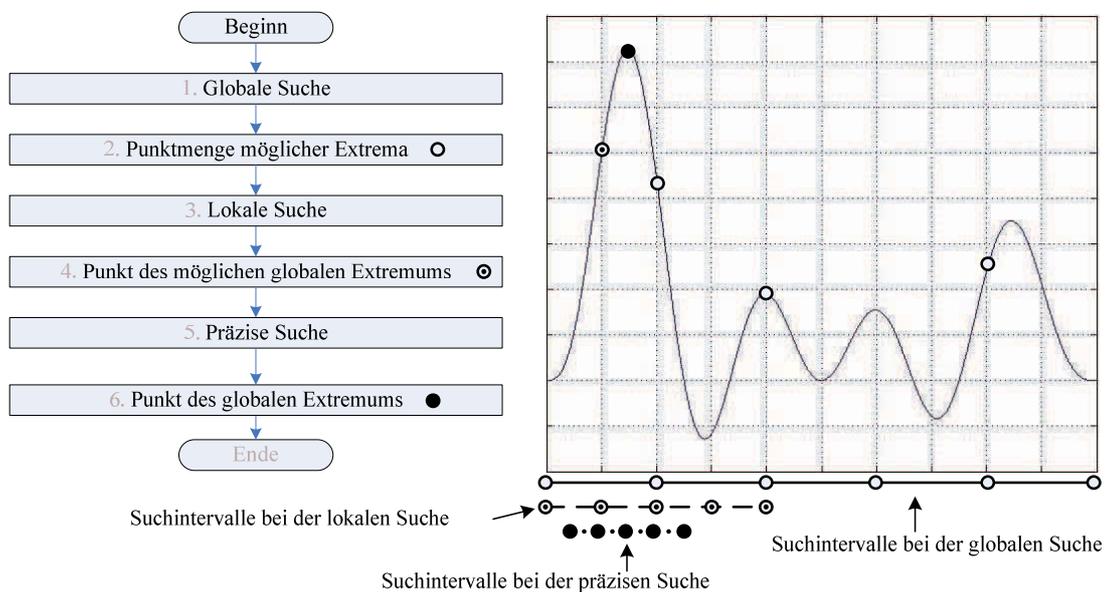


Abbildung 3.6: Algorithmus der hierarchischen Optimierung der Gütefunktion

Die hierarchische Optimierung ermöglicht sowohl die Erhöhung der Wahrscheinlichkeit für das Auffinden eines globalen Extremums als auch die Reduzierung des Aufwands für die Optimierung einer Gütefunktion.

Verwendung von Kombination der räumlich-linearen, räumlich-axialen und parametrischen Argumenten der Gütefunktion

Die klassische Powell-Methode wurde für die Optimierung von mehrdimensionalen Funktionen bei der Verwendung von linear unabhängigen Koordinaten entwickelt. Z.B. jede Iteration der Optimierung einer Gütefunktion mit 6 Koordinaten (3 Rotationen um x -, y - und z -Achsen und 3 Translationen entlang x -, y - und z -Achsen) durch die klassische Powell-Methode schließt eine Optimierung dieser Funktion entlang aller 6 Koordinaten ein. Die durchgeführten Experimente haben gezeigt, dass die Verwendung der Kombinationen von linear abhängigen Koordinaten (räumlich-lineare, räumlich-axiale und parametrische) die Effizienz der Optimierung erhöht. Z.B. bei der Tomogrammregistrierung werden auch die folgende Kombinationen der Koordinaten verwendet: (x, y) , $(x, -y)$, (x, z) . Der Einsatz dieses Verfahrens führt zur wesentlichen Reduzierung der Optimierungsdauer, sowie zur Erhöhung der Wahrscheinlichkeit des Auffindens eines globalen Extremums.

Verwendung von parametrischen Koordinaten des Objektprototyps

Beim Matching des Prototyps mit zu erkennendem Objekt werden in den Lösungen vieler Aufgaben Gütefunktionen optimiert, die 6 Koordinaten des Prototyps als Argumente annehmen: 3 Translationen (räumlich-lineare Koordinaten) und 3 Rotationen (räumlich-axiale Koordinaten), d.h. alle diese Koordinaten sind räumliche Koordinaten. Bei der Optimierung solcher Gütefunktion reicht es aus, ein Extremum entlang dieser räumlichen Koordinaten zu finden. Für die Erhöhung der Genauigkeit der Objekt- und Lageerkennung werden in dem entwickelten Algorithmus nicht nur räumliche, sondern auch die parametrische Koordinaten des Objektprototyps bei der Optimierung benutzt (Skalierungs- und Verzerrungskoeffizienten):

$$F = f(d_1, d_2, \dots, d_k, d_{k+1}, \dots, d_l, d_{l+1}, \dots, d_n) \quad (3.4)$$

mit f - Gütefunktion, (d_1, \dots, d_k) - räumlich-lineare Koordinaten, (d_{k+1}, \dots, d_l) - räumlich-axiale Koordinaten, (d_{l+1}, \dots, d_n) - nicht-räumliche parametrische Koordinaten.

Der Einsatz dieses Verfahrens ermöglicht die Reduzierung des Einflusses von Rauschen und Verzerrungen auf die Objekt- und Lageerkennung.

Multikriterielle Optimierung

Der Einsatz von mehreren Gütekriterien beim Matching des Prototyps mit dem zu erkennenden Objekt führt zur Erhöhung der Genauigkeit der Erkennung. Solche Gütekriterien müssen verschiedene Information benutzen, die aus den Eingangsdaten extrahiert werden können. Also müssen sie unabhängig sein. Diese Kriterien müssen mit der Hilfe von Funktionen berechnet werden können, die mehrdimensional sind und dieselben Koordinaten als Argumente verwenden.

Mehrere Gütekriterien können folgendermaßen berechnet werden:

- Verwendung verschiedener Gütekriterien in den bestimmten Suchschritten nachein-

ander, z.B. bei der globalen Suche wird ein Kriterium K_1 , bei der lokalen Suche K_2 und bei der präzisen Suche K_3 verwendet.

- Gleichzeitige Berechnung der Gütekriterien. In diesem Fall wird die Optimierung einer mehrdimensionalen Funktion (kombinierte Funktion) durchgeführt, welche auf mehreren Gütefunktionen nach einer Regel basiert.

Nehmen wir an, gegeben sind die folgende Gütefunktionen f_1, f_2, \dots, f_m :

$$\begin{aligned} F_1 &= f_1(d_1, d_2, \dots, d_n) \\ F_2 &= f_2(d_1, d_2, \dots, d_n) \\ &\dots = \dots \\ F_m &= f_m(d_1, d_2, \dots, d_n) \end{aligned} \quad (3.5)$$

Dadurch sieht die kombinierte Gütefunktion folgendermaßen aus:

$$\begin{aligned} F &= f(d_1, d_2, \dots, d_n) = \\ &= f(f_1(d_1, d_2, \dots, d_n), f_2(d_1, d_2, \dots, d_n), \dots, f_m(d_1, d_2, \dots, d_n)) \end{aligned} \quad (3.6)$$

oder

$$F = f(f_1, \dots, f_m) = k_1 \cdot f_1 + k_2 \cdot f_2 + \dots + k_m \cdot f_m \quad (3.7)$$

mit den wählbaren Koeffizienten k_1, k_2, \dots, k_m .

Experimente unter Verwendung des entwickelten Algorithmus der hierarchischen Optimierung mehrdimensionaler Gütefunktionen

Für die Bewertung der Effizienz des entwickelten Optimierungsalgorithmus wurde eine Reihe von Experimenten unter Verwendung von verschiedenen Optimierungsalgorithmen und Methoden durchgeführt. Als zu vergleichende Algorithmen wurden die klassische Powell-Methode (gehört zu der Klasse von deterministischen Methoden) und den Algorithmus der simulierten Abkühlung (gehört zu der Klasse der stochastischen Methoden) ausgewählt.

In Abbildung 3.7 (Daten in Tabelle A.1 im Anhang) sind die experimentellen Ergebnisse dargestellt, die im Rahmen des Griff-in-die-Kiste-Systems für die Lösung der Aufgabe von Objekt- und Lageerkennung mit Hilfe der hierarchisch-iterativer Methode (HIMOLE) durchgeführt wurden.

Beim ersten Versuch wurde die hierarchisch-iterativen Methode ohne die Erweiterungsalgorithmen eingesetzt, welche die Powell-Methode und den Berechnungsalgorithmus für Gütekriterien mittels Analyse der Sphärendarstellung von Quelldaten [58] verwenden. Beim zweiten Versuch wurde die Powell-Methode durch den Algorithmus der simulierten Abkühlung und beim dritten Versuch durch den Algorithmus der hierarchischen Optimierung mehrdimensionaler Funktionen ersetzt.

Für den Vergleich wurden folgende Kriterien verwendet:

1. Durchschnittliche Zeit zur Erkennung eines Objektes (K_1)

2. Genauigkeit der Lageerkennung, die durch die euklidische Metrik der Translation von der berechneten und wirklichen Lage eines Objektes definiert ist (K_2):

$$K_2 = \sqrt{(x_0 - x)^2 + (y_0 - y)^2 + (z_0 - z)^2} \quad (3.8)$$

3. Genauigkeit der Lageerkennung, die durch die euklidische Metrik der Rotation von der berechneten und wirklichen Lage eines Objektes definiert ist (K_3):

$$K_3 = \sqrt{(\alpha_0 - \alpha)^2 + (\beta_0 - \beta)^2 + (\gamma_0 - \gamma)^2} \quad (3.9)$$

4. Anzahl der korrekten Erkennungen (in %), d.h. die Anzahl von Objekten, deren Klassen richtig erkannt wurden (K_4)

Als die experimentelle Daten wurden 5 Datensätze verwendet, von denen jeder 20 3D-Szenen mit 2-3 Objekte verschiedener Klassen je Szene enthält.

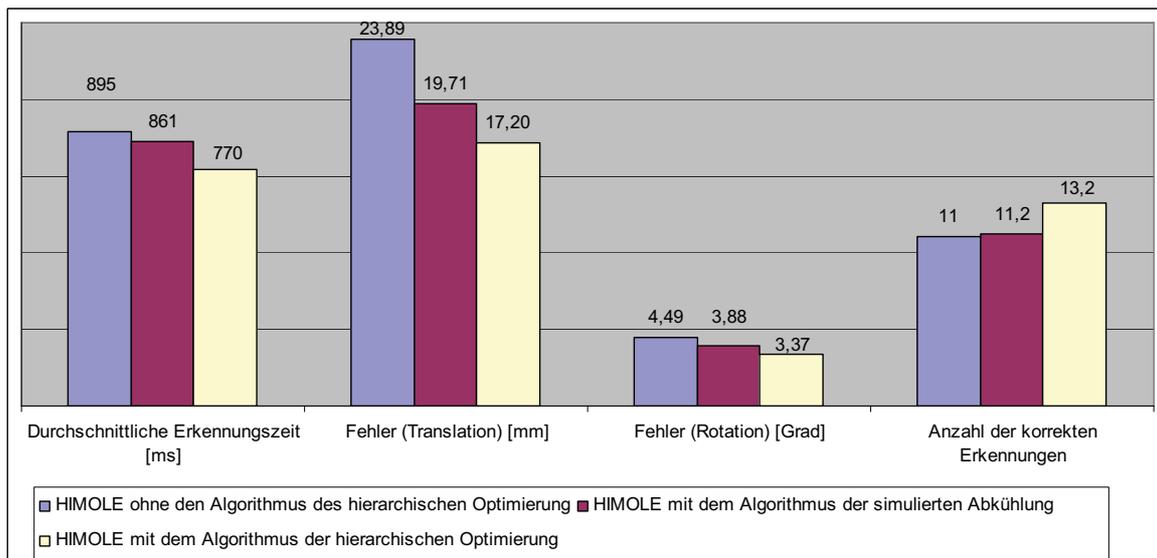


Abbildung 3.7: Experimentelle Ergebnisse (Hierarchische Optimierung)

Nach der Analyse der experimentellen Ergebnisse aus der Tabelle A.1 ergibt sich folgendes Bild:

- Der Einsatz des Algorithmus der simulierten Abkühlung hat ermöglicht, die durchschnittliche Erkennungszeit um 3,8% zu reduzieren, die Genauigkeit der Translations- und Rotationsbestimmung bei der Lageerkennung um 17,5% bzw. 13,5%, sowie die Anzahl der korrekten Erkennungen um 2% zu erhöhen.
- Der Einsatz des Algorithmus der hierarchischen Optimierung hat ermöglicht, die Erkennungszeit um 14% zu reduzieren, die Genauigkeit der Translations- und Rotationsbestimmung bei der Lageerkennung um 28% bzw. 24,8%, sowie die Anzahl der korrekten Erkennungen um 20% zu erhöhen.

3.3.2 Hierarchisches Matching

Die Vorgehensweise des hierarchischen Matchings ist wie folgt:

- Die Daten werden als eine Pyramide dargestellt. Jede Stufe dieser Pyramide enthält die Daten in verschiedenen Auflösungen (Detaillierung). An der Spitze dieser Pyramide (erste Stufe) befinden sich die Daten mit der kleinsten Auflösung und Detaillierung. Am Fußpunkt dieser Pyramide (letzte Stufe) liegen die Originaldaten.
- Der Vorgang des Matchings von Objektprototyp mit zu erkennendem Objekt (Eingangsdaten) beginnt auf der ersten Stufe der Pyramide. Auf dieser Stufe wird die grobe Position und Orientierung des Prototyps berechnet. Auf den nächsten Stufen wird die Position und Orientierung des Prototyps präziser festgestellt. Das Matching wird auf den letzten Stufe (originale Daten) beendet.

Die Algorithmen für die Berechnung jeder Stufe der Pyramiden hängt vom Typ der Daten ab, die in folgende Klassen unterteilt werden können:

- 2D- oder 3D-Daten, die Binärinformation enthalten (z.B. Gradientenbilder)
- 2D- oder 3D-Daten, die Grauwertinformation enthalten
- 2D- oder 3D-Daten, die Farbinformation enthalten (z.B. Farbbilder)

Für jede Datenklasse gibt es bestimmte Algorithmen für die Reduzierung der Auflösung und Detaillierung:

- Bei der Verarbeitung der 2D-Daten, die Grauwert- oder Farbinformation enthalten, werden die Laplace- und Gauß-Filter verwendet. Diese Filter ermöglichen die Komprimierung der Bilder ohne Verlust von wesentlicher Information [10].
- Bei der Verarbeitung der 3D-Daten, die Binärinformation enthalten, werden verschiedene Algorithmen der Dezimierung von Dreiecksnetzen verwendet.

Um die Speicheranforderungen für die Pyramidendarstellung der Daten zu berechnen, kann man die Formel der geometrischen Folge benutzen. Nehmen wir an, die Daten sind N -dimensional. Jede Stufe der Pyramide enthält die Daten, deren Auflösung jeweils um den Faktor 2^N zur vorherigen Stufe reduziert wurde. Die gesamte Pyramide besitzt M -Stufen.

Nach der Verwendung der Formel der geometrischen Folge ergibt sich folgende Gleichung:

$$S = (d_1 \cdot d_2 \cdot \dots \cdot d_N) \cdot \frac{\left(\frac{1}{2^N}\right)^M - 1}{\frac{1}{2^N} - 1} = (d_1 \cdot \dots \cdot d_N) \cdot \frac{2^N \cdot (2^{N \cdot M} - 1)}{2^{N \cdot M} \cdot (2^N - 1)} \quad (3.10)$$

Der zweite Faktor kann durch folgende Formel berechnet werden:

$$S_2 = (d_1 \cdot \dots \cdot d_N) \cdot \frac{2^N \cdot (2^{N \cdot M} - 1)}{2^{N \cdot M} \cdot (2^N - 1)} = 1 + \frac{1 - 2^{N \cdot (1-M)}}{2^N - 1} \quad (3.11)$$

Da $N \geq 1$ und $M \geq 1$, gilt es $2^N - 1 \geq 1$ bzw. $1 - 2^{N \cdot (M-1)} < 1$. Dadurch ergibt sich folgende Gleichung:

$$0 \leq \frac{1 - 2^{N \cdot (1-M)}}{2^N - 1} < 1 \quad (3.12)$$

Unter Berücksichtigung der Bedingung 3.12, erhält man folgende Gleichung:

$$1 \leq 1 + \frac{1 - 2^{N \cdot (1-M)}}{2^N - 1} < 2 \quad (3.13)$$

Die Bedingung 3.13 bestätigt, dass die Speicheranforderung für die Pyramidendarstellung der Daten sich um weniger als den Faktor 2 im Vergleich zu den originalen Daten erhöht.

Der Algorithmus des hierarchischen Matchings ist in Abbildung 3.8 dargestellt.

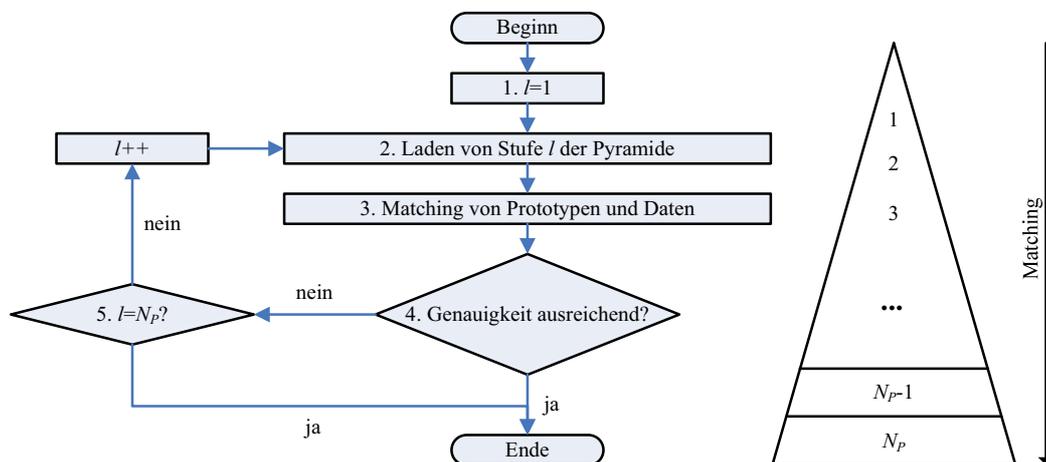


Abbildung 3.8: Algorithmus des hierarchischen Matchings

Der Einsatz des Algorithmus des hierarchischen Matchings ermöglicht die Reduzierung der Dauer der Optimierung einer mehrdimensionalen Funktion, weil bei der globalen Suche grobe Datenmodelle verwendet werden und erst bei der präzisen Suche ein Extremum in den originalen Daten gesucht wird. Die Reduzierung der Optimierungsdauer führt zur Reduktion der Erkennungszeit.

Da bei der globalen Optimierung grobe Datendarstellungen verwendet werden, wird die Gütefunktionen geglättet, d.h. sie verliert manche lokale Extrema. Dies führt zusätzlich zu einer Erhöhung der Genauigkeit der Lage- und Objekterkennung, sowie zur Erhöhung der Anzahl an korrekte Erkennungen.

Experimente unter Verwendung des entwickelten Algorithmus des hierarchischen Matchings

Für die Bewertung der Effizienz des entwickelten Algorithmus des hierarchischen Matchings wurde eine Reihe von Experimenten durchgeführt, deren Ergebnisse in Abbildung

3.9 (Daten in Tabelle A.2 im Anhang) dargestellt werden. Als zu vergleichende Methoden wurden die hierarchisch-iterative Methode (HIMOLE) ohne und mit dem Algorithmus des hierarchischen Matchings verwendet.

Wie im vorherigen Kapitel werden für den Vergleich sowohl dieselbe Kriterien K_1 , K_2 , K_3 und K_4 als auch dieselbe 5 Datensätze, jeder von denen 20 3D-Szenen mit 2-3 Objekte verschiedener Klassen in der Szene enthält, verwendet.

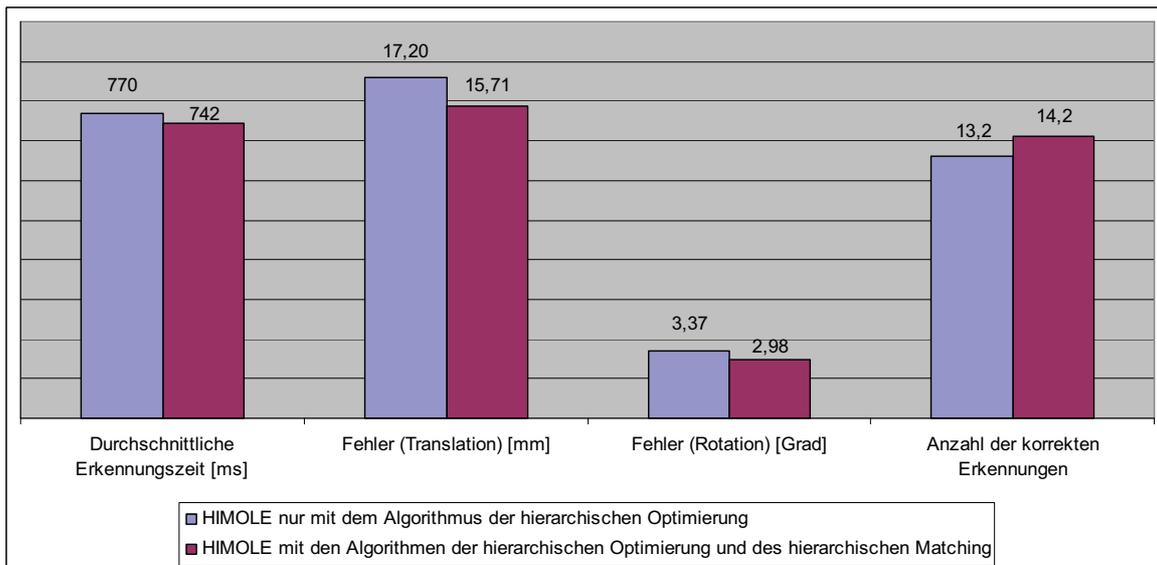


Abbildung 3.9: Experimentelle Ergebnisse (Hierarchisches Matching)

Die Analyse der experimentellen Ergebnisse aus der Tabelle A.2 zeigt: Der Einsatz des Algorithmus des hierarchischen Matchings ermöglicht hat, die Erkennungszeit um 3,6% zu reduzieren, die Genauigkeit der Translations- und Rotationsbestimmung bei der Lageerkennung um 8,9% bzw. 11,7%, sowie die Anzahl der korrekten Erkennungen um 7,6% zu erhöhen.

3.3.3 Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten

Um die Güte der Überlagerung des Objektprototyps mit den Daten (mit dem zu erkennenden Objekt) zu bewerten, ist es notwendig den Wert der Gütefunktion zu berechnen. Allerdings sind manche Gütefunktionen sehr rechenintensiv. Dies führt zur wesentlichen Reduzierung der Erkennungsgeschwindigkeit. Das liegt u.a. daran, dass zu bearbeitende Daten Millionen von Elementen (z.B. Punkte, Voxel, Polygone) enthalten können und der Erkennungsprozess iterativ durchgeführt wird, d.h. bei jeder Iteration findet eine neue Berechnung der Gütefunktion statt. Als solche Gütekriterien kann man folgende nennen:

- Summe der quadratischen Abständen von jedem Element des Prototyps zum nächsten Element der Szene.

- Korrelationskoeffizient oder Überdeckungsgrad des Prototyps und der Daten, d.h. Summe der Abstände von jedem Element des Prototyps zum nächstgelegenen Element der Szene, die kleiner als ein Wert δ sind.

Bei der Berechnung der oben genannten Gütekriterien werden 2D- oder 3D-Binärdaten behandelt. Als Elemente in diesen Daten können Pixel oder Punkte (2D-Daten) und Voxel oder Polygone (3D-Daten) verwendet werden.

Für die Berechnung der oben genannten Gütekriterien ist es notwendig, die Suche nach einem nächstgelegenen Element der Szene für jedes Element des Prototyps durchzuführen. Eine solche Suche kann man unter Einsatz der Brute-Force-Methode durchführen.

Auch im Fall der Verwendung einer ICP-Methode kann der Suchprozess nach den nächstgelegenen Elementen nicht vermieden werden. Mit der Absicht diesen Prozess in wenigen Iterationen durchzuführen, wurde ein Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten entwickelt. Die Vorgehensweise dieses Algorithmus sieht folgendermaßen aus:

1. Eine Matrix (2-, 3- oder mehrdimensional) wird erstellt. Jede Zelle dieser Matrix enthält die Information über die nächstgelegenen Elemente in den Daten.
2. Für jedes Element des Prototyps wird seine Position als Zelle in der erstellten Matrix bestimmt. Die Information aus dieser Zelle wird für die Suche nach dem nächstgelegenen Element verwendet.

Ein Beispiel der Erstellung der Matrixdarstellung der Daten für die Schrifterkennung (Matrixdarstellung für den zuerkennenden Text/Symbol) ist in der Abbildung 3.10 dargestellt.

Die Verwendung der Matrixdarstellung ermöglicht die wesentliche Reduzierung der Dauer der Berechnung des Gütekriteriums. Dabei ist es möglich, zwei Varianten der Matrixdarstellung zu verwenden. Diese Varianten unterscheiden sich in der Genauigkeit der Berechnung der Abstände zwischen Elementen des Prototyps und der Daten. Diese Varianten sind folgende:

- **Distanzmatrix.** Das Prinzip dieser Matrix besteht darin, dass jede Zelle den Abstand zum nächstgelegenen Element der Daten enthält (s. Abb. 3.10). Dabei werden beliebige unterschiedliche Elemente des Prototyps des zu erkennenden Objekts, die zu einer Zelle dieser Matrix gehören, von dem Algorithmus als ein Element angenommen. Für dieses Element wird dann ein Abstand, der in dieser Zelle gespeichert wird, dementsprechend berechnet. Die Verwendung einer solchen Matrix ist dann zweckmäßig, wenn die Dauer der Gütekriteriumberechnung sehr kritisch ist und der Verlust an Genauigkeit vernachlässigt werden kann.
- **Zeigermatrix.** Jede Zelle einer solchen Matrix enthält den Zeiger auf das nächstgelegene Element der Daten, d.h. die Berechnung des Abstandes zwischen Elementen wird zusätzlich durchgeführt unter Berücksichtigung der genauen Position des Prototypelementes. In diesem Fall ist der berechnete Abstand genauer im Vergleich zu dem Abstand, der mit Hilfe einer Distanzmatrix berechnet wird, obwohl es nur ein nächstgelegenes Element für die ganze Zelle der Zeigermatrix gibt. Somit ist die Verwendung der Zeigermatrix dann zweckmäßig, wenn die Genauigkeit der Berechnung sehr wichtig ist und die Anforderung an die Geschwindigkeit weniger kritisch ist.

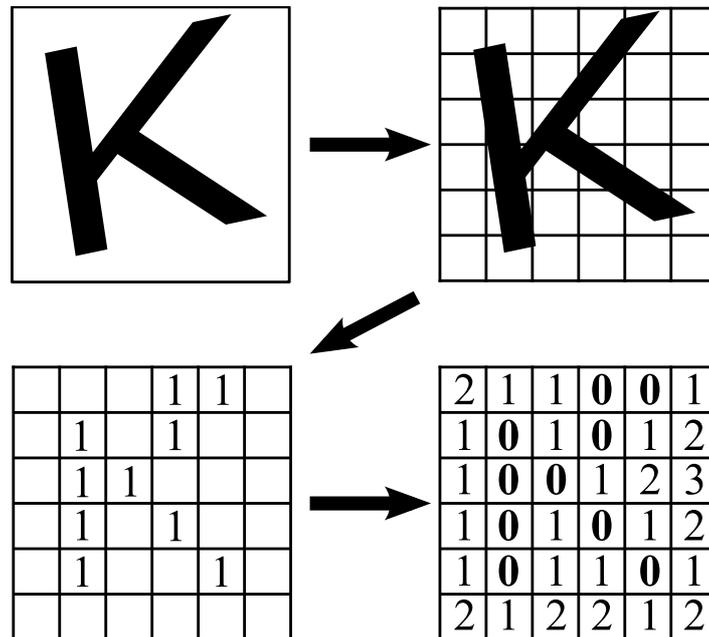


Abbildung 3.10: Erstellung einer Matrixdarstellung des zu erkennenden Textes

In Abbildung 3.11 sind die Beispiele von Distanz- und Zeigermatrizen dargestellt.

2	1	1	0	0	1
1	0	1	0	1	2
1	0	0	1	2	3
1	0	1	0	1	2
1	0	1	1	0	1
2	1	2	2	1	2

1	1	7	0 ₇	0 ₈	8
1	0 ₁	1	0 ₆	6	8
2	0 ₂	0 ₅	5	6	6
3	0 ₃	3	0 ₉	9	10
4	0 ₄	4	9	0 ₁₀	10
4	4	4	10	10	10

Abbildung 3.11: Distanzmatrix (links) und Zeigermatrix (rechts)

Der Prozess der Berechnung einer Matrixdarstellung der Daten ist ziemlich rechenintensiv. Um die Berechnungszeit zu reduzieren, wurde ein iterativer Algorithmus für das Ausfüllen der Matrixzellen (s. Abb. 3.12), entwickelt:

- Bei der ersten Iteration werden die Zellen mit Information (Abstände oder Zeiger) gefüllt, die alle Elemente (Punkte, Dreiecke, usw.) der Daten enthalten.
- Bei der zweiten Iteration werden die, zu den bereits ausgefüllten Zellen nächstliegenden, Zellen bearbeitet.
- Bei den darauffolgenden Iterationen werden die benachbarten Zellen mit Information gefüllt, so lange bis die ganze Matrix ausgefüllt ist.

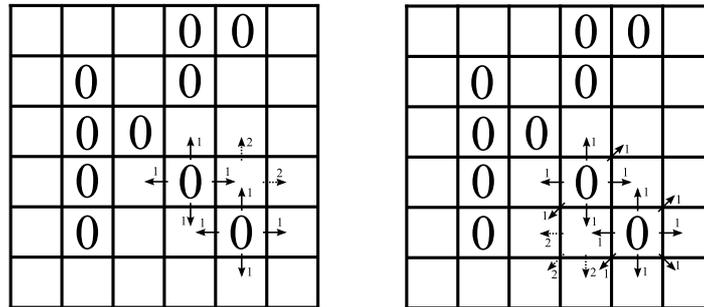


Abbildung 3.12: Algorithmus für das Ausfüllen der Matrixzellen

In der Abbildung 3.13 ist ein Beispiel der Berechnung von Gütekriterien mit Hilfe der Distanzmatrix beim Matching von Symbolprototyp mit dem zu erkennenden Text dargestellt.

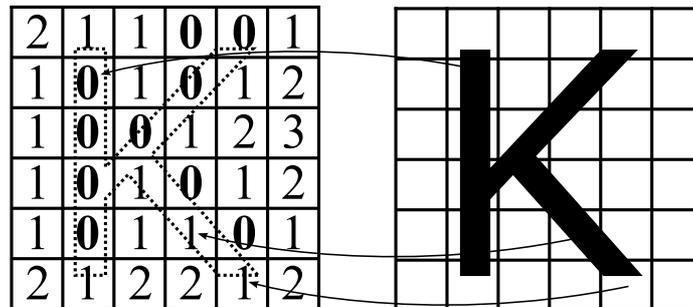


Abbildung 3.13: Berechnung des Gütekriteriums mit Hilfe von Distanzmatrix beim Matching des Symbolprototyps mit dem zu erkennenden Text

Im Beispiel, das in der Abbildung 3.13 dargestellt ist, die Güte der Überlagerung (in diesem Fall die Summe der quadratischen Abstände) ist: $F = 4$.

Die Verwendung des Algorithmus zur Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten ermöglicht die wesentliche Reduzierung der Berechnungszeit von Gütekriterien. Seinerseits führt dies zur wesentlichen Reduzierung der Erkennungszeit.

Experimente unter Verwendung des entwickelten Algorithmus zur Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten

Für die Bewertung der Effizienz des entwickelten Algorithmus zur Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten wurde eine Reihe von Experimenten durchgeführt, deren Ergebnisse in Abbildung 3.14 (Daten in Tabelle A.3 im Anhang) zusammengefasst sind. Als zu vergleichende Methoden wurden die hierarchisch-iterative Methode ohne und mit dem Algorithmus zur Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten verwendet.

Wie im vorherigen Kapitel werden für den Vergleich sowohl dieselben Kriterien K_1 , K_2 ,

3. Hierarchisch-iterative Methode für die Objekt- und Lageerkennung

K_3 und K_4 als auch dieselben 5 Datensätze, von denen jeder 20 3D-Szenen mit 2-3 Objekte verschiedener Klassen je Szene enthält, verwendet.

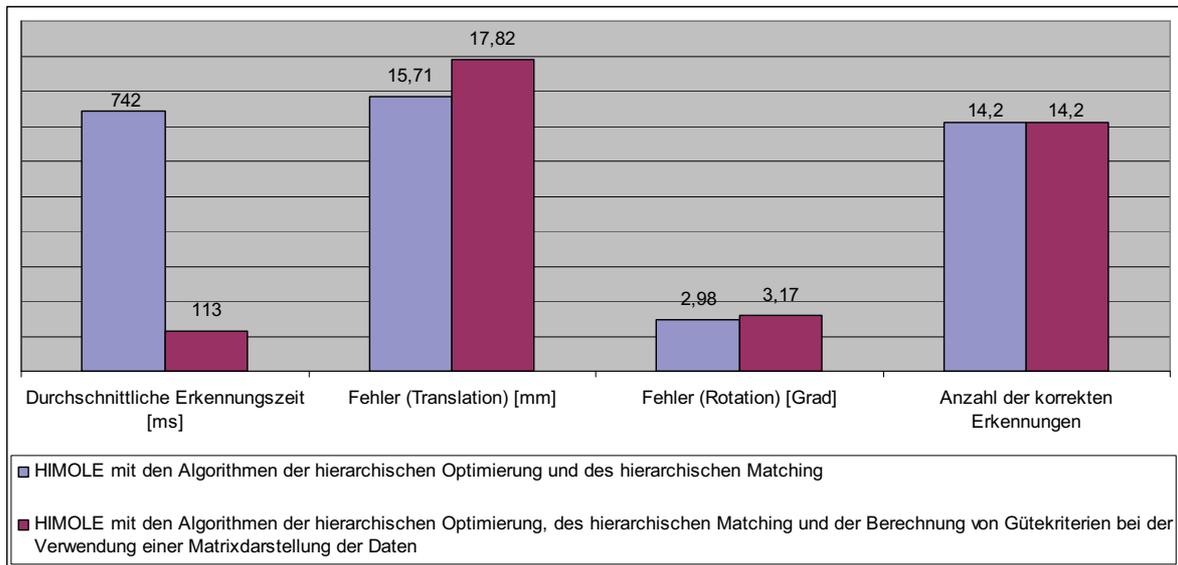


Abbildung 3.14: Experimentelle Ergebnisse (Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten)

Die Analyse der experimentellen Ergebnisse aus der Tabelle A.3 ergibt: Der Einsatz des Algorithmus zur Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten hat ermöglicht, die Erkennungszeit um 84,8% zu reduzieren, wobei die Genauigkeit der Translations- und Rotationsbestimmung bei der Lageerkennung wurden um 13,4% bzw. 6,4% reduziert. Die Anzahl der korrekten Erkennungen wurde nicht beeinflusst.

3.3.4 Algorithmus der Deformation des Objektprototyps

Daten, die zu erkennende Objekte enthalten, können verzerrt und verrauscht sein. Also können auch die Objekte verzerrt und deformiert sein. Beim Matching der Objektprototypen aus der Datenbank mit den Daten sind Fälle von unkorrekten Erkennungen möglich, weil ein Prototyp ein ideales Objekt beschreibt, aber geometrische und andere Variationen dieses Objektes nicht berücksichtigt.

Mit der Absicht alle mögliche Verzerrungen der Quelldaten zu berücksichtigen und Zuverlässigkeit der Erkennung zu erhöhen, wurde im Rahmen der vorliegenden Arbeit ein Algorithmus der Deformation des Objektprototyps entwickelt. Das Prinzip dieses Algorithmus wird im Folgenden dargestellt:

1. Es wird eine Gütefunktion definiert, die als ihre Argumente die Parameter des Prototyps des zu erkennendem Objekt annimmt, d.h. die parametrischen Koordinaten des Prototyps:

$$F = f(d_1, d_2, \dots, d_k, d_{k+1}, \dots, d_l, d_{l+1}, \dots, d_n) \quad (3.14)$$

mit (d_1, \dots, d_k) - Position des Prototyps, die durch räumlich-linearen Koordinaten gegeben ist; (d_{k+1}, \dots, d_l) - Orientierung des Prototyps, die durch räumlich-axialen Koordinaten gegeben ist; (d_{l+1}, \dots, d_n) - Deformation des Prototyps, die durch parametrische Koordinaten gegeben ist.

Parametrische Koordinaten eines Prototyps des zu erkennendem Objekts können im einfachsten Fall Verzerrungsfaktoren in Richtung x -Achse (s_x), y -Achse (s_y) und z -Achse (s_z) sein. Es ist auch möglich, andere Arten der Deformation zu benutzen (z.B. tangentielle Verzerrungen, usw.)

Mögliche Verzerrungen der zu erkennenden Objekte werden während der Entwicklung eines Objekterkennungssystems untersucht und analysiert. Danach werden die parametrischen Koordinaten des Prototyps definiert.

2. Im Laufe der Optimierung einer mehrdimensionalen Gütefunktion werden bei der Variation der Prototypsparameter nicht nur räumliche, sondern auch parametrische Koordinaten (Verzerrungsfaktoren, etc.) variiert. Die Suche nach einem globalen Optimum wird in einem Raum durchgeführt, dessen Dimensionen räumlich-lineare, räumlich-axiale und parametrische Koordinaten des Prototyps sind.

In Abbildung 3.15 ist der Algorithmus der Veränderung von Prototypsparameter dargestellt. In diesem Fall wird das Prototyp als ein 3D-Objekt gegeben, das mit 3D-Daten gematcht wird.

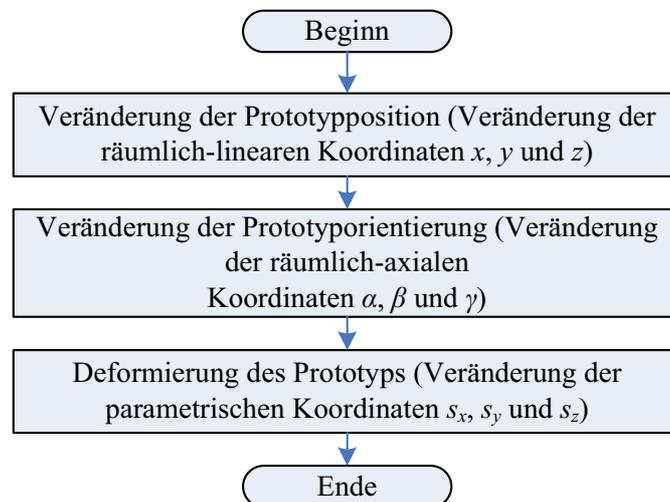


Abbildung 3.15: Algorithmus der Veränderung der Prototypsparameter

Als Beispiel sind in Abbildung 3.16 verschiedene Arten von Deformationen der Schablone des zu erkennendem Symbols dargestellt.

Ein ähnliches Verfahren wird in [45] vorgeschlagen. Im Gegensatz zu dem hier vorgeschlagenen Algorithmus wird dieses aber nur für die Erzeugung der Prototypen verwendet.

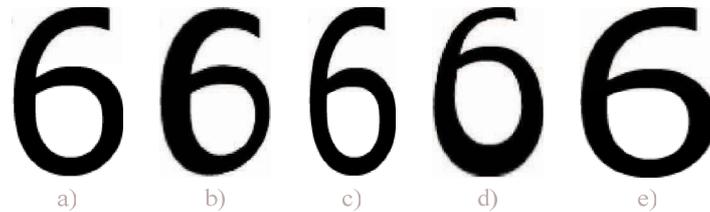


Abbildung 3.16: Mögliche Deformationen der Schablone des zu erkennendem Symbols

Experimente unter Verwendung des entwickelten Algorithmus der Deformation des Objektprototyps

Für die Bewertung der Effizienz des entwickeltem Algorithmus der Deformation des Objektprototyps wurde eine Reihe von Experimenten durchgeführt, deren Ergebnisse in Abbildung 3.17 (Daten in Tabelle A.4 im Anhang) dargestellt sind. Als zu vergleichende Methoden wurden die hierarchisch-iterative Methode ohne und mit dem Algorithmus der Deformation des Objektprototyps verwendet.

Wie im vorherigen Kapitel werden für den Vergleich sowohl dieselben Kriterien K_1 , K_2 , K_3 und K_4 als auch dieselbe 5 Datensätze, von denen jeder 20 3D-Szenen mit 2-3 Objekten verschiedener Klassen je Szene enthält, verwendet.

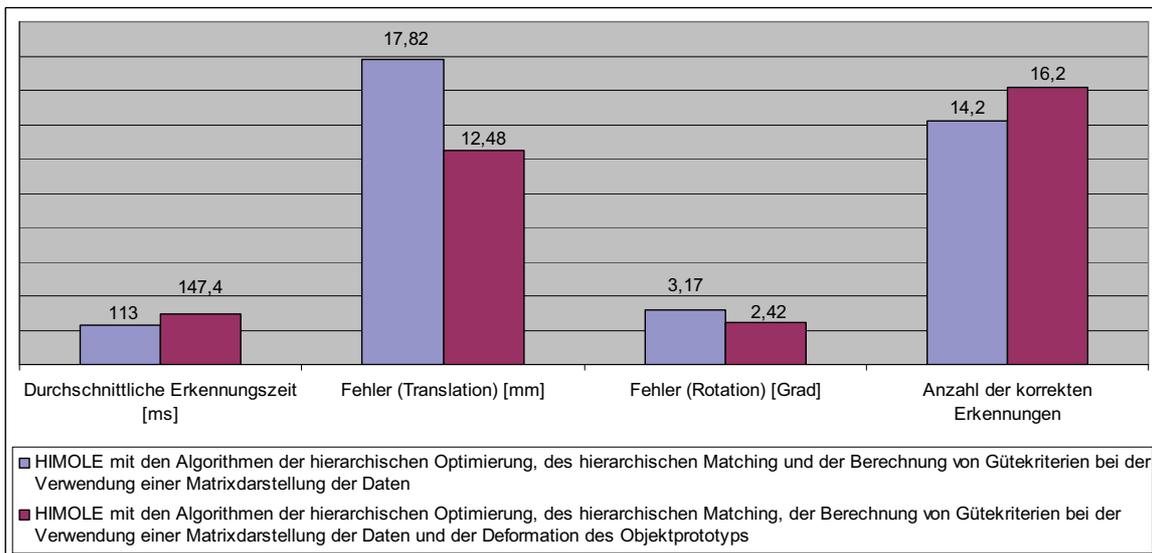


Abbildung 3.17: Experimentelle Ergebnisse (Algorithmus der Deformation des Objektprototyps)

Nach der Analyse der experimentellen Ergebnisse aus der Tabelle A.4 zeigt sich, dass der Einsatz des Algorithmus der Deformation des Objektprototyps ermöglicht hat, die Genauigkeit der Translations- und Rotationsbestimmung bei der Lageerkennung um 29,9% bzw. 23,6%, sowie die Anzahl der korrekten Erkennungen um 14,1% zu erhöhen, wobei die

Erkennungszeit um 30,4% gestiegen ist.

3.3.5 Algorithmus des iterativen Matchings

Das Prinzip dieses Algorithmus basiert auf der Vermutung, dass verschiedene Objekte nur im Fall der Vorverarbeitung der Daten mit verschiedenen Filterparametersätzen korrekt erkannt werden. Also muss der Prozess des Matchings iterativ und mit der Veränderung der Filterparameter für die Vorverarbeitung der Daten bei jeder Iteration durchgeführt werden.

In dem Algorithmus der hierarchischen iterativen Methode für die Objekt- und Lageerkennung, der in Abbildung 3.2 dargestellt ist, wird der Algorithmus des iterativen Matchings durch die Schritte 3, 4 und 20 realisiert. Diesen Algorithmus kann man auch folgendermaßen darstellen (s. Abb. 3.18).

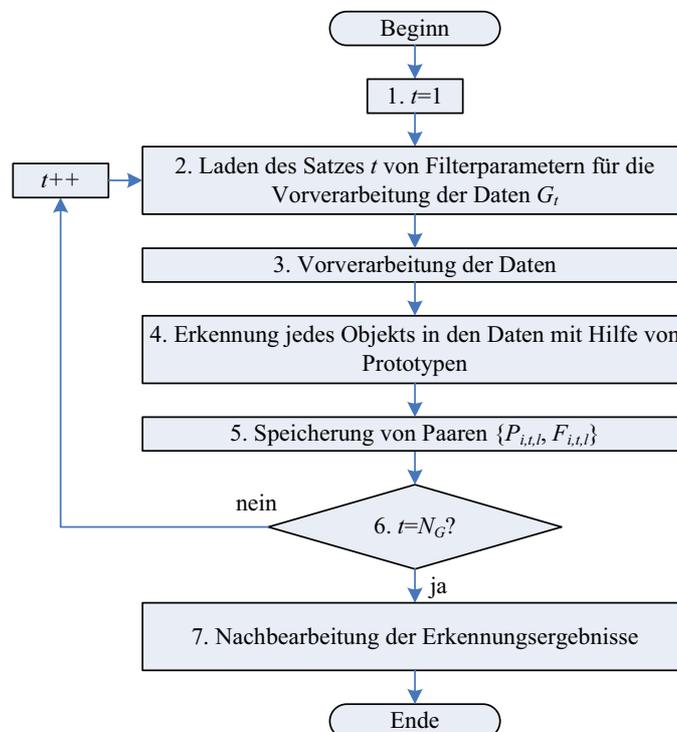


Abbildung 3.18: Algorithmus des iterativen Matchings

Der Einsatz des Algorithmus des iterativen Matchings ermöglicht die Lösung der Aufgabe der Objekterkennung auch in dem Fall, der in der Abbildung 3.19 dargestellt ist.

In der Abbildung 4.23 ist ein Beispiel der Schrifterkennung dargestellt. In dieser Abbildung wird das originale Bild, das zu erkennenden Text enthält, durch „a“ bezeichnet. Die Binärbilder, die mit der Verwendung von verschiedenen Schwellwerten erstellt wurden, werden durch „b“, „c“ und „d“ bezeichnet (75% weiße Pixel - b, 70% - c, 65% - d). Es ist zu sehen, dass die Zahl „0“ beim Schwellwert von 75% schwer zu erkennen ist. Beim Schwellwert von 70% werden die Zahlen mehr deutlicher. Immerhin beim Schwellwert von 70% wird

3. Hierarchisch-iterative Methode für die Objekt- und Lageerkennung

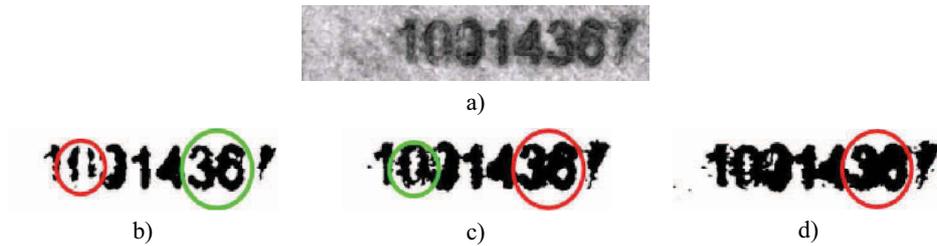


Abbildung 3.19: Beispiel der Verwendung von Algorithmus des iterativen Matchings

es kompliziert die Zahlkombination von „36“ zu erkennen, was beim Schwellwert von 65% trivial zu machen ist. Somit ist der Algorithmus des iterativen Matchings geeignet, dieses Problem zu lösen.

Experimente unter Verwendung des entwickelten Algorithmus

Für die Bewertung der Effizienz des entwickeltem Algorithmus des iterativen Matchings wurde eine Reihe von Experimenten durchgeführt, deren Ergebnisse in Abbildung 3.20 (Daten in Tabelle A.5 im Anhang) dargestellt sind. Als zu vergleichende Methoden wurden die hierarchisch-iterative Methode ohne und mit dem Algorithmus des iterativen Matchings verwendet.

Wie im vorherigen Kapitel werden für den Vergleich sowohl dieselbe Kriterien K_1 , K_2 , K_3 und K_4 als auch dieselbe 5 Datensätze, von denen jeder 20 3D-Szenen mit 2-3 Objekte verschiedener Klassen je Szene enthält, verwendet.

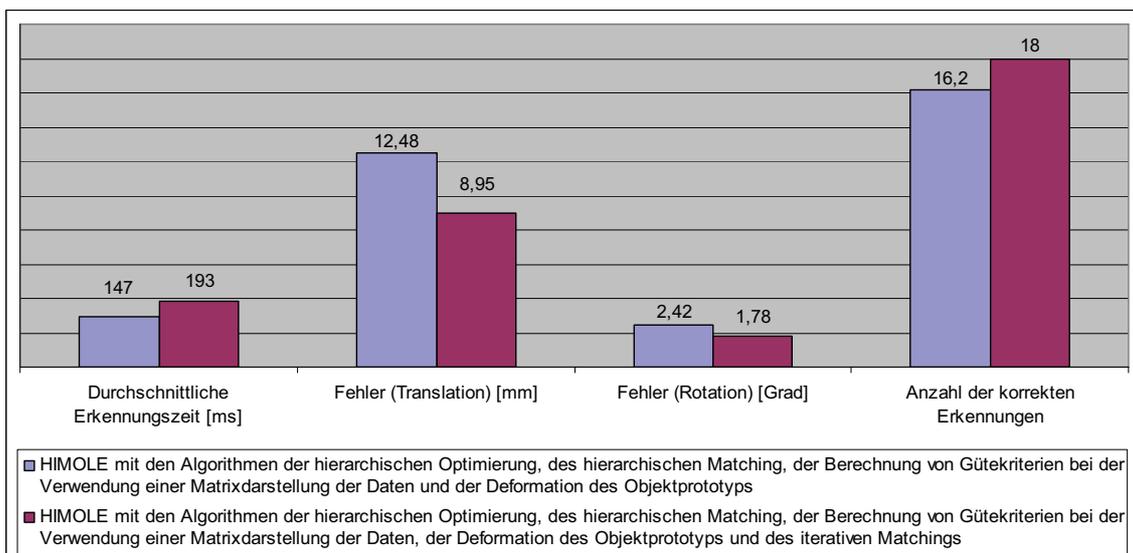


Abbildung 3.20: Experimentelle Ergebnisse (Algorithmus des iterativen Matchings)

Aus der Analyse der experimentellen Ergebnisse in Tabelle A.5 ergibt sich, dass der Einsatz

von Algorithmus des iterativen Matchings ermöglicht hat, die Genauigkeit der Translations- und Rotationsbestimmung bei der Lageerkennung um 28,3% bzw. 26,3%, sowie die Anzahl der korrekten Erkennungen um 11,1% zu erhöhen, wobei die Erkennungszeit um 30,9% gestiegen ist.

3.3.6 Algorithmus der Hierarchisierung von Objektprototypen

Die hierarchisch-iterative Methode für die Objekt- und Lageerkennung verwendet Prototypen idealer Objekte für die Erkennung realer Objekte in der 2D- und 3D-Daten. Solche Prototypen können auf folgende Weise bestimmt und erstellt werden:

- manuell: z.B. Definition eines Prototyps durch ein CAD-Modell.
- automatisch: z.B. einzelne reale Objekte werden aufgenommen und mittels dieser Daten werden Prototypen für diese Objekte erstellt.

Die Verwendung von Prototypen der zu erkennenden Objekte ermöglicht eine schnelle und robuste Identifikation von Objekten in 2D- und 3D-Daten, sowie die gleichzeitige Bestimmung ihrer Lage in der Szene. Dabei stellt die Erkennung von überlappenden Objekten verschiedener Klassen ein kritisches Problem dar. In diesem Fall schlägt der Algorithmus fehl, d.h. die Genauigkeit der Erkennung nimmt ab.

Für die Lösung des oben beschriebenen Problems wurde der Algorithmus der Hierarchisierung von Objektprototypen entwickelt, dessen Prinzip in Folgendem besteht:

- Modell eines Prototyps eines zu erkennenden Objekts wird nach der Analyse in Komponenten zerlegt.
- Ein hierarchisches Modell dieses Prototyps wird erstellt.
- Räumlich-topologische Beziehungen zwischen den Objektkomponenten werden identifiziert.
- Jede Komponente auf jeder Stufe des hierarchischen Modells wird gewichtet. Somit berücksichtigt der Algorithmus der Hierarchisierung von Objektprototypen die Gewichte, d.h. die Wertigkeit, der Objektkomponenten.

Nehmen wir an, das Prototyp des zu erkennenden Objekts ist durch folgendes hierarchischen Modell gegeben (s. Abb. 3.21). Jede Komponente hat das Gewicht $w_{L_i}^j$, mit Ordnungszahl der Komponente j und Nummer der Hierarchiestufe i . Der Algorithmus der Berechnung von Gütekriterium mit der Verwendung von Algorithmus der Hierarchisierung von Objektprototypen ist in der Abbildung 3.22 dargestellt.

Die Einschränkung der Verwendung des Algorithmus der Hierarchisierung von Objektprototypen besteht darin, dass dieser Algorithmus nur bei der lokalen oder präzisen Suchen eingesetzt werden kann, wenn die Lage des Prototyps ungefähr der Lage des zu erkennenden Objekts entspricht.

3. Hierarchisch-iterative Methode für die Objekt- und Lageerkennung

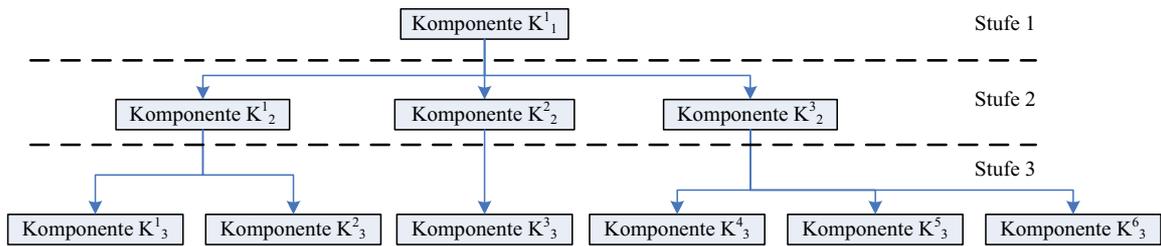


Abbildung 3.21: Hierarchisches Modell von einem Prototyp eines zu erkennenden Objekts

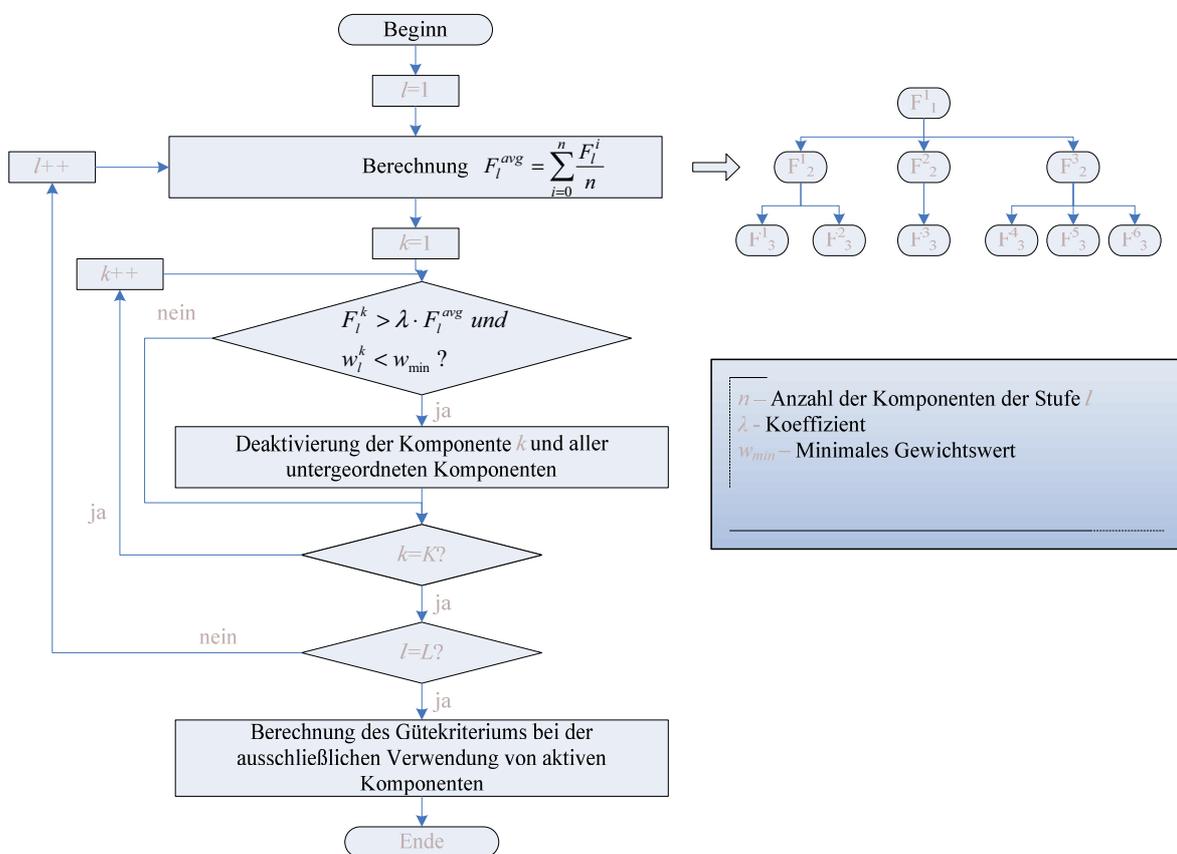


Abbildung 3.22: Algorithmus der Hierarchisierung von Objektprototypen

Der Algorithmus der Hierarchisierung von Objektprototypen erhöht die Genauigkeit der Objekterkennung in Fällen mit komplexen Szenen. Ein Beispiel einer solcher Szene ist in der Abbildung 3.23 dargestellt.

Aus der Abbildung 3.23 lässt sich ersehen, dass es für die Erkennung des Y -Objektes nur die Komponenten, die durch Kreise gekennzeichnet sind, nötig sind (nur Geometrie und topologische Beziehungen dieser Komponenten). Deswegen müssen die gekennzeichneten

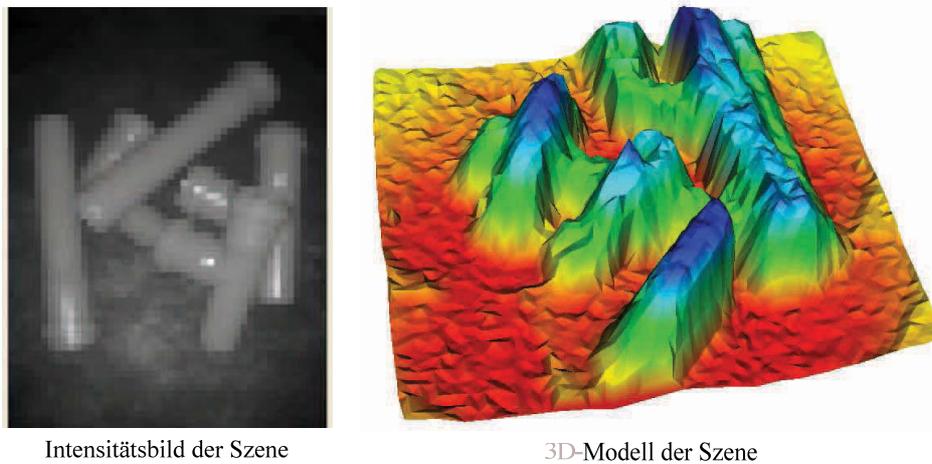


Abbildung 3.23: Beispiel einer komplexen 3D-Szene

Komponenten maximale Gewichte im hierarchischen Modell des Objektprototyps haben.

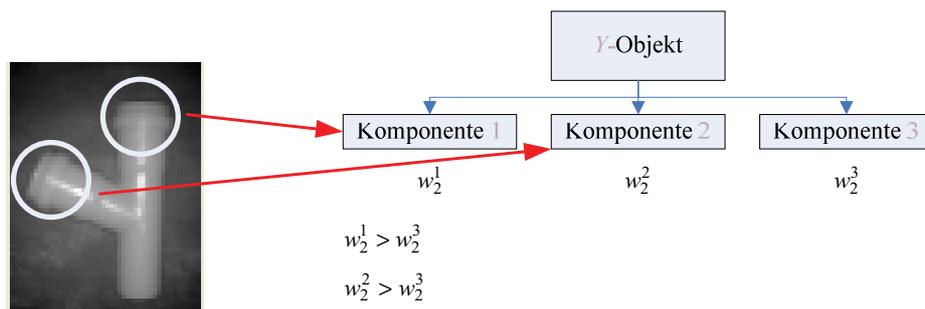


Abbildung 3.24: Beispiel des Y-Objekts und seines hierarchische Modell

Experimente unter Verwendung des entwickelten Algorithmus

Für die Bewertung der Effizienz des entwickeltem Algorithmus der Hierarchisierung von Objektprototypen wurde eine Reihe von Experimenten durchgeführt, deren Ergebnisse in Abbildung 3.25 (Daten in Tabelle A.6 im Anhang) dargestellt sind. Als zu vergleichende Methoden wurden die hierarchisch-iterative Methode ohne und mit dem Algorithmus der Hierarchisierung von Objektprototypen verwendet.

Wie im vorherigen Kapitel werden für den Vergleich sowohl dieselben Kriterien K_1 , K_2 , K_3 und K_4 als auch dieselben 5 Datensätze, von denen jeder 20 3D-Szenen mit 2-3 Objekten verschiedener Klassen je Szene enthält, verwendet.

Aus der Analyse der experimentellen Ergebnisse in Tabelle A.6 ergibt sich, dass der Einsatz von Algorithmus der Hierarchisierung von Objektprototypen ermöglicht hat, die Erkennungszeit um 9,6% zu reduzieren, die Genauigkeit der Translations- und Rotationsbe-

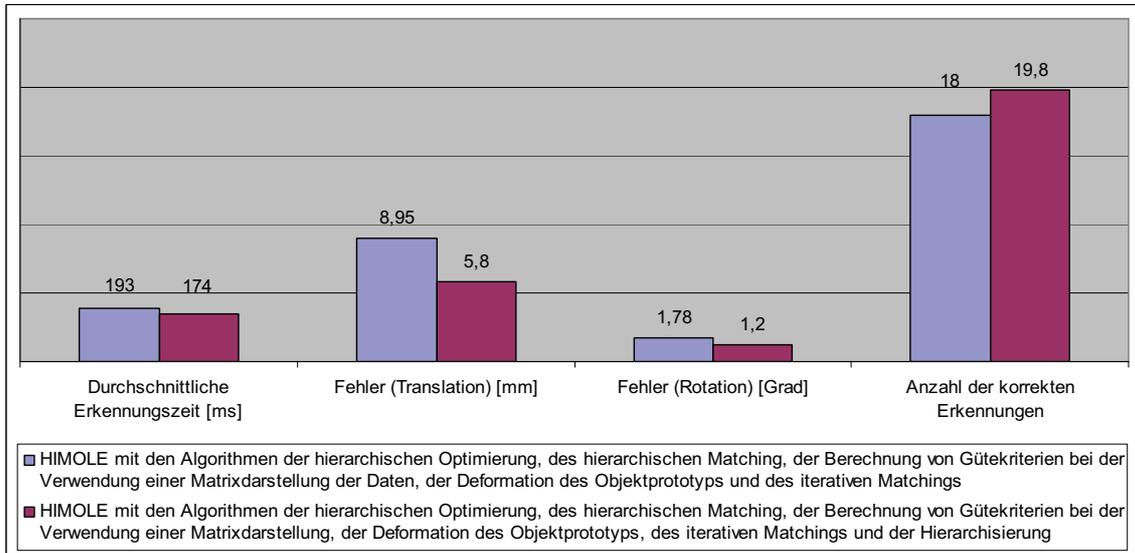


Abbildung 3.25: Experimentelle Ergebnisse (Algorithmus der Hierarchisierung von Objektprototypen)

stimmung bei der Lageerkennung um 35,2% bzw. 32,7%, sowie die Anzahl der korrekten Erkennungen um 10% zu erhöhen.

3.3.7 Analyse der entwickelten Algorithmen

In den vorherigen Kapiteln wurden die Experimente dargestellt, die den Beitrag jedes Algorithmus zur Erhöhung der Effizienz der hierarchisch-iterativen Methode (die Erhöhung der Genauigkeit der Translations- und Rotationsbestimmung bei der Lageerkennung und der Anzahl der korrekten Erkennungen, sowie die Reduzierung der Erkennungszeit) gezeigt haben.

In Abbildung 3.26 und in Tabelle 3.3 sind die Ergebnisse der Experimenten bei der Lösung der Griff-in-die-Kiste-Aufgabe dargestellt. Als zu vergleichende Methoden wurden die hierarchisch-iterative Methode ohne und mit allen entwickelten Algorithmen verwendet.

Für den Vergleich werden vier Kriterien K_1 , K_2 , K_3 und K_4 und 5 Datensätze, von denen jeder 20 3D-Szenen mit 2-3 Objekten verschiedener Klassen je Szene enthält, verwendet.

Die Analyse der experimentellen Ergebnisse aus der Tabelle 3.3 bestätigt: Der Einsatz der entwickelten Algorithmen hat ermöglicht, die Erkennungszeit um den Faktor 5,1 zu reduzieren, die Genauigkeit der Translations- und Rotationsbestimmung bei der Lageerkennung um die Faktoren 4,2 bzw. 3,7, sowie die Anzahl der korrekten Erkennungen um den Faktor 1,8 zu erhöhen.

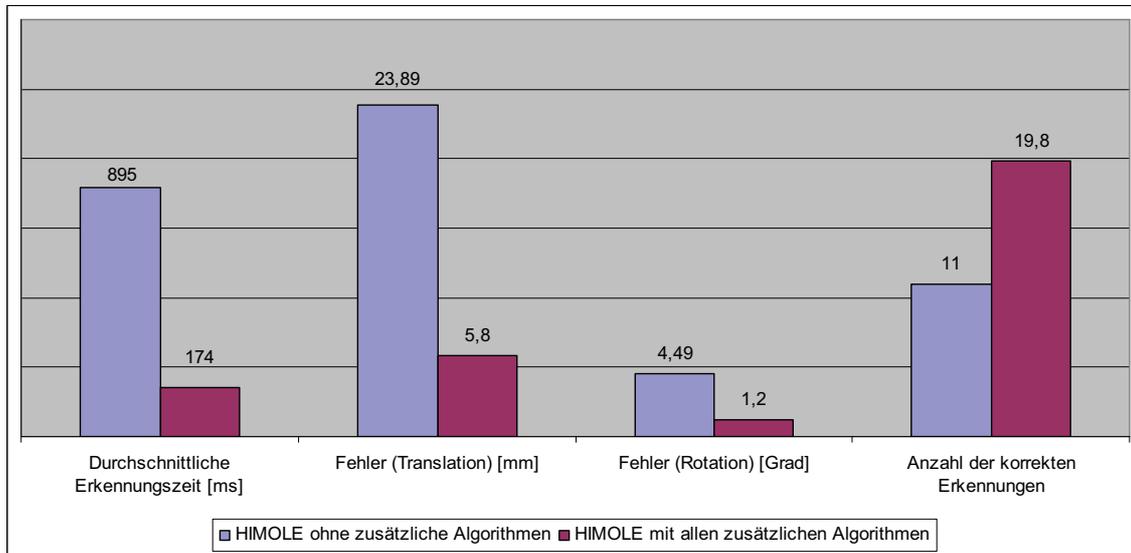


Abbildung 3.26: Experimentelle Ergebnisse (Hierarchische Optimierung)

Tabelle 3.3: Experimentelle Ergebnisse bei der Lösung Griff-in-die-Kiste-Aufgabe

Datensatz	Durchschnittliche Erkennungszeit [ms] (K_1)	Fehler (Translation) [mm] (K_2)	Fehler (Rotation) [Grad] (K_3)	Anzahl der korrekten Erkennungen (K_4)
HIMOLE ohne zusätzliche Algorithmen				
1	846	13,51	4,26	11
2	860	20,74	5,31	11
3	779	23,21	3,91	10
4	856	33,75	3,84	11
5	1133	28,25	5,11	12
HIMOLE mit allen zusätzlichen Algorithmen				
1	170	5,10	1,20	20
2	166	5,50	1,30	20
3	190	6,00	1,00	19
4	158	6,50	0,90	20
5	188	5,90	1,60	20

3.3.8 Fazit

In diesem Kapitel wurde die allgemeine Problemstellung der Erkennung eines Objekts und seiner Lage in der Szene definiert.

Da keine der Methode, die im ersten Kapitel analysiert wurden, allen Anforderungen, die in Tabelle 2.1 dargestellt sind, entspricht, wurde die hierarchisch-iterative Methode entwickelt, die auf der Kombination des Matching-Verfahrens mit Optimierungsalgorithmen basiert.

3. Hierarchisch-iterative Methode für die Objekt- und Lageerkennung

Die folgenden Algorithmen, die zu der hierarchisch-iterativen Methode gehört, wurden entwickelt:

- Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen
- Algorithmus des hierarchischen Matchings
- Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten
- Algorithmus der Deformation des Objektprototyps
- Algorithmus des iterativen Matchings
- Algorithmus der Hierarchisierung von Objektprototypen

Der Algorithmus der hierarchisch-iterativen Methode, sowie die experimentellen Ergebnisse wurden angegeben. Diese experimentellen Ergebnisse haben gezeigt, dass der Einsatz der hierarchisch-iterativen Methode mit erweiterten Algorithmen die Reduzierung der Erkennungszeit und Erhöhung der Anzahl von korrekten Erkennungen sowie der Genauigkeit der Translations- und Rotationsbestimmung bei der Lageerkennung ermöglicht.

Kapitel 4

Systeme für die Objekt- und Lageerkennung

4.1 Entwicklungsverfahren von Systemen für die Objekt- und Lageerkennung

Alle Aufgaben der Objekterkennung können in vier Klassen eingeteilt werden:

1. Erkennung der Lage eines Objektes in der Szene (Position und Orientierung)
2. Erkennung eines Objektes und seiner Lage in der Szene
3. Erkennung mehrerer Objekte
4. Erkennung mehrerer Objekte und ihrer Lagen in der Szene

Mit Hilfe der entwickelten hierarchisch-iterativen Methode für die Objekt- und Lageerkennung wurden folgende Aufgaben erfolgreich gelöst:

1. Registrierung medizinischer Bilddatensätze. Diese Aufgabe besteht in der Analyse von Tomogrammen eines Organs oder Körperteils und in der Suche nach ähnlichen Tomogrammen in einem Tomogrammatlas (Datenbank von Tomogrammen) für die Diagnosefeststellung oder für die Fusion mit einem Tomogramm anderes Typs (z.B. CT- und MRT-Tomogramme) des selben Organs.
2. Schrifterkennung in der Industrie, die sich durch Rauschen, Verzerrungen und Deformationen des Textes erschwert.
3. Erkennung mehrerer Objekte verschiedener Klassen, die überlappt sein können, und ihrer Lagen für das Griff-in-die-Kiste-System.

Die Lösungen für diese Aufgaben sind nicht trivial zu finden, weil viele existierende Methoden die folgende Besonderheiten dieser Aufgaben nicht berücksichtigen:

4. Systeme für die Objekt- und Lageerkennung

- Umfangreiche Modifikation oder Anpassung bei der Veränderung der Datensätze sind zu vermeiden.
- Es ist notwendig, nicht nur die Objekte zu erkennen sondern auch ihre Lage in der Szene zu bestimmen (gleichzeitige Objekt- und Lageerkennung).
- Hohe Geschwindigkeit, Genauigkeit und Zuverlässigkeit der Objekt- und Lageerkennung sind sehr wichtig.

Der Einsatz der hierarchisch-iterativen Methode für die Objekt- und Lageerkennung hat ermöglicht, alle diese Besonderheiten zu berücksichtigen, und hat die hohe Effizienz bei der Lösung der oben beschriebenen Aufgaben gezeigt.

In Kapitel 3 wurde der generische Algorithmus der hierarchisch-iterativen Methode beschrieben. Für die Entwicklung des realen Systems für die Objekt- und Lageerkennung ist es notwendig:

1. Die Klasse der Eingangsdaten zu bestimmen:
 - 2D- und 3D-Daten, die Binärinformation enthalten
 - 2D- und 3D-Daten, die Grauwertinformation enthalten
 - 2D- und 3D-Daten, die Farbinformation enthalten
2. Die Methode für Bestimmung und Erstellung von Objektprototypen zu definieren:
 - manuell
 - automatisch
3. Wesentliche Merkmale von Objektprototypen, die bei der Objekterkennung verwendet werden, zu bestimmen und ihre Hierarchie zu definieren
4. Gütekriterien, sowie die Gütefunktionen festzulegen
5. Filter für die Vorverarbeitung der Daten festzulegen
6. Die Parameter jedes Algorithmus der hierarchischen iterativen Methode festzusetzen:
 - Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen:
 - Stufen der Optimierung, die verwendet werden: globale, lokale und präzise Suche
 - Kombinationen der linear abhängigen Koordinaten
 - nichträumliche parametrische Koordinaten
 - Gütefunktionen und Varianten ihrer gleichzeitigen Optimierung
 - Algorithmus des hierarchischen Matchings
 - Filter für die Erstellung einer Matrixdarstellung der Eingangsdaten und Prototypen von zu erkennenden Objekten
 - Anzahl der Pyramidenstufen und die Parameter jeder Stufe

- Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten
 - Matrixtyp: Distanz- oder Zeigermatrix
 - Parameter des iterativen Algorithmus des Ausfüllens der Matrixzellen
 - Algorithmus der Deformation des Objektprototyps
 - nichträumliche Parameter von Objektprototypen
 - Gütefunktionen, die diese Parameter als Argumente annehmen können
 - Algorithmus des iterativen Matchings
 - Filter für die Vorverarbeitung der Eingangsdaten
 - Parameter dieser Filter, die im Laufe verändert werden können
 - Algorithmus der Hierarchisierung von Objektprototypen
 - Besonderheiten der Hierarchisierung von Objektprototypen
7. Lernstichprobe für das Lernen, d.h. Menge der Objekte von bekannten Klassen festlegen
8. Lernverfahren für das Lernen des zu entwickelnden Erkennungssystems wählen
- mit Vorablernverfahren
 - gleichzeitige Erkennung mit Lernen
9. Das entwickelte System für die Lösung konkreter Aufgaben optimieren, d.h. die Parameter dieses Systems müssen adaptiert und angepasst werden, um Zeit-, Speicher- und andere Einschränkungen zu berücksichtigen

Einer der wichtigsten Schritte bei der Entwicklung von Erkennungssystemen ist der Lernvorgang dieses Systems. Der Lernvorgang eines Erkennungssystems wird mit Hilfe einer Lernstichprobe durchgeführt. Dabei erfolgt die Anpassung der Algorithmenparameter mit dem Ziel, alle Objekte aus der ausgewählten Lernstichprobe zu erkennen.

Meistens ist es unmöglich, alle Parameter so anzupassen, dass alle Objekte korrekt erkannt werden. In diesem Fall ist es notwendig die Gütekriterien oder wesentliche Merkmale von Prototypen der zu erkennenden Objekte neu festzulegen.

Es gibt zwei Lernverfahren:

- Vorablernverfahren: Vor der Erkennung wird ein Satz der bekannten Objekte für die Anpassung der Systemparameter verwendet. Diese angepassten Parameter werden dann für die Erkennung unbekannter Objekte benutzt.
- Gleichzeitige Erkennung mit Lernen: Die Parameter, die mit der Hilfe von den bekannten Objekten angepasst wurden, werden für die Erkennung benutzt. Aber im Laufe des Erkennungsprozesses werden sie nochmal verfeinert. Dies passiert bei jeder Iteration und die Parameter werden iterativ korrigiert.

4.2 Entwicklung des Lageerkennungssystems für die Registrierung medizinischer Bilddatensätze

4.2.1 Problemstellung

Eine Aufgabe, die mit der Hilfe der hierarchisch-iterativen Methode effizient gelöst werden kann, ist die Aufgabe der Registrierung medizinischer Bilddatensätze (s. Abb. 4.1). Sie besteht aus folgenden zwei Teilaufgaben:

1. Registrierung (Überlagerung) von CT- und MRT-Tomogrammen eines Organs (Aufgabe der Klasse 1 - Erkennung der Lage eines Objektes in der Szene (Position und Orientierung)).
2. Analyse von Tomogrammen eines Organs oder Körperteils und die Suche nach ähnlichen Tomogramme in einem Tomogrammenatlas (Datenbank von Tomogrammen) für die Diagnosefeststellung. In diesem Fall kann es sich um gleich- oder verschiedenartige Tomogramme handeln (CT, MRT) (Aufgabe der Klasse 2 - Erkennung eines Objektes und seiner Lage in der Szene).

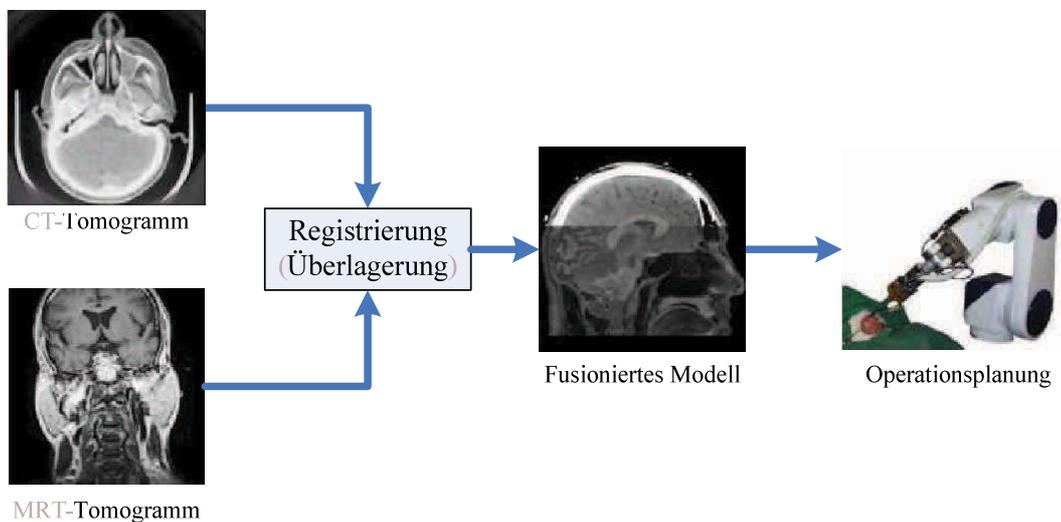


Abbildung 4.1: Registrierung medizinischer Bilddatensätze

Im Rahmen dieser Dissertation wurde ein System, das auf der hierarchisch-iterativen Methode für die Objekt- und Lageerkennung basiert und die Aufgabe der Registrierung medizinischer Bilddatensätze löst, entwickelt ([29, 78]). Dieses System ist auch in der Lage die Aufgabe der Tomogrammanalyse und die Suche nach ähnlichen Tomogrammen im Atlas zu lösen.

4.2.2 Architektur des Lageerkennungssystems

Basierend auf den in Kapitel 4.1 beschriebenen Schritten zur Entwicklung eines Objekt- und Lageerkennungssystems wurde das System für die Registrierung medizinischer Bild- datensätze entwickelt und implementiert.

Die Klasse der Eingangsdaten

Eingangsdaten sind medizinische Bilddatensätze in Form verschiedener Tomogramme: Computertomogramm, Magnetresonanztomogramm usw. Die Eingangsdaten sind dreidi- mensionale Daten mit Grauwertinformation.

Die Methode für die Bestimmung und Erstellung von Objektprototypen

Ein Prototyp des zu erkennenden Objekts ist ebenfalls ein Tomogramm. Ein solches Tomo- gramm wird automatisch durch einen Tomographen aufgenommen.

Wesentliche Merkmale von Objektprototypen

Die wesentliche Merkmale von Objektprototypen sind folgende (s. Abb. 4.2):

- anatomische Merkmale wie Augen und Ohren
- geometrische Merkmale wie verschiedene Konturen, Oberflächen und Volumen

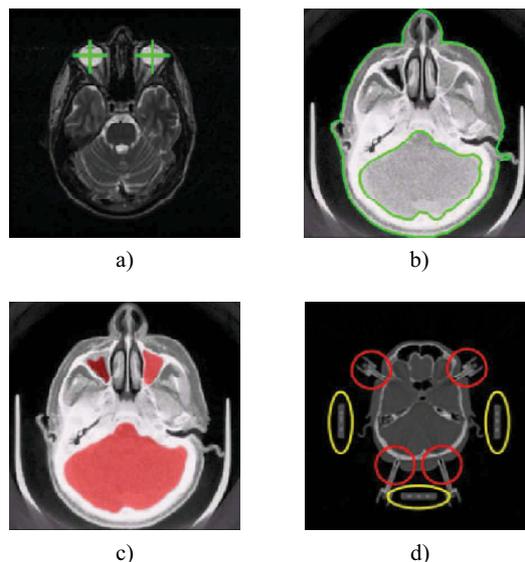


Abbildung 4.2: Wesentliche Merkmale von Tomogrammen

Gütekriterien

Für die Lösung der Aufgabe der Registrierung medizinischer Bilddatensätze werden folgende zwei Gütekriterien verwendet:

1. **Syntropie (Mutual entropy)**. In diesem Fall wird das Gütekriterium mit Hilfe folgender Formel berechnet [70]:

$$F_{ME} = - \sum_{i=0}^{|v|} \sum_{j=0}^{|v|} p(i, j) \log \frac{p(i, j)}{p_x(i) p_y(j)} \quad (4.1)$$

mit

$$p(i, j) = \frac{f(i, j)}{N}; p_x(i) = \sum_{j=0}^{|v|} \frac{f(i, j)}{N}; p_y(j) = \sum_{i=0}^{|v|} \frac{f(i, j)}{N} \quad (4.2)$$

mit N - Anzahl aller Voxel im Tomogramm; $p_x(i), p_y(i)$ - normierte Häufigkeit des Auftretens einer bestimmten Grauwertkombination im Histogramm mit insgesamt N erfolgten Inkrementierungen (Spalte bzw. Zeile).

Für die Berechnung eines solchen Gütekriteriums wird in [70] vorgeschlagen, ein zweidimensionales Grauwert-Histogramm zu benutzen. Dieses Histogramm hat die Form einer 2D-Matrix. Jede Zelle dieser Matrix entspricht der Anzahl der Voxel mit der Grauwertkombination, die durch die Nummer von Spalte und Zeile gegeben sind. Die Anzahl an Zeilen entspricht der Anzahl der Grauwerte des ersten Tomogramms. Die Anzahl an Spalten entspricht der Anzahl der Grauwerte des zweiten Tomogramms (normalerweise $0 \dots 255$) (s. Abb. 4.3).

2. **Gradienteninformation** (Konturen und Oberflächen verschiedener anatomischer Merkmale). In diesem Fall wird das Gütekriterium durch folgende Formel berechnet:

$$F_{SSD} = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K (f_d^m(x+i, y+j, z+k)^2 \cdot \bar{M}(i, j, k)) \quad (4.3)$$

mit f_d^m - Distanzmatrix, die aus erstem Tomogramm berechnet wurde; (x, y, z) - Lage des Tomograms \bar{M} im Bezug auf Distanzmatrix; I, J, K - Größe des Tomogramms \bar{M} ; \bar{M} kann durch folgende Formel berechnet werden:

$$\bar{M} = T(M, \alpha, \beta, \gamma, s_x, s_y, s_z) \quad (4.4)$$

mit M - zweites Tomogramm, das mit erstem Tomogramm gematcht wird; T - Funktion zur Transformation dieses Tomogramms; α, β, γ - Rotationen um x -, y - und z -Achse; s_x, s_y, s_z - Verzerrungsfaktoren entlang x -, y - und z -Achse.

Unter Verwendung der Zeigermatrix f_p^m wird F_{SSD} folgendermaßen berechnet:

$$F_{SSD} = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K (((f_p^m(x+i, y+j, z+k)_x - i)^2 + (f_p^m(x+i, y+j, z+k)_y - j)^2 + (f_p^m(x+i, y+j, z+k)_z - k)^2)^{\frac{1}{2}} \cdot \bar{M}(i, j, k)) \quad (4.5)$$

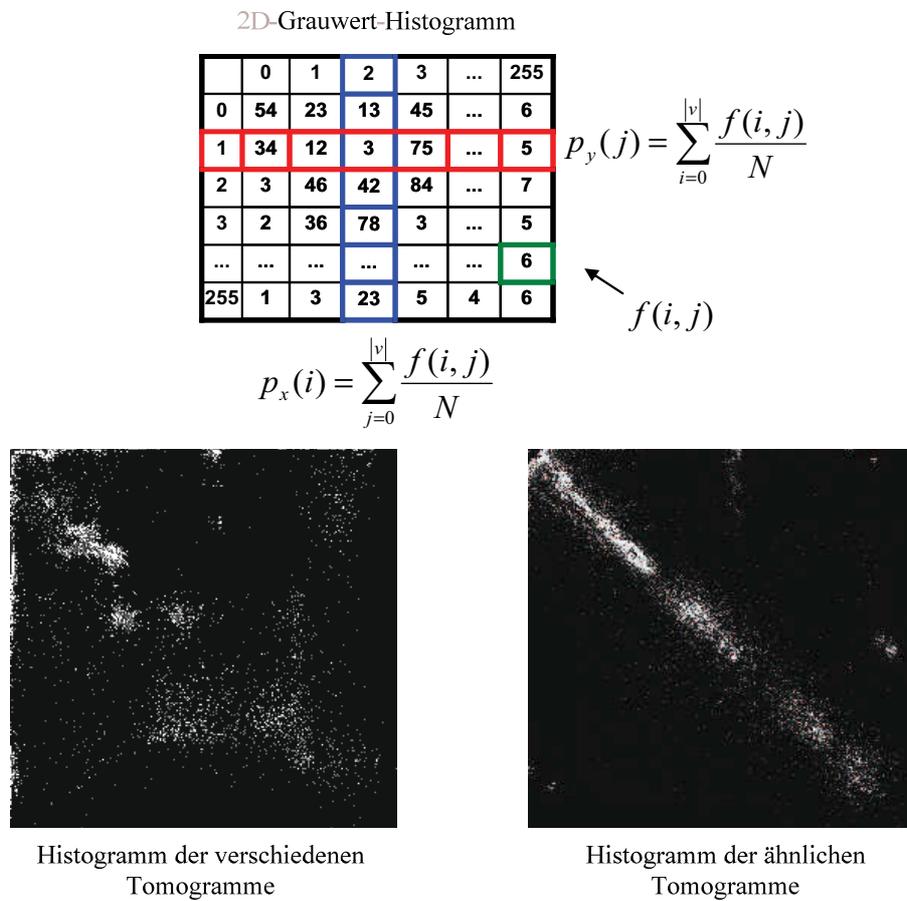


Abbildung 4.3: 2D-Grauwert-Histogramm

Kurz kann man beide Formeln folgendermaßen definieren:

$$F_{SSD} = \sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K d_{NN}(R^3, \bar{M}, x, y, z, i, j, k)^2 \quad (4.6)$$

mit d_{NN} - Abstand von einem Punkt des Gradientbildes eines Tomogramms \bar{M} mit Koordinaten (i, j, k) bis zum nächstgelegenen Punkt des Gradientbildes eines anderen Tomogramms R^3 .

Filter für die Vorverarbeitung der Eingangsdaten

Die Filter, die für die Vorverarbeitung der Eingangsdaten verwendet werden, kann man in 2 Klassen unterteilen:

- Rauschfilter
- Gradientenfilter

Rauschunterdrückung. Um die besserer Resultate der Erkennung zu erzielen, ist es sinnvoll, einen Filter für die Rauschunterdrückung auf die Eingangsdaten anzuwenden. Als solcher Filter wird der anisotropische Diffusionsfilter benutzt. Der anisotropische Diffusionsfilter ([38]) ermöglicht die Reduzierung des Rauschens in den Daten unter Beibehaltung wesentlicher Information trotzdem behalten (s. Abb. 4.4).

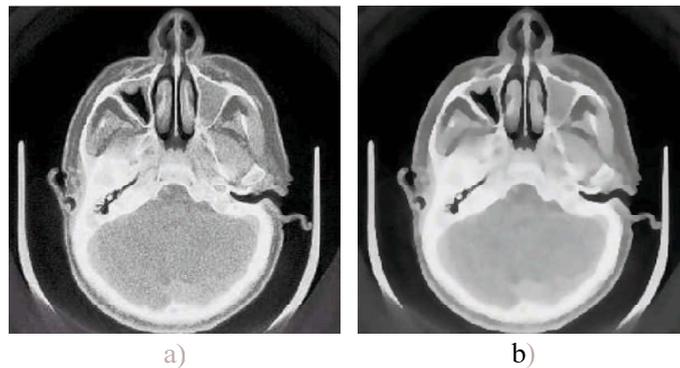


Abbildung 4.4: Originales CT-Tomogramm (a) und Tomogramm, das mit dem anisotropischen Diffusionsfilter bearbeitet wurde (b)

Berechnung der Gradientenbilder. Gradientenbilder helfen anatomische und geometrische Merkmale von Tomogrammen effizient zu extrahieren. Um Gradientenbilder zu berechnen, ist es möglich verschiedene Filter zu benutzen: Laplace-, Sobel-, Canny- und morphologische Filter (s. Abb. 4.5).

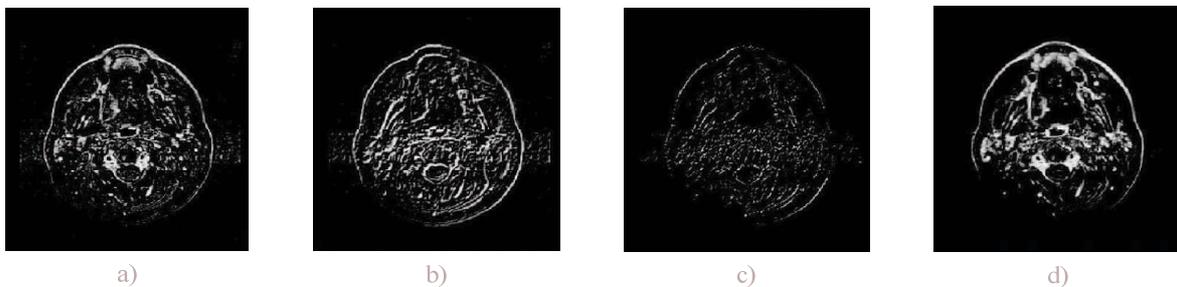


Abbildung 4.5: MRT-Tomogramm, das mit Laplace- (a), Sobel- (b), Canny- (c) und morphologischem (d) Filter bearbeitet wurde

Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen

Alle drei Stufen der Optimierung werden verwendet - globale, lokale und präzise Suche. Folgende Dimensionen und ihre Kombinationen werden benutzt:

- Translationen entlang x -, y - und z -Achse
- Rotation um x -, y - und z -Achse

- Translationen entlang folgender Richtungen: (x, y) , $(-x, y)$, (x, z) , $(-x, z)$, (y, z) , $(-y, z)$, (x, y, z) , $(-x, y, z)$, $(-x, -y, z)$, $(x, -y, z)$

Als nichträumliche parametrische Koordinaten werden Verzerrungsfaktoren des Tomogramms verwendet (s_x , s_y und s_z).

Zwei Gütekriterien werden verwendet:

- Synentropie (Mutual entropy) - F_{MI} (s. Formel 4.1)
- Summe der quadratischen Abstände - F_{SSD} (s. Formel 4.3).

Die gleichzeitige Verwendung dieser Gütekriterien, die verschiedene Information benutzen, ermöglicht das Problem der vorzeitigen Konvergenz im Punkt des lokalen Extremums bei der Optimierung zu vermeiden.

Folgende Kombinationen der gleichzeitigen Verwendung der Gütekriterien werden verwendet:

- aufeinanderfolgende Verwendung der Gütekriterien. Bei der globalen Suche wird das Gütekriterium 1 (F_{MI}) und bei der lokalen Suche Gütekriterium 2 (F_{SSD}) verwendet.
- Verwendung der kombinierten Gütefunktion bei der präzisen Suche. Diese Gütefunktion sieht folgendermaßen aus:

$$\begin{aligned}
 F &= f(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z) \\
 &= f(f_{MI}(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z), \\
 &\quad f_{SSD}(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z)) \\
 &= f_{MI}(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z) \cdot \\
 &\quad f_{SSD}(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z)
 \end{aligned} \tag{4.7}$$

mit t_x, t_y, t_z - Translation entlang x -, y - und z -Achse; α, β, γ - Rotationen um x -, y - und z -Achse; s_x, s_y und s_z - Verzerrungsfaktoren entlang x -, y - und z -Achse.

Algorithmus des hierarchischen Matchings

Für den Einsatz des Algorithmus des hierarchischen Matchings ist es nötig, die Eingangsdaten (in diesem Fall Tomogramme) als Pyramide darzustellen. Eine solche Pyramide kann mit Hilfe von Filtern, die die Größe des Tomogramms ohne Verlust wesentlicher Information reduzieren können, erstellt werden. Als solche Filter kann man Gauß- oder Laplace-Filter benutzen. In der Abbildung 4.6 wird die Pyramide eines Tomogramms dargestellt, die unter Anwendung wiederholter Gauß-Filterung erstellt wurde.

Obwohl eine solche Datenstruktur mehr Speicher benötigt, nimmt der notwendige Speicherplatz nur um den Faktor 1,5 zu (s. Formel 3.13). Außerdem haben die durchgeführten Experimente gezeigt, dass die Gütefunktion genauer optimiert wird, weil die Gütefunktion beim Matching von Tomogrammen in groben Stufen mehr geglättet wird und deswegen wenige lokale Extrema enthält. Dies führt zur Erhöhung der Wahrscheinlichkeit des Auffindens eines globalen Extremums (s. Abb. 4.7).

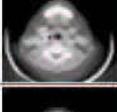
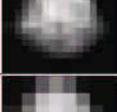
Auflösung	Größe	CT	Größe	MRT
original	512x512		256x256	
1/2	256x256		128x128	
1/4	128x128		64x64	
1/8	64x64		32x32	
1/16	32x32		16x16	
1/32	16x16		8x8	

Abbildung 4.6: Gauß-Pyramide der CT- und MRT-Tomogramme

Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten

Um die Berechnungszeit von Gütekriterien zu reduzieren, wird eine Distanzmatrix verwendet. Da ein 3D-Tomogramm eine große Anzahl von Voxeln enthält (normalerweise ist die Auflösung des CT-Tomogramms $512 \times 512 \times 50$ Voxel), erhöht der Einsatz der Zeigermatrix die benötigte Berechnungszeit.

Die Distanzmatrix wird nur für die Berechnung des Gütekriteriums, das auf Gradienteninformationen basiert, verwendet:

- Bei der Registrierung von zwei Tomogrammen eines Organs, ist ein Tomogramm fixiert (normalerweise - das größte Tomogramm) und es wird eine Distanzmatrix für dieses Tomogramm berechnet. Aus dem zweiten Tomogramm wird die Gradienteninformation extrahiert und das berechnete Gradientenbild wird mit dem fixierten Tomogramm (genauer: mit seiner Distanzmatrix) gematcht.
- Bei der Analyse eines Tomogramms und bei der Suche nach einem ähnlichen Bild im Atlas wird dieses Tomogramm fixiert und seine Distanzmatrix berechnet. Aus den Tomogrammen des Atlas werden Gradientenbilder extrahiert, die mit dem zu analysierenden Tomogramm gematcht werden.

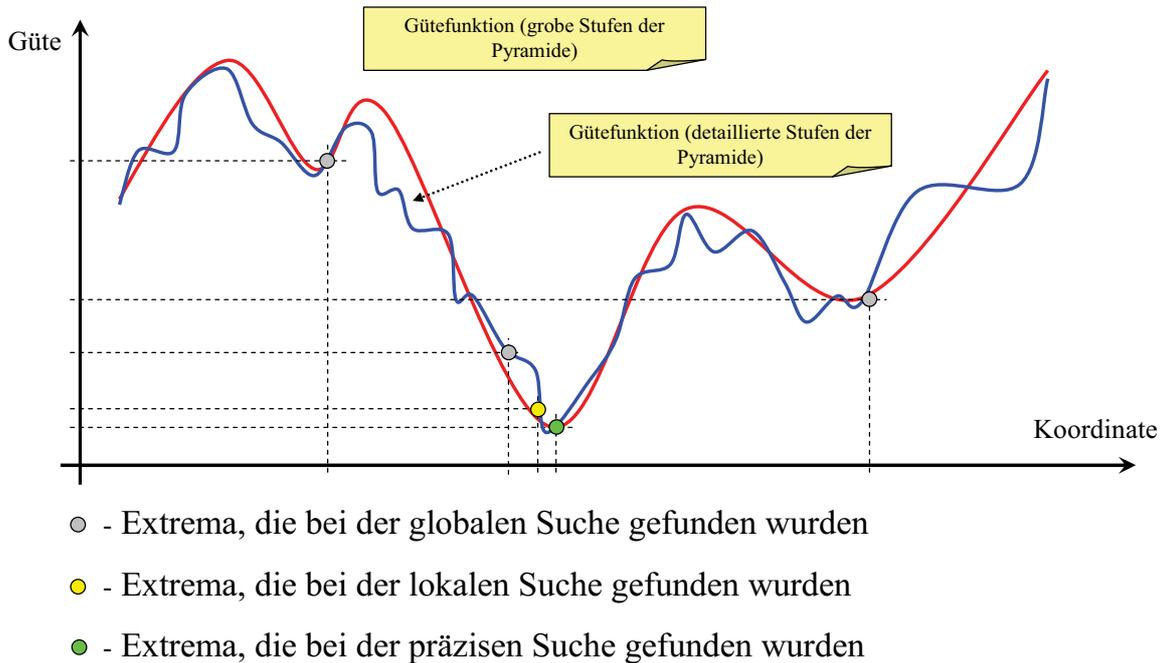


Abbildung 4.7: Gütefunktion unter Verwendung von Algorithmus des hierarchisches Matchings

Algorithmus der Deformation des Objektprototyps

Bei der Registrierung werden zwei Gütekriterien F_{MI} und F_{SSD} verwendet. Jedes Kriterium basiert auf der Gütefunktion, die als Argumente folgende nicht-räumliche parametrische Koordinaten von Objektprototypen annimmt: Skalierungsfaktoren s_x - entlang x -Achse, s_y - entlang y -Achse und s_z - entlang z -Achse.

Jeder Faktor definiert die Streckung oder Stauchung des Tomogramms entlang einer Achse (s. Abb. 4.8).

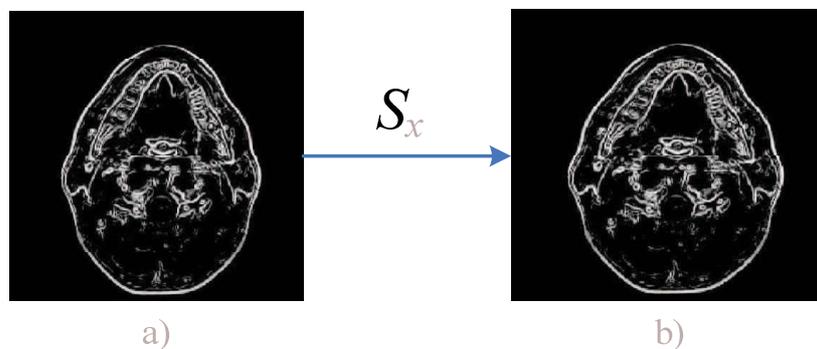


Abbildung 4.8: Skalierungsfaktor s_x entlang x -Achse

Gütefunktion, die die oben genannten Verzerrungsfaktoren verwenden, werden durch folgende Formeln definiert:

$$F_{MI} = f_{MI}(x, y, z, \alpha, \beta, \gamma, s_x, s_y, s_z) \quad (4.8)$$

$$F_{SSD} = f_{SSD}(x, y, z, \alpha, \beta, \gamma, s_x, s_y, s_z) \quad (4.9)$$

Algorithmus des iterativen Matchings

Die Genauigkeit der entwickelten Methode für die Objekt- und Lageerkennung hängt von der Qualität der Eingangsdaten, die mit verschiedenen Filtern (Gradienten, Rauschunterdrückung, etc.) vorverarbeitet wurden, ab. Außerdem sind einige Tomogrammpaare korrekt bei einem Satz von Filterparametern überlagert. Andere Tomogramme liefern eine hohe Übereinstimmung bei einem anderen Satz von Filterparametern.

Folgende Filter für die Vorverarbeitung der Daten werden vom Algorithmus des iterativen Matchings verwendet:

- anisotropisches Diffusionsfilter
- Gradientenfilter:
 - Laplace
 - Canny
 - morphologische Filter

Wesentliche Parameter dieser Filter, die im Prozess verändert werden, sind die folgende:

- Kerngröße des Filters
- Anzahl der Iterationen
- Schwellwert für die Binarisierung (für die Gradientenfilter)

Während des Prozesses der Tomogrammregistrierung werden nicht nur die Filterparameter, sondern auch der Filtertype geändert (z.B. Gradientenfilter).

Algorithmus der Hierarchisierung von Objektprototypen

Bei der Vorverarbeitung der Eingangsdaten und während der Optimierung der Gütefunktion wird die Hierarchisierung und Gewichtung der Komponenten eines Tomogramms durchgeführt:

- bei der Vorverarbeitung der Eingangsdaten werden einige Tomogrammkomponenten durch die Filterung gelöscht oder niedrig gewichtet;
- während der Optimierung der Gütefunktion wird der Durchschnittsabstand zwischen zwei benachbarten Elementen der zwei Tomogramme berechnet. Aufgrund dieses Wertes werden die Tomogrammkomponenten erneut gewichtet (z.B. werden ferne Komponenten minimal gewichtet).

Lernverfahren

Das System für die Registrierung medizinischer Bilddatensätze wird einem Vorabverfahren trainiert. Eine Lernstichprobe von Tomogrammen wurde bestimmt, mit der die initialen Parameter der Algorithmen und Filter berechnet wurden. Aufgrund dieser Lernstichprobe wurde das System für die Registrierung unbekannter Tomogramme angepasst.

4.2.3 Experimentelle Ergebnisse

Es wurden zahlreiche Experimente durchgeführt, die gezeigt haben, dass die entwickelte hierarchisch-iterative Methode für die Objekt- und Lageerkennung die Aufgabe der Registrierung von Tomogrammen effizient und mit hinreichender Genauigkeit lösen kann.

Als zu vergleichende Methoden wurden folgende Methoden gewählt:

- Klassisches Matching-Verfahren: Das Hauptprinzip dieser Methode besteht in der Überlagerung zweier Tomogramme und Bewertung dieser Überlagerung mit Hilfe eines Gütekriteriums, das basierend auf dem 2D-Grauwert-Histogramm berechnet wird. Für die Suche nach der besten Überlagerung werden alle möglichen Positionen dieser Tomogramme geprüft, d.h. jede von 6 Koordinaten (3 Translationen und 3 Rotationen) wird innerhalb eines Intervalls mit bestimmter Schrittweite verändert.
- Probabilistisches Verfahren ([70]): Es benutzt dasselbe Gütekriterium wie ein klassisches Matching-Verfahren und sucht nach der besten Position von Tomogrammen mit Hilfe einer linearen Optimierung der Gütefunktion.
- Hierarchisch-iterative Methode für die Objekt- und Lageerkennung (HIMOLE)

Für den Vergleich wurden folgende Kriterien verwendet:

1. Durchschnittliche Zeit zur Erkennung eines Objektes (K_1)
2. Durchschnittlicher Fehler der Lageerkennung, die durch die euklidische Distanz der berechneten und manuell bestimmten Lage eines Objektes definiert ist. Der Fehler teilt sich auf in den euklidischen Abstand der Tomogrammschwerpunkte (K_2) und den Winkelfehler (K_3):

$$\begin{aligned} K_2 &= \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \\ K_3 &= \sqrt{\Delta \alpha^2 + \Delta \beta^2 + \Delta \gamma^2} \end{aligned} \quad (4.10)$$

3. Anzahl der korrekten Erkennungen (in %), d.h. die Anzahl von Objekten, deren Klassen richtig erkannt wurden (K_4)

Als experimentelle Daten wurden 10 Datensätze verwendet, von denen jeder 3 Tomogramme enthält. Die Ergebnisse des Vergleichs sind in Tabelle 4.1 dargestellt.

Nach der Analyse der experimentellen Ergebnisse aus der Tabelle 4.1 zeigt sich, dass der Einsatz der hierarchisch-iterativen Methode im Vergleich zum klassischen Matching-Verfahren

4. Systeme für die Objekt- und Lageerkennung

Tabelle 4.1: Experimentelle Ergebnisse (Registrierung medizinischer Bilddatensätze)

Methode	Durchschnittliche Erkennungszeit [s] (K_1)	Fehler (Translation) [mm] (K_2)	Fehler Rotation [Grad] (K_3)	Anzahl der korrekten Erkennungen (K_4)
Klassisches Matching-Verfahren	408	12,42	2,26	3
Probabilistisches Matching	64,4	7	1,1	7,5
Hierarchisch-iterative Methode	56	1,92	0,81	9,8

und probabilistischen Matching ermöglicht hat, die Erkennungszeit um den Faktor 7,2 bzw. 1,15, den Fehler der Positionsbestimmung um den Faktor 6,4 bzw. 2,6, den Fehler der Orientierungsbestimmung um den Faktor 3,6 bzw. 1,4 zu reduzieren, sowie die Anzahl der korrekten Erkennungen um den Faktor 2,3 bzw. 1,3 zu erhöhen.

In der Abbildung 4.9 ist die benötigte Erkennungszeit der verschiedenen Methoden dargestellt.

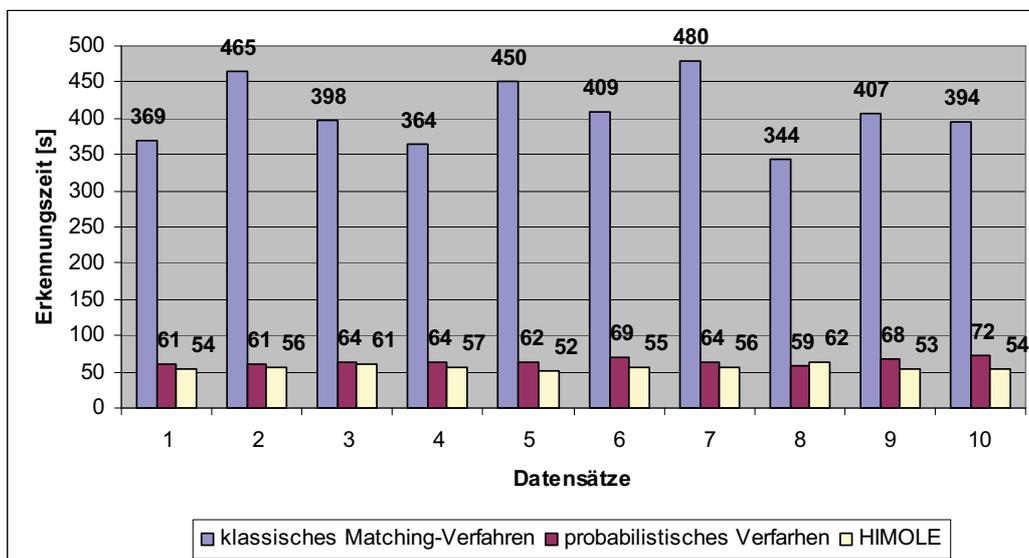


Abbildung 4.9: Erkennungszeit bei der Registrierung von Tomogrammen

In den Abbildungen 4.10 und 4.11 ist die Genauigkeit der Positions- und Orientierungserkennung der verschiedenen Methoden dargestellt.

In der Abbildung 4.12 ist die Anzahl der korrekten Erkennungen mit verschiedenen Methoden dargestellt.

Die visuelle Ergebnisse der Registrierung von Tomogrammen sind in der Abbildungen 4.13

4.2 Entwicklung des Lageerkennungssystems für die Registrierung medizinischer Bilddatensätze

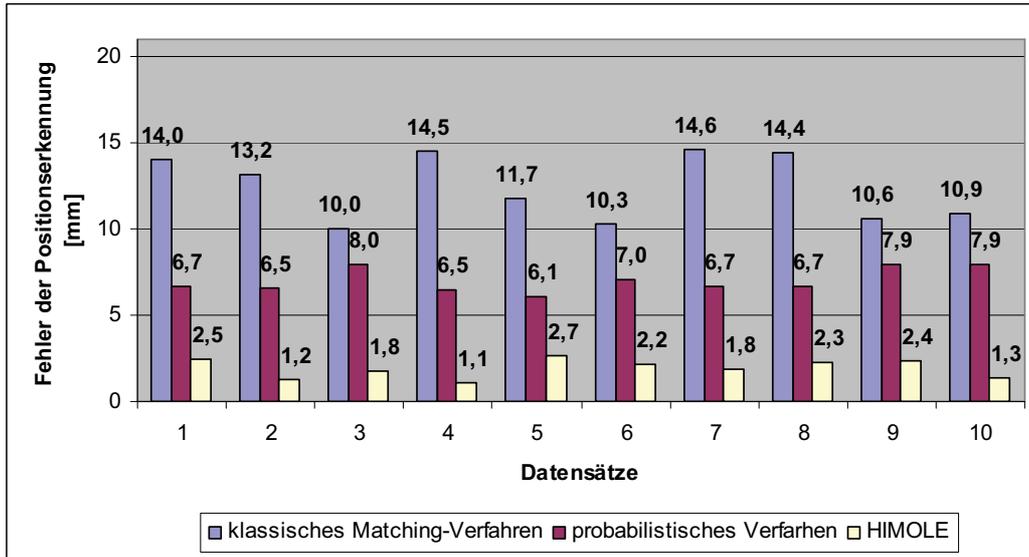


Abbildung 4.10: Genauigkeit der Positionserkennung bei der Registrierung von Tomogrammen

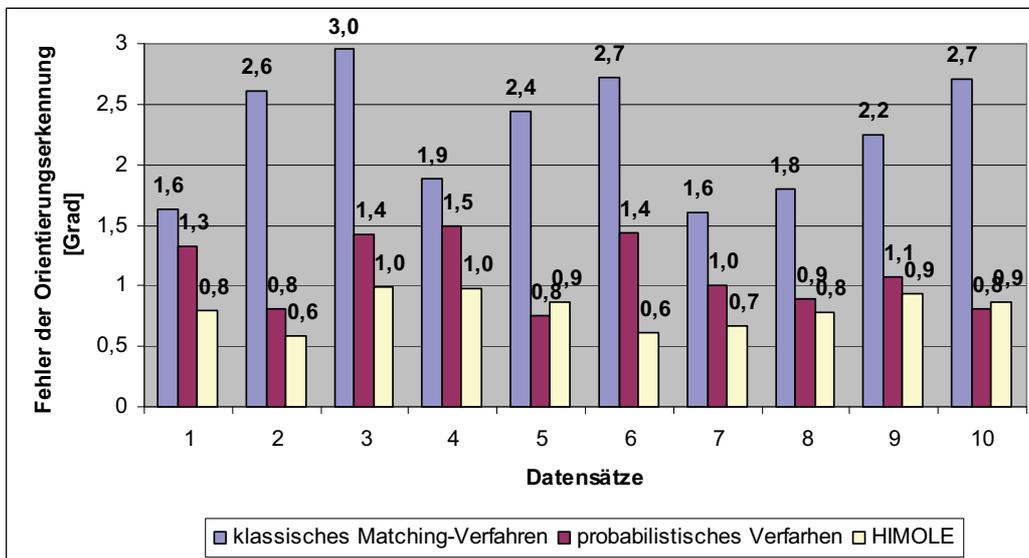


Abbildung 4.11: Genauigkeit der Orientierungserkennung bei der Registrierung von Tomogrammen

und 4.14 dargestellt.

4. Systeme für die Objekt- und Lageerkennung

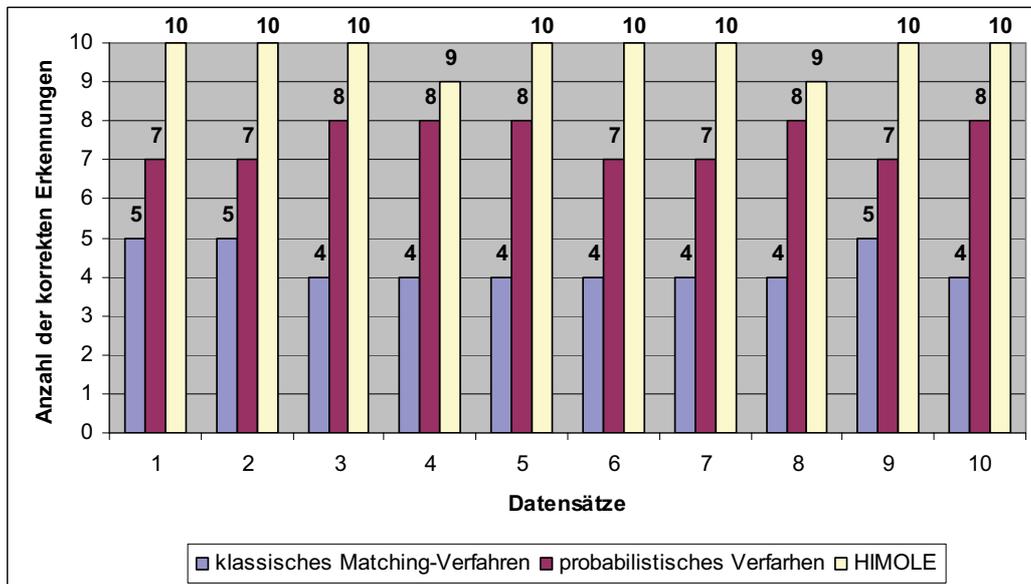


Abbildung 4.12: Anzahl der korrekten Erkennungen bei der Registrierung von Tomogrammen

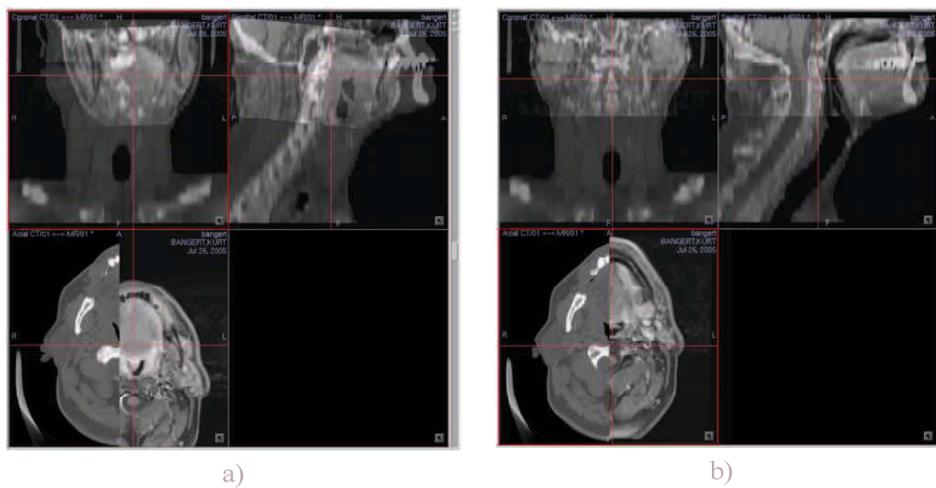


Abbildung 4.13: Registrierung von Tomogrammen mit dem probabilistischen Verfahren (a) und mit der hierarchischen iterativen Methode (b)

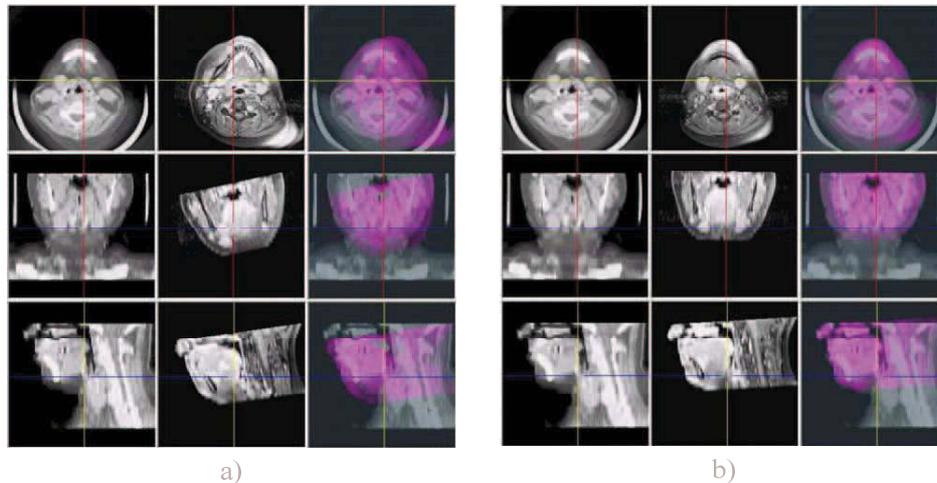


Abbildung 4.14: Registrierung von Tomogrammen mit dem probabilistischen Verfahren (a) und mit der hierarchischen iterativen Methode (b)

4.3 Entwicklung des Objekterkennungssystems für die Schrifterkennung

4.3.1 Problemstellung

Das zweite Problem, das mit Hilfe von der hierarchisch-iterativen Methode effizient gelöst wurde, ist die Schrifterkennung in der Industrie ([79]). Das Problem ist vergleichbar mit der Aufgabe von Schrifterkennung von gedruckten Texten, weist aber einige Besonderheiten auf.

Die wichtigsten Anforderungen in dieser Aufgabenklasse sind Zuverlässigkeit und Robustheit, weil sogar einzelne Fehler bei der Erkennung von kurzen Texten ein großes Problem darstellen können. Algorithmen, die in den industriellen Systemen verwendet werden, müssen gegen verschiedene Defekte, die sich aus Verschiebungen und Deformationen von Objekten, Verzerrungen, Rauschen, Farb- und Belichtungsänderungen, Verschmutzungen von Werkstücken verursacht sind, robust sein. Algorithmen für die Schrifterkennung gedruckter Texte zeigen hier schlechte Ergebnisse.

Das Problem der Schrifterkennung gehört zu der dritten Klasse der Erkennungsaufgaben: Erkennung mehrerer Objekte.

Die Methoden der Schrifterkennung können in folgende Klassen eingeteilt werden:

- musterbasierte Systeme
- strukturbasierte Systeme
- merkmalsbasierte Systeme
- Neuronale Netze

Alle diese Systeme haben sowohl Vorteile als auch die Nachteile:

- Musterbasierte Systeme sind robust gegen der kleinen Defekten von Bildern und haben eine ziemlich hohe Erkennungsgeschwindigkeit. Allerdings können sogar minimale Verzerrungen von Bildern, die zur Veränderung der Schrift führen, die Erkennungsergebnisse negativ beeinflussen.
- Strukturbasierte Systeme sind sehr empfindlich gegenüber Defekten von Bildern. Außerdem gibt es für diese Systeme im Gegensatz zu den musterbasierten und merkmalsbasierten Systemen keine effizienten automatisierten Lernverfahren.
- Merkmalbasierte Systeme erreichen bei Eingangsdaten mit einer Vielzahl an kleinen Defekten schlechte Erkennungsraten.
- Neuronale Netze, die in der Lage sind, verschiedene Schriften mit Defekten und Verzerrungen zu erkennen, müssen eine komplexe, mehrschichtige Struktur haben und erfordern das dauerhafte Lernen unter Verwendung einer hohen Anzahl an Lernstichproben. Unter Produktionsbedingungen ist dies kaum zu erfüllen.

Nach der Analyse von existierenden Methoden zur Schrifterkennung mit der Berücksichtigung der oben genannten Anforderungen (Zuverlässigkeit und Robustheit) wurde beschlossen, die entwickelte hierarchisch-iterative Methode für die Objekt- und Lageerkennung einzusetzen.

Die durchgeführten Experimente haben gezeigt, dass das Hauptproblem bei der Schrifterkennung in der Industrie die starke Verzerrung von Symbolen ist. Dies verhindert einen Vergleich dieser Symbole mit den Schablonen. Dieses Problem kann durch den Einsatz von Algorithmen, die zu der hierarchisch-iterativen Methode gehören, gelöst werden:

- Deformation der Schablone des zu erkennenden Symbols (Algorithmus der Deformation von Objektprototypen)
- Variation der Filterparameter während des Matching-Prozesses (Algorithmus des iterativen Matchings)

Der Algorithmus der Schrifterkennung unter Verwendung von der hierarchisch-iterativen Methode ist in der Abbildung 4.15 dargestellt.

4.3.2 Architektur des Lageerkennungssystems

Basierend auf den in Kapitel 4.1 beschriebenen Schritten bei der Entwicklung eines Objekt- und Lageerkennungssystems wurde das System für die Schrifterkennung entwickelt und implementiert.

Die Klasse der Eingangsdaten

Eingangsdaten sind Bilder von Werkstücken, die die Seriennummer dieser Werkstücke enthalten, d.h. 2D-Grauwert- oder Farbbilder (s. Abb. 4.16).

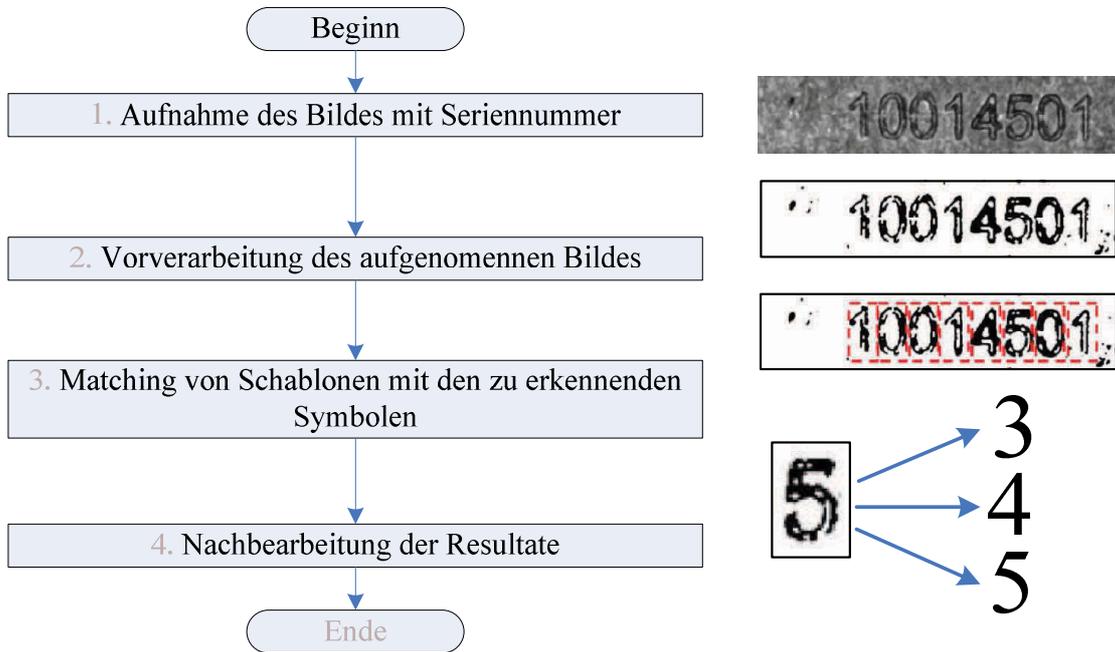


Abbildung 4.15: Algorithmus der Schrifterkennung



Abbildung 4.16: Bilder mit Seriennummern

Die Methode für Bestimmung und Erstellung von Objektprototypen

Als Prototyp des zu erkennenden Objektes wird eine Schablone des zu erkennenden Symbols verwendet, die durch den Schrifttyp und Schriftgröße gegeben ist und als 2D-Binärbild dargestellt ist.

0123456789abc

Abbildung 4.17: Schablone der zu erkennenden Symbole

Wesentliche Merkmale von Objektprototypen

Da die aufgenommenen Bilder der Seriennummern sehr stark verzerrt und verrauscht sind, ist die Verwendung von Merkmalen nicht sinnvoll.

Gütekriterien

Bei der Schrifterkennung werden folgende Gütekriterien verwendet:

1. Summe der quadratischen Abstände:

$$F_{SSD} = \sum_{i=0}^I \sum_{j=0}^J (f_d^m(x+i, y+j) \cdot \bar{M}(i, j))^2 \quad (4.11)$$

mit f_d^m - Distanzmatrix, die aus dem Bild, das die Seriennummer enthält, berechnet wurde; (x, y, z) - Lage der Symbolschablone M im Bezug auf Distanzmatrix; \bar{M} kann durch folgende Formel berechnet werden:

$$\bar{M} = T(M, t_x, t_y, \gamma, s_x, s_y, s_z) \quad (4.12)$$

mit M - Schablone eines Symbols; T - Funktion zur Transformation dieser Schablone; t_x, t_y - Translationen; γ - Rotation um z -Achse; s_x, s_y, s_z - Skalierungsfaktoren.

Unter Verwendung von Zeigermatrix f_p^m wird F_{SSD} folgendermaßen berechnet:

$$F_{SSD} = \sum_{i=0}^I \sum_{j=0}^J (\sqrt{(f_p^m(x+i, y+j)_x - i)^2 + (f_p^m(x+i, y+j)_y - j)^2} \cdot \bar{M}(i, j)) \quad (4.13)$$

Kurz kann man diese Formel folgendermaßen definieren:

$$F_{SSD} = \sum_{i=0}^I \sum_{j=0}^J d_{NN}(R^2, \bar{M}, x, y, i, j)^2 \quad (4.14)$$

mit d_{NN} - Abstand von einem Punkt der Symbolschablone \bar{M} mit Koordinaten (i, j) bis zum nächstgelegenen Punkt des Bildes R^2 , das Seriennummer enthält.

2. Korrelationskoeffizient:

$$F_{CC} = \frac{N_e}{N_t} \cdot \frac{N_e}{N_c} = \frac{(\sum_{i=0}^I \sum_{j=0}^J (R^2(x+i, y+j) \cdot \bar{M}(i, j)))^2}{\sum_{i=0}^I \sum_{j=0}^J R^2(x+i, y+j) \cdot \sum_{i=0}^I \sum_{j=0}^J \bar{M}(i, j)} \quad (4.15)$$

mit N_e - Anzahl der Punkten von Symbolschablone, die den Punkten des zu erkennenden Symbols entsprechen; N_t - Anzahl aller Punkte von Symbolschablone; N_c - Anzahl der Punkten eines Bildbereiches, das durch die Schablone überlappt ist; R^2 - 2D-Bild, das eine Seriennummer enthält.

Filter für die Vorverarbeitung der Eingangsdaten

Für die Rauschunterdrückung werden im entwickelten System folgende Filter eingesetzt:

- Filter der anisotropischen Diffusion (s. Abb. 4.18)
- Binärfilter (s. Abb. 4.19)
- morphologische Filter (s. Abb. 4.19)



Abbildung 4.18: Originales Bild mit der Seriennummer (a) und das mit dem anisotropischen Diffusionsfilter verarbeitete Bild (b)

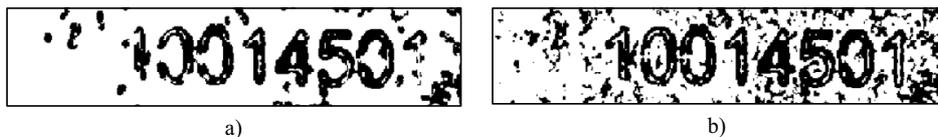


Abbildung 4.19: Bild mit der Seriennummer, das mit dem Binärfilter (a) und einem morphologischen Filter (b) verarbeitet wurde

Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen

Da die Bilder, die die Seriennummer von Werkstücke enthalten, wesentlich kleiner sind als Tomogramme, werden nur zwei Stufen der hierarchischen Optimierung verwendet: lokale und präzise Suche.

Folgende Dimensionen und ihre Kombinationen werden benutzt:

- Translation entlang x -, y - und z -Achse
- Rotation um z -, sowie auch um x - und y -Achse
- Translation entlang folgender Richtungen: (x, y) und $(x, -y)$

Als nicht-räumliche parametrische Koordinaten werden Verzerrungsfaktoren des Tomogramms verwendet (s_x und s_y).

Zwei Gütekriterien werden verwendet:

- Summe der quadratischen Abstände - F_{SSD} (s. Formel 4.11).

4. Systeme für die Objekt- und Lageerkennung

- Korrelationskoeffizient - F_{CC} (s. Formel 4.15)).

Diese Kriterien werden aufeinanderfolgend verwendet, d.h. bei der lokalen Suche wird das Kriterium F_{SSD} (Summe der quadratischen Abstände) und bei der präzisen Suche das Kriterium F_{CC} (Korrelationskoeffizient) berechnet.

Die Gütefunktionen, die auf den obenbeschriebenen Gütekriterien basieren, haben folgende Form:

$$F = f(R^2, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y) \quad (4.16)$$

Algorithmus des hierarchischen Matchings

Unter Verwendung des Algorithmus des hierarchischen Matchings wird das Bild mit der Seriennummer als Pyramide dargestellt. Um eine solche Pyramidendarstellung zu erstellen, wird der Gauß-Filter eingesetzt, der das Bild ohne Verlust von wesentlicher Information reduzieren kann (s. Abb. 4.20).

Auflösung	Größe	Symbol	Größe	Symbol	Punktenanzahl
originale	$m \times m$		$n \times n$		N
$1/k$	$m/k \times m/k$		$n/k \times n/k$		N/k^2

Abbildung 4.20: Gauß-Pyramide von Symbolen

Da ein Bild mit Seriennummer vergleichsweise wenig Bildpunkte enthält, wird eine Pyramide verwendet, die nur drei Stufen hat. Auf diese Weise steigt der Speicherbedarf für eine solche Pyramidendarstellung nur um ca. 3% im Vergleich zu den Originaldaten (s. Formel 3.13):

Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten

Um die Geschwindigkeit der Berechnung von Gütekriterien zu erhöhen, wird eine Distanz- oder Zeigermatrix verwendet. Da das Bild, das die Seriennummer enthält, ziemlich klein ist (normalerweise 200×50 Pixel), ist es in diesem Fall sinnvoll, eine Zeigermatrix einzusetzen (s. Abb. 4.21).

Eine solche Zeigermatrix wird für die Berechnung aller Gütekriterien (sowohl für die Summe der quadratischen Abstände als auch für den Korrelationskoeffizienten) verwendet.

Algorithmus der Deformation des Objektprototyps

Bei der Schrifterkennung werden zwei Gütekriterien F_{SSD} und F_{CC} verwendet. Jedes Kriterium basiert auf der Gütefunktion, die als ihre Argumente folgende, nichträumliche para-

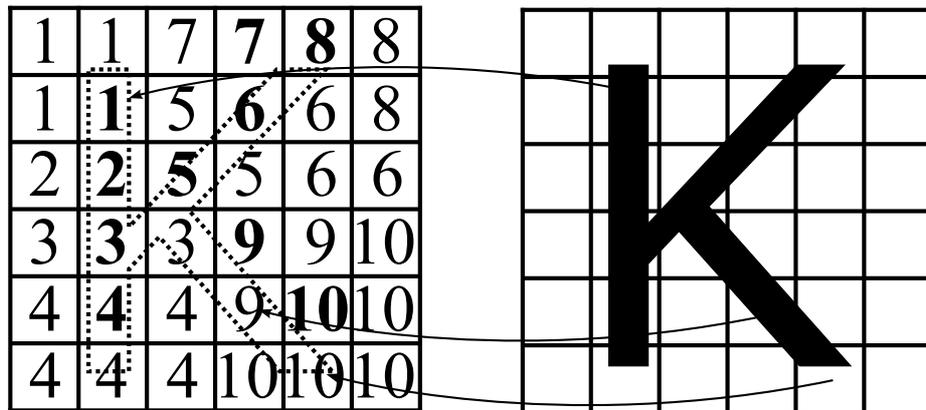


Abbildung 4.21: Verwendung der Zeigermatrix

metrische Koordinaten von Objektprototypen, die die mögliche Deformation von Symbolen darstellen können, annimmt (s. Abb. 4.22):

- Rotation um x -Achse - α
- Rotation um y -Achse - β
- Verzerrungsfaktor entlang x -Achse - s_x
- Verzerrungsfaktor entlang y -Achse - s_y
- Verzerrungsfaktor entlang z -Achse - s_z

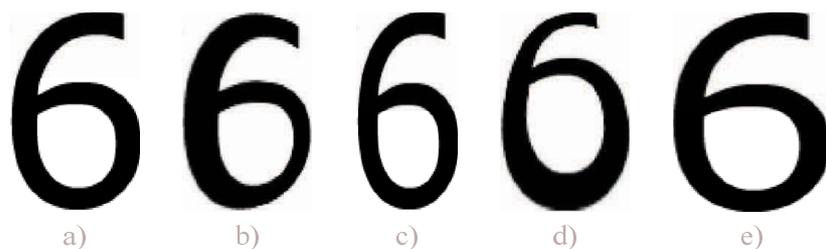


Abbildung 4.22: Mögliche Deformationen einer Symbolschablone: a) originale Schablone; b) Rotation um y -Achse; c) Verzerrung um y -Achse; d) Rotation um x -Achse; e) Verzerrung um x -Achse

Algorithmus des iterativen Matchings

Folgende Filter der Vorverarbeitung der Daten werden verwendet:

- anisotropisches Diffusionsfilter

4. Systeme für die Objekt- und Lageerkennung

- Binärfilter
- morphologische Filter

Die Parameter dieser Filter, die während des Matching-Prozesses verändert werden, sind die folgenden:

- Kerngröße des Filters
- *Diffusion conductance*
- Anzahl der Iterationen
- Schwellwert für die Binarisierung

In der Abbildung 4.23 sind das ursprüngliche Grauwertbild (a) und drei Binärbilder (b, c, d) dargestellt. Diese Binärbilder wurden mit verschiedenen Schwellwerten berechnet. Es ist zu sehen, dass nicht jede Zahl bei einem Schwellwert erkannt werden kann: „0“ kann man leicht mit dem Schwellwert von 70% erkennen; „36“ können korrekt mit dem Schwellwert von 75% erkannt werden, etc.

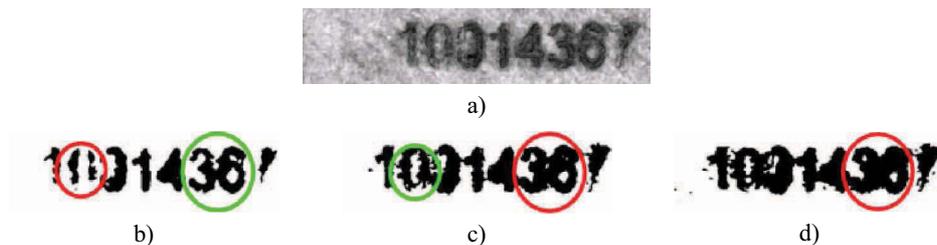


Abbildung 4.23: Der Einfluss des Binarisierungsschwellwert: a) Originalbild; b) Binärbild mit dem Schwellwert von 75% (75% der weissen Pixel); c) Binärbild mit dem Schwellwert von 70%; d) Binärbild mit dem Schwellwert von 65%

Algorithmus der Hierarchisierung von Objektprototypen

Da die Bilder, die die Seriennummer enthalten, sowie die Symbolschablonen eine niedrige Auflösung haben, ist es nicht sinnvoll, die Hierarchisierung von Objektprototypen durchzuführen.

Lernverfahren

Das System für die Schrifterkennung wird mit einem Vorablernverfahren trainiert. Eine Lernstichprobe von Bildern, die verschiedene bekannte Seriennummern enthalten, wurde bestimmt, mit der die initialen Parameter der Algorithmen und Filter berechnet wurden. Aufgrund dieser Lernstichprobe wurde das System für die Erkennung unbekannter Seriennummern angepasst.

4.3.3 Experimentelle Ergebnisse

Um die Güte des entwickelten Systems zu bewerten, wurden mehrere Experimente mit existierenden Systemen für die Schrifterkennung durchgeführt. Die folgenden Systeme wurden ausgewählt: *SimpleOCR*, *GOOCR*, *TopOCR*, *Readiris Pro*, *ScreenOCR*, *OpenOCR*, *Finereader*, *TOCR*.

Für den Vergleich wurden folgende Kriterien verwendet:

1. Durchschnittliche Zeit zur Erkennung eines Objektes (K_1)
2. Anzahl der korrekten Erkennungen (in %), d.h. die Anzahl von Objekten, deren Klassen richtig erkannt wurden (K_2)

Als experimentelle Daten wurden 10 Datensätze verwendet, von denen jeder 10 Bilder enthält. Die Ergebnisse des Vergleichs sind in der Tabelle 4.2 dargestellt.

Tabelle 4.2: Experimentelle Ergebnisse (Schrifterkennung)

Methode	Durchschnittliche Erkennungszeit [ms] (K_1)	Anzahl der korrekten Erkennungen (K_2)
SimpleOCR	204,7	25,3
GOOCR	274,6	23,1
TopOCR	99,5	30,1
Readiris Pro	190,3	5,9
ScreenOCR	138	34,2
OpenOCR	66,7	24,8
Finereader	61,1	35
TOCR	125	55,6
Hierarchische iterative Methode	205,1	99,7

In der Abbildung 4.24 ist die benötigte Erkennungszeit der verschiedenen Methoden dargestellt.

In der Abbildung 4.25 ist die Anzahl der korrekten Erkennungen der verschiedenen Methoden dargestellt.

In der Abbildung 4.26 werden die Methoden verglichen.

Aus der Tabelle 4.2 und aus den Abbildung 4.26 lässt sich ersehen, dass *TOCR* das effizienteste der Vergleichssysteme ist. Dieses System hat die besten Ergebnisse bei der Schrifterkennung geliefert. Der Einsatz des Systems, das auf der hierarchisch-iterativen Methode basiert, hat ermöglicht, die Anzahl der korrekten Erkennungen um den Faktor 1,8, was 99,7% aller bearbeiteten Bilder entspricht, im Vergleich zu den *TOCR*-Ergebnissen zu erhöhen.

In der Tabelle 4.3 sind Beispiele der Auswertungsergebnisse mit den HIMOLE- und *TOCR*-System dargestellt.

4. Systeme für die Objekt- und Lagererkennung

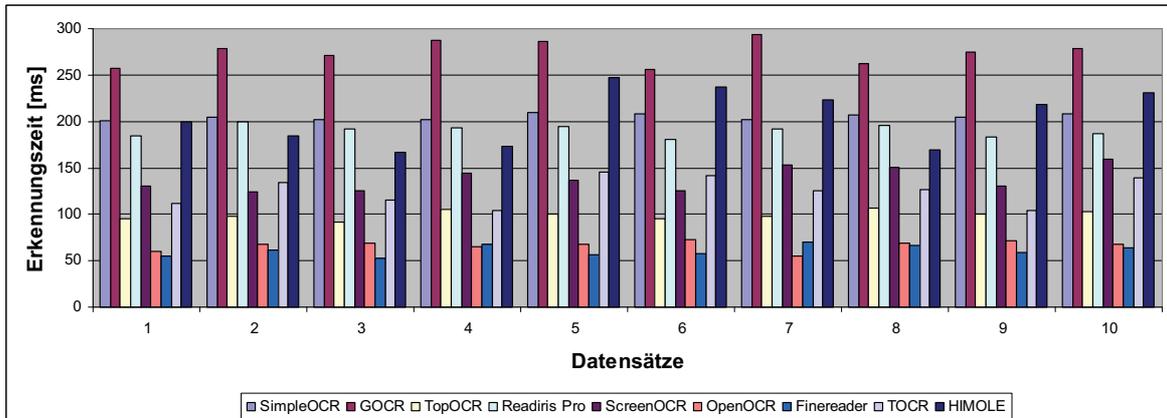


Abbildung 4.24: Erkennungszeit bei der Schrifterkennung

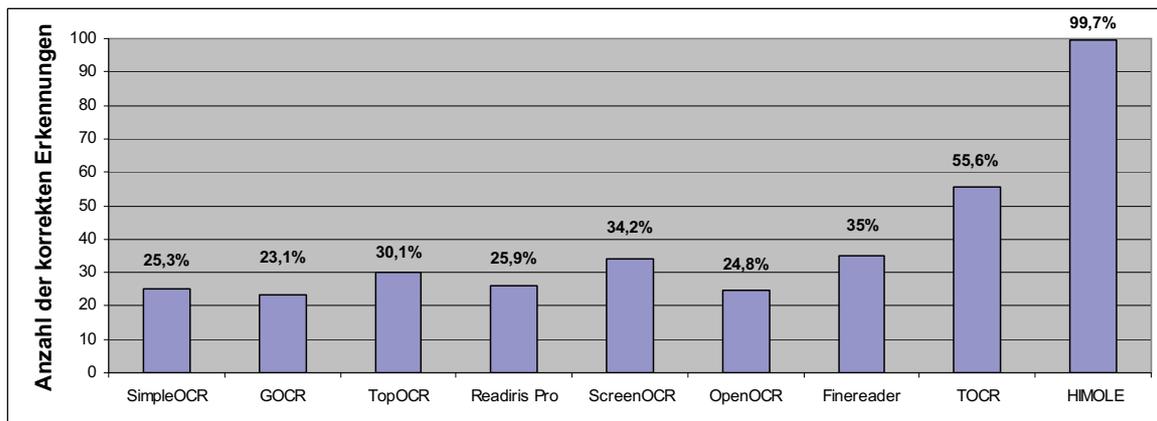


Abbildung 4.25: Anzahl der korrekten Erkennungen bei der Schrifterkennung

Aus der Tabelle 4.3 lässt sich ersehen, dass die Genauigkeit der Schrifterkennung unter Verwendung des HIMOLE-Systems wesentlich gesteigert werden konnte. Es wurde 8 Beispiele betrachtet, 7 von denen vom HIMOLE-System korrekt erkannt wurden. TOCR hat bei den 7 Fällen fehlgeschlagen. Diese Verbesserung wird allerdings mit einer Erhöhung der Erkennungszeit erkaufte, was aber für die Applikation von untergeordneter Bedeutung ist, weil der Fokus auf eine hohe Erkennungsrate lag.

Die visuellen Ergebnisse der Schrifterkennung mit den HIMOLE- und TOCR-Systemen sind in der Abbildungen 4.27 dargestellt.

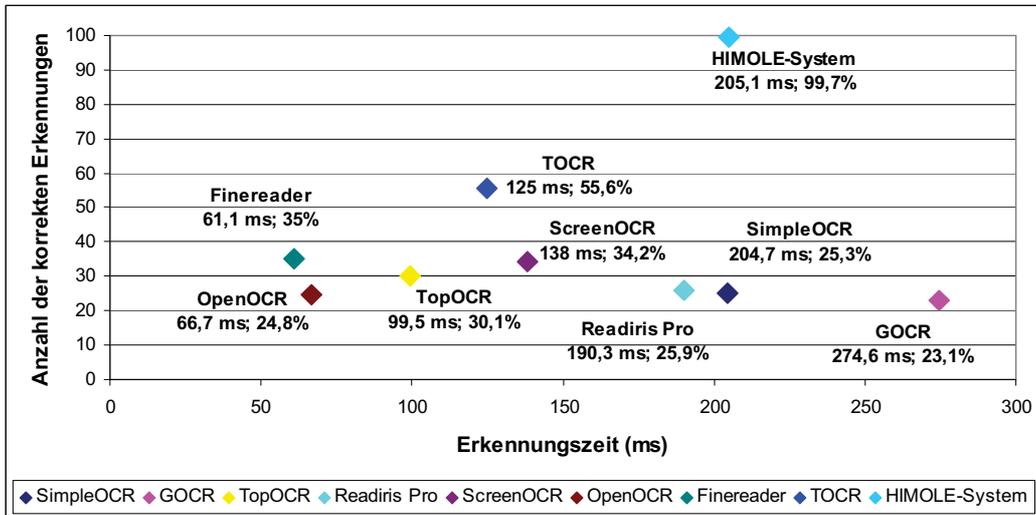


Abbildung 4.26: Vergleich der Methoden/Systeme für die Schrifterkennung

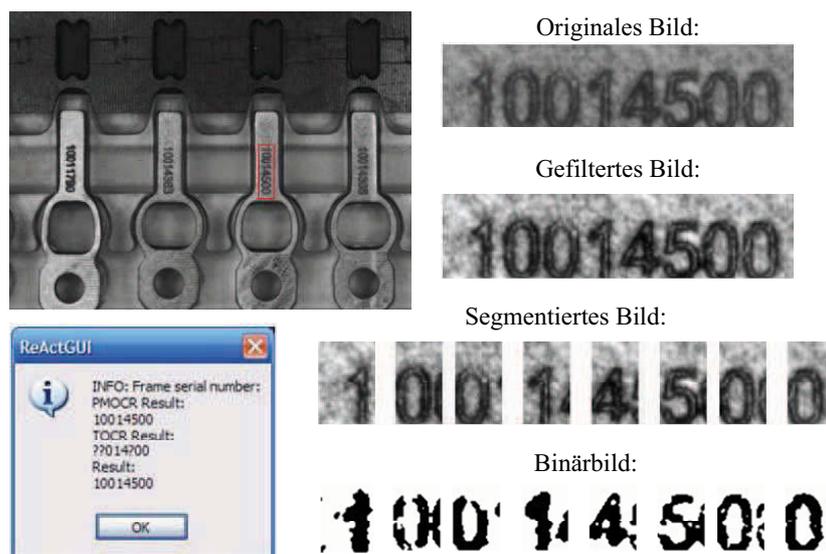


Abbildung 4.27: Schrifterkennung mit den HIMOLE- und TOCR-Systemen

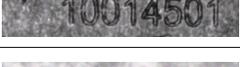
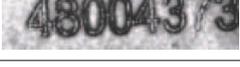
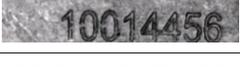
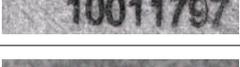
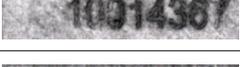
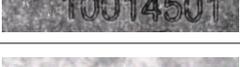
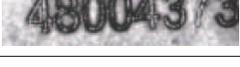
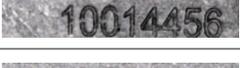
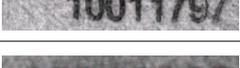
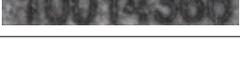
4.4 Entwicklung des Objekterkennungssystems für das Griff-in-die-Kiste-Problem

4.4.1 Problemstellung

Das dritte Problem, das mit Hilfe der hierarchisch-iterativen Methode erfolgreich gelöst wurde, ist das Problem der Erkennung mehrerer Objekte verschiedener Klassen im Rahmen eines Griff-in-die-Kiste-Systems ([81, 80, 59]). Dieses Problem gehört zu der vierten

4. Systeme für die Objekt- und Lageerkennung

Tabelle 4.3: Schrifterkennungen mit den HIMOLE- und TOCR-Systemen

System	Bild mit der Seriennummer	Zeit	Erkannter Text
HIMOLE-System		125 ms	10014353
		122 ms	10014367
		154 ms	1001436 ¹
		137 ms	10014501
		166 ms	48004373
		175 ms	10014456
		127 ms	10011797
		162 ms	10014366
TOCR		52 ms	10014353
		63 ms	1001436 ¹
		54 ms	10 ³ 1436 ¹
		69 ms	10014501
		43 ms	4 ³ 0043 ¹ 3
		56 ms	10014 ¹ 56
		67 ms	1001179 ⁴
		62 ms	1001 ¹⁶ 66

Klasse der Objekterkennungsaufgaben: Erkennung mehrerer Objekte und ihrer Lagen in der Szene.

Architektur eines Griff-in-die-Kiste-Systems ist in der Abbildung 4.28 dargestellt.

Dieses Problem ist ein sehr bekanntes Problem in der Robotik und es gibt viele verschiede-

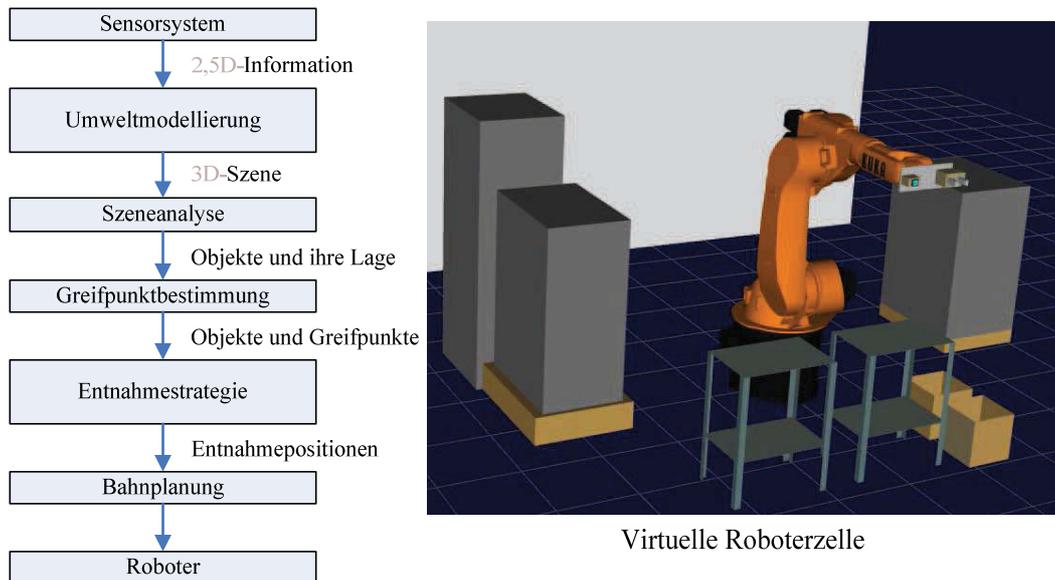


Abbildung 4.28: Architektur eines Griff-in-die-Kiste-Systems

ne Lösungen dieses Problems, von denen die meistens auf Model- oder Feature-Matching basieren ([60, 46, 55, 61, 62, 63, 64]).

4.4.2 Architektur des Objekt- und Lageerkennungssystems

Basierend auf den in Kapitel 4.1 Schritten bei der Entwicklung eines Objekt- und Lageerkennungssystems wurde das System für die Erkennung mehrerer Objekte verschiedener Klassen und ihrer Lage entwickelt und implementiert.

Die Klasse der Eingangsdaten

Die Eingangsdaten sind in diesem Fall Punktwolken im 3D-Raum: Ein Voxelmodell, das mit Hilfe eines Sensorsystems (PMD-Kamera [105]), aufgenommen wurde und eine 3D-Szene, die eine Anhäufung von Objekten verschiedener Klassen enthält, beschreibt, d.h. 3D-Daten, die Binärinformation enthalten.

In der Abbildung 4.29 sind 2,5D-Bilder, die mit Hilfe einer PMD-Kamera aufgenommen wurden, dargestellt.

Die Methode für Bestimmung und Erstellung von Objektprototypen

Als Prototyp des zu erkennenden Objektes wird ein 3D-Voxelmodell verwendet, das folgendermaßen erzeugt werden kann:

- automatisch: Es wird ein 3D-Modell eines idealen Objektes erstellt, das alle Objekte dieser Klasse beschreibt.

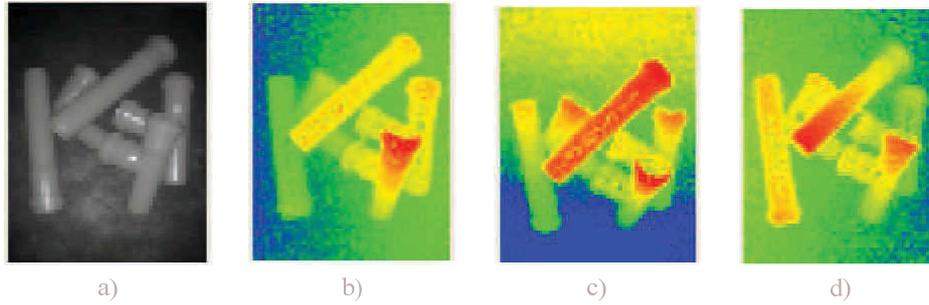


Abbildung 4.29: Grauwertbild einer Szene (a); 2,5D-Bilder der selben Szene (b, c, d)

- mit Hilfe eines CAD-Modells.

Wesentliche Merkmale von Objektprototypen

Die wesentlichen Merkmale von Objektprototypen sind geometrische Merkmale wie Konturen, Oberflächen, Kreise.

Die Hierarchie der geometrischen Merkmale wird manuell vorgegeben, wenn CAD-Modelle von zu erkennenden Objekten verwendet werden, oder kann automatisch bei der Erstellung des 3D-Modells eines Prototyps berechnet werden.

Gütekriterien

Die folgenden Gütekriterien wurden verwendet:

1. Gewichtete Summe der quadratischen Abstände:

$$F_{WSSD} = \frac{\sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K (f_d^m(x+i, y+j, z+k) \cdot \bar{M}(i, j, k))^2}{\sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \bar{M}(i, j, k)} \quad (4.17)$$

mit f_d^m - Distanzmatrix der 3D-Szene; (x, y, z) - Lage des Objektprototyps \bar{M} im Bezug auf Distanzmatrix; \bar{M} kann durch folgende Formel berechnet werden:

$$\bar{M} = T(M, \alpha, \beta, \gamma, s_x, s_y, s_z) \quad (4.18)$$

mit M - Prototyp eines 3D-Objektes; T - Funktion zur Transformation dieses Prototyps; α, β, γ - Rotationen; s_x, s_y, s_z - Skalierungsfaktoren.

Kurz kann F_{WSSD} folgendermaßen definiert werden:

$$F_{WSSD} = \frac{\sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K d_{NN}(R^3, \bar{M}, x, y, z, i, j, k)^2}{N} \quad (4.19)$$

mit d_{NN} - Abstand von einem Element des Objektprototyps M mit Koordinaten (i, j, k) bis zum nächstgelegenen Element der Oberfläche der 3D-Szene R^3 ; N - Anzahl von Elementen des Objektprototyps.

Unter Verwendung der Zeigermatrix f_p^m , wird d_{NN} folgendermaßen berechnet:

$$d_{NN}(R^3, \bar{M}, x, y, z, i, j, k) = ((f_p^m(x + i, y + j, z + k)_x - i)^2 + (f_p^m(x + i, y + j, z + k)_y - j)^2 + (f_p^m(x + i, y + j, z + k)_z - k)^2) \cdot \bar{M}(i, j, k) \quad (4.20)$$

2. Gewichtete Summe der absoluten Abstände:

$$F_{WSSRD} = \frac{\sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \sqrt{d_{NN}(R^3, \bar{M}, x, y, z, i, j, k)}}{N} \quad (4.21)$$

3. Korrelationskoeffizient (Überdeckungsgrad des Prototyps und der Daten):

$$F_{CC} = \frac{N_e}{N_t} = \frac{\sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K [(f_d^m(x + i, y + j, z + k) < \delta) \cdot \bar{M}(i, j, k)]}{\sum_{i=0}^I \sum_{j=0}^J \sum_{k=0}^K \bar{M}(i, j, k)} \quad (4.22)$$

mit N_e - Anzahl von Elementen des Objektprototyps, die nicht weiter als irgendein δ von den nächsten Elementen der 3D-Szene entfernt sind; N_t - Anzahl aller Elemente des Objektprototyps.

Unter Verwendung von Zeigermatrix f_p^m , wird N_e folgendermaßen berechnet:

$$N_e = \sum_i \sum_j \sum_k (((f_p^m(x + i, y + j, z + k)_x - i)^2 + (f_p^m(x + i, y + j, z + k)_y - j)^2 + (f_p^m(x + i, y + j, z + k)_z - k)^2)^{\frac{1}{2}} \cdot \bar{M}(i, j, k) < \delta) \quad (4.23)$$

Filter für die Vorverarbeitung der Eingangsdaten

Die Eingangsdaten im Fall der Verwendung einer PMD-Kamera sind eine Punktwolke als Voxelmodell der Szene. Um die aufgenommene Szene mit Hilfe der hierarchisch-iterativen Methode effizient zu analysieren, ist es sinnvoll, aus einem Voxelmodell ein Polygonmodell zu berechnen. Der Prozess der Umweltmodellierung (Berechnung des Polygonmodells aus den Tiefenbildern) ist in der Abbildung 4.30 dargestellt ([34, 16]).

Als Triangulationsmethoden werden 3D-Delaunay-Triangulierung und ein in [23] beschriebenes Verfahren verwendet.

Nachdem das Polygonmodell einer 3D-Szene berechnet wurde, ist es nötig, folgende Filterungen durchzuführen:

4. Systeme für die Objekt- und Lageerkennung

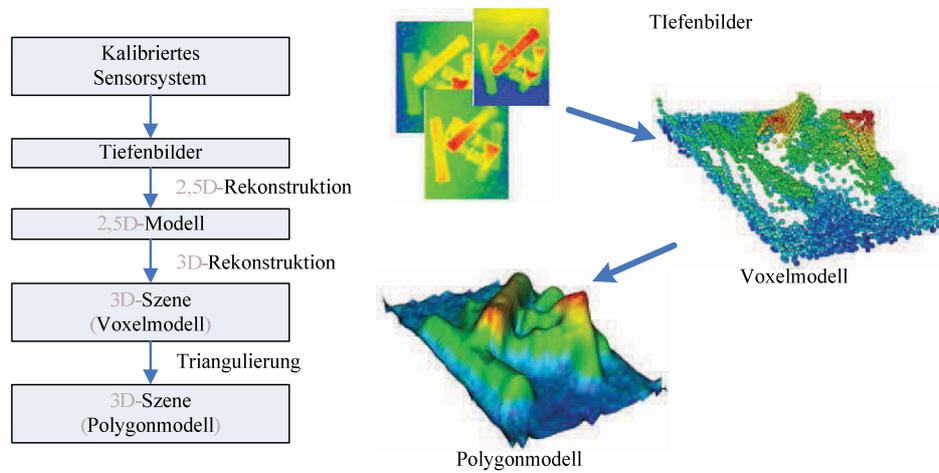


Abbildung 4.30: Umweltmodellierung

- Entfernung der Ausreißer. Aufgrund der Analyse von Oberflächen, die bei der Triangulierung berechnet wurden, werden Punkte und Dreiecke entfernt, die zu weit weg von den Hauptelementen der Szene liegen.
- Glättung der Oberflächen. Diese Glättung wird mit einem Laplace-Filter durchgeführt (s. Abb. 4.31).

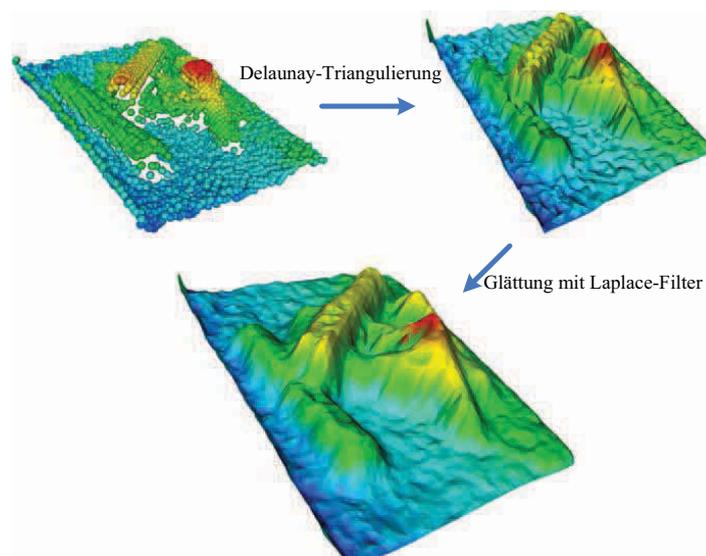


Abbildung 4.31: Glättung mit Laplace-Filter

Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen

Alle drei Stufen der hierarchischen Optimierung werden verwendet: globale, lokale und präzise Suche.

Folgende Dimensionen und ihre Kombinationen werden benutzt:

- Translation entlang x -, y - und z -Achsen
- Rotation um x -, y - und z -Achsen
- Translation entlang folgende Richtungen: (x, y) , $(-x, y)$, (x, y) , $(-x, z)$, (y, z) , $(-y, z)$, (x, y, z) , $(-x, y, z)$, $(-x, -y, z)$, $(x, -y, z)$

Als nichträumliche parametrische Koordinaten werden Verzerrungsfaktoren von Objektprototypen verwendet.

Drei Gütekriterien werden verwendet:

- Gewichtete Summe der quadratischen Abstände F_{WSSD} (s. Formel 4.17)
- Gewichtete Summe der absoluten Abstände F_{WSSRD} (s. Formel 4.21)
- Korrelationskoeffizient F_{CC} (s. Formel 4.22)

Diese Gütekriterien werden folgendermaßen verwendet:

- Aufeinanderfolgende Verwendung von Gütekriterien. Die Experimente haben gezeigt, dass die Gütekriterien bei den verschiedenen Stufen der Optimierung Ergebnisse mit verschiedener Genauigkeit liefern, z.B. bei der globalen Suche werden die besten Ergebnisse unter Verwendung des Kriteriums der gewichteten Summe der quadratischen Abstände (F_{WSSD}) erreicht. Bei der lokalen Suche ist die Verwendung des Kriteriums der gewichteten Summe der absoluten Abstände (F_{WSSRD}) sinnvoll. Bei der präzisen Suche wird das kombinierte Gütekriterium verwendet.
- Kombiniertes Gütekriterium bei der präzisen Suche. Dieses Kriterium basiert auf den zwei anderen Gütekriterien F_{WSSD} (gewichtete Summe der quadratischen Abstände) und F_{CC} (Korrelationskoeffizient) und wird durch folgende Formel definiert:

$$\begin{aligned}
 F &= f(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z) \\
 &= f(f_{WSSD}(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z), \\
 &\quad f_{CC}(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z)) \\
 &= f_{WSSD}(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z) \\
 &\quad \cdot f_{CC}(R^3, M, t_x, t_y, t_z, \alpha, \beta, \gamma, s_x, s_y, s_z)
 \end{aligned} \tag{4.24}$$

Algorithmus des hierarchischen Matchings

Unter Verwendung des Algorithmus des hierarchischen Matchings wird das Polygonmodell einer 3D-Szene als Pyramide dargestellt. Um eine solche Pyramidendarstellung zu erstellen, wird der Algorithmus für die Dezimierung von Dreiecksnetzen [85] eingesetzt. Dieser Algorithmus reduziert die Anzahl von Polygonen ohne den Verlust von wesentlicher Information (s. Abbildung 4.32).

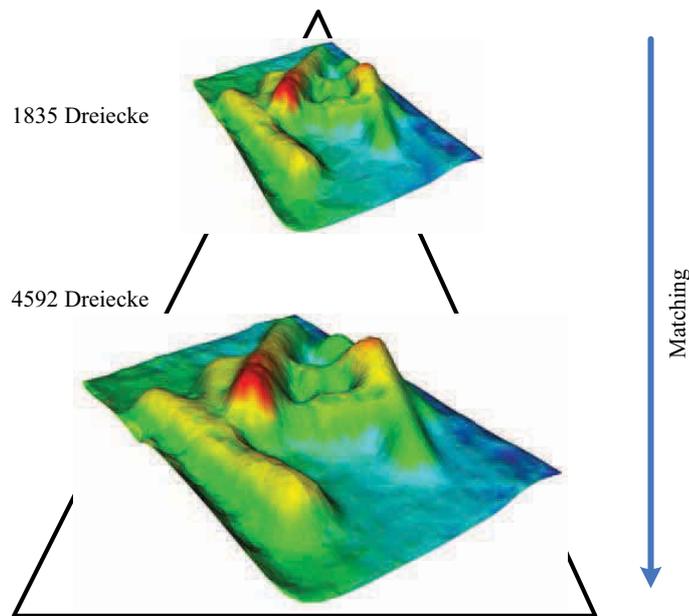


Abbildung 4.32: Pyramidendarstellung einer 3D-Szene

Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten

Um die Geschwindigkeit der Berechnung von Gütekriterien zu erhöhen, wird die Zeigermatrix (s. 3.3.3) verwendet. Da die Anzahl von Elemente einer 3D-Szene (Voxel oder Polygone) ziemlich groß ist, ist es sinnvoll, eine Distanzmatrix zu verwenden. Wenn die Anforderung an die Genauigkeit der Lageerkennung hoch ist, wird vorgeschlagen, eine Zeigermatrix bei der präzisen Suche zu verwenden.

Algorithmus der Deformation von Objektprototypen

Bei der Erkennung werden drei Gütekriterien verwendet: Summe der quadratischen Abstände (F_{WSSD}), Summe der absoluten Abstände (F_{WSSRD}) und Korrelationskoeffizient (F_{CC}). Jede Gütefunktion, die diese Kriterien berechnet, nimmt als Argumente folgende Koordinaten, die eine mögliche Deformation von Objekten darstellen können, an (s. Abb. ??):

- Verzerrungsfaktor (Skalierung) entlang der x -Achse - s_x
- Verzerrungsfaktor (Skalierung) entlang der y -Achse - s_y
- Verzerrungsfaktor (Skalierung) entlang der z -Achse - s_z

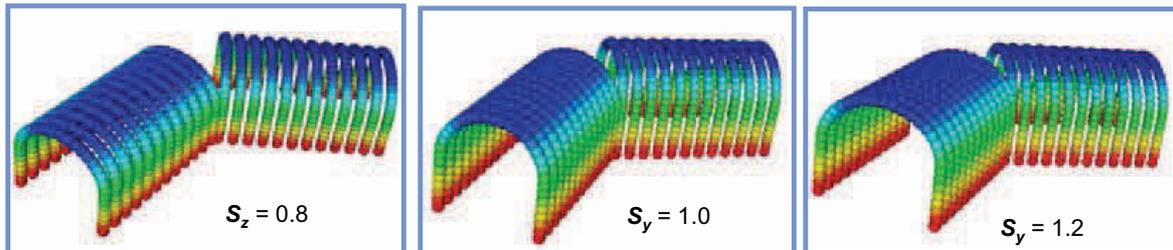


Abbildung 4.33: Deformationen des Prototyps

Algorithmus des iterativen Matchings

Folgende Algorithmen und Filter der Vorverarbeitung der Eingangsdaten werden verwendet:

- Delaunay-Triangulierung
- Laplace-Glättung [65, 33]
- Algorithmus der Dezimierung von Dreiecknetzen [85]
- Algorithmus der Entfernung von Ausreißern

Wesentliche Parameter dieser Filter, die im Prozess variiert werden, sind die folgende:

- Laplace-Glättung:
 - Anzahl von Iterationen
 - *Feature Angle*
 - Kantenwinkel (*Edge Angle*)
 - Relaxationswert (Intervall [0; 1])
- Dezimierung: Reduzierungsfaktor
- Algorithmus der Entfernung von Ausreißern: maximaler Abstand zwischen zwei Voxeln

Algorithmus der Hierarchisierung von Objektprototypen

1. Bei der automatischen Erstellung von Objektprototypen wird das Modell eines Prototyps in Bestandteile zerlegt, jede Komponente wird gewichtet und die Position im hierarchischen Modell wird bestimmt.
2. Bei der manuellen Erstellung von Objektprototypen als CAD-Modell wird das Modell manuell zerlegt und gewichtet.

Lernverfahren

Das System für die Objekt- und Lageerkennung wird mit dem Vorablernverfahren trainiert. Eine Lernstichprobe von 3D-Szenen, die verschiedene bekannte Objekte enthalten, wurde bestimmt, mit der die initiale Parameter der Algorithmen und Filter berechnet wurden. Aufgrund dieser Lernstichprobe wurde das System für die Erkennung anderer Objekte angepasst.

Es wurden zahlreiche Experimente durchgeführt, die gezeigt haben, dass die entwickelte hierarchisch-iterative Methode für die Objekt- und Lageerkennung das Griff-in-die-Kiste-Problem effizient und mit hinreichender Genauigkeit lösen kann.

Als mit der HIMOLE zu vergleichende Methode wurde das klassische Matching-Verfahren gewählt. Das Prinzip dieser Methode besteht in der Überlagerung von Objektprototypen und einer 3D-Szene und Bewertung dieser Überlagerung mit Hilfe von Gütekriterien, das als Summe der quadratischen Abstände definiert ist. Für die Suche der besten Überlagerung werden alle möglichen Positionen eines Objektprototyps im Bezug auf einer 3D-Szene geprüft, d.h. jede von 6 Koordinate (3 Translationen und 3 Rotationen) wird innerhalb eines Intervalls mit einer bestimmten Schrittweite verändert.

Für den Vergleich wurden folgende Kriterien verwendet:

1. Durchschnittliche Zeit zur Erkennung eines Objektes (K_1)
2. Durchschnittlicher Fehler der Lageerkennung, die durch die euklidische Distanz der berechneten und manuell bestimmten Lage eines Objektes definiert ist. Der Fehler teilt sich auf in den euklidischen Abstand der Tomogrammschwerpunkte (K_2) und den Winkelfehler (K_3):

$$\begin{aligned} K_2 &= \sqrt{\Delta x^2 + \Delta y^2 + \Delta z^2} \\ K_3 &= \sqrt{\Delta \alpha^2 + \Delta \beta^2 + \Delta \gamma^2} \end{aligned} \quad (4.25)$$

3. Anzahl der korrekten Erkennungen (in %), d.h. die Anzahl von Objekten, deren Klassen richtig erkannt wurden (K_4)

Aus der Tabelle 4.4 lässt sich ersehen, dass der Einsatz von der hierarchischen iterativen Methode hat im Vergleich zu dem klassischen Matching-Verfahren ermöglicht, die Erkennungszeit um den Faktor 7,2 zu reduzieren, den Fehler der Positions- und Orientierungsbestimmung um den Faktor 4,4 bzw. 3,8, sowie die Anzahl der korrekten Erkennungen um den Faktor 2,2 zu erhöhen.

Tabelle 4.4: Experimentelle Ergebnisse (Griff-in-die-Kiste-System)

Methode	Durchschnittliche Erkennungszeit [ms] (K_1)	Fehler (Translation) [mm] (K_2)	Fehler (Rotation) [Grad] (K_3)	Anzahl der korrekten Erkennungen (K_4)
Klassisches Matching-Verfahren	1148,7	22,86	4,54	4,5
Hierarchisch-iterative Methode	170,1	5,25	1,19	9,8

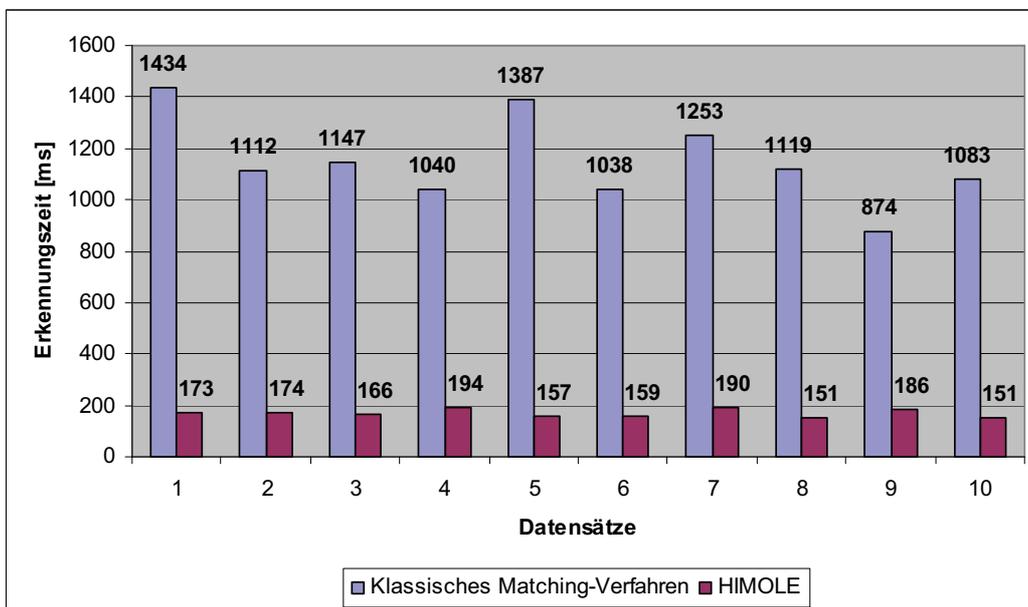


Abbildung 4.34: Erkennungszeit der verschiedenen Methoden

In der Abbildung 4.34 ist die benötigte Erkennungszeit der verschiedenen Methoden dargestellt.

In den Abbildungen 4.35 und 4.36 ist die Genauigkeit der Positions- und Orientierungserkennung der verschiedenen Methoden dargestellt.

In der Abbildung 4.37 ist die Anzahl der korrekten Erkennungen der verschiedenen Methoden dargestellt.

In der Abbildung 4.38 ist ein Beispiel der Erkennung von einem Objekt (I-Rohr) dargestellt. Der Prototyp wurde automatisch erzeugt: eine Szene, die nur ein Objekt enthält, wurde mit der PMD-Kamera aufgenommen und aus dieser Szene wurde der Behälter ausgefiltert.

In Abbildung 4.39 ist ein Beispiel der Erkennung verschiedener Objekte dargestellt. Die Prototypen wurden synthetisch als Punktwolken aus geometrischen Primitiven erzeugt.

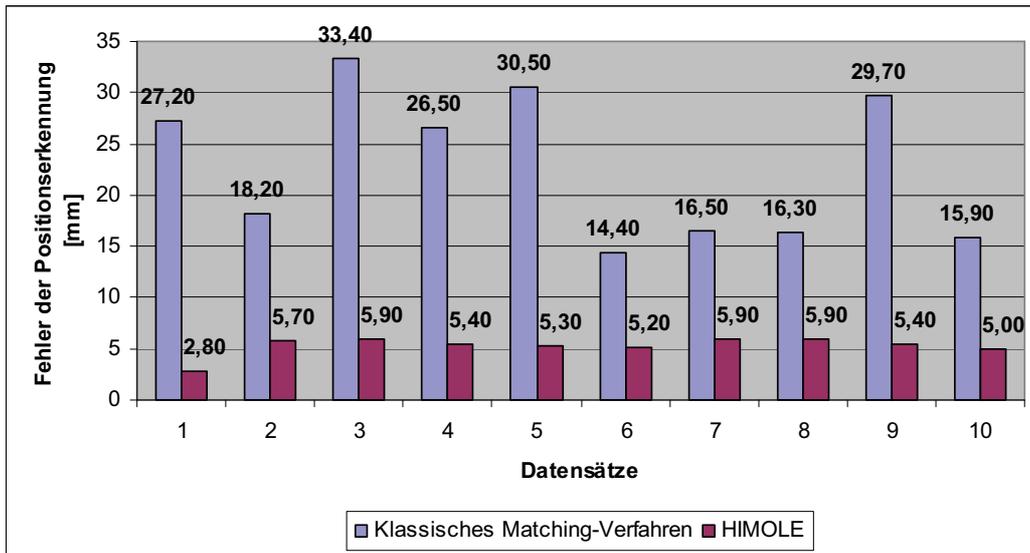


Abbildung 4.35: Genauigkeit der Positionserkennung

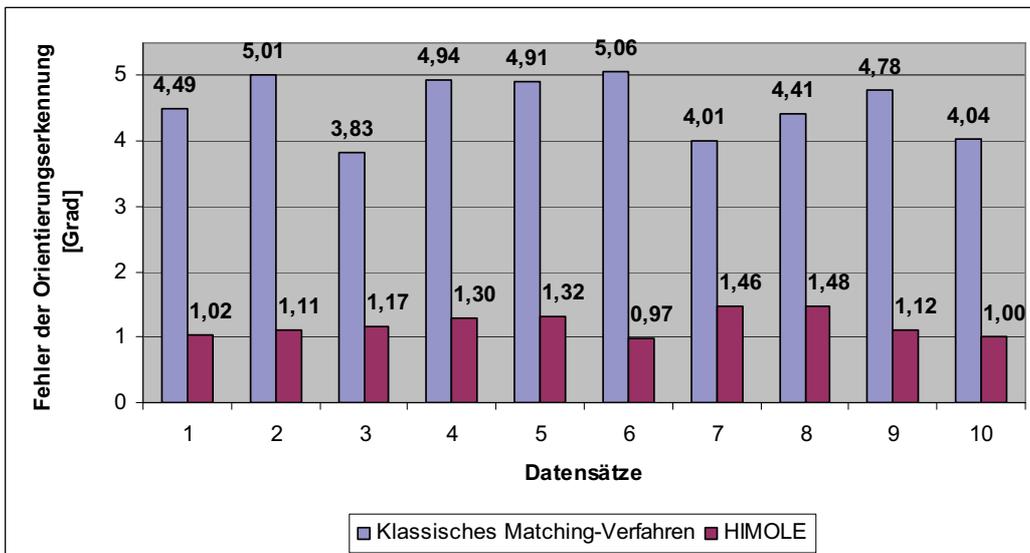


Abbildung 4.36: Genauigkeit der Orientierungserkennung

4.4.3 Zusammenfassung

In diesem Kapitel wurde die Architektur der Systeme beschrieben, die auf der hierarchisch-iterativen Methode für die Objekt- und Lagererkennung basieren. Diese Systeme lösen folgende Aufgaben:

- Registrierung medizinischer Bilddatensätze

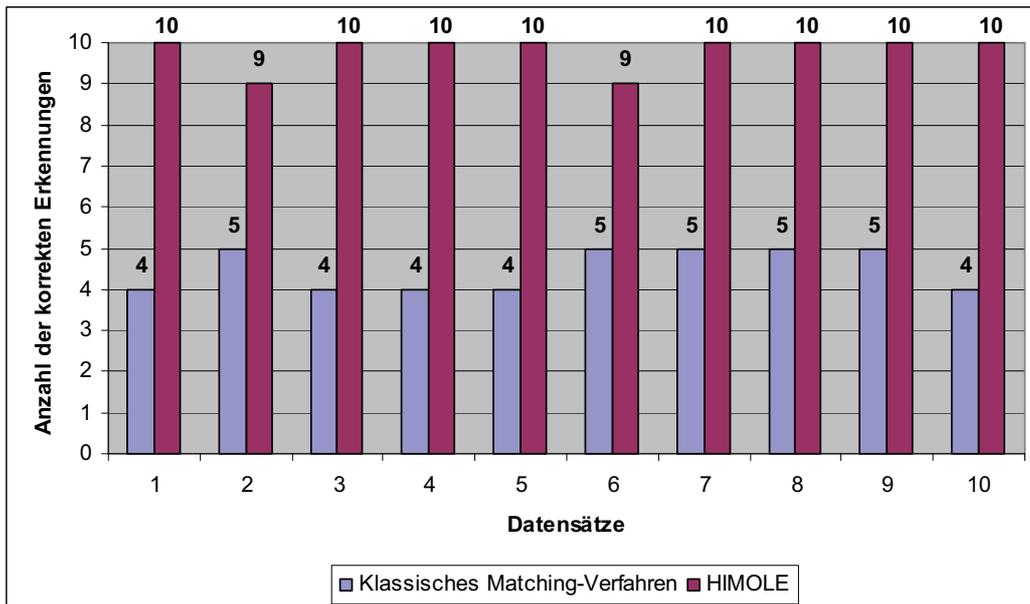


Abbildung 4.37: Anzahl der korrekten Erkennungen

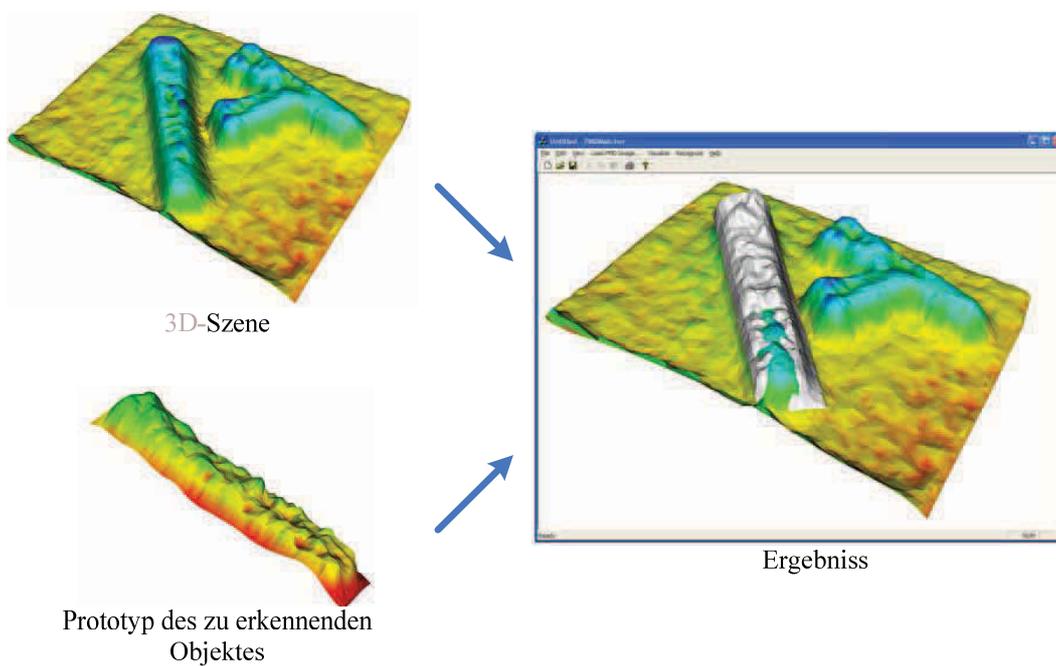
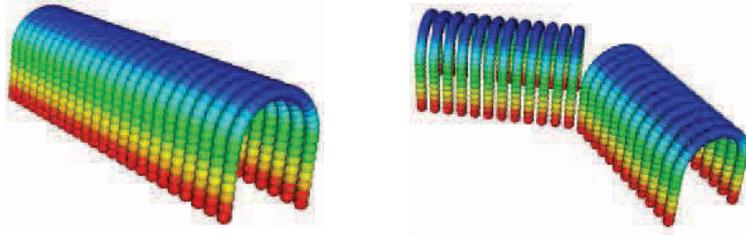
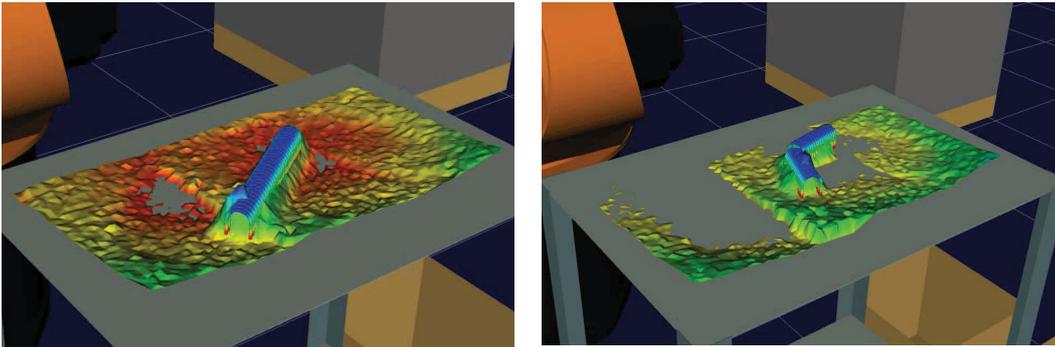


Abbildung 4.38: Objekterkennung in der 3D-Szene

- Schrifterkennung in der Industrie
- Erkennung mehrerer Objekte verschiedener Klassen, die sich überlappen können, und



Prototype von zu erkennenden Objekten



Szenen mit den erkannten Objekten

Abbildung 4.39: Objekterkennung in der 3D-Szene

ihrer Lagen für das Griff-in-die-Kiste-System

Basierend auf der entwickelten Architektur wurden Erkennungssysteme entwickelt, mit deren Hilfe zahlreiche Experimente durchgeführt wurden, die die Leistungsfähigkeit der Verwendung der hierarchisch-iterativen Methode gezeigt haben.

Kapitel 5

Entwicklung und Untersuchung von HIMOLE-basierten Software-Systemen

Als Softwarerealisierung der hierarchisch-iterativen Methode für die Objekt- und Lageerkennung wurde ein Softwaremodul als dll-Bibliothek implementiert, das eine Schnittstelle für die Verbindung mit externen Modulen und Systemen zur Verfügung stellt, deswegen kann dieses Modul in einem beliebigen Bildverarbeitungssystem integriert werden kann. Dieses Modul wurde in der Programmiersprache C++ unter Verwendung der Klassenbibliothek MFC und der Bildverarbeitungsbibliotheken VTK und OpenCV entwickelt.

Die Architektur des Moduls für die Objekt- und Lageerkennung, das auf der vorgeschlagenen hierarchisch-iterativen Methode basiert, ist auf der Abbildung 5.1 dargestellt.

Dieses Modul besteht aus folgenden Komponenten:

- Leitinstanz: Realisiert den Hauptalgorithmus der Objekterkennung, sowie die Interaktion zwischen den Modulen
- Module für die Vorverarbeitung der Eingangsdaten, das folgende Komponenten enthält:
 - Modul für die Erstellung von Matrixdarstellungen der Eingangsdaten (Distanz- oder Zeigermatrix)
 - Modul für die Erstellung der Pyramidendarstellung der Eingangsdaten
 - Modul für die Normalisierung der Eingangsdaten (Rauschfilterung und Rauschunterdrückung, etc.)
- Modul für das Matching-Verfahren: Realisiert den Hauptalgorithmus des Matchings mit der Verwendung der nachfolgenden Module
- Modul für die Prototypshierarchisierung: Realisiert den Algorithmus von der Hierarchisierung der Objektprototypen

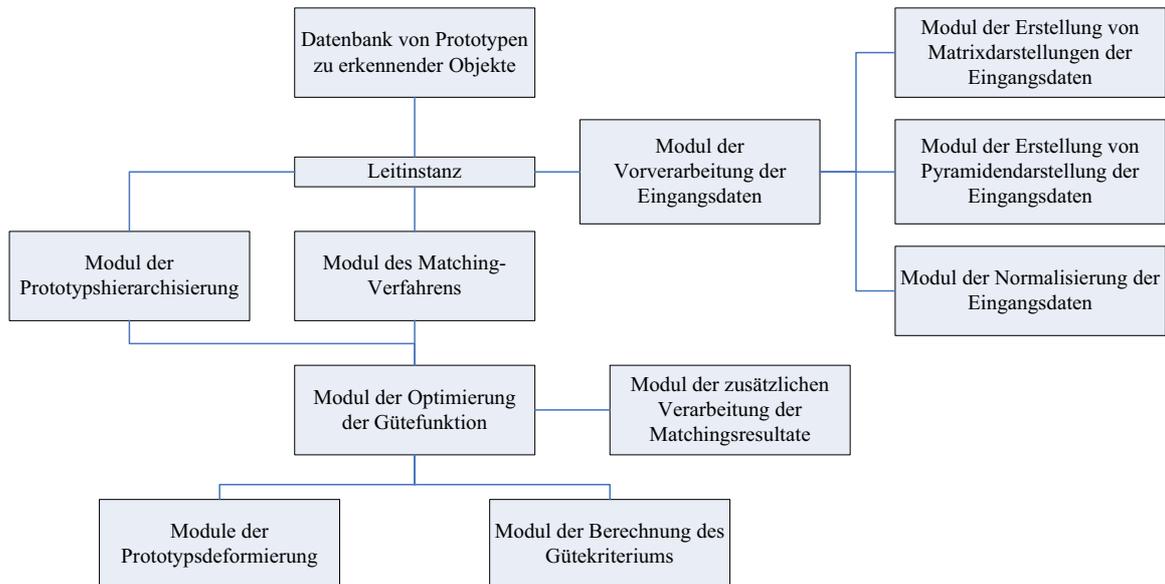


Abbildung 5.1: Architektur des Systems für die Objekt- und Lageerkennung

- Modul für die Optimierung der Gütefunktion: Realisiert den Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen
- Modul für die Prototypsdeformation: Realisiert den Algorithmus der Deformation des Objektprototyps
- Modul für die Berechnung des Gütekriteriums: Realisiert den Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten
- Modul für die zusätzliche Verarbeitung der Matchingsresultate
- Datenbank von Prototypen zu erkennender Objekte

Das entwickelte System wurde erfolgreich in die folgende Automatisierungssysteme integriert:

1. System für die automatisierte OP-Planung der Fa. Stryker-Leibinger: Registrierung medizinischer Bilddatensätze (CT und MRT). Das entwickelte Modul wird verwendet, um die automatische Fusion tomographischer Aufnahmen zu realisieren. Normalerweise werden hier CT- und MRT-Tomogramme verwendet.
2. System für die Qualitätssicherung bei der Herstellung von Piezo-Motoren der Fa. Elliptec - ReActVision: Erkennung der Seriennummer auf Piezo-Motoren.
3. Griff-in-die-Kiste-Anwendung: Erkennung von mehreren Objekten verschiedener Klassen und ihrer Lage (vom BMBF gefördertes Projekt - Lynkeus) mittels einer neuartigen PMD-basierten Kamera.

5.1 Softwaresystem für die automatisierte OP-Planung

Das Objekterkennungssystem, das auf der hierarchisch-iterativen Methode basiert, wurde adaptiert und im System der automatisierten OP-Planung integriert (sowohl für Linux- als auch für die Windows-Version dieses Systems). Da die plattformunabhängige Sprache C++ und die Bildverarbeitung-Bibliothek OpenCV bei der Entwicklung dieses Moduls verwendet wurden, konnte die Adaption des Moduls leicht durchgeführt werden.

Die Hauptaufgabe des Objekterkennungsmoduls besteht in der automatischen Fusion tomographischer Aufnahmen (s. 4.2) verschiedener Gerätetypen, z.B. CT- und MRT-Tomogramme.

In der Abbildung 5.2 ist die Benutzeroberfläche des Programms *Stryker Leibinger* mit dem integrierten Modul für die Objekt- und Lageerkennung für das Betriebssystem Windows dargestellt.



Abbildung 5.2: Benutzeroberfläche von Stryker Leibinger unter Windows

In der Abbildung 5.3 ist die Benutzeroberfläche des Programms für die Tomogrammregistrierung *Karma*, das am IPR entwickelt wurde, mit dem integrierten Modul für die Objekt- und Lageerkennung für das Betriebssystem Linux dargestellt.

5.2 Softwaresystem für die Qualitätssicherung bei der Herstellung von Piezo-Motoren

Das Softwaresystem für die Qualitätssicherung bei der Herstellung von Piezo-Motoren der Fa. Elliptec hat folgende Aufgaben.

- Lageerkennung des Piezo-Elements in Motoren



Abbildung 5.3: Benutzeroberfläche von Karma unter Linux

- Bestimmung der Verzerrungen des Motorrahmens nach dem Einpressen von Piezo-Element

Nachdem alle Prüfungen für die Qualitätssicherung durchgeführt wurden, muss das System die berechneten Parameter in die Datenbank eintragen. Dafür braucht man die Seriennummer, die in den Rahmen des Piezo-Motors gefräst ist. Diese Seriennummern werden mit Hilfe des entwickelten HIMOLE-Moduls erkannt (s. 4.3).

In der Abbildung 5.4 ist die Benutzeroberfläche des Programms *ReActVision* mit der integrierten HIMOLE-Modul dargestellt.

5.3 Softwaresystem für das Griff-in-die-Kiste-Problem

Das Lynkeus-System löst folgende Aufgaben:

- Automatische Entnahme einzelner Werkstücke aus einer Kiste
- Sichere Mensch-Roboter-Kooperation

Die Besonderheit des Lynkeus-Systems besteht in der Verwendung von neuen Sensoren, die auf der PMD-Technologie basieren. Diese Sensoren sind als kompakte Kameras realisiert und können gleichzeitig 2D-(Grauwert-) und 2,5D-(Tiefen)Bilder aufnehmen. Beim Einsatz dieser Technologie und der entwickelten hierarchisch-iterativen Methode für die Objekt- und Lageerkennung wurden Ergebnisse erreicht, aus denen sich einer Erhöhung der Effizienz bei der Lösung von Griff-in-die-Kiste-Aufgabe erkennen lässt.

Das Griff-in-die-Kiste-Problem teilt sich in die folgenden Aufgaben auf:



Abbildung 5.4: GUI von ReActVision

- Modellierung der 3D-Szene
- Szenenanalyse
- Greifpunktbestimmung
- Entwicklung einer Entnahmestrategie
- Kollisionsfreie Bahnplanung

Die Effizienz und die Genauigkeit der Lösung der drei zuletzt genannten Aufgaben hängen stark von der Szenenanalyse, d.h. von der Objekt- und Lageerkennung einzelner Objekte, ab. Für die Lösung dieser Aufgabe ist im Lynkeus-System das Modul eingesetzt, das auf der hierarchisch-iterativen Methode basiert (s. 4.4).

Die Benutzeroberfläche der im Rahmen des Lynkeus-Projekts entwickelten Software ist in Abbildung 5.5 dargestellt.

In der Abbildung 5.6 ist die Architektur des Lynkeus-Frameworks dargestellt. Dieses Framework enthält das HIMOLE-Modul.

5.4 Zusammenfassung

In diesem Kapitel werden Software-Systeme für die Lösung folgender Aufgaben unter Verwendung des entwickelten HIMOLE-Systems beschrieben. Dabei wurde das entwickelte System erfolgreich in die folgenden Anwendungen integriert:

- In ein System für die automatisierte OP-Planung der Fa. Stryker-Leibinger für die Registrierung medizinischer Bilddatensätze (CT und MRT)

5. Entwicklung und Untersuchung von HIMOLE-basierten Software-Systemen

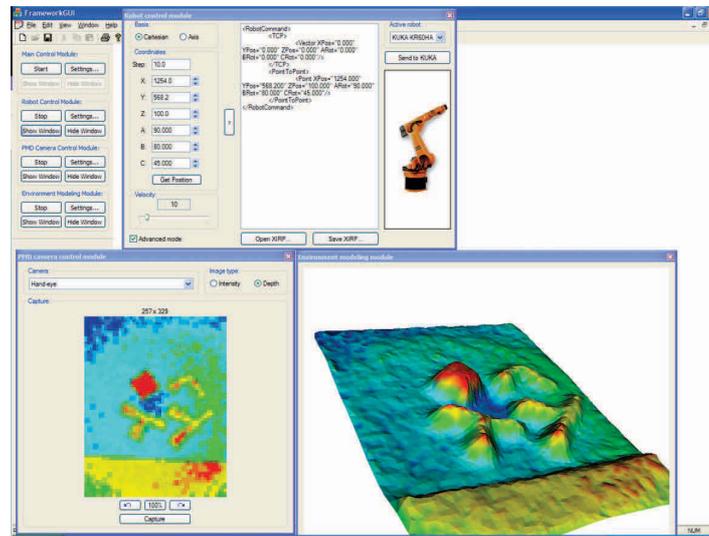


Abbildung 5.5: GUI von Lynkeus-Framework

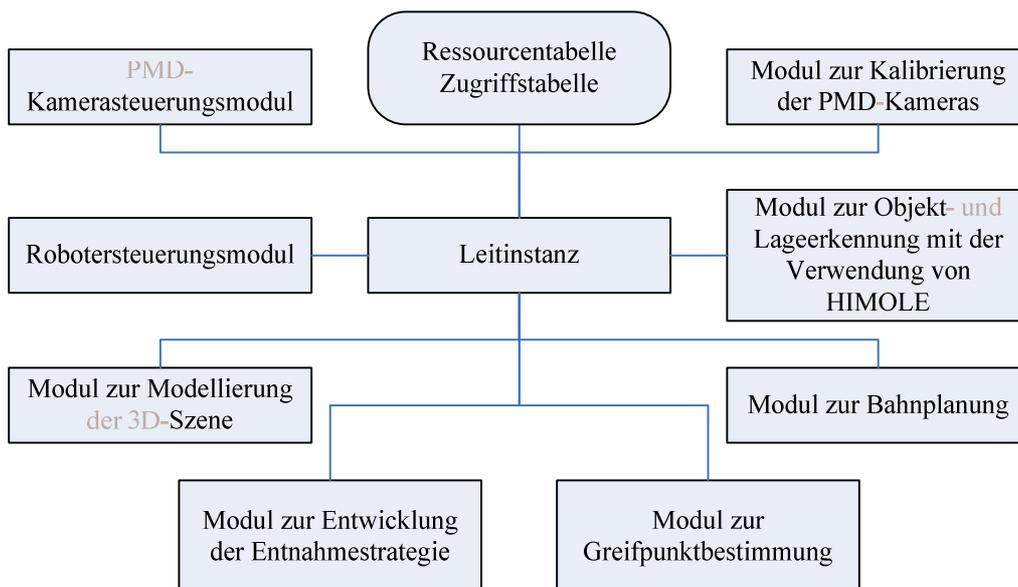


Abbildung 5.6: Architektur des Lynkeus-Frameworks

- In das System für die Qualitätssicherung bei der Herstellung von Piezo-Motoren der Fa. Elliptec - ReActVision zur Erkennung der Seriennummern von Piezo-Motoren
- In das Lynkeus-Griff-in-die-Kiste-System zur Erkennung von mehreren Objekten verschiedener Klassen und ihrer Lage

Kapitel 6

Zusammenfassung und Ausblick

Im Rahmen der hier vorgestellten Arbeit wurden folgende Ergebnisse erzielt:

1. Eine Methode zur Auswertung von 2D und 3D unstrukturierten Daten für die Objekt- und Lageerkennung wurde vorgeschlagen. Die Neuheit dieser Methode besteht in einer Kombination von Matching-Verfahren mit Optimierungsalgorithmen, sowie in der Interpretation des Matching-Verfahrens, die im Auffinden und Lokalisieren des, durch einen Prototypen vorgegebenen Objektes in 2D und 3D Daten besteht. Die entwickelte Methode besitzt eine hohe Rauschunempfindlichkeit, hängt nicht von Datenarten und zuerkennenden Objekten ab, erhöht nicht nur wesentlich die Wahrscheinlichkeit und Geschwindigkeit der korrekten Objekterkennung sondern auch die Genauigkeit der Lage- und Positionsbestimmung des Objektes. All das ermöglicht die Entwicklung von Objekterkennungssystemen, die in der Lage sind, eine Reihe von praktischen Aufgaben zu lösen.
2. Ein System von Algorithmen und Prozeduren, die Bestandteil der vorgeschlagenen Methode sind und die Erhöhung der Erkennungsgenauigkeit und -geschwindigkeit von Objekten und ihrer Lage ermöglichen, wurde entwickelt. Dieses System enthält folgende Algorithmen: Algorithmus der hierarchischen Optimierung von mehrdimensionalen Funktionen, Algorithmus des hierarchischen Matchings, Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten, Algorithmus der Deformation des Objektprototyps, Algorithmus des iterativen Matchings, Algorithmus der Hierarchisierung von Objektprototypen
3. Die Architektur des universellen Systems zur Objekterkennung, die auf der vorgeschlagenen hierarchisch-iterativen Methode basiert, sowie die Methodik der Verwendung dieses Systems bei der Lösung verschiedener Aufgaben wurden entwickelt.
4. Die Software, welche die vorgeschlagene hierarchisch-iterative Methode realisiert, wurde in C++ unter Verwendung der Bildverarbeitungsbibliotheken OpenCV und VTK implementiert. Mit Hilfe dieser Implementierung wurde eine Reihe von praktischen Aufgaben im Rahmen von Automatisierungssystemen gelöst.
5. Basierend auf den entwickelten Architekturen wurden Systeme zur Objekt- und Lageerkennung entwickelt, mit deren Hilfe zahlreiche Experimente durchgeführt wurden,

die die Effizienz der Verwendung von der hierarchisch-iterativen Methode gezeigt haben:

- Registrierung medizinischer Bilddatensätze: ein typisches herkömmliches Matching-Verfahren, probabilistisches Matching und hierarchisch-iterative Methode wurden verglichen. Die Experimente haben gezeigt, dass der Einsatz der hierarchisch-iterativen Methode im Vergleich zum klassischen Matching-Verfahren und probabilistischen Matching ermöglicht hat, die Erkennungszeit um den Faktor 7,2 bzw. 1,15, den Fehler der Positionsbestimmung um den Faktor 6,4 bzw. 2,6, den Fehler der Orientierungsbestimmung um den Faktor 3,6 bzw. 1,4 zu reduzieren, sowie die Anzahl der korrekten Erkennungen um den Faktor 2,3 bzw. 1,3 zu erhöhen.
- Schrifterkennung: aktuelle Software-Systeme für die Schrifterkennung (TOCR, SimpleOCR, etc.) und das hier vorgeschlagene System wurden verglichen. Die Experimente haben gezeigt, dass der Einsatz des Systems, das auf der hierarchisch-iterativen Methode basiert, ermöglicht, die Anzahl der korrekten Erkennungen um den Faktor 1,8, was 99,7% aller bearbeiteten Bilder entpricht, im Vergleich zu den TOCR-Ergebnissen zu erhöhen.
- Griff-in-die-Kiste-System: ein typisches herkömmliches Matching-Verfahren und das vorgeschlagene System wurden verglichen. Die Experimente haben gezeigt, dass der Einsatz von der hierarchischen iterativen Methode hat im Vergleich zu dem klassischen Matching-Verfahren ermöglicht, die Erkennungszeit um den Faktor 7,2 zu reduzieren, den Fehler der Positions- und Orientierungsbestimmung um den Faktor 4,4 bzw. 3,8, sowie die Anzahl der korrekten Erkennungen um den Faktor 2,2 zu erhöhen.

Der Ansatz des entwickelten Systems, das auf der vorgeschlagenen hierarchisch-iterativen Methode basiert, ist nicht auf die drei in Kapitel 4 beschriebene Aufgaben beschränkt. Als mögliche Anwendungsfelder des HIMOLE-basierten Systems sind die folgende weiteren Anwendungsgebiete möglich:

- Qualitätssicherung: Feststellung von Fehlern in komplexen Gussbauteilen
- Sortieraufgaben: Sortierung der Anhäufung von Objekten verschiedener Größen und Klassen
- verschiedene Suchaufgaben, z.B. Wortsuche in gescannten Texten: dafür wird Software zur Schrifterkennung nicht eingesetzt, sondern die geschilderte Methode, um kürzere Suchzeiten zu realisieren

Die vorgeschlagene Methode und das entwickelte System können auch in die folgenden Richtungen erweitert werden:

- Automatische oder halbautomatische Erweiterung der Template-Datenbank: die automatische Erzeugung von Templates neuer unbekannter Objekte, die Erstellung von Templates neuer Objekte anhand mehrerer Aufnahmen
- Selbstständige Parameteradaption während der Erkennung

-
- Verbindung mit einer Entnahmemodellierung: Nach der Szeneanalyse und der Entnahme des erkannten Objekts ist es sinnvoll, die Szene ohne des entnommenen Objekts zu analysieren und nicht erneut zu modellieren
 - Umweltaufnahme mit Semantik: Die Erkennung nicht nur von Objekte und ihrer Lage, sondern auch deren Zustände und die Zusammenhänge von Objekten (ob eine Tür geöffnet ist, ob die Hindernisse die Bewegung stören)

Als Fernziel dieser Arbeit kann die Weiterentwicklung des HIMOLE-basierten Systems im Rahmen eines universellen Systems, das selbständig neue Objekte erkennen und anschließend in beliebigen Szenedaten (2D, 3D, taktile Daten) wieder erkennen und interpretieren kann, betrachtet werden.

Anhang A

Experimentelle Ergebnisse

Tabelle A.1: Experimentelle Ergebnisse (Hierarchische Optimierung)

Datensatz	Durchschnittliche Erkennungszeit [ms] (K_1)	Fehler (Translation) [mm] (K_2)	Fehler (Rotation) [Grad] (K_3)	Anzahl der korrekten Erkennungen (K_4)
HIMOLE ohne den Algorithmus des hierarchischen Optimierung				
1	846	13,51	4,26	11
2	860	20,74	5,31	11
3	779	23,21	3,91	10
4	856	33,75	3,84	11
5	1133	28,25	5,11	12
HIMOLE mit dem Algorithmus der simulierten Abkühlung				
1	861	12,39	4,62	11
2	757	17,65	3,94	11
3	771	18,18	3,29	10
4	870	25,60	3,00	11
5	1045	24,77	4,55	13
HIMOLE mit dem Algorithmus der hierarchischen Optimierung				
1	717	11,16	3,64	13
2	682	15,48	3,61	13
3	756	16,23	2,84	12
4	725	22,65	2,78	13
5	968	20,47	3,99	15

A. Experimentelle Ergebnisse

Tabelle A.2: Experimentelle Ergebnisse (Hierarchisches Matching)

Datensatz	Durchschnittliche Erkennungszeit [ms] (K_1)	Fehler (Translation) [mm] (K_2)	Fehler (Rotation) [Grad] (K_3)	Anzahl der korrekten Erkennungen (K_4)
HIMOLE nur mit dem Algorithmus der hierarchischen Optimierung				
1	717	11,16	3,64	13
2	682	15,48	3,61	13
3	756	16,23	2,84	12
4	725	22,65	2,78	13
5	968	20,47	3,99	15
HIMOLE mit den Algorithmen der hierarchischen Optimierung und des hierarchischen Matching				
1	664	10,94	3,09	14
2	669	13,94	3,17	14
3	741	16,73	2,53	13
4	748	19,87	2,44	14
5	888	17,06	3,66	16

Tabelle A.3: Experimentelle Ergebnisse (Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten)

Datensatz	Durchschnittliche Erkennungszeit [ms] (K_1)	Fehler (Translation) [mm] (K_2)	Fehler (Rotation) [Grad] (K_3)	Anzahl der korrekten Erkennungen (K_4)
HIMOLE nur mit den Algorithmen von hierarchischer Optimierung und des hierarchischen Matchings				
1	664	10,94	3,09	14
2	669	13,94	3,17	14
3	741	16,73	2,53	13
4	748	19,87	2,44	14
5	888	17,06	3,66	16
HIMOLE mit den Algorithmen von hierarchischer Optimierung, des hierarchischen Matchings und der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten				
1	110	13,19	3,28	14
2	100	15,85	3,27	14
3	116	17,99	2,61	13
4	112	23,11	2,37	14
5	127	18,95	4,31	16

Tabelle A.4: Experimentelle Ergebnisse (Deformation des Objektprototyps)

Datensatz	Durchschnittliche Erkennungszeit [ms] (K_1)	Fehler (Translation) [mm] (K_2)	Fehler (Rotation) [Grad] (K_3)	Anzahl der korrekten Erkennungen (K_4)
HIMOLE nur mit den Algorithmen der hierarchischen Optimierung, des hierarchischen Matchings und der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten				
1	110	13,19	3,28	14
2	100	15,85	3,27	14
3	116	17,99	2,61	13
4	112	23,11	2,37	14
5	127	18,95	4,31	16
HIMOLE mit den Algorithmen der hierarchischen Optimierung, des hierarchischen Matchings, der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten und Deformation des Objektprototyps				
1	134	8,62	2,47	16
2	134	11,48	2,75	16
3	155	13,84	2,19	15
4	147	15,30	1,74	16
5	167	13,16	2,95	18

Tabelle A.5: Experimentelle Ergebnisse (Iteratives Matchings)

Datensatz	Durchschnittliche Erkennungszeit [ms] (K_1)	Fehler (Translation) [mm] (K_2)	Fehler (Rotation) [Grad] (K_3)	Anzahl der korrekten Erkennungen (K_4)
HIMOLE nur mit den Algorithmen der hierarchischen Optimierung, des hierarchischen Matchings, der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten und der Deformation des Objektprototyps				
1	134	8,62	2,47	16
2	134	11,48	2,75	16
3	155	13,84	2,19	15
4	147	15,30	1,74	16
5	167	13,16	2,95	18
HIMOLE mit den Algorithmen der hierarchischen Optimierung, des hierarchischen Matching, der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten, der Deformation des Objektprototyps und des iterativen Matchings				
1	179	6,68	1,88	18
2	176	7,87	1,98	18
3	209	9,48	1,59	17
4	186	11,51	1,16	18
5	214	9,20	2,30	19

Tabelle A.6: Experimentelle Ergebnisse (Hierarchisierung von Objektprototypen)

Datensatz	Durchschnittliche Erkennungszeit [ms] (K_1)	Fehler (Translation) [mm] (K_2)	Fehler (Rotation) [Grad] (K_3)	Anzahl der korrekten Erkennungen (K_4)
HIMOLE nur mit den Algorithmen der hierarchischen Optimierung, des hierarchischen Matchings, der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten, der Deformation des Objektprototyps und des iterativen Matchings				
1	179	6,68	1,88	18
2	176	7,87	1,98	18
3	209	9,48	1,59	17
4	186	11,51	1,16	18
5	214	9,20	2,30	19
HIMOLE mit den Algorithmen der hierarchischen Optimierung, des hierarchischen Matchings, der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten, der Deformation des Objektprototyps, des iterativen Matchings und der Hierarchisierung von Objektprototypen				
1	170	5,10	1,20	20
2	166	5,50	1,30	20
3	190	6,00	1,00	19
4	158	6,50	0,90	20
5	188	5,90	1,60	20

Anhang B

Abkürzungsverzeichnis

CAD Computer Aided Design

CT Computertomogram

HIMOLE Hierarchisch-iterative Methode zur Objekt- und Lageerkennung

MRT Magnetresonanztomogramm

OP Operation

PMD Photonic Mixer Device

Anhang C

Symbolverzeichnis

$c(x, y)$	Kreuzkorrelationsfunktion
d_i	Entscheidungsfunktion, die ein Muster x der Klasse i zuordnet (primitiver Klassifikator)
d_{NN}	Abstand von einem Element des Objektprototyps bis zum nächstgelegenen Element der Daten
$D(x, m_i)$	Distanz zwischen den Merkmalsvektoren zwei Objekten x und m_i
$f(x, y)$	2D-Bild
f	Gütefunktion
\check{f}	Kombinierte Gütefunktion
f_d^m	Distanzmatrix
f_p^m	Zeigermatrix
F	Gütemaß
F_{min}	Minimaler Wert des Gütekriteriums
F_{CC}	Korrelationskoeffizient
F_{MI}	Synentropie
F_{SDD}	Summe der quadratischen Abstände
F_{WSDD}	Gewichtete Summe der quadratischen Abstände
F_{WSSRD}	Gewichtete Summe der absoluten Abstände
$F(x)$	Synthetischer Klassifikator
G	Formale Grammatik
G_i	Parametersatz der Filter i zur Vorverarbeitung der Daten

C. Symbolverzeichnis

K_1	Durchschnittliche Zeit zur Erkennung eines Objektes
K_2	Durchschnittlicher Fehler der Positionserkennung
K_3	Durchschnittlicher Fehler der Orientierungserkennung
K_4	Anzahl der korrekten Erkennungen
L_{ij}	Verlust durch Zuteilung von Muster x , das zur Klasse ω_i gehört, zur Klasse ω_j
$L(G)$	Sprache, die von Grammatik G definiert ist
m_i	Erwartungswertvektor von Merkmalen der Klasse ω_i
M_i	Prototyp der Klasse ω_i
\bar{M}_i	Transformiertes Prototyp der Klasse ω_i
N_G	Anzahl an Parametersätze der Filter für die Vorverarbeitung
N_P	Anzahl der Pyramidenstufen
$p(x \omega_k)$	Wahrscheinlichkeitsverteilung des Musters x , wenn die Klasse ω_k vorgegeben ist
p_{ij}	Lageparameter j des Objekts i
P_i	Lage des Objektes X_i
$P(\omega_k)$	die Wahrscheinlichkeit, dass Klasse ω_k auftritt
r_i	Mittlerer Verlust durch Zuteilung des Musters x zur Klasse ω_i
R	Ähnlichkeitsgrad von zwei Objekten
R_i	System der logischen Entscheidungsregeln, die Klasse ω_i definiert
R^K	Daten (K - Anzahl der Dimensionen)
s_x	Skalierung entlang x -Achse
s_y	Skalierung entlang y -Achse
s_z	Skalierung entlang z -Achse
t_x	Translation entlang x -Achse
t_y	Translation entlang y -Achse
t_z	Translation entlang z -Achse
$T(M_i)$	Funktionen zur Transformation von Objektprototypen
T_i	Anzahl an logischen Regeln im System R_i
w	Gewicht

$w(x, y)$	Schablone als 2D-Bild
W	Anzahl an Objektklassen
x	Muster (Merkmalsvektor)
α_i^t	Gewicht der logischen Regel φ_i^t
α	Rotation um x -Achse
β	Rotation um y -Achse
γ	Rotation um z -Achse
$\gamma(x, y)$	Korrelationskoeffizient
$\Gamma_i(x)$	Anzahl an die Regeln, die das Muster x der Klasse ω_i zuordnen
φ_i^t	Logische Regel, die das Muster x der Klasse ω_i zuordnet
ω_i	Objektklasse

Abbildungsverzeichnis

2.1	Algorithmus zur Erkennung eines Objektes	13
2.2	Algorithmus zur Suche und Erkennung mehrerer Objekte in der Szene . . .	14
2.3	Methode für die Objekterkennung	17
2.4	Hypothetischer Ähnlichkeitsbaum	20
2.5	Objekt. Strukturelemente. Struktur, die von der Grammatik definiert wird. .	21
2.6	Endlicher Zustandsautomat für die syntaktische Erkennung von Symbolfolgen	21
2.7	Neuronales Netz, das zwei Klassen erkennt	22
2.8	Mehrschichtiges neuronales Netz, das mehrere Klassen erkennen kann . . .	23
2.9	Berechnung der Tangentendistanz	24
2.10	Sequentieller Multiklassifikator	28
2.11	Paralleler Multiklassifikator	29
3.1	Algorithmus des klassischen Matching-Verfahrens	37
3.2	Algorithmus der hierarchisch-iterativen Methode für die Objekt- und Lage- erkennung	39
3.3	Algorithmen der hierarchischen iterativen Methode für die Objekt- und La- geerkennung	44
3.4	Powell-Algorithmus zur Optimierung von mehrdimensionalen Funktionen .	46
3.5	Komponenten einer Stufe der hierarchischen Optimierung	47
3.6	Algorithmus der hierarchischen Optimierung der Gütefunktion	48
3.7	Experimentelle Ergebnisse (Hierarchische Optimierung)	51
3.8	Algorithmus des hierarchischen Matchings	53
3.9	Experimentelle Ergebnisse (Hierarchisches Matching)	54
3.10	Erstellung einer Matrixdarstellung des zu erkennenden Textes	56
3.11	Distanzmatrix (links) und Zeigermatrix (rechts)	56
3.12	Algorithmus für das Ausfüllen der Matrixzellen	57

3.13	Berechnung des Gütekriteriums mit Hilfe von Distanzmatrix beim Matching des Symbolprototyps mit dem zu erkennenden Text	57
3.14	Experimentelle Ergebnisse (Algorithmus der Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten)	58
3.15	Algorithmus der Veränderung der Prototypsparameter	59
3.16	Mögliche Deformationen der Schablone des zu erkennendem Symbols	60
3.17	Experimentelle Ergebnisse (Algorithmus der Deformation des Objektprototyps)	60
3.18	Algorithmus des iterativen Matchings	61
3.19	Beispiel der Verwendung von Algorithmus des iterativen Matchings	62
3.20	Experimentelle Ergebnisse (Algorithmus des iterativen Matchings)	62
3.21	Hierarchisches Modell von einem Prototyp eines zu erkennenden Objekts . .	64
3.22	Algorithmus der Hierarchisierung von Objektprototypen	64
3.23	Beispiel einer komplexen 3D-Szene	65
3.24	Beispiel des Y -Objekts und seines hierarchische Modell	65
3.25	Experimentelle Ergebnisse (Algorithmus der Hierarchisierung von Objektprototypen)	66
3.26	Experimentelle Ergebnisse (Hierarchische Optimierung)	67
4.1	Registrierung medizinischer Bilddatensätze	72
4.2	Wesentliche Merkmale von Tomogrammen	73
4.3	2D-Grauwert-Histogramm	75
4.4	Originales CT-Tomogramm (a) und Tomogramm, das mit dem anisotropischen Diffusionsfilter bearbeitet wurde (b)	76
4.5	MRT-Tomogramm, das mit Laplace- (a), Sobel- (b), Canny- (c) und morphologischem (d) Filter bearbeitet wurde	76
4.6	Gauß-Pyramide der CT- und MRT-Tomogramme	78
4.7	Gütefunktion unter Verwendung von Algorithmus des hierarchisches Matchings	79
4.8	Skalierungsfaktor s_x entlang x -Achse	79
4.9	Erkennungszeit bei der Registrierung von Tomogrammen	82
4.10	Genauigkeit der Positionserkennung bei der Registrierung von Tomogrammen	83
4.11	Genauigkeit der Orientierungserkennung bei der Registrierung von Tomogrammen	83
4.12	Anzahl der korrekten Erkennungen bei der Registrierung von Tomogrammen	84
4.13	Registrierung von Tomogrammen mit dem probabilistischen Verfahren (a) und mit der hierarchischen iterativen Methode (b)	84

4.14	Registrierung von Tomogrammen mit dem probabilistischen Verfahren (a) und mit der hierarchischen iterativen Methode (b)	85
4.15	Algorithmus der Schrifterkennung	87
4.16	Bilder mit Seriennummern	87
4.17	Schablone der zu erkennenden Symbole	87
4.18	Originales Bild mit der Seriennummer (a) und das mit dem anisotropischen Diffusionsfilter verarbeitete Bild (b)	89
4.19	Bild mit der Seriennummer, das mit dem Binärfilter (a) und einem morphologischen Filter (b) verarbeitet wurde	89
4.20	Gauß-Pyramide von Symbolen	90
4.21	Verwendung der Zeigermatrix	91
4.22	Mögliche Deformationen einer Symbolschablone: a) originale Schablone; b) Rotation um y -Achse; c) Verzerrung um y -Achse; d) Rotation um x -Achse; e) Verzerrung um x -Achse	91
4.23	Der Einfluss des Binarisierungsschwellwert: a) Originalbild; b) Binärbild mit dem Schwellwert von 75% (75% der weissen Pixel); c) Binärbild mit dem Schwellwert von 70%; d) Binärbild mit dem Schwellwert von 65%	92
4.24	Erkennungszeit bei der Schrifterkennung	94
4.25	Anzahl der korrekten Erkennungen bei der Schrifterkennung	94
4.26	Vergleich der Methoden/Systeme für die Schrifterkennung	95
4.27	Schrifterkennung mit den HIMOLE- und TOCR-Systemen	95
4.28	Architektur eines Griff-in-die-Kiste-Systems	97
4.29	Grauwertbild einer Szene (a); 2,5D-Bilder der selben Szene (b, c, d)	98
4.30	Umweltmodellierung	100
4.31	Glättung mit Laplace-Filter	100
4.32	Pyramidendarstellung einer 3D-Szene	102
4.33	Deformationen des Prototyps	103
4.34	Erkennungszeit der verschiedenen Methoden	105
4.35	Genauigkeit der Positionserkennung	106
4.36	Genauigkeit der Orientierungserkennung	106
4.37	Anzahl der korrekten Erkennungen	107
4.38	Objekterkennung in der 3D-Szene	107
4.39	Objekterkennung in der 3D-Szene	108
5.1	Architektur des Systems für die Objekt- und Lageerkennung	110
5.2	Benutzeroberfläche von Stryker Leibinger unter Windows	111
5.3	Benutzeroberfläche von Karma unter Linux	112

5.4	GUI von ReActVision	113
5.5	GUI von Lynkeus-Framework	114
5.6	Architektur des Lynkeus-Frameworks	114

Tabellenverzeichnis

2.1	Anforderungen an Methoden für die Objekt- und Lageerkennung	15
2.2	Anwendungsbedingungen und Nachteile von existierenden Methoden für die Objekt- und Lageerkennung	31
3.1	Eingangsparameter des Algorithmus der hierarchisch-iterativen Methode für die Objekt- und Lageerkennung	38
3.2	Schritte des Algorithmus der hierarchischen iterativen Methode für die Objekt- und Lageerkennung	40
3.3	Experimentelle Ergebnisse bei der Lösung Griff-in-die-Kiste-Aufgabe	67
4.1	Experimentelle Ergebnisse (Registrierung medizinischer Bilddatensätze)	82
4.2	Experimentelle Ergebnisse (Schrifterkennung)	93
4.3	Schrifterkennungen mit den HIMOLE- und TOCR-Systemen	96
4.4	Experimentelle Ergebnisse (Griff-in-die-Kiste-System)	105
A.1	Experimentelle Ergebnisse (Hierarchische Optimierung)	119
A.2	Experimentelle Ergebnisse (Hierarchisches Matching)	120
A.3	Experimentelle Ergebnisse (Berechnung von Gütekriterien bei der Verwendung einer Matrixdarstellung der Daten)	120
A.4	Experimentelle Ergebnisse (Deformation des Objektprototyps)	121
A.5	Experimentelle Ergebnisse (Iteratives Matchings)	122
A.6	Experimentelle Ergebnisse (Hierarchisierung von Objektprototypen)	123

Literaturverzeichnis

- [1] B. V. Anisimov, V. D. Kurganov, and V. K. Zlobin. *Raspoznavanie i cifrovaya obrabotka izobrazheniy*. Vysshaya shkola, 1983.
- [2] A. Bachem, T. Müller, and H. H. Nagel. Robust visual servoing: Examination of cameras under different illumination conditions. 2001.
- [3] L. M. Bahvalov. *Chislennye metody*. Nauka, 1973.
- [4] R. Basri. Recognition by prototypes. 1993.
- [5] G. Bebis, S. Louis, and S. Fadali. Using genetic algorithms for 3d object recognition. 1998.
- [6] S. Belongie, J. Malik, and J. Puzicha. Matching shapes. 2001.
- [7] M. Ben-Ezra, S. Peleg, and M. Werman. Model based pose estimator using linear-programming. 2000.
- [8] M. Berger, T. Auer, and G. Bachler. 3d model based pose determination in real-time: Strategies, convergence, accuracy. 2000.
- [9] P. J. Besl and R. C. Jain. Three-dimensional object recognition. 1985.
- [10] J. Beyerer. Automatische Sichtprüfung und Bildverarbeitung. Vorlesung an der Universität Karlsruhe (TH), 2008. http://www.ies.uni-karlsruhe.de/ies/de/lehre/html/lehre_asb.html#lehre_asb_material.
- [11] J. Beyerer. Mustererkennung. Vorlesung an der Universität Karlsruhe (TH), 2009. http://www.ies.uni-karlsruhe.de/lehre_mustererkennung.php.
- [12] M. M. Bongard. *Pattern Recognition*. Spartan Books, 1970.
- [13] R. T. Chin and C. R. Dyer. Model-based recognition in robot vision. 1986.
- [14] R. Cipolla and N. Hollinghurst. Visually guided grasping in unstructured environments. 1996.
- [15] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience, 2nd edition, 2000.
- [16] O. Faugeras and Q. T. Luong. *The Geometry of Multiple Images*. The MIT Press, 2001.
- [17] N. G. Fedotov. *Metody stokhasticheskoy geometrii v raspoznavanii obrazov*. Radio i svyaz, 1990.
- [18] R. Fletcher. *Practical Methods of Optimization*. Wiley, 2nd edition, 2000.
- [19] D. A. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice Hall, 2002.
- [20] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Journal of Computer and System Sciences*, pages 119–139, 1997.
- [21] Ya. A. Furman, editor. *Vvedeniye v konturnyy analiz i ego prilozhenie k obrabotke izobrazheniy i signalov*. FIZMATLIT, 2002.

- [22] Ya. A. Furman, A. N. Yuryev, and V. V. Yanshin. *Cifrovye metody obrabotki i raspoznavaniya binarnykh izobrazheniy*. Izdatelstvo Krasnoyarsk, 1992.
- [23] F. Gaßmann and K.-H. Franke. Verfahren zur triangulation ungeordneter 3d-punktmengen.
- [24] M. V. Gashnikov, N. I. Glumov, N. Yu. Ilyasova, V. A. Soyfer, et al. *Metody kompyuternoy obrabotki izobrazheniy*. FIZMATLIT, 2nd edition, 2003.
- [25] R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Prentice Hall International, 2nd edition, 2001.
- [26] A. N. Gorban. *Obuchenie neyronnykh setey*. SP Paragraf, 1990.
- [27] A. L. Gorelik, I. B. Gurevich, and V. A. Skripkin. *Sovremennoe sostoyanie problemy raspoznavaniya. Nekotorye aspekty*. Radio i svyaz, 1985.
- [28] A. L. Gorelik and V. A. Skripkin. *Metody raspoznavaniya*. Vysshaya shkola, 2nd edition, 1984.
- [29] J. Graf, K. Safronov, I. Tchouchenkov, and H. Wörn. Automated registration of medical head volume images using gradient based method. In *Proceedings of the 7th International Workshop on Computer Science and Information Technologies (CSIT'2005)*, volume 1, pages 174–179, 2005.
- [30] U. Grenader. *Lectures in Pattern Theory: Volume 1: Pattern Synthesis*. Springer, 1978.
- [31] U. Grenader. *Lectures in Pattern Theory: Volume 2: Pattern Analysis*. Springer, 1978.
- [32] A. Gueziec and N. Ayache. Smoothing and matching of 3-d space curves. In *Int'l Jour. Computer Vision*, volume 12, pages 79–104, 1994.
- [33] J. Haase. *Glättung von Polygonnetzen in medizinischen Visualisierungen*. PhD thesis, Universität Magdeburg, 2005.
- [34] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. University Press, second edition, 2000.
- [35] A. Hauck, J. Rüttinger, M. Sorg, and G. Färber. Visual determination of 3d grasping points on unknown objects with a binocular camera system. 1999.
- [36] K. Ikeuchi, B. K. P. Horn, and S. Nagata. Picking up an object from a pile of objects. 1983.
- [37] K. Ikeuchi, T. Shakunaga, M. Wheeler, and T. Yamazaki. Invariant histograms and deformable template matching for sar target recognition. In *Computer Vision and Pattern Recognition (CVPR '96)*, pages 100–105, 1996.
- [38] B. Jaehne. *Digital image processing*. Springer, 5th edition, 2002.
- [39] A. K. Jain. Statistical pattern recognition: A review. 2000.
- [40] A. E. Johnson and M. Hebert. Surface matching for object recognition in complex 3-d scenes. 1998.
- [41] A. Kak and J. Edwards. Experimental state of the art in 3d object recognition and localization using range data. 1996.
- [42] A. Kak and A. Kosaka. Multisensor fusion for sensory intelligence in robotics. 1996.
- [43] F. Kececi, M. Tonko, H. H. Nagel, and V. Gengenbach. Improving visually servoed disassembly operations by automatic camera placement. 1998.
- [44] M. G. Keller. *Matching Algorithms and Feature Match Quality Measures for Model-Based Object Recognition with Applications to Automatic Target Recognition*. PhD thesis, New York University, 1999.
- [45] D. Keysers, T. Deselaers, C. Gollan, and N. Hermann. Deformation models for image recognition. 2007.

- [46] S. Kim and I. S. Kweon. Automatic model-based 3d object recognition by combining feature matching with tracking. 2005.
- [47] L. G. Komarcova and A. V. Maksimov. *Neyrokompyutery*. Izdatelstvo MGTU im. N. E. Baumana, 2nd edition, 2004.
- [48] D. Kragic and H. I. Christensen. Robust visual servoing. *The International Journal of Robotics Research*, 22(10-11):923–939, 2002.
- [49] D. Kragic and H. I. Christensen. Vision techniques for robotic manipulation and grasping. In *Proc. International Symposium on Robotics (ISR 2002)*, October 7-11 2002.
- [50] D. Kragic and H. I. Christensen. Visual servoing meets the real world. In *Visual Servoing*, Lausanne, October 2002. IEEE.
- [51] G. F. Lakin. *Biometriya*. Vysshaya shkola, 1990.
- [52] A. Laubenheimer. *Automatische Registrierung adaptiver Modelle zur Typerkennung technischer Objekte*. PhD thesis, Universität Karlsruhe, 2004.
- [53] B. Leibe. *Segmentierungsfreie Objekterkennung unter Verwendung von Tiefenbildern*. PhD thesis, Universität Stuttgart, 2001.
- [54] V. Lepetit and P. Fua. Monocular model-based 3d tracking of rigid objects: A survey. 2005.
- [55] H. Li and R. Hartley. Feature matching and pose estimation using newton iteration. 2005.
- [56] Yu. Livshic. Statisticheskie metody raspoznavaniya obrazov. <http://yury.name/modern/07modernnote.pdf>.
- [57] M. A. Magnor. Geometry-based automatic object localization and 3-d pose detection. 2002.
- [58] D. Maier, J. Hesser, and R. Männer. Fast and accurate closest point search on triangulated surfaces and its application to head motion estimation. 2000.
- [59] I. R. Mamaev, K. V. Safronov, I. M. Tchouchenkov, and H. Wörn. Real-time framework for bin-picking problem using advantages of pmd-technology. In *Proceedings of the 9th International Workshop on Computer Science and Information Technologies (CSIT'2008)*, pages 21–26, 2008.
- [60] E. Marchand and F. Chaumette. Features tracking for visual servoing purpose. In *Advances in Robot Vision - From Domestic Environments to Medical Applications*, pages 10–20, Sendai, Japan, September 2004.
- [61] M. Müller. *Entnahmestrategien zur Objekteinzelung mittels Industrierobotern: Picking Strategies for Object Singularization using Industrial Robots*. PhD thesis, Universität Karlsruhe (TH), 2001.
- [62] M. Müller, T. Längle, and H. Wörn. Grasping and reliable picking of overlapping objects in unstructured scenes. 1999.
- [63] M. Müller, T. Längle, and H. Wörn. Secure picking of disarranged objects using probabilistic reasoning. 2000.
- [64] M. Müller, S. Stepanov, T. Längle, and H. Wörn. Robust detection of object displacements in unstructured scenes. 1999.
- [65] W. Narkbuakaew, S. Sotthivirat, D. Gansawat, P. Yampri, K. Koonsanit, Areeprayolkij W., and W. Sinthupinyo. 3d surface modeling and clipping of large volumetric data using visualization toolkit library. In *Proceedings of the 13th International Conference on Biomedical Engineering*, pages 1144–1148, 2009.
- [66] K. Ohba and K. Ikeuchi. Recognition of the multi specularity objects for bin-picking task. 1996.
- [67] K. Ohba and K. Ikeuchi. Recognition of the multi-specularity objects using the eigen-window. 1996.

- [68] K. Ohba, Y. Sato, and K. Ikeuchi. Visual learning and object verification with illumination invariance. 1995.
- [69] P. V. Pocepavev and I. B. Petrov. Effektivnyy algoritm predobrabotki izobrazheniy dlya strukturnykh metodov raspoznavaniya rukopisnykh simvolov. In *Issledovano v Rossii*, volume 19, pages 181–190, 2003.
- [70] P. Pokrandt. *Automatische Überlagerung medizinischer Bilddatensätze*. PhD thesis, Universität Karlsruhe (TH), 1997.
- [71] J. Ponce, S. Lazebnik, F. Rothganger, and C. Schmid. Toward true 3d object recognition. In *CVPR Workshop on Generic Object Recognition and Categorization*, 2004.
- [72] A.R. Pope. Model-based object recognition: A survey of recent research. Technical report, University of British Columbia, 1994.
- [73] D. A. Pospelov, editor. *Iskusstvennyy intellekt: Modeli i metody*. Radio i svyaz, 1990.
- [74] W. H. Press. *Numerical Recipes*. University Press, Cambridge. <http://http://www.nr.com/>.
- [75] E. P. Putyatin and S. I. Averin. *Obrabotka izobrazheniy v robototekhnike*. Mashinostroyeniye, 1990.
- [76] K. Rahardja and A. Kosaka. Vision-based bin-picking: Recognition and localization of multiple complex objects using simple visual cues. 1996.
- [77] A. Ruf, M. Tonko, R. Horaud, and H. H. Nagel. Visual tracking of an end-effector by adaptive kinematic prediction. 1996.
- [78] K. Safronov, I. Tchouchenkov, and H. Wörn. Combined medical image registration method using both mutual and gradient information. In *Proceeding of the 8th International Workshop on Computer Science and Information Technologies (CSIT'2006)*, volume 1, pages 7–10, 2006.
- [79] K. Safronov, I. Tchouchenkov, and H. Wörn. Optical character recognition using optimisation algorithms. In *Proceeding of the 8th International Workshop on Computer Science and Information Technologies (CSIT'2007)*, volume 1, pages 85–89, 2007.
- [80] K. V. Safronov, I. M. Tchouchenkov, and H. Wörn. Hierarchical iterative object recognition method for a pmd-sensor supplied bin-picking system. In *8th Asia-Pacific Conference on Control and Measurement (APCCM 2008), Harbin, China*, pages 80–84, 2008.
- [81] K. V. Safronov, I. M. Tchouchenkov, and H. Wörn. Hierarchical iterative pattern recognition method for solving bin picking problem. In *Robotik 2008*, pages 3–6, 2008.
- [82] P. J. Sanz, A. P. del Pobil, and J. M. Inesta. Real-time grasping of unknown objects based on computer vision. 1997.
- [83] M. Schael. *Methoden zur Konstruktion invarianter Merkmale für die Texturanalyse*. PhD thesis, Universität Freiburg, 2005.
- [84] R. E. Schapire and Y. Singer. Improved boosting algorithms using confidence-rated predictors. In *Machine Learning*, pages 297–336, 1999.
- [85] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen. Decimation of triangle meshes. In *Conference Proceedings of SIGGRAPH*, pages 65–70, 1992.
- [86] N. T. Siebel, O. Lang, F. Wirth, and A. Gräser. Robuste positionierung eines roboters mittels visual servoing unter verwendung einer trust-region-methode. 1999.
- [87] D. V. Smolin. *Vvedenie v iskusstvennyy intellekt*. FIZMATLIB, 2004.
- [88] M. Sorg. *Visual Tracking and Grasping of a Dynamic Object: From the Human Example to an Autonomous Robotic System*. PhD thesis, Universität München, 2003.

- [89] P. I. Sosnin. Metodicheskiy kompleks po discipline: Raspoznavanie obrazov i analiz scen. http://old.ulstu.ru/people/sosnin/umk/Image_Recognition_and_Scene_Analysis.
- [90] V. V. Starovoytov. *Lokalnye geometricheskie metody cifrovoy obrabotki i analiza izobrazheniy*. Institut tehnikoy kibernetiki, 1997.
- [91] H. A. Taha. *Operations Research: An Introduction*. Prentice Hall, 8th edition, 2006.
- [92] A. Takenouchi, N. Kanamaru, and M. Mizukawa. Hough-space-based object recognition tightly coupled with path planning for robust and fast bin-picking. 1998.
- [93] S. Theodoridis and K. Koutroumbas. *Pattern recognition*. Elsevier, 2nd edition, 2003.
- [94] N. S. Tomashevich, D. S. Tomashevich, and A. I. Galushkin. Metod realizacii invariantnosti k affinnym preobrazovaniyam pri raspoznavanii dvumernyh izobrazheniy. In *Informaionnye tehnologii*, 2001.
- [95] M. Tonko, V. Gengenbach, K. Schäfer, S. Picard, R. Horaud, R. Mohr, and H. H. Nagel. Towards the integration of object recognition and visual servoing for disassembly of used cars. 1997.
- [96] M. Tonko, K. Schäfer, F. Heimes, and H. H. Nagel. Towards visually servoed manipulation of car engine parts. 1997.
- [97] J. T. Tou and R. C. Gonzalez. *Pattern Recognition Principles*. Addison-Wesley Publishing Company, 1974.
- [98] K. Umeda and T. Arai. Industrial vision system by fusing range image and intensity image. 1994.
- [99] V. N. Vapnik and A. Ya. Chervonenkis. *Teoriya raspoznavaniya obrazov. Statisticheskie problemy obucheniya*. Nauka, 1974.
- [100] K. V. Vorontsov. Logische klassifizierungsalgorithmen. <http://http://www.ccas.ru/voron/download/LogicAlgs.pdf>.
- [101] S. Wang, R. Cromwell, A. Kak, I. Kimura, and M. Osada. Model-based vision for robotic manipulation of twisted tubular parts: Using affine transforms and heuristic search. 1994.
- [102] A. Wendt. 3D-Matching von ähnlichen Punktwolken in der Hüftendoprothetik. 2000.
- [103] S. Winkelbach. *Gradientenbasierte Rekonstruktion von 3d-Oberflächen*. PhD thesis, Universität Braunschweig, 2001.
- [104] P. Wunsch and G. Hirzinger. Registration of cad-models to images by iterative inverse perspective matching. 1996.
- [105] Z. Xu, R. Schwarte, H. Heinol, B. Buxbaum, and R. Ringbeck. Smart pixel - photonic mixer device (pmd). In *Proceedings of the International Conference on Mechatron and Machine Vision*, pages 259–264, 1998.
- [106] S. M. Yamany, A. A. Farag, and A. El-Bialy. Free-form object recognition and registration using surface signatures. 1999.
- [107] V. I. Zhuravlyov. *Izbrannye nauchnye trudy*. Magistr, 1998.

Eigene Veröffentlichungen

1. J. Graf, K. Safronov, I. Tchouchenkov, H. Wörn. Automated Registration of Medical Head Volume Images using Gradient Based Method. In *Proceedings of the 7th International Workshop on Computer Science and Information Technologies (CSIT'2005)*, Ufa, Russia, 2005.
2. K. Safronov, I. Tchouchenkov, H. Wörn. Combined Medical Image Registration Method using both Mutual and Gradient Information. In *Proceedings of the 8th International Workshop on Computer Science and Information Technologies (CSIT'2006)*, Karlsruhe, Germany, 2006.
3. K. Safronov, I. Tchouchenkov, H. Wörn. Optical Character Recognition Using Optimization Algorithms. In *Proceedings of the 9th International Workshop on Computer Science and Information Technologies (CSIT'2007)*, Ufa, Russia, 2007.
4. K. Safronov, I. Tchouchenkov, H. Wörn. Hierarchical Iterative Pattern Recognition Method for Solving Bin Picking Problem. In *VDI-Berichte 2012, Robotik 2008*, München, Germany, 2008.
5. K. Safronov, I. Tchouchenkov, H. Wörn. Hierarchical Iterative Object Recognition Method for a PMD-Sensor Supplied Bin-Picking System. in *Proceedings of the 8th Asia-Pacific Conference on Control and Measurement (APCCM 2008)*, Harbin, China, 2008.
6. I. Mamaev, K. Safronov, I. Tchouchenkov, H. Wörn. Real-time Framework for Bin-Picking Problem using Advantages of PMD-Technology. In *Proceedings of the 10th International Workshop on Computer Science and Information Technologies (CSIT'2008)*, Antalya, Turkey, 2008.
7. H. Wörn, J. Graf, K. Safronov, I. Mamaev. 3D-Umwelterfassung in der Produktion mit neuer 3D-Kamera. In *AVN: Allgemeine Vermessungs-Nachrichten*. 5/2009