

Wissensbasierte 3D-Analyse von Gebäudeszenen aus mehreren frei gewählten Stereofotos

Vom Fachbereich Elektrotechnik und Informationstechnik

der Universität Hannover

zur Erlangung des akademischen Grades

Doktor-Ingenieur

genehmigte

Dissertation

von

Dipl.-Ing. Oliver Grau

geboren am 28. Dezember 1963 in Hannover

2000

Referent: Prof. Dr.-Ing. C.-E. Liedtke
Korreferent: Prof. Dr.-Ing. H. Pralle
Tag der Promotion: 18.6.1999

Kurzfassung

Stichworte: Bildanalyse, wissensbasiertes System, 3D-Rekonstruktion

Diese Arbeit beschreibt ein Analysesystem zur automatischen Rekonstruktion dreidimensionaler Gebäudeoberflächen aus stereoskopischen Kameraaufnahmen. Nach einer Interpretation wird szenenspezifisches Wissen für die einzelnen Objektkomponenten ausgewählt und zusammen mit Meßdaten aus den Bildern zu einer effizienten und konsistenten Oberflächenbeschreibung vereint.

Für die Visualisierung in Sichtsystemen müssen die erzeugten Oberflächenbeschreibungen sehr effizient sein. Das heißt, die Anzahl der verwendeten Oberflächenprimitive soll möglichst gering sein und trotzdem soll das 3D-Modell realistisch wirken. Bei der herkömmlichen Gewinnung durch CAD-Systeme bringt ein Operator Wissen über Art und Aufbau der zu modellierenden Objekte ein, während automatisierte, bildgestützte Verfahren dies bisher nur ansatzweise tun. Die vorliegende Arbeit stellt daher einen Ansatz zur Darstellung und automatischen Nutzung von Vorwissen für die gestellte Rekonstruktionsaufgabe vor.

Zur Darstellung der erforderlichen Daten und des Vorwissens wurde ein generisches strukturiertes Szenenmodell entwickelt, das auf semantische Netze zur expliziten Wissensrepräsentation abgebildet wurde. Für die Anwendung des Wissens analysiert eine leistungsfähige Interpretationskomponente den Szeneninhalt. Diese Analyse stützt sich auf eine vorangegangene Segmentierung von stereoskopisch gewonnenen Tiefenkarten. Die entwickelte Oberflächenrekonstruktion integriert die Tiefen- und Konturmessungen der Bildverarbeitung mehrerer Ansichten unter Verwendung von in der Interpretation ausgewählten Randbedingungen zu einer konsistenten, geometrischen Oberflächenbeschreibung. Diese wird abschließend durch Gewinnung von Texturkarten aus den Kamerabilddern vervollständigt.

Die entwickelte Interpretationskomponente ist in der Lage reale, komplexe Gebäudeszenen zu analysieren. Die nachfolgende Oberflächenrekonstruktion berechnet aus den gewonnenen Daten und Randbedingungen eine effiziente, geschlossene Objektgeometrie. Durch die abschließende Texturierung mit den Eingangsbildern erreichen die Oberflächenbeschreibungen in der Visualisierung ein fotorealistisches Aussehen. Das entwickelte Analysesystem stellt damit ein wertvolles Werkzeug zur Erstellung von Datenbasen für Sichtsysteme dar.

Abstract

Keywords: image analysis, knowledge-based system, 3-D reconstruction

This thesis presents an analysis system for the automatically reconstruction of three-dimensional surfaces of buildings from stereoscopic camera images. To accomplish this, scene specific knowledge for the scene components is selected by an interpretation. This knowledge is used in conjunction with measurements from image processing to generate an efficient and consistent surface description.

For visualization in visual systems, the generated surface descriptions have to be efficient, i. e. the number of surface primitives should be low while the visual impression of the generated 3-D model should be realistic. In conventional creation using CAD methods this efficiency is reached by a human operator, who applies knowledge about the kind of the objects to model and their properties. Up to now image processing approaches rarely use this feature. This thesis presents an approach for the representation and automatic processing of a priori knowledge for reconstruction.

A generic structured scene model was developed for the representation of data and knowledge and semantic nets are used to represent this model. An effective interpretation component was developed that analyses the scene content and enables further use of the a priori knowledge. The analysis is based on a segmentation of depth maps, which are computed using a stereoscopic approach. The surface reconstruction integrates scene depths and scene contours extracted by image processing by using further constraints selected by the interpretation into a consistent geometrical surface description. Finally, the surface description is completed by computing texture maps from the camera images.

For the interpretation component will be shown that it is able to analyze complex building scenes. The following surface reconstruction computes efficient and complete object geometry. The surface descriptions are reaching photo realistic impression after the final texturing. The system offers a powerful tool for the creation of databases for visual systems.

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung der Universität Hannover. Herrn Prof. Dr.-Ing. C.-E. Liedtke danke ich für die Anregung zum Thema dieser Arbeit, die Übernahme des Hauptreferats und ganz besonders für die freie und angenehme Arbeitsatmosphäre in seiner Arbeitsgruppe.

Herrn Prof. Dr.-Ing. H. Pralle danke ich für die Übernahme des Korreferats.

Mein besonderer Dank gilt all denen, die direkt oder indirekt zum Gelingen dieser Arbeit beigetragen haben. Besonders erwähnen möchte ich dabei meine Kollegen Arnold Blömer, Jürgen Bückner, Hellwart Broszio, Oliver Dehning, Lutz Falkenhagen, Gabriel Gaus, Stefan Growe, Reinhard Koch, Ralf Tönjes und Wolfgang Wölker, die durch Diskussionen und aktive Zusammenarbeit am Entstehen der Arbeit beteiligt waren.

Die Implementierung der in der Arbeit vorgestellten Ansätze wäre nicht ohne die Mithilfe der beteiligten studentischen Hilfskräfte und Diplomanden möglich gewesen. Besonders hervorheben möchte ich den Einsatz der Herren Christos Colios, Stefan Growe, Jens Teichert und Sebastian Weik.

Für die kritische Durchsicht der Arbeit danke ich Herrn P. F. Exner, Carsten Lehr und Frau Ursula (Ulla) Rost.

Meinen Eltern danke ich für die frühe Anregung zum selbstständigen Arbeiten.

Das Anfertigen einer Dissertation erfolgt leider nicht in Einklang mit dem Familienleben. Meiner Frau Anke Exner-Grau und meinem Sohn Justus danke ich daher für das entgegengebrachte Verständnis.

Inhaltsverzeichnis

1. Einleitung	1
2. Konzept der wissensbasierten Analyse von Gebäudeszenen	7
2.1. Begriffsdefinitionen	7
2.2. Synthesemodell	9
2.3. Struktur des entwickelten Analysesystems	11
3. Ein Szenenmodell für die Analyse von Gebäudeszenen	13
3.1. Anforderungen und Aufgaben	13
3.2. Kameramodell	15
3.3. Oberflächenmodell	15
3.4. Strukturiertes generisches Szenenmodell	17
3.5. Vorgehen bei der Entwicklung des Szenenmodells	18
3.5.1. Bildung von Objektklassen und -hierarchien	19
3.5.2. Beschreibung von Objektrelationen	21
3.6. Modellgültigkeitsbereich	23
3.7. Diskussion des entwickelten Szenenmodells	24
4. Wissens- und Datenrepräsentation	25
4.1. Repräsentation von Objektoberflächen für Visualisierungsaufgaben	25
4.1.1. Geometrische Beschreibung der Objektoberfläche	25
4.1.2. Texturkarten	26
4.2. Kamerarepräsentation	27

4.3.	Repräsentation von Vorwissen	28
4.3.1.	Prädikatenlogik, Produktionssysteme und Regeln	28
4.3.2.	Semantische Netze	29
4.3.3.	Prozedurales Wissen	30
5.	Bildverarbeitung	31
5.1.	Kamerakalibrierung	31
5.2.	Stereoskopische Tiefenschätzung	32
5.3.	Segmentierung der Tiefenkarten	34
5.4.	Diskussion der Bildverarbeitungskomponenten	35
6.	Netzwerksprache zur szenenspezifischen Wissensrepräsentation	37
6.1.	Knoten	37
6.2.	Attribute	39
6.3.	Netzwerkanten (Relationen)	40
6.4.	Netzmengen	43
6.5.	Repräsentation von prozeduralem Wissen	44
6.5.1.	Berechnungsfunktionen für Attributwerte und -wertebereiche	45
6.5.2.	Suchfunktionen	46
6.5.3.	Benutzerdefinierte Funktionen	46
6.5.4.	Regeln für die Wissensverarbeitung	47
6.6.	Implementierungsaspekte	47
6.7.	Ein semantisches Netz für die Analyse von Gebäudeszenen	49
6.8.	Diskussion der entwickelten Repräsentation	51
7.	Interpretation des Szeneninhaltes	55
7.1.	Konzept der entwickelten Interpretationskomponente	55
7.2.	Ein problemunabhängiger Kontrollalgorithmus	56

7.2.1. Formulierung und Anwendung der Netzwerktransformationen als Inferenzregeln	57
7.2.2. Behandlung von Mehrdeutigkeiten	59
7.2.3. Suchalgorithmus	60
7.3. Bewertung von Teilinterpretationen	63
7.4. Bindung von Segmentierungsdaten (Matching)	68
7.5. Datenfluß während der Analyse	70
7.6. Analysestrategien zur Interpretation von Gebäudeszenen	73
7.7. Entwicklungswerkzeuge und Erklärungskomponente	78
7.8. Diskussion der entwickelten Interpretationskomponente	80
8. Erzeugung der Oberflächenbeschreibung	83
8.1. Konzept der entwickelten Rekonstruktionskomponente	83
8.2. Schätzung der geometrischen Oberflächenbeschreibung	84
8.2.1. Relationales generisches Kostennetz zur Darstellung von Randbedingungen und Meßwerten	85
8.2.2. Erzeugung des Kostennetzes aus der Szenenbeschreibung .	88
8.2.3. Gewinnung der Oberflächengeometrie durch Minimierung des Kostennetzes	89
8.3. Schätzung der Kamerabewegung (Registrierung)	92
8.4. Schätzung der Oberflächentextur	94
8.5. Diskussion des entwickelten Rekonstruktionsmoduls	97
9. Zusammenfassung	101
A. Ergebnisse der Interpretation	113
B. Kostenfunktionen des Kostennetzes	115

Abkürzungsverzeichnis

Allgemeine Schreibweisen:

$\vec{R} = (r_x, r_y, r_z)^T$	Dreidimensionaler Vektor
$\vec{r} = (r_x, r_y)^T$	Zweidimensionaler Vektor
$\mathbf{M} = \{I_1, I_2, \dots, I_N\}$	Menge mit den Elementen $I_1, I_2, \dots, I_N$
$\mathbf{L} = \{I'_1, I'_2, \dots, I'_N\}$	Geordnete Liste mit den Elementen $I'_1, I'_2, \dots, I'_N$
$\{nil\}$	Leere Menge oder Liste

Liste der verwendeten Abkürzungen und Begriffe:

$\delta(n)$	Wertigkeit eines Knotens n
$\delta_{Prio}(R)$	Priorität einer Regel R
$\epsilon(A)$	Wertebereich von Attribut A
σ	Bewertungssicherheit
$\sigma(A)$	Bewertungssicherheit eines Attributes A
$\sigma(N)$	Bewertungssicherheit eines Knotens N
$\sigma(S)$	Bewertungssicherheit eines Suchbaumknotens S
ρ_1	Radialverzerrungskoeffizient
φ	Winkel in [Rad]
Φ	Kostenfunktion
A	Attribut
$\mathbf{A}(N)$	Menge der Attribute des Knotens N
$B(L)$	Bindungen einer Kante L
<i>boolean</i>	Logischer Datentyp einer Variablen oder Funktion
C	Konzeptknoten
$\vec{C}, \vec{A}, \vec{H}, \vec{V}$	CAHV-Kamerarepräsentation
$c(S)$	Tatsächliche Kosten eines Suchbaumknotens S
$c_{ges}(S)$	Gesamtkosten eines Suchbaumknotens S
c_i	Kante des Kostennetzes vom Typ i
c_{ij}	Kante j des Kostennetzes vom Typ i
<i>CDPart-Of</i>	Kontextabhängige Teil-von-Kante des semantischen Netzes
<i>Concrete-Of</i>	Konkretisierung-von-Kante des semantischen Netzes
$\mathbf{D}$	Definitionsmenge eines Attributes
<i>Data-Of</i>	Datum-von-Kante des semantischen Netzes

E_i	Ebene im Raum
$E_i(p_1, p_2, p_3)$	Ebene im Raum in Achsenabschnittsform
$fc_i(\vec{x})$	Kostenfunktion zur Kante des Kostennetzes vom Typ i
<i>float</i>	Fließkommadatentyp einer Variablen oder Funktion
$\vec{h} = (h_x, h_y)^T$	Kamerahauptpunktverschiebung
<i>integer</i>	Ganzzahldatentyp einer Variablen oder Funktion
$I_j^H(C)$	Hypothetischer Instanzknoten
$I_j^P(C)$	Partieller Instanzknoten
$I_j^K(C)$	Vollständiger Instanzknoten
$I_j^F(C)$	Fehlinstanzknoten
<i>Instance-Of</i>	Instanz-von-Kante des semantischen Netzes
<i>Is-A</i>	Ist-ein-Kante des semantischen Netzes
$J = (\xi, v, \sigma)^T$	Bewertung, mit Zulässigkeit ξ , Güte v und Sicherheit σ
$J(A)$	Bewertung eines Attributes A
$J(N)$	Bewertung eines Knotens N
$J(S_i)$	Bewertung eines Suchbaumknotens S_i
K	Kostennetz
k_1, k_i	Konstanten
L_j	Schnittgerade zweier Ebenen im Raum
$L_i(N_1, N_2)$	Kante zwischen zwei Knoten N_1 und N_2 des semantischen Netzes
L	Menge von Kanten $\mathbf{L} = \{L_1, L_2, \dots\}$ des semantischen Netzes
<i>m</i>	Meter (Maßeinheit einer Strecke)
N, N_i	Knoten des semantischen Netzes
$\mathbf{N}(S)$	Menge aller Knoten eines Suchbaumknotens S
M	Menge von Knoten $\mathbf{M} = \{N_1, N_2, \dots\}$
$\vec{P}, \vec{p}$	3D-Punkt im Raum, 2D-Punkt in der Bildebene
PD_i	Polygondeskriptor
<i>pel</i>	Maßeinheit von Strecken in der Bildebene als Vielfache eines Bildelementes
<i>Part-Of</i>	Teil-von-Kante des semantischen Netzes
$Q(L)$	Quantität einer Kante L des semantischen Netzes
R, R_i	Transformationsregel zur Interpretation
R	Menge der reellen Zahlen
$\mathbf{R}_{\text{Trans}}$	Menge der Transformationsregeln zur Interpretation
$\vec{R}$	Rotationsvektor
$r(S)$	Restkosten eines Suchbaumknotens S
$r^*(S)$	Abgeschätzte Restkosten eines Suchbaumknotens S
$S(\mathbf{M}, \mathbf{L})$	Semantisches Netz
S_i	Suchbaumknoten
s_x, s_y	Größe eines Bildpunktes in der Kameraebene in $\left[\frac{m}{pel}\right]$

<i>string</i>	Zeichenkettendatentyp einer Variablen oder Funktion
$\vec{T}$	Translationsvektor
$V(A)$	Wert des Attributes A
$v(A)$	Bewertungsgüte eines Attributes A
$v(N)$	Bewertungsgüte eines Knotens N
$v(S)$	Bewertungsgüte eines Suchbaumknotens S
<i>void</i>	(Quasi-)Datentyp einer Funktion, die keinen Wert zurückliefert
<i>VRML</i>	Virtual Reality Modeling Language
w_j	Gewichtsfaktor
$w(A)$	Gewichtsfaktor der Bewertung von Attribut A
$w(N)$	Gewichtsfaktor der Bewertung von Knoten N
$\vec{x}$	Parametervektor

1. Einleitung

Aufgrund der raschen Entwicklung der Rechner-Hardware in den letzten Jahren werden eine Reihe von Anwendungen, die die Visualisierung von dreidimensionalen Objekten zum Gegenstand haben, im größeren Maßstab durchführbar. Bei Anwendungen wie Fahr- und Flugsimulation, der Architekturvisualisierung, 3D-Informations- und Leitsystemen oder Kommunikationsanwendungen, wie der Telepräsenz entsteht ein wachsender Bedarf, reale Objekte möglichst naturgetreu als Datensatz im Rechner zu beschreiben und anschließend zu visualisieren.

Die Darstellung der Objekte im Rechner erfolgt dabei als dreidimensionale Oberflächenbeschreibung. Die Gewinnung einer solchen Beschreibung von realen bekannten Objekten mit Hilfe von CAD-Methoden ist sehr langwierig und kostenintensiv und erreicht oftmals nicht die gewünschte Realitätstreue. Daher werden in der Literatur verschiedene Verfahren zur automatischen Gewinnung von Oberflächenbeschreibungen durch Auswertung von Kameraaufnahmen vorgeschlagen. Die meisten Ansätze nutzen dabei kein Vorwissen über die betrachteten Objekte. Die resultierenden Oberflächenbeschreibungen weisen daher nicht die Effizienz der manuell, durch CAD-Systeme erstellten Beschreibungen auf, bei denen ein Operator Vorgaben über Art und Aufbau der Objekte einbringt.

Diese Arbeit beschäftigt sich mit der automatischen Erzeugung von dreidimensionalen Oberflächenbeschreibungen starrer, unbewegter Objekte aus mehreren Stereobildpaaren. Dabei wird insbesondere Vorwissen über Gesetzmäßigkeiten der zu beschreibenden Objekte in einer Wissensbasis abgelegt und von dem System genutzt. Der Ansatz ermöglicht so die Gewinnung sehr effizienter Beschreibungen, die sonst nur durch CAD-Methoden gewonnen werden können und sich beispielsweise für den Einsatz in Simulationssystemen eignen. Die Wirkungsweise der entwickelten Verfahren wird anhand der Analyse von Gebäudeszenen demonstriert.

Stand der Technik

Bei der Projektion eines realen Objektes in die Bildebene einer Kamera wird nur der gerade sichtbare Teil der Oberfläche abgebildet. Ferner geht die Tiefeninformation verloren. Das primäre Ziel der automatischen Gewinnung von Oberflächenbeschreibungen aus Kamerabildern ist somit die Rekonstruktion der dreidimensionalen Form der Objektoberfläche. Um dieses mathematisch unterbestimmte Problem zu lösen, werden in der Literatur verschiedene modellbasierte Ansätze vorgeschlagen. Ein Modell der zu beschreibenden Szene umfaßt dabei ein Oberflächenmodell, ein Kameramodell und ein Beleuchtungsmodell. Letzteres wird oft stark vereinfacht. Die Aufgabe besteht nun darin, die Modellparameter aus den Kameraaufnahmen zu schätzen.

Die meisten Ansätze zur Rückgewinnung der Tiefeninformation setzen voraus, daß die Parameter der Kamera bekannt sind. Üblich ist eine Vorabkalibrierung der Kamera mit einem Kalibriermuster (zum Beispiel [Tsai87]). Mit diesem Verfahren werden die sogenannten inneren Parameter, wie die Brennweite der Kamera geschätzt und während der nachfolgenden Aufnahmen nicht mehr verändert. Werden mehrere Aufnahmen eines Objektes aus verschiedenen Blickwinkeln aufgenommen, so müssen die äußeren Kameraparameter, das sind die Position und Orientierung der Kamera im Raum, fortlaufend gemessen werden. Der ICP-Algorithmus [Besl92] fügt unabhängig erzeugte Teiloberflächen geometrisch zusammen und berechnet daraus die Parameter der neuen Ansichten. Ist ein großer Teil der bis dahin gewonnenen Oberfläche im aktuellen Kamerabild sichtbar, so können die Kameraparameter durch einen photometrischen Vergleich bestimmt werden [Koch97]. Die Ansätze zur Selbstkalibrierung schätzen sowohl die inneren als auch die äußeren Kameraparameter aus einer Folge von Kamerabildern [Luo97, Poll98].

Zur Rückgewinnung der Tiefeninformation aus zwei oder drei kalibrierten Kameraansichten eignen sich vorzugsweise stereoskopische Verfahren [Dhon89, Lem88, Cox92, Fal94]. Diese berechnen für jeden Bildpunkt (z.B. [Cox92, Fal94]) oder für einzelne Merkmale vorzugsweise Linien die entsprechende Tiefeninformation (Übersicht in [Lem88]). Durch letzteren Ansatz gewonnene Oberflächenbeschreibungen sind kantentreu, das heißt, die Rückprojektion einer gewonnenen 3D-Kante fällt auf die 2D-Linie, die der Berechnung diene. Gegenüber dem blockbasierten Ansatz ist jedoch die Rekonstruktion der Objekttopologie problematischer.

Die gewonnene Tiefeninformation wird anschließend unter Verwendung eines Oberflächenmodells zu einer Oberflächenbeschreibung integriert. Dazu dienen meist Spline-Oberflächenmodelle [Fol92, Hos92] mit anschließender Approxima-

tion durch Dreiecksnetze [Chen93, Koch97, Rie97].

In die obengenannten Verfahren fließt kein spezielles Vorwissen über die zu rekonstruierenden Objekte ein. Sie liefern jedoch akzeptable Ergebnisse, wenn die Anzahl der Flächenprimitive zur Oberflächenbeschreibung hoch gewählt wird. Bei der Echtzeitvisualisierung von 3D-Objekten zum Beispiel in Simulatoren ist es jedoch erforderlich, die Anzahl von 3D-Polygonen so gering wie möglich zu halten, um die Visualisierungs-Hardware nicht zu überfordern. Daher ist es erforderlich, die Objekte mit möglichst speziellen Modellen zu beschreiben, die mit einer minimalen Anzahl von Polygonen oder Dreiecken einen guten visuellen Eindruck erreichen. Sind die in der Szene vorkommenden Objekte in ihrer Form bekannt und liegen als CAD-Beschreibung [Bhan87] vor, so besteht die Aufgabe darin, zuerst die richtige CAD-Beschreibung auszuwählen und darauf die Lage des Objektes in der Kameraaufnahme zu bestimmen [Hara93, Zhan91, Stre96]. Die richtige Auswahl der CAD-Beschreibung ist hierbei ein Problem der Objekterkennung. Ein CAD-Modell ist jedoch sehr speziell und daher nur auf wenige gleichartige Objekte anwendbar. Mehr Freiheitsgrade erlauben parametrische Modelle, die neben der Lage die Variation der Größenabmessungen der Objekte zulassen. Parametrische Modelle wurden beispielsweise für die Modellierung von Personen [Ryd87, Fish93] und Gebäuden (zum Beispiel [Deb96]) entwickelt.

Zur Beschreibung komplexer Szenen sind generische Modelle erforderlich. Eine Interpretation des Szeneninhaltes kann eine objektspezifische Auswahl für die Oberflächenmodelle treffen. Für die Gewinnung von Oberflächenbeschreibungen gibt es bislang wenige Ansätze, die auf der Ebene der Bildverarbeitung arbeiten und beispielsweise eine Klassifizierung von Luftbildern nutzen, um Waldgebiete und Straßen unterschiedlich detailliert zu behandeln [McKe96, McKe98, Toen96]. Andere Ansätze sind stark abgestimmt auf bestimmte Anwendungen und Bildmerkmale, insbesondere konturbasierte Verfahren für die Luftbilddauswertung [Fis97, Lang96] oder Robotik [Winz95, Sand97].

Um einen fotorealistischen Eindruck der visualisierten Objekte zu erhalten, ist es erforderlich, neben der Formbeschreibung der Objektoberfläche deren Feinstruktur und Farbe zu erfassen. Diese werden in Form einer Texturkarte [Fol92] gespeichert. Die Gewinnung der Texturkarte erfolgt unter Verwendung der berechneten Objektform und den Kameraparametern aus den Kameraaufnahmen. Für Beschreibungen mit Dreiecksnetzen werden dabei oft affine Abbildungsmodelle verwendet [Kapp89, Niem95]. Bei Verwendung von großflächigen Polygonen müssen jedoch perspektivische Abbildungsmodelle verwendet werden [Hil97, Col98]. Die Texturkarte bildet zusammen mit der Objektform eine vollständige Oberflächenbeschreibung für die hier betrachteten Visualisierungsaufgaben.

Problemstellung der Arbeit

Zur automatischen Gewinnung einer Oberflächenbeschreibung von komplexen Szenen müssen spezielle, angepaßte Modelle für die einzelnen Szenenteile ausgewählt werden. Dazu muß ein geeignetes generisches Szenenmodell entwickelt werden, das sich sowohl zur automatischen Interpretation des Szeneninhaltes eignet, als auch für die Oberflächenrekonstruktion. Bisherige Ansätze betrachten Interpretation und Rekonstruktion meist als getrennte Probleme. Um szenenspezifisches Vorwissen für die Gewinnung einer Oberflächenbeschreibung zu nutzen, müssen jedoch beide Ansätze miteinander kombiniert werden.

Die Aufgabe einer Szeneninterpretation ist es, unter Verwendung des Szenenmodells den Szeneninhalt zu ermitteln. Zu Beginn der Verarbeitung ist weder bekannt, welche Objekte sich in der betrachteten Szene befinden, noch wo und in welcher Orientierung sie relativ zur Kamera liegen. Ferner ist aufgrund von Verdeckungen jeweils nur ein Teil der Objektteilkomponenten in einer Bildansicht sichtbar. Diese besonderen Probleme der Nahbereichsbildauswertung stellen hohe Anforderungen an das Szenenmodell und die Szeneninterpretation.

Die Rekonstruktion der Oberflächengeometrie aus blockbasierten Tiefenmessungen [Cox92, Fal94] ermöglicht die nachfolgende Zerlegung des Bildinhaltes in zusammenhängende Regionen. Um eine bessere Wiedergabe von Kanten zu erreichen, die besonders bei den betrachteten Gebäudeszenen wichtig sind, sollte das Verfahren jedoch mit kantenbasierten stereoskopischen Verfahren kombiniert werden. Die daraus berechnete Oberflächengeometrie muß darüber hinaus möglichst gut zu den in der Interpretation ausgewählten Randbedingungen passen. Abschließend ist die Oberflächenbeschreibung durch die Erzeugung von Texturkarten zu vervollständigen.

Lösungsansätze und Aufbau der Arbeit

Das Ziel dieser Arbeit ist der Entwurf eines Systems zur automatischen Gewinnung von Oberflächenbeschreibungen komplexer Objekte unter Verwendung von szenenspezifischem Wissen. Zur Formulierung dieses Wissens wurde ein strukturiertes generisches Szenenmodell entwickelt.

Aufbauend auf der Auswertung von Stereobildpaaren [Koch97, Fal94] werden die Szenen interpretiert. Anschließend werden Randbedingungen und spezifische Objekteigenschaften ausgewählt, die für die Gewinnung der Oberflächenbeschreibung nutzbar sind.

Die gewonnenen Daten, die ausgewählten Randbedingungen und die Objekteigenschaften werden zu einer konsistenten geometrischen Rekonstruktion der Form der Objekte genutzt. Abschließend wird die Oberflächenbeschreibung durch die Gewinnung von Texturkarten aus mehreren Ansichten vervollständigt.

Die Arbeit gliedert sich in folgende Abschnitte:

Das Konzept des entwickelten Systems und die Definition einiger in der Arbeit verwendeter Begriffe sind in Abschnitt 2 zusammengestellt.

Der Abschnitt 3 beschreibt das entwickelte Szenenmodell.

Eine Übersicht der notwendigen Repräsentationsformen für die erforderlichen Daten und Wissensinhalte sind in Abschnitt 4 zusammengestellt.

Im Abschnitt 5 sind die verwendeten Bildverarbeitungsverfahren beschrieben.

Der Abschnitt 6 beschreibt die entwickelte Wissensrepräsentation, mit der das strukturierte generische Szenenmodell formuliert werden kann. Ferner wird hier die zur Lösung der Aufgabe entwickelte Wissensbasis vorgestellt.

Der Abschnitt 7 beschreibt die zur Nutzung des Wissens notwendige Interpretation des Szeneninhalts.

Die Erzeugung der Oberflächenbeschreibung wird in Abschnitt 8 vorgestellt.

Die Arbeit schließt mit einer Zusammenfassung und Bewertung der Ergebnisse in Abschnitt 9.

2. Konzept der wissensbasierten Analyse von Gebäudeszenen

Das Ziel dieser Arbeit ist der Entwurf eines Szenenanalysesystems, das die Rekonstruktion von Oberflächen real existierender Objekte, wie zum Beispiel von Gebäuden, erlaubt. Insbesondere soll dabei untersucht werden, wie Vorwissen über die zu rekonstruierenden Objekte repräsentiert und automatisch verarbeitet werden kann. Dieser Abschnitt erläutert das Konzept des entwickelten Analysesystems und definiert einige in der Arbeit verwendete Begriffe.

2.1. Begriffsdefinitionen

Die Szenenanalyse hat das Ziel, aus Beobachtungen unter Verwendung eines Modells eine Szenenbeschreibung zu gewinnen [Lie89]. Die Beobachtungen stammen hier vor allem aus Kameraaufnahmen der Szene. Ferner gehen alle Angaben von szenenspezifischen Daten, wie Abmaßen von Kalibriermustern oder gegebenenfalls Benutzereingaben als Beobachtungen ein.

Es gelten die folgenden Definitionen:

Def. 2.1 Szene: Menge der beobachtbaren Objekte mit (absoluten) Bezügen in Raum und Zeit, ferner die Menge der für die Erklärung der Bildinhalte relevanten, physikalischen Erscheinungen.

Def. 2.2 Objekt: Durch einen Begriff bezeichneter Gegenstand des Interesses, insbesondere der Beobachtung.

Def. 2.3 Szenenbeschreibung: Enthält als Ergebnis der Szenenanalyse eine Beschreibung der beobachteten Objekte. Die Szenenbeschreibung enthält dabei die zum Erreichen des Analyseziels erforderlichen Daten und die Nutzdaten.

Def. 2.4 Repräsentation: Legt in Form von Konventionen fest, wie eine Klasse von Informationen beschrieben wird.

Für die Szenenanalyse sind besonders die folgenden drei (physikalischen) Objekttypen relevant:

Def. 2.5 (physikalischer) Körper: *Materiemenge, die einen zusammenhängenden, dreidimensionalen Raumbereich ausfüllt.*

Def. 2.6 Lichtquelle: *sendet Licht aus, wodurch in der Szene enthaltene Körper visuell in Erscheinung treten.*

Def. 2.7 Kamera: *Sensor der visuellen Erscheinungen und hier die wichtigste Quelle von Beobachtungen für das Szenenanalysesystem.*

Ein Teilziel dieser Arbeit ist die Rekonstruktion der Körperberandung, also der Oberfläche der sichtbaren festen Körper. Diese ist eindeutig definiert und beobachtbar an der Grenze von festen Körpern zur umgebenden Luft. Die Trennung zusammengefügtter Körper, wie zum Beispiel eines Hauses vom Untergrund, ist jedoch nicht visuell möglich, sondern nur mit Hilfe von spezifischen Modellannahmen über die betreffenden Objekte. Diese Annahmen sollen als explizites Wissen formuliert werden. Als Definition für ein wissensbasiertes System gilt dann:

Def. 2.8 Wissensbasiertes System: *ein System, in dem zur Lösung eines Problems geeignetes Wissen in einer expliziten Repräsentation getrennt von der Systemkontrolle vorliegt. Dabei ist es möglich, das System durch Austausch der Wissensbasis an andere ähnliche Aufgaben anzupassen, ohne daß eine Modifikation der Kontrollstrukturen des Systems erforderlich wäre.*

Das in dieser Arbeit beschriebene Szenenanalysesystem erzeugt mit Hilfe eines Szenenmodells eine Szenenbeschreibung, die neben der Oberflächenbeschreibung eine Reihe von weiteren Informationen enthält. Der Systembenutzer kann über einen Filter diejenigen Informationen auswählen, die für seine Aufgabenstellung relevant sind. Neben einer geometrischen und topologischen Beschreibung sind das insbesondere symbolische Informationen und strukturelle Relationen. Die letzteren Daten sind im allgemeinen gewünscht, wenn das Analysesystem zu Aufgaben der Objekterkennung eingesetzt wird. Das ist in dieser Arbeit jedoch nicht das primäre Analyseziel, sondern ein zur Lösung der Aufgabe der wissensbasierten Oberflächenrekonstruktion erforderlicher Zwischenschritt.

Die von dem System gewonnene Oberflächenbeschreibung kann mit oder ohne die zusätzlichen Informationen in weiterverarbeitenden Systemen importiert werden und dort als 3D-Modell zur Visualisierung der Szene genutzt werden. Die Abbildung 2.1 verdeutlicht diesen Zusammenhang.

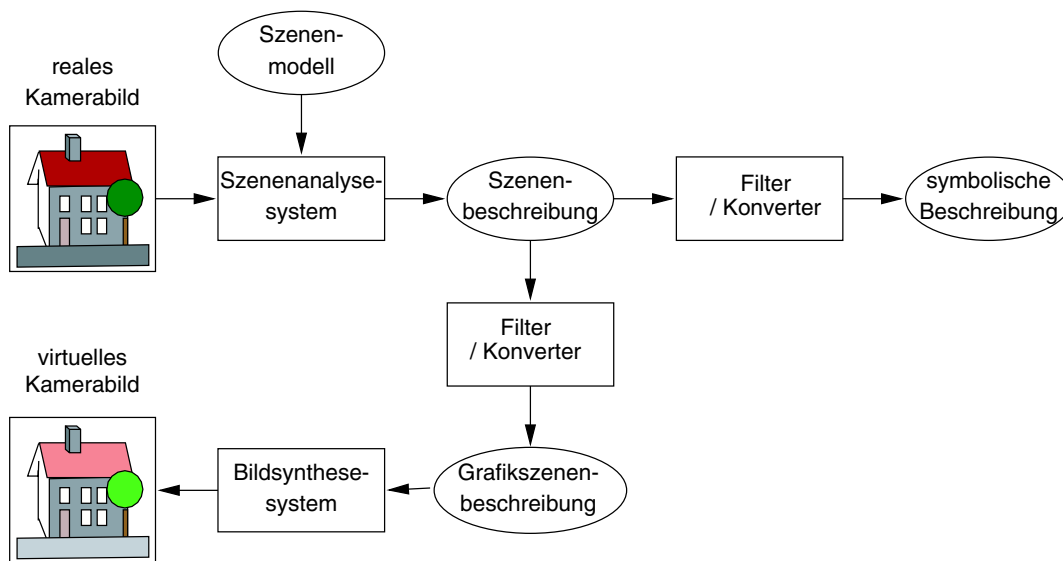


Abbildung 2.1.: Zusammenhang zwischen Szenenanalyse und der Visualisierung (Synthese)

2.2. Synthesemodell

Die Güte der automatischen Rekonstruktion von 3D-Oberflächen für Visualisierungsaufgaben wird daran bemessen, ob der visuelle Eindruck der synthetisierten Szene subjektiv befriedigend ist. Das ist gleichbedeutend mit der Frage, ob ein menschlicher Betrachter die visualisierte Szene als hinreichend realistisch empfindet.

Prinzipiell wird eine Rekonstruktion um so realistischer wirken, desto detailgetreuer die Oberflächenbeschreibung ist. Auf der anderen Seite weisen technische Visualisierungssysteme Grenzen in ihrer Verarbeitungskapazität auf, so daß immer ein Kompromiß gefunden werden muß, zwischen möglichst detailgetreuer Wiedergabe von Objekten und dem maximal darstellbaren Komplexitätsgrad des Visualisierungssystems. Insbesondere bei einer Echtzeit-3D-Visualisierung, zum Beispiel in Flug- und Fahrsimulatoren, führt eine komplexe und damit datenreiche virtuelle Szene dazu, daß die Bildwiederholrate sinkt. Das äußert sich bei der Wiedergabe von Bewegung in einer nicht mehr flüssigen Darstellung und wird wiederum als unnatürlich empfunden.

In der Computergrafik wurden verschiedene Techniken beziehungsweise Repräsentationen entwickelt, um die Datenkomplexität während der Visualisierung zu reduzieren, ohne daß das Ergebnis deutlich unrealistischer wirkt. Einige Re-

präsentationen, wie VRML [VRML97] für die Darstellung von 3D-Objekten im Internet, sind bereits standardisiert worden.

Die zugrundeliegenden Prinzipien, die zur Lösung unterschiedlicher Aufgaben geeignet sind, werden in dieser Arbeit als Synthesemodell bezeichnet. Sie basieren auf Beobachtungen der Eigenschaften der menschlichen visuellen Wahrnehmung von 3D-Visualisierungen:

- Die realen kontinuierlichen Objektoberflächen können durch einfache, generische Oberflächenmodelle approximiert werden. Die notwendige Komplexität hängt ab vom Abstand zwischen virtuellem Objekt und virtueller Kamera während der Visualisierung. Das heißt, bei großem Abstand kann die Oberfläche mit geringer Komplexität wiedergegeben werden, ohne daß die Visualisierung unrealistisch erscheint, da in den kleiner abgebildeten Objekten die Details aufgrund der begrenzten Auflösung nicht mehr sichtbar sind.
- Die Objektform kann wesentlich gröber angenähert werden, wenn gleichzeitig die Objektfarbe und die Objektfeynstruktur als Textur wiedergegeben wird. So ist es beispielsweise nicht erforderlich, alle Mauersteine eines Gebäudes als einzelne 3D-Objekte zu formulieren, sondern es reicht meist aus, die Wände geometrisch nur durch ein Polygon und die Objektfarbe und -feynstruktur als Textur zu beschreiben.

Die Visualisierung einer derart beschriebenen virtuellen Szene wirkt realistisch, solange keine signifikanten Fehler durch die vereinfachte Geometrie sichtbar werden. Der Zusammenhang der notwendigen Modellkomplexität vom Abstand des Modells zur virtuellen Kamera kann genutzt werden, um während der Visualisierung die augenblickliche Anzahl der 3D-Primitive (Komplexität des Modells) nahezu konstant zu halten. Dazu werden die darzustellenden Objekte in unterschiedlichen Komplexitätsgraden erzeugt.

Neben dem Abstand der virtuellen Kamera zum rekonstruierten Objekt hängt die subjektive Qualität von Art und Aufbau des Objektes selbst ab. Der Betrachter nimmt mangelnde Realitätstreue an Objekten wahr, die er erkennt und mit seinem Vorwissen vergleicht.

Analog zum Synthesemodell wird in dieser Arbeit ein Szenenmodell für die Analyse vorgeschlagen, das als Wissensbasis repräsentiert wird und es erlaubt, objektspezifisches Vorwissen für die Rekonstruktion der Objektoberfläche zu formulieren. Das Szenenmodell wird ausführlich im Abschnitt 3 behandelt.

2.3. Struktur des entwickelten Analysesystems

Die Abbildung 2.2 gibt eine Übersicht über Komponenten und Struktur des entwickelten Analysesystems. Als Eingangsdatum wird jeweils ein Stereobildpaar in dem Block *Bildverarbeitung* verarbeitet. Der Stereosensor, der die Aufnahmen liefert, wird vor der Aufnahme kalibriert. Dabei werden die Kameraparameter Brennweite, Radialverzerrung, Abstand der beiden Kameras voneinander (Basisabstand) und relative Orientierung der Kameras zueinander erfaßt. Mit diesen Daten errechnet eine Korrespondenzanalyse für jeden Bildpunkt die Szenentiefe. Darauf basierend wird die Szene segmentiert, das heißt in Bildregionen zerlegt. Zusätzlich liefert die Bildverarbeitung Konturen der Szene als Linienelemente.

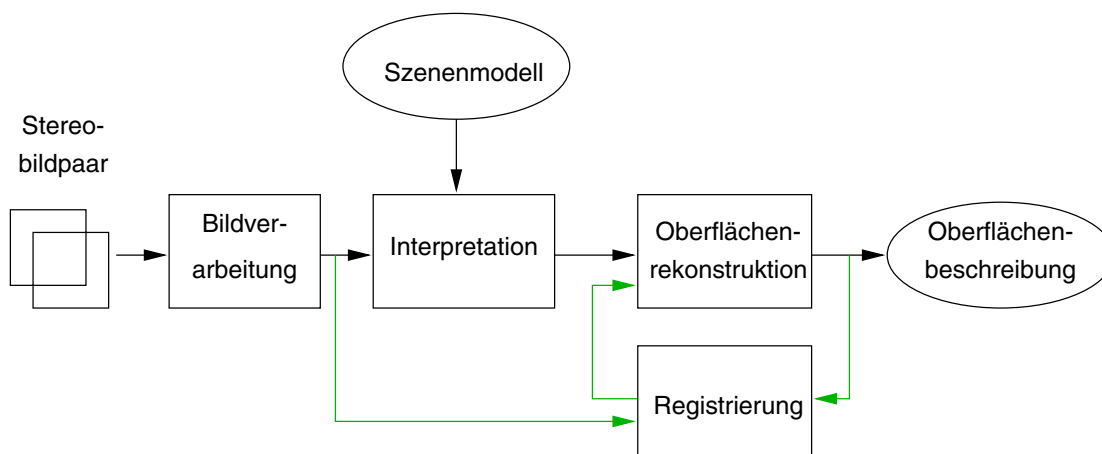


Abbildung 2.2.: Überblick über die Komponenten des entwickelten Analysesystems

Die Interpretation erzeugt unter Verwendung eines als Wissensbasis repräsentierten Szenenmodells eine symbolische Beschreibung der zuvor extrahierten Bildmerkmale. Die Oberflächenrekonstruktion berechnet aus den vorliegenden Daten eine konsistente Oberflächenbeschreibung der interpretierten Objekte.

Nach Verarbeitung des ersten Bildpaares kann der Stereosensor an eine andere Position gebracht werden. Die Aufnahme dieser neuen Ansicht wird als neues Eingangsdatum analysiert und durchläuft dabei wie zuvor die einzelnen Komponenten des Analysesystems. Um die dreidimensionale Information in die vorangegangene Beschreibung integrieren zu können, muß zuvor die Position und Orientierung des Stereosensors bestimmt werden. Die Berechnung dieser Parameter erfolgt unter Verwendung der bereits erstellten Oberflächenbeschreibung in der Registrierung.

3. Ein Szenenmodell für die Analyse von Gebäudeszenen

Die Aufgabe des hier vorgestellten Analysesystems ist die Erzeugung einer Beschreibung des Szeneninhaltes aus Kamerabildern. Die Wiederherstellung der dreidimensionalen Form und darüber hinaus der Zusammenhänge (Semantik) der Szenenkomponenten durch eine Interpretation erfordern einen modellbasierten Ansatz. In diesem Abschnitt wird das Szenenmodell entwickelt, das dem System zugrunde liegt. Das Szenenmodell umfaßt ein Kamera-, ein vereinfachtes Beleuchtungsmodell und ein Oberflächenmodell. Diese (Teil-)Modelle werden mit der Szenenstruktur und -semantik durch ein generisches strukturiertes Szenenmodell zusammengefaßt.

3.1. Anforderungen und Aufgaben

In der Signalverarbeitung dient ein Modell dazu, ein Signal zu erklären, um damit eine bestimmte Aufgabe zu lösen. Die Anwendung eines Modells erfolgt in drei Schritten:

1. Auswahl eines der Aufgabe angemessenen Modells,
2. Schätzung (Anpassung) der Modellparameter,
3. Nutzung des angepaßten Modells zur Lösung der Aufgabe.

Die Modellauswahl umfaßt neben der Festlegung der Art des Modells auch Einschränkung oder Konfiguration der Modellparameter, insbesondere in der Anzahl der Parameter sowie der zugehörigen Wertebereiche.

Die wesentliche Motivation für den Einsatz eines modellbasierten Ansatzes für die Analyse ist die Fähigkeit, Information zu integrieren. Im vorliegenden Anwendungsfall liefern die einzelnen Kameraaufnahmen aufgrund von Verdeckungen und Projektion vom dreidimensionalen in den zweidimensionalen Raum nur

unvollständige Informationen über die betrachteten Objekte. Ein richtig gewähltes Szenenmodell ermöglicht die Integration von Information, wie strukturelle Relationen, geometrische Randbedingungen und Fusion mehrerer Bilder. Damit ist eine konsistente Rekonstruktion der Objektoberflächen möglich.

Aufgrund der Komplexität der realen Welt sind Szenenmodelle stark auf die jeweilige Anwendung und das Ziel zugeschnitten und weisen einen abgegrenzten *Modellgültigkeitsbereich* auf (siehe Abschnitt 3.6). Das Modell wird für jede Anwendung und zumeist auch jede Szene individuell durch einen Operator ausgewählt. Desweiteren muß dieser eine (Vor-)Einstellung der Systemparameter und der Parameter des ausgewählten Modells vornehmen. Bei letzteren sind vor allem die Randbedingungen zur Lösung der gestellten Aufgabe wichtig.



Abbildung 3.1.: Gebäudeszene „Haus-1“.

Diese Arbeit verfolgt den Ansatz, die Modellauswahl und das Aufstellen von geometrischen Randbedingungen durch eine Interpretation zu steuern. Die Funktionstüchtigkeit des Ansatzes wird anhand der Analyse von realen Gebäudeszenen gezeigt. Da diese bereits einen hohen Komplexitätsgrad besitzen (siehe Abbildung 3.1), wird hier zur Realisierung ein wissensbasierter Ansatz verfolgt. Wissensbasierte Systeme bieten zahlreiche Konzepte zur Lösung komplexer Aufgaben. Zudem bieten sie eine hohe Flexibilität bei der Entwicklung und Realisierung komplexer Modelle, die hochsprachlich in Form einer Wissensbasis vorliegen. Aufgrund der Trennung von Methoden und Fakten können Erweite-

rungen oder Abänderungen an andere, verwandte Aufgaben leicht vorgenommen werden.

3.2. Kameramodell

Das Kameramodell beschreibt die Beziehungen zwischen Punkten im Raum und deren Abbildung auf die Bildebene der Kamera. Hier wird angenommen, daß die Abbildung als Zentralprojektion (Lochkameramodell) hinreichend genau beschrieben werden kann und nichtlineare Linsenfehler mit einer Radialverzerrung 3. Ordnung beschrieben werden können. Zusätzlich wird eine Verschiebung des Kameratargets um den Hauptpunkt (Durchstoßpunkt der optischen Achse, siehe Abbildung 4.1) berücksichtigt. Das Modell entspricht dem von Tsai beschriebenen [Tsai87].

3.3. Oberflächenmodell

Das Oberflächenmodell ist ein spezieller Teil des Szenenmodells. Es dient der Beschreibung ¹ von Objektoberflächen, was das eigentliche Ziel der Analyse darstellt.

Das grundsätzliche Vorgehen zur Gewinnung einer Oberflächenbeschreibung gliedert sich hier in drei Schritte:

- Sensorbezogene Rückgewinnung von 3D-Tiefeninformation,
- Auswahl eines Oberflächenmodells und Voreinstellung der Parameter,
- Objektbezogene Schätzung der Oberflächenmodellparameter.

Zur Rückgewinnung der 3D-Tiefeninformation wird hier eine stereoskopische Tiefenschätzung eingesetzt (siehe Abschnitt 5.2). Diese liefert für jeden Bildpunkt (sofern meßbar) eine 3D-Koordinate, die, wenn auch fehlerbehaftet, einen Punkt auf der Objektoberfläche darstellt. Nach erfolgter Modellauswahl wird hieraus eine Oberflächenbeschreibung berechnet.

Die Modellauswahl hat entscheidenden Einfluß auf Qualität und Effizienz der Oberflächenbeschreibung. Die meisten in der Literatur beschriebenen Ansätze

¹aus Gründen der Durchgängigkeit und der Verwechslungsgefahr wird in dieser Arbeit das angepaßte Oberflächenmodell als Oberflächenbeschreibung bezeichnet

nutzen ein festes Oberflächenmodell. In [Koch97, Rie97] werden beispielsweise Splines zur Interpolation der Tiefenmessungen und anschließender Approximation durch Dreiecksnetze verwendet. Die Anzahl der Modellparameter und damit der möglichen Freiheitsgrade wird so eingestellt, daß die betrachteten Objektoberflächen mit einer hinreichenden Detailtreue rekonstruiert werden. Dies hat zum einen den Nachteil, daß ein Operator die Modellauswahl, bzw. die Einstellung der Modellparameteranzahl für jede Szene individuell durchführen muß. Desweiteren muß immer ein Kompromiß gefunden werden; hohe Detailtreue bedeutet hohe Anzahl an Modellparametern. Wodurch die Beschreibung von Objekten oder Objektkomponenten, die bestimmte einschränkende Oberflächeneigenschaften, wie beispielsweise Planarität, besitzen, ineffizient wird.

Mehr Effizienz in der Beschreibung ist möglich durch eine Zerlegung in Einzelobjekte und Objektkomponenten und anschließender Modellauswahl für jede Objektkomponente. Abbildung 3.2 zeigt dazu (als Aufsicht) ein Beispiel: Wird aus einer Anzahl von 3D-Punkten eine Oberfläche als Spline mit hoher Parameterzahl modelliert und anschließend geschätzt, so weist die Oberflächenbeschreibung eine hohe Detailtreue auf, die allerdings bei Überanpassung (zu viele freie Modellparameter) die Meßfehler der 3D-Punkte wiedergibt. Stellen die Punkte jedoch zwei sich schneidende Hauswände dar, so können sie sehr viel effizienter durch zwei Polygone beschrieben werden. Die Parameterreduktion ermöglicht, neben der Ersparnis an Parametern und der damit verbundenen Datenreduktion, eine Interpolation und einen Fehlerausgleich der Meßwerte.

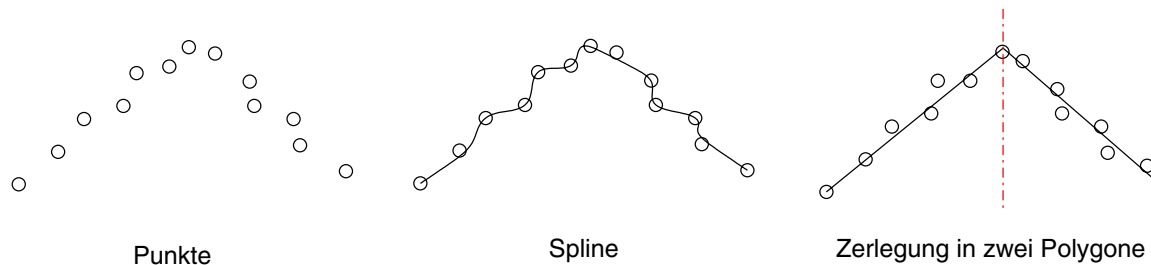


Abbildung 3.2.: Schätzung einer 3D-Oberflächenbeschreibung mit allgemeinem Oberflächenmodell (Mitte) und speziellem Modell bei erfolgter Zerlegung (rechts).

Aus dieser Betrachtung ergeben sich zwei Schlüsse: Zum einen zeigt es die Notwendigkeit einer kompositionellen Zerlegung der Szenenbeschreibung, zum anderen die Bedeutung von szenenspezifischem Vorwissen („ist eine Hauswand“).

Die Frage, ob ein Detail als geometrische Form oder als Textur beschrieben wird, ist ebenfalls bedeutungsabhängig. So wird es beispielsweise ein menschlicher

Betrachter als störend empfinden, wenn ein Schornstein nicht geometrisch beschrieben wird und nur in der Textur des Daches vorhanden ist. Während ein Gegenstand, der die gleichen Abmessungen besitzt, an anderer Stelle als Texturbeschreibung akzeptiert wird. Der Grund liegt darin, daß der menschliche Betrachter weiß, daß ein Dach meist einen Schornstein besitzt und er diesen in der 3D-Beschreibung erwartet.

Die kompositionelle Zerlegung der Szene erfordert die Möglichkeit einer strukturellen Beschreibung der Szene. Dies bietet neben der Möglichkeit, das Oberflächenmodell (und die Modellparameter) individuell auszuwählen, einen weiteren Vorteil: Objektspezifische Eigenschaften können in Form von Relationen zwischen Objekten oder Objektkomponenten formuliert werden. So können Gesetzmäßigkeiten wie beispielsweise Rechtwinkeligkeit zwischen Objektkomponenten für die Oberflächenrekonstruktion genutzt werden. Das Wissen über die Geschlossenheit eines Objektes führt zur Erzeugung konsistenter Oberflächenbeschreibungen.

3.4. Strukturiertes generisches Szenenmodell

Um den obengenannten Anforderungen von Interpretation und Rekonstruktion gerecht zu werden, wird hier ein strukturiertes generisches Szenenmodell vorgeschlagen, das folgende Inhalte und Eigenschaften umfaßt:

- a. Eine kompositionelle Zerlegung der Szene.
- b. Eine Menge von Oberflächenmodellen für die Rekonstruktion der Objekt-oberfläche.
- c. Formulierung von Randbedingungen und Einschränkungen der Oberflächenmodellparameter.
- d. Angabe von topologischen und geometrischen Beziehungen zwischen Objekten und Objektkomponenten.
- e. Zusammenfassung von Objekten (und deren Komponenten), die Ähnlichkeiten aufweisen zu Objektklassen.
- f. Herstellung von Bezügen zwischen Signal, Modell und Daten.
- g. Formulierung von alternativen Objektkomponenten.

Nach erfolgter kompositioneller Zerlegung können die Wissensinhalte aus den Punkten b,c und d unter Nutzung der Daten für die Gewinnung der Oberflächenbeschreibung genutzt werden. Dies erfordert jedoch zuvor eine erfolgreiche Interpretation. Dazu werden die Wissensinhalte in einer strukturellen, intensionellen Beschreibung zusammengefaßt.

Die Form der intensionellen Beschreibung, in der objektspezifische Wissensinhalte zusammengefaßt werden, ermöglicht eine sehr kompakte Wissensdarstellung durch die Einführung von Symbolen oder Objektklassen (Punkt e). Erwähnt werden soll an dieser Stelle der Unterschied zum Klassenbegriff bei objektorientierten Programmiersprachen: Bei objektorientierten Programmiersprachen umfaßt der Begriff der Klasse Konventionen, die festlegen, welche Daten die Objektinstanzen beinhalten (Liste der Daten-Member), wie diese zu erzeugen sind (Konstruktoren) und weitere Funktionen (Methoden). Es wird jedoch nicht festgehalten, welche Ausprägungen die Werte der einzelnen Instanzen annehmen dürfen. Eine intensionelle Beschreibung gibt in Form einer Interval- oder Mengenbeschreibung unter anderem genau dies an. Bei Attributen beispielsweise ist das durch Bereichsangaben möglich, die angeben, in welchem Intervall die Werte eines Attributs einer Instanz der jeweiligen Objektklasse liegen können.

Die Überprüfung der Szenenbeschreibung wie auch die Oberflächenrekonstruktion erfolgt anhand des Eingangssignals. Das Modell muß daher, wie in Punkt f aufgeführt, einen Bezug zwischen Signal, Modell und den gewonnenen Daten der Szenenbeschreibung erlauben.

Damit eine Vielzahl von Objekten, die sich zwar ähneln, aber in Details voneinander abweichen, beschrieben werden können, sieht Punkt g in der Liste oben, die Möglichkeit vor, alternative Komponenten zu definieren. Gebäude können beispielsweise durch eine Beschreibung des Mauerwerks und Kombination mit verschiedenen Dachtypen in einer Reihe von Häusertypen zusammengefaßt werden. Die Möglichkeit, Alternativen und Optionen zu beschreiben, ist somit ein wichtiges Merkmal, das eine Wissensrepräsentation aufweisen muß, um kompakte und pflegbare Wissensbasen zu ermöglichen. Ferner können damit Modelle generisch formuliert werden. Das heißt, es lassen sich beliebig komplexe Modelle in einer Wissensbasis beschreiben.

3.5. Vorgehen bei der Entwicklung des Szenenmodells

Bei der Entwicklung des Szenenmodells steht zuerst im Vordergrund, eine Beschreibung des möglichen Szeneninhalts zu finden, die eine automatische Er-

kennung der Objekte und Komponenten der Szene erlaubt. Dazu ist ein Vorrat an Symbolen zu definieren, der die Szenenobjekte und deren Komponenten benennt. Diese können dann durch ihre (inneren) Eigenschaften und durch Relationen zueinander beschrieben werden.

3.5.1. Bildung von Objektklassen und -hierarchien

Der Entwurf des Szenenmodells wird schrittweise durchgeführt. Das heißt, daß die bestehende Beschreibung immer weiter verfeinert wird. Die zu gewinnende intensionelle Beschreibung entsteht, wie in Abbildung 3.3 dargestellt, im wesentlichen durch Anwendung von zwei Prinzipien: Erstens durch eine Zerlegung der Szene bzw. der Szenenobjekte in Teile. Dadurch entstehen Beschreibungen von immer spezielleren Komponenten. Zweitens findet durch die Bildung von Objektklassen aus der Analyse von Daten eine Verallgemeinerung der Beschreibung statt. Diese beiden Prinzipien dürfen nicht unabhängig voneinander betrachtet werden, wie im folgenden erläutert wird.

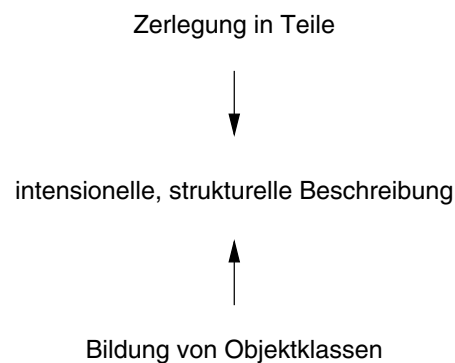


Abbildung 3.3.: *Vorgehen bei der Entwicklung eines strukturierten Szenenmodells*

Durch die hierarchische Zerlegung von Objekten können für die einzelnen Komponenten spezielle Randbedingungen formuliert und in der Oberflächenrekonstruktion genutzt werden. Auf der anderen Seite führt jedoch eine starke Spaltung in Teile unter Umständen zu einer langen (tiefen) Suche während der Interpretation. Das heißt, die Interpretation wird, insbesondere wenn Alternativen untersucht werden müssen, komplexer und damit kostenintensiv. Eine Aufspaltung in Komponenten sollte daher nur durchgeführt werden, wenn die Teilkomponenten gut interpretierbar sind und Randbedingungen für die Oberflächenrekonstruktion angegeben werden können, das heißt, wenn sie einen Beitrag zum Erreichen des Analyseziels liefern.

Die Frage, ob ein Objekt oder eine Objektkomponente gut interpretierbar ist, entscheidet sich zum einen dadurch, inwieweit sich dieses von anderen alternativen Interpretationen abgrenzt. Dies hat direkten Einfluß auf die Robustheit der Interpretation. Weiterhin muß ein Objekt durch das Signal unterstützt sein: Ist ein Objekt zu klein abgebildet, so wird es schwierig oder unmöglich, dieses zu interpretieren. Das Objekt ist dann nicht meßbar (vergleiche Abschnitt 3.6) und eine Darstellung im Szenenmodell ist nicht sinnvoll. Kann allerdings das Vorhandensein der Komponente sicher gefolgert werden, so ermöglicht dies eine Vervollständigung der Szenenbeschreibung. Ist beispielsweise eine Dachhälfte sichtbar, kann angenommen werden, daß eine zweite vorhanden ist.

Neben der kompositionellen Zerlegung der Szene ist die Bildung von Objektklassen das zweite wichtige Prinzip bei der Entwicklung eines strukturierten generischen Szenenmodells. Dabei wird durch Analyse der Daten ähnlichen Objekten ein Symbol zugeordnet und die gemeinsamen Eigenschaften in einer geeigneten Form festgehalten. Dieses Prinzip wird ebenso in der objektorientierten Programmierung angewendet: Aufbauend auf einer Analyse von Gemeinsamkeiten werden allgemeine Klassen entworfen und davon abgeleitete Klassen erhalten speziellere Eigenschaften.

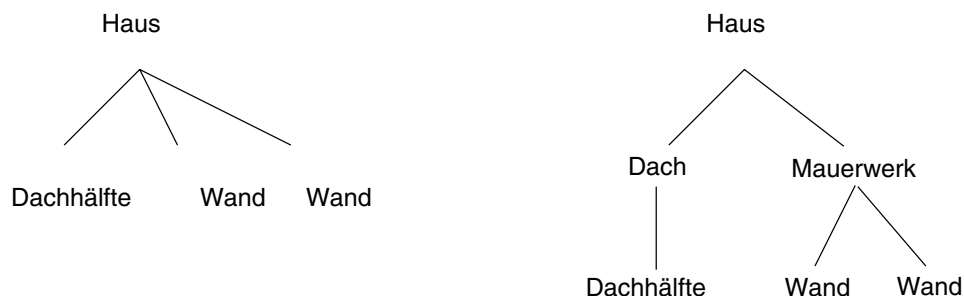


Abbildung 3.4.: Einführung von zusätzlichen Knoten für die Akkumulation von Informationen während der Interpretation

Viele Klassen können geradlinig durch direkte Umsetzung des Problems beschrieben werden: So ist bei der Anwendung hier klar, daß es Klassen für Wände und Dachhälften geben wird, da die Gewinnung ihrer dreidimensionalen Form das Analyseziel darstellt. Andere Klassen ergeben sich als zweckmäßig oder sogar notwendig für die Interpretation. So ist es beispielsweise günstig, wie in Abbildung 3.4 grafisch dargestellt, die Knoten *Dach* und *Mauerwerk* einzufügen. Diese stellen einen zusätzlichen Aufpunkt für das Ansammeln (Akkumulieren) von Informationen dar und werden beispielsweise für die Propagation von Vorwissen während der Interpretation genutzt. Eine Beschreibung dieses Vorgangs liefert der Abschnitt 7.5.

Um dem Anspruch einer generischen Beschreibung gerecht zu werden, muß das Szenenmodell in der Lage sein, alternative Interpretationen zu liefern. Als Repräsentation dienen hierfür optionale Teile und Spezialisierungen. Eine Spezialisierung bedeutet die Einführung einer neuen Klasse, die über das Vererbungskonzept die Eigenschaften der generelleren Klasse erbt und durch Überschreiben oder Hinzufügen von Eigenschaften sich von dieser unterscheidet. Es gibt Wissensinhalte, die sich sowohl durch Angabe von optionalen Teilen, als auch durch Einführung von spezialisierten Klassen lösen lassen: Beispielsweise hat ein Dach üblicherweise zwei Dachschrägen; Walmdächer besitzen vier Dachschrägen. Es ergeben sich zwei alternative Formen der Beschreibung:

- a. *Ein Dach besitzt mindestens zwei Dachschrägen und zwei optionale Dachschrägen.*
- b. *Ein Dach besitzt zwei Dachschrägen. Ein Walmdach besitzt vier Dachschrägen.*

Die Entscheidung, welche der beiden Beschreibungsformen gewählt wird, muß von Fall zu Fall entschieden werden. Die Darstellung als optionale Teile führt zu sehr kompakten Wissensbasen und sollte gewählt werden, wenn aufgrund des Signals relativ sicher entschieden werden kann, ob optionale Teile vorliegen oder nicht. Ist diese Entscheidung nicht so einfach zu treffen oder unterscheiden sich die Objekte in weiteren Eigenschaften voneinander, so ist die Einführung von spezialisierten Objektklassen mit dem damit verbundenen Aufwand gerechtfertigt.

3.5.2. Beschreibung von Objektrelationen

Relationen geben die Beziehungen zwischen Objekten oder Komponenten¹ an. Bei Verwendung einer hierarchischen Modellzerlegung, wie vorgeschlagen, stellt die Beziehung „Teil“² eine wichtige Relation dar und verbindet Objektklassen des Szenenmodells in Form eines Baumes miteinander. Zur Erhöhung der Sicherheit von Interpretation und Oberflächenrekonstruktion spielen jedoch weitere Relationen eine Rolle. Dies sind vor allem topologische und räumliche Relationen.

Es gibt verschiedene Möglichkeiten einen Sachverhalt, wie beispielsweise „Objekt A ist (räumlich) über Objekt B“ auszudrücken. Die erste Variante ist, wie in Abbildung 3.5 links dargestellt, eine spezielle Relation l zu definieren, die den

¹Teile oder Komponenten eines Objektes stellen wiederum Objekte dar und werden im folgenden als solche bezeichnet.

²Später wird diese Beziehung wie in der Literatur üblich als part-of bezeichnet.

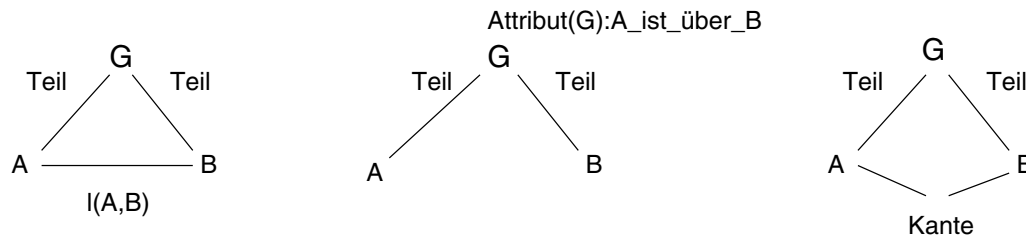


Abbildung 3.5.: Räumliche Beziehungen können durch eine Relation $l(A, B)$ zwischen Objekten (links), durch Attribute (Mitte) oder ein weiteres Objekt (rechts) dargestellt werden

räumlichen Bezug „ist über“ zwischen den Objekten A und B herstellt. Zum zweiten kann die Aussage durch Attribute dargestellt werden, wie in Abbildung 3.5 (Mitte) gezeigt: Das Attribut $A_ist_über_B$ des Objektes G erhält den Wert „wahr“, wenn sich Objekt A räumlich über Objekt B befindet.

Welche der beiden Varianten gewählt wird, muß von Fall zu Fall entschieden werden. Tritt beispielsweise das Objekt A optional auf, und kann das Erscheinen in der Szene nur unsicher aus den Signaleigenschaften geschlossen werden, so kann dies mit der Darstellung als Attribut flexibel und einfach ausgedrückt werden³. Bei der Darstellung als Relation müßte diese zusätzliche Attribute erhalten. Des weiteren wird die Auswertung der relationalen Darstellung aufwendig und kann unter Umständen nur problemabhängig noch effizient verarbeitet werden. In SIGMA [Mat90] wird hierzu vorgeschlagen, alle möglichen Ausprägungen der Relationen in Form von Hypothesen zu expandieren und anschließend durch Fusion sich „überlappender“ Hypothesen die Sicherheit der Ausgangshypothesen zu erhöhen. Die Überlappung wird dort problemabhängig im Bild überprüft.

Für die in dieser Arbeit betrachtete Problemstellung treten anders als in Luftbildszenarien Verdeckungen auf, und daher müssen mehrere Ansichten der Szenen verarbeitet werden. Die Wahrscheinlichkeit, daß Objekte in mindestens einer Teilansicht verdeckt sind, ist groß. Es erscheint hier daher die Darstellung als Attribut für geeignet, wenn die Objekte unsicher detektiert werden können.

Eine dritte Variante der Darstellung ist in Abbildung 3.5 rechts gezeigt. Soll eine räumliche, topologische Beziehung dargestellt werden, die sich zusätzlich auf einer anderen Abstraktionsebene manifestiert, so kann es zweckmäßig sein, die Beziehung durch eine oder mehrere Objektklassen auszudrücken. Stellen die Objekte A und B beispielsweise Hauswände dar, so kann die räumliche Beziehung „ A ist mit B verbunden“ durch ein Objekt $Kante$ dargestellt werden, das mit den

³In diesem Fall nimmt das Attribut den Wert „unbestimmt“ an.

Objekten A und B verbunden ist. Darüber hinaus kann die Erscheinung einer dreidimensionalen Kante in der Projektion, also im Kamerabild überprüft werden. Kann das Vorkommen der Objekte A und B sicher geschlossen werden, so ist diese Darstellung zweckmäßig.

3.6. Modellgültigkeitsbereich

Die in dieser Arbeit entwickelten Konzepte zur Darstellung und Nutzung von Wissen für die 3D-Szenenanalyse sind auf eine Vielzahl von Problemstellungen anwendbar. Die Funktionsweise wird am Beispiel der 3D-Beschreibung von Gebäudeszenen im Nahbereich demonstriert. Die für diese Anwendung entwickelte Wissensbasis, die darin enthaltenen oder verwendeten Modellannahmen und die Module zur Oberflächenrekonstruktion sind auf diese Problemstellung abgestimmt. Dieser Abschnitt beschreibt die dabei zugrundeliegenden Modellrandbedingungen, innerhalb derer das vorgeschlagene System anwendbar ist. Es gelten folgende Annahmen:

- Als Beleuchtungsmodell wird von einer diffusen Beleuchtung ausgegangen.
- Die zu analysierenden Körper sind starr und unbewegt.
- Die beiden (Stereo-)Kameras werden vor der Aufnahme kalibriert. Die inneren Kameraparameter und relative Orientierung der beiden Kameras zueinander sind damit bekannt (vergleiche Abschnitt 5.1).

Objekte werden von dem Analysesystem erfaßt, wenn sie sichtbar und vom System meßbar sind. Der letzte Punkt hängt von einer Vielzahl von Faktoren ab:

- den Signaleigenschaften der Kameras (Bildauflösung, Kontrast, Rauschen,...) und dem Abstand zwischen Objekt und Kamera und der Szenenbeleuchtung.
- Die Körperoberfläche muß eine gewisse Textur aufweisen, damit das verwendete Stereotiefenmeßverfahren arbeitet (vergleiche Abschnitt 5.2).
- Die initiale Segmentierung der Szene muß bestimmten Anforderungen genügen (Abschnitt 5.3). Insbesondere müssen die signifikanten Komponenten im Segmentierungsergebnis enthalten sein.

3.7. Diskussion des entwickelten Szenenmodells

Aufgrund der Komplexität der gestellten Aufgabe wurde ein strukturiertes generisches Szenenmodell gewählt. Dieses Modell ist in der Lage, die für die Rekonstruktion erforderlichen objektspezifischen Eigenschaften zu berücksichtigen.

Aus den Überlegungen zum systematischen Entwurf des Szenenmodells folgen konkrete Anforderungen an die gewählte Repräsentation von Daten und Wissensinhalten. Diese müssen insbesondere in der Lage sein, die strukturellen Eigenschaften und Attribute zu erfassen.

Die Betrachtung der Frage, ob ein Wissensinhalt relational oder als Attribut formuliert werden soll, zeigt, daß die Wahl der Darstellung durch die spätere automatische Auswertung beeinflusst wird. Eine getrennte Betrachtung von Wissensrepräsentation und Kontrolle ist somit nicht sinnvoll. Statt dessen müssen die Auswertungskomponente und die Darstellung aufeinander abgestimmt sein.

4. Wissens- und Datenrepräsentation

Once a problem is described using an appropriate representation, the problem is almost solved.

P.H. Winston

Eine Repräsentation legt in Form von Konventionen fest, wie die zur Lösung eines Problems relevante Information beschrieben wird. Neben der reinen Nutzinformation, das sind die durch die Analyse gewonnenen Oberflächenbeschreibungen, müssen die Eingangsdaten, Vorwissen und Zwischenergebnisse repräsentiert werden. Der problemgerechte Entwurf einer geeigneten Repräsentation ist entscheidend für die Effizienz des gesamten Analysesystems.

Dieser Abschnitt stellt einige Repräsentationsformen für die Daten des wissensbasierten Analysesystems zusammen. Die Auswahl ist dabei sehr stark eingeschränkt auf die für diese Arbeit relevanten Darstellungsformen. Aus den hier vorgestellten Repräsentationen werden in den kommenden Abschnitten dieser Arbeit die zur Lösung der Aufgabe verwendeten Datenstrukturen entwickelt.

4.1. Repräsentation von Objektoberflächen für Visualisierungsaufgaben

Die in dieser Arbeit verwendete Repräsentation von Objektoberflächen basiert auf dem in Abschnitt 2.2 beschriebenen Synthesemodell. Die wesentliche Annahme des Synthesemodells besteht darin, die Darstellung in eine (vereinfachte) Oberflächenform und die Oberflächentextur zu trennen. Damit wird eine fotorealistische Darstellung von komplexen Objekten mit endlichem Aufwand möglich.

4.1.1. Geometrische Beschreibung der Objektoberfläche

Objekte können durch verschiedene, problem- und anwendungsabhängige Repräsentationen geometrisch beschrieben werden. So werden beispielsweise für

bestimmte Anwendungen Volumendarstellungen oder Punktwolken verwendet. Für die Anwendung der Visualisierung ist eine Repräsentation als Oberflächenmodell hinreichend und notwendig: Die inneren Eigenschaften, die beim Volumenmodell dargestellt werden können, sind bei nichttransparenten Objekten im allgemeinen nicht sichtbar. Auf der anderen Seite bietet ein geschlossenes Oberflächenmodell alle Möglichkeiten Objektverdeckungen oder Lichtinteraktionen (z.B. Schattenwurf) realistisch und effizient zu behandeln.

Als Oberflächenmodelle sind üblich: Splines (Bézier-Splines, NURBS, Thin-Plate-Splines) für die Darstellung kontinuierlicher Oberflächen und Polygone für stückweise planare Oberflächen [Hos92, Fol92]. Ferner werden Dreiecksnetze verwendet, um kontinuierliche Oberflächen zu approximieren. Der Grund für die Wahl der letzteren gegenüber der Darstellung als Splines liegt in der einfacheren Beschreibung und Mathematik von Dreiecksnetzen, sowohl für Analysezwecke [Grau96, Grau97], als auch für die Bildsynthese [Fol92].

4.1.2. Texturkarten

Die nicht als geometrische Form ausgeprägten Strukturen oder Details einer Oberfläche werden als Textur erfaßt und parametrisch oder in Form von Texturkarten beschrieben. Die Textur wird in der Bildsynthese eingesetzt, um je nach Beleuchtungsmodell die diffuse Reflexion, die Oberflächenemission (kein Beleuchtungsmodell) oder die Oberflächennormale (Bump-Mapping [Fol92]) zu modulieren. In dieser Arbeit wird gemäß des Analysemodells (Abschnitt 3.6) die Textur zur Modulation der diffusen Reflexionseigenschaften der Objektoberfläche verwendet.

Die parametrische Darstellung wird in der Bildanalyse (z.B. [Hara92]) verwendet, um anhand der im Bild gemessenen Parameter Bildbereiche zu klassifizieren (auch zur Segmentierung). In der Computergrafik sind parametrische Texturmodelle für die Texturierung synthetischer Objekte üblich. Von Vorteil ist hier, daß die Textur phasenrichtig über praktisch unbegrenzt ausgedehnte Oberflächen gespannt werden kann und relativ wenige Daten beansprucht.

Zur Darstellung von beliebigen Texturen realer Objekte dienen Texturkarten [Fol92]. Diese erfassen die Oberflächentextur in einem zweidimensionalen Feld mit den Koordinaten (u,v) . Das Koordinatensystem der Texturkarte wird auf das lokale Koordinatensystem eines 3D-Oberflächenprimitivs (Polygon oder Dreieck) abgebildet. Bei der Bildsynthese wird aus dieser Zuordnung eine perspektivische Projektion der Texturkarte auf die Bildebene berechnet. Bei der Erzeugung von Texturkarten wird die Abbildung entsprechend umgekehrt (Abschnitt 8.4).

4.2. Kamerarepräsentation

Als Modell, das die Abbildung dreidimensionaler Punkte in die Bildebene (Kameratarget) beschreibt, wurde in dieser Arbeit die perspektivische Abbildung (Zentralprojektion) unter Berücksichtigung von Radialverzerrungen gewählt (vergleiche Abschnitt 3.2). Dieses Kameramodell stellt die zur Rekonstruktion notwendigen Bezüge zwischen Bild- und Raumelementen her.

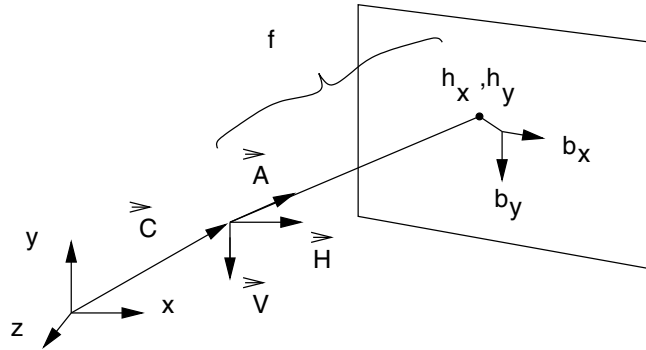


Abbildung 4.1.: Die CAHV-Kamerarepräsentation

Als Repräsentation einer Kamera dient die von Yakimovski und Cunningham [Yak78] vorgeschlagene CAHV-Kamerarepräsentation (Abbildung 4.1), die um die Radialverzerrung erweitert wurde. Die Gleichung zur Abbildung eines Punktes $\vec{P}$ in $[m] \in \mathbf{R}^3$ auf einen Punkt $\vec{p} = (b_x, b_y)^T$ in $[pel] \in \mathbf{R}^2$ in der Bildebene lautet:

$$\vec{p} = \frac{1}{(\vec{P} - \vec{C}) \cdot \vec{A}} \cdot \begin{pmatrix} (\vec{P} - \vec{C}) \cdot \vec{H} \\ (\vec{P} - \vec{C}) \cdot \vec{V} \end{pmatrix} \quad (4.1)$$

$$\vec{H} = \frac{f}{s_x} \cdot \vec{H}_0 + h_x \cdot \vec{A} \quad (4.2)$$

$$\vec{V} = \frac{f}{s_y} \cdot \vec{V}_0 + h_y \cdot \vec{A} \quad (4.3)$$

Dabei stellen das Kameraprojektionszentrum $\vec{C}$ in $[m] \in \mathbf{R}^3$, der Einheitsvektor der optischen Achse $\vec{A}$ und die Einheitsvektoren $\vec{H}_0, \vec{V}_0$ die sogenannten **äußeren Kameraparameter** dar. Die Vektoren $\vec{A}, \vec{H}_0, \vec{V}_0$ bilden ferner ein Orthonormalsystem.

Die **inneren Kameraparameter** sind: Die Brennweite f in [m], die Hauptpunktverschiebung $\vec{h} = (h_x, h_y)^T$ in [pel] $\in \mathbb{R}^2$, die Größe eines Bildpunktes in der Bildebene s_x, s_y in $[\frac{m}{pel}]$, sowie der Radialverzerrungskoeffizient ρ_1 in [pel²].

Die Vektoren $\vec{H}$ und $\vec{V}$ erhalten die Einheit [pel] $\in \mathbb{R}^3$.

Die Radialverzerrung wird nach Anwendung von Gl. 4.1 in der Bildebene berücksichtigt:

$$\vec{p} = (1 + \rho_1 \cdot |\vec{p}' - \vec{h}|^2) \cdot (\vec{p}' - \vec{h}) \quad (4.4)$$

Dabei stellt $\vec{p}'$ den real gemessenen und $\vec{p}$ den verzerrten Bildpunkt dar.

4.3. Repräsentation von Vorwissen

Für die Formulierung von Fakten in einer Wissensbasis sind verschiedene Repräsentationen entwickelt worden. Dieser Abschnitt stellt eine Auswahl der für die Lösung des betrachteten Problems relevanten Wissensrepräsentationen zusammen. Für eine vollständigere Übersicht sei auf die entsprechende Literatur verwiesen (z.B. [Win92, Shap92]).

4.3.1. Prädikatenlogik, Produktionssysteme und Regeln

Die *Prädikatenlogik* erster Ordnung gestattet die Darstellung von Begriffen und Beziehungen zwischen Begriffen und ist als klassische Logikrepräsentation weit verbreitet. Die Darstellung von Unsicherheiten oder numerischer Daten, wie in der vorliegenden Anwendung gefordert, ist jedoch nicht ohne weiteres möglich.

In *Produktionssystemen* ist Wissen in Form von Regeln repräsentiert. Eine Regel besteht aus einem Bedingungsteil und einem Aktionsteil. Zur Auswertung wird für alle Regeln überprüft, ob ihr Bedingungsteil für die Daten erfüllt ist. Dies kann jedoch für mehrere Regeln gleichzeitig der Fall sein. Die Kontrolle des Produktionssystems muß dann mit Hilfe einer Konfliktlösungsstrategie eine Regel auswählen. Die Auswertung von Regeln ist daher recht aufwendig. Ferner neigen Wissensbasen, die durch eine große Zahl von Regeln realisiert sind, dazu, unübersichtlich zu werden.

4.3.2. Semantische Netze

Semantische Netze haben den Vorteil, daß selbst größere Mengen an Fakten, die untereinander in Beziehung stehen, für den Menschen übersichtlich und verständlich bleiben. Sie setzen sich zusammen aus Knoten, die einen Begriff oder ein Objekt charakterisieren, und den Kanten zwischen den Knoten, die die Beziehungen der Knoten zueinander repräsentieren. Die Kanten tragen wie die Knoten eine Bezeichnung, die die Bedeutung der Kante festlegt.

Werden keine weiteren Konventionen festgelegt, werden semantische Netze unstrukturiert und die automatische Auswertung problematisch. In der Literatur wurden daher unterschiedliche Erweiterungen und Konventionen bezüglich der Kanten und Knoten semantischer Netze festgelegt (Übersicht z.B.: [Win92, Shap92]):

Die von Minsky beschriebenen Frames [Min75] gestatten beispielsweise eine sehr kompakte Darstellung. Ein Frame enthält sogenannte Slots, die wiederum weitere Frames enthalten können oder Eigenschaften in Form von Werten beinhalten. Ziel ist es, so alle Objekteigenschaften in einem Frame zusammenzufassen.

Weitere Festlegungen betreffen die Netzwerkkanten. Prinzipiell sind wie bei den Knoten beliebig viele Relationen möglich. Um die Verarbeitung zu vereinfachen, beschränken sich die meisten realisierten Systeme auf einen mehr oder weniger kleinen Satz von Kantentypen. Mit Hilfe der Netzwerkkante *part-of* ist eine kompositionelle Zerlegung der Wissensbasis möglich. Insbesondere können damit Objekt- und Teile-Hierarchien dargestellt werden. Eine weitere wichtige Kante ist die *is-a*-Kante. Sie ist, wie die *part-of*-Kante, bei praktisch allen Netzwerksprachen vorhanden. Über diese Kante werden allgemeine Eigenschaften von einem Knoten auf den oder die untergeordneten Knoten vererbt. In diesem können die Eigenschaften überschrieben oder ergänzt werden (Spezialisierung).

Die Netzwerksprache des Analysesystems ERNEST [Nie90, Kum93] reduziert die Anzahl der erlaubten Netzwerkkanten auf nur vier Typen: *part-of*, *is-a*, *concrete-of* und *instance-of*. Die Kante *concrete-of* verbindet dabei Knoten unterschiedlicher Abstraktionsebenen und entspricht der Kante *appearance-of* in SIGMA [Mat85, Mat90]. Die *instance-of*-Kante verbindet Konzepte mit Instanzen. Ein Konzept ist ein Knoten der Wissensbasis, der eine intensionelle Beschreibung eines Begriffes oder Objektes darstellt, das heißt, er gibt an, welche Merkmale und Beziehungen ein Objekt aufweisen muß, um Ausprägung des Begriffes zu sein. Die Ausprägungen werden als Instanz bezeichnet und mit einer *instance-of*-Kante mit dem dazugehörenden Konzept verbunden.

4.3.3. Prozedurales Wissen

Prozedurales Wissen dient der Beschreibung von Verarbeitungswissen, das insbesondere die Verarbeitungsstrategie festlegt. Im Zusammenhang mit einer strukturierten Wissensbasis dient es vor allem der Formulierung von Transformationen zwischen den unterschiedlichen Schichten der Wissensbasis [Nie90a]. So kann beispielsweise das Wissen über den Abbildungsvorgang in einer Kamera als Projektionsvorschrift in einer Prozedur formuliert werden und so die zweidimensionale mit der dreidimensionalen Schicht verbinden.

Als Repräsentation von prozeduralem Wissen eignen sich neben allgemeinen Programmiersprachen besonders Skriptsprachen, wie beispielsweise Tcl/Tk [Oust94].

5. Bildverarbeitung

Die Bildverarbeitung umfaßt alle Verarbeitungsschritte, die unmittelbar Bildinformation verarbeiten. Das sind insbesondere die digitalisierten Kamerabilder aus denen Bildmerkmale, wie Liniensegmente, extrahiert werden. Zur Berechnung von Tiefenwerten durch einen stereoskopischen Ansatz wird jeweils ein Kamerapaar verwendet, das zuvor kalibriert wurde. Das dazu verwendete Kalibrierverfahren wird im nächsten Abschnitt beschrieben. Die gewonnenen Tiefenwerte zerlegt eine Segmentierung in Regionen, die zu einer planaren Oberfläche gehören.

5.1. Kamerakalibrierung

Zur Erfassung der Szenentiefe dient ein Stereosensor, bestehend aus zwei fest auf einem Stativ montierten Digitalkameras vom Typ Kodak DCS 410 mit 14 mm Objektiven. Die Anordnung ist in Abbildung 5.1 dargestellt.

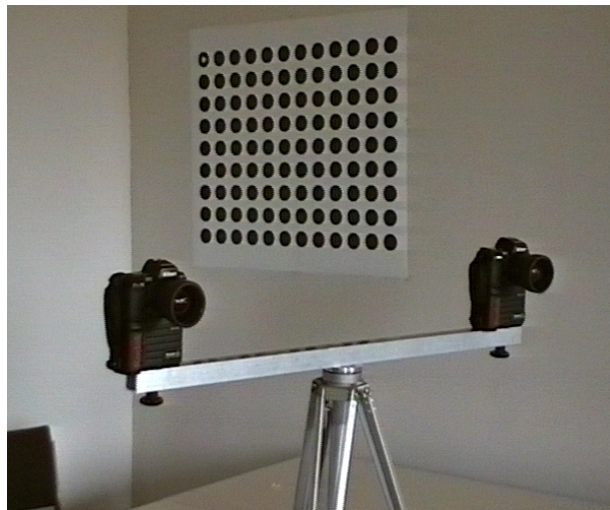


Abbildung 5.1.: *Stereosensor mit Kalibriermuster*

Zur Kalibrierung des Stereosensors wird das von Tsai [Tsai87] beschriebene Verfahren eingesetzt. Dazu wird das ebenfalls in Abbildung 5.1 dargestellte Kalibriermuster von beiden Kameras aufgenommen. Die geometrische Anordnung der Kalibriermarken ist durch vorherige Vermessung bekannt. Anschließend werden die Projektionen der Kalibriermarken automatisch im Bild gesucht und der Mittelpunkt exakt bestimmt. Mit Hilfe der Zuordnung der Marken im Bild zu den bekannten dreidimensionalen Positionen rekonstruiert das in [Tsai87] beschriebene Verfahren die Kameraparameter. Dabei werden die inneren Kameraparameter Brennweite, Hauptpunktverschiebung und Radialverzerrung, sowie die äußeren Kameraparameter Position und Orientierung der Kameras zum Kalibriermuster berechnet. Die für die beiden Einzelkameras gewonnenen äußeren Kameraparameter werden in eine relative Kameraorientierung umgerechnet. Das heißt, es wird die Position und Orientierung der zweiten Kamera auf die erste bezogen.

5.2. Stereoskopische Tiefenschätzung

Die aus der Kalibrierung gewonnenen Parameter beschreiben die Kameraanordnung geometrisch. Sind die Koordinaten von zwei Bildpunkten, die jeweils eine Projektion desselben Raumpunktes in die beiden Kamerabildebenen darstellen, bekannt, so kann daraus die Lage des Raumpunktes berechnet werden. Diese ergibt sich durch Triangulation. Der Raumpunkt ergibt sich dabei durch die Schnittbildung der zwei Sichtlinien der beiden Punkte. Eine Sichtlinie ist durch den Kamerabrennpunkt und einen Bildpunkt festgelegt.

Die eigentliche Aufgabe der stereoskopischen Tiefenschätzung besteht nun darin, zu einem Bildpunkt oder Bildmerkmal eines Kamerabildes automatisch den entsprechenden Punkt oder das entsprechende Merkmal in einem zweiten Bild zu finden. Diese Zuordnung heißt Korrespondenzanalyse. In der Literatur sind dabei eine Reihe von kantenorientierten und auch flächenorientierten, beziehungsweise blockbasierten Verfahren beschrieben worden (Übersicht [Dhon89, Lem88]).

In dieser Arbeit wird ein blockbasiertes Verfahren eingesetzt, das mit Hilfe der normierten Kreuzkorrelation zwischen zwei Fenstern die Korrespondenzen bestimmt. Zur Erhöhung der Zuordnungssicherheit werden zusätzliche Randbedingungen wie das Eindeutigkeits- und das Ordnungsprinzip eingebracht. Zur Beschränkung des Suchaufwandes werden die Kamerabilder auf Standardepipolargeometrie entzerrt. Dadurch liegen im zweiten Bild die zu einem Bildpunkt des ersten Bildes korrespondierenden Punkte in einer horizontalen Bildzeile. Damit reduziert sich die Korrespondenzsuche auf eine eindimensionale Suche ent-

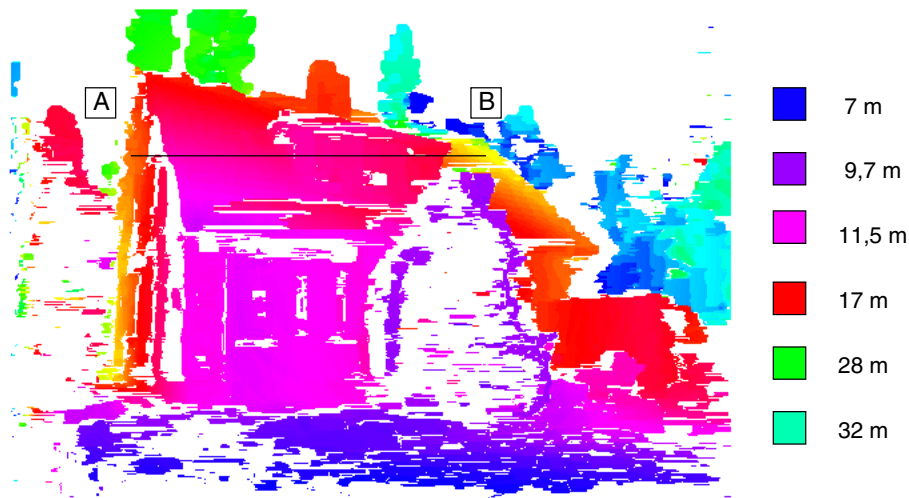


Abbildung 5.2.: Tiefenkarte der Szene „Haus-1“ in Farbdarstellung.

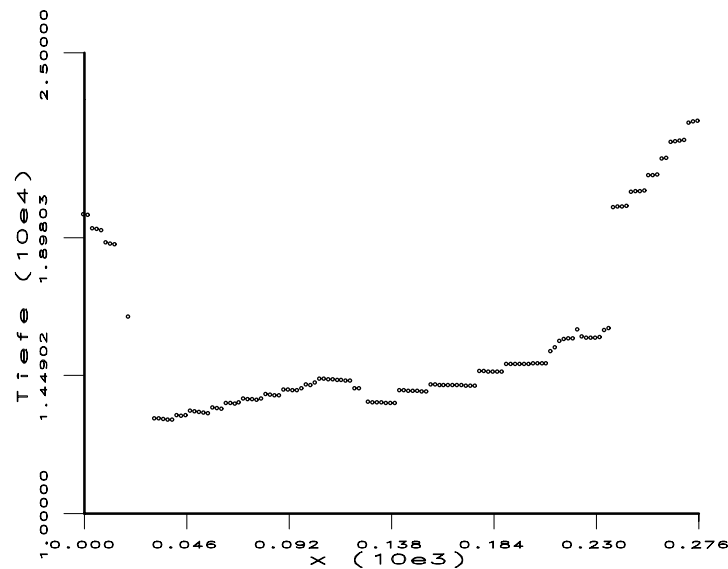


Abbildung 5.3.: Profil der Tiefenkarte entlang des Schnittes A-B in Abb. 5.2.

lang dieser Zeile. Eine dynamische Programmierung berechnet für jeweils eine Zeile die korrespondierenden Bildpunkte unter Minimierung einer Kostenfunktion, die die aufgestellten Randbedingungen und Korrelationen der Bildfenster berücksichtigt [Cox92]. Die Einbeziehung der benachbarten Bildpunkte in die Kostenfunktion [Fal94] und ein hierarchischer Ansatz [Fal97a] erlauben eine robuste Korrespondenzanalyse, die relativ dichte Disparitätskarten liefert. Die Dis-

parität gibt dabei die Verschiebung des Punktes eines Bildes in die Position in einem zweiten Bild an. Das Verfahren wird daher als Disparitätsschätzer bezeichnet.

Mit Hilfe der Triangulation werden die Disparitäten in Tiefenwerte umgerechnet und in Tiefenkarten gespeichert. Die Abbildung 5.2 zeigt die Tiefenkarte der Szene „Haus-1“, das dazugehörige linke Kamerabild ist in Abbildung 3.1 dargestellt. Eine Beschreibung des Stereosensors mit den hier kurz skizzierten Verarbeitungsschritten gibt [Koch97].

5.3. Segmentierung der Tiefenkarten

Die Segmentierung hat die Aufgabe, Bilder anhand eines Einheitlichkeitskriteriums in Teile zu zerlegen. Wie bei der Segmentierung von Luminanz- oder Farbbildern sind für die Segmentierung von Tiefenkarten eine Reihe von unterschiedlichen Verfahren beschrieben worden.

Die meisten in der Literatur bekannten Verfahren basieren auf der Auswertung des lokalen Oberflächennormalenvektors (beispielsweise [Hoff87, Koch96, Lej96, Sab93], Hover et al. [Hoov96] geben eine tabellarische Übersicht über einige Verfahren zur Tiefensegmentierung und deren Leistungsfähigkeit). Der Normalenvektor wird aus der lokalen Nachbarschaft des betrachteten Tiefenwertes berechnet. Das setzt zum einen voraus, daß die verwendeten Tiefenkarten sehr dicht sind, das heißt, es darf nur wenige unbestimmte Tiefenwerte geben. Zum anderen dürfen die Tiefenwerte nicht zu stark diskretisiert sein. Wie in Abbildung 5.3 zu sehen ist, liefert der hier verwendete Disparitätsschätzer jedoch relativ stark diskretisierte Tiefenwerte.

Besser geeignet für den verwendeten Schätzer sind daher Segmentierungsverfahren, die ein lokales Oberflächenmodell annehmen. Durch ein iteratives Region-Growing werden die dazugehörigen Tiefenwerte bestimmt und anschließend die Modellparameter des lokalen Oberflächenmodells angepaßt. Abbildung 5.4 zeigt das Segmentierungsergebnis der Tiefenkarte aus Abbildung 5.2. Der dabei verwendete Segmentierungsalgorithmus basiert auf dem von Leonardis et al. vorgestellten Verfahren [Leon93] unter Verwendung von planaren Teilflächen (Ebenen). Zusätzlich wurde der Algorithmus um eine adaptive, tiefenabhängige Schwellenberechnung erweitert [Teich96], wodurch das Verfahren robuster gegenüber Variation der Parametereinstellungen wird.



Abbildung 5.4.: Ergebnis der Tiefenkartensegmentierung

5.4. Diskussion der Bildverarbeitungskomponenten

Die in diesem Abschnitt vorgestellten Bildverarbeitungskomponenten stellen die Basis für das Szenenanalysesystem dar. So gehen die aus dem Stereosensor gewonnenen Tiefenmessungen in die Oberflächenrekonstruktion und zum Teil auch in die Interpretation ein. Die Qualität und Genauigkeit der berechneten Tiefenkarten hat somit großen Einfluß auf das Endergebnis. Aufgrund des durch die nachfolgende Interpretation eingebrachten Zusatzwissens war für die untersuchten Szenen die Genauigkeit in jedem Fall ausreichend.

Viel kritischer ist hingegen der Einfluß der Segmentierung. Diese stellt die Basis für die Interpretation dar. Anhand der konkreten Analysesituationen wird aus den Eigenschaften der Segmentierung die Analysestrategie für die Interpretation entwickelt. Für die hier betrachteten Gebäudeszenen liefert das entwickelte Segmentierungsverfahren eine relativ kleine Anzahl unterschiedlicher Regionen (ca. 15-30 je nach Einstellung der Segmentierungsparameter). Für die nachfolgende Interpretation ist es dabei wichtig, daß die wichtigen Komponenten der Szene im Segmentierungsergebnis unterschieden werden. Die Segmentierung führt eine Gruppierung unter Verwendung von lokalen Signaleigenschaften durch. Daher muß im Ergebnis immer mit fehlerhaften Zuordnungen gerechnet werden. Für die Anwendung hier ist daher damit zu rechnen, daß einige Komponenten, obwohl sie im Kamerabild sichtbar sind, nicht im Segmentierungsergebnis unterschieden werden. Fehlt eine Komponente in allen Segmentierungsergebnissen der zur Analyse verwendeten Ansichten, so ist diese für das System nicht meßbar.

Für die Segmentierung bedeutet das, daß die Segmentierungsparameter so eingestellt werden müssen, daß die signifikanten Szenenkomponenten unterschieden werden. Da die Analyse auf der initialen Segmentierung basiert, ist diese Einstellung relativ kritisch. Ein robusteres Verhalten ist zu erwarten, wenn die Bildverarbeitung, insbesondere die Segmentierung, nicht einschrittig arbeitet, sondern eng mit der Interpretationskomponente gekoppelt ist. Das würde es ermöglichen, Modellwissen aus der Wissensbasis zur Segmentierung zu verwenden. Für die Verwendung von Vorwissen sind aber die bekannten Segmentierungsverfahren bislang nicht ausgelegt. An dieser Stelle besteht ein Entwicklungsbedarf.

6. Netzwerksprache zur szenenspezifischen Wissensrepräsentation

Zur Repräsentation des in Abschnitt 3 entworfenen strukturierten generischen Szenenmodells wurde eine Netzwerksprache entwickelt, die die zur Lösung der Aufgabe erforderlichen Daten und Wissensinhalte in einer einheitlichen, objektorientierten Darstellung verkörpert. Für Interpretation und Oberflächenrekonstruktion sind neben den objektbezogenen Eigenschaften insbesondere die Beziehungen zwischen den Objekten von großer Bedeutung. Daher basiert die gewählte Darstellung auf einem semantischen Netz.

Die entworfene Netzwerksprache enthält als Bestandteile Knoten, Kanten und Attribute zur deklarativen Darstellung der objektbezogenen Eigenschaften. Prozedurale Wissensinhalte können in Form von Berechnungsfunktionen und Regeln angegeben werden. Die einzelnen Elemente der Netzwerksprache und einige Implementierungsaspekte werden in den folgenden Abschnitten dieses Kapitels vorgestellt. In Abschnitt 6.7 wird darauf aufbauend die zur Analyse von Gebäudeszenen entwickelte Wissensbasis vorgestellt.

Neben dem hier verfolgten Ziel der Beschreibung von Szenenwissen über Gebäude im Nahbereich [Grau95, Grau97a] wurden die Netzwerksprache und die Kontrollmechanismen für die Nutzung im Bereich der Fernerkundung aus Luftbildern erweitert und eingesetzt [Toen96, Lie97].

6.1. Knoten

Die Knoten eines semantischen Netzes repräsentieren ein Objekt oder einen Begriff und tragen einen eindeutigen Namen. Abbildung 6.3 zeigt ein einfaches semantisches Netz. Die Knoten, in der Abbildung 6.3 als Rechtecke dargestellt, sind Konzepte, die die dargestellten Begriffe intensionell beschreiben (vergleiche Abschnitt 4.3.2). Das Ziel der Interpretation ist es, mit Hilfe der intensionellen Darstellung eine Beschreibung der Szene zu finden. Die durch das Signal gestützte

Beschreibung wird durch Instanzen dargestellt. Die Ellipse in Abbildung 6.3 stellt eine Instanz des Konzeptes *Haus* dar. Instanzen sind mit genau einer *Instance-Of*-Kante mit dem dazugehörigen Konzept verbunden und damit Bestandteil des semantischen Netzes. Da es mehrere Instanzen eines Konzeptes geben kann, tragen die Instanzen den Namen des Konzeptes und eine eindeutige Nummer.

Die während der Interpretation erzeugten Instanzen liegen gemeinsam mit ihren Konzepten in einem Netz vor. Um zwischen Wissensbasis (nur Konzepte) und Szenenbeschreibung unterscheiden zu können, gelten folgende Namensfestlegungen:

Def. 6.1 Ein semantisches Netz, das nur Konzeptknoten enthält, heißt Wissensbasis.

Def. 6.2 Ein semantisches Netz, das neben Konzeptknoten Instanzen enthält, heißt Interpretationszustand oder bei abgeschlossener Interpretation Szenenbeschreibung.

Im folgenden werden Konzepte mit dem Buchstaben C bezeichnet, die j -te Instanz des Konzeptes C als $I_j(C)$.

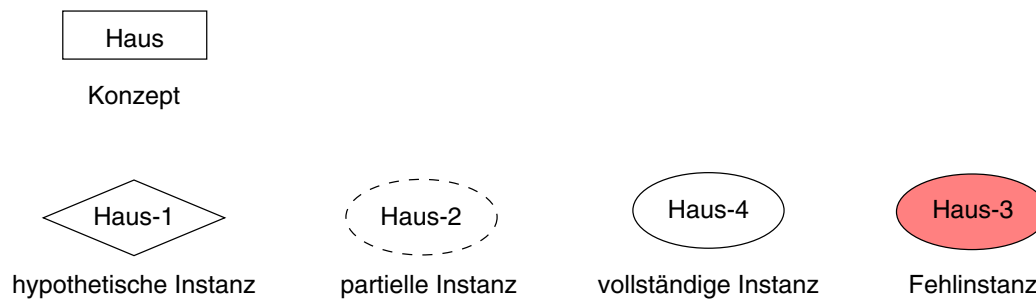


Abbildung 6.1.: Darstellung verschiedener Knotentypen

Neben den in Abbildung 6.3 dargestellten Konzepten und den sogenannten kompletten Instanzen können während der Interpretation mehrere Zwischenstufen von Netzwerkknoten auftreten, die als Zwischenergebnisse erzeugt werden. Diese werden hier wie folgt bezeichnet:

- *Hypothetische Instanzen* $I_j^H(C)$ werden direkt von dem Konzept abgeleitet und stellen die erste Stufe in der Interpretation dar, das heißt, sie sind noch nicht durch die Daten bestätigt.
- *Partielle Instanzen* $I_j^P(C)$ stellen (noch) unvollständige Instanzen dar, die aber bereits einen ersten Bezug zu den realen Daten haben.

- *Vollständige Instanzen* $I_j^K(C)$ sind strukturell vollständig und alle Daten sind berechnet.
- *Fehlinstanzen* $I_j^F(C)$ sind Instanzen, bei denen kein Bezug zu den Daten hergestellt werden konnte. Ein solcher Fall tritt beispielsweise auf, wenn zu einem optionalen Teil kein Segmentierungsergebnis gefunden wurde.

Die im folgenden verwendeten grafischen Symbole für die verschiedenen Knotentypen sind in Abbildung 6.1 dargestellt.

6.2. Attribute

Attribute repräsentieren die Objekteigenschaften eines Konzeptes. In der Regel stellen sie meßbare Werte dar, wie zum Beispiel die Farbe oder Höhe einer Wand. Sie können jedoch auch abstrakte Werte repräsentieren, die für die Interpretation benutzt werden.

Es gibt unterschiedliche Wertetypen von Attributen. Die gebräuchlichsten sind: Fließkommaattribut und (3D-)Vektor. Daneben können mit einem generischen Typ beliebige benutzerdefinierte Größen dargestellt werden.

Fließkommaattribut

Ein Attribut A enthält als Daten einen minimalen und einen maximalen Attributwert $V(A) = (V_{min}, V_{max})$ und einen Wertebereich $\epsilon(A) = (\epsilon_{min}, \epsilon_{max})$. Die Werte liegen im Definitionsbereich D des Attributs. Für ein Fließkommaattribut ist $D = \mathbf{R}$. Abbildung 6.2 stellt die Größen für ein Fließkommaattribut auf dem Zahlenstrahl dar.

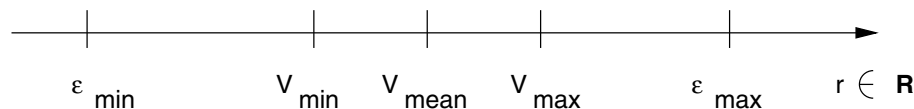


Abbildung 6.2.: Größen eines Fließkommaattributs

Mit Hilfe des Wertebereiches ist es möglich, Wertetoleranzen und daran geknüpft Unsicherheiten bei der Berechnung der Attributwerte darzustellen. Die Höhe eines Hauses berechnet sich beispielsweise aus der Höhe der Stockwerke plus der

Höhe des Daches. Zu Beginn der Interpretation kann der Wert nur geschätzt werden (er wird aus der Wissensbasis übernommen) und muß den möglichen Toleranzbereich abdecken. Das heißt, V_{min} = „kleinste vorkommende Haushöhe“ und V_{max} = „größte Höhe“. Im Laufe der Interpretation kann, sobald die entsprechenden Daten aus der Bildverarbeitung zugeordnet wurden, der Toleranzbereich weiter eingeschränkt werden: der Attributwert wird sicherer. Im Falle $V_{min} = V_{max}$ gilt der Attributwert als sicher bestimmt.

Der Wertebereich $\epsilon(A)$ dient der Bewertung eines Attributes während der Interpretation (siehe Abschnitt 7.3). Ferner wird er zur Top-Down-Propagierung des Vorwissens benutzt. Dieses Prinzip und das Zusammenspiel mit der Werteberechnung während der Interpretation behandelt Abschnitt 7.5 ausführlich.

Vektorattribute

Vektorattribute stellen dreidimensionale Vektoren in $\mathbf{R}^3$ dar. Das Attribut $V_{min} \neq V_{max}$ stellt dabei ein Volumen dar.

Zur Darstellung von Richtungsvektoren für Flächennormalen oder ähnliches werden hier Kugelkoordinaten (r, θ, ϕ) mit $r = 1$ gewählt. Attributwert und -bereich werden somit durch einen zweidimensionalen Vektor in $\mathbf{R}^2$ beschrieben: $V(A) = ((\theta_{min}, \phi_{min})^T, (\theta_{max}, \phi_{max})^T)$, mit $0 < \theta < \pi$ und $0 < \phi < 2\pi$. Der Wertebereich von $\epsilon(A)$ ist entsprechend.

Bei Angabe der Bereiche ist zu beachten, daß die Variablen ϕ und θ zyklisch sind. Dieser Sonderfall muß beim Vergleich von Attributwert und -wertebereich berücksichtigt werden.

6.3. Netzwerkkanten (Relationen)

Eine Kante $L_i(N_1, N_2)$ der entwickelten Netzwerksprache stellt eine Beziehung zwischen zwei Knoten N_1 und N_2 des semantischen Netzes her. Anders als in den von Minsky vorgeschlagenen Frames oder der Netzwerksprache in ERNEST [Nie90, Kum93] (siehe auch Abschnitt 4.3.2) werden Kanten in der hier beschriebenen Repräsentation als Objekte realisiert (vergleiche hierzu auch Abschnitt 6.6). Die objektorientierte Repräsentation der Netzwerkkanten als Klassen erlaubt eine leichte Erweiterung der Netzwerksprache um spezielle Kantentypen. Als weiterer wesentlicher Vorteil können Daten, die sich auf die Kanten beziehen, auch dort gespeichert werden und müssen nicht in parallel verwalteten Datenstrukturen abgelegt werden. Die Knoten selbst beinhalten nur zwei Listen: eine für

alle abgehenden Relationen und eine für die eingehenden Relationen (Doppelverkettung). Im Gegensatz dazu wird in den oben erwähnten Repräsentationen für jeden Relationstyp eine Liste angelegt, in der Referenzen auf die Folgeknoten eingetragen werden.

Um die Verarbeitung des Netzes zu vereinfachen, ist es sinnvoll, die Anzahl der verschiedenen Kantentypen möglichst gering zu halten. Es werden folgende Grundtypen unterschieden:

- Die Kante *Is-A* erlaubt Spezialisierung und Vererbung von Konzepten.
- Durch *Part-Of*-Kanten werden die Teilobjekte oder -konzepte mit dem übergeordneten Knoten verbunden.
- Die *CDPart-Of*-Kanten (Context Dependant Part-Of) verbinden wie die *Part-Of*-Kanten Teile mit ihren übergeordneten Knoten. Im Gegensatz zu *Part-Of* können die Teile nur in Verbindung mit dem übergeordneten Knoten auftreten.
- Die *Instance-Of*-Kante verbindet Konzepte mit deren Instanzen.
- Die *Concrete-Of*-Kante ¹ verbindet Knoten unterschiedlicher Abstraktionsniveaus.
- Die *Data-Of*-Kante verbindet Knoten des Netzes mit einem Datenprimitiv.

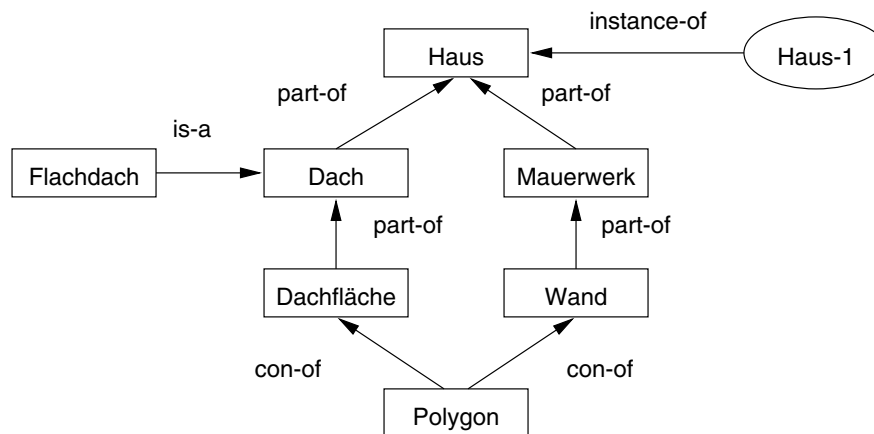


Abbildung 6.3.: Semantisches Netz

¹In den Abbildungen wird die *Concrete-Of*-Kante mit der Abkürzung *con-of* bezeichnet

Die Kanten *Is-A*, *Part-Of* und *Instance-Of* sind vielfach in der Literatur beschrieben und Bestandteil der meisten bekannten Netzwerksprachen (vergleiche auch Abschnitt 4.3.2). Die Kante *Concrete-Of* weist Ähnlichkeiten zur *Part-Of*-Kante auf, verbindet aber unterschiedliche Abstraktionsniveaus oder Systemschichten. In Abbildung 6.3 verbindet eine *Concrete-Of*-Kante das Symbol *Wand* mit seiner Konkretisierung *Polygon* auf einem geometrischen Abstraktionsniveau.

Mit Hilfe der *Concrete-Of*-Kante können sehr unterschiedliche Aufgaben gelöst werden, ohne daß eine Änderung an der Kontrolle der Analysekomponente des Systems notwendig ist. So wurden mit ERNEST Applikationen zur Interpretation von Bildern und Sprache mit unterschiedlicher Anzahl von Schichten beschrieben. Andere Systeme weisen dagegen oft ein starres Organisationsschema auf (z.B. Foresti, et al. [For93]) und können nicht ohne weiteres an Aufgaben, die eine andere Datenstrukturierung erfordern, angepaßt werden.

Die Kante *Data-Of* ist eine für die entwickelte Netzwerksprache spezifische Relation und der *Concrete-Of*-Kante verwandt. Sie bindet Datenprimitive an das semantische Netz und besitzt dazu spezifische Eigenschaften, die während der Interpretation genutzt werden. Das ist im wesentlichen die Suchfunktion (siehe Abschnitt 6.5.2) zum Matching der Segmentierungsergebnisse, die aufgabenspezifisch formuliert werden kann.

Die obengenannten Kanten sind gerichtet und weisen vom untergeordneten auf den übergeordneten Knoten ². Daneben sind für einige Anwendungen ungerichtete, bzw. bidirektionale Kanten bedeutsam:

- *ConnectedWith* gibt die räumliche Nachbarschaft zweier Objekte an und kann mit einer Liste von Attributen versehen sein.

Als Daten beinhalten alle Kantenobjekte einen Namen, eine Reihe von Statusvariablen, die während der Interpretation genutzt werden und Festlegungen über die Bindungen, die während der Interpretation hergestellt werden. Eine Bindung bezeichnet die Herstellung einer Verknüpfung eines Instanzknotens. Während in der Wissensbasis Konzepte über verschiedene Pfade zu erreichen sind (zum Beispiel das Konzept *Polygon* in Abbildung 6.3), sind Instanzen in der Regel eindeutig an einen übergeordneten Knoten gebunden. Eine Ausnahme hierzu stellen die sogenannten Mehrfachbindungen dar. Die Abbildung 6.4 stellt dazu ein Beispiel einer Mehrfachbindung dar: Der Knoten *Verbindung* soll als Instanz mit zwei Knoten vom Typ *Wand* verbunden sein. Dazu sind in den Kanten der Wissensbasis folgende Angaben möglich:

²In den Abbildungen wird eine Kante durch einen Pfeil vom unter- zum übergeordneten Knoten dargestellt.

- Die Quantität $Q(L) = (Q_{min}, Q_{max})$ gibt an, wie oft eine Kante L in der Szenenbeschreibung mindestens vorhanden sein muß und wie oft sie maximal vorhanden sein kann. Für $Q_{max} > Q_{min}$ stellt die Differenz $Q_{max} - Q_{min}$ die Anzahl optionaler Verbindungen dar.
- Die Anzahl der Bindungen $B(L) = (B_{min}, B_{max}, B_{und})$ erlaubt die mehrfache Bindung von Knoten (siehe Beispiel in Abbildung 6.4).

Die Angabe der Quantität einer Kante dient zur Festlegung von obligatorischen und/oder optionalen Bestandteilen eines Netzes.

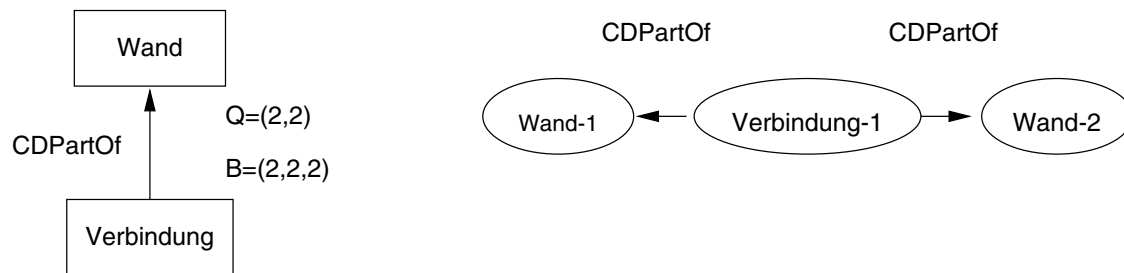


Abbildung 6.4.: Mehrfachbindung. Darstellung in der Wissensbasis (links) und in der Szenenbeschreibung (rechts)

Einen weiteren Einfluß auf die Suche in der Interpretation hat die Priorität einer Kante. Während der Interpretation werden die Kanten nach absteigender Priorität geordnet expandiert. Damit kann heuristisches Wissen formuliert werden. Ist beispielsweise ein bestimmtes Objekt besonders häufig, so ist es sinnvoll, daß die verbindende Kante die höchste Priorität erhält. Dieses Wissen kann die Interpretation später nutzen.

6.4. Netzmengen

Mit Hilfe von Netzmengen $M_i = \{N_1, N_2, \dots\}$ kann ein semantisches Netz in Teilnetze gegliedert werden. Die Teilnetze dürfen dabei überlappen, das heißt, Knoten dürfen in mehreren Teilnetzen vorhanden sein. Das ermöglicht die Darstellung konkurrierender Interpretationszustände, ohne alle Netzwerkknotten kopieren zu müssen.

Um die Konsistenz des Teilnetzes zu gewährleisten, ist eine Kante $L(N_1, N_2)$ zwischen den Knoten N_1 und N_2 nur zulässig, wenn die folgende Bedingung $\rho(L, M)$ erfüllt ist:

$$\rho(L, \mathbf{M}) = N_1 \in \mathbf{M} \wedge N_2 \in \mathbf{M} \quad (6.1)$$

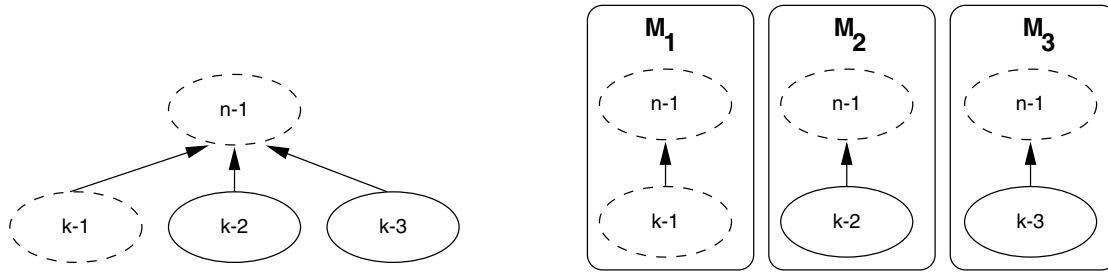


Abbildung 6.5.: Mit Hilfe von Netzmengen entstehen Teilnetze

Das heißt, eine Kante zwischen zwei Knoten ist nur gültig, wenn beide Knoten in derselben Netzmenge $\mathbf{M}$ enthalten sind. Im Speicher kann ein Knoten somit diverse Folgeknoten besitzen, die jedoch wie in Abbildung 6.5 dargestellt, nicht in allen Teilnetzen „sichtbar“ sind. Das Konzept der Netzmengen ermöglicht eine sehr effiziente Verwaltung von Hypothesen während der Interpretation.

6.5. Repräsentation von prozeduralem Wissen

Die bislang beschriebenen Teile der Netzwerksprache dienen der Repräsentation von deklarativem Wissen. Mit den vorgeschlagenen Konstrukten können mit Hilfe von Knoten und Attributen Objekteigenschaften attribuiert und durch die Netzwerkanten strukturell beschrieben werden. Die *Part-Of*- und *Concrete-Of*-Kanten beschreiben dabei Übergänge zwischen unterschiedlichen Abstraktions- bzw. Hierarchieebenen. Ein Informationsaustausch über diese Kanten hinweg, der während der Interpretation erfolgt, wird mit Hilfe von Transformationen vermittelt, die hier als prozedurales Wissen repräsentiert werden.

Die nächsten beiden Abschnitte erläutern die Nutzung von prozeduralem Wissen zur Attributberechnung. Insbesondere werden durch Attributberechnungsfunktionen die Transformationen zwischen unterschiedlichen Abstraktionsebenen des semantischen Netzes spezifiziert. Desweiteren dient prozedurales Wissen zur Beschreibung von Suchfunktionen, die die Bindung von Datenprimitiven und die Zusammenfassung von redundanten Knoten, die bei Verwendung der Mehrfachbindung entstehen, steuern.

Das Wissen zur Auswertung der szenenspezifischen Wissensbasis während der Analyse wird zweckmäßigerweise ebenfalls prozedural angegeben. Dieses Wis-

sen wird nicht im semantischen Netz abgelegt, sondern in Form von Regeln in einer der Analysekontrolle zugänglichen Liste.

6.5.1. Berechnungsfunktionen für Attributwerte und -wertebereiche

Attributberechnungsfunktionen sind eine Vorschrift zur Berechnung des Wertes $V(A)$ eines Attributes A . Den Attributberechnungsfunktionen wird eine Liste von Ausdrücken übergeben, die in Form einer Pfadgrammatik beschreiben, wie die Argumente der Funktion aus dem semantischen Netz gewonnen werden. Die Pfadgrammatik beschreibt dabei einen Pfad von dem Ausgangsattribut, beziehungsweise dem Knoten des Attributes, zu einem anderen Knoten oder einem anderen Attribut im semantischen Netz:

$$arg := \langle Pfad \rangle \quad (6.2)$$

$$Pfad := \langle Knoten \rangle [.attributname] \quad (6.3)$$

$$Knoten := \langle Kante \rangle , \langle Kante \rangle \quad (6.4)$$

$$Kante := kantename[: INV] \quad (6.5)$$

Vor dem Aufruf einer Berechnungsfunktion werden die Ausdrücke expandiert, das heißt, sie werden durch die adressierten Werte ersetzt. Die Produktion 6.5 gibt dabei den Übergang von einem Knoten zum Folgeknoten in Form einer Kante an. Ist beispielsweise der Knoten n_1 der Ausgangspunkt, so adressiert der Pfad „*part-of*“ den Knoten, der durch die (sofern eindeutig) *part-of*-Kante mit n_1 verbunden ist. Der Zusatz *:INV* kennzeichnet die Umkehrung der Übergangsrichtung. So adressiert der Pfad „*part-of,part-of.Höhe*“ ausgehend von dem Knoten *Wand* in Abbildung 6.3 das Attribut *Höhe* des Knotens *Haus* und der Pfad „*con-of:INV.Fläche*“ das Attribut *Fläche* des Knotens *Polygon*. Für den allgemeinen Fall ist zu beachten, daß ein Pfad nicht eindeutig sein muß. In diesem Fall wird eine Liste von Werten erzeugt.

In aller Regel werden einer Attributberechnungsfunktion Attributwerte von Attributen übergeben, die dichter am Signal liegen. Das heißt, die Argumentdaten stammen aus untergeordneten Knoten des jeweiligen Attributknotens. So könnte beispielsweise ein Attribut *Höhe* des Knotens *Haus* aus den Attributwerten der Teile dieses Knotens, also den Instanzen des Konzeptes *Mauerwerk* und *Dach* berechnet werden. Der Informationsfluß ist somit Bottom-Up.

Umgekehrt dienen Berechnungsfunktionen für den Attributwertebereich $\epsilon(A)$ dazu, einen Top-Down Informationsfluß herzustellen: Werden während der In-

terpretation hypothetische Instanzen erzeugt, so werden für die Attribute dieser Knoten die Berechnungsfunktionen für den Attributwertebereich aufgerufen. Die Argumente stammen von den übergeordneten Knoten. So wird beispielsweise für eine hypothetische Instanz des Konzeptes *Mauerwerk* der Wertebereich für das Attribut *Höhe* aus der dem Attribut *Haus.Höhe* berechnet. Diese Flußrichtung dient damit der Constraint-Propagierung.

Die Berechnungsfunktionen für den Attributwert wie auch für den Wertebereich können, wie in Abschnitt 6.2 beschrieben, einen minimalen und einen maximalen Wert berechnen. Wird nur ein Wert zurückgeliefert, so gilt $V_{max} = V_{min}$, bzw. $\epsilon_{max} = \epsilon_{min}$.

6.5.2. Suchfunktionen

Suchfunktionen dienen während der Interpretation dazu, eine bereits vorhandene Instanz zu finden. Dieser Fall tritt in der Regel ein, wenn für einen Knoten $I_j^H(C)$ des Netzes ein Datenknoten gesucht wird. Die Datenknoten werden zu Beginn der Interpretation beispielsweise aus dem Segmentierungsergebnis in das Netz geladen. Stößt die Interpretation auf einen Knoten $I_j^H(C)$ mit einer nicht gebundenen *data-of*-Kante, so ruft sie für diesen Knoten eine Datensuchfunktion auf. Diese kann wie die Attributberechnungsfunktionen eine Liste von Argumenten haben. Als Ergebnis liefert die Funktion eine Liste mit passenden Datenknoten. Die Kontrolle der Interpretation muß für den Fall, daß die Ergebnisliste mehr als einen Datenknoten enthält, diese jeweils als konkurrierende Hypothesen behandeln (Siehe Abschnitt 7.2).

Ebenfalls Verwendung finden Suchfunktionen bei der Verarbeitung von Mehrfachbindungen. Dieser Fall wird in Abschnitt 7.2.1 behandelt.

Die Suchfunktionen, wie auch die nachfolgend beschriebenen benutzerdefinierten Funktionen, werden an Knoten im semantischen Netz entlang der *is-a*-Kanten vererbt und können gegebenenfalls überschrieben werden.

6.5.3. Benutzerdefinierte Funktionen

Den Knoten des semantischen Netzes können benutzerdefinierte Funktionen zugewiesen werden, zum Beispiel eine spezielle Visualisierungsmethode, die von der grafischen Benutzeroberfläche genutzt wird.

Eine spezielle benutzerdefinierte Funktion wird für die Oberflächenrekonstruktion genutzt, um neben den im semantischen Netz vorhandenen Daten zusätzliche

Randbedingungen auszuwählen, die zur Erzeugung der Oberflächenbeschreibung nützlich sind (siehe Abschnitt 8.2.2).

6.5.4. Regeln für die Wissensverarbeitung

Die bisher vorgestellten Repräsentationen dienen der Formulierung von szenenspezifischem Wissen. Dies wird überwiegend deklarativ und im Falle der oben beschriebenen Transformationen zwischen Abstraktionsebenen prozedural angegeben. Neben dem szenenspezifischen Vorwissen, das in dem semantischen Netz angegeben wird, können Fakten über Vorgehen und Strategie für die Auswertung des szenenspezifischen Vorwissens formuliert werden. Hierzu zählt auch heuristisches Wissen. Als Repräsentation eignen sich Regeln, die aus einem Bedingungsteil und einem Aktionsteil bestehen. Strategie und Heuristik werden durch Angabe einer Priorität beschrieben.

Die Regeln zur Auswertung sind nicht Bestandteil der Netzwerksprache im engeren Sinne, sondern werden eher der Analyse zugeordnet. Eine genauere Beschreibung über Art und Anwendung befindet sich daher in Abschnitt 7.2.1 des Kapitels zur Interpretation.

6.6. Implementierungsaspekte

Die Entwicklung einer geeigneten Repräsentation für die Daten und die Systemimplementierung sind entscheidend für die Systemperformance. Ferner entscheiden sie, wie effizient ein Problem formuliert werden kann, was sich vor allem in der Dauer der Entwicklungszeit niederschlägt. Der technische Fortschritt im System-Engineering ist unter anderem darin zu sehen, daß ein gegebenes Problem schneller und effizienter formuliert werden kann. Dies wird durch an das Problem angepaßte Beschreibungsformen und Programmierumgebungen erreicht. Neben dem reinen Sprachumfang einer Netzwerksprache spielt hier auch die Implementierung des Gesamtsystems eine entscheidende Rolle.

Als erstes stellt sich hierbei die Frage nach einer geeigneten Programmiersprache. Während für die numerisch sehr rechenintensiven bild- und geometriedatenverarbeitenden Algorithmen meist die maschinennahe Programmiersprache „C“ [Kern78] eingesetzt wird, werden für die Verarbeitung auf der symbolischen Ebene sogenannte höhere Programmiersprachen, wie beispielsweise LISP [Ste90] eingesetzt. Es gibt eine ganze Reihe von Beispielen für die Implementierung wissensbasierter Systeme in LISP, bzw. der objektorientierten LISP-Erweiterung CLOS

(zum Beispiel [SNePS, Rost98]). Die wichtigsten Eigenschaften, die für den Einsatz von LISP-basierten Systemen sprechen, sind die mächtigen Möglichkeiten der symbolischen Datenverarbeitung und während der Entwicklungsphase die Tatsache, daß in LISP die Programme interpretiert werden. Das ermöglicht zum Beispiel die Abfrage von Variablen zur Laufzeit. Ferner können einzelne Programmteile manuell in beliebiger Reihenfolge gestartet werden.

Demgegenüber steht jedoch der Nachteil von LISP-basierten Systemen, daß diese nur unbefriedigend mit C-Programmteilen kombiniert werden können ³. Ferner sind LISP-Programme sehr speicherintensiv und langsam in der Ausführungszeit. Erste Implementierungen des Analysesystems [Gro94, Grau95, Lie95] ergaben, daß LISP sich zwar als Prototyping-Plattform eignet, aber keine effiziente Systemimplementierung zuläßt.

Das in dieser Arbeit beschriebene System wurde daher objektorientiert unter Verwendung der Programmiersprache C++ [Stro92] und Anbindung an Tcl/Tk [Oust94] implementiert. Die Sprache C++ ist eine Erweiterung der Sprache C um objektorientierte Ansätze. Programme können damit sehr effizient in Bezug auf Rechenzeit und Speicherbedarf implementiert werden. Das Toolkit Tcl/Tk ist eine Kombination aus Interpreter und X11-Fensteroberfläche und sieht eine Schnittstelle zu C- oder C++-Funktionen vor. Durch Verwendung des Paketes Objectify [Chri95] können alle Datenobjekte sowohl von C++, als auch vom Tcl/Tk-Interpreter bearbeitet werden.

Dies ermöglicht zum einen die Nutzung der hohen Verarbeitungsgeschwindigkeit von C++, zum anderen bietet der Interpreter die Vorzüge einer komfortablen Experimentalumgebung. So wurde eine grafische Entwicklungsumgebung für das System (siehe Abschnitt 7.7) mit Hilfe von Tcl/Tk realisiert. Darüber hinaus ist es möglich, die in Abschnitt 6.5 beschriebenen Berechnungs- und Suchfunktionen als Tcl-Skripte zu spezifizieren. Diese können damit zur Laufzeit anhand einer realen Analysesituation entwickelt und getestet werden. Nach Erstellung der Funktion kann diese, wenn sie zeitkritisch ist, in einer speziellen C++-Funktion implementiert werden. Das System ist damit in Hinsicht auf die Entwicklungsfreundlichkeit und Verarbeitungsgeschwindigkeit skalierbar.

³Der Grund hierfür liegt vor allem in dem unterschiedlichen Speicherverwaltungskonzept von LISP und C/C++

6.7. Ein semantisches Netz für die Analyse von Gebäudeszenen

Dieser Abschnitt beschreibt das für die Anwendung der Analyse von Gebäudeszenen in der zuvor beschriebenen Netzwerksprache entwickelte semantische Netz. In diesem Netz spiegeln sich alle für die Aufgabe benötigten Daten und deren Beziehungen zueinander in Form einer strukturierten generischen Modellbeschreibung wieder. Ebenso ist die Anbindung an das Signal berücksichtigt.

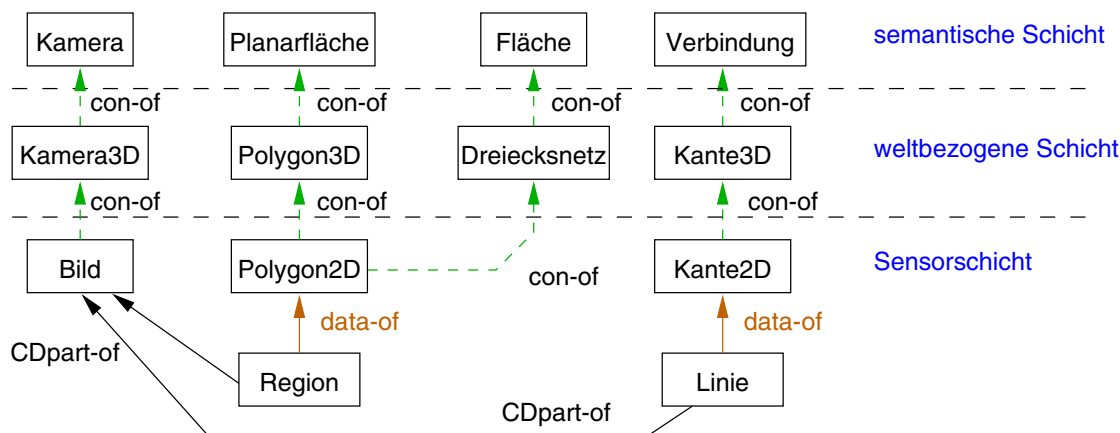


Abbildung 6.6.: Für die Analyse bedeutsame Daten sind als Knoten im semantischen Netz repräsentiert

Die Abbildung 6.6 zeigt einen Ausschnitt der erstellten Wissensbasis mit den Konzepten, die die für die Analyse bedeutsamen Daten repräsentieren. Das Netz ist in drei Schichten organisiert:

Die untere Schicht ist die Sensorschicht. Dementsprechend finden sich hier Knoten, die sich auf den Sensor beziehen. Das ist das Konzept *Bild* und die Bildprimitive *Polygon2D* und *Kante2D*. Das Konzept *Bild* ist ein spezieller Knotentyp, der als Datencontainer verschiedene Bilddaten beinhaltet: Neben dem Farbbild werden hier die Tiefenkarte, das Segmentierungsergebnis und die extrahierten Liniensegmente gespeichert. Innerhalb dieser Schicht liegen auch die Datenknoten *Region* und *Linie*. Während der Initialisierung vor der Interpretation werden Instanzen dieser Konzepte aus den Segmentierungsdaten erzeugt. So wird für jeden Label-Wert des Segmentierungsergebnisses (Abschnitt 5.3) ein Instanzenknoten vom Typ *Region* erzeugt und in das Netz eingefügt.

Die mittlere Schicht ist die weltbezogene Schicht, in der Objekte dreidimensional raumbezogen beschrieben sind. In der Abbildung 6.6 sind das die Konzepte *Po-*

lygon3D und *Dreiecksnetz* als dreidimensionale Flächen, *Kante3D* als Pendant zu *Kante2D* und *Kamera3D*, das die Kameraparameter zum Konzept *Bild* beinhaltet.

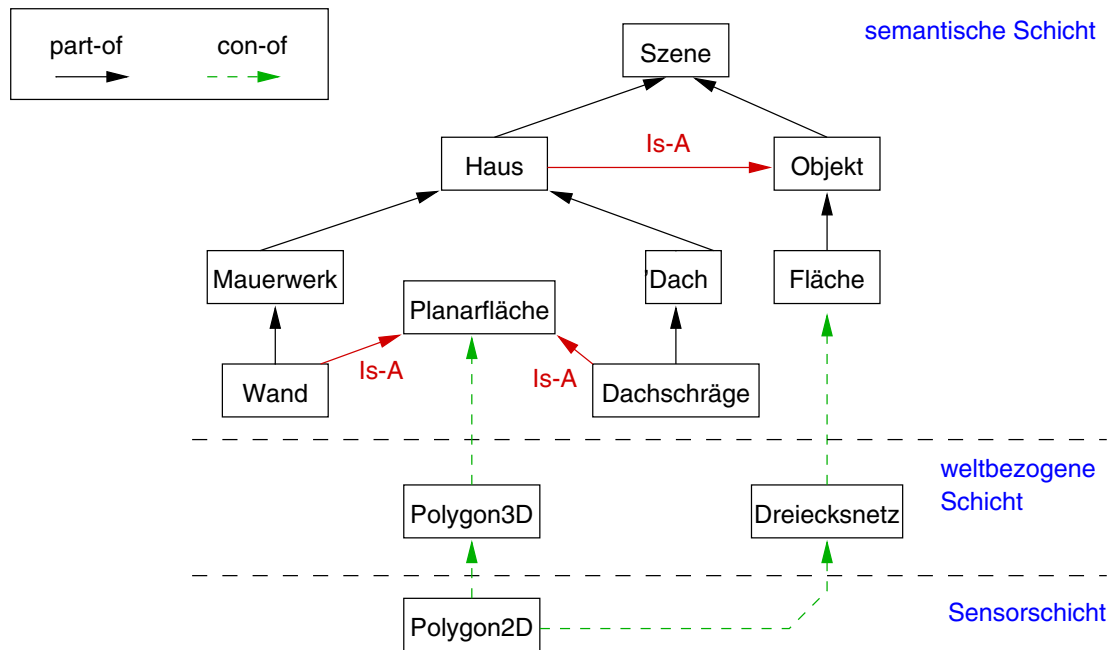


Abbildung 6.7.: Szenenspezifische Eigenschaften in der semantischen Schicht

Die obere, semantische Schicht beschreibt Objekte auf einer abstrakten Ebene. Hier werden spezielle Objekteigenschaften und Beziehungen der Objekte zueinander dargestellt. Das Konzept *Kamera* und dessen Konkretisierungen (*Kamera3D*, *Bild*) stellen den Sensor für die Analyse dar. Eine Szenenbeschreibung kann mehrere Instanzen dieser Konzepte enthalten, wenn entsprechend viele Eingangsbilder verarbeitet werden. Die Konzepte *Planarfläche* und *Fläche* stellen die zur Verfügung stehenden Flächenbeschreibungen dar: 3D-Polygon und 3D-Dreiecksnetz. Das Konzept *Verbindung* repräsentiert die Verbindung zwischen zwei Polygonflächen. Diese wird, wie bereits in Abschnitt 6.3, Abbildung 6.4 gezeigt, als Mehrfachbindung zwischen Polygonflächeninstanzen, zum Beispiel Hauswände, gebunden.

Die Abbildung 6.7 zeigt die wichtigsten Konzepte der Wissensbasis in einem Ausschnitt des Netzes. Das Konzept *Szene* ist der oberste Knoten in der Teile-Hierarchie. Eine „Szene“ kann nach diesem Szenenmodell die Knoten *Haus* und *Objekt* enthalten. Ein Knoten vom Typ *Objekt* hat als Komponente einen (oder mehrere) Knoten vom Typ *Fläche*, der als Dreiecksnetz konkretisiert ist. Das Konzept *Haus* besteht aus den Komponenten *Mauerwerk* und *Dach*, die jeweils die Teile *Wand*, beziehungsweise *Dachschräge* besitzen, die als 3D-Polygone konkreti-

siert sind. Das Konzept *Haus* ist spezieller als das Konzept *Objekt* und demzufolge durch eine *Is-A* Kante mit letzterem verbunden.

Durch Einfügen oder Weglassen der *Part-Of*-Kante, die das Konzept *Haus* als Teil der Szene beschreibt, wird die Abarbeitungsreihenfolge während der Interpretation beeinflusst: Durch die *Is-A*-Kante wird diese Kante ohnehin vom Konzept *Objekt* vererbt. Ohne die zusätzliche *Part-Of*-Kante würde die Interpretation jeweils zwei konkurrierende Hypothesen für *Haus* und *Objekt* erzeugen und anhand des Signals verifizieren. Ist bekannt, daß genau ein Haus in der Szene ist, kann dies genutzt werden, um zielgerichteter danach zu suchen. In diesem Falle muß die *Part-Of*-Kante wie in Abbildung 6.7 gezeigt zwischen den Konzepten *Haus* und *Szene* als obligat eingetragen werden. Die Interpretation versucht darauf dieses als erstes zu finden.

Die Abbildung 6.8 zeigt eine Übersicht über das gesamte entwickelte semantische Netz. Die Konzepte zur Beschreibung von Kanten, beziehungsweise Verbindungen zwischen den Polygonen sind der Übersicht halber nicht eingezeichnet. Neben weiteren spezifischen Komponenten des „Hauses“ ist noch das Konzept *Boden* dazugekommen. Dieses dient als Referenzfläche, um beispielsweise eine Aussage über die senkrechte Orientierung von Teilflächen, wie Hauswänden treffen zu können.

Die Konzepte enthalten entsprechende Attribute, die die Objekteigenschaften beschreiben. So besitzt das Konzept *Haus* beispielsweise die Attribute *Haus.Höhe*⁴ und *Haus.Breite*. Die Attribute der Wissensbasis enthalten dabei, wie in Abschnitt 6.2 erläutert, Werte in Form von Wertebereichen. Das sind Intervalle, in denen sich die gemessenen Attribute der instanziierten Objektbeschreibung üblicherweise befinden. Die Wertebereiche werden zur Interpretation verwendet und stellen Wissen über das Objekt in intensioneller Form dar.

6.8. Diskussion der entwickelten Repräsentation

Die in diesem Abschnitt beschriebene Netzwerksprache erlaubt die Darstellung des entwickelten strukturierten generischen Szenenmodells. Das Vorwissen über die Szene und deren Komponenten kann in einer expliziten Repräsentation formuliert werden. Die Wissensinhalte werden zur Interpretation wie auch für die nachfolgende Rekonstruktion genutzt.

Die entwickelte Netzwerksprache vereint mehrere Darstellungsformen, mit denen die Vielfalt der Wissensinhalte verkörpert werden können. So kann das struk-

⁴Zur Schreibweise: *Haus.Höhe* bezeichnet das Attribut *Höhe* des Knotens *Haus*

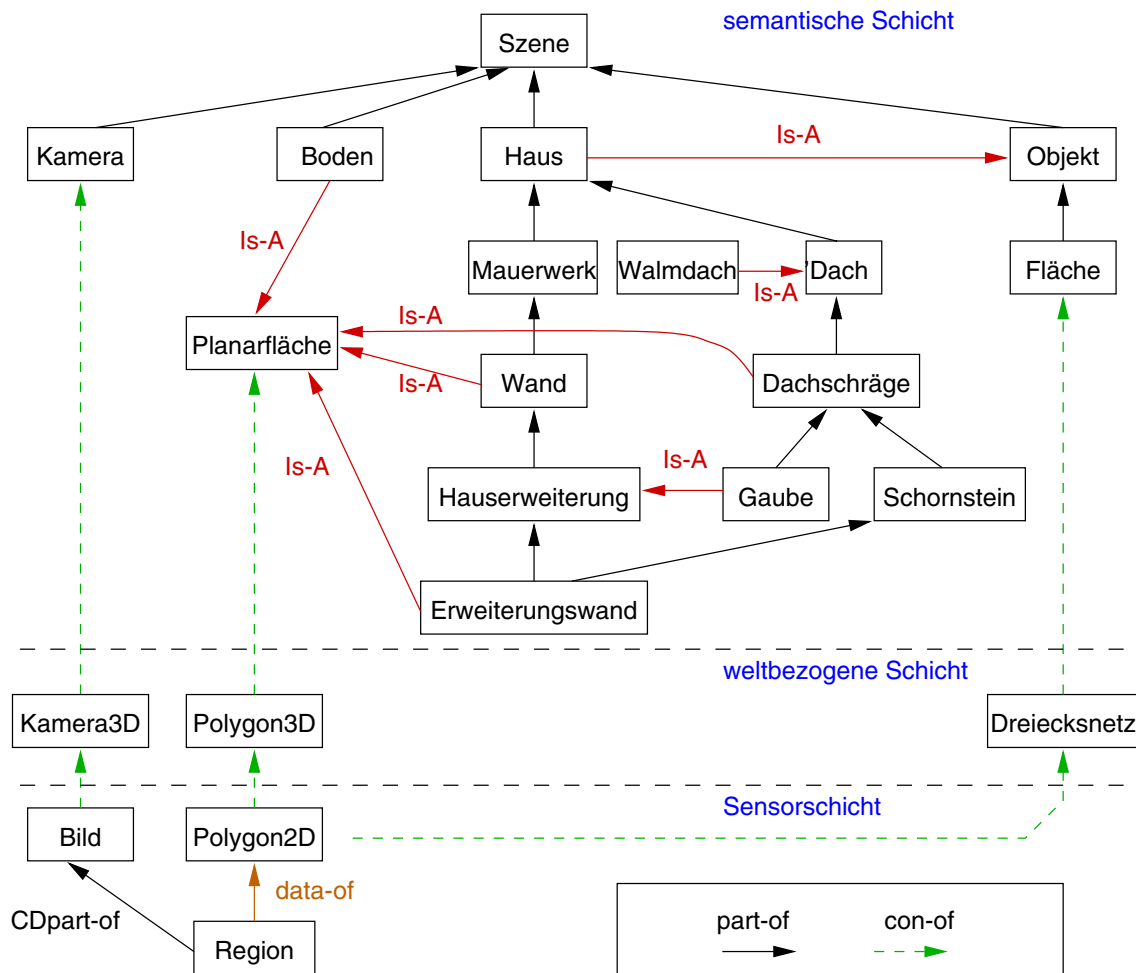


Abbildung 6.8.: Semantisches Netz für die Analyse von Gebäudeszenen (Ausschnitt)

turierte generische Szenenmodell direkt deklarativ umgesetzt werden. Durch Konzeptbildung und Angabe von Relationen mit Hilfe der Netzwerkkanten kann die Szene kompositionell zerlegt und in ihrer Struktur beschrieben werden. Die geometrischen, photometrischen und sonstigen Objekteigenschaften werden durch Attribute angegeben. Die intensionelle Repräsentation führt dabei zu einer kompakten Darstellung der Wissensinhalte.

Die deklarative Szenenbeschreibung unterstützt die Gliederung der Wissensbasis in Abstraktionsebenen, die durch die spezielle *concrete-of*-Kante vorgenommen wird. Die Übergänge zwischen Teilen der Wissensbasis unterschiedlichen Abstraktionsniveaus, die insbesondere während der Interpretation dem Datenfluß dienen, werden prozedural durch Transformationsfunktionen beschrieben.

Ferner ist die Formulierung von heuristischem Wissen, beispielsweise als Prioritäten von Kanten (siehe Abschnitt 6.3) oder für die Auswertungsregeln möglich, wodurch die Effizienz der Analyse gesteigert wird.

Der strukturellen Beschreibung kommt eine mehrfache Bedeutung zu: Zum einen ist die Interpretation von Objektkomponenten in natürlichen Umgebungen allein aufgrund ihrer Attribute meist nicht möglich. Eine sichere Interpretation ist nur durch Ansätze der strukturellen Mustererkennung möglich. Die Wissensrepräsentation trägt daher diesem Gesichtspunkt verstärkt Rechnung. Die Darstellung der Abstraktionsebenen durch die *concrete-of*-Kante führt zudem zu einer konsequenten Trennung zwischen dem strukturierten generischen Szenenmodell und der Systemkontrolle. Des weiteren sind es die strukturellen Eigenschaften und Relationen, die für die Oberflächenrekonstruktion eine wichtige Rolle spielen.

Der Entwurf und die Implementierung der Bestandteile der Netzwerksprache erfolgte durchgängig objektorientiert und erlaubt so ein komfortables Entwickeln sowohl von Wissensbasen als auch von Systemerweiterungen. Dies wird unterstützt durch das Konzept der alternativen Verwendung von (Tcl-)Interpreter oder (C++-)Compiler für die Implementierung von prozeduralen Wissensinhalten. Die Ausführung der Netzwerkkanten als Objekte⁵ ermöglicht eine effiziente Haltung von kantenbezogenen Daten, wie beispielsweise Quantitäten oder temporäre Zustandsvariablen, die während der Interpretation genutzt werden.

⁵Im Sinne der objektorientierten Programmierung

7. Interpretation des Szeneninhaltes

Die Interpretation stellt einen Bezug her zwischen den Daten aus der Bildverarbeitung und dem strukturierten generischen Szenenmodell, das als Wissensbasis vorliegt. Die dadurch gewonnene objektspezifische Szenenbeschreibung wird genutzt, um Randbedingungen für die nachfolgende Gewinnung der Oberflächenbeschreibung zu erzeugen.

Die Interpretation realer Szenen aus Bildern oder Bildfolgen stellt ein schwieriges Problem dar. Zum einen ist - gerade bei natürlichen Szenen - mit einer großen Anzahl von verschiedenartigen und vielfältigen Objekten zu rechnen. Zum anderen erschwert, wie bei der Oberflächenrekonstruktion, der Informationsverlust während der Abbildung die Lösung des Problems.

7.1. Konzept der entwickelten Interpretationskomponente

Das Ziel der Interpretation ist es, ausgehend von der Wissensbasis in Form eines semantischen Netzes und den Daten aus der Bildverarbeitung eine Bedeutungszuweisung des Szeneninhaltes vorzunehmen. Die resultierende Szenenbeschreibung liegt ebenfalls in Form eines semantischen Netzes vor. Sie enthält neben den Instanzen, die einen Bezug zu den Daten haben, auch die Konzepte der ursprünglichen Wissensbasis.

Zu Beginn der Interpretation werden neben der Wissensbasis eine initiale Szenenbeschreibung und die Daten aus der Bildverarbeitung in das semantische Netz geladen und bilden zusammen den Interpretationszustand S_0 (vergleiche Abschnitt 6.7). Unter Anwendung einer Reihe von Transformationen, die sukzessiv ein Instanzennetz aufbauen, entsteht daraus nach N Schritten die resultierende Szenenbeschreibung als Interpretationszustand S_N , wie in Abbildung 7.1 dargestellt.

Die Transformationen werden explizit als Regeln formuliert. Dieses Konzept wird in Abschnitt 7.2.1 kurz skizziert. Beim Auftreten von Mehrdeutigkeiten

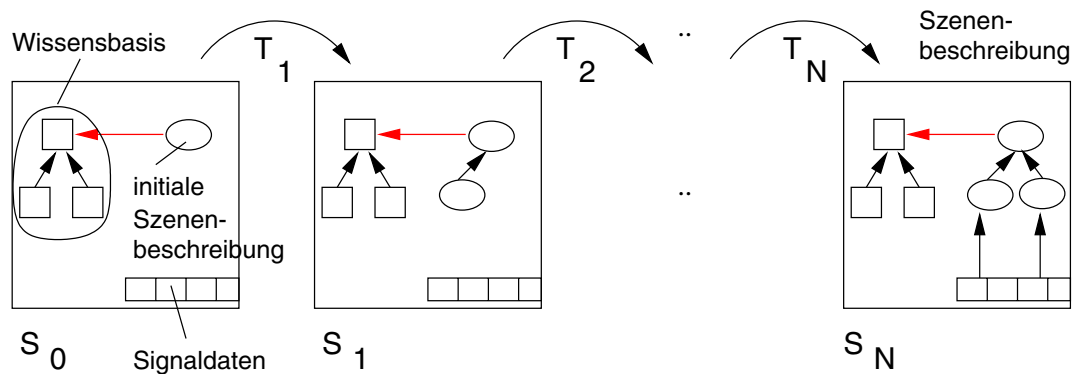


Abbildung 7.1.: Die Szenenbeschreibung entsteht durch wiederholtes Anwenden von Transformationen T_i auf das semantische Netz

kann die Szenenbeschreibung nicht mehr durch eine lineare Kette von Transformationen erreicht werden. Statt dessen spaltet sich die Kette in einem Suchbaum auf. Die konkurrierenden Szenenbeschreibungen werden bewertet und ein Suchalgorithmus sucht daraufhin die Interpretation mit der besten Bewertung. Die Abarbeitungsreihenfolge der Regeln und die Behandlung von Mehrdeutigkeiten ist in einem problemunabhängigen Kontrollalgorithmus festgelegt, der im folgenden Abschnitt beschrieben wird.

Der problemunabhängige Kontrollalgorithmus bildet zusammen mit der in Abschnitt 6 beschriebenen Netzwerksprache und der in Abschnitt 7.7 kurz skizzierten graphischen Oberfläche den Kern des an der Universität Hannover entwickelten Rahmensystems AIDA zur wissensbasierten Analyse von Szenen [Bueck98].

Neben dem problemunabhängigen Kontrollalgorithmus benutzt die hier entwickelte Interpretation weitere Prinzipien zur Einschränkung der Suche nach der richtigen Szenenbeschreibung. Dies sind die Constraint-Propagierung, Matching von Segmentierungsdaten und die Bewertung konkurrierender Deutungen. Diese Themen sind anwendungsspezifisch und werden im Einzelnen in den weiteren Abschnitten dieses Kapitels beschrieben.

7.2. Ein problemunabhängiger Kontrollalgorithmus

Die meisten Ansätze zur Szeneninterpretation verwenden anwendungsspezifische Regeln, wie in den Produktionssystemen SPAM [McKe85], den Arbeiten von

Henderson, et al. [Hend88] oder BPI [Stil95]. Das szenenspezifische Wissen ist damit sehr eng mit der Kontrollstrategie verbunden. Kummert [Kum91] beschreibt einen festen Satz von problemunabhängigen Regeln zur Transformation eines semantischen Netzes, das das szenenspezifische Wissen beinhaltet, in Kombination mit einem Suchalgorithmus. Dieses Konzept wird hier übernommen. Statt einer festen impliziten Implementierung der Regeln als Programm werden in AIDA die Transformationsregeln jedoch explizit repräsentiert. Dadurch ist es möglich, sehr flexibel unterschiedliche Ausführungsreihenfolgen der Transformationsregeln zu erreichen, was unterschiedliche, dem jeweiligen Problem besser angepaßte Strategien erlaubt.

Im folgenden wird der in AIDA implementierte Kontrollalgorithmus und das Konzept der expliziten Transformationsregeln kurz vorgestellt. Eine weiterführende Beschreibung gibt [Toen99].

7.2.1. Formulierung und Anwendung der Netzwerktransformationen als Inferenzregeln

Die zur Interpretation auf das semantische Netz angewandten Transformationen bewirken die Erzeugung von Instanzen, die Propagierung von Hypothesen, Spezialisierung und Bindung von Instanzen. Eine flexible Darstellung ist die durch Regeln. Diese bestehen aus einem Bedingungsteil und einem Aktionsteil. Der Bedingungsteil prüft, ob die Regel für einen Knoten des Interpretationszustandes zutrifft. Ist das der Fall, so wird der Aktionsteil ausgeführt. Die Überprüfung des Bedingungsteils erfolgt lokal für die Netzwerkknotten, das heißt, nur der Status eines Knotens und seiner direkten Nachbarn gehen in die Überprüfung ein. Eine typische Regel ist die Regel $R_{H:inv}$ zum Erzeugen einer modellgetriebenen Hypothese:

Wenn	für zwei Konzepte A, B gilt: $A \xleftarrow{Part-Of} B$ oder $A \xleftarrow{Con-Of} B$ und dem Knoten $I(A)$ ein obligatorischer Bestandteil oder eine Konkretisierung fehlt
Dann	generiere eine hypothetische Instanz $I^H(B)$, verbinde $I^H(B)$ mit $I(A)$, rufe für alle Attribute deren Bereichsberechnungsfunktionen auf

Tabelle 7.1 stellt die übrigen verwendeten Regeln zusammen. In der Spalte *Aktion* sind die wesentlichen Manipulationen angegeben. Neben der beschriebenen Regel zur modellgetriebenen Erzeugung einer hypothetischen Instanz $R_{H:inv}$ gibt es jeweils eine Regel zur datengetriebenen Hypothese R_H und zur Erzeugung von Hypothesen optionaler Teile $R_{H:invopt}$.

Kürzel	Bezeichnung der Regel	Aktion
R_P	partielle Instanzierung	$I^H(N) \rightarrow I^P(N), AB \uparrow, B$
R_K	vollständige Instanzierung	$I^P(N) \rightarrow I^K(N), AB \uparrow, B$
$R_{H:dat}$	Datenbindung	$\rightarrow I^K(C), AB \uparrow, B$ oder $\rightarrow I^F(N)$
R_F	Propagiere Status Fehlinstanz	$N \rightarrow I^F(N)$
$R_{H:inv}$	Erzeuge modellgetriebene Hypothese	$\rightarrow I^H(C), AB \downarrow$
R_H	Erzeuge datengetriebene Hypothese	$\rightarrow I^H(C), AB \uparrow$
$R_{H:invopt}$	Erzeuge optionale Hypothese	$\rightarrow I^H(C), AB \downarrow$
R_{Spez}	Erzeuge Spezialisierung	$\rightarrow I^H(C), AB \downarrow$
R_{MB}	Mehrfachbindung	$I_1^H(C) + I_2^H(C) \rightarrow I^P(C)$ $, AB \uparrow, B$

Tabelle 7.1.: Transformationsregeln. ($AB \uparrow$ = Attributberechnung ausführen, $AB \downarrow$ = Attributbereichsberechnung ausführen, B = Bewertungsfunktionen ausführen)

Die ersten vier Regeln in Tabelle 7.1 manipulieren einen Knoten vom Status *hypothetische Instanz* über die *partielle Instanz* bis zur *vollständigen Instanz* oder zur *Fehlinstanz*, falls kein Segmentierungsergebnis gefunden wurde. Neben der Änderung des Status werden für die Attribute die Berechnungsfunktionen aufgerufen und die Knotenbewertung, die ihrerseits die Bewertungsfunktionen für alle Attribute des Knotens ausführt. Mit Hilfe der Knotenbewertung wird die Gesamtbewertung für den aktuellen Interpretationszustand berechnet, die von dem im Abschnitt 7.2.3 beschriebenen Suchalgorithmus ausgewertet wird.

Die Auswertung der Transformationsregeln erfolgt in einer Inferenzmaschine, die eine Liste R_{Trans} aller Regeln enthält. Zur Auswertung besitzen die Regeln eine Priorität $\delta_{Prio}(R)$ als freien Parameter. Dieser beeinflusst die Reihenfolge der Regelanwendung und legt die Strategie der Analyse fest. Soll beispielsweise die datengetriebene Erzeugung von Hypothesen der modellgetriebenen bevorzugt werden, so wird $\delta_{Prio}(R_H) > \delta_{Prio}(R_{H:inv})$ gewählt.

Die Inferenzmaschine arbeitet die Regeln mit hoher Priorität bevorzugt ab. Es kann allerdings sein, daß mehrere Regeln, für die der Bedingungsteil erfüllt ist, dieselbe Priorität besitzen. Für diesen Fall hat Sagerer [Sag90] ein problemunabhängiges Prioritätsmaß $\delta(n)$ eingeführt. Dieses berücksichtigt die Distanz des Knotens n vom Signal, die Abstraktionsstufe, den Verifikationsaufwand, die Generalisierungsstufe und die Kompositionsstufe. Die einzelnen Wertigkeiten werden für die Konzepte der Wissensbasis berechnet und dienen gleichzeitig als Konsistenzüberprüfung. Die Einbeziehung des zusammengesetzten Prioritätsmaßes hat sich zur Steuerung der Ausführung der Regeln als vorteilhaft erwiesen [Gro94].

Instanzen sind normalerweise mit genau einer Kante an ihren übergeordneten Knoten gebunden. Es gibt jedoch Anwendungen, wie in Abbildung 6.4 gezeigt, bei denen die Bindung einer Instanz an mehrere übergeordnete Knoten gewünscht ist. Dieser Fall wird von der Regel R_{MB} behandelt. Dabei werden für die übergeordneten Knoten zuerst unabhängig *hypothetische Instanzen* erzeugt. Anschließend ruft der Aktionsteil von Regel R_{MB} eine Suchfunktion auf, die bei Erfolg eine Instanz oder eine Liste von Instanzen zurückliefert. Daraufhin wird der die Regel aktivierende Knoten mit dem gefundenen fusioniert und der resultierende Knoten erhält die Kanten der Ausgangsknoten.

Die Ausführung der Inferenzregeln erfolgt nach folgendem Algorithmus:

Funktion $Inferenz(S, \mathbf{R}_{Trans})$	
erzeuge sortierte Liste N_{sort} aller Knoten aus S nach absteigender Wertigkeit $\delta(n)$	
Liste konkurrierender Knoten $N_{konkurr} = \{nil\}$	
Für alle Regeln R_i aus $\mathbf{R}_{Trans}$ nach absteigender Priorität sortiert	
Für alle Knoten N_i aus N_{sort}	
Prüfe, ob Bedingungsteil von Regel R_i für N_i in S erfüllt ist	
Falls ja	führe Aktionsteil von Regel R_i aus
	kopiere dabei entstehende konkurrierende Knoten in $N_{konkurr}$
	liefere $N_{konkurr}$ und Status „erfolgreich“ als Funktionswerte zurück
terminiere Funktion und liefere $\{nil\}$ und Status „nicht erfolgreich“ zurück	

7.2.2. Behandlung von Mehrdeutigkeiten

Die Regel R_{Spez} erzeugt bei Vorhandensein einer *IsA*-Kante in der Wissensbasis eine Spezialisierung, die als konkurrierende Hypothese behandelt wird. Neben der Spezialisierung sind auch die Regeln zur initialen Instanzierung $R_{H:dat}$ und für Mehrfachbindung R_{MB} in der Lage, konkurrierende Hypothesen zu erzeugen. Tritt dieser Fall ein, so dokumentiert die Inferenzmaschine dies durch Rückgabe einer Liste von neu erzeugten Knoten, die jeweils eine konkurrierende Hypothese darstellen (normalerweise befindet sich kein oder nur ein Element in der Rückgabeliste).

Der übergeordnete Kontrollalgorithmus behandelt den Fall des Auftretens von Mehrdeutigkeiten durch Aufspaltung des Interpretationszustandes: Jede konkurrierende Hypothese wird als neuer Folgeknoten des aktuellen Suchbaumknotens in einem Suchgraphen eingetragen, dabei entsteht ein Suchbaum.

Jeder Knoten des Suchbaumes enthält einen kompletten Interpretationszustand. Die Inferenzmaschine zur Interpretation dokumentiert alle Änderungen am In-

terpretationszustand in dem aktuellen Suchbaumknoten S_i . Spaltet sich der Suchbaum auf, so müssen die neuen Suchbaumknoten den Inhalt des ursprünglichen Knotens und die neu entstandenen Knoten enthalten.

Anstatt den Interpretationszustand komplett mit allen Knoten, deren Attributen und den Relationen zu kopieren, werden die Nachfolgezustände durch Eintragen von Referenzen auf die Knoten und durch Speichern von inkrementellen Änderungen realisiert. Da im Laufe der Interpretation eine Reihe von Suchbaumknoten entstehen, bietet diese Art der Dokumentation eine enorme Speicher- und Zeitersparnis bei der Verarbeitung.

Zur inkrementellen Speicherung des veränderten Interpretationszustandes werden alle unveränderten Knoten als Referenz in die neue Menge S_i übernommen und modifizierte Knoten durch ein neues Objekt ersetzt. Ein Knoten muß somit ersetzt werden, sobald beispielsweise ein Attribut einen neuen Wert erhält.

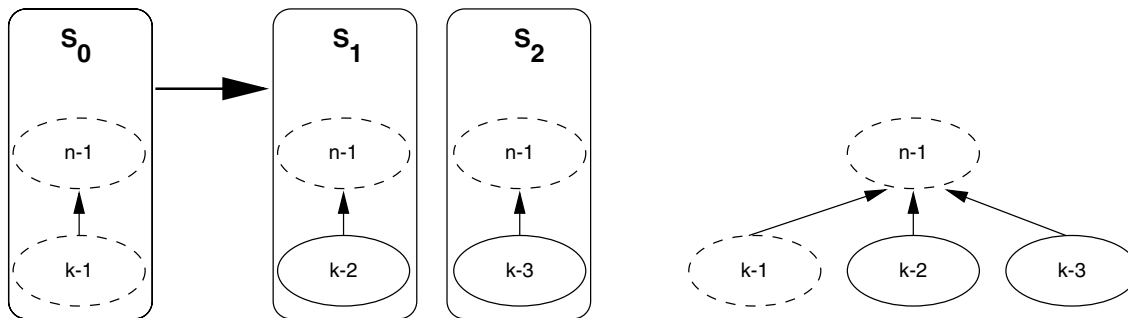


Abbildung 7.2.: Durch inkrementelle Modifikation des Netzes entstehen Knoten mit mehreren Kanten.

Die Verwendung der in Abschnitt 6.4 eingeführten Netzmengen für die Suchbaumknoten, erlaubt die Mehrfachbenutzung von Knoten, wie in Abbildung 7.2 dargestellt. Durch Erzeugung konkurrierender Interpretationszustände können Knoten, wie der Knoten $n-1$ (Abbildung 7.2 rechts) eine Reihe von Kanten zu verschiedenen Knoten erhalten, die zu unterschiedlichen Hypothesen gehören. Die Bedingung 6.1, daß beide Knoten einer binären Relation in derselben Netzmenge enthalten sein müssen, erlaubt eine eindeutige Trennung.

7.2.3. Suchalgorithmus

Würden die in Abschnitt 7.2.1 beschriebenen Transformationen rekursiv auf alle entstehenden Suchbaumknoten angewendet, bis keine Regel mehr feuert, das

heißt, kein Aktionsteil mehr erfüllt ist, so würden alle möglichen Interpretationszustände erreicht werden. Die „beste“ Interpretation wäre dann ein Endknoten des Suchbaums, der die beste Bewertung aufweist. Dieses Vorgehen wäre eine vollständige Suche, die aufgrund der großen Menge an möglichen Interpretationszuständen bei praktischen Anwendungen nicht bewältigt werden kann. In der Literatur [Win92, Pear84] sind daher eine Reihe von Graphsuchverfahren beschrieben worden, die eine effizientere Suche erlauben, bei der nicht mehr alle möglichen Verzweigungen des Suchbaumes expandiert werden.

Als Abbruchkriterium wird ein Analyseziel definiert: Erreicht im aktuellen Suchbaumknoten durch Anwendung der Transformationsregeln ein Knoten des frei gewählten Zielkonzeptes C_{Ziel} den Status *komplette Instanz*, so ist das Analyseziel erreicht und die Suche terminiert.

Die in der Literatur beschriebenen Graphsuchverfahren besitzen unterschiedliche Eigenschaften bezüglich der Sucheeffizienz. Die einfach zu implementierende Bestensuche beispielsweise ist zwar in der Lage, mit einer minimalen Anzahl von expandierten Suchbaumknoten zum Ziel zu gelangen, dabei muß das Ziel aber nicht der Suchbaumknoten mit der besten Bewertung sein. Der A*-Algorithmus liefert als sogenanntes Optimalsuchverfahren den Suchbaumknoten mit der besten Bewertung. Dabei muß die Bewertung jedoch noch gewissen Bedingungen genügen.

Seine Leistungsfähigkeit erreicht der A*-Algorithmus unter anderem dadurch, daß die Bewertung einen additiven Term erhält, der die Differenz zur Bewertung des Zielknotens abschätzt. In der Literatur wird dies meist als Kostenfunktion ausgedrückt ¹:

$$c_{ges}(S_i) = c(S_i) + r(S_i) \quad (7.1)$$

Die Kosten $c_{ges}(S_i)$ setzen sich dabei aus den tatsächlichen Kosten $c(S_i)$ und den Restkosten $r(S_i)$ zusammen. Letztere können in der praktischen Anwendung nicht immer exakt berechnet werden und daher ersetzt man sie durch eine Abschätzung $r^*(S_i)$. In dem Fall einer optimistischen Restkostenabschätzung

$$r^*(S_i) \leq r(S_i) \quad (7.2)$$

findet der A*-Algorithmus immer den optimalen Pfad. Wird $r^*(S_i)$ größer als die tatsächlichen Restkosten geschätzt (Überschätzung der Restkosten), so wird unter Umständen die optimale Lösung nicht gefunden. In praktischen Anwendun-

¹Bei Verwendung einer Kostenfunktion besteht das Problem in einer Minimierung anstatt einer Optimierung bei Verwendung einer Bewertung

gen wird daher oft $r^*(S_i) = 0$ gewählt. Damit verfolgt die Suche zwar im allgemeinen mehr Möglichkeiten als notwendig, aber es wird immer der optimale Pfad gefunden.

Der zur Interpretation verwendete Algorithmus, basierend auf den Prinzipien des A*-Algorithmus, ergibt sich somit zu:

Funktion <i>Suche</i>
Liste der offenen Suchbaumknoten Offen = $\{S_0\}$
Solange Offen $\neq \{nil\}$ und Ziel nicht gefunden
rufe Funktion <i>Inferenz</i> für den ersten Suchbaumknoten aus Offen
falls erfolgreich
falls Rückgabewert von <i>Inferenz</i> $N_{konkurr} \neq \{nil\}$
entferne ersten Knoten aus Offen
erzeuge für jeden Knoten aus $N_{konkurr}$ einen Folgeknoten
trage Folgeknoten in Offen ein
rufe Bewertungsfunktion für Folgeknoten
entferne unzulässige und redundante Knoten aus Offen
sonst
rufe Bewertungsfunktion für ersten Suchbaumknoten aus Offen
sonst
entferne ersten Knoten aus Offen
sortiere Offen nach absteigender Bewertung unter Verwendung der Funktion $ORD(S_i, S_j)$

S_0 bildet die Wurzel des Suchbaums und enthält den initialen Interpretationszustand. Dazu zählen neben der Wissensbasis die Signaldaten und eine initiale Szenenbeschreibung. Für die in Abbildung 6.8 gezeigte Wissensbasis wird zweckmäßigerweise eine hypothetische Instanz des Konzeptes *Szene* als initiale Szenenbeschreibung verwendet.

Die Bewertung der Suchbaumknoten geht in der letzten Zeile des Algorithmus bei der Erstellung einer sortierten Liste der offenen Suchbaumknoten ein. Dabei wird die Funktion $ORD(S_i, S_j)$ gerufen, die zwischen zwei Suchbaumknoten eine relative Ordnung angibt. Der Abschnitt 7.3 beschreibt diese Funktion für das in dieser Arbeit verwendete Bewertungssystem.

Die Funktion zum Entfernen unzulässiger Knoten aus der Menge der offenen Suchbaumknoten folgt im A*-Algorithmus der Grundidee, daß Suchpfade, die mit höheren Kosten als ein alternativer Pfad zum selben Ziel führen, keinen besseren Weg mehr beinhalten und daher nicht mehr berücksichtigt werden müssen. Im Abschnitt 7.6 wird eine mögliche Implementierung dieses Prinzips vorgeschlagen und diskutiert.

7.3. Bewertung von Teilinterpretationen

Das Abspeichern von Teilinterpretationen in dem Suchbaum ermöglicht dem Suchalgorithmus, nach Abstieg in einen Pfad, der zu einem falschen Interpretationsergebnis führt, die Suche an einer anderen Stelle des Suchbaums nach einem alternativen Pfad fortzusetzen. Ausschlaggebend dafür, welcher Pfad verfolgt, das heißt, welcher Suchbaumknoten der offenen Liste des Suchalgorithmus expandiert wird, ist die Bewertung. Die Bewertung steuert damit ganz entscheidend den Verlauf der Suche.

Die Bewertung berechnet basierend auf den Attributen ein Maß, das angibt, wie gut der gegenwärtige Interpretationszustand mit dem intensionellen Modell übereinstimmt. Dieses wird für Instanzen und Suchbaumknoten zusammengefaßt, so daß jedes dieser Objekte eine Bewertung enthält.

Die Bewertung für Attribute, Instanz- und Suchbaumknoten wird einheitlich dargestellt und enthält drei Komponenten:

$$J = (\xi, v, \sigma)^T, 0 \leq \xi \leq 1, 0 \leq v \leq 1, 0 \leq \sigma \leq 1 \quad (7.3)$$

Die Zulässigkeit ξ gibt die Modellgültigkeit an. Stellt ein Suchbaumknoten einen gültigen Interpretationszustand dar, so erhält er den Wert eins. Wird ein Suchbaumknoten mit einer Zulässigkeit $\xi < \xi_{min}$ bewertet, ist er ungültig und wird aus der Liste der offenen Suchbaumknoten entfernt. Die Güte v ist das Maß der Übereinstimmung zwischen Modell und Signal, σ gibt die Sicherheit dieser Aussage an. Die Güte v beinhaltet gegebenenfalls eine Restkostenabschätzung. Entsprechend der Normierungsbedingung in Gleichung 7.3 nimmt die Güte v im Falle der optimistischen Restkostenabschätzung den Wert eins an².

Die Verwendung der Zulässigkeit ist meist optional und, soweit nicht anders vermerkt, wird sie hier mit dem Wert eins angenommen. Im folgenden werden daher nur die Bewertungsvariablen v und σ betrachtet und, soweit nicht anders vermerkt, zur Bewertung J' zusammengefaßt:

$$J = (1, v, \sigma)^T \rightarrow J' = (v, \sigma)^T \quad (7.4)$$

²Das heißt, den Kosten null entspricht eine Bewertung eins. Der Ausdruck Kosten wird hier benutzt, weil er in der Literatur für die optimalen Suchverfahren eingeführt ist. Vergleiche dazu auch Abschnitt 7.2.3.

Attributbewertung

Die Attributbewertung $J'(A)$ ist eine Funktion, die angibt, wie gut der Attributwert $V(A)$ zum Attributwertebereich $\epsilon(A)$ paßt. Sie wird daher hier als Kompatibilität bezeichnet. Der Attributwert wird aus den Signaldaten berechnet. Der Wertebereich wird entweder als Festwert aus der Wissensbasis übernommen - das entspricht einem Prototypen - oder dynamisch, modellgetrieben aus der bereits gewonnenen (Teil-)Szenenbeschreibung durch eine Attributwertebereichsberechnungsfunktion gewonnen.

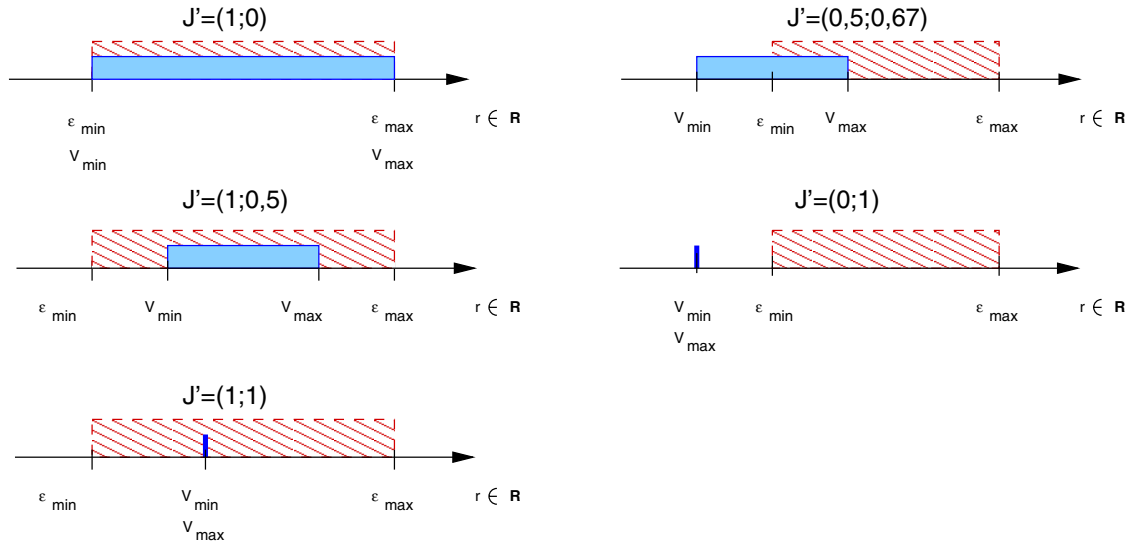


Abbildung 7.3.: Beispiele für Bewertungen eines Fließkommaattributes

Die Attributbewertung soll im folgenden anhand eines Fließkommaattributes erläutert werden. Durch die Einbindung in die Analyseabläufe ergeben sich mehrere Festlegungen an die Attributbewertungsfunktion:

- Bei der modellgetriebenen Erzeugung eines Instanzknotens werden die Attributwerte den Attributwertebereichen gleichgesetzt ($V = \epsilon$). Da für diesen Zustand kein Informationsgehalt in dem Attributwert steckt, wird durch die Forderung nach optimistischer Restkostenabschätzung $v = 1$ und die Sicherheit $\sigma = 0$. Dieser Fall ist in Abbildung 7.3 links oben dargestellt.
- Liegt der Attributwert $V(A) = (V_{min}, V_{max})$ innerhalb des Wertebereiches $\epsilon(A) = (\epsilon_{min}, \epsilon_{max})$, gilt $v = 1$.
- Liegt der Attributwert außerhalb des Wertebereiches, so wird $v = 0$ (Abbildung 7.3 rechts Mitte).

- Liegt der Attributwert teilweise außerhalb des Wertebereiches, so entspricht v der Überlappung zwischen Attributwert und -wertebereich.
- Die Sicherheit σ wird maximal, wenn der Attributwert „scharf“ ist, das heißt, $V_{min} = V_{max}$ ist. Sonst ist σ das Verhältnis zwischen $\epsilon_{max} - \epsilon_{min}$ und $V_{max} - V_{min}$.

Daraus ergeben sich die Rechenvorschriften für die Bewertungsgüte und -sicherheit eines Fließkommaattributes zu:

$$D = V_{max} - V_{min} \quad (7.5)$$

$$D' = \epsilon_{max} - \epsilon_{min} \quad (7.6)$$

$$a = \begin{cases} (V_{max} - \epsilon_{max})/D' & \text{für } (V_{max} - \epsilon_{max})/D' > 0 \\ 0 & \text{sonst} \end{cases} \quad (7.7)$$

$$b = \begin{cases} (\epsilon_{min} - V_{min})/D' & \text{für } (\epsilon_{min} - V_{min})/D' > 0 \\ 0 & \text{sonst} \end{cases} \quad (7.8)$$

$$v(A) = \begin{cases} 1 - a - b & \text{für } 0 \leq 1 - a - b \leq 1 \\ 1 & \text{für } 1 - a - b > 1 \\ 0 & \text{sonst} \end{cases} \quad (7.9)$$

$$\sigma(A) = \begin{cases} (D' - D)/D' & \text{falls } (D' - D)/D' > 0 \\ 0 & \text{sonst} \end{cases} \quad (7.10)$$

$$\xi = 1 \quad (7.11)$$

Für vektorielle Attribute kann diese Berechnungsvorschrift verallgemeinert werden. So ergibt sich die Überlappung bei 2D-Vektoren anhand einer Überprüfung der sich schneidenden Flächen und bei 3D-Vektoren anhand der Volumenschnitte.

Die beschriebene Berechnungsvorschrift ist als Standardbewertungsfunktion für Fließkommaattribute implementiert. Für spezielle Attribute kann diese überschrieben werden. Das ist beispielsweise der Fall, wenn der Zulässigkeit ξ ein Wert ungleich eins zugewiesen werden soll. Dies gilt in gleicher Weise für die im folgenden beschriebene Instanz- und Suchbaumknotenbewertung.

Instanzbewertung

In die Instanzbewertung $J'(N)$ gehen standardmäßig die Bewertungen aller Attribute $A(N)$ durch Bildung eines gewichteten Mittelwertes ein:

$$W_N = \sum_i^{\mathbf{A}(N)} w(A_i) \quad (7.12)$$

$$v(N) = \frac{1}{W_N} \sum_i^{\mathbf{A}(N)} w(A_i) v(A_i) \quad (7.13)$$

$$\sigma(N) = \frac{1}{W_N} \sum_i^{\mathbf{A}(N)} w(A_i) \sigma(A_i) \quad (7.14)$$

$$\xi(N) = \frac{1}{W_N} \sum_i^{\mathbf{A}(N)} w(A_i) \xi(A_i) \quad (7.15)$$

$$(7.16)$$

Die Gewichte $w(A_i)$ ermöglichen die heuristische Betonung wichtiger Attribute in der Bewertung oder bei $w(A_i) = 0$ die Nichtwertung spezieller Attribute, die beispielsweise als Datencontainer dienen.

Suchbaumknotenbewertung

In die Bewertung $J'(S)$ eines Suchbaumknotens S gehen alle Instanzknoten mit dem Status partielle und vollständige Instanz ein. Fehlinstanzen, sowie Konzepte und hypothetische Instanzen gehen nicht ein. Letztere werden ausgenommen, weil sie noch keinen Bezug zum Signal haben und daher keine Entscheidungsgrundlage bieten.

Eine naheliegende Lösung zur Bewertung des Suchbaumknotens ist die Addition der Instanzbewertungen. Ein Problem ergibt sich allerdings bei der Abschätzung der Restkosten: Zu Beginn der Analyse ist nicht abzusehen, wie viele Instanzen die Szenenbeschreibung am Ende enthält, insbesondere wenn optionale Teile gesucht werden.

Daher wird ein gewichteter Mittelwert als Suchbaumknotenbewertung eingeführt, der die Eigenschaft einer Normierung hat und Suchbaumknoten unterschiedlicher Tiefe im Suchbaum miteinander vergleichbar macht:

$$\mathbf{N}'(S) = \{n | n \in \mathbf{N}(S) \text{ und Status}(n) \text{ partiell oder vollständig}\} \quad (7.17)$$

$$W_S = \sum_i^{\mathbf{N}'(S)} w(N_i) \quad (7.18)$$

$$v(S) = \frac{1}{W_S} \sum_i^{N'(S)} w(N_i)v(N_i) \quad (7.19)$$

$$\sigma(S) = \frac{1}{W_S} \sum_i^{N'(S)} w(N_i)\sigma(N_i) \quad (7.20)$$

$$\xi(S) = \frac{1}{W_S} \sum_i^{N'(S)} w(N_i)\xi(N_i) \quad (7.21)$$

Für einen Suchbaumknoten ohne Instanzen wird die Bewertung optimistisch angegeben, das heißt, $J'(S) = (1; 0)$.

Da die Netzwerkkanten während der Analyse durch die Transformationsregeln ausgewertet werden, gehen sie nicht zusätzlich explizit noch mal in die Bewertung ein.

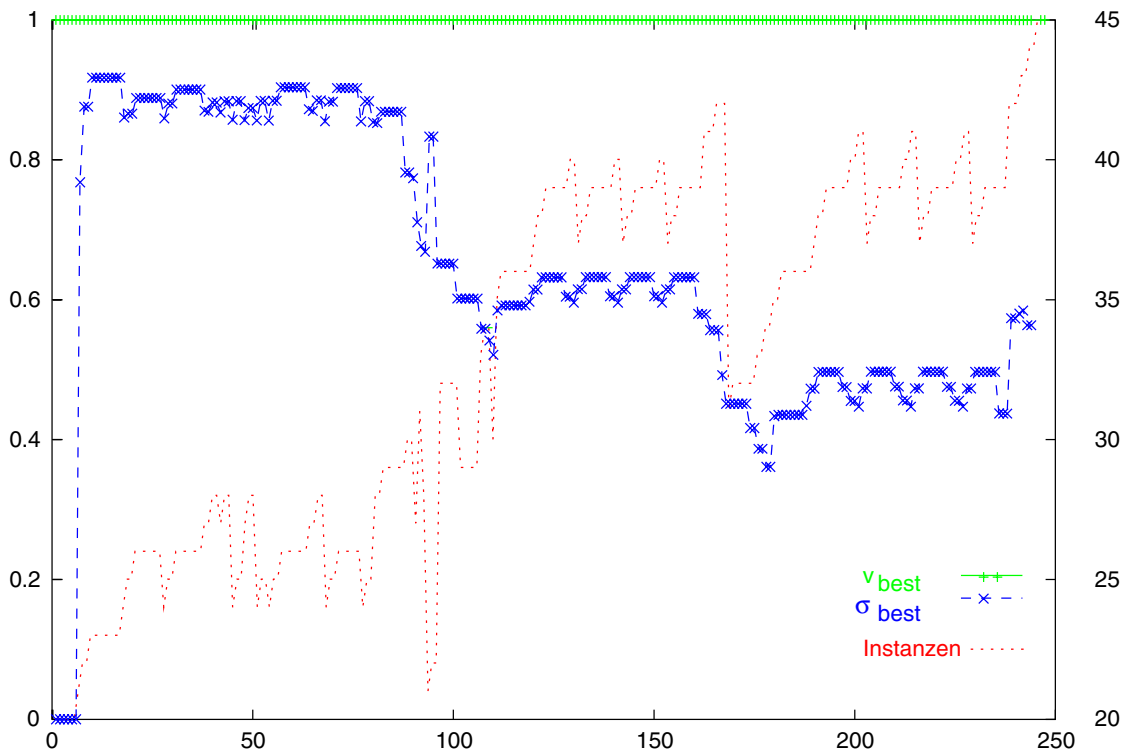


Abbildung 7.4.: Anzahl der Instanzen und Verlauf der Bewertungsgüte und -sicherheit des besten Suchbaumknotens während der Analyse der Szene „Restaurant“

Einen typischen Verlauf der Bewertungsgüte und -sicherheit zeigt Abbildung 7.4. Dargestellt ist die Bewertung und Anzahl der Instanzen des jeweils besten Such-

baumknoten während der Analyse der Szene „Restaurant“. Gesucht wird nur nach obligatorischen Teilen für ein Konzept *Haus*. Die resultierende Szenenbeschreibung ist in Abbildung A.2 im Anhang dargestellt. Die Ordinate von Abbildung 7.4 zeigt die Anzahl der ausgeführten Transformationsgruppen³. Die Abszisse zeigt die Bewertungsgüte und -sicherheit des bestbewerteten Suchbaumknotens und gleichzeitig die jeweilige Anzahl der Instanzen in diesem Knoten.

Die Bewertungsgüte liegt durchweg bei eins. Die Sicherheit liegt am Anfang bei Null und steigt dann schnell gegen einen Wert von ca. 0,9. Das liegt daran, daß zu Beginn der Analyse die generierten hypothetischen Instanzen noch nicht anhand des Signals verifiziert werden können, so daß die Restkostenabschätzung optimistisch ausfällt.

Die Anzahl der Instanzen schwankt. Ein starker Sprung nach unten rührt daher, daß der gerade expandierte Suchbaumknoten nicht mehr die beste Bewertung aufweist und der Suchalgorithmus einen Knoten weiter expandiert, der im Suchbaum weiter oben liegt und demzufolge noch nicht so viele Instanzen enthält.

Nach ca. 100 Regelgruppen sinkt die Bewertungssicherheit auf einen Wert um ca. 0,6. Das liegt daran, daß zu diesem Zeitpunkt „verdeckte“ Objekte in die Szenenbeschreibung aufgenommen werden. Diese werden optimistisch abgeschätzt, da sie nicht im Signal gefunden wurden.

7.4. Bindung von Segmentierungsdaten (Matching)

Die Bindung der Datenknoten, die den Regionen aus der Segmentierung nach Abschnitt 5.3 entsprechen, erfolgt modellgetrieben. Ausgehend von einer initialen Szenenbeschreibung werden dabei in einer Top-Down-Phase hypothetische Instanzen erzeugt (siehe Beispiel in Abbildung 7.5 links).

Für eine hypothetische Instanz des Konzeptes *Polygon2D* wertet die Inferenzmaschine als nächstes die *data-of*-Kante anhand der Regel $R_{H:dat}$ zur Datenbindung aus (Schritt 6. in Abbildung 7.5). Dabei wird für die hypothetische Instanz des Konzeptes *Polygon2D* dessen Suchfunktion gerufen (vergleiche Abschnitt 6.5.2). Die gerufene Suchfunktion liefert eine Liste von Instanzen des Konzeptes *Region* zurück, die im aktuellen Suchbaumknoten keine *data-of*-Kante besitzen, also noch frei zur Bindung sind und möglichst gut zur Modellerwartung passen. Für jedes zurückgegebene Datenobjekt erzeugt der Kontrollalgorithmus einen neuen Suchbaumknoten, das heißt, der Suchbaum spaltet sich auf; sofern mehr als ein

³Innerhalb einer Gruppe von Regeln erfolgt kein Wechsel zu einem anderen Suchbaumknoten

Datenobjekt zurückgeliefert wird.

Die Modellerwartung ist bei der Suche nach der ersten Wand (Abbildung 7.5 links) gleich dem in der Wissensbasis abgelegten Modellwissen, daß eine Wand senkrecht steht. Das heißt, für den dargestellten Fall würde die Suchfunktion alle freien Regionen zurückliefern, die eine innerhalb eines Toleranzwinkels senkrechte Flächennormale besitzen. Die Suchfunktion wertet dazu das Attribut *Normale* der hypothetischen Instanz *Wand-1* und die Flächennormale der Region aus⁴.

Bei der Suche nach der nächsten Wand wird die Modellerwartung unter Verwendung des Wissens angepaßt, daß Hauswände senkrecht aufeinander stehen. Dies geschieht durch die Wertebereichsberechnungsfunktion des Attributes *Wand.Normal*. Durch die Einschränkung des Suchbereiches reduziert sich die Liste der möglichen Kandidaten erheblich. Für die Szene „Haus-1“ mit dem in Abbildung 5.4 dargestellten Segmentierungsergebnis mit 16 Regionen gibt die Suchfunktion für die erste Wand eine Liste von 10 Regionen und für die zweite Wand eine Liste von nur noch vier Regionen zurück. Bei vollständiger Expansion des Suchbaumes entspricht das einer Reduktion des Suchaufwandes von $16 \times 15 = 240$ auf $10 \times 4 = 40$, um den Faktor sechs.

Steht keine Region mehr als Kandidat zur Verfügung, so liefert die Datensuchfunktion eine leere Liste $\{nil\}$ zurück. In diesem Fall erhält die rufende, hypothetische Instanz den Status *Fehlinstanz*. Dieser Status wird im folgenden vom Kontrollalgorithmus Bottom-Up auf die übergeordneten Knoten übertragen. Der Vorgang stoppt, wenn ein Knoten optionaler Teil des übergeordneten ist.

Behandlung des Verdeckungsproblems

Die Datensuchfunktion wird für alle polygonalen Komponenten der Szene gerufen. Dabei ergibt sich aufgrund der dreidimensionalen Natur des Problems die teilweise oder vollständige Verdeckung von Komponenten. Um verdeckte, oblique Komponenten trotzdem ordnungsgemäß instanzieren zu können, wird ein spezieller Knoten eingeführt: Der Knoten *HiddenRegion* kennzeichnet eine Region, die verdeckt ist. Er kann beliebig oft gebunden werden.

Ob eine Region sichtbar sein kann oder grundsätzlich verdeckt sein muß, ist anhand der Erwartung für die Flächennormale überprüfbar: Sichtbare Flächen besitzen eine Normale, die in Richtung der Kamera weist.

⁴die Flächennormale der als Region abgebildeten Oberfläche steht aufgrund der Stereoauswertung zur Verfügung

Bei der Suche nach optionalen Komponenten müssen weitere Fälle berücksichtigt werden. Optionale Teile werden nach den obligaten gesucht (vergleiche Abschnitt 7.6). Ist die Anzahl der möglichen optionalen Komponenten, zum Beispiel einer Hauserweiterung, groß oder unendlich, so wird solange nach neuen Komponenten gesucht, bis die Suchfunktion keine Region mehr zurückliefert. Dies ist der Hinweis, daß die optionale Komponente nicht existiert. Gibt es noch Kandidaten in der Liste der freien Regionen, so heißt das noch nicht, daß die gesuchte Komponente tatsächlich existiert. Um diesen Fall zu berücksichtigen, fügt die Suchfunktion ein *nil*-Objekt in die zurückgegebene Liste der Regionen ein. Für dieses *nil*-Objekt wird ein Suchbaumknoten mit einer *Fehlinstanz* erzeugt. Das heißt, die Möglichkeit, daß die Komponente nicht vorhanden ist, wird in die Menge der offenen Suchbaumknoten als zusätzliche Hypothese aufgenommen und anhand der Bewertung, die mehr Daten einbezieht, wird entschieden, welche Deutung zutrifft.

Ein weiterer Spezialfall ergibt sich für bereits als nicht sichtbar klassifizierte Komponenten: Die Suche nach optionalen Teilen dieser Komponenten liefert immer das *nil*-Objekt zurück, das heißt, es werden keine Kandidaten ausgewählt, da sie nicht sichtbar sein können.

7.5. Datenfluß während der Analyse

Dieser Abschnitt zeigt anhand eines Beispiels, wie die bislang erläuterten Prinzipien während der Analyse zusammenspielen. Die Interpretation arbeitet mehrstufig und sieht verschiedene Phasen mit zwei Richtungen des Datenflusses vor: In den Top-Down-Phasen werden Modellerwartungen in Form von Randbedingungen propagiert. In Bottom-Up-Phasen werden die Signalwerte und daraus berechnete Attributwerte propagiert und die Hypothesen verifiziert.

Eine tragende Rolle bei dem Austausch und der Propagation von Informationen spielen die Attribute: Die Attributwertebereiche dienen dem Datenfluß in Top-Down-Richtung, die Attributwerte in Bottom-Up-Richtung. Dieses Konzept soll am Beispiel der Attribute *Haus.Höhe*, *Mauerwerk.Höhe* und *Wand.Höhe* und den Abbildungen 7.5 und 7.6 erläutert werden.

Die Analyse wird mit der Wissensbasis aus Abbildung 6.8 und einer hypothetischen Instanz des Konzeptes *Szene* gestartet. In der Folge feuert fünfmal die Regel $R_{H:INV}$ und erzeugt dabei jeweils die hypothetischen Instanzen *Haus-1*, *Mauerwerk-1*, *Wand-1*, *Polygon3D-1* und *Polygon2D-1*⁵.

⁵Tatsächlich wird zuerst eine Instanz des Konzeptes *Boden* erzeugt, was hier der Einfachheit halber nicht dargestellt ist.

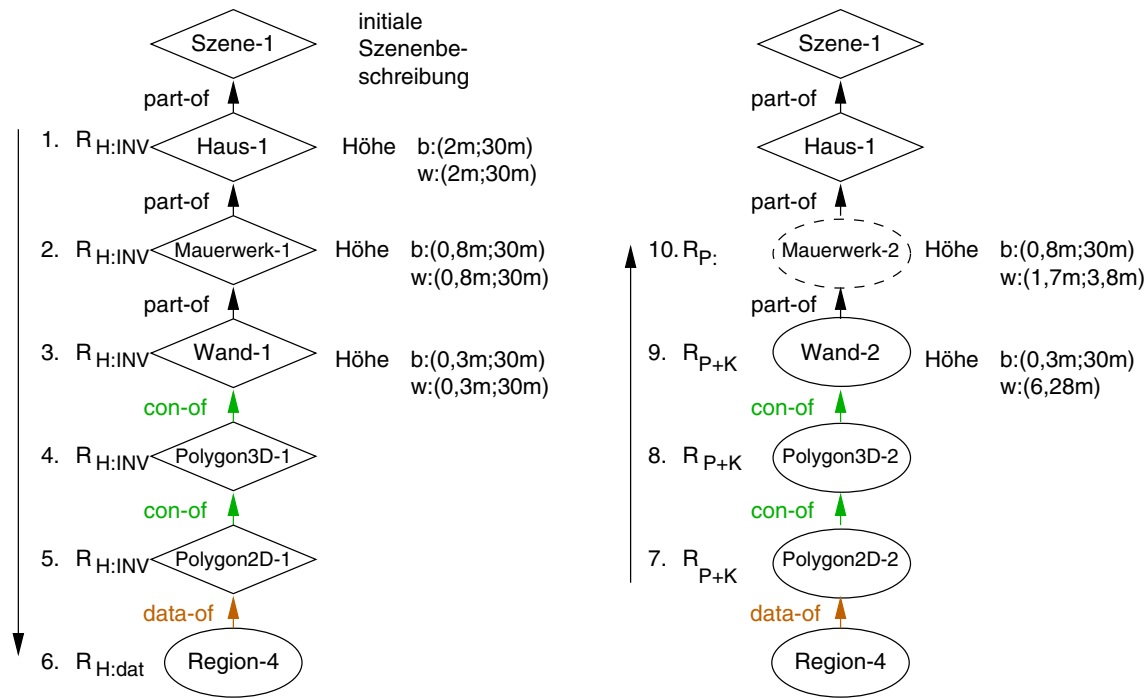


Abbildung 7.5.: In einer Top-Down-Phase (links) werden Erwartungen als Randbedingungen propagiert. In einer Bottom-Up-Phase (rechts) werden Daten aus dem Signal berechnet.

Der Wertebereich des Attributes *Haus-1.Höhe* wird bei der Erzeugung der Instanz *Haus-1* aus der Wissensbasis kopiert und beträgt $\epsilon(\text{Haus-1.Höhe}) = (2m; 30m)$. Das heißt, es wird erwartet, daß die Höhe eines Hauses im Bereich 2-30 m liegt. Der Attributwert $V(\text{Haus-1.Höhe})$ wird beim Instanzieren auf den Wert des Wertebereichs vorinitialisiert.

Der Wertebereich des Attributes *Mauerwerk-1.Höhe* wird bei der Instanzierung auf den Wert $\epsilon(\text{Mauerwerk-1.Höhe}) = (0, 8m; 30m)$ gesetzt. Der Minimalwert ist kleiner als der des entsprechenden Attributes des Knotens *Haus-1*, um eine größere Toleranz zu gewährleisten. Der Wertebereich $\epsilon(\text{Wand-1.Höhe})$ wird aus dem Attributwert *Höhe* des übergeordneten Knotens, also *Mauerwerk-1* berechnet:

$$\epsilon(\text{Wand-1.Höhe}) = (0, 4 \times V_{\min}(\text{Mauerwerk-1.Höhe}); V_{\max}(\text{Mauerwerk-1.Höhe})) \quad (7.22)$$

In Schritt 6 wird ein Datum gebunden. In den folgenden Schritten (Abbildung 7.5 rechts) findet eine Bottom-Up-Phase statt. Die Schritte 7-9 fassen jeweils zwei

Regeln zusammen: Die Regel R_P erzeugt eine partielle Instanz und die Regel R_K erzeugt eine vollständige Instanz. Dabei werden die Werteberechnungsfunktionen für alle Attribute der jeweiligen Knoten aufgerufen. Für die gefundene Region *Region-4* erhält das Attribut *Wand-2.Höhe*⁶ den Wert 6,28 m, der aus den Daten berechnet wird.

In Schritt 10 wird durch die Regel R_P eine partielle Instanz *Mauerwerk-2* erzeugt. Die Regel R_K kann noch nicht feuern, solange nicht alle obligatorischen Teile instanziiert sind (das sind mindestens vier Wände). Das Attribut *Mauerwerk-2.Höhe* erhält jedoch einen neuen Wert: $V(\text{Mauerwerk-2.Höhe}) = (1,7m; 3,8m)$. Dieser Attributwert ist noch nicht „scharf“, weil noch nicht alle obligaten Teile gefunden worden sind, der Bereich ist jedoch aufgrund der Daten der ersten gefundenen Wand eingeschränkt.

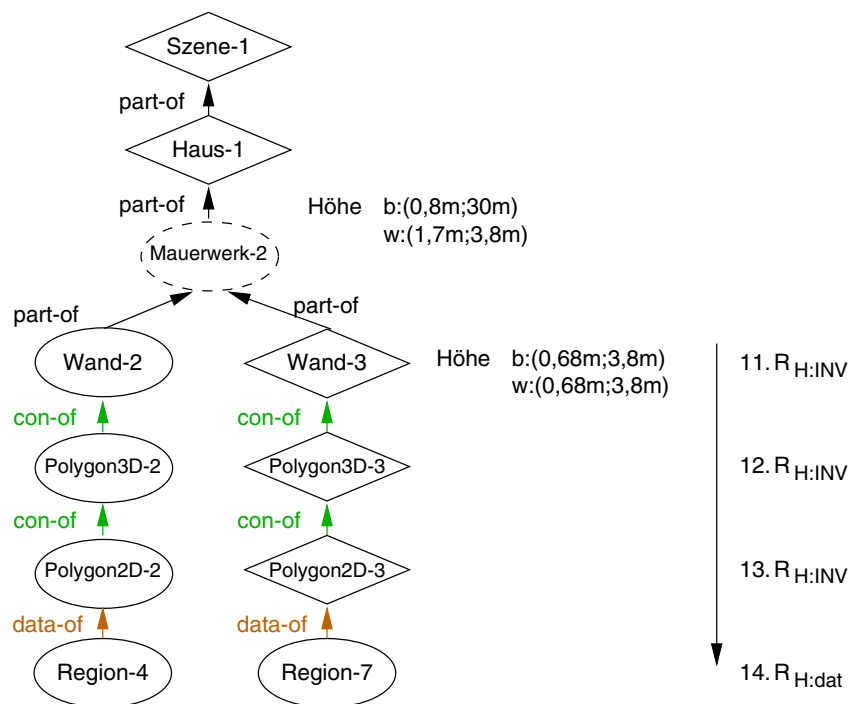


Abbildung 7.6.: In einer erneuten Top-Down-Phase kann aufgrund des Vorwissens der Wertebereich für das Attribut *Höhe* der hypothetischen Instanz *Wand-3* eingeschränkt werden

In den nächsten Analyseschritten, die in Abbildung 7.6 dargestellt sind, wird ein

⁶Wie in Abschnitt 7.2.2 erläutert, wird bei der inkrementellen Dokumentation des Analysefortschrittes ein Knoten bei Änderung eines Attributes durch einen neuen ersetzt. Daher erhalten modifizierte Knoten eine andere, eindeutige Nummer als vorher.

zweiter Instanzknoten *Wand-3* erzeugt. Da der Wert des Attributes *Mauerwerk-2.Höhe* neu ist, ergibt sich nach Gleichung 7.22 ein Wertebereich $\epsilon(\text{Wand-3.Höhe}) = (0, 68m; 3, 8m)$. Nach erfolgter Bindung des Datenknotens *Region-7* erhält das Attribut den Wert $V(\text{Wand-3.Höhe}) = (1, 69m)$ (nicht abgebildet). Diejenigen Hypothesen, in denen eine Region gebunden wurde, die nicht in dem Attributwertebereich liegt, erhalten eine schlechtere Bewertung und werden (vermutlich) nicht weiter expandiert.

Die nächsten beiden Wände sind nicht mehr sichtbar, es wird deshalb das Spezialobjekt *HiddenRegion* gebunden. Nach vollständiger Instanzierung der vier obligaten Wände feuert die Regel R_P für den Knoten *Haus-1*, der Status wechselt auf *partielle Instanz* und das Attribut *Höhe* erhält einen neuen Wert, der sich aus der Höhe des Mauerwerks plus einer Abschätzung der Dachhöhe ergibt, die relativ angegeben wird (das 1,8-fache der Mauerwerkhöhe). Für das als nächstes zu suchende Dach ergibt sich damit ein eingeschränkter Wertebereich.

Das sukzessive Einengen der Attributwerte durch die mehrfachen Berechnungsphasen erfolgt durch Akkumulation von Informationen, die notwendig sind, um das komplexe Problem lösen zu können.

Die dynamische Anpassung des Attributwertebereichs an die bereits erstellte Szenenbeschreibung durch modellgetriebene Propagation ist ein Kennzeichen der intensionellen Beschreibung. Im Unterschied dazu findet in prototypischen oder extensionellen Beschreibungen keine solche Anpassung statt. Die Bewertung vergleicht die aus dem Signal gewonnene Szenenbeschreibung mit jeweils genau einem individuellen Prototyp, wodurch der Suchaufwand (Anzahl der zu vergleichenden Prototypen) unter Umständen enorm groß wird.

7.6. Analysestrategien zur Interpretation von Gebäudeszenen

Obwohl die hier betrachteten Szenen für den menschlichen Betrachter relativ „einfach“ erscheinen, stellt sich für die automatische Analyse bereits ein beträchtlicher Suchaufwand ein. So ergibt sich für die Szene „Haus-1“, mit den in Abschnitt 5.3 vorgestellten Segmentierungsergebnissen, für die Suche nach den obligaten Komponenten eines Hauses (2 sichtbare Wände, 1 Dachhälfte und eine „Bodenregion“) unter Verwendung der gefundenen 16 Regionen folgende Anzahl an möglichen Permutationen: $16 \times 15 \times 14 \times 13 = 43680$. Dabei ist bereits berücksichtigt, daß nur maximal zwei Wände und eine Dachhälfte sichtbar sein können. Unter Berücksichtigung der optionalen Komponenten und Spezialisierungen steigt die Anzahl der theoretisch möglichen Deutungen auf $16! = 2,1 \times 10^{13}$. Aus die-

sen Überlegungen wird deutlich, daß das Problem nur durch Einschränkung des Suchraums und eine entsprechende Analysestrategie zu lösen ist.

Als erste Maßnahme wird daher die Interpretation in zwei Phasen gegliedert: Die erste Phase bestimmt die obligatorischen Komponenten. Dabei wird davon ausgegangen, daß sich genau ein Haus in der Szene befindet. In der zweiten Analysephase wird nach optionalen Teilen gesucht. Aufgrund des Vorwissen und der bestehenden Szenenbeschreibung ist die Suche dabei sehr zielgerichtet.

Beschränkung der Suche

Eine Möglichkeit zur Beschränkung der Suche ist die bereits beschriebene Verwendung einer Datensuchfunktion, die nur geeignete Kandidaten aus der Liste der Segmentierungsdaten zurückliefert, wie in Abschnitt 7.4 gezeigt. Darüber hinaus bietet der in Abschnitt 7.2.3 beschriebene Kontrollalgorithmus verschiedene Möglichkeiten zur Reduktion des Suchaufwandes. Im einzelnen sind das:

- Abbruch der Suche, wenn das vorgegebene Zielkonzept vollständig instanziiert wurde.
- Entfernen von unzulässigen Suchbaumknoten.
- Entfernen von redundanten Suchbaumknoten.
- Verwendung einer Restkostenabschätzung.

Als Abbruchbedingung dient in der ersten Phase das Kriterium, ob im gerade expandierten Suchbaumknoten eine vollständige Instanz des Konzeptes *Szene* erzeugt wurde. Ist das der Fall, so ist die erste Phase beendet, es wurden alle obligaten Komponenten instanziiert. Das Ergebnis der Suche ist der bestbewertete Suchbaumknoten des Suchbaums.

Die Bewertung kennzeichnet Suchbaumknoten, die unmögliche Deutungen darstellen, als unzulässig. In der hier zugrunde liegenden Wissensbasis wird beispielsweise für Instanzen des Konzeptes *Mauerwerk* überprüft, ob sich die gefundenen Wände im Winkel von ca. 90° schneiden oder ob der Winkel ca. 270° entspricht. Der letztere Fall wird durch die Suchfunktion für Regionen nicht überprüft, so daß etwa eine Hauswand und eine Anbauwand als Hypothese für die Grundmauern des Hauses erzeugt werden können, die einen Außenwinkel von 270° statt der erwarteten 90° besitzen. Diese Hypothese wird daher durch die Bewertung als unzulässig zurückgewiesen.

Die Entfernung redundanter Suchbaumknoten basiert auf dem Prinzip der dynamischen Programmierung [Win92]: Wird auf der Suche nach einem Zielpfad im Suchbaum ein (Zwischen-)Knoten über verschiedene Pfade erreicht, so kann in den schlechter bewerteten Pfaden kein anderer Weg mehr enthalten sein, der eine bessere Bewertung liefert. Mit anderen Worten: Wird eine bestimmte Szenenbeschreibung auf unterschiedlichen Pfaden erreicht, so braucht nur der beste Pfad weiter betrachtet zu werden, alle anderen können entfernt werden. Das gezielte Entfernen von Suchbaumknoten heißt Pruning.

Das Problem besteht nun aber darin, effizient zu erkennen, wann zwei Suchbaumknoten dieselbe Szenenbeschreibung darstellen. Dies kann problemunabhängig nur schwer ausgedrückt werden. Daher wird hier von den folgenden Überlegungen ausgegangen: Redundante, gleichwertige Interpretationen entstehen in der vorliegenden Anwendung, wenn beispielsweise zwei Wänden eines Hauses einmal die Region A für die erste Wand und B für die zweite zugeordnet wird und als zweite Variante aufgrund der Kombinatorik als erste Wand B und A als zweite. In diesem Fall ist der schlechter (oder gleich) bewertete Suchbaumknoten redundant und kann entfernt werden. Der Vergleich zweier Suchbaumknoten prüft daher, ob 1. dieselben Regionen als Daten gebunden sind, ob 2. diese Regionen einen Vaterknoten (auf der semantischen Ebene) desselben Konzeptes besitzen und ob 3. beide Knoten denselben Expansionsgrad besitzen. Das letztere Kriterium vergleicht, ob die Suchbaumknoten dieselbe Anzahl von Instanzen enthalten. Nur so ist der folgende Vergleich der Bewertungen zwischen zwei Suchbaumknoten zulässig. Dieses spezielle für das Problem entwickelte Kriterium kann rechentechnisch schnell überprüft werden.

Die Verwendung eines Restkostenterms $r(S_i)$ in Gleichung 7.1 bewirkt, daß nur noch die Pfade mit den geringsten Kosten expandiert werden [Win92]. Als eine mögliche Abschätzung der Restkosten $r^*(S_i)$ wird ein empirisches Maß eingeführt, das auf einer Abdeckung des Signals $a_{\text{Signalabdeckung}}$ beruht und in die Bewertungsgüte $v(S)$ eines Suchbaumknotens eingeht:

$$a_{\text{Signalabdeckung}} = \frac{N_{\text{gebunden}}}{N_{\text{total}}} \quad (7.23)$$

$$v'(S_i) = v(S_i) + k_1 \cdot a_{\text{Signalabdeckung}} + k_2 \quad (7.24)$$

Dabei stellt N_{total} die Anzahl aller Pixel ungleich Null im Label-Bild dar und N_{gebunden} die Anzahl der Pixel der im Suchbaumknoten als Datenobjekte gebundenen Regionen. Die Konstanten k_1, k_2 wurden empirisch wie folgt eingestellt: $k_1 = 0, 1, k_2 = -0, 4$.

Abbruchbed.	Pruning	Restkosten	SB-Knoten	Regeln	Ergebnis
Szene „Haus-1“					
theoretische maximale Anzahl			43680		
nein	nein	nein	1618	22945	richtig
nein	ja	nein	976	12735	richtig
ja	nein	nein	164	1705	richtig
ja	ja	nein	128	1185	richtig
ja	ja	ja	24	142	falsch
Szene „Restaurant“					
theoretische maximale Anzahl			43680		
nein	nein	nein	271	6262	richtig
nein	ja	nein	231	4707	richtig
ja	nein	nein	28	269	richtig
ja	ja	nein	28	265	richtig
ja	ja	ja	21	146	richtig

Abbildung 7.7.: Reduktion des Suchaufwandes

Die Wirkungsweise der einzelnen Maßnahmen zur Sucheinschränkung ist in der Abbildung 7.7 für die Szenen „Haus-1“ und „Restaurant“ dokumentiert. Die ersten drei Spalten geben verschiedene Kombinationen über die Verwendung des Abbruchkriteriums, des Prunings und einer Restkostenabschätzung wieder. Das Pruning umfaßt hierbei sowohl das Entfernen unzulässiger, als auch redundanter Suchbaumknoten.

Die jeweils erste Zeile (Abbruch nein, Pruning nein, Restkosten nein) entspricht einer vollständigen Suche unter Verwendung eines modellbasierten Matching der Segmentierungsdaten. Der Vergleich zu den maximal möglichen Kombinationen von 43680⁷ zeigt den deutlichen Gewinn dieses Vorgehens. Dieses Ergebnis spiegelt sich auch in der Anzahl der angewandten Regeln (Spalte Regeln).

Die Verwendung des Prunings bringt vor allem bei der Anzahl der gefeuerten Regeln, die die Dauer der Suche bestimmen, eine Reduktion des Aufwandes um ca. 24-45 %. Die größte Verringerung des Suchaufwandes liefert die Verwendung des Abbruchkriteriums. Die Verwendung einer empirischen Restkostenabschätzung bringt ebenfalls eine Reduktion des Aufwandes. Sie führt für die Szene „Haus-1“ jedoch zu einer falschen Interpretation. Dies ist darauf zurückzuführen, daß die empirische Restkostenabschätzung nicht gut an das Problem angepaßt ist. Eine Optimierung für die hier betrachteten Szenen wäre sicher möglich. Das Versagen zeigt aber die Gefahr der empirischen Restkostabschätzung: Sie kann bei leichter

⁷Die Szenen „Haus-1“ und „Restaurant“ haben zufällige dieselbe Anzahl von 16 Regionen aus der Segmentierung.

Fehlanpassung zum falschen Interpretationsergebnis führen. Besteht das Ziel in einer robusten Interpretation, so wird man wahrscheinlich auf eine empirische Restkostabschätzung verzichten und dafür eine etwas umfangreichere Suche in Kauf nehmen. In Anhang A ist das Ergebnis der Interpretation ohne optionale Komponenten für die Szene „Restaurant“ abgebildet.

Suche nach optionalen Komponenten

Nach erfolgter Instanziierung der obligatorischen Szenenkomponenten erfolgt die Suche nach optionalen Komponenten. Diese ist wiederum zweigeteilt: zuerst werden die Komponenten des Hauses (oder Häuser, falls die Szene mehrere enthält) gesucht, danach weitere Szenenobjekte, die nicht näher spezifiziert sind und dem Knotentyp *Objekt* in der Wissensbasis entsprechen. Der Grund für diese Zweiteilung ist, daß die Suche nach den Komponenten eines Hauses aufgrund des allgemeinen Modellwissens und des Wissens über die Szene sehr zielgerichtet arbeitet. Für Komponenten, wie Hausanbauten oder Dachgauben ist sehr gut vorherzusagen, wo diese geometrisch auftreten. Mit diesem Vorwissen können die optionalen Komponenten sehr sicher bestimmt werden, da die Bewertung sehr selektiv ist.

Die Suche nach den verbleibenden unspezifischen Szenenobjekten endet, wenn keine ungebundenen, das heißt, noch nicht interpretierten Segmentierungsdaten (Regionen) mehr vorhanden sind.

Zuordnung der Szenenkonturen

Nachdem der Szeneninhalte anhand des regionenbasierten Segmentierungsergebnisses gedeutet wurde, werden die Verbindungen der zusammenhängenden Szenenkomponenten, die sich als Schnittkanten der dreidimensionalen Polygon konkretisieren, den Szenenkonturen im (Kontur-)Bild zugeordnet. Die Szenenkonturen spielen damit für die Interpretation eine untergeordnete Rolle, da sie keinen Einfluß auf die Deutung haben. Sie sind jedoch für die nachfolgende Oberflächenrekonstruktion wichtig.

Aufgrund der aus der Stereoauswertung grob bekannten Lage der Polygone im Raum kann die Lage der Konturen durch Projektion der Schnittkanten der Raumpolygone in die Bildebene relativ gut vorhergesagt werden. Mit dieser Vorgabe findet eine Datensuchfunktion [Wiebe98] die besten Kandidaten der zugehörigen Kontursegmente im Bild.

Für das Rahmensystem AIDA [Bueck98] wurden daher die folgenden Komponenten entwickelt:

- Der Kontroll-Browser erlaubt den Zugriff auf die Kontrollstrukturen der Interpretation. Das ist im einzelnen: Die (Re-)Initialisierung der Suche, das Setzen eines Zielkonzeptes und die Manipulation der Inferenzregeln in einem separaten Strategie-Editor. Zur Verfolgung des Analyseablaufes kann die Suche schrittweise, das heißt, immer genau eine Inferenzregel pro Schritt ausgeführt werden oder solange, bis der Knoten eines Konzeptes manipuliert wird, der als Breakpoint gesetzt wurde. Der Analysefortschritt wird für jeden Schritt im Suchbaum- und Netz-Browser dokumentiert.
- Der Suchbaum-Browser zeigt den aktuellen Suchbaum als Graph an. Bei Auswahl eines Suchbaumknotens wird dieser im Netz-Browser dargestellt.
- Der Netz-Browser zeigt den Inhalt eines Suchbaumknotens. Bei schrittweiser Suche (Schrittfunktion des Kontroll-Browsers) wird automatisch nach jedem Schritt der beste Suchbaumknoten angezeigt. Der Browser kennzeichnet die unterschiedlichen Instanz-Stati und verschiedenen Kantentypen farblich.
- Bei Betätigung eines Knotens oder einer Netzwerkkante wird für das betreffende Objekt ein Daten-Browser geöffnet. Für Knoten können deren Attribute aus einem Menü ausgewählt und in einem weiteren Browser geöffnet werden.
- In zwei speziellen Bild-Browsern werden die Segmentierungs- und Bilddaten (Regionen und Kanten) dargestellt. Durch Betätigung einer Region wird der dazu gehörende Netzwerkknoten im Daten-Browser geöffnet. Ferner sind spezielle Visualisierungsfunktionen vorgesehen.

Die Abbildungen 7.8 und 7.9 zeigen eine typische Arbeitssitzung mit geöffneten Browsern. Mit ihrer Hilfe können die Daten und Analyseabläufe dargestellt und verfolgt werden. Die Daten-Browser unterstützen einen Editier-Modus. Damit ist es möglich, die Wissensbasis - auch zur Laufzeit - zu verändern. Sehr zweckmäßig ist beispielsweise die Möglichkeit, eine Bewertungsfunktion anhand einer konkreten Analysesituation zu testen und interaktiv zu verändern. Diese Eigenschaft ist durch die Einbindung des Tcl/Tk-Interpreters (vergleiche Abschnitt 6.6) möglich, was das System zu einem sehr flexiblen Experimentalsystem macht.

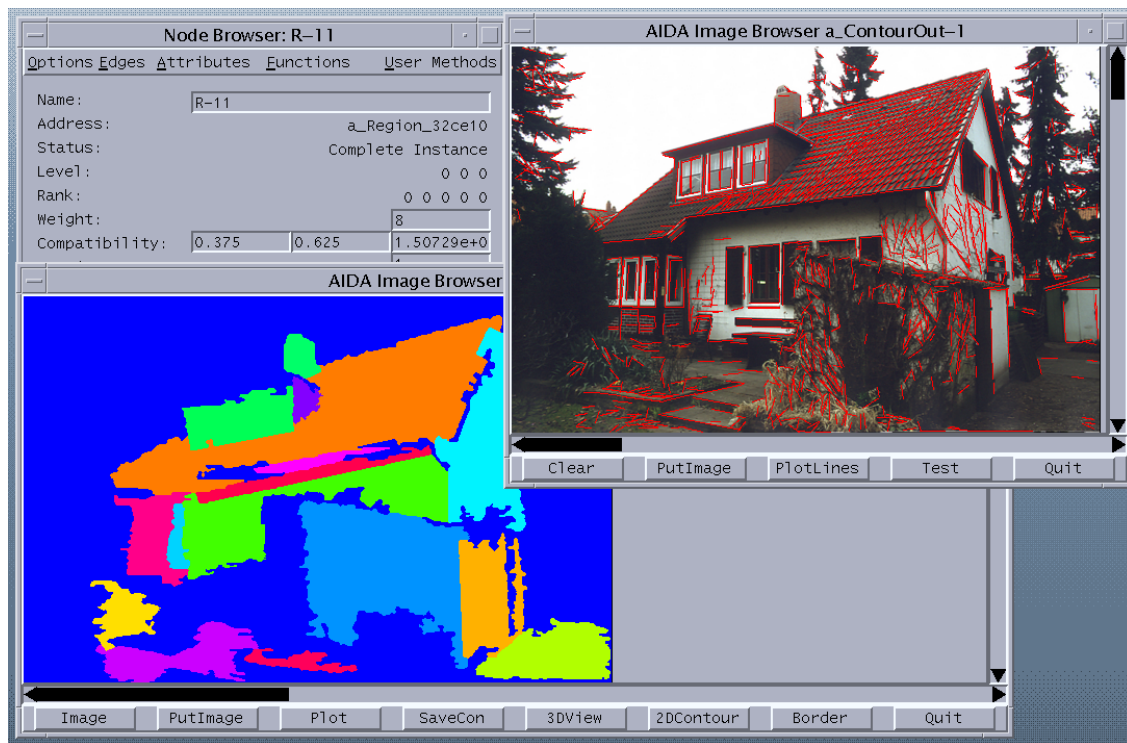


Abbildung 7.9.: Arbeitssitzung mit den Bild-Browsern

7.8. Diskussion der entwickelten Interpretationskomponente

Die entwickelte Interpretationskomponente ist von zentraler Bedeutung für das vorgeschlagene Analysesystem zur Gewinnung einer effizienten Oberflächenbeschreibung von Gebäudeszenen aus Stereobildern.

Da eine Erkennung von Gebäuden wegen der Vielfalt der realen Szenen als Ganzes nicht möglich ist, wird das Problem in Teilprobleme zerlegt. Dazu wurde das strukturierte generische Szenenmodell entwickelt, das eine Zerlegung der Szene in Komponenten erlaubt. Die Interpretation erzeugt daraus Hypothesen für die Komponenten und muß diese darauf unter Berücksichtigung von Unsicherheiten zu einer Gesamtinterpretation zusammenfassen. Das vorliegende System erreicht dies durch ein aufeinander abgestimmtes Bewertungssystem und einen problemunabhängigen Kontrollalgorithmus. Die Verwendung von Top-Down- und Bottom-Up-Phasen erlaubt die Akkumulation von Information, eine Einschränkung des Suchraumes und damit verbunden eine Reduktion der Unsicherheiten.

Für die untersuchten Szenen konnte gezeigt werden, daß die entwickelte Interpretationskomponente eine Deutung des Bildinhaltes findet. Wie die experimentellen Untersuchungen weiterhin zeigen, ist der problemunabhängige Kontrollalgorithmus geeignet, eine Interpretation mit Hilfe des entwickelten generischen Szenenmodells in Form des beschriebenen semantischen Netzes zu liefern. Zur Erhöhung der Effizienz während der Suche wurden in Abschnitt 7.6 weitere wirkungsvolle Prinzipien vorgeschlagen. Diese sind auf andere Probleme übertragbar, müssen jedoch individuell angepaßt werden. Der Systemingenieur muß dabei im allgemeinen einen Kompromiß finden zwischen einer möglichst schnellen Suche und einem robusten Verhalten des Systems.

Zum Gelingen der Interpretation trägt in jedem Fall die Qualität der Wissensbasis eine entscheidende Rolle. Die entwickelte grafische Oberfläche mit den Daten-Browsern und Monitor-Funktionen zum Visualisieren der Analyseabläufe stellen eine notwendige Hilfe für den Entwickler der Wissensbasis dar. Insbesondere für die prozeduralen Inhalte, wie die Berechnungsfunktionen, ist die ständige Überprüfung anhand konkreter Analysesituationen erforderlich. Das heuristische Wissen steuert darüber hinaus die Analysestrategie und somit die Analyseabläufe. Ohne eine entsprechende Visualisierung würde die Entwicklung der Wissensbasis sehr viel umständlicher.

8. Erzeugung der Oberflächenbeschreibung

Die Oberflächenrekonstruktion hat die Aufgabe, die Messungen aus der Bildverarbeitung und die Ergebnisse der Interpretation zu einer konsistenten Oberflächenbeschreibung zusammenzufassen. Zur Oberflächenbeschreibung werden entweder Dreiecksnetze für allgemeine Objekte oder Polygone für die speziellen Gebäudekomponenten verwendet.

Neben den Meßwerten aus der Bildverarbeitung werden topologische Informationen und Randbedingungen aus der Szenenbeschreibung in eine geeignete Form gewandelt und für die Oberflächenrekonstruktion benutzt. Um zu einer vollständigen Oberflächenbeschreibung zu gelangen, werden mehrere Bildansichten integriert. Als letzter Analyseschritt wird eine Texturkarte aus den Eingangsbildern erzeugt.

8.1. Konzept der entwickelten Rekonstruktionskomponente

Die Rekonstruktionskomponente besteht aus drei Teilkomponenten: Der geometrischen Rekonstruktion, der Registrierung zur Fusion von Teilansichten und der Schätzung der Oberflächentextur.

Die geometrische Rekonstruktion berechnet aus der während der Interpretation erzeugten Szenenbeschreibung eine polygonale Oberflächenbeschreibung. Die polygonalen Komponenten der Szenenbeschreibung, die im semantischen Netz vorliegen, werden dazu in eine für die Oberflächenrekonstruktion geeignete Darstellung gewandelt, die im folgenden als relationales, generisches Kostennetz bezeichnet wird. Dieses Kostennetz stellt die in der Bildverarbeitung extrahierten Meßwerte und die aus der Wissensbasis abgeleiteten Randbedingungen in einer einheitlichen Darstellung als Kostenterme in Bezug zueinander. Die Anzahl der Daten und Randbedingungen ist dabei variabel und hängt davon ab, wie viele Meßdaten, Fakten aus der Interpretation und verschiedene Kameraansichten

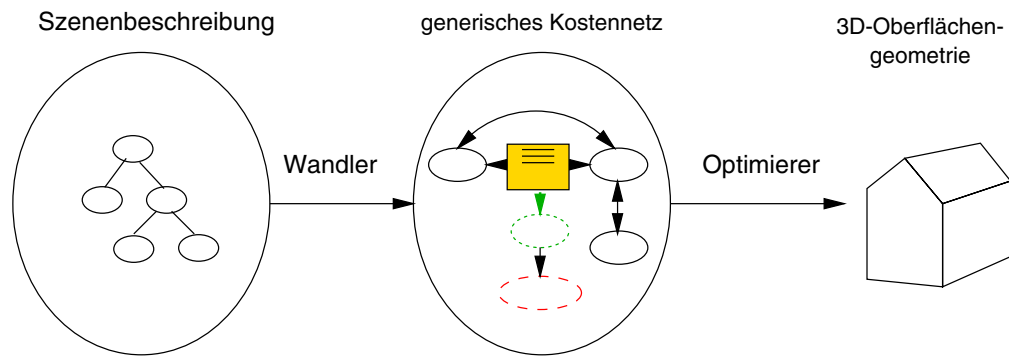


Abbildung 8.1.: Konzept der geometrischen Rekonstruktionskomponente

gesammelt werden konnten.

Durch eine numerische Minimierung der Gesamtkosten wird aus dem Kostennetz eine Oberflächengeometrie berechnet, die möglichst gut zu den Daten und den zusätzlichen Randbedingungen paßt. Die Abbildung 8.1 stellt dies in der Übersicht dar. Der nächste Abschnitt erläutert die Schätzung der geometrischen Oberflächenbeschreibung im Detail.

Um mehrere Kameraansichten zur Gewinnung der Oberflächenbeschreibung nutzen zu können, müssen die einzelnen Kameraansichten auf ein und dasselbe dreidimensionale Koordinatensystem bezogen werden. Dieser Vorgang heißt Registrierung. Das zu lösende Problem ist die Schätzung von Position und Orientierung des Stereosensors. Der Abschnitt 8.3 erläutert diesen Vorgang.

Nach erfolgter Registrierung und Schätzung der Oberflächengeometrie schließt sich als letzter Verarbeitungsschritt die Schätzung der Oberflächentextur an, was in Abschnitt 8.4 beschrieben ist.

8.2. Schätzung der geometrischen Oberflächenbeschreibung

Die Interpretation liefert eine Beschreibung aller Regionen des in der Bildverarbeitung gewonnenen Segmentierungsergebnisses. Nach dem strukturierten generischen Modell aus Abschnitt 6.7 besteht die Szene aus Polygoneiteilflächen für die speziellen Gebäudekomponenten und aus Dreiecksnetzen für Objekte, die nicht näher spezifiziert werden können. Für die Erzeugung eines Dreiecksnetzes aus einer Tiefenkarte können die Verfahren aus [Koch97] und [Rie97] eingesetzt

werden, auf die hier nicht weiter eingegangen wird.

Der folgende Abschnitt beschäftigt sich vielmehr mit der Erzeugung einer polygonalen Oberflächenbeschreibung, die die Randbedingungen aus der Interpretation und die Meßwerte der Bildverarbeitung berücksichtigt.

8.2.1. Relationales generisches Kostennetz zur Darstellung von Randbedingungen und Meßwerten

Um Randbedingungen und Daten einheitlich darzustellen, wurde eine Repräsentation in Form eines relationalen, generischen Kostennetzes entwickelt. Die zu schätzenden Polygone und Hilfsdaten stellen die Knoten dieses Netzes dar. Die Kanten beschreiben die geometrischen Randbedingungen und Relationen zwischen den Knoten.

Knoten des Kostennetzes

Das Kostennetz K sieht als Knoten die Typen *Polygondeskriptor*, *Sichtebene* und *Kamera* vor (Siehe Abbildung 8.2 links). Die Knoten sind als Klassen implementiert und erben von der Klasse *CN-Knoten* die folgende abstrakte Schnittstelle:

CN-Knoten (abstrakte Klasse)	
string	Name
boolean	SchätzungEnabled
integer	AnzahlParameter()
void	SetParameter(integer no, float value)
float	GetParameter(integer no)

Der Knotenname *Name* wird aus den Knoten der Szenenbeschreibung übernommen. Der Slot *SchätzungEnabled* ist ein Flag, das angibt, ob die Parameter des Knotens in der Optimierungsphase variiert, das heißt, geschätzt werden sollen. Die Methode *AnzahlParameter()* wird von jeder nicht abstrakten Klasse überschrieben und liefert die Anzahl von Parametern, die diese Klasse enthält. Mit den Methoden *SetParameter* und *GetParameter* kann auf die Parameter zugegriffen werden.

Die Klasse *Polygondeskriptor* enthält drei Parameter, die eine Ebene E_i im Raum repräsentieren und in der Achsenabschnittsform angegeben werden:

$$E_i(p_1, p_2, p_3) : \frac{1}{p_1} \cdot x + \frac{1}{p_2} \cdot y + \frac{1}{p_3} \cdot z = 1 \quad (8.1)$$

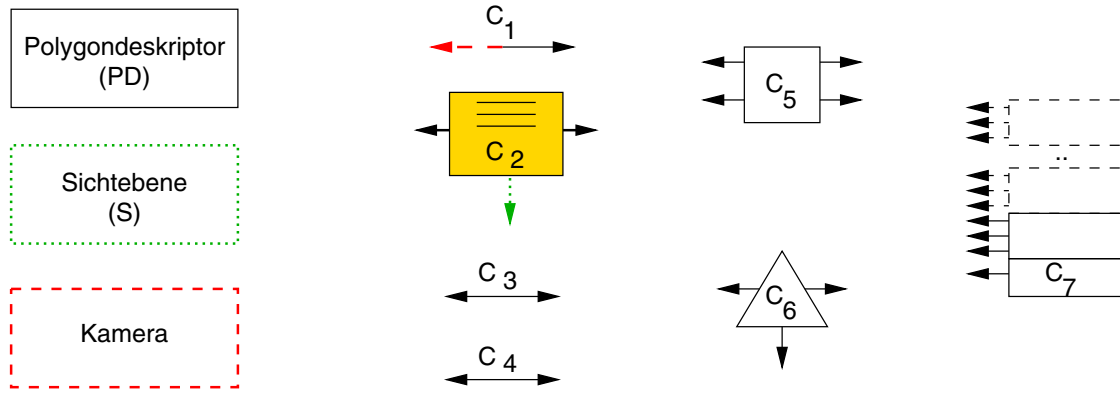


Abbildung 8.2.: Knotentypen (links) und Relationen (rechts) des Kostennetzes

Das zu schätzende Polygon liegt in der Ebene E_i , die durch diese Parameter angegeben werden. Die Begrenzung, das heißt, die Kanten des Polygons werden durch Erzeugung der Schnittgeraden L_j der Ebene E_i mit den Nachbarn gefunden:

$$L_j = \text{Schnitt}(E_i, E_j) \quad E_j \in \mathbf{M}_{\text{nachbarn}}(PD_i) \quad (8.2)$$

Die Objekte PD_i der Klasse *Polygondeskriptor* besitzen dazu eine geordnete Liste der topologischen Nachbarpolygone $\mathbf{M}_{\text{nachbarn}}(PD_i)$, beziehungsweise Polygondeskriptoren. Nach erfolgter Parameterschätzung werden daraus durch Schnittbildung die Schnittkanten und die Eckpunkte der Polygone bestimmt.

Die Klasse *Sichtebene* stellt wie der Polygondeskriptor eine Ebene im Raum dar, die durch den Kamerabrennpunkt und den Endpunkten eines Liniensegments in der Kamerabildebene festgelegt ist (siehe Abbildung 8.3). Die Parameter dieser Ebene werden nicht variiert. Um dies zu kennzeichnen, liefert die Methode *AnzahlParameter()* immer den Wert Null zurück. Die Sichtebene repräsentiert in dem Kostennetz die Liniensegmente, die aus dem Eingangsbild extrahiert wurden.

Die Objekte der Klasse *Kamera* enthalten die inneren Kameraparameter aus der Kalibrierung (siehe Abschnitt 5.1). Die äußeren Kameraparameter werden durch jeweils drei Parameter für Position und Orientierung (drei Rotationswinkel um die Koordinatenachsen) beschrieben. Die Methode *AnzahlParameter()* liefert daher für Objekte dieser Klasse den Wert sechs zurück.

Kanten des Kostennetzes

Die Kanten des Kostennetzes (Abbildung 8.2 rechts) stellen attributierte Relationen zwischen den Knoten dar. Abhängig davon, was die Relationen darstellen, sind die Kanten binär (c_1, c_3, c_4), trinär (c_2 und c_6), quadrinär (c_5) und multinär (c_7). Ferner sind Verbindungen nur mit bestimmten Knotentypen möglich. Die Kante c_1 verbindet einen Polygondeskriptor mit einer Kamera, Kante c_2 verbindet zwei Polygondeskriptoren mit einer Sichtebe. Die folgende Tabelle stellt die verwendeten Relationen zusammen:

Abk.	Beschreibung	Wertigkeit
c_1	Gleichheit der Flächenorientierung und -position	2
c_2	Schnitt von Polygonkante mit Sichtebe	3
c_3	Einhaltung eines Winkels zwischen zwei Ebenen	2
c_4	Parallelität zweier Ebenen	2
c_5	Parallelität zweier Polygonkanten	4
c_6	Gleichheit der Winkel von zwei Ebenen zu einer dritten	3
c_7	Gleichheit der Längen zweier / mehrerer Polygonkanten	$1 + 3 \cdot N$

Wie die Knoten sind die Kanten als Klassen realisiert. Sie besitzen folgende Eigenschaften, die über eine abstrakte Basisklasse definiert sind:

CN-Kante (abstrakte Klasse)	
float	Gewicht
float	Kosten()

Die Methode *Kosten()* berechnet einen Kostenterm aus den Daten, die in den Knoten repräsentiert sind und gegebenenfalls, zum Beispiel als Winkelangabe, in der Kante gesetzt wurden. Jede Kante stellt somit eine Randbedingung (Constraint) dar, die von den zu schätzenden Parametern eingehalten werden soll. Das Gewicht geht als Wichtungsterm w in die Bildung der Gesamtkosten ein (siehe unten).

Die Abbildung 8.3 zeigt ein Beispielnetz mit drei Polygondeskriptoren ($PD-1, PD-2$ und $PD-3$), die mit den links abgebildeten Szenenkomponenten korrespondieren. Zusätzlich enthält das Netz eine Kamera (*Kamera-1*) und eine Sichtebe ($S-1$). Die Kante c_{21} ¹ verbindet die Sichtebe mit den Polygondeskriptoren $PD-1$ und $PD-2$. Diese Relation repräsentiert als Kosten die Abweichung der Sichtebe von der Polygonkante, die durch den Schnitt der Polygondeskriptoren $PD-1$ und $PD-2$ gebildet wird. Dieser Kantentyp stellt somit einen Bezug zu den Meß-

¹Der erste Index des Kantenobjektes kennzeichnet die Art der Relation, der zweite ist eine eindeutige Objekt Nummer.

werten in Form eines Liniensegmentes in der Kamerabildebene her.

Die Kante c_{11} ist der zweite Kantentyp, mit einem Bezug zu den Messungen der Bildverarbeitung: In dem Kantenobjekt werden die Oberflächennormale und -position als Sollwerte gespeichert. Sie werden aus den Tiefenmessungen der Bildverarbeitung gewonnen. Die Methode *Kosten()* berechnet die Abweichung der Flächennormalen und -position des Polygondeskriptors *PD-1* zu diesen Werten.

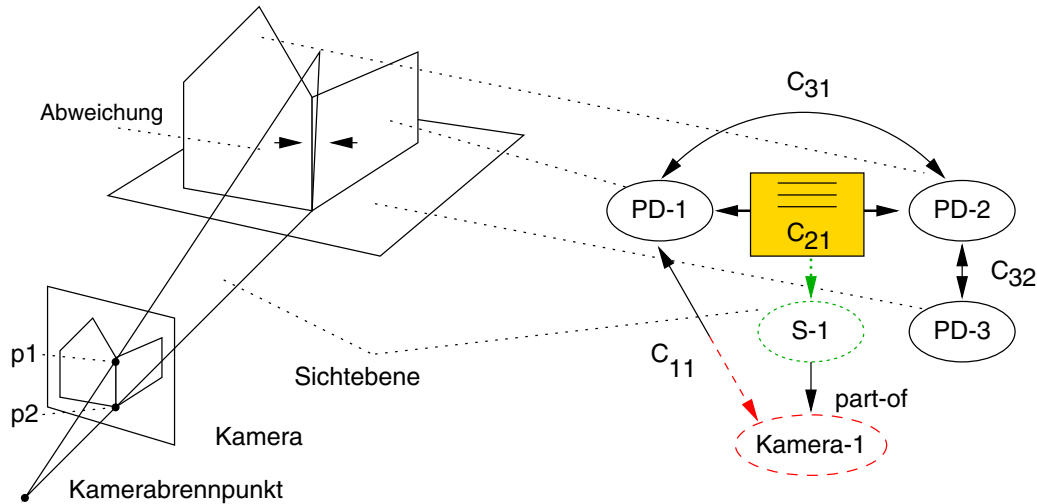


Abbildung 8.3.: Beispiel eines Kostennetzes mit drei Polygondeskriptoren

Die Relation c_3 prüft die Einhaltung des Schnittwinkels zwischen zwei Polygondeskriptoren. Im Beispiel von Abbildung 8.3 ist für die Relationen c_{31} und c_{32} ein Winkel von 90° vorgegeben. Die Relationen c_{5-7} eignen sich für die Darstellung von Symmetrieeigenschaften, die bei Gebäuden sehr ausgeprägt sind.

8.2.2. Erzeugung des Kostennetzes aus der Szenenbeschreibung

Das Kostennetz K entsteht durch Transformation aus der Szenenbeschreibung S . Die Umsetzung $S \rightarrow K$ erfolgt durch die folgenden Schritte:

- Erzeuge für jeden Instanzenknoten P_i vom Typ *Polygon3D* aus S einen Polygondeskriptor PD_i . Kopiere den Namen des Vaterknotens $P_{Vater} \xleftarrow{\text{part-of}} P_i$ in den Slot *Name* des erzeugten Deskriptors. Erzeuge eine geordnete Liste mit den Nachbarn des Deskriptors.

- Erzeuge für jeden Instanzenknoten K_j vom Typ *Kamera3D* eine *Kamera*-Instanz. Kopiere den Namen.
- Erzeuge für jeden Instanzenknoten E_k vom Typ *Kante2D* aus S eine Sichte Ebene S_k . Erzeuge eine *part-of*-Kante zum dazugehörenden *Kamera*-Objekt. Erzeuge eine Kante c_{2l} und deren Zeiger zu den erzeugten Sichte ebenen und den Polygondeskriptoren, die sich aus den Nachfolgern des Instanzenknotens *Verbindung* aus der Szenenbeschreibung S ergeben.
- Rufe für alle Instanzenknoten von S die Methode *ErzeugeConstraint*.

Die speziellen Constraint-Kanten werden mit Hilfe der Methode *ErzeugeConstraint* erzeugt. Diese ist als benutzerdefinierte Funktion in den Knoten des semantischen Netzes implementiert (Abschnitt 6.5.3). So wird beispielsweise die 90°-Schnittwinkelbedingung (c_3) für Häuserwände, durch Aufruf der Methode für die Instanzen des Konzeptes *Verbindung* in das Kostennetz eingefügt. Des weiteren werden Symmetrieeigenschaften für die Konzepte *Haus* und *Dach* mit dieser Methode in die entsprechenden Constraints umgesetzt.

8.2.3. Gewinnung der Oberflächengeometrie durch Minimierung des Kostennetzes

Das Kostennetz enthält mit den Meßdaten, den zusätzlichen Randbedingungen und der topologischen Information alle zur Gewinnung der Oberflächengeometrie notwendigen Daten. Freie Parameter sind die Ebenen E_i , die durch die gesuchten Polygone aufgespannt werden. Die Schätzung dieser Parameter erfolgt durch Minimierung der globalen Kostenfunktion $\Phi_{global}(\vec{x})$:

$$\Phi_{global}(\vec{x}) = \sum_j w_j \cdot \Phi_{ij}(\vec{x}) \rightarrow min. \quad (8.3)$$

Dabei stellen $\Phi_{ij}(\vec{x})$ die Kostenfunktionen der Netzkanten dar und w_j die Gewichte der Relationen. Der Parametervektor $\vec{x}$ ist ein zusammengesetzter Vektor, der sich aus den Komponenten der Ebenenparametern E_i aller zu schätzender Polygone zusammensetzt:

$$\vec{x} = (E_0 | E_1 | \dots | E_N)^T = (p_{01}, p_{02}, p_{03}, p_{11}, \dots, p_{N3})^T \quad (8.4)$$

Die Wichtungsfaktoren w_j in Gleichung 8.3 dienen im wesentlichen einer Skalierung der Einzelkostenterme in die gleiche Größenordnung. Für die resultierende

Oberflächenbeschreibung besonders wichtige Kosten können jedoch hiermit betont werden. Dies ist beispielsweise für die Einhaltung der Kantenschnittbedingung c_2 sinnvoll. Die Übereinstimmung der projizierten Polygonkanten mit den Kanten im Eingangsbild ist notwendig, damit die nachfolgend erzeugten Texturkarten zu der Oberflächengeometrie passen: Verläuft in der Texturkarte eine Kante, die nicht mit der entsprechenden Polygonkante übereinstimmt, so wirkt das resultierende Oberflächenmodell unrealistisch. Das Übereinstimmen der in die Kameraaufnahmen zurückprojizierten Polygonkanten ist daher ein Gütemaß für die geometrische Oberflächenrekonstruktion (siehe Abbildungen 8.4-8.7).

Die Kostenminimierung des Kostennetzes erfolgt durch das Verfahren der konjugierten Gradienten, das sich am vorliegenden Problem als hinreichend schnell und robust erwiesen hat [Weik95]. Im Anhang B sind die Kostenfunktionen der einzelnen Kantentypen angegeben.



Abbildung 8.4.: Drahtnetzdarstellung der initialen Oberflächenbeschreibung

Die Anzahl der Parameter in Gleichung 8.4 kann für komplexe Szenen groß werden. Um die Anzahl der gleichzeitig zu schätzenden Parameter möglichst klein zu halten, ist ein hierarchisches Vorgehen bei der Schätzung der Parameter sinnvoll. Dazu wird zuerst die Oberflächenbeschreibung der „Hauptkomponenten“ der Szene geschätzt. Diese zeichnen sich dadurch aus, daß sie in der *part-of*-Hierarchie des semantischen Netzes weiter an dem Wurzelknoten der Szenenbeschreibung² als untergeordnete Komponenten liegen. Dies entspricht der Kompositionsstufe, die jedem Knoten des semantischen Netzes zugeordnet wird (vergleiche Abschnitt 7.2.1 zur Berechnung der Prioritäten).

²Der (Sub-)Graph der *part-of*-Hierarchie ist ein Baum

Die Kompositionsstufe kann daher zur hierarchischen Schätzung genutzt werden: Zuerst werden Polygondeskriptoren der Komponenten mit einer kleinen Kompositionsstufe erzeugt und die Parameter geschätzt. Danach werden die Polygondeskriptoren durch Rücksetzen des Flags *SchätzungEnabled* markiert und damit die Parameter bei der nächsten Minimierung nicht mehr verändert. Als nächstes wird das Kostennetz um Komponenten erweitert, die die nächst größere Kompositionsstufe besitzen, und durch Minimierung werden darauf die Parameter der dazugekommenen Polygondeskriptoren geschätzt. Der Vorgang kann im Prinzip iterativ für eine beliebige Tiefe von Kompositionsstufen durchgeführt werden. Für die Gebäudeszenen hat sich alternativ ein zweistufiger Ansatz bewährt, wobei zuerst die obligatorischen und im zweiten Schritt die optionalen Teile erzeugt und geschätzt werden.



Abbildung 8.5.: Drahtnetzdarstellung nach Optimierung

Die Abbildungen 8.4-8.7 zeigen die Ergebnisse der einzelnen Stufen zur Gewinnung der geometrischen Oberflächenbeschreibung. Die Abbildung 8.4 stellt die initiale Oberflächenbeschreibung der obligaten Komponenten dar, die allein unter Verwendung der topologischen Information und den Meßwerten aus der Szenenbeschreibung entstanden sind. Das heißt, es wurden keine geometrischen Randbedingungen verwendet³. Aufgrund der Meßfehler, die aus der Kalibrierung, der Tiefenmessung und der Segmentierung stammen, sind deutliche Abweichungen zu dem Bildinhalt des Eingangsbildes zu erkennen.

Die Abbildung 8.5 zeigt das Ergebnis der Kostenminimierung. Die rückprojizierten Polygonkanten stimmen recht gut mit dem Bildinhalt des Eingangsbildes

³Die Polygone werden in der Rückprojektion als Dreiecke dargestellt



Abbildung 8.6.: Drahtnetzdarstellung mit optionalen Teilen nach Optimierung

überein. Nach diesem ersten Minimierungsschritt werden alle Knoten des Kostennetzes, die obligatorische Komponenten darstellen, bezüglich der weiteren Minimierung gesperrt (durch Rücksetzen des Flags *SchätzungEnabled*). Die Abbildung 8.6 zeigt das Ergebnis nach Einfügen und Schätzung der optionalen Teile.

8.3. Schätzung der Kamerabewegung (Registrierung)

Mit Hilfe einer Ansicht und entsprechendem Vorwissen können, wie die Abbildungen oben zeigen, Gebäude rekonstruiert werden. Um die verdeckten Komponenten mit zu erfassen, müssen jedoch mehrere Ansichten verwendet werden. Dazu ist es notwendig, die Kameraparameter Position und Orientierung für jede neue Ansicht zu schätzen und auf ein gemeinsames (Welt-)Koordinatensystem zu beziehen. Dieser Vorgang wird auch als Registrierung bezeichnet.

Der Ansatz zur Schätzung der Kameraparameter einer neuen Ansicht setzt voraus, daß eine erste Oberflächenbeschreibung des abgebildeten Objektes vorhanden ist und eine Korrespondenz der Polygonkanten zu den entsprechenden Kanten im Bild hergestellt werden kann. Der Vorgang der Korrespondenzzuordnung wird bislang manuell unterstützt durchgeführt.

Für die neue Ansicht werden ein Kameraobjekt und die Bildkanten als Sichtebenen in das Kostennetz eingeführt. Die Korrespondenzen zu den vorhandenen

Polygonkanten werden, wie in Abbildung 8.3, mit Hilfe der c_2 -Kanten ins Kostennetz eingetragen. Zur Schätzung der Kameraparameter Position und Orientierung wird Gleichung 8.3 abgewandelt:

$$\Phi_{global}(\vec{x}') = \sum_j w_j \cdot \Phi_{ij}(\vec{x}') \rightarrow \min. \quad (8.5)$$

Der Parametervektor $\vec{x}'$ setzt sich hier aus den Kameraparametern $\vec{T}$ und $\vec{R}$ zusammen:

$$\vec{x}' = (\vec{T}|\vec{R})^T \quad (8.6)$$

Dabei entspricht $\vec{T}$ dem C-Vektor der CAHV-Kamerarepresentation. Der Vektor $\vec{R}$ beschreibt die Orientierung der Kamera als drei Rotationswinkel um die Koordinatenachsen.



Abbildung 8.7.: Drahtnetzdarstellung nach erfolgter Registrierung

Zur Schätzung der Kameraparameter werden im Kostennetz die Polygondeskriptoren von der Minimierung ausgenommen, indem die Flags *SchätzungEnabled* für die entsprechenden Knoten zurückgesetzt werden. Statt dessen wird das Flag für den zu schätzenden Kameraknoten gesetzt. Die Kostenminimierung ergibt die gesuchten Kameraparameter.

Nach erfolgter Registrierung können bislang verdeckte Komponenten des abgebildeten Objektes als entsprechende Polygondeskriptoren in das Kostennetz aufgenommen und geschätzt werden. Die Schätzung der Polygone kann die Daten

aus mehreren Ansichten verwerten. Die Abbildung 8.7 zeigt die Drahtgitterdarstellung der vollständigen Oberflächenbeschreibung der Szene „Restaurant“ nach erfolgter Registrierung in der neuen Ansicht.

8.4. Schätzung der Oberflächentextur

Als letzter Schritt der Oberflächenrekonstruktion wird eine Texturkarte (vergleiche Abschnitt 4.1.2) der Oberfläche angelegt, die die Oberflächenfarbschattierung als zweidimensionales Bild festhält. Die Textur wird dabei aus den Kamerabilddern unter Zuhilfenahme der Kameraparameter und der zuvor erzeugten geometrischen Oberflächenbeschreibung gewonnen.

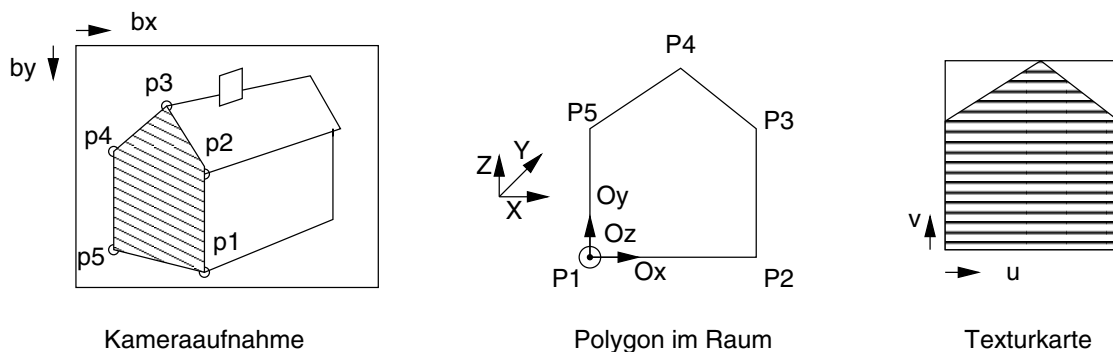


Abbildung 8.8.: Abbildung von Koordinatensystemen während der Texturierung

Zur Beschreibung der Oberflächentextur wird eine Texturkarte angelegt, die einer Abwicklung der Objektoberfläche entspricht. Da das zugrunde liegende Polygon bei der Projektion in die Kamera perspektivisch abgebildet wurde, muß zur Erzeugung einer Abwicklung eine perspektivische Entzerrung des Bildausschnittes vorgenommen werden.

Das in Abbildung 8.8 rechts dargestellte (u,v) -Koordinatensystem der Texturkarte entspricht den ersten beiden Komponenten des Objektkoordinatensystems des Polygons, das seinen Ursprung im ersten Eckpunkt hat und dessen ersten beiden Komponenten dieselbe Ebene wie das Polygon aufspannt. Die (u,v) -Koordinaten sind üblicherweise auf das Intervall $[0,1]$ normiert [VRML97]. Das heißt, der äußere rechte Punkt in der Texturkarte in Abbildung 8.8 rechts besitzt die Koordinaten $(1,1)$, der Ursprung der Texturkarte die Koordinaten $(0,0)$.

Die perspektivische Abbildung des (u,v) -Koordinatensystems in das Koordinatensystem der Kamerabildebene erfolgt durch folgende Abbildung:



Abbildung 8.9.: Texturkarte für Dachfläche aus einer Ansicht



Abbildung 8.10.: Gebäudemodell mit kombinierten Texturkarten

$$b_x = \frac{a \cdot u + b \cdot v + c}{g \cdot u + h \cdot v + i} \quad (8.7)$$

$$b_y = \frac{d \cdot u + e \cdot v + f}{g \cdot u + h \cdot v + i} \quad (8.8)$$

Zur Bestimmung der acht Koeffizienten ($a-i$) werden die (normierten) Eckkoordinaten der Texturkarte $[(0,0),(1,0),(1,1),(0,1)]$ erst in Objektkoordinaten des Polygons und dann in Weltkoordinaten umgerechnet und anschließend mit Hilfe des Kameramodells in die Bildebene projiziert. Nach Einsetzen der Wertepaare in die Gleichungen 8.7 und 8.8 berechnen sich die Koeffizienten durch Lösen des so gewonnenen Gleichungssystems.

Beim Füllen der Texturkarte entstehen im allgemeinen Zwischenrasterwerte, die ein Resampling erfordern. Zur Berechnung der Pixel wurde hier eine bilineare Interpolation eingesetzt. Ferner sind Verdeckungen zu berücksichtigen. Dazu wird mit Hilfe einer virtuellen Kamera ein synthetisches Bild der gewonnenen Ober-

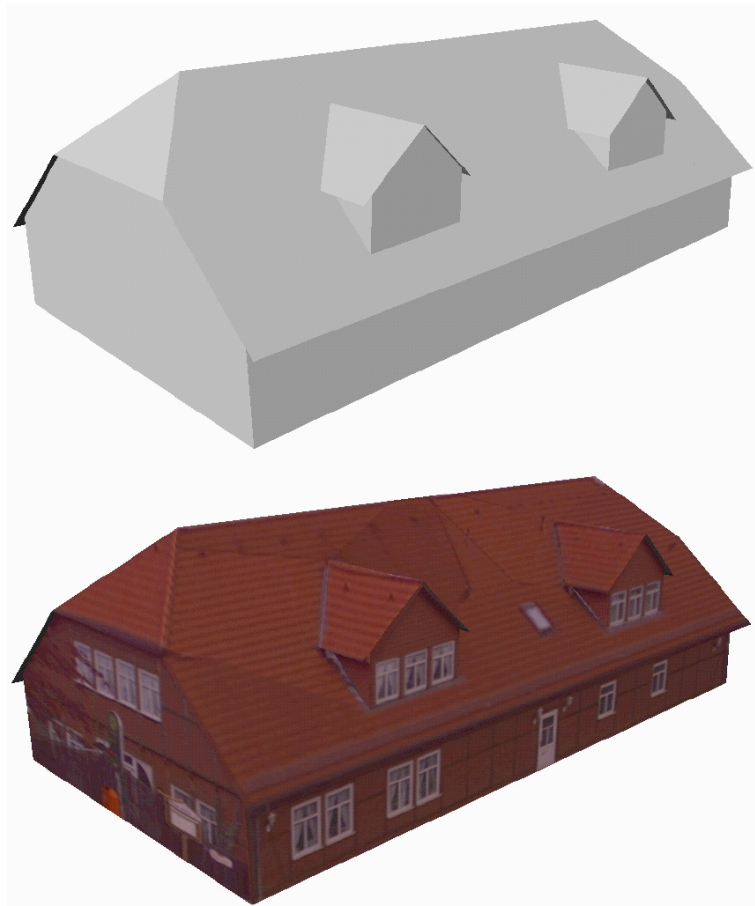


Abbildung 8.11.: *Visualisierung der Oberflächenbeschreibung „Restaurant“*

flächenbeschreibung erzeugt und mit dem Z-Buffer-Algorithmus [Fol92] überprüft, ob ein adressiertes Pixel im (Original-)Kamerabild tatsächlich auf der gerade betrachteten Polygonoberfläche liegt oder ob diese durch andere Flächen verdeckt ist [Col98]. Die Abbildung 8.9 zeigt die Texturkarte einer Dachhälfte aus der Szene „Restaurant“, die aus einer Ansicht (Abbildung 8.6) gewonnen wurde.

Durch Akkumulation der Texturkarten aus den Teilansichten [Col98, Grau94] kann eine vollständige Texturkarte gewonnen werden. Die Abbildung 8.10 zeigt eine Visualisierung der aus zwei Kameraansichten gewonnenen Oberflächenbeschreibung. Die Abbildungen 8.11 + 8.12 zeigt eine Darstellung der Szene „Haus“ und „Restaurant“ in schattierter und texturierter Darstellung.

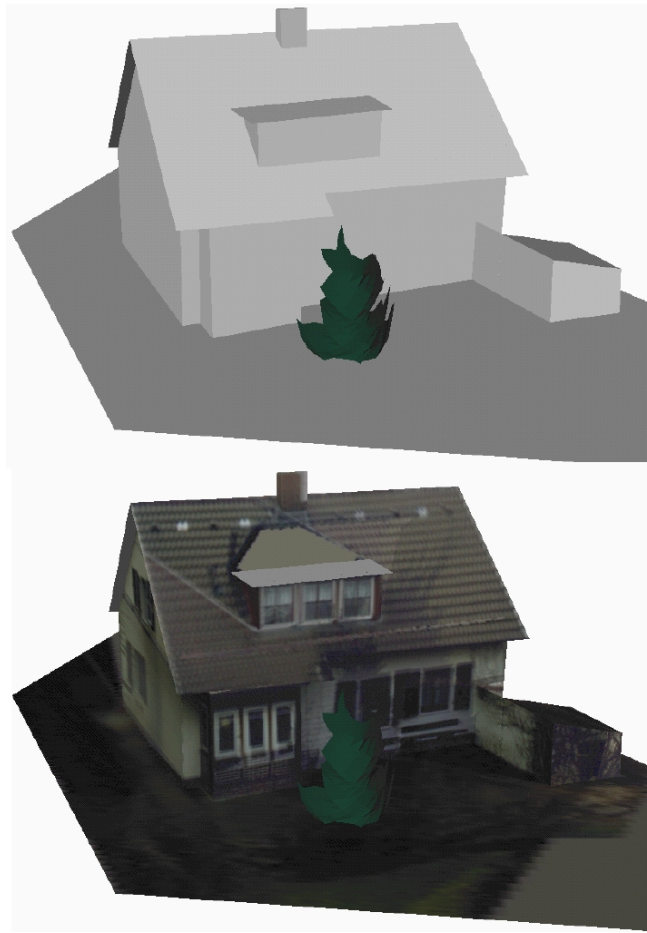


Abbildung 8.12.: Visualisierung der Oberflächenbeschreibung „Haus“

8.5. Diskussion des entwickelten Rekonstruktionsmoduls

Zur geometrischen Rekonstruktion der Objektoberflächen wurde eine generische Repräsentation in Form eines Kostennetzes entwickelt, die eine einheitliche Darstellung von Meßdaten und Randbedingungen für die Rekonstruktion polygonaler Oberflächen erlaubt. Diese Repräsentation ermöglicht es, je nach Analysesituation die verschiedenen Fakten für eine Teilfläche anzusammeln. Dadurch wird der Informationsverlust durch die Projektion vom Raum in die Bildebene ausgeglichen.

Der Gewinn durch Einbringung von Vorwissen ist prinzipiell um so größer, desto spezieller die Objekt beziehungsweise deren Komponenten sind. So können

beispielsweise für die betrachteten Gebäude Symmetrieeigenschaften vorteilhaft für die Oberflächenrekonstruktion genutzt werden. Durch Einbringung von entsprechendem Vorwissen ist dabei die Verwendung von Stereobildern nicht unbedingt erforderlich. In diesem Falle müßten die aus der Stereoauswertung fehlenden Meßdaten (Tiefenmessungen) durch entsprechende Randbedingungen im Kostennetz kompensiert werden. Zur Auswertung würden dann als Daten beispielsweise Liniensegmente aus monokularen Kameraansichten ausreichen.

Zur Berechnung unter Verwendung aller Daten wird eine globale Kostenfunktion der für die Optimierung ausgewählten Knoten des Kostennetzes minimiert. Dabei können mit derselben Darstellung sowohl die Form der Oberfläche als auch die Kameraparameter Position und Orientierung berechnet werden. Um komplexe Objekte behandeln zu können, ist ein hierarchischer Ansatz sinnvoll.

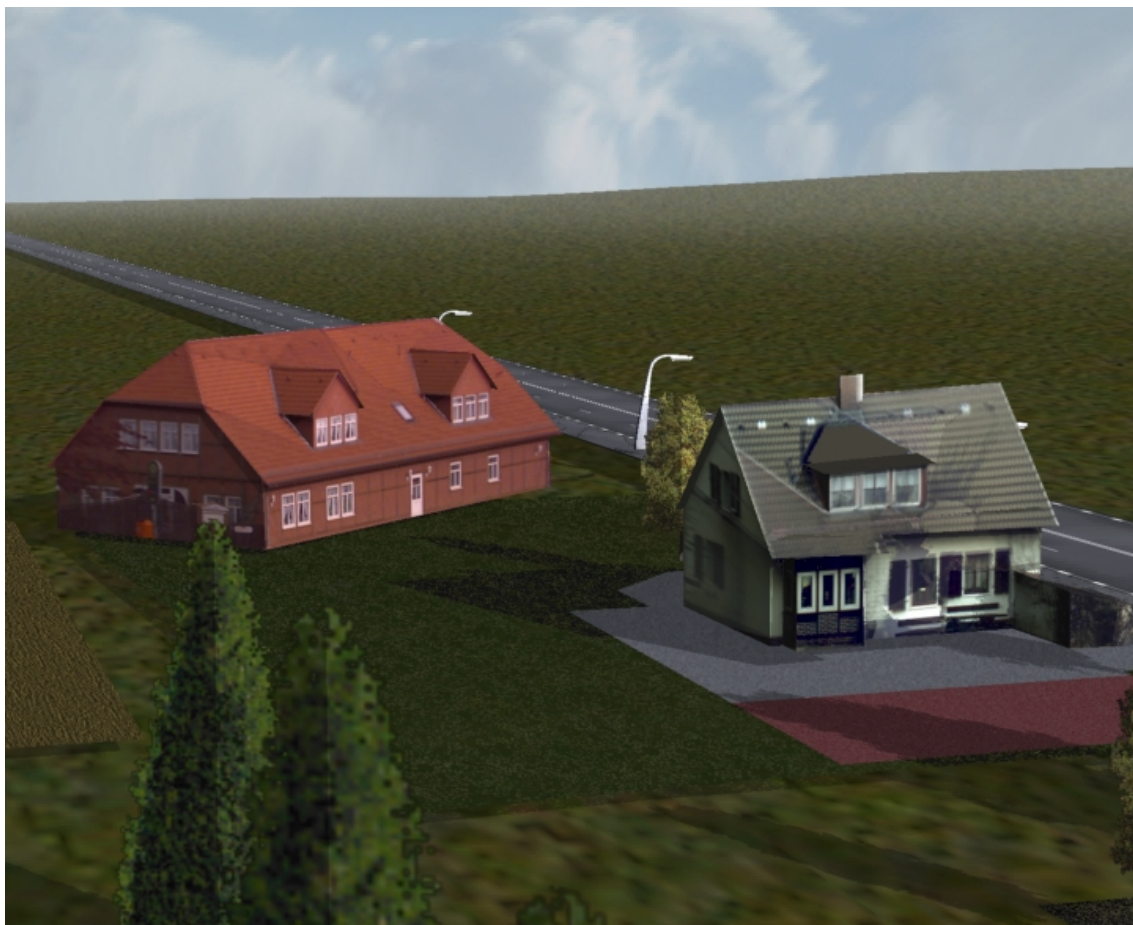


Abbildung 8.13.: Virtuelle Szene mit den gewonnenen Gebäudemodellen

Die Ergebnisse der Oberflächenrekonstruktion führen zu Beschreibungen mit ei-

ner sehr geringen Anzahl von 3D-Primitiven (Polygonen), was insbesondere bei zeitkritischen Visualisierungsanwendungen, wie Fahr- und Flugsimulatoren, gefordert ist. Die subjektive Qualität ist aufgrund der Verwendung von Texturkarten trotzdem sehr gut. Die Abbildung 8.13 zeigt als Anwendungsbeispiel eine virtuelle Szene, in der die gewonnenen Gebäudemodelle zusammengestellt wurden.

9. Zusammenfassung

Diese Arbeit beschreibt ein Analysesystem zur automatischen dreidimensionalen Rekonstruktion von Gebäudeoberflächen aus stereoskopischen Kameraaufnahmen. Bei der Analyse wird durch eine Interpretation szenenspezifisches Wissen für die einzelnen Objektkomponenten ausgewählt und zusammen mit Meßdaten, die aus mehreren Kamerabildern gewonnen werden, zu einer effizienten und konsistenten Oberflächenbeschreibung vereint.

Die einzelnen 3D-Modelle müssen für die Verwendung in Sichtsystemen sehr effizient dargestellt sein, da die Verarbeitungskapazität begrenzt ist. Zum einen muß dabei die Anzahl der zur geometrischen Beschreibung verwendeten Oberflächenprimitive möglichst gering sein, zum anderen soll die Visualisierung trotzdem realistisch wirken. Bei der herkömmlichen Gewinnung durch CAD-Systeme bringt ein Operator Wissen über Art und Aufbau der zu modellierenden Objekte ein. In den bisherigen automatisierten, bildgestützten Verfahren ist dies nur ansatzweise der Fall. Die vorliegende Arbeit stellt daher einen Ansatz zur Darstellung und automatischen Nutzung von Vorwissen für die gestellte Rekonstruktionsaufgabe vor.

Zur einheitlichen Darstellung der für das System erforderlichen Daten und des szenenspezifischen Vorwissens wurde ein generisches strukturiertes Szenenmodell entwickelt, das auf semantische Netze zur Wissensrepräsentation abgebildet wurde. Mit Hilfe der semantischen Netze können die Komponenten der Szene in ihren geometrischen, photometrischen, topologischen und strukturellen Eigenschaften beschrieben werden. Die explizite Darstellungsform erlaubt es, das Analysesystem durch Abänderung der Wissensbasis an andere, ähnliche Aufgaben zu adaptieren, ohne daß Änderungen an den Verarbeitungskomponenten notwendig werden.

Als erster Schritt der Analyse wird aus einem mit einer vorkalibrierten Stereokameraanordnung aufgenommenen Stereobildpaar durch ein stereoskopisches Tiefenschätzverfahren für jeden Bildpunkt die Szenentiefe geschätzt. Eine Segmentierung zerlegt darauf den Bildinhalt in Regionen, die sich stückweise als Ebenen in der zuvor gewonnenen Tiefenkarte ausprägen. Zusätzlich erfolgt eine Extraktion von geradlinigen Konturen in dem Kamerabild.

Ausgehend von den Bildregionen aus der Segmentierung führt die Interpretationskomponente, unter Verwendung der Wissensbasis schrittweise eine Bedeutungszuweisung durch. Dabei werden aus den in der Wissensbasis abgelegten Konzepten, die eine intensionelle Beschreibung der zu erwartenden Komponenten der Szene darstellen, Hypothesen erzeugt. Nach erfolgter Verifikation einer Hypothese anhand der Bildmerkmale und der weiteren Eigenschaften, die in der Wissensbasis abgelegt sind, wird eine Instanz aus dem ursprünglichen Konzept erzeugt. Treten bei der Erzeugung von Hypothesen Mehrdeutigkeiten auf, so werden diese als konkurrierende Hypothesen verfolgt und bewertet. Ein problemunabhängiger Kontrollalgorithmus, der auf dem A*-Algorithmus basiert, findet die bestbewertete Alternative. Verschiedene Suchstrategien wurden entwickelt, um den Suchaufwand möglichst gering zu halten. Nach erfolgreicher Interpretation liegt eine strukturelle und attributierte Beschreibung der meßbaren Szenenkomponenten vor.

Die Ergebnisse der Interpretation für die ausgewählten Beispielszenen zeigen, daß die erstellte Wissensbasis und die entwickelte Analysekomponente in der Lage sind, komplexe Gebäudeszenen zu analysieren. Für den praktischen Einsatz ist die Reduktion des Suchaufwandes in dem problemunabhängigen Kontrollalgorithmus bedeutsam. Durch die vorgeschlagenen Maßnahmen ist eine Reduktion der expandierten Hypothesen um ca. das 400-1500-fache möglich, ohne die Interpretation zu verfälschen. Durch Verwendung einer Restkostenabschätzung in der Bewertung der Hypothesen ist im Grenzfall sogar ein lineares Suchverhalten möglich, wenn die Suche nur die jeweils „richtigen“ Hypothesen verfolgt. Dieses Vorgehen ist jedoch nicht empfehlenswert, da die Gefahr besteht, daß der Suchalgorithmus nicht die optimale Interpretation hinsichtlich der Bewertung findet.

Für die Oberflächenrekonstruktion wird die Szenenbeschreibung aus der Interpretation in ein Kostennetz umgewandelt. Dieses besteht aus Knoten, die die zu rekonstruierenden Teilflächen durch Polygonflächen darstellen. Die Kanten des Kostennetzes repräsentieren geometrische Randbedingungen oder Relationen zu Bildmerkmalen, insbesondere den extrahierten Bildkanten. Mit Hilfe des Kostennetzes lassen sich Daten und Randbedingungen aus einer oder mehreren Bildansichten in einer einheitlichen Darstellung in Beziehung zueinander setzen. Die Berechnung der Oberflächengeometrie erfolgt durch Minimierung der Gesamtkosten des Kostennetzes. Mit Hilfe des Kostennetzes können neben den Oberflächenparametern auch Kameraparameter einer Aufnahme geschätzt werden. Diese Kameraregistrierung erfordert eine korrekte Zuordnung korrespondierender Bildkonturen, die zur Zeit manuell unterstützt vorgenommen wird. Die geometrische Oberflächenbeschreibung wird abschließend durch Gewinnung von Texturkarten aus den Kamerabildern vervollständigt.

Die von der Oberflächenrekonstruktion berechnete Objektgeometrie ist durch

die Verwendung der stereoskopischen Tiefenmessungen zusammen mit den ausgewählten geometrischen Randbedingungen und der Konturinformation kantentreu. Das heißt, bei Rückprojektion der gewonnenen Oberflächenbeschreibung in die Kameraaufnahmen stimmen die Polygonkanten mit den tatsächlichen Objektkanten gut überein. Dieses Kriterium ist insbesondere wichtig, um korrekte Texturkarten erzeugen zu können. Die aus dem Ansatz resultierenden, texturierten Oberflächenmodelle erreichen ein fotorealistisches Aussehen und sind aufgrund der niedrigen Anzahl von 3D-Polygonen gut für den Einsatz in Sichtsystemen geeignet. Der Grad der Detailliertheit kann dabei frei in der Wissensbasis eingestellt werden. Aufgrund des wissensbasierten Ansatzes können dabei insbesondere bedeutsame Details durch Einfügen entsprechender Objektbeschreibungen in die Wissensbasis berücksichtigt werden.

Literaturverzeichnis

- [Besl92] Besl, P., McKay, N.D., *A method for registration of 3-d shapes*, IEEE Tr. on PAMI, vol. 14, no. 2, 1992.
- [Bhan87] Bhanu, B., Ho, C.-C., *CAD-Based 3D Object Representation for Robot Vision*, IEEE Computer, Aug. 1987.
- [Bueck98] Bückner, J., Grau, O., Growe, S., Tönjes, R., *AIDA - Automatic Image Data Analyser*, <http://www.tnt.uni-hannover.de/soft/tnt/aida/overview.html>, 1998.
- [Col98] Colios, Ch., *Entwurf eines Verfahrens zur Erzeugung perspektivisch entzerrter Texture-Maps aus mehreren Ansichten für 3D-Computer-Grafik-Modelle*, Studienarbeit, Universität Hannover, 1998.
- [Cox92] Cox, I., Hingorani, S., Maggs, B., Rao, S., *Stereo without Regularisation*, NEC Research Institute, Princeton, NJ, USA, Oct. 1992.
- [Chen93] Chen, X., Schmitt, F., *Vision-Based Construction of CAD Models from Range Images*, Proc. of ICCV'93, Berlin, Germany, 1993.
- [Chri95] Christopher, W., *Objectify - turn C++ classes into Tcl objects*, <ftp://ftp.aud.alcatel.com/tcl/code/objectify-3.1.tar.gz>.
- [Deb96] Debevec, P. E., Taylor, C. J., Malik, J., *Modeling and Rendering Architecture from Photographs: A hybrid geometry- and image-based approach*, Technical Report UCB//CSD-96-893, Computer Science Division, University of California at Berkeley, Jan. 1996.
- [Dhon89] Dhond, U. R., Aggarwal, J.K., *Structure from Stereo - A Review*, IEEE Tr. on Systems, Man, And Cybernetics, Vol. 19, No. 6, Nov./Dez. 1989.
- [Fal94] Falkenhagen, L., *Depth Estimation from Stereoscopic Image Pairs Assuming Piecewise Continuous Surfaces*, European Workshop on Combined real and synthetic image processing for broadcast and video production, 23-24.11.1994, Hamburg.

- [Fal97] Falkenhagen, L., *Block-Based Depth Estimation from Image Triples with Unrestricted Camera Setup*, IEEE Workshop on Multimedia Signal Processing, June 23 - 25, 1997, Princeton, New Jersey, USA.
- [Fal97a] Falkenhagen, L., *Hierarchical Block-Based Disparity Estimation Considering Neighbourhood Constraints*, International workshop on SNHC and 3D Imaging, September 5-9, 1997, Rhodes, Greece.
- [Fis97] Fischer, A., Kolbe, T. H., Lang, F., *Integration of 2D and 3D Reasoning for Building Reconstruction using a Generic Hierarchical Model*, Proc. of Semantic Modeling for the Acquisition of Topographic Information from Images and Maps SMATI '97, Ed. Förstner/Plümer, Birkhäuser Verlag, pages 159-180, Mai 1997.
- [Fish93] Fischl, J., Miller, B., Robinson, J., *Parameter tracking in a muscle-based analysis/synthesis coding system*, Picture coding Symposium (PCS'93), Lausanne, Switzerland, No. 2.3, March 1993.
- [Fol92] Foley, J. D., van Dam, A., van Dam, A., Feiner, S.K., Hughes, J.F., *Computer Graphics: Principles and Practice*, Second Edition, Addison-Wesley Publishing Company, 1992.
- [For93] Foresti, G.L., Murino, V., Regazzoni, C. S., Vernazza, G., *Distributed spatial reasoning for multisensory image interpretation*, Signal Processing 32, pp. 217-255, Mai 1993.
- [Grau94] Grau, O., *Bewegungskompensierte Akkumulation verrauschter Bildsequenzen*, Mustererkennung 1994. Veröffentlichungen des 16. Symposium der DAGM und 18. Workshop der ÖAGM. 21.-23. September 1994 Wien.
- [Grau95] Grau, O., *Ein Szeneninterpretationssystem zur Modellierung dreidimensionaler Körper*, 17. DAGM-Symposium Mustererkennung '95, Bielefeld, 13.-15. Sep. 1995.
- [Grau96] Grau, O., *Definitions of 3-D Model Objects*, Technical Report, ACTS-PANORAMA Project Deliverable AC092/UH/DS/P/004, Aug. 1996.
- [Grau97a] Grau, O., *A Scene Analysis System for the Generation of 3-D Models*, IEEE Proc. of the Int. Conf. on Recent Advances in 3D Imaging and Modelling, Ottawa, Canada 12-15 May 1997.
- [Grau97] Grau, O., *Representation of Temporal Changes of Flexible 3-D Objects*, Int. Workshop on Synthetic - Natural Hybrid Coding and Three Dimensional Imaging (IWSNHC3DI'97). 5.-9. September 1997, Rhodos.

- [Gro94] Growe, S., *Entwurf einer Kontrollstrategie zur wissensbasierten Modellierung dreidimensionaler Objekte unter Verwendung von semantischen Netzen*, Diplomarbeit Universität Hannover, 1994.
- [Hara92] Haralick, R.M., Shapiro, L.G., *Computer and Robot Vision*, Addison-Wesley Publishing Company, Vol.I, 1992.
- [Hara93] Haralick, R.M., Shapiro, L.G., *Computer and Robot Vision*, Addison-Wesley Publishing Company, Vol.II, 1993.
- [Hend88] Henderson, T., Weitz, E., Hansen, C., Mitiche, A., *Multisensor Knowledge Systems: Interpreting 3D Structure*, The Int. Journal of Robotics Research, Vol. 7, No. 6, Dec. 1988.
- [Hil97] Hildebrand, A., *Von der Photographie zum 3D-Modell*, Dissertation, Springer Verlag, 1997.
- [Hos92] Hoschek, J., Lasser, D., *Grundlagen der geometrischen Datenverarbeitung*, B.G. Teubner, Stuttgart, 1992.
- [Hoff87] Hoffman, R.L., Jain, A.K., *Segmentation and Classification of Range Images*, IEEE T-PAMI, Vol. 9, no. 5, Sept. 1987.
- [Hoov96] Hoover, A., et al., *An Experimental Comparison of Range Image Segmentation Algorithms*, IEEE T-PAMI, Vol. 18, No. 7, July 1996.
- [Kapp89] Kappei, F., *Modellierung und Rekonstruktion bewegter dreidimensionaler Objekte in einer Fernsehbildfolge*, Dissertation, Universität Hannover, 1989.
- [Kern78] Kerningham, B., W., Ritchie, D., M., *The C Programming Language*, Prentice-Hall, 1978.
- [Koch96] Koch, R., *Surface Segmentation and Modeling of 3-D Polygonal Objects from Stereoscopic Image Pairs*, ICPR Conference '96, Wien, Aug. 1996.
- [Koch97] Koch, R., *Automatische Oberflächenmodellierung starrer dreidimensionaler Objekte aus stereoskopischen Rundum-Ansichten*, Dissertation, Universität Hannover, VDI-Verlag, Reihe 10, Nr. 499, 1997.
- [Kum91] Kummert, F., *Flexible Steuerung eines sprachverstehenden Systems mit homogener Wissensbasis*, Dissertation, Universität Erlangen-Nürnberg, 1991.
- [Kum93] Kummert, F., Niemann, H., Prectel, R., Sagerer, G.: *Control and explanation in a signal understanding environment*, Signal Processing, Vol. 32, Mai 1993.

- [Lang96] Lang, F., Förstner, W., *Surface Reconstruction of Man-Made Objects using Polymorphic Mid-Level Features and Generic Scene Knowledge*, Zeitschrift für Photogrammetrie und Fernerkundung, Wichmann Verlag, Heft 6, Seiten 193-202, Nov. 1996 .
- [Lej96] Lejeune, A., Ferrie, F.P., *Finding the Parts of Objects in Range Images*, CVGIP: Image Understanding, Vol. 61, No. 2, pp. 230-247, Sept. 1996.
- [Lem88] Lemmens, M.J.P.M., *A survey on stere matching techniques*, Int. Archives of Photogrammetry and Remote Sensing, vol. 27, Comm. V., pp. 11-23. 1988.
- [Leon93] Leonardis, A., Gupta, A., Bajcsy, R., *Segmentation of Range Images as the Search for Geometric Parametric Models*, Int. J. of Computer Vision, Vol. 14, no. 3, Nov. 1993.
- [Lie89] Liedtke, C.-E., Ender, M., *Wissensbasierte Bildverarbeitung*, Springer-Verlag, 1989.
- [Lie95] Liedtke, C.-E., Grau, O., Growe, S., *Use of Explicit Knowledge for the Reconstruction of 3-D Object Geometry*, 6th Int. Conf. Computer Analysis of Images and Patterns CAIP'95. 6.-8.Sept. 1995 Prag.
- [Lie97] Liedtke, C.-E., Bückner, J., Grau, O., Growe, S., Tönjes, R., *AIDA: A System for the Knowledge Based Interpretation of Remote Sensing Data*, 3rd International Airborne Remote Sensing Conference, 7.-10. Juli 1997, Copenhagen, Dänemark.
- [Luo97] Luong, Q.-T., Faugeras, O. D., *Self-calibration of a moving camera from point correspondences and fundamental matrices*, Int. J. Comput. Vis. (NL), vol.22, no.3, p.261-89 (Aug.1997).
- [Mat85] Matsuyama, T., Hwang, V.S.-S., *SIGMA: A framework for image understanding - Integration of bottom-up and top-down analysis*, In Proc. of the 9th Int. Joint Conf. on Artificial Intelligence, 1985.
- [Mat90] Matsuyama, T., Hwang, V.S.-S., *SIGMA: A Knowledge-Based Image Understanding System*, Plenum Press, New York, 1990.
- [McKe85] McKeown, D.M., Harvey, W.A., McDermott, J., *Rule-Based Interpretation of Aerial Imagery*, IEEE Transactions, Bd. PAMI-7, Nr. 5, S. 570-585, 1985.
- [McKe96] McKeown, D.M. Jr., Gifford, S.J., Polis, M.F., McMahon, J., *Progress in Automated Virtual World Construction*, ARPA Image Understanding Workshop, Palm Springs, California, 12-15. Feb., 1996.

- [McKe98] McKeown, D.M. Jr., Bullwinkle, G.E., Cochran, S.D., Harvey, W.A., McGlone, C., McMahil, J., Polis, M.F., Shufelt, J.A., *Research in Image Understanding and Automated Cartography: 1997-1998*, DARPA Image Understanding Workshop, Monterey, California, 20-23 Nov. 1998.
- [Min75] Minsky, M., *A framework for representing knowledge*, In Patrick H. Winston, Hrsg., *The Psychology of Computer Vision*, McGraw-Hill, 1975.
- [Nie90a] Niemann, H., *Pattern Analysis and Understanding*, Springer-Verlag, 1990.
- [Nie90] Niemann, H., Sagerer, G., Schröder, S., Kummert, F., *ERNEST: A Semantic Network System for Pattern Understanding*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 12, No. 9, pp. 883-905, Sept. 1990.
- [Niem95] Niem, W., Broszio, H., *Mapping Texture from Multiple Camera Views onto 3D-Object Models for Computer Animation*, Proceedings of the International Workshop on Stereoscopic and Three Dimensional Imaging, September 6 - 8, 1995, Santorini, Greece.
- [Oust94] Ousterhout, J.K., *Tcl and the Tk Toolkit*, Addison-Wesley Publishing Company, 1994.
- [Pear84] Pearl, J., *Intelligent Search Strategies for Computer Problem Solving*, Addison-Wesley Publishing Company, 1984.
- [Poll98] Pollefeys, M., Koch, R., Van Gool, L., *Self-calibration and Metric Reconstruction in spite of Varying and Unknown Internal Camera Parameters*, Proc. of ICCV'98, Bombay, Indien, Jan. 1998.
- [Rie97] Riegel, T., Manzotti, R., Pedersini, F., *3-D shape approximation for objects in multiview image sequences*, - International Workshop on Synthetic-Natural Hybrid Coding and 3D Imaging (IWSNHC3DI '97), Rhodes, Sept. 5-9 1997.
- [Rost98] Rost, U., Munkel, H., *Knowledge Based Configuration of Image Processing Algorithms*, Proc. of the Int. Conf. on Computational Intelligence & Multimedia Applications 1998 (ICCIMA98), 9.-11.2.98, Monash University, Gippsland Campus, Australia.
- [Ryd87] Rydfalk, R., *CANDIDE, A parameterised face*, Internal Report Lith- ISY-I-0866, Linköping University, 1987.
- [Sab93] Sabata, B., Arman, F., Aggarwal, J.K., *Segmentation of 3D Range Images Using Pyramidal Data Structures*, CVGIP: Image Understanding, Vol. 57, No. 3, pp 373-387, 1993.

- [Sag90] Sagerer, G., *Automatisches Verstehen gesprochener Sprache*, Vol.74 Reihe Informatik, Bibliographisches Institut, Mannheim, 1990.
- [Sand97] Sandakly, F., Giraudon, G., *3D Scene Interpretation for a Mobile Robot*, Robotics and Autonomous Systems 21 (1997) 399-414.
- [Shap92] Shapiro, Stuart C. (ed), *Encyclopedia of Artificial Intelligence*, 2nd Edition, John Wiley and Sons, New York, 1992.
- [SNePS] Shapiro, S. C., *SNePS - The Semantic Network Processing System*, Dep. of Computer Science, State University of New York at Buffalo, <ftp://ftp.cs.buffalo.edu/pub/sneps>.
- [Ste90] Steele, Guy, L., *Common Lisp the Language*, Digital Press, 1990.
- [Stil95] Stilla, U., Michaelsen, E., Lütjen, K., *Structural 3D-Analysis of Aerial Images with a Blackboard-based Production System*, in Automatic Extraction of Man-Made Objects from Aerial and Space Images, Ed. Gruen, Kuebler, Agouris, Birkhäuser Verlag, 1995.
- [Stre96] Streilein, A., *Utilization of CAD models for the object oriented measurement of industrial and architectural objects*, Int. Archives of Photogrammetry and Remote Sensing, Vol. XXI, Part B5, Vienna 1996, pp. 548-553.
- [Stro92] Stroustrup, B., *Die C++-Programmiersprache*, Addison-Wesley, 1992.
- [Teich96] Teichert, J., *Segmentierung von Tiefenkarten unter Verwendung von parametrischen Modellen*, Studienarbeit, Universität Hannover, 1996.
- [Toen96] Tönjes, R., *Knowledge Based Modelling of Landscapes*, XVIII. Congress of International Society of Photogrammetry and Remote Sensing, 9.-19. Juli 1996, Wien.
- [Toen99] Tönjes, R., *Wissensbasierte Interpretation und 3D-Rekonstruktion von Landschaftsszenen aus Luftbildern*, Dissertation Universität Hannover, 1999.
- [Tsai87] R. Tsai: *A versatile Camera Calibration Technique for High-Accuracy 3D Machine Vision Metrology Using Off-the-Shelf TV Cameras and Lenses*, IEEE Journal of Robotics and Automation, Vol. RA-3, No. 4, pp. 323-344, Aug. 1987.
- [VRML97] *The Virtual Reality Modeling Language (VRML)*, International Standard ISO/IEC 14772-1:1997.
- [Weik95] Weik, S., *Erzeugung dreidimensionaler Oberflächenmodelle unter Einhaltung geometrischer Randbedingungen*, Diplomarbeit Universität Hannover, 1995.

- [Wiebe98] Wiebelitz, J., *Entwicklung eines Verfahrens zur modellbasierten Suche von Linienelementen in Konturbildern*, Diplomarbeit Universität Hannover, 1998.
- [Win92] Winston, P. H., *Artificial intelligence*, Addison-Wesley, 1992.
- [Winz95] Winzen, A., *Automatische Erzeugung dreidimensionaler Modelle für Bildanalysesysteme*, Dissertation, Universität Erlangen, 1995.
- [Yak78] Yakimovski, Y., Cunningham, R., *A System for Extracting 3D Measurements from a Stereo Pair of TV Cameras*, CGVIP, Vol. 7, 1978, pp. 195 - 210.
- [Zhan91] Zhang, S., Sullivan, G., D., Baker, K.D., *Automatic construction of a relational model for recognition of a 3D object*, SPIE Proc., Vol. 1609, Model-Based Vision Development and Tools, pp. 161, 1991.

A. Ergebnisse der Interpretation

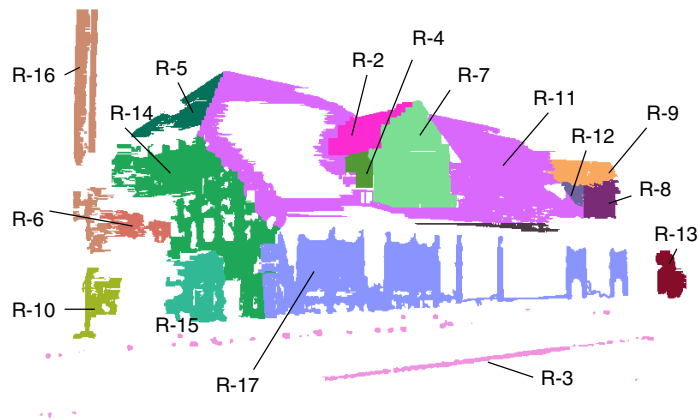
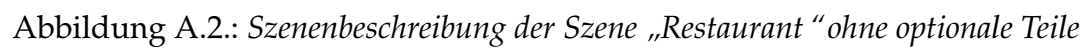


Abbildung A.1.: Segmentierungsergebnis der Szene „Restaurant“



B. Kostenfunktionen des Kostennetzes

Die Kanten des in Abschnitt 8.2.1 beschriebenen Kostennetzes repräsentieren die im folgenden aufgeführten Kostenfunktionen als Funktion von $\vec{x}$ (siehe Gleichung 8.4 und 8.6):

Einhaltung eines Winkels zwischen zwei Ebenen c_3 :

$$\Phi_3(\vec{x}) = (\varphi_{soll} - \varphi_{ist}(\vec{x}))^2 \cdot w_{3j} \quad (B.1)$$

Der Winkel φ_{ist} wird zwischen den Normalenvektoren der beiden Ebenen gemessen. Der Winkel φ_{soll} gibt den vorgegebenen Winkel an.

Parallelität zweier Ebenen c_4 :

$$\Phi_4(\vec{x}) = (\varphi_{ist}(\vec{x}))^2 \cdot w_{4j} \quad (B.2)$$

Der Winkel φ_{ist} wird zwischen den Normalenvektoren der beiden Ebenen gemessen.

Parallelität zweier Polygonkanten c_5 :

$$\Phi_5(\vec{x}) = (\varphi_{ist}(\vec{x}))^2 \cdot w_{5j} \quad (B.3)$$

Der Winkel φ_{ist} wird zwischen den Richtungsvektoren der beiden Schnittgeraden L_1, L_2 gemessen.

Gleichheit der Winkel von zwei Ebenen zu einer dritten c_6 :

$$\Phi_6(\vec{x}) = (\varphi_1(\vec{x}) - \varphi_2(\vec{x}))^2 \cdot w_{6j} \quad (B.4)$$

Der Winkel $\varphi_1(\vec{x})$ wird zwischen den Normalenvektoren der ersten Ebene zur Referenzebene gemessen. Der Winkel $\varphi_2(\vec{x})$ entsprechend.

Gleichheit der Längen zweier / mehrerer Polygonkanten c_7 :

$$\Phi_7(\vec{x}) = w_{7j} \cdot \sum_{i=1}^N (\delta_m(\vec{x}) - \delta^{(i)}(\vec{x}))^2 \quad \text{mit} \quad \delta_m = \frac{1}{N} \cdot \sum_{i=1}^N \delta^{(i)}(\vec{x}) \quad (\text{B.5})$$

Die Kostenkante C_7 enthält eine Referenzebene und eine Liste der Länge N von Tripeln aus jeweils drei Ebenen. Der Ausdruck $\delta^{(i)}(\vec{x})$ gibt den Abstand des jeweiligen Schnittpunktes eines Tripels zur Referenzebene an, $\delta_m(\vec{x})$ den mittleren Abstand.

Index

- 3D-Modell, 8
- A*, 61
- Attribut, 39
 - Richtungsvektor, 40
 - Vektor, 40
- Attribute, 21
- Bewertung
 - Attributbewertung, 64
 - Instanzbewertung, 65
 - Suchbaumknotenbewertung, 66
- Disparität, 32
- Disparitätsschätzer, 32
- dynamische Programmierung, 32, 74
- ERNEST, 29
- Frames, 29
- generisches Modell, 18
- Instanz, 29, 38
 - Fehlinstanz, 39, 69
 - hypothetische, 38
 - partielle, 38
 - vollständige, 39
- intensionelle Beschreibung, 18, 29
- Interpretationszustand, 38
- Körper, 8
 - Körperberandung, 8
- Kamera, 8
 - CAHV, 27
 - Kalibrierung, 31
 - Zentralprojektion, 27
- Kamerabewegung, 92
- Kante, 29
- Knoten, 37
 - des Kostennetzes, 85
- Konzept, 29, 37
- Kostennetz, 85
 - Erzeugung aus der Szenenbeschreibung, 88
- Lichtquelle, 8
- Mehrfachbindung, 42, 58
- Modell, 13
- Netzwerkkanten, 40
 - Concrete-Of, 41
 - Data-Of, 41
 - des Kostennetzes, 87
 - Instance-Of, 41
 - Is-A, 41
 - Part-Of, 41
 - Priorität, 43
- Oberflächenbeschreibung
 - Approximation, 10
 - Erzeugung, 83
 - Geometrische Beschreibung, 25
 - Schätzung der geometrischen, 84
 - Schätzung der Oberflächentextur, 94
 - Textur, 10, 26
- Objekt, 7
- Registrierung, 92
- Rekonstruktion, 8
- Relationen, 21, 40, 87
- Repräsentation, 7
- Restkosten, 61
- Segmentierung, 34

- Semantische Netze, 29
- Suchbaum, 59
- Suchbaumknoten, 59
- Suchfunktionen, 46, 68
- Synthesemodell, 9
- Szene, 7
- Szenenanalyse, 7
- Szenenbeschreibung, 7, 38
- Szenenmodell, 8, 13
 - strukturiertes, 17
- Textur, 10, 94
 - Texturkarte, 26
- Tiefenkarte, 34
- Transformationen
 - zur Erzeugung einer Szenenbeschreibung, 55
- Verdeckungsproblem, 69
- Vorwissen
 - Regeln für die Wissensverarbeitung, 47, 57
 - Repräsentationsformen, 28
- Wissensbasiertes System, 8
- Wissensbasis, 38

Lebenslauf

Oliver Grau

28. Dezember 1963	geboren in Hannover Eltern: Bodo Grau Anni Grau, geb. Hischer
1970-74	Grundschule in Altwarmbüchen
1974-77 1977-83	Gymnasium Großburgwedel Gymnasium Isernhagen Abschluß der allgemeinen Hochschulreife
1983-91	Studium der Elektrotechnik an der Universität Hannover Studienschwerpunkt Nachrichtenverarbeitung Abschluß Diplom
März 1991 - Sep. 1997	Wissenschaftlicher Mitarbeiter am Institut für Theoretische Nachrichtentechnik und Informations- verarbeitung der Universität Hannover
seit Okt. 1997	Wissenschaftlicher Mitarbeiter am Laboratorium für Informationstechnologie der Universität Hannover