

Abbildung 3.9: Repräsentation des Verkehrszeichens 101

Am Beispiel des Verkehrszeichens *Achtung Gefahrenstelle*, kodiert als Verkehrszeichen 101 gemäß StVO, soll die Repräsentationsform verdeutlicht werden. Betrachtet man die Konturen des Zeichens, so stellt man fest, daß es sich aus zwei ineinanderliegenden Dreiecken zusammensetzt, die ein Ausrufezeichen als Piktogramm enthalten. Daraus ergeben sich die Netzwerkknoten $vz101 \in SN$, sowie $Dreieck1, Dreieck2 \in CN$ und $Ausrufezeichen \in PN$. Der Einfachheit halber betrachten wir hier als Lagerrelation lediglich die *enthält* Relation, so daß wir die zwei Kanten $enthält(Dreieck2, Dreieck1) \in RL$ und $enthält(Ausrufezeichen, Dreieck2) \in RL$ in das Netzwerk einfügen müssen. Zuletzt werden alle erzeugten Knoten und Kanten durch Kanten des Typs CL mit dem Knoten $vz101$ verbunden. Diese Kanten werden, gemäß ihrer Wichtigkeit für das Schild, mit Gewichten versehen. Da das Piktogramm das größte Unterscheidungsmerkmal zu anderen Verkehrszeichen darstellt, wird es dementsprechend höher gewichtet²⁵.

Weitere Eigenschaften des Netzwerkes Die Beschränkung auf die *enthält*-Relation in obigem Beispiel ist nicht zufällig. Gerade diese Relation wird benutzt, um durch das Netzwerk zu navigieren. Dies bietet sich an, da die *enthält*-Relation bereits von der Segmentierungskomponente²⁶ geliefert wird, daher also nicht erst berechnet werden muß.

Eine weitere Eigenschaft ist der Verzicht auf Redundanz. Eine Kontur eines

²⁵Unterschiedliche Gewichtungen für die zwei Dreiecke sind ebenfalls sinnvoll. Siehe dazu auch 3.4.2.7.

²⁶siehe 3.4.2.1

bestimmten Typs²⁷ kommt im Verkehrszeichennetz nur sooft vor, wie in der Beschreibung des Verkehrszeichen das sie am häufigsten enthält.

Weiterhin soll die Bedingung gelten, daß der Knoten $a \in CN_{vs}$ eines Verkehrszeichensubnetzes VS , der die Konturinformation des äußersten zu dem dargestellten Verkehrszeichen gehörenden geometrischen Objekts speichert, durch eine Kante des Typs E mit dem Startknoten s verbunden wird. Knoten a ist identisch mit dem Wurzelknoten des durch die Knoten CN_{vs} und PN_{vs} und den Kanten $\{k \mid k \in RL_{vs}, V(k) = \text{enthält}\}$ aufgespannten Baumes. Diese von s durch E -Kanten erreichbaren Knoten werden als *Eintrittsknoten* bezeichnet.

Abbildung 3.10b zeigt den Aufbau eines Beispielnetzwerkes, das die in 3.10a dargestellten Verkehrszeichen repräsentiert. Dabei wurde das Netz aus Gründen der Übersichtlichkeit in zwei Ebenen aufgeteilt. Die untere Ebene stellt das Subnetz dar, welches von der Knotenmenge $PN \cup CN \cup \{s\}$ und der Kantenmenge $RL \cup E$ aufgespannt wird. Relationen sind als Pfeile dargestellt, wobei die des Typs E mit e_i bezeichnet sind. Die mit i bezeichneten Pfeile stellen die *enthält* Relation dar, während dc ²⁸ und dut ²⁹ Relationen bezeichnen, die die Lage zwischen zwei Konturen genauer festlegen.

Die obere Ebene zeigt die modellierten Verkehrszeichen und die auf diesen definierte Klasseneinteilung³⁰. Kanten des Typs CL , die die Verkehrszeichenknoten SN mit den Knoten und Kanten der unteren Ebene verbinden, wurden nicht dargestellt.

3.4.1.2 Wissen über Konturen

Bei der in 3.4.1.1 gelieferten Beschreibung der Struktur des Verkehrszeichennetzwerkes, wird noch nicht weiter auf die Speicherung der Konturen eingegangen, aus denen sich ein Verkehrszeichen zusammensetzt.

Ein Verkehrszeichen wird in der Szene durch Formprimitive beschrieben. Die Aufgabe besteht jetzt darin, die Konturen in einer Form abzuspeichern, die sich leicht mit den zu erkennenden Formprimitiven vergleichen läßt, andererseits jedoch eine flexiblere und einfachere Struktur aufweist. Eine Lösung bietet die nachfolgend definierte Konturbeschreibung, die für alle zusammenhängenden Linienzüge, d.h. für alle Formprimitive außer *circular* und *circular.line*, eingesetzt werden kann.

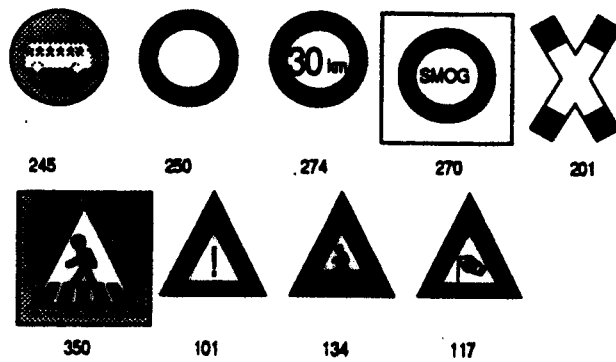
²⁷ Ein Konturtyp ist beispielsweise die Beschreibung eines Dreiecks, dessen Spitze nach oben zeigt.

²⁸ dc - *double.circle*

²⁹ dut - *double.up.triangle*

³⁰ Diese wird implizit durch die Struktur der Verkehrszeichen bestimmt und liegt nicht als Datenstruktur vor.

a)



b)

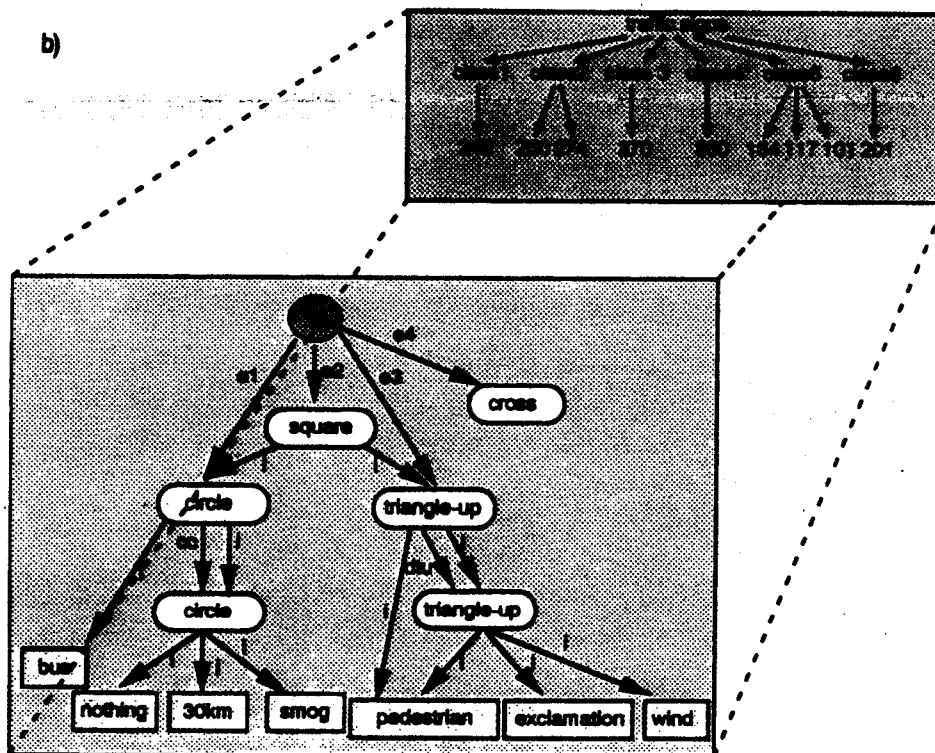


Abbildung 3.10: Der Aufbau eines Verkehrszeichennetzwerkes

Definition : Winkel/Längen - Konturbeschreibung

Eine Winkel/Längen - Konturbeschreibung besteht aus zwei Vektoren α und l .

$$\forall i : \alpha_i \in [0..359], \forall i : l_i \in \text{REAL}_0^+$$

Dabei bezeichnet α einen Vektor von Winkeln³¹, l einen Vektor von Kantenlängen. Ist x eine Winkel/Längen-Konturbeschreibung, so bezeichnet $\alpha(x)$ den Winkelvektor und $l(x)$ den Längenvektor.

Es fällt auf, daß bei dieser Beschreibung keine konkrete Information über die Position der Kontur in einem Koordinatenkreuz gegeben ist, so wie das bei den Formprimitiven der Fall ist. Vielmehr handelt es sich bei einer wie oben definierten Konturbeschreibung um ein Muster, das sich frei über ein beliebiges Koordinatenkreuz bewegen, und seine Größe ebenso frei verändern kann. Wird ein beliebiger Eckpunkt fixiert und die Länge einer beliebigen Kante angegeben, so erhält man eine konkrete Kontur.

Aufgrund ihrer flexiblen Größe und freien Beweglichkeit ist diese Repräsentation ideal für die Erkennung von Konturen aus Formprimitiven geeignet. Die Verkehrszeichenkonturen werden in Form einer Winkel/Längen-Beschreibung gespeichert, wobei die Längen so normiert sind, daß die kleinste Länge 1 beträgt.

Abbildung 3.11 zeigt ein Beispiel für die Kodierung einer Kontur mit Hilfe der Winkel/Längen-Konturbeschreibung. Die Ausnutzung der Konturbeschreibung, bei der Erkennung, sowie bei der Rekonstruktion von aufgebrochenen und verrauschten Konturen aus einer Verkehrsszenenbeschreibung, wird in 3.4.2.6 behandelt.

Setzt man flexible Größe und freie Beweglichkeit voraus, so besteht die Konturbeschreibung eines Kreises lediglich aus der Information, daß es sich um einen Kreis handelt. Die Menge der Konturbeschreibungen wird also um die spezielle Konturbeschreibung *circular.shape* ergänzt. Die Kontur eines Kreisbogens wird durch dessen Anfangs- und Endpunkt, sowie durch Kreismittelpunkt und Radius beschrieben. Wie bei der Konturbeschreibung eines Kreises, fallen auch hier der Mittelpunkt und Radius weg. Die Konturbeschreibung eines Kreisbogens wird somit nur durch zwei Winkel³² bestimmt, die Anfang und Ende des Kreisbogens festlegen³³.

³¹ Die α_i stellen *Absolutwinkel* bezüglich des Koordinatenkreuzes dar.

³² Auch diese Winkel stellen *Absolutwinkel* bezüglich des Koordinatenkreuzes dar.

³³ Da der Kreisbogen als Kontur eines Verkehrszeichens nicht auftritt, ist eine Konturbeschreibung eines Kreisbogens nicht implementiert.

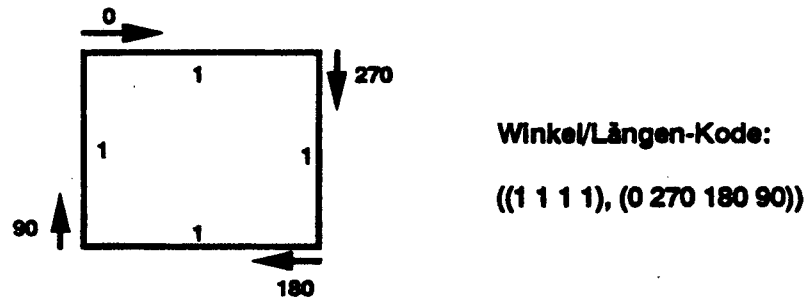


Abbildung 3.11: Beispiel für die Winkel/Längen-Konturbeschreibung

3.4.1.3 Wissen über Lagerrelationen

Lagerrelationen sind jeweils zwischen zwei Konturen definiert. Die Komplexität der Beschreibung einer Lagerrelation hängt stark von den beteiligten Konturen ab. Um die Berechnung zu vereinfachen werden Lagerrelationen als Funktionen definiert, deren Definitionsbereich jeweils auf bestimmte Formprimitivtypen begrenzt ist. Beispielsweise kann die Relation *double.circle* nur Argumente des Typs *circular* oder *circular.line* verarbeiten. Diese Einschränkung wird jedoch größtenteils bereits durch das Einfügen der Relation als RL-Kante in das Netzwerk gewährleistet, da damit die Argumenttypen durch die verknüpften Knoten bestimmt werden.

Das Wissen über die Lagerrelation zwischen zwei Konturen ist in einer Funktion $R: \Gamma \times \Gamma \rightarrow [0..1]$ kodiert. Diese Funktion vergleicht die Lagerrelation von zwei Formprimitiven mit einer Musterrelation und liefert einen Ähnlichkeitswert zu dieser. Dabei ist es möglich, eine solche Funktion R aus mehreren, einfacheren, Relationenfunktionen zusammenzusetzen. Dabei werden deren errechnete Ähnlichkeitswerte gemäß ihrer Wichtigkeit für die zusammengesetzte Gesamtrelation gewichtet und kombiniert.

$$R(x,y) = C\left((R_1(x,y), \dots, R_n(x,y)), (w_1, \dots, w_n)\right)$$

Dabei bezeichnet C eine Funktion zur Verknüpfung der beiden Vektoren. Für die Funktionen R_i gilt $R_i \in \Delta$, der Wertebereich der Gewichtung ist aber mit dieser Definition noch nicht festgelegt. Verschiedene Ansätze zur Behandlung von Ähnlichkeitswerten, oder Evidenzen³⁴, gehen von verschiedenen Voraussetzungen aus, die an die w_i gestellt werden. Eine Möglichkeit zur Behandlung von Evidenzen wird von Valdes [Val91] vorgeschlagen.

³⁴Bei Vergleich mit Mustern kann statt von Ähnlichkeitswerten auch von Evidenzen gesprochen werden.

Hier sollen die beiden Vektoren in Form einer normierten gewichteten Summe zusammengefaßt werden. Dadurch ist es möglich, daß $R(x,y) > 0$ liefert, obwohl es eine Funktion R_i , $1 \leq i \leq n$, gibt mit $R_i(x,y) = 0$. Dies ist bei Valdes Ansatz nicht möglich. Durch diese Eigenschaft ist es einfacher, mit verrauschten Eingangsdaten zurechtzukommen, denn es ist durchaus möglich, daß zwei Objekte eine bestimmte (komplexere) Lagerrelation erfüllen, auch wenn durch gestörte Eingangsinformation Teile der Objekte fehlen, wodurch eine Subfunktion der Relationsfunktion den Wert 0 liefern kann.

$$R(x,y) = \frac{R_1(x,y)w_1 + \dots + R_n(x,y)w_n}{\sum_{i=1}^n w_i}$$

Die Berechnung der Ähnlichkeit zwischen der Musterrelation und der zu untersuchenden Relation in der Szene erfolgt in zwei Schritten. Zuerst wird der Abstand zwischen gemessenen Werten und Muster durch ein geeignetes Distanzmaß bestimmt. Der durch das Distanzmaß bestimmte Abstand muß dann in einen Ähnlichkeitswert überführt werden. Eine einfache Funktion $f: \text{REAL} \rightarrow [0,1]$ kann diese Aufgabe erledigen³⁵. An dieser Stelle ergeben sich Möglichkeiten, das Ähnlichkeitsmaß einzustellen. Genauer dazu in 3.4.2.7.

Wissen über Piktogramme Das Verkehrszeichen untergliedert sich in *große* Konturen, die die Verkehrszeichenklasse festlegen, und in Bilder und Schrift, die zur Identifikation des Verkehrszeichens und seiner genauen Bedeutung benötigt werden. Aufgrund ihrer Größe sowie der Komplexität der Kontur eines Bildes oder eines Schriftzeichens ist es nicht möglich (bis auf einige Ausnahmen) ein solches Objekt mit Hilfe des in dieser Arbeit vorgestellten Gesamtsystems³⁶ zu erkennen. Dazu sind folgende Gründe anzuführen:

- Die erste Problemquelle ist die natürliche Szene selbst. Abgesehen von Verdeckung durch andere Szenenobjekte kann der zur Konturbestimmung eines Objekts benötigte Kontrast durch die enge Strichführung bei Schriftzeichen gestört werden. Betrachtet man beispielsweise eine Fotografie, so fällt es oft schwer, geschriebenen Text auf dieser zu erkennen.
- Ein technisches Problem ist die Auflösung der verarbeiteten Kamerabildes. Hier stellt die aktuelle Bildverarbeitungstechnologie nur ein 512×512 Pixel Bild zur Verfügung³⁷. Geht man von der, schon sehr klein gewählten, Mindestzahl von 16×16 Pixeln zur Erkennung eines einzelnen Buchstaben aus,

³⁵ Stetigkeit und Monotonie der Funktion sind Voraussetzung für eine korrekte Transformation des gegebenen Distanzmaßes in ein Ähnlichkeitsmaß.

³⁶ Die im Anhang aufgeführten Bilderkennungssysteme lösen dieses Problem ebensowenig.

³⁷ Aus technischen Gründen wird zur Zeit in diesem Projekt nur ein 256×256 Pixel Bild verwendet.

so können bei Ausnutzung der vollen Breite des Bildes gerade 32 Buchstaben ohne Zwischenräume erkennbar dargestellt werden, was zweifellos nicht ausreicht um Texte erkennen zu können. Dabei ist anzumerken, daß ein Verkehrszeichen in der Regel $\frac{1}{4}$ bis $\frac{1}{8}$ der untersuchten Fläche bedeckt³⁸, womit nicht einmal mehr das Wort *Einbahnstraße* erkannt werden könnte.

- Betrachtet man die Kontur eines Piktogramms, so besteht diese aus einer Verkettung von Linien und Kreisbögen. Auf einer solchen Beschreibung basiert das System MIRABELLE³⁹ von R. Mohr [Mo87], welches recht erfolgreich auch komplexere Zeichnungen erkennt. Die Bildvorverarbeitung liefert jedoch eine Menge von Formprimitiven. Bei deren Erzeugung werden Linienzüge geglättet, was für komplexere, kleine Objekte dazu führt, daß deren Beschreibung unbrauchbar wird, da Details verloren gehen. Je nach Größe werden Piktogramme bereits vorher aussortiert, da ihre HSC-Wurzel auf einer zu niedrigen Ebene liegt.
- Text ist nicht zusammenhängend, d.h. jedes einzelne Zeichen wird für sich betrachtet. Das führt zu einer Menge von kleinen Objekten, die sich nur geringfügig von Rauschen unterscheiden. Sie werden also bei der Rauschunterdrückung der Bildvorverarbeitung bereits ausgesondert und nicht weiter betrachtet.

Zur Lösung des Piktogramm-Problems können einige Ansätze vorgeschlagen werden:

- Benutzung von Kameras mit höherer Auflösung. Die Folge wäre ein besseres Erkennen von Ziffern (auf Pixelebene). Es würde sich jedoch nichts ändern in Bezug auf die oben angeführte Rauschunterdrückung der Vorverarbeitung.
- Delegation von Texterkennungsaufgaben an speziell dafür konstruierte Systeme. Da diese Systeme andere Vorverarbeitungsschritte benötigen, muß ihnen eine Pixelmatrix übergeben werden. Um diesen Schritt überhaupt zu ermöglichen muß vorher der genaue Bereich gefunden werden, in dem ein Text zu erwarten ist.
- Vergrößern des Ausschnitts, in dem ein Bild oder Text erwartet wird. Nach der Vergrößerung, der auch wieder ein Vorverarbeitungsschritt folgt, kann das neu erhaltene Bild entweder an spezielle Systeme übergeben werden, oder aber, da

³⁸ Diese Werte ergeben sich aus den vorliegenden Testszenen. Die Einschränkung ist durch die Anbringung der Kamera im Fahrzeug bedingt.

³⁹ Das System untersucht zweidimensionale, aus Linien bestehende Zeichnungen. Die Eingabe erfolgt per Hand auf einem Grafiktablett.

ein Großteil der oben aufgeführten Probleme beseitigt ist, mit symbolischen Methoden weiter untersucht werden. In diesem Fall kann ein Piktogrammnetzwerk nach Art des Verkehrszeichennetzwerks mit diesem gekoppelt werden und Piktogramme ebenfalls durch Konturen⁴⁰ und zwischen diesen Lagerrelationen beschrieben werden.

In dieser Arbeit wurde eine Schnittstelle zur Verfügung gestellt, die es offen läßt, welche Lösung implementiert wird. Dabei wird an der Schnittstelle ein Szenenbereich angegeben, der ein Piktogramm aus einer ebenfalls übergebenen Menge von möglichen Piktogrammen, enthalten könnte. Erwartet werden Ähnlichkeitswerte für die übergebenen Piktogramme.

Die Lagerrelation *double.circle* Als Beispiel zur Repräsentation von Wissen über Lagerrelationen soll die Relation *double.circle* beschrieben werden. Dazu betrachten wir Verkehrszeichen 250 *Durchfahrt verboten*. Das Zeichen besteht aus zwei ineinanderliegenden Kreisen. Abbildung 3.12 verdeutlicht, daß diese Information alleine nicht ausreicht, um das Verkehrszeichen zu beschreiben.

Welche Eigenschaften sind noch vorhanden? Die Mittelpunkte der beiden Kreise liegen auf dem selben Punkt und die Radien weisen ein festes Längenverhältnis auf. Dies kann durch zwei Relationenfunktionen R_{center} , $R_{radius} \in \Delta$ beschrieben werden, die jeweils den Abstand der im Bild ermittelten Relation von der implizit repräsentierten Musterrelation berechnen, und diesen anschließend mit Hilfe der Funktionen f_1 und f_2 in einen Ähnlichkeitswert überführen.

$$R_{center}(a, b) = f_1\left(\sqrt{(center(a)_x - center(b)_x)^2 + (center(a)_y - center(b)_y)^2}\right)$$

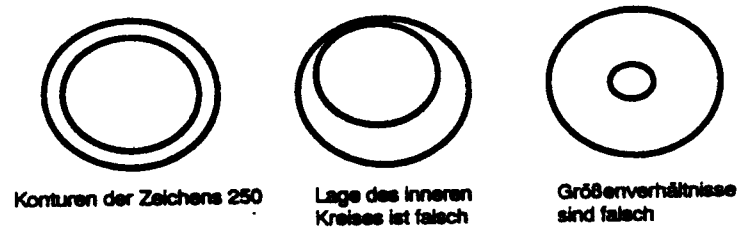
R_{center} berechnet den Abstand der Mittelpunkte der beiden Kreise. Betrachtet man die Distanz zu der Musterrelation, so ist diese identisch mit dem Abstand der Kreismittelpunkte. Mit steigendem Abstand wird die Ähnlichkeit zu dem Muster geringer.

$$R_{radius}(a, b) = f_2\left(abs\left(Muster - Min\left(\frac{radius(a)}{radius(b)}, \frac{radius(b)}{radius(a)}\right)\right)\right)$$

R_{radius} mißt das Längenverhältnis der Radien der beiden Kreise und bildet die Differenz zwischen Musterverhältnis und Meßergebnis.

$$f_1(x) = 1 - \left(\frac{x}{1+x}\right)^2$$

⁴⁰Eine Möglichkeit der Kombination von Linien und Kreissegmenten muß dazu realisiert werden.

Abbildung 3.12: Die Lagerrelation *double.circle*

$$f_2(x) = 1 - x$$

Die Funktionen f_1 und f_2 transformieren die errechnete Distanz in einen Ähnlichkeitswert. Betrachtet man den Wertebereich der beiden Distanzberechnungen, so stellt man fest, daß dieser im zweiten Fall auf das Intervall $[0,1]$ beschränkt ist, im ersten jedoch in $[0,+\infty[$ ⁴¹ liegt. Eine Distanz von 1 bedeutet also im ersten Fall eine sehr große Ähnlichkeit zwischen Muster und Kandidat, im zweiten Fall würde derselbe Abstand absolute Unähnlichkeit bedeuten. Dies begründet die zwei verschiedenen Transformationsfunktionen.

Die in 3.12a dargestellte Konstellation liefert folgende Ergebnisse:

$$\begin{aligned} \text{center}(a) &= (10, 8) & \text{center}(b) &= (9, 7) \\ \text{radius}(a) &= 5 & \text{radius}(b) &= 3 \end{aligned}$$

Die Radiusrelation des Musterverkehrszeichens sei 0.8 und die Gewichte der beiden Teilrelationen seien mit $w_{\text{center}} = 5$ und $w_{\text{radius}} = 7$ festgelegt.

$$\begin{aligned} R_{\text{center}}(a, b) &= f_1\left(\sqrt{(10-9)^2 + (8-7)^2}\right) \\ &= f_1(1) \\ &= 0.75 \\ R_{\text{radius}}(a, b) &= f_2\left(\text{abs}\left(0.8 - \frac{3}{5}\right)\right) \\ &= f_2(0.2) \\ &= 0.8 \end{aligned}$$

Die Lagerrelation *double.circle* setzt sich aus den beiden obigen Teilrelationen zusammen und errechnet sich aus der normierten gewichteten

⁴¹In der realen Anwendung ist der Wertebereich nur durch die Größe der betrachteten Bildmatrix beschränkt

Summe der durch diese gelieferten Ähnlichkeitswerte und der ihnen zugeordneten Gewichte:

$$\text{double.circle}(a, b) = \frac{0.75 \cdot 5 + 0.8 \cdot 7}{5 + 7} = 0.78$$

3.4.1.4 Netzwerkoperationen

Um das Netzwerk aufzubauen und zu modifizieren werden Zugriffsfunktionen zur Verfügung gestellt. Jedem Knoten und jeder Kante des Netzwerks wird dazu ein eindeutiger Schlüssel zugeordnet, mit dessen Hilfe die Netzwerkelemente identifiziert werden können. Beim Einfügen können zwei Fälle unterschieden werden:

Einfügen einer Kante: Zum Einfügen einer Kante werden die beiden Knoten angegeben, die durch die neue Kante verbunden werden sollen. Dabei ist auf die Reihenfolge zu achten, da die Kante gerichtet ist und anderenfalls die Argumente der repräsentierten Funktion vertauscht werden könnten.

Einfügen eines Knotens: Da das Netzwerk zusammenhängend sein muß, kann ein Knoten nicht eingefügt werden, ohne gleichzeitig eine Kante einzufügen, die den Knoten mit dem bestehenden Netzwerk verbindet. Eine Ausnahme bildet hier die Erzeugung eines neuen Netzwerks durch Einfügen eines Knotens in ein *leeres Netzwerk*.

Das Löschen von Knoten und Kanten unterliegt gewissen Einschränkungen, da hier sehr leicht Bedingungen des Netzwerks, die im ersten Abschnitt von 3.4.1.1 beschrieben wurden, verletzt werden können.

Löschen von Knoten: Es ist nicht möglich, einen Knoten zu entfernen, ohne gleichzeitig alle Kanten, die ihn mit dem Netz verbinden, zu löschen. Gibt es eine Kante, die nicht gelöscht werden darf, dann ist die Operation nicht durchzuführen.

Löschen von Kanten: Beim Löschen von Kanten ist zu beachten, ob durch das Löschen ein Knoten vom Netzwerk abgespalten wird. Ist dies der Fall, so kann das Löschen der Kante nur durch zusätzliches, explizites Löschen des Knotens erfolgen.

Einen Sonderfall stellt das Löschen von Knoten des Typs SN und Kanten des Typs CL dar. Auf das Einfügen von Knoten dieses Typs wird in 3.4.1.5 näher eingegangen. Das Entfernen eines Knotens $v \in \text{SN}$ bedeutet das Entfernen eines

Verkehrszeichens aus der Wissensbasis. Um das Netzwerk möglichst klein zu halten, wird überprüft, ob CN- oder PN-Knoten, sowie RL-Kanten existieren, die nach Löschen der zu v führenden CL Kanten mit keiner weiteren Kante dieses Typs verknüpft sind. Da diese Knoten und Kanten keine relevante Information mehr tragen, sollen auch sie gelöscht werden.

$$\text{Edge-Deletion-Set} = \{x \in RL \mid \nexists e \in CL, e = (a, b) : a = v \wedge b := x\}$$

$$\text{Node-Deletion-Set} = \{x \in RN \cup PN \mid \nexists e \in CL, e = (a, b) : a = v \wedge b = x\}$$

Das Löschen der Knoten aus dem Node-Deletion-Set veranlaßt wiederum das Löschen der mit ihnen verbundenen Kanten.

Desweiteren wird überprüft, ob durch das Löschen von v eine Kante $e \in E$ gelöscht werden muß. Dies ist der Fall, wenn ein Knoten zwar im Netzwerk verbleibt, aber jetzt nicht mehr die äußere Kontur eines Verkehrszeichens repräsentiert.

3.4.1.5 Wissensakquisition

Zur Akquisition von Wissen wird eine Schnittstelle zur Verfügung gestellt, die es dem Benutzer erlaubt ein neues Verkehrszeichen in die Wissensbasis einzutragen. Dazu wird eine Regelsprache definiert, mit deren Hilfe ein Verkehrszeichen beschrieben werden kann. Eine solche Beschreibung wird in ein Verkehrszeichensubnetz übersetzt, welches dann in ein bestehendes Verkehrszeichennetz eingefügt werden kann.

Die Regelsprache Um ein Verkehrszeichen in das Netzwerk einzufügen, ist eine Reihe von Operationen notwendig, wobei jeweils Vorschriften bezüglich der Struktur des Netzwerks, eingehalten werden müssen. Die Regelsprache soll die Akquisition erleichtern. Durch folgendes Syntaxdiagramm soll die Sprache definiert werden:

```

Verkehrszeichen ::= ( Vz-Name Aufbau Gewichtsvektor )
Vz-Name ::= VZ-42
Gewichtsvektor ::= ( {  $x \in [0, 1]$  }n )
Aufbau ::= ( {Kontur}+ | {Lagerrelation}+ | {Piktogramm}* )
Kontur ::= ( Mustername Variable )
Piktogramm ::= ( PICTURE Piktogrammname Variable )
Lagerrelation ::= ( RELATION Funktionsname Variable-1 Variable-2 )
Die durch Kontur- und Piktogramm-Definitionen gebundenen Variablen
müssen auch durch Lagerrelationen referenziert werden. Dabei wird die

```

⁴²x entspricht Verkehrszeichenkode nach StVO

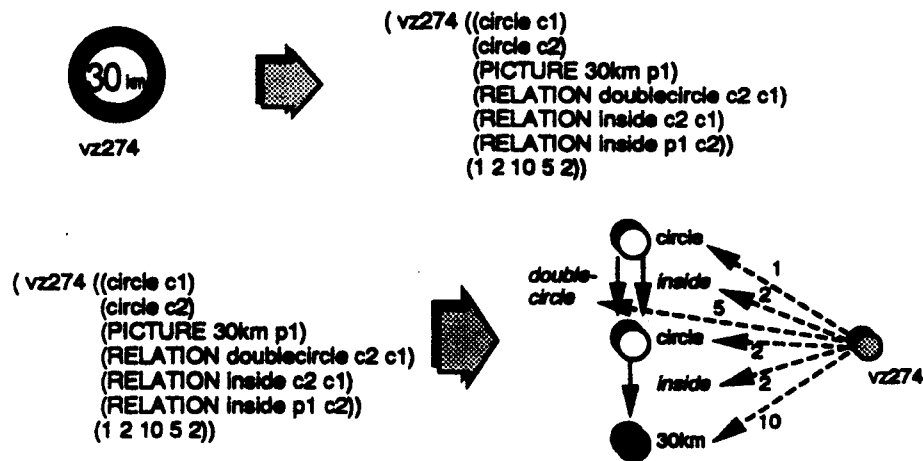


Abbildung 3.13: Übersetzung einer Verkehrszeichendefinition

Eindeutigkeit der Variablennamen vorausgesetzt. In *Lagerrelationen* benutzte Variablen müssen an Konturen oder Piktogramme gebunden sein.

Außer den oben angeführten Bedingungen muß sichergestellt werden, daß die durch ihre Namen referenzierten Muster, Piktogramme und Lagerrelationen definiert sind.

Übersetzung in ein Verkehrszeichensubnetz Nachdem die an die Eingabe gestellten Bedingungen überprüft worden sind, werden für jede Kontur ein Knoten des Typs CN und für jedes Piktogramm ein Knoten des Typs PN erzeugt und mit einer eindeutigen Identifikationsnummer, sowie mit den gegebenen Muster- und Piktogrammnamen versehen. Danach wird jeder Lagerrelation eine Kante des Typs RL zugeordnet. Diese Kanten werden dann gemäß der durch die Variablen gegebenen Referenzen den vorher erzeugten Knoten zugeordnet. Anschließend wird ein Knoten des Typs SN erzeugt und mit dem Vz-Namen versehen. Dieser Knoten wird mit allen Knoten und Kanten des zuvor erzeugten Teilnetzes durch CL-Kanten verbunden. Die Gewichtung der CL-Kanten erfolgt gemäß der Reihenfolge der Gewichte und der Einträge im *Aufbau*-Vektor. Abbildung 3.13 zeigt die Definition eines Verkehrszeichens mit Hilfe der Regelsprache sowie die Übersetzung in ein Verkehrszeichensubnetz.

Einfügen des Verkehrszeichensubnetzes in das Verkehrszeichennetz Zuletzt ist das aus der Verkehrszeichenbeschreibung erzeugte Verkehrszeichensubnetz in das bestehende Verkehrszeichennetz einzufügen. Dabei muß beachtet werden, daß keine Redundanzen entstehen dürfen. In einem ersten Schritt werden die CN- und PN-Knoten des Verkehrszeichensubnetzes mit den Netzwerkknoten verglichen. Da die Knoten typisiert sind⁴³, ist diese Einschränkung der Knoten leicht zu treffen. Seien k_i ($i = 1, \dots, n$) die Knoten des Verkehrszeichensubnetzes und n_j ($j = 1, \dots, m$) die Knoten des bestehenden Verkehrszeichennetzwerks. Dann wird eine Menge M erzeugt, die Paare von Knoten des neuen Verkehrszeichensubnetzes und des bestehenden Verkehrszeichennetzes enthält, deren Konturbeschreibungen übereinstimmen.

$$M = \{(k_i, n_j) \mid V(k_i) = V(n_j) \text{ mit } i = 1, \dots, n; j = 1, \dots, m\}$$

Aus der Menge M werden anschließend Untermengen S_k gebildet, die keinen Knoten k_i und keinen Knoten n_j mehrfach enthalten. Eine solche Menge stellt einen Knotenmatch zwischen dem Verkehrszeichensubnetz und dem bestehenden Verkehrszeichennetz dar. Aus diesen Untermengen wird diejenige (bzw. werden diejenigen) mit der größten Elementenzahl ausgewählt. Die (k_i, n_j) Paare aus S_k werden als Bindung bezeichnet. Für jeden Knoten k_i , für den keine Bindung in S_k existiert, wird eine Kopie $k_{i,k}$ angelegt. Wird zu einem Knoten eine Kopie angelegt, so bedeutet dies, daß ein neuer Knoten in das Netzwerk eingefügt werden muß, um das aufzunehmende Verkehrszeichen zu repräsentieren. Das Paar $(k_i, k_{i,k})$ wird zu S_k hinzugefügt. Weiterhin bezeichne $b_k(k_i) = n_j$ eine Funktion, die jedem k_i die durch die Menge S_k bestimmte Bindung zuordnet.

Als nächstes sollen die RL-Kanten des Verkehrszeichensubnetzes betrachtet werden. Könnten mehrere Untermengen S_k gebildet werden, so werden die Kanten zunächst zur Bestimmung der besten Bindungsmenge S_k herangezogen.

Seien e_i ($i = 1, \dots, r$) die RL-Kanten des Verkehrszeichensubnetzes und r_j ($j = 1, \dots, s$) die RL-Kanten des bestehenden Verkehrszeichennetzes. Eine Bewertungsfunktion $f: \{S_k \mid S_k \subseteq M\} \rightarrow \text{Nat}$ sei wie folgt definiert:

$$f(S_k) = \sum_{i=1}^r g_k(e_i) + em(S_k)$$

$$em(S_k) = \begin{cases} 1 & \text{Wurzelknoten ist an Eintrittsknoten gebunden} \\ 0 & \text{sonst} \end{cases}$$

⁴³Die Knoten werden nach ihrem Typ in Listen verwaltet, so daß ein Zugriff auf Knoten eines bestimmten Typs möglich ist.

$$g_k(e_i) = \begin{cases} 1 & e_i = (a, b), \exists r_j, j = 1, \dots, s; r_j = (x, y) : \\ & V(e_i) = V(r_j) \wedge x = b_k(a) \wedge y = b_k(b) \\ -1 & e_i = (a, b), \exists r_j, j = 1, \dots, s; r_j = (x, y) : \\ & V(e_i) = V(r_j) \wedge x = b_k(b) \wedge y = b_k(a) \\ 0 & \text{sonst} \end{cases}$$

Die Menge S_k mit der höchsten Bewertung wird ausgewählt. Ist dabei $em(S_k) = 0$, so muß eine zusätzliche Kante vom Typ E eingefügt werden, die den Netzwerkknoten s mit dem Knoten des Verkehrszeichensubnetzes verbindet, der die äußere Kontur des Verkehrszeichens repräsentiert. Danach wird eine Bindungsmenge R erzeugt, die den Kanten e_i jeweils eine Kante r_j zuordnet. Für jede Kante e_i , der keine Kante r_j zugeordnet werden konnte, wird eine Kopie e_{ik} angelegt. Diese wird zusammen mit e_i als Bindungspaar in R eingetragen, nachdem die Knotenreferenzen durch Referenzen auf die Netzwerkknoten, die den Knoten durch S_k zugeordnet wurden, ersetzt worden sind. Auch hier bezeichnet b eine Funktion, die jedem e_i eine Kante des Verkehrszeichennetzwerks zuordnet. Die angelegten Kopien der Knoten und Kanten werden in einem Schritt in das Verkehrszeichennetzwerk eingefügt. Danach wird der Verkehrszeichenknoten v des Verkehrszeichensubnetzes, sowie die CL-Kanten in das Netzwerk eingefügt. Dabei werden lediglich die CL-Kanten durch Ersetzen der k_i und e_i durch $b(k_i)$ und $b(e_i)$ geändert. Zuletzt werden alle CN- und PN-Knoten des Verkehrszeichensubnetzes, sowie dessen RL-Kanten, gelöscht.

Beispiel:

Das Verkehrszeichensubnetz, welches bereits in Abbildung 3.13 dargestellt wurde, soll jetzt in ein sehr klein gehaltenes Verkehrszeichennetzwerk eingefügt werden. Abbildung 3.14 zeigt das einzufügende Verkehrszeichensubnetz, sowie das Verkehrszeichennetz vor und nach dem Einfügen. Dabei wurden aus Gründen der Übersichtlichkeit der Darstellung die CL-Kanten sowie die SN-Knoten des Verkehrszeichennetzes ausgeblendet. Zur Identifikation der einzelnen Knoten sind diese mit Nummern versehen.

Für die Knoten des einzufügenden Netzes können zwei verschiedene Bindungsmengen S_1 und S_2 gefunden werden. Dabei wird jeweils dem Piktogrammknoden kein Knoten des Verkehrszeichennetzwerks zugeordnet, so daß für diesen eine Kopie angelegt wird. Wie das einzufügende Netz hat auch das bestehende Verkehrszeichennetzwerk zwei Knoten, die die Kontur *circle* repräsentieren. Es gibt also zwei mögliche Bindungsmengen. In Bindungsmenge S_1 wird der *äußere circle*-Knoten (1) an den *circle*-Knoten (a) des Netzwerks gebunden, der als Eintrittsknoten gespeichert

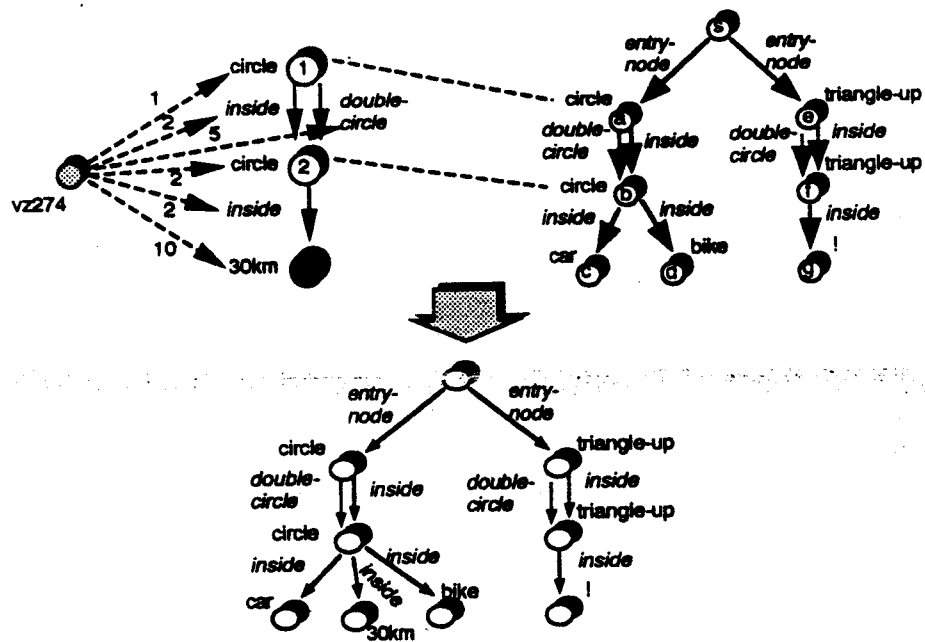


Abbildung 3.14: Einfügen eines neuen Verkehrszeichens

ist.

$$\begin{aligned} S_1 &= \{(1,a)(2,b)\} \\ S_2 &= \{(1,b)(2,a)\} \end{aligned}$$

Die Bewertungsfunktionen ergeben dann:

$$\begin{aligned} f(S_1) &= (1+1)+1 = 3 \\ f(S_2) &= (-1+(-1))+0 = -2 \end{aligned}$$

S_1 wird also ausgewählt. Die Kanten sind bis auf die *inside*-Kante zwischen Piktogramm und *innerem circle*-Knoten bereits vorhanden. Die Menge R enthält also lediglich keine Bindung für diese Kante. Für die fehlende Kante wird eine Kopie der entsprechenden Kante des Verkehrszeichensubnetzes angelegt. Danach werden die Knotenreferenzen der kopierten Kante anhand der Bindungsmenge S_1 an so verändert, daß die Kante in das Verkehrszeichennetz eingefügt wird. Dabei wird die Referenz auf Knoten (2) durch eine Referenz auf Knoten (a) und die Referenz auf den Piktogrammknoten *30km* durch eine Referenz auf die angelegte Kopie dieses Knotens ersetzt.

Zuletzt wird der Knoten *uz274* sowie die gewichteten Kanten in das Netzwerk eingefügt, indem auch hier die Knotenreferenzen anhand der Menge S_1 und die Kantenreferenzen anhand der Menge R angepaßt worden sind.

3.4.2 Klassifikation

Dieser Abschnitt befaßt sich mit der Klassifikation von Verkehrszeichen. Die Klassifikation unterteilt sich in mehrere Schritte. Zunächst wird in einem *Segmentierungsschritt* bestimmt, in welchen Regionen des Bildes sich ein Verkehrszeichen befinden kann. Diese Regionen werden in nachfolgenden Klassifikationsschritten weiter untersucht. Nach einem ersten Schritt zur Bestimmung der Verkehrszeichenklasse, der *Umrissklassifikation*, wird eine *Piktogrammklassifikation* durchgeführt⁴⁴.

Zum Schluß werden die Ergebnisse der ersten beiden Schritte zu einem Gesamtergebnis verknüpft.

3.4.2.1 Segmentierung einer Verkehrsszene

Nachdem das von der Kamera erfaßte Bild in eine symbolische Beschreibung umgeformt worden ist, sollen nun Verkehrszeichen aus den Formprimitiven extrahiert

⁴⁴Diese Komponente ist nicht realisiert, es existiert jedoch eine Schnittstelle.

und klassifiziert werden.

Als *Verkehrsszene* soll in diesem Zusammenhang eine Menge von Formprimitiven verstanden werden, die aus einem Bild der Fahrzeugumgebung, welches aus Perspektive des Fahrers aufgenommen wurde, durch einen Vorverarbeitungsprozeß bestimmt werden konnten.

Eine solche Verkehrsszene enthält weiterhin eine Vielzahl von Daten (Formprimitive), die nicht zu Verkehrszeichen gehören. Auch können in einer Verkehrsszene mehrere verschiedene Verkehrszeichen enthalten sein. Diese beiden Faktoren wirken sich auf die Komplexität der Klassifikation aus, zum einen indem Formprimitive fälschlicherweise Verkehrszeichen zugeordnet werden oder der Suchraum durch irrelevante Primitive stark vergrößert wird, und zum anderen indem mehrere in einer Verkehrsszene vorhandene Verkehrszeichen gegenseitig in Konkurrenz treten können.

Um diese Probleme einzugrenzen soll der Klassifikation eine Segmentierung der Verkehrsszene in Bereiche vorausgehen, die für die Klassifikation interessant sein könnten.

Ein *Verkehrsszenensegment*, im Folgenden auch als *Block* bezeichnet, soll einen Szenenausschnitt beschreiben, der für die nachfolgende Klassifikation interessant sein könnte, also Strukturen enthält, die auf die Existenz eines Verkehrszeichens in diesem Szenenausschnitt hinweisen.

Kriterien zur Bildung von Blöcken Betrachtet man den syntaktischen Aufbau von Verkehrszeichen (siehe auch 3.2), so fällt auf, daß zum größten Teil ineinanderliegende Strukturen⁴⁵ den Aufbau bestimmen. Außerdem wird jedem Verkehrszeichen durch die Umrisse des Schildes eine äußere Form gegeben. Diese wird in der Verkehrsszene durch ein geschlossenes⁴⁶ Formprimitiv dargestellt. Alle für das Verkehrszeichen relevanten Bildinformationen liegen innerhalb dieser äußeren Kontur.

Diese Eigenschaften der Domäne legen es nahe, die Relation *Formprimitiv A enthält Formprimitiv B*, sowie die besondere Stellung von geschlossenen Formen, auszunutzen, um ein Kriterium für die Relevanz von Formprimitiven zu erhalten. Um die Komplexität der Berechnung dieser Eigenschaft zu reduzieren, soll die folgende Definition der Relation anstelle der realen geometrischen in Beziehung zur Berechnung herangezogen werden.

⁴⁵ Beispiele: Kreis in Kreis für Verbotschilder oder Dreieck in Dreieck für Warnschilder

⁴⁶ Dabei wird hier zunächst vom Idealfall ausgegangen, d.h. der Umriss eines Verkehrsschildes wird nicht durch Überdeckung oder andere Störungen unterbrochen, so daß nur geschlossene Formprimitive betrachtet werden müssen. Mit der Behandlung von aufgerissenen Konturen befaßt sich 3.4.2.6

Definition: Umschreibendes Rechteck

Sei Γ die Menge der Formprimitive und $A \in \Gamma$. Sei Ψ_A die Menge der Bildpunkte $p \in A$.

Das umschreibende Rechteck R_A wird durch die folgenden Eckpunkte bestimmt:

$$\begin{aligned} & (\text{Min}\{x \mid (x, y) \in \Psi_A\} , (\text{Min}\{y \mid (x, y) \in \Psi_A\})) \\ & (\text{Min}\{x \mid (x, y) \in \Psi_A\} , (\text{Max}\{y \mid (x, y) \in \Psi_A\})) \\ & (\text{Max}\{x \mid (x, y) \in \Psi_A\} , (\text{Max}\{y \mid (x, y) \in \Psi_A\})) \\ & (\text{Max}\{x \mid (x, y) \in \Psi_A\} , (\text{Min}\{y \mid (x, y) \in \Psi_A\})) \end{aligned}$$

Definition: Formprimitiv A enthält Formprimitiv B

Sei Γ die Menge der Formprimitive, $\Gamma_G \subset \Gamma$ die Menge der geschlossenen Formprimitive und $A, B \in \Gamma$. Ferner bezeichne in die natürliche Lagerrelation, wie sie zwischen zwei zweidimensionalen geometrischen Objekten definiert ist.

Dann gilt A *enthält* B genau dann, wenn gilt:

$$A \in \Gamma_G \wedge R_B \text{ in } R_A$$

Diese Einschränkung der *enthält* - Beziehung erhält *nicht* die Korrektheit, d.h. es kann durchaus vorkommen, daß zwei Formprimitive durch das oben definierte Prädikat fälschlicherweise als ineinanderliegend bezeichnet werden, wie dies in Abbildung 3.15 dargestellt wird.

Andererseits kann es nicht vorkommen, daß zwei Formprimitive ineinanderliegen, die vom oben definierten Prädikat zurückgewiesen werden. Daraus ergeben sich folgende Beziehungen zwischen dem definierten Begriff *enthält* und der *in* Beziehung:

$$\begin{array}{lll} A \text{ in } B & \longrightarrow & B \text{ enthält } A \quad (\text{vollständig}) \\ A \text{ enthält } B & \not\longrightarrow & B \text{ in } A \quad (\text{nicht korrekt}) \end{array}$$

Es stellt sich die Frage, ob es überhaupt zulässig sein kann, eine unkorrekte Berechnung als Grundlage für die Segmentierung einer Verkehrsszene zu benutzen. Dazu lassen sich einige Punkte anführen, die die Benutzung rechtfertigen.

1. Die Berechnung vereinfacht das Prädikat, so daß es wesentlich schneller berechenbar wird. Betrachtet man das ursprüngliche Prädikat, so verdeutlicht Abbildung 3.15b die Komplexität seiner Berechnung.