



KfK 4941
Oktober 1991

Simulation der Zielfahrt eines autonomen Fahrzeugs in labyrinthartiger Umgebung durch KISMET

H. Knüppel, U. Kühnapfel, D. Smidt
Institut für Reaktorentwicklung

Kernforschungszentrum Karlsruhe

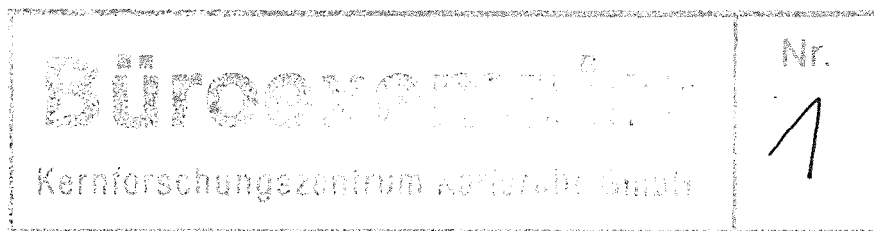
KERNFORSCHUNGSZENTRUM KARLSRUHE

Institut für Reaktorentwicklung

KfK 4941

**Simulation der Zielfahrt eines autonomen Fahrzeugs
in labyrinthartiger Umgebung durch KISMET**

H. Knüppel, U. Kühnappel, D. Smidt



Kernforschungszentrum Karlsruhe GmbH, Karlsruhe

Als Manuskript gedruckt
Für diesen Bericht behalten wir uns alle Rechte vor

Kernforschungszentrum Karlsruhe GmbH
Postfach 3640, 7500 Karlsruhe 1

ISSN 0303-4003

Simulation der Zielfahrt eines autonomen Fahrzeugs in labyrinthartiger Umgebung durch KISMET.

H.Knüppel, U.Kühnapfel, D. Smidt
Institut für Reaktorentwicklung
Kernforschungszentrum Karlsruhe

Zusammenfassung.

Unter Benutzung der besonderen Fähigkeiten des KISMET-Programmpakets für geometrische Operationen und graphische Präsentation wurde ein Algorithmus für die kollisionsfreie Zielfahrt eines autonomen Fahrzeugs entwickelt, implementiert und an KISMET angebunden. Er besteht aus einem einfachen globalen Routenplaner, der ohne Berücksichtigung der endlichen Dimensionen des Fahrzeugs eine Datei von Zwischenzielen erzeugt, die dann in einen sensorgestützten lokalen Routenplaner eingebracht werden. Dieser weicht aufgrund seines empfangenen Sensormusters den Hindernissen aus und erzeugt für jeden Zeitschritt ein Bewegungsmuster, das die Zielverfolgung einschließt. Einige Beispiele und ein Videofilm zeigen, daß das Ziel auch in komplexer und labyrinthartiger Umgebung auf dem kürzesten Wege kollisionsfrei erreicht wird.

Abstract.

By using the special capabilities of the KISMET software-Package and hardware for geometric operations and graphical presentation we have developed, implemented, and linked to KISMET an algorithm for the collision-free target-oriented driving of an autonomous vehicle. The algorithm employs a simple global route-planner. It creates the global path neglecting the finite vehicle dimensions as input to the sensor-based local route-planner. The local planner for each time step transforms the sensor pattern, received by a number of ultrasonic sensors, to the movement-pattern. The target oriented global information influences the local operations. Some examples and a video demonstrate, the the target will be reached collision free and close to the shortest path even in a labyrinthic environment.

Simulation der Zielfahrt eines autonomen Fahrzeugs in labyrinthartiger Umgebung durch KISMET.

H. Knüppel, U. Kühnapfel, D. Smidt
Institut für Reaktorentwicklung.
Kernforschungszentrum Karlsruhe.

1. Einleitung.

Als Werkzeug zur graphischen und kinematischen Simulation bewegter Objekte hat sich KISMET (Kinematic Simulation, Monitoring and Off-Line Programming Environment for Telerobotics) [1-4] in vielen Anwendungsfällen bewährt. Die vorliegende Arbeit soll am Beispiel eines autonomen Fahrzeugs demonstrieren, wie flexibel der einmal entwickelte KISMET-Apparat, ohne ihn selbst verändern zu müssen, an neue Aufgabenstellungen angepaßt und dazu mit zusätzlicher Intelligenz ausgerüstet werden kann.

Die Entwicklung und Erprobung der erforderlichen Software kann unabhängig von KISMET erfolgen; im vorliegenden Beispiel wurde hierfür ein PC verwendet.

Bereits früher wurde in einer zusammen mit der Firma Pietzsch durchgeführten Studie [5] deren auf einem neuronalen Netz beruhender Steueralgorithmus für ein autonomes Fahrzeug erfolgreich mit KISMET verküpft. Das neuronale Netz war dazu auf einem eigenen Rechner implementiert. Seine Fähigkeiten sind aber darauf beschränkt, unter Auswertung einer Anzahl (eines "Musters") von Ultraschall-Abstandssignalen das Fahrzeug in der Nähe von Hindernissen Ausweichbewegungen zur Kollisionsvermeidung ausführen zu lassen. Ein zielgerichtetes Fahren zwischen vorgegebenen Start- und Zielpunkten ist nicht möglich, die Reaktion auf die Hindernisse ist rein lokal.

Im Abschnitt 2 soll ein Verfahren zur Generierung einer globalen Route zwischen Start und Ziel beschrieben werden, das die auf diese oder ähnliche Weise verfügbaren Fähigkeiten der lokalen Hindernisvermeidung, wie sie in Abschnitt 3 dargestellt werden, benutzt und deshalb sehr einfach wird. Abschnitt 4 beschreibt die Verknüpfung mit KISMET und erläutert die Arbeitsweise des Gesamtverfahrens an einigen Beispielen.

2. Die globale Routenplanung.

Bestünde das Fahrzeug aus einem dimensionslosen Punkt, so setzte sich die kürzeste Route zwischen zwei Punkten in der Gegenwart von Hindernissen in einer zweidimensionalen Umgebung aus Hinderniskanten und ihren Verbindungen zusammen und die globale Routenplanung wäre ein einfach zu lösendes Problem. Erst die Berücksichtigung der endlichen Fahrzeugdimensionen und der Wunsch, die globale Route von vorneherein kollisionsfrei zu planen, erschwert die Aufgabe erheblich und erhöht den Rechenaufwand wesentlich. Es gibt verschiedene Verfahren zur Lösung dieses Problems:

a.. Man vergrößert die Hindernisse unter Berücksichtigung aller Positionen und Orientierungen um die Fahrzeugdimension, transformiert das Problem dadurch in den sog. *Konfigurationsraum* [6], [7] und kann das Fahrzeug dann wieder als Punkt behandeln. Leider nimmt dadurch die Zahl der Freiheitsgrade zu und durch Hinzunahme von Drehungen des Fahrzeugs werden die Hinderniskanten nichtlinear.

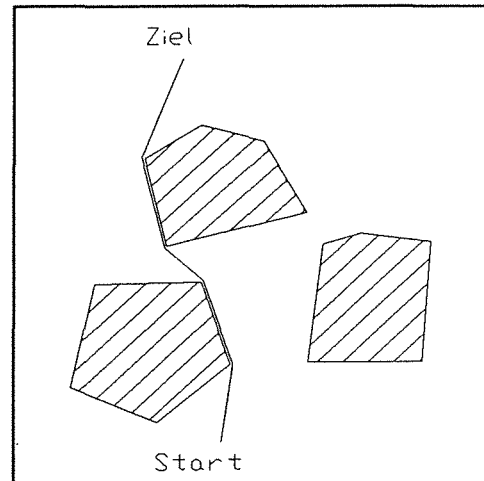


Bild 1. Kürzeste Route zwischen Polygonen.

Bild 1 zeigt eine typische, nach dieser Methodik im zweidimensionalen Raum berechnete Route. Bei mehr als zwei Dimensionen wird die Auswertung dadurch erschwert, daß die Kanten der Hindernisobjekte nicht mehr Teil der kürzesten Route sein müssen.

b. Man bildet das *Voronoi-Diagramm* der Anordnung der Hindernisse, das ist die Menge der Punkte, die von zwei oder mehr Hinderniskanten den gleichen Abstand haben. Routen aus Teilen eines Voronoi-Diagramms passieren Engstellen mit einem definierten größtmöglichen Abstand und erlauben in Verbindung mit einer geeigneten Drehung des Fahrzeugs [8] die Prüfung auf Passierbarkeit. Auch dieses Verfahren erfordert einen erheblichen Rechenaufwand ($n \log(n)$ bzw. $n \log^2(n)$), wo n die Zahl der Eckpunkte als polygonal angenommener

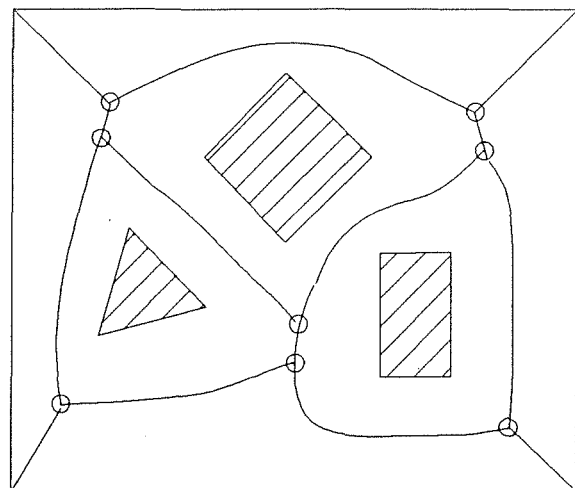


Bild 2. Voronoi-Diagramm

kursiven Prozedur die kürzeste Route durch Gradientenverfolgung unmittelbar auslesen und als Datei einer Serie konsekutiver Punkte speichern. Wie aus Bild 3 hervorgeht, werden zur Beschleunigung der Rechnung nur die vier nächsten Nachbarpunkte oben, unten, rechts und links berücksichtigt und die sich ergebende Route läuft im Zickzack. In Verbindung mit der lokalen Intelligenz des Fahrzeugs werden aber nicht alle Punkte benötigt, sodaß sich ein gewisser Ausgleich ergibt. Im nächsten Abschnitt wird gezeigt werden, wie eine Datei, die jeden zweiten oder dritten Punkt der globalen Route enthält, dem Fahrzeug mit seinen lokalen Fähigkeiten ein komplettes Zielfahren ermöglicht. Die Manövrierbarkeit des Fahrzeugs kommt hier nur insoweit ins Spiel, als die Abstände der Punkte des verwendeten Rasters etwas größer als die Abmessungen des Fahrzeugs sein sollten.

3. Kombination von globaler und lokaler Routenplanung.

Die lokale Routenplanung bzw. Hindernisvermeidung basiert auf der Auswertung der Signale von Abstandssensoren. Die hier verwendete Anordnung von 7 Ultraschallsensoren ist aus den Bildern 4-6 zu erkennen. Jedes Muster aus gemessenen Abständen zwischen dem Fahrzeug und den Objekten seiner Umgebung ergibt eine Vorschrift für die Größe und Richtung der Verschiebung des Fahrzeugs im nächsten Zeitschritt, die man als das jeweilige Bewegungsmuster auffassen kann. Laufend wird so aus dem *Sensormuster* ein *Bewegungsmuster* generiert.

Man kann die hierfür geltenden Regeln entweder explizit formulieren oder auch, wie in dem schon erwähnten Fall von Pietzsch [5] über eine Auswahl von Beispielen in ein neuronales Netz einbringen.

Um zu einer zielgerichteten Fahrt zu kommen, muß die *globale* Routenplanung in die Erzeugung des Bewegungsmusters einbezogen werden. Es liegt nahe, hierzu das Sensormuster um Komponenten zu ergänzen, die die Richtung und Entfernung zu einem Ziel beinhalten und den Regelsatz für die Erzeugung des Bewegungsmusters entsprechend erweitern. Da zur Erreichung des endgültigen Ziels zwischen den Hindernissen möglicherweise Umwege gefahren werden müssen, ist dessen (des endgültigen Ziels) Richtung nicht geeignet, das Sensormuster für den jeweils aktuellen Fall sinnvoll zu ergänzen und dem Fahrzeug die im Augenblick erforderliche Fahrtrichtung aufzuzwingen. Steht aber, wie oben beschrieben, die globale Route mit einer hinreichenden Anzahl von Zwischenpunkten zur Verfügung, so können genau diese als Zwischenziele für die Festlegung der momentanen Fahrtrichtung dienen. In dieser Vorgehensweise liegt

die Besonderheit unseres aus lokalen und globalen Strategien kombinierten Routenplaners.

Im Prinzip ergibt sich dann der folgende Algorithmus:

1. Initialisiere den ersten Punkt der globalen Route als erstes Zwischenziel.
2. Erzeuge unter Auswertung des Sensormusters und der Richtung des Zwischenziels das Bewegungsmuster und fahre das Zwischenziel an.
3. Unterschreitet der Abstand zum Zwischenziel einen gegebenen Betrag, setze den nächsten Punkt der Globalroute als Zwischenziel und wiederhole 2.
4. Ist das letzte Zwischenziel (=Endziel) erreicht, ist die Zielfahrt erfolgreich abgeschlossen.

Im Interesse einer größeren Transparenz haben wir die Regeln für die Erzeugung des Bewegungsmusters aus dem Sensormuster explizit und nicht durch ein neuronales Netz definiert. Dabei wird auch deutlich, wie die vorgeschlagene Verwendung von Zwischenzielen und die Hindernisvermeidung zusammenwirken.

Wir unterscheiden die folgenden Fahrklassen, zwischen denen auf definierte Weise umgeschaltet werden kann:

Klasse 1: Freie Fahrt ("free_cruise"), die Sensoren detektieren keine Hindernisse innerhalb ihrer Reichweite. Während jedes Zeitschrittes wird ein Ortsschritt in Richtung auf das geltende Zwischenziel zurückgelegt.

Klasse 2: Die Sensoren entdecken ein Hindernis ("hit_object"). Über die geometrische Relation der gemessenen Abstände wird ermittelt, ob und unter welchem Winkel das

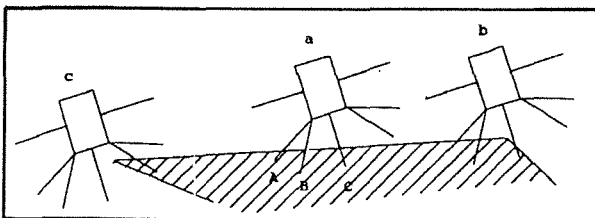


Bild 4. Unterschiedliches Auftreffen auf ein Hindernis.

Fahrzeug sich einer geraden Kante (a in Bild 4) oder aber einer Ecke des Hindernisses (b) nähert oder ob es nur streifend vorbeifährt. Im letzteren Fall wird lediglich eine leichte Ausweichbewegung ausgeführt, dann geht die Fahrt gemäß Klasse 1 in Richtung Zwischenziel weiter (c). Bei Vorliegen der ersteren Situation wird in Klasse 3 umgeschaltet.

Klasse 3: Wenn ein Hindernis dem Fahrzeug den Weg versperrt, folgt es so lange der Kante dieses Objektes, bis es wieder frei in Richtung Ziel fahren kann. In Klasse 3 wird dies vorbereitet ("prep_following"). Dazu muß die Richtung der zu verfolgenden Kante berechnet werden. Ob das Fahrzeug das Objekt dabei im Uhrzeiger- oder Gegenuhrzeigersinn umfährt, hängt wieder von der Richtung des Zwischenziels ab.

Klasse 4: Wenn nicht das nächste Zwischenziel direkt angefahren werden kann, folgt

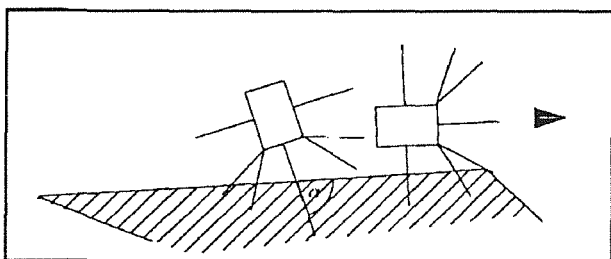


Bild 5. Ausrichten parallel zur Hinderniskante.

das Fahrzeug in einem definierten Abstand der Hinderniskante ("perhaps_along") und orientiert sich dabei mit seinen seitlichen Sensoren.

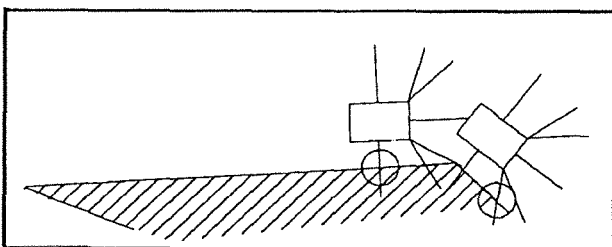


Bild 6. Umfahren konvexer Ecken.

Klasse 5: Dabei umrundet das Fahrzeug auch konvexe Ecken des Objekts und bestimmt zunächst aus den Sensordaten den auf der Ecke liegenden Mittelpunkt der Umrundungskurve ("rumprep"), dann mit

Klasse 6: ("turn_about") mit jedem Zeitschritt einen Ortsschritt in der Umrundung der Ecke, bis die Sensoren feststellen, daß wieder in Klasse 4 der Kante gefolgt werden kann (und die freie Fahrt zu nächsten Zwischenziel noch nicht möglich ist).

Klasse 7: Analog zum Umfahren konvexer (äußerer) Ecken können auch konkave Ecken erkannt und in definiertem Abstand passiert werden, obwohl hier ein "Abschneiden" der Ecke meist kürzere Wege ermöglicht.

Weitere Ergänzungen sind möglich, wie z. B. das Passieren von Kanälen, wo die Sensoren auf beiden Seiten die Nähe von Wänden signalisieren u. a. m..

Die explizite Definition der Regeln für die Übertragung des Sensormusters in das Bewegungsmuster macht aber jetzt eine zusätzliche Voraussetzung für die Zielfahrt deutlich: *Um auf einer möglichst kurzen Route an das Ziel zu kommen, ist auch die in den Klassen 3-6 definierte Fähigkeit des Kantenfolgens notwendig.* Es ist nicht zu erkennen, wie dies mit einem neuronalen Netz realisiert werden kann.

4. Integration in das KISMET-System.

Für die Simulation der beschriebenen Algorithmen sind neben der anspruchsvollen graphischen Darstellung von Fahrzeug und Hindernissen die Sensoroperationen von Bedeutung. Sie werden auf Schnitte zwischen den Sensorstrahlen und den Objekten der Umgebung zurückgeführt, für die der KISMET-Rechner aufgrund seiner speziellen Hard- und Software besonders gut geeignet ist. Dazu wurden, wie in Bild 7 gezeigt, im Wesentlichen zwei zusätzliche Programm-Module an KISMET angebunden.

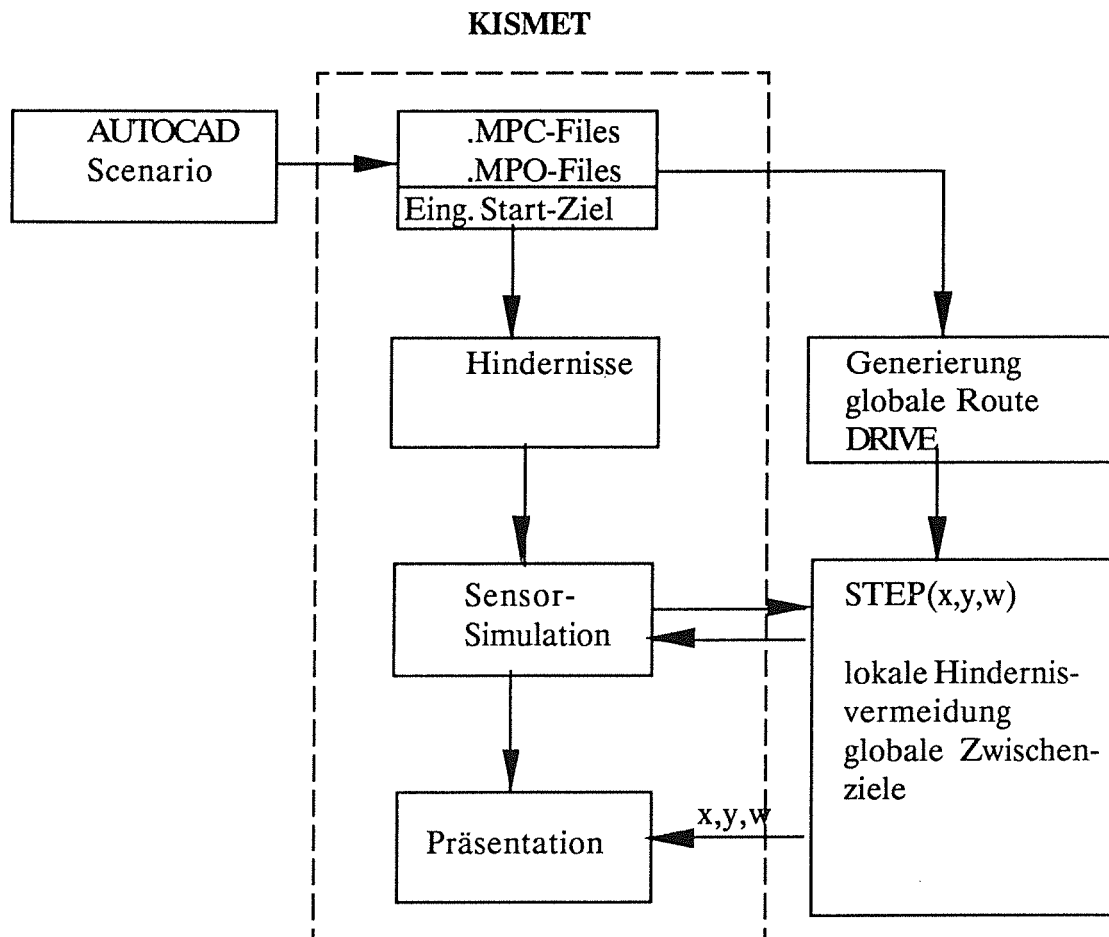


Bild 7. Die Anbindung des Zielfahrtalgorithmus an KISMET.

Der gestrichelt umrandete Kasten in Bild 7 kennzeichnet das KISMET-Programmpaket. Die hier graphisch dargestellten Objekte sind durch ASCII-Files definiert. Für jedes Objekt gibt es ein File mit dem Zusatz .MPO, während die Lage eines Referenzpunktes der Objekte durch ein File vom Typ .MPC angegeben wird. Beim gegenwärtigen Stand ist die Darstellung auf quaderförmige Objekte beschränkt, eine Erweiterung nach den KISMET-Konventionen auf andere Objekttypen ist prinzipiell möglich. Der Benutzer baut sich sein Szenario als zweidimensionale Anordnung mit einem CAD-System. Für unser Beispiel wurde AUTOCAD benutzt und ein DXF-File erzeugt, das mit einem kleinen PC-Programm interpretiert und in die .MPC und .MPO-Files umgewandelt wird. Die Höhendimension, die allein für die Präsentation gebraucht wird, kann automatisch oder von Hand ergänzt werden. Die Daten werden dann über das lokale Netzwerk an KISMET übertragen.

Zu KISMET wird ein zweiter Prozeß (DRIVE) gestartet, der über zwei Kommunikationskanäle (UNIX-Message Kommunikation) mit KISMET verbunden ist. Über eine grafische Eingabe werden die Zielpunkt-Koordinaten des Fahrzeugs definiert. Als Startpunkt für die Routenplanung wird die augenblickliche Fahrzeugposition verwendet. Die Start- und Zielpunktkoordinaten werden nun von KISMET an DRIVE gesendet und gleichzeitig damit die Routenplanung gestartet. In DRIVE wird eine globale Route zwischen Start- und Zielpunkt geplant. Zwischenziele werden dabei, wie bereits oben beschrieben, in einer Datei temporär zwischengespeichert. KISMET liefert nun gleichzeitig vom Startpunkt ausgehend ein Sensormuster der simulierten Ultraschall-Abstandssensoren für die augenblickliche Fahrzeugposition an den Prozeß DRIVE. Hier werden in der Prozedur STEP anhand der von KISMET gelieferten Sensormuster und den Koordinaten des nächsten Zwischenziels neue Fahrzeug-Sollkoordinaten (x,y und Orientierungswinkel) für den nächsten Zeitschritt berechnet. Diese neue Fahrzeugposition wird an KISMET zurückgesendet, das Fahrzeug entsprechend neu positioniert und ein neues Sensormuster bestimmt. Damit kann das simulierte Fahrzeug mit den in KISMET verfügbaren grafischen Darstellungsmöglichkeiten [1] zwischen den Hindernissen präsentiert werden. Dieser Vorgang wird solange wiederholt, bis der zu Beginn des Planungszyklus vom Bediener definierte Zielpunkt erreicht ist.

Neben der grafischen Darstellung des Bahnplanungsvorgangs und der Bereitstellung von Sensormustern führt KISMET außerdem eine numerische Kollisionserkennung zwischen dem Fahrzeug und den Hindernissen durch. Diese Information diene während der Implementierungs- und Testphase zur Optimierung der Berechnungsparameter für die Prozedur DRIVE.

5. Ergebnisse.

Die folgenden Bilder zeigen einige Ergebnisse, wie sie sich in der KISMET-Grafik darstellen. Die rein zweidimensionale Bahnplanung kann dazu in eine dreidimensionale Umgebung eingebettet werden. Bild 8 zeigt das Modellfahrzeug, das in Bild 9 in einer Umgebung aus dreidimensionalen Blöcken dargestellt wird, die man z.B. als Straßen und Gebäude einer Stadt interpretieren kann.

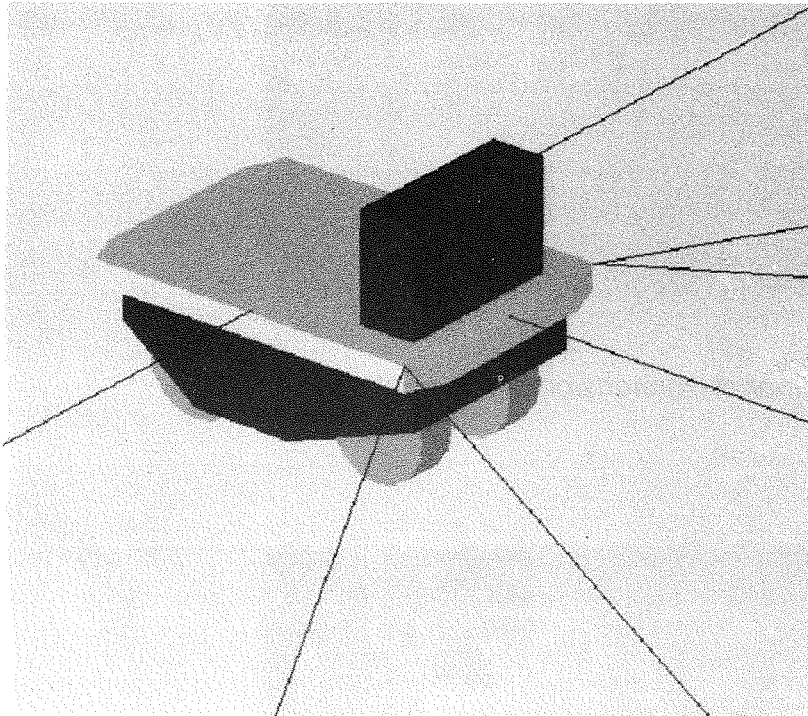


Bild 8. Fahrzeugmodell.

Bild 10 und 11 zeigen das Fahrzeug während es seine Route zurücklegt. In Bild 10 führt diese durch eine "Stadt", in Bild 11 durch ein "Labyrinth". Der bereits zurückgelegte Weg ist als ausgezogene Linie zu erkennen, während die durch die globale Routenplanung gelieferten Zwischenziele als schwarze Punkte dargestellt sind. Da die globale Route die endliche Größe des Fahrzeugs nicht berücksichtigt, muß es die Hindernisse mit seinen Ultraschallsensoren "abtasten". Dies äußert sich in der zurückgelegten Route in gelegentlichen Ausbuchtungen und Knicken. Dies ist ein Preis, der für den Verzicht auf ein aufwendigeres globales Planungsverfahren erbracht werden muß. Man erkennt auch, daß die durch die globale Routenplanung nicht genau auf der bereits zurückgelegten Route liegen, da die lokale Hindernisvermeidung Abweichungen erzeugen kann.

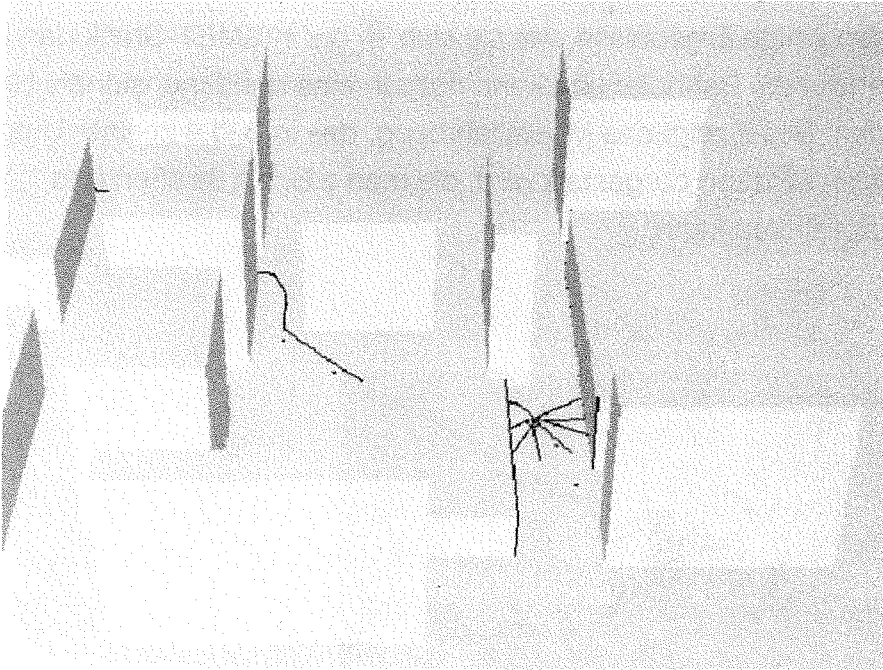


Bild 9. Das Fahrzeug in einer dreidimensionalen Umgebung.

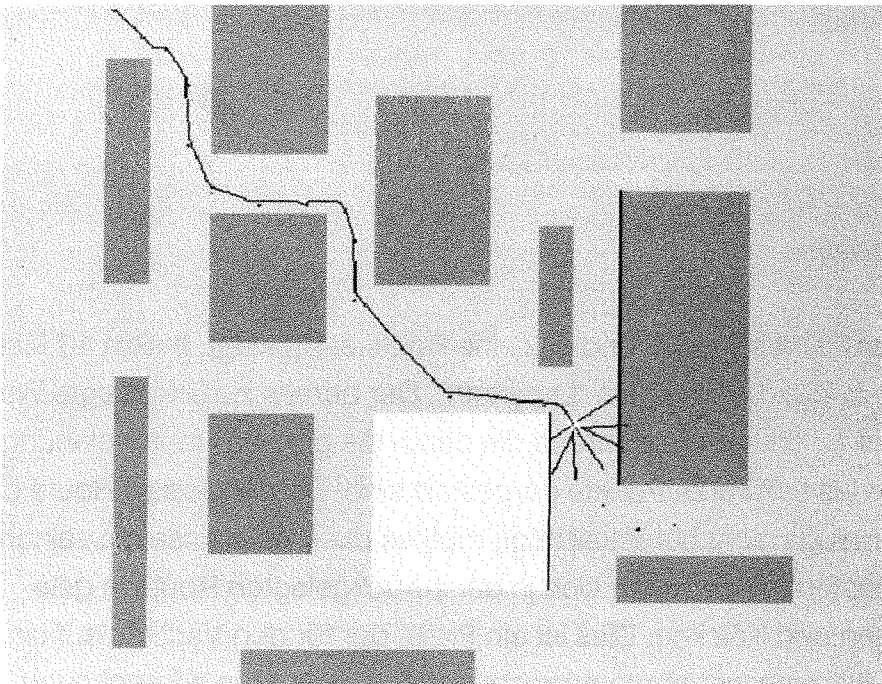


Bild 10 Bahnplanung zwischen Gebäuden.

6. Literatur.

- [1] Kühnapfel, U., KISMET-3D-Grafik zur Planung, Programmierung und Überwachung von Telerobotics-Applikationen, S. 71-86, VDI-Berichte Nr. 861.3, 1990.
- [2] Kühnapfel, U., Leinemann, K., Schlechtendahl, E. G., Graphics Support for JET Boom Control. Proc. ANS 2nd Int. Top. Meeting on Remote Systems and Robotics in Hostile Environments, S. 28-34, Pasco, Wash., (1988).
- [3] Kühnapfel, U., Ludwig, A., Graphics and CAD Support for NET / ITER Boom Control, Proc. 16th SOFT, North-Holland, Vol. 2, S. 1347-1352, London, Sept. 3-7, 1990.
- [4] Leister, P., Kühnapfel, U., Ludwig, A., Computer Aided Simulation of a Remote Steam-Jet Exchange in a Dissolver Cell, PROC: ANS 4th Topical Meeting on Robotics and Remote Systems, S. 353-364, Albuquerque, NM, (1991).
- [5] Hohn, H.U., Knüppel, U., Kühnapfel, U., Müller, W., Opitz, R., Weber, Th., Ein mit neuronaler Software gesteuertes Fahrzeug, simuliert auf der KISMET-Robotersimulationsumgebung, KfK-Bericht 4892, August 1991.
- [6] Lozano-Perez, T., Spatial Planning: A Configuration Space Approach, IEEE Trans. on Computers, C-32, No. 2, 108-120, IEEE, Feb. 1983
- [7] Udupa, S.M., Collision Detection and Avoidance in Computer Controlled Manipulators, 5th Int. Joint Conf. on Artificial Intelligence, Vol. 2, 737-748, MIT, Cambridge, Mass., Aug. 22-25, 1977
- [8] Takahashi, O., Schilling, R.J., Motion Planning in a Plane Using Generalized Voronoi Diagrams, IEEE Transactions on Robotics and Automation, Vol. 5, No. 2, 143 - 150, IEEE, April 1989
- [9] Kirkpatrick, D.G., Efficient Computation of Continuous Skeletons, 20th Annual Symp. on Foundations of Computer Science, IEEE 79CH1471-2C, 18-27, 1979
- [10] Soetadji, T., Methode zur Lösung des Routenplanungsproblems für ein Navigationssystem eines autonomen Fahrzeugs, VDI Fortschrittberichte, Reihe 10, Informatik Kommunikationstechnik Nr. 70, VDI-Verlag, Düsseldorf, 1987

- [11] Noborio, H., Naniwa, T., Arimoto, S., A Feasible Motion-Planning Algorithm for a Mobile Robot on a Quadtree Representation, Proc. 1989 IEEE Int. Conf. on Robotics and Automation, 327-332, Scottsdale ARIZ, 1989

- [12] Wong, E.K., Fu, K.S., A Hierarchical Orthogonal Space Approach to Three-Dimensional Path Planning, IEEE Journal of Robotics and Automation, Vol. RA-2, No. 1, 42-53, 1986

- [13] Noborio, H., Naniwa, T., Arimoto, S., A Quadtree-Based Path-Planning Algorithm for a Mobile Robot, Journal of Robotic Systems, 7(4), 555-574, 1990

- [14] Fredman, M.L., Tarjan, R.E., Fibonacci Heaps and Their Uses in Improved Network Optimization Algorithms, Proc. 25th Ann. IEEE Symp. on Foundations of Computer Science, 338 - 345, IEEE, 1984