

# Informatik aktuell

---

Herausgeber: W. Brauer  
im Auftrag der Gesellschaft für Informatik (GI)

Wolfgang A. Halang  
Peter Holleczek (Hrsg.)

# Aktuelle Anwendungen in Technik und Wirtschaft

Fachtagung des GI/GMA-Fachausschusses  
Echtzeitsysteme (real-time)  
Boppard, 27. und 28. November 2008



**VDE**

VDI/VDE-Gesellschaft  
Mess- und Automatisierungstechnik



Springer

## Herausgeber

Wolfgang A. Halang  
FernUniversität in Hagen  
Lehrstuhl für Informationstechnik,  
insb. Realzeitsysteme  
58084 Hagen  
wolfgang.halang@fernuni-hagen.de

Peter Holleccek  
Regionales Rechenzentrum  
der Universität Erlangen-Nürnberg  
Kommunikationssysteme  
Martensstraße 1, 91058 Erlangen  
peter.holleccek@rrze.uni-erlangen.de

## Programmkomitee

|                  |               |
|------------------|---------------|
| R. Arlt          | Hannover      |
| J. Bartels       | Krefeld       |
| J. Benra         | Wilhelmshaven |
| F. Dressler      | Erlangen      |
| W. Gerth         | Hannover      |
| W.A. Halang      | Hagen         |
| H. Heitmann      | Hamburg       |
| P. Holleccek     | Erlangen      |
| R. Müller        | Furtwangen    |
| F. Saglietti     | Erlangen      |
| G. Schiedermeier | Landshut      |
| U. Schmidtman    | Emden         |
| U. Schneider     | Mittweida     |
| H.F. Wedde       | Dortmund      |
| H. Windauer      | Lüneburg      |
| D. Zöbel         | Koblenz       |

Bibliographische Information der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

CR Subject Classification (2001): C3, D.4.7

ISSN 1431-472-X

ISBN-13 978-3-540-85323-7 Springer Berlin Heidelberg New York

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils geltenden Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

Springer Berlin Heidelberg New York  
Springer ist ein Unternehmen von Springer Science+Business Media  
[springer.de](http://springer.de)

© Springer-Verlag Berlin Heidelberg 2009  
Printed in Germany

Satz: Reproduktionsfertige Vorlage vom Autor/Herausgeber  
Gedruckt auf säurefreiem Papier SPIN: 12466221 33/3180-543210

# Vorwort

Der seit 1980 immer im Spätherbst zu Boppard stattfindende und mittlerweile schon zur Tradition gewordene „PEARL-Workshop über Realzeitsysteme“ zunächst des PEARL-Vereins e.V. und dann der Fachgruppe Echtzeitprogrammierung und PEARL der Gesellschaft für Informatik e.V. (GI) kommt in diesem Jahr mit einigen Neuerungen daher.

Da ist zuerst der neue Untertitel „Echtzeit 2008“ anstelle von „PEARL 2008“ zu nennen, der jedoch mitnichten eine Abkehr von der Echtzeitprogrammiersprache PEARL bedeutet. Im Gegenteil, die Fachgruppe hat sich die Pflege dieser bewährten und mittlerweile auf ihrem Gebiet weltweit konkurrenzlos dastehenden höheren Programmiersprache auf ihre Fahnen geschrieben, wozu auch die in einigen Jahren vorzunehmende Weiterentwicklung der Norm DIN 66253-2 dienen wird.

Der neue Untertitel reflektiert nur die durch die Fusion mit der Fachgruppe Echtzeitsysteme der GI vollzogene organisatorische Veränderung, durch die sie zu einem gemeinsamen Fachausschuss unter dem Namen Echtzeitsysteme der GI und der GMA, der Gesellschaft für Mess- und Automatisierungstechnik, geworden ist, wobei es sich bei letzterer wiederum um eine gemeinsame Fachgesellschaft des Vereins Deutscher Ingenieure e.V. (VDI) und des Verbandes der Elektrotechnik Elektronik Informationstechnik e.V. (VDE) handelt. Nach dem Motto „nomen est omen“ stellen Fachausschuss und Fachtagung mithin die ersten Adressen für die Gebiete der Echtzeit- und der eingebetteten Systeme im deutschsprachigen Raum dar.

Der Fachausschuss fühlte sich schon immer der Nachwuchsförderung verpflichtet. Als konkrete Maßnahme dazu hat er in diesem Jahr zum ersten Male drei Diplomarbeitspreise für Studienabschlussarbeiten ausgelobt und vergeben. Wichtiger aber als der Geldpreis ist, dass die Preisträger zur aktiven Teilnahme an der Tagung eingeladen werden. Dies umfasst die Erstellung schriftlicher und im vorliegenden Tagungsband abgedruckter Beiträge sowie deren mündliche Präsentation im Rahmen der vierten Sitzung bei der Tagung in Boppard.

Eine weitere Neuerung in diesem Jahr ist schließlich, beim Tagungsthema zur ersten Male den Blick über den Tellerrand von den bisher allein betrachteten eher technischen Anwendungen von Echtzeitsystemen hin zu solchen im Wirtschaftsleben und Finanzwesen erweitert zu haben. Die dazu eingegangenen Beiträge sind in der ersten Sitzung zusammengefasst. Sie beschäftigen sich mit Fragen aus Logistik und Verkehr.

Traditionsgemäß ist eine Sitzung der Hochschulausbildung im Bereich der Echtzeitsysteme gewidmet. Hier erfahren wir auch über den Stand der Entwicklung eines neuen, äußerst preisgünstigen PEARL-Rechners. Die übrigen beiden Sitzungen lassen sich unter dem Leitthema Vorhersehbarkeit des Ausführungsverhaltens einordnen. Eine beleuchtet dies für Kommunikationsvorgänge und

die andere hinsichtlich Befehlsverarbeitung und Speicherverwaltung im Rechnerkern.

Mit leichtem Druck auf einige Autoren wurde erreicht, dass sie von Word abließen und alle Beiträge zur Verarbeitung mit dem Formatierungssystem LaTeX vorbereiteten. Im Ergebnis kommen alle im Tagungsband enthaltenen Kapitel in einheitlichem Erscheinungsbild daher. Die Herausgeber bedanken sich bei den Autoren sehr herzlich für dieses Entgegenkommen und für die zumeist vorzeitige Ablieferung ihrer Beiträge. Im Ergebnis gestaltete sich die Fertigstellung des druckfertigen Manuskripts als völlig stressfrei. Unser besonderer Dank gilt Frau Dipl.-Ing. Jutta Düring für ihre wertvolle Mitarbeit bei der Formatierung des Tagungsbandes.

Das Programmkomitee und das Leitungsgremium des Fachausschusses wünschen den Teilnehmern der Fachtagung einen intensiven und anregenden Erfahrungsaustausch sowie den Lesern des vorliegenden Bandes eine interessante Lektüre.

Hagen

Erlangen

im September 2008

Wolfgang A. Halang

Peter Holleczek

# Inhaltsverzeichnis

## Logistik und Verkehr

- Echtzeiterkennung von befahrbaren Bereichen in urbanen Szenarien . . . . . 1  
*Kai Berger, Christian Linz, Christian Lipski, Timo Stich,  
Marcus Magnor*
- Berührungslose Winkelbestimmung zwischen Zugfahrzeug und Anhänger . 11  
*Jacek Schikora, Uwe Berg, Dieter Zöbel*
- Von eingebetteten Systemen zu Cyber-Physical Systems . . . . . 21  
*Horst F. Wedde, Sebastian Lehnhoff, Christian Rehtanz, Olav Krause*

## Ausbildung

- Eine Plattform für die studentische Ausbildung im Echtzeit- und  
Feldbusbereich . . . . . 31  
*Markus Hilmer, Karl-Heinz Niemann*
- HighTecBot . . . . . 41  
*Martina Lehser, Benjamin Behringer, Eric Wagner*

## Kommunikation

- Einsatz einer Echtzeit-Publish/Subscribe-Kommunikation für die  
Teleoperation mobiler Roboter . . . . . 51  
*Timo Lindhorst, André Herms, Michael Schulze*
- Improving IEEE 802.15.4 for Low-Latency Energy-Efficient Industrial  
Applications . . . . . 61  
*Feng Chen*

## Studentische Beiträge

- Virtualisierung im Echtzeitbereich . . . . . 71  
*Andreas Hollmann*
- Fehlertoleranzanalyse des FlexRay Startup-Prozesses . . . . . 81  
*Sven Bünte, Paul Milbredt*
- Konzeption und Entwicklung eines echtzeitfähigen Lastgenerators für  
Multimedia-Verkehrsströme in IP-basierten Rechnernetzen . . . . . 91  
*Andrey Kolesnikov*

## **Forschung**

|   |     |
|---|-----|
| Dynamische Verwaltung Virtueller Speichers für Echtzeitsysteme . . . . .  | 101 |
| <i>Martin Böhnert, Thorsten Zitterell, Christoph Scholl</i>               |     |
| WCET-Analyseverfahren in der automobilen Softwareentwicklung . . . . .    | 111 |
| <i>Martin Däumler, Robert Baumgartl, Matthias Werner</i>                  |     |
| Echtzeitrechnerarchitektur mit exakt vorhersehbarer Befehlsverarbeitung . | 121 |
| <i>Helmut Stieger</i>   |     |

# Echtzeiterkennung von befahrbaren Bereichen in urbanen Szenarien

Kai Berger, Christian Linz, Christian Lipski, Timo Stich, Marcus Magnor

Institut für Computergrafik,  
Mühlenpfordtstraße 23, 38106 Braunschweig  
{berger,linz,lipski,stich,magnor}@cg.tu-bs.de

**Zusammenfassung.** Unser Artikel beschreibt ein Echtzeitverfahren zur kamerabasierten Fahrbereichserkennung, welches in urbanen bzw. ländlichen Fahr Szenarien eingesetzt wird. In dem Eingabebild eines monokularen Kamerasystems, welches auf dem Dach eines Automobils in Fahrtrichtung montiert ist, wird pro Zeitschritt ein kleiner Bereich vor der Motorhaube als befahrbar vorausgesetzt. Der Algorithmus berechnet die vorherrschenden Farben innerhalb dieses befahrbaren Bereiches und vergleicht die Farben mit den Farbwerten jedes Pixels im Eingabebild: Je ähnlicher die Pixelfarben zu den vorherrschenden sind, desto höher ist die Wahrscheinlichkeit, dass die entsprechenden Bereiche befahrbar sind. Um den Algorithmus auch im städtischen Umfeld einsetzen zu können, muss ein vorverarbeitendes Modul vorgeschaltet werden, welches Fahrspuren, Schatten und überbelichtete Bereiche ausmaskiert und in der entgültigen Befahrbarkeitskarte als unbekannt (rot) markiert. Weiterhin wird ein dynamisches Suchpolygon vorgestellt, um den Algorithmus unabhängig von weiteren Eingabesensoren zu gestalten.



## 1 Einleitung

Das im folgenden beschriebene Verfahren ist ein Teil der Software, die im autonomen Fahrzeug *Caroline* verwendet wurde. *Caroline* hat im Finale der *DARPA Urban Challenge 2007* teilgenommen. Der vorgestellte Algorithmus ist unablässig in Situationen, in denen die Fahrbahnfläche von anderen Sensoren nicht ausreichend klassifiziert werden kann, z.B. auf Straßen ohne Fahrspurmarkierungen bzw. ohne ausreichend hohe Bordsteinkanten oder auf Feldwegen.

Um ein besseres Verständnis der Aufgabe des vorgestellten Algorithmus zu bekommen, ist ein kurzer Überblick über die im Fahrzeug verwendete Sensorik notwendig. Diese unterteilt sich in aktive Sensoren (drei Laserscanner, zwei Radarsensoren und zwei Lidar-Sensoren), welche große Objekte, wie Mauern, Autos oder Straßensperren wahrnehmen sollen und passive Sensoren (7 Kameras), welche Straßenmarkierungen [5] und die Befahrbarkeit des Untergrundes [4] erkennen sollen. In einer nachgeschalteten Sensorfusion werden die einzelnen Informationsquellen in einer 2D-Karte fusioniert. Zwei Laserscanner werden von einem Bodenerkennungsmodul zur Rekonstruktion der Oberflächengeometrie ausgelesen, um kleine Hindernisse oder unebenes Terrain zu umfahren. Fahrspurmarkierungen und Stoplinien werden von einem Modul zur Fahrspurerkennung mit Hilfe vierer Farbkameras wahrgenommen. Die künstliche Intelligenz wertet daraufhin alle vorliegenden Informationen aus und berechnet eine optimale Trajektorie, welche an die Lenk- und Beschleunigungsaktorik des Fahrzeugs gesendet wird.

## 2 Stand der Technik

Die wesentliche Grundlage für die folgenden Betrachtungen bildet der Algorithmus auf der folgenden Seite, welcher von Thrun et al. [1] vorgestellt und ausführlich in der DARPA Grand Challenge im Oktober 2005 ausgetestet wurde. Die Idee ist, in einem gegebenen Kamerabild einen kleinen Bereich als befahrbar anzunehmen und die darin enthaltenen Farben mit dem gesamten Bild zu vergleichen um ähnliche Farben als befahrbar zu markieren. Allerdings ist der Algorithmus nur für Geländefahrten entwickelt worden um den Anforderungen der Grand Challenge zu genügen. Der im folgenden vorgestellte Algorithmus erhält zusätzlich als Eingabe die Tiefenwerte eines Laserscanners. In einer Höhenkarte werden die Scanlinien normalerweise über die Zeit ausgewertet. Um die Datenmenge, die zwischen Laserscanner und dem Algorithmus ausgetauscht wird, gering zu halten, wird pro Zeitschritt ein Hüllpolygon definiert. Dieses umfasst den Bereich vor dem Auto in der Höhenkarte, welcher als befahrbar betrachtet wird. Das übertragene Polygon wird dann in die Bildkoordinaten der Eingabekamera transformiert und gegebenenfalls geclippt, so dass es nun den befahrbaren Bereich im Bildraum markiert. Alle Farben innerhalb dieses Bereiches werden nun betrachtet und statistisch ausgewertet, um die vorherrschenden Farbtöne zu bestimmen, z.B. straßengrau oder grasgrün. Diese Farbtöne werden nun mit allen Pixeln im aktuellen Kamerabild verglichen, wobei ein im verwendeten Farbraum anwendbares Distanzmaß benutzt wird. Ist die resultierende Distanz eines Pixels kleiner als ein vorgegebener Schwellwert, wird der Pixel und damit der Bereich in der Welt, den er abbildet, als befahrbar angenommen. Zusammengefasst erweitert der Algorithmus den Bereich vor dem Auto, über den eine Aussage zur Befahrbarkeit des Untergrundes getroffen werden kann, von einigen wenigen Metern auf über 50 Meter. In der städtischen Umgebung stößt das beschriebene Verfahren an seine Grenzen und muß weiterentwickelt werden. Beim intensiven Testen des Algorithmus in städtischer Umgebung sind neue Probleme festgestellt

---

**Algorithm 1:** Der grundlegende Algorithmus zur Berechnung befahrbarer Regionen
 

---

**Data:** Das Eingabe-Bild  $I_{kamera}$  und das Laserscanner-Polygon  $P_{Scanner}$ .

**Result:** Ein klassifiziertes Bild  $I_{befahrbar}$ .

```

1 begin
2   Hole das Eingabe-Bild  $I_{kamera}$  und das Laserscanner-Polygon  $P_{Scanner}$ 
3   Kopiere  $I_{kamera}$  nach  $I_{befahrbar}$ 
4   Erstelle eine Liste der aktuellen Farben von den Bildpixeln von  $I_{befahrbar}$ 
   die innerhalb  $P_{Scanner}$  sind
5   Berechne daraus  $n$  vorherrschende Farben und verbessere diese mit
   Cluster-Methoden (z.B. EM-Algorithmus [2], [3])
   /* Füge diese  $n$  Farbverteilungen in eine Liste von  $m > n$ 
   Farbverteilungen */
6   if Die Listengröße kleiner  $m$  ist then
7     | Füge den Wert in die Liste ein
8   else
9     /* Vergleiche jede Farbverteilung  $i$  aus  $n$  mit jedem  $j$  aus  $m$ 
   mit Mahalanobis-Dist.  $d(i, j) = (\mu_i - \mu_j)^\perp * (\Sigma_i + \Sigma_j)^{-1}(\mu_i - \mu_j)$  */
10    foreach Farbverteilung  $j$  do
11      | Speichere die kleinste Farbverteilung  $d(i, j)$  und die Position  $j$ 
12    end
13    if  $d(i, j)$  ist kleiner oder gleich einem Schwellwert  $\phi$  then
14      /* Farbverteilung  $j$  wird angepasst an Farbverteilung  $i$  */
15       $\mu_j \leftarrow \frac{1}{\alpha_i + \alpha_j}(\alpha_i * \mu_i + \alpha_j * \mu_j)$ 
16       $\Sigma_j \leftarrow \frac{1}{\alpha_i + \alpha_j}(\alpha_i * \Sigma_i + \alpha_j * \Sigma_j)$ 
17       $\alpha_j \leftarrow \alpha_i + \alpha_j$ 
18      /* Dabei ist  $\alpha$  die Gewichtung der Verteilung je nach
   Anzahl der zu Grunde liegenden Farben */
19    else
20      | Ersetze die Farbverteilung  $k$  mit der kleinstem  $\alpha_k$  durch  $i$ 
21    end
22  end
23  foreach Gespeicherte Farbverteilung  $j$  do
24    | Multipliziere das Gewicht  $\alpha_j$  mit einem Faktor  $\gamma < 1$ 
25    /* So wird das Gewicht der Verteilung pro Frame geringer */
26  end
27  foreach Pixel des Bildes  $I_{befahrbar}$  do
28    foreach Gespeicherte Farbverteilung  $j$  do
29      Berechne die Verteilung  $p(x, \mu_j, \Sigma_j)$  mit der Verteilungsfunktion
30      if  $p(x, \mu_j, \Sigma_j)$  ist größer als ein Schwellwert  $\chi$  für ein gegebenes  $j$ 
31      then
32        | Klassifiziere den Pixel  $x$  als befahrbar
33      else
34        | Klassifiziere den Pixel  $x$  als nicht befahrbar
35      end
36    end
37  end
38 end

```

---

worden, da die Straßen nun andersgefärbte Fahrbahnmarkierungen besitzen und große Gebäude weite Schatten über die Straße werfen. So kommt es vor, dass sich gelbe Fahrbahnmarkierungen, wie sie z.B. in den USA vorkommen, nicht innerhalb des Bereiches befinden, der vom Laserscanner als befahrbar erkannt wird ( $P_{Scanner}$ ). Infolgedessen sind sie im Kamerabild als nicht befahrbar markiert. Eine durchgezogene Linie in der Mitte der Fahrbahn verhindert z.B. einen Fahrspurwechsel, während Stoplinien als blockierende Objekte vor dem Auto erkannt werden, Abb. 1.



(a) Eingabebild



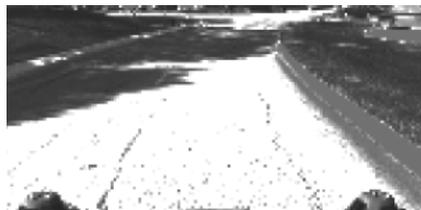
(b) Normale Befahrbarkeitskarte

**Abb. 1.** Die Befahrbarkeitskarte (b) wird vom grundlegenden Algorithmus zur Berechnung befahrbarer Regionen erstellt. Ein weißer Pixel ist befahrbar, ein schwarzer ist unbefahrbar. Eine gelbe Fahrspur (a) allerdings wird als unbefahrbar (b, schwarz) erkannt, weil die Farbe nicht im Suchpolygon auftritt.

Desweiteren erweisen sich großflächige Schatten von höheren Gebäuden als problematisch, während schmale Schatten von Bäumen die Fahrbahnfarbe im Kamerabild nur leicht beeinflussen. Die großflächigen Gebäudeschatten wurden in Gänze als unbefahrbar interpretiert, Abb. 2. Sobald das Auto sich allerdings innerhalb dieser Schattenfläche befindet, passt sich die Belichtungsregelung der Kamera an die verminderte Helligkeit an mit der Folge dass alle Flächen ausserhalb des Schattens überbelichtet erscheinen und ebenfalls als nicht befahrbar erkannt werden, Abb. 3.



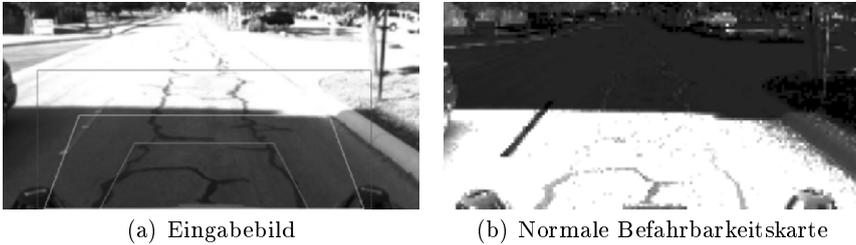
(a) Eingabebild



(b) Normale Befahrbarkeitskarte

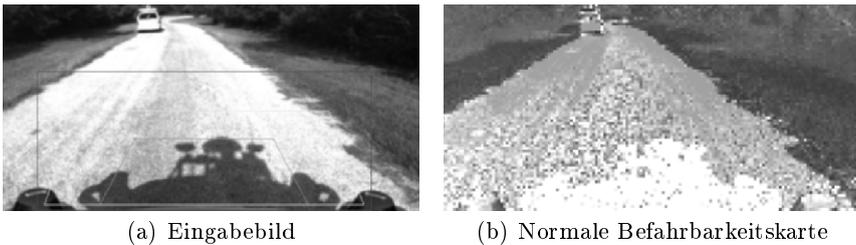
**Abb. 2.** Große dunkle Schatten (a, links) unterscheiden sich zu stark von der Straßenfarbe (b, dunkel).

Bei Fahrten am frühen Vormittag und am Nachmittag erweist sich auch der eigene Schatten des Fahrzeugs als Störfaktor, sobald sich die Sonne hinter dem Auto befindet. In diesem Fall taucht der Eigenschatten im Kamerabild vor dem



**Abb. 3.** Sobald sich die Kamera im Schatten befindet, wird die Belichtung nachgeregelt (a). Unglücklicherweise erscheinen Bereiche ausserhalb des Schattens überbelichtet und werden deshalb als nicht befahrbar erkannt (b, dunkel).

Fahrzeug auf und wird entweder als nicht befahrbar oder als einzig befahrbarer Bereich im gesamten Bild markiert, Abb. 4.



**Abb. 4.** Der Eigenschatten des Fahrzeugs verursacht Probleme (a), zum Beispiel, wenn der Schatten in dem Suchbereich für die befahrbaren Farben liegt (b, weiß).

Schlussendlich ist es problematisch, in Gegenden zu testen, in denen kaum ein Höhenunterschied zwischen der Straße und dem seitlich angrenzendem Gelände, z.B. Grasnarbe oder Sand, festzustellen ist. Die vorgenommenen Änderungen werden ausführlich in Abschnitt 3 beschrieben, es kann aber bereits festgestellt werden, dass anstelle des im Algorithmus verwendeten EM-Verfahrens [3] auch das KMeans-Verfahren angewendet werden sollte. Ebenso können die Abstandsberechnungen in unterschiedlichen Farbräumen, wie RGB, aber auch YUV,  $L^*a^*b$  oder HSV durchgeführt werden, ohne dass der vorgestellte Algorithmus eingeschränkt wird.

### 3 Der Algorithmus für urbane Szenarien

Um die oben vorgestellten Probleme zu bewältigen, wurde ein zusätzliches System entwickelt, welches die Bilder der Eingabekamera vorbearbeitet. Bevor die Eingabebilder an Algorithmus 1 gegeben werden, werden sie an die folgenden Präprozessoren verteilt: Präprozessor für dunkle Bereiche, Präprozessor für überbelichtete Bereiche, Präprozessor für Fahrbahnmarkierungen und Präprozessor für Eigenschatten-Pixel. Jeder Präprozessor liefert nach erfolgter Bearbeitung

eine Bitmaske, welche die Pixel ausmaskiert, die jeweils als kritisch bewertet werden. Die ausmaskierten Pixel werden in der endgültigen Befahrbarkeitskarte nicht wie bisher mit einem Farbwert besetzt, sondern als *unbekannt* gekennzeichnet.

**Der Präprozessor für dunkle Bereiche** maskiert Pixel aus, die zu dunkel und damit potentiell im Schatten sind. Das Eingabebild wird in den HSV-Farbraum konvertiert, der Helligkeitswert des Pixels mit einem gegebenen Schwellwert verglichen. Wenn der Wert kleiner ist, wird der Pixel in der Ausgabemaske auf 1 gesetzt, ansonsten auf 0.

**Der Präprozessor für überbelichtete Bereiche** maskiert überbelichtete Pixel aus. Dazu wird das Eingabebild in den HSV-Farbraum konvertiert. Anschließend wird der Helligkeitswert jedes Pixels mit einem vorgegebenem Schwellwert verglichen. Wenn der Wert größer ist, wird der Pixel in der Ausgabemaske auf 1 gesetzt, ansonsten auf 0.

**Der Präprozessor für Fahrbahnmarkierungen** sucht Pixel, die im RGB-Farbraum nahe an gelben Farbtönen sind, und somit potentiell zu gelben Fahrspuren, wie sie in den USA auftreten, gehören. Wenn der Grün-Wert des Pixels größer als der Rot-Wert und der Blau-Wert ist, so wird er nicht als gelblich betrachtet. Ebenso wenig gelblich ist er, wenn der Rot-Wert größer als die Summe des Blau- und Grün-Wertes ist. In allen anderen Fällen wird das Verhältnis  $\frac{\min(R,G)}{B} - 1$  betrachtet. Ist es größer als ein gegebener Schwellwert, so wird der Pixel in der Ausgabemaske mit 1 gesetzt, ansonsten mit 0. Um lediglich schmale Fahrbahnspuren herauszufiltern und große gelbe Bereiche (z.B. sandige Flächen am Fahrbahnrand) unberührt zu lassen, wird eine Kopie der Ausgabemaske noch geglättet, mit einer Dilatation versehen und schließlich von der Ausgabemaske subtrahiert.

**Der Präprozessor für Eigenschatten-Pixel** maskiert die Pixel aus, die potentiell zu dem eigenen Schatten des Autos gehören. Dazu wird eine feste Punktmenge  $p(x)$  im Bild definiert, welche sich im Bild am Rand der Motorhaube befinden. Pro Frame wird dann per *FloodFill*-Operation ein zusammenhängender Bereich im Bild gesucht, welcher einen geringen Helligkeitswert aufweist. Um zu verhindern, dass der gefundene Bereich beliebig groß ist, wird die Summe gefundener Pixel mit einem Schwellwert verglichen, der die maximale Fläche des Autoschattens darstellt. Die gefundene Fläche wird in der Ausgabemaske markiert.

Wie bereits erwähnt, lässt sich das Eingabepolygon von einem Laserscanner dann gut verwenden, wenn der tatsächlich befahrbare Bereich durch relativ hohe Objekte, wie z.B. Sträucher oder Sandhügel, begrenzt ist. Im urbanen Szenario kann der Höhenunterschied zwischen Straße und Bordstein oder Grasnarbe sehr gering sein. Würde der Schwellwert der Befahrbarkeitserkennung des Laserscan-

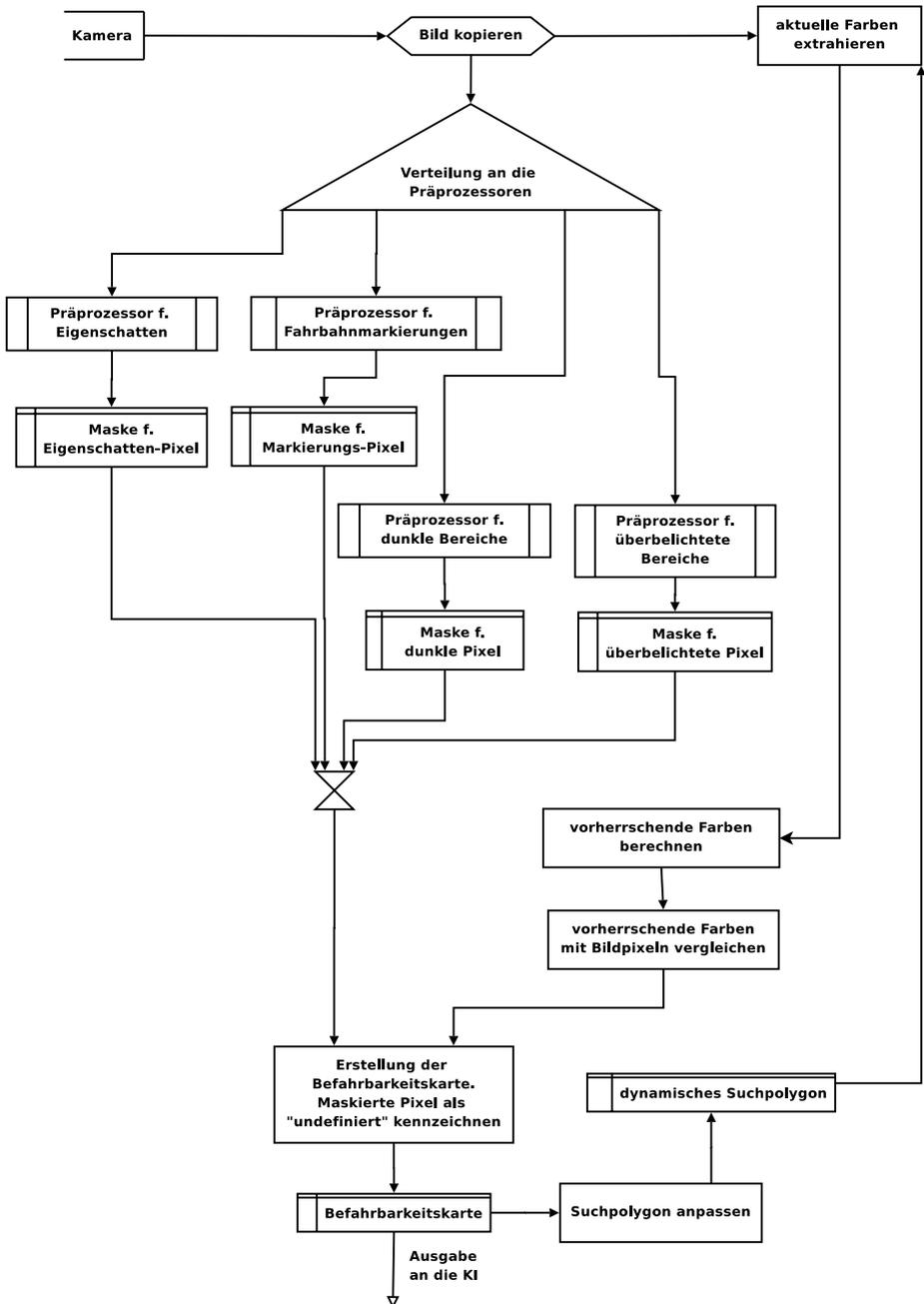


Abb. 5. Die Verarbeitungsschritte des Algorithmus