Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminararbeit

Thorsten Jost

Betrachtung von Verfahren zur Posenbestimmung und Merkmalsexktraktion

Fakultät Technik und Informatik
Department Informatik

Faculty of Engineering and Computer Science
Department of Computer Science

Thorsten Jost

Betrachtung von Verfahren zur Posenbestimmung und Merkmalsexktraktion

Seminararbeit eingereicht im Rahmen der Vorlesung SR
im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Abgegeben am 27. Februar 2009

Thorsten Jost

Thema der Seminararbeit

Betrachtung von Verfahren zur Posenbestimmung und Merkmalsexktraktion

Stichworte

Merkmalsextraktion, Posenbestimmung, Lokalisierung, Kartenerzeugung, Map Building, SIFT-Features, MOPS, SURF, SLAM, Monte Carlo

Kurzzusammenfassung

In dieser Seminararbeit werden unterschiedliche Verfahren zur Merkmalsextraktion aus Bilddaten untersucht. Außerdem werden Verfahren zur Posenbestimmung verglichen. Im Kern der Betrachtung liegt dabei die mögliche Verwendung der Verfahren im Bereich der Indoor-Navigation.

Inhaltsverzeichnis

1 Einleitung

Im Bereich der Technischen Informatik an der HAW Hamburg wird im Kontext der FAUST-Projektes Technologie für Fahrerassistenz- und Autonome Systeme entwickelt. Schwerpunkte in diesen Projekten bilden die Sensorik, Telemetrie und digitale Bildverarbeitung, Echtzeitsysteme und Bussysteme, Software- und Hardwarearchitekturen und Algorithmen und Steuerung. Die FAUST-Projekte dienen der Durchführung von Entwicklungs- und Forschungsprojekten im Bereich von verteilten Hard Real-Time Systemen.
Es wurden bereits automatische Brems-, Ausweich- und Einparkassistenten mit Laserscanner-basierten Objekterkennungsmodellen für das SCV (Sensor Controlled Vehicle) entwickelt und auf verteilten PC-104 Rechnerkomponenten impelemtiert. Eine weitere notwendige Funktion für ein autonomes Fahrzeug ist ein Navigationsmodul für virtuelle Fahrspuren und zum autonomen Abfahren einer vorgegebenen Strecke. Außerdem fänden Funktionen zur autonomen Positionsbestimmung in unbekannten Gebieten großes Interesse. Zur Realisierung eines Navigationssystemes ist zunächst eine Posenbestimmung[1] notwendig, weil Informationen über die Position und Lage des Objektes für Entscheidungen bezüglich der Wegfindung vorausgesetzt werden. Des Weiteren soll dieses Modul zur Verwendung auf nahezu allen mobilen Plattformen geeignet sein.

Aus einer Vorlesung des Sommersemesters 2007 (AW1 TI) entstand die Idee die digitale Bildverarbeitung als Grundlage für die Umsetzung dieses Moduls zu nutzen. In Manske und Jost (2007) wurde bereits ein Verfahren zur Posenbestimmung vorgestellt. Dieses ist in der Lage mittels im Raum verteilter Passmarken, die Pose einer Kamera zu berechnen. Das 13 Parameter 3D-Kameramodell aus der Vermessungstechnik wurde für die Entwicklung eines prototypischen Posenbestimmung-Moduls im Rahmen des Projektes eingesetzt. Ein wesentlicher Bestandteil des Verfahrens war die Verwendung von vorher bekannten Passmarken im Raum. Das heißt, die Orientierungspunkte waren dem System vorher bekannt. Sie müssen dem System also in Form einer Konfiguration bekannt gemacht werden.
Ein anderer Ansatz soll es aber ermöglichen, auf das manuelle Bekanntmachen von Raumpunkten verzichten zu können. Darüber hinaus soll es auch möglich sein die Navigation anhand von markanten Raumpunkten durchzuführen, die schon vorhanden sind. Es sollen also keine Passmarken mehr nötig sein. Dies kann mit dem „Scale Invariant Feature Transform“ erreicht werden. Denn dieser Algorithmus ist in der Lage eindeutige Bildmerkmale aus Abbildungen zu extrahieren. Diese können dann für die Navigation verwendet werden.

In Jost (2008) wurde dieses Verfahren vorgestellt. In dieser Arbeit sollen nun weitere Verfahren zur Merkmalsextraktion und Posen- bzw. Pfaderkennung vorgestellt und verglichen werden. Die Arbeit gliedert sich fünf Kapitel. Das vorliegende Kapitel 1 gibt einen kurzen Überblick über die Motivation, Zielsetzung und den Aufbau der Arbeit. In Kapitel 2 werden Verfahren zur Merkmalsextraktion vorgestellt. Im anschließenden Kapitel 3 werden zwei Verfahren vorgestellt, die speziell die Lokalisierung mit Kartenmaterial umsetzen. Kapitel 4 beendet diese Arbeit und zieht ein Fazit.

[1] Pose - kombinierte Angabe der Position und der Orientierung im dreidimensionalen Raum

2 Featrue Detection

Im Bereich der Bild-Verarbeitung (Computer-Vision) versteht man unter Feature Detection das Suchen und Finden von markanten Bildpunkten. Dabei liefern Algorithmen Kenngrößen die es erlauben, diese Punkte zu identifizieren. Somit wird es ermöglicht, gleiche Punkte in mehreren Abbildungen zu identifizieren.
Sind diese Punkte möglichst einmalig, so können diese Rückschlüsse auf das abgebildete Objekt liefern; wenn sich beispielsweise eine bestimmte Anzahl Punkte in einer räumlichen Korrelation befinden, die auf ein Referenzbild „passen“. Das heisst, wenn die räumliche Korrelation und die Kenngrößen der Punkte mit einer Abbildung eines Referenzobjektes übereinstimmen, so kann daraus geschlossen werden, dass das Abbild dieses Objekt zeigt.
Hier sollen nun drei Verfahren vorgestellt werden.

2.1 Multi-Image Matching using Multi-Scale Oriented Patches

SIFT - Scale Invariant Feature Transform - wurde im Jahre 1999 von David G. Lowe an der University of British Columbia veröffentlicht (Lowe (1999)).

2.1.1 Schlüssel lokalisieren

Zunächst soll festgehalten werden, welche Anforderungen an die gefunden „Key-Points“ gestellt werden. Sie sollen auf der einen Seite möglichst eindeutig sein, damit sie später wieder identifiziert werden können. Auf der anderen Seite sollen sie aber auch so flexibel sein, dass sie invariant sind gegenüber Rotation, Skalierung, affiner und drei-dimensionaler Transformation. Außerdem sollen Bildrauschen und Änderungen der Beleuchtung das Ergebnis möglichst nicht beeinträchtigen.
In Lowe (1999) wird darauf hingewiesen, das unter bestimmten Annahmen, nur der Gauß-Kernel eine Glättung erzeugt, welche eine Invarianz gegenüber Skalierung erhält. Diese Glättung ist zur Bestimmung der „Key-Points“ erforderlich. Um die Key-Points zu erzeugen wird ausgehend von einer Abbildung zunächst eine so genannte Gauß-Pyramide erzeugt. Dabei wird die Abbildung mittels Gauß-Filter geglättet. Das geglättete Abbild wird abermals geglättet. Anschließend wird das Differenzbild der beiden geglätteten Abbildungen berechnet. Anschließend wird das Bild neu skaliert. Es wird auf die Hälfte der Bildinformationen gebracht und der Vorgang wird von vorne begonnen. Abhängig von der Anwendung oder der gewünschten Informationsdichte gestaltet sich dann die Größe der Gauß-Pyramide. Abbildung 1 veranschaulicht dieses Vorgehen. Innerhalb der Gauß-Differenz-Bilder werden nun minimale und maximale Pixelwerte gesucht. Betrachten werden nun die acht benachbarten Pixel der selben Abbildung und 18 weitere Nachbarn, jeweils neun auf einer höheren und niedrigeren Gauß-Ebene. Insgesamt ergeben sich daraus 26 Nachbarn, die das Pixel in einem 3*3-Würfel umschließen.
Wurde das Pixel $L(x, y)$ als Maxima/Minima identifiziert, werden sein Betrag $m(x, y)$ und seine Richtung $\theta(x, y)$ berechnet:

$$m(x,y) = \sqrt{(L(x+1,y) - L(x-1,y))^2 + (L(x,y+1) - L(x,y-1))^2} \tag{1}$$

$$\theta(x,y) = \tan^{-1} \frac{L(x,y+1) - L(x,y-1)}{L(x+1,y) - L(x-1,y)} \tag{2}$$

Zusätzlich werden die normierten Grauwerte der Pixel innerhalb eines bestimmten Radius in einem in 36 Klassen unterteiltem Histogramm akkumuliert. Die Werte sind nach ihrem Abstand

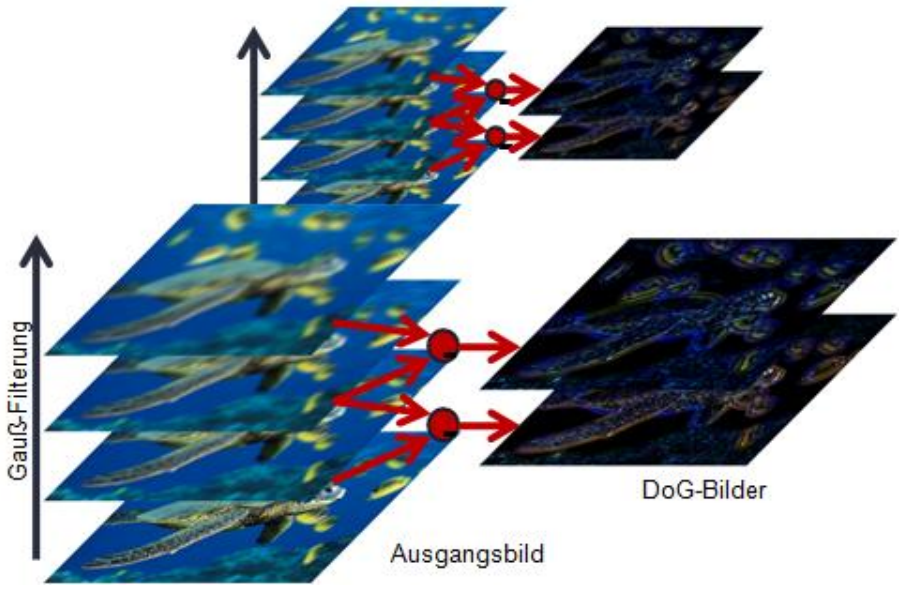


Abb. 1: Gauß-Filter und Gauß-Pyramide

zum Mittelpunkt und ihrem Betrag gewichtet. Nun stelle man sich dieses Histogramm als Kreis vor, in dem die 36 Klassen jeweils 10° des Vollkreises repräsentieren. Man drehe diesen Kreis mit dem höchsten berechneten Wert nach oben. Dadurch wird dieser Merkmalsvektor invariant gegen Rotation.
Die Skalierungsinvarianz wird dadurch erreicht, dass innerhalb der Gauß-Differenz-Bilder die Pixel verglichen werden, um das Maxima/Minima in der Glättungs-Dimension zu finden.

2.1.2 Lokale Abbildungs-Beschreibung

Um nun einen Merkmalsvektor zu erzeugen, werden für die Nachbarn des Key-Points in allen Gauß-Differenz-Bildern die räumliche Orientierung berechnet. Diese ergibt sich in Relation zum Key-Point, indem man dessen Orientierung subtrahiert. Die in Lowe (2004) vorgestellten Experimente wurden mit zwei Skalierungsebenen und jeweils acht Gauß-Glättungsebenen durchgeführt. Dabei würden sich $8 * 4 * 4 + 8 * 2 * 2 = 160$ Elemente ergeben, die in den Merkmalsvektor des Key-Points eingetragen werden können. Genaue Angaben zu den Experimenten und statistische Auswertungen finden sich an der oben zitierten Stelle.
Dieses Verfahren wurde an Anlehnung an das Sehvermögen von Säugetieren entwickelt. Dort werden Kanten anhand ihrer Form und räumlichen Ausbreitung erkannt. Die genau Position auf der Netzhaut ist dabei unerheblich. Kleine Änderungen bewirken dort keine Beeinträchtigung. Verwendet man ein solches Modell, ist ebenfalls das Problem der affinen bzw. räumlichen Transformation gelöst. Denn diese werden durch dieses Vorgehen ebenfalls gut kompensiert.

2.1.3 Indizierung von Objekten

Gefundene Key-Points einer Abbildung können nun in einer Datenbank gespeichert werden. Um ein effizientes Suchen der Merkmalsverktoren zu gewährleisten, wurde eine Abwandlung des K-D-Baumes entwickelt. Denn die Größe des Vektors von 160 Elementen stellt ein Problem für das finden des besten Referenzvektors dar. Dieses „Best-Bin-First“ genannte Verfahren ermöglicht das Finden von ähnlichen Vektoren mit hoher Wahrscheinlichkeit bei gleichzeitig minimierter Rechenzeit.

Werden nun in einer Vergleichsabbildung ausreichend passende Vektoren gefunden, müssen diese noch durch eine Model-Hypothese bestätigt werden. Zu diesem Zweck wird die Hough-Transformation angewendet. Es wird eine Hash-Tabelle angelegt, in der die Key-Points in Abhängigkeit der Model-Lokation, der Orientierung und der Skalierung eingetragen werden. Insgesamt werden für jeden Key-Point 16 Einträge in der Tabelle erzeugt um die Stabilität zu erhöhen. Anschließend wird die Tabelle in absteigender Reihenfolge der Anzahl gleicher Hash-Schlüssel sortiert. Schlüssel, die seltener als drei Mal auftreten, werden aussortiert. Für die Keys, die im gleichen Hash-Bereich liegen, also einen Cluster bilden, werden nun mittels Kleinster-Quadrate-Methode die Paramter für eine affine Transfomation gesucht. Diese können mit der folgenden Gleichung gefuden werden. Wobei x und y den Koordinaten des Models entsprechen und u und v denen der Abbildung:

$$\begin{bmatrix} x & y & 0 & 0 & 1 & 0 \\ 0 & 0 & x & y & 0 & 1 \\ & & & \dots & & \\ & & & \dots & & \end{bmatrix} * \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \\ t_x \\ t_y \end{bmatrix} = \begin{bmatrix} u \\ v \\ . \\ . \end{bmatrix} \tag{3}$$

$$A * x = b \tag{4}$$

Die Gleichung 4 zeigt die Kurzform der Gleichung 3, wobei x dem gesuchten Vektor der Translations-Parameter entspricht. Nun können die Paramter aller Cluster miteinander verglichen werden. Bleiben die Abweichungen in einem bestimmten Maß, gilt die These als bestätigt. Bleiben weniger als drei Cluster übrig, wird die These verworfen. Mit Hilfe der Transformationsparameter ist es nun auch möglich, die gefundenen Objekte im Vergleichsbild zu markieren. Abbildung 2 verdeutlich dies. Dort wurden die Objekte Eisenbahn und Frosch von einer SIFT-Anwendung mit farbigen Rechtecken umrandet. Der Frosch wurde zwei Mal gefunden. Wie man auf der Abbildung ebenfalls erkennen kann, ist es auch Möglich, verdeckte Objekte zu erkennen. Dies liegt an der Tatsache, dass schon drei Key-Points ausreichend sein können, um ein Objekt zu erkennen. Aber auch die verdeckten Objekte ließen die Identifikation von mehr als zehn Key-Points zu.

Es ist also möglich, mit Hilfe der SIFT-Features abgebildete Objekte mit hoher Wahrschein-

Abb. 2: Beispiel einer SIFT Anwendung - Links sind zwei Referenz-Abbildungen zu sehen. In der der Mitte sieht man das Vergleichsbild, in dem die Objekte gefunden werden sollen. Auf der rechten Abbildung wurden die gefundenen Objekte markiert. Die kleinen Rechtecke verweisen auf die Key-Points, die zur Identifikation dienten.

lichkeit wieder zu erkennen bzw. zu identifizieren. Dabei sind diese Features robust gegenüber

Störungen wie Verdeckung, Bild-Rauschen, affine und dreidimensionale Transformationen oder auch Änderungen der Beleuchtungsintensität.
Durch eine Weiterentwicklung des Abbildungsmodells, bei dem Objekte von mehreren Seiten aufgezeichnet würden, könnte auch das Problem der dreidimensionalen Transformation gelöst werden. Die SIFT-Features der Abbildungen könnten so miteinander verknüpft werden, so dass ein räumliches Modell des Objektes entstünde. Dieses würde es dann unerheblich machen, in welecher Perspektive das Bild aufgezeichnet wurde, in dem das Objekt identifiziert werden soll.

2.2 Multi-Image Matching using Multi-Scale Oriented Patches

Die Arbeit von Matthew Brown u.a. (Brown u. a. (2004)) stellt ein weiteres Verfahren zur Merkmalsextraktion dar. Multi-Image Matching using Multi-Scale Oriented Patches oder kurz MOPS wurde von Mircosoft Research veröffentlicht. Dieser Ansatz beinhaltet Verbesserungen die im wesentlichen auf der Arbeit von Lowe Lowe (1999), Lowe (2004) beruhen.

2.2.1 Multi-Scale Oriented Patches

Die Komplexität der tatsächlichen Bedingungen (Geometrische und Photometrische Eigenschaften der Szenen und Kameras) wurde für dies Arbeit reduziert. Sechs Parameter sind für die Transformation zwischen zusammengehörigen Image-Patches ausreichend. Dies sind t_1, t_2, θ, s, α und β (Position, Orientierung, Skalierung, Gain und Bias).

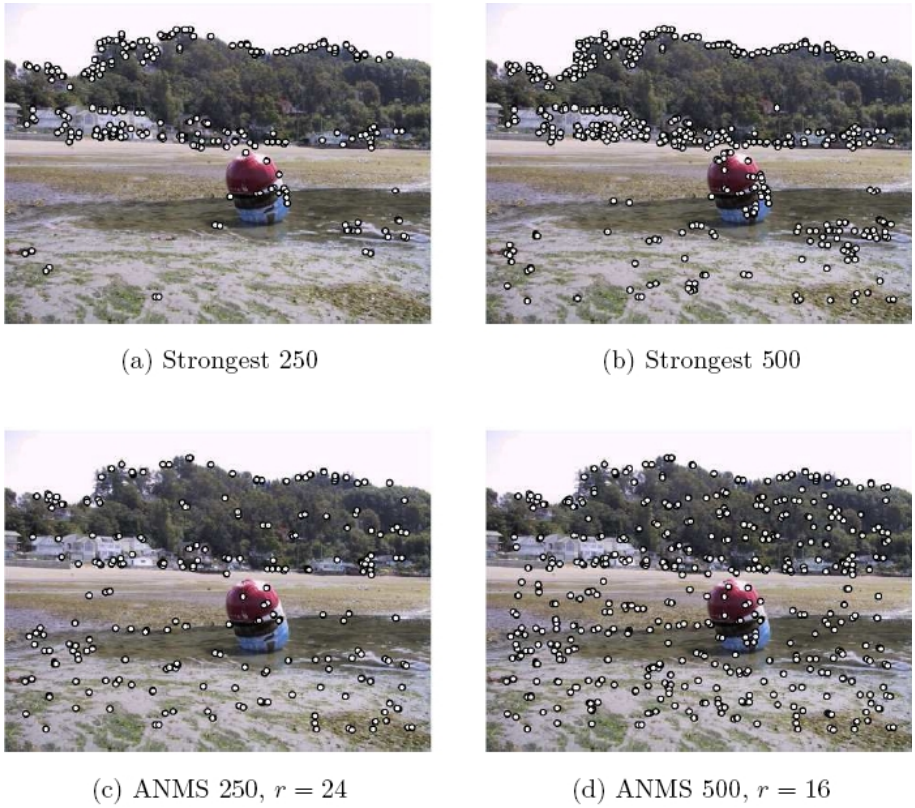
$$I'(x') = \alpha I(x) + \beta + n(x) \tag{5}$$

mit

$$x' = Ax + t \tag{6}$$

$$A = s \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{7}$$

Der Wert $n(x)$ beschreibt den Fehler, der durch Bildrauschen und Modell-Fehler verursacht wird. Für jedes Feature wird dann ein Satz von Modell-Parametern festgelegt, wodurch dann Transformations-Parameter zwischen Feature-Paaren impliziert sind. Über den Fehlerwert $n(x)$ kann dann eine Aussage getroffen werden, ob eine Übereinstimmung korrekt ist oder nicht.
Wie auch bei SIFT wird bei MOPS mit einer Grauwert-Bildpyramide gearbeitet. Höhere Ebenen werden durch Gauß-Filterung geglättet und die Auflösung wird reduziert. Intrest-Points werden auf jeder Ebene der Pyramide extrahiert. Voraussetzung dieser Arbeit ist, dass die Bilder eine gleiche Skalierung haben. Aus diesem Grund wurde auf die Verwendung von Sub-Oktave-Ebenen verzichtet. Zur Erkennung der Intrest-Points wird der Algorithmus „Harris Corner Detector“ verwendet. Siehe dazu Brown u. a. (2004) S.3ff. Um eine möglichst optimale Überdeckung von Bidern zu erreichen wurde hier ein Verfahren zur besseren Verteilung der Intrest-Points über ein Bild implementiert. Dieses Verfahren wurde „Adaptive Non-maximal Suppression“ genannt. Dabei werden Intrest-Points verworfen, die innerhalb eines Radius kein Maximum sind. Der Radius ist dabei adaptiv pro Intrest-Point, sodass eine möglichst gleichbleibende räumliche Verteilung erreicht wird. Abbildung 3 zeigt die Verteilung im Vergleich zu einer Verteilung, die auf einem globalen Grenzwert beruht. Gut zu erkennen ist hierbei, dass sich bei Verwendung eines globalen Grenzwertes Häufungen von Intrest-Points an markanten Stellen des Bildes entwickeln. Die erreichte Verteilung ist dabei jedoch ein Kompromiss, da „starkte“ Intrest-Points auf Kosten von „schwächeren“ verworfen werden. Die räumliche Ver-



(a) Strongest 250 (b) Strongest 500

(c) ANMS 250, $r = 24$ (d) ANMS 500, $r = 16$

Abb. 3: Verteilung durch Ädaptive Non-maximal Suppression

teilung ist aber ausgewogener.
Alle Intrest-Points besitzen auch eine Orientierung θ , die anhand des lokalen Gradienten berechnet werden kann. Ein weicher Verlauf über das Gradientenfeld $u_l(x, y)$ führt zu einer stabileren Orientierungs-Bestimmung. Dafür wird das Bild zuvor geglättet.

2.2.2 Feature Descriptor

Um nun Intrest-Points in mehreren Bildern miteinander vergleichen zu können, benötigen diese nun auch noch eine Beschreibung. Zu diesem Zweck werden dem Intrest-Point umgebende Bildinformationen herangezogen: Für einen IP (x, y, l, θ) wird um die Position (x, y) im Winkel θ von der Pyramiden-Ebene $l + l_s$ ein $8x8$-großer Patch entnommen. Zwischen den Samples liegt ein Abstand von $s = 5$ Pixeln (siehe Abbidlung 4). Der Wert für l_s berechnet sich aus dem Pixelabstand s. Für $s = 5$ ergibt sich ein Wert von 2. Das bedeutet, dass der Patch aus einer zwei Ebenen höher gelegenen Pyramidenebene entnommen wird. Dadurch lassen sich Aliasing-Effekte vermeiden. Die Pixelwerte werden dann durch bilineare Interpolation berechnet. Nach einer Normalisierung dieses Vektors sind die Informationen invariant gegen Änderungen der Intensität (Bias und Gain).

2.2.3 Feature Matching

Von einer Sammlung von n Bildern wurden die Multi-Scale Oriented Patches extrahiert. Als nächster Schritt werden nun Übereinstimmungen von Bildern anhand der Patches bestimmt. Des geschieht in drei Schritten:

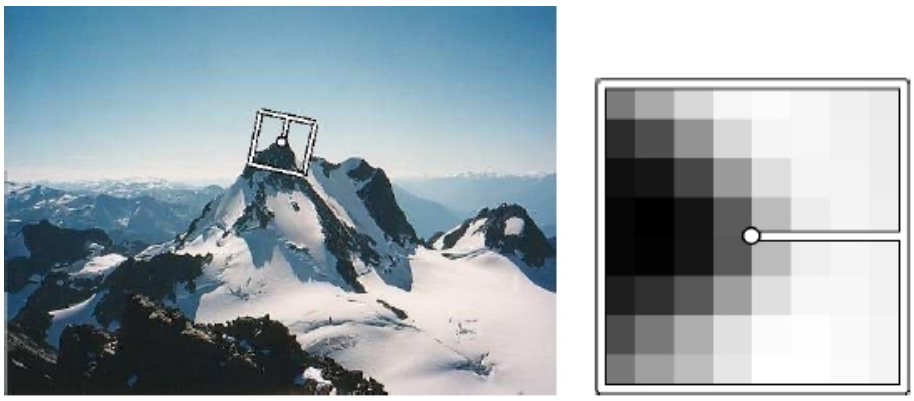


Abb. 4: Feature Descriptor

- Suchen von Kandidat-Matches mit Hilfe eines angenäherten Neares-Neighbour-Algorithmus;
  Dabei wird ein Wavelet-Verfahren eingesetzt. Wavelets sind $2x2$ Pixel große Suchmuster. Anhand dieser Suchmuster werden die Features in Kategorien eingeteilt. Dies entspricht einer Vorsortierung.

- Ausfiltern von Matches durch eine Outlier-Rejection-Procedure;
  Ausreißer wreden verworfen, sobald deren Fehler $n(x)$ (s. Gleichung 5) die Outlier-Distance übersteigen. Die Outlier-Distance wird folgendermaßen berechnet: Zu einem Patch im Quellbild gibt es einige Kandidat-Matches. Beim Kandidat-Match mit dem kleinsten $n(x)$ wird von einem passenden Feature ausgegangen. Der Match mit dem zweitkleinsten $n(x)$ ist dagegen kein Treffer. Für das gesamte Bild wird dann der Durchschnitt der zweitkleinsten $n(x)$ berechnet. Dies ist dann die Grenze für gültige Übereinstimmungen.

- Bestimmen von geometrischen Bedingungen durch RANSAC.
  Dadurch werden weitere falsche Treffer herausgefiltert, indem man die räumliche Verteilung der Features überprüft. Je nachdem, wie die Bilder aufgezeichnet wurden, können unterschiedliche geometrische Modelle verwendet werden, um die Lage der Features untereinander zu vergleichen. Beispielsweise könnte eine statische Szene mit einer beweglichen Kamera aufgezeichnet worden sein. Hier wäre die Verwendung einer Fundamental-Matrix angebracht.

2.3 Speeded Up Robust Features

Die Arbeit von Bay u. a. (2008) stellt einen Ansatz mit dem Namen „Speeded Up Robust Features“ vor, der eine möglichst hohe Verarbeitungsgeschwindigkeit bieten soll. Hierzu wurden in der Literatur bekannte Verfahren bearbeitet und so modifiziert, sodass das angestrebte Ziel erreicht werden kann.

2.3.1 Fast Hessian Detector

Anstelle eines Gauß-Filters mit weichem Verlauf wird ein stark vereinfachter Filter verwendet. Abbildung 5 zeigt dies als eine Annäherung an die Gaußglocke 2. Ordnung mit $\sigma = 1.2$. Der vereinfachte Filter wurde Box-Filter genannt. Die zuvor beschriebenen Verfahren verwenden

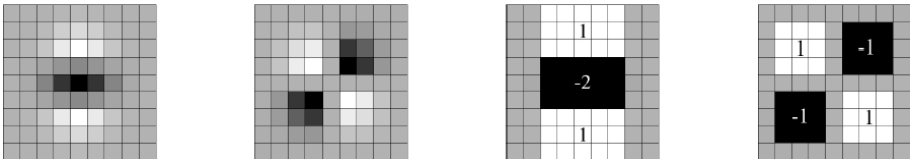


Abb. 5: Partielle Ableitung der Gaußglocke 2. Ordnung, x- und xy- Richtung und die angenäherten Box-Filter

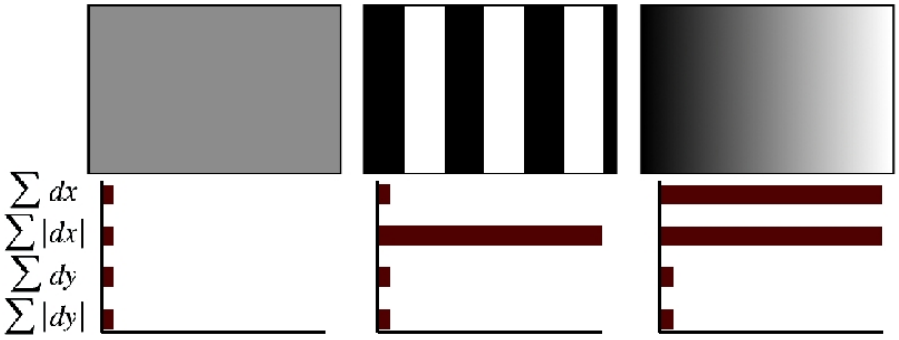


Abb. 6: Beispile der Haar-Wavelet-Antworten für einfache Muster

Bild-Pyramiden für die räumliche Skalierung. SURF dagegen verwendet immer das gleiche Ausgangsbild und ändert die Größe des Box-Filters. Die berechneten Werte fließen dann in die Erstellung der Hesse-Matrix ein:

$$H_{approx}(x,\sigma) = \begin{bmatrix} D_{xx}(x,\sigma) & D_{xy}(x,\sigma) \\ D_{xy}(x,\sigma) & D_{yy}(x,\sigma) \end{bmatrix} \tag{8}$$

Wobei x die Position des Bildpunktes und σ den Skalierungsgrad angibt. D_{xx}, D_{xy} und D_{yy} stellen die oben erwähnte Annäherung dar. Intrest-Points werden dann durch Maxima der Determitante $det(H_{approx})$ in einer $3x3x3$-Nachbarschaft (Bildposition und Skalierungsgrad) definiert.

2.3.2 Descriptor-Komponenten

Zur Beschreibung der Intrestpoints wird im ersten Schritt dessen Orientierung durch ein Haar-Wavelet Filterung bestimmt. Dies sind einfache Faltungsmasken, horizontal oder vertikal in schwarz und weiß unterteilt sind. Diese Masken werden in x- und y-Richtung um den Intrest-Point angewendet (mit einem Radius $6s$, wobei s Skalierungsgrad). Die Ergebnisse werden in einem Ergebnisvektor aufaddiert, der die Orientierung um den Ursprung angibt. Der höchste Wert ergibt die Orientierung. Der eigentliche Descriptor der den Intrest-Point beschreibt besteht aus vier Vektoren, die jeweils eine Sub-Region um den Intrest-Point bewerten. Jeder Vektor enthält jeweils die Haar-Wavelet-Antworten in x- und y-Richtung, als Summe der Werte und als Summe der Beträge siehe (Abbildung 6):

$$v = \left(\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|\right) \tag{9}$$

Um räumliche Eigenschaften zu erhalten, werden die Haar-Wavelet-Antworten in Abhängigkeit zur Entfernung zum Intrest-Point gewichtet.

3 Map Building

Die nächsten beiden Kapitel beschäftigen sich nun mit Verfahren, deren Ziel die Lokalisierung mit Hilfe von Landkarten ist. Dabei werden auch Methoden der Feature-Detection verwendet. Landkarten erlauben es den autonomen mobilen Systemen sich in ihrer Umgebung zurecht zu finden. So können sie beispielsweise Ziele besuchen oder festgelegte Routen verfolgen. Kennt man die Umgebung, in der sich das System befindet, kann man dem System fertige Karten zur Verfügung stellen, anhand deren es navigiert. Kennt man die Umgebung dagegen nicht, muss das System selbst das Kartenmaterial erzeugen.

3.1 Simultaneous Localization and Map-Building Using Active Vision

Ein Verfahren zur Simultanen Lokalisierung und Karten-Erzeugung beschreiben Davison und Murray (2002) in ihrer Arbeit. Ein Schlüsselelement des Verfahrens ist die Verwendung eines beweglichen Stereo-Kamerasystems. Dieses System ist in der Lage, Raumpunkte bei eigener Bewegung zu fixieren. Gleichzeitig kann die Position der Raumpunkte relativ zum System durch das Stereo-System ermittelt werden. Angebracht ist die Kamera auf einem Roboter, der auf Rädern durch die Umgebung fahren kann.

3.1.1 Globaler Zustandsvektor

Die Position des Roborters und die Parameter des Kartenmaterials werden in einem Vektor $\hat{x}$ zusammengefasst. Dazu gibt es noch eine Kovarianzmatrix P, welche die Element von $\hat{x}$ in Beziehung zueinander setzt:

$$\hat{x} = \begin{pmatrix} \hat{x}_v \\ \hat{y}_1 \\ \hat{y}_2 \\ \vdots \end{pmatrix}, P = \begin{pmatrix} P_{xx} & P_{xy1} & P_{xy2} \dots \\ P_{y1x} & P_{y1y1} & P_{y1y2} \dots \\ P_{y2x} & P_{y2y1} & P_{y2y2} \dots \\ \vdots & \vdots & \vdots \end{pmatrix} \tag{10}$$

Die Position $\hat{z}, \hat{x}$ und die Orientierung $\hat{\phi}$ des Roboters ist in $\hat{x}_v = (\hat{z}, \hat{x}, \hat{\phi})$ zusammengefasst. Das Kartenmaterial wird in Form von Feature-Positionen abgespeichert, wobei $\hat{y}_i = (\hat{X}_i, \hat{Y}_i, \hat{Z}_i)^T$. All diese Informationen werden verwendet, um ein Modell des Raumes mit dem darin enthaltenen Roboter zu erzeugen. Die Kovarianzmatrix wird zur Gewichtung der Features genutzt.

3.1.2 Karten-Erzeugung

Nach einem festgelegtem Zeitintervall Δt_k werden die Werte für $\hat{x}$ und P neu berechnet. Durch weiter Berechnungen ergeben sich daraus Wahrscheinlichkeiten, die auf die Bildposition eines Features im nächsten Bild schließen lassen. Diese Wahrscheinlichkteiten werden verwendet, um die bewegliche Kamera in die entsprechende Richtung schwenken zu lassen. Außerdem werden um die Features Ellipsen angelegt, in deren Bereich die entsprechenden Features gesucht werden. Je nach Wahrscheinlichkeit wird die Größe der Ellipse festgelegt. In Abbildung 7 ist ein Versuchsaufbau dargestellt. Teil (a) zeigt den Roboter und Passmarkten, die als Features erkannt werden sollten. Teil (b) zeigt die gefundenen Features und deren geschätzte Positon (grau). Im Teil (c) und (d) fuhr der Roboter die selbe Strecke wieder zurück. Im Teil (d) jedoch,

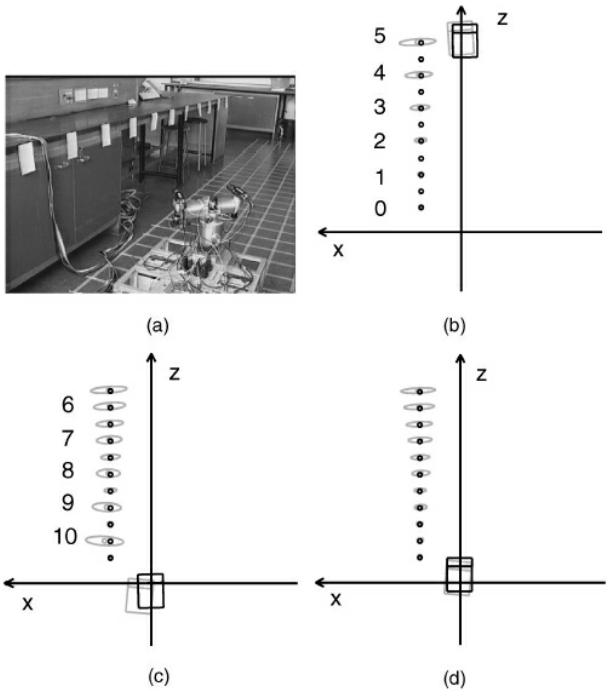


Abb. 7: SLAM Versuchsaufbau

wurde das Feature 1 wieder erkannt, im Teil (c) mussten alle Features neu entdeckt werden. Durch die Verknüpfung der bekannten Informationen konnte die Genauigkeit der Berechnungen erhöht werden. Dies ist durch die bessere Überdeckung der grauen und schwarzen Siluette des Roboters und der kleineren Ellipsen erkennbar. Können also die gleichen Features immer wieder erkannt werden, so trägt dies zu einer erhöhten Genauigkeit der Positionsangabe bei. Werden die Features jedoch nicht erkannt, obwohl sie vom System erwartet wurden, werden sie vom globalen Zustandsvektor entfernt. So ergibt sich automatisch eine Karte der Umgebung, deren Fehler nicht von der gesamt gefahrenen Strecke abhängig ist.

3.2 Monte Carlo

Anders als das zuvor dargestellte Verfahren arbeitet das System von Wolfy u. a. (2002) in einer bekannten Umgebung. Bekannte Bilder aus einer Datenbank werden laufend mit einer aktuellen Abbildung einer Roboter-Kamera verglichen und so die Pose bestimmt.

3.2.1 Bildeigenschaften und Feature Detection

Um ganze Bilder miteinander vergleichen zu können, wird für jedes Bild eine globale Histogramm-Matrix gebildet. Diese Matrix beinhaltet mehrere Histogramme von verschiedenen lokalen Features und verschiedenen Feature-Funktionen. Diese Feature-Funktionen errechnen eine Kennzahl, die es erlaubt, die Ähnlichkeit von Abbildungen festzustellen.
Diese Eigenschaft wird dazu genutzt, das aktuelle Bild mit den Bildern aus der Datenbank zu vergleichen. Dabei erhält man die Bilder mit den höchsten Übereinstimmungen. Beispielhaft zeigt die Abbildung 8 ein Aufnahme des Roboters. Die Abbildung 9 zeigt die neun am Besten übereinstimmenden Bilder aus der Datenbank. Die Datenbank enthält in diesem Fall 936 Bilder.

Abb. 8: Suchanfrage des Roboters

Abb. 9: Antwort des Algorithmus, Übereinstimmung: 81.67%, 80.18%, 77.49%, 77.44%, 77.43%, 77.19%, 77.13%, 77.06%, und 76.42%.

3.2.2 Monte-Carlo Lokalisierung

Um die Monte-Carlo Lokalisierung anwenden zu können, müssen die „Samples“ (Mögliche Posen des Roboters) anhand der zuvor bestimmten Ähnlichkeiten der Bilder gewichtet werden. Eine „Visibility Area“ σ_M für jedes Bild M bestimmt, ob sich ein „Sample“ für die Berechnung eignet. Liegt es außerhalb von σ_M wir es für die Gewichtung nicht verwendet. Ein „Sample“ wird also dann hoch gewichtet (ω ist groß), wenn es sich innerhalb von σ_M befindet und M eine hohe Ähnlichkeit zu dem Anfragebild hat:

$$\omega = \sum_{i=1}^{n} I\left(\langle x, y\rangle, \sigma_i\right) * d(\psi) * \zeta_i \tag{11}$$

Wobei x, y die Koordinaten des Sample angeben. $d(\psi)$ ist die Gewichtung der Abweichung der Orientierung des Samples. ζ_i ist die Ähnlichkeit von M_i zum Abfragebild und I ist eine Funktion die 1 ergibt, wenn $\langle x, y\rangle$ innerhalb von σ_i liegt, ansonsten 0.
Anhand der Gewichtungen wird dann die wahrscheinlichste Pose l des Systems bestimmt. Die Samples werden dann gemäß ihrer Gewichte und der im Iterationsschritt vollzogenen Bewegung neu berechnet. So vereinigen sich dann die zuvor zufällig verteilten Samples nach einiger Zeit um die tatsächliche Position des Roboters. Die Abbildung 10 zeigt dies.

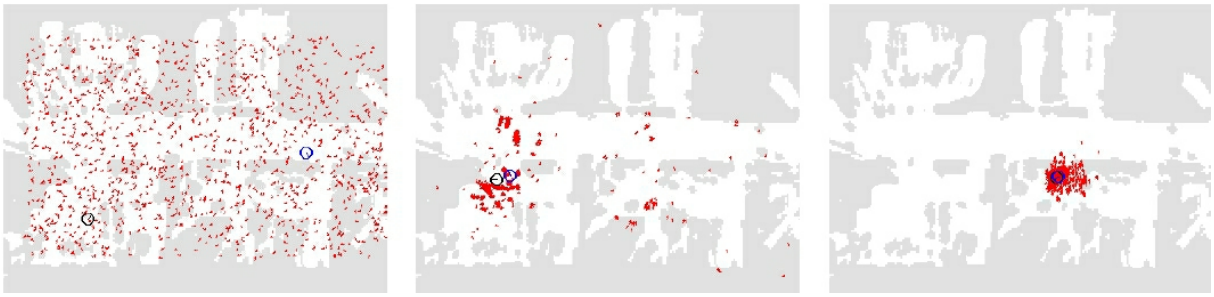


Abb. 10: Anordnung der Samples bei Initialisierung, nach vier und nach 35 Iterationen

4 Fazit

Insgesamt wurden drei Verfahren zur Merkmalsextraktion und zwei weitere Verfahren zur Lokalisierung mit Hilfe von Merkmalsextraktion vorgestellt. Im Hinblick auf die in Kapitel 1 genannten Ziele können die vorgestellten Verfahren eingesetzt werden. Je nach Zielvorgabe können die Verfahren jedoch unterschiedliche Priorisierungen erfahren. Zum Beispiel unterscheiden sich die Verfahren Kapitel 3.1 und Kapitel 3.2 darin, dass SLAM sich in fremder Umgebung eine Karte selbst erzeugt und das Monte-Carlo-Verfahren mit einer zuvor erstellten Bilddatenbank arbeitet. Anhand der verwendeten Hardware-Platform könnte sich des weiteren entscheiden, welches Verfahren zur Merkmalsextraktion genutzt werden kann. Einige sind genauer und benötigen mehr rechnerischen Aufwand, bei anderen ist es entgegengestzt.
Abschließend kann gesagt werden, dass es im Umfeld der Kartenerzeugung und Lokalisierung zahlreiche Ansätze gibt, die je nach Anwendung gewisse Vor- und Nachteile aufweisen. Die Auswahl eines Verfahrens sollte aus diesem Grund sorgfältig getroffen werden.

Abbildungsverzeichnis

Bis auf Abbildung 1 wurden alle Abbildungen aus der jeweils in den Kapiteln benannten Literatur entnommen.

Literatur

[Bay u. a. 2008] BAY, Herbert ; ESS, Andreas ; TUYTELAARS, Tinne ; GOOL, Luc V.: Speeded-Up Robust Features (SURF) / ETH Zurich BIWI and K.U. Leuven ESAT-PSI. 2008. – Forschungsbericht

[Brown u. a. 2004] BROWN, Matthew ; SZELISKI, Richard ; WINDER, Simon: Multi-Image Matching using Multi-Scale Oriented Patches / Microsoft Research, Microsoft Corporation. 2004. – Forschungsbericht

[Davison und Murray 2002] DAVISON, Andrew J. ; MURRAY, David W.: Simultaneous Localization and Map-Building Using Active Vision / IEEE. 2002. – Forschungsbericht

[Jost 2008] JOST, Thorsten: Navigation anhand natürlicher Landmarken mit Hilfe der 'Scale Invariant Feature Transform' / HAW Hamburg, Fakultät Technik und Informatik. 2008. – Forschungsbericht. Eingereicht im Rahmen der Vorlesung AW1

[Lowe 1999] LOWE, David G.: Object Recongition from Locale Scale-Invariant Features / Computer Schience Department, Univerity of British Columbia. 1999. – Forschungsbericht

[Lowe 2004] LOWE, David G.: Distinctive Image Features from Scale-Invariant Keypoints / Computer Schience Department, Univerity of British Columbia. 2004. – Forschungsbericht

[Manske und Jost 2007] MANSKE, Nico ; JOST, Thorsten: Posenbestimmung in Räumen mit einem 3D-Kameramodell / HAW Hamburg, Fakultät Technik und Informatik. 2007. – Forschungsbericht. Eingereicht im Rahmen der Vorlesung AW2

[Meisel 1994] MEISEL, Andreas: *3D-Bildverarbeitung für feste und bewegte Kameras.*, Rheinisch Westfälische Technische Hochschule (RWTH), Dissertation, 1994

[Meisel 2005] MEISEL, Prof. Dr.-Ing. A.: *Vorlesungsunterlagen Robot Vision*. HAW Hamburg. 2005. – URL https://users.informatik.haw-hamburg.de/home/pub/prof/meisel/WP_RV_RobotVision/

[Wolfy u. a. 2002] WOLFY, Jürgen ; BURGARDZ, Wolfram ; BURKHARDTZ, Hans: Robust Vision-based Localization for Mobile Robots Using an Image Retrieval System Based on Invariant Features / Department of Computer Science, University of Hamburg and Department of Computer Science, University of Freiburg. 2002. – Forschungsbericht