



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminararbeit

Thorsten Jost

Navigation anhand natürlicher Landmarken mit
Hilfe der „Scale Invariant Feature Transform“

Thorsten Jost

Navigation anhand natürlicher Landmarken mit
Hilfe der „Scale Invariant Feature Transform“

Seminararbeit eingereicht im Rahmen der Vorlesung AW1
im Studiengang Master Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer : Prof. Dr.-Ing. Andreas Meisel
Zweitgutachter : Prof. Dr.-Ing. Bernd Schwarz

Abgegeben am 23. Juli 2008

Thorsten Jost

Thema der Seminararbeit

Navigation anhand natürlicher Landmarken mit Hilfe der „Scale Invariant Feature Transform“

Stichworte

Merkmalsextraktion, SIFT-Features, Posenbestimmung, Positionsbestimmung, Indoor, Navigation, 3D-Kameramodell

Kurzzusammenfassung

In dieser Seminararbeit werden die Eigenschaften von SIFT-Features erläutert. Dabei handelt es sich um ein Verfahren, um Bildmerkmale in Abbildungen zu identifizieren. Mit deren Hilfe ist es anschließend möglich, die selben Merkmale in anderen Abbildungen zu suchen. Außerdem sollen Möglichkeiten aufgezeigt werden, wie es mit Hilfe von SIFT-Features möglich sein kann, Navigation und Lokalisierung durchzuführen.

Inhaltsverzeichnis

1	Einleitung	2
2	Feature Detection	3
2.1	Schlüssel lokalisieren	3
2.2	Lokale Abbildungs-Beschreibung	4
2.3	Indizierung von Objekten	4
3	Navigation mit SIFT	6
3.1	Kameramodell	6
3.2	Stereo-Abgleich	7
3.3	Kartenerzeugung	8
4	Stitching von Bildern	9
5	Ausblick	10
5.1	Implementierung	10
5.2	Erweiterung der Navigation	10
5.3	Anwendungen	11
5.4	Fazit	11
	Abbildungsverzeichnis	12
	Literatur	13
	Glossar	14

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 Laserscannerbasierten Objekterkennungsmodellen für das SCV (Sensor Controlled Vehicle) entwickelt und auf verteilten PC-104 Rechnerkomponenten implementiert. Eine weitere notwendige Funktion für ein autonomes Fahrzeug ist ein Navigationsmodul für virtuelle Fahrspuren und zum autonomen Abfahren einer vorgegebenen Strecke. Außerdem wären Funktionen zur autonomen Positionsbestimmung in unbekannt Gebieten von großem Interesse. Zur Realisierung eines Navigationssystems ist zunächst eine Posenbestimmung¹ 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.

Dieser Ansatz soll es nun aber ermöglichen, auf das manuelle bekannt machen 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.

Die Arbeit gliedert sich vier Kapitel. Das vorliegende Kapitel 1 gibt einen kurzen Überblick über die Motivation, Zielsetzung und den Aufbau der Arbeit. In Kapitel 2 wird die Funktionsweise von SIFT vorgestellt. Im anschließenden Kapitel 3 wird dann bezug auf die Navigation genommen. In Kapitel 4 folgt die Erläuterung für die Durchführung einer Masterarbeit in diesem Bereich und eine Bewertung des vorgestellten Vorhabens.

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

2 Feature Detection

Grundlegende Idee des Algorithmus ist es, Punkte in Abbildungen zu finden, deren Beschaffenheit es ermöglicht, sie als eindeutig zu identifizieren. Diese Punkte werden „Key-Points“ genannt. Dazu werden benachbarte Punkte herangezogen und auf bestimmte Weise miteinander verglichen. Das Vorgehen des Algorithmus wird nun in diesem Abschnitt erläutert.

2.1 Schlüssel lokalisieren

Zunächst soll festgehalten werden, welche Anforderungen an die gefundenen „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, dass 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 Pi-

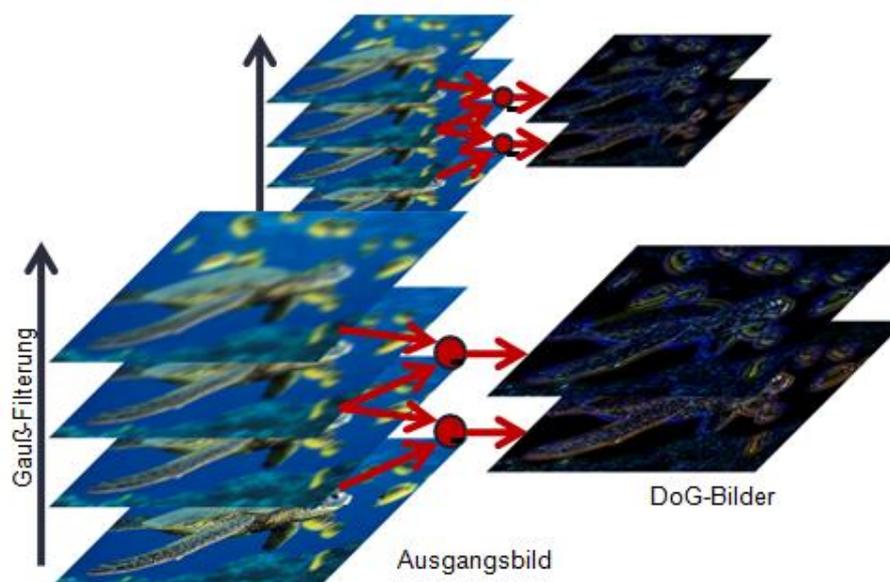


Abb. 1: Gauß-Filter und Gauß-Pyramide

xelwerte 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 sei-

ne Richtung $\theta(x, y)$ berechnet:

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

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

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 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. Dann kann man das Feld, mit dem höchsten berechneten Wert nach oben drehen. 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.

2.2 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.

2.3 Indizierung von Objekten

Gefundene Key-Points einer Abbildung können nun in einer Datenbank gespeichert werden. Um ein effizientes Suchen der Merkmalsvektoren 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 die 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-Quadrat-Methode“ die Parameter für eine affine Transformation gesucht. Diese können mit der folgenden Gleichung gefunden werden. Wobei x und y den Koordinaten des Models

entsprechen und u und v denen der Abbildung:

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

$$A * x = b \quad (4)$$

Die Gleichung 4 zeigt die Kurzform der Gleichung 3, wobei x dem gesuchten Vektor der Translation-Parameter entspricht. Nun können die Parameter 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 verdeutlicht 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 Wahrscheinlichkeit



Abb. 2: Beispiel einer SIFT Anwendung - Links sind zwei Referenz-Abbildungen zu sehen. In 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 dienen.

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 welcher Perspektive das Bild aufgezeichnet wurde, in dem das Objekt identifiziert werden soll.

3 Navigation mit SIFT

Es wurde gezeigt, dass mit Hilfe der SIFT-Features markante Punkte einer Abbildung erfasst und identifiziert werden können. Diese Eigenschaft wird nun dazu verwendet, natürliche Passmarken in einer Umgebung zu definieren. „Natürlich“ werden die Passmarken genannt, da sie nicht künstlich in die Umgebung gebracht werden, wie ist etwa in [Manske und Jost \(2007\)](#) beschrieben wurde. Die Umgebung muss also nicht auf die Verwendung des Navigationssystems vorbereitet werden. Dadurch erhöht sich die Zahl der möglichen Anwendungsgebiete erheblich.

3.1 Kameramodell

Um die Position und Lage einer Kamera anhand einer Abbildung und der darin enthaltenen Passmarken bestimmen zu können, wird ein Modell benötigt. In [Manske und Jost \(2007\)](#) wurde ein 3D-Kameramodell mit 13 Parametern verwendet. Dieses Modell soll nun erläutert werden. Die 13 Parameter des Modells setzen sich aus sechs externen und sieben internen Parametern zusammen. Sowohl die externen als auch die internen Parameter lassen sich nicht direkt messen, sie müssen bestimmt beziehungsweise berechnet werden.

Externe Parameter Die sechs externen Kameraparameter geben die Position und die Orientierung (Pose) der Kamera in Bezug zum Weltkoordinatensystem an. Zur Angabe der Position werden die 3D-Koordinaten t_x, t_y, t_z verwendet. Sie bezeichnen den folgenden Translationsvektor \vec{t} .

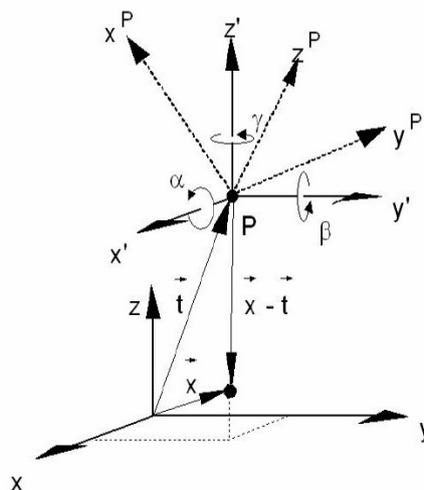


Abb. 3: Kameraparameter des Modells im Weltkoordinatensystem

Die Orientierung der Kamera wird mit den Winkeln α, β, γ angegeben. In Abbildung 3 werden die externen Parameter und deren Lage im Weltkoordinatensystem veranschaulicht. **Interne Parameter** Die Bildhauptpunktverschiebung angegeben durch eine 2D-Koordinate $(\Delta u_h, \Delta v_h)$ enthält zwei der sieben internen Parameter. Durch die Bildhauptpunktverschiebung wird der Durchstoßpunkt der optischen Achse durch die Bildebene beschrieben. Ein weiterer Parameter ist die Kammerkonstante oder auch Bildweite c . Die Linsenverzeichnungskorrektur wird durch die Parameter K_1, K_2, P_1, P_2 angegeben. Für die Bestimmung der internen Kameraparameter muss kein Aufwand betrieben werden, sie können durch die initiale Kalibrierung der verwend-

ten Kamera bestimmt werden. Die photogrammetrische Vermessungssoftware *Photomodeler*² stellt Funktionen zur initialen Kalibrierung mit Hilfe einer ausgedruckten Kalibrieranordnung bereit. Anschließend können die internen Parameter in der Software ausgelesen werden.

Die internen Parameter des Kameramodells sind also bekannt. Weiter gehen wir davon aus, dass eine Abbildung vorhanden ist, auf der mindestens drei Passmarken vorhanden sind. Von den Passmarken sind die 3D-Welkoordinaten und die entsprechenden 2D-Abbildungskordinaten bekannt. Mit Hilfe verschiedener Gleichungen, die an oben genannter Stelle zur Verwendung kamen, ist es dann möglich die unbekannt Parameter t_x , t_y , t_z , α , β und γ zu bestimmen. Somit erhält man dann die Pose des Kamerasystems und somit dessen Position.

3.2 Stereo-Abgleich

Ein Stereo-Kamerasystem mit drei Kameras wird für die Identifizierung der Features verwendet. Neben der Epipolar- und Disparitätsbedingung werden auch die SIFT Skalierungs- und Orientierungsbedingung verwendet, um die zwei auf der horizontalen Ebene gelbenen Abbildungen abzugleichen. Die Ergebnisse werden dann mit der Abbildung der oben liegenden Kamera verglichen.

Daraus ergeben sich nun die Abbildungskordinaten von Punkten auf drei Abbildungen. Aus diesen 2D-Koordinaten lassen sich nun die 3D-Koordinaten der Punkte im Raum relativ zum Kamerasystem berechnen. Dazu wird die Kleinste-Quadrate-Methode zur Annäherung der Lösung verwendet. Abbildung 4 zeigt eine Umgebung mit darin identifizierten Raumpunkten. Anhand der odometrischen Daten lassen sich die relative Position des Systems und die zu erwartenden Koordinaten der Raumpunkte in den Abbildungen berechnen. Daher fällt das Auffinden der Raumpunkte relativ leicht, da der zu untersuchende Bereich eingeschränkt wird. Die tatsächliche Berechnung bestätigt oder verbessert dann die vorherige Schätzung.



Abb. 4: Ergebnisse des Stereo-Abgleichs. Die Längen der Linien symbolisieren die horizontale und vertikale Disparität der Punkte. Daraus ergibt sich der Abstand des Punktes zum Kamerasystem. Nahe Objekte haben längere Linien.

²Photomodeler - photogrammetrische Vermessungssoftware, bewährt in der Forensik, Unfall-Rekonstruktion, Erstellung von 3D-Modellen ...

3.3 Kartenerzeugung

Sind nun Raumpunkte und deren Koordinaten bekannt, so können dies in einer Datenbank erfasst werden. Während sich nun das System in der Umgebung bewegt, wird kontinuierlich eine dreidimensional Karte der Umgebung erzeugt. Algorithmen, die in [Stephen Se \(2001\)](#) beschrieben wurden, gewährleisten, dass die Daten konsistent bleiben. Raumpunkte werden aus der Datenbank gelöscht, wenn sie nicht mehr zu finden sind und neue werden erfasst.

Probleme werden natürlich dadurch erzeugt, dass in der Umwelt nicht alle Punkte stets an der selben Stelle zu finden sind. Aus diesem Grund ist es auch möglich, in der Datenbank Raumpunkte anzulegen, die als permanent gekennzeichnet sind. Diese werden dann nicht mehr gelöscht und bilden sozusagen das Grundgerüst für eine Navigation.

Abbildung 5 zeigt eine zweidimensionale Abbildung einer Kartendatenbank, die die Raumpunkte dreidimensional speichert. Die Karte wurde in einem Raum erzeugt, durch den das Roboter-System eine Schleife gefahren ist.

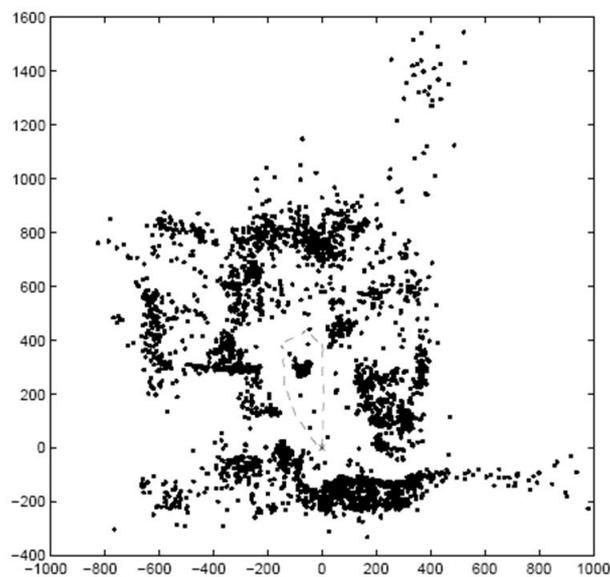


Abb. 5: SIFT-generierte Karte einer Testumgebung. Die gestrichelte Linie stellt den gefahrenen Weg des Roboter-Systems dar. Die Punkte identifizieren die gefundenen SIFT-Keypoints.

4 Stitching von Bildern

Bis hierher wurden die SIFT-Features vorgestellt und die Anwendung anhand der Navigation beschrieben. Daneben sind aber noch andere Anwendungen denkbar. Bereits entwickelt wurde beispielsweise ein System zum „stitchen“ von Bildern. Darunter versteht man das Zusammenfügen von Bildern zu einem einzigen Panoramabild. [Brown und Lowe \(2007\)](#) beschreibt das Vorgehen näher. Bisherige Stitching-Programme benötigten in der Regel die manuelle Abgleichung der einzelnen Bilder. Dabei mussten von Hand Überlappungspunkte definiert werden. Die Software konnte dann daraus ein Panoramabild erzeugen. Die Qualität des Panoramabildes hing dann sehr stark von der Genauigkeit ab, mit der die Überlappungspunkte gewählt wurden. Man kann sich vorstellen, dass dies bei einem manuellen Vorgang zu erheblichen Schwankungen kommen kann.

Bei der Verwendung der SIFT-Features kann die Qualität erhöht werden. Dies hat im Kern zwei Gründe. Zum Ersten werden Key-Points pixelgenau identifiziert. Das heißt, dass die Überlappungspunkte viel genauer den gleichen Raumpunkt identifizieren, als das man es mit der Hand und Computermaus erreichen könnte. Zum zweiten ist die Anzahl der gefundenen Key-Points in der Regel recht groß. Durch die große Anzahl der Überlappungspunkte lässt sich eine Panoramabild viel genauer berechnen. Perspektivische Störungen lassen sich so besser beseitigen. Abbildung 6 zeigt zwei Bilder und den darin gefundenen Überlappungspunkten. Spezielle

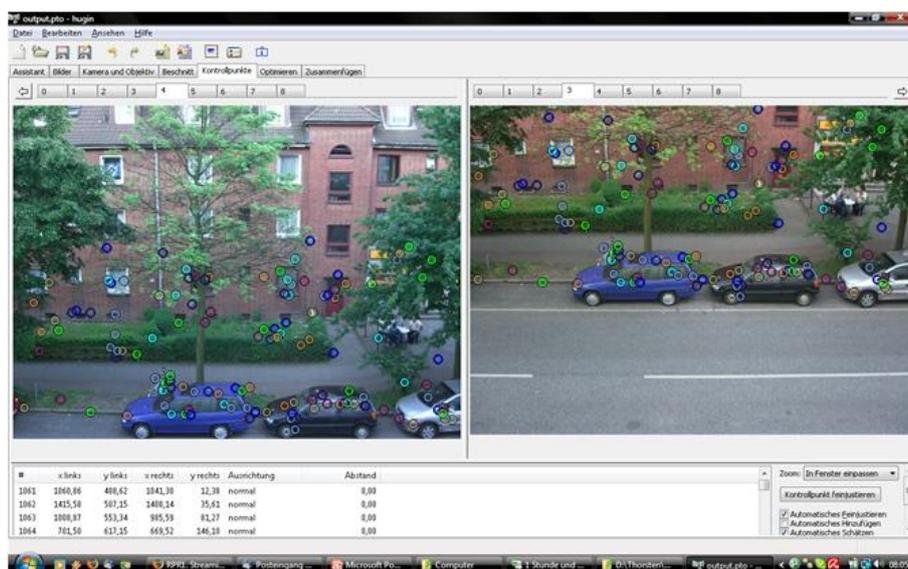


Abb. 6: Beispielabbildung einer Stitching-Software. Die kleinen Kreise in der Abbildung geben SIFT-Key-Points an, die für das Stitching als Überlappungspunkte verwendet werden. Zwei Kreise mit gleicher Farbe identifizieren den gleichen Raumpunkt.

Algorithmen gleichen dann perspektivische Verzerrungen und Unterschiede der Intensität der Belichtung der Abbildungen aus.

5 Ausblick

Am Ende dieser Arbeit sollen nun einige Ideen erarbeitet werden, wie in diesem Themenbereich weiter vorgegangen werden kann. Dabei sollen spezielle Themenvorschläge für Masterarbeiten im Studiengang Informatik entwickelt werden. Diese werden nun vorgestellt.

5.1 Implementierung

Ein Bereich, in dem sich ein Navigationssystem einsetzen lässt, ist der der Robotik oder auch der autonomen Fahrzeuge. Eine Einschränkung die dort immer wieder thematisiert wird, ist der Energiebedarf. Da solche Systeme möglichst unabhängig von externer Stromversorgung funktionieren sollen, ist eine effektive Implementierung und eine ressourcenschonende Wahl von Prozessoren und Elektronik unabdingbar. Um von dieser Seite eine Entwicklung anzugehen, wäre ein Vorschlag, die Algorithmen zur Lokalisierung der Key-Points in einem separaten Rechenkern durchzuführen. Dieser könnte auf der Hardware-Ebene, als Logik-Schaltung implementiert werden. Gängige Techniken dazu sind FPGAs, die mit der Hardwarebeschreibungssprache VHDL programmiert werden können.

Integrierte Systeme sind sparsamer im Energieverbrauch als PC-basierte Systeme. Außerdem kann die Verarbeitungsgeschwindigkeit erhöht werden.

5.2 Erweiterung der Navigation

Im Bereich der Navigation wären auch verschiedene Verbesserungen denkbar. Ein Vorschlag ginge in die Richtung, dass man die Navigation mit Objekterkennung verknüpft. Man denke beispielsweise an eine Navigation, die es erlaubt dem System Befehle zu übergeben. Diese könnten dann beispielsweise beinhalten, dass sich das System zu einem bestimmten Objekt bewegen soll, es holen oder einfach nur finden soll. Um dies zu realisieren, müsste die Karten-Datenbank um eine Objekt-Datenbank erweitert werden. Wie dies effizient gelöst werden kann, die Entwicklung einer entsprechenden Benutzer-Schnittstelle wäre Punkte einer weiteren Arbeit.

Ein weiterer Ansatz wäre das Vernetzen von mehreren autonomen Systemen, um eine globale verteilte Karte einer Umgebung zu erstellen. Da eine Karte eines Systems zunächst relativ vom System aus gesehen wird, stellt die Vernetzung mehrerer Karten eine gewisse Herausforderung dar.

Um eine Navigation im Umfeld einer Straßenführung mit Begrenzungslinien, Verkehrszeichen etc. zu realisieren, müsste eine neue Klasse von SIFT-Objekten eingeführt werden. Diese würden dann nicht die absolute Position in einer Karte darstellen, sondern beispielsweise die Position zwischen zwei Führungslinien, die eine Straße begrenzen. Gleichzeitig könnte natürlich auch die absolute Position interessant sein. Wenn man aber weiß, dass es nur eine Straße gibt, die zwei Bereiche miteinander verbindet, wäre die absolute Positionsangabe nicht besonders wichtig, solange man sich auf der Straße befindet. Hier wäre dann zu überprüfen, wie Begrenzungslinien beschaffen sein müssen, um sie für eine Navigation verwenden zu können. Wichtig wäre hierbei, dass Key-Points solcher Markierungen auch als solche erkannt werden können und nicht in die globale Positionierung ebezogen werden.

5.3 Anwendungen

Die Möglichkeiten der SIFT-Features beschränken sich nach Ansicht des Autors nicht nur auf die Verwendung im Bereich der Bildverarbeitung. Eventuell könnte es auch möglich sein, diese Technik in andere Bereiche zu übertragen, in denen es auf die Erkennung von Mustern in Datenbeständen ankommt. Experimente könnten belegen, dass sich die SIFT-Features auch zur Erkennung von Mustern in Audio-Strömen nutzen lassen. Im Prinzip lässt sich eine Audio-Datei als 2-dimensionales Feld betrachten, in der die Zeit die eine Dimension und die Tonhöhe (bzw. Frequenz) die zweite darstellt (siehe Abbildung 7. Auch hier sind verschiedene Szenari-

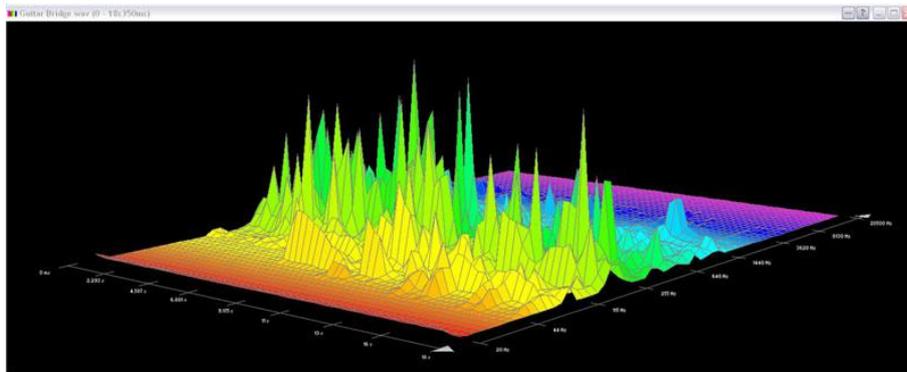


Abb. 7: Dreidimensionale Darstellung einer Frequenzanalyse

en denkbar. Es könnte ein System zur Stimm-Identifikation entwickelt werden, das Anhand von Mustern eine bestimmte Person identifizieren kann. Oder es könnte eine sprachgesteuerte Benutzerschnittstelle entwickelt werden, die Befehle von beliebigen Personen annehmen kann. Beide Varianten müssten zunächst durch Experimente bestätigt werden.

5.4 Fazit

Mit den SIFT-Features wurde ein mächtiges Instrument zur Objekterkennung geschaffen. Einige weiter führende Ansätze sind entwickelt worden. Unter anderem auch Algorithmen, die die Verwendung der SIFT-Features schneller macht. Nach Meinung des Autors wurde das ganze Potential dieser Technik jedoch noch nicht ausgeschöpft. Es wird vermutlich noch recht lange Entwicklungen in verschiedene Richtungen geben können. Aus diesem Grund ist dieses Thema für Abschlussarbeiten sehr interessant.

Abbildungsverzeichnis

1	Gauß-Filter und Gauß-Pyramide	3
2	Beispiel einer SIFT Anwendung	5
3	Kameraparameter des Modells im Weltkoordinatensystem	6
4	Ergebnisse des Stereo-Abgleichs	7
5	SIFT-generierte Karte	8
6	Beispielabbildung einer Stitching-Software	9
7	Dreidimensionale Darstellung einer Frequenzanalyse	11

Literatur

- [Brown und Lowe 2007] BROWN, Matthew ; LOWE, David G.: Automatic Panoramic Image Stitching using Invariant Features / Computer Science Department, University of British Columbia. 2007. – Forschungsbericht
- [Lowe 1999] LOWE, David G.: Object Recognition from Local Scale-Invariant Features / Computer Science Department, University of British Columbia. 1999. – Forschungsbericht
- [Lowe 2004] LOWE, David G.: Distinctive Image Features from Scale-Invariant Keypoints / Computer Science Department, University 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/
- [Stephen Se 2001] STEPHEN SE, Jim L.: Vision-based Mobile Localization and Mapping using Scale-Invariant Features / Computer Science Department, University of British Columbia. 2001. – Forschungsbericht

Glossar

Differenz-Gauß-Pyramide Eine Pyramide von Abbildungen, die aus den Differenzen von jeweils zwei Abbildungen aus einer Gauß-Pyramide berechnet wurden.

FAUST Fahrerassistenz- und Autonome Systeme Besteht aus mehreren Projekten zu den Themen Fahrerassistenzsysteme, Sensorik und Telemetrie in Fahrzeugen und autonome Fahrzeuge.

FPGA Field Programmable Gate Array Programmierbarer Logik-Baustein. Kann als komplexe Schaltungen (Microcontroller) eingesetzt werden.

Gauß-Pyramide Mehrfach mittels Gauß-Filter geglättete Abbildung, deren einzelne Stufen als Pyramide gestapelt werden.

Pose Die Pose ist die Kombination von Position und Orientierung im dreidimensionalen Raum.

Key-Point Ein Maxima- bzw. Minima-Grauwertpixel zwischen mehreren Ebenen einer Differenz-Gauß-Pyramide.

SCV Sensor Controlled Vehicle Eine mobile Plattform im FAUST Kontext.