Visual Odometry

Matthias Nieuwenhuisen

Uni Bonn

07.05.2007

Überblick

- Intention
- Featureerkennung
- Featurematching
- Trajektorienverfolgung
  - Stereokamera
  - Monokamera
- Ergebnisse

Intention

Probleme

- Dead Reckoning führt zu schlechter Odometrie
- Sensorsysteme wie DGPS und Laserscanner sind teuer und/oder unhandlich
- Navigation mit Abstandssensoren ist schwierig in flachem Gelände
- Bisherige visuelle Navigation basiert meist auf Landmarken oder ähnlichem

Visual Odometry

- Kameras sind günstige Sensoren
- Häufig sind Kameras für andere Anwendungen mitnutzbar
- Visual Odometry kann natürliche Merkmale oder Strukturen des Untergrundes zur Orientierung nutzen

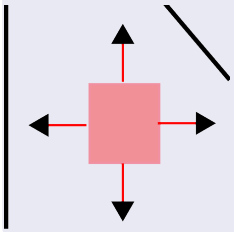
Featureerkennung

Zur Bestimmung der Roboterposition müssen Bildpunkte zu 3D Punkten trianguliert werden. Um verarbeitbare Punkte zu bekommen müssen zuerst Features aus den Bilder extrahiert werden.
Features müssen

- in mehreren Frames vorkommen um als Referenzpunkt zu dienen
- in ausreichender Anzahl vorhanden sein
- immer den selben Raumpunkt markieren

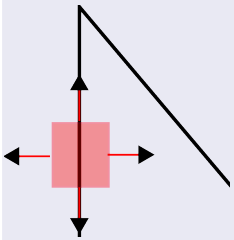
Harris Corner

Fläche



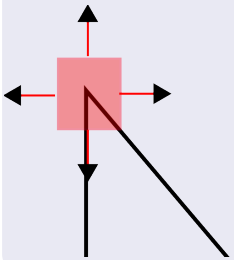
Keine signifikanten Intensitätsänderungen beim Verschieben in jede Richtung

Kante



Keine signifikante Intensitätsänderung beim Verschieben in Richtung einer Kante, Intensitätsänderung beim Verschieben in alle anderen Richtungen

Ecke

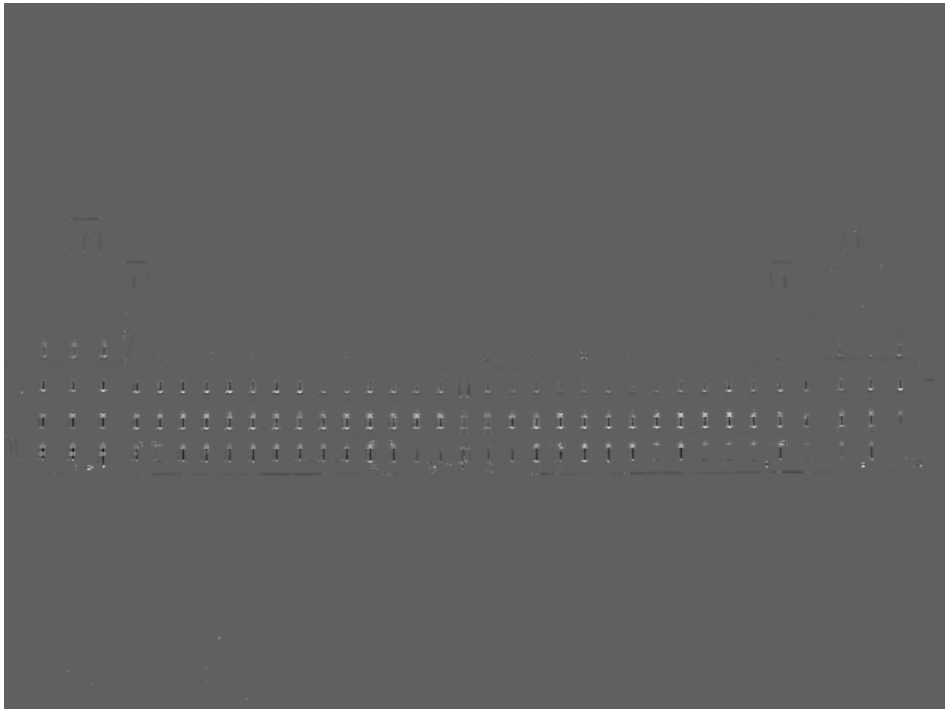


Signifikante Intensitätsänderungen beim verschieben in jede Richtung

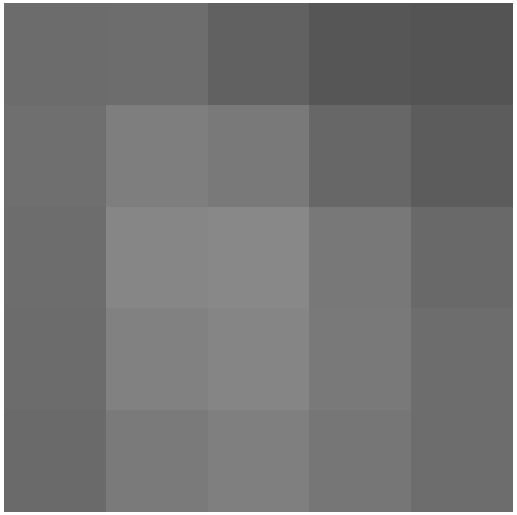
Harris Corner

Original

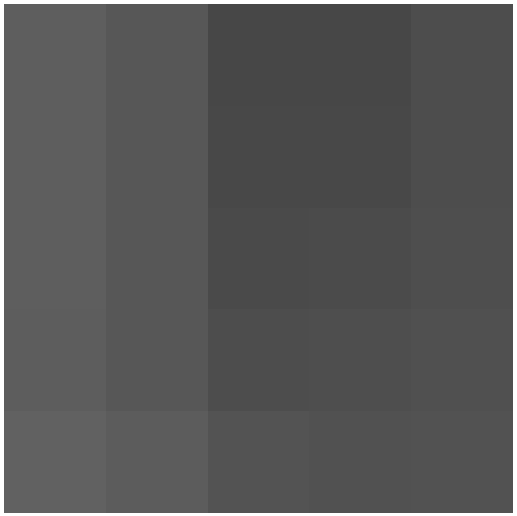
gefiltert mit Harris Corner Detektor



Harris Corner



Signifikantes Feature



Kein signifikantes Feature

Der Harris Corner Detektor findet bei kleinen Bildverschiebungen die gleichen Ecken.

Video liefert mehrere Frames pro Sekunde
$\Rightarrow$ kleine Bildverschiebungen zwischen 2 Frames

Als Feature sind nur Ecken interessant. Gewählt werden alle Pixel die in der Nachbarschaft keinen Pixel mit stärkerer Corner Response haben. So werden auch in Naturszenen ohne harte Ecken viele Features gefunden.

Harris Corner

Die Corner Response ist definiert als:

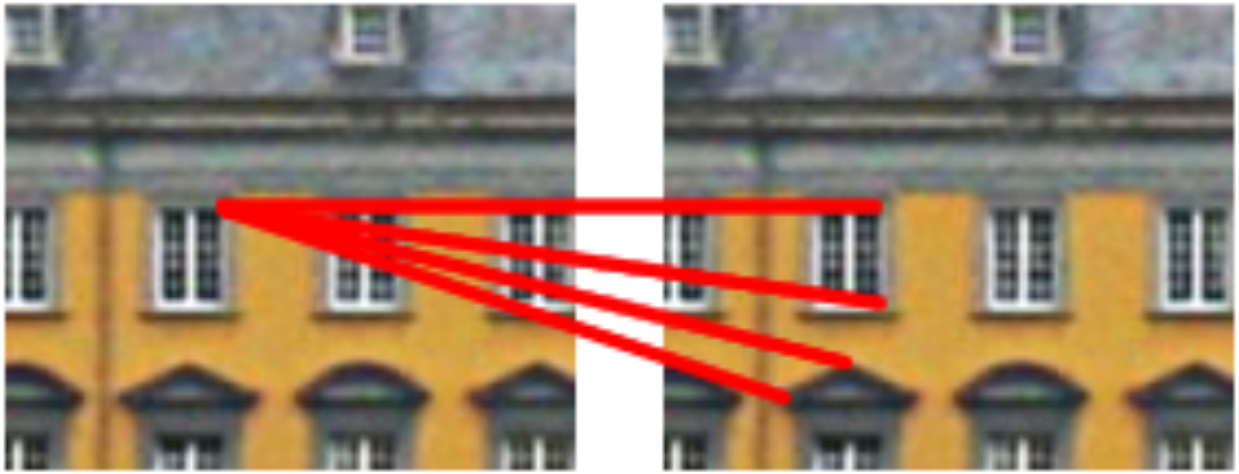
$$r = \det(M) - k\,\mathrm{trace}(M)^2$$

Die Matrix M basiert auf den Ableitungen des Bildes:

$$M = \begin{pmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{pmatrix}$$

Alle Werte werden zeilenweise berechnet.
So muss wenig im Speicher gehalten werden und man erreicht eine gute Cacheausnutzung.
Die Werte $I_x^2, I_x Iy, I_y^2$ können mit MMX parallelisiert berechnet werden.

Featurematching



Information welche Features in mehreren Bildern die selbe Stelle markieren wird benötigt.

Welche Features sind dies?

Featurematching

Normalisierte Korrelation über Intensitäten in 11×11 Pixelfeld mit Feature als Mittelpunkt:

Korrelation

$$r = \frac{121 \sum I_1 I_2 - \sum I_1 \sum I_2}{\sqrt{121 \sum I_1^2 - (\sum I_1)^2} \sqrt{121 \sum I_2^2 - (\sum I_2)^2}}$$

I sind die Intensitäten im Pixelfeld als 121 elementiger Vektor
Features werden verknüpft wenn sie sich jeweils gegenseitig als bevorzugten Partner haben.

Featurematching

Jedes Feature wird gegen viele andere Features getestet.
Die Vorberechnung aller nur von einem Feature abhängigen Werte ist somit sinnvoll:

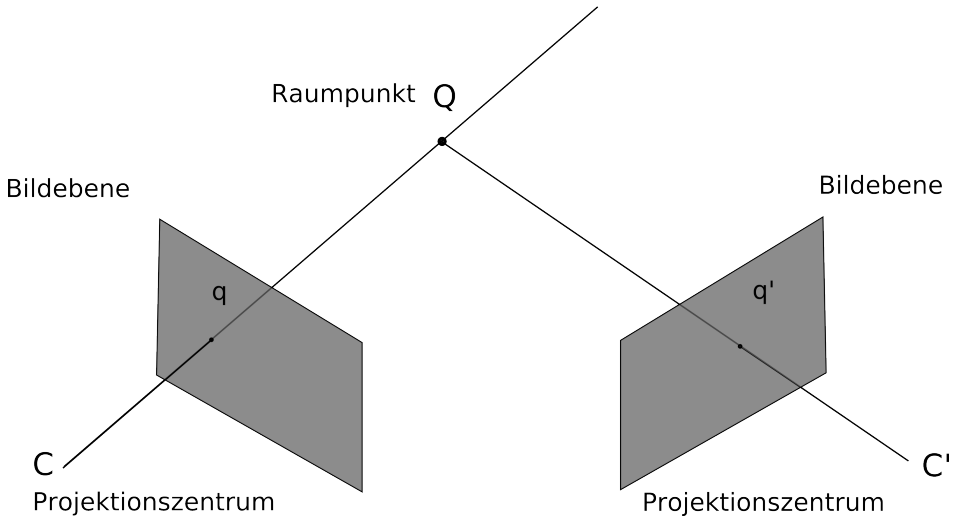
$$\begin{aligned} r &= \frac{n\sum I_1 I_2 - \sum I_1 \sum I_2}{\sqrt{n\sum I_1^2 - \left(\sum I_1\right)^2}\sqrt{n\sum I_2^2 - \left(\sum I_2\right)^2}} \\ &= \left(n\sum I_1 I_2 - A_1 A_2\right) B_1 B_2 \end{aligned}$$

A und B sind skalar $\Rightarrow$ Nur noch $\sum I_1 I_2$ ist bei jedem Test zu berechnen.
Mit MMX können alle 121 Multiplikationen von I_1, I_2 parallel durchgeführt werden.

Featurematching

Über mehrere Frames entstehen so Historien über die Bewegung der Features im Bild.

Trajektorienbestimmung



Gegeben:
Positionsänderung von 2D Features über mehrere Frames

Gesucht:
Position der Features im Raum und Bewegung der Kamera

Da mehrere Perspektiven der Features bekannt sind ist die Triangulation in Raumpunkte möglich.

Stereokamera

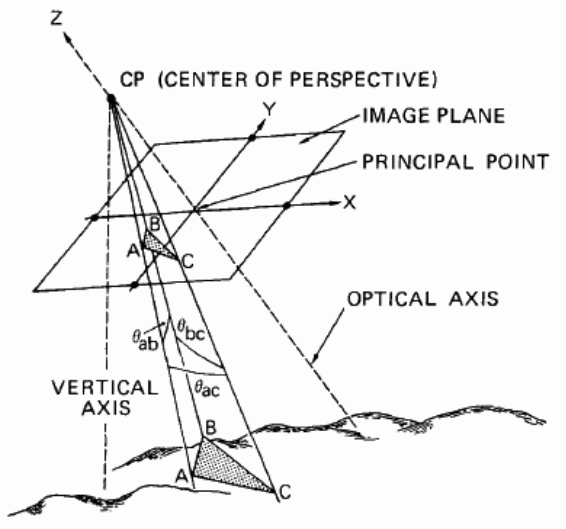
- Bei einer Stereokamera wird die Umgebung von zwei Kameras aufgenommen.
- Die Position der Kameras zueinander ist bekannt, es können direkt relative 3D Koordinaten jedes Features zurückgeliefert werden.
- Zur Bestimmung der Kameraposition wird RANSAC mit Dreipunktalgorithmus als Hypothesengenerator verwendet.

Erinnerung: Preemptive RANSAC

Zur Modellbildung wird an vielen Stellen ein preemptive RANSAC Algorithmus verwendet um die unsicheren Daten zu verarbeiten.

1. Der RANSAC-Algorithmus bekommt Features von 2 oder 3 Bildern als Eingabe
2. Zufällig werden Features ausgesucht und diese als Eingabe für einen beliebigen weiteren Algorithmus (Hypothesengenerator) genutzt.
3. Viele Hypothesen werden generiert und gegen alle Features getestet.
4. Die beste Hypothese wird als Ergebnis zurückgeliefert.

Dreipunktalgorithmus

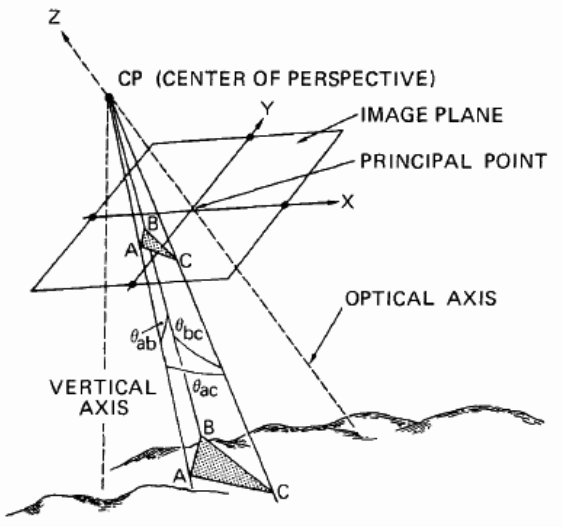


Die triangulierten Features werden als Eingabe für einen RANSAC-Algorithmus verwendet.

Da ihre Position im Raum bekannt ist kann man jeweils Dreiecke bekannter Seitenlänge finden.

Diese Dreiecke werden zur Lösung des Location-Determination-Problems mit dem Dreipunktalgorithmus verwendet.

Dreipunktalgorithmus



- Der preemptive RANSAC-Algorithmus erstellt 500 Hypothesen.
- Der bisher gefahrene Pfad wird zu dem Punkt der besten Hypothese verlängert.
- Dieser Schritt wird für mehrere Frames durchgeführt.

Bestimmung neuer Features

- Alte Features verlassen durch die Bewegung der Kamera den sichtbaren Bereich.
- Neue Features werden in jedem Frame gefunden und Historien gebildet.
- Diese werden aus Effizienzgründen nicht direkt in 3D Punkte trianguliert.
- Beim triangulieren können durch die bekannte Kameraposition direkt absolute Koordinaten bestimmt werden.

Firewall

- Gegenseitige Abhängigkeit von Positionsbestimmung der Kamera und Triangulation von Punkten führt mit der Zeit zu Fehlerakkumulation.
- Eine gelegentlich eingesetzte Firewall verhindert Fehlerpropagation und -akkumulation.
- Für die weitere Trajektorienbestimmung wird das Koordinatensystem beibehalten.
- Alle anderen Parameter wie triangulierte Features und Historien werden gelöscht.

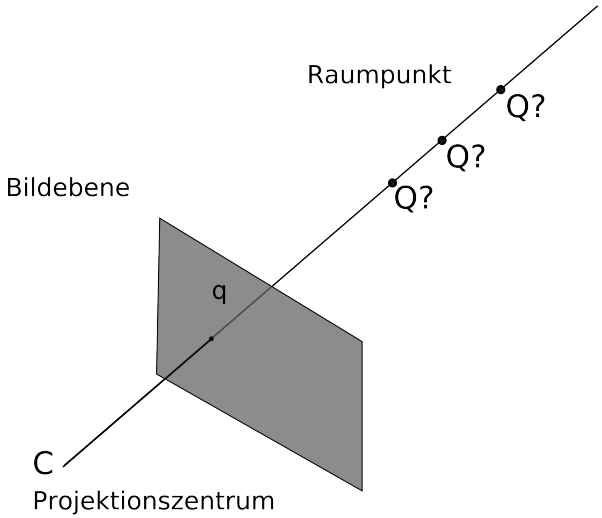
Stereokamera

Für einen Roboter mit zwei Kameras läuft die Trajektorienbestimmung so ab:

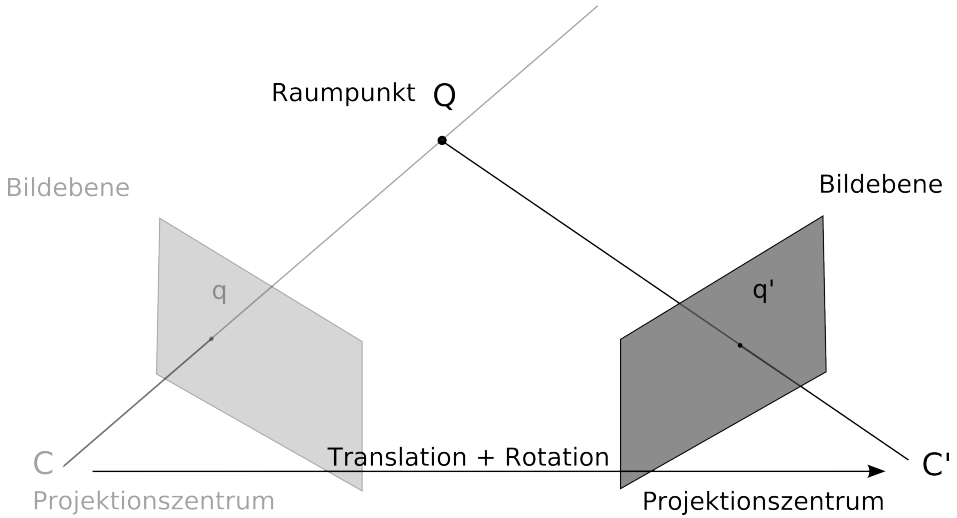
1. Featurematching zwischen linkem und rechtem Kamerabild
2. Trianguliere die Features in 3D Punkte
3. Verfolge die Position einige Zeit mit dem Dreipunktalgorithmus/RANSAC mit einer Kamera
4. Trianguliere neu gefundene Features mit den Aufnahmen aus beiden Kameras
5. Trianguliere alle Punkte neu mit den Aufnahmen der jeweils ersten und letzen Beobachtung und beginne bei Schritt 3
6. Setze gelegentlich eine Firewall ein

Monokamera

Eine einzelne Kamera liefert keine Tiefeninformation
Mehrere Perspektiven eines Punkte werden benötigt für eine Triangulation



Mit einer Kamera möglich durch Positionsänderung



Monokamera

Relative Pose Estimation Problem

- Um die neue Position bestimmen zu können werden 3D Referenzpunkte benötigt.
- Um 3D Referenzpunkte zu bestimmen wird die Position der Kamera benötigt.

Fünfpunktalgorithmus

Eine Möglichkeit Rotation und Translation einer Kamera zwischen zwei Aufnahmen zu bestimmen ohne auf bekannte Punkte im Raum zurückzugreifen ist der Fünfpunktalgorithmus.

Ohne bekannte Raumpunkte kann die Translation nur bis auf einen Skalierungsfaktor bestimmt werden.
Somit fällt ein Freiheitsgrad weg und es werden fünf Korrespondenzpunkte benötigt.

Kameramodell

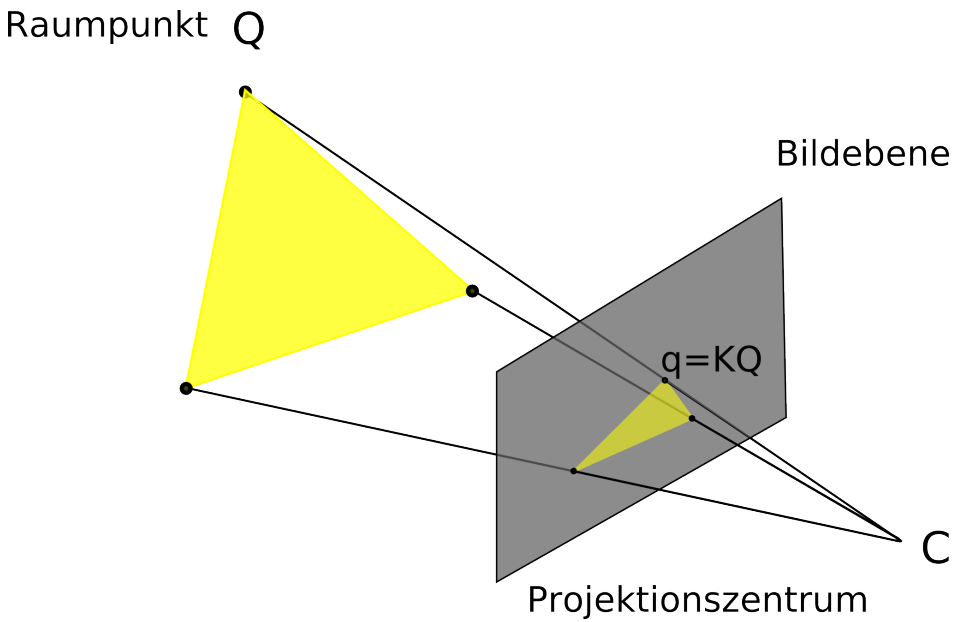
Innere Kameraparameter

- Brennweite
- Linsenverzerrung
- Pixelgröße
- ...

Äussere Kameraparameter

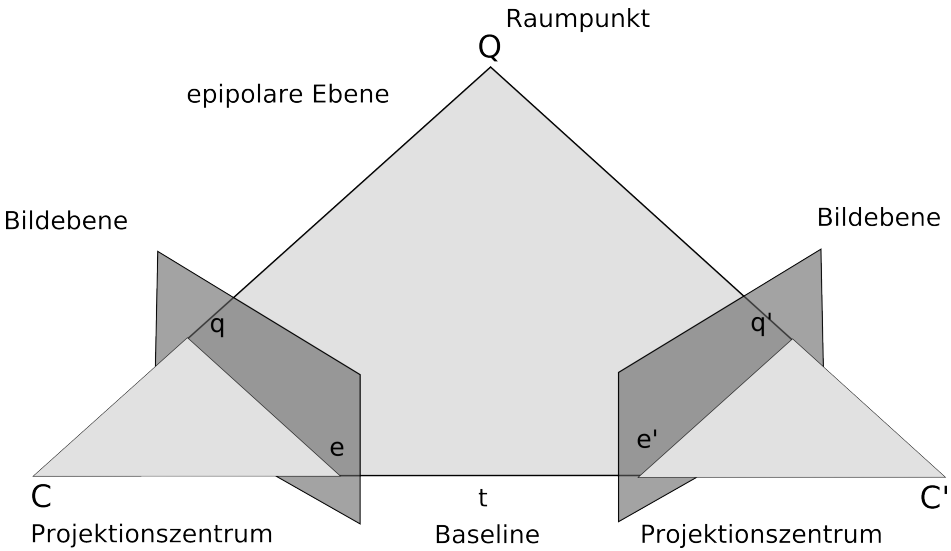
- Rotation
- Translation

Fünfpunktalgorithmus



- Kamerabild entspricht perspektivischer Projektion der Umgebung auf die Bildebene
- Die Projektion wird vorher bestimmt und kann umgekehrt werden
- Unterschied zwischen den Bildern ist nur noch Rotation und Translation der Kamera

Komplanaritätsbedingung



Vektoren die in einer Ebene liegen spannen keinen Raum auf.

Andere Formulierung:
Für Vektoren a, b, c ist $(a \times b)\, c = 0$.

Für die Vektoren
$(Q - C), (Q - C'), t$
gilt dies offensichtlich.

Fundamentalmatrix

Die Umkehrabbildung der perspektivischen Projektion ergibt Vektoren vom Projektionszentrum durch die Raumpunkte.

Zu jedem Raumpunkt gibt es einen solchen Vektor zu beiden Projektionszentren.

Daraus ergibt sich die Fundamentalmatrix:

Fundamentalmatrix

$$F = K_2^T [t]_x R K_1^T$$

Komplanaritätsbedingung

Für alle Abbildungen q und q' eines Punktes Q im Raum gilt die Komplanaritätsbedingung:

$$\begin{aligned} F &= K_2^T \, [t]_x \, R K_1^T \\ q'^T F q &= 0 \end{aligned}$$

R ist die Rotation der zweiten Kameraperspektive
K_i beschreiben Abbildung von Raumpunkten auf die Bildebene
t ist die Translation der zweiten Kameraperspektive

Essentialmatrix

Ziel ist es die Fundamentalmatrix zu bestimmen.

In einem späteren Schritt werden Rotation und Translation daraus extrahiert.

Da die K_i bekannt sind muss nur der Rest der Fundamentalmatrix gefunden werden: Die Essentialmatrix E.

Die Komplanaritätsbedingung lässt sich umformulieren zu $Q\tilde{E} = 0$.

Gleichungssystem

Um die Gleichung $Q\tilde{E} = 0$ zu erfüllen muss $\tilde{E}$ im Kern der Matrix Q liegen.

Eine Singulärwertzerlegung von Q findet eine Basis des Kerns ist aber zu teuer.

Eine QR-Zerlegung kann effizient durchgeführt werden und ist numerisch nur geringfügig schlechter

Gleichungssystem

Sind Basisvektoren des Kerns von Q gefunden kann daraus die Essentialmatrix konstruiert werden:

$$E = xX + yY + zZ + wW$$

Theorem

Eine reelle nicht triviale 3×3 Matrix E ist eine Essentialmatrix genau dann wenn gilt

$$EE^T E - \frac{1}{2}\text{trace}\left(EE^T\right)E = 0$$

Durch einsetzen von E entsteht ein Gleichungssystem mit dem gültige x, y, z, w bestimmt werden können.

Nullstellenbestimmung

Durch partielles lösen des aufgestellten Gleichungssystems erhält man ein weiteres Gleichungssystem:

$$\begin{pmatrix} [3] & [3] & [4] \\ [3] & [3] & [4] \\ [3] & [3] & [4] \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = 0$$

[N] sind Polynome vom Grad N in z.
Um das System zu lösen muss die Determinante verschwinden.
Dies führt zu einem Polynom vom Grad 10.
Die Nullstellen dieses Polynoms sind gültige Lösungen zur Konstruktion von E.

Nullstellenbestimmung

Eine einfach zu implementierende Methode ist die Eigenwertzerlegung einer Begleitmatrix

$$\begin{pmatrix} -n_9 & -n_8 & \dots & -n_0 \\ 1 & & & \\ & \ddots & & \\ & & 1 & \end{pmatrix}$$

Sturm-Sequenzen ermöglichen eine effiziente Eingrenzung der Nullstellensuche auf kleine Intervalle.
Die Sequenzen bestehen aus einer Zerlegung des Polynoms in Polynome niedrigeren Grades.

Rotation und Translation

Durch Singulärwertzerlegung der Essentialmatrix kann aus dem Ergebnis die Rotation und Translation der Kamera abgelesen werden.

Es gibt hier vier mögliche Lösungen von denen nur eine korrekt ist.

Ob eine Lösung korrekt ist wird geprüft durch die Lage der Raumpunkte zu den Bildebenen.

Wenn ein Raumpunkt hinter der Bildebene liegt wird die nächste Lösung überprüft.

Fünfpunktalgorithmus im RANSAC

- Eingabemenge: Features aus drei Frames
- Jeweils aus zwei Frames werden fünf Punkte für Hypothesengenerierung gezogen
- Trianguliere die Features mit den gefundenen Hypothesen
- Nutze diese 3D Punkte um die Kameraposition im dritten Frame zu finden
- Führe Scoring über die drei Frames durch

Monokamera

Für einen Roboter mit einer Kamera läuft die Trajektorienbestimmung so ab:

1. Verfolge Features über einige Frames und bestimme Positionen mit dem Fünfpunktalgorithmus/RANSAC
2. Trianguliere mit den gefundenen Kamerapositionen die Features
3. Bringe die Features ins korrekte Koordinatensystem mit Least-Squares/RANSAC
4. Verfolge die Position einige Zeit mit dem Dreipunktalgorithmus/RANSAC
5. Trianguliere alle Punkte neu mit den Aufnahmen der jeweils ersten und letzen Beobachtung und beginne von vorn
6. Setze gelegentlich eine Firewall ein

Effizienz

Mit QR-Zerlegung zur Bestimmung der Nullvektoren und Sturm-Sequenzen zur Nullstellenbestimmung ergibt sich folgende Laufzeit (auf einem 550MHz System):

Laufzeiten

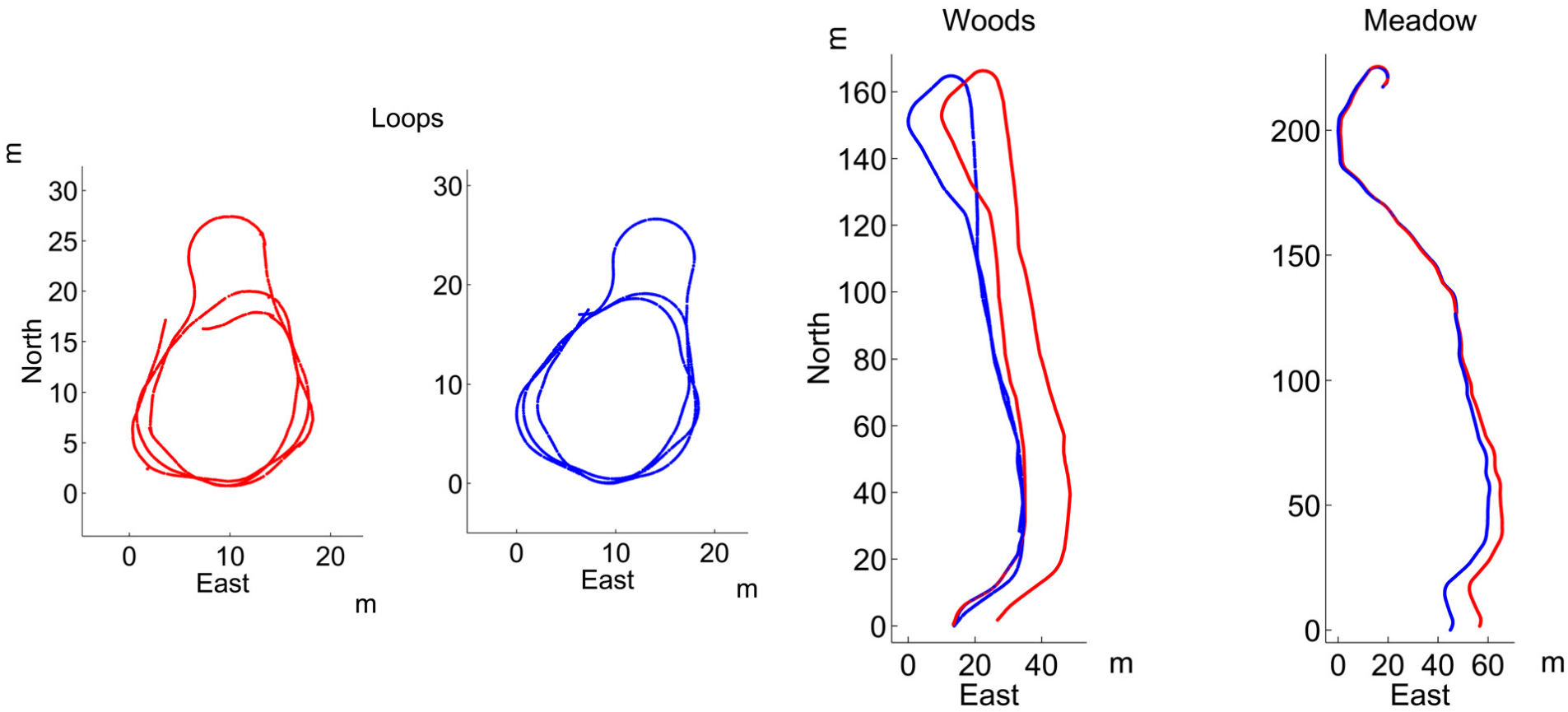
Featureerkennung: 30ms
Featurematching: 34ms - 160ms
Positionsbestimmung 5-Punkt: 50ms
Positionsbestimmung 3-Punkt: 2,5ms

Die großen Zeitunterschiede beim Featurematching kommen daher, dass die Suchweite für Korrespondenzen nicht fest ist und vom Videosignal und den gefundenen Features abhängt.

Ergebnisse

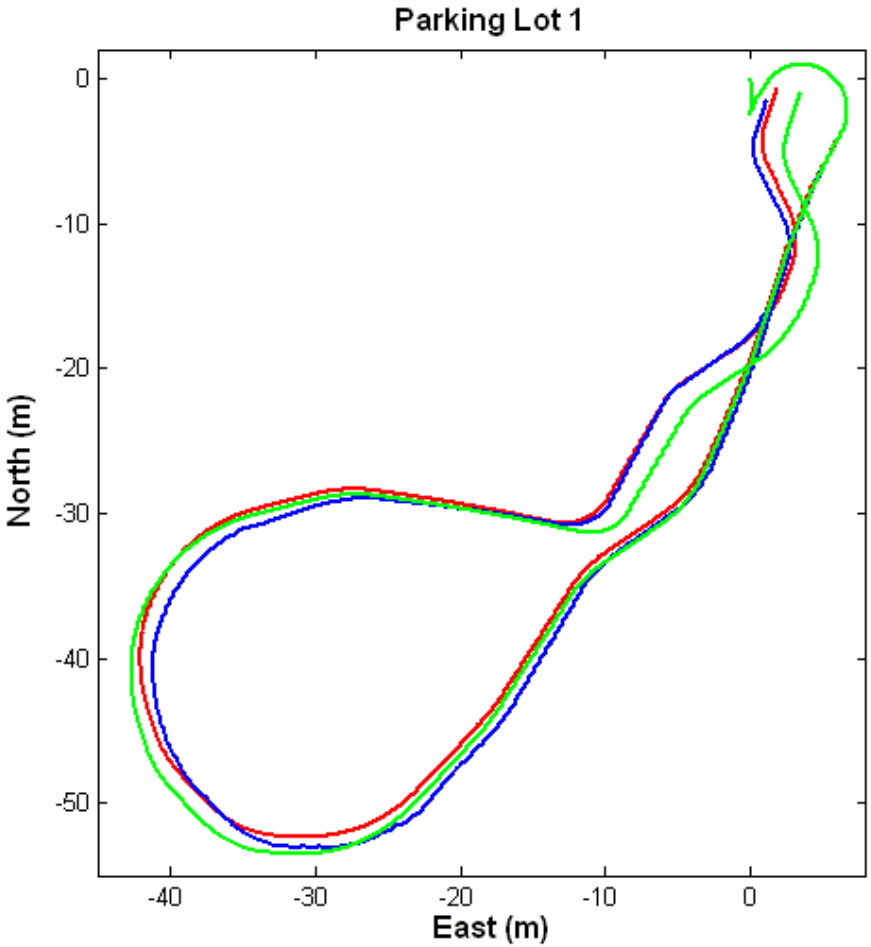
- Stereokameras
- Laserscanner
- Differential GPS
- Inertiales Navigationssystem

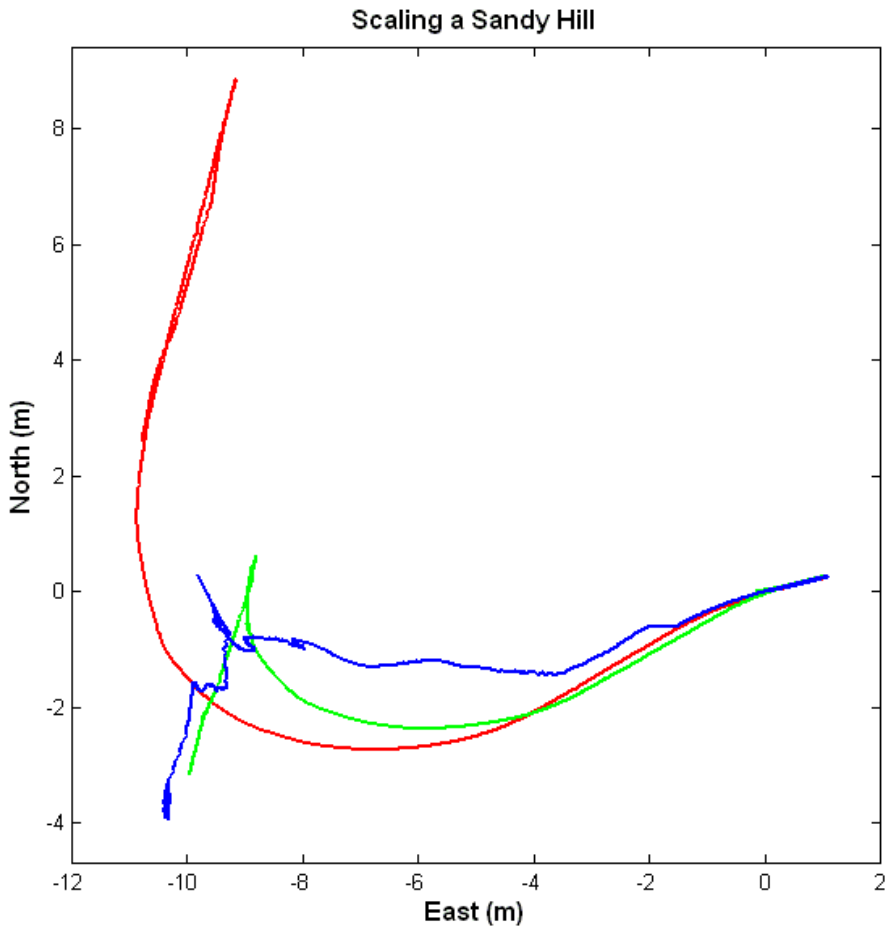
Ergebnisse



DGPS, Visual Odometry

Ergebnisse





Radencoder, DGPS, Visual Odometry

Zusammenfassung

- Features müssen in jedem Frame gefunden werden
- Featurematching lässt sich mit Vorverarbeitung beschleunigen
- Mit Stereokameras kann die Positionsbestimmung schnell durchgeführt werden
- Mit einer Kamera lassen sich mit Hilfe des Fünfpunktalgorithmus auch 3D Positionen gewinnen
- Für Echtzeitanwendung ist es nötig jeden Schritt effizient zu implementieren
- Die Trajektorienerkennung ist bei schlechtem Untergrund erheblich besser als mit herkömmliche Odometriedaten
- Messfehler in der zurückgelegten Strecke sind sehr klein

Literatur

David Nistér.
An efficient solution to the five-point relative pose problem.
In *CVPR*, pages 195–202. IEEE Computer Society, 2003.

David Nistér, Oleg Naroditsky, and James R. Bergen.
Visual odometry.
In *CVPR (1)*, pages 652–659, 2004.