

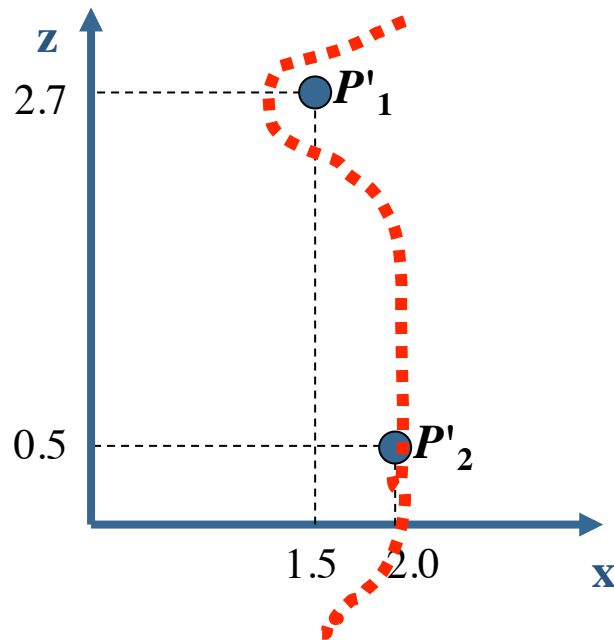
Aber...

Das Problem Punktkorrespondenz

Gegeben „Modell“ (z.B. Referenz-Scan, kontinuierliche Kurve),
welche Scanpunkte P'_i („Datenpunkte“) entsprechen
Modell-Punkten P_i ?

Wieviele/welche haben überhaupt eine Entsprechung?

Und wenn – welche?



Beispiel: Messpunkte P'_i nach Schätzung
der Poseänderung überlagert einer
Modellmenge (rot)

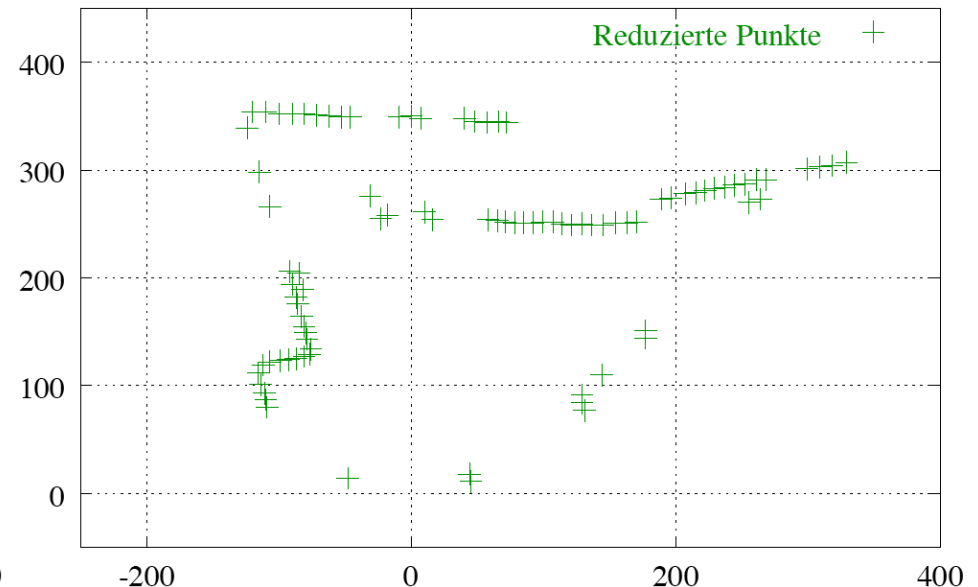
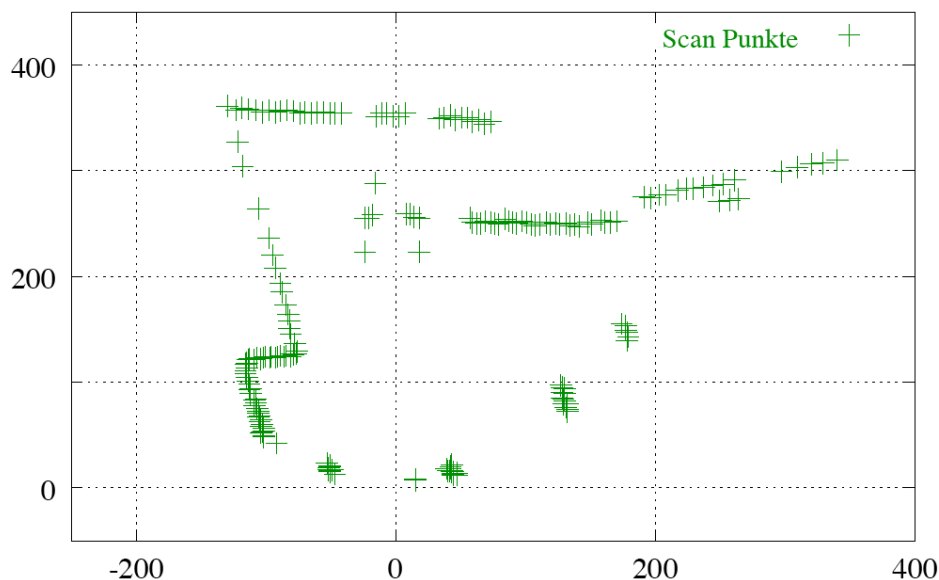
- Abstände minimieren?
- Überdeckung maximieren?
- (θ, T) minimieren?
- Sichtbarkeit berücksichtigen?

Vorarbeit für Korrespondenzbestimmung

Nicht alle Punkte aus Modell- und Datenmenge müssen überhaupt fürs Scanmatching herangezogen werden!

Beispiel: Reduziere Datenpunkte durch Reduktionsfilter

😊linear in Menge der Datenpunkte



Korrespondenz I: Die Regel *closest-point*

- Geh die Punkte (ggf. gefiltert) der Datenmenge durch
- Wenn für Datenpunkt p_i' in einer ε -Umgebung mindestens 1 Modellpunkt liegt, dann nimm den nächsten Modellpunkt dieser Umgebung als korrespondierenden Punkt; definiere dabei Nähe als Euklidischen Abstand.
(Gibt es innerhalb der ε -Umgebung keinen Modellpunkt, hat p_i' keinen korrespondierenden)
- ➔ Grundlage des Algorithmus **ICP** (*Iterative Closest Points*, Besl/McKay, 1992) (z.T. *Iterative Corresponding Points*)
- ➔ Analytisch: Konvergiert (gemeinsam mit Fehlergradientenabstieg) garantiert auf lokales Minimum von $E(\theta, T)$
- ➔ Empirisch: Kompensiert Translationsfehler gut, wenn Rotationsfehler gering

2D-Scanmatching mit ICP

Eingabe : Punktmenge M , Scan D .

Ausgabe: Transformation $\mathbf{T} = (t_x, t_z, \theta)^T$, die D mit M registriert.

```
1: if Initiale Poseschätzung existiert then
2:   Setze  $\mathbf{T}$  gleich dem geschätzten Poseversatz
3:   Transformiere  $D$  um  $\mathbf{T}$ 
4: else
5:    $\mathbf{T} = (t_x, t_z, \theta)^T = (0, 0, 0)^T$ 
6: end if
7: repeat
8:   Bestimme die Paarungen korrespondierender Punkte (z.B. mittels ICP-Regel)
9:   Berechne  $\Delta\mathbf{T}$  durch Minimierung der Fehlerfunktion  $E$  wie in Folie 224 (Lu/Milios)
10:   $\mathbf{T} = \mathbf{T} + \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \Delta\mathbf{T}$  // Aktualisierung der Transformation
11:  Transformiere  $D$  mit  $\Delta\mathbf{T}$ 
12: until Transformations-Inkrement  $\Delta\mathbf{T}$  ist betragsmäßig unter einer Grenze.
13: return  $\mathbf{T}$ 
```

K II: Die Regel *matching-range-point* (Lu/Milios)

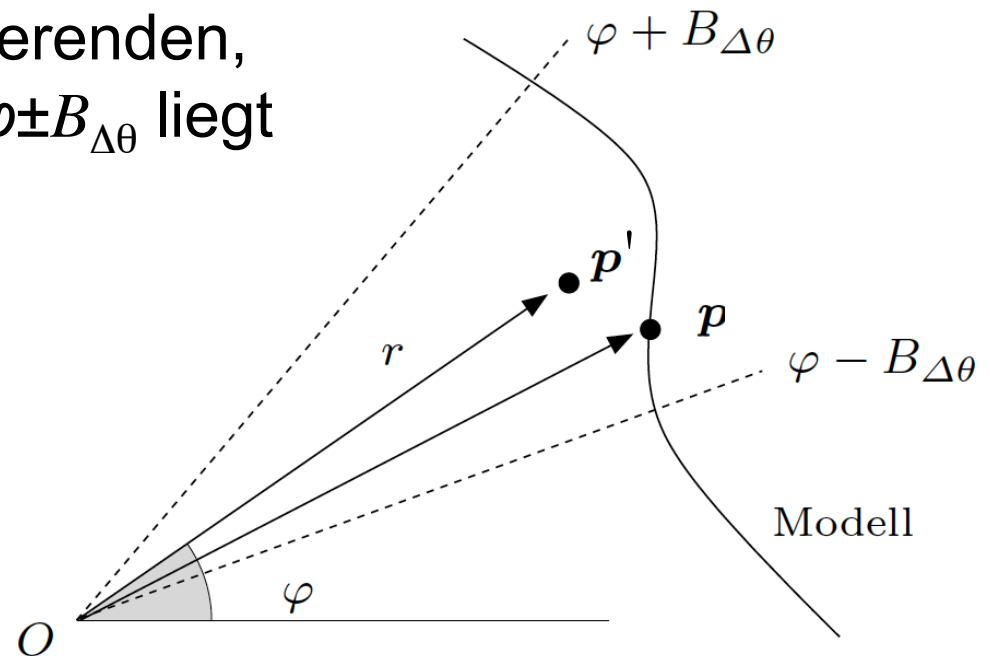
- Verwende die Datenpunkte in Polardarstellung (r, φ) (wie sie vom Scanner kommen)
- Sei $\Delta\theta$ die zuletzt ermittelte Änderung der Rotationsschätzung. ($\Delta\theta$ nimmt normalerweise ab über die Laufzeit des Scan-Matching.)

Für Datenpunkt $p'=(r, \varphi)$ wähle, wenn vorhanden, den

Modellpunkt p als korrespondierenden, der in einem Winkelsegment $\varphi \pm B_{\Delta\theta}$ liegt und für den $(\|p\| - r)^2$

minimal ist

$B_{\Delta\theta}$ Schranke für Winkelirrtum;
sinkt über Iterationen des Algorithmus

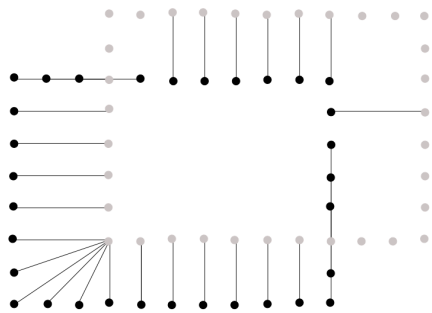


Unterschied *matching-range*-*closest-point*

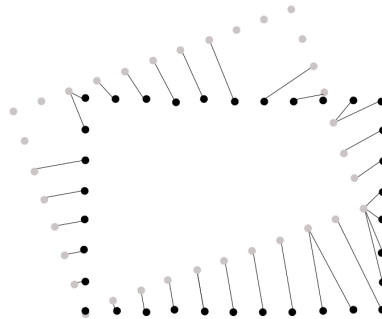
Beispiele

closest point

schwarz: Daten;
grau: Modell



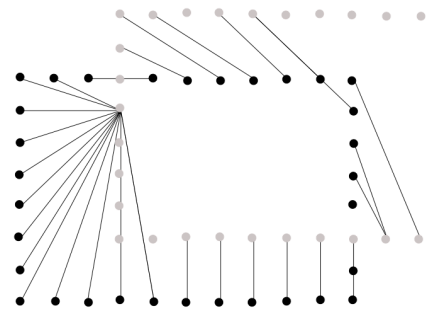
Translation



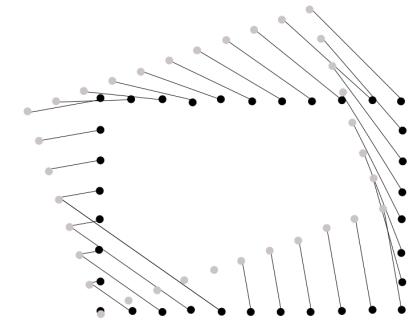
Rotation

matching range

schwarz: Daten;
grau: Modell



Translation



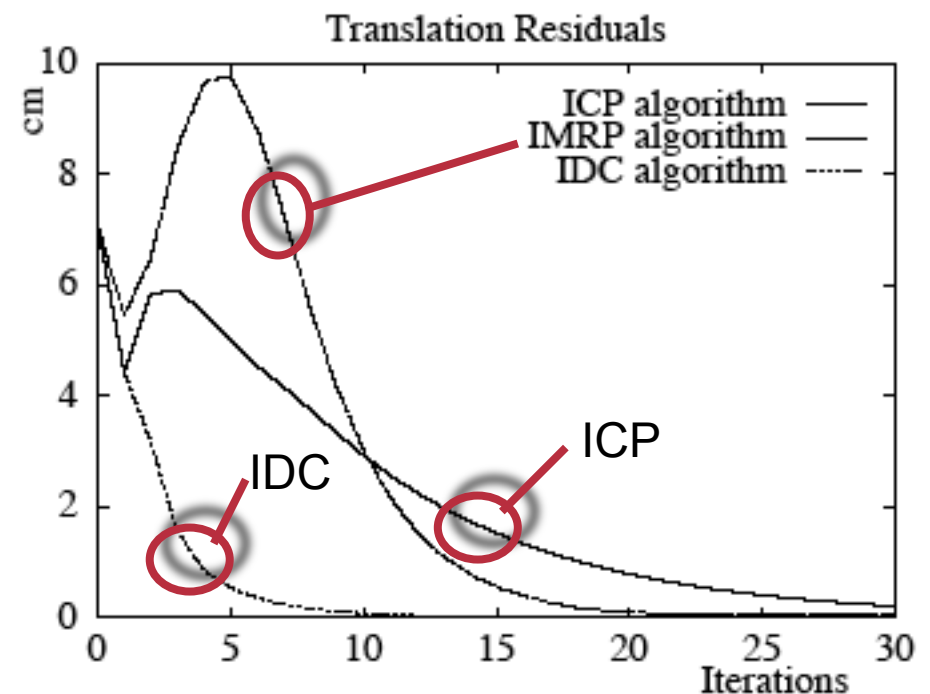
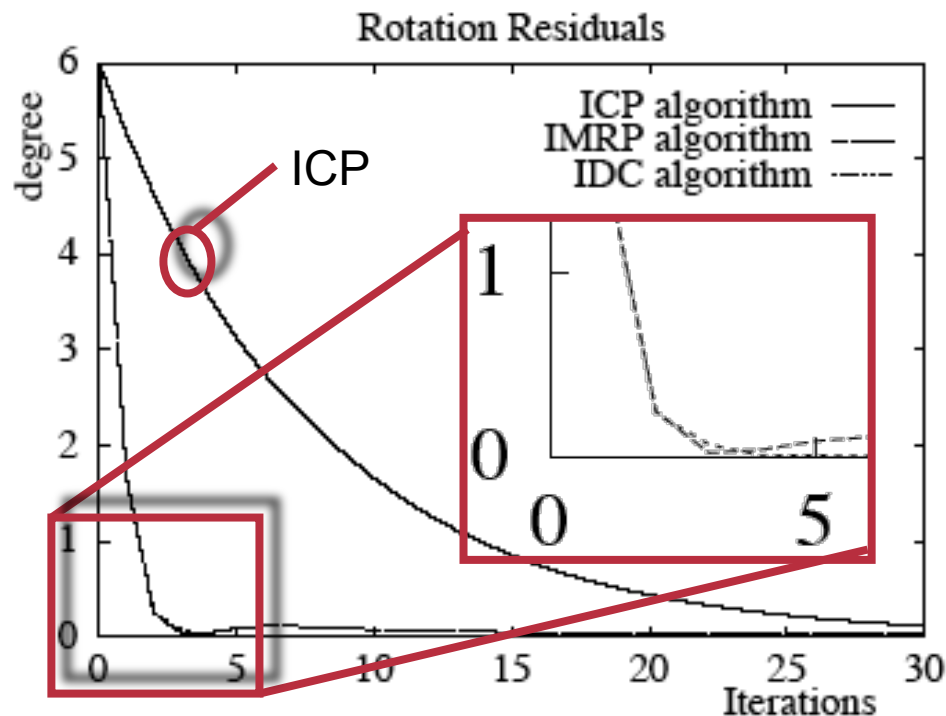
Rotation

matching range wählt Korrespondenzen in einheitlicher Rotationsrichtung

- ↪ Kompensiert Rotationsfehler gut;
übersieht Translationsfehler, solange B_θ noch hoch ist

IDC (*Iterative Dual Correspondence*)

- Ermittle Korrespondenzen je nach *c-p-* und nach *m-r-p*-Regel
- Berechne für beide Korrespondenzen jeweils optimale Pose-schätzung (z.B. minimaler Restfehler, s.o.; maximale Kovarianz, s.u.)
- Ergebnis: Rotation aus *m-r-p*- und Translation aus *c-p*-Regel



Lokalisierung für schnelle Roboter

Kai Lingemann&al.:
High-Speed Laser Localization for Mobile Robots. J.RAS, 2005

... den Film kennen
wir schon (#142)



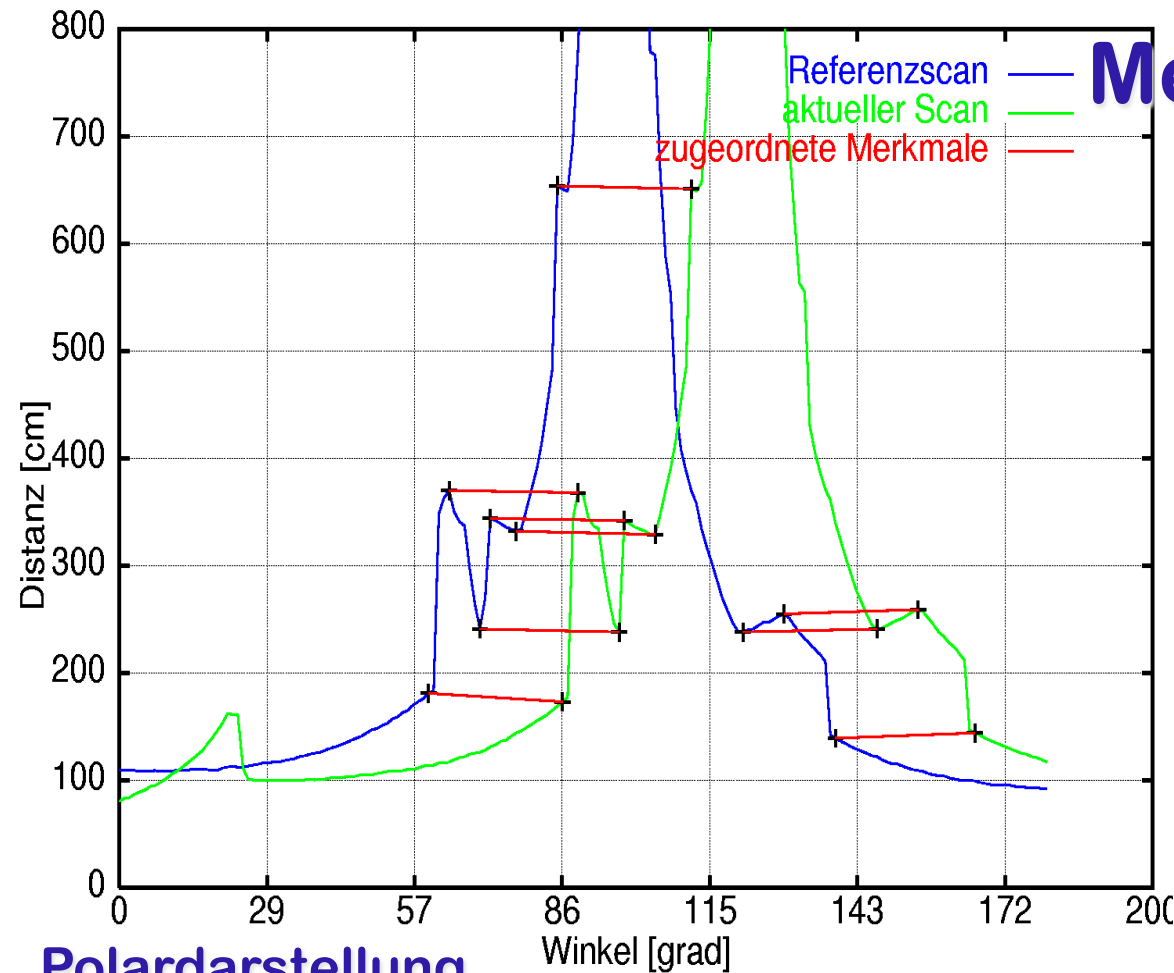
HAYAI:
Highspeed
And Yet Accurate
Indoor/outdoor
Tracking

速い

Ideen zum schnellen Pose-Tracking

- Geh aus von Scanmatching
(Rotation/Translation konsekutiver Scans)
- Betrachte **lokale Merkmale**
(Extrema/Ableitung(en)=0 in geglätteten Daten)
- Arbeite in Polardarstellung
(nutze Extrema-Reihenfolge in Scans)
- Translation&Rotation in geschlossener Form
wie bei Scanmatching (keine Iteration)
- Füge nötigenfalls (merkmalsarme Scans) Information hinzu
 - Winkelhistogramme
 - EKF

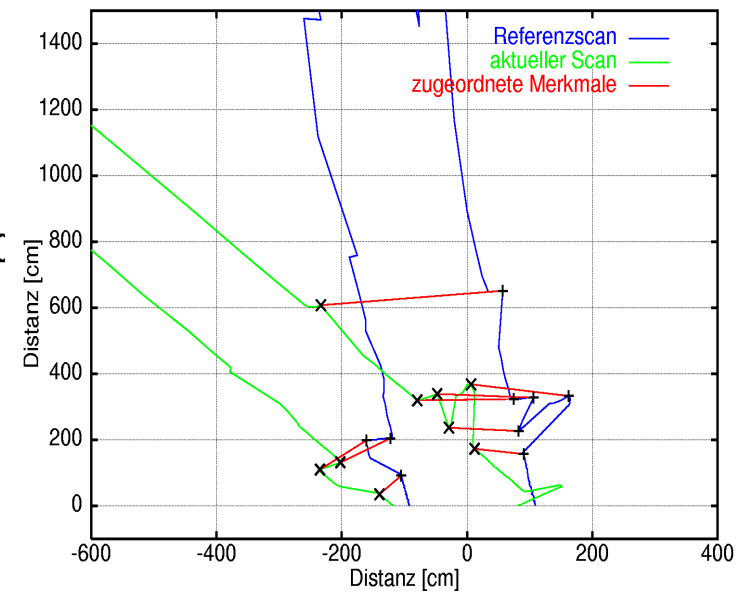
Merkmalszuordnung



Polardarstellung

Scans mit linearen Filtern vorverarbeitet

Kartesische Darstellung



HAYAI-Mathematik: Ansatz

Gegeben zwei Scans M, D . Finde Rotation \mathbf{R} und Translation \mathbf{t} , unter denen D „möglichst exakt“ auf M abgebildet wird; Evaluation über zu minimierende Fehlerfunktion E :

Bei paarweiser Kongruenz von N Punkten aus M und D (benutze genau N extrahierte Merkmale, $N \ll |D|$):

$$E(\mathbf{R}, \mathbf{t}) = \sum_{i=1}^N \left\| \mathbf{m}_i - (\mathbf{R} \mathbf{d}_i + \mathbf{t}) \right\|^2$$

➡ berechnen wie in ICP/IDC
s. Beweis Folie 225

Lokalisierung bei schneller Fahrt

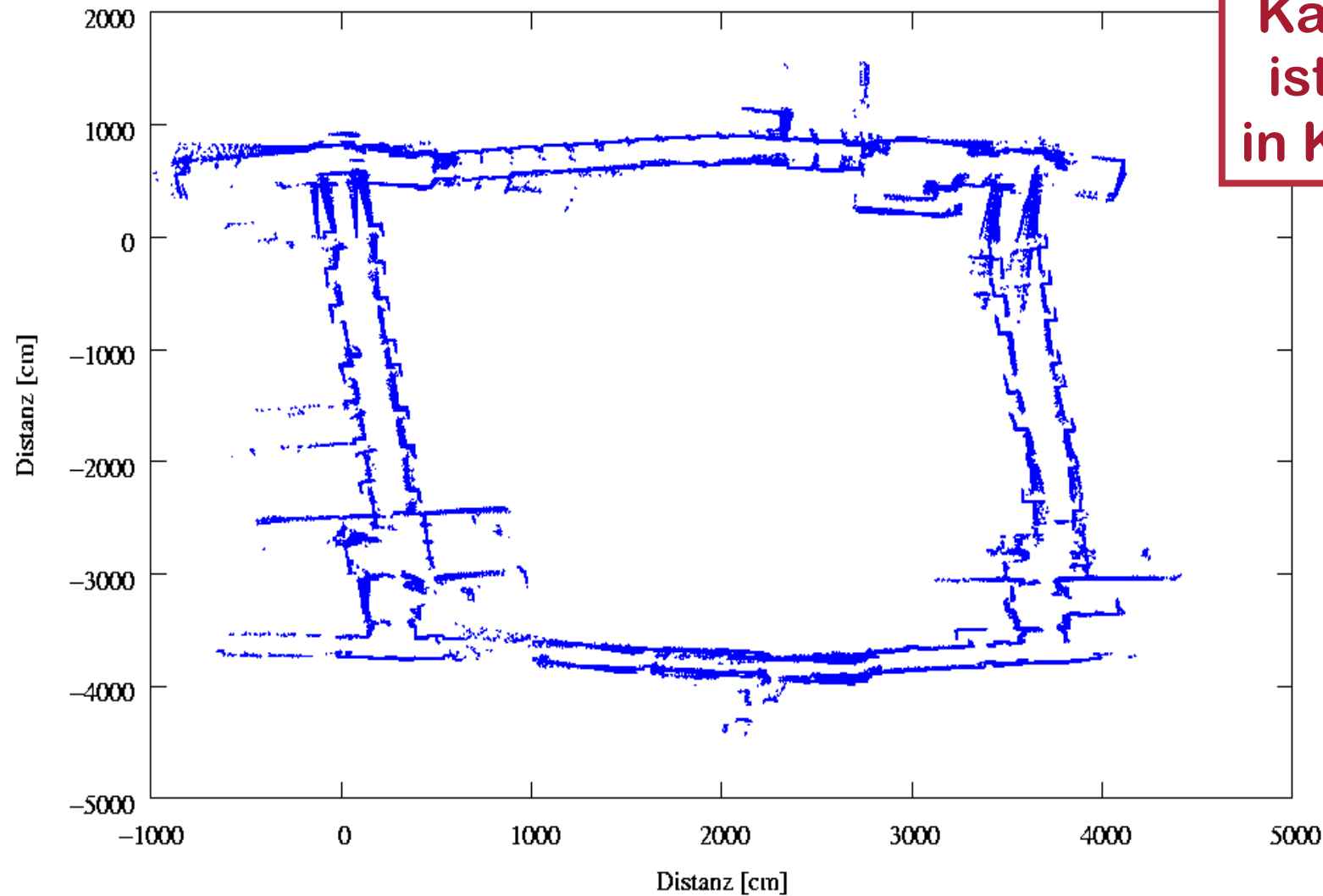
1. HAYAI allein
2. ... plus EKF
3. ... plus Karte



- Erhöhtes Rauschen durch Vibrationen
- Höhere Differenz zwischen Scans durch Fahrgeschwindigkeit

„Karte“ einer Rundfahrt

Kartenbau
ist Thema
in Kapitel 6!



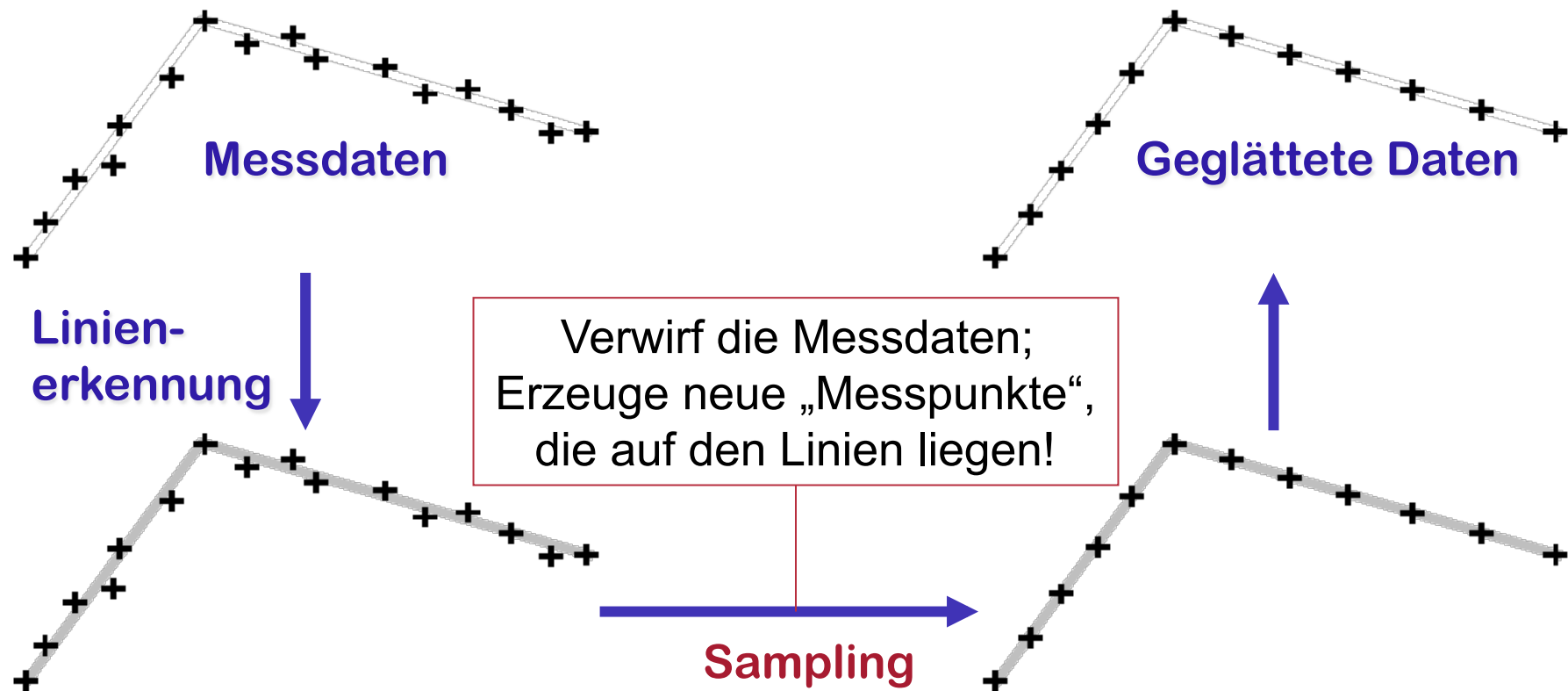
160 m,
12.727 Scans

5.3.2 Lokalisierung an Linien

- Statt Punktmengen mit Punktmengen matche Linienmengen mit Linienmengen zur Lokalisierung
 - Macht nur Sinn in überwiegend polygonalen Umgebungen!
 - Modell-Linien kommen aus Karte oder vorherigem/n Scan/s
 - Frage: Wie kommt man an Linien aus Laserscans?
- ➔ Methode, Gerade durch Punktmenge zu legen
s. Linien-Erkenner Folien 87–95!
(z.B. online-Linienfinder, Regressionsgeraden, Hough-Transformation)

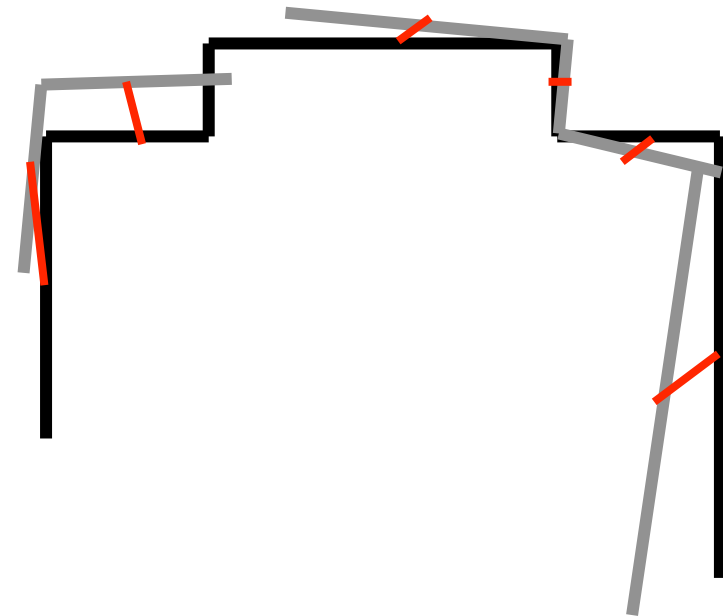
Linien zurück zu Punkten: Sampling (Folie 87)

Will man lieber Punkte als Linien matchen (ICP/IDC),
kann Linienfinden zum Glätten der Daten dienen!



Einzellinienmatching

- Beschreibe Linie durch Mittelpunkt, Länge, Neigung:
 $L = \langle c, l, \phi \rangle$
- Ordne Linien mit nächsten Mittelpunkten einander zu;
ggf. verwende Schwellen der Unterschiede von ϕ und l
- nimm z.B. IDC für korrespondierende
Mittelpunkte zum Matchen
aller Modelllinien mit allen
Datenlinien
- ggf. justiere Daten-Linien
einzeln nach (Werte c, ϕ)



**Für Polylinien:
s. Masterarbeit Rosemann!**