

Aufgabe Scan Matching

Einführung

Beim Scan-Matching sind zwei Punktwolken derselben Szene gegeben, die aber aus verschiedenen Positionen und Orientierungen aufgenommen wurden.

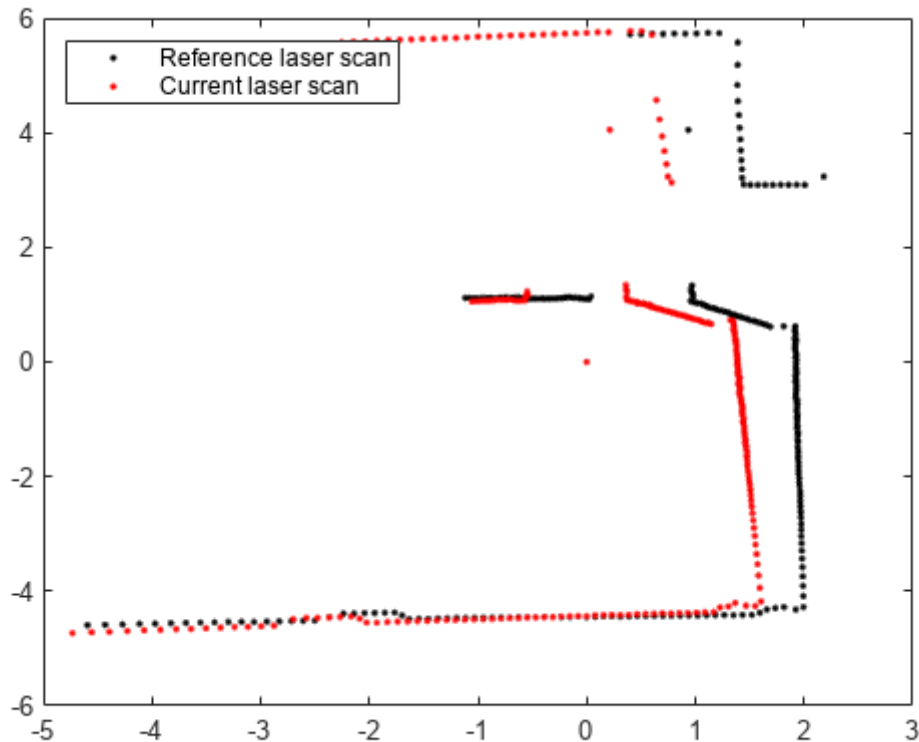


Abbildung 1: Scan Matching. Quelle: <https://de.mathworks.com/help/nav/ug/estimate-robot-pose-with-scan-matching.html>

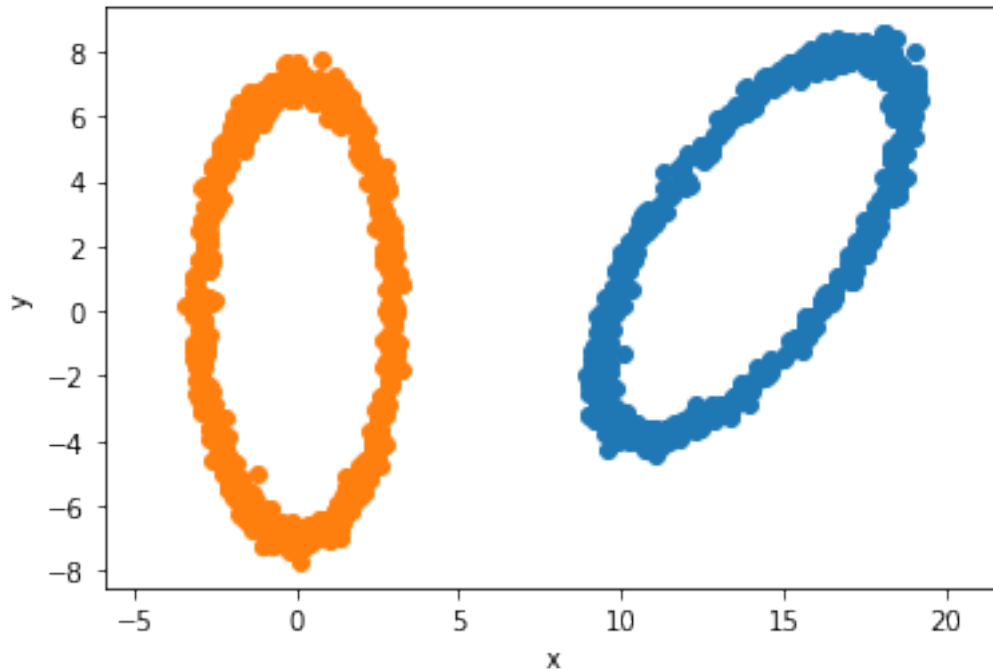
Gesucht ist die Transformationsmatrix, die die eine Punktwolke auf die andere abbildet (rotiert und translatiert). Eine Anwendung ist beispielsweise, wenn die erste Punktwolke eine Karte der Umgebung ist und die zweite aus den Sensoren eines Roboters gewonnen wurde. Kennt man die Transformation zwischen den Punktwolken, kennt man damit auch die Position und Orientierung des Roboters in Bezug zur Karte (Lokalisierung innerhalb einer Karte ist Grundvoraussetzung um eine Karte bspw. Zur Pfadplanung zu nutzen). Eine andere Anwendung ist, wenn die Punktwolken vom selben Sensor von zwei kurz aufeinander folgenden Zeitpunkten stammen. In diesem Fall erlaubt das Scan-Matching die Eigenbewegung des Sensors zwischen den Aufnahmezeitpunkten zu schätzen.

Ein populärer Algorithmus für Scan-Matching ist der Iterative-Closest-Point Algorithmus, über den Sie hier mehr lernen können:

<https://www.ipb.uni-bonn.de/html/teaching/msr2-2020/sse2-03-icp.pdf>

Aufgabe

In ScanMatching.txt finden Sie zwei Punktwolken:



Die orangene Punktwolke A ist in den ersten beiden Spalten der Textdatei abgelegt: X_a, Y_a . Die blaue Punktwolke B in den letzten beiden Spalten: X_b, Y_b .

Die Aufgabe des Scan-Matchings ist für diese Aufgabe insofern vereinfacht, als dass direkte Korrespondenzen zwischen den Punktwolken bestehen. Der n -te Punkt der Punktwolke A entspricht dem n -ten Punkt der Punktwolke B. Dies ist im Allgemeinen beim Scan-Matching nicht gegeben und Korrespondenzen müssen zuerst geschätzt werden.

Ihre Aufgabe ist es die Transformationsmatrix zu finden, die die Punktwolke B auf die Punktwolke A abbildet. Lesen Sie die Folien 1-14 hier:

<https://www.ipb.uni-bonn.de/html/teaching/msr2-2020/sse2-03-icp.pdf>

um zu lernen, wie dies möglich ist. Hinweis: Die Korrespondenzmenge C über die bspw. In Folie 11 summiert wird ist für diese Aufgabe $C = \{(0,0), (1,1), (2,2), \dots\}$. Die folgende Quelle könnte auch hilfreich sein:

https://www.ltu.se/cms_fs/1.51590!/svd-fitting.pdf

Im Algorithmus wird die Singulärwertzerlegung (Singular Value Decomposition) verwendet. Diese können Sie in numpy mittels 'np.linalg.svd' verwenden.