

Extrinsische Kamera zu Lidar Kalibrierung in Virtual Reality

Elias Birkefeld[†], Florian Wirth[†] und Christoph Stiller

KIT, Institut für Mess- und Regelungstechnik (MRT),
Engler-Bunte-Ring 21, 76131 Karlsruhe

Zusammenfassung In dieser Arbeit wurde eine Anwendung für die Bestimmung der extrinsischen Kalibrierungsparameter zwischen einer Kamera und einem Lidar entwickelt. Dies wird anhand einer Projektion des Kamerabildes auf eine Oberflächengeometrie, die aus der Lidar-Punktwolke generiert wurde, umgesetzt. Die Pose der Sensoren relativ zueinander wird manuell mittels VR-Technologie gesetzt. Anschließend werden die sechs Parameter der Kamerapose manuell approximiert, indem Featurepaare in Bild und Punktwolke gefunden und überlagert, gefundene Featurepaare zueinander fixiert und somit gezielt Freiheitsgrade der Kamerabewegung eliminiert werden. Eine erfahrene Testperson erreicht eine Standardabweichung von rund $\pm 0.11m$ und $\pm 0.34^\circ$ innerhalb weniger Minuten. Dementsprechend kann dieses Tool verwendet werden, wenn robuste Initialparameter für die maschinelle extrinsische Kalibrierung benötigt werden und nur eine statische Messung verfügbar ist.

Keywords Kalibrierung, Virtuelle Realität, Tooling, Kamera zu Lidar

1 Einleitung

Mit der Erschließung der Unterhaltungsbranche über das letzte Jahrzehnt haben verbrauchernahe Virtual Reality (VR) Systeme an Popularität zugenommen. Nicht nur Videospiele profitieren von der

[†] Gleichwertiger Beitrag.

verlustfreien Darstellung dreidimensionaler Szenen, auch bei technischen Anwendungen eröffnet VR neue Möglichkeiten Probleme einfacher, schneller und effizienter zu lösen. Insbesondere bei Problemen, die dreidimensionale Messungen mit hoher Informationsdichte umfassen wie beispielsweise die Kalibrierung einer Kamera zur dreidimensionalen Punktwolke eines Lidar¹-Sensors, kann VR als Eingabeschnittstelle bisher ungenutzte Potentiale fördern. Die Ausmessung der Relativanordnungen von Sensoren, welche als extrinsische Kalibrierung bezeichnet wird, ist von zentraler Bedeutung für die Sensordatenfusion. Die Vorteile der hier vorgestellten manuellen Methode gegenüber herkömmlicher Methoden werden im folgenden Szenario erörtert:

Im Jahre 2040 fahre ein lenkradloses vollautomatisches Taxi auf einer schlechtbeleuchteten Landstraße. Es kommt zu einem Wildunfall, bei der die vordere Stoßstange signifikant deformiert wird. Der an der Stoßstange montierte Lidar verdreht sich in allen Dimensionen. Ohne Neukalibrierung kann die Fahrt nicht fortgesetzt werden, da sonst eine sichere computergestützte Wahrnehmung nicht gewährleistet ist. Die Sensorik muss nun vorort hinreichend genau kalibriert werden.



Abbildung 1.1: In VR visualisierte dreidimensionale Darstellung.

Die im System eingebetteten automatischen Methoden können aus den Sensordaten nicht hinreichend viele visuelle Merkmale extra-

¹ Light detection and ranging, *kurz:* LiDAR oder Lidar, *deutsch:* Lichtdetektion und Abstandsmessung

hieren, um die erforderliche Kalibrierengenauigkeit zu erreichen. Die Sensordaten werden daher an ein Servicecenter des Fahrzeugherstellers geschickt, wo ein Servicemitarbeiter mithilfe des hier vorgestellten Tools schnell und routiniert eine initiale Kalibrierung erstellt und zurücksendet. Diese wird während der Weiterfahrt algorithmisch optimiert.

Die in dieser Arbeit vorgestellte Anwendung soll einen Einblick in die Anwendungsmöglichkeiten von Virtual Reality im Bereich Kalibrierung zeigen. In der Regel werden digitale Repräsentationen dreidimensionaler Objekte auf zweidimensionale Bildschirmflächen projiziert, woraus eine Mehrdeutigkeit resultiert. Ein modernes Virtual Reality System ermöglicht eine intuitive Darstellung dreidimensionaler Szenen, indem es diese Mehrdeutigkeit durch stereoskopische Darstellung umgeht. Diese Vorteile wurden bereits durch das Label Tool PointAtMe [1] gezeigt, das der manuellen Generierung von Objektannotationen in Punktwolken dient.

2 Stand der Wissenschaft

Kalibrierung von Sensoren ist ein altes Forschungsgebiet, das mit der Entwicklung neuer Sensoren und Messprinzipien aktuell bleibt. Bei der Kalibrierung eines Sensors werden zwei Teilbereiche unterschieden: Das intrinsische Sensormodell projiziert den realen Sensor auf ein mathematisches Modell; die extrinsische Kalibrierung besteht aus Translation und Rotation, die der Sensor relativ zu einem Bezugskoordinatensystem aufweist.

Extrinsische Parameter können basierend auf Kalibrierobjekten mit bekannter Geometrie oder ohne Kalibrierobjekte bestimmt werden. Für die objektbasierte Kalibrierung werden Objekte wie bspw. Schachbrettmuster [2], Kugeln [3] oder Kreise [4] im Sichtfeld aller zu kalibrierenden Sensoren platziert. Diese werden algorithmisch detektiert. Durch die Position und Orientierung des Objektes im Sichtfeld jedes einzelnen Sensors können Rückschlüsse auf die Ausrichtung der Sensoren zueinander gezogen werden. Dies setzt ein Kalibrierobjekt bekannter Geometrie und in der Regel eine Vielzahl von Aufnahmen voraus. Von Geiger *et al.* [2] werden intrinsische und extrinsische Parameter durch das automatische Detektieren mehre-

rer Schachbrettmuster im überlappenden Lidar- und Kamerasichtfeld bestimmt. Heng *et al.* [5] bestimmen intrinsische Kameraparameter per Schachbrettmuster, extrinsische Parameter werden durch Odometrie und Bewegungserkennung im jeweiligen Kamerasichtfeld bestimmt.

Im Falle einer unkontrollierten Umgebung wurden objektunabhängige Methoden entwickelt. Diese basieren beispielsweise auf der Extrahierung von primitiven optischen Merkmalen wie Kanten aus Aufnahmen der zu kalibrierenden Sensoren, welche durch Minimierung einer Kostenfunktion wie von Moghadam *et al.* [6] oder Kang *et al.* [7] vorgeschlagen überlagert werden. Diese Methoden funktionieren in strukturreicher Umgebung mit Beleuchtung, Gebäuden, Pfeilern, Pfosten und Straßenmarkierungen präzise und robust. Für kantenarme Regionen, wie sie auf Überlandstraßen durch Wälder und über Felder vorzufinden sind, büßen sie an Genauigkeit ein. Ansätze wie von Taylor *et al.* [8] sind auf ein problemspezifisches Umfeld angepasst und somit nicht universell einsetzbar. Manuelle Methoden wie von Scaramuzza *et al.* [9] funktionieren dagegen für beliebige Szenen zuverlässig.

Für die Projektion eines Bildes auf ein dreidimensionales Objekt wird eine Projektionsfläche benötigt. Es gibt bereits Verfahren, die sich mit der Oberflächengenerierung aus Punktwolken beschäftigen [10, 11]. Allerdings konzentrieren sich diese Verfahren auf die Generierung einer lückenfreien Oberfläche. In dieser Arbeit sollen Bereiche der Punktwolke mit wenigen Punkten aber explizit als Loch dargestellt werden, damit sich der Nutzer nicht an automatisch generierten Oberflächen orientiert, sondern nur an tatsächlich eingemessenen.

3 Ansatz

3.1 Oberflächengenerierung aus Punktwolken

Die Oberfläche soll aus so wenigen Polygonen wie möglich bestehen, um Echtzeitfähigkeit zu gewährleisten.

In einem ersten Schritt werden Punkte zusammengefasst, die nahe beieinander liegen. Hierfür werden räumliche Sektoren s_{ϕ_i, θ_j} für

$i \in [1, \dots, n_s]$ und $j \in [1, \dots, n_z]$ eingeführt, die Winkelabschnitten in Kugelkoordinaten ϕ, θ, r im Lidarsichtfeld entsprechen. n_s ist die gewählte horizontale Auflösung, n_z ist die Zeilen-/Diodenanzahl des Lidars. Alle Punkte, die nach Projektion auf diese Kugel im selben Sektoren liegen, werden als Punktgruppe zusammengefasst. Ist die größte Distanz eines Punktes in dieser Punktgruppe zu einem anderen kleiner als ein Schwellwert $d_{\text{Limit, PG}}$, werden die Koordinaten all dieser Punkte gemittelt und dem entsprechenden Sektoren zugeordnet.

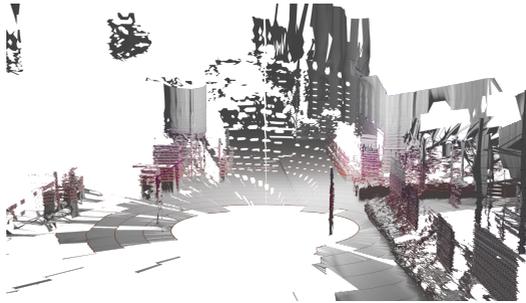


Abbildung 3.1: Generierte Oberfläche aus einer Punktwolke.

Ist die Distanz größer, wird die Punktgruppe unterteilt und die Schwellwertbedingung erneut überprüft. Für jede so unterteilte Punktgruppe werden die einzelnen Koordinaten der Untergruppen nicht gemittelt sondern die äußersten Punkte extrahiert, um Kanten in der Punktwolke zu erhalten. Idealerweise wird die Größe und die damit verbundene Anzahl der Sektoren so gewählt, dass pro Sektor maximal zwei Punktgruppen vorliegen.

Im zweiten Schritt sind Polygone aus den verbleibenden Punkten zu generieren. Es werden jeweils vier Sektoren $s_{\phi_i, \theta_j}, s_{\phi_i, \theta_{j+1}}, s_{\phi_{i+1}, \theta_j}, s_{\phi_{i+1}, \theta_{j+1}}$ überprüft. Im einfachsten Fall enthält jeder Sektor genau eine Punktgruppe. Dann werden zwei Dreiecke gebildet, wenn für den

Normalenvektor $n_{i,j}$ des jeweiligen Dreiecks und die Verbindungsgerade $l_{i,j}$ vom Lidar zur Punktgruppe des Sektors s_{ϕ_i,θ_j}

$$\arccos\left(\frac{s_{\phi_i,\theta_j} \cdot l_{i,j}}{|s_{\phi_i,\theta_j}| \cdot |l_{i,j}|}\right) \geq 90^\circ - \alpha_{\text{Limit}} \quad (3.1)$$

gilt, wobei $\alpha_{\text{Limit}} < 5^\circ$ gewählt wird. Falls in mindestens einem Sektor mehr als eine Punktgruppe vorhanden ist, wird jede mögliche Punktekombination in Betracht gezogen, die aus drei Punkten in unterschiedlichen Sektoren besteht. Jede aus diesen Kombinationen entstehenden Dreiecke, die die obige Bedingung erfüllen, werden generiert. Ein Beispiel für eine so generierte Oberfläche und die zugehörige Punktwolke ist in Abb. 3.1 zu sehen.

3.2 Projektion des Bildes auf die Oberfläche

Das Bild wird mithilfe der Projektionsmatrix des Lochkameramodells auf die generierte Oberfläche projiziert. Bisher wird das Lochkameramodell unterstützt. Die Projektion wird mithilfe eines Shaders durchgeführt, der parallelisiert und somit auch für große Datenmengen in Echtzeit auf einer Grafikkarte berechnet wird. Die Oberflächen ermöglichen eine gegenseitige Abschattung, wodurch verhindert wird, dass der gleiche Bildausschnitt auf mehreren Oberflächen abgebildet wird.

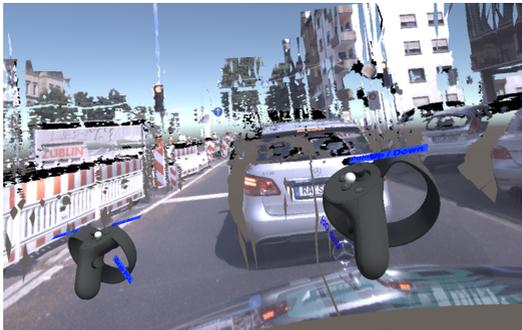


Abbildung 3.2: Auf die Oberfläche projiziertes Kamerabild.

3.3 Ankerpunkte

Das wichtigste der zahlreichen Hilfsmittel, die dem Nutzer zur Verfügung gestellt werden, sind Ankerpunkte, die Freiheitsgrade der Kamerapose beschränken und so den Kalibriervorgang beschleunigen. Es können bis zu zwei Ankerpunkte gesetzt werden. Diese Anker unterbinden eine Verschiebung des projizierten Bildes an dem Punkt, an dem sie in die Punktwolke gesetzt werden.

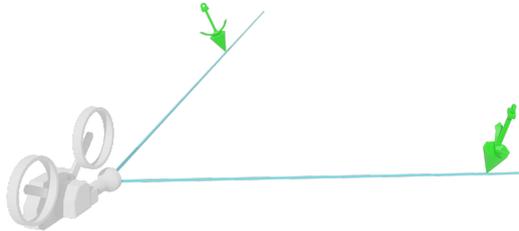


Abbildung 3.3: Wenn die grünen Anker gesetzt werden, verlaufen zwei Sichtstrahle der Kamera gezwungenermaßen durch die Anker. Dies schränkt die Freiheitsgrade der Kamerapose ein.

Wenn ein Anker gesetzt ist, werden zwei (Rotations-) Freiheitsgrade eliminiert. Wenn zwei Ankerpunkte gesetzt sind, können nur noch zwei (Translations-) Freiheitsgrade verändert werden. In beiden Fällen ist keiner der Parameter der Kamerapose konstant, wird aber in Abhängigkeit der verbleibenden Freiheitsgrade angepasst. Abb. 3.3 zeigt beispielhaft zwei Anker und die zugehörigen Sichtstrahle.

3.4 Eingabeschnittstelle

Es wird das VR Gerät Oculus Rift verwendet, das Touch Controller mit je 6 Freiheitsgraden und mehreren Tasten bereitstellt. Anhand der Controller können sowohl die virtuellen Sensoren, als auch die Anker gegriffen und neu positioniert werden.

4 Evaluation

4.1 Sensorsetup

Das Tool wurde anhand des Sensorsetups des teilautomatischen Versuchsfahrzeugs des MRT evaluiert. Die verwendete Kamera ist vom Typ FLIR BlackFly S 9 MPx mit einem Weitwinkelobjektiv mit rund 120° Öffnungswinkel. Das Rohbild wurde auf ein Lochkameramodell umgerechnet. Der verwendete Lidar ist vom Typ Velodyne VLS-128 Alpha Prime.

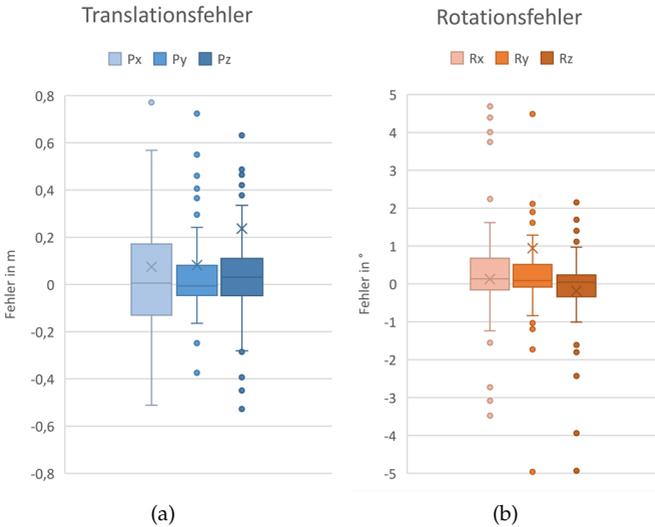


Abbildung 4.1: Ergebnisse aller Probanden über alle Szenarien.

4.2 Probandenexperimente und Randbedingungen

Die Testszenen stammen von 21 diversitären Verkehrsszenarien. Vier Probanden wurden gebeten auf allen 21 Szenarien die extrinsische Kalibrierung mithilfe des vorgestellten Tools zu erstellen. Die Probanden bekamen keine Rückmeldung bezüglich ihrer aktuellen Kalibrierengenauigkeit. Für alle Versuche wurde die gleiche maschinen-

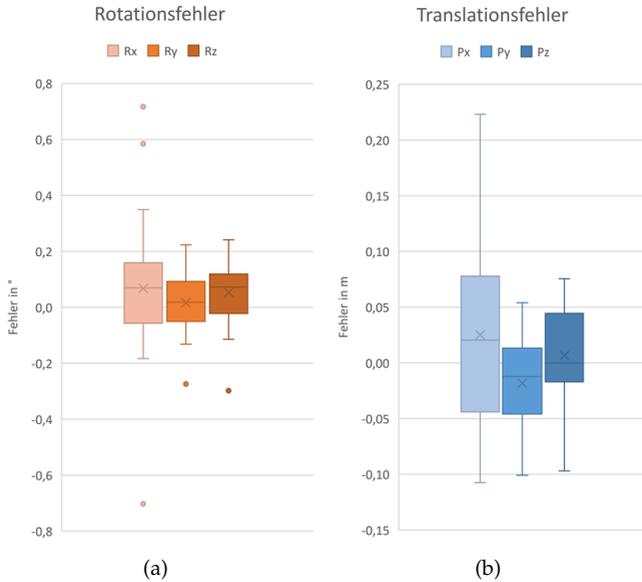


Abbildung 4.2: Ergebnisse des routiniertesten Probanen über alle Szenen.

erstellte Kalibrierung nach Strauss *et al.* [12] als Referenz verwendet. Aufgrund eines Rotationsfehlers um die Nick-Achse wurde die Rotation in dieser Komponente angepasst. Ob dieser Rotationsfehler aus der Umrechnung der Koordinatensysteme herrührt oder die wahre Kalibrierung tatsächlich eine Ungenauigkeit besitzt, konnte nicht geklärt werden. Nach dieser minimalen Anpassung kann man sich in VR davon überzeugen, dass Bild und Punktwolke augenscheinlich besser zueinander passen.

4.3 Ergebnisse

In Abb. 4.1 ist der Translationsfehlervektor aufgeteilt in x , y und z Anteile, sowie die Eulerwinkel um die x , y und z Achse aufgetragen. Hierbei entspricht die x -Achse der optische Achse, die y -Achse zeigt in Fahrtrichtung rechts und die z -Achse entsprechend nach oben.

Erwartungsgemäß findet sich der größte Fehler in Richtung der optischen Achse.

Als besonders klein stellen sich die Rotationsfehler heraus, die Translationsfehler sind vermutlich sogar größer als mit einfachsten Messinstrumenten erzielbar, womit Fehler unterhalb von 10 cm problemlos erreichbar sein sollten. Dennoch stellt das vorgestellte Tool für den eingangs geschilderten Anwendungsfall eine schnelle Fernkalibrierung als Servicemaßnahme eine geeignete, innovative Lösung dar, insbesondere aufgrund der geringen Rotationsfehler.

Des Weiteren sind die Ergebnisse in Abb. 4.2 zu beachten, die vom geübtesten der Probanden unter Zuhilfenahme der Anker erzeugt wurden. Hierbei lässt sich erkennen, dass durch den erfahrenen Umgang mit dem Tool deutliche Verbesserungen im Vergleich zu den Laienprobanden erzielt werden können.

5 Zusammenfassung

In der Arbeit wird ein manuelles Kalibrierwerkzeug vorgestellt, bei dem die Möglichkeiten von Virtual Reality als Hilfsmittel zur Lösung technischer Aufgabenstellungen am Anwendungsfall der extrinsischen Sensorkalibrierung erprobt werden soll. Aus einer Lidar-Punktwolke wird eine 3D Oberfläche in VR generiert, auf welches anhand des Lochkameramodells ein Kamerabild projiziert wird. Umfassende Tools zur leichteren Handhabung wurden integriert, mit welchen die Relativposition der Sensoren zueinander dem angepasst werden kann. Anhand einer Probandenstudie wurde die erzielbare Genauigkeit festgestellt. Demnach kann das Tool vor allem den Rotationsfehler stark verringern. Um eine schnelle, datengestützte Fernwartung zu bewerkstelligen, liefert das Tool gute Initialwerte, die danach bei Weiterfahrt algorithmisch verbessert werden können.

Literatur

1. F. Wirth, J. Quehl, J. Ota, and C. Stiller, "PointAtMe: Efficient 3D Point Cloud Labeling in Virtual Reality," in *2019 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2019, pp. 1693–1698.

2. A. Geiger, F. Moosmann, Ö. Car, and B. Schuster, "Automatic camera and range sensor calibration using a single shot," in *2012 IEEE International Conference on Robotics and Automation*. IEEE, 2012, pp. 3936–3943.
3. T. Kühner and J. Kümmerle, "Extrinsic Multi Sensor Calibration under Uncertainties," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 3921–3927.
4. H. Alismail, L. D. Baker, and B. Browning, "Automatic calibration of a range sensor and camera system," in *2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization & Transmission*. IEEE, 2012, pp. 286–292.
5. L. Heng, B. Li, and M. Pollefeys, "Camodocal: Automatic intrinsic and extrinsic calibration of a rig with multiple generic cameras and odometry," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 1793–1800.
6. P. Moghadam, M. Bosse, and R. Zlot, "Line-based extrinsic calibration of range and image sensors," in *2013 IEEE International Conference on Robotics and Automation*. IEEE, 2013, pp. 3685–3691.
7. J. Kang and N. L. Doh, "Automatic targetless camera–LIDAR calibration by aligning edge with Gaussian mixture model," *Journal of Field Robotics*, vol. 37, no. 1, pp. 158–179, 2020.
8. Z. Taylor and J. Nieto, "A mutual information approach to automatic calibration of camera and lidar in natural environments," in *Australian Conference on Robotics and Automation*, 2012, pp. 3–5.
9. D. Scaramuzza, A. Harati, and R. Siegwart, "Extrinsic self calibration of a camera and a 3d laser range finder from natural scenes," in *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2007, pp. 4164–4169.
10. N. Amenta, M. Bern, and M. Kamvysselis, "A new Voronoi-based surface reconstruction algorithm," in *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998, pp. 415–421.
11. M. Kolahdouzan and C. Shahabi, "Voronoi-based k nearest neighbor search for spatial network databases," in *Proceedings of the Thirtieth international conference on Very large data bases-Volume 30*, 2004, pp. 840–851.
12. T. Strauß, J. Ziegler, and J. Beck, "Calibrating multiple cameras with non-overlapping views using coded checkerboard targets," in *17th international IEEE conference on intelligent transportation systems (ITSC)*. IEEE, 2014, pp. 2623–2628.