

Erweiterung eines landmarkenbasierten Innenraumortungsverfahrens für Mobilgeräte mit ARCore

Jan Handrich und Ralf Vandenhouten*

Zusammenfassung

Die Lokalisierung in Innenräumen, die sich nicht auf Satellitennavigation wie GPS verlassen kann, ist nach wie vor eine Herausforderung. Viele Methoden zur Lokalisierung in Innenräumen beruhen auf Funksignalen oder anderen Sendern (z.B. Infrarot), die eine kostenintensive Infrastruktur erfordern, die in einem Gebäude installiert werden muss. Außerdem sind diese Techniken ungenau, wenn Reflexionen der Signale in einem Gebäude auftreten. Die Forschungsgruppe Telematik der TH Wildau hat für diese Problemstellung ein optisches Verfahren zur Innenraumlokalisierung entwickelt, das die Kamera eines mobilen Geräts verwendet. Dieser Ansatz erfordert nur die Verfügbarkeit von mehreren identifizierbaren Landmarken, die kostengünstig in einem Gebäude installiert werden können. In diesem Beitrag soll eine Möglichkeit vorgestellt werden, dieses landmarkenbasierte Verfahren mit der Augmented Reality-Bibliothek ARCore von Google zu erweitern, um eine kontinuierliche Positionsermittlung zu ermöglichen, auch wenn vorübergehend keine Landmarken durch die Gerätekamera erfasst werden können.

Abstract

Indoor localization, which cannot rely on satellite navigation such as GPS, remains a challenge. Many indoor localization methods depend on radio signals or other transmitters (e.g. infrared) that require a costly infrastructure to be installed in a building. In addition, these techniques are inaccurate when reflections of the signals occur in a building. The telematics research group at TH Wildau has developed an optical method for indoor localization for this problem that uses the camera of a mobile device. This approach only requires the availability of several identifiable landmarks that can be installed cost-effectively in a building. In this article, a way is presented to extend this landmark based method with Google's Augmented Reality library ARCore to enable continuous position tracking even if no landmarks can be captured by the device camera temporarily.

1. Einführung

1.1 Motivation

Ortung und Navigation sind seit vielen Jahrhunderten essentielle Bestandteile des Lebens und aus der Gesellschaft nicht mehr wegzudenken. Nutzte man früher die Sterne zum Navigieren, vertraut man heute primär auf weltumspannende Satellitennetze wie GPS oder Galileo für verschiedene Lokalisierungsanwendungen rund um den Globus. Durch die Beschleunigung von Prozessen bei gleichzeitiger Qualitätssteigerung sind sie als wesentlicher Teil von Wertschöpfungsketten verschiedenster Branchen unabdingbar.

Satellitengestützte Ortungssysteme ermöglichen weltweit eine metergenaue Ortung eines entsprechenden Empfängers, haben jedoch auch einen großen Nachteil: Im Innenraum oder an Orten, wo keine direkte Sichtverbindung zu den Satelliten besteht, können diese Systeme nicht eingesetzt werden. Die Signale der Satelliten können hier nicht auf direktem Wege empfangen werden, was dazu führt, dass keine zuverlässige Position ermittelt werden kann. Daher ist es nötig für den Innenraum auf alternative Ortungssysteme zurückzugreifen.

Nach wie vor gibt es heute aber noch kein Pendant zu GPS als allgemeingültige Lösung für alle Anwendungsberei-

che der Innenraumortung. So haben einige Ansätze Probleme, die hohen Anforderungen an die Präzision, Zuverlässigkeit oder Robustheit für spezielle Anwendungsfälle zu erfüllen, wie beispielsweise der zentimetergenauen Steuerung eines fahrerlosen Transportfahrzeuges, da keine ausreichende Ortungsgenauigkeit bereitgestellt werden kann (Lane et al. 2010, Kárník & Streit 2016). Andere Systeme bringen teilweise hohe Kosten für die nötige Infrastruktur mit sich.

Ein weiterer Lösungsansatz sieht die Nutzung von Bild- und Kameradaten vor, um Subjekte in einem entsprechenden Raum zu orten. Dabei können, je nach Verfahren, Subjekte durch

* korrespondierender Autor

fest installierte Kameras in einer definierten Umgebung erkannt und geortet werden oder mithilfe einer Kamera, die sie mit sich führen, ihre eigene Pose bestimmen.

Nach dem Beginn des Siegeszuges des Smartphones vor nicht einmal 10 Jahren sind sie heute zu einem Gegenstand geworden, der aus dem Alltag nicht mehr wegzudenken ist. Moderne Smartphones sind mit Kameras, Kommunikationstechnik (z. B. WLAN- und Mobilfunkempfänger) und vielerlei anderen Sensoren ausgestattet. Durch immer fortschreitende Prozessortechnik werden die Mobilgeräte immer leistungsfähiger und sind in der Lage aufwendige Berechnungen, wie sie für Bildverarbeitungsprozesse nötig sind, in Echtzeit durchzuführen. Dank ihrer umfangreichen Ausstattung bieten sie aber auch die ideale Plattform zur Erprobung und Nutzung alternativer Innenraum-Ortungsmethoden. Dieser Beitrag widmet sich daher dem Thema optischer Ortungsverfahren auf mobilen Endgeräten. Dabei soll dargestellt werden, inwiefern ein bestehendes optisches Ortungsverfahren durch zusätzliche Bildverarbeitungsverfahren sowie zusätzlichen Sensordaten erweitert und verbessert werden kann.

1.2 Codeloc

Die Forschungsgruppe Telematik der Technischen Hochschule Wildau um Prof. Dr. Vandenhouten hat ein markerbasiertes, optisches Ortungsverfahren entwickelt, das verspricht, ohne hohe Infrastrukturkosten, eine Ortungs- und Navigationslösung zu realisieren, die sowohl außerhalb als auch innerhalb von Gebäuden eine Genauigkeit im Zentimeterbereich liefern (Vandenhouten et al. 2014). Dieses als *Codeloc* bezeichnete innovative Verfahren nutzt eine Kamera um bekannte, eindeutige Markierungen (hier Landmarken genannt) zu erfassen und ermittelt durch die Abbildung dieser Landmarken im Kamerabild die Position im betrachteten Ortungsgebiet. Hierfür muss das Areal, in dem *Codeloc* verwendet werden soll, mit entsprechenden Landmarken ausgestattet werden. Die Form der Landmarken ist dabei relativ frei wählbar. So sind sowohl zweidimensionale (z. B. Vierecke oder Kreise) als auch

dreidimensionale Formen (z. B. Quader oder Kugeln) als Landmarken möglich. Die Landmarken sollten gut sichtbar im Areal positioniert und nicht verdeckt werden. Weiterhin kann durch ein farbliches Hervorheben der Landmarken von der Umgebung ihre Erkennung verbessert und beschleunigt werden. Essentiell für das Design der Landmarken ist, dass sie eindeutig voneinander unterschieden werden können. Hierfür bietet es sich an, eine optische Kodierung der Marken zu verwenden. So können u. a. Barcodes, QR-Codes oder Datamatrix-Codes für eine solche Identifizierung der Marken verwendet werden.

Zur Lokalisierung des Mobilgerätes wird dessen Kamera für die Aufnahme der Umgebung verwendet. Mittels Bildverarbeitungsalgorithmen werden die Landmarken im Kamerabild erkannt und anhand ihrer eindeutigen Markierungen identifiziert. Die Relativposition und Orientierung der Kamera werden darauffolgend anhand der Größe der sichtbaren Landmarken, ihrer Positionen im Bild sowie ihrer perspektivischen Verzeichnung ermittelt. Dies geschieht durch Schätzen der Kamerapose aus 3D-2D-Punktkorrespondenzen (Levenberg 1944, Marquardt 1963, Lepetit et al. 2009). Durch die Relativposition des Mobilgerätes zu den Landmarken sowie dem Wissen über die Position und Ausrichtung der Landmarken im Ortungsareal, kann die gesuchte absolute Position und Orientierung des Mobilgerätes errechnet werden. Mit der Positions- und Lagebestimmung des Mobilgerätes kann das *Codeloc*-Verfahren für das Bereitstellen von *Location-Based-Services* genutzt werden. Anhand der Absolutposition des Gerätes kann der Nutzer mit positionsbezüglichen Benachrichtigungen versorgt werden. Da mittels *Codeloc* aber auch die Pose des Mobilgerätes bestimmt wird, kann diese auch genutzt werden um Zusatzinformationen direkt im Kamerabild mittels *Augmented Reality* darzustellen.

Zur Validierung der Ortungsmethode wurde durch die Forschungsgruppe Telematik ein Versuchsaufbau mit drei Landmarken an festen, bekannten Positionen im Raum erstellt und Messungen an unterschiedlichen

Positionen im 3D-Raum getätigt. Die ermittelte Position der Software wurde mit der tatsächlichen, manuell ausgemessenen Position verglichen. Um die Genauigkeit der Methode zu quantifizieren wurden der durchschnittliche Messfehler zwischen der berechneten und tatsächlichen Position bestimmt. Zur Darstellung der Präzision wurde die Standardabweichung über alle Messungen ermittelt. Der Test zeigt, dass das Verfahren relativ stabile Werte für die Position ermittelt, mit einer Genauigkeit von 10–12 cm und einer Präzision von ungefähr 2,2 cm Abweichung.

Das *Codeloc*-Verfahren ist ein leistungsstarkes Verfahren zum Orten mobiler Endgeräte im Innenraumbereich. Verglichen zu anderen Ansätzen ist die Methode sehr präzise und robust gegenüber Veränderungen in der Umgebung. Ein großer Vorteil des Verfahrens besteht darin, dass, abgesehen von den kostengünstigen Landmarken, keine zusätzlichen Bauteile oder Komponenten benötigt werden und es auf jedem aktuellen Smartphone oder Tabletcomputer implementiert werden kann.

Im Rahmen des Forschungsprojektes *INNOMAP* (Vandenhouten 2018a) soll dieses Lokalisierungsverfahren nun verbessert und für die Bereitstellung von *Location-Based-Services* weiterentwickelt werden. Der Nachteil des *Codeloc*-Verfahrens besteht darin, dass eine Lokalisierung der Kamera bzw. des Endgerätes nur dann möglich ist, wenn mindestens eine Landmarke im Kamerabild erfasst und identifiziert werden kann. Daher gilt es *Codeloc* um zusätzliche Verfahren zu erweitern, um die Position des Gerätes in diesen Situationen mit hinreichender Genauigkeit kontinuierlich weiterverfolgen zu können.

In diesem Beitrag wird dabei dargestellt, wie *Codeloc* für Android-Geräte durch die neue Augmented Reality-Bibliothek *ARCore* von Google erweitert werden kann, um die Bestimmung der Position und Ausrichtung eines Smartphones auch dann zu ermöglichen, wenn keine optischen Marker erkannt werden können.

1.3 ARCore

Bei *ARCore* (Google Inc. 2018a) handelt es sich um eine Highlevel-API für Android Geräte, die es dem Anwender ermöglicht virtuelle 3D-Objekte in das Kamerabild des Smartphones zu integrieren. Um 3D-Objekte perspektivisch korrekt im Kamerabild einblenden zu können, wird das Koordinatensystem der Kamera mit realweltlichen Koordinaten überlagert und die relative Pose der Kamera kontinuierlich verfolgt. *ARCore* verwendet hierfür einen Prozess namens *Concurrent Odometry and Mapping* bzw. *COM* (Nerurkar et al. 2017), um zu verstehen, wie sich das Gerät, relativ zur Welt um es herum, bewegt. *ARCore* erkennt, ähnlich wie Feature-basierte *SLAM*-Verfahren, Feature-Punkte im aufgenommenen Kamerabild und verwendet diese zur Berechnung der Ortsveränderung. Die visuellen Informationen werden mit den Daten der inertialen Messeinheit (IMU), bestehend aus Beschleunigungs- und Lagesensorik, des Geräts kombiniert, um die Pose der Kamera relativ zur Welt über die Zeit abzuschätzen.

Zur Verfolgung der Gerätepose wird versucht, die Translations- und Rotations-schätzung der Kamera voneinander zu entkoppeln und sie stattdessen als eigenständige Optimierungsprobleme zu behandeln. Zunächst wird die 3D-Kamerabewegung ausschließlich aus dem visuellen Signal der Kamera ermittelt. Hierzu wird die 2D-Translation und der relative Maßstab einer Zielregion über mehrere Bilder hinweg beobachtet. Durch ein Lochkameramodell wird diese Translation sowie die Maßstabsänderungen in der Bildebene mit der 3D-Translation der Kamera in Verbindung gesetzt (Wei & Mullen 2018). Da das Kameramodell jedoch nicht von der Brennweite des Kameraobjektivs ausgeht, ist nicht bekannt, wie groß die tatsächliche Entfernung bzw. die Tiefe der nachgeführten Ebene ist. Um dies zu berücksichtigen, wird eine Skalenschätzung sowie eine Regionsverfolgung außerhalb des Sichtfeldes der Kamera eingesetzt. Anschließend wird die 3D-Orientierung des Geräts mit dem eingebauten Gyroskop bestimmt und die geschätzte 3D-Translation mit der 3D-Rotation kombiniert.

ARCore versucht weiterhin kontinuierlich sein Verständnis der Umgebung durch die Erkennung von Merkmalspunkten und Ebenen zu verbessern. Hierfür wird nach Gruppen von Feature-Punkten gesucht, die auf gemeinsamen horizontalen oder vertikalen Flächen, wie Tischen oder Wänden liegen. Die erkannten Features werden weiterhin in einer digitalen Karte gespeichert, um, ähnlich wie in einem *SLAM*-Verfahren, die Position in einer bereits bekannten Umgebung wieder zu finden (Sturm 2017).

Für die Erweiterung von *Codeloc* können die Funktionen von *ARCore*

Idee für die Kombination stellt daher eine Initialisierung der Position mittels *Codeloc* und ein anschließendes Verfolgen der Position durch *ARCore* dar. Da *ARCore* keine explizite Startpose vorgegeben werden kann, muss durch ein *Middleware*-Modul zunächst die Landmarkensuche und Posenbestimmung von *Codeloc* ausgeführt werden. Beim Erfassen einer stabilen Pose muss darauffolgend der Start von *ARCore* ausgelöst und die *ARCore*-Posen mit der *Codeloc*-Pose verrechnet werden.

Dies wird in Abb. 1 vereinfacht dargestellt:

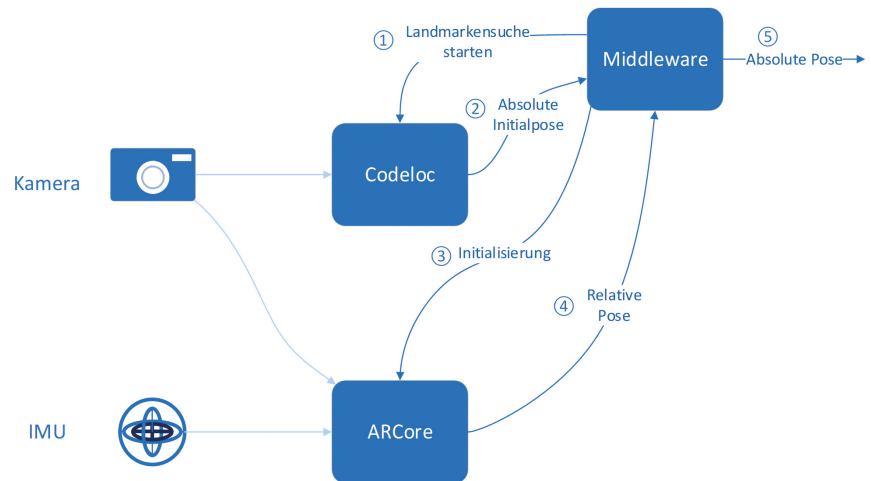


Abb. 1) Integration *Codeloc* und *ARCore*.

zur Darstellung virtueller Inhalte im Kamerabild eher als nebensächlich betrachtet werden, da hier die Bewegungsverfolgung benötigt wird, um die Kamerapose zu bestimmen. Für die Implementierung stellt *Google ARCore-SDKs* für Android, Android NDK, iOS, Unity sowie Unreal zur Verfügung.

2. Methodik

2.1 Integrationskonzept

Codeloc und *ARCore* stellen zunächst autark arbeitende Systeme dar. Während *Codeloc* in der Lage ist, die absolute Kamerapose im Bezugssystem zu bestimmen, wenn sich eine Landmarke im Kamerabild befindet, kann *ARCore* kontinuierlich Bewegungen verfolgen. *ARCore* ermittelt dabei immer die relative Position zum jeweiligen Initialisierungsort, der als der Ursprung (0, 0, 0) definiert wird. Die grundlegende

- Durch die *Middleware* wird *Codeloc* initialisiert und sucht kontinuierlich nach einer Landmarke im Kamerabild
- Bei einer erfassten Landmarke liefert *Codeloc* die absolute Kamerapose an die *Middleware*. Wird über einen definierten Zeitraum (bis 3 s) eine stabile Pose durch *Codeloc* ermittelt, wird diese Pose als Initialpose gesetzt und die Landmarkensuche in *Codeloc* gestoppt.
- *ARCore* wird initialisiert. Bis dies abgeschlossen ist, darf die Pose des Gerätes nicht verändert werden.
- *ARCore* liefert nun kontinuierlich die relativen Posen, ausgehend von der Initialpose.
- Die relativen Posen werden mit der Initialpose in der *Middleware*

verrechnet um die aktuelle Absolutpose zu bestimmen. Diese wird an die jeweilige Applikation weitergeleitet.

Der entscheidende Schritt in diesem Ablauf stellt dabei die Bestimmung der Initialpose dar. Um in Schritt 2 einzelne Messfehler von *CodeLoc* ausschließen zu können, wird die Pose über einen kurzen Zeitraum bestimmt. Verändert sich diese nur in einem geringen Maß, kann davon ausgegangen werden, dass das Smartphone augenblicklich nicht bewegt wird. Da *ARCore* in seiner aktuellen Version nicht die Möglichkeit bietet eine explizite Pose von außen zu setzen, kann dieses erst nach dem Bestimmen der Initialpose gestartet werden. Dabei muss sichergestellt werden, dass der Ursprung des *ARCore*-Koordinatensystems und die aktuelle Kameraausrichtung mit der ermittelten *CodeLoc*-Pose übereinstimmen, indem das Smartphone nicht bewegt wird. Der vollständige Initialisierungsprozess von *ARCore* dauert auf einem aktuellen Smartphone ca. 2–3 s. *ARCore* benötigt weiterhin während der Laufzeit den vollständigen Zugriff auf die Gerätekamera. Da das Android-System jedoch nur den Zugriff auf die Kamera durch einen Prozess ermöglicht, muss bereits vor dieser Initialisierung *CodeLoc* gestoppt werden.

Die Gleichausrichtung der Kamera und Position bei der Übergabe von *CodeLoc* zu *ARCore* ist wichtig, damit eine Verrechnung der absoluten Initialpose mit der relativen *ARCore*-Pose vorgenommen werden kann. Hierfür muss die Relativpose aus dem *ARCore*-Koordinatensystem in das *CodeLoc*-Koordinatensystem transformiert werden. Bezüglich der Skalierung der Positionen ist diese Transformation eher unkritisch zu betrachten, da beide Ortungsverfahren Translationswerte im metrischen System liefern. Entscheidend ist jedoch die Transformation der Raum-Achsen. Während bei *CodeLoc* die Ausrichtung der Koordinatenachsen vorhergehend beim Einmessen der Landmarken festgelegt werden, spannt *ARCore* bei jeder Initialisierung ein neues Koordinatensystem auf. Dieses ist dabei von der Lage des Gerätes abhängig.

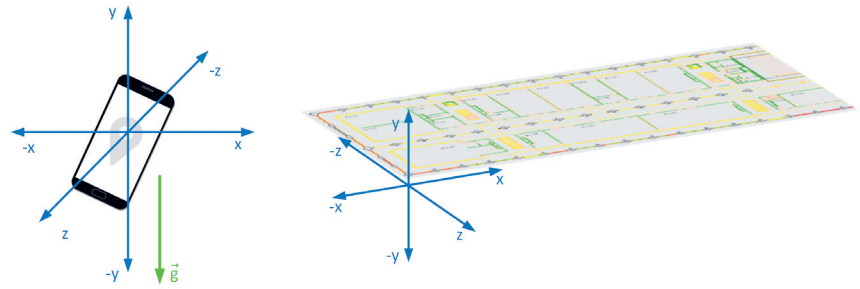


Abb. 2) Ausrichtung der Koordinatenachsen von *ARCore* und *CodeLoc*.

Wie in Abb. 2 dargestellt, wird unabhängig von der Lage des Smartphones die Ausrichtung der y-Achse des Koordinatensystems durch die Erdbeschleunigung g bestimmt. Diese wird durch die Beschleunigungssensoren des Gerätes ermittelt. Die x- und z-Achse werden jeweils orthogonal zu dieser als rechtshändiges Koordinatensystem definiert. Dabei zeigt die z-Achse in Richtung des Bildschirms des Gerätes. Die Ausrichtung des Koordinatensystems von *CodeLoc* kann beim Einmessen der Landmarken theoretisch frei gewählt werden. Vorgesehen ist hier jedoch, dass die y-Achse als Höhe verwendet wird, während die x- und z-Achse durch eine Karte oder den Grundriss des Gebäudes ausgerichtet werden. *CodeLoc* spannt dabei auch ein rechtshändiges Koordinatensystem auf. Somit kann per Definition davon ausgegangen werden, dass die y-Achsen beider Systeme gleich ausgerichtet sind. Um das *ARCore*-Koordinatensystem in das *CodeLoc*-Koordinatensystem projizieren zu können, muss dieses um seine y-Achse rotiert werden, bis die x- und z-Achsen beider Systeme gleich ausgerichtet sind. Nach dieser Rotation muss die relative Pose aus dem *ARCore*-System lediglich mit der vorhergehend bestimmten Initialpose verrechnet werden.

Um die x- und z-Achse gleich auszurichten, gilt es, den Winkel α in der x-z-Ebene zu ermitteln, mit dem um die y-Achse rotiert werden muss. Um diesen zu bestimmen, wird die Rotationsmatrix R_i der Initialpose verwendet. Zunächst wird der Einheitsvektor der z-Achse \vec{k} mit R_i multipliziert um die gedrehte z-Achse durch \vec{k}_r zu bestimmen. Dieser Vektor stellt gleichzeitig die aktuelle Blickrichtung der Kamera im *CodeLoc*-Koordinatensystem dar.

$$R_i \cdot \vec{k} = \begin{pmatrix} r_{i11} & r_{i12} & r_{i13} \\ r_{i21} & r_{i22} & r_{i23} \\ r_{i31} & r_{i32} & r_{i33} \end{pmatrix} \cdot \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \vec{k}_r = \begin{pmatrix} k_{rx} \\ k_{ry} \\ k_{rz} \end{pmatrix}$$

Anschließend wird der Vektor \vec{k}_r in die x-z-Ebene projiziert, indem seine y-Komponente gleich 0 gesetzt wird.

$$\vec{k}_r = \begin{pmatrix} k_{rx} \\ k_{ry} \\ k_{rz} \end{pmatrix} \rightarrow \vec{k}_r' = \begin{pmatrix} k_{rx} \\ 0 \\ k_{rz} \end{pmatrix}$$

Nun wird durch das Skalarprodukt der Vektoren \vec{k}_r' und \vec{k} der Drehwinkel α in der x-z-Ebene bestimmt.

$$\cos(\alpha) = \frac{\vec{k}_r' \cdot \vec{k}}{|\vec{k}_r'| \cdot |\vec{k}|}$$

Mit dem Winkel α kann zuletzt eine Drehmatrix R_y erstellt werden, welche für die Drehtransformation um die y-Achse eingesetzt werden kann.

$$R_y = \begin{pmatrix} \cos \alpha & 0 & \sin \alpha \\ 0 & 1 & 0 \\ -\sin \alpha & 0 & \cos \alpha \end{pmatrix}$$

Die durch *ARCore* ermittelte Pose mit der Rotationsmatrix R_{ar} und dem Translationsvektor \vec{t}_{ar} kann nun durch die Rotationsmatrix R_y und dem Translationsvektor \vec{t}_i der Initialpose in das *CodeLoc*-Koordinatensystem überführt werden.

$$R_{pose} = R_y \cdot R_{arcore}$$

$$\vec{t}_{pose} = \vec{t}_i + R_y \cdot \vec{t}_{ar}$$

Bei diesem Integrationskonzept kommt *CodeLoc* lediglich zur Initialisierung der Ortung zum Einsatz, um eine möglichst genaue Ausgangspose für die Relativortung von *ARCore* zu ermitteln. Während der Laufzeit von *ARCore*

könnte weiterhin nach Landmarken im Bild gesucht werden, um die Absolutpose durch *Codeloc* zu bestimmen. Jedoch kann während der Laufzeit von *ARCore* dessen interne Pose nicht durch äußeren Zugriff verändert werden. Eine Veränderung der Initialpose, die mit den Relativbewegungen verrechnet wird, wäre zwar denkbar, bei einer internen Korrektur des aktuellen Drifts durch ein *Loop-Closing* von *ARCore* würde dabei jedoch wiederum ein Positionsfehler entstehen.

2.2 Evaluation

Nach der Implementierung des Integrationskonzeptes in einer Android-Applikation gilt es, die Ortungsgenauigkeit und Präzision des Verfahrens zu bestimmen. Hierzu soll nach der Initialisierung mit *Codeloc* ein Smartphone an Positionen mit bekannten Koordinaten bewegt werden. Abb. 3 zeigt den schematischen Aufbau des Experiments.

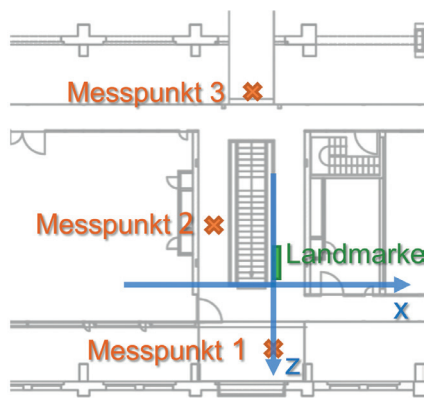


Abb. 3) Schematischer Aufbau des Experiments.

Anhand eines vorhandenen Grundrisses wird ein Koordinatensystem im Inneren eines Gebäudes angelegt. Dargestellt sind hier lediglich die Ausrichtungen der x- und z-Achsen. Die y-Achse bildet per Definition die Höhe ab. Zur Vereinfachung des Einmessvorganges, wird der Ursprung an eine, in der Nähe befindliche, Raumecke verschoben. Nahe dem Koordinatenursprung wird an einer Wand eine *Codeloc*-Landmarke angebracht und eingemessen (grüne Markierung). Ausgehend vom Koordinatenursprung werden drei Messpunkte auf dem Boden angebracht. Die Positionen der Landmarke und der Messpunkte sind dabei so gewählt, dass Messpunkt 1, je nach

Blickrichtung bei der Initialisierung, auf direktem Weg erreicht werden kann, während für Messpunkt 2 mindestens eine 90°-Kurve und für Messpunkt 3 mindestens eine 45°-Kurve zurückgelegt werden muss. Damit soll verhindert werden, dass *ARCore* lediglich geradlinige Bewegungen erkennen muss. Das Testgerät wird nach der Initialisierung durch Erfassen der *Codeloc*-Landmarke von Hand zu den einzelnen Messpunkten getragen. Da nicht gewährleistet werden kann, dass das Gerät beim Erfassen der Ergebnisse an den Messpunkten immer auf der gleichen Höhe gehalten wird, werden nur die x- und z-Koordinaten ausgewertet. Jeder Messpunkt wird bei diesem Experiment zehnmal erfasst. Nach jedem Erreichen eines Messpunktes und Aufnehmen der ermittelten Koordinaten, wird das Ortungsverfahren neu durch die Landmarke initialisiert. Die Initialisierung findet dabei von unterschiedlichen Startpositionen und Winkeln statt.

Um die Genauigkeit und Präzision der *Codeloc*-initialisierten *ARCore*-Ortung zu quantifizieren, werden zunächst für jede getätigte Messung die x-z-Abweichung zum Sollwert des jeweiligen Messpunktes als Messfehler bestimmt. Anhand dieser wird der durchschnittliche Messfehler als Maß für die Genauigkeit ermittelt. Zur Darstellung der Präzision wird die Standardabweichung der Messfehler berechnet, um die Streuung um den Mittelwert zu erhalten. Tab. 1 stellt die ermittelten Ergebnisse dar.

Die Ergebnisse des Experiments zeigen, dass das Verfahren für kurze Distanzen gute, reproduzierbare Werte liefert. Der

durchschnittliche Fehler zu Messpunkt 1 liegt bei 8 cm mit einer Abweichung von 3,5 cm. Die weiter entfernten und nicht direkt erreichbaren Messpunkte 2 und 3 sind mit einem höheren Messfehler von ca. 12 cm und 19 cm versehen, wobei sich die Präzision bei Messpunkt 2 stark verschlechtert. Anhand der schlechter werdenden Genauigkeit lässt sich gut der Drift der relativen *ARCore*-Ortung erkennen: Je größer die zurückgelegte Distanz zum Messpunkt, desto größer der ermittelte Messfehler.

Die Genauigkeit von 18,8 cm bei Messpunkt 3 zeigt, dass das Verfahren nach ca. 10 m zurückgelegtem Weg auf eine neue Initialpose gestützt werden sollte, damit der Drift nicht zu stark anwächst. Weiterhin beträgt die Präzision zu Messpunkt 2 lediglich 7,7 cm, womit die Messfehler doppelt so stark streuen, wie zu den anderen Messpunkten. Im Vergleich zu den anderen Messpunkten gab es bei Messpunkt 2 aufgrund des Sonnenstandes, sehr stark wechselnde Lichtverhältnisse. Daher muss während der Bewegungserfassung durch *ARCore* die Belichtungszeiten der Kamera stetig angepasst werden, was das Feature-Tracking entsprechend einschränkt. Im Vergleich zu reinen optischen Verfahren kann *ARCore* jedoch mit den Daten der IMU die Bewegungen weiterverfolgen, auch wenn diese stärker fehlerbehaftet sind.

Tab. 1) Genauigkeit und Präzision des Verfahrens.

	Durchschnittlicher Messfehler in cm	Standardabweichung der Messfehler in cm
Messpunkt 1	8,0	3,5
Messpunkt 2	12,4	7,7
Messpunkt 3	18,8	3,0
Gesamt	12,7	6,8

3. Zusammenfassung

Das *Codeloc*-initialisierte Verfahren stellt eine leistungsstarke Kombination von absoluter und relativer Ortung dar. Es ist in der Lage, die absolute Pose des Gerätes im Bezugssystem mit einer Frequenz von 30 Hz zu bestimmen und liefert dabei zunächst eine Genauigkeit im Dezimeterbereich, unabhängig von erkannten Landmarken.

Ein Problem stellt der größerwerdende Drift der *ARCore*-Ortung dar. Aufbauend muss daher ein Konzept bzw. Verfahren entwickelt werden, um die Ortung bei Sichtung einer Landmarke neu zu initialisieren, ohne dass die aktuelle Bewegung des Gerätes gestoppt werden muss. Nach aktuellem Entwicklungsstand müsste die *ARCore*-Sitzung hierfür gänzlich neu gestartet werden, was jedoch zu viel Zeit in Anspruch nimmt. Dabei sollte auch untersucht werden, wie die IMU-Daten bei dem Übergabe-Prozess von *Codeloc* zu *ARCore* eingebunden werden können, um Bewegungen vor der vollständigen Initialisierung von *ARCore* zu erfassen und dem Nutzer mitzuteilen, dass die Initialisierung ungültig ist und wiederholt werden sollte.

Das präzise Verfolgen der Kamerapose hat sich in den vergangenen Jahren zu einem Schwerpunkt in Wissenschaft und Forschung entwickelt. Gerade das Aufleben der Augmented Reality im Consumer-Bereich stellt einen starken Antrieb für dieses Thema dar. Das hier vorgestellte Verfahren ermöglicht es dabei, nicht nur die relativen Bewegungen des Gerätes zu ermitteln, sondern diese auch in einen absoluten Kontext für das Bezugssystem zu setzen, ohne dabei Leistung in Form der Ortungsfrequenz einzubüßen. Damit bietet es einen großen Mehrwert für die Bereitstellung von Location-Based-Services.

LITERATUR

- Google Inc. (2018a) *ARCore, Fundamental Concepts*. <https://developers.google.com/ar/discover/concepts>. Accessed 18 Jul 2018
- Google Inc. (2018b) *Github | arccore-android-sdk*. <https://github.com/google-ar/arccore-android-sdk>. Accessed 14 Sep 2018
- Handrich J (2018) *Entwicklung einer Softwarebibliothek für Mobilgeräte zur Innenraumortung mittels Bildverarbeitung und Bewegungssensorik*. Masterarbeit, Wildau
- Kárník J, Streit J (2016) *Summary of available indoor location techniques*. *IFAC-PapersOnLine* 49:311–317. doi:10.1016/j.ifacol.2016.12.055
- Lane N, Miluzzo E, Lu H, Peebles D, Choudhury T, Campbell A (2010) *A survey of mobile phone sensing*. *IEEE Commun. Mag.* 48:140–150. doi:10.1109/MCOM.2010.5560598
- Lepetit V, Moreno-Noguer F, Fua P (2009) *EPnP; An Accurate O(n) Solution to the PnP Problem*. *International Journal of Computer Vision* 81:155–166. doi:10.1007/s11263-008-0152-6
- Levenberg K (1944) *A method for the solution of certain non-linear problems in least squares*. *Quart. Appl. Math.* 2:164–168. doi:10.1090/qam/10666
- Marquardt DW (1963) *An Algorithm for Least-Squares Estimation of Nonlinear Parameters*. *Journal of the Society for Industrial and Applied Mathematics* 11:431–441. doi:10.1137/0111030
- Nerurkar E, Lynen S, Zhao S (2017) *System and method for concurrent odometry and mapping*. Patent US15595617
- OpenCV (2018) *OpenCV; Open Source Computer Vision Library*. <https://opencv.org/>. Accessed 14 Sep 2018
- Sturm J (2017) *The Computer Vision Technology Underlying ARCore*, München
- Vandenhouten R (2018a) *INNOMAP; Innenbereichs-Ortungssystem auf der Basis einer Bildanalyse markanter Punkte*. *Forschungsbericht 2017 2018:32*. doi:10.15771/FB_2018_1
- Vandenhouten R (2018b) *Orten eines mobilen Endgeräts unter Nutzung von optisch detektierbaren Landmarken*. Patent DE 10 2012 221 921
- Vandenhouten R, Kistel T, Wendlandt O (2014) *A method for optical indoor localization of mobile devices using multiple identifiable landmarks*. Paper, Wildau
- Wei J, Mullen T (2018) *The Instant Motion Tracking Behind Motion Stills AR*. <https://ai.googleblog.com/2018/02/the-instant-motion-tracking-behind.html>. Accessed 10 Aug 2018

AUTOREN

Jan Handrich, M. Eng.
Prof. Dr. rer. nat. Ralf Vandenhouten
Forschungsgruppe Telematik
Fachbereich Ingenieur- und Naturwissenschaften
Technische Hochschule Wildau

E-Mail für Korrespondenz:
ralf.vandenhouten@th-wildau.de

