

Beuth Hochschule für Technik Berlin

Masterarbeit

zur Erlangung des akademischen Grades
Master of Engineering

Thema: Graphoptimierung einer Trajektorie
anhand von Referenzpunkten

Autor: Okhan Tayfur
<okhan.tayfur@gmx.de>
MatNr. 814503

Version vom: 17. Juni 2018

1. Betreuer an der BHT: Prof. Dr. Roland Wagner
2. Betreuer am DLR: Dr. Sergey Zuev

Eidesstattliche Erklärung

Eidesstattliche Erklärung zur Masterarbeit

Ich versichere, die von mir vorgelegte Arbeit selbstständig verfasst zu haben. Alle Stellen, die wörtlich oder sinngemäß aus veröffentlichten oder nicht veröffentlichten Arbeiten anderer entnommen sind, habe ich als entnommen kenntlich gemacht. Sämtliche Quellen und Hilfsmittel, die ich für die Arbeit benutzt habe, sind angegeben. Die Arbeit hat mit gleichem Inhalt bzw. in wesentlichen Teilen noch keiner anderen Prüfungsbehörde vorgelegen.

Unterschrift :

Ort, Datum :

Zusammenfassung

Technologien zur Positionsbestimmung und Lokalisierung von beweglichen Objekten in Indoorbereichen oder in städtischen Bereichen, wo Satellitennavigation nicht möglich ist, nutzen entweder infrastrukturbasierte Sensordaten, wie WLAN, RFID oder infrastrukturunabhängige Sensorik, z.B. Kameras.

Das Institut für Optische Sensorsysteme am Deutschen Zentrum für Luft- und Raumfahrt e.V. (DLR-OS) hat in den letzten Jahren das Integrated Positioning System (IPS) entwickelt. IPS ist ein System zur Eigenverortung auf Basis einer Stereokamera und einer inertialen Messeinheit (IMU). Dieses System liefert in Echtzeit Koordinaten in einem relativen, ortsfesten Koordinatensystem. Über die Laufzeit entsteht aus den Einzelmessungen eine relative Trajektorie. Die Genauigkeit der relativen Trajektorie hängt maßgeblich von der Qualität der eingebauten Sensorik, z.B. Drift und Rauschen der IMU und der Datenverarbeitung, z.B. Feature-Detektion bei der visuellen Odometrie, ab. Die Akkumulation der Fehler führt dazu, dass die Qualität der Eigenverortung mit der Laufzeit nachlässt. Eine Verbesserung der Genauigkeit der Trajektorie kann durch Einbeziehung zusätzlicher Informationen, z.B. Referenzpunkte, erreicht werden. Dabei wird die Trajektorie durch geeignete Transformationsverfahren von relativen Koordinaten in ein übergeordnetes, absolutes Koordinatensystem übertragen und anschließend optimiert.

In der vorliegenden Masterarbeit wird ein Verfahren zur Verbesserung von IPS-Trajektorien auf Basis von Referenzpunkten und Graphenoptimierung implementiert und validiert. Ausgehend von der Tatsache, dass Starrkörpertransformationen, z.B. Helmert-Transformation, die akkumulierten Fehler relativer IPS-Trajektorien nicht ausgleichen können, bedarf es eines flexiblen Verfahrens, das auch nichtlineare Fehler berücksichtigt. Aus diesem Grund wird das g^2o -Framework verwendet, bei dem u.a. auch graphbasierte Optimierungsverfahren implementiert sind. Dazu wird in dieser Arbeit ein Konzept entwickelt und umgesetzt, das auf dem bisherigen Stand der IPS-Softwareentwicklung aufbaut und das IPS-Softwaresystem um das graphbasierte Optimierungsverfahren erweitert. Als Ergebnis erhält man eine anhand von Referenzpunkten optimierte IPS-Trajektorie in absoluten Koordinaten. Die Ergebnisse der Tests und Validierung zeigen, dass das umgesetzte Konzept zu einer maßgeblichen Verbesserung der verwendeten IPS-Trajektorie führt.

Inhaltsverzeichnis

Abbildungsverzeichnis	6
Tabellenverzeichnis	7
Abkürzungsverzeichnis	8
1 Einleitung	9
2 Grundlagen	11
2.1 Integrated Positioning System	11
2.2 Starrkörpertransformation von Koordinaten	13
2.2.1 Räumliche Helmert-Transformation	13
2.2.2 Bestimmung der 7 Parameter der Helmert-Transformation	14
2.3 Graphentheorie	17
2.4 Graphoptimierung	19
3 Softwaremethoden	23
4 Konzept der graphbasierten Optimierung einer absoluten Trajektorie anhand von Ground Control Points	24
5 Implementierung der graphbasierten Optimierung von IPS-Trajektorien	27
6 Validierung	28
7 Ergebnisse	31
8 Zusammenfassung	41
9 Ausblick	41
Literaturverzeichnis	42

Abbildungsverzeichnis

1.1	Das Integrated Positioning System-Mini als tragbare Version . . .	10
2.1	IPS-Sensorkopf [1]	11
2.2	IPS-Prozesskette[1]	12
2.3	Eine relative Indoor-IPS-Trajektorie, die auf einen Gebäudegrundriss projiziert ist [1]	12
2.4	Räumliche Koordinatentransformation nach Helmert, Koordinate \vec{q} wird vom $x'y'z'$ -Koordinatensystem in die Koordinate \vec{p} des xyz-Koordinatensystem umgerechnet	13
2.5	ungerichteter Graph mit und ohne Mehrfachkante	18
2.6	Bespie einerl Adjazenzmatrix und Inzidenzmatrix	18
2.7	Beispiel eines gerichteten Graphen mit gewichteten Kanten	19
2.8	Darstellung einer Trajektorie als Graph	20
4.1	Prozesskette der graphbasierten Optimierung einer IPS-Trajektorie	25
5.1	Überblick verwendeter Softwarebibliotheken zur Implementierung des Konzepts zur graphbasierten Optimierung von IPS-Trajektorien	27
6.1	Übersicht der Ground Control Points	28
6.2	Ground Control Points (GCPs)-Vermessung mittels RTK-GPS-Verfahren	30
7.1	2D-Horizontaldarstellung der Helmert-transformierten (rot) - und graphoptimierten Trajektorie (grün) und GCPs (Blau)	32
7.2	2D-Seitenansicht der Helmert-transformierten (rot) - und graphoptimierten Trajektorie (grün) und GCPs (Blau)	33
7.3	Absolute Abweichung der Helmert-transformierten Trajektorie an den GCPs	34
7.4	Absolute Abweichung der graphoptimierten Trajektorie an den GCPs	35
7.5	Absolute Abweichung der graphoptimierten Trajektorie an den GCPs mit den Kontrollpunkten)	36
7.6	Absolute Abweichung der Helmert-transformierten zur graphoptimierten Trajektorie mit GCPs zum Zeitpunkt t auf der Wegstecke	37
7.7	Darstellung der Helmert-transformierten und graphoptimierten Trajektorie in GoogleEarth	38
7.8	Darstellung der Helmert-transformierten und graphoptimierten Trajektorie in GoogleEarth mit 3D-Gebäude-Projektion	39

Tabellenverzeichnis

1	Dateistruktur der Helmert-transformierten - und graphoptimierten Trajektorie	26
2	Dateistruktur der GCP-Datei	26
3	GCP Abfolge und ID	29
4	Systemspezifikation des Testrechners	30
5	Rechenaufwand der Datenverarbeitung für die Helmert-transformierte - und graphoptimierte Trajektorie	40

IPS	Integrated Positioning System	4
GCP	Ground Control Point	6
GPS	Global Positioning System	9
DLR	Deutsches Zentrum für Luft- und Raumfahrt e.V.	28
DLR-OS	Institut für Optische Sensorsysteme am Deutschen Zentrum für Luft- und Raumfahrt e.V.	4
WGS84	World Geodetic System 1984 (Definition der Erde als Ellipsoiden-Modell seit 1984)	12
IMU	Inertial Measurement Unit	9
ECEF	Earth-Centered-Earth-Fixed	26
I/O	Input/Output	27
g²o	General (hyper) Graph Optimization	23
RFID	Radio-Frequency Identification	
WLAN	Wireless Local Area Network	
GNSS	Global Navigation Satellite System	41
NFC	Near Field Communication	

1 Einleitung

Navigation ist seit jeher ein bestimmendes Thema der Menschheit und entwickelte sich über die Jahrtausende. Navigation besteht aus der Bestimmung der eigenen Position im Raum sowie der Richtung, in die man sich bewegen muss, um sein Ziel zu erreichen. Die eigene Position im Raum wird heutzutage beispielsweise mittels des Global Positioning System (GPS), welches eine im Idealfall auf bis zu wenige Zentimeter genaue Eigenverortung ermöglicht, erreicht. Die Verfügbarkeit von GPS ist jedoch auf den Außenbereich beschränkt und die Genauigkeit von der Signalqualität abhängig. In urbanen Bereichen, in denen die Signale umgebungsbedingt, vor allem durch eingeschränkte Satellitensichtbarkeit und Multipath an Häusern, gestört oder gar nicht empfangbar sind, wird eine Positionsbestimmung sehr ungenau oder unmöglich. In diesen Bereichen bedarf es für zukünftige Anwendungen, z.B. autonomes Fahren, seamless Navigation (Verknüpfung von Indoor- und Outdoor-Navigation), alternativer Verortungskonzepte.

Das Institut für Optische Sensorsysteme am Deutschen Zentrum für Luft- und Raumfahrt e.V. beschäftigt sich mit der Entwicklung eines technischen Systems zur Eigenverortung, das den menschlichen Orientierungssinn nachbildet und für die Verortung im Innen- wie auch Außenbereich ohne Verwendung externer Ortungsinfrastruktur einsetzbar ist. Es handelt sich um das Integrated Positioning System, s. Abb. 1.1, welches mittels einer Stereokamera und einem Inertialsensor (Inertial Measurement Unit (IMU)), seine Position und Lage im Raum bestimmt. [2]. Über die Laufzeit entsteht anhand der Einzelmessungen eine Trajektorie in relativen Koordinaten.



Abb. 1.1: Das Integrated Positioning System-Mini als tragbare Version

Die relative IPS-Trajektorie akkumuliert die Fehler, die z.B. durch Sensor-drift, Unsicherheiten und Fehler bei der Datenverarbeitung, nicht perfekte Sensor-Kalibrierung und Registrierung der Sensoren zu einander (Stereokameras zu IMU) oder auch durch nicht exakte zeitliche Synchronisation der Messungen zurückzuführen sind. Eine Transformation der Trajektorie von lokalen in ein übergeordnetes, absolutes Koordinatensystem kann mit Hilfe von GCPs mit bekannten Koordinaten erfolgen.

Die Aufgabe ist eine Verbesserung der absoluten Trajektorie mittels eines graph-basierten Optimierungsverfahren und zu Validieren, unter Verwendung der DLR-OS Software-Bibliothek. In den folgenden Abschnitten werden Grundlagen und Softwaremethoden zur Entwicklung eines Konzeptes aufgeführt. Das Entwickelte Konzept wird unter Verwendung eines IPS-Trajektorien-Datensatzes validiert

2 Grundlagen

In diesem Kapitel werden die für die Masterarbeit relevanten Grundlagen vorgestellt. Dabei wird ein kurzer Überblick zu IPS (Kapitel 2.1) gegeben. Im Anschluss wird die Koordinatentransformation nach Helmert (Kapitel 2.2.1) beschrieben, die zur Transformation der Trajektorie des IPS vom relativen in ein absolutes Koordinatensystem genutzt wird sowie die Bestimmung der Transformationsparameter (Kapitel 2.2.2). Kapitel 2.3 gibt eine kurze Einführung in die Graphentheorie. Im Kapitel 2.4 beschreibt das Verfahren zur Lösung von nichtlinearen Optimierungsproblemen und ihre Anwendung im graphbasiertem Optimierungsverfahren.

2.1 Integrated Positioning System

IPS wurde am Institut für Optische Sensorsysteme am Deutschen Zentrum für Luft- und Raumfahrt DLR-OS entwickelt.



Abb. 2.1: IPS-Sensorkopf [1]

IPS besteht aus Hard- und Softwarekomponenten. Die Hardware besteht aus einer IMU, einem Tilt-Sensor, einer Stereokamera, einem Rechner, weiterer elektronischer Komponenten sowie einer Beleuchtungseinheit. Die genannten Komponenten bilden zusammen den Sensorkopf (siehe Abb. 2.1). Der Sensorkopf liefert Rohdaten, die durch verschiedene Softwareinstanzen verarbeitet werden. Die Prozesskette ist in Abb. 2.2 dargestellt. Die Prozesskette besteht aus mehreren umfangreichen Bibliotheken zur Synchronisation der Rohdaten, zur Datenverarbeitung und Hardwaresteuerung.

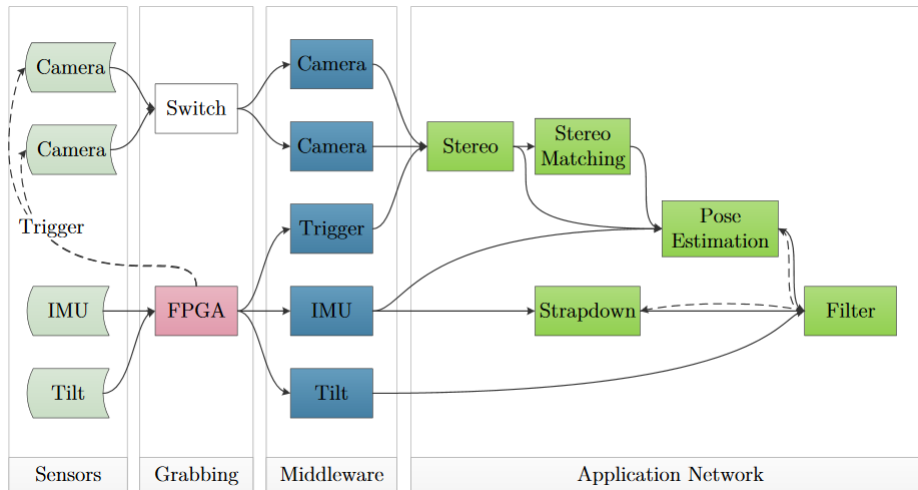


Abb. 2.2: IPS-Prozesskette[1]

IPS ermittelt seine eigene Position und Lage in relativem Koordinatensystem. Dabei werden alle Daten aus sämtlichen Sensoren in einem fehlertoleranten Filter (erweitertes Kalman-Filter) fusioniert. Während die IMU die Lage in Bezug auf die Gravitationsachse misst, wird das Stereo-Kamera-System zur optischen Odometrie genutzt.

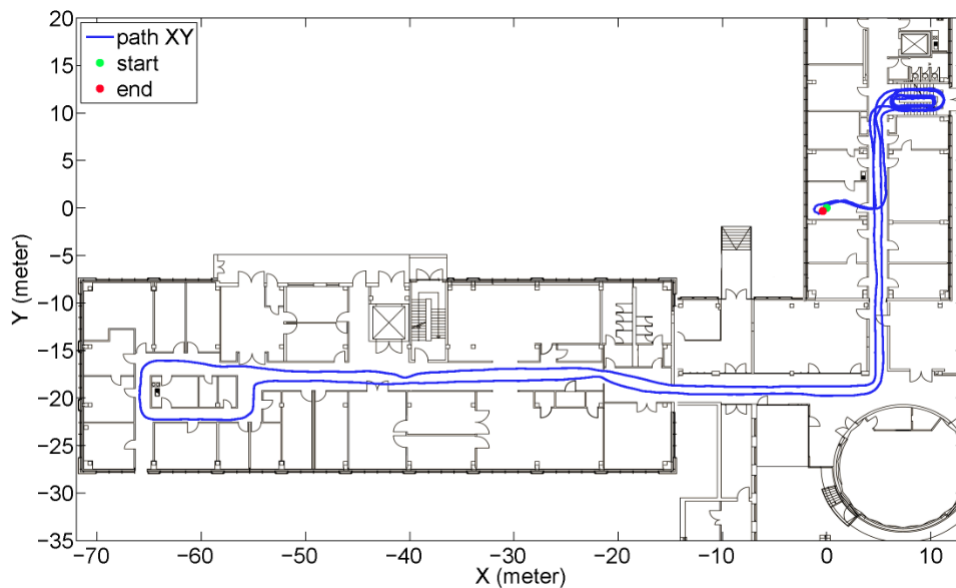


Abb. 2.3: Eine relative Indoor-IPS-Trajektorie, die auf einen Gebäudegrundriss projiziert ist [1]

Um die Trajektorie vom relativen Koordinatensystem in ein absolutes Koordinatensystem zu übertragen, wurde DLR-OS die Methode nach Horn (siehe 2.2.1) implementiert. Damit werden die relativen Koordinaten der IPS-Trajektorie mit Hilfe von Ground Control Points in das globale Koordinatensystem World Geodetic System 1984 (Definition der Erde als Ellipsoiden-Modell seit 1984) (WGS84) umgerechnet.

2.2 Starrkörpertransformation von Koordinaten

Dieses Unterkapitel greift das Thema der Starrkörper-Koordinatentransformation auf. Es wird die räumliche Transformation nach Helmert erklärt und eine direkte Methode [3] zur Bestimmung der Transformationsparameter beschrieben.

2.2.1 Räumliche Helmert-Transformation

Die räumliche Helmert-Transformation ist eine Koordinatentransformation von dreidimensionalen kartesischen Koordinaten in ein beliebiges anderes dreidimensionales kartesisches System. Sie findet häufig Anwendung in der Geodäsie zur verzerrungsfreien Umrechnung zwischen verschiedenen landesspezifischen Koordinatensystemen.

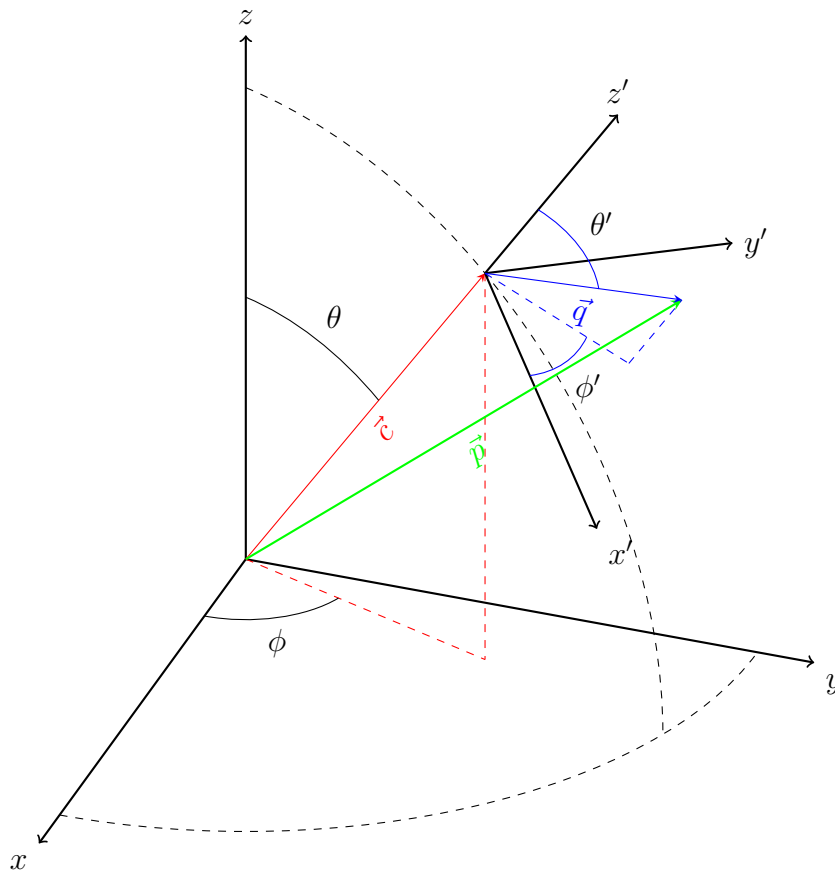


Abb. 2.4: Räumliche Koordinatentransformation nach Helmert, Koordinate \vec{q} wird vom $x'y'z'$ -Koordinatensystem in die Koordinate \vec{p} des xyz -Koordinatensystem umgerechnet

Die Helmert-Transformation[4, S.41-45] wird wie folgt definiert:

$$\vec{p} = \vec{c} + \mu * R * \vec{q} \quad (2.2.1.1)$$

$$\begin{array}{ll}
\vec{p} \dots \textit{transformierter Vektor} & \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix}_{3 \times 1} \\
\vec{q} \dots \textit{Ausgangsvektor} & \begin{pmatrix} q_x \\ q_y \\ q_z \end{pmatrix}_{3 \times 1} \\
\vec{c} \dots \textit{Verschiebungsvektor} & \begin{pmatrix} c_x \\ c_y \\ c_z \end{pmatrix}_{3 \times 1} \\
\mu \dots \textit{Maßstabsfaktor (Skalar)} & \\
R \dots \textit{Drehmatrix} & R(\phi, \theta, \psi)_{3 \times 3}
\end{array}$$

Die Helmert-Transformation [5, S. 1-11] ist auch unter dem Namen 7-Parameter-Transformation bekannt. Sie setzt sich zum einen aus den drei Rotations-Parametern (ϕ , θ und ψ) aus der Drehmatrix R für die Rotationslage des einen Koordinatensystems im Bezug zum Zielkoordinatensystem und zum anderen aus drei Translationen (c_x , c_y und c_z) auf der jeweiligen X-, Y- und Z-Achse. Letztendlich wird noch der Maßstabsfaktor/Scaling (μ) ermittelt.

Zur Berechnung der sieben Transformationsparametern werden sieben Gleichungen benötigt. Dazu werden mindestens zwei identische Raumkoordinaten aus jedem System und eine weitere Koordinaten-Komponente benötigt. Damit können die sieben Gleichung zur Parameterbestimmung bereits aufgestellt werden. In der Praxis werden mehr als drei Punkte bei der Parameterberechnung verwendet. Dies bewirkt, dass Fehler, die bei der Messung der Koordinaten durch Abrundung entstehen, über die gesamte Anzahl der Koordinatenpunkte gleichmäßig verteilt werden. Die dabei entstehende Überbestimmung wird mittels einer Ausgleichsrechnung der kleinsten quadratischen Fehler gelöst.

Mit gegebenen Transformationsparametern der Rotationen und der Translation lassen sich so Koordinaten vom Start- in das Zielsystem umrechnen.

2.2.2 Bestimmung der 7 Parameter der Helmert-Transformation

Da man in der Regel eine Vielzahl von Koordinatensätzen in ein anderes Koordinatensystem übertragen will, müssen hierfür die Transformationsparameter mittels Passpunkten (identische Koordinaten) ermittelt werden. Hierbei kann man iterativ durch Umstellung nach den sieben gesuchten Transformationsparametern ($\vec{c}_{x,y,z}, R_{\phi,\theta,\psi}, \mu$) der Formel (2.2.1.1) und Ableitung vorgehen. Meist sind die Mess-

werte der Koordinaten fehlerbehaftet. So ergibt sich eine Restabweichung (Residuen) \vec{e}_i :

$$\vec{e}_i = \vec{p}_i - R\vec{q}_i - \vec{c}$$

Wobei es die quadratische Fehler zu minimieren zu gilt:

$$\sum_{i=1}^n \|\vec{e}_i\|^2$$

Eine direktere Methode (*Closed-form solution of absolute orientation using unit quaternions* [3]) wurde durch Berthold K. P. Horn vorgestellt, bei der die bestmöglichen Parameter ermittelt werden.

Zur Bestimmung der Startwerte der 7 Parameter-Transformation geht man nach Horn wie folgt vor:

- a) Bestimmung der Schwerpunktkoordinaten für Start- und Zielsystem:

$$\vec{p} = \frac{1}{n} \sum_{i=1}^n \vec{p}_i \quad \vec{q} = \frac{1}{n} \sum_{i=1}^n \vec{q}_i$$

- b) Berechnung der Schwerpunktreduziertenkoordinaten:

$$\vec{p}_i = \vec{p}_i - \vec{p} \quad \vec{q}_i = \vec{q}_i - \vec{q}$$

- c) Berechnung der Translationskonstanten:

Durch die ermittelten Schwerpunktkoordinaten lässt sich die Fehlergleichungen mit den reduzierten Schwerpunktkoordinaten umschreiben :

$$e_i = \vec{p}_i - \mu R \vec{q}_i - \vec{c}$$

wobei

$$\vec{c} = \vec{c} - \vec{p}_i + \mu R \vec{q}_i$$

gilt. Horn zeigt hierbei, dass die Translation c , mit $\dot{c} = 0$, als Differenz der Schwerpunktkoordinaten \vec{p} bzw. \vec{q} aufgefasst werden kann. So lässt sich die Translation direkt berechnen, sobald die optimale Rotation ermittelt ist:

$$\vec{c} = \vec{p}_i - \mu R \vec{q}_i$$

- d) Bestimmung des Skalierungsfaktors:

$$\mu = \sqrt{\frac{\sum_{i=1}^n \|\vec{p}_i\|^2}{\sum_{i=1}^n \|\vec{q}_i\|^2}}$$

e) Bestimmung der Rotationsparameter:

Die Darstellung der Rotationsparameter durch Quaternionen ist bei einigen Anwendungen vorteilhaft. Der Nachteil bei einer orthonormalen Rotationsmatrix mit Eulerwinkeln ist, dass die Reihenfolge nicht kommutativ (mehrdeutig) ist und somit die Gefahr eines Gimbal-Lock entstehen kann. Dieses Problem gibt es mit Quaternionen nicht. Ein Quaternion wird wie folgt beschrieben:

$$s + v_x i + v_y j + v_z k$$

Ein Quaternion besteht aus vier Elementen, dem Realteil/ Skalaranteil s , welches den Rotationswinkel angibt, sowie dem Imaginärteil/Vektoranteil \vec{v}_e , welcher den Vektor beschreibt, um den die Achsen i, j, k rotiert werden. Damit wird eine Rotation um ein Achsenvektor beschrieben.

$$\vec{v} = [s, \vec{v}_e]^T$$

Die Rotationsparameter werden in Zwischenschritten aus dem Produkt M der schwerpunktreduzierten Koordinaten gebildet, indem \dot{p} und \dot{q} als $n \times 3$ -Matrix S_p und S_q aufgefasst werden.

$$M = S_p^T S_q = \begin{pmatrix} m_{xx} & m_{xy} & m_{xz} \\ m_{yx} & m_{yy} & m_{yz} \\ m_{zx} & m_{zy} & m_{zz} \end{pmatrix}$$

Aus der quadratischen Matrix M lässt sich die spiegelsymmetrische Matrix N bilden.

$$N = \begin{pmatrix} m_{xx} + m_{yy} + m_{zz} & m_{yz} - m_{zy} & m_{zx} - m_{xz} & m_{xy} - m_{yx} \\ m_{yz} - m_{zy} & m_{xx} - m_{yy} - m_{zz} & m_{xy} + m_{yx} & m_{zx} + m_{xz} \\ m_{zx} - m_{xz} & m_{xy} + m_{yx} & -m_{xx} + m_{yy} - m_{zz} & m_{yz} + m_{zy} \\ m_{xy} - m_{yx} & m_{zx} + m_{xz} & m_{yz} + m_{zy} & -m_{xx} - m_{yy} + m_{zz} \end{pmatrix}$$

Durch Ermittlung $\det(N - \lambda I) = 0$ des größten Eigenwertes mit dem korrespondierendem Eigenvektor der Matrix N , lässt sich das gesuchte Quaternion $v = [s, v_e]^T$ bestimmen.

Das ermittelte Quaternion wird in eine Rotationsmatrix R_v überführt.

$$R_v = (s^2 - \vec{v}_e^T \vec{v}_e)I + 2(\vec{v}_e \vec{v}_e^T + s[\vec{v}_e \times])$$

I sei hier eine 3×3 Identitätsmatrix und $\vec{v}_e \times$ sei:

$$v_e \times = \begin{pmatrix} 0 & -v_z & v_y \\ v_z & 0 & -v_x \\ -v_y & v_x & 0 \end{pmatrix}$$

Daraus ergibt sich R_v als Rotationsmatrix eines Einheitsquaternions $|v| = 1$

$$R_v = \begin{pmatrix} s^2 + v_x^2 - v_y^2 - v_z^2 & 2(v_x v_y - s v_z) & 2(v_x v_z + s v_y) \\ 2(s v_z v + v_x v_y) & s^2 - v_x^2 + v_y^2 - v_z^2 & 2(v_y v_z - s v_x) \\ 2(v_x v_z - s v_y) & 2(s v_x + v_y v_z) & s^2 - v_x^2 - v_y^2 + v_z^2 \end{pmatrix}$$

Anschließend kann die Rotationsmatrix R_v in die gesuchten Eulerwinkel (ϕ, θ, ψ) einer starrkörper Rotation umgerechnet werden.

$$\begin{aligned} \phi &= \arctan \left(\frac{2(v_y v_z - s v_x)}{s^2 - v_x^2 - v_y^2 + v_z^2} \right), \\ \theta &= \arcsin (2(v_x v_z + s v_y)), \\ \psi &= \arctan \left(\frac{2(v_x v_y - s v_z)}{s^2 + v_x^2 - v_y^2 - v_z^2} \right) \end{aligned}$$

2.3 Graphentheorie

In diesem Kapitel werden die Grundtypen von Graphen in Bezug zur Graphentheorie [6] erläutert und erhebt keinen Anspruch auf Vollständigkeit. Ein Graph dient zur Visualisierung von abstrakten Strukturen, wie zum Beispiel U-Bahnnetz- oder Straßennetzplan. Dabei werden eine Menge von Objekten und die Beziehungen zwischen den Objekten dargestellt. Objekte werden als Knoten (V_j) und Beziehungen als Kanten (e_i) visualisiert. Die Gesamtheit aller Knoten und Kanten bilden somit den Graphen. Geht nur eine Kante vom Knoten ab, wird solch ein Graph als einfacher Graph (siehe Abb. 2.5a und 2.5c) bezeichnet. Es können auch mehrere Kanten von einem Knoten abgehen oder auf einen Knoten zulaufen. Dieser Graph wird dann Multigraph (siehe Abb. 2.5b und 2.5d) genannt. Des Weitern

unterscheidet man zwischen ungerichteten und gerichteten (mit Pfeilspitze) Graphen. Während bei ungerichteten Kanten die Verlaufsrichtung beliebig ist, gibt die gerichtete Kante den Ablauf explizit vor. Somit kann man zum Beispiel vom Knoten A zwar zum Knoten D gelangen, aber von Knoten D nirgendwo hin, da hier keine abgehende Kante vorliegt (siehe Abb. 2.5d).

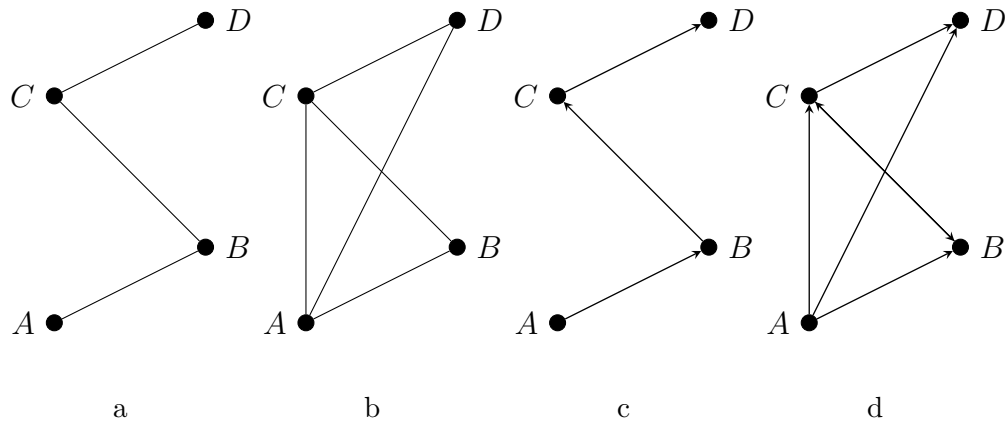


Abb. 2.5: ungerichteter Graph mit und ohne Mehrfachkante

Die Kantenbeziehungen lassen sich einer Adjazenzmatrix oder Inzidenzmatrix abbilden. Die Adjazenzmatrix, s. Abb. 2.6a, beinhaltet die Information, ob zwischen zwei Knoten eine Kante existiert. Die Inzidenzmatrix, s. Abb. 2.6b, eine Aussage gibt, ob ein Knoten auf einer bestimmten Kante liegt. Mit negativen Vorzeichen wird die Kantenrichtung für gerichtete Graphen angegeben.

	A	B	C	D
A	1	1	1	1
B	1	0	1	0
C	1	0	0	1
D	1	0	1	0

	e_1	e_2	e_3
A	1	0	0
B	-1	0	0
C	0	-1	1
D	0	0	-1

Adjazenzmatrix vom Graphen 2.5b Inzidenzmatrix vom Graph 2.5c

Abb. 2.6: Beispiel einer Adjazenzmatrix und Inzidenzmatrix

Eine Folge von Knoten werden als Weg bzw. Pfad bezeichnet. Sie beschreiben die Abfolge der Knoten, die ab dem Startknoten zum Zielknoten durchlaufen werden. Um unterschiedliche Pfade zum selben Ziel abzugrenzen, werden die Kanten zwischen den Knoten unterschiedlich gewichtet. Addiert die gewichteten Kanten (s. Abb. 2.7) auf, lassen sich unterschiedliche Kosten für die jeweiligen Pfade berechnen.

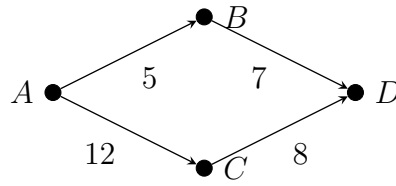


Abb. 2.7: Beispiel eines gerichteten Graphen mit gewichteten Kanten

Für diese Masterarbeit sind nur gerichtete Graphen ohne Mehrfachkanten (siehe Abb.2.5c) relevant, da die Trajektorie sich aus fortlaufenden Posen bildet.

2.4 Graphoptimierung

Im folgenden Abschnitt wird die Graphoptimierung nach dem g^2o -Framework [7] (siehe Abschnitt 3) beschrieben. In diesem Framework sind die in diesem Abschnitt beschriebenen Formeln implementiert.

Das Graphoptimierung ist ein Schätzverfahren zu Lösung des nichtlinearen Problems der kleinsten Quadrate betrachten. Es findet in vielen Bereichen Verwendung, wie in der Robotik sowie im Bereich der Computer Vision.

Allgemein sei $x = (x_1^T, x_2^T, \dots, x_n^T)$ ein Vektor von Parametern, wobei jedes x_i bzw. x_j einen generischen Parameterblock der Menge C von Bedingungen darstellt. z_{ij} und Ω_{ij} beinhalten den Mittelwert und die Kovarianzmatrix der Bedingungen im Bezug der Parameter x_i und x_j . Mit dem Vektor der Fehlerfunktion $e(x_i, x_j, z_{ij})$ wird gemessen, wie gut die Parameterblöcke x_i und x_j die Bedingung z_{ij} erfüllen. Bei perfekter Übereinstimmung nimmt dieser Wert 0 an.

$$F(x) = \sum_{i,j \in C} e(x_i, x_j, z_{ij})^T \Omega_{ij} e(x_i, x_j, z_{ij}) \quad (2.4.0.1)$$

$$x^* = \underset{x}{\operatorname{argmin}}(F(x)) \quad (2.4.0.2)$$

Durch das Lösen von x^* nach der Kleinste-Quadratische-Fehlerminimierung wird der Vektor von Parametern gefunden, der den Graph unter seinen Bedingungen am besten wiedergibt.

Angewendet auf die Aufgabenstellung dieser Arbeit lassen sich durch das Einbeziehen von Zusatzinformationen(GCPs) die Fehler der Posen an den GCPs mit jeder Iteration minimieren. Die Trajektorie wird als Graph dargestellt, wobei jede Pose X_i und Referenzpunkt L_l (GCP) einen Knoten des Graphen repräsentiert und die Kanten Abhängigkeiten zwischen den Knoten wiedergeben.

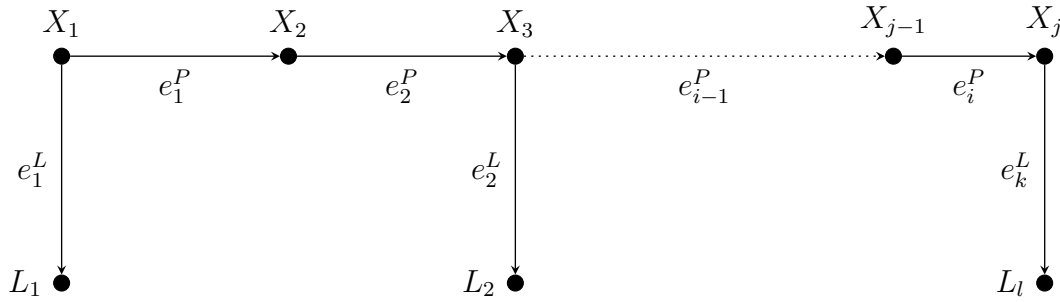


Abb. 2.8: Darstellung einer Trajektorie als Graph

Graphoptimierung nach der Methode der kleinsten Quadrate [7]

Die Methode der kleinsten Quadrate ist ein mathematisches Ausgleichsverfahren für überbestimmte Systeme mit mehr Gleichungen als Unbekannten. Hierbei werden die gesuchten Parameter so geschätzt und iterativ ihrem optimalen Wert angenähert, dass die Summe der Fehlerquadrate minimiert wird. Entwickelt wurde diese Methode von Carl Friedrich Gauß und im Jahr 1809 veröffentlicht.

Zur Vereinfachung wird für die Kostenfunktion $e(x_i, x_j, z_{ij})$ folgende Definition vereinbart:

$$e(x_i, x_j, z_{ij}) \stackrel{def.}{=} e(x_i, x_j) \stackrel{def.}{=} e_{ij}(x) \quad (2.4.0.3)$$

Sobald eine Initialbelegung \check{x} der Parameter x bekannt ist, lässt sich die Gleichung (2.4.0.2) durch ein Gauß-Newton-Verfahren oder Levenbreg-Marquardt-Algorithmus numerisch lösen. Die Fehlerfunktion $e_{ij}(x)$ wird durch eine Taylor-expansion der ersten Ordnung J_{ij} um die Initialbelegung \check{x} approximiert.

$$\begin{aligned} e_{ij}(\check{x}_i + \Delta x_i, \check{x}_j + \Delta x_j) &= e_{ij}(\check{x} + \Delta x) \\ &\simeq e_{ij} + J_{ij} \Delta x \end{aligned} \quad (2.4.0.4)$$

Durch Substitution der Formel (2.4.0.4) in die Fehlergleichung F_{ij} der Formel (2.4.0.1) erhält man folgende neue Gleichung:

$$\begin{aligned}
F_{ij}(\check{x} + \Delta x) &= e_{ij}(\check{x} + \Delta x)^T \Omega_{ij} e_{ij}(\check{x} + \Delta x) \\
&\simeq (e_{ij} + J_{ij} \Delta x)^T \Omega_{ij} (e_{ij} + J_{ij} \Delta x) \\
&= \underbrace{e_{ij}^T \Omega_{ij} e_{ij}}_{c_{ij}} + 2 \underbrace{e_{ij}^T \Omega_{ij} J_{ij}}_{b_{ij}} \Delta x + \Delta x^T \underbrace{J_{ij}^T \Omega_{ij} J_{ij}}_{H_{ij}} \Delta x \\
&= c_{ij} + 2b_{ij} \Delta x + \Delta x^T H_{ij} \Delta x
\end{aligned} \tag{2.4.0.5}$$

Setzt man diese lokale Approximation (2.4.0.5) in die Funktion $F(x)$ aus der Formel (2.4.0.1) ein und fasst $c = \sum c_{ij}$, $b = \sum b_{ij}$ und $H = \sum H_{ij}$ zusammen, so lässt sich die Gleichung wie folgt umschreiben:

$$\begin{aligned}
F(\check{x} + \Delta x) &= \sum_{i,j \in C} F_{ij}(\check{x} + \Delta x) \\
&\simeq \sum_{i,j \in C} c_{ij} + 2b_{ij} \Delta x + \Delta x^T H_{ij} \Delta x \\
&= c + 2b^T \Delta x + \Delta x^T H \Delta x
\end{aligned} \tag{2.4.0.6}$$

Um Δx zu minimieren, wird folgendes Gleichungssystem gelöst:

$$H \Delta x^* = -b \tag{2.4.0.7}$$

H ist die Kovarianzmatrix des gesamten Systems (Adjazenzmatrix des Graphen) und hat nur an den Stellen Einträge ungleich 0, an denen zwei Blöcke durch eine Bedingung verknüpft sind. Durch iteratives Inkrementieren durch Δx^* des Initialwertes \check{x} lässt sich x^* bestimmen

$$x^* = \check{x} + \Delta x^* \tag{2.4.0.8}$$

Mit dem Gaus-Newton-Algorithmus wird die Gleichung (2.4.0.6) iterativ linearisiert, in dem die Gleichung (2.4.0.7) gelöst und \check{x} durch die Gleichung (2.4.0.8) in jeder Iteration erneuert wird. Dies wird wiederholt, bis ein Abbruchkriterium (z.B. Anzahl der Iterationen oder keine wesentlichen Änderungen ϵ zwischen aktuellem Wert und neuem Wert) erreicht ist.

Der Levenberg-Marquardt-Algorithmus (2.4.0.9) erweitert den Gauß-Newton-Algorithmus durch einen Dämpfungsfaktor λ .

$$(H + \lambda I) \Delta x^* = -b \tag{2.4.0.9}$$

Dieser Dämpfungsfaktor dient als Regulator der Konvergenz. Bei einem ungünstigen Initialwert ist die Konvergenz beim Gauß-Newton-Algorithmus nicht gegeben oder erreicht ihn sehr langsam. Der Dämpfungsfaktor verbessert die Konvergenzeigenschaft, d.h. dass der Levenberg-Marquardt-Algorithmus mit höherer

die Wahrscheinlichkeit konvergiert, bei schlechten Initialwerten. Je höher λ gewählt wird, desto kleiner ist Δx . Ist der berechnete Fehler kleiner als der zuvor berechnete, so wird λ kleiner.

3 Softwaremethoden

Dieser Abschnitt beschreibt die Softwarebibliotheken, die zur Aufgabenumsetzung genutzt wurden.

IPS-Software-Bibliothek

Die IPS-Bibliothek ist eine umfangreiche C++ Softwarebibliothek des DLR-OS. Aus dieser Bibliothek werden grundlegende Funktionen für das Einlesen der IPS-Trajektorie, für die Matrixoperationen und deren Klassen, sowie die Implementierung der Helmert-Transformation genutzt.

g^2o - Framework

Das *General (hyper) Graph Optimization (g^2o)* - Framework [7] ist eine Programmbibliothek, welche in C++ geschrieben ist und zur Lösung allgemeiner Graphoptimierungsprobleme genutzt wird. Es stellt Klassen und Funktionen bereit, um ein kleinste-Quadrate-Minimierungsproblem mit nichtlinearen Optimierungsverfahren, wie Levenberg-Marquardt-Methode und Gauss-Newton-Verfahren zu lösen. Dieses Framework [7] wird genutzt, um die IPS-Trajektorie zu verbessern.

4 Konzept der graphbasierten Optimierung einer absoluten Trajektorie anhand von Ground Control Points

Dieses Kaptiel befasst sich mit der Entwicklung eines Konzepte zur graphenbasierten Optimierung einer absoluten Trajektorie mittels Ground Control Points unter Verwendung der in Kapitel 2.4 und 2.2 vorgestellten Methoden. Zur Lösung der Aufgabenstellung aus dem Abschnitt 1 wurde folgendes Konzept ausgearbeitet, das die graphbasierte Optimierung zur Verbesserung der absoluten Trajektorie realisiert.

Es wird davon ausgegangen, dass die zu optimierende Trajektorie bereits prozessiert ist und vollständig vorliegt (siehe Abb. 4.1). Die relative IPS-Trajektorie (*navigationFilteredSynced.mat*) wird zusammen mit der Datei *GPSs.csv*, welche die absoluten kartesischen WGS84 Koordinaten der GCPs enthält, die auf dieser Trajektorie liegen, eingelesen. Mit den ersten drei GCPs aus dieser *GPSs.csv* Datei wird die Transformation (siehe Abb. 4.1 Helmert-Transformation, orange) mit den in Abschnitt 2.2.1 und 2.2.2 beschriebenen Methoden vom relativen - in das absolute Koordinatensystem umgerechnet. Die transformierte Trajektorie wird gespeichert (*trajectoryHT.txt*). Im nächsten Schritt werden die transformierten Koordinaten vom graphbasierten Optimierungsprozess (siehe Abb.4.1 Graph-Optimierung, grün) verarbeitet. In diesem Prozess wird die Trajektorie als Graph aufgebaut. Jede Koordinate stellt einen Knoten da und die Kanten bilden sich aus den Beziehungen der Knoten x_i zu x_{i+1} , wie in Abschnitt 3 beschrieben. Die Graphenoptimierung wird mit dem Levenberg-Marquart-Verfahren realisiert. Daraus resultiert eine graphoptimierte Trajektorie, welche als Datei gespeichert wird (*trajectoryG2O.txt*).

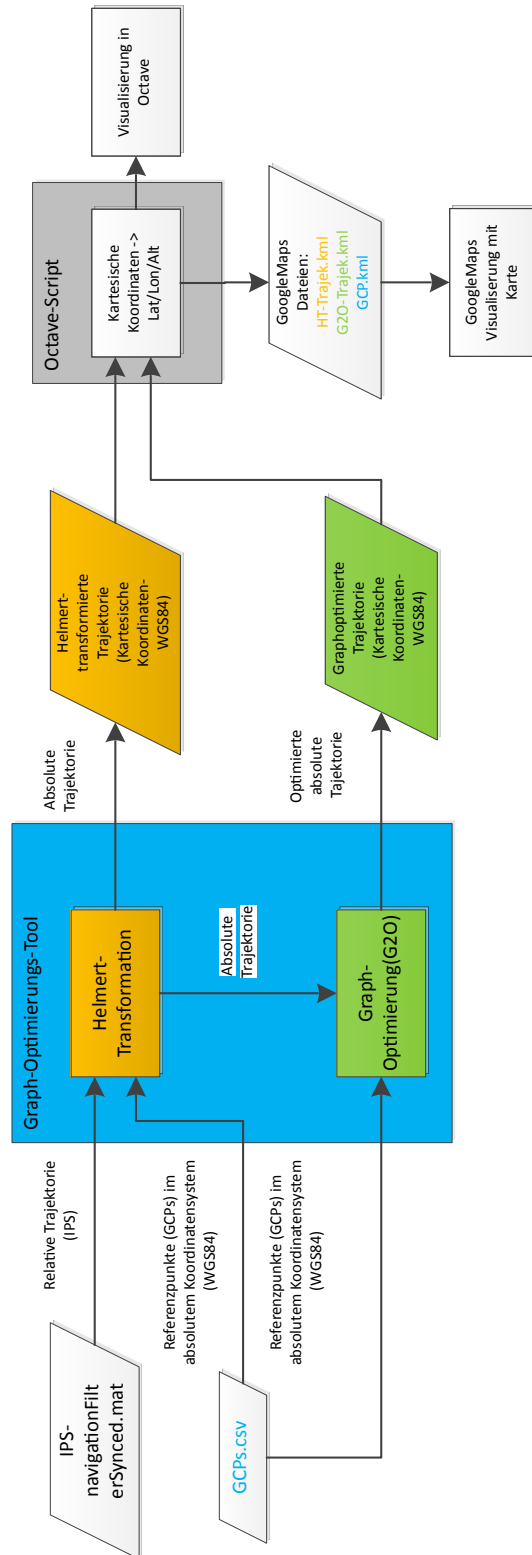


Abb. 4.1: Prozesskette der graphbasierten Optimierung einer IPS-Trajektorie

Die Struktur der beiden gespeicherten Trajektorien (*trajectoryHT.txt* und *trajectoryG2O.txt*) ist wie folgt in Tabelle 1 definiert:

Zeitstempel vom IPS	Orientierung als Quaterni- on	IPS-Position als ECEF Koordinate	als Kovarianzmatrix
------------------------	-------------------------------------	-------------------------------------	------------------------

Tab. 1: Dateistruktur der Helmert-transformierten - und graphoptimierten Trajektorie

Jede Zeile entspricht somit einer Pose und die Gesamtheit aller Posen bilden die Trajektorie.

Der Aufbau der GCP-Dateien (*GCPs.csv*) ist in Tabelle 2 wie folgt definiert:

Zeitstempel vom IPS zu dem die- ser GCP erreicht wurde	GCP-ID	GCP-Position als ECEF Koordinate
---	--------	-------------------------------------

Tab. 2: Dateistruktur der GCP-Datei

Jede Zeile entspricht dabei einem Referenzpunkt (GCP).

Im Anschluss wird ein Octave-Script ausgeführt, das das Ergebnis der Helmert-transformierten - und graphoptimierten Trajektorie visualisiert. Hierfür werden die zuvor gespeicherten Trajektorien und die Liste der GCPs eingelesen. Anschließend werden die Earth-Centered-Earth-Fixed (ECEF)-Koordinaten des kartesischen im WGS84-Bezugssystems in räumliche Polarkoordinaten (Längengrad, Breitengrad, Höhe) des WGS84 umgerechnet. Anschließend werden Graphiken erstellt, in denen die Helmert-transformierte - und graphoptimierte Trajektorie, sowie die GCPs abgebildet sind. Außerdem werden *KML-Dateien* der Trajektorien generiert und damit eine Visualisierung des Ergebnisses in GoogleEarth ermöglicht.

5 Implementierung der graphbasierten Optimierung von IPS-Trajektorien

Die Implementierung des Konzeptes aus Abschnitt 4 erfolgt mittels Modulen aus der IPS-Bibliothek und des g^2o -Framework (s. Abb. 5.1)

Die IPS-Bibliothek stellt Funktionen zur Umsetzung der in Abschnitt 2.2.1 gezeigten Formel (2.2.1.1) und zur Ermittlung der Transformationsparameter nach Horn (Kapitel 2.2.2) bereit. Zudem werden auch Input/Output (I/O)-Routinen von der IPS-Bibliothek genutzt, die das Einlesen und Speichern der IPS-Trajektorie ermöglichen.

Das g^2o -Framework wird zur Realisierung der Graphoptimierung der IPS-Trajektorie, wie in Abschnitt 2.3 und 3 beschrieben, genutzt. Das Framework beinhaltet auch eine Umsetzung des Levenberg-Marquardt-Verfahrens.

In Graphoptimierungs-Software werden die Posen der IPS-Trajektorie in Objektklassen des g^2o -Framework überführt. Dazu wurden Funktionen implementiert, die die Kompatibilität der Objekte-Klassen zwischen der IPS-Bibliothek und dem g^2o -Framework gewährleistet.

Mittels der zuvor genannten Softwarebibliotheken wurde und entsprechend des vorgestellten Konzeptes wurde eine Software zur graphbasierten Optimierung von IPS-Trajektorien (s. Abb. 4.1) in C++ implementiert. Als Ergebnis wird eine Helmert-transformierte - und eine graphoptimierte IPS-Trajektorie gespeichert.

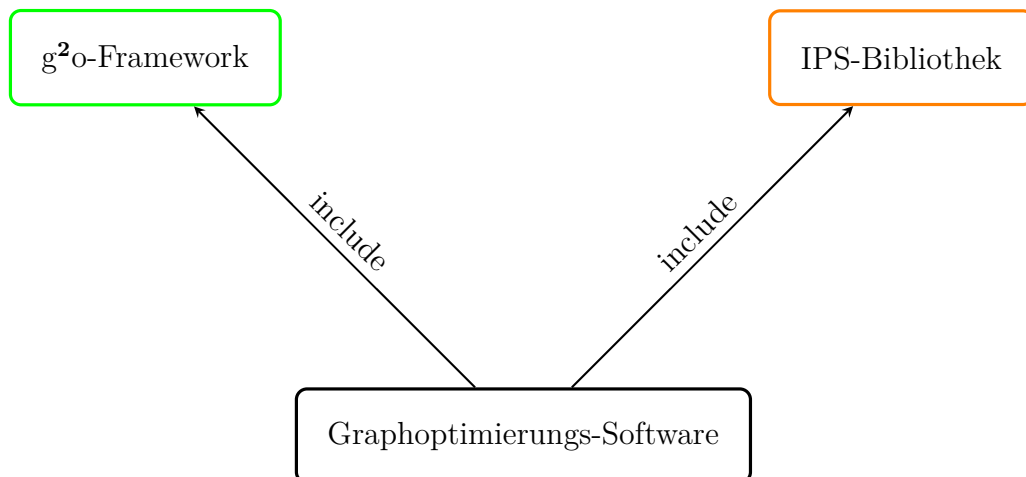


Abb. 5.1: Überblick verwendeter Softwarebibliotheken zur Implementierung des Konzeptes zur graphbasierten Optimierung von IPS-Trajektorien

6 Validierung

Die in dieser Arbeit entwickelte Software zur graphbasierten Optimierung von IPS-Trajektorien wird mit einem IPS-Datensatz getestet. Hierfür wurde eine Route um das Gebäude des Deutschen Zentrum für Luft- und Raumfahrt e.V. (DLR)s in Berlin-Adlershof, s. Abb. 6.1 mit 23 GCPs mit IPS abgelaufen.

Beschreibung des Testlaufs:

IPS wurde von einer Person entlang der mit Referenzpunkten belegten Route um das Gelände des DLR getragen. Dabei wurden die auf dem Weg liegenden 23 GCP mit dem Gerät angelaufen und IPS kurzzeitig auf dem Boden am GCP abgesetzt. Die GCPs sind markierte Punkte am Boden, ihre absolute Koordinaten sind bekannt. Die Koordinaten der GCP wurden durch RTK-GPS-Verfahren (RTK - Real Rime Kinematic), s. Abb. 6.2, vermessen und haben eine Positionsgenauigkeit von 2 cm. Der Testlauf hatte eine Gesamtlaufzeit von 20 min, die Gesamtlänge betrug 1059 m.

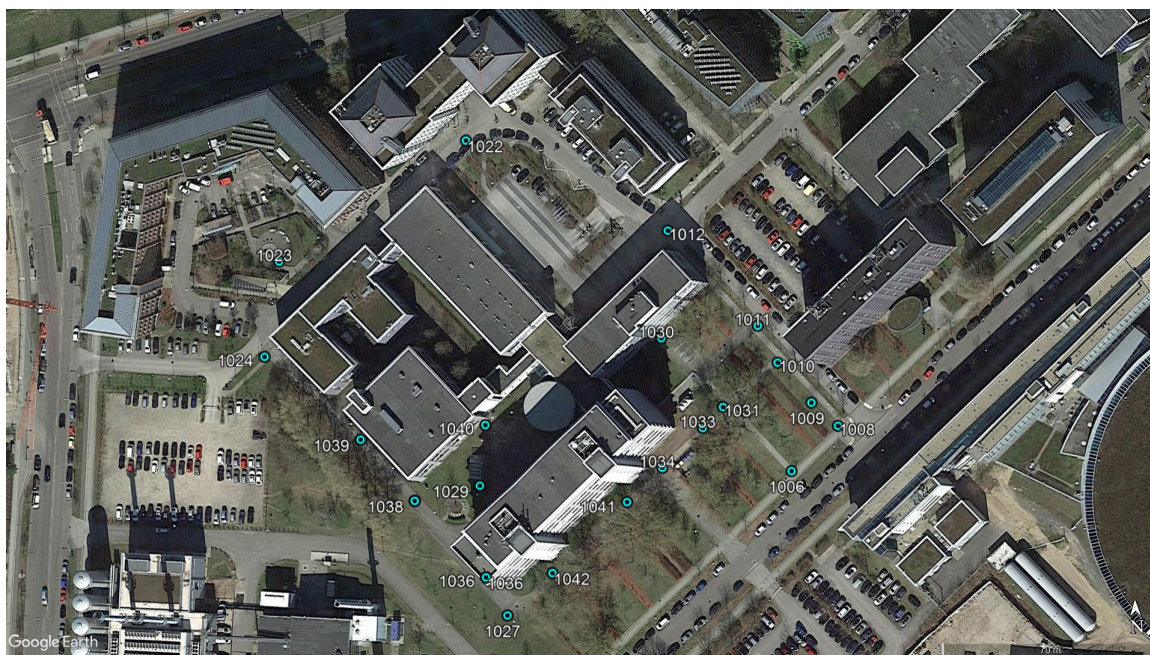


Abb. 6.1: Übersicht der Ground Control Points

In der Tabelle 3 ist die Abfolge der angegangenen GCPs und ihre ID aufgelistet. Sich wiederholende GCP-IDs geben an, dass der GCP zu einem späteren Zeitpunkt erneut als Referenzpunkt genutzt wurde.

Abfolge	GCP-ID
1	1031
2	1006
3	1008
4	1009
5	1010
6	1033
7	1034
8	1041
9	1042
10	1036
11	1029
12	1040
13	1038
14	1027
15	1036
16	1038
17	1039
18	1024
19	1023
20	1022
21	1012
22	1011
23	1030

Tab. 3: GCP Abfolge und ID

Für die Berechnung der IPS-Trajektorie wurden die Messdaten des Stereokamerasystems und der IMU benutzt. Bei diesem Datensatz lag ein "veraltetes" Kalibrations-File vor, das eine aktuelle Änderung der IPS-Geometrie, die durch eine kurz zuvor durchgeführte Reparatur des Sensors, nicht berücksichtigt hat. Die Auswirkung einer graphbasierten Optimierung ist in diesem Fall besonders interessant, da davon auszugehen ist, dass die Sensordrift über die Strecke der Trajektorie dazu führt, dass die Helmert-transformierte-Trajektorie von der tatsächlich gelaufenen Trajektorie noch stärker als bei einem perfekt kalibrierten System abweicht.



Abb. 6.2: GCPs-Vermessung mittels RTK-GPS-Verfahren

Um den Prozessierungsaufwand messen zu können, wurde eine Zeitmessung zur Datenverarbeitung der gesamten Trajektorie mit dem Testrechner, s. Tab. 4, durchgeführt.

Prozessor	i7-6700 @ 3.40GHz
Arbeitsspeicher	32 GB RAM
Betriebssystem	Windows 7 64 Bit Enterprise Service Pack 1

Tab. 4: Systemspezifikation des Testrechners

7 Ergebnisse

Dieser Abschnitt zeigt die Ergebnisse, die mit dem IPS-Datensatz aus dem Testlauf entlang des Referenzpunktfeldes, s. Abschnitt 6 Abb. 6.1, erzielt wurden.

Für die Helmert-Transformation wurden nur die ersten drei GCPs verwendet. Die Fehler werden deshalb nicht über die gesamte transformierte Trajektorie gleichmäßig verteilt, was dazu führt das die Drift der Trajektorie sehr deutlich wird.

Zum Verständnis der Abbildungen sei folgende Konvention zu beachten:

Farbe	Symbol	Erklärung	Abkürzung
rote Trajektorie	—	absolute Trajektorie mittels der Helmert-Transformation	HT
grüne Trajektorie	—	absolute Trajektorie mittels der Helmert-Transformation und anschließender Graphoptimierung	G2O
blauer Kreis	•	absolute Referenzpunkte (Ground Control Point)	GCP
rotes Dreieck	▼	Position des IPSs zum Zeitpunkt an einem GCP in der Helmert-transformierten Trajektorie	HT-GCP
grünes Dreieck	▲	Position des IPSs zum Zeitpunkt an einem GCP in der graphoptimierten Trajektorie	G2O-GCP
gelbes Dreieck	▶	Startpunkt der IPSs-Trajektorie	Start
türkises Dreieck	◀	Endpunkt der IPSs-Trajektorie GCP	Ende

Abbildung 7.1 zeigt die jeweiligen Trajektorien in einem zweidimensionalen Längengrad/Breitengrad-Graphik (x-Achse/y-Achse). Man erkennt, dass die graphoptimierte Trajektorie (grüne Trajektorie) auf den GCPs liegt, während die Helmert-transformierte Trajektorie (rote Trajektorie) mit wachsender Wegstrecke immer mehr in der x-y-Ebene abweicht.

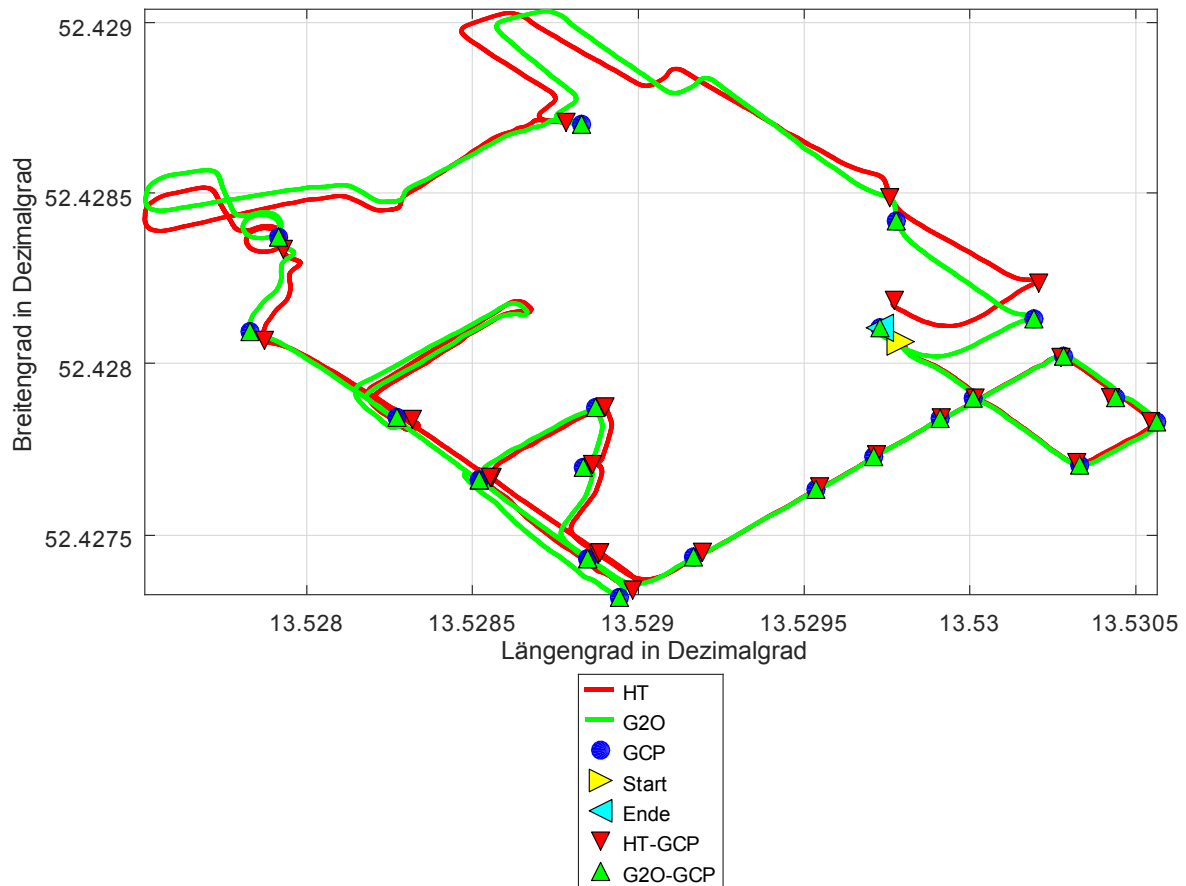


Abb. 7.1: 2D-Horizontaldarstellung der Helmert-transformierten (rot) - und graphoptimierten Trajektorie (grün) und GCPs (Blau)

Abbildung 7.2 zeigt die jeweiligen Trajektorien in eine Längengrad/Höhe-Graphik (x-Achse/y-Achse). In dieser Ansicht wird deutlich, an welchen Stellen IPS an einem GCP abgelegt wurde (ca. 1 Meter Höhenunterschied). Deutlich kann man erkennen, dass auch eine Abweichung in der Höhe (Altitude) mit Zunahme der Wegstrecke voranschreitet.

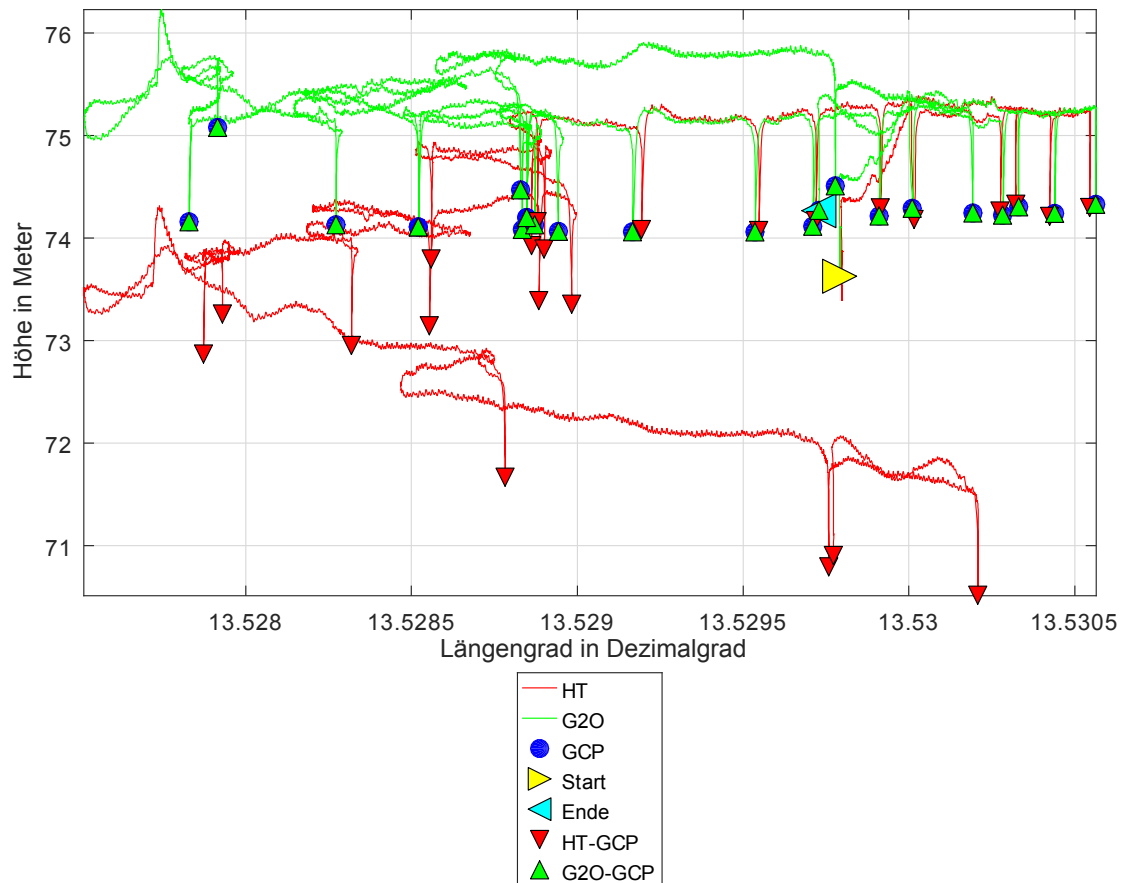


Abb. 7.2: 2D-Seitenansicht der Helmert-transformierten (rot) - und graphoptimierten Trajektorie (grün) und GCPs (Blau)

Die Abweichung der Posen der Helmert-transformierten Trajektorie an den GCPs ist in Abbildung 7.3 dargestellt. In dem Diagramm wird sichtbar, wie sich die Abweichung zwischen GCP und transformierter IPS-Trajektorie bei Verwendung von lediglich der ersten drei GCP für die Helmert-Transformation kontinuierlich vergrößert.

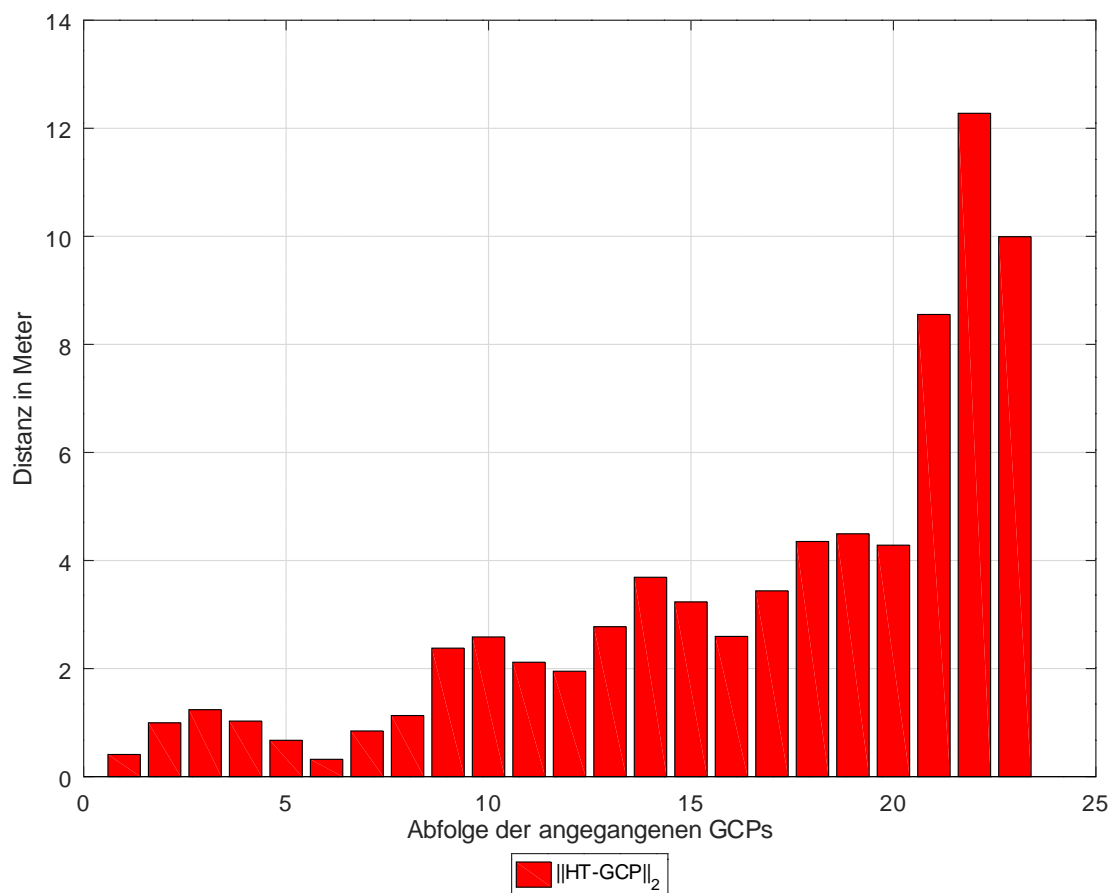


Abb. 7.3: Absolute Abweichung der Helmert-transformierten Trajektorie an den GCPs

Zur Validierung der Graphoptimierung werden die Posen der optimierten Trajektorie an den GCPs als Differenzdiagramm 7.4 dargestellt. Die Abbildung zeigt, dass die Fehler an den GCPs durch die Optimierung vernachlässigbar klein geworden sind und deutlich im Sub-Millimeterbereich liegen.

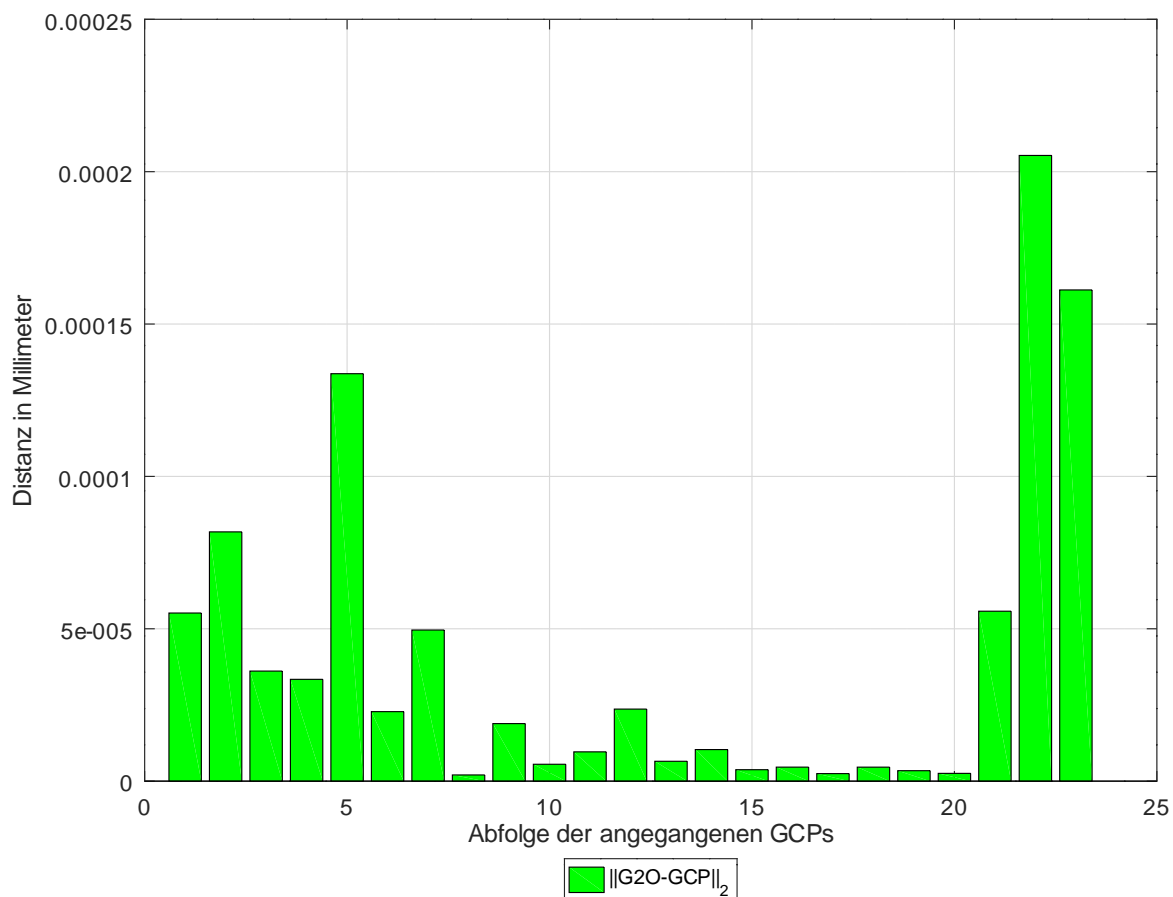


Abb. 7.4: Absolute Abweichung der graphoptimierten Trajektorie an den GCPs

Um die Auswirkung der Graphoptimierung auf die Bereiche der Trajektorie außerhalb von GCP zu beurteilen, wurden einzelne GCP (No. 6, 11, 16, 19, 20 und 21) aus der Graphoptimierung entfernt und als Kontrollpunkte verwendet. Abbildung 7.5 zeigt, dass die Verbesserung an den Kontrollpunkten geringer ist, als an den in GCPs, die bei der Optimierung berücksichtigt worden sind. Dennoch ist im Vergleich zu Abb. 7.3 an den GCPs 6, 11, 16, 19, 20 und 21 eine wesentliche Verbesserung zu sehen. Da die IPS-Trajektorie in der Gesamtheit eine Verbesserung erfahren hat, wirkt sich diese ebenso auf Streckenabschnitte der Trajektorie zwischen den GCP aus. Die Wahl der Anzahl und der Dichte der GCPs bestimmen dabei das Maß der Verbesserung der IPS-Trajektorie durch die Graphoptimierung.

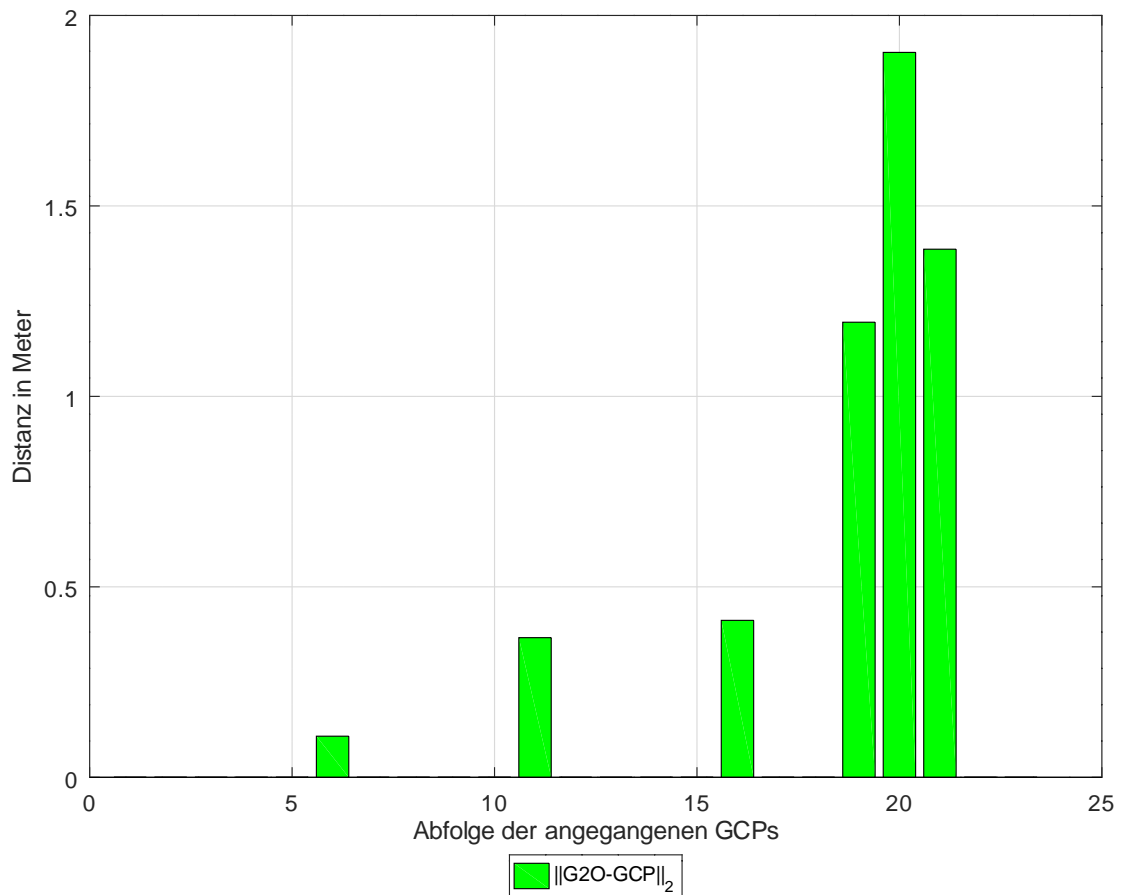


Abb. 7.5: Absolute Abweichung der graphoptimierten Trajektorie an den GCPs mit den Kontrollpunkten)

Abbildung 7.6 die absolute Differenz der Trajektorien über die Zeit. Zur Orientierung sind die Markierungen (blaue Punkte) auf dem Graph eingetragen, als sich IPS auf einem GCP befand.

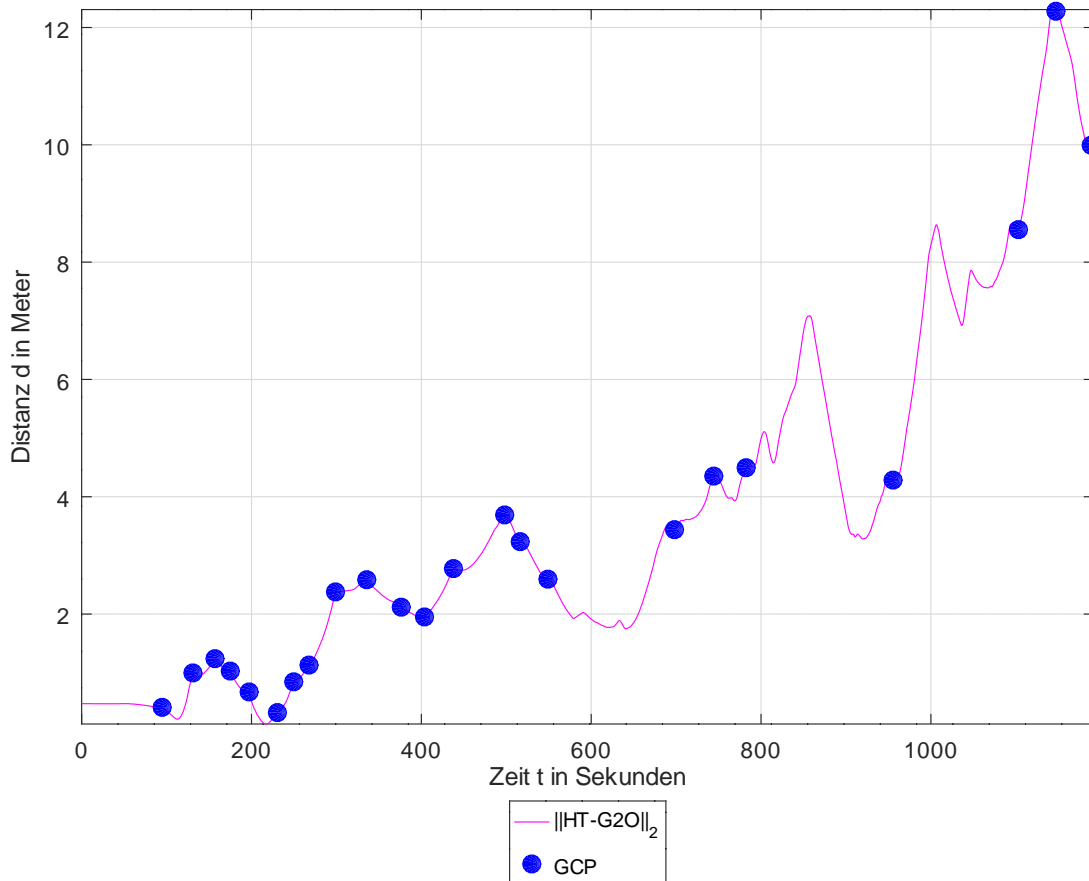


Abb. 7.6: Absolute Abweichung der Helmert-transformierten zur graphoptimierten Trajektorie mit GCPs zum Zeitpunkt t auf der Wegstecke

Abbildung 7.7 zeigt die Projektion der Trajektorien in einer zweidimensionalen Satellitenabbildung in GoogleMaps. Da die markierten GCPs in dieser Abbildung schwer sichtbar sind, wurden die ID-Nummern der GCPs noch zusätzlich eingeblendet. Da das Satellitenbild nicht perspektivisch korrekt über dem Testgebiet liegt, sieht es fälschlicherweise so aus, als ob die Trajektorien durch Gebäude verlaufen. Für eine bessere Darstellung können die KML-Dateien mit GoogleMaps visualisiert und die 3D-Gebäude-Projektion aktiviert werden. Dann wird deutlich, dass die graphoptimierte Trajektorie nicht durch Gebäude verläuft (s. Abb 7.8).

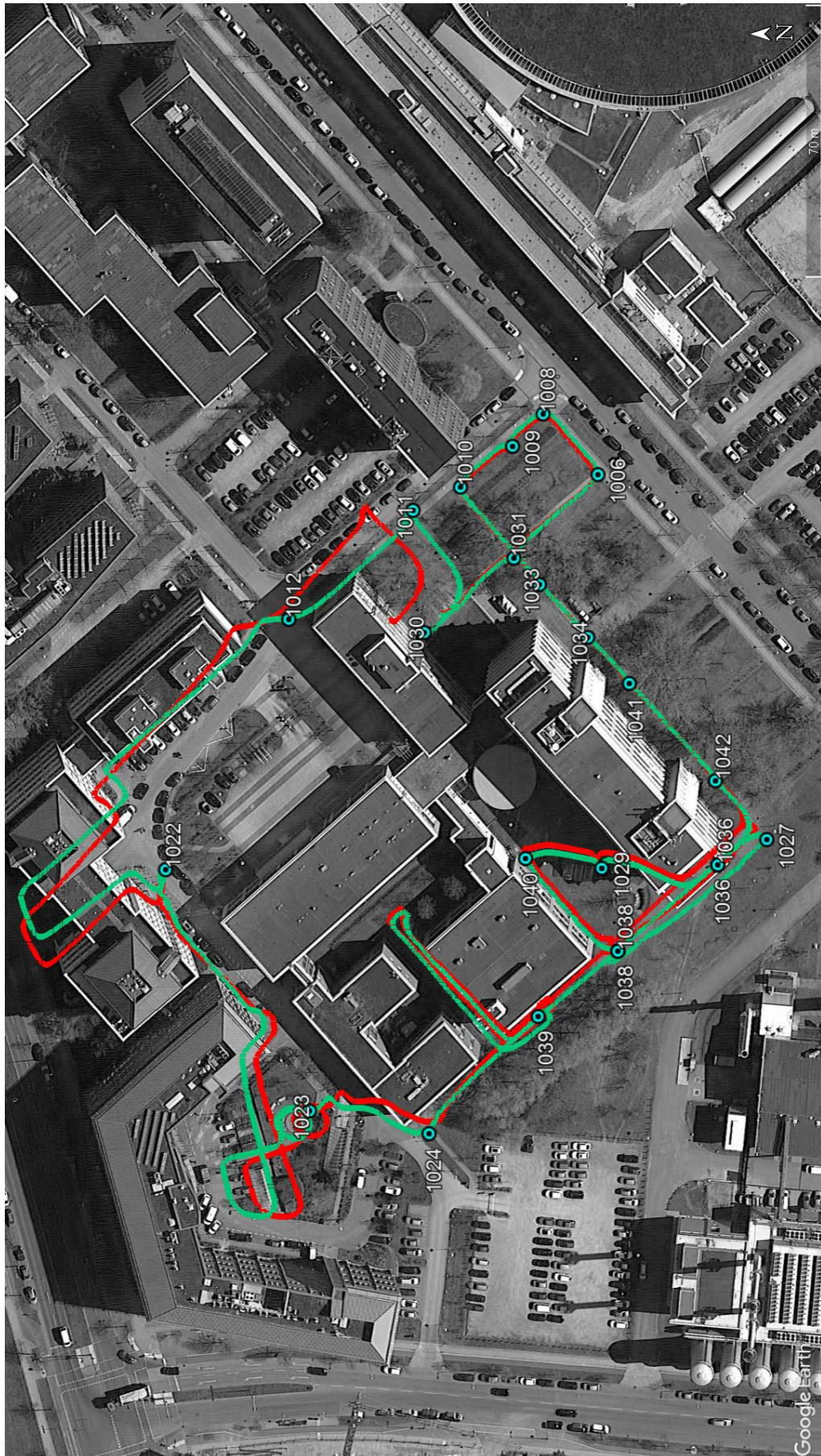


Abb. 7.7: Darstellung der Helmert-transformierten und graphoptimierten Trajektorie in GoogleEarth

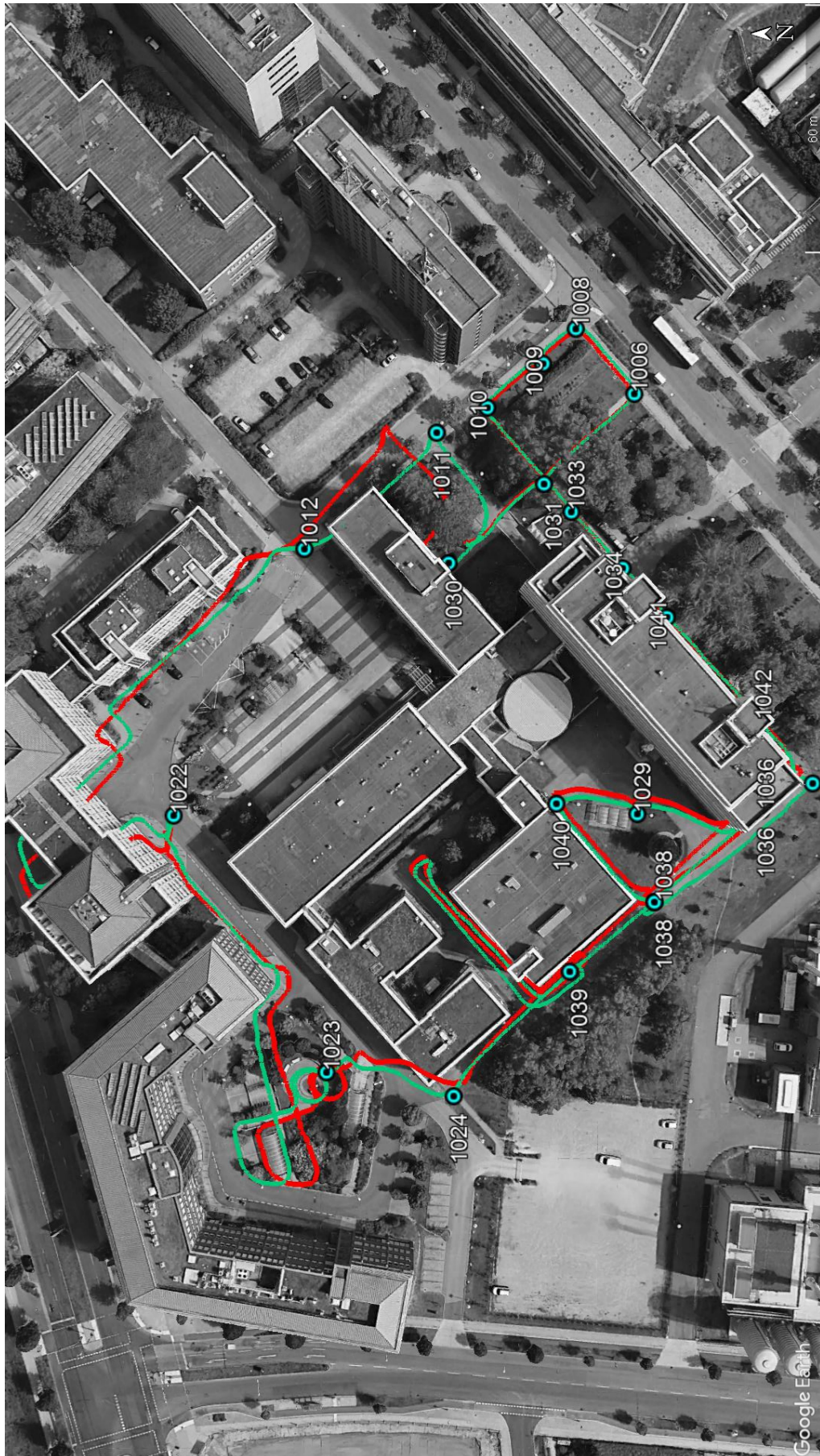


Abb. 7.8: Darstellung der Helmert-transformierten und graphoptimierten Trajektorie in GoogleEarth mit 3D-Gebäude-Projektion

Die Zeitmessung, s. Tab. 5, wurde in drei Durchläufen für 11834 Posen mit 23 GCPs über die gesamte Steckle ausgeführt und brachte folgendes Ergebnis:

Durchlauf	Helmert-Transformation	Graphoptimierung
1	45 ms	3338 ms
2	44 ms	3893 ms
3	43 ms	3908 ms
Ø Zeit	44 ms	3713 ms

Tab. 5: Rechenaufwand der Datenverarbeitung für die Helmert-transformierte - und graphoptimierte Trajektorie

Die Spalte Helmert-Transformation beinhaltet die Zeit, die für die Ermittlung der Transformationsparameter mit drei GCPs und die Umrechnung der gesamten relativen IPS-Trajektorie in eine absolute Trajektorie benötigt wird. Die Spalte Graphoptimierung gibt die Zeit wieder, die benötigt wurde, um die Trajektorie und die GCPs in die Graph-Struktur zu überführen und die Optimierung mit maximal 100 Iterationen durchzuführen. Für die drei Durchläufe sind die Durchschnittszeiten der Helmert-Transformation und der Graphoptimierung berechnet worden.

8 Zusammenfassung

Diese Arbeit hatte Ziel, der IPS-Trajektorien mittels Graphoptimierung zu verbessern. Mittels einer Stereokamera und IMU erzeugt IPS eine relative Trajektorie. Durch eine Starrkörpertransformation und Referenzpunkte wird die relative Trajektorie in absolute Koordinaten transformiert. Die Ermittlung der Transformationsparameter erfolgt nach der Methode von Horn. Nichtlineare Fehler, die bei IPS mit fortschreitender Laufzeit und Wegstrecke aufakkumuliert werden, können bei Eigenverortung mit einfließen, können jedoch durch die Starrkörpertransformation von relativen zu absoluten Koordinatensystem nicht kompensiert werden. Das Resultat der Helmert-transformierten Trajektorie ist eine über alle Referenzpunkte gemittelte fehlerbehaftete absolute Trajektorie. Zur Verbesserung der absoluten IPS-Trajektorie wurde in dieser Masterarbeit ein Konzept zur graphbasierten Optimierung entwickelt und implementiert. Das graphbasierte Konzept nutzt eine Implementierung zur Graphoptimierung mittels g^2o -Framework sowie über die Trajektorie verteilte Ground Control Point zur Koordinatentransformation. Als Ergebnis erhält man eine mittels Referenzpunkten und Graphoptimierung verbesserte IPS-Trajektorie. Das Verfahren wurde anhand eines IPS-Testdatensatzes an einem GCP-Feld am DLR in Berlin-Adlershof validiert. Die Anwendung der Graphoptimierung auf die absolute IPS-Trajektorie hat gezeigt, dass mit diesem Ansatz nichtlineare Fehler, die bei IPS während der Laufzeit entstehen, an den Referenzpunkten korrigiert werden. An Kontrollpunkten ergeben sich ebenfalls Verbesserungen der Trajektoriengenauigkeit. Allerdings erfolgt dies in einem geringeren Umfang als an den GCP, die zur Graphoptimierung der IPS-Trajektorie verwendet werden.

9 Ausblick

In dieser Arbeit wurde gezeigt, dass eine Graphoptimierung der IPS-Trajektorie mit GCPs zu einer verbesserten Trajektorie führt. Eine Randbedingung war, dass IPS an den Referenzpunkten abgelegt werden kann. Damit der Ansatz praktikabel wird, müssten Referenzpunkte ohne Abstellen von IPS in den Datenstrom integriert werden, z.B. durch die Hinzunahme externer Verortungssignale wie Global Navigation Satellite System (GNSS) oder optischer Marker. Denkbar wäre auch hier die Ausweitung auf RFID - oder NFC-Technologie-basierte Referenzpunkte.

Ein weiterer Punkt betrifft die Echtzeitfähigkeit. Die in dieser Arbeit entwickelte Software könnte in die Echtzeit-IPS-Navigationslösung integriert werden. Die Graphoptimierung von IPS-Trajektorien in Echtzeit bedarf jedoch einer entsprechenden Konzepterweiterung und Umsetzung.

Literaturverzeichnis

- [1] D. Griebbach, D. Baumbach, A. Börner, and S. Zuev, “Accuracy evaluation of stereo vision aided inertial navigation for indoor environments,” in *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 40, pp. 13–17, 2013.
- [2] D. Griebbach, D. Baumbach, A. Börner, M. Buder, I. Ernst, E. Funk, J. Wohlfeil, and S. Zuev, “IPS - A system for real-time navigation and 3D modeling,” *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, vol. 39, no. September, pp. 21–26, 2012.
- [3] B. K. P. Horn, “Closed-form solution of absolute orientation using unit quaternions,” *Journal of the Optical Society of America A*, vol. 4, p. 629, apr 1987.
- [4] T. Luhmann, *Nahbereichsphotogrammetrie: Grundlagen, Methoden und Anwendungen*. Wichmann, 2010.
- [5] G. A. Watson, “Computing Helmert transformations,” *Journal of Computational and Applied Mathematics*, vol. 197, no. 2, pp. 387–394, 2006.
- [6] U. Brandes, “Graphentheorie,” in *Handbuch Netzwerkforschung*, pp. 345–353, 2010.
- [7] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, “G2o: A general framework for graph optimization,” in *Proceedings - IEEE International Conference on Robotics and Automation*, pp. 3607–3613, 2011.

