

GEOMETER – Der Landvermesser-Roboter

Laurin Keanu Webers, Elektrotechnik/Informationstechnik
Otto-von-Guericke-Universität Magdeburg

Zusammenfassung—Dieser Beitrag beschreibt den "Geometer", einen differentiell angetriebenen Roboter auf Basis des LEGO Mindstorms EV3, der als messendes Linienfolgesystem konzipiert ist. Entwickelt wurde ein Fahrzeug, das eine dreifarbig kodierte Strecke selbsttätig abfährt, seine Bewegung per Odometrie protokolliert und daraus ein mathematisches Modell des zurückgelegten Pfades generiert. Auf Grundlage dessen lassen sich sowohl Distanzen als auch Flächeninhalte präzise ermitteln. Die Steuerung erfolgt mit MATLAB anhand der Daten zweier Farbsensoren, aus deren Signalen die erforderlichen Fahrmanöver abgeleitet werden. Die Weg- und Winkelmessung nutzt die Drehgeber der Motoren, deren Rohwerte durch eine Kalibrierung in physikalische Größen überführt werden. Die Validierung zeigt, dass der Roboter einfache geometrische Formen exakt rekonstruieren kann. Insgesamt wird deutlich, dass selbst anspruchsvolle Vermessungsaufgaben mit LEGO-Hardware lösbar sind.

Schlagwörter—Differentieller Antrieb, LEGO Mindstorms EV3, MATLAB, Odometrie, Zustandsautomat, Farbsensor, Linienfolger

I. GRUNDIDEE

Die Fähigkeit, Strecken und Flächen autonom zu erfassen, zählt zu den grundlegenden Anforderungen an robotische Systeme und findet in zahlreichen Bereichen Anwendung, von der Kartierung bis zur industriellen Qualitätskontrolle. Plattformen wie LEGO Mindstorms und MATLAB erlauben einen niedrighwelligen Zugang zur praktischen Erprobung von Verfahren der Odometrie und Pfadrekonstruktion. Das hier vorgestellte Projekt verfolgt das Ziel, einen Roboter zu entwickeln, der eine vorgegebene Route selbstständig befährt, seine Bewegung odometrisch dokumentiert und daraus ein mathematisches Abbild der Strecke gewinnt, um geometrische Größen wie Entfernungen und Flächeninhalte zu bestimmen.

II. KONZEPT

Die Aufgabe des Roboters besteht darin, eine vorgegebene Wegstrecke zu durchfahren. Hierfür muss festgelegt werden, wie diese Strecke definiert wird und wie die Steuerung des Roboters erfolgen soll. Da die zurückgelegte Strecke anschließend vermessen werden muss, ist es erforderlich, das Bewegungsverhalten des Roboters entsprechend darauf auszuliegen.

A. Steuerbarkeit

Grundsätzlich lassen sich zwei methodisch verschiedene Ansätze zur Steuerung des Roboters entlang einer Strecke unterscheiden: Entweder folgt der Roboter automatisch einer vorher festgelegten Route, oder er wird durch manuelle Eingriffe gelenkt. Die automatische Steuerung bietet den Vorteil, dass die Messergebnisse besser reproduzierbar und

nachvollziehbar sind. Dadurch lassen sich auftretende Probleme und Abweichungen leichter identifizieren und ihren Ursachen zuordnen. Die automatische Steuerung wird daher mittels Linienverfolgung realisiert.

B. Fahrweise

Um während der Fahrt verwertbare Messdaten zu erhalten, muss sich der Roboter nach einem definierten Bewegungsmuster verhalten. Ein geeignetes Schema besteht darin, Phasen der Geradeausfahrt mit Drehungen auf der Stelle zu kombinieren. Auf diese Weise lassen sich die erfassten Daten klar in zurückgelegte Wegstrecken und Drehwinkel unterteilen. Daraus folgt, dass die zu durchfahrende Strecke aus mehreren geradlinigen Segmenten aufgebaut sein muss, die in definierten Winkeln zueinander angeordnet sind.

Die zu verfolgende Linie muss Eigenschaften aufweisen, die es dem Roboter ermöglichen, den Zeitpunkt und die Richtung einer erforderlichen Drehung eindeutig zu identifizieren. Hierfür bietet sich der Einsatz einer dreifarbigem Linie an. Dabei steht eine Farbe für den Mittelbereich der Linie, während die beiden anderen Farben die Bereiche rechts bzw. links vom Mittelbereich kennzeichnen.

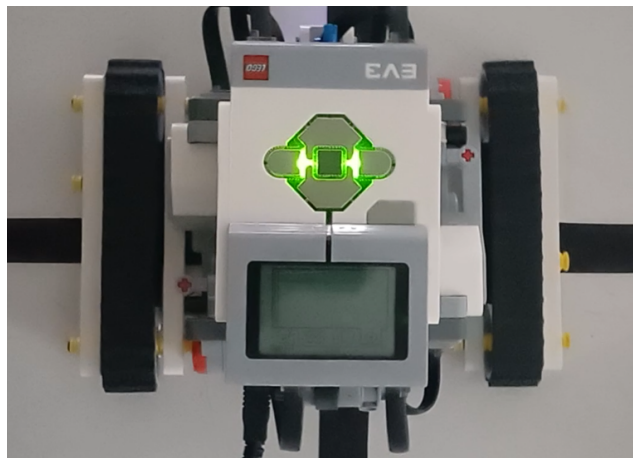


Abbildung 1. Aufbau des Geometer-Roboters mit Kettenantrieb

III. KONSTRUKTION

Aus dem erarbeiteten Konzept ergeben sich drei grundlegende Anforderungen an die mechanische Gestaltung des Roboters. Er muss in der Lage sein, sowohl Geradeausfahrten als auch Drehungen auf der Stelle durchzuführen und gleichzeitig die Farbe der Fahrbahn unterhalb seiner Basis zu erfassen. Um das geforderte Fahrverhalten zu realisieren, kommt ein kettenbasierter Differenzialantrieb zum Einsatz (siehe Abb. 1).

Zur frühzeitigen Erkennung einer bevorstehenden Richtungsänderung ist ein Farbsensor an der Vorderseite des Roboters angebracht. Ein zweiter Sensor ist im Zentrum des Roboters positioniert (siehe Abb. 2). Dessen Aufgabe ist es, dauerhaft die Ausrichtung des Roboters auf den Mittelbereich der Fahrspur zu gewährleisten. Als Zentrum wird hierbei der Punkt definiert, um den der Roboter bei Drehungen auf der Stelle rotiert. Um einen eindeutigen Drehpunkt zu gewährleisten, wurde der Roboter punktsymmetrisch aufgebaut. Dies wird unter anderem durch die entgegengesetzte Ausrichtung der beiden Antriebsmotoren für die linke und rechte Kette erreicht.

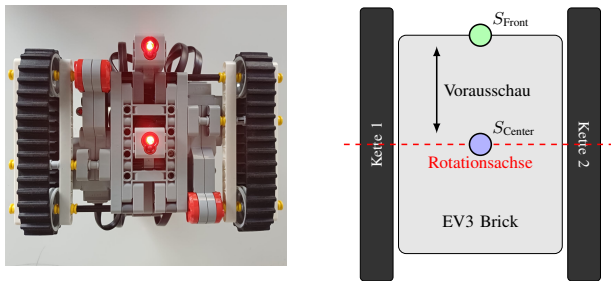


Abbildung 2. Unterseite mit Sensorplatzierung (links) und schematische Ansicht (rechts)

IV. AUFBAU DER STRECKE

In Abschnitt II wurde die Verwendung einer dreifarbigem Linie als Grundlage für die Steuerung festgelegt. Es bleibt zu klären, ob die Auswahl der Farben von Bedeutung ist und welche Breite die einzelnen Streifen aufweisen sollten.

A. Farbwahl

Tests mit dem Farbsensor ergaben Ungenauigkeiten in der Farberkennung. Rot und Grün wurden am zuverlässigsten unterschieden; bei direkt nebeneinanderliegenden roten und grünen Flächen wurde der Übergang jedoch als Gelb interpretiert. Dieses Problem konnte umgangen werden, indem Rot und Grün an den Außenseiten platziert und Weiß für den Mittelbereich festgelegt wurden (siehe Abb. 5). Da Weiß vom Sensor überwiegend als Blau erkannt wurde, wurde im Programm eine Umdeutung implementiert: Blau wird als Weiß gewertet. Zur Markierung des Streckenendes wurde Gelb gewählt, da diese Farbe im gewählten Schema eindeutig identifiziert wird und nicht mit anderen Farbzuständen kollidiert.

B. Streifenbreite

Im Gegensatz zu den seitlichen Streifen ist die Breite des Mittelstreifens von entscheidender Bedeutung, da sie den zulässigen Toleranzbereich für die Positionierung des Roboters festlegt. Zwar besitzt das Sensorgehäuse eine Breite von 2 cm, die aktive Sensorfläche ist mit 0,75 cm jedoch deutlich schmaler. Für eine zuverlässige Erkennung muss der Mittelbereich mindestens diese aktive Fläche abdecken. Um auch bei leichter Schräglage eine stabile Geradeausfahrt über längere Distanzen zu gewährleisten und ein permanentes Nachjustieren bei kleinsten Abweichungen zu vermeiden, ist eine Breite von mehr als 0,75 cm anzustreben. Gleichzeitig

erfordert eine hohe Messgenauigkeit einen möglichst schmalen Streifen. In praktischen Versuchen wurden 2,5 cm als geeignet ermittelt.

V. PROGRAMMIERUNG

Die Aufgabe des Steuerungsprogramms in MATLAB besteht darin, die von den Sensoren erfassten Farbkombinationen auszuwerten, daraus die erforderlichen Fahrmanöver abzuleiten und während der Fahrt die Bewegungsdaten aufzuzeichnen.

A. Interpretation der Farbkombinationen

Da beide Sensoren drei Farben (Weiß, Rot, Grün) unterscheiden können, ergeben sich neun mögliche Kombinationen. Erkennt der mittlere Sensor Rot oder Grün anstelle von Weiß, ist der Roboter dezentriert und führt eine Lagekorrektur durch. Dies trifft auf sechs Farbkombinationen zu. Bei korrekter Zentrierung (mittlerer Sensor auf Weiß) leitet der Frontsensor das eigentliche Fahrmanöver ein: Beide Sensoren auf Weiß bedeuten Geradeausfahrt; erkennt nur der Frontsensor Rot oder Grün, wird eine Kurve eingeleitet (siehe Tabelle I).

Tabelle I
SENSORZUSTÄNDE UND ABGELEITETE FAHRMANÖVER

S_{Center}	S_{Front}	Motor-Aktion
Weiß	Weiß	Geradeausfahrt
Weiß	Grün	Rechtskurve einleiten
Weiß	Rot	Linkskurve einleiten
Grün	Grün	Lagekorrektur nach rechts
Rot	Rot	Lagekorrektur nach links
Gelb	Gelb	Stopp (Ziel erreicht)

B. Rotation

Beim Erkennen einer Farbe (Rot oder Grün) durch den Frontsensor, während der Mittelsensor gleichzeitig Weiß erkennt, wird eine Kurve eingeleitet – Rot signalisiert eine Linkskurve, Grün eine Rechtskurve (siehe Tabelle I). Die Drehung auf der Stelle erfolgt solange, bis der Frontsensor wieder Weiß erfasst (siehe Abb. 3). Da das Drehzentrum (Position des Mittelsensors) den Schnittpunkt der Mittellinien noch nicht erreicht hat, muss vor der Rotation eine Vorwärtsbewegung erfolgen, bis das Drehzentrum über dem Eckpunkt liegt. Aufgrund des unveränderlichen Sensorabstands ist diese Vorlaufstrecke stets identisch und wird als fester Wert im Programm eingefügt; eine dynamische Ermittlung ist obsolet.

C. Lagekorrektur

Bei sechs der neun Farbkombinationen weicht der Roboter von der Sollposition ab und die korrekte Weiterfahrt erfordert eine Lagekorrektur. Wenn der mittlere Sensor nicht Weiß erkennt, unterscheidet die Korrekturstrategie zwei Fälle:

Schräglage – Unterschiedliche Farben an den Sensoren. Der Roboter fährt zunächst geradeaus, bis der Mittelsensor den weißen Mittelstreifen erreicht. Anschließend dreht er sich, bis auch der Frontsensor auf dem Mittelstreifen liegt.

Parallelversatz – Beide Sensoren erkennen die gleiche Farbe. Der Roboter dreht sich zuerst, bis der Frontsensor den Mittelbereich erfasst. Es folgt die Lagekorrektur analog zur Schräglage.

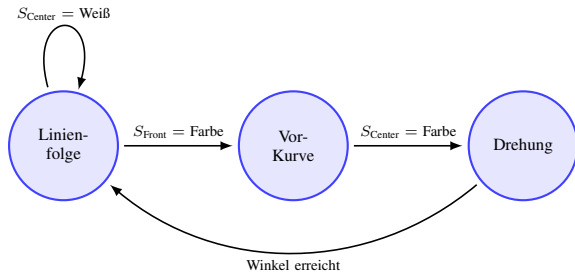


Abbildung 3. Zustandsautomat der Steuerungslogik

D. Aufzeichnung der Bewegungsdaten

Zur Erfassung der zurückgelegten Distanz und der Drehwinkel wird die Funktion `tachoCount()` verwendet, die den aktuellen Zählerstand der integrierten Motordrehgeber im Gradmaß zurückgibt. Vor und nach jeder Aktion (Geradeausfahrt oder Drehung) wird dieser Wert für einen der beiden Motoren ausgelesen. Aus der Differenz der beiden Messungen wird die inkrementelle Motordrehung ermittelt und in einem Array abgelegt, wobei zwischen den beiden Aktionstypen unterschieden wird. Da der Zähler bei Rückwärtsdrehungen dekrementiert wird, haben Rechtsdrehungen das umgekehrte Vorzeichen von Linksdrehungen. Diese Rohdaten bilden die Grundlage für die spätere Pfadrekonstruktion.

VI. INTERPRETATION DER BEWEGUNGSDATEN

Die erfassten Rohdaten repräsentieren die gemessene Motorrotation für jede durchgeführte Aktion, erlauben jedoch keine direkten Rückschlüsse auf die tatsächlich zurückgelegten Streckenlängen oder die Drehwinkel des Roboters. Hierfür müssen zwei Kalibrierungsfaktoren ermittelt werden: das Verhältnis zwischen Motordrehung und zurückgelegter Strecke sowie das Verhältnis zwischen Motordrehung und Roboterdrehung.

A. Motordrehung / Strecke

Zur Bestimmung des Übersetzungsverhältnisses zwischen Motorrotation und Fahrtstrecke wurde der Roboter wiederholt auf einer geraden Teststrecke von 6m gefahren und dabei die Motorrotation aufgezeichnet. Die Auswertung ergab eine systematische Abweichung vom idealen Geradeauskurs: Über die gesamte Distanz trat ein seitlicher Drift von 20 cm auf (siehe Abb. 4). Als wahrscheinliche Ursache kommt eine ungleichmäßige Bodenreibung infolge von Kettenverunreinigungen infrage. Aus den Messreihen wurde der mittlere Drehwinkel pro Meter ermittelt und auf 3300° pro Meter gerundet. Bei einem Validierungsversuch legte der Roboter mit diesem Wert jedoch 1,10 m zurück. Eine zweite Kalibrierung mit 3000° ergab exakt 1 m. Daraus wurde der Faktor 30° pro cm abgeleitet.

B. Motordrehung / Roboterdrehung

Die Bestimmung des Zusammenhangs zwischen Motorrotation und resultierendem Drehwinkel des Roboters erfolgte durch eine Kalibrierungsmessung bei einer vollständigen Eigenrotation. Hierfür wurde der Roboter derart auf einem schwarzen, rechtwinkligen Kreuz positioniert, dass sein Drehzentrum

exakt mit der Kreuzmitte übereinstimmte und der Frontsensor auf einem der vier Kreuzarme lag. Anschließend rotierte der Roboter solange, bis der Frontsensor viermal die Farbe Schwarz detektiert hatte, was einer vollständigen Umdrehung entspricht. Bei jeder Detektion wurde die seit der letzten Markierung aufgelaufene Motordrehung gespeichert, sodass vier Werte für die jeweiligen Vierteldrehungen vorlagen. Dieser Prozess wurde dreimal wiederholt; aus allen Einzelwerten wurde der Mittelwert gebildet, ganzzahlig gerundet und auf eine Volldrehung hochgerechnet. Daraus ergab sich ein Verhältnis von 2240° Motorrotation pro vollständiger Roboterdrehung. Ein Validierungslauf, bei dem der Roboter angewiesen wurde, genau diesen Winkel zu rotieren, bestätigte die korrekte Kalibrierung: Es wurde exakt eine Umdrehung ausgeführt. Mit den beiden ermittelten Faktoren können die erfassten Rohdaten in die entsprechenden physikalischen Größen umgerechnet werden.

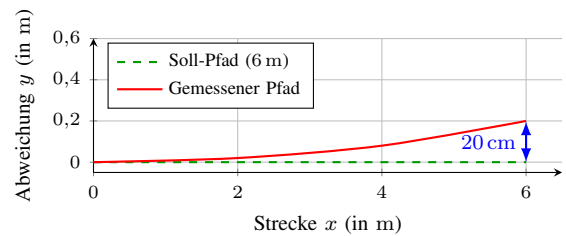


Abbildung 4. Seitliche Abweichung auf einer geraden Strecke von 6 m

VII. DATENAUSWERTUNG

Nach Abschluss der Datenerfassung werden die aufgezeichneten Bewegungsdaten einer mathematischen Transformation unterzogen, um eine grafische Rekonstruktion der gefahrenen Strecke zu ermöglichen. Hierzu werden die einzelnen geradlinigen Segmente der Strecke als Vektoren modelliert, die anschließend zu einer Gesamtstrecke zusammengesetzt werden.

A. Algebraische Transformation

Zur Beschreibung der Bewegung in der Ebene wird ein kartesisches Koordinatensystem eingeführt, dessen Ursprung im Startpunkt des Roboters liegt und dessen x-Achse mit der ersten geradlinigen Teilstrecke zusammenfällt. Alle weiteren Streckensegmente werden zunächst ebenfalls als Vektoren in Richtung der x-Achse betrachtet. Anschließend werden diese Vektoren mittels Rotationsmatrizen um den jeweiligen kumulierten Drehwinkel rotiert, um ihre tatsächliche Orientierung in der Ebene abzubilden. Der kumulierte Drehwinkel ergibt sich aus der vorzeichenbehafteten Summe der zuvor ermittelten Einzeldrehungen. Die Transformation eines beliebigen Segments der Länge Δd mit dem zugehörigen kumulierten Winkel φ lässt sich beschreiben durch (vgl. [1]):

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix} \begin{pmatrix} \Delta d \\ 0 \end{pmatrix}$$

B. Konstruktion des Graphen

Die Rekonstruktion der Gesamtstrecke als Graph erfolgt über die Bestimmung der Koordinaten sämtlicher Knotenpunkte, an

denen die geradlinigen Segmente aufeinandertreffen. Hierzu werden die transformierten Vektoren in der Reihenfolge ihrer Erfassung aufsummiert. Die resultierenden Koordinaten der Knotenpunkte können anschließend mit dem Befehl `plot()` dargestellt werden (siehe Abb. 5).

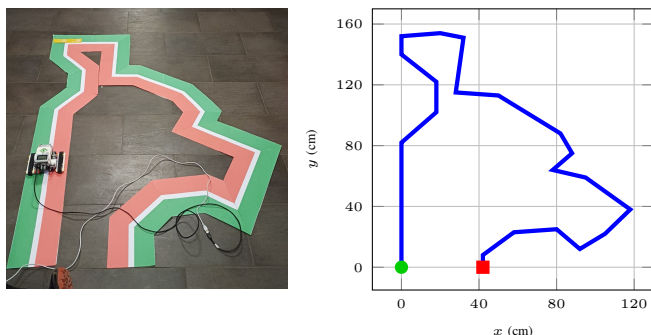


Abbildung 5. Gegenüberstellung Teststrecke vs. Odometrie-Rekonstruktion

VIII. GEOMETRISCHE BERECHNUNGEN

Aus den verarbeiteten Bewegungsdaten lassen sich verschiedene geometrische Kenngrößen der befahrenen Strecke ableiten. Dazu zählen insbesondere der euklidische Abstand zwischen Start- und Endpunkt sowie der Flächeninhalt einer durch die Strecke umschlossenen Fläche.

A. Euklidischer Abstand

Die direkte Entfernung zwischen Anfangs- und Endpunkt wird durch die Länge des Verbindungsvektors bestimmt. Da der Startpunkt definitionsgemäß im Koordinatenursprung liegt, entsprechen die Komponenten dieses Vektors den kartesischen Koordinaten des Endpunktes. Die Berechnung erfolgt über den Satz des Pythagoras (siehe Abb. 6).

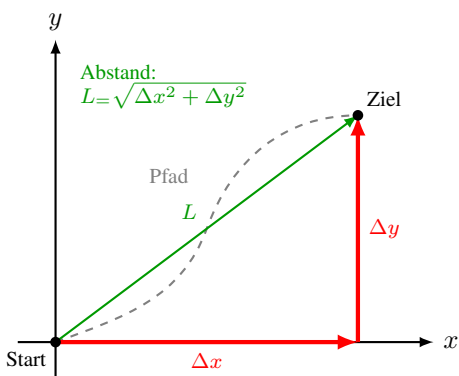


Abbildung 6. Veranschaulichung des euklidischen Abstands L

B. Flächeninhalt

Eine Fläche ist als ein abgeschlossener Bereich definiert, dessen Berandung einen geschlossenen Weg bildet. Da die gefahrene Strecke im Allgemeinen nicht zum Startpunkt zurückführt, liegt kein geschlossener Weg vor. Um dennoch eine Flächenberechnung zu ermöglichen, wird der Weg durch

eine Verbindungsstrecke zwischen Start- und Endpunkt ergänzt, die jedoch nicht eindeutig ist. Dennoch ist eine naheliegende Option die direkte lineare Verbindung beider Punkte (siehe Abb. 6). Der eingeschlossene Flächeninhalt kann so mit der MATLAB-Funktion `polyarea()` berechnet werden.

IX. ERGEBNIS UND DISKUSSION

Zur Validierung der Funktionsfähigkeit und Messgenauigkeit des Roboters wurde eine reale Teststrecke aufgebaut und abgefahren (siehe Abb. 5). Die aus den Odometriedaten rekonstruierte Wegstrecke zeigt eine hohe Übereinstimmung mit der tatsächlichen Geometrie der Teststrecke. Insbesondere die berechnete Distanz von 43 cm zwischen Start- und Endpunkt entspricht dem real gemessenen Wert.

Trotz dieser grundsätzlich guten Genauigkeit sind folgende Einschränkungen zu beachten:

- 1) Bei längeren Geradeausfahrten weicht der Roboter infolge asymmetrischer Bodenreibung von der idealen Linie ab und bewegt sich leicht bogenförmig. Das Programm behandelt diese Bewegung dennoch als geradliniges Segment, was zu einer systematischen Verfälschung der rekonstruierten Geometrie führt.
- 2) Ein Verlassen der farbigen Fahrspur hat zweierlei Konsequenzen: Zum einen werden die erfassten Bewegungsdaten fehlerhaft, zum anderen kann der Roboter die Orientierung verlieren und unkontrollierte Manöver ausführen.
- 3) Fehlerkennungen des Farbsensors, beispielsweise durch Umgebungslicht oder Verschmutzungen, beeinträchtigen den korrekten Ablauf der Zustandsmaschine und können Fehlmanöver auslösen.

X. ZUSAMMENFASSUNG UND FAZIT

Im Rahmen dieser Arbeit wurde ein Messroboter auf Basis des LEGO Mindstorms EV3 und MATLAB entwickelt, der in der Lage ist, eine vorgegebene, farbkodierte Strecke autonom abzufahren und aus den erfassten Bewegungsdaten eine grafische Rekonstruktion des Pfades zu erstellen. Die Realisierung einer zuverlässigen Pfadverfolgung und Datenerfassung gelang durch den Einsatz eines differentiellen Kettenantriebs in Kombination mit zwei Farbsensoren. Die Bestimmung der Fahrposition erfolgte odometrisch über die Auswertung der Motorrotationswinkel. Eine Kalibrierung der Motordrehung erlaubte die Umrechnung der Rohdaten in Streckenlängen und Drehwinkel. Ein Vergleich der rekonstruierten Strecke mit der realen Teststrecke bestätigte die hohe Genauigkeit des Gesamtsystems. Das Projekt demonstriert, dass sich mit vergleichsweise einfachen technischen Möglichkeiten komplexe Aufgabenstellungen aus den Bereichen der Landvermessung und digitalen Kartierung erfolgreich umsetzen lassen.

LITERATURVERZEICHNIS

- [1] WIKIPEDIA, THE FREE ENCYCLOPEDIA: *Drehmatrix*. <https://de.wikipedia.org/wiki/Drehmatrix>. Version: Dezember 2025