

# GEOMETER – Der Landvermesser-Roboter

Yannick Christopher Kastens, Elektrotechnik/Informationstechnik  
Otto-von-Guericke-Universität Magdeburg

**Zusammenfassung**—In diesem Beitrag wird der „Geometer“ vorgestellt, ein differentiell angetriebener LEGO-Mindstorms-EV3-Roboter, der als linienfolgender Vermessungsroboter fungiert. Ziel der Arbeit ist es, dass der Roboter eine dreifarbige Linie selbstständig abfährt, seine Bewegung odometrisch erfasst und die gefahrene Strecke mathematisch rekonstruiert. Dadurch können unterschiedliche Strecken und Flächen genau vermessen werden und Abstände bestimmt werden.

Die Steuerung erfolgt vorausschauend mithilfe von MATLAB über zwei Farbsensoren, aus denen die notwendigen Fahrmanöver abgeleitet werden. Die Odometrie erfolgt über die Motordrehwinkel, die nach Kalibrierung in reale Strecken- und Winkeldaten umgerechnet werden. Die Auswertung zeigt, dass der Roboter einfache geometrische Strecken präzise rekonstruieren kann. Die Arbeit demonstriert, dass auch mit einfacher LEGO-Hardware komplexe Vermessungsaufgaben umsetzbar sind.

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

## I. GRUNDIDEE

Die autonome Vermessung von Strecken und Flächen ist eine grundlegende Aufgabe in der Robotik mit vielfältigen Anwendungen, von der Kartierung bis zur Qualitätskontrolle. Leicht zugängliche Plattformen wie LEGO Mindstorms und MATLAB ermöglichen es, zentrale Konzepte der Odometrie und Pfadrekonstruktion praktisch zu erproben. Ziel dieses Projekts ist die Entwicklung eines Roboters, der eine vorgegebene Strecke abfährt, seine Bewegung odometrisch erfasst und daraus ein mathematisches Abbild der Strecke erstellt, um geometrische Kenngrößen wie Abstände und Flächeninhalte zu berechnen.

## II. KONZEPT

Der Roboter soll eine bestimmte Strecke abfahren. Dafür muss entschieden werden, wie diese Strecke festgelegt bzw. wie der Roboter gesteuert werden kann. Da die gefahrene Strecke auch vermessen werden soll, sollte das Bewegungsmuster des Roboters daran angepasst sein.

### A. Steuerbarkeit

Es gibt zwei grundlegende Prinzipien, nach denen der Roboter eine gewisse Strecke entlang gesteuert werden kann: Entweder fährt der Roboter automatisch eine vordefinierte Strecke ab oder er wird manuell gesteuert. Wenn der Roboter automatisch gesteuert wird, hat das den Vorteil, dass das Ergebnis der Vermessungen des Roboters genauer reproduziert und überprüft werden kann. Potenzielle Probleme und Fehler können so besser erkannt und einfacher auf ihre jeweiligen Ursachen zurückgeführt werden. Eine mögliche Umsetzung ist, den Roboter einer Linie folgen zu lassen.

### B. Fahrweise

Damit während der Fahrt brauchbare Bewegungsdaten erhoben werden, sollte der Roboter sich möglichst strukturiert bewegen. Ein geeignetes Bewegungsprofil ist geradliniges Fahren kombiniert mit einer Drehung auf der Stelle. Dadurch lassen sich die Bewegungsdaten eindeutig in Streckenlänge und Drehwinkel aufteilen. Das bedeutet, dass die Strecke aus einzelnen geradlinigen Teilen bestehen muss, die zueinander in einem bestimmten Winkel stehen.

Die abzufahrende Linie muss derart beschaffen sein, dass der Roboter erkennt, wann er sich in welche Richtung drehen soll. Dafür eignet sich eine dreifarbige Linie. Eine Farbe steht dabei für die Mitte der Linie, eine Farbe für rechts der Mitte und eine Farbe für links der Mitte.

## III. KONSTRUKTION

Das bisherige Konzept stellt folgende drei Anforderungen an die Konstruktion des Roboters: Er kann geradeaus fahren, er kann sich auf der Stelle drehen und er kann die Farbe der Linie unter ihm erkennen. Für die passende Fahrweise wird ein differentieller Kettenantrieb verwendet (siehe Abb. 1).

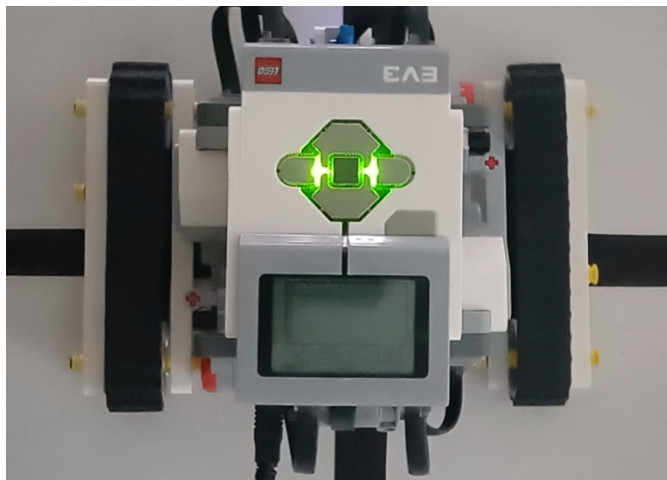


Abbildung 1. Aufbau des Geometer-Roboters mit Kettenantrieb

Damit der Roboter erkennt, welche Farbe vor ihm liegt, also eine vorausliegende Kurve vorzeitig erkennt, hat er einen Farbsensor an der Front. Um sicherzustellen, dass sich der Roboter selbst stets in der Mitte der Linie befindet, hat er einen zweiten Sensor in seiner Mitte (siehe Abb. 2). Als Mitte wird hier das Drehzentrum des Roboters bezeichnet. Damit dieses Zentrum eindeutig ist, ist der Roboter punktsymmetrisch gebaut; die zwei Motoren für den jeweils links- und rechtsseitigen Antrieb sind dafür entgegengesetzt orientiert.

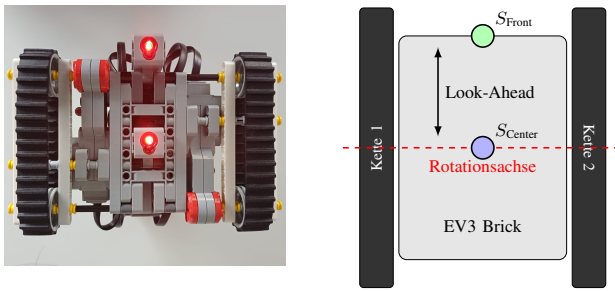


Abbildung 2. Unterseite mit Sensorplatzierung (links) und schematische Ansicht (rechts)

#### IV. AUFBAU DER STRECKE

Bisher festgelegt ist, dass die Linie dreifarbig sein soll. Zu klären bleibt, ob die konkrete Farbwahl entscheidend ist und welche Breite die einzelnen Streifen haben sollten.

##### A. Farbwahl

Beim Test des Farbsensors zeigte sich, dass dieser nicht immer die korrekte Farbe erkannte. Am zuverlässigsten erkannte er die Farben Rot und Grün; lagen Rot und Grün aber nebeneinander, erkannte der Sensor auf der Grenze Gelb. Dieses Problem wurde gelöst, indem Rot und Grün jeweils außen positioniert wurden und für den mittleren Streifen Weiß gewählt wurde. Die Farbe Weiß erkannte der Sensor oft als Blau, deswegen wird die Farbe Blau im Programm als Weiß interpretiert. Es wurde festgelegt, dass der rechte Streifen rot und der linke Streifen grün sein soll. Um das Ende der Strecke zu markieren, wird die Farbe Gelb verwendet, da sie in der gewählten Anordnung nicht fälschlicherweise erkannt wird.

##### B. Streifenbreite

Im Gegensatz zur Breite der äußeren Streifen ist die genaue Breite des mittleren Streifens elementar, da sie bestimmt, wie weit der Roboter von der tatsächlichen Mitte der Linie abweichen darf. Der Farbsensor ist 2 cm breit; es ist daher sinnvoll, dass der mittlere Streifen mindestens ebenso breit ist. Damit der Roboter auch in leicht schiefer Lage zur Linie eine längere gerade Strecke fahren kann und sich nicht durchgehend bei minimalen Abweichungen selbst korrigiert, sollte die Linie breiter als 2 cm sein. Damit die Messergebnisse möglichst präzise sind, sollte der Streifen möglichst schmal sein. In Tests erwies sich eine Streifenbreite von 2,5 cm als geeignet.

#### V. PROGRAMMIERUNG

Das Steuerungsprogramm wird in MATLAB umgesetzt. Es muss die Farbkombinationen der beiden Sensoren interpretieren und den entsprechenden Bewegungsablauf ausführen, gleichzeitig muss es die Bewegungsdaten erfassen.

##### A. Interpretation der Farbkombinationen

Da beide Sensoren jeweils drei unterschiedliche Farben auf der Linie erkennen können, sind insgesamt neun Farbkombinationen möglich. Erkennt der mittlere Sensor eine Farbe und nicht Weiß, ist der Roboter falsch positioniert

und muss seine Position korrigieren; das trifft auf sechs der neun Farbkombinationen zu. Ist der Roboter hingegen richtig positioniert, erfolgt die eigentliche Lenkung; das ist für die übrigen drei Farbkombinationen der Fall. Erkennen beide Sensoren Weiß, fährt der Roboter geradeaus.

Tabelle I  
SENSORZUSTÄNDE UND ABGELEITETE FAHRMANÖVER

$S_{Center}$	$S_{Front}$	Motor-Aktion
Weiß	Weiß	Geradausfahrt
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

Erkennt nur der vordere Sensor eine Farbe, der mittlere aber Weiß, soll der Roboter abbiegen – nach links bei Rot und nach rechts bei Grün. Der Roboter muss sich dafür so lange drehen, bis der vordere Sensor wieder Weiß erkennt (siehe Tabelle I). Da der Roboter auf der Stelle rotiert, darf er sich nicht unmittelbar nach Erkennen der Farbe drehen; andernfalls wäre er nach Abschluss der Drehung nicht in einer Linie mit der nächsten Teilstrecke, da das Drehzentrum über dem mittleren Sensor liegt, welcher sich zu diesem Zeitpunkt noch nicht auf dem Treffpunkt der Teilstrecken befindet. Stattdessen muss der Roboter noch so lange weiterfahren, bis das Drehzentrum genau auf der Ecke liegt, also dem Treffpunkt der Mittellinien der zusammenlaufenden Teilstrecken. Da beide Sensoren einen festen Abstand zueinander haben, ist diese zusätzliche Strecke immer gleich lang; sie kann im Programm festgelegt werden und muss nicht situativ ermittelt werden (siehe Abb. 3).

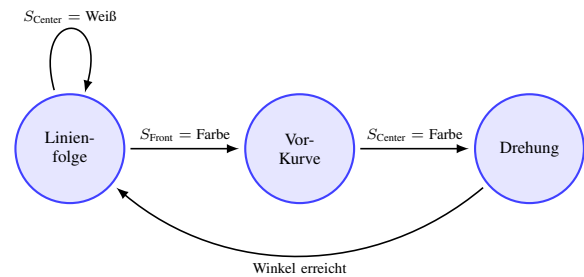


Abbildung 3. Zustandsautomat der Steuerungslogik

##### C. Lagekorrektur

Wird eine der anderen sechs Farbkombinationen erkannt, muss sich der Roboter neu positionieren. Erkennen beide Sensoren unterschiedliche Farben, steht der Roboter schief zur Strecke. In diesem Fall fährt der Roboter so lange geradeaus, bis der Mittelsensor den Mittelstreifen erreicht hat. Dann dreht er sich zur Mitte, bis der Frontsensor ebenfalls auf dem Mittelstreifen liegt. Erkennen beide Sensoren die gleiche Farbe, liegt der Roboter parallel zum Mittelstreifen; er muss sich daher zuerst zur Mitte drehen, bis der Frontsensor den Mittelstreifen erreicht hat.

#### D. Aufzeichnung der Bewegungsdaten

Um zu ermitteln, welche Distanz und welchen Winkel der Roboter zu welchem Zeitpunkt zurückgelegt hat, kann der Befehl `tachoCount` verwendet werden. Dieser Befehl gibt in seiner Grundeinstellung im Gradmaß aus, wie viel sich ein Motor seit dem letzten Zurücksetzen gedreht hat. Um daraus Bewegungsdaten zu generieren, wird bei jeder Aktion des Roboters diese Zahl für einen der beiden Motoren sowohl davor als auch danach erhoben. Aus beiden Werten wird die Differenz ermittelt, die dann in Reihenfolge des Bewegungsablaufs in einem Array gespeichert wird. Dabei wird zwischen Drehungen und Geradeausbewegung unterschieden. Da bei einer Rückwärtsdrehung des Motors zurückgezählt wird, ist eine Rechtsdrehung in der Differenz vorzeichenumgekehrt zu einer Linksdrehung.

### VI. INTERPRETATION DER BEWEGUNGSDATEN

Die erhobenen Daten geben an, wie viel sich der gewählte Motor bei einer Aktion gedreht hat; für sich genommen geben diese noch keine Auskunft über die Länge der einzelnen Teilstrecken bzw. die Größe der jeweiligen Winkel der Roboterdrehungen. Hierzu müssen die Verhältnisse zwischen Motordrehung und zurückgelegter Strecke bzw. zwischen Motordrehung und Roboterdrehung ermittelt werden.

#### A. Motordrehung und Strecke

Um das Verhältnis zwischen Motordrehung und Strecke zu ermitteln, fährt der Roboter eine bestimmte Strecke, während die Motordrehung gemessen wird; die Länge der Strecke und die entsprechende Motordrehung werden dann zueinander ins Verhältnis gesetzt. Die Teststrecke war 6 m lang; es wurden mehrere Messungen auf dieser Strecke durchgeführt. Es stellte sich heraus, dass der Roboter auf längeren Strecken nicht in einer geraden Linie fuhr, sondern in einem Bogen; auf 6 m gerader Strecke zeigte sich eine seitliche Abweichung von 20 cm (siehe Abb. 4). Vermutlich geht diese Abweichung auf Verunreinigungen der Ketten zurück, die auf beiden Seiten zu unterschiedlicher Bodenreibung führen.

Die gemessenen Werte wurden jeweils zu Motordrehwinkel je gefahrenem Meter heruntergerechnet und der Durchschnitt aller Messergebnisse berechnet. Auf Zehner gerundet ergaben die Messungen einen Durchschnitt von  $3300^\circ$  je m. Dieser Wert wurde anschließend überprüft, indem der Roboter angewiesen wurde, genau  $3300^\circ$  Motordrehung auf gerader Strecke zurückzulegen. Der Roboter ist dabei 10 cm weiter als 1 m gefahren. Daher wurde ein zweiter Test mit einem entsprechend angepassten Wert von  $3000^\circ$  durchgeführt; dabei legte Roboter 1 m zurück. Daraus wurde abgeleitet, dass eine  $30^\circ$  Motordrehung einer Strecke von 1 cm entspricht.

#### B. Motordrehung und Roboterdrehung

Das Verhältnis zwischen Motordrehung und Roboterdrehung lässt sich bestimmen, indem die Motordrehung bei einer Volldrehung des Roboters gemessen wird. Hierzu wird der Roboter auf einem schwarzen, rechtwinkligen Kreuz so positioniert, dass das Drehzentrum auf der Mitte liegt und der Frontsensor

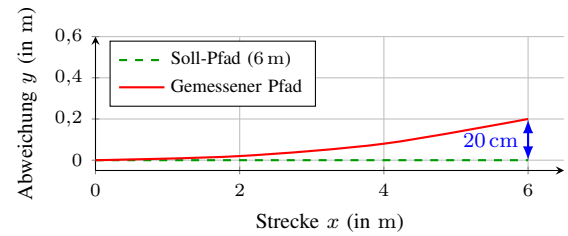


Abbildung 4. Veranschaulichung der seitlichen Abweichung auf einer geraden Strecke von 6 m

auf einem der vier Arme. Der Roboter soll sich nach dem Start so lange drehen, bis der Frontsensor das vierte Mal schwarz erkennt; er sollte dann eine volle Drehung durchgeführt haben. Jedes Mal, wenn der Sensor schwarz erkennt, speichert er die zwischen zwei Markierungen durchgeführte Motordrehung. Das bedeutet, dass er vier Werte speichert, die jeweils einer Vierteldrehung entsprechen.

Der Test wurde drei Mal durchgeführt und es wurde der Durchschnitt aus allen gemessenen Werten berechnet. Dieser wurde auf ganze Zahlen gerundet und auf volle Drehungen hochgerechnet, was ein Verhältnis von  $2240^\circ$  je Volldrehung ergab. Zur Überprüfung des Messwertes lässt man den Roboter diesen Winkel als Drehung zurücklegen; der Wert ist korrekt, wenn genau eine Volldrehung zurückgelegt wird.

Der Test bestätigte, dass  $2240^\circ$  Motordrehung einer vollständigen Rotation entspricht. Die Bewegungsdaten des Roboters können durch die beiden ermittelten Verhältnisse nun umgewandelt werden. Längendaten werden in cm umgerechnet, indem sie durch das gemessene Verhältnis von  $30^\circ$  je cm geteilt werden; Winkeldaten werden ins Bogenmaß umgewandelt, indem sie durch das ermittelte Verhältnis von  $2240^\circ$  je  $2\pi$  rad geteilt werden.

### VII. DATENAUSWERTUNG

Nachdem der Roboter die vorgegebene Strecke abgefahren hat, werden die Bewegungsdaten mathematisch derart transformiert, dass die Strecke grafisch dargestellt werden kann. Dazu werden Länge und Richtung der geradlinigen Teilstrecken jeweils durch einen Vektor repräsentiert und anschließend zu einem Gesamtstreckengraphen zusammengefügt.

#### A. Algebraische Transformation

In der Ebene der gefahrenen Strecke wird ein Koordinatensystem wie folgt definiert: Der Koordinatenursprung entspricht dem Startpunkt, ab dem der Roboter seine Fahrt beginnt; die erste geradlinige Teilstrecke ab diesem Punkt liegt auf der  $x$ -Achse. Zunächst werden alle Teilstrecken als Vektoren entlang der  $x$ -Achse dargestellt. Diese müssen jeweils um den Winkel rotiert werden, in dem sie zur  $x$ -Achse stehen. Dieser Winkel ergibt sich durch vorzeichenbehaftete Kumulation der einzelnen Drehwinkel. Die Vektoren werden mit der Rotationsmatrix des entsprechenden Winkels multipliziert und dadurch rotiert (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

Zur Konstruktion des Gesamtstreckengraphen werden die Eckpunkte der Strecke, an denen die Teilstrecken jeweils zusammenlaufen, in ein Koordinatensystem eingezeichnet und der Reihenfolge nach mit Linien verbunden. Dazu werden die einzelnen rotierten Vektoren kumuliert. Daraus ergeben sich die Koordinaten der Eckpunkte, die mit dem Befehl `plot` dargestellt werden können (siehe Abb. 5).

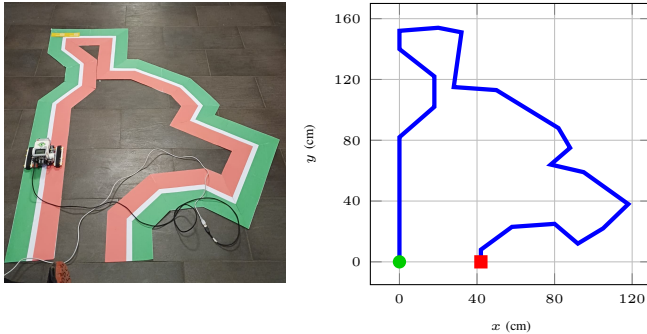


Abbildung 5. Gegenüberstellung Teststrecke vs. Odometrie-Rekonstruktion

### VIII. GEOMETRISCHE BERECHNUNGEN

Aus den verarbeiteten Daten lassen unterschiedliche geometrische Eigenschaften der Strecke bestimmen, beispielsweise der euklidische Abstand zwischen Start und Endpunkt sowie die durch die Strecke eingeschlossene Fläche.

#### A. Euklidischer Abstand

Der Abstand zwischen Start und Endpunkt lässt sich ermitteln, indem der Betrag des Verbindungsvektors bestimmt wird. Da der Startpunkt mit dem Koordinatenursprung identisch ist, entsprechen die Komponenten des Verbindungsvektors den Koordinaten des Endpunktes (siehe Abb. 6).

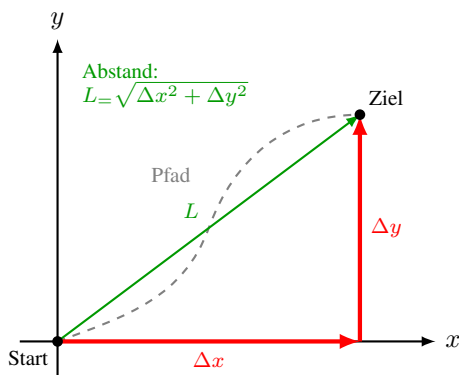


Abbildung 6. Veranschaulichung des euklidischen Abstands  $L$

#### B. Flächeninhalt

Eine Fläche ist ein abgeschlossener Bereich, der durch einen geschlossenen Weg begrenzt wird. Da nicht festgelegt ist, dass der Roboter zum Startpunkt zurückkehrt, bildet die gefahrene Strecke im Allgemeinen keinen geschlossenen Weg.

Daher existiert allgemein keine Begrenzung, die eine Fläche definiert. Der Weg wird geschlossen, indem er durch eine Verbindungsstrecke zwischen Start- und Endpunkt ergänzt wird. Dieser Abschluss ist nicht eindeutig, da unterschiedliche Verbindungen möglich sind. Eine naheliegende Möglichkeit ist die direkte lineare Verbindung. Nach dem Schließen des Weges, lässt sich die Größe der eingeschlossenen Fläche mithilfe der Funktion `polyarea` berechnen.

### IX. ERGEBNIS UND DISKUSSION

Um die Funktionsfähigkeit und Präzision des Programms zu überprüfen, wurde eine Teststrecke gebaut, die der Roboter vermessen sollte (siehe Abb. 5). Das Ergebnis ist eine präzise grafische Rekonstruktion der Strecke. Insbesondere der im Model ermittelte Abstand von 43 cm zwischen Start- und Endpunkt entspricht der Realität. Trotz der hohen Genauigkeit im Test gibt es Einschränkungen:

- 1) Der Roboter kann auf längeren Strecken nicht geradeaus fahren, sondern weicht leicht zur Seite ab; das Programm interpretiert diesen Bogen dennoch als gerade Strecke.
- 2) Kommt der Roboter von der Strecke ab, sind einerseits die erhobenen Daten verfälscht, andererseits besteht das Risiko, dass der Roboter sich unkontrolliert bewegt.
- 3) Erkennt der Farbsensor eine falsche Farbe, wird der Bewegungsablauf des Roboters gestört.

### X. ZUSAMMENFASSUNG UND FAZIT

Im Rahmen dieser Arbeit wurde ein Messroboter auf Basis des LEGO Mindstorms EV3 entwickelt, der eine vordefinierte Strecke eigenständig abfahren und anschließend grafisch rekonstruieren kann. Durch die Verwendung eines differentiellen Kettenantriebs kombiniert mit zwei Farbsensoren konnte eine zuverlässige Pfadverfolgung und Datenerfassung realisiert werden.

Die Fahrposition wurde mittels Odometrie aus den Motordaten bestimmt. Durch die Ermittlung der Verhältnisse zwischen Motordrehung und zurückgelegter Strecke bzw. Rotationswinkel ließen sich genaue Bewegungsdaten ableiten, durch die die Strecke rekonstruiert werden konnte, woraus ihre geometrischen Eigenschaften bestimmt wurden.

Der Vergleich von der realen Teststrecke mit dem daraus rekonstruierten Pfad bestätigte die Genauigkeit des Roboters. Insgesamt zeigt das Projekt, dass sich mit vergleichsweise einfachen technischen Mitteln komplexe Aufgaben der Landvermessung und digitalen Kartierung erfolgreich umsetzen lassen.

### LITERATURVERZEICHNIS

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