

Hough-Transformation

Silas Ueberschaer

26. März 2021

1 Einleitung

Immer mehr Autos fahren autonom und können ohne die Hilfe des Fahrers perfekt einparken. Straßen und Parklücken werden automatisch erkannt. Ein weiteres Beispiel sind Flugzeuge, bei denen Landebahnen in Echtzeit erfasst werden. Der Anteil der Fahrzeuge, die autonom fahren können, liegt im Jahr 2020 zwar nur bei 2,4 Prozent, soll jedoch möglicherweise im Jahr 2050 schon bei 70 Prozent liegen [1]. Doch wie kann aus einem einfachen Bild eine Straße oder eine Landebahn erkannt werden?

Bilder bestehen aus extrem vielen Daten. Die Hough-Transformation hilft dabei aus einem Kantenbild vorhandene Linien zu erkennen, um zum Beispiel erkennen zu können, wo eine Straße anfängt und aufhört.

Das Ziel dieser Arbeit ist es, die Hough-Transformation zu erklären und für die Leser:innen verständlich zu vermitteln. Es wird auf die Entstehung, die Theorie, so wie die Umsetzung eingegangen.

2 Erfindung der Hough-Transformation

Am 18. Dezember 1962 meldete Paul V. C. Hough ein Patent mit dem Titel "Method And Means for Recognizing Complex Patterns" [3] an. Das ungefähr zwei Seiten lange Patent gibt Möglichkeiten an, Punkte genauer zu beschreiben. Im gesamten Patent lassen sich keine mathematischen Formel finden, stattdessen beinhaltet es Abbildungen, die die Idee hinter dem Verfahren erklären sollen. Hough war sich dem Problem bewusst, dass die normale Geradenfunktion

$$y = a \cdot x + b \tag{1}$$

nicht funktioniert, um Linien in Bildern zu finden. Das hängt damit zusammen, dass das Gleichstellen von Punkten die auf einer Vertikalen Linie liegen nicht möglich ist, da die Gleichung gegen unendlich gehen würde (dazu mehr in Abschnitt 3). Im Jahr 1969 veröffentlichte Informatiker Azriel Rosenfeld das Buch "Picture Processing by Computer" [6]. Dieses handelt von der Verarbeiten und Analyse von Bildern und bezieht sich auf das Patent von Paul Hough. Drei Jahre später, 1972, brachten die Informatiker Richard O. Duda und Peter E.

Hart einen Artikel mit dem Titel "Use of the Hough transformation to detect lines and curves in pictures" in der Association for Computing Machinery (ACM) Zeitschrift raus [2]. Dort wurde erstmals die Hessesche Normalform

$$\rho = x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta) \quad (2)$$

im Zusammenhang mit Bildanalyse benutzt. Ausgehend von dieser Formel wird die heutzutage bekannte Hough-Transformation verwendet, um Linien in Bildern zu erkennen.

3 Linien zwischen zwei Punkten berechnen

Um die Linie zwischen zwei Punkten zu berechnen, kann die Geradengleichung beider Punkte errechnet und gleichgesetzt werden. Sind die Punkte $a = (a_x, a_y)$ und $b = (b_x, b_y)$ gegeben, ergeben sich eingesetzt in die Geradengleichung die Funktionen

$$a_y = a_x \cdot a + b \text{ und } b_y = b_x \cdot a + b. \quad (3)$$

Um den Schnittpunkt beider Geraden zu errechnen, werden die Gleichungen nach b aufgelöst, gleichgesetzt und anschließend nach a aufgelöst.

$$a = (b_y - a_y) / (a_x - b_x). \quad (4)$$

Um b zu erhalten, kann nun a in die Anfangsfunktion eingesetzt werden.

$$b = -a_x \cdot ((b_y - a_y) / (a_x - b_x)) + a_y. \quad (5)$$

Werden nun beide Variablen a und b in die Funktion (1) eingesetzt, ergibt sich die Geradengleichung der Gerade, die beide Punkte schneidet

$$y = ((b_y - a_y) / (a_x - b_x)) \cdot x + (-a_x \cdot ((b_y - a_y) / (a_x - b_x)) + a_y). \quad (6)$$

Diese Variante der Linienerkennung zwischen zwei Punkten ist mathematisch leicht lösbar. Jedoch entsteht ein Problem, wenn die gefundene Gerade vertikal verlaufen würde. In diesem Fall wären die x Werte beider Punkte gleich und die Geradenfunktion würde eine Division durch Null erhalten (siehe Funktion 4). Diese Division würde gegen unendlich gehen und würde sich folglich nicht mit der normalen Geradenfunktion darstellen lassen. Um dieses Problem zu umgehen, kann stattdessen Funktion (2) genutzt werden. Diese enthält keine Division, wodurch sich auch vertikale Linien darstellen lassen.

4 Linien Erkennung mit der Hough Transformation

Um die Vorgangsweise der Hough-Transformation zu verstehen, ist es hilfreich die einzelnen Teile und Zeichen der Funktion zu verstehen. ρ (Rho) beschreibt

jeweils den Abstand vom Ursprung (0,0) zu einer Linie durch den Punkt. Hierbei ist ausschlaggebend, dass der Winkel zwischen der Linie und ρ 90° (Lot) beträgt. θ (Theta) beschreibt den Winkel zwischen ρ und der x-Achse.

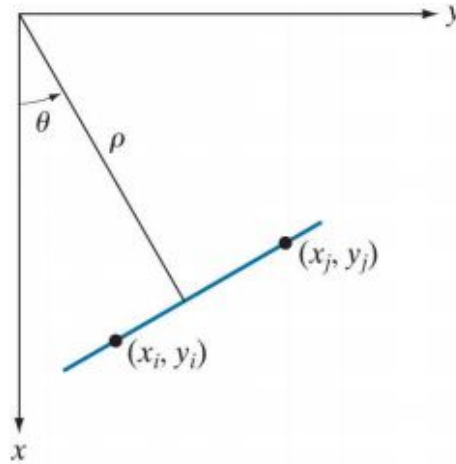


Abbildung 1: Darstellung von ρ und θ [5]

Diese beiden Werte reichen aus, um die Gerade an dem Punkt X zu beschreiben. Um nun die Gerade zwischen zwei Punkten zu ermitteln, wird die Hough-Transformation angewandt. Es wird jedes ρ für jedes θ von dem ausgewählten Punkt berechnet. In den meisten Fällen wird θ auf den Bereich von -90° bis 90° beschränkt. Jedes in diesem Bereich gefundene ρ wird in die sogenannte $\rho\theta$ -plane eingetragen (Abbildung 2). Dadurch entsteht eine Sinuskurve für jeden Punkt, an dem die Transformation angewandt wurde.

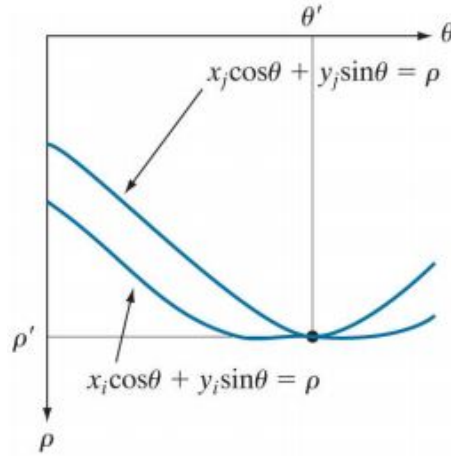


Abbildung 2: $\rho\theta$ -plane mit Beispielkurven [5]

Der Schnittpunkt beider Kurven gibt uns jeweils den ρ und θ Wert der Gerade, die zwischen den beiden Punkten verläuft. Soll dies mathematisch errechnet werden, können die Gleichungen der beiden Punkte gleichgesetzt und nach θ umgestellt werden. Während dies bei zwei Punkten noch leicht umsetzbar ist, wird die Berechnung bei vielen Punkten wesentlich komplexer. Deshalb wird ein neues Koordinatensystem, der Accumulator, erstellt (Abbildung 3b). Dieser ist eingeteilt in kleine Zellen. Genau wie die $\rho\theta$ -plane bildet er die Sinuskurven für die Punkte ab. Der Unterschied liegt darin, dass im Accumulator der Wert einer Zelle um eins erhöht wird, wenn eine Kurve durch diese verläuft. Dies ermöglicht es, die Anzahl der Kreuzungen der Sinuskurven zu zählen. Durch das Einführen eines Schwellwerts kann so bestimmt werden, ab welchem Wert in dem Accumulator eine Linie erkannt werden soll. Das Einführen eines Schwellwerts ist notwendig, da nicht jeder Schnittpunkt von zwei Sinuskurven eine Linie im Bild darstellt.

Je höher der Wert einer Zelle im Accumulator ist, desto mehr Überschneidungen von Kurven gibt es und desto mehr Pixel liegen auf dieser Gerade. An der Stelle $\theta = 45^\circ$ $\rho = 0$ und $\theta = -45^\circ$ $\rho = 71$ des Beispiels in Abbildung 3, befinden sich die Zellen mit den höchsten Wert innerhalb des Accumulators. Diese Werte repräsentieren die diagonalen Linien in der Abbildung 3c. Je nachdem, ob der Schwellwert gleich drei oder zwei ist, können nun zwei oder sechs Linien erkannt werden. Zusammenfassend lässt sich die Transformation in vier Schritte unterteilen.

1. Es wird ein Kantenbild zum Eingabebild erstellt.
2. Für jeden Pixel im Kantenbild wird die Sinuskurve bestimmt und im Accumulator eingetragen.
3. Die ρ und θ Werte werden von dem Accumulator abgelesen. Sie entsprechen den Zellen, in denen der Wert größer oder gleich dem Schwellwert ist.
4. Mit den erhaltenen Werten werden Linien erstellt.

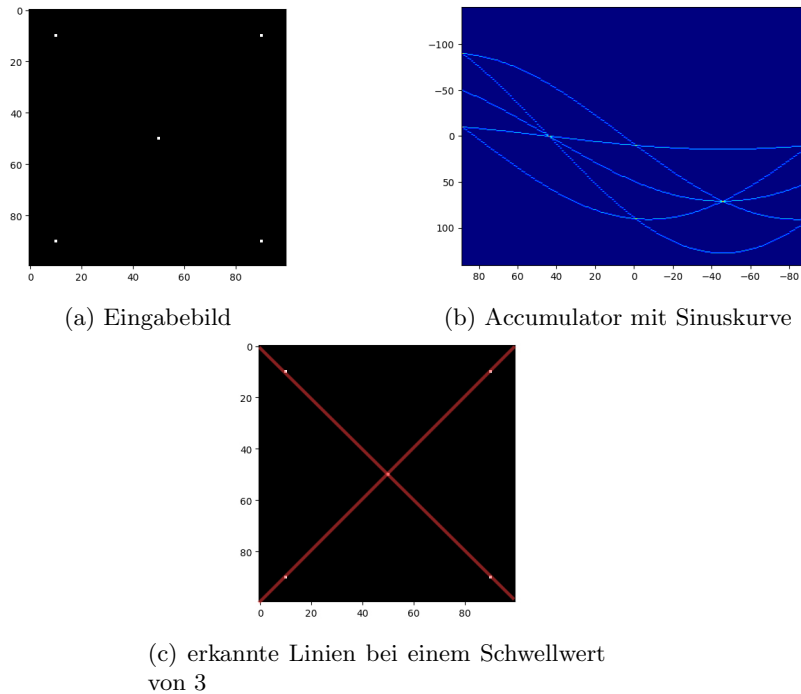


Abbildung 3: Beispiel Anwendung der Hough-Transformation

Ein Problem welches entsteht, ist, dass jede erkannte Linie unendlich lang ist. Aus den gefundenen ρ und θ lässt sich zwar eine Gerade bilden, aber weder Start- noch Endpunkt sind bekannt. Je nach Anwendungsgebiet ist dieses Problem mal weniger und mal mehr problematisch. Eine Möglichkeit dieses Problem zu lösen, ist die Progressive Probabilistic Hough Transformation (PPHT).

5 Progressive Probabilistic Hough Transformation

Das Ziel der PPHT ist grundsätzlich, wie bei der normalen Hough Transformation, das Finden von Linien in Kantenbildern. Zusätzlich dient die PPHT jedoch auch dem Finden von Start- und Endpunkt und dem Reduzieren von Rechenzeit. Das Berechnen von ρ von 180 (-90 bis 90) θ für jedes Pixel im Bild ist rechenaufwendig. Die Idee hinter der PPHT, ist die zufällige Auswahl an Pixeln, anstelle das Berechnens jedes einzelnen Pixels. Hierzu wird zunächst zufällig ein Pixel bestimmt, die dazugehörige Sinuskurve berechnet und in den Accumulator eingetragen. Dies geschieht so lange, bis eine Zelle im Accumulator einen höheren Wert annimmt, als den vorher bestimmten Schwellwert. Ist dies der Fall, wird im Kantenbild an der gefundenen Gerade nach dem ersten

und letzten Pixel auf der Gerade gesucht. Dann werden alle Pixel die auf der Geraden liegen vom Kantenbild entfernt. Durch das Ablaufen der Gerade kann nun Start- und Endpunkt der Gerade ermittelt werden. Der Algorithmus kann mit folgenden Schritten dargestellt werden [4]:

1. Es wird geprüft, ob das Kantenbild leer ist. Ist dies der Fall, wird der Algorithmus beendet.
2. Ein zufälliger Pixel wird ausgewählt, die Sinuskurve berechnet und der Accumulator mit den entsprechenden Werten aktualisiert.
3. Das ausgewählte Pixel wird aus dem Kantenbild entfernt.
4. Es wird geprüft, ob der höchste Wert im Accumulator höher ist, als der Schwellwert. Ist dies der Fall, wird eine Linie erstellt. Ist dies nicht der Fall, wird wieder Schritt 1 ausgeführt.
5. Die Linie wird mit den gefundenen ρ und θ des Accumulator konstruiert. Anhand der Linie werden im Kantenbild alle Pixel gesucht, die auf dieser Linie liegen und deren Abstand nicht größer ist als erlaubt.
6. Alle Pixel auf der Gerade werden im Kantenbild entfernt.
7. Für jeden der gefundenen Pixel wird die Sinuskurve aus dem Accumulator entfernt.
8. Ist die gefundene Linie länger als das angegebene Minimum, wird sie zum Ergebnisbild hinzugefügt.
9. Ist der Algorithmus durchgelaufen, fängt er erneut bei Schritt 1 an.

Ein Problem, welches die PPHT hat, ist, dass das Ergebnis nicht mehr konstant ist. Durch das Entfernen von Pixeln im Kantenbild kann es bei der Kreuzung zweier Gerade dazu führen, dass eine von ihnen zu kurz ist, um als Linie erkannt zu werden. Um diesem Problem entgegenzuwirken, gibt es einen Parameter, der bestimmt, wie groß der Abstand der Pixel sein darf, damit sie noch als Linie erkannt werden.

6 Zusammenfassung

Die Hough-Transformation ist eine gute Möglichkeit, um Linien in Bildern zu erkennen. Falls es ein Problem ist, dass die Linien unendlich lang sind, gibt es die Progressive Probabilistic Hough Transformation, welche das Problem löst. Die Nutzungsbereiche der Linienerkennung mit der Hough-Transformation sind vielfältig. Ein bekanntes Beispiel, wie schon in der Einleitung besprochen, ist das autonome Fahren. Doch auch für Technologien in der Zukunft kann der Algorithmus sehr hilfreich sein.

Literatur

- [1] ADAC. *Autonomes Fahren: Digital entspannt in die Zukunft*. URL: <https://www.adac.de/rund-ums-fahrzeug/ausstattung-technik-zubehoer/autonomes-fahren/technik-vernetzung/aktuelle-technik/>. (accessed: 02.02.2021).
- [2] Richard O. Duda und Peter E. Hart. "Use of the Hough transformation to detect lines and curves in pictures". In: *Communications of the ACM* 15 (1972). DOI: <https://doi.org/10.1145/361237.361242>.
- [3] Paul V C Hough. "Method and means for recognizing complex patterns". US3069654A. März 1960.
- [4] C. Galambosy J. Matasy und J. Kittlery. *Progressive Probabilistic Hough Transform*. Aug. 1988. URL: <https://cmp.felk.cvut.cz/~matas/papers/matas-bmvc98.pdf>.
- [5] Rafael C. Gonzalez Richard E. Woods. *Digital Image Processing*. Rafael C. Gonzalez, 2018. ISBN: 9353062985.
- [6] Azriel Rosenfeld. *Picture Processing by Computer*. Academic Press, 1969. ISBN: 0125973500.