**Seminar in Simulations-Tools**

**Erkennung von Geraden in Bildern**

Einleitung

* ***Thema***
* Geradenerkennung, kurze Einführung
* Bei der Hough-Transformation handelt es sich um ein Verfahren zur Erkennung von Strukturen (Geraden, Kreise,…) in Bildern. In unserem Fall wird die Hough-Transformation zur Erkennung von Geraden verwendet.
* Hough Transformation wurde 1959 erfunden und Ursprünglich zur maschinellen “Untersuchung” von sogenannten “Bubble-Chamber” benutzt ( !!! WIKIPEDIA!!! )

* **Bilder**:
* Im Wesentlichen 2 Arten von Bildern. Es gibt Farbbilder und Schwarz-Weiß-Bilder.
* In beiden Fällen handelt es sich um ein Gitter mit einzelnen Zellen, den so genannten Pixeln.
* Beim Schwarz-Weiß-Bild hat jede Zelle genau einen Wert zwischen 0 und 255. Diese 256 Werte beschreiben die Graustufe des jeweiligen Pixels, wobei “reines Weiß” bei dem Wert 255 erreicht wird und “reines Schwarz” bei 0. (<https://programmerwiki.com/article/64781044523/>)
* Im Gegensatz dazu hat eine Zelle des Farbbildes 3 Farbkanäle
* Rot , Grün, Blau
* Jeder Wert kann wiederum einen Wert zwischen 0 und 255 annehmen wobei der jeweilige Wert der Farbe die Intensität der Farbe widerspiegelt
* “Jedes Pixel kann als drei Kanäle betrachtet werden, und jeder Kanal ist ein unabhängiges Graustufenbild” ( wortwörtlich so aus der Seite <https://programmerwiki.com/article/64781044523/>  übernommen)
* Bei Bildern liegt nicht, wie in der Mathematik, der Ursprung des Koordinatensystems oben links.

* Thema: automatisierte Strukturerkennung
* Computer soll in einigen Anwendungen Objekte in Bildern erkennen
* Identifizierung Personen, Gesichtserkennung bei Geräten
* wir behandeln nur Geradenerkennung

* ***Motivation***
* Idee, fertige Beispiele
* Spurhalteassistent, Pflanzenbeobachtung, Barcodescanning

* Beispiele:
* Pflanzenbeobachtung -> Bastl
* Barcodescanning
* Beispielbilder:
* Spurhalteassistent Auto: mehrere Stufen des Spurverlassenswarnsystem
* Nur zB akustische Warnung
* System ergreift Maßnahme, Fahrzeug in der Spur zu halten
* Übernahme von Fahrzeug, Halten in Spur
* Dafür wird ein (Geraden-) Erkennungssystem benötigt, um Spurlinien in Echtzeit-Bildern zu erkennen, um dann schnell zu reagieren
* Bild?

Hauptteil

* ***Theorie, Verfahren***
* Bild in Graustufen umwandeln, map erstellen
* Allgemein: Theorie Gradient in Skalarfeldern
* Sobel-Filter, Matrizen drüber laufen lassen
* Hough Transformation
* Genau! Welcher punkt wo im H Room
* Geraden einzeichnen

* Bild zuerst in Graustufen umwandeln: Reduzierung auf einen Farbkanal, damit Sobelfilter richtig arbeiten kann
* Erstellung einer Matrix: Weißanteile als Werte, zwischen 0 (schwarz) und 255 (weiß)

Diese Werte werden auf das Intervall [0;1] normiert, sodass 0  o bleibt und 255 jetzt 1 ist, die Verhältnisse zwischen den Werten bleiben aber bestehen.

* Skalarfeld:
* \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
* Gradient: Produkt des Nabla-Operators mit einem Skalarfeld;
* => Richtung des steilsten Anstiegs/der maximalen Änderung des Skalarfeldes
* Resultat Vektorfeld/Gradientenfeld
* -----------------------------------------------------------------------------

Nachdem ein Graustufenbild vorliegt werden die Kanten des Bildes identifiziert.

Dafür wird der Sobel-Filter verwendet.

Hierbei werden 2 3x3-Matrizen erstellt (???eine für den horizontale und eine für vertikale Kanten???)

Außerdem wird zusätzlich für jede Matrix ein leeres Wertefeld der Größe des Bildes erstellt.

Danach wird über jedes Pixel des Graustufenbildes iteriert.

Ausgenommen hierbei ist der Rand, das ist dem Algorithmus an sich geschuldet, da das zu bearbeitende Pixel immer den Mittelpunkt einer 3x3 Matrix darstellt, welcher aus dem Pixel an sich und den 8 umliegenden Pixeln (Pixelwerte) besteht.

Die 3x3 Matrix des Arbeitspixel und des Sobelfilters werden übereinandergelegt und die übereinanderliegenden Werte miteinander multipliziert.

Anschließend werden alle 9 Produkte aufaddiert. Und die Summe in das entsprechende Wertefeld an der gleichen Stelle wie das Arbeitspixel eingetragen ????

Dieser Vorgang wird zuerst für die horizontale und danach für die vertikale Matrix durchgeführt.

Da der Sobelfilter eine Kante

* Diese beiden Gradienten werden
* Auch hier: Sobel Filter = image gradient; je stärker Farbänderung (heller/dunkler), desto größer Wert des Gradienten an dieser Stelle
* Sobel Filter: Gradient, vertikale und horizontale Matrix ermitteln stärkste Farbänderungen (Schwellwert) und ermitteln somit Kanten im Bild

* // Beispielbild, Matrizen dazu, erklären
* Hough-Transformation: Verfahren zur Erkennung von Geraden oder anderen Objekten, wie Kreisen, in Bildern
* Grober Überblick, bevor genaue Erläuterung:
* Nach Anwendung des Gradienten: Erschaffen eines Dualraums (Hough-Raum)
* Für aller Punkte, die auf einer erkannten Kante liegen, werden alle möglichen Parameter in den Hough-Raum eingetragen, die für eine Gerade durch diese Punkte möglich wäre
* Parametergleichung: Polarform x\*cos(phi) + y\*sin(phi) = d
* Neue Parameterwerte: Winkel phi durch Lot ablesen, dann mit Gleichung d bestimmen
* Raum mit d-phi-Achsen beschriften
* Für jeden einzelnen Punkt wird die dazugehörige Kurve im H-R eingetragen.
* // Beispiel!! Welche Gerade entsteht für Bsp-Punkt
* Nun Analyse nach Häufungspunkten im Raum: Aufteilen des Raumes in Matrix: wie oft gehen die Geraden jeweils durch Kästchen der Matrix? Zählen -> Maxima (Schnittpunkte) deuten auf Gerade im ursprünglichen Bild hin

* ***Programm***
* Grober Überblick
* Anwendung der Theorie?
* Segmentierung

* ***Bedienungsanleitung:***
* Starten des Programms “Geradenerkennung.py” nach starten des Programmes ist muss das zu bearbeitende Bild eingelesen werden. Dazu wählen sie den Bildpfad aus und ersetzen Sie den den Text <path>
* Als nächstes muss die Anzahl der Inkremente für sowohl Winkel Phimit dem abgetastet wird und der Unterteilung (für die Feinheit) des Ergebniswertes d.
* Die Inkremente beschreiben die Anzahl der Wertepunkte/Wertemöglichkeiten auf jeder Achse.
* Bei Phi von –90 bis + 90
* Da d sowohl positive als auch negative Werte annehmen kann, wird hier die Inkrementzahl effektiv halbiert, da die Hälfte der Unterteilung für alle positiven Werte und die andere Hälfte für alle negativen Werte zur “Verfügung” stehen. Damit das Programm ohne Probleme ausgeführt wird, wird deshalb Programmintern der eingestellt Wert der Inkremente für d verdoppelt.
* Um somit einen quadratischen Hough-Raum zu erhalten sollte die Formel
* Inkremente\_d \* 2 = Inkremente\_phi    oder   eben Inkremente\_Phi \* ½ = Inkremente\_d
* Die eingestellten Inkremente bestimmen somit die Dimension des Hough-Raumes.
* Im nächsten Schritt muss der Schwellwert eingestellt werden, den ein Punkt im Hough-Raum überschreiten muss, damit der Punkt im Hough-Raum als Gerade gewertet wird.
* Optional: Letztlich kann bei der Maximasuche nur noch der Suchradius eingestellt werden. Dieser Radius beschreibt einen Kreis in dem von einem Startpunkt ausgehend nach Punkten mit größeren Werten gesucht wird.

* ***Beispiele***
* Einfach: Quadrat/Dreieck
* Ohm?
* Evtl. komplexeres Bild (Rechenzeit)

Schluss

* ***Diskussion -> Vor-/Nachteile***
* „Brute-Force“
* Rechenzeit, bei welcher Pixelzahl wielang, wo Grenze
* Flexibilität Anwendung, gui, Eingabe?