

# **Geschwindigkeitserkennung von Autos auf der Autobahn**

**EDBV WS 2018/2019: AG\_B5**

Dario Sommer (11777720)  
Dominik Gangl (11776203)  
Florian Holzmann (11776181)  
Joshua Oblong (01631789)  
Martin Steinmetz (11777767)

6. Januar 2019

## **1 Problemstellung**

### **1.1 Ziel**

Ziel des Projekts ist die Erkennung und das berechnen der Geschwindigkeit von auf der Autobahn fahrenden Autos anhand eines 10 sekundigen Videos, gefilmt von einer Autobahnbrücke.

### **1.2 Eingabe**

Als Input wird ein Video im MP4 Format erwartet, welches Autos von einer Brücke gefilmt zeigt. Genauere Eigenschaften, wie das Video aussehen sollte, werden in Punkt *1.4 Vorraussetzungen und Bedingungen* festgelegt. Als zusätzlicher Parameter wird die Geschwindigkeitsbegrenzung auf dem gefilmten Autobahnabschnitt erwartet.

### **1.3 Ausgabe**

Ausgegeben wird ein Video, in welchem die Fahrzeuge markiert sind und dessen Durchschnittsgeschwindigkeit angezeigt wird. Außerdem werden jene Fahrzeuge über dem Geschwindigkeitslimit rot markiert.

## 1.4 Voraussetzungen und Bedingungen

Der Input ist ein Video, welches folgende Eigenschaften erfüllt:

- Das Video zeigt fahrende Autos auf einer mindestens 2-spurigen Autobahn, welche sich von der Kamera wegbewegen.
- Das Video wurde von einem Stativ aus gefilmt und die Kamera bewegt sich während des gesamten Videos nicht.
- Das Video ist im Querformat, MP4 und hat folgende Qualität: 626x352 Pixel mit 30fps
- Die Länge darf 10 Sekunden nicht überschreiten.
- Das Video wurde bei Tag und guten Lichtverhältnissen aufgenommen.
- Die Kamera muss so platziert werden, dass sich die zwei zu erkennenden Fahrspuren genau links bzw. rechts zur Mitte und die strichlierte Fahrbahnmarkierung dazwischen genau in der Mitte befindet.
- Die strichlierte Fahrbahnmarkierung muss genau senkrecht ausgerichtet sein.
- Die gefilmten Autos müssen den verpflichtenden Mindestabstand auf der Autobahn einhalten.

Als zusätzlicher Parameter wird die Geschwindigkeitsbeschränkung des gefilmten Abschnitts erwartet, um evaluieren zu können, welches Fahrzeug sich nicht an die vorgegebene Geschwindigkeit gehalten hat.

## 1.5 Methodik

Methodik- Pipeline

1. Threshold (Erkennung der Fahrbahnmarkierungen)
2. Thinning (Erkennung der Fahrbahnmarkierungen)
3. Opening (mit Linien-Strukturelement zur Erkennung der Fahrbahnmarkierungen/Entfernen von Bildrauschen)
4. Opening (mit einem Quadratischen-Strukturelement zur Erkennung der Fahrzeuge/Entfernen von Bildrauschen)
5. Connected Component Labelling (einzelne Fahrzeuge segmentieren)

## 1.6 Evaluierungsfragen

- Sind die Abstände der Trennlinien korrekt erkannt worden?
- Konnten die Fahrzeuge erkannt werden?
- Wurden jene bewegende Objekte, die nicht Autos sind, entfernt?
- Konnte die Geschwindigkeit der Autos erkannt werden?
- Ist die erkannte Geschwindigkeit legitim?

## 1.7 Zeitplan

Meilenstein	abgeschlossen am		Arbeitsaufwand in h	
	geplant	tatsächlich	geplant	tatsächlich
Erstellung eines Prototyps	7.11	30.11	25h	35h
Erstellung des Datensatzes	7.11	28.11	25h	60h
Bestimmen der selbst zu implementierenden Methoden	13.11	15.12	1h	1h
Implementierung der Methoden	5.12	03.01	175h	150
Testen und Fehlerbehebung	6.1	05.01	74h	70
Abgabe des Projekts	6.1	6.1	1h	1h

## 2 Arbeitsteilung

Name	Tätigkeiten
Martin Steinmetz	- Filmen und Bearbeitung der Videos - Erstellung der Datenbank - Gestalten der Endpräsentation
Dario Sommer	- Implementierung Hintergrunderkennung, Linienerkennung, Autoerkennung - Implementierung von Matlab-Methoden Opening, Erosion, Dilation - Zusammenfassen aller Matlab-Teile
Dominik Gangl	- Bericht - Implementierung-GUI - Gestalten der Zwischenpräsentation
Florian Holzmann	- Implementierung Geschwindigkeitsmessung - Implementierung Video-Output (Umrandung der Autos, Anzeigen der Geschwindigkeit, Färben der Umrandungen)
Joshua Oblong	- Implementierung Thinning - Versuch der Implementierung Connected Component Labeling

## 3 Methodik

### 3.1 Thinning

Beim Thinning wird der Vordergrund eines Binärbildes bis auf ein Pixel breite verdünnt. [1] Nach Anwendung des Thinning-Algorithmus erhält man eine bis auf das Minimum reduzierte Form des Eingabebildes. Die grundlegende Struktur des Eingabebildes wird dabei nicht verändert.

Thinning wurde bei unserem Projekt für die pixelgenaue Bestimmung der Fahrbahnmarkierungen benutzt. Hierbei wurden die mithilfe der Thresholdoperationen erkannten Linien auf die Stärke von einem Pixel reduziert um die Geschwindigkeit genau bemessen zu können. Unsere Berechnung basiert auf den genormten Abständen der Fahrbahnmarkierungen und daher ist es besonders wichtig diese Linien sehr genau zu bestimmen. In weiterer Folge legen wir horizontale Linien an den verdünnten Fahrbahnmarkierungen an um so errechnen zu können, welche Zeit die Autos brauchen um den Abstand zwischen den Linien zurückzulegen.

### 3.2 Connected Component Labeling

Connected Component Labeling ist eine Methodik, welche für die Erkennung der Zusammenhangskomponenten eines Bildes verwendet wird. [2] Hierbei wird jedem Bildpunkt ein Zahlenwert zugeordnet, welcher dem Segment in dem er sich befindet entspricht. Dieser Zahlenwert wird als Label bezeichnet. Als Eingabebild wird meist ein Binärbild erwartet, bei welchem ausschließlich der Vordergrund gelabelt werden soll. Mithilfe von Connected Component Labeling wird eine Einteilung in mehrere Objekte möglich.

Da wir bereits Vordergrund von Hintergrund unterschieden haben, müssen wir den relevanten Vordergrund nun segmentieren. Dazu haben wir Connected Component Labeling verwendet. Die einzelnen Fahrzeuge sind momentan als weiße Flächen auf dem Binärbild erkennbar. Mithilfe von CCL können wir die Flächen nun als verschiedene Objekte erkennen und somit die einzelnen Fahrzeuge mit Bounding Boxes versehen. Das ermöglicht in weiterer Folge eine Berechnung der Autogeschwindigkeiten.

### 3.3 Threshold

Bei einer Thresholdoperation wird ein Graustufenbild in ein Binärbild umgewandelt. [3] Alle Pixel, welche einen gewissen Helligkeitsschwellwert überschreiten, werden weiß eingefärbt und der Rest wird schwarz eingefärbt. Dadurch kann der relevante Vordergrund vom nicht relevanten Hintergrund eines Bildes unterschieden werden. Die Methodik wird oft zur Vorverarbeitung am Anfang der Methodiken-Pipeline verwendet.

In unserem Projekt wurden Thresholdoperationen zu Erkennung der Fahrbahnmarkierungen, welche für die Distanz- bzw. Geschwindigkeitsmessung wichtig sind, verwendet.

Hierbei wurde ein hoher Wert im Weißbereich angesetzt, um nur die Fahrbahnmarkierungen zu erkennen. Dadurch erhalten wir ein Binärbild, welches wir für die weiteren Methodiken benötigen.

### 3.4 Opening

Die morphologische Operation Opening beschreibt eine Erosion gefolgt von einer Dilation. [4] Opening haben wir auf die Binärbilder zur Verminderung von Bildstörungen durch helle Bildpunkte verwendet, da wir nur bestimmte Elemente mit unserem Programm erkennen wollen. Um anschließend die Geschwindigkeit messen zu können, haben wir die Autos durch minimale Rechtecke eingegrenzt, welche alle Pixel, die zum jeweiligen Fahrzeug gehören, berücksichtigen. Diese Rechtecke nennt man Bounding Boxes. Durch das Opening konnten genauere Bounding Boxes um die Fahrzeuge gezeichnet werden, die Fahrbahnmarkierungen erkannt und Bildrauschen vermindert werden.

Bei morphologischen Operationen wird oft ein Strukturelement zur genaueren Anpassung der Ergebnisse verwendet. Es funktioniert wie eine Maske, nach deren Anwendung die gewünschten Strukturen erhalten bleiben und unerwünschte nicht.

Zuerst haben wir die Operation mit einem Linien-Strukturelement auf das Linien Bild nach dem Thining angewendet um die Fahrbahnmarkierungen zu erkennen. Für die Autoerkennung wurde ein quadratisches Strukturelement verwendet um anschließend die Bounding Boxes möglichst genau um die Fahrzeuge zeichnen zu können.

## 4 Implementierung

Bei dem Projekt wurden alle Funktionen in die Main-Methode geschrieben und können mithilfe eines GUI ausgeführt werden. Die folgende Aufzählung behandelt alle Funktionen und Teilbereiche des Projekts:

- **Voreinstellungen:** Zu Beginn des Projekts werden allgemeine Einstellungen festgelegt. Das Video wird eingelesen, die Framesize und die Gesamtanzahl der Frames erkannt.
- **Background:** Um den Hintergrund zu bestimmen werden 50 Frames betrachtet und der Modus von jedem Pixel bestimmt. Es werden nicht alle Bilder betrachtet um die Laufzeit geringer zu halten. Mit dem errechneten Bild kann man im nächsten Schritt bestimmte Teile eines Frames segmentieren.
- **Fahrbahnmarkierungen und Methode Thinning:** Da die Fahrbahnmarkierungen weiß sind, können wir mithilfe eines Threshold alle Linien relativ einfach aus dem Hintergrund segmentieren. Um die Fahrbahnmarkierungen bis auf wenige Pixel zu verdünnen, haben wir die Methodik *Thinning* angewendet. Dazu haben wir eine Funktion mit dem Zhang-Suen Thinning-Algorithmus [5] erstellt.

Als Eingabeparameter werden einzelne Frames benötigt und der Algorithmus entscheidet anhand verschiedener Kriterien in jedem Schleifendurchlauf, ob das Pixel weiß gefärbt wird. Wenn kein Pixel mehr weiß gefärbt wird, hat man eine minimale Darstellung des Eingabebildes erhalten.

- **Opening:** Die morphologische Operation Opening beschreibt eine Erosion gefolgt von einer Dilatation. Daher wurden die zwei Funktionen *Erosion* und *Dilatation* implementiert. Bei beiden Funktionen werden als Eingabeparameter die Frames und ein strukturgebendes Element benötigt. Bei der *Erosion* wird das Strukturelement nun Pixelweise über das gesamte Bild bewegt und es wird überprüft ob das Strukturelement vollständig in die Menge der aktuellen Pixel passt. Falls ja, so gehört das Pixel, an der sich der Bezugspunkt momentan befindet, zur erodierten Menge. Bei der *Dilatation* wird das Strukturelement ebenfalls über das gesamte Bild bewegt und an den Grenzen der Menge um den Teil des Strukturelements erweitert, welcher die Grenzen überschreitet. Man dehnt also jeden Bildpunkt auf die Größe des strukturgebenden Elements aus.
- **Detektionslinien:** Nachdem die Fahrbahnmarkierungen erkannt und verdünnt wurden, werden horizontale Linien am Ende der ersten Markierung und am Anfang der zweiten Markierung angebracht, um den genormten Abstand zu erhalten. Das geschieht im Abschnitt *Get Detection Lines*. Es wird ein kleiner Betrag addiert um Fehler von Thinning und die Position der Linien auszugleichen. Mithilfe dieser Linien kann errechnet werden wann die Autos den Messbereich betreten und wann sie ihn verlassen. Dadurch kann man sich die Geschwindigkeit der Autos im besagten Bereich errechnen.
- **Vordergrund- und Autoerkennung:** Anhand der Frames, welche in Binärbilder umgewandelt werden, können die Autos erkannt werden. Dies erfolgt indem der zuvor berechnete Hintergrund vom aktuellen Frame abgezogen wird. Das resultierende Bild enthält dann nur den Unterschied, die Autos. Dann wird die morphologische Operation Opening mit einem quadratischen Strukturelement auf die Bilder angewendet, um aus den nur schwach sichtbaren Autos ein klar erkennbares Element zu machen.
- **CCL, Bounding Boxes und Autofärbung:** Connected Component Labeling wird benutzt um den Vordergrund in verschiedene Segmente zu unterteilen um die einzelnen Autos zu erkennen. Jedoch wurde mit dieser Methode bei einem Video mit vielen Frames die Performance extrem geschwächt, da für jede Bounding Box und jeden Frame eine Schleife durchlaufen werden muss, die jede Bounding Box auf die Größe überprüft. Deshalb haben wir uns letztendlich für das Tool *vision.BlobAnalysis* entschieden, wodurch wir eine viel bessere Performance erhielten. (Leider haben wir es aufwands- und zeittechnisch nicht mehr geschafft, diese Methode selbst zu implementieren, weshalb keine Implementierungsdetails behandelt werden können.) Dann werden die Autos mit Bounding Boxes versehen um die Berechnung zu ermöglichen. Es kann auch der Fall eintreten, dass zwei Bounding

Boxes verbunden werden müssen. Falls sie einen gewissen Abstand voneinander haben, wird die geringste Distanz der Kanten im Abschnitt *"returns minimal distance between two rectangles"* berechnet. Mit dieser Information können die Rechtecke optimal verbunden werden. Abhängig von der aktuellen Position und vom Ergebnis der Geschwindigkeitsberechnung, werden die Autos entsprechend eingefärbt.

- **Geschwindigkeitsberechnung:** Für die Geschwindigkeitsberechnung werden am hinteren Ende der Bounding Boxes der Autos horizontale Linien angebracht. Diese bewegen sich mit dem Auto mit und ermöglichen eine Geschwindigkeitsberechnung.
- **Graphical User Interface:** Für die einfachere Bedienung wurde ein GUI in der Appdesigner Umgebung erstellt. Um die Appdesigner Einstellungen und den Code einzusehen, muss man in der Matlab-Konsole Appdesigner('GUI') aufrufen. Im GUI wird als Eingabeparameter ein Video erwartet, welches sich im selben Ordner wie das Projekt befinden muss. Nachdem man den Titel eines Videos eingegeben hat, wird eine Vorschau mit dem ersten Frame des Videos angezeigt. Dann kann man mithilfe der Buttons entweder das unbearbeitete Video oder die Geschwindigkeitsberechnung starten.
- **Start des Programms:** Neben dem GUI gibt es auch die Möglichkeit das Programm normal zu starten. Hierbei ist nur zu beachten, dass sich das am Anfang des Codes festgelegte Video im selben Ordner wie das Projekt befindet. Man kann im Abschnitt mit dem Kommentar *"car is in speed projection area"* die gewünschte Höchstgeschwindigkeit verändern. Der gewählte Standardwert ist 130 km/h. Für die Ausführung der Anwendung benötigt man die **Computer Vision Toolbox**.

## 5 Evaluierung

Der Datensatz unseres Projekts besteht aus einigen 10 sekündigen Clips. Für Testzwecke wurden meist längere Videos eingelesen, um die Ergebnisse genauer analysieren zu können. Die kurzen Clips wurden aus den längeren Videos ausgeschnitten um die Laufzeit zu verringern.

Die Videos wurden von einer Autobahnbrücke gefilmt und zeigen fahrende Autos. Genauere Details zur Aufnahme befinden sich in Punkt 1.4 *Voraussetzungen und Bedingungen*. Es wurde mit einer Handkamera in 720p gefilmt. Dabei wurde ein Stativ verwendet und auf möglichst gute Umweltverhältnisse geachtet. Schon geringes Ruckeln bei der Aufnahme kann Fehler produzieren, daher ist unbedingt auf eine ruhige Aufnahme zu achten. Weiters wurden die Videos mit einer externen Software stabilisiert um die Kamerabewegung zu minimieren.

Die genaue Größe der Frames unserer Videos beträgt  $626 * 352$  Pixel, kann aber leicht abweichen und wurde auch mit verschiedenen Größen getestet. Die Videos sind 10 Sekunden lang, also hat jeder Clip etwa 300 Frames (30 Frames pro Sekunde vorausgesetzt).

Es folgt die Diskussion der Evaluierungsfragen:

- **Sind die Abstände der Trennlinien korrekt erkannt worden?**

Die Abstände der Trennlinien werden sehr genau erkannt. Mithilfe der Methodik Thinning konnte der Abstand pixelgenau berechnet und die horizontalen Bemessungslinien angebracht werden.

- **Konnten die Fahrzeuge erkannt werden?**

Die Fahrzeuge werden gut erkannt. Es gibt keine Fahrzeuge, die sich unerkannt durch das Bild bewegen. Diese Tatsache führt leider zu dem Problem, dass Autos, welche sich auf der entgegengesetzten Fahrbahn befinden, genauso erkannt und mit einer Bounding Box versehen werden, wie die Autos auf der richtigen Seite der Autobahn. Die Geschwindigkeit der Autos auf der entgegengesetzten Fahrbahn wird zwar nicht berechnet, aber die unnötig erkannten Fahrzeuge können die Geschwindigkeitsberechnung der richtigen Autos erschweren bzw. ungenauer machen.

- **Wurden jene bewegende Objekte, die nicht Autos sind, entfernt?**

Die Objekte, welche sich nur flüchtig durch das Bild bewegen, werden durch das Opening gut herausgefiltert. LKWs bereiten unserem Programm leider immer noch Probleme, da sie die Lichtverhältnisse des ganzen Frames verändern. Dadurch werden die Helligkeitswerte des ganzen Bildes geändert und da unser Programm mit fixen Schwellwerten für die Linien- und Autoerkennung arbeitet, werden manchmal zu viele bzw. zu wenige Autos erkannt.

- **Konnte die Geschwindigkeit der Autos erkannt werden?**

Die Geschwindigkeit der Autos kann in den meisten Fällen erkannt werden. Problematisch ist es nur, wenn LKWs oder Autos von der entgegengesetzten Fahrbahn die Bounding Boxes der korrekt erkannten Autos verändern. Die Geschwindigkeit wird nämlich nicht erkannt, wenn sich die Bounding Box nicht zu Beginn und am Ende innerhalb der horizontalen Messlinien befindet. Wenn mehrere Bounding Boxes für ein Auto erkannt werden, ist die Geschwindigkeitsberechnung meist ungenau.

- **Ist die erkannte Geschwindigkeit legitim?**

Die erkannte Geschwindigkeit ist in den meisten Fällen legitim und weicht nur bei offensichtlichen Fehlern stark von der Realität ab. Wenn das Programm von LKWs oder erkennbarem Ruckeln irritiert wird, wird oft keine oder eine zu hohe Geschwindigkeit erkannt. Die falsch erkannte Geschwindigkeit ist dann aber meist so hoch, dass man sofort erkennt, dass etwas falsch gemessen wurde.

Es wurden etwa 60 Videos evaluiert. Die Laufzeit beträgt pro 10 sekündigem Video etwa 5 Minuten und betrug beim Prototyp nur etwa 20 Sekunden. Die Laufzeit erhöht sich besonders durch die wiederholte Anwendung der selbst implementierten Funktionen *Erode* und *Thinning*.



Die Implementierung funktioniert besonders gut bei Videos ohne LKWs und Gegenverkehr. Wenn sich die Lichtverhältnisse oder die Kameraposition während des Videos verändern, kommt es zu Ungenauigkeiten und Fehlern. Der häufigste Fehler ist, dass zwei horizontale Messlinien an den Fahrzeugen angebracht werden und eine Messung somit unmöglich wird. Besonders wenn LKWs im Video vorkommen, werden zu viele Bounding Boxes erkannt.

## 6 Schlusswort

Zusammenfassend kann man sagen, dass wir die umfangreiche Problemstellung, bis auf Ausnahmen, gut lösen konnten. Man muss aber auch beachten, dass es noch einige Problemstellen gibt. Die Laufzeit könnte zum Beispiel durch effizientere Algorithmen verbessert werden oder die Behandlung von Ausnahmefällen wie LKWs genauer betrachtet werden. In den Monaten des Projekts ist es uns leider nicht gelungen diese Probleme zu beheben, teils ist es vielleicht auch nicht möglich. Offene Probleme bleiben die Effizienz, LKWs, die Reaktion der Software auf geringes Ruckeln der Kamera, zu viele Messlinien oder Umwelteinflüsse wie Schnee oder starke Sonne.

Es war interessant zu sehen, dass es in einer Gruppe ohne Vorerfahrung möglich war, eine so umfangreiche Problemstellung zu lösen. Wir als Gruppe sind zufrieden mit den Ergebnissen, aber auch motiviert, die verbleibenden Ungenauigkeiten zu beseitigen.

## Literatur

- [1] Richard E. Woods Rafael C. Gonzalez. *Digital Image Processing*, pages 541–543. 1992.
- [2] Kurt Schwenk. *Methoden zur Bildsegmentation und zum Connected Component Labeling auf einem FPGA für eine zukünftige On-Board Bilddatenverarbeitung*. PhD thesis, 2015.
- [3] Linda Shapiro. *Computer Vision*, pages 97–98. 2000.
- [4] Linda Shapiro. *Computer Vision*, pages 75–81. 2000.
- [5] C. Y. Suen T. Y. Zhang. A fast parallel algorithm for thinning digital patterns. *Communications of the ACM*, 27(3):1–4, 1984.