

Nutzerhandbuch: Höchstgeschwindigkeitsschilder

Projekt 8 - Digitale Bildverarbeitung

Andrej Lisnitzki,, Michael Horn

16. Juni 2016

1 Allgemeines

In diesem Projekt war es das Ziel mit Hilfe von ImageJ eine Erkennung von Höchstgeschwindigkeitsschildern zu implementieren. Die Implementierung erfolgte als ausführbares ImageJ-Plugin.

Zur Erkennung von Schildern wird zunächst ein Farberkennung realisiert und im Anschluss mit einer Hough-Transformation nach Kreisen gesucht. Gefundene Kreise werden im nächsten Schritt durch eine OCR-Anwendung analysiert. Werden dabei gültige Geschwindigkeitsinformationen geliefert, so wird das Schild als Höchstgeschwindigkeitsschild erkannt und entsprechend markiert.

Damit die Erkennung der Geschwindigkeitsschilder funktioniert, wird GOCR¹ als ausführbare OCR-Anwendung benötigt. Diese Anwendung muss in dem entsprechenden Ordner des Plugins hinterlegt sein. Weiterhin muss diese Binärdatei ausführbar sein (Executable Flag gesetzt).

Ergebnisse werden im Ordner der Eingangsdatei gespeichert.

¹<http://jocr.sourceforge.net/> Stand: 16. Juni 2016

2 Voraussetzungen

ImageJ muss mindestens mit der Java Version 7 arbeiten und der Compiler von ImageJ muss auf 1.7 gestellt werden! Ältere Versionen werden nicht unterstützt!

GOOCR muss im entsprechenden Unterordner des Plugins ausführbar sein.

3 Ablauf

Das ImageJ-Plugin arbeitet wie folgt:

Als erstes erfolgt eine Farbdetektion des Schildes. Jedes Höchstgeschwindigkeitsschild besitzt einen roten Kreis. Im ersten Schritt werden rote Punkte durch die Farbe Weiß ersetzt und alle andere Farben auf Schwarz. Dadurch wird der Hintergrund Schwarz und relevante Punkte Weiß.

Im nächsten Schritt wird eine Hough-Transformation für Kreise durchgeführt. Der verwendete Quellcode für die Hough-Transformation stammt von <https://imagej.nih.gov/ij/plugins/hough-circles.html> (Stand: 16. Juni 2016). Diese Transformation sucht nach Mittelpunkten von potenziellen Kreise. Damit die Suche erfolgreich ist müssen für die Transformation verschiedene Parameter angegeben werden. Zum einen Parameter über den minimalen und maximalen Radius des Kreises der erkannt werden soll sowie die Größe der Iterationsschritte vom kleinsten zum größten Radius. Ebenfalls angegeben werden muss die maximale Anzahl an Kreisen die gefunden werden soll.

Nachdem Kreismittelpunkte durch die Hough-Transformation gefunden wurde wird überprüft ob die gefunden Kreismittelpunkte wirklich einen Kreis beschreiben. Dazu werden die Abstände vom Kreismittelpunkt zum Rand kontrolliert. Die Kontrolle erfolgt vertikal, horizontal und diagonal, sodass maximal 8 mögliche Übereinstimmungen für einen Kreis gefunden werden können. Über den Parameter

Hit wird der Grenzwert an Position festgelegt, damit der Kreis als Kreis erkannt wird.

Nachdem nun potenzielle Kandidaten für Höchstgeschwindigkeitsschilder gefunden wurden, werden bestimmte Bereiche aus dem Originalbild kopiert. Aus den Bereichen wird die Farbe rot entfernt und anschließend eine Binarisierung durchgeführt. Im Anschluss wird das Bild gespeichert und an die externe GOCR-Anwendung übergeben. Diese überprüft das Bild auf Text und gibt einen String von Text zurück. Über Patternmatching wird anschließend geprüft ob eine Geschwindigkeit im Schild steht oder nicht.

Falls ein Geschwindigkeitsschild gefunden wurde, so wird eine rote Bounding-Box um das Schild gezeichnet und die erkannte Geschwindigkeit unter das Schild geschrieben. Sollte keine Geschwindigkeit erkannt werden, so erfolgt keine Hervorhebung des Schildes! Das Ergebnisbild wird als „_RSD.png“ im Quellordner gespeichert. Zusätzlich wird in diesen Ordner die Ergebnis-XML gespeichert. Die Speicherung von Ergebnissen erfolgt nur wenn Geschwindigkeitsschilder vollständig erkannt wurden.

4 Aufruf

Aufruf per Kommandozeile:

```
./ImageJ -eval "run('Street Speed Sign', 'i=plugins/Project/result/test.png  
rmin=10 rmax=50 rinc=2 cnum=6 hits=6');"
```

Folgende Parameter werden benötigt, damit die Erkennung funktioniert:²

- Pfad zur Bilddatei
 - i=plugins/Project/result/vlcsnap-2016-05-04-14h27m18s219.png

HoughCircle-Parameter:

- Kleinster Radius des zu erkennenden Kreises
 - rmin=10
- Größter Radius des zu erkennenden Kreises
 - rmax=50
- Schrittweite der Radiuserhöhung bei der Erkennung
 - rinc=2
- Anzahl der zu erkennenden Kreise
 - cnum=6
- Anzahl der Treffer, damit der Kreis erkannt wird ($0 \leq 8$), es werden Senkrechte, Vertikale und Diagonale Punkte verwendet um einen Kreis zu definieren
 - hits=6

²**Achtung:** Derzeit erfolgt keine Überprüfung der Eingangswerte auf Korrektheit!