Projektdokumentation im Modul Digitale Bildverabeitung

Detektion und Erkennung von QR-Verkehrszeichen

Tina Neumann und Minh Pham

June 20, 2017

Mit dem vorliegenden ImageJ-PlugIn ist es möglich, QR-Codes in einem RGB Bild zu erkennen.

1 Ansatz

Das Erkennen von QR-Codes wird mit einer horizontalen Linescan untersucht. Dabei wird das Bild zunächst binarisiert und im darauffolgenden wird dies mit horizontalen Linescan untersucht.

2 Verwendung von Parametern

binMode: Unterscheidung zwischen 0 und 1. Bei der Auswahl von 0, wählt man die Otsu Methode. Bei der Auswahl von 1 wählt man die HSB Methode

Boxgröße: Hierbei werden die Boxgrößen mit minimalen und maximalen Boxbreite und -höhe in Pixeln bestimmt.

scanColDist: Mit scanDolDist werden die vertikalen ScanLine pixelweise festgesetzt.

Bildvariablen: Dabei werden das Originalbild und das Binärbild gespeichert als ImagePlus. In dem Originalbild werden die gefundenen Bounding-Boxen gezeichnet und die Bearbeitung erfolgt im Binärbild.

3 run-Methode

3.1 Öffnen & Binarisieren des Bildes

Im ersten Schritt wird das PlugIn im Makro aufgerufen, danach wird das Bild des übergebenen Pfad geöffnet und hinzu binarisiert. Desweiteren gibt es die Möglichkeit das Bild im ImageJ zu öffnen, wenn das Bild im Makro nicht gefunden wird. Es existiert zwei Möglichkeiten für das Binarisieren des Bildes. Mit binmode kann festgelegt werden, ob die Ostu-Threshhold-Methode (Standrd ImageJ) oder die selbstimplementierte colorThresholdBinary verwendet werden.

3.2 Auffinden von weißen Liniensegmenten

Hierbei werden die Segmente im Linescan untersucht. Ziel ist es hierbei weißen Segmente zu finden. Im ersten Schritt erfolgt das Auffinden von Segmenten. Die Spaltenprofile werden untersucht, ob diese mehr weiße Pixelanteile enthalten sind. Diese werden in der Hash-Map hinzugefügt.

3.3 Ausschluss von Segmenten

Der folgenden Hash-Map wird iteriert und die Segemente untersucht. Dabei erfolgt das Ausschließen von Segementen, die schwarze Pixelanteile enthalten. Desweiteren werden Linien, die keine Kanten sind und die über dem Standardwert von 350 sind, entfernt.

3.4 Vergleich von Segmenten

In zwei verschachteltet For schleifen werden die Linien der segmentMap miteinander Verglichen. Dabfür wird zunächt geprüft ob die Segment Nummern und damit die Linien sich unterscheiden und ob die Linien in unterschiedliche Spalten (x Position) aufweisen. Als nächstes wir überprüft ob der horizontale(x) Abstand beider Linien den begrenzungen von minBoxWidth und maxBoxWidth entspricht. Trifft all dies zu werden zwei horizontale Linien erzeugt die oberes(rot) und Unteres(blau) Ende der beiden vertikalen Linien(magenta) verbindet. Dannach wird untersucht ob das Linienprofil der oberen und unteren Linie ausschließlich Weiße Pixel enthält. Sind Die vertikalen Linien Verbunden wird noch der Wikel von Oberer und Unterer überprüft. Ist er innerhalb der von maxAngle Vorggegebenen Grenze wird Ein Rectangle erzeugt und zur ArrayList boundingBoxes hinzufügt Anschließend wird die innere Schwarze box mit der Methode innerBlackBox() gesucht und zu einer weiteren ArrayListe NAME hinzugefügt. Dannach werden die Boxen beider Ĺisten in gelb und cyan ins originalbild gezeichnet.

4 colorThresholdBinary

Das Original RGB Bild wird zunächst kopiert und der Kontrast erhöht. Die Arrays min, max und filter enthalten Einstellungen für die akzeptierten Grenzwerte. Das Bild wird in drei Grauwertbilder zerlegt, jeweils für Farbton, Sättigung und Helligkeit. Diese werden umbenannt, um sie nacheinander mit For-Schleife zu bearbeiten. Innerhalb der For-Schleife werden die Grauwertbilder mit minimalem und maximalem Grenzwert binarisiert. Nach dem die Schleife durchlaufen wurde, werden die drei binären Ergebnisbilder mit der ADD Funktion des ImageCalculators zussamengeführt und geschlossen. Rückgabewert ist das zusammengesetzte Ergebnisbild(Binär)

5 innerBlackbox

Eingabe: awt.Rectangle das die äußeren Grenzen des weißen Rahmens festlegt. Die Methode sucht innerhalb des äußeren Weißen Rahmens das schwarze Quadrate (das die Kodierten informationen enthält). In einer geschachtelten For Schleife,wird mit einem gewissen Abstand (int padding) zum äußeren Rand nach schwarzen Pixeln gesucht. Dabei wird das padding verwendet um die Erkennung der inneren Box auch bei perspektivisch verzerrten Bildern zu verbessern.Innerhalb der Schleifen werden die äußeren Grenzen der schwarzen Box ermittelt. Am Ende der Methode wird ein neues Rectangle mit den gefundenen grenzen Konstruiert und zurückgegeben.