

# Optische Verbrauchszählerauslesung

**Blendl, Herbert; Janker, Robert; Klein, Andreas; Wutzer, Felix; Rausch, Mathias**

Hochschule Landshut

## Kurzfassung

Energie- und Stoffströme wie Gas, Wasser, Heizung und Elektroenergie werden beim Endverbraucher heute mit unterschiedlichen Systemen meist manuell erfasst. Um alle Verbräuche mit einem System einheitlich zu erfassen, bietet sich die optische Erfassung der Verbrauchsstände an, da diese Schnittstelle vorhanden ist und keinen Eingriff in den Verbrauchszähler erfordert. Im Rahmen einer Projektarbeit wird eine Hard- und Softwarelösung erarbeitet, die mittels einer Kamera ein Foto eines beliebigen Zählers aufnimmt, an einen Auswerterechner überträgt und auf diesem mittels Bildverarbeitung und Texterkennung den Zählerstand erfasst.

## 1. Motivation

Gas-, Wasser-, Strom- und Heizungszähler werden eingesetzt, um den Verbrauch der jeweiligen Ressource zu erfassen und mit dem Versorgungsunternehmen abzurechnen. Diese Zähler, die in verschiedensten Varianten bei Privat- und Geschäftskunden zum Einsatz kommen, müssen in festgelegten Intervallen abgelesen werden. Das Ablesen der Zähler erfolgt meistens manuell. Aus Kostengründen sind die Ableseintervalle sehr groß. Dadurch ist es nicht möglich, den Verbrauchsverlauf über ein Jahr, einen Monat oder einen Tag zu erfassen sowie Verbrauchsoptimierungen, z.B. eine bessere Lastverteilung, vorzunehmen.

Sowohl für Versorger als auch für Kunden besteht das Interesse, den Zählstand automatisch in wesentlich kleineren Intervallen zu erfassen.

## 2. Anforderungen

Die folgenden Anforderungen werden an ein System zur optischen Zählwerterfassung gestellt.

- **Keine Manipulation am Zähler:** Aus Gründen der Sicherheit (Gas) sowie zur Unterbindung von Betrug ist es gesetzlich verboten, an den Zählersystemen Veränderungen vorzunehmen. Es ist nicht zulässig, das Gehäuse zu öffnen bzw. elektronische Komponenten direkt am Zähler einzusetzen. Diese Einschränkungen können eingehalten werden, wenn der Zählstand

optisch erfasst und ausgewertet wird, da am geeichten Zähler selbst keine Manipulation notwendig ist.

- **Drahtlose Steuerung:** Für beide Anwendergruppen, Versorger und Kunden, ist es ein entscheidendes Kriterium, unabhängig vom Standort den Zählstand auslesen zu können. Als Gründe hierfür kann genannt werden, dass dadurch kein Personal zum Ablesen anreisen muss sowie die freie Wahl des Zeitpunkts der Auswertung.
- **Verschiedene Zählertypen:** Da das System bei Verbrauchszählern in unterschiedlichen Bereichen (Gas, Wasser, Elektrizität) zum Einsatz kommen soll und diese Zähler ganz unterschiedlich gestaltet sind, muss das System so konzipiert sein, dass es für verschiedene Zählervarianten einfach angepasst werden kann.
- **Preis:** Ein weiteres Kriterium für das System als marktreifes Produkt ist ein niedriger Preis, um eine rasche Verbreitung erreichen zu können. Für die Prototypenentwicklung spielt dieses Kriterium zunächst eine untergeordnete Rolle, da in erster Linie das Funktionsprinzip zu beweisen ist. Ein Redesign mit Fokus auf Kostenersparnis ist obligatorisch und erfolgt nach erfolgreichem Prototypentest.

### 3. Aufbau

#### 3.1. Prinzipieller Aufbau

Der grundsätzliche Aufbau besteht aus (siehe Abb. 1)

- einer Kamera, die den Zählerstand als Bild erfasst,
- einem Mikrocontroller, der das Bild aus der Kamera ausliest und an den
- Auswerterechner überträgt.

Die Bildauswertung kann prinzipiell sowohl auf dem Mikrocontroller als auch auf dem Auswerterechner erfolgen. Der Vorteil der Auswertung auf dem Mikrocontroller ist, dass nur sehr wenige Daten an den Auswerterechner übertragen werden müssen. Nachteilig ist aber die höhere Rechenleistung, die der Mikrocontroller für die Bildauswertung zur Verfügung stellen muss. Beim Prototyp erfolgt die Bildauswertung auf dem Auswerterechner, weil die Entwicklung der Software wesentlich schneller möglich ist.

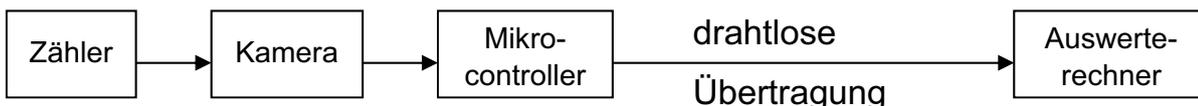


Abb. 1: Prinzipieller Aufbau der optischen Zählerstandserfassung

Unter Berücksichtigung der vorgegebenen Kriterien lassen sich die weiteren Arbeitsschritte auf Hardware und Software aufteilen.

## 3.2. Hardware

Sowohl die Auswahl des Kamerasystems als auch die der Beleuchtung des Objekts spielen hierbei eine entscheidende Rolle. Es ist notwendig, die verschiedenen Kameratechnologien gegenüberzustellen und die richtige Kamera zu wählen. Um eine gute Beleuchtung zu gewährleisten, wird ein Beleuchtungsdom angefertigt.

Für den entwickelten Prototyp wurde ein RaspberryPi Computer eingesetzt, welcher außer der drahtlosen Kommunikationsanbindung weitere Aufgaben wie Kamera- und Beleuchtungssteuerung übernimmt.

- **Kamerasystem:** Für den Prototyp wurde eine c920 HD Webcam der Firma Logitech verwendet, die über USB kommuniziert. Bei einem Redesign ist ein preiswertes CMOS Kameramodul eingeplant. Dabei kann auch die Schnittstelle gewechselt werden.
- **Mikrocontroller:** Der Prototyp wurde auf Basis eines RaspberryPi angefertigt.

## 3.3. Software

Das Herzstück der optischen Zählwerterfassung bildet die Software. Dabei sind unter anderen folgende Punkte hervorzuheben:

- **Bildverarbeitung:** Das gewonnene Bild des Zählstands muss soweit optimiert werden, dass es durch eine Texterkennungssoftware ausgewertet werden kann. Zum Einsatz kommt hierfür das Open-Source Softwarepaket OpenCV.
- **Texterkennung:** Um aus dem aufbereiteten Bild den Zählwert extrahieren zu können, ist eine Texterkennung notwendig. Verwendet wird das von Google weiterentwickelte Tesseract OCR, welches in diesem Fall auf die Erfassung von Zahlen spezialisiert wird.
- **Datenübertragung:** Zur Datenübertragung zwischen dem Bilderfassungssystem und dem zentralen Rechner, an dem die Texterkennung durchgeführt wird, gibt es verschiedene Optionen. In diesem Fall wird die Datenübertragung über das Secure Shell (SSH-2) Netzwerkprotokoll realisiert. Dieses Protokoll erlaubt eine verschlüsselte Kommunikation über das Internet. Die Datenübertragung ist dabei über das SSH File Transfer Protocol (SFTP) möglich.

## 4. Realisierung

### 4.1. Beleuchtungsart

Für das Projekt der optischen Zählwerterfassung hat sich nach mehreren Versuchsreihen gezeigt, dass eine Square-Tunnelbeleuchtung (Abb. 2) am geeig-

netsten ist. Ein entscheidender Grund für die Auswahl der Square-Tunnel Beleuchtung war, dass im Gegensatz zur direkten Domebeleuchtung eine größere Fläche (Zähleroberfläche ist meist rechteckig) gleichmäßig belichtet werden kann. Dies ist z.B. dann von Vorteil, wenn neben dem aktuellen Zählwert auch der Zählertyp ermittelt werden soll. Ein weiteres Auswahlkriterium war die „eckige“ Konstruktion des Gehäuses. Dadurch können Reflexionen, die bei Testreihen mit anderen Gehäusetypen (z.B. runder Aufbau) aufgetreten sind und die Auswertung des Bildes per Software deutlich erschwert haben, eliminiert werden.

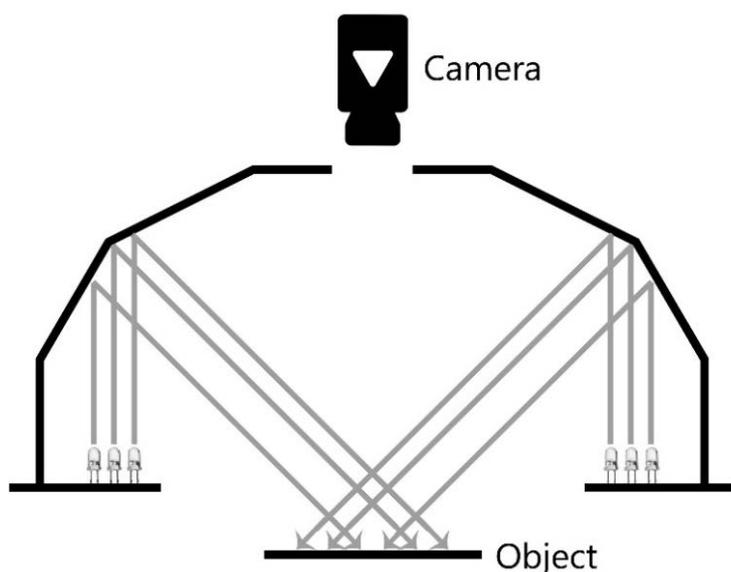


Abb. 2: Square-Tunnel Beleuchtung

#### 4.2. Bildverarbeitung und Texterkennung

Die digitale Bildverarbeitung bildet das Fundament für eine korrekte Texterkennung. Aus diesem Grund ist es wichtig, ein entzerrtes Bild, welches lediglich die wesentlichen Informationen enthält, an Tesseract (optical character recognition (OCR) Software Engine von Google) zu übergeben. Der folgende Algorithmus wurde entwickelt und wird nachfolgend beschrieben:

- Originalbild
- Graustufenbild
- Binarisiertes Bild mittels adaptivem Thresholding
- Aufbereitung des Graustufenbildes mittels Blurfilter
- Erzeugung eines Kantenbildes
- Konturensuche im erstellten Kantenbild
- Verstärkung der gefundenen Konturen durch Sobelfilter
- Konturenfilterung mit Hilfe von Bounding Boxes
- Entzerrung des Bildes von Fehlerkonturenflächen (Pixelflächenfilterung)
- Endbilderzeugung mittels Maskierung
- Auslesen des Bildes mittels OCR Tesseract

Im ersten Schritt wird der Software eine Bildquelle zur Verfügung gestellt. Bei diesem Prototyp wurde eine Webcam verwendet. Das von der Kamera aufgenommene Farbbild wird anschließend in ein 8-Bit Graustufenbild konvertiert. Es ist dabei unerheblich, ob der abzulesende Zählerstand in weißer Schrift auf schwarzem Grund ist (Zählerstand in Abb. 3), oder in schwarzer Schrift auf weißem Grund (Zählernummer in Abb. 3). Es muss lediglich vor der Auswertung bekannt sein, welche Zahl ausgewertet werden soll.

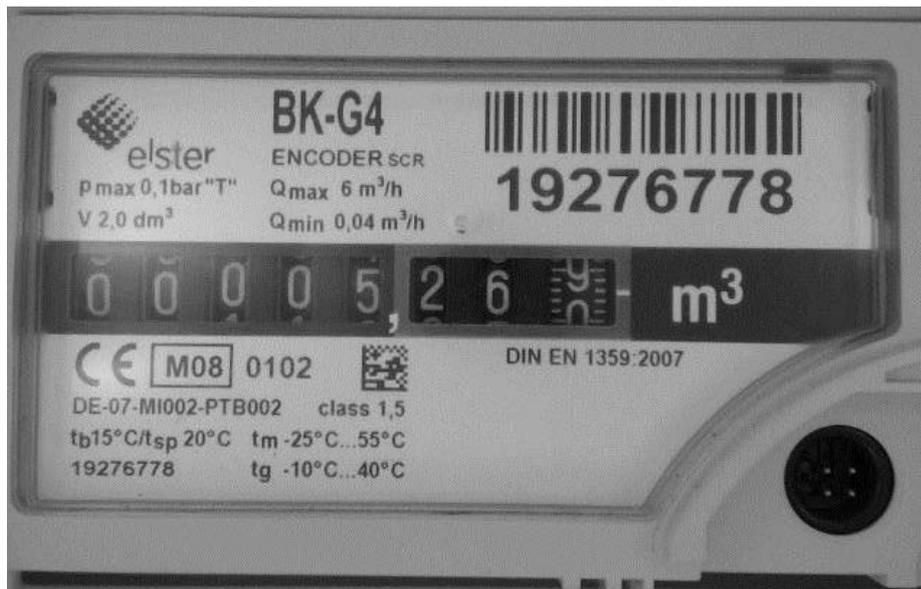


Abb. 3: Graustufenbild

Dieses Vorgehen ist notwendig, da ein Binärbild nur aus einem Graustufenbild erzeugt werden kann. Durch Auswahl einer Region of Interest (ROI) kann der gewünschte auszulesende Bereich festgelegt werden (Abb. 4). Das erstellte Binärbild bildet eine Maskierungsgrundlage für die Endbilderzeugung, worauf im späteren Verlauf weiter eingegangen wird.



Abb. 4: Binärbild der Region of Interest (ROI)

Das erstellte Binärbild wird mit einer adaptiven Threshold-Methode verarbeitet, welche Störeinflüsse durch Lichtreflexionen grob herausfiltern kann. Nachfolgend wird das gewonnene Graustufenbild mit einer Filteranwendung entört, wobei der sogenannte Blurfilter bzw. Gaußfilter zum Einsatz kommt. Dieser dient zur Entstörung des Bildes, da das gewonnene Bild durch die begrenzte Auflösung im Detail unsaubere Übergänge enthält. Auf dieser entörteten Aufnahme wird der Canny-Algorithmus angewandt, welcher die Kanten eines Bildes selektiert und diese anschließend mittels OpenCV-Funktionen visualisiert (Abb. 5).



Abb. 5: Kantenbild mit Canny-Algorithmus

Die gefundenen Kanten enthalten teilweise keine sauberen Übergänge, da in einer weißen Pixelreihe vereinzelt schwarze Unterbrechungspixel zu finden sind. Diese Unterbrechungen sind mit weißen Pixeln aufzufüllen, um eine vollständige Kante zu erhalten. Eine Möglichkeit, diese Unterbrechungspixel herauszufiltern, ist der Sobel-Filter. Er verstärkt die Kantenlinien mit einem vorgegebenen Faktor und schließt dadurch die durch die schwarzen Pixel entstandenen Lücken (Abb. 6).



Abb. 6: Kantenverstärkung mit Sobelfilter

Aus dem resultierenden Kantenbild werden die enthaltenen Konturen der Aufnahme herausgeschnitten und vorerst in einem Datenarray hinterlegt. Für das Ausschneiden der Konturen können diverse Einstellmöglichkeiten konfiguriert werden. Diese Datensammlung enthält alle Konturzüge des Bildes unter Angabe der Form, Länge, Breite und des Start- und Endpunktes der einzelnen Konturen. Über jede einzelne Kontur wird anschließend eine Bounding Box gelegt (Abb. 7). Hierbei handelt es sich um rechteckige Boxen, die die jeweilige Kontur der einzelnen Arrayelemente umfasst.



Abb. 7: Bounding Boxes visuell dargestellt

Diese Boxen enthalten ebenfalls eine Längen- und Breitenangabe der Box, welche nachfolgend mit Werten verglichen werden kann. Anhand dieser Informationen kann entschieden werden, ob die jeweilige Länge und Breite des einzelnen Elements zu klein oder zu groß für eine Zahl ist.

Somit gelingt es, fast alle Störelemente, welche keine Zahlen sind, aus der Aufnahme zu eliminieren (Abb. 8).



Abb. 8: Filterung mit Bounding Boxes

Durch die Anwendung der Bounding Boxes können grob betrachtet alle zu großen Konturen aus dem Bild entfernt werden. Da es jedoch auch vorkommen kann, dass kleine Störelemente entfernt werden müssen, wird eine Pixelflächenfilterung eingefügt, welche speziell für die Eliminierung dieser kleinen Störelemente ausgerichtet ist. Hierbei wird die eingeschlossene Fläche der jeweiligen Kontur mit dem Wert 255 (weiß) gefüllt. Die weißen Pixel werden mit einer OpenCV Funktion gezählt und anschließend mit einem einstellbaren Zahlenwert verglichen. Hierbei durchlaufen alle Konturen diesen Ausfüllvorgang, was jedoch auch dazu führt, dass die richtigen Zahlen, beispielsweise bei der Zahl Null, komplett ausgefüllt werden (Abb. 9).



Abb. 9: Pixelflächenfilterung

Durch die bereits vorgestellten Bildbearbeitungsvorgänge gelingt es, die kleineren und größeren Fehlkonturen eines Bildes zu entfernen. Das so entstandene Bild enthält nun die reinen Zahlenkonturen in ausgefüllter weißer Farbe (Abb. 9).

Um die Richtigkeit der inneren Zahlenkonturen wiederherzustellen (Ziffern „0“ und „6“ in Abb. 9), wird das aktuelle Bild mit dem binarisierten Bild aus Abb. 4 maskiert. Ein schwarzer Pixel entspricht dabei einer logischen „0“ und ein weißes Pixel dem Wert „1“. Durch eine einfache UND-Operation der beiden Bilder werden somit die inneren Konturen, welche bereits im binarisierten Bild in Abb. 4 sauber dargestellt sind, wiederhergestellt. Dieser letzte Schritt ist notwendig, um ein entstörtes Bild zu erhalten, welches abschließend an die Tesseract-Engine zum Auswerten übergeben werden kann.



Abb. 10: Endmaskierung

## 5. Ausblick

Mithilfe des entwickelten Algorithmus lässt sich, wie in Abb. 10 zu sehen, ein gut aufbereitetes Binärbild erstellen, welches durch die Texterkennungssoftware Tesseract OCR ausgewertet werden kann. Der Machbarkeitsnachweis, dass eine optische, nicht-invasive Zählwerterfassung durchgeführt werden kann, ist somit erbracht.

Dennoch gibt es noch offene Punkte sowohl bei Hard- als auch Software, welche zur Komplettierung des Projektes bearbeitet werden müssen.

Hinsichtlich der Hardware ist der bestehende Prototypenaufbau zu überarbeiten. Der Beleuchtungsdom ist kompakter zu konstruieren. Zudem sind Befestigungsmechanismen für eine breitere Auswahl an Zählersystemen anzufertigen. Die Elektronik ist in das Gesamtsystem zu integrieren. Das Raspberry-Pi-Entwicklungsboard kann durch eine anwendungsspezifische Platine ersetzt werden.

Der Entwicklungsstand der Software erlaubt bereits jetzt die Durchführung reproduzierbarer Messreihen. Jedoch ist die Robustheit und Zuverlässigkeit der Software noch zu verbessern. Eine Nutzeroberfläche der Software ist für mobile Plattformen wie Smartphones oder Tablets zu implementieren. Dies würde dem Anwender eine komfortable Inbetriebnahme der automatischen optischen Zählererfassung ermöglichen.

Die drahtlose Kommunikationsschnittstelle zum Kamerasystem ist dahingehend zu erweitern, dass die Beleuchtungs- und Kamerasteuerung von der Anwendersoftware übernommen werden kann.

## Literatur

- [1] Parker, J. R.: Algorithms for Image Processing and Computer Vision. Wiley, 2010.
- [2] Engelhardt, E. F.: Foto und Video mit RaspberryPi. Franzis, 2014.
- [3] Bradski, Gary; Kaehler, Adrian: Learning OpenCV Computer Vision with the OpenCV Library. O'Reilly, 2008.

## Weiterführende Weblinks

- [4] <https://code.google.com/p/tesseract-ocr/> - Tesseract Projektseite.
- [5] <http://opencv.org/> - OpenCV Projektseite.
- [6] <http://www.libssh2.org/> - libssh2 Projektseite.

## Kontakt

Prof. Dr.-Ing. Mathias Rausch  
Hochschule für angewandte Wissenschaften Landshut  
Am Lurzenhof 1  
84036 Landshut  
E-Mail: [rausch@haw-landshut.de](mailto:rausch@haw-landshut.de)