

Lib-Indexer

EDBV WS 2019/2020: AG_C_3

Anand Eichner (11808244)

Laurenz Edmund Fiala (11807869)

Anna Nieto-Berezhinskaya (01223066)

Aleksandar Vucenovic (01635282)

5. Januar 2020

1 Gewählte Problemstellung

(1-1,5 Seiten)

entspricht dem (aktualisierten) Konzept

1.1 Ziel

Das Projekt soll Bücher in einem Bücherregal erkennen, in Bücher-Koordinaten umwandeln und nach ihrem Label abspeichern.

1.2 Eingabe

JPG-Bild eines Bücherregals mit Büchern, auf denen eindeutige TU-Bibliotheks-Labels (schwarz-auf-weiß) kleben.

1.3 Ausgabe

visuell:



Abbildung 1: Input-Bild

textuell:

Strukturierte Klartext-Datei mit Inhalt:

- Standort der Bücher (in Büchern zum Ursprung - links-oben)
- Vier Pixel-Vektoren, die ein Label in einem Viereck umschließen

1.4 Voraussetzungen und Bedingungen

- Die Bücher müssen gerade (+/- 5°) stehen.
- Das Bild darf nicht mehr als 30° von der Waagrechte abweichen.
- Das Bild muss eine für die Texterkennung der Labels ausreichende Auflösung aufweisen (Abhängig von der Entfernung).
- Das Bild muss farbig sein.
- Das Bild muss ausreichend hell sein.
- Ein Weißabgleich muss durchgeführt worden sein.

Neue Anforderungen:

- Der Hintergrund der Labels muss weiß sein (Intensität muss über dem globalen Otsu-Threshold liegen)

1.5 Methodik

Methodik- Pipeline

1. Hough-Transformation
Finden der Regalfächer zum Korrigieren der Perspektive
2. Perspektivenkorrektur
tuwe Mittels Transformationsmatrix aus HT berechnet

3. Eckenerkennung
Finden der Ecken von Labels
4. Integral imaging
Finden von Labels innerhalb eines Akzeptanzbereichs, es werden nur Bereiche zwischen verschiedenen, zuvor erkannten, Ecken überprüft
5. Eigene Heuristik
Einordnen von Labels in Buch-Koordinaten
6. Optical Character Recognition
Erkennen von Text auf den Labels in den zuvor erkannten Bereichen

1.6 Evaluierungsfragen

Interaktion zwischen den Komponenten:

- *Werden die Regalfächer korrekt erkannt?*
Voraussetzungen: Seite der Regalfächer, die zur Kamera zeigt, ist schwarz.
Ergebnis: An jedem Fach liegt eine Gerade an.
- *Wird die Perspektive korrekt angepasst?*
Voraussetzungen: Korrekte Geraden der Regalfächer.
Ergebnis: Bücher und Labels sind im Bild weitestgehend rechteckig.
- *Werden alle Labels erkannt?*
Voraussetzungen: Perspektivenkorrigiertes Bild
Ergebnis: Bounding-Boxes der gefundenen Labels
- *Sind die Bounding Boxes der gefundenen Labels korrekt? Ist der gesamte Text darin enthalten?*
Voraussetzungen: Korrekt erkanntes Label oder ein false-positive.
Ergebnis: Vier Vektoren, die den gesamten Text umschließen (und nicht mehr). Bei false-positives ist das Ergebnis nicht relevant, jedoch sollte es nicht zu groß sein (z.B. das gesamte Bild überdecken).
- *Werden die Labels korrekt in Bücher-Koordinaten umgewandelt?*
Voraussetzungen: Bounding Boxes der Labels sind korrekt.
Ergebnis: Bücher-Koordinatensystem als 2D-Array mit Ursprung links-oben.
- *Werden die Labels der TU-Bibliothek korrekt gelesen und in Text umgewandelt?*
Voraussetzungen: Label Bounding-Boxes wurden korrekt berechnet (enthalten keinen unnötigen Text).
Ergebnis: String-Repräsentation des Labels. Erwartete Korrektheit: > 90% für typische Datensätze.

- Wird die Wahrscheinlichkeit der Label-Korrektheit angemessen berechnet?
 Voraussetzungen: Korrekt in Text umgewandeltes Label.
 Ergebnis: Floating-point Wert im Intervall [0, 1]. Alle Labels mit Wahrscheinlichkeit 0 wurden entfernt.

1.7 Zeitplan

Meilenstein	abgeschlossen am	Arbeitsaufwand in h
Prototyp	10.11.	30
Hough-Transformation	17.11.- 20.12.	50 2
Perspektivenkorrektur	17.11.- 23.12.	6 20
Labelerkennung	25.11.- 14.11.	25
Labelerkennung - Verbesserungen	25.11.- 5.1.	30
Labels in Buch-Koordinaten	1.12.- 4.1.	4 0.2
Optical Character Recognition	10.12.- 3.1.	110 40
Labels filtern	15.12. nicht abgeschl.	6 nicht abgeschl.
Daten in Output-Format umwandeln	5.1.	4 0
Tests	18.12.- 5.1.	5
Evaluierung	20.12. 5.1.	5 4

2 Arbeitsteilung

Name	Tätigkeiten
Anna Nieto-Berezhinskaya	Harris Eckendetektor (zu feinfühlig und kann deshalb nicht eingesetzt werden)
Anand Eichner	houghTransform (+ lokale Funktionen), PerspectiveCorrection (+ lokale Funktionen)
Laurenz Edmund Fiala	label_detection (+ lokale Funktionen),
Aleksandar Vučenović	ocr (+ locals), NCC, SSD_naive (+ lokale Funktionen), preprocessing (+ lokale Funktionen)

3 Methodik

(2-3 Seiten)

Hier wird die verwendete Methodik in der Theorie vorgestellt:

Welche Methodik wurde verwendet? Warum eignet sich diese Methodik für die gewählte Problemstellung? Habt ihr Methoden verändert (Einschränkungen, Abwandlungen, Parameter), wenn ja wie? etc.

Die erwähnten Methoden werden zum größten Teil auf Beschreibungen in Büchern oder wissenschaftlichen Artikeln beruhen. Daher ist hier auch der richtige Platz für Zitate. Die hier zitierten Publikationen sollten mittels Abkürzung bzw. Nummer referenziert sein und sich in der Referenzliste am Ende des Berichts über diese Bezeichnung finden lassen.

Ein Beispielsatz (inkl. entsprechender Literaturangabe am Ende des Berichts): Interest Points wurden mittels Scale Invariant Feature Transform [?] detektiert.

Bei der Verwendung von Latex gestaltet sich das Zitieren besonders einfach - siehe Beispielssatz im Source der Latex-Vorlage.

Wichtig in diesem Abschnitt ist, dass sich der Leser Eures Berichts mit den verwendeten Methodiken auskennt und weiß, weshalb ihr diese Methodiken verwendet habt und keine anderen. Es soll dem Leser helfen den nächsten Abschnitt des Berichts besser zu verstehen.

Label Detection

Harris-Eckendetektor [?]

Der Harris-Eckendetektor wird genutzt, um die Ecken der einzelnen Labels zu erkennen. Ursprünglich war geplant, diese Methode selbst zu implementieren, jedoch war die selbst implementierte Variante zu sensibel und das konnte aufgrund von fehlenden Projektmitgliedern nicht mehr ausgebessert werden.

Summed Area Table / Integral Imaging [?]

Das Integral Imaging wird benutzt, um die Intensität der einzelnen Label-Kandidaten effizient berechnen zu können. Im Programmverlauf wird das genutzt, um das Verhältnis von dunklen zu hellen Bildabschnitten zu berechnen und damit bestimmen zu können, ob ein Bereich als Label gelten kann oder nicht.

Otsu-Threshold [?]

Es wird ein globaler Threshold benötigt um Labels vom Hintergrund abzugrenzen. Das Integral Image beruht auf dem Ergebnis der Bildmaske dieses Thresholds. Ein Otsu-Threshold eignet sich sehr gut um dynamisch in jedem Bild einen akzeptablen Threshold zu ermitteln.

Der Threshold wird auch lokal pro Label-Kandidat angewandt, um den oft sehr hellgrauen Text vom Hintergrund (dem Label) abzugrenzen. Damit ist das Ergebnis des Programms wesentlich zuverlässiger.

4 Implementierung

(1-X Seiten)

Hier gebt ihr einen Überblick über Eure Implementierung:

Wie habt ihr die im vorhergehenden Abschnitt vorgestellte Methodik praktisch umgesetzt? Wie werden die einzelnen Methoden kombiniert (zB. Implementierungspipeline)?

Hier ist Platz für Implementierungsdetails wie zB. gewählte Parameter.

Wie startet der User das Programm? Welche Parameter hat der User zu setzen?

Auch in diesem Abschnitt können Referenzen und Zitate notwendig sein.

Wichtig in diesem Abschnitt ist, dass der Leser Eures Berichts versteht wie

ihr Euer Projekt in MATLAB umgesetzt habt um sich auch im Quelltext leichter zurechtfinden zu können.

Label Detection

Zuerst werden mit einem Harris-Eckendetektor (MATLAB-builtin) die Ecken im Bild gefunden [?]. Danach wird ein globaler Otsu-Threshold berechnet [?], der genutzt wird, um eine binäre Bildmaske zu erstellen, wobei alle Pixel mit größerer Intensität als der Threshold, auf 1 gesetzt werden, und alle anderen auf 0.

Mit dieser Maske wird wiederum ein Summed Area Table erstellt [?].

Für alle gefundenen Ecken wird nun versucht Nachbarn zu finden, die ein plausibles Label ergeben könnten. Dabei wird aus dem Vorschritt der Perspektivenkorrektur die Höhe der Bücherregale berechnet und daraus die ungefähren Label-Dimensionen abgeleitet. Für alle gefundenen Ecken-Nachbar-Kombinationen wird nun überprüft, ob der Schwarzanteil dem eines typischen Labels entspricht. Dafür wird ein lokaler Otsu-Threshold berechnet, der den Label-Text vom Hintergrund trennen soll. Im Falle eines schrägen Labels, werden mittels der MATLAB-Funktion `imclearborder` eventuell vorhandene Ränder ignoriert. Dann wird mit dieser Maske die anzahl an schwarzen Pixeln summiert und in Verhältnis zu dem Ergebnis des Integral Imaging (weiße Bildteile) verglichen. Wenn dieser Wert im Akzeptanzbereich liegt, nehmen wir an, im Bereich zwischen der Ecke-Nachbar-Kombination befindet sich ein Label.

Um die Labels zu erkennen, haben wir uns aufgrund der verschiedenen Label-Dimensionen am Anfang des Projekts dazu entschieden, mittels Integral Imaging (Summed Area Table) zu bestimmen, ob ein Bereich als Label gelten kann oder nicht.

Der Prototyp war auch recht erfolgversprechend. Jedoch sind wir im Verlauf des Projekts auf einige Limitationen bei dieser Herangehensweise gestoßen.

Es hat sich herausgestellt, dass es oft nicht ganz trivial ist, bei verschiedenen Auflösungen die korrekten Ecken mit dem Harris-Eckendetector zu finden. Hier spielt das Preprocessing, welches ein einfacher Gaußfilter ist, eine große Rolle. Auch die Größe des Gauß-Konvolutionskernels für den Harris-Detektor ist wichtig. Aufgrund von Zeitmangel gegen Ende des Projekts, konnten diese Werte leider nicht mehr so genau bestimmt werden und deshalb ist das Ergebnis leider nicht so genau, wie eigentlich möglich bzw. gewünscht.

5 Evaluierung

(2-X Seiten)

Hier stellt ihr Euren Datensatz vor und beantwortet Evaluierungsfragen:

z.B. Fakten zum Datensatz: Anzahl der Bilder, Größe der Bilder, Quelle des Datensatzes (falls selbst aufgenommen: Aufnahmegerät, Einstellungen,... / falls nicht selbst erstellt: Datenbank vorstellen... → Referenzen!)

Diskussion der Evaluierungsfragen: Beantwortung der Fragen, Diskussion anhand von Beispielen, Diskussion von Grenzfällen: für welche Bilder funktioniert die Implementierung, für welche nicht? Worin unterscheiden sich diese Bilder? Warum funktionieren sie nicht? etc.

Evaluiert wird der ganze Datensatz, nicht nur einzelne Bilder. Einzelne Bilder können zum Aufzeigen von Fehlern/Problemen/besonders guten Ergebnissen... genutzt werden.

Zur Evaluierung gehört auch das Testen der einzelnen Methodiken (separat), mit Erwähnung eventueller Einschränkungen.

6 Schlusswort

(max. 1 Seite)

Hier fasst ihr Ergebnisse Eures Projekt zusammen:

Welche Schlussfolgerung lässt sich ziehen? Gibt es offene Probleme? Wie lässt sich Eure Lösung noch verbessern? etc.

Literatur

- [1] Mike Stephens Chris Harris. A combined corner and edge detector. *Alvey Vision Conference*, 1988. Zugriff 28.10.2019; Verantwortliche: Anna Nieto-Berezhinskaya, Laurenz E. Fiala.
- [2] Franklin C. Crow. Summed-area tables for texture mapping. 1984. Zugriff 12.11.2019; Verantwortliche: Laurenz E. Fiala.
- [3] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Computer Graphic*, 1979. Zugriff 5.1.2019; Verantwortliche: Laurenz E. Fiala.

Webseiten werden als Fußzeilen (an jener Stelle wo sie verwendet werden) eingebunden, nicht als Literature!