

# Automatisierte Aufbereitung archäologischer Grabungsfotos mittels Computer Vision

Simon Metzger

## **Masterarbeit**

zur Erlangung des akademischen Grades Master of Arts im  
Studiengang Digitale Methodik der Geistes- und  
Kulturwissenschaften

Johannes-Gutenberg-Universität Mainz und Hochschule Mainz

## **Zusammenfassung**

Im Format abstract

# Inhaltsverzeichnis

<b>1 Einleitung</b>	<b>2</b>
1.1 Grabung Kapitol . . . . .	2
1.2 Datensatz vorstellen . . . . .	2
1.3 Material vorstellen . . . . .	2
1.3.1 Tafeln . . . . .	2
1.3.2 Tafelvergleiche . . . . .	4
1.3.3 Schrift . . . . .	4
1.4 Pipeline . . . . .	5
<b>2 Objektdetektion</b>	<b>6</b>
2.1 Detektionsverfahren . . . . .	6
2.1.1 Feature Detection . . . . .	6
2.1.2 CNN . . . . .	6
2.1.3 Contours . . . . .	6
2.2 Cropverfahren . . . . .	14
2.2.1 simple crop . . . . .	14
2.2.2 Hough . . . . .	14
2.3 Experimentelles . . . . .	14
2.4 Zusammenfassung . . . . .	14
<b>3 Texterkennung</b>	<b>15</b>
3.1 Theorie Texterkennung . . . . .	15
3.2 Software: Tesseract . . . . .	15
3.3 Pre-Processing . . . . .	15
3.4 OCR . . . . .	15
3.5 Evaluation . . . . .	15
<b>4 Ergebnisse</b>	<b>17</b>
<b>5 Fazit</b>	<b>17</b>

# 1 Einleitung

Einleitung und Fragestellung  
[Hough, 1962]

## 1.1 Grabung Kapitol

Grabungsverlauf bis 2014 (recherchieren)  
Übernahme durch DAI (recherchieren)

## 1.2 Datensatz vorstellen

Herkunft  
Umfang  
Fragestellungen des Projektes

## 1.3 Material vorstellen

### 1.3.1 Tafeln

Die Verwendung von Tafeln zur Dokumentation von Fund- und Grabungsarealen ist in allen, im weitesten Sinne grabenden, Wissenschaften weit verbreitet. So setzt auch die Archäologie diese Methode ein. Dabei werden neben den zu dokumentierenden Gebieten verschiedenste Formen von Tafeln oder Schildern platziert, auf denen Zeit und Ort der Aufnahme sowie weitere bild- und motivbezogene Informationen festgehalten werden können. Der Vielfalt von Form und Material der Tafeln ist dabei keine Grenze gesetzt.

Bei den Tafeln, die Gegenstand dieses Projektes sind, handelt es sich um Schiefertafeln mit einem Holzrahmen, die mit Kreide beschriftet wurden (Vgl. Abb 1). Für die Detektion der Tafeln ergeben sich daraus folgende Faktoren:

1. Die Tafeln haben grundsätzlich eine rechteckige Form.
2. Durch die Breite des Rahmens können bis zu zwei Rechtecke erkannt werden, ein Inneres und ein Äußeres.
3. Durch die große Differenz zwischen dem hellen Holzrahmen und der dunklen Schieferplatte sollte der innere Rand in der Regel gut detektierbar sein.



Abbildung 1: Beispiel eines Fotos der verwendeten Tafel. GOT bezeichnet die Kampagne, darunter folgt das Datum. US ist die Abkürzung für *unità stratigrafica*, die stratigrafische Einheit.

Die im Beispielbild gezeigte Tafel stellt ein Idealbild dar: Die Tafel nimmt einen relativ großen Teil des Originalbildes ein. Sie ist frontal vor der Kamera positioniert. Die Beleuchtung ist gut und indirekt. Keines der weiteren Bildelemente verdeckt die Tafel. Diese Beschreibung impliziert schon die Problemfelder, die bei der Detektion beachtet werden müssen:

1. Die Tafel ist unter Umständen stark rotiert (Vgl. Abb 2).
2. Die Distanz der Tafel zur Kamera und damit ihre Größe im Bild kann stark variieren.
3. Der Rahmen der Tafel kann teilweise verdeckt oder anderweitig durch Gegenstände überlagert sein (Vgl. Abb 2).
4. Die Farbe des Tafelrahmens kann dazu führen, dass sie sich nicht klar vom Hintergrund abhebt, was die Detektion des (äußeren) Randes erschweren kann.
5. Unregelmäßigkeiten im Rahmen, die auf grobe Verarbeitung oder Abnutzung zurückzuführen sind, können die Detektion erschweren.
6. Die Beleuchtung kann zu Problemen führen. Grundsätzlich sind alle Fotos hell und gut ausgeleuchtet, direktes Licht kann sich aber negativ auf die Kontraste auswirken.
7. Weitere Gegenstände, die den Spezifika der Tafeln entsprechen, können im Bild vorhanden sein.



Abbildung 2: Schwierigere Detektion: Rotation und teilweise verdeckter Rahmen.

Teilweise werden die hier genannten Probleme auch bei der Texterkennung wieder relevant. Auf diese und auf weitere wird an geeigneter Stelle zurückgegriffen.

### 1.3.2 Tafelvergleiche

Im Rahmen der Arbeit wurden weitere Tafeln exemplarisch dem Algorithmus unterzogen. Dabei handelte es sich um Aufnahmen der späteren Grabungen des Deutschen Archäologischen Instituts am Kapitol in Rom sowie um vergleichbare Fotos von Bodenuntersuchungen der Gruppe Terrestrische Ökohydrologie der Friedrich-Schiller-Universität Jena. Der ursprüngliche Gedanke dahinter war eine möglichst universale Detektion von Tafeln aller Art anzustreben. Die unterschiedlichen Daten konnten dabei vor allem Stärken und Schwächen der letztlich gewählten Technik aufzeigen.

Die Tafeln beider Projekte sollen im Folgenden kurz vorgestellt werden, um das Spektrum der Komplexität evtl. Vergleiche zu Tafeln aus späterer Grabung als Positivbeispiel:

besser gearbeitete Tafeln

besser lesbare Schrift

evtl. Vergleiche zu Tafeln der Bodenkunde als Negativbeispiel:

Tafel schwierig durch Form und Farbe

Klarsichthülle: Reflektion und Formveränderung

oft verdeckt

Bilder zur Veranschaulichung einfügen

### 1.3.3 Schrift

Kreide auf Schiefer Probleme wie Handschrift, Verwischung, Karomuster

## **1.4 Pipeline**

Struktur der Arbeit wie Pipeline: Bildakquise Objekterkennung Crop-Verfahren  
Pre-Processing OCR Evaluation Ergebnis

## 2 Objektdetektion

Das folgende Kapitel befasst sich mit den ersten beiden Schritten in der automatisierten Analyse der Grabungsfotos: der Erkennung der Schiefertafeln und ihrer Extraktion aus dem Gesamtbild.

Zunächst werden verschiedene Möglichkeiten der Erkennung präsentiert und diskutiert. Der Schwerpunkt liegt hier auf dem schlussendlich umgesetzten Verfahren. Abschließend wird das mit der Tafeldetektion verbundene Ausschneiden der gefundenen Tafeln aus dem Gesamtbild vorgestellt.

### 2.1 Detektionsverfahren

Aus den Anforderungen der Aufgabenstellung lässt sich als erster Arbeitsschritt die Detektion der Tafeln ableiten. Die wichtigste Prämisse ist dabei, dass Falsch-Negative, also nicht erkannte Tafeln, vermieden werden. Grund dafür ist, dass diese für die weitere Bearbeitung komplett verloren sind. Falsch-Positive sollten ebenfalls weitestgehend ausgeschlossen werden. Diese können jedoch in späteren Bearbeitungsschritten, vor allem der Texterkennung, erkannt und aussortiert werden. Daher ist dieses Kriterium von niedrigerer Priorität. Dieses Kapitel wird sich daher verschiedenen Verfahren widmen, mit denen rechteckige, beschriftete Objekte erkannt werden können. Diese Verfahren werden vorgestellt und ihre Ergebnisse in einer ersten, explorativen Umsetzung präsentiert. Anschließend wird begründet, warum diese Verfahren im Rahmen des Projektes nicht nicht zum Einsatz kommen. Final wird dann das Kontur-basierte Erkennungsverfahren erläutert, mit dem die besten Ergebnisse erzielt wurden.

#### 2.1.1 Feature Detection

Was ist Feature Detection? Wie funktioniert sie? Was war die Idee hinter dem Ansatz? Wie sehen die Ergebnisse aus?

#### 2.1.2 CNN

Hier werden CNNs vorgestellt. Was sind CNNs? Was können sie, wie funktionieren sie? Warum habe ich sie ausprobiert, was war die Idee dahinter? Warum habe ich nicht selbst trainiert? Wie sehen die Ergebnisse aus?

#### 2.1.3 Contours

Cv2.Contours basiert auf einem Algorithmus, der Punkte gleicher Farbe und Intensität umrandet [Suzuki and Abe, 1985]. Das Resultat ist eine Liste von Punkten, die ein geschlossenes Polygon ergeben. Die gefundenen Konturen können mit

einer Hierarchie versehen werden, bei der Konturen, die sich innerhalb anderer Konturen befinden, als deren „Kinder“ gelten. Das Verfahren ist darauf ausgelegt, auf binäre Bilder angewendet zu werden. Die Implementierung in OpenCV unterscheidet in Nullwerte und Nicht-Nullwerte [OpenCVDocumentation, ]. Nicht-binäre Bilder werden also automatisch binarisiert. Für die Erzielung optimaler Ergebnisse kommt dem Binarisierungsverfahren allerdings eine große Bedeutung zu. Auf Basis dieses Algorithmus lässt sich Detektionsverfahren aufbauen, das Objekte, die sich vom Hintergrund der Bilder abheben, zu erkennen. Die Herausforderung besteht, nach diesem Ansatz, in zwei Punkten: Erstens muss die Binarisierung so erfolgen, dass eine möglichst saubere Trennung von Vordergrund (Objekten) und Hintergrund (vor allem Erde) der Bilder stattfindet und zweitens müssen aus den gefundenen Vordergrundobjekten diejenigen ausgewählt werden, die als Tafeln in Frage kommen. Wie bereits erwähnt hat dabei die Vermeidung Falsch-Negativer eine höhere Priorität als die von Falsch-Positiven. Bei der Binarisierung haben sich zwei Wege als praktikabel herausgestellt, die im Folgenden beide präsentiert werden sollen. Im Flowchart sind die Abläufe der Rechtecksdetektion schematisch dargestellt (Vgl. Abb.3).

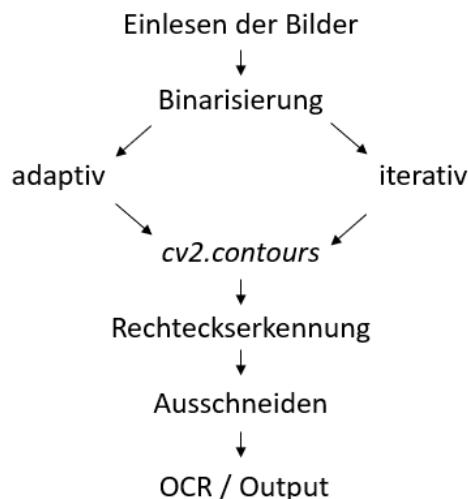


Abbildung 3: Flowchart Rechteckserkennung.

### Adaptive Binarisierung

Das einfachste Verfahren der Binarisierung eines Bildes besteht darin, einen Threshold, also einen Grenzwert festzulegen. Dieser muss auf der Spanne der Farbwerte, also zwischen 0 und 255 liegen. Farbwerte unterhalb dieses Thresholds werden

zu Nullen, Farbwerte darüber zu Einsen. Das Ergebnis ist ein reines Schwarz-Weiß-Bild. Bei komplexen Szenerien, wie den Grabungsfotos, ist dieses Verfahren jedoch zu einfach. So kann ein Foto beispielsweise stark unterschiedliche Beleuchtung, wie direktes Sonnenlicht und Schatten, enthalten. Eine Differenzierung innerhalb dieser Zonen ist so nicht möglich (Vgl. Abb. 4).

Eine Lösung für dieses Problem ist die Anwendung eines adaptiven Thresholds

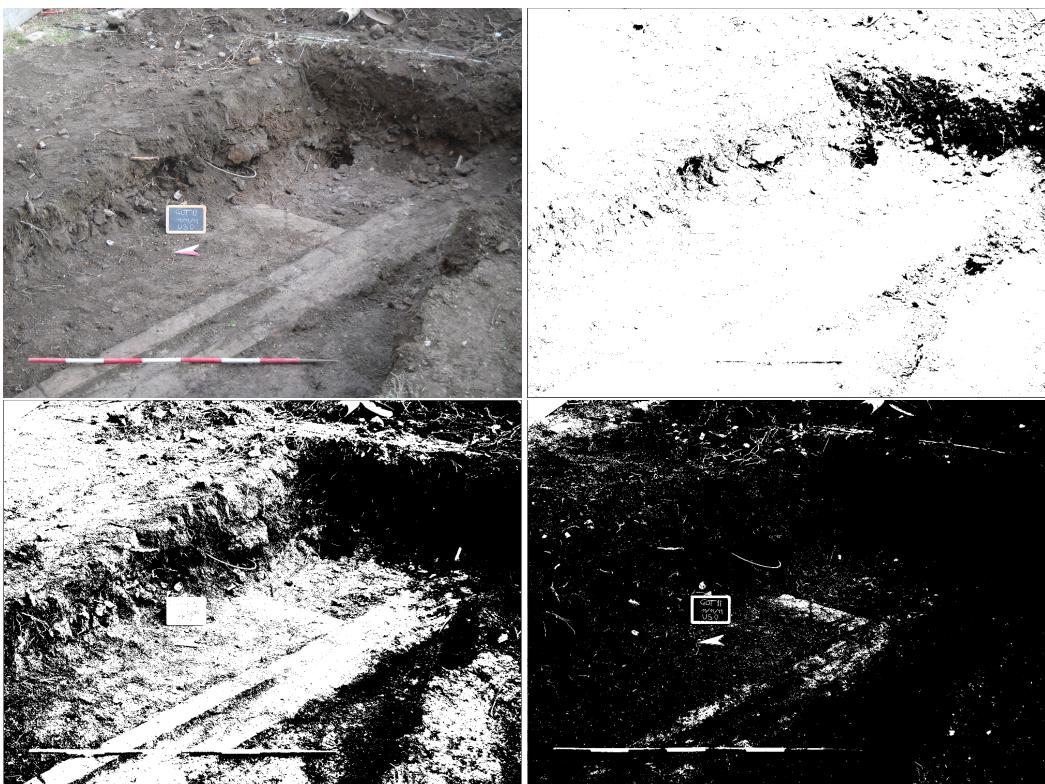


Abbildung 4: Grabungsfoto im Original (o.l.), mit niedrigem (60, o.r.), mittlerem (125, u.l.) und hohem (185, u.r.) Threshold.

[?]): Statt global, über das gesamte Bild, einen Grenzwert festzulegen, können lokale Grenzwerte errechnet werden. Die Größe des lokalen Ausschnittes sowie das exakte Verfahren können dabei frei gewählt werden. In diesem Fall wird für die Binarisierung ein Gauss-Verfahren auf einen Kernel von  $11 \times 11$  Pixeln angewendet. Die Beleuchtung oder Farbunterschiede innerhalb des Bildes können so ausgeglichen werden (Vgl. Abb. 5). Vor der Binarisierung muss das Bild in ein Grauskala-Bild umgewandelt verkleinert<sup>1</sup>. Das verbessert die Genauigkeit der Detektion und verringert die zu verarbeitende Datenmenge, wodurch der Prozess

<sup>1</sup>Genauer: Der längere Bildrand wird auf 1000 Pixel reduziert, der kürzere entsprechend angepasst. Die Grabungsfotos haben in der Regel eine Auflösung von 3264 x 2448 Pixeln. Es findet



Abbildung 5: Grabungsfoto im Original sowie mit adaptivem Threshold.

beschleunigt wird. Die Verkleinerung des Bildes ist allerdings ein Prozess, der später rückgängig gemacht werden muss, um Datenverlust bei der Texterkennung zu verhindern.

### Iterative Binarisierung

Die adaptive Binarisierung bringt einige Nachteile mit sich. So ist hier die Erkennung von Falsch-Positiven relativ hoch. Die Konturen werden auf einem verkleinerten Bild gesucht und müssen später auf Originalgröße skaliert werden, was ein verlustbehafteter Prozess ist. Daher gibt es einen zweiten Ansatz, hier als iterativer Ansatz bezeichnet. Die Grundidee besteht darin, das Bild mit einem globalen Threshold zu binarisieren. Die Problematik davon wurde bereits diskutiert: Bei großen Beleuchtungsunterschieden oder sehr hellen oder dunklen Objekten auf dem Bild werden ganze Bereiche durch den Threshold von der weiteren Bearbeitung ausgeschlossen. Der iterative Ansatz sieht daher vor, den Threshold von 20 auf 200 in Fünferschritten zu erhöhen. Dadurch entstehen pro Foto 37 binäre Bilder, auf die die Konturenerkennung angewendet werden kann. Auf jedem dieser Bilder können mehrere mögliche Tafeln erkannt werden<sup>2</sup>. Der nächste Schritt besteht also darin, aus diesen möglichen Tafeln die auszuwählen, die am wahrscheinlichsten tatsächlich eine ist. In diesem Kontext wird eine Grundannahme

---

also eine Reduktion auf ca.  $\frac{1}{9}$  der Fläche statt. Bilder kleiner als 1000 Pixel würden theoretisch auf 1000 Pixel vergrößert werden. Da für die hier beschriebenen Prozesse und vor allem für die Texterkennung später aber eine gewisse Bildqualität erforderlich ist, ist von diesem Fall nicht auszugehen.

<sup>2</sup>Die eigentliche Tafelerkennung wird erst im folgenden Abschnitt beschrieben. Da die dort gewonnenen Informationen nicht, wie beim adaptiven Ansatz, an das Hauptprogramm übergeben, sondern innerhalb der Funktion des iterativen Ansatzes weiter verarbeitet werden, ist hier ein Vorgriff nötig.

getroffen: Es wird davon ausgegangen, dass in dem Bereich, in dem auf den meisten der 37 Bilder eine Tafel vermutet wird, sich tatsächlich eine Tafel befindet. Alle anderen werden als Falsch-Positive betrachtet. Diese Annahme beruht auf zwei Faktoren: Erstens hat sich gezeigt, dass Objekte, die keine Tafeln sind, aber als solche erkannt werden können – beispielsweise Fenster, Türen oder Plakate – nur unter wenigen Thresholds als solche eingeordnet werden. Das liegt unter anderem darin begründet, dass die Tafeln einen hellen Holzrahmen und eine dunkle Innenfläche haben, was zum maximalen Kontrasten führt. Zweitens werden durch eben diesen Rahmen in mittleren Threshold-Bereichen die Tafeln zweimal erkannt: Einmal an der Außenkante und einmal an der Innenkante des Rahmens. Dadurch häuft sich die Detektion möglicher Tafeln in diesem Bereich. Basierend auf dieser Annahme wird auf alle möglichen Tafeln immer paarweise die `intersection_over_union` angewendet. Dieser Algorithmus basiert auf dem Jaccard-Koeffizienten zur Berechnung der Ähnlichkeit zweier Mengen [Jaccard, 1902]. Der Koeffizienten wird errechnet, indem die Schnittmenge durch die Vereinigungsmenge geteilt wird. Das Ergebnis liegt zwischen 0 und 1. Je mehr es sich der 1 annähert, desto ähnlicher sind die Mengen. Diese Berechnung lässt sich auch auf die Rechtecke, mit denen die möglichen Tafeln verortet werden, anwenden (Vgl. Abb. 6).

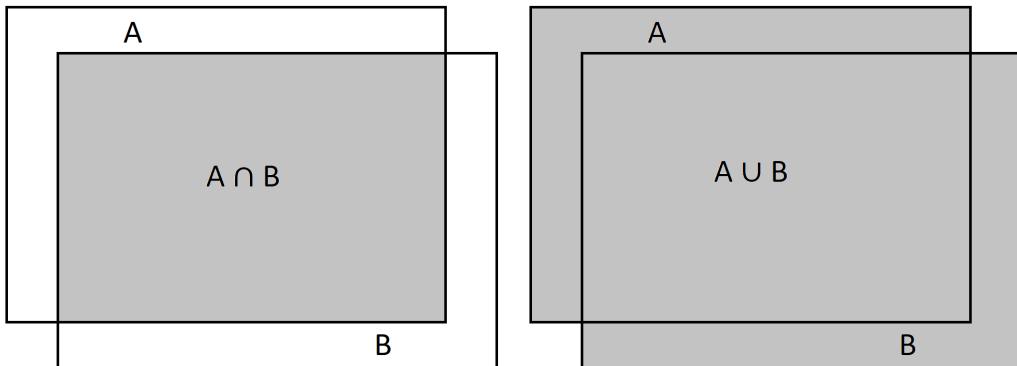


Abbildung 6: Schnittmenge und Vereinigungsmenge.

Im Rahmen dieses Projekts gelten zwei Rechtecke dann als hinreichend ähnlich, wenn der Jaccard-Koeffizient über 0.9 liegt. Jedes Mal, wenn ein Rechteck im Vergleich mit einem anderen diesen Wert erzielt, wird für das Rechteck ein Zähler erhöht. Das Rechteck, welches am Ende der Vergleiche den höchsten Wert in diesem Zähler erzielt, wird als tatsächliche Tafel angenommen. Eine Ausnahme ergibt sich, wenn dieser Wert unter 6 liegt. In diesem Falle, so hat sich ergeben, handelt es sich ausschließlich um Falsch-Positive. Je höher der Wert, desto sicherer ist die Identifikation der Tafel. Unter guten Bedingungen erreicht er Werte um

die 20.

### Rechteckserkennung

Ist die Binarisierung der Bilder nach einer der beiden vorgestellten Methoden erfolgt, können mittels `cv2.findContours` die Konturen der Objekte darauf gefunden werden (Vgl. Abb. 7). Die Hierarchie der Konturen wird dabei außer Acht gelassen, da sich daraus keine verlässlichen Informationen gewinnen lassen. Als nächstes müssen unter diesen Konturen die ausgewählt werden, die als Tafel



Abbildung 7: Grabungsfoto im Original sowie nach der Anwendung der Konturerkennung.

in Frage kommen. Dazu wird die Tatsache genutzt, dass es sich bei den Tafeln um Gegenstände handelt, die auf den Fotos als Rechtecke abgebildet werden. Durch ihre Lage zur Kamera können sie zu einem gewissen Grad davon abweichen, grundsätzlich bleibt diese Form aber annähernd erhalten. Andere Rechtecke kommen zwar vor, wie Plakate, Fenster, Türen, Ziegel und Fliesen, unterscheiden sich jedoch meist signifikant in der tatsächlichen Form oder können im Zweifelsfall bei der späteren Texterkennung ausgeschlossen werden. Dementsprechend geht es in der weiteren Tafelerkennung darum, Rechtecke zu finden und diese durch verschiedene weitere Kriterien so zu sortieren, dass alle Tafeln und möglichst nur Tafeln erkannt werden. Das wird in mehreren Schritten erreicht: Zunächst wird die Fläche der Konturen berechnet. Genutzt wird hierfür die OpenCV-Funktion `cv2.contourArea`. Ist diese Fläche zu klein, wird die Kontur aussortiert. Als Vergleichswert wird hier die Kantenlänge der längsten Kante des Bildes genommen, damit bei höherer Auflösung weiterhin korrekt sortiert werden kann (Vgl. Abb. `reffig:adaptivecontsize`). Eine Größenbegrenzung ist sinnvoll, da die Tafeln in der Regel eher prominent im Bild zu sehen sind. Sollte tatsächlich eine Tafel durch dieses Kriterium aussortiert werden, wäre der Text darauf nicht mehr lesbar

und somit ohnehin nicht interessant für die weitere Verarbeitung.

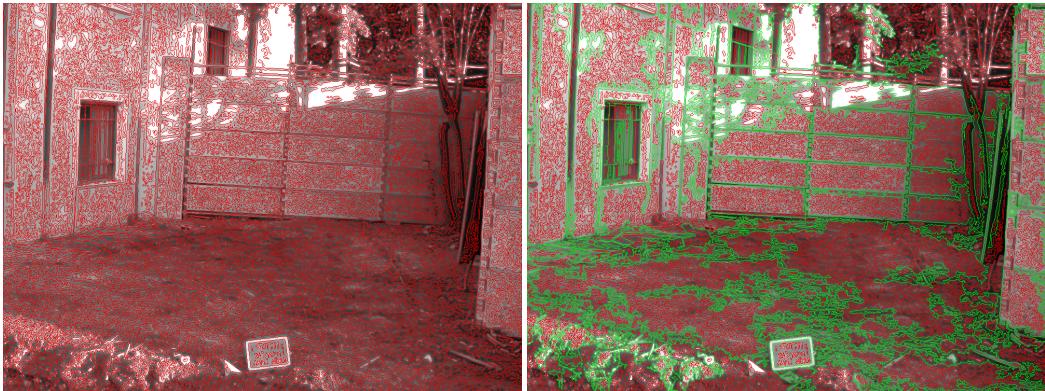


Abbildung 8: Konturen vor und nach der Selektion nach Fläche.

Der nächste Schritt besteht darin, das kleinstmögliche Rechteck um die Kontur zum bilden. Auch hier kommt mit `cv2.minAreaRect` eine Bibliotheksfunktion zum Einsatz (Vgl. Abb. 9).

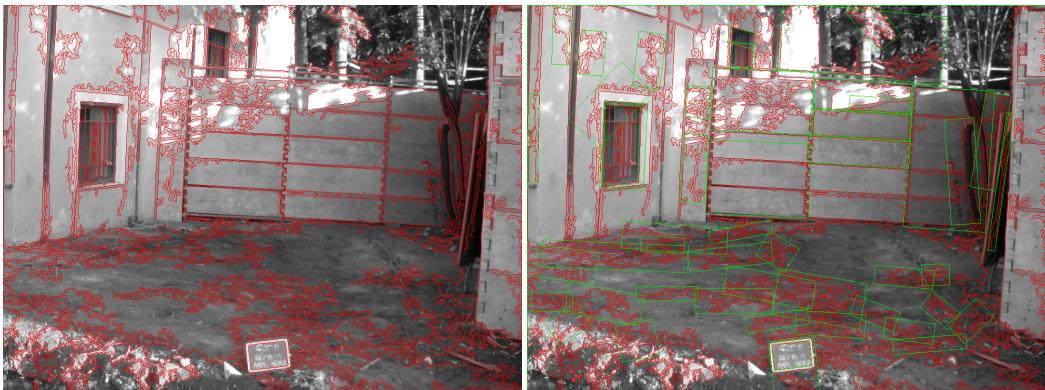


Abbildung 9: Konturen ohne und mit Rechtecken.

Anschließend werden die bereits berechneten Flächen der Konturen mit denen der kleinstmöglichen Rechtecke verglichen. Die Annahme: Nähert sich der Flächeninhalt der Kontur dem des Rechtecks an, handelt es sich bei der Kontur selbst wahrscheinlich um ein Rechteck. Als Grenzwert hat sich ein Verhältnis von 0.85 bewährt. Ebenfalls ausgeschlossen wird ein Rechteck, wenn es die Fläche des gesamten Bildes ausmacht, da teilweise, wie auch bei diesem Beispiel, der Rand des Bildes als Kontur erkannt werden kann (Vgl. Abb. 10).

Als letztes Kriterium wird das Seitenverhältnis herangezogen. Im Programm besteht die Möglichkeit, vor der Untersuchung aller Bilder ein Template zu hinterlegen, auf dem eine Tafel gut und eindeutig zu sehen ist. Aus diesem Template



Abbildung 10: Rechtecke aller Konturen und jene derer mit annähernd rechteckiger Fläche.

wird das Seitenverhältnis der Tafel berechnet. Findet dieser Schritt nicht statt, wird ein maximales Seitenverhältnis von 2:1 angenommen, da Tafeln selten in länglichen Formaten zu finden sind<sup>3</sup>. Mit einer deutlichen, aber bewährten Toleranz von 30% in jede Richtung wird das Seitenverhältnis der verbliebenen Rechtecke mit dem des Templates verglichen. Sind die Werte sich ähnlich genug, wird das Rechteck als Tafel interpretiert und gilt als das Ergebnis des Algorithmus(Vgl. Abb. 11).



Abbildung 11: Die bisher verbliebenen Rechtecke und die mit dem richtigen Seitenverhältnis.

<sup>3</sup>Sollte das Format einer Tafel doch einmal von dieser Annahme abweichen, lässt sich das problemlos entsprechend anpassen.

## 2.2 Cropverfahren

Was ist die Aufgabe beim Crop? Worin liegen hier die Schwierigkeiten? Auch hier wieder Zweispurigkeit der Ansätze erklären

### 2.2.1 simple crop

Was ist die Idee? Wie wurde sie umgesetzt? Wo liegen die Probleme?

### 2.2.2 Hough

Was ist die Idee? Wie wurde sie umgesetzt? Wo liegen die Probleme?

## 2.3 Experimentelles

## 2.4 Zusammenfassung

evtl zusammenfassen wie vorgegangen wurde, warum dieser Weg gut ist und was das Wichtigste Ergebnis ist

## 3 Texterkennung

### 3.1 Theorie Texterkennung

Ursprünge der Texterkennung

Aktueller Forschungsstand

Wechsel auf machine learning

Schwerpunkte

Handschrift vs. Druckschrift

### 3.2 Software: Tesseract

evtl ALternativen

Tesseract: was kann es, wie funktioniert es

Wechsel von Charactererkennung via CV zu Zeilenerkennung via machine learning

wie funktioniert die Box-detection?

### 3.3 Pre-Processing

Preprocessing: besondere Herausforderungen, vorgehen, beide Varianten vorstellen

### 3.4 OCR

normales Modell

eigenes Modell

Vergleich: Tafeln aus späterer Grabung (gesetzte Lettern)

evtl. Vergleich Tafeln Bodenkunde

### 3.5 Evaluation

(Kapitel evtl. vorziehen wegen Bedeutung für den ganzen Prozess -*i* auch als Maß für Objektdetektion)

Evaluation: Vorgehen, Überlegungen

Verwendete Kennzahlen

Vergleichbarkeit der Ergebnisse

## **4 Ergebnisse**

Ergebnisse aus dem kompletten Datensatz präsentieren

## **5 Fazit**

Auswertung des geschriebenen Codes Materialkritik Ausblick und weitere Ideen

## Literatur

- [Hough, 1962] Hough, P. V. C. (1962). Method and Means for Recognizing Complex Patterns. *U.S. Patent*, (3,069,654).
- [Jaccard, 1902] Jaccard, P. (1902). Tois de distribution florale dans la zone alpine. *Bulletin de la Société Vaudoise des Sciences Naturelles*, (38):72.
- [OpenCVDocumentation, ] OpenCVDocumentation. Structural analysis and shape descriptors.
- [Suzuki and Abe, 1985] Suzuki, S. and Abe, K. (1985). Topological Structural Analysis of Digitized Binary Images by Border Following. *COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING*, (30):32–46.