

# **Schnapskönig**

**EDBV WS 2017/2018: AG\_A2**

Jan Michael Lajarno (01425799)

Andreas Brunner (01429369)

Miran Jank (01526438)

Thorsten Korpitsch (01529243)

Aleksandar Marinkovic (01634028)

4. Januar 2018

## **1 Gewählte Problemstellung**

### **1.1 Ziel**

Beim Schnapsen treten 2 Spieler gegeneinander an. Zu Beginn bekommt jeder 5 Karten vom Stapel, danach wird jede Runde von beiden Spielern eine Karte gezogen. Jede Runde wird jeweils 1 Karte von jedem Spieler auf den Tisch gelegt wobei beide Spieler versuchen einen Stich zu machen. Stechen bedeutet, dass die eigene Karte entweder ein Adut ist und die Karte des gegenüber keines ist, beziehungsweise, dass die eigene Karte einen höheren Wert hat als die des Gegenübers. Es wird solange gespielt bis der Stapel leer ist und beide Spieler keine Karten mehr auf der Hand haben, beziehungsweise einer der Spieler Karten im Wert von 66 Punkten gestochen hat. Sind alle Karten ausgespielt worden, wird der Gewinner ermittelt indem die gewonnenen Punkte beider Spieler gezählt werden und danach der mit der höheren Punktezahl gewinnt.

### **1.2 Eingabe**

Der Benutzer muss pro Spielzug ein Farbbild, in einem gängigen Format(.PNG/.JPG/.JPEG), der Karten in das Programm laden.

### **1.3 Ausgabe**

In der Konsole wird ein Zwischenstand nach jeder Runde ausgegeben, am Ende wird der Gewinner ausgegeben und der Endstand.

## 1.4 Voraussetzungen und Bedingungen

Ein Farbbild der Karten. Der Hintergrund sollte möglichst Einfarbig sein (nicht weiß, texturarm). Die Kamera soll sich in einem Winkel von 45 bis 135 Grad befinden. Die Karten müssen mit einem dünnen schwarzen Rand preperiert sein. Die obere Karte darf maximal 45% der anderen Karte überdecken.

## 1.5 Methodik

1. Threshold nach Otsu: Um in Kombination mit Zusammenhangskomponenten die Karten zu trennen
2. Geometrische Transformation: Um das Eingabebild vorzubereiten wird aus dem, bis zu 45 Grad schrägen Bild, ein Bild aus der Vogelperspektive (90 Grad) transformiert, die hochkant steht
3. Template-Matching: Um die Spielkarte zu identifizieren

## 1.6 Evaluierung

- Werden beide Karten erkannt?
- Wird das Bild richtig transformiert, oder werden Buchstaben/Symbole verzerrt?
- Wird die Karte richtig identifiziert?

## 1.7 Datenbeispiel



Abbildung 1: Karo-Ass sticht Kreuz-Koenig (Farbzwang), 16 Punkte gewonnen



Abbildung 2: Karo-König sticht Karo-Dame, 7 Punkte gewonnen

## 1.8 Zeitplan

Meilenstein	abgeschlossen am		Arbeitsaufwand in h	
	geplant	tatsächlich	geplant	tatsächlich
Vorarbeit Projekt	20.10.2017	21.10.2017	20	23
Prototyp erstellen	10.11.2017	01.12.2017	40	80
Geometrische Transformation	22.12.2017	27.12.2017	105	92
Threshold nach Otsu	18.12.2017	25.12.2017	67	60
Pattern-Matching	18.12.2017	30.12.2017	43	30
GT-Test	31.12.2017	31.12.2017	10	12
TH-Test	27.12.2017	29.12.2017	5	5
PM-Test	27.12.2017	29.12.2017	5	3
Unit-Test	05.01.2017	04.01.2017	5	2

## 2 Arbeitsteilung

Name	Tätigkeiten
Jan Michael Lajarno	Matlab: Prototyp, ausführbares Programm, Bericht: Methodik/Implementierung(Template Matching), Gewählte Problemstellung
Andreas Brunner	Matlab: Prototyp, tmc.m Bericht: Arbeitsteilung, Evaluierung
Miran Jank	Matlab: geometricTransform Bericht: Methodik(Geometrische Transformation)
Thorsten Korpitsch	Matlab: Prototyp, splitCards.m, thresholdOtsu.m Bericht: Evaluierung, Methodik/Implementierung (Vorverarbeitung, Trennen der Karten), Schlusswort
Aleksandar Marinkovic	Matlab: Prototyp, geometricTransform Bericht: Implementierung(Geometrische Transformation)

## 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), 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 Beispielsatz im Source der Latex-Vorlage.

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

### 3.1 Threshold nach Otsu

Die Aufgabe für die erste verwendete Methode ist das Trennen der Karten vom Hintergrund. Hier haben wir uns für ein Schwellenwertverfahren entschieden. Bei einem

Schwellenwertverfahren wird ein Graustufenbild in ein Binärbild umgewandelt. Wir haben uns für den Threshold nach Otsu entschieden, da es in der Kombination mit der Zusammenhangskomponente besonders dazu eignet um Objekte vom Hintergrund zu trennen.

Beim Threshold nach Otsu wird versucht, die beiden Segmente (Vordergrund/Hintergrund) so kompakt wie möglich zu machen und die Überschneidung gering zu halten.

Um das Trennen der Karten zu vereinfachen schränken wir die Eingabe dahingehend ein, dass beim Bild der Karten der Hintergrund nicht weiß sein darf und möglichst texturarm. Diese Einschränkungen machen unseren Threshold nach Otsu stabiler und sehr erfolgssicher beim Trennen der Karten vom Hintergrund.

## 3.2 Geometrische Transformation

## 3.3 Template-Matching

Um die Karte schlussendlich zu identifizieren haben wir uns für das Template Matching entschieden. Das Template-Matching bietet genau die Lösung zu unserer letzten Problemstellung. Mithilfe des Template-Matching versuchen wir herauszufinden ob es sich bei der Karte um Herz, Pik, Kreuz oder Karo handelt und ob es sich um Ass, König, Dame, Bube oder Zehn handelt. Aus der Kombination dieser beiden Symbole können wir die Karte eindeutig identifizieren.

Beim Template-Matching wird versucht ein Bild beziehungsweise einen Bildausschnitt in einem anderen Bild wiederzufinden. User Template-Matching wird mittels einer Correlation-Matrix realisiert, es wird durch alle Möglichkeiten durchiteriert und das passendste wird der Karte zugewiesen.

# 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.**

## 4.1 Vorverarbeitung

Zuerst wird der Datensatz geladen und in ein Graustufenbild umgewandelt.

## 4.2 Karten trennen

Der Thershold ist die erste Methodik in der Methodik-Pipeline, in Kombination mit der Zusammenhangskomponente. Der Funktion `splitCards.m` wird das Eingabebild übergeben. Als erstes wird dieses in ein Graustufenbild umgewandelt und an die Funktion `thresholdotsu.m` übergeben.

Dort werden zuerst die aufkommenden Grauwerte gezählt mittels der `histcounts`-Funktion. Danach wird die gewichtete Summe aller Pixel berechnet, indem man die Anzahl der Pixel mit ihrem jeweiligen Grauwert multipliziert. Danach wird in einer `for`-Schleife durch alle Grauwerte durchiteriert. In jedem Schleifendurchgang werden die Pixel, die den aktuellen Grauwert besitzen, zum Hintergrund dazugerechnet. Der Vordergrund wird neu berechnet indem von der Summe der Pixel, die Pixel vom Hintergrund abzieht. Danach wird die Gewichtete Summe des Hintergrunds berechnet, analog dazu, wie am Beginn die gewichtete Summe aller Pixel berechnet wurde. Jetzt können die Durchschnittswerte des Hinter- und Vordergrunds berechnet werden. Als vorletzten Schritt berechnen wir die Between Class Variance.

Als letztes wird überprüft ob die Between Class Variance, die bisher größte ist. Ist das der Fall, wird der Threshold auf den aktuell iterierten Grauwert gesetzt und die maximale Between Class Variance auf die gerade berechnete gesetzt. Nach dem durchlaufen der Schleife, wird das Bild mittels des optimalen Thresholds in ein Binärbild umgewandelt.



Abbildung 3: Binärbild mittels Threshold nach Otsu

Auf dieses Binärbild wird die Zusammenhangskomponente angewendet um die Karten zu trennen. Aus den 2 größten Zusammenhangskomponenten werden 2 Bilder erstellt, die die Karten repräsentieren. Bei diesen werden noch die Löcher gefüllt. Dies passiert, da wir nicht nur die Karten trennen wollen, sondern auch gleichzeitig herausfinden wol-

len, welche Karte die Obere und welche die Untere ist. Das ganze berechnen wir ganz einfach indem wir die Fläche beider Karten vergleichen und die Karte mit der größeren Fläche die sein muss, die oben liegt, da sie ja einen Teil der unteren Karte verdeckt. Das Füllen der Löcher funktioniert vorallem bei Bildkarten unterschiedlich gut, wie man im nachfolgenden Bild erkennen kann, das kann zu Problemen führen bei der Erkennung welche Karte die Obere und welche die Untere ist.

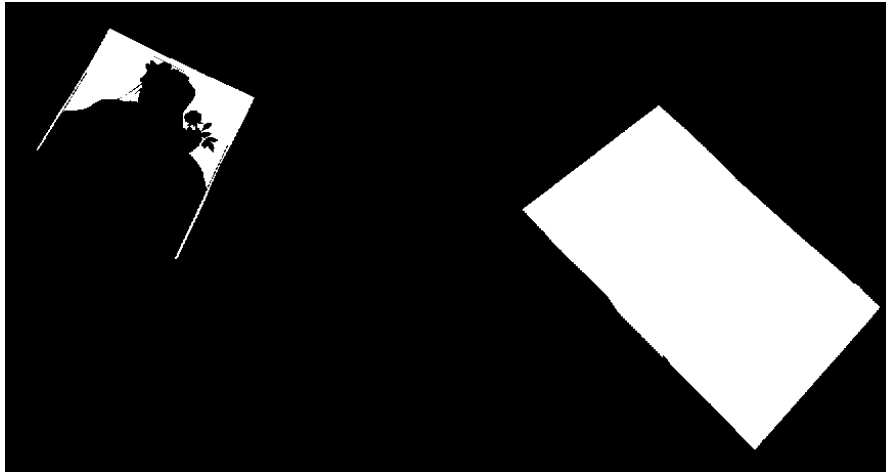


Abbildung 4: Binärbilder der getrennten Karten als Montage. Dame(links), König(rechts)

Danach werden beide Karten an die 2. Methodik weitergegeben, die geometrische Transformation.

## 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

Unsere Schlussfolgerung ist, dass der Anfang des Projekts, also das finden geeigneter Methoden um unsere Problem zu lösen, beziehungsweise einen ersten Prototypen zu entwickeln der schwierigste Schritt für uns waren. Insbesondere das finden der Karten auf dem Bild, bzw. die Findung der Kanten und Ecken um eine Geometrische Transformation im nächsten Schritt zu ermöglichen. Sobald das erledigt war und wir beim Prototyping eine Methode (Canny-Edge-Detection) gegen eine andere ausgetauscht hatten (Threshold nach Otsu) ging es sehr zueigig voran und wir konnten die Zeit die wir aufholen, welche wir beim Prototyping verloren hatten.

Zurzeit gibt es noch Probleme, wenn der Hintergrund zu hell ist, werden die Karten teilweise nicht getrennt, da unser Programm die Karten mittels Threshold nach Otsu und der Zusammenhangskomponente trennt. Wird dann der Hintergrund als Vordergrund erkannt und ist eine größere Zusammenhangskomponente als die kleinere Karte, wird diese nicht mehr erkannt und stattdessen wird der Hintergrund als 2. Karte erkannt.

Eine grosse Verbesserung unserer Loesung waere, wenn die Karten nicht mit einem schwarzen Rand praepariert werden muessten. beziehungsweise wenn man einen Schritt weiter geht, dass nicht ein bestimmtes Kartendeck verwendet werden muss, sondern ein beliebiges (Kartendecks unterscheiden sich oft in der Symbolik, was bei uns dazu fuehrt, dass die Karten nicht mehr richtig identifiziert werden).

## Literatur