

# Optischer Münzenzähler



## Dokumentation

14. Januar 2015

Julien Villiger, Daniel Inversini

Modul BZG1301: Programmierung Matlab/Octave

## Inhalt

Zusammenfassung .....	3
Einleitung .....	4
Vorgehen, Methoden Analysen .....	5
Vorarbeiten .....	5
Bilder aufnehmen .....	5
Grunddaten ermitteln .....	6
Optimales Verhältnis von Bildauflösung zu Abstand der Kamera .....	7
Bildaufbereitung .....	7
Graustufenerkennung .....	7
Binarisierung des Bildes .....	8
Image Fill .....	8
Closing .....	9
Bildanalyse .....	9
Berechnung Radien .....	9
Kantendedektion .....	9
Hough-Transformation .....	10
Template-matching .....	11
Resultate .....	12
Diskussion .....	13
Anhang .....	14
Glossar * .....	15

## Zusammenfassung

Bei dieser Arbeit wurde eine Methode entwickelt, um Münzen auf einer Fotoaufnahme zu erkennen und dessen Werte zu ermitteln. Somit sollte es mit einem Knopfdruck möglich sein, die Summe sämtlicher auf dem Foto abgebildeter Münzen auszugeben.

Es stellte sich die Frage, ob die Bilder (meist aufgenommen mit einem Smartphone) genügend detaillierte Informationen liefern, um den Erkennungsprozess durchzuführen und ob die möglichen Fehlerquellen tolerierbar sind.

Bewahrheitet hat sich die Annahme, dass leichte Verschiebungen des Kamerawinkels und dessen Distanz zu den Münzen einen entscheidenden Einfluss auf das Resultat haben. Diese mussten mit verschiedenen Algorithmen soweit es geht ausgemerzt werden.

Der Erkennungsprozess des Bildes unterliegt etlichen Bildanalyse-Algorithmen und Verarbeitungen wie einer Kantenerkennung [1], Binarisierung [2] und weiteren Prozessen. (Siehe Kapitel Vorgehen)

Resultiert hat eine Matlab [3] Anwendung, welche zuverlässig die Summe der zu erkennenden Münzen ermittelt, vorausgesetzt das Foto entspricht gewissen Kriterien.

# Einleitung

Der Rahmen dieser Arbeit wurde durch uns selbst definiert. Wir steckten uns somit das Ziel, dass ein normales Foto analysiert werden kann.

Somit kann dieses Matlab-Script unter Umständen sehr schnell und unabhängig eingesetzt werden. (Münzen auf dem Tisch auslegen, mit dem Smartphone fotografieren, per Mail/Bluetooth an den PC senden, Script ausführen und Resultat Anzeigen).

Ob dies schneller und effizienter ist, als die Münzen klassisch von Hand zu zählen, können und wollen wir nicht beantworten. Mit dieser kleinen Anwendung kann aber auch informatik-unbegeisterten gezeigt werden, wie mit einfachen Methoden, etwas Kreativität und einem Grundwissen von Bildverarbeitung ein alltägliches Problem gelöst werden kann.

Denkbar wäre demnach auch, grosse Mengen an Münzen automatisch zählen zu lassen, welche die Möglichkeit einer Handzählung übersteigen würden.

Die verwendeten Methoden\* in der Übersicht (geordnet nach Anwendungsschritt):

- Graustufenerkennung [4]
- Binarisierung des Bildes
- Löcher füllen
- Closing (Dilatation / Erosion) [6]
- Kantendetektion
- Hough Transformation [7]
- Templatematching mit Kreuzkorrelation [8]

\*Folgende kleinere subsidiäre Funktionen wurden ebenfalls verwendet, auf diese wird aber nicht speziell eingegangen:

- Statische Grunddaten (Durchmesser und Verhältnis aus einer Config-Datei)
- Timer (zum Messen der Performance)
- Schwarzen und weisse Pixel zählen (Durchmesser ermitteln)
- Matrixoperationen

# Vorgehen, Methoden Analysen

Dieses Kapitel enthält drei Teile:

- Vorarbeiten  
Ein Ausgangsbild wird beschafft, eine Grössen- und Wertetabelle der CH-Münzen werden erstellt.
- Bildaufbereitung  
Das Bild wird zur Analyse und Ermittlung der Münzen-Werte vorbereitet.
- Bildanalyse  
Münzen werden identifiziert und die Werte werden bestimmt.

## Vorarbeiten

### Bilder aufnehmen

Folgende Bilder dienen uns als Entwicklungshilfe, jeweils von uns selbst erstellt:



Abbildung 1: Referenzbild 1a, verschiedene Münzen und Referenzmünze (1-Räppler)

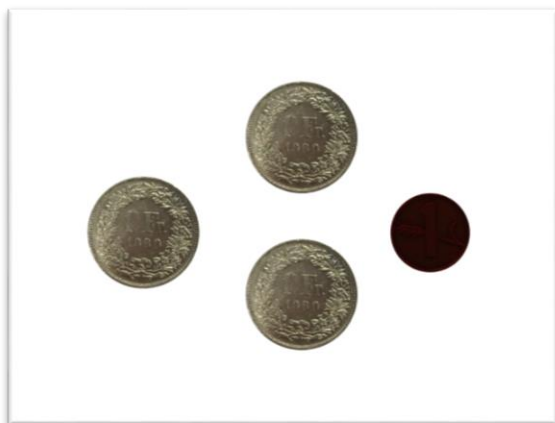


Abbildung 2: Referenzbild 1, verschiedene 1-Fränkler und Referenzmünze (1-Räppler)



Abbildung 3: Referenzbild 1b

Auf allen Referenzbildern ist der 1-Räppler (schwarz eingefärbt) erkennbar.

### Grunddaten ermitteln

Folgende Daten mussten wir uns extern beschaffen\*:

Wert der Münze [SFr.]	Durchmesser in [mm]
5	31.45
2	27.4
1	23.2
0.5	18.2
0.2	21.05
0.1	19.15
0.05	17.15

Tabelle 1: Grössendaten Schweizer Münzen

\*Quelle: [http://de.wikipedia.org/wiki/Schweizer\\_Franken](http://de.wikipedia.org/wiki/Schweizer_Franken)

Dazu kommt noch unsere Referenzmünze, ein Schweizer 1-Räppler:

Wert der Münze [SFr.]	Durchmesser in [mm]
0.01	16

## Optimales Verhältnis von Bildauflösung zu Abstand der Kamera

Da die heutigen Smartphones extrem hochauflösende Fotos schießen, mussten wir einen optimalen Abstand von Kamera zum fotografierenden Objekt (Münzen) finden.

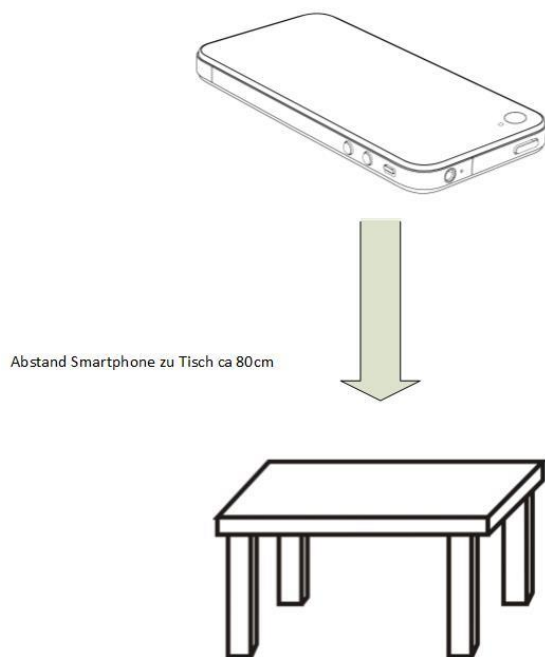


Abbildung 4: Abstand Smartphone zu Tisch

Warum ist dies so wichtig?

Da ein Bild schnell eine Auflösung von 8-11 Megapixel haben kann, können die Flächen der Münzen auch relativ viele Pixel haben. Somit wird es dann später schwieriger, mit einem relativ knappen Threshold [9], ähnlich grosse Münzen zu Unterscheiden.

(Siehe oben, bspw. 50 Rappen und 10 Rappen)  
Durch diesen selbst eruierten Abstand sind die Münzen etwas kleiner (was aber bei der grossen Auflösung nicht weiter ins Gewicht fällt), und eine perspektivische Verzerrung [10] kann auch eliminiert werden.

## Bildaufbereitung

Folgende der bereits erwähnten Methoden verwenden wir, um das Bild für die Analyse vorzubereiten:

### Graustufenerkennung

Mit der Graustufenerkennung suchen wir die Referenzmünze. Diese kann gut über das Histogramm erfolgen, weil sie sich durch die schwarze Einfärbung und dem hellen Hintergrund gut hervorhebt.

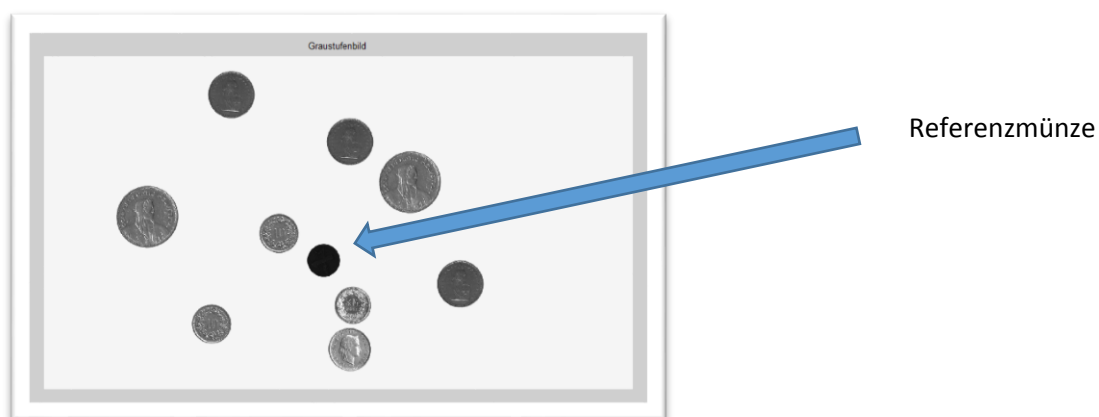


Abbildung 5: Graustufenbild

## Binarisierung des Bildes

Die Binarisierung erfolgt zweimal mit unterschiedlichen Thresholds. Einmal sollen sämtliche Münzen erkennbar sein, einmal bloss die Referenzmünze. Wieso das so ist wird später bei der Bildanalyse erläutert.

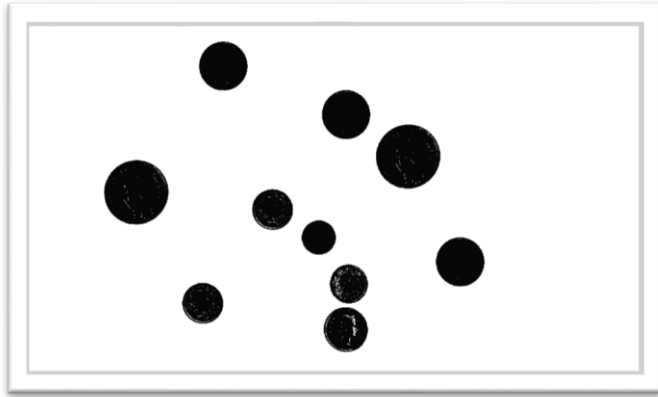


Abbildung 6: Binäres Bild mit hohem Threshold

Mit dem viel tieferen Threshold resultiert nur die schwarze Referenzmünze. Darum ist es wichtig, dass sich der 1-Räppler stark von den anderen Münzen unterscheidet, also eine dunklere Farbe hat.

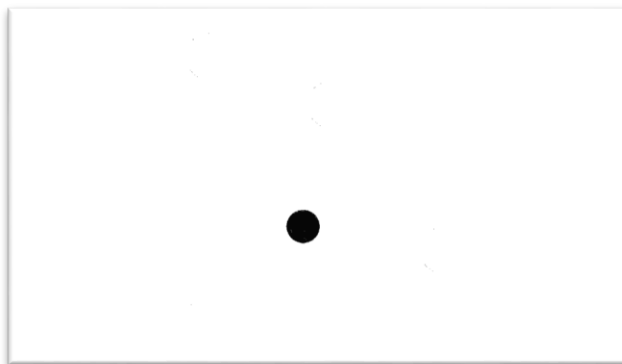


Abbildung 7: Binarisierung mit tiefem Threshold.

Das binäre Bild erleichtert uns später die Kantendetektion. Rein durch das Binarisieren mit dem ermittelten Threshold enthalten die Münzen aber noch weisse, störende Flecken, welche die weiteren Schritte beeinträchtigen könnten.

## Image Fill

Matlab bietet eine Funktion um Löcher in Regionen zu füllen. Löcher können beispielsweise durch den Blitz, also sehr helle reflektierende Regionen auf den Münzen, erzeugt werden. Folgende Methode wird `imfill()` genannt. Ausgewählt werden die Pixel, welche nicht vom Rand aus erreichbar sind. Es werden also nur abgeschottete Regionen aufgefüllt.

Bessere Resultate werden erzeugt, wenn das Image Fill auf das Komplement des Bildes angewandt wird. Deshalb wird das Bild vorgängig zum Image Fill konvertiert und anschliessend wieder zurückkonvertiert.



## Closing

Falls noch weitere Löcher vorhanden sind, werden diese mittels Closing geschlossen. Closing besteht aus morphologischen Transformationen. Zuerst erfolgt eine Dilatation, anschliessend eine Erosion.

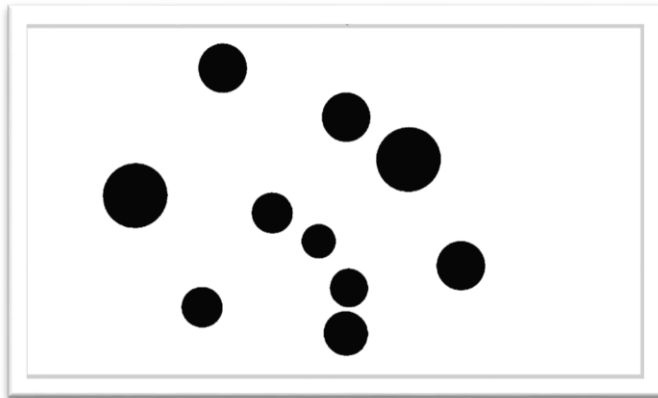


Abbildung 8: Closing

## Bildanalyse

### Berechnung Radien

Nun wird der Radius der Referenzmünze ermittelt. Das geschieht, indem beim binären Referenzmünzenbild sämtliche schwarzen Pixel summiert werden. Das entspricht dann gleich dem Flächeninhalt der Referenzmünze und somit kann der Radius eruiert werden.

Basierend auf dem Pixelradius der Referenzmünze und der Grössentabelle der CH-Münzen können die gesuchten Pixelradien berechnet werden. Alleine die Grössenangaben der Münzen in der Realität reichen nicht aus, weil wir vorgängig nicht wissen, mit welcher Distanz das Foto aufgenommen wurde. Die Grösse auf dem Foto variiert also extrem.

### Kantendetektion

Wir verwendeten den Canny-Algorithmus, um die Kanten zu detektieren. Wir haben uns für diesen entschieden, weil dieser allgemein als bester klassischer Kantenerkennungsalgorithmus anerkannt ist. \*

\* [http://www.mathematik.uni-ulm.de/stochastik/lehre/ws05\\_06/seminar/wagner.pdf](http://www.mathematik.uni-ulm.de/stochastik/lehre/ws05_06/seminar/wagner.pdf)

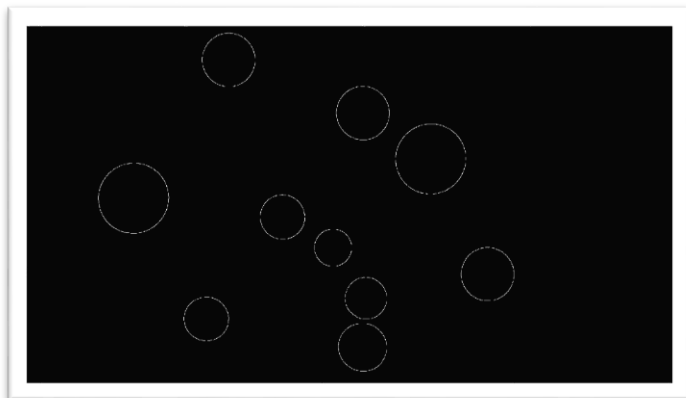


Abbildung 9: Canny Edge

## Hough-Transformation

Nun wurde das Foto der Münzen soweit vorbereitet, um die Hough-Transformation durchzuführen. Voraussetzung dieser Transformation ist, dass der Radius des gesuchten Kreises bekannt ist. Wir führen also mit jedem von uns gesuchten Radius die Hough-Transformation durch. Diese wurden ja im vorgegangenen Schritt ermittelt.

Die Hough-Transformation funktioniert so, dass auf jedem weissen Pixel im binarisierten Bild ein Kreis mit dem gesuchten Radius erzeugt wird. Im Hough-Raum[11] werden diese Kreise kumuliert und dargestellt. Sollten die gezeichneten Kreise dem Radius entsprechen, entsteht ein dunkler Fleck in der Mitte eines Kreises, weil sich dort viele Kreislinien überschneiden. Solche dunklen Flecke bedeuten, dass der Kreis dem gesuchten Radius gefunden wurde.

Diese Transformation haben wir nicht selber geschrieben sondern eine Implementation der Universität Teheran übernommen. Diese liefert uns x- und y-Koordinaten der gefundenen Kreise.

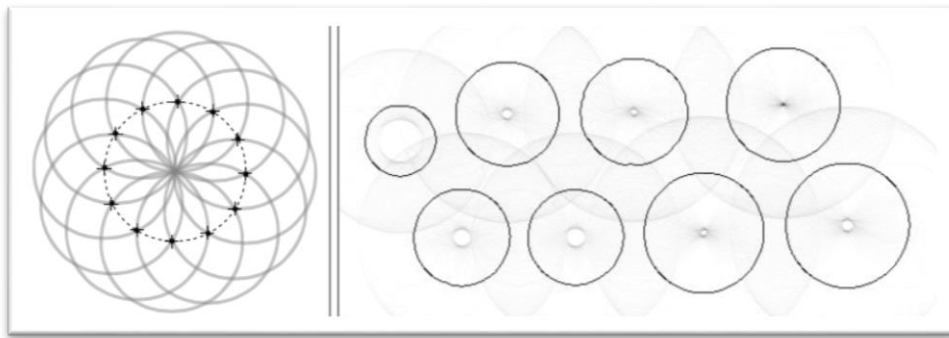


Abbildung 10: Circle Hough-Transformation. Der gesuchte Radius entspricht dem Radius des Kreises oben rechts.

In der Praxis hat sich dieses Vorgehen nur bedingt bewährt. Selbst kleine perspektivische Verzerrungen der Münzen haben dazu geführt, dass sie unter Umständen nicht erkannt wurden.

Um diesem Problem entgegenzuwirken, haben wir für jeden gesuchten Radius die Hough-Transformation mit unterschiedlichem Offset durchgeführt. Eine gesunde Mischung zwischen Performance und guten Resultaten hat sich mit dem Offset (-2 / -1 / 0 / +1 / +2) ergeben. Somit waren wir nicht fix auf einen pixelgenauen Radius fixiert und konnten flexibler die Münzen erkennen.


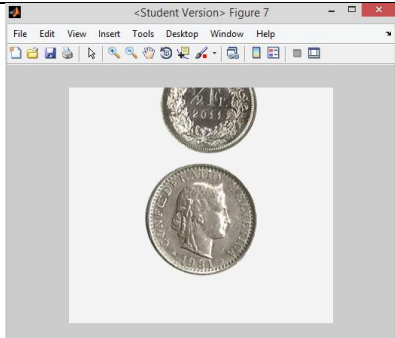
Mit den strikten Vorgaben der Bildaufnahmen und der flexiblen Hough-Transformation wurden die Münzwerte zuverlässig ermittelt.

## Template-Matching

Zusätzlich zu der Bildanalyse mit der Hough-transformation führen wir ein Template-Matching durch. Das heisst, dass wir eine Mustersuche durchführen. Unsere Muster sind logischerweise die Schweizer Münzen.

Wir haben uns aus persönlichen und herausfordernden Gründen entschlossen, diese zusätzliche Methode zu implementieren. Im günstigsten Fall erhalten wir gerade noch einen Vergleich des Resultates zur Hough-Transformation. Somit haben wir mit der Hough-Transformation eine Analyse im Hough-Raum und mit dem Template-Matching eine Analyse im Bildraum (resp. Matrizen).

Der Ablauf gestaltet sich wie folgt:

Templates auslesen	Vorlagedateien werden ausgelesen, das Bild und die Werte (aus Dateinamen) gespeichert.	 <i>Abbildung 8: 500_front.jpg</i>
Münzflächen im Bild suchen (1)	Im Binärbild aus der Hough-transformation werden die Kreise gesucht mit der Funktion „regionprops“	
Münzflächen im Bild suchen (2)	Die gefundenen Münzen werden dann mit einer adjustierten Boundingbox[12] (je 150 Pixel mehr pro Seite) ausgeschnitten	
Loop über die gefundenen Münzen	Über das Array der gefundenen Münzen wird geloopt, und der Durchmesser dieser Münze eruiert	
Loop über die Referenzmünzen (1)	Jede Referenzmünze wird an den Durchmesser der gefundenen Münze angepasst	
Loop über 360 Grad	Die angepasste Referenzmünze wird gedreht, in 1 Grad Schritten	
Vergleich	Beide Bilder (ausgeschnittenes Bild und angepasstes, gedrehtes Referenzbild) werden in eine Matrix konvertiert und verglichen	

## Schwierigkeiten des Template-Matching

Dieses Verfahren hat sich als sehr Rechenintensiv herausgestellt. Dazu ein kleines Beispiel:  
10 Münzen auf dem Tisch, 4 verschiedene (5.-- , 2.-- , 1.-- , 50 Rp.) Typen.

= 10 Loops auf Stufe 1

14 Templates (jede Schweizer Münze, vor und Rückseite)

= 14 Loops auf Stufe 2

Skalieren und Rotieren des Templates

= 360 Loops auf Stufe 3

Vergleich

Umwandeln der Bilder in Double-Matrizen und Vergleich.

Somit kommen wir auf 50'400 Vergleiche. (von Beispielsweise 300x300 Matrizen)

## Resultate

Bewährt hat sich das Verfahren mit der Referenzmünze in Zusammenspiel mit der Hough-transformation. Durch die Anpassung der Radien mit  $(-2 / -1 / 0 / +1 / +2)$  Pixeln konnten wir alle Münzen fehlerfrei identifizieren und das Geld zählen. Wir mussten aber einen zusätzlichen Vergleich einbauen, denn eine Münze mit Radius 323 Pixel kann bei den Schritten mit 322 (-1), 323 (0) und 324 (+1) Pixeln gefunden werden. Hier mussten wir erkennen, dass es sich um dieselbe Münze handelt.

Nur teilweise hat sich das Template-Matching bewährt. Da sehr viele Rechenoperationen ausgeführt werden müssen, dauerte dies bei unserem Referenzbild mit nur zwei Templates (10 Rappen und 5 CHF) 1382.888460 Sekunden, 23 Minuten.

Trotzdem konnten mit einem angepassten Setting die Münzen gefunden werden:

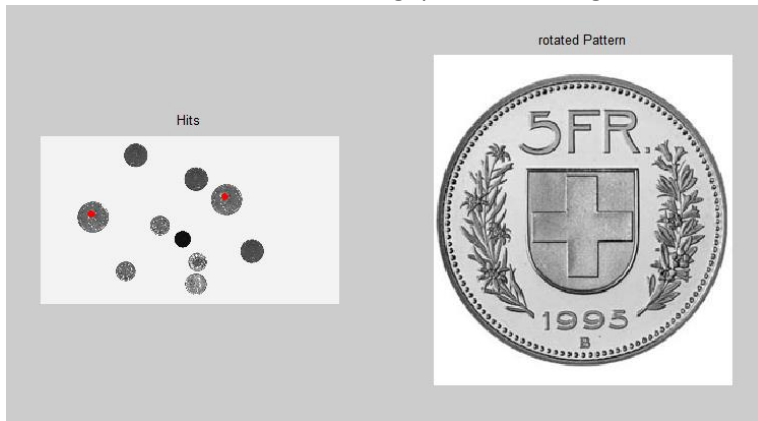


Abbildung 11: Gefundene Münzen Template-Matching

Das angepasste Setting enthielt nur ein Template, und es wurde nur jeweils um 2 Grad rotiert. Somit verringerten sich die Operationen von 50'400 Vergleichen zu  $10 \times 1 \times 2$ , 20 Vergleichen.

Als Resultat zu unserem Testbild (Titelblatt) können wir nun folgendes präsentieren:

*Final Money: CHF*

*13.9000*

*Elapsed time is 18.946052 seconds.*

## Diskussion

Da uns einige Schwierigkeiten aufgefallen sind, möchten wir auf folgende drei Probleme hinweisen, die u.U. in einer weiteren Analyse genauer betrachtet werden können:

1. Die Hough-Analyse ohne Referenzmünze durchführen. Dies wird anspruchsvoll wenn nur eine Münze oder nur Münzen eines Types vorhanden sind.  
(Da keine Relation der Durchmesser mehr möglich ist).
2. Bildauflösung begrenzen.  
Für die Hough-Analyse und die Template-Analyse wäre es von Vorteil, wenn nicht allzu grosse Bilder verarbeitet werden müssten.  
Die Hough-Analyse könnte mit genaueren Durchmessern arbeiten (ohne  $-2 / -1 / 0 / +1 / +2$  Schritte) und die Template-Analyse würde von kleineren Matrixvergleichen profitieren
3. Münzen mit Kantenerkennungsalgorithmem vorbereiten.



Die Kantenerkennung für die Templates könnte eine verbesserte Performance bringen. (Da damit auch ein Graustufenbild noch weiter reduziert werden kann, ohne das signifikante Information verloren gehen)

Achtung, eventuell würden die Jahreszahlen auf der Münze hier aber dann einen negativen Einfluss haben, da sie so auch stärker ins Gewicht fallen.

## Anhang

Unter folgendem Link können Sie unsere Scripts betrachten:

<https://www.dropbox.com/sh/gt93u0gooiropos/AAABuRGNsKrDB9nT9pYUkqyPa?dl=0>

## Glossar \*

1	Kantenerkennung	Die Kantenerkennung ist ein digitales Bildanalyse-Verfahren. Eine Kante wird als Grenzfläche zwischen zwei unterschiedlichen Bildbereichen definiert. Kanten können durch eine Veränderung des Lichts (Farbe, Schatten) oder durch unterschiedliche Texturen (Oberflächen) bedingt sein.
2	Binarisierung	Ein Binärbild, oder eine Binarisierung ist eine digitale Rastergraphik, wo die Pixel nur aus Schwarz oder Weiss bestehen.
3	Matlab	Matlab steht für MATrix LABoratory, und ist eine kommerzielle Software von „The MatWorks“.
4	Graustufenerkennung	Dies bezeichnet ein Bild nur bestehend aus Abstufungen von reinem Weiss bis zu reinem Schwarz. So kann ein RGB Bild mit bspw 16.7 Mio Farben in 255 Graustufen unterteilt werden.
5	Image fill	Mittels Matlab Funktion imfill(). Es werden Hintergrundpixel gefüllt, welche nicht vom Rand aus erreichbar sind.
6	Closing	Bezeichnet ein morphologisches Verfahren auf Binärbildern, um Löcher zu füllen (Dilatation gefolgt von Erosion)
7	Hough Transformation	Die Hough-Transformation ist ein robustes globales Verfahren zur Erkennung von Geraden, Kreisen oder beliebigen anderen parametrisierbaren geometrischen Figuren in einem binären Gradientenbild, also einem Schwarz-Weiß-Bild, nach einer Kantenerkennung. Das Verfahren wurde 1962 von Paul V. C. Hough unter dem Namen „Method and Means for Recognizing Complex Patterns“ patentiert.
8	Templatematching mit Kreuzkorrelation	Das Templatematching ist ein Verfahren, wo verschiedene nicht durch Merkmale parametrisierbare Muster in einem Bild gesucht werden. Die Kreuzkorrelation zeigt an, wie Ähnlich die beiden Signale (Bilder) sind.
9	Treshhold	Schwellenwert, auch Schwellenwertverfahren. Definiert, welche Bildpunkte das gesuchte Objekt darstellen, und welche der Umgebung angehören. (Bei Beispielsweise Graustufen)
10	Perspektivische Verzerrung	Dies ist der geometrische Abbildungsfehler, welcher entsteht wenn gleiche Objekte unterschiedlich weit vom Zentrum (Fotokamera, Auge) entfernt sind.
11	Hough Raum	Der Hough Raum ist anders als ein kartesisches Koordinatensystem nicht in x- & y-Richtung aufgeteilt, sondern nach Winkel & Abstand zum Mittelpunkt. So lassen sich verschiedene Geometrische Figuren als Geraden abbilden.
12	Boundingbox	Umschreibt vollständig ein Objekt.

\*Quelle: Wikipedia.org