

Bearbeitung des 1. Übungsblatts

Wir werden folgendes Szenario betrachten: Gegeben sei ein Bild von einer ebenen, rechteckigen Messfläche, die wir mit einer Kamera beobachten. Auf diese Fläche werden Euro-Münzen gelegt. Die Position dieser Münzen und ihre Wertigkeit soll automatisch erkannt werden.

Um dieses Problem zu lösen, gehen wir wie folgt vor:

- Messfläche erkennen, projektiv entzerren und ausschneiden.
- Trennen von Vordergrund (Münzen) und Hintergrund (Messfläche).
- Finden von Zusammenhangskomponenten (einzelne Münzen).
- Klassifizieren der Zusammenhangskomponente, d.h. ihnen mittels Merkmalen (z.B. Größe) einen Münztyp zuweisen.

Aufgabe 1 Projektives Entzerren und Ausschneiden

Im ersten Schritt sollen Sie eine Funktion schreiben, die aus einer beliebigen Ansicht der Messfläche eine Draufsicht berechnet. Gehen Sie hierbei wie folgt vor:

- Laden Sie ein Bild aus der Bilddatenbank (\rightarrow `imread`).
- Lassen Sie sich das Bild anzeigen (\rightarrow `imshow`).
- Bestimmen Sie manuell die Position der vier Kreuze, die die Messebene markieren (\rightarrow `p=ginput(4)`).
- Vervollständigen Sie die Funktion `p = mark(I)`.
- Führen Sie eine projektive Transformation auf das Bild aus, so dass die 4 Markierungen auf die Positionen

$$P_0 = \left\{ \begin{pmatrix} 1 \\ 2000 \end{pmatrix}, \begin{pmatrix} 1450 \\ 2000 \end{pmatrix}, \begin{pmatrix} 1450 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 1 \end{pmatrix} \right\}$$

abgebildet werden. Schneiden Sie das neue Bild dabei so zurechnet, dass es nur die Messebene abbildet (\rightarrow `cp2tform`, `imtransform` [verwenden Sie die Parameter `XData` und `YData`]).

- Schreiben bzw. vervollständigen Sie die Funktion

```
function Iout = projectiveCrop(I,p,p0),
```

die die Bildregion, die durch ein Polygon (`p`) beschrieben wird, auf ein Bild mit den Ecken (`p0`) abbildet und zurückgibt.

Aufgabe 2 Hintergrundsubtraktion

Im zweiten Schritt wollen wir nun die Vordergrundregionen finden. Gehen Sie hierbei wie folgt vor:

- Laden Sie ein Bild aus der Bilddatenbank, bestimmen Sie die Position der Messebenmarkierungen und wenden Sie die Funktion `projectiveCrop` an.
- Wechseln Sie in den HSV-Farbraum indem Sie das RGB-Bild der Messebene in den HSV-Farbraum transformieren (\rightarrow `rgb2hsv`).
- Betrachten Sie sich die drei Farbkanäle des Bildes im HSV-Raumes. Was ist die Bedeutung der Kanäle und welcher eignet sich am besten um den Hinter- bzw. Vordergrund zu erkennen?
- Wandeln Sie das HSV-bild in ein Binärbild um, dabei sollen alle Pixel, deren Wert bezüglich des Kanals unterhalb oder überhalb eines geeigneten Schwellwertes liegen, auf 1 gesetzt werden, und alle anderen auf 0 (\rightarrow `find`, `A=B<T`).
- Finden Sie einen geeigneten Schwellwert und verwenden Sie morphologische Operationen, um kleinere Löcher zu schließen (\rightarrow `imerode`, `imdilate`).

Bearbeitung des 1. Übungsblatts

- Vervollständigen Sie die Funktion `B = backgroundSubtraction(I, t)`, die ein Bild und einen Schwellwert erwarten und eine binäre Segmentierung zurückliefert (1=Vordergrund, 0=Hintergrund).

Aufgabe 3 Zusammenhangskomponenten

Für eine gegebene Vorder-/Hintergrundsegmentierung wollen wir nun die Zusammenhangskomponenten (Münzen) detektieren.

Hinweis: 2 Pixel gehören zu derselben Zusammenhangskomponente, wenn sie direkt aneinander grenzen. Jedes Pixel gehört genau zu einer Zusammenhangskomponente. Die Größe einer Zusammenhangskomponente ist die Anzahl der Pixel die zu dieser gehören. Der Schwerpunkt einer Zusammenhangskomponente ist der Mittelwert der Positionen ihrer Pixel.

- Verwenden Sie die Matlab-Funktion `bwconncomp`, um die Zusammenhangskomponenten eines Binärbildes zu bestimmen.
- Schreiben Sie die Funktion

```
function [coinList, L] = findCoins(B),
```

der eine Segmentierung des Messbereichs übergeben wird und die eine Liste der Zusammenhangskomponenten und eine Segmentierung L zurückgibt. Dabei soll L die gleiche Größe wie B haben und das Pixel p den Wert k wenn p zur k-ten Zusammenhangskomponente gehört (0 = Hintergrund). Zum Beschreiben der Objektliste verwenden Sie die Klasse `ObjectList` (\rightarrow `objectList.m`).

- Um die Markierungen und sonstige Fehldetektionen nicht als Objekt zu berücksichtigen, entfernen Sie zu kleine Objekte aus der Liste bzw. fügen Sie diese nicht hinzu.

Aufgabe 4 Zwischenabgabe 1

Wenn Sie alle Aufgaben korrekt bearbeitet haben, können Sie das Skript `zwischenabgabe.m` aufrufen, um für alle Trainingsbilder eine Registrierung und Hintergrundsubtraktion durchzuführen. Sie können den Vorgang jeder Zeit abbrechen. Beim Neustart überspringt das Skript schon bearbeitete Bilder. Wollen Sie ein Bild erneut bearbeiten, müssen Sie die gespeicherte Segmentierung von Hand löschen. Am Ende berechnet das Skript die Zusammenhangskomponenten, zeigt diese an und speichert die Objekt-Liste.

Zur ersten Zwischenabgabe sollte das Skript fehlerfrei durchlaufen. Was beobachten Sie bezüglich der Segmentierungen?

Aufgabe 5 Merkmale

Erweitern Sie die Funktion `saveLists` um die Berechnung der Münzengröße als erstes Merkmal.

- Verwenden Sie das Labelbild L um die Anzahl der Pixel zu zählen, die zur k-ten Zusammenhangskomponente gehören.
- Setzen Sie mittels `coinList.setObjectFeature(k, kSize)`; die Größe der k-ten Komponente auf kSize.

Aufgabe 6 Testdatensatz

Um Ihren Klassifikator vernünftig zu evaluieren, benötigen Sie einen unabhängigen Testdatensatz. Implementieren Sie, aufbauend auf der Funktion `saveLists`, eine Funktion um die Objektlisten des Testdatensatzes zu erzeugen. Im Gegensatz zu den Trainingsdaten können Sie die Wertigkeit der Münzen nicht mehr aus den Dateinamen generieren! Sie können jedoch ausnutzen, dass manche Testdaten die gleiche Szene aus verschiedenen Sichten zeigen.

Bearbeitung des 1. Übungsblatts

Überlegen Sie sich wie Sie dies bewerkstelligen können und beachten Sie, dass Sie den Testdatensatz später für andere Merkmale erneut benutzen möchten.

Aufgabe 7 Klassifikation

Anhand der Größe eines Objekts soll nun die Wertigkeit der Münze erkannt werden.

- Verwenden Sie die Trainingsdaten, um die Verteilung der Objektgröße für die verschiedenen Münzen zu bestimmen. Gesucht ist also die bedingte Dichte $p(\text{Merkmal}|\text{Münztyp})$, hierbei ist das Merkmal die Objektgröße. Approximieren Sie diese Wahrscheinlichkeitsverteilung/-dichte mit einer Gauss-Verteilung.

$$p(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$$

beziehungsweise

$$p(x) = \frac{1}{|\Sigma|^{0.5}(2\pi)^{d/2}} \exp\left(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu)\right).$$

- Schreiben Sie eine Funktion

`function list = detectedCoins(I,p,p0),`

die das Bild, p und $p0$ übernimmt und jedem Objekt die Wertigkeit (Münztyp) zuweist, für die $P(\text{Münztyp}|\text{Merkmal})$ maximal wird. Gehen Sie davon aus, dass alle Münzen gleich häufig auftauchen und verwenden Sie das Bayestheorem

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}.$$

Desweiteren empfiehlt es sich, zu großen und zu kleinen Objekten unabhängig vom Klassifikatorergebnis die Wertigkeit 0 zuzuweisen, bzw. sie nicht in die Objektliste aufzunehmen.

- Evaluieren Sie ihren Münzenerkennung in dem Sie die Konfusionsmatrix für Test- und Trainingsdatensatz berechnen ($\leftarrow \text{confusionmat}$).

Aufgabe 8 Zwischenabgabe 2

Bereiten Sie eine Präsentation (15 Minuten) mit Ihren bisherigen Ergebnissen vor. Betrachten Sie Stärken und Schwächen der verwendeten Verfahren und schlagen Sie Alternativen vor.

Aufgabe 9 Endabgabe

Verbessern Sie den bisherigen Ansatz in dem Sie weitere/andere Merkmale und/oder Klassifikatoren verwenden. Dokumentieren Sie die Verbesserung und erstellen Sie einen Abschlussvortrag (15 Minuten).