

Übung Computergrafik 2

Abgabetermin:

Die Lösung zu diesem Übungsblatt ist bis zum **18.07.2012** abzugeben.

Form der Abgabe:

- Die Übungen können in Gruppen von 2–3 Studenten bearbeitet werden.
- Die Abgaben bestehen aus den Python-Quelltexten samt aller Bilder, die für die Bearbeitung der Übung verwendet wurden. Idealerweise wird das gesamte PyDev-Projektverzeichnis für die jeweilige Übung über Eclipse als .zip Archiv exportiert. Wichtig: Zur Vermeidung von Namenskonflikten in Eclipse, euren Nachnamen bitte als Präfix vor die Namen der PyDev-Projekte setzen, z.B.: „müller-übung-x“.
- Teilaufgaben, deren Python-Dateien wegen Syntaxfehlern nicht ausführbar sind, werden nicht weiter korrigiert.
- Textaufgaben sollten in Form einer .doc, .odt oder (bevorzugt) als PDF-Datei abgegeben werden, und sollten (falls erforderlich) die benötigten Ausgabebilder der Aufgaben enthalten.
- Alle Übungsabgaben erfolgen über UniWorX¹.

Inhalt: In dieser Übung wird ein Demonstrator entwickelt, mit dem verschiedene Klassifizierungsverfahren von Bildregionen anhand von Haralick'schen Texturmerkmalen evaluiert werden können.

¹<https://uniworx.ifi.lmu.de>

Aufgabe 1 Haralick'sche Texturemerkmale, (single-)Nearest-Neighbor-Klassifizierung (★★)

Haralick'sche Texturmaße

Die Haralick'schen Texturmaße beruhen auf der Berechnung einer Co-Occurrence-Matrix der Pixel einer Bildregion R . Die Co-Occurrence-Matrix gibt an, wie wahrscheinlich es ist, dass zwei Pixel p_1 und p_2 von R , die um den Winkel α und die Distanz d versetzt sind, p_1 den Grauwert g_1 und p_2 den Grauwert g_2 haben (Abbildung 1). Die Co-Occurrence-Matrix für eine Region R stellt also ein normalisiertes Histogramm dar und lässt sich wie folgt berechnen:

$$P_{d_x, d_y}(g_1, g_2) = \frac{1}{|R|} \sum_{(x,y) \in R} \delta_D(f(x,y) - g_1) \delta_D(f(x+d_x, y+d_y) - g_2)$$

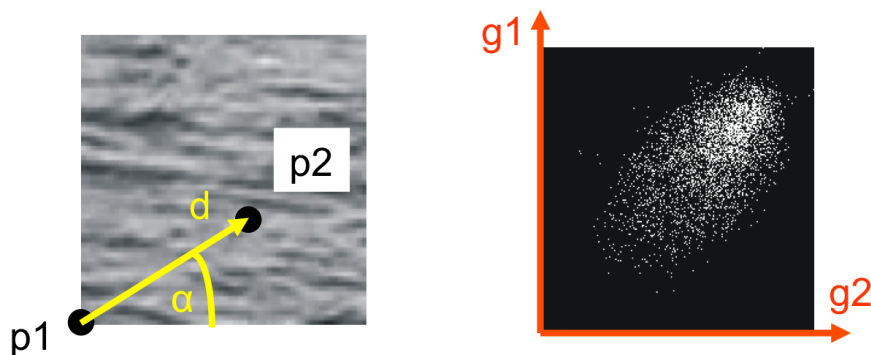


Abbildung 1: Skizzierung des Korrelationsmaßes der Co-Occurrence-Matrix (links). Co-Occurrence Matrix der Textur (rechts).

Algorithmisch lässt sich die obige Formel durch die Berechnung eines Histogramms über die Häufigkeit der gefundenen Grauwertpaare in der Region R (Typischerweise 40×40 Pixel) umsetzen. Achten Sie darauf, das Histogramm zu normalisieren, d.h. dass die Summe aller Einträge 1 ergibt. Es reicht ferner aus, ein Binning der Grauwerte g_1, g_2 auf $K = 64$ Bins vorzunehmen. Anhand der Co-Occurrence-Matrix lassen sich die Haralick'schen Texturmaße berechnen, die als Merkmalsvektor für die Klassifizierung verwendet werden können. Besonders geeignete Texturmaße für die Klassifizierung sind die folgenden:

$$\text{Entropie: } - \sum_{g_1=0}^{K-1} \sum_{g_2=0}^{K-1} P_{d_x, d_y}(g_1, g_2) \log P_{d_x, d_y}(g_1, g_2)$$

$$\text{Kontrast: } \sum_{g_1=0}^{K-1} \sum_{g_2=0}^{K-1} (g_1 - g_2)^2 P_{d_x, d_y}(g_1, g_2)$$

$$\text{Korrelation: } \sum_{g_1=0}^{K-1} \sum_{g_2=0}^{K-1} \frac{(g_1 - \mu_1)(g_2 - \mu_2) P_{d_x, d_y}(g_1, g_2)}{\sigma_1 \sigma_2}$$

Beachten Sie bitte bei der Entropie das negative Vorzeichen im Summenterm. Formeln zur Berechnung von $\mu_1, \mu_2, \sigma_1, \sigma_2$ für die Korrelation finden Sie in den Vorlesungsfolien.

Grafische Oberfläche zur Evaluierung von Klassifikatoren

Für diese Aufgabe haben wir eine graphische Oberfläche (GUI), Abbildung 2, entwickelt, die als Grundlage für die Implementierung dieser Aufgabe dienen wird. Die GUI ermöglicht es, bestimmte Pixel eines Bildes einer Klasse zuzuordnen, einen Klassifikationsalgorithmus zu trainieren, und die Klassifikation zu verifizieren.

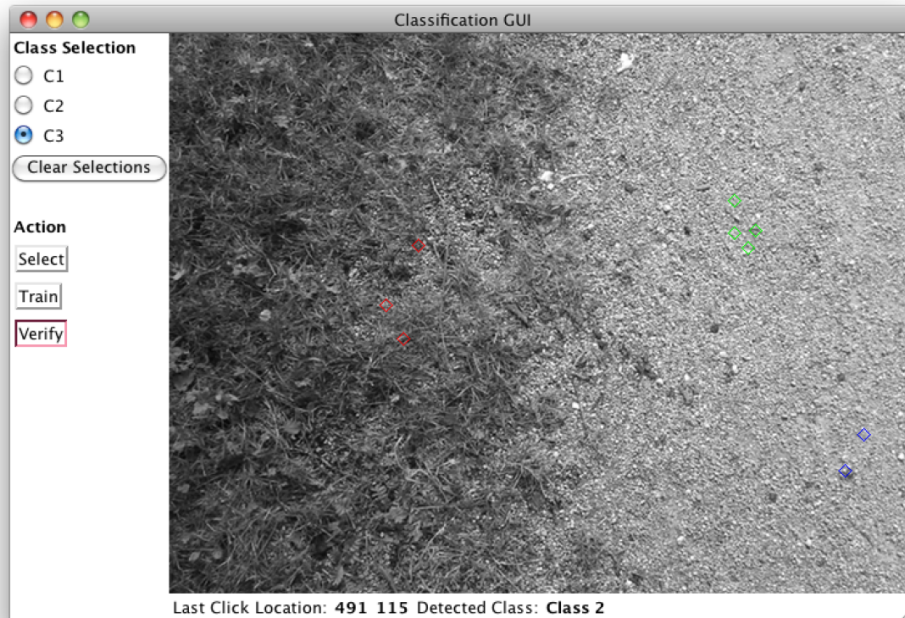


Abbildung 2: Grafische Oberfläche zur Evaluation von Klassifikationsverfahren von Bildbereichen.

Im Select-Modus lassen sich über die **Class Selection** Bildpunkte durch Mausklick auf das Bild einer Klasse (C1, C2 oder C3) zuordnen. Die Aktion **Train** erlaubt es, nach der Selektion einen Klassifikator zu trainieren. Beim Klick auf den Button **Train** wird folgender Callback eines zu implementierenden Klassifikators aufgerufen, der die Zuordnung der Klassen zu den Bildpunkten (selections) und das Quellbild (image) dem Klassifikator übergibt:

```
1 def training_callback(selections, image):
```

Ist die Aktion Verify ausgewählt, lässt sich durch Mausklicks auf dem Bild für einzelne Bildpunkte die Zugehörigkeit zu einer Klasse verifizieren. Bei jedem Klick auf das Bild wird der folgende Callback ausgelöst:

```
def verification_callback(x, y, image):  
    ...  
    return classID
```

Der Callback erhält die Koordinaten des Klicks und muss die ID der erkannten Klasse (C1, C2 oder C3) an die GUI zurückgeben.

Die Aufgabenstellung

- Entwickeln Sie einen auf Haralick'schen Texturmaßen beruhenden Classifier, der die beiden Callbacks der GUI, training_callback und verification_callback implementiert. Gehen Sie dazu wie folgt vor:

- Implementieren Sie anhand der ersten Gleichung die Berechnung der Co-Occurance-Matrix eines gegebenen Bildbereiches (für die Größe R des Bildbereichs z.B. 40×40 Pixel wählen).
 - Implementieren Sie anhand der Gleichungen für Entropie, Kontrast und Korrelation die Berechnung der Haralick'schen Texturmaße zu einer gegebenen Co-Occurance-Matrix.
 - Implementieren Sie nun den `training_callback`. Berechnen Sie dazu für alle vom Benutzer eingegebenen Punkte jeweils eine Co-Occurance-Matrix und die Texturmaße für $d = (1,0)$ und $d = (0,1)$.
 - Stellen Sie die Texturmaße zu einem 6-dimensionalen Merkmalsvektor (Entropie, Kontrast und Korrelation, je zwei Richtungen) für jede Co-Occurance-Matrix zusammen.
 - Berechnen Sie für jede Klasse den Mittelpunkt im Merkmalsraum und speichern Sie diesen.
 - Implementieren Sie nun den `verification_callback`. Dieser berechnet anhand der Eingabekoordinaten (x,y) eine Co-Occurance-Matrix und die dazugehörigen Haralick'schen Texturmaße mit $d = (1,0)$ und $d = (0,1)$ um (x,y) und lokalisiert somit (x,y) im Merkmalsraum. Die Klasse, zu der der Bildpunkt (x,y) ist die des Klassenmittelpunktes mit dem geringsten Abstand zu (x,y) .
- b) Implementieren Sie einen auf Haralick'schen Texturmaßen beruhenden Classifier, der jedoch im Gegensatz zum Aufgabenteil (a), ein single-Nearest-Neighbor-Verfahren zur Klassifizierung verwendet.
- c) Optional: Implementieren Sie einen Bayes'schen Classifier für Bildregionen der auf Haralick'schen Texturmaßen beruht.