

Intelligente Sehsysteme – Übungsblatt 8

Prof. Dr. W. Koch, Dr. F. Govaers, H. Hoelzemann, F. Oßwald, R. Bartolosch
*E-Mail: {wolfgang.koch, felix.govaers, henry.hoelzemann, florian.osswald,
roman.bartolosch}@fkie.fraunhofer.de}*

Abgabe bis Sonntag, 14.12.2024, 12:00 Uhr in Gruppen von 3 Personen

1 Harris Corner Detector (2P)

Leiten Sie für das zentrale Pixel \mathbf{p} des abgebildeten 5×5 -Bildes den Wert der Antwortfunktion R ab. Gehen Sie bitte schrittweise vor:

1. Berechnen Sie zunächst $\langle I_x^2 \rangle, \langle I_y^2 \rangle, \langle I_x I_y \rangle$ (s. Folie 10, Vorl. 8). Der Einfachheit halber soll die Gewichtsfunktion $w(u, v)$ anstelle einer Gauß-Funktion alle Pixel innerhalb des in \mathbf{p} zentrierten 3×3 -Umgebungsfensters mit dem Faktor 1 gewichten und alle Pixel außerhalb des Fensters mit dem Faktor 0 gewichten.
2. Notieren Sie daraus folgend die Harris-Matrix \mathbf{A} .
3. Berechnen Sie $R = \det(\mathbf{A}) - \kappa \cdot \text{trace}^2(\mathbf{A})$ mit $\kappa = 0, 1$.

1	1	1	1	1
1	1	1	1	0
0	0	1	0	0
0	0	0	0	0
0	0	0	0	0

2 Harris Corner Detector in Python (3P)

Implementieren Sie den *Harris Corner Detector* in Python. Sie können von Grauwertbildern als Eingabe ausgehen. Die Parameter κ, σ sowie der Schwellwert t (s. Folie 19, Vorl. 8) können vom Benutzer festgelegt werden. Das Ergebnis soll eine Kopie des Eingabebildes sein, aber an den detektierten *Interest Points* soll das Pixel sowie dessen 8-Nachbarschaft rot (RGB = (255,0,0)) gefärbt sein (ähnlich wie auf Folie 25, Vorl. 8).

Evaluieren Sie Ihre Implementierung an den beiden Bildern *Letter_A.png* und *Alphabet.png*, die in eCampus zusammen mit diesem Übungsblatt hinterlegt sind.

Laden Sie mit Ihrem Code die beiden entsprechenden Ergebnisbilder hoch. Geben Sie die dabei jeweils benutzten Werte der Parameter κ, σ und t an.

3 SIFT Keypoint Descriptor (2P)

Bemerkung: Die Aufgabe ist eigentlich einfach, aber ihre Spezifikation erfordert leider etwas mehr Text.

Der *SIFT Keypoint Descriptor* wird abgeleitet, indem die Gradientenbeträge und -orientierungen jedes Pixels in einem 16×16 -Sample Array um den *Keypoint* erfasst und mit einer Gauß-Funktion gewichtet werden. Das 16×16 -Sample Array wird dann in $4 \times 4 = 16$ Teilregionen unterteilt und die Orientierungen in 8 Orientierungen klassifiziert: $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$. Dies führt zu $16 \cdot 8 = 128$ Deskriptorkomponenten (s. Folie 68, Vorl. 8).

Der Einfachheit halber soll das Sampling der Gradienten in dieser Aufgabe in einem 4×4 -Sample Array (s. Abb.) um den *Keypoint* und ohne eine gewichtende Gauß-Funktion erfolgen. Das 4×4 -Sample Array wird dann in $2 \times 1 = 2$ Teilregionen (eine obere und eine untere Teilregion) unterteilt und die Orientierungen in 8 Orientierungen klassifiziert: $0^\circ, 45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ$. Jedes Feld im 4×4 -Sample Array zeigt die Betragssumme (oben) sowie die gemittelte Orientierung (unten). Der Eintrag „./.“ bedeutet, dass keine signifikanten Gradienten gefunden wurden.

Leiten Sie den *SIFT Keypoint Descriptor* für dieses einfache Szenario ab. Schreiben Sie den Deskriptor auf, indem Sie mit der oberen Teilregion und mit 0° beginnen.

10 45°	1 45°	./.	./.
./.	20 45°	1 90°	1 135°
1 270°	./.	20 90°	1 90°
./.	1 225°	1 45°	10 90°

4 Kombination von Harris Corner Detector und SIFT Descriptor in Python (5P)

Wenden Sie Ihre Implementierung des *Harris Corner Detector* auf die beiden Bilder *Letter_A.png* und *Alphabet.png* an. Wenn Sie Aufgabe 2 nicht bearbeitet haben, benutzen Sie die in den beiden Bildern *Letter_A_Harris.png* und *Alphabet_Harris.png* angezeigten *Key Points*. Beide Bilder sind in eCampus zusammen mit diesem Übungsblatt hinterlegt. Beachten Sie, dass sich diese Ergebnisse von denen Ihrer eigenen Implementierung aufgrund anderer Parameterwahlen unterscheiden können.

1. Implementieren Sie die Schritte 3 und 4 zur Erzeugung des *SIFT Keypoint Descriptors* (s. Folie 56, Vorl. 8) um jetzt SIFT-Deskriptoren für solche *Key Points* zu finden, die mit dem *Harris Corner Detector* abgeleitet wurden. Für beide Schritte wird ein einheitliches Sigma genutzt.
2. Implementieren Sie eine Funktionalität, die in zwei Eingabebildern korrespondierende *Key Points* und deren 8-Nachbarschaft mit derselben Farbe markiert. Testen Sie mit den beiden Bildern *Letter_A.png* und *Alphabet.png* an den *Key Points*, die in *Letter_A_Harris.png* und *Alphabet_Harris.png* markiert sind - also die Wiedererkennung der *Key Points* des Buchstabens A im Alphabet. Zur Korrespondenzbestimmung nutzen Sie bitte die minimale Euklidische Distanz der SIFT-Deskriptoren der *Key Points*.