

Intelligente Sehsysteme – Übungsblatt 6

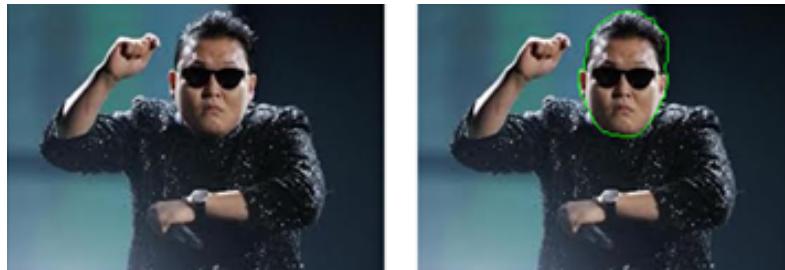
Prof. Dr. W. Koch, Dr. F. Govaers, H. Hoelzemann, F. Oßwald

E-Mail: {wolfgang.koch, felix.govaers, henry.hoelzemann, florian.osswald}@fkie.fraunhofer.de

Abgabe bis Sonntag, 30.11.2024, 12:00 Uhr in Gruppen von 2 bis 3 Personen.

1 Intelligent Scissors (4P)

Bitte implementieren Sie *Intelligent Scissors* entsprechend Vorlesung 6, Folie 48 basierend auf der interaktiven Suche nach optimalen Kantenzügen entsprechend Vorlesung 6, Folien 38 ff. Testen Sie Ihre Implementierung von *Intelligent Scissors* auf dem Bild **Testbild_Gangman_300x200.png**, das in eCampus zusammen mit diesem Übungsblatt hinterlegt ist, indem Sie *Intelligent Scissors* zur Segmentierung des Gesichts des Sängers nutzen.



Bildquelle: <https://sites.google.com/site/aminzhenghomepage/teaching> (30 Nov 2017)

Sie können das Bild zuvor in ein Grauwertbild umwandeln und *Intelligent Scissors* für Grauwertbilder umsetzen.

1. Implementieren Sie zunächst die auf dem Dijkstra-Algorithmus basierende Suche nach einem optimalen Kantenzug für ein durch den Cursor definiertes Punktpaar (Startpunkt und Endpunkt) und testen Sie diese Suche an einer Kante des Oberarms des abgebildeten Sängers (3P)
2. Erweitern Sie die Suche um die schrittweise Erweiterung durch Folgen von weiteren Endpunkten zu *Intelligent Scissors* und testen Sie diese zur Segmentierung des Gesichts des Sängers. (1P)

Hinweise:

1. Die von Tönnies verwendete Kostenfunktion $c(k) = |g(k) - g_{mod}|$ mit $g_{mod} = 0,5 \cdot (|\nabla I(e)| + |\nabla I(s)|)$ ist vergleichsweise einfach. Daher ist auch eine deutlich einfachere Version des Bildes in eCampus zusammen mit diesem Übungsblatt zum Testen hinterlegt als Bild **Testbild_GangmanSketch_300x200.png**. Aber auch dort ist darauf zu achten, dass die interaktiv gesetzten Start- und Endpunkte hinreichend nahe an Kanten platziert werden. Ansonsten kann es bei der Rückwärtsverfolgung bei Start-/Endpunkten auf homogenen Flächen zu einer Dauerschleife kommen, da dann der kürzeste Pfad nicht eindeutig ist. Dieser Spezialfall kann aber auch abgefangen werden, indem beim Backtracking bereits besuchte Knoten nicht noch einmal besucht werden und bei mehreren Nachbarn mit entsprechenden Kosten der Knoten mit der geringsten euklidischen Distanz gewählt wird.
2. Dokumentieren Sie alle Tests durch Beifügen der Ergebnisse zu Ihrer Lösungsabgabe.

2 Flutungsalgorithmus (2P)

Gegeben sind die folgende 5x5-Grauwertmatrix ($I_{max} = 15$), deren Pixel bereits die Gradientenbeträge eines Originalbildes darstellen, sowie die binarisierte Variante (Schwellwert $t_b = 12$) dieser Matrix.

3	2	13	7	10
6	5	14	15	4
9	15	7	6	14
13	4	5	14	13
6	2	13	15	3

0	0	15	0	0
0	0	15	15	0
0	15	0	0	15
15	0	0	15	15
0	0	15	15	0

Wenden Sie den Flutungsalgorithmus auf die beiden Bilder bzw. Matrizen an, wobei Sie diese von links nach rechts und von oben nach unten durchlaufen. Gehen Sie dabei zusätzlich von einer 8-Nachbarschaft aus.

3 Canny Edge Detector (4P)

Bitte implementieren Sie den Canny-Operator entsprechend Vorlesung 6, Folien 29 ff. und wenden ihn auf das Bild *Testbild_Lena_512x512.png* an. Geben Sie für die Parameter σ , t_1 und t_2 die verwendeten Werte an und laden Sie analog zu den Vorlesungsfolien neben Ihrem Quellcode je ein Bild hoch, das die jeweilige Verarbeitungsstufe zeigt:

1. Das per Gaußfilter geglättete Bild,
2. das Bild der Gradientenbeträge,
3. das Bild der Gradientenbeträge nach Anwendung der Non-Maximum-Unterdrückung,
4. die binarisierten Kantenzüge.

Dokumentieren Sie die Tests aller vier Schritte am Bild *Testbild_Lena_512x512.png* durch Beifügen der Ergebnisse zu Ihrer Lösungsabgabe.