

Praktikum Bildverarbeitung, Übung 4

Morphologie

Ziel dieser Praktikumsübung ist es, mit morphologischen Filtern Strukturen auch bei schwierigen Beleuchtungsverhältnissen zu detektieren. Als Strukturen werden beispielhaft Buchstaben in inhomogen ausgeleuchteten Bildern gewählt.

Die Ergebnisse der schrittweise aufeinanderfolgenden Filter sollen betrachtet werden, um ein Gefühl für deren Eigenschaften zu gewinnen.

In Aufgabe 1 wird die optimale Schwelle auf ein inhomogen beleuchtetes Quellbild angewendet. Das Ergebnis ist unbefriedigend.

In Aufgabe 2 wird das Quellbild mit einem vorgegebenen, quadratischen SE erodiert und dilatiert.

In Aufgabe 3 wird ein annähernd rundes SE berechnet und damit das Quellbild erodiert und dilatiert.

In Aufgabe 4 wird mit dem runden SE ein Opening und Closing auf dem Bild durchgeführt.

In Aufgabe 5 werden der schwarze und der weiße Zylinderhut berechnet und die optimale Schwelle auf die Zylinderhut-Bilder angewendet. Nun sollten die Buchstaben gut erkennbar sein.

Bereiten Sie alle Aufgaben vor dem Praktikumstermin gründlich vor und erstellen Sie soweit wie möglich schon vor dem Praktikumstermin Programmcode.

Aufgabe 1: Optimale Schwelle auf Grauwertbild anwenden

Im Programmrahmen wird eine Funktion `optimal_threshold` aufgerufen, die aus dem Luminanzbild `src` ein Binärbild `erstes_Binaerbild` berechnet.

⇒ Erstellen Sie den Code der Funktion `optimal_threshold` mithilfe der Vorlesungsunterlagen.

⇒ Berechnen Sie aus den Quellbildern "*Inhomogen_1.bmp*" und "*Inhomogen_2.bmp*" die optimalen Binärbilder.

Ausarbeitung: Beschreiben Sie das Resultat in wenigen Sätzen.

Aufgabe 2: Erosion und Dilation mit quadratischem SE

Das Luminanzbild `src` soll so stark erodiert und dilatiert werden, dass vorgegebene helle Strukturen (Erosion) bzw. dunkle Strukturen (Dilation) im Quellbild entfernt werden. Wie ist die Größe des strukturierenden Elements zu wählen?

Das strukturierende Element wird durch xy-Positionen relativ zum Bezugspunkt spezifiziert, die als `pair<int, int>`-Objekte in einem `vector`-Container abgelegt sind.

⇒ Wählen Sie im Programmrahmen eine sinnvolle Größe für das quadratische strukturierende Element passend zur Größe der Buchstaben in den Quellbildern "*Inhomogen_1.bmp*" und "*Inhomogen_2.bmp*".

⇒ Erstellen Sie den Code der Template-Funktion `erode<Pixel>`.

⇒ Erstellen Sie den Code der Template-Funktion `dilate<Pixel>`.

⇒ Berechnen Sie jeweils das erodierte und dilatierte Bild zu den Quellbildern "*Inhomogen_1.bmp*" und "*Inhomogen_2.bmp*".

Ausarbeitung: Diskutieren Sie kurz, ob Ihre Wahl der Größe des strukturierenden Elements sinnvoll ist.

Aufgabe 3: Erosion und Dilation mit rundem SE

Alternativ zum quadratischen SE soll nun ein rundes SE mit Durchmesser d zur Erosion und Dilation verwendet werden. Dazu werden in dem `vector`-Container, der das strukturierende Element beschreibt, nur Positionen abgelegt, die einen maximalen Abstand von $d/2$ zum Bezugspunkt (kleiner als der Radius) besitzen. Eine Position (x, y) gehört also dann zum strukturierenden Element, wenn gilt:

$$\frac{d}{2} \geq \sqrt{x^2 + y^2} \quad \text{oder} \quad \left(\frac{d}{2}\right)^2 \geq x^2 + y^2$$

⇒ Erstellen Sie den Code der Funktion `create_round_SE`.

⇒ Berechnen Sie jeweils das erodierte und dilatierte Bild zu den Quellbildern `"Inhomogen_1.bmp"` und `"Inhomogen_2.bmp"`

Ausarbeitung: Erläutern Sie den Unterschied zu den Ergebnissen mit dem quadratischem SE.

Aufgabe 4: Opening und Closing

Die verbliebenden Strukturen in erodierten bzw. dilatierten Bildern können wieder auf ihre ursprüngliche Größe gebracht werden. Dazu dient eine nachfolgende Dilation bzw. Erosion mit dem punktgespiegelten strukturierenden Element. Man erhält damit ein geöffnetes (Opening) bzw. geschlossenes (Closing) Bild.

Im Programmrahmen werden die Template-Funktionen `opening` und `closing` auf dem Quellbild `src` mit rundem SE aufgerufen. Diese sind zu kodieren. Innerhalb dieser Funktionen wird das punktgespiegelte SE benötigt. Dazu ist eine Hilfsfunktion `mirror_SE` vorgesehen, die ebenfalls zu kodieren ist.

⇒ Erstellen Sie den Code der Funktion `mirror_SE`.

⇒ Erstellen Sie den Code der Template-Funktion `opening`.

⇒ Erstellen Sie den Code der Template-Funktion `closing`.

⇒ Berechnen Sie jeweils das geöffnete und geschlossene Bild zu den Quellbildern `"Inhomogen_1.bmp"` und `"Inhomogen_2.bmp"`.

Ausarbeitung: Vergleichen Sie die erzeugten Bilder mit dem Quellbild.

Aufgabe 5: Zylinderhut-Operationen und Binarisierung

Das weiße und schwarze Zylinderhut-Bild wird durch Differenzbildung des geöffneten bzw. geschlossenen Bilds mit dem Originalbild berechnet. Zur einfachen Berechnung ist die Subtraktion für `Img`-Bilder überladen.

Im Programmrahmen werden das weiße und schwarze Zylinderhut-Bild berechnet und anschließend darauf die optimale Schwelle angewendet.

⇒ Erstellen Sie den Code des überladenen Subtraktionsoperators für `Img`-Bilder.

⇒ Berechnen Sie die Zylinderhut-Bilder und die daraus berechneten Binärbilder zu den Quellbildern "*Inhomogen_1.bmp*" und "*Inhomogen_2.bmp*".

Ausarbeitung: Inspizieren und diskutieren Sie die Ergebnisse.