Aufgabenblatt 2

Einführung in die Bildverarbeitung
Christian Wilms und Simone Frintrop
SoSe 2020

Ausgabe: 8. Mai 2020 - **Abgabe bis: 15. Mai 2020, 10:00**

Aufgabe 1 — **Anwendung Fernerkundung** - 15 Punkte - Programmieraufgabe
Erlaubte (Sub-)Pakete: `numpy`, `skimage.io`, `matplotlib`
Das Satellitenbild in Abbildung 1 scheint einige grüne Punkte in der Wüste zu zeigen. Sind dies Pflanzen? Findet es heraus, indem ihr den Vegetationsindex NDVI (normalized difference vegetation index) für jedes Pixel im Bild berechnet. NDVI ist ein Indikator für lebende, grüne Vegetation in Bildern. Dazu wird für den selben Bildbereich eine Aufnahme des roten Lichts (Red, Wellenlänge $0.63\mu\text{m} - 0.69\mu\text{m}$) und des nahen Infrarot (NIR, Wellenlänge $0.78\mu\text{m} - 0.90\mu\text{m}$) benötigt. Mithilfe dieser Aufnahmen kann der NDVI für jeden Pixel als

$$\text{NDVI} = \frac{(\text{NIR} - \text{Red})}{(\text{NIR} + \text{Red})}$$

berechnet werden. Führt dies durch, indem ihr euch aus der zip-Datei `landsatBild.zip` im Moodle die entsprechenden Bänder (jedes Band ist ein Graustufenbild) des Landsat 5-Satellitenbildes heraussucht und verarbeitet. Das Ergebnis soll mit `matplotlib` wie ein übliches Graustufenbild angezeigt werden. Worum handelt es sich nun bei den grünen Punkten? Was sagen euch die Abstufungen der Grautöne im Ergebnis? Antwortet in ein einem kurzen Text (max. 4 Sätze).
Hinweis: Das Ergebnis des NDVI je Pixel soll im Bereich von -1 bis 1 liegen (achtet auf den Elementtyp bei der Division). Trotzdem kann das Bild angezeigt werden, wenn es einen Elementtyp aus der Familie `np.float` besitzt.

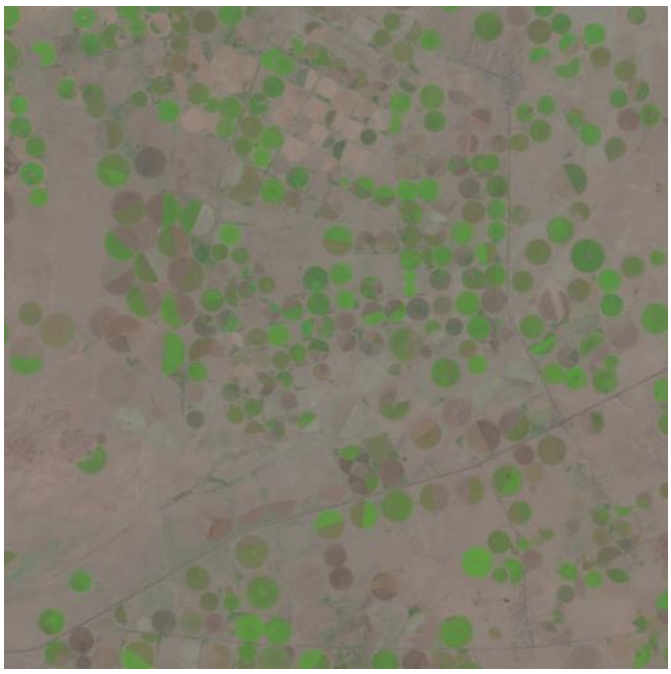

Abbildung 1: Echtfarben-Darstellung eines Landsat 5-Satellitenbildes.

Aufgabe 2 — Abtastung und Quantisierung - 25 Punkte - Programmieraufgabe
Erlaubte (Sub-)Pakete: `numpy`, `math`, `scipy.stats`, `matplotlib`
Erstellt ein Python-Skript, das ein Bild mit 100 identischen Bildzeilen erzeugt. Jede Bildzeile soll dabei die identische Abtastung und Quantisierung der Gauß-Funktion

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

mit $\mu = 0$ und $\sigma = 1$ beinhalten, wobei der Definitionsbereich der Funktion von -5 bis $+5$ an 51 regelmäßig verteilten Punkten abgetastet werden soll. Das Bild soll folglich 100 gleiche Bildzeilen mit jeweils 51 Bildspalten haben. Alle Pixelwerte sollen zudem so skaliert werden, dass der höchste Wert des Bildes bei 255 und der niedrigste Wert bei 0 liegt. Wie üblich soll der Wertebereich des Bildes $\{0, 1, 2, \ldots, 255\}$ sein. Zeigt das entstandene Bild mit `matplotlib` an. Beantwortet schriftlich, welche Auswirkungen die Veränderung von μ und σ auf das Bild hat (max. 2 Sätze).
Hinweis: Pixelwerte sollen stets kaufmännisch gerundet werden.

Aufgabe 3 — Abtasttheorem - 15 Punkte - Theorieaufgabe
Ihr sitzt an Bord eines Propellerflugzeugs am Fenster mit bester Sicht auf einen der Propeller und könnt dessen Rotation klar und korrekt wahrnehmen. Nun wollt ihr die Rotation des Propellers mit eurer Kamera in einem Video festhalten. Ist dies aus Sicht der Signalverarbeitung eine sinnvolle Idee? Beantwortet dazu folgende Fragen schriftlich mit kurzer Begründung. Nehmt dazu an, dass sich der Propeller pro Minute 1000 mal dreht und eure Kamera 30 vollständige Bilder pro Sekunde aufnehmen kann.

1. Ist das Abtasttheorem im Sinne der zeitlichen Abtastung der Propellerdrehung durch die Kamera erfüllt? Rechnet es nach!
2. Was wird schätzungsweise auf dem Video erkennbar sein und was nicht?
3. Ändert sich etwas, wenn eure Kamera statt 30 Bilder nun 60 Bilder pro Sekunde aufnehmen kann?

Aufgabe 4 — Gaußsches Rauschen - 25 Punkte - Theorieaufgabe
Nehmt an, dass ihr M verrauschte Varianten $\mathrm{g}_m(x, y)$ eines Bildes $\mathrm{f}(x, y)$ habt. Unter der Annahme von additivem Rauschen bedeutet dies:

$$\mathrm{g}_m(x, y) = \mathrm{f}(x, y) + \eta_m(x, y).$$

Alle η_m repräsentieren dabei Gaußsches Rauschen mit einem Erwartungswert $\mu = 0$ sowie einer Varianz $\sigma^2 = 1$ und sind für alle Paare (x, y) unkorreliert. Das Rauschen kann reduziert werden, indem je Pixel ein Mittelwert über die verrauschten Varianten berechnet wird:

$$\bar{\mathrm{g}}(x, y) = \frac{1}{M} \sum_{m=1}^{M} \mathrm{g}_m(x, y).$$

Beweist nun, dass der Erwartungswert E des Ergebnisses $\bar{\mathrm{g}}(x, y)$ wiederum dem Bild $\mathrm{f}(x, y)$ entspricht:

$$\mathrm{E}(\bar{\mathrm{g}}(x, y)) = \mathrm{f}(x, y).$$

Aufgabe 5 — Rauschen - 20 Punkte - Programmieraufgabe
Erlaubte (Sub-)Pakete: `numpy`, `skimage.io`, `matplotlib`
Erstellt ein Python-Skript mit zwei Funktionen, die jeweils das Gaußsche Rauschen bzw. das salt-and-pepper noise auf ein Bild applizieren. Bei der Funktion zum Gaußschen Rauschen soll die Standardabweichung übergeben werden können und bei der Funktion zum salt-and-pepper noise die Wahrscheinlichkeit für eine Veränderung (salt und pepper sind dann gleich wahrscheinlich). Testet die beiden Funktionen am Bild `mandrill.png` vom Aufgabenblatt der letzten Woche und betrachtet die Ergebnisse. Achtet darauf, dass das Ergebnis wieder im Wertebereich $\{0, 1, 2, \ldots, 255\}$ sein muss. Beschreibt kurz, welche Auswirkungen die Veränderung der jeweiligen Parameter der beiden Funktionen (Standardabweichung bzw. Wahrscheinlichkeit für eine Veränderung) auf das Ergebnis hat.
Hinweis 1: Nutzt das Paket `numpy.random`, um zufällige Werte zu ziehen.
Hinweis 2: Mit der Funktion `numpy.clip` lässt sich ein bestimmter Wertebereich für ein Array erzwingen.