

CHARAKTERISIERUNG UND RADIOMETRISCHE KALIBRIERUNG VON BILDSSENSOREN

Bildgebende Optische Systeme, Teil 3 - Software, Übung 1

1. Aufgabe 1: Photonentransfer

Vorbereitung: Machen Sie sich mit dem Messplatz und den verschiedenen Komponenten vertraut: Stromquelle, Ulbrichtkugel, Kameramontierung. Sie brauchen zusätzlich ein Photometer. Notieren Sie sich die Wellenlänge λ der Leuchtdiode in der Ulbrichtkugel.

Sie werden eine Reihe von Bildern aufnehmen. Diese Bilder sollten Sie im *png*-Format speichern. Falls das nicht geht, müssen Sie die Bilder nachträglich in dieses Format umwandeln. Zum Einlesen der Bilder benutzen Sie am Besten die Funktion `matplotlib.image.imread()` in Python (Beispiel unter http://matplotlib.org/users/image_tutorial.html) oder `imread()` in Matlab.

Aufgabe:

- Entfernen Sie die Optik von der Kamera. Stellen Sie sicher, dass der Sensor komplett von der Ulbrichtkugel ausgeleuchtet ist. Dazu sollte die Entfernung des Sensors maximal das 8–9fache der Öffnungsbreite der Ulbrichtkugel betragen.
- Messen Sie die Bestrahlungsstärke E der Lichtquelle in W/m^2 mit dem Photometer in gleichem Abstand wie die Kamera. Achten Sie dabei auf den korrekten Umrechnungsfaktor für die Wellenlänge der Leuchtdiode. Mit der Pixelfläche A , der Wellenlänge λ der Leuchtdiode und der Belichtungszeit können Sie so die mittlere Anzahl der einfallenden Photonen berechnen (s. Vorl.2/5).
- Nehmen Sie mindestens 9 verschiedene Belichtungszeiten. Für jede Belichtungszeit werden 4 Bilder aufgenommen, je zwei von der Ulbrichtkugel und je zwei bei abgedeckter Kamera. Bestimmen Sie den Sättigungspunkt der Kamera, d.h. die maximale Belichtungszeit, ab der die Pixelvarianz wieder abzunehmen beginnt (s. Vorl. 2/15).
- Verwenden Sie die Photonentransfermethode nach EMVA 1288, um eine Sensitivitäts- und eine Photonentransferkurve zu erstellen. Schauen Sie sich dazu die Beispiele in den Vorlesungsunterlagen bzw. in der EMVA 1288 an.
- Bestimmen Sie die Responsivität R , Kameraverstärkung K , Quanteneffizienz η und das Dunkelsignal σ_d^2 , wie in der Vorlesung gezeigt.

Tips: In Python gibt es für die lineare Regression die Funktion `scipy.stats.linregress()` aus dem Paket `scipy` oder `sklearn.linear_model.LinearRegression()` aus dem Paket `scikit-learn`. Die Vorgehensweise unter Matlab finden Sie unter https://de.mathworks.com/help/matlab/data_analysis/linear-regression.html beschrieben.

2. Aufgabe 2: Kennzahlen zur Bewertung der Kamera

- Bestimmen Sie aus den für Aufgabe 1 aufgenommenen Daten die SNR-Kurve der Kamera und tragen Sie diese wie in der Vorlesung in einem doppelt logarithmischen Diagramm auf (s.Vorl. 2/20).
- Zeichnen Sie den Sättigungspunkt und die kleinste nutzbare Bestrahlungsstärke (s. Vorlesung) ein, außerdem die SNR-Kurve des idealen Sensors und die theoretische SNR-Kurve, wie sie sich aus den in Aufgabe 1 bestimmten Parametern ergibt (Vorl. 2/18).
- Bestimmen Sie den Dynamikbereich Ihrer Kamera. Vergleichen Sie Ihre Ergebnisse mit den anderen Gruppen. Welche Kamera ist besser?

Tips: Ein Beispiel für ein doppelt logarithmisches Diagramm in Python findet man unter http://matplotlib.org/examples/pylab_examples/log_demo.html, in Matlab geschieht dies mit der Funktion `loglog()`.

3. Aufgabe 3: Räumliche Inhomogenitäten

- Nehmen Sie einen Satz von mindestens 16 Bildern bei 50% des Sättigungswertes und mindestens 16 Bilder bei abgedeckter Kamera auf. Verwenden Sie dabei für beide Bildersätze die gleiche Belichtungszeit.
- Berechnen Sie daraus ein Weißbild und ein Dunkelbild.
- Berechnen Sie DSNU und PRNU nach EMVA 1288 Ihrer Kamera.
- Berechnen Sie die horizontalen und vertikalen Spektrogramme und stellen Sie diese so wie in der Vorlesung dar. Hat Ihre Kamera periodische Inhomogenitäten?

Tips: Im Python-Paket `numpy` berechnet man die eindimensionale Fouriertransformierte mit der Funktion `numpy.fft.fft()`, in Matlab mit `fft()`.

4. Aufgabe 4: Defektpixel

- Erstellen Sie eine hochpassgefilterte Version Ihres Weißbildes aus Aufgabe 3.
- Berechnen Sie das logarithmierte DSNU-Histogramm aus dem Dunkelbild aus Aufgabe 3 und das logarithmierte PRNU-Histogramm aus dem hochpassgefilterten Weißbild.

- Bestimmen Sie die Positionen der *dead pixels* in dem hochpassgefilterten Weißbild anhand eines aus dem PRNU-Histogramm gewählten Schwellwertes.
- Bestimmen Sie die Positionen der *hot pixels* und *stuck pixels* im Dunkelbild anhand eines aus dem DSNU-Histogramm gewählten Schwellwertes.

Tips: Ein Beispiel für die Anwendung eines Boxfilters in Python findet sich unter http://docs.opencv.org/3.1.0/d4/d13/tutorial_py_filtering.html. Matlab hat hierfür die Funktion *imboxfilt()* in der *image processing toolbox*. Für die logarithmischen Histogramme in Python gibt es ebenfalls ein Beispiel unter http://matplotlib.org/examples/pylab_examples/log_demo.html, in Matlab gibt es hierfür die Funktion *semilogy()*.

5. Aufgabe 5: Radiometrische Kalibrierung

- Erzeugen Sie ein zusätzliches Dunkelbild für eine möglichst kleine Belichtungszeit.
- Nehmen Sie ein Weißbild bei aufgesetzter Kameraoptik auf.
- Nehmen Sie ein Bild einer Testchart bei beliebiger anderer Belichtungszeit auf.
- Interpolieren Sie aus den beiden Dunkelbildern ein neues Dunkelbild für die Belichtungszeit des Testbildes.
- Kalibrieren Sie das Testbild radiometrisch, wie in der Vorlesung gezeigt. Ergibt sich eine sichtbare Verbesserung? Bestimmen Sie die mittlere Anzahl der einfallenden Photonen pro Pixel für einen von Ihnen ausgewählten homogenen Bildbereich mithilfe der Responsivität aus Aufgabe 1.
- Zusatzaufgabe: ersetzen sie alle Defektpixel durch Interpolation über seine Nachbarpixel.