Ludwig-Maximilians-Universität München
LFE Medieninformatik, Mobile Interaction Group
Prof. Dr. Michael Rohs
Dipl.-Inform. Sven Kratz

Sommersemester 2012
Übungsblatt 5
06.06.2012

Übung Computergrafik 2

Abgabetermin:

Die Lösung zu diesem Übungsblatt ist bis zum **13.06.2012** abzugeben.

Form der Abgabe:

- Die Übungen können in Gruppen von 2–3 Studenten bearbeitet werden.
- Die Abgaben bestehen aus den Python-Quelltexten samt aller Bilder, die für die Bearbeitung der Übung verwendet wurden. Idealerweise wird das gesamte PyDev-Projektverzeichnis für die jeweilige Übung über Eclipse als .zip Archiv exportiert. Wichtig: Zur Vermeidung von Namenskonflikten in Eclipse, euren Nachnamen bitte als Präfix vor die Namen der PyDev-Projekte setzen, z.B.: „müller-übung-x“.
- Teilaufgaben, deren Python-Dateien wegen Syntaxfehlern nicht ausführbar sind, werden nicht weiter korrigiert.
- Textaufgaben sollten in Form einer .doc, .odt oder (bevorzugt) als PDF-Datei abgegeben werden, und sollten (falls erforderlich) die benötigten Ausgabebilder der Aufgaben enthalten.
- Alle Übungsabgaben erfolgen über UniWorx[1].

Inhalt:

In diesem Übungsblatt behandeln wir die Fourier Transformation und ihre Anwendung in der Bildverarbeitung. Die grundlegenden Konzepte der Fourier Transformation (FT) haben Sie in der Vorlesung kennen gelernt, hier sollen Sie ihr wissen praktisch anwenden und vertiefen.

[1] https://uniworx.ifi.lmu.de

Aufgabe 1 Diskrete Fourier Transformation (⋆⋆⋆)

Jedes beliebige Signal, z.B. Töne im 1-dimensionalen bzw. Bilder im 2-dimensionalen, besteht aus einer Kombination von Signalen unterschiedlicher Frequenz.

Anhand eines Prismas können Sie sich sehr gut veranschaulichen, wie Signale aus unterschiedlichen Signalen zusammen gesetzt sind. Wenn weißes Licht auf das Prisma trifft, wird das Signal in seine konstituierenden Signale (farbiges Licht) aufgespalten.

Analog lässt sich jede periodische Funktion durch eine gewichtete Summe von Sinusoiden (Sinus und Cosinus Wellen) ausdrücken. Diese Erkenntnis ist die Grundlage der Fouriertheorie. Die einzelnen Funktionen der Summe heißen Basisfunktionen. Die Fouriertheorie gibt uns ein Werkzeug um die Anteile jeder einzelnen Basisfunktion in der Repräsentation einer Funktion $f(x)$ zu analysieren. Für Bilder gilt, dass die Originalfunktion zweidimensional ist.

Die FT in zwei Dimensionen lässt sich folgendermaßen angeben:

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(m,n) * e^{-i*2\pi(um+vn)}\, dmdn \tag{1}$$

Analog ist die inverse FT als

$$f(m,n) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) * e^{i*2\pi(um+vn)}\, dudv \tag{2}$$

definiert.

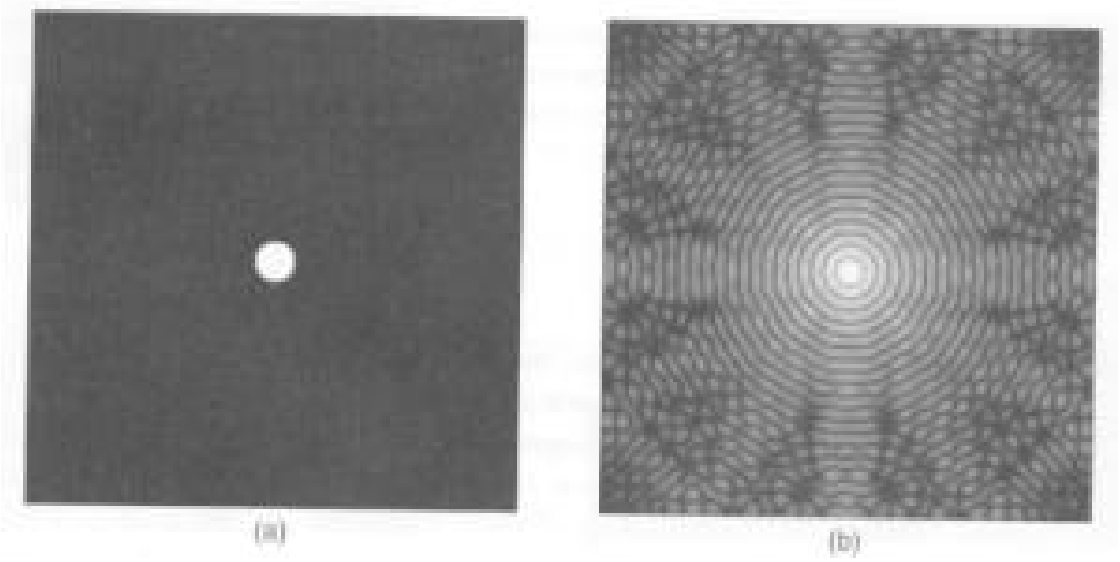

Abbildung 1: Fourier-Transformierte eines Kreises. (a) der Ortsraum. (b) der Frequenzraum

Dabei bezeichnet $F(u,v)$ die Funktion im *Frequenzraum*, oder die Fourier-Transformierte, und $f(m,n)$ die Originalfunktion im *Ortsraum*. Beachten Sie dabei, dass $F(u,v)$ eine komplexe Zahl ist. Im Frequenzraum bezeichnet u die Frequenz entlang der originalen m-Achse und v die Frequenz entlang der originalen n-Achse. In der Darstellung in Abb. 1 ist der Koordinatenursprung des Frequenzraums in die Mitte des Bildes verschoben worden; im Ortsraum liegt er links oben.

Wenn wir mit digitalen Bildern arbeiten, haben wir niemals eine kontinuierlich definierte Funktion sondern diskrete Samples - die Pixel des Bildes. Um mit solchen Daten trotzdem im Frequenzraum arbeiten zu können, benötigen Sie einen Spezialfall der FT - die diskrete FT. Anstatt der Integration für kontinuierliche Funktionen, verwenden wir hier Summen über die diskreten Samples.

Die Formel um eine Bild der Dimension MxN in den Frequenzraum zu transformieren ist als

$$F(u,v) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) * e^{-i*2\pi*\left(\frac{um}{M}+\frac{vn}{N}\right)} \tag{3}$$

gegeben. Die Formel um zurück in den Ortsraum zu gelangen ist

$$f(m,n) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} F(u,v) * e^{i*2\pi*\left(\frac{um}{M}+\frac{vn}{N}\right)} \tag{4}$$

Der Unterschied zwischen den beiden Formeln, ist das Vorzeichen im Exponenten. Die FT ist in beide Richtungen verlustfrei, also eine eineindeutige Abbildung. Aus den Formeln wird unmittelbar deutlich, dass sich der selbe Code verwenden lässt, um die FT und die inverse FT zu implementieren.

Um zu verstehen, was die Fouriertransformierte eines Bildes ausdrückt, muss man sich nochmal verdeutlichen, das $F(u,v)$ eine komplexe Zahl ist.

$$F(u,v) = R(u,v) + i*I(u,v) = |F(u,v)| * e^{i*\phi(u,v)} \tag{5}$$

Der reelle $R(u,v)$ und imaginäre $I(u,v)$ Anteil an sich haben noch keine besondere Bedeutung, allerdings ist $|F(u,v)|$ die Amplitude und $\phi(u,v)$ ihre Phase. Zerlegt man nun ein Bild im Frequenzraum in ein Array von Amplituden und ein Array von Phasen, dann erhalten wir das Amplituden-Spektrum, respektive das Phasen-Spektrum eines Bildes. Diese Spektren lassen sich wiederum als Bilder (Abb. 2) zeichnen.

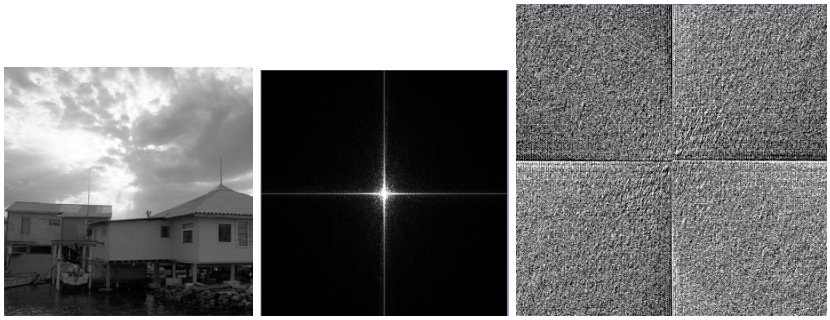

Abbildung 2: Ein Bild im Ortsraum, sein Amplituden-Spektrum, und das Phasen-Spektrum.

a) Machen Sie sich mit den Konzepten der Fourier Transformation (FT) und insbesondere der diskreten Variante vertraut. Benutzen Sie dazu die Vorlesungsfolien.

b) Schreiben Sie die Basisfunktionen der diskreten 1D-Fouriertransformation für N = 7 auf und zeigen Sie, dass diese Funktionen eine orthogonale Basis der Dimension N bilden.

c) Welche Länge haben die Basisvektoren?

d) Berechnen Sie die Basisfunktionen der inversen diskreten 1D-Fouriertransformation. Multiplizieren Sie nun die Matrix der Vorwärtstransformation mit Matrix der Rückwärtstransformation. Wie lautet das Ergebnis?

e) Implementieren Sie in Python sowohl die (Fourier-)Transformation einer Bilddatei in den Frequenzraum sowie ihre Rücktransformation. Verwenden Sie hierbei die Eigenschaft der Separabilität. D.h. dass jede 2-dimensionale FT in zwei 1-dimensionale FTs aufgespalten werden kann. Zuerst wird eine 1-dimensionale FT entlang den Zeilen eines Bildes ausgeführt um ein Zwischenergebnis zur berechnen. Anschließend wird eine zweite FT entlang der Spalten dieses Zwischenergebnisses berechnet um zum finalen Ergebnis zu gelangen.

Aufgabe 2 Fast Fourier Transform ($\star\star\star$)

Um ein einen Wert von $F(u,v)$ nach der Standardformel zu berechnen, muss einmal über alle Pixel summiert werden. Bei einem NxN Bild ergibt das eine Komplexität von $O(N^2)$ und folglich für alle Werte eine Gesamtkomplexität von $O(N^4)$. Das führt schon bei sehr kleinen Bildern zu nicht akzeptablen Laufzeiten.

Glücklicherweise hat die FT neben der Separabilität noch die Eigenschaft der Symmetrie:

Symmetrie Die 1-dimensionale FT der Länge N kann als Summe zweier FTs der Länge $\frac{N}{2}$ ausgedrückt werden. Ist nun N eine 2er Potenz, dann kann man diesen Schritt rekursiv anwenden, bis Transformationen der Länge 2 betrachtet werden.

Diese beiden Eigenschaften erlauben es die Laufzeiten der FT signifikant zu verbessern.

Die *Fast Fourier Transformation* (FFT) ist ein *Divide-and-Conquer* Algorithmus der obige Eigenschaften ausnützt und damit die Komplexität auf $O(N^2 \log N)$ senkt. Die FFT für Bilder erfordert, dass beide Bild-Dimensionen 2er Potenzen sind. Ist das nicht der Fall, muss das Bild entweder beschnitten oder mit Nullen aufgefüllt werden.

a) Machen Sie sich mit dem FFT Algorithmus und seiner Implementierung vertraut. Gute Ausgangspunkte sind der Wikipedia Eintrag zum originalen FFT
http://en.wikipedia.org/wiki/Cooley-Tukey_FFT_algorithm
außerdem:
http://dspguru.com/dsp/faqs/fft
http://www.fftw.org/links.html
http://www.dspguide.com/CH12.PDF

b) Implementieren Sie den FFT Algorithmus in Python.

Aufgabe 3 Konvolution im Frequenzraum vs. Ortsraum ($\star$)

Wann ist Konvolution im Frequenzraum schneller als im Ortsraum? Nehmen Sie an, das Eingabebild habe die Größe $2^n \times 2^n$.

a) Der Konvolutionskern habe die Größe $m \times m$ und sei separierbar.

b) Der Konvolutionskern habe die Größe $m \times m$ und sei nicht separierbar.