

Bildverarbeitung

Fragen

Manuel Pauli

Sebastian Schweikl

Thomas Lang

23. Juni 2016

1 Allgemein

Erkläre Operationen: Skalierung, Translation

Erkläre das Prinzip einer Lochkamera Eine Lochkamera ist der Vorläufer einer modernen Kamera. Dabei fällt das natürliche Licht durch ein kleines Loch (die Blende) ein. Das Bild trifft dann auf der Rückwand der Kamera auf. Bedingt durch die Geometrie steht das Bild auf dem Kopf, dessen Höhe/Breite wird dabei durch den Abstand des Objektes zur Blende, des Abstandes der Blende zur Rückwand und von der Größe des Objektes und der Blende bestimmt.

2 Fourier

2.1 Fouriertransformation

Nenne die Gleichung Für eine Funktion $f \in L_1(\mathbb{R})$ ist die Fouriertransformation durch

$$\hat{f}(\xi) = \int_{\mathbb{R}} f(t) e^{-i\xi t} dt$$

für alle $\xi \in \mathbb{R}$ definiert.

Das ist die Gleichung für L_1 -Funktionen. Wie sieht das im L_2 aus? Die Fouriertransformation im Signalraum L_2 kann ganz analog zum obigen Fall verwendet werden. Die resultierende Fouriertransformierte liegt dann in $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$.

Formal beruht dies darauf, dass der Raum $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ dicht in $L_2(\mathbb{R})$ liegt. Deshalb lässt sich also immer eine Funktionenfolge aus $L_1(\mathbb{R}) \cap L_2(\mathbb{R})$ finden, die im Grenzwert gegen unsere gesuchte Fouriertransformierte konvergiert¹.

Wie sieht das Ergebnis der FT bei einer reellwertigen Funktion aus?

¹Geht immer, zwecks Banachraum und so.

Was sagt der Satz von Parseval/Plancherel? Der Satz von Parseval/Plancherel beschreibt den Wechsel zwischen einem Skalarprodukt im Zeit- und einem Skalarprodukt im Frequenzbereich.

Formal gilt für Funktionen $f, g \in L_1(\mathbb{R})$:

$$\int_{\mathbb{R}} f(t)g(t)dt = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(\vartheta)\overline{\hat{g}(\vartheta)}d\vartheta$$

Im Speziellen für $f = g$ folgt, dass

$$\|f\|_1^2 = \frac{1}{2\pi} \|\hat{f}\|_2^2$$

gilt, also dass hier (bis auf Normierung) eine Isometrie vorliegt.

Sie sprachen von Isometrie, warum ist dann der konstante Faktor vor dem rechten Term?

Dieser Faktor kommt daher, weil wir bei der Definition der Fouriertransformation keinen Faktor dabei hatten. Hätte man dort einen Faktor $1/\sqrt{2\pi}$ hinzugefügt, so hätte man hier eine perfekte Isometrie, was man ja mit Energieerhaltung² identifizieren kann.

Wie sieht die inverse Fouriertransformation aus? Unter der Voraussetzung, dass sowohl für f als auch für \hat{f} gilt, dass diese Funktionen aus L_1 sind, ist die inverse Fouriertransformation definiert als

$$f(\xi) = (\hat{f})^\wedge(\xi) = \frac{1}{2\pi} \int_{\mathbb{R}} \hat{f}(t)e^{i\xi t} dt$$

Zusammenhang zwischen Faltung und Fouriertransformation Ein wichtiger Zusammenhang ist, dass sich die Faltung zweier Funktionen als Produkt ihrer Fouriertransformierten darstellen lässt (vorausgesetzt deren Existenz).

Es gilt also:

$$\begin{aligned} (f * c)^\wedge(\xi) &\stackrel{\text{def}}{=} \int_{\mathbb{R}} (f * c)(t)e^{-i\xi t} dt \\ &\stackrel{\text{def}}{=} \int_{\mathbb{R}} \int_{\mathbb{R}} f(t-x)c(x)dx e^{-i\xi t} dt \\ &\stackrel{\text{Tonelli}}{=} \int_{\mathbb{R}} \int_{\mathbb{R}} f(t-x)c(x)e^{-i\xi t} dx dt \\ &\stackrel{t=t+x}{=} \int_{\mathbb{R}} \int_{\mathbb{R}} f(t)c(x)e^{-i\xi(t+x)} dx dt \\ &= \int_{\mathbb{R}} \int_{\mathbb{R}} f(t)e^{-i\xi t} c(x)e^{-i\xi x} dx dt \\ &\stackrel{\text{Tonelli}}{=} \int_{\mathbb{R}} f(t)e^{-i\xi t} dt \int_{\mathbb{R}} c(x)e^{-i\xi x} dx \\ &= \hat{f}(\xi)\hat{c}(\xi) \end{aligned}$$

²Die 2-Norm $\|\cdot\|_2$ beschreibt ja i.W. Energie.

Erkläre den Weg zur FFT Der Weg von der DFT zur FFT ist der, dass man einen Baumalgorithmus implementiert. Konkret arbeitet ein Teilbaum alle Folgenglieder mit geradem Index ab, während der andere Teilbaum den Rest bearbeitet, wobei sich natürlich die Indexmenge in den Teilbäumen vom Vaterknoten verändert.

Formal habe der Vaterknoten die Indexmenge $\mathbb{Z}_n = \{0, \dots, n-1\}$ zur Folge c und sei $n = 2 * m, m \in \mathbb{N}$. Dann ergibt sich für die Teilbäume, dass der eine alle Folgenglieder $c(2k)$ und der andere alle Folgenglieder $c(2k+1), k \in \mathbb{Z}_m = \{0, \dots, m-1\}$ verarbeitet.

Wie schnell ist die FFT? Da man dabei einen Baumalgorithmus implementiert, hat man also *nur* $\log n$ Ebenen abzuarbeiten, wobei man immer noch alle Folgenglieder $c(k), k \in \mathbb{Z}_n$ verarbeiten muss, es entsteht also ein Aufwand in $O(n \log n)$. Nachzuweisen ist dies mit dem Mastertheorem.

Theorie schön und gut, aber wozu braucht man die Fouriertransformation nun in der Praxis?

Mit der ganzen Theorie (insbesondere mit dem Zusammenhang zwischen der Faltung und der Fouriertransformation) können Bildverarbeitungsalgorithmen in verschiedenster Hinsicht optimiert werden. Beispielsweise um die reine Berechnung zu beschleunigen (Multiplikation vs Faltung) oder um die numerische Stabilität zu erhöhen.

Der Standardweg ist dabei, dass man das Signal fouriertransformiert, auf dem Ergebnis die Berechnungen durchführt und anschließend wieder zurücktransformiert.

Wie funktioniert das JPEG-Kompressionsverfahren mit Hilfe der DFT? Die Idee ist hierbei, die Tatsache der Abhängigkeitsfreiheit zu nutzen um den Algorithmus massiv parallel zu machen. Dabei ist noch nicht mal die FFT gemeint, sondern das Vorgehen das Bild in Makroblöcke der fixen Größe 8×8 aufzuspalten. Diese fixe Größe kommt daher, dass man also pro Makroblock 16×4 - Vektoren vorliegen hat, welche z.B. auf Graphikkarten oder modernen CPUs sehr effizient verarbeitet werden können. Falls die Dimensionen des Bildes keine Vielfachen von 8 sind, so können die fehlenden Ränder z.B. auf 0 gesetzt oder das Bild periodisch fortgesetzt werden.

Hat man nun die Makroblöcke vorliegen, so wendet man auf diese die Transformation an (sei es die DFT, die FFT oder die DCT). Das Ergebnis wird dann mit einem Kompressionsverfahren verknüpft, um die Größe jedes Makroblokes (und damit die Bildgröße) dramatisch zu verringern. Beispiele dafür wäre z.B. der Huffman-Code, welcher eindeutige Codes unterschiedlicher Länge generiert, die die ursprünglichen Werte darstellen. Dieses Encoding wäre sogar verlustfrei.

Bei der Dekompression geht man entsprechend umgekehrt vor: Man Ersetzt die gekürzten Codes mit den ursprünglichen Werten und wendet die inverse Transformation (IDFT, IFFT, IDCT) an, um das alte Bild z.B. darstellen zu können.

Macht das JPEG wirklich so? Nein, da die DFT/FFT u.U. imaginäre Werte liefert. Diese lassen sich aber nur auf wenigen (heute nicht mehr aktuellen) Maschinen in Hardware darstellen, und die Verarbeitung in Software dauert sehr lange.

Als Alternative wird daher in JPEG (vor Standard JPEG-2000) die *Diskrete Cosinus-Transformation*(DCT) verwendet, welche den selben Effekt wie die DFT erzielt, aber nur reelle Werte liefert.

Wo liegen die Informationsmaxima? Die Informationsmaxima liegen am Rand, da die höherfrequenten Anteile dort liegen.

Nenne die Formel der DFT

Fouriertransformation im \mathbb{R}^2 Die Fouriertransformation im zweidimensionalen Raum ist entsprechend die Variante für Bilder.

Hierbei hat man also Punkte $p = (x, y) \in \mathbb{Z}^2$, entsprechen betrachtet man auch die Fouriertransformation in zwei unterschiedlichen Richtungen (in x - und in y -Richtung). Formal übersetzt ergibt sich also die Fouriertransformation im \mathbb{R}^2 als

$$\hat{f}(\xi) = \int_{\mathbb{R}^2} f(t) e^{-i\xi^T t} dt \quad , \xi, t \in \mathbb{R}^2$$

2.2 Faltung

Was ist formal eine Faltung? Formal gesehen ist eine Faltung ein Integral. Dieses ist für Funktionen $f, g \in \mathbb{R}^n \rightarrow \mathbb{C}$ definiert als

$$(f * g) = \int_{\mathbb{R}^n} f(\cdot - t)g(t)dt.$$

Prinzipiell ist dieser Ausdruck definiert für alle Funktionen, die für *fast alle* $x \in \mathbb{R}$ wohldefiniert sind. In unserem eingeschränkten Kontext der integrierbaren Funktionen ist dies automatisch erfüllt.

Wie kann man sich so eine Faltung graphisch vorstellen? Man kann sich das so vorstellen, dass man eine Funktion über die zweite *schiebt* und dort die Übereinstimmung berechnet. Bei einer hohen Übereinstimmung zwischen den Funktionen wird dieses Integral einen hohen Wert liefern, während sich bei wenig Übereinstimmung nur ein kleiner Wert ergibt.

Was bringt so eine Faltung? Das im letzten Punkt beschriebene Verhalten der Faltung kann man sich z.B. in der Texterkennung zu nutze machen. Dabei prüft man z.B., ob es eine hohe Übereinstimmung zwischen einem Bild eines Buchstaben und dem vorliegenden Bild gibt, indem man über diese Bildteile die Faltung berechnet. Ein hoher Wert indiziert dabei, dass mit hoher Wahrscheinlichkeit der gesuchte Buchstabe vorliegt. Ganz allgemein ist dieses Prinzip die Grundlage von Filterungen.

3 Filter

Was ist ein Filter?

Wie sieht ein Filter allgemein aus?

Wie sieht ein Tiefpass aus? Zeichne den Graph

Forderungen an die Transferfunktion für Tiefpass

Erkläre das Gibbs-Phänomen

Welche Arten von Filter gibt es? Erklären Sie diese

Zeichnen Sie ein Schaltbild zu einem Filter (Addierer, Verzögerer, Multiplizierer)

Welchen Filter zur Kantenerkennung?

Was ist eine Impulsantwort?

Spielt die Laufzeit von Filtern in der Praxis eine Rolle?

3.1 Was ist ein Gradientenfilter?

Wozu ist er gut?

Wie funktioniert er?

Wie sieht so ein Filter aus? (Gradient + Filtermatrix)

Wie kommt man auf die Impulsantwort?

Nachteile + mögliche Gegenmaßnahmen

Wie sieht so ein Filter aus?

3.2 Filterbänke

Skizziere eine Filterbank und beschreibe den Vorgang

Nenne 3 typische Filter

Was ist die zentrale Eigenschaft von Filterbänken? Welche Voraussetzung muss dazu gelten?

Kann man Filterbänke hierarchisch aufbauen?

4 Abtastatz

Welche Eigenschaften müssen für eine Abtastung gelten?

Erkläre den Shannonschen Abtastatz

Wie funktioniert Abtasten überhaupt?

Was ist die kritische Abtastrate?

Wie berechnet man die kritische Abtastrate?

Was ist ein bandbeschränktes Signal?

Was ist der Träger eines Signals?

5 Hugh-Transformation

Was ist die Hugh-Transformation?

Wie funktioniert sie?

Was ermöglicht sie?

Welcher Filter spielt dabei eine Rolle?

Gibt es weitere Filter zur Kantenerkennung?

6 Muss noch zugeordnet werden

Erklären Sie die Heisenbergboxen