Bildverarbeitung: Mathematik arbeiten sehen

B. Burgeth, F. Kern
(Universität des Saarlandes)

burgeth@math.uni-sb.de
florkern@math.uni-sb.de

Bildverarbeitung, wozu?

Der Mensch ist ein visuelles Wesen, der den weitaus größten Teil der Informationen über seine Umwelt durch seinen Sehsinn erhält. Umgekehrt verbreitet er auch gezielt Wissen und Kenntnisse an andere über Zeichen („Lesen“) und Bilder, vor allem in neuerer Zeit, da ihm die Segnungen des modernen Technologie- und Medienzeitalters zur Verfügung stehen. Auch Schüler haben oft schon wie selbstverständlich Zugriff auf Computer und Digitalkamera (integriert im „Handy“). Damit sind ihnen auch schon die Möglichkeiten gegeben, Bilder zu machen, zu speichern und, vor allem, zu verändern. Diese Manipulation von Bildern geschieht mit mathematischen Methoden, und was kann deswegen näher liegen als die Wirkungsweise elementarer Mathematikkonzepte an Bildern zu veranschaulichen, mit Bildern einsehbar zu machen?
Um die Verdeutlichung einfacher mathematischer Konzepte an Bildern soll es in diesem Artikel gehen. Dabei werden wir uns auf folgende Fragen konzentrieren:
Was sind Bilder, wie können sie mathematisch beschrieben werden? Wie werden sie im Computer repräsentiert? Was bedeuten dabei Diskretisierung und Quantisierung? Wie kann man Bilder untersuchen, analysieren und auf einfache Weise gezielt verändern?

Die Verarbeitung aller Bilder und die Erstellung sämtlicher Graphiken in diesem Beitrag ist mit Hilfe des Computeralgebrasystems MAPLE15 [3] geschehen. Es bietet mit seinem *ImageTools Package* eine recht bequeme Möglichkeit verschiedene Bildfile-Formate, wie z. B. das bekannte .jpg-Format und fast kompressionsfreie .tif-Format, als Files ein- und auszulesen und ins ascii-format, also in eine Datei mit lesbaren Zahlen, umzuwandeln. Mit seinen anderen Funktionalitäten, z. B. dem *Statistics Package*, und dem *LinearAlgebra Package* gelingt dann eine statistische Analyse und die Verarbeitung der Bilddaten. Wer schon ein wenig Erfahrung mit MAPLE hat, wird durch die Hilfe-Funktion einen schnellen Einstieg in die Handhabung des Bildverarbeitungsmoduls finden. Zweifelsohne gibt es noch weitere leistungsfähige Computeralgebrasysteme, die Werkzeuge zur Bildverarbeitung anbieten. Dass die Wahl der Autoren auf Maple fiel, ist Zufall.

Repräsentationen von Bildern

Bilder als Funktionen
Wir betrachten hier so genannte Grauwertbilder, da diese grundlegenden Charakter auch für Farbbilder haben, und letztere konzeptionell für einen Einstieg ungeeignet erscheinen.
Ein kontinuierliches Grauwertbild kann als eine Funktion von zwei Veränderlichen betrachtet werden, also als Abbildung f vom Definitionsbereich $\Omega = [0, a] \times [0, b]$ in den Zielbereich $\mathbb{R}$:

$$f : \mathbb{R}^2 \supset \Omega \longrightarrow \mathbb{R}\,.$$

Ihre Definitionsmenge, der rechteckige Bereich Ω heißt Bildbereich oder auch, etwas missverständlich, Bildebene. Der Wertebereich dieser Funktion ist die Menge aller Grauwerte des Bildes. Dabei werden niedrige Grauwerte dunkel, hohe Grauwerte hell dargestellt.

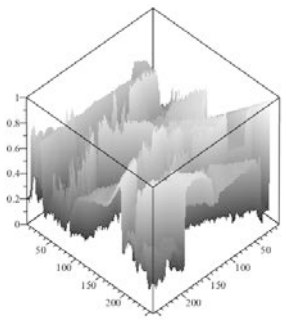

Abbildung 1: *Lena, eines der bekanntesten Testbilder in der Bildverarbeitung. Links: Original. Rechts: Als Funktionsgraph einer Funktion in zwei Veränderlichen.*

In dieser Seitenansicht des dreidimensionalen Graphen G_f ist Lena nur schwer zu erkennen. Eine schrittweise Rotation des Graphen, so dass man

am Ende von oben auf ihn blickt, bringt Klarheit.

Abbildung 2: *Von links oben nach rechts unten: Schrittweise Rotation.*

Diskretisierung 1: Bilder als Matrizen

Natürlich kann man keinen Funktionsterm angeben, dessen Graph das Bild von Lena darstellt. Man kann das Bild nur in Form einer recht umfangreichen Wertetabelle im Computer abspeichern. Das Erstellen dieser Wertetabelle nennt man abtasten und meint die Diskretisierung des Bildbereiches. Die Bilddaten sind dann nur auf den Punkten (i, j) eines Rechteckgitters in Ω gegeben und wir haben auf diese Weise ein **digitales Bild** erzeugt

$$\{f_{i,j} = f(i,j) \,|\, i = 1, \ldots, N, j = 1, \ldots, M\},$$

das auch als Matrix angesehen werden kann:

$$(f_{i,j})_{i,j} = \begin{bmatrix} 0.38 & 0.48 & 0.51 & 0.73 & 0.43 & 0.55 & 0.91 & 0.20 \\ 0.38 & 0.75 & 0.44 & 0.72 & 0.81 & 0.67 & 0.17 & 0.63 \\ 0.39 & 0.82 & 0.57 & 0.46 & 0.73 & 0.69 & 0.65 & 0.59 \\ 0.44 & 0.58 & 0.30 & 0.41 & 0.74 & 0.25 & 0.62 & 0.50 \\ 0.45 & 0.60 & 0.63 & 0.61 & 0.81 & 0.14 & 0.57 & 0.81 \\ 0.48 & 0.34 & 0.25 & 0.36 & 0.25 & 0.61 & 0.45 & 0.19 \\ 0.58 & 0.18 & 0.40 & 0.55 & 0.68 & 0.56 & 0.86 & 0.36 \\ 0.37 & 0.29 & 0.42 & 0.54 & 0.61 & 0.46 & 0.29 & 0.42 \end{bmatrix}$$

Die Gitterpunkte (man spricht auch von Gitterzellen) (i, j) heißen **Pixel**. Es ist in der Bildverarbeitung aus Gründen der Einfachheit üblich den Abstand der Gitterpunkte (bzw. Gitterzellenseitenlängen) auf 1 zu normieren. Dies bringt auch eine vereinfachte und sparsamere Speicherung der Bilder auf dem Computer mit sich, da keine expliziten Gitterpunktkoordinaten gespeichert werden müssen. Wenn man einen Ausschnitt eines digitalen Bildes stark vergrößert, tritt die Pixelstruktur deutlich sichtbar hervor.

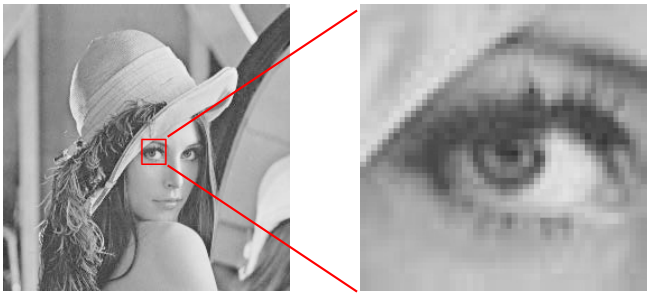

Abbildung 3: *Links: Lena. Rechts: Auge vergrößert.*

Tastet man beispielsweise die Funktion auf einem gröberen Gitter ab, so erhält man eine kleinere Matrix, die durch dieses **Vergröbern** (engl. down sampling) allerdings als ein „verpixeltes“ Bild interpretiert wird, bei dem viele Deteils verlorengingen.

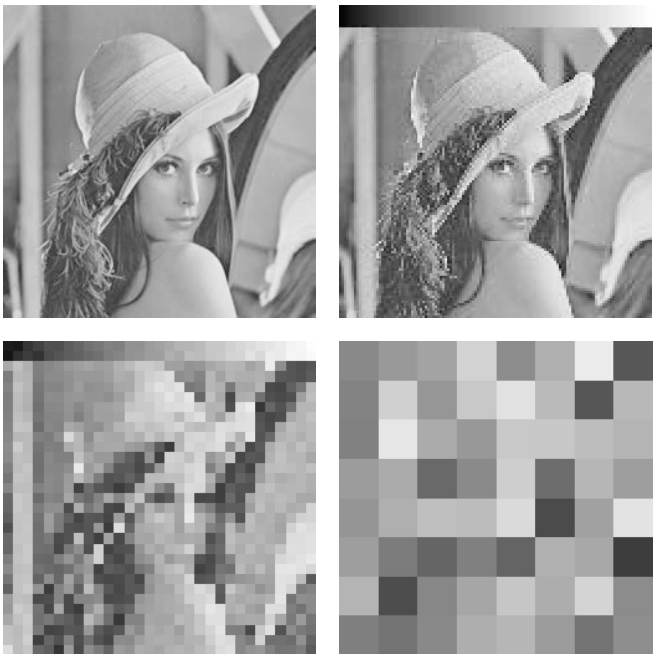

Abbildung 4: *Auflösung in $n \times n$ Pixel. Links oben:* 512×512. *Rechts oben:* 128×128. *Links unten:* 32×32. *Rechts unten:* 8×8.

Man sieht sofort: bei einer Auflösung von 8×8 Pixel ist Lena nicht mehr zu erkennen. Das letzte Bild entspricht übrigens gerade der eben genannten 8×8-Matrix $(f_{i,j})$. Es sei noch angemerkt, dass es sich bei dem Streifen am oberen Bildrand um kein Diskretisierungsartefakt handelt, es ist vielmehr ein Teststreifen, der eine gleichmäßig von null (schwarz) nach eins (weiss) ansteigende lineare Funktion darstellt. Dies wird bei noch folgenden Ausführungen noch eine Rolle spielen.

Diskretisierung 2: Quantisierung

Mit **Quantisierung** meint man in der Bildverarbeitung die Diskretisierung des Wertebereichs $f(\Omega)$. Das führt im Extremfall zu binären Bildern, also Schwarz-Weiss-Bildern: $f(\Omega) = \{0, 1\}$. Als Speicherplatz noch kostbar war, benutzte man auch noch die byte-Codierung der Grauwerte, die eine Unterteilung in 256 Graustufen

erlaubte: $f(\Omega) = \{0, 1, \ldots, 255\}$. Dies ist völlig ausreichend, da Menschen durch ihre Physiologie bestimmt nur etwa 40 verschiedene Grauwerte unterscheiden können. In den Zeiten leistungsfähiger Computer findet man heute oft einen quasi-kontinuierlichen Wertebereich bei Bildern: $f(\Omega) = [0, 1]$. Dies wollen wir auch für diesen Aufsatz annehmen, falls nicht anderweitig angemerkt.

Abbildung 5: *Quantisierung beim Bild Kameramann. Links:* 256 *Grauwertstufen. Rechts:* 4 *Grauwertstufen, mit deutlichem Effekt auch auf den Teststreifen.*

Wann ist ein Bild ein Bild? Verteilung von Grauwerten

Ein digitales Bild ist eine Art diskretisierte Funktion und kann als Matrix betrachtet werden, deren Einträge Zahlen aus dem Intervall $[0, 1]$ sind. Ordnet man diese Einträge in einer fest vorgegebenen Reihenfolge (etwa Zeile für Zeile der Matrix) hintereinander an, so erhält man einen Vektor, aber einen sehr hochdimensionalen: das 256×256 Bild von Lena ergibt einen Vektor der Dimension $256 \cdot 256 = 65536$! Ein Schüler, dem das klar geworden ist, wird dem allgemeinen Konzept eines n-dimensionalen Vektors im Mathematikunterricht wohl etwas offener gegenüber stehen.
Die in einem Bild enthaltene Information steckt in der räumlichen Verteilung der Grauwerte, und diese können wir uns „in Schichten“ ansehen, in Form von **Niveaumengen**. Dazu denken wir uns ein digitales Bild mit 256 Graustufen gegeben $f : \Omega \longrightarrow \{c_0, \ldots, c_{255}\} \subset [0, 1]$. Mit dem Begriff der Niveaumenge haben Schüler und Anfänger im Studium in der Regel Schwierigkeiten. Dieses und verwandte Konzepte lassen sich an einem Bild sinnfällig veranschaulichen. Wie zu erwarten ist die **Niveaumenge** zum Grauwert $c_k \in \{c_0, \ldots, c_{255}\}$ als die Menge

$$\{(i, j) \in \Omega \,|\, f(i, j) = c_k\} = f^{-1}(c_k)$$

definiert, also als Teilmenge des Bildbereichs, mathematisch eine Urbildmenge. Anders das **Niveaubild**:

$$I^f_{c_k}(i, j) = f(i, j) \cdot \mathbf{1}_{f^{-1}(c_k)}(i, j) = c_k \cdot \mathbf{1}_{f^{-1}(c_k)}(i, j)$$

mit $\mathbf{1}_M$ als der Indikatorfunktion einer Menge M,

$$\mathbf{1}_M(x) = \begin{cases} 1 \text{ falls } x \in M \\ 0 \text{ falls } x \notin M \end{cases}$$

Das Niveaubild hat einen bestimmten Grauwert (nämlich c_k, siehe Abb. 2) und ist eine Teilmenge des Graphen der (diskretisierten) Funktion.
Auch die Wirkungsweise der Indikatorfunktion als wichtiges Instrument einer Fallunterscheidung kann damit demonstriert werden.

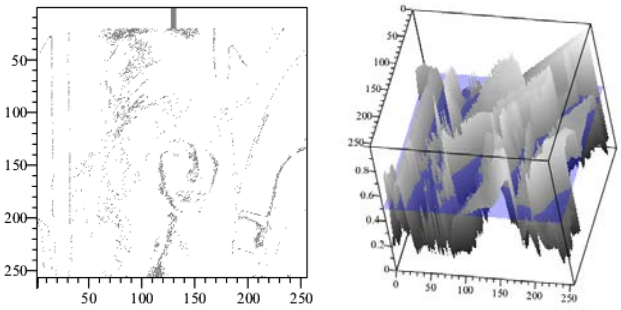

Abbildung 6: *Räumliche Verteilung von Grauwerten. Links: Niveaubild zum Grauwert* 0.5 *in 2D. Rechts: Niveaubild zum Grauwert* 0.5 *in 3D.*

Diese Schichten eines Bildes lassen sich auch wieder zusammensetzen, man hat die interessante Zerlegung:

$$f = \sum_{k=0}^{255} I^f_{c_k} = \sum_{k=0}^{255} c_k \cdot \mathbf{1}_{f^{-1}(c_k)} \,.$$

Einen ganz anderen Blick auf ein Bild gewinnt man, wenn die **Häufigkeitsverteilung der Grauwerte** betrachtet wird. Dies bietet einen anwendungsnahen Einstieg zum Thema **Histogramme** über der Menge der Grauwerte eines Bildes.
Es geht also darum, wie oft ein Grauwert in einem diskreten Bild $f : \Omega \longrightarrow [0, 1]$ vorkommt, genauer: Die Zuordnung

$$H : c \longmapsto |f^{-1}(c)| = \text{Anzahl}(f^{-1}(c))$$

liefert das Histogramm H zur Verteilung der Grauwerte, $H : [0, 1] \longrightarrow \{0, \ldots, |\Omega|\}$. Da es nur um Anzahlen geht, ist die räumliche Anordnung der Pixel in Ω ohne Bedeutung für das Histogramm. Und gerade die Chance, an einem Objekt, dem Bild, diese beiden Aspekte von „Verteilung“ gegenüber stellen zu können macht den Reiz dieser Betrachtungsweise aus. Das Histogramm selbst kann in Form eines Punkte-, Stäbchen- oder, am häufigsten, in der Gestalt eines Balkendiagramms dargestellt werden. Die folgenden Abbildungen zeigen einige Beispiele.

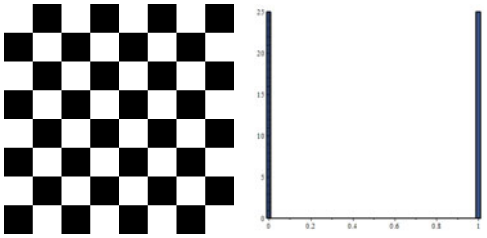

Abbildung 7: *Häufigkeitsverteilung von Grauwerten bei einem binärem Bild. Links: Schachbrettmuster. Rechts: Zugehöriges Histogramm.*

Und manchmal zeigt ein Histogramm doch mehr als ein Blick auf das Bild, zum Beispiel beim sogenannten

„Rauschen“.

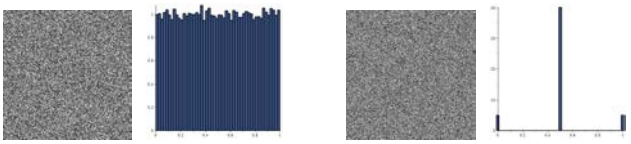

Abbildung 8: *Häufigkeitsverteilung von Grauwerten bei zwei häufig auftretenden Arten von Rauschen. Oben: Gleichverteiltes Rauschen mit Histogramm. Oben: „Salz und Pfeffer“-Rauschen mit Histogramm.*

Auch solche Bilder, die nur Rauschen, d. h., von einem Pixel zum anderen einen starken und zufälligen Wechsel in den Grauwerten zeigen, sind Bilder. Am zugehörigen Histogramm wird deutlich, um welche Art von Rauschen es sich handelt, was wichtig sein kann zur Entwicklung von „Entrauschungsalgorithmen“.

Nicht nur schauen: Verarbeiten von Bildern

Bis jetzt haben wir Bilder mehr oder weniger nur analysiert. Jetzt geht es um die tatsächliche Verarbeitung von Bildern und wir stellen uns die Frage, ob man reellwertige Funktionen $T : D \supset \mathbb{R} \longrightarrow \mathbb{R}$ auf Bilder anwenden und sie auf diese Art transformieren kann. Die Antwort ist, wie zu erwarten, ja, denn diese Anwendung läuft auf eine Verknüpfung von T mit der Funktion f hinaus, deren dreidimensionaler Graph G_f das Bild darstellt (vgl. Abschnitt 1). Allerdings taugt nicht jede Funktion als transformierende Abbildung T, genauer: zu einem Bild repräsentiert durch $f : \Omega \longrightarrow [0,1]$ und der Funktion $T : [0,1] \longrightarrow [0,1]$ (!) liefert die Verknüpfung $T \circ f$ von Transformation T und Bild f das **transformierte Bild**,

$$T \circ f : \Omega \longrightarrow [0,1]\,.$$

Im ersten Beispiel werden wir sehen, dass der Begriff des „Negativs“ eines Bildes nicht völlig zutreffend ist.

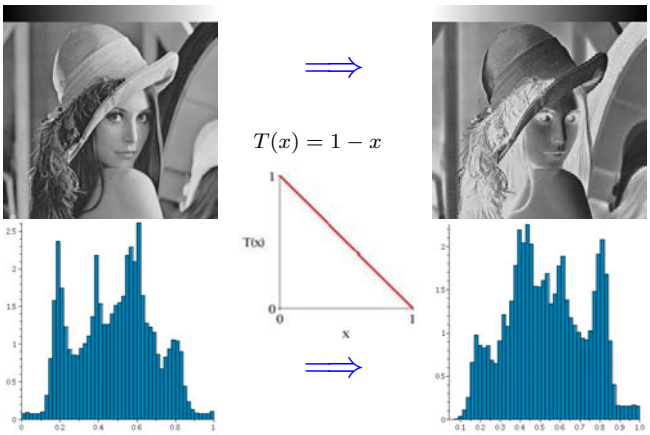

Abbildung 9: *Transformationen von Bildern. Rechte Spalte: Lena (Bild f) und zugehöriges Histogramm. Mittlere Spalte: Transformation T und Graph. Rechte Seite: „Transformierte“ Lena (Bild $T \circ f$) mit (gespiegeltem) Histogramm.*

Die Transformation verändert auch die Häufigkeitsverteilung der Grauwerte, d. h. die Histogramme. An den folgenden Beispielen wird „sichtbar“ wie sich Monotonie- und Stetigkeitseigenschaften von T im Ergebnisbild und -histogramm widerspiegeln.

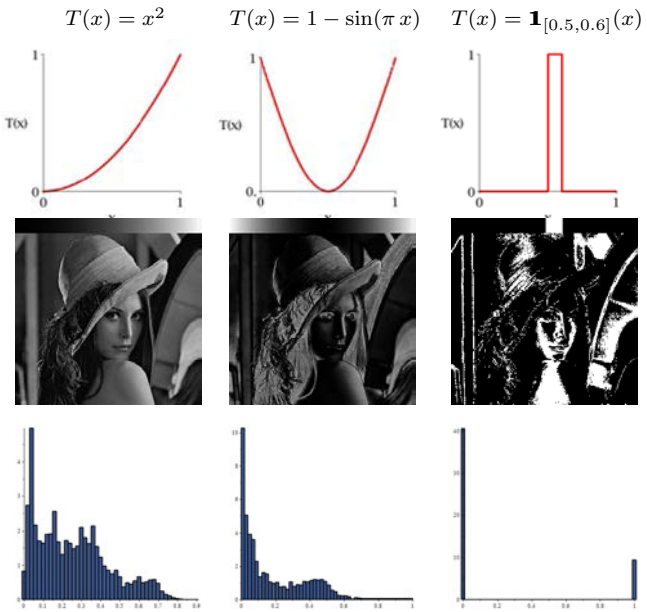

Abbildung 10: *Beispiele (spaltenweise gelesen) von Transformationen. Die angegebenen Transformationen werden jeweils auf das originale Lena-Bild angewandt. Aus dem Teststreifen läßt sich die Gestallt des Graphen G_T der Transformation T ableiten.*

Man darf hoffen, dass die/der interessierte Schüler/in vielleicht von sich heraus versucht, mit verschiedenen Transformationen ansprechende Effekte zu erzielen (und z. B. aus einer Bildergeschichte mit selbstaufgenommenen Fotos ein Comic-Strip zu machen).
Es ergeben sich auch didaktisch interressante Fragestellungen:
Bei gegebenem T, kann man grob voraussagen, wie Bild und Histogramm verändert werden wird? Wann kann ein transformiertes Bild wieder rekonstruiert werden? Das führt zu einer sinnhaften Auseinandersetzung mit dem Konzept der Umkehrfunktion in Form der Umkehrabbildung T^{-1}. Was ist zu tun, wenn der Wertebereich einer Funktion T nicht von vorne herein in $[0,1]$ enthalten ist, $T([0,1]) \not\subset [0,1]$? Welches Funktionsdesign ist dann nötig, stauchen, „klippen“,...?

Rückblick

Wir hoffen mit diesem Aufsatz wenigstens ansatzweise aufgezeigt zu haben, welches Potenzial die mathematische Bildverarbeitung bietet, zahlreiche Zusammenhänge und Konzepte der (Schul-)Mathematik vernetzt darzustellen und visuell erfahrbar zu machen. Als Bücher, die allgemein von Bildverarbeitung handeln, seien hier [1] und [2] genannt.

Literatur

[1] Rafael C. Gonzalez und Richard E. Woods. *Digital Image Processing*, 2008.

[2] Milan Sonka, Vaclav Hlavac und Roger Boyle. *Image Processing, Analysis and Machine Vision*, 1999.

[3] *Maple 15.* www.maplesoft.com (zuletzt aufgerufen am 14.02.2013)