

Principles of Image Generation

Copyright © by V. Miszalok, last update: 15-08-2006

- ↓ [Definition](#)
- ↓ [Die Bildmatrix](#)
- ↓ [Prinzipien der Diskretisierung = Abtastung](#)
- ↓ [Kompression](#)

Definition

Image Processing = Picture Processing = Bildverarbeitung ist die Umwandlung redundanzreicher Bilder in andere (bessere) redundanzreiche Bilder und deren platz sparende Speicherung (=Kompression).

Beispiele: Scanner, Digitalkopierer, DigitalPhoto, DigitalTV, dig. Röntgen, Computertomographie, Radar, Ultraschall, Wettersatelliten.

Wichtig: Verwechseln Sie nicht Image Processing mit Computer Vision ! Beide verarbeiten Bilder, aber Computer Vision erzeugt aus Bildern keine neuen Bilder, sondern einen oder wenige Messwerte.

Wichtig: Verwechseln Sie nicht Computer Vision mit Kompression ! Beide vernichten Redundanz aber Kompression konserviert dabei das Bild, während Computer Vision das Bild vollständig vernichtet.

Die Bildmatrix

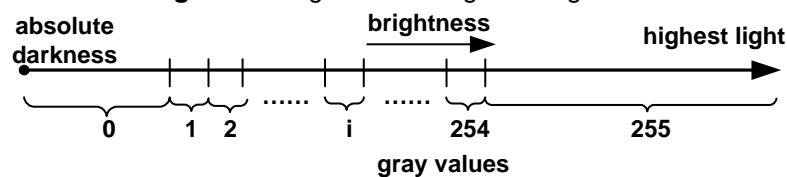
Bildmatrizen sind Rastermatrizen (siehe Vorlesung: [Raster Graphik](#)), die in Kameras und Scannern durch Diskretisierung und Quantisierung (= Digitalisierung) entstehen.

Selten entstehen sie durch Menschenhand: orthogonale Mosaik, Kachelwände, Icons.

Wichtige Definitionen:

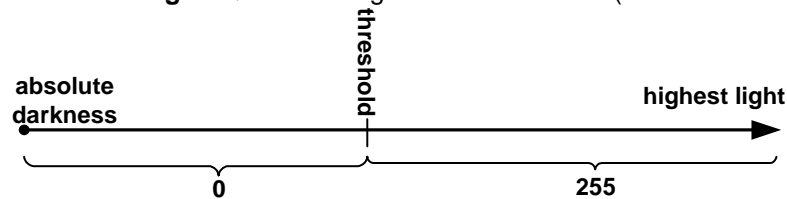
Diskretisierung: Abbildung der realen Welt auf ein Gitter von lichtempfindlichen Sensoren = Abtastung.

Quantisierung: Abbildung einer analogen Helligkeitsachse auf eine Menge ganzer Zahlen = AD-Wandlung.



Digitalisierung: = Hintereinanderausführung von 1) Diskretisierung und 2) Quantisierung.

Binarisierung: = Quantisierung mit exakt 2 Stufen (oft: 0-1 oder 0-255) mit Hilfe einer Schwelle = threshold.



Prinzipien der Diskretisierung = Abtastung

Um eine Rastermatrix zu erzeugen, muss ein Abtastgerät liefern:

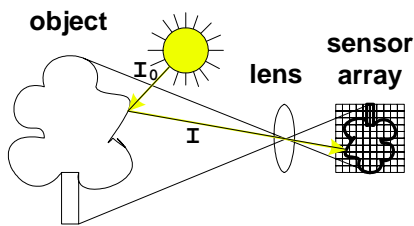
1. Spalten- und Zeilennummern x und y

2. für jede Stelle x, y der realen Welt folgenden Quotienten (messen oder schätzen):

Energie I des von der Stelle x, y der realen Welt ausfallenden Lichtes dividiert durch Energie I_0 des auf die Stelle x, y der realen Welt einfallenden Lichtes: $\text{quotient}(x, y) = I / I_0$.

Dieser Quotient muss in einen Grau-/Farbwert umgerechnet werden: $\text{grayvalue}(x, y) = \text{function}(I / I_0)$

Alternative Erzeugungen einer Rastermatrix $M(x, y)$:



a) Abtastung mit ungerichtetem Licht und vielen Sensoren (Photoprinzip):

Die Sonne oder sonst eine nichtsteuerbare Lichtquelle beleuchtet die Außenwelt. Eine Blende oder eine Linse entwirft ein verkleinertes Bild der Außenwelt auf eine Ebene, wo in Spalten und/oder Zeilen lichtempfindliche Sensoren angeordnet sind. Die Sensorsignale werden in ganze Zahlen verwandelt und positionsrichtig in die Matrix M eingeschrieben. Das Sensorgitter = Diskretisierer liefert x, y und ein Analog-Digital-Wandler = Analog-Digital-Converter = ADC = Quantisierer liefert den Grauwert. Vorteile: 1) natürliches Licht 2) hohe Zeitauflösung = schnelle Abtastung bewegter Objekte möglich

Nachteile: 1) sehr viel Licht I_0 notwendig

2) niedere Ortsauflösung durch begrenzte Spalten/Zeilenzahl des Sensors

3) häufig Bildfehler: unterbelichtet, überbelichtet, Shading, Unschärfe

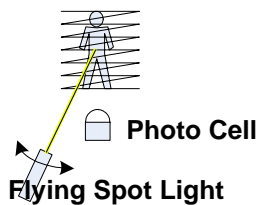
b) Abtastung mit gerichtetem Licht und einem Sensor (Scanner-Prinzip):

Eine steuerbare Lichtquelle bewegt und zielt einen Lichtstrahl (flying spot) mit bekanntem Spektrum und bekannter Energie auf einen Punkt x, y . Der Lichtstrahl wird von der Außenwelt teils verschluckt (Absorption), teils durchgelassen (Transmission), teils zurückgeworfen (Reflektion). Ein einziger Sensor = Photozelle fängt einen Teil der durchgelassenen oder der reflektierten Energie ein, misst oder schätzt den Quotienten und ein ADC erzeugt daraus eine ganze Zahl. Zu jedem Zeitpunkt liefern die Digitalmotoren der Strahlsteuerung die Position x, y und Photozelle+ADC liefern genau einen Grau- oder Farbwert `grayvalue`.

Vorteile: 1) hohe Ortsauflösung = hohe Spalten- und Zeilenzahlen möglich

2) kaum Streulicht, daher hohe Bildschärfe 3) kein Shading

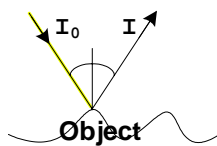
Nachteil: geringe Zeitauflösung: = Abtastung bewegter Objekte nicht möglich



c) Abtastung undurchsichtiger Objekte = Reflektions-Abtastung:

Vorteile: 1) Geht bei fast allen Objekten, die in der Regel undurchsichtig sind. 2) weite Abstände zwischen Lichtquelle, Objekt und Sensor möglich

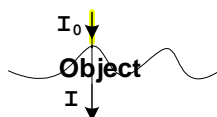
Nachteil: Der `quotient` ist niedrig, weil das meiste Licht verschluckt = absorbiert oder gestreut wird.



d) Abtastung durchsichtiger Objekte = Transmissions-Abtastung:

Vorteil: Der `quotient` ist hoch, weil wenig Licht verschluckt = absorbiert oder gestreut wird

Nachteile: 1) Geht oft nur mit speziellem Licht z.B. Röntgen, Gamma-Strahlung. 2) Kurze Abstände zwischen Lichtquelle, Objekt und Sensor erforderlich.



Beispiele:

Photoabtastung undurchsichtiger Objekte: Photo, Video

Scannerabtastung undurchsichtiger Objekte: Flachbettscanner, Kopierer, Radar

Photoabtastung durchsichtiger Objekte: Diapositiv, Overheadfolie, LCD-Beamer, Kinofilm, Röntgen, Mikroskopie von Zellen und Geweben

Scannerabtastung durchsichtiger Objekte: Computertomographie, Ultraschall, Gamma-Kamera

Mischformen: Die handelsüblichen Flachbettscanner und Digitalkopierer sind eine Mischung aus Photoabtastung und Scannerabtastung. Sie besitzen ein oder mehrere einzeilige Sensorarrays, in der Regel mit 4096 Sensoren = Photoprinzip, die sie mit Linearmotoren parallel verschieben = Scannerprinzip.

Beste Bildqualität liefert eine Kombination aus Scanner- und Transmissionsabtastung: viel Zeit, kein Streulicht, kein Shading und hoher `quotient` ermöglichen hohe Ortsauflösung und Bildschärfe.

Kompression

Bilder enthalten enorme Redundanz und stellen deshalb enorme Anforderungen an Speicherplatz und Übertragungsbandbreite. Bei begrenzter Kapazität entsteht der Zwang zur Redundanzvernichtung = Verringerung der Codelänge = Kompression.

Als Maß für die Codeverkürzung dient folgender Quotient:

$$\text{Kompression} = \frac{\text{Codelänge nachher}}{\text{Codelänge vorher}} * 100\%$$

Dabei muss selbstverständlich im Zähler und im Nenner mit einem einheitlichen Datentyp (normalerweise Bit) gemessen werden.

Man unterscheidet:

a) verlustfreie Kompression, d.h. aus den komprimierten Daten lässt sich das Ursprungsbild exakt wiederherstellen.

b) verlustbehaftete Kompression, d.h. das wiederhergestellte Bild ist schlechter als das Original.

Nachteile von Kompression:

- 1) Das Bild muss vor dem speichern/versenden von einem Kompressionsprogramm verarbeitet werden.
- 2) Das Bild muss nach vor dem laden/empfangen von einem Dekompressionsprogramm verarbeitet werden.
- 3) Kompressions- und Dekompressionsprogramm müssen exakt spiegelbildlich zueinander passen.
- 4) Kompressions- und Dekompressionsprogramme stellen hohe Anforderungen an Hardware und temporären Speicherplatz.
- 5) Die Kompression ist nur statistisch aber nicht individuell pro Bild voraussagbar.
- 6) Bei verlustbehafteten Kompressionen ist der Verlust nur statistisch voraussagbar.

Das älteste und am häufigsten verwendete Verfahren ist:

Verlustfreie Kompression von Binärbildern durch Lauflängencode = Run Length Code = RLC = Run Length Encoding = RLE

wird in der Fax-Übertragung verwendet.

Es werden die Anzahlen aufeinander folgender Nullen und Einsen übertragen.

Es codiert die erste, dritte, fünfte etc. Zahl jeder Zeile die Nullen, die zweite, vierte, sechste etc. Zahl codiert die Einsen. Eine leere Zeile wird codiert durch ein einziges Integer = Anzahl der Pixel pro Zeile.

Zeilen, die mit eine Eins beginnen, bekommen eine führende Null.

Oft endet eine Zeile mit einem Sonderzeichen = Zeilentrenner zur Fehlererkennung (siehe unten).

Starke Kompression bei Schriftgut (Fax), da eine durchschnittliche Schriftgutseite nur ca. 2% geschwärzt ist und deshalb die Zahl der Nullen exorbitant hoch ist.

Beispiel: 4-zeiliges Binärbild B

```

0011001000      22213
B = 0101010000  -> RLC = 1111114
1010101001      0111111121
0111110000      154

```

Summe = 40 Bit Summe = 25 Integer

Das Beispiel komprimiert 40 Bit zu 25 Integer. Letztere speichert man normalerweise mit dem Datentyp UInt16 (= 16 Bit ohne Vorzeichen). Die Kompression beträgt damit $25 * 16 \text{Bit} * 100\% / 40 \text{Bit} = 1000\%$.

Das komprimierte Bild ist also 10 mal länger als das unkomprimierte.

Kompressionen über 100% sind natürlich unerwünscht. Sie können auftreten bei:

- 1) kleinen Bildern
- 2) Bildern mit schnell wechselndem Inhalt, wie etwa einem Schachbrettmuster
- 3) Bildern, die bereits komprimiert sind (Kompressionen sind nicht kaskadierbar !).

Bei realen Binärbildern mit Zeilenlängen von 4096 (wie bei Fax) sind aber Kompressionen von 5% die Regel.

Man kann auch Grauwertbilder und Farbbilder mit dem RLC komprimieren. Man muss vor jeder Zahl noch den zugehörige Grau-/Farbwert übertragen. Bei Bildern mit großen homogenen Grau-/Farbflächen kann die Kompression trotzdem effektiv sein.

Beispiel: AAAAAAABBBB -> A7B4

Wenn die Anzahl der Pixel pro Zeile bekannt ist (z.B. beim Fax), dann benötigt der RLC keine Zeilentrennzeichen.

Begründung: Der Empfänger addiert die ankommenden Zahlen und weiß damit, wann eine Zeile voll ist. Er kann aber nicht kontrollieren, ob der Sender der gleichen Meinung ist, d.h. er kann Übertragungsfehler (fehlende Zahlen oder falsche Längen) nicht erkennen.

Beispiel für einen schwerwiegenden Fehler: Wenn eine einzelne Zahl verloren geht, dann verwechselt der Empfänger für den ganzen Rest des Blattes Vordergrund (=schwarz) und Hintergrund (=weiß).

Der Einbau der Trennzeichen erlaubt folgende Fehlerbehandlung: Stimmen die Position des Trennzeichens und das vom Empfänger errechnete Zeilenende nicht überein, dann wird eine Zeilenwiederholung angefordert. Die redundanten Trennzeichen erhöhen also die Sicherheit der Übertragung, verschlechtern allerdings die Kompression.

Weitere verlustfreie Kompressionsverfahren:

Huffman: häufige Grauwerte durch wenige Bits, seltene durch mehr Bits codieren.

Lempel-Zif-Welch = LZW in ZIP, GIF, PDF: häufige Grauwert-/Farbfolgen nur einmal speichern, ab dann genügt kurzer Verweis.

Verlustbehaftete Kompressionsverfahren:

JPEG, MPEG auf der Basis der Fouriertransformation