

Punktoperationen

Methoden der Bildverarbeitung auf einzelnen
betrachteten Bildpunkten

Wolfgang Heiden © 2014-22, mit Beiträgen von R. Herpers, F. Mannuß, B. Kahl, G. Heisenberg

Wolfgang Heiden © 2014-22 wolfgang.heiden@h-brs.de

-- auf der Grundlage einer Lehrveranstaltung von Prof. Dr. Rainer Herpers sowie Folien
von Florian Mannuß 2011, Dr. Björn Kahl 2012, Prof. Dr. G. Heisenberg, 2013 --

Fachbereich Informatik (Dpt. Computer Science)

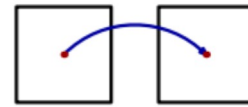
Hochschule Bonn-Rhein-Sieg – University of Applied Sciences,

53754 Sankt Augustin

Germany

● Punktoperatoren

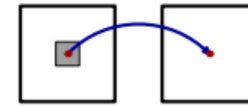
- Unabhängige Betrachtung einzelner Pixel/Punkte



Punktoperator

● Lokale Operatoren (Maskenoperatoren)

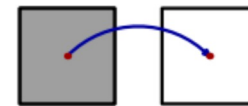
- Betrachtung von Bildbereichen unter einer Maske
- Es werden einzelne Pixel und deren Umgebung transformiert.



Lokaler Operator

● Globale Operatoren

- Berücksichtigung des Bildes als Gesamtheit
- aufwändig



Globaler Operator

Operatoren zur Transformation von Bildern lassen sich in drei Kategorien unterteilen:

1. Punktoperatoren: Hier werden sämtliche Bildpunkte des Ursprungsbildes (UB) einzeln und unabhängig voneinander mittels derselben Operation bearbeitet, die zu einem entsprechenden Wert beim zugeordneten Bildpunkt im Ergebnisbild (EB) führt. solche Operationen lassen sich sehr gut parallelisieren.
2. Maskenoperatoren: Ein Bildbereich definierter Größe im UB, ausgehend von einem einzelnen Bildpunkt, beeinflusst mittels der Maskenoperation einen einzelnen (korrespondierenden) Bildpunkt im EB. In diese Kategorie fallen die typischen sog. "Faltungsfiler".
3. Globale Operatoren: Die Operation zur Erzeugung von Werten auf den Bildpunkten des EG aus den Bildpunkten des UB bezieht für jeden einzelnen Bildpunkt das gesamte UB mit ein. Solche Operationen sind mit sehr hohem Rechenaufwand verbunden.

[Folie nach G. Heisenberg]

● Was sind Punktoperationen?

- bezogen auf einzelne Pixel
- anwendbar auf ein Bild oder Bildsequenzen
- Homogene Operationen
- unabhängig von Pixel (u, v)
(jedes Pixel wird gleich behandelt)
- $I'(u, v) = f(I(u, v))$ für Einzelbild
- $I'(u, v) = f(I_1(u, v), \dots, I_n(u, v))$ für
Bildsequenzen

Punktoperationen bilden Pixel homogen ab, d.h. ein Pixel des Ursprungsbildes wirkt sich auf genau ein Pixel des Ergebnisbildes aus.

Sie können sowohl auf einzelne Bilder angewendet werden (z.B. RGB->Graustufen) oder auf Bildsequenzen (z.B. Differenz zweier Graustufenbilder identischer Abmessungen).

Beispiele für Punktoperationen

● Einzelbilder

- Umwandlung
Farbwerte zu Graustufen
- Kontrast- und
Helligkeitskorrektur
- Schwellwert
- Nichtlineare
Grauwertkorrektur
 - Gammakorrektur
- Histogrammskalierung

● mehrere Bilder

- Mittelwertbildung
(Rauschminderung von
Bildsequenzen)
- Differenzbild zweier Bilder
- Logische Operationen
- Quantisierung

● Helligkeitsänderung

- Addition/Subtraktion eines festen Wertes zu $I(u, v)$

$$g'_i = g_i + a; \quad g'_i, g_i \in [0, 255]$$

- g' , g Grauwerte; a Verschiebungswert

● Kontraständerung

- Spreizung der Grauwerte

$$g'_i = g_i \cdot b$$

● Kombination

$$g'_i = g_i \cdot b + a$$

Globale **Helligkeitsänderungen** addieren einen positiven (heller) oder negativen (dunkler) Betrag zu den Intensitätswerten jedes Pixels. Dabei ist allerdings darauf zu achten, dass die Grenzen des Wertebereichs (i.d.R. 8 bit \rightarrow 0-255) nicht verlassen werden. **Clamping** setzt alle negativen Ergebniswerte auf 0 und alle Werte >255 auf 255. Eventuell vorhandene Helligkeitsunterschiede im Ursprungsbild gehen dabei in den betroffenen Wertebereichen allerdings verloren.

Spreizen der Grauwerte durch Multiplikation mit einem bestimmten Faktor kann den **Kontrast** in einem Bild verstärken (Verbreiterung, $b > 1$) oder vermindern (Stauchung, $0 < b < 1$). Bei Kontrastverstärkung ist auch hier darauf zu achten, dass der Wertebereich nicht verlassen wird.

Bedeutung der Variablen:

g_i = Grauwert eines Pixels i (Original)

g'_i = Grauwert eines Pixels i (nach Punktoperation)

a = Offset, wird hinzu addiert

b = Multiplikator, Skalierungsfaktor

• Was tun bei Verletzung des Wertebereichs?

- **Clamping**
$$g'_i = \begin{cases} g'_i & , \text{ wenn } g'_i < 255 \\ 255 & , \text{ sonst} \end{cases}$$
- oder **Skalierung**, z.B. Division durch 2



Original



Korrekturwerte: $a = 10$; $b = 1,25$

Clamping = Deckelung am Rand des zulässigen Wertebereichs

Skalierung ist nur dann sicher, wenn $a \leq g_{\max}/b$

g_i = Grauwert eines Pixels i (Original)

g'_i = Grauwert eines Pixels i (nach Punktoperation)

a = Offset, wird hinzu addiert

b = Multiplikator, Skalierungsfaktor

Automatische Kontrastanpassung durch Grauwertspreizung (Wiederholung)

Spreizen des Grauwertbereichs des Bildes auf $[0, 255]$

$$i_{new} = b_{low} + \frac{(b_{high} - b_{low})}{(a_{high} - a_{low})} \cdot (i - a_{low}); \quad a_{low} \leq i \leq a_{high}$$



$a_{low} = 100$; $a_{high} = 200$



$b_{low} = 0$; $b_{high} = 255$

vgl. Grauwertspreizung auf Histogramm-Basis

Schwellwert

- Spezielle Form der Quantisierung → **Binarisierung**
- Trennung der Bildwerte in zwei Klassen → schwarz/weiß

$$g'_i = \begin{cases} a_{low} & , \text{ wenn } g_i \leq a_S \\ a_{high} & , \text{ wenn } g_i > a_S \end{cases}$$

- Schwellwert a_S
- Bsp.: $a_{low} = 0$ und $a_{high} = 255$



Schwellwert: 128

Für spätere Verarbeitungsschritte ist gelegentlich die Binarisierung eines Bildes erforderlich, so dass jeder Bildpunkt entweder den Wert 0 (schwarz) oder 1 (weiß) erhält.

Im einfachsten Fall wird dafür ein Schwellwertverfahren angewandt, d.h. alle Werte unterhalb des Schwellwerts werden im Ergebnisbild schwarz, alle darüber weiß. Welcher Kategorie der Schwellwert selbst zugeordnet wird, ist zu definieren.

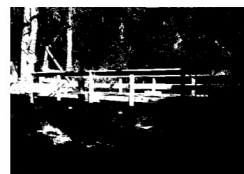
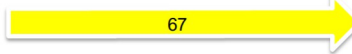
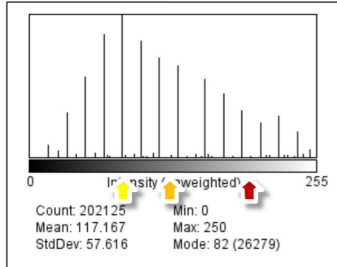
Im Beispiel wird in einem Bild mit Grauwerten von 0-250 der Schwellwert der unteren Kategorie zugewiesen.

Bsp.: Originalbild: 0-250 (mit ImageJ gemessen)

Schwellwert-Beispiele



300x240 pixels; RGB; 281K



Die Wahl eines geeigneten Schwellwerts ist kritisch.

Im Beispiel zeigen sich zwischen den Schwellwerten 67 und 66 bei nur einer Grauwertstufe Unterschied deutlich sichtbare Auswirkungen auf das Ergebnis!

Treten im Histogramm eines Bildes mehrere Maxima auf, ist es meist sinnvoll, bei der Binarisierung einen Schwellwert zwischen den Maxima zu wählen.

Gamma-Korrektur

- Überführung einer linear wachsenden Größe (Bildverarbeitung: Intensitätsskala) in eine nicht linear wachsende Größe
→ **nicht lineare Transformationsfunktion**
- **Potenzfunktion** mit einem **Exponenten γ** (*Gamma*) als einzigem Parameter
- Anwendung:
 - Überführung einer physikalisch proportional (d. h. linear) wachsenden Größe in eine dem menschlichen Empfinden gemäß nicht linear wachsende Größe
 - z.B. **Fotoverbesserung** bei schlechter Ausleuchtung

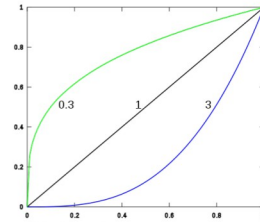


Bild: Gammafrp.svg
Wikimedia, Creative Commons

$$I_{out} = I_{in}^{\gamma}$$

$$0 \leq \{I_{in}, I_{out}\} \leq 1$$

Wie das Farbempfinden, so folgt auch die Helligkeitswahrnehmung beim Menschen nicht einer linearen Skala. So werden geringe Helligkeitsunterschiede zwischen sehr hellen und sehr dunklen Bildpunkten beim Betrachten oft nicht erkannt. Bei über- oder unterbeleuchteten Bildern gehen dadurch leicht viele Details verloren. Eine Verbesserung kann durch die sog. **γ -Korrektur** erreicht werden, indem mit einer Exponentialfunktion (Exponent γ) die Kurve der Helligkeitssteigerung für die Neuberechnung der Bildpunkte im hellen ($\gamma < 1$) oder im dunklen ($\gamma > 1$) Wertebereich abgeflacht wird.

Quelle: Wikipedia <https://de.wikipedia.org/wiki/Gammakorrektur> [18.04.2019]

- **Ungleiche Anpassung der Wertebereiche** in unterschiedlichen Bildregionen
- verschiedene **mathematische Funktionen** mit unterschiedlicher Auswirkung

- Quadratwurzel (1)
- Logarithmus (2)
- Quadratfunktion (3)
- Exponentialfunktion (4)



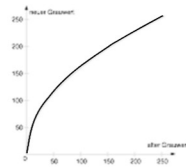
Durch die Wahl einer geeigneten Transferfunktion lässt sich steuern, in welchen Wertebereichen und in welchem Ausmaß eine Spreizung oder eine Stauchung der Intensitäten erfolgt.

Quadratwurzel und Logarithmus betonen Unterschiede in helleren Bildregionen und schwächen diese in dunkleren, Quadrierungs- und Exponentialfunktion haben den gegenteiligen Effekt.

Grauwertkorrektur über Wurzelfunktion

- Dehnung der Grauwertskala in dunklen Bildbereichen

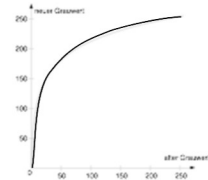
$$g' = \sqrt{g \cdot 255}$$



Logarithmische Grauwertkorrektur

- ähnlich Quadratwurzel, aber stärker
- Skalierungsfaktor a

$$g' = a \cdot \ln(g + 1)$$



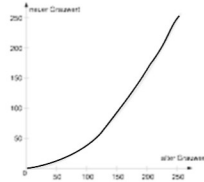
$a = 45,9$

$\ln(g+1)$, weil $\ln(0)$ nicht definiert wäre

$a = 46$ führt (fast) genau zu einem Wert von 255 für g' , wenn $g = 255$

- Dehnung der Grauwertskala in hellen Bildbereichen

$$g' = \frac{g^2}{255}$$

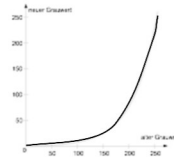


Teiler 255, weil maximaler Wert (in einer 8 bit-Skala)

Exponentielle Grauwertkorrektur

- ähnlich quadratisch, aber stärker
- Skalierungsfaktor a

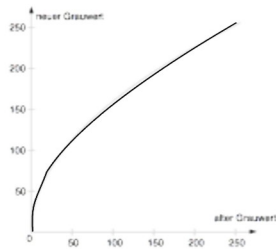
$$g' = e^{g \cdot a} - 1$$



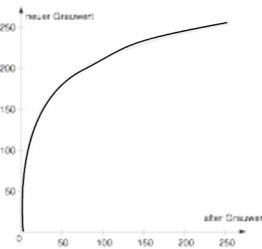
$a = 0,04$

Vergleich der Grauwertkurven

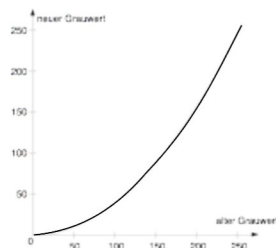
Wurzel



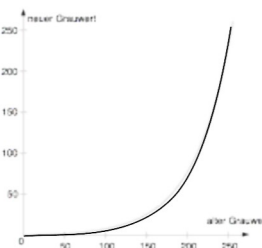
Logarithmus



Quadrat



Exponentialfunktion



Quadratwurzel und Logarithmus ähneln sich in ihrem Effekt – ebenso wie die zugehörigen Umkehrfunktionen Quadrierung und Exponentialfunktion. Nur das Ausmaß des jeweiligen Effekts ist unterschiedlich.

Histogrammskalierung

- **Histogramm** → Verteilung der Grauwerte im Bild
- **wünschenswert**: möglichst **gleichmäßige Verteilung**
 - *ideal*: identische Häufigkeit aller Grauwerte (vgl. (f))

→ Spreizungsfunktion $f(g)$

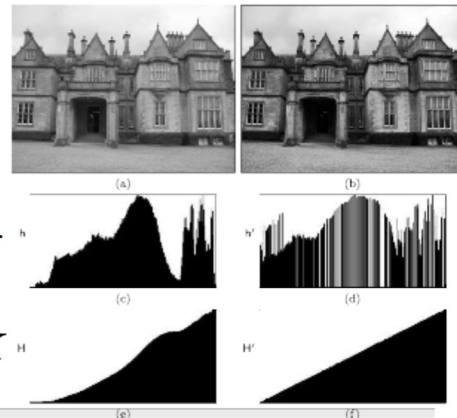
$$f(g) = \left\lfloor \frac{K-1}{U \cdot V} \sum_{i=0}^g h(i) \right\rfloor$$

- **kumulatives Histogramm**

$H(g)$: an jeder Position Anzeige der Summe der Häufigkeiten bis dahin

(vgl. (e))

$$H(g) = \sum_{i=0}^g h(i), \quad 0 \leq g < K$$



Eine Optimierung der Informationscodierung wird erreicht, wenn das verfügbare Spektrum an Grauwerten (= Intensitätswerten) möglichst flächendeckend ausgenutzt wird. Dies wird erreicht, wenn die Häufigkeitsverteilung der Grauwerte im Histogramm möglichst wenige Lücken (und Täler) aufweist.

Eine Spreizungsfunktion (ungleich der gleichmäßigen Grauwertspreizung!) kann die Grauwerte eines Bildes neu berechnen, um eine gleichmäßigere Häufigkeit der Grauwerte zu erreichen.

Ein kumulatives Histogramm (e) zeigt den Anstieg der Grauerhäufigkeiten mit zunehmendem Grauwert. Ein linearer Verlauf (f) spricht dabei für maximale Gleichverteilung.

g = Grauwerte, 0-255

$h(g)$ = Histogrammfunktion = Häufigkeit der einzelnen Grauwerte

$H(g)$ = Summe der Häufigkeiten aller Grauwerte unterhalb eines Werts g → zur Zieldefinition für gleichmäßige Neuaufteilung

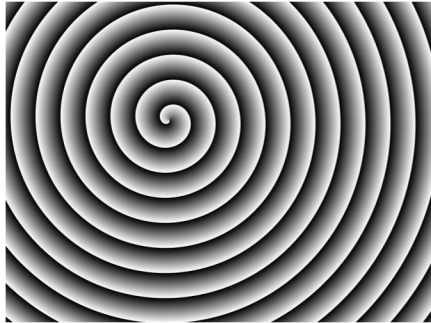
K = Anzahl der möglichen Grauwerte (i.d.R. 256)

U, V = Anzahl der Pixel (horizontal, vertikal)

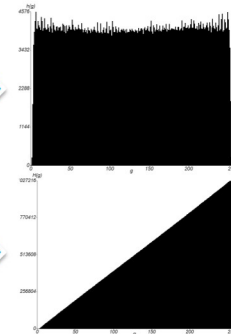
Kumulatives Histogramm

- $H(g)$: an jeder Position Anzeige der Summe der Häufigkeiten bis dahin

$$H(g) = \sum_{i=0}^g h(g), \quad 0 \leq g < K$$



Histogramm



kumul. Histogramm

siehe: http://wwwpub.zih.tu-dresden.de/~photo/teachlets/Histogramme_Punktoperatoren/Bilder (URL s.o.): © 2014 Johannes Pietrzyk, Benjamin Worpitz

[Zugriff: 18.04.2019]

Weitere Beispiele unter: : http://wwwpub.zih.tu-dresden.de/~photo/teachlets/Histogramme_Punktoperatoren/ [© 2014 Johannes Pietrzyk, Benjamin Worpitz , letzter Zugriff 18.04.2019]

g = Grauwerte, 0-255

$h(g)$ = Histogrammfunktion = Häufigkeit der einzelnen Grauwerte

$H(g)$ = Summe der Häufigkeiten aller Grauwerte unterhalb eines Werts $g \rightarrow$ zur Zieldefinition für gleichmäßige Neuaufteilung

K = Anzahl der möglichen Grauwerte (i.d.R. 256)

U, V = Anzahl der Pixel (horizontal, vertikal)

Effiziente Punktoperatoren

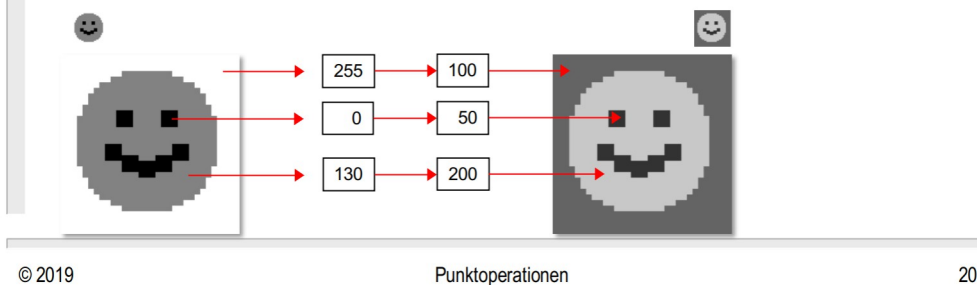
- **Beobachtung: Ein Graustufenbild besitzt weniger Graustufenwerte als Pixel.**

→ mehrmalige Anwendung einer Punktoperation auf einen Graustufenwert

- ineffizient – besonders bei komplexen Berechnungen

- **Lösungsansatz: Look-Up Tables (LUT)**

- Graustufenabbildung wird vorberechnet und in einem Array abgespeichert; $L(a) \leftarrow f(a)$ mit $0 \leq a < K$
- Arraygröße = 256 ; Index = Originalgrauwert
- Berechnung pro Pixel ist dann nur ein Nachsehen in der Look-Up Table:
 $I'(u,v) = L(I(u,v))$



Wir haben Look-up-Tables (LUT) bereits als nützliches Hilfsmittel bei der komprimierten Speicherung von Bildern mit einer geringen Zahl an unterschiedlichen Farb- bzw. Intensitätswerten kennengelernt.

Unter dieser Voraussetzung lassen sich solche Tabellen auch zur Effizienzsteigerung bei der Berechnung von Intensitätstransformationen mit Punktoperatoren nutzen.

Besonders bei großen Bildern mit großen einfarbigen Flächen lassen sich Transformationen signifikant beschleunigen, wenn komplexe Umrechnungen nur selten durchgeführt werden müssen und bei den meisten Bildpunkten dann auf den Ergebniseintrag dieser Berechnung in einer Transfertabelle zurückgegriffen werden kann.

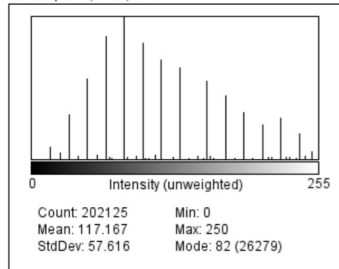
Oder kurz gesagt:

Ziel: schnellere Berechnung des Ergebnisbilds, wenn viele Pixelwerte mehrfach vorkommen; (bzw. wenn wenige Pixelwerte möglichst oft auftreten)

Beispiel Grauwert-Indizierung



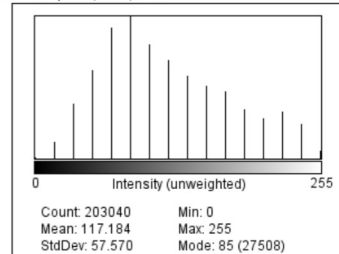
300x240 pixels; RGB; 281K



256 → 16



300x240 pixels; RGB; 281K



Im Beispiel nutzt das Originalbild zwar Grauwerte aus allen Bereichen des 8bit-Spektrums, zeigt im Histogramm aber zwischen den verschiedenen Helligkeitsstufen deutliche Lücken. (Ein gewisses Rauschen mit selten auftretenden Zwischenwerten geht vermutlich auf Artefakte durch Antialiasing bei einer Verkleinerung oder durch DCT-Koeffizienten bei der Speicherung im JPEG-Format zurück.

Die Abbildung des Histogramms auf wenige diskrete Grauwerte verändert nicht nennenswert die Bildqualität.

Mittelwertbildung über Bildreihen

- **Ziel:**

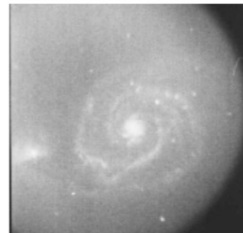
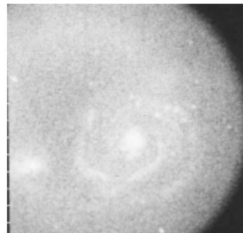
- Verringerung zufälliger Bildstörungen einer Bildreihe
- Hervorheben wesentlicher gemeinsamer Bildinhalte

- **Beachte: Dabei handelt es sich um dasselbe Motiv!!!**

$$I'(u, v) = \frac{1}{n} \sum_{i=1}^n I_n(u, v)$$

n = Anzahl der Einzelbilder

keine Mittelung



Mittelung über 4 Aufnahmen

Wesentliche Informationen werden dadurch oft erst sichtbar.

Verdeutlicht (geschärft) werden Bildeigenschaften, die in den verschiedenen Einzelbildern identisch sind.

u, v = Pixelkoordinaten

$I(u, v)$, $I'(u, v)$ = Intensität vor (I) und nach (I') der Operation

- **Pixelweise Subtraktion zweier Bilder**

$$I'(u, v) = I_1(u, v) - I_2(u, v)$$

- **Anwendungsbereiche:**

- Segmentierung
- Änderungsdetektion
- $I'(u, v) \neq 0$ nur wenn Pixeländerung auftritt
- trotzdem Schwellwert auf Differenzbild anwenden
 - ┆ z.B. wegen Helligkeitsschwankungen

Beispiel: Finde die Unterschiede!

Problematik: Wie geht man mit negativen Differenzen um?

→ z.B. nur Differenzbeträge, Grauwertbereichreduktion (an Ausgangsbilder oder – i.d.R. besser – Differenzwerte) und Offset, Falschfarben, etc.

Differenzbild Cornell Box

● **Cornell Box images – real and virtual**



Source: <http://www.graphics.cornell.edu/online/box/compare.html>

Wie realistisch kann man werden?

... und wozu?

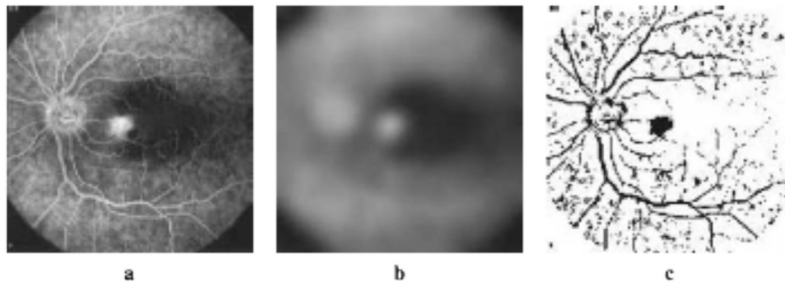
Cornell Univ., Comp. Graphics Dpt.

The original Cornell box, as simulated by Cindy M. Goral, Kenneth E. Torrance, and Donald P. Greenberg for the 1984 paper *Modeling the interaction of Light Between Diffuse Surfaces*, Computer Graphics (SIGGRAPH '84 Proceedings), Vol. 18, No. 3, July 1984, pp. 213-222. Because form factors were computed analytically, no occluding objects were included inside the box.
(<http://www.graphics.cornell.edu/online/box/history.html>)

Segmentierung über Differenzbild

● *Beispiel:* Segmentierung von Blutgefäßen im Augenhintergrund

- Segmentierung über Histogramm nicht möglich
- Subtraktion des Originalbildes (a) von einer mit Unschärfefilter behandelten Kopie (b) liefert (c)



Bildquelle: Ehrhard

Eine häufige Aufgabe der Bildverarbeitung besteht in der Segmentierung von Bildern (d.h. der Aufteilung eines Bildes in einzelne, in sich zusammenhängende Bereiche anhand eines Homogenitätskriteriums).

Im Beispiel besteht das Ziel in der Darstellung der Blutgefäße im Augenhintergrund. Wegen unterschiedlicher Ausleuchtung verschiedener Bildbereiche lässt sich dies nicht mit Histogramm-basierten Operationen (z.B. Schwellwert) erreichen. Bildet man aber die Differenz des Originalbilds mit einer unscharfen Kopie seiner selbst, bleiben im Ergebnis die bei der Weichzeichnung verwischten Feinstrukturen – und damit im Wesentlichen das Aderngeflecht – übrig.

Änderungsdetektion über Differenzbild

- Änderungen zwischen (a) und (b) sind im Differenzbild (c) schwarz dargestellt.



Bildquelle: Efford

Differenzbildung zwischen einzelnen Bildern einer zeitlichen Serie kann Bewegungsvorgänge sichtbar machen, indem unveränderte Bildbereiche ausgeblendet werden.

Der helle Bereich zwischen den beiden Personensilhouetten im Ergebnisbild kommt durch Überlappung des Bildbereichs durch die Person zustande.

- **Pixelweise Vergleiche mehrerer Bilder**
- **z.B. Boole'sche Operatoren**
 - Differenz, Vereinigung, Schnittmenge
- **allg. vergleichende Abfragen**
 - z.B. Pixel hervorheben, die in Bild A rot und in Bild B blau sind
 - praktischer: Pixel löschen, die in einer Bildserie innerhalb einer gewissen Toleranzgrenze unverändert bleiben → Hintergrund entfernen

Außer der Differenzbildung können auch weitere logische Operatoren beim automatisierten Vergleich mehrerer Bilder nützlich sein.

Anwendungsbeispiel Punktoperatoren

● Suche nach Ostereiern (SS'14ff)

Suchen Sie in dem Beispiel-Bildpaar (vorher/nachher) nach versteckten Ostereiern. Erstellen Sie dazu die für geeignete Punktoperationen erforderlichen ImageJPlugins und wenden Sie diese so an, dass die versteckten Eier sichtbar werden. Markieren Sie die gefundenen Eier und dokumentieren Sie Ihre Vorgehensweise. Für jedes (nachvollziehbar) gefundene Ei erhalten Sie einen **Bonuspunkt** für die Klausur.

- FROHE FEIERTAGE! -

- z.B.
- Grauwertskalierung
 - Kontrastverstärkung (Helligkeit, Farbe)
 - Differenzbildung



Kompetenzcheck

- **Bildtransformation**
 - Varianten
 - Prinzipien & Beispiele von Punktoperationen
 - Effizienz (LUT)
- **Wertebereiche**
 - Kontrast & Helligkeit, Skalierungsgrenzen
 - Kumulative Histogramme
- **Binarisierung**
 - Schwellwert
- **Nicht-lineare Grauwertskalierung**
 - Gammakorrektur, Exponentialfunktionen
- **Punktoperationen an Bildserien**
 - Entrauschen, Segmentierung, Dynamik
 - Logische Operatoren

