



DIGITALE BILDBEARBEITUNG

ZPG IMP KLASSE 10

HINTERGRUND

Dieses Werk ist unter einem **Creative Commons 3.0 Deutschland Lizenzvertrag** lizenziert:

- Namensnennung
- Keine kommerzielle Nutzung
- Weitergabe unter gleichen Bedingungen

Um die Lizenz anzusehen, gehen Sie bitte zu <http://creativecommons.org/licenses/by-nc-sa/3.0/de> oder schicken Sie einen Brief an Creative Commons, 171 Second Street, Suite 300, San Francisco, California 94105, USA.

Thomas Schaller – E-Mail: thomas.schaller@zsl-rska.de. – April 2020



Inhaltsverzeichnis

Digitale Bildbearbeitung	3
Was ist digitale Bildbearbeitung?.....	3
Aufbau eines digitalen Bildes.....	3
Farben	3
Farbschemata.....	3
RGB-Codierung:.....	4
HSB-Codierung.....	5
Grauwertbestimmung.....	5
Schwarzweißbilder.....	6
Farbanpassung.....	7
Histogrammanpassung.....	9
Mehrpixeloperationen	12
Faltung.....	12
Weichzeichnen (Lowpass-Filter).....	13
Kanten finden.....	14
Schärfen.....	16
Bildkombinationen	18
Addition.....	18
Subtraktion.....	18
Multiplikation.....	19
Manipulation der Pixelpositionen	20
Nearest Neighbor.....	20
Bilineare Interpolation.....	21
Sonstiges	22
Bit-Ebenen / Steganographie.....	22
Pointillismus.....	23



Digitale Bildbearbeitung

Was ist digitale Bildbearbeitung?

Bildbearbeitung ist die Veränderung von Fotos, Negativen, Dias oder digitalen Bildern. Sie ist zu unterscheiden von der Bildverarbeitung, mit deren Hilfe die Inhalte von Bildern verarbeitet werden. Oft wird die Bildbearbeitung dazu angewandt, um Fehler zu beseitigen, die beim Fotografieren oder anderen Bilderfassungen entstehen können. Dazu gehören beispielsweise Über- und Unterbelichtung, Unschärfe, Kontrastschwäche, Bildrauschen, Rote-Augen-Effekt und stürzende Linien. Durch diese Fehler wirken Bilder oft zu dunkel, zu hell, zu unscharf oder anderweitig mangelhaft. Die Ursachen können technische Probleme oder mindere Qualität der Aufnahmegeräte (Digitalkamera, Objektiv, Scanner), deren Fehlbedienung sowie ungünstige Arbeitsbedingungen oder mangelhafte Vorlagen sein.¹

Aufbau eines digitalen Bildes

Um digitale Bilder herzustellen muss ein analoges Bild diskretisiert werden (vgl. IMP Klasse 9: 3.2.1.1 Daten und Codierung). Die Diskretisierung findet auf zwei Arten statt: zum einen werden die Bilder in Pixel (**P**icture **E**lement) unterteilt. Jedes Pixel hat eine bestimmte Farbe. Zum anderen muss der Farbraum diskretisiert werden. Dafür gibt es eine ganze Reihe von Möglichkeiten, die im Folgenden betrachtet werden.

Digitale Bilder kann man als zweidimensionales Array von Farbinformationen speichern. Durch Manipulation der Farbinformation oder durch Änderung der Positionen im Array können verschiedene Bildbearbeitungsoperationen realisiert werden. Für weitergehende Bildoperationen ist die Umwandlung des Bildes mit Hilfe der Fouriertransformation in Koeffizienten einer Wellenfunktion notwendig. Dies ist nicht Inhalt dieses Unterrichtsganges. Daher wird hier darauf verzichtet, die Repräsentation eines Bildes als Wellenfunktion zu erläutern.

Farben

Farbschemata

Um die Farbe eines Pixels zu beschreiben, gibt es viele verschiedene Möglichkeiten, die je nach Anwendungsbereich zum Einsatz kommen:

- Bildschirme stellen die Pixel als eine Kombination aus drei leuchtenden Farbpunkten in den Farben rot, grün und blau dar. Diese Farben korrelieren mit den von unseren Augen vorhandenen Sinneszellen. Der Mensch verfügt über drei Typen von Zapfen, deren höchste Empfindlichkeit bei Wellenlängen von 445nm (blau), 535nm (grün) und 575nm (rot) liegen. Alle anderen Farben werden additiv² gemischt. Dies führt zur **RGB-Codierung**.
- Bei gedruckten Bildern kommt eine andere Art zum Einsatz, da sich hier die wahrgenommene Farbe durch Absorption des einfallenden Lichtes und damit eine subtraktive Farbmischung³ ergibt. Die Farbe entsteht durch die Kombination von Cyan, Magenta und Gelb - den

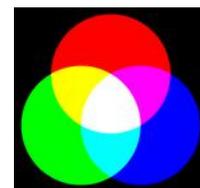


Abb. 1: Additive Farbmischung

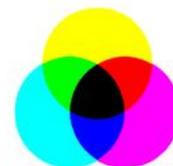


Abb. 2: Subtraktive Farbmischung

¹ Seite „Bildbearbeitung“. In: Wikipedia, Die freie Enzyklopädie. Bearbeitungsstand: 3. April 2020, 09:56 UTC. URL: <https://de.wikipedia.org/w/index.php?title=Bildbearbeitung&oldid=198416232> (Abgerufen: 14. April 2020, 09:02 UTC)

² Abbildung 1: *Additive Farbmischung*, Quark67 (Lizenz: *CC BY-SA 3.0*) via Wikimedia Commons, URL: <https://commons.wikimedia.org/wiki/File:Synthese%2B.svg>.



Komplementärfarben von Rot, Grün und Blau. Da sich auf diese Art nur schlecht Schwarz mischen lässt, wird oft als vierte Druckfarbe noch Schwarz ergänzt. Man erhält eine CMY oder **CMYK-Codierung**.

- Beide Varianten entsprechen aber nicht dem menschlichen Empfinden von Farbe. Es fällt uns nicht leicht, bei einer Farbe den Rot-, Grün- und Blauanteil anzugeben. Wir geben eher den Farbton, die Helligkeit und die Reinheit der Farbe an. Damit ergibt sich das **HSB-Farbmodell** (Hue-Saturation-Brightness).

RGB-Codierung:

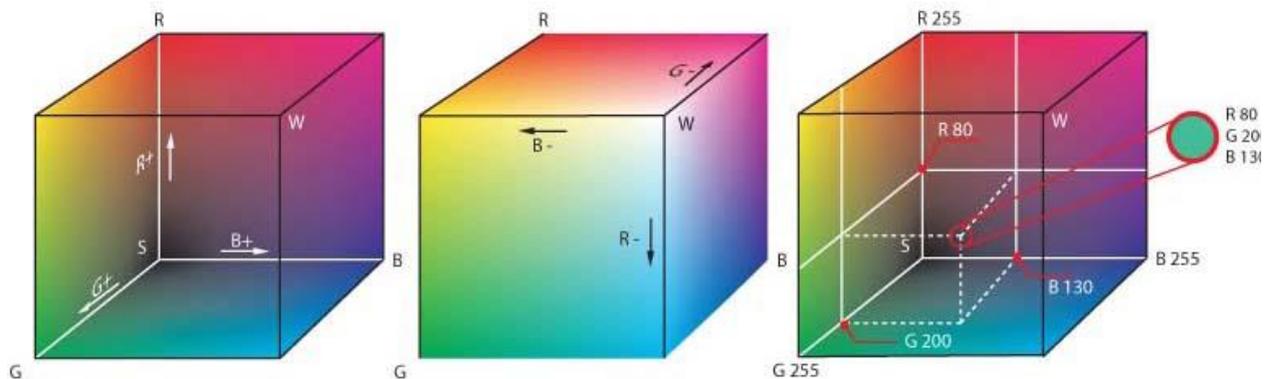


Abb. 3: RGB-Farbwürfel, Horst Frank (Lizenz: CC BY-SA 3.0) via Wikimedia Commons: URL: https://commons.wikimedia.org/w/index.php?title=File:RGB_farbwuerfel.jpg&oldid=193805576.

Die RGB-Codierung ist der Standard für die Color-Klasse von Java und wird auch in den meisten anderen Programmiersprachen eingesetzt. Dabei wird jede Farbkomponente mit einem Wert zwischen 0 und 255 (oder zwischen 0 und 1) angegeben. Haben alle drei Komponenten den Wert 0 erhält man schwarz. Weiß ergibt sich bei 255 für jede Farbkomponente. Trägt man die drei Komponenten in die drei Raumrichtungen auf, kann der Farbraum durch einen Farbwürfel dargestellt werden. Die Diagonale durch den Farbwürfel (d.h. alle drei Komponenten sind gleich) enthält alle Graustufen.

Bemerkenswert ist, dass trotz der vielen darstellbaren Farben nicht alle vom menschlichen Augen wahrnehmbaren Farben (grauer Bereich) dargestellt werden können. Dies bezeichnet man als Gamut-Effekt. Nur die Farben des RGB-Dreiecks können am Monitor dargestellt werden. Andere Medien (z.B. Drucker) haben noch kleinere darstellbare Bereiche.

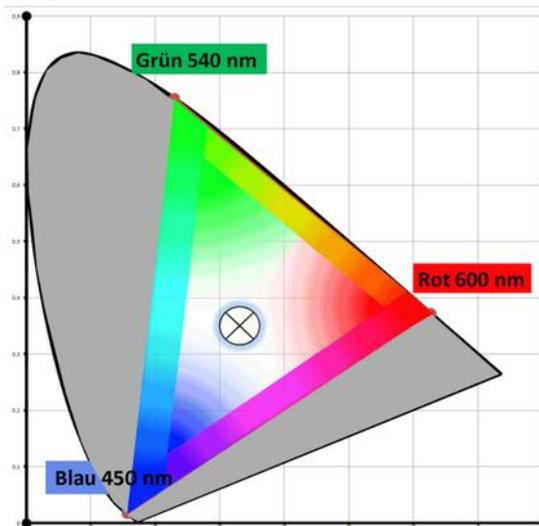


Abb. 4: Gamut-Effekt des RGB-Farbraums, Boonekamp (Lizenz Public Domain) via Wikimedia Commons, URL: <https://commons.wikimedia.org/wiki/File:Gamut-effect.png>

3 Abbildung *Subtraktive Farbmischung*, ZPG Physik (Lizenz: CC BY-NC-SA 3.0) via Lehrerfortbildungsserver, URL: https://lehrerfortbildung-bw.de/st_digital/medienkompetenz/gestaltung-farbe/physik/subtrakt/subinter/



HSB-Codierung

Vom RGB-Modell kommt man zum HSB-Modell, wenn man den RGB-Würfel auf die Seite kippt, so dass Schwarz unten und Weiß oben ist. Die drei Primärfarben (R,G,B) und die drei Sekundärfarben (C,Y,M) befinden sich dann etwa auf halber Höhe. Man vereinfacht das zu einem Doppelkegel. Die Verbindungsstrecke der beiden Spitzen sind die Graustufen.

Die Helligkeit einer Farbe (Brightness) ist nun definiert durch die Höhe im Doppelkegel. Brightness 0 ist schwarz, 255 ist weiß. Der Farbton ergibt sich aus der Richtung, in der man sich von der Graustufenstrecke entfernt. Der Farbton wird (meistens) als Winkel ausgehend von der Rotrichtung angegeben (0° = Rot, 60° = Gelb usw.). Die "Entfernung" von der Mittelachse ist dann die Sättigung einer Farbe. Sättigung 0 bedeutet grau, da man sich auf der Mittelachse befindet, Sättigung 255 bedeutet maximale Sättigung, d.h. man befindet sich auf der Mantelfläche des Doppelkegels.

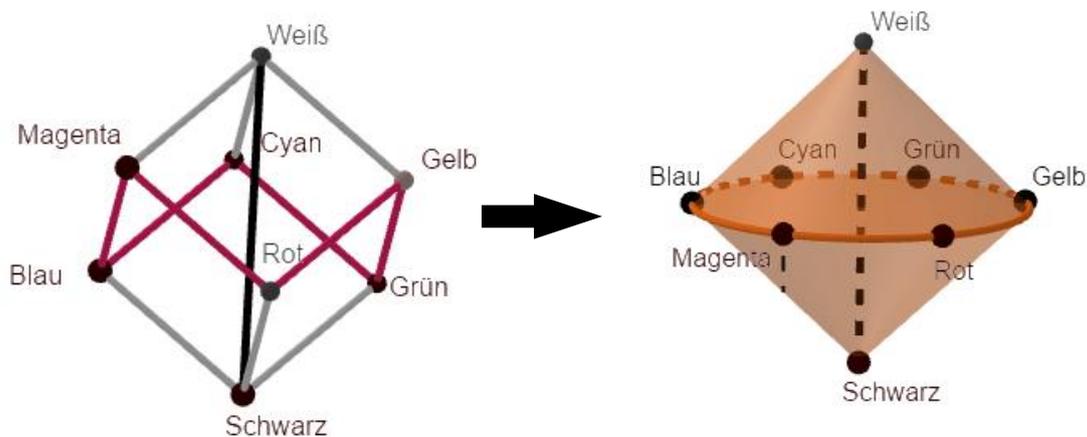


Abb. 5: Doppelkegel des HSB-Farbmodells (eigene Abbildung)

Die Interpretation als Doppelkegel ist allerdings nur eine Möglichkeit, oft wird auch ein einfacher Kegel oder ein Zylinder verwendet. Die genauen Formeln für die Umrechnung⁴ werden hier nicht benötigt, da Java Funktionen zur Umrechnung bereitstellt.

Grauwertbestimmung

Viele der Algorithmen zur digitalen Bildbearbeitung sind leichter an Graustufen-Bildern nachzuvollziehen. Daher sollte es möglich sein, aus einem Farbbild ein Graustufen-Bild zu berechnen.

Verwendet man das HSB-Farbmodell ist dies sehr einfach, da man lediglich den Wert der Sättigung (Saturation) auf 0 setzen muss.

Im RGB-Farbmodell muss man aus den drei Farbkomponenten einen Graustufenwert zwischen 0 und 255 ermitteln. Diesen weist man dann allen Farbkomponenten zu. Um den Graustufenwert zu ermitteln, kann man verschiedene Möglichkeiten verwenden, die zu unterschiedlichen Ergebnissen führen. Je nach Anwendungsbereich können aber alle sinnvoll sein.

⁴ 4 Formeln für die Umrechnung von RGB- in HSB-Farbraum findet man z.B. auf der Seite HSV-Farbraum (URL: <https://de.wikipedia.org/wiki/HSV-Farbraum>), Abrufdatum: 13.11.2019



- Minimum oder Maximum der einzelnen Farbkomponenten.
- Durchschnitt der Farbkomponenten:
Dies wird z.B. für die Umrechnung von RGB- in den HSB-Farbraum verwendet.
- Natürlicher Eindruck:
Unser Auge beurteilt die Helligkeit der einzelnen Farben unterschiedlich, da die Empfindlichkeit der Sinneszellen für die drei Wellenlängen und die Anzahl der Sinneszellen für die einzelnen Farben (z.B. nur 2% erkennen Blau) unterschiedlich ist. Um diesem Rechnung zu tragen, kann man folgende Formel verwenden:

$$\text{Grauwert} = 0,299 \times \text{Rotanteil} + 0,587 \times \text{Grünanteil} + 0,114 \times \text{Blauanteil}$$

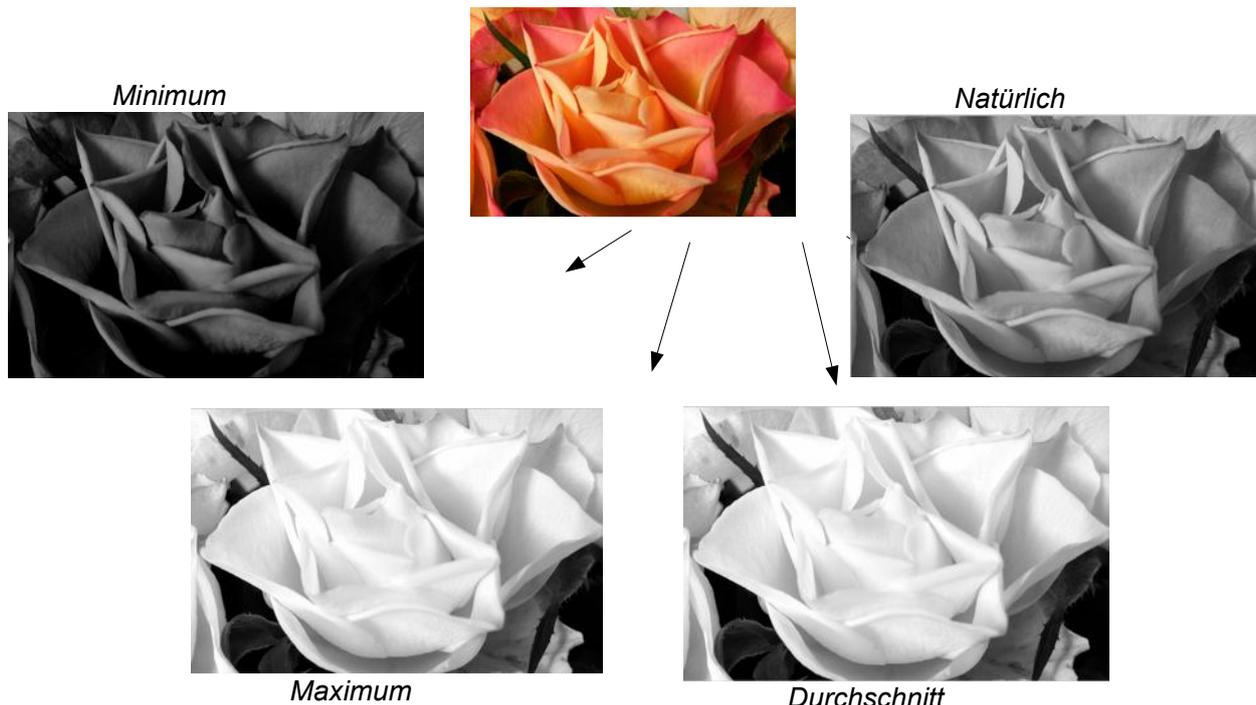


Abb. 6: Verschiedene Umwandlungen in Graustufen (eigenes Werk)

Schwarzweißbilder

Noch extremer als die Umwandlung von Farbbildern in 256 verschiedene Grauwerte ist die Umwandlung in Schwarz-Weiß-Bilder mit nur 2 Farbwerten. Die Weiterverarbeitung von reinen Schwarz-Weiß Bildern ist aber manchmal einfacher. Man könnte zum Beispiel zeigen, in welchen Pixeln sich zwei Bilder unterscheiden.

Für die Umrechnung wird in der Regel der Graustufenwert herangezogen und eine Grenze festgelegt. Alle kleineren Helligkeiten werden zu Schwarz, alle größere Helligkeiten werden zu Weiß.



Differenzbildung und Umwandlung in Schwarz-Weiß



Abb. 7: Differenz der beiden oberen bei Wind aufgenommenen Bilder, man sieht die sich bewegenden Äste (eigenes Werk).

Farbanpassung

Die Umwandlung der Bilder in Graustufen- oder Schwarz-Weiß-Bilder ist eine erste Form der Farbanpassung. In der Regel werden Farbanpassungen aber durchgeführt, um die Qualität der Bilder zu verbessern. In der Schule kann man weitere Farbanpassungen vornehmen, um den Schülerinnen und Schülern den RGB-Farbraum nahezubringen.

Invertieren

Tauscht man jede Farbe gegen ihr Komplement, dann erhält man Negative wie man sie früher von der Analogfotografie kannte.



Abb. 8: Invertieren eines Bildes (eigenes Werk)

Im RGB-Farbraum muss man die Intensität jeder Farbkomponente mit der Formel $I_{\text{neu}} = 255 - I_{\text{alt}}$ umrechnen.

Im HSB Farbraum muss der Farbton (Hue) nach der Formel $H_{\text{neu}} = (H_{\text{alt}} + 180) \bmod 360$, um die



Komplementärfarbe zu bekommen, und die Intensität nach der Formel $I_{\text{neu}} = 255 - I_{\text{alt}}$ angepasst werden.

Tausch von Farbkomponenten

Tauscht man die Intensitäten der einzelnen Farbkomponenten kann man auch schöne Effekte erreichen. Hier wurden z.B. die Rot- und die Grünkomponente vertauscht.

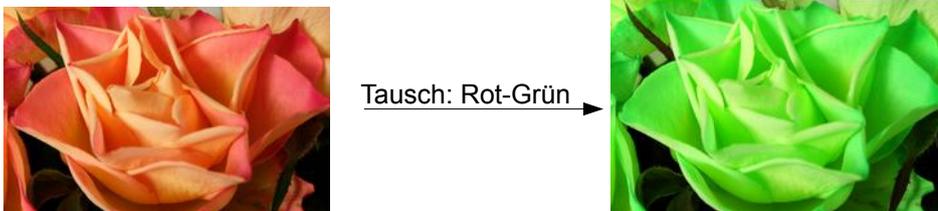


Abb. 9: Tausch von Farbkomponenten (eigenes Werk)

Allgemeine Anpassung

Um eine Verbesserung der Bildqualität im RGB-Farbraum nach optischer Einschätzung vornehmen zu können, können die einzelnen Farbkomponenten mit jeweils mit einem eigenen Faktor multipliziert werden, den der Benutzer wählt. Dies führt zu einer Verstärkung oder Abschwächung einzelner Komponenten.

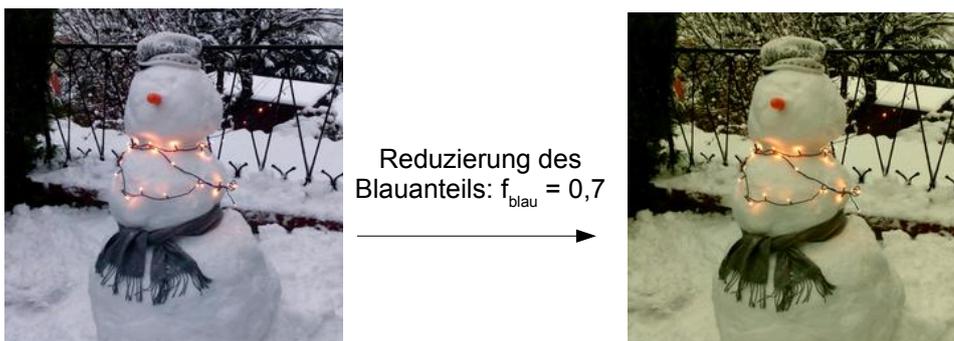


Abb. 10: Multiplikation der Farbkomponenten mit einem Faktor (eigenes Werk)

Im HSB-Farbraum ist eine ähnliche Anpassung möglich. Der Farbton sollte allerdings nicht mit einem Faktor multipliziert werden, sondern man erreicht eine Farbverschiebung, indem man zum H-Wert (=Winkel des Farbtons) die Verschiebung addiert. Aufgrund der Struktur des HSB-Farbraums kann hier sehr leicht eine Entsättigung eines Bildes vorgenommen werden.



Abb. 11: Entsättigung (eigenes Werk)



Histogrammanpassung

Ein gutes Mittel der Bildverbesserung ist die Anpassung der Verteilung der Helligkeitswerte. Am besten ist dies bei Graustufenbildern zu verstehen. Zählt man für jede Graustufe im gesamten Bild wie viele Pixel diese Helligkeit haben, erhält man ein Histogramm. In diesem kann man z.B. gut erkennen, ob ein Bild über- oder unterbelichtet ist.



Viele dunkle Pixel



Verteilung gleichmäßig (bis auf dunkle Ecke)



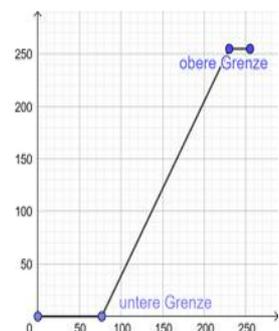
Viele sehr helle Pixel

Abb. 12: Histogramme von verschiedenen belichteten Bildern (eigenes Werk)

Um ein schlecht belichtetes Bild nachträglich zu verbessern, müssen die Helligkeitswerte besser verteilt werden. Bei unterbelichteten Bildern muss der linke Bereich des Histogramms gedehnt und der rechte gestaucht werden. Bei überbelichteten Bildern genau umgekehrt. Dafür gibt es viele verschiedene Möglichkeiten. Hier werden nur drei angesprochen:

Lineare Histogrammanpassung

Die erste Möglichkeit besteht darin, alle Farbwerte bis zu einer unteren Grenze in Schwarz umzuwandeln. Dieser Bereich wird also extrem gestaucht (zu einem einzigen Farbwert). Oberhalb der oberen Grenze wird alles zu Weiß. Dazwischen werden die alten Farbwerte (x-Werte) mit einer abschnittsweise linearen Funktion in neue Farbwerte (y-Werte) umgerechnet.





Diese Umrechnungsfunktion ist also beschrieben durch:

$$f(i_{alt}) = \begin{cases} 0 & \text{für } i_{alt} < \text{untere Grenze} \\ 255 * \frac{(i_{alt} - u)}{(o - u)} & \text{für } \text{untere Grenze} \leq i_{alt} < \text{obere Grenze} \\ 255 & \text{für } i_{alt} \geq \text{obere Grenze} \end{cases}$$

Um ein unterbelichtetes Bild aufzuhellen, wählt man die obere Grenze an der Stelle, an der das Histogramm nur noch wenige Pixel zeigt. Die untere Grenze legt man auf 0 fest.

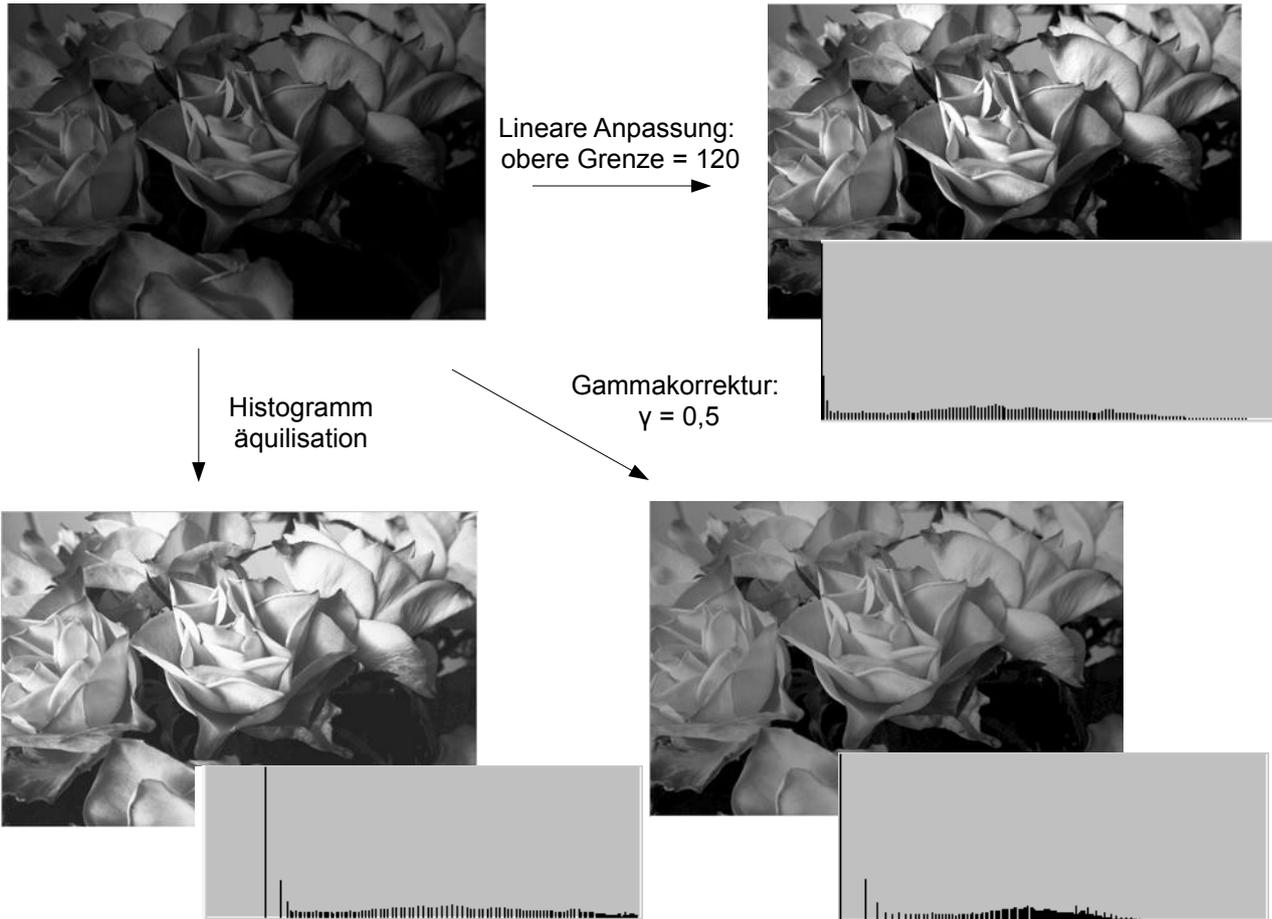


Abbildung 13: Histogrammanpassung eines unterbelichteten Bildes (eigenes Werk)



Gammakorrektur

Die Gammakorrektur ist eine andere Art der Umrechnung von alten Intensitätswerten in neue. Statt einer linearen Funktion wird eine Potenzfunktion verwendet. Durch Variation des Exponenten kann man eine Aufhellung ($\gamma < 1 \Rightarrow$ die neuen Intensitäten gehen schnell in den hellen Bereich) oder eine Abdunklung ($\gamma > 1 \Rightarrow$ die neuen Intensitäten bleiben lange im dunklen Bereich) erreichen.

$$f(i_{alt}) = 255 \cdot \left(\frac{i_{alt}}{255}\right)^\gamma$$

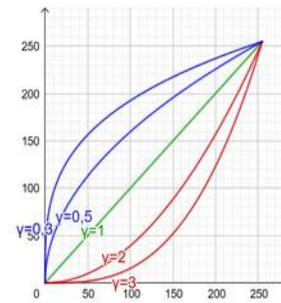
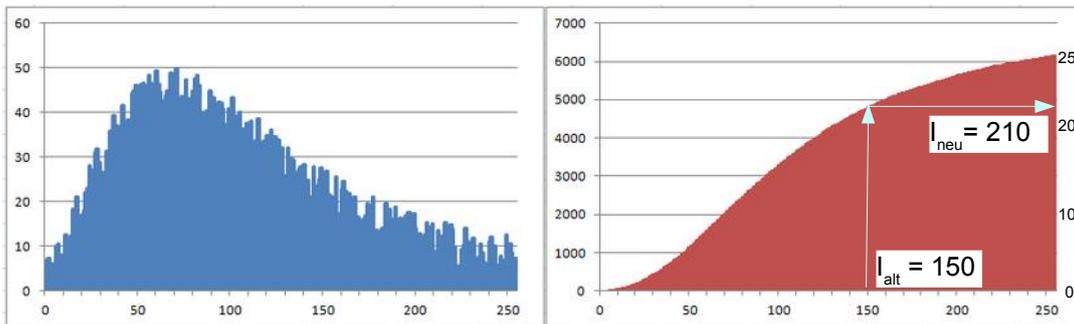


Abbildung 14: Graphen der Umrechnungsfunktionen zu verschiedenen Gammawerten (eigenes Werk)

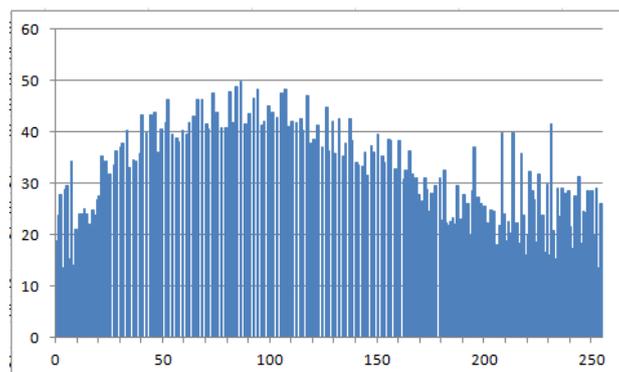
Automatische Histogrammäquilisation

In einem gut belichteten Bild verteilen sich die Helligkeiten relativ gleichmäßig. Dies kann man ausnutzen, um zu versuchen, das Histogramm automatisch zu einer Gleichverteilung zu machen. Dabei ist es nicht möglich, die Häufigkeit einzelner Helligkeitswerte zu reduzieren, da immer alle Pixel der gleichen Helligkeit auch nach der Transformation die gleiche Helligkeit haben. Man kann aber erreichen, dass der gesamte Helligkeitsbereich gleichmäßiger genutzt wird, indem manche Bereiche gedehnt und andere gestaucht werden. Man nutzt dabei das Histogramm, um die Funktion für die Berechnung des neuen Intensitätswertes zu bestimmen.



Zur Berechnung der neuen Intensität für I_{neu} addiert man alle Häufigkeiten bis zu I_{alt} (z.B. 4900 Pixel haben eine Helligkeit kleiner oder gleich $I_{alt}=150$). Diesen Wert teilt man noch durch die Gesamtzahl der Pixel und multipliziert ihn danach mit 255 (rechte Skala). I_{neu} ist im Beispiel also 210. Damit erhält man Werte für I_{neu} zwischen 0 und 255.

Man sieht, dass in diesem Beispiel der Bereich zwischen 30 und 120 gestreckt wird (Steigung der roten Kurve ist groß, das neue Histogramm enthält Lücken) und die ganz dunklen und hellen Bereiche eher gestaucht werden (Werte des neuen Histogramms erhöhen sich). Im Ergebnis ist das neue Histogramm recht gleichmäßig verteilt.





Mehrpixeloperationen

Die bisherigen Bildbearbeitungsalgorithmen verwendeten zur Berechnung der Farbwerte des neuen Pixels immer nur die Farbwerte des entsprechenden Originalpixels. Für manche Operationen ist es aber notwendig, die Umgebung eines Pixels mit einzubeziehen. So können z.B. Kanten nur dann erkannt werden, wenn die Farbe der Nachbarpixel bekannt ist.

Faltung

Die im folgenden besprochenen Algorithmen arbeiten alle auf die gleiche Weise. Die neue Intensität berechnet sich als Linearkombination der umliegenden Pixel:

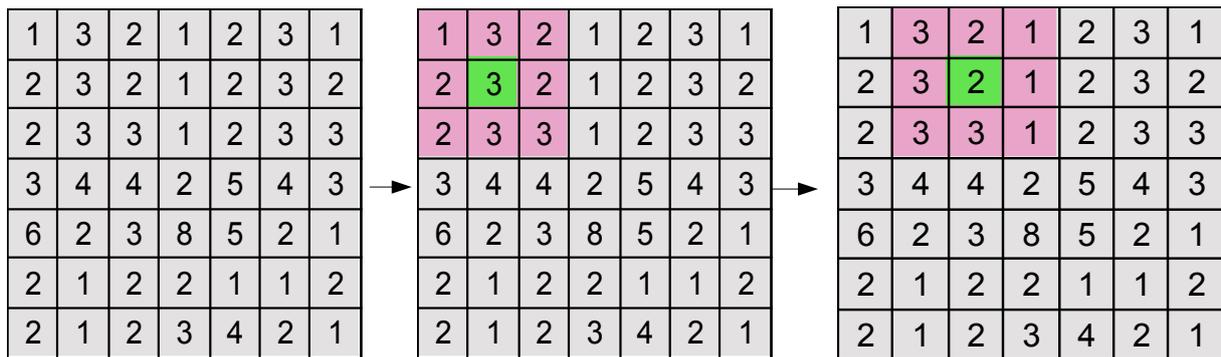
$$I_{neu}(x, y) = \sum_{i=0}^N \sum_{j=0}^N a_{ij} \cdot I_{alt}(x + (i - \frac{N}{2}), y + (j - \frac{N}{2}))$$

-1	-2	-1
0	0	0
1	2	1

Anschaulich bedeutet dies:

Ist $N=2$ werden die 8 umliegenden Pixel mit zur Berechnung herangezogen. Eine Filtermatrix gibt an, wie stark diese gewichtet werden. Diese Filtermatrix wird quasi über die Pixel gelegt und Pixel für Pixel verschoben und angewendet, um die Intensität der neuen Pixel zu berechnen.

Abb. 15: Filtermatrix für Kantenerkennung (Sobelfilter)



Für den Pixel mit $x=1$ und $y=1$ ergibt sich dann z.B.

$$I_{neu}(1,1) = (-1) \cdot I_{alt}(0,0) + (-2) \cdot I_{alt}(1,0) + (-1) \cdot I_{alt}(2,0) + (0) \cdot I_{alt}(0,1) + (0) \cdot I_{alt}(1,1) + (0) \cdot I_{alt}(2,1) + (1) \cdot I_{alt}(0,2) + (2) \cdot I_{alt}(1,2) + (1) \cdot I_{alt}(2,2) = -1 - 6 - 2 + 2 + 6 + 3 = 2$$

Insgesamt ergibt sich damit folgende neue Intensitätsverteilung:

1	3	2	1	2	3	1
2	2	2	1	0	2	2
2	5	6	7	8	6	3
3	2	6	17	12	-1	3
6	-9	-7	-6	-11	-11	1
2	-7	-8	-12	-7	-1	2
2	1	2	3	4	2	1

Die Intensität der Pixel am Rand kann man nicht nach obiger Formel berechnen, da Pixel außerhalb des Bildes herangezogen würden. Um diesem Problem zu begegnen, gibt es unterschiedliche Verfahren: Man kann Pixel außerhalb des Bildes einfach als Schwarz werten, die Farbe der Randpixel nach außen fortsetzen oder die Pixel von innen nach außen spiegeln. Für die Schule sind die Randpixel aber egal, da man sie ohnehin kaum wahrnimmt. Am einfachsten ist es daher, für die Randpixel einfach die Originalfarbe zu übernehmen.



Man erkennt, dass dieser Filter kleine Werte liefert, wenn die vorangegangene und die nachfolgende Zeile kleine Unterschiede haben und große Werte, wenn diese Zeilen sich stark unterscheiden.

Weichzeichnen (Lowpass-Filter)

Beim Weichzeichnen ist das Ziel, starke Schwankungen in der Intensität auszugleichen. Dadurch verschwimmen feine Strukturen. Da langsame Änderungen durchgelassen werden, spricht man von einem "Lowpass"-Filter. Gut sieht man das an einer sogenannten Zoneplate⁵.

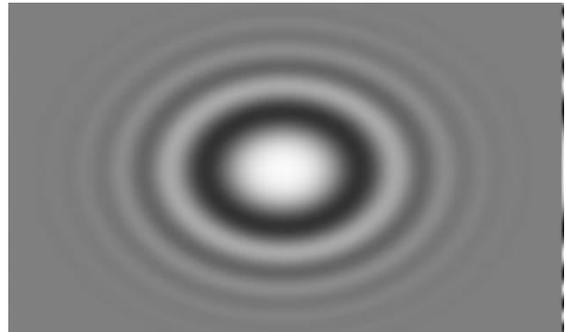
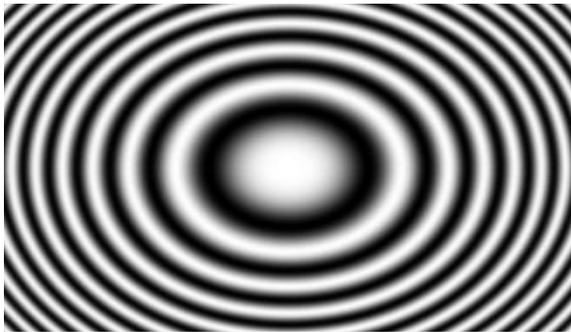
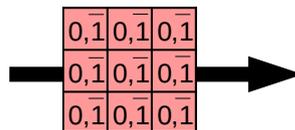


Abbildung 16: Anwendung eines Lowpass-Filters (hier 31x31 Gaußfilter) (eigenes Werk)

Die einfachste Version, einen Lowpass-Filter zu erstellen, ist es, eine Matrix zu verwenden, die alle Pixel gleich gewichtet. Damit sich die Helligkeit des Bildes nicht ändert, muss die Summe aller Gewichte immer Eins ergeben. Die Werte der Matrix hängen also von ihrer Größe ab. Bei einer 3x3 Matrix haben die Einträge alle den Wert $1/9$ ($=0,1$), bei einer 4x4 Matrix den Wert $1/16$. Man sieht, dass die Werte des neu berechneten Bildes deutlich weniger schwanken. Da nur ganzzahlige Intensitäten zulässig sind, sind die Werte hier gerundet.

1	5	3	2	3	2	1	2	2
1	3	2	1	2	3	1	1	1
2	3	2	1	2	3	2	2	2
2	3	3	1	2	3	3	3	3
3	4	4	2	5	4	3	5	5
6	2	3	8	5	2	1	2	2
2	1	2	2	1	1	2	3	3
2	1	2	3	4	2	1	3	3
2	3	2	4	2	2	2	2	2
1	2	7	2	3	2	4	2	2



	2	2	2	2	2	2	2	
	2	2	2	2	2	2	2	
	3	3	2	3	3	3	3	
	3	3	4	4	3	3	3	
	3	3	4	3	3	3	3	
	2	3	3	3	2	2	2	
	2	2	2	2	2	2	2	
	2	3	3	3	2	2	2	

⁵ Abbildung Cosinus-Zoneplate, Georg Wiora (Lizenz: CC-BY-SA 2.5, 2.0) via Wikimedia-Commons, URL: https://commons.wikimedia.org/wiki/File:Zonenplatte_Cosinus.png (17.11.19)



Eine weitere Möglichkeit ist die Verwendung eines „Gaußfilters“. Dieser gewichtet durch die Verwendung einer Gaußverteilung die nahegelegenen Pixel stärker als weiter entfernte Pixel. Die Gewichte berechnen sich nach der Formel:

$$\text{Gewicht}(i, j) = K \cdot e^{-\frac{r^2}{2\sigma^2}}$$

Dabei ist r der Abstand vom Mittelpunkt der Matrix und Sigma (σ) die Standardabweichung der Gaußverteilung⁶ und definiert damit die Größe des Filters. Jenseits vom 3-fachen der Standardabweichung ist das Gewicht so gering, dass es vernachlässigt werden kann. Die Größe der Matrix beträgt daher 6σ . Der Faktor K wird so gewählt, dass die Summe aller Filtergewichte wieder Eins ergibt.

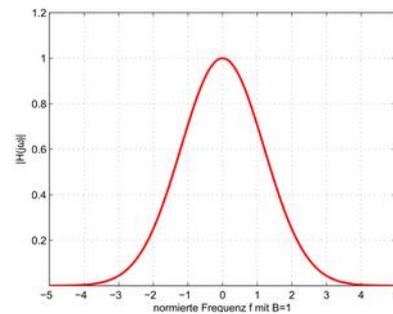


Abb. 17: Gauß-Verteilung

Für Bilder mit höherer Auflösung müssen größere Matrizen gewählt werden, um einen sichtbaren Effekt zu erzeugen.

Kanten finden

Das Gegenstück zu Lowpass-Filtern sind Highpass-Filter. Sie lassen starke Änderungen gut durch und eliminieren langsame Änderungen. Sie können verwendet werden, um Kanten im Bild zu finden. Dies benötigt man für die Erkennung von Objekten in Bildern (z.B. beim autonomen Fahren) oder auch für das Schärfen von Bildern (siehe unten).

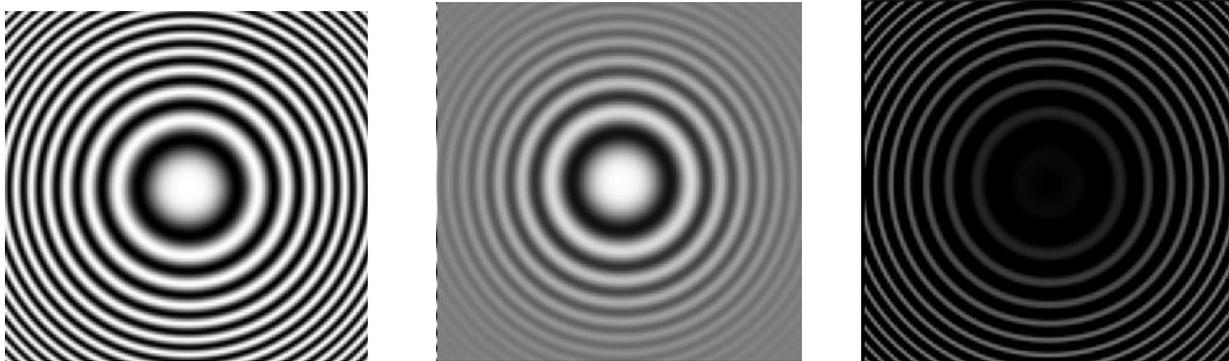
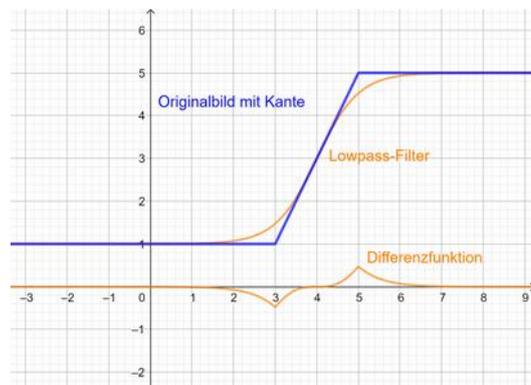


Abbildung 18: Highpass-Filter (rechtes Bild) aus Differenz von Originalbild der Zoneplate und Lowpass-Filter (mittleres Bild) angewendet (eigenes Werk)

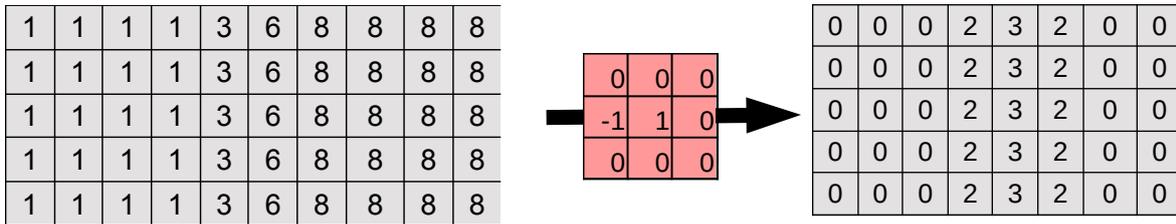
Man kann einen Highpass-Filter erzeugen, indem man zunächst einen Lowpass-Filter anwendet und das Ergebnis vom Originalbild subtrahiert. Dadurch verschwinden die Strukturen, die beim Lowpass-Filter erhalten geblieben waren, und die vom Lowpass-Filter geglätteten Strukturen erscheinen wieder. Addiert man diese Differenzfunktion zum Originalbild werden Kanten überzeichnet. Dadurch erreicht man eine Schärfung des Bildes.



⁶ Abbildung: Gaußverteilung, Chris828 (Lizenz: Public Domain) via Wikimedia.org, URL: <https://commons.wikimedia.org/wiki/File:Gauss-Filter-Uebertragungsfunktion.svg> (17.11.2019)



Eine andere Möglichkeit Kanten zu entdecken, ist die Anwendung einer Faltung mit einer geeigneten Matrix.



Diese Matrix berechnet immer die Differenz zweier nebeneinanderliegender Pixel. Sie führt dazu, dass 0 herauskommt, wenn sich die Intensitäten nicht ändern, positive Werte, wenn die Intensität zunimmt, und negative Werte, wenn die Intensität abnimmt. Im Grunde genommen, berechnet der Filter die erste Ableitung der Intensitätsfunktion. Möchte man die neu errechneten Intensitäten direkt als Bild verwenden, dann muss man sich überlegen, wie man mit den negativen Werten umgeht. Normalerweise werden Werte außerhalb des zulässigen Bereichs 0-255 einfach abgeschnitten. Damit gehen die Information über einen Intensitätsabfall verloren.

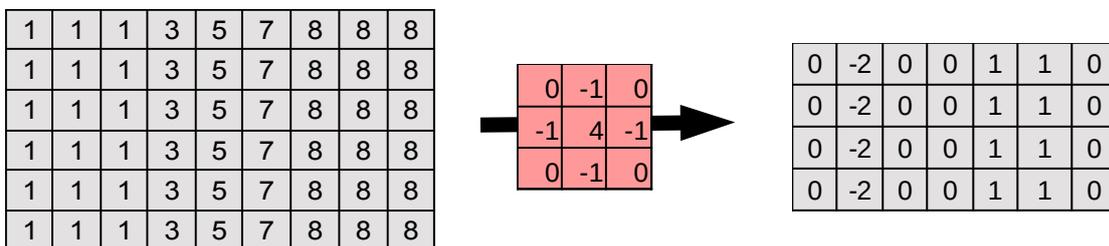
Dieser Filter findet nur senkrechte Kanten im Bild. Um auch die anderen Raumrichtungen erkennen zu können, muss man analog auch eine senkrechte Komponente einführen.

Alternativ zu diesem ganz einfachen Filter werden Sobel-Filter für die Erkennung von senkrechten und waagerechten Kanten eingesetzt. Sie vergleichen zwei Intensitätswerte von Vorgänger- mit Nachfolgezeile, wobei nicht nur der direkt darüber und darunterliegende Pixel berücksichtigt wird, sondern auch deren beide Nachbarpixel mit geringerer Gewichtung.



Abb. 19: Sobel-Filter für waagerechte und senkrechte Kanten

Ergänzt man den Filter um eine -1 nach rechts und nach unten, führt das dazu, dass nicht nur 0 herauskommt, wenn die Intensität sich nicht ändert, sondern auch 0 herauskommt, wenn die Intensität gleichmäßig ansteigt oder abfällt. Nur Änderungen in der Steigung werden erkannt. Dieser Filter berechnet also so etwas wie die zweite Ableitung der Intensitätsfunktion.



Der Filter erkennt also exakt den Rand der Änderung. Er entspricht der Differenzfunktion von Originalbild und Lowpass-Filter-Bild.

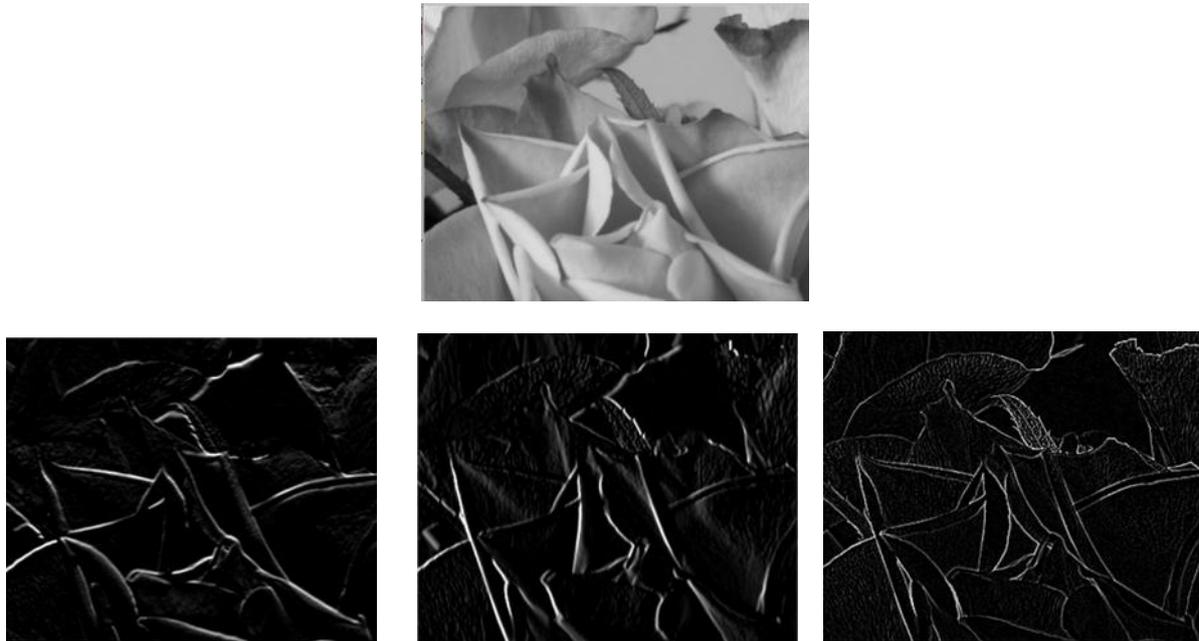


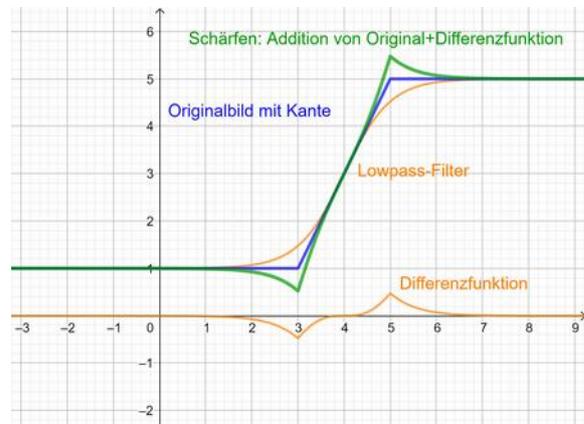
Abbildung 20: waagerechter und senkrechter Sobelfilter, Filter mit der 2. Ableitung (mit anschließender Anpassung des Kontrastes) (eigenes Werk)

Schärfen

Beim Schärfen eines Bildes müssen die Kanten im Bild verstärkt werden. Man nutzt also die beim Kantenfinden vorgestellten Varianten und addiert sie zum Originalbild dazu.

Durch Addition der Differenzfunktion zum Originalbild erreicht man eine Überzeichnung der Ränder eines Intensitätsanstiegs und damit eine Schärfung des Bildes.

Gleiches kann durch die Anwendung einer Filtermatrix erreicht werden:



$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Abbildung 21: Filtermatrix zum Schärfen aus der Addition eines Kantenfinde-Filters + Originalbild



Abb. 22: Schärfen eines Bildes mit Filtermatrix (eigenes Werk)

Addiert man das Originalbild zu einem „schrägen Sobelfilter“, erreicht man auch eine Verstärkung der Kanten, die optisch einem Relief gleicht.

-2	-1	0
-1	0	1
0	1	2

0	0	0
0	1	0
0	0	0

-2	-1	0
-1	1	1
0	1	2

Abb. 23: Filtermatrix zur Reliefbildung aus der Addition eines Sobel-Filters + Originalbild



Abb. 24: Relief (eigenes Werk)



Bildkombinationen

Im vorangehenden Text ist schon einige Male von der Addition oder Subtraktion von Bildern die Rede gewesen. Diese Operationen werden vor allem in Kombination mit anderen Algorithmen eingesetzt.

Addition

Bei der Addition zweier Bilder werden die Intensitäten der Pixel addiert. Da jede Farbkomponente einen Wert zwischen 0 und 255 haben kann, kommen am Ende Werte zwischen 0 und 510 heraus. Alle Werte über 255 müssen auf 255 abgeschnitten werden.

Bei der Addition zweier normaler Bilder kommt aufgrund diesen Effekts ein Bild heraus, das insgesamt zu hell ist und sehr viele weiße Pixel hat. Denkbar wäre es, die Helligkeit der Bilder zunächst zu reduzieren.



↓ Addition ↓



Oft werden aber nur Bilder geringer Helligkeit (z.B. aus der Kantendetektion) zu normalen Bildern addiert. Dort besteht dieses Problem dann nicht.

Subtraktion

Bei der Subtraktion besteht das umgekehrte Problem. Die Differenz der Intensitäten kann negativ werden. Auch hier werden negative Werte zu Null, um im Definitionsbereich zu bleiben. Dies kann aber z.B. beim Schärfen ein Problem werden, da hier auch die negativen Werte bei der Differenzbildung aus Originalbild und Lowpass-Filter entscheidend sind. Nun muss man entweder vorübergehend negative Werte zulassen (und kann das dann nicht als Bild darstellen) oder man berechnet die Schärfung als:



Geschärftes Bild = Original
 + (Original – Lowpass-Bild)
 – (Lowpass-Bild – Original)

Eine andere mögliche Anwendung der Subtraktion ist die Erkennung von Unterschieden in Bildern. Das kann z.B. bei der Erkennung von bewegten Objekten (siehe Beispiel bei Schwarz-Weiß-Bildern) in Bildfolgen (Filmen) verwendet werden. Zum Beispiel müssen beim autonomen Fahren vor allem bewegte Objekte beachtet werden.

Multiplikation

Bei der Multiplikation von zwei Bildern muss auch wieder der Definitionsbereich berücksichtigt werden. Durch die Multiplikation entstehen Werte zwischen 0 und $255^2 = 65025$. Sinnvoller wäre hier eine Normierung der Intensitätswerte vor der Multiplikation auf einen Wertebereich von 0 bis 1. Dadurch bleibt der Wertebereich auch nach der Multiplikation erhalten. Danach wird der Bereich wieder auf 0-255 erweitert.

$$I_{neu} = \frac{I_{Bild1}}{255} \cdot \frac{I_{Bild2}}{255} \cdot 255 = I_{Bild1} \cdot I_{Bild2} / 255$$

Man kann die Multiplikation beispielsweise nutzen, um eine Maske auf das Bild anzuwenden. Multipliziert man ein Bild mit einem anderen, das einen Farbverlauf von schwarz nach weiß hat, erhält man einen Verlauf von dunkel nach hell im Originalbild.

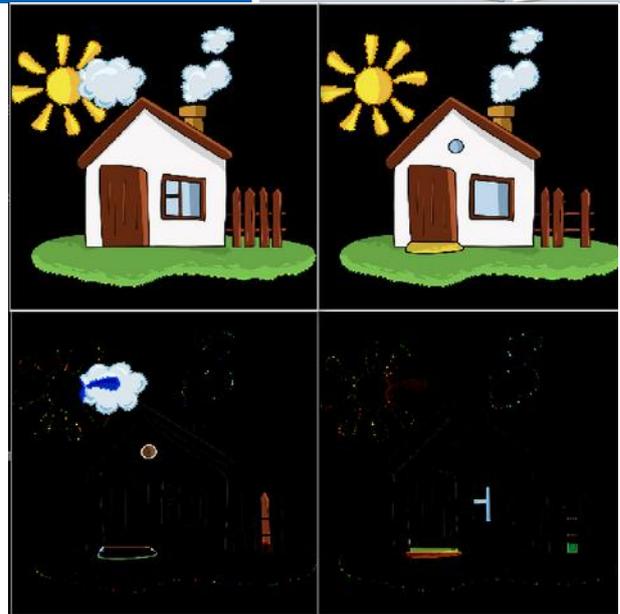


Abb. 25: Differenzbildung bei „Finde den Unterschied“, creozavr (Pixabay Lizenz) via Pixabay, URL: <https://pixabay.com/de/illustrations/haus-häuser-finde-den-unterschied-3208132/> (17.11.19)

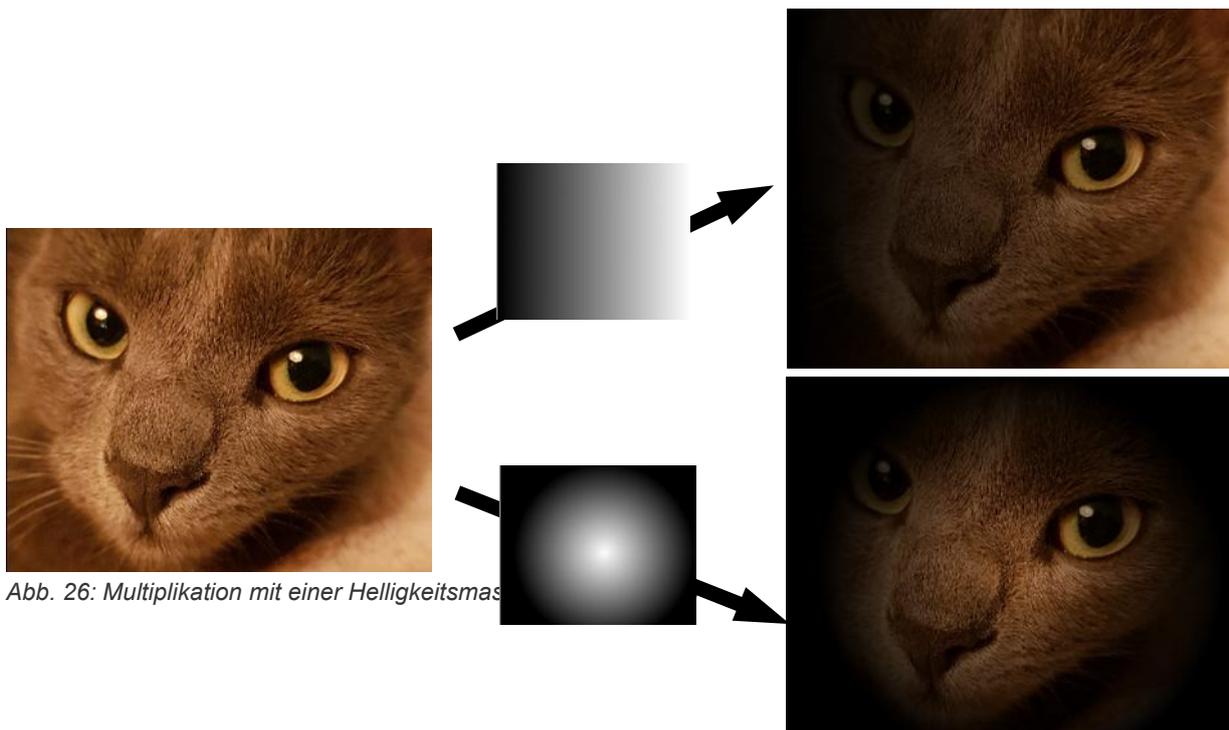


Abb. 26: Multiplikation mit einer Helligkeitsmaske



Manipulation der Pixelpositionen

Bei allen bisherigen Operationen sind die Pixel an ihren Positionen verblieben. Viele Operationen (Spiegeln, Drehen, Größe ändern, Verzerrern usw) verändern aber die Position der Pixel. Dabei sind zwei Vorgehensweisen denkbar:

1. Man durchläuft jedes Pixel des neuen Bildes und berechnet dessen Farbe aus den alten Pixeln. Im einfachsten Fall kann man die Farbe eines bestimmten alten Pixels übernehmen.
2. Man durchläuft jedes Pixel des alten Bildes und überlegt, wo das Pixel im neuen Bild zu finden ist.

z.B. horizontales Spiegeln:

Variante 1: $\text{pixelNeu}[x][y] = \text{pixel}[(\text{breite}-1)-x][y];$

Variante 2: $\text{pixelNeu}[(\text{breite}-1)-x][y] = \text{pixel}[x][y];$

Variante 2 passt besser zu der Vorstellung, dass die alten Pixel auf neue Positionen verschoben werden. Dies setzt aber voraus, dass jedes Pixel auf genau eine neue Position abgebildet wird. Das ist bei der Spiegelung der Fall, aber im Allgemeinen nicht. Bei einer Vergrößerung oder Verkleinerung des Bildes oder auch beim Drehen eines Bildes gilt dies nicht. Dort muss für jedes Pixel entschieden werden, wie die Farbe ermittelt wird. Das kann z.B. mit nearest-Neighbor-Verfahren oder mit bilinearer Interpolation geschehen.

Relativ leicht kann man sich das vorstellen, wenn die Breite und Höhe halbiert werden. Bei der Halbierung werden je 2x2 Pixel zu einem neuen Pixel. Dieses Pixel kann nun entweder die Farbe des linken oberen Pixels bekommen oder den Durchschnitt der vier Pixel. Wird nur ein Pixel verwendet, nimmt man in der Regel das, das dem neuen Pixel bei der Skalierung am nächsten liegt. Dieses Verfahren nennt sich „nächster Nachbar“-Verfahren (nearest neighbor). Der Durchschnitt der vier Pixel ergibt sich, wenn man eine bilineare Interpolation wählt, da das neue Pixel genau in der Mitte der vier alten Pixel liegt.

Nearest Neighbor

Bei diesem Verfahren muss für jedes Pixel des neuen Bildes berechnet werden, an welcher Stelle dieses Pixel im alten Bild gelegen hätte:

$$x_{alt} = \frac{x_{neu}}{\text{breite}_{neu} - 1} \cdot (\text{breite}_{alt} - 1)$$

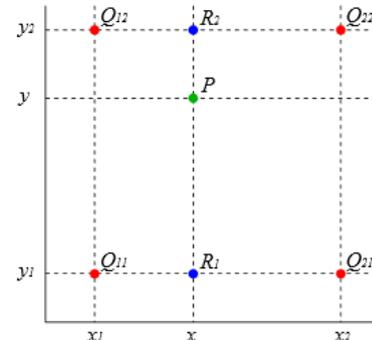
$$y_{alt} = \frac{y_{neu}}{\text{höhe}_{neu} - 1} \cdot (\text{höhe}_{alt} - 1)$$

Diese Werte werden auf ganze Zahlen gerundet, um den nächsten Nachbarn zu finden. Dieses Verfahren hat zwei Nachteile: Zum einen bekommen schräg verlaufende Linien durch die Rundungen „Knicke“. Zum anderen wird das Bild bei Vergrößerung pixelig, da z.B. bei einem Vergrößerungsfaktor von 2 ein 2x2 Feld die Farbe eines einzigen Pixels des alten Bildes bekommt.



Bilineare Interpolation

Zunächst wird wie beim nearest neighbor-Verfahren die Position des neuen Pixels (P) im alten Bild berechnet. Diese Koordinaten fallen in der Regel nicht genau auf ein Pixel des alten Bildes (Q_{11} bis Q_{22}). Um die Farbe des neuen Pixels zu ermitteln, werden nun die vier umliegenden Pixel des alten Bildes in richtigen Gewichtungen herangezogen. Dabei werden zunächst die Farben von den Punkten R_1 und R_2 durch lineare Interpolation zwischen den Pixels Q_{11} und Q_{21} bzw. Q_{12} und Q_{22} und dann die Farbe von P durch lineare Interpolation zwischen R_1 und R_2 bestimmt.



Wenn x und y Werte zwischen 0 (genau bei x_1 bzw. y_1) und 1 (genau bei x_2 bzw. y_2) sind, dann ergibt sich folgende Formel:

Abb. 27: Bilineare Interpolation (Lizenz: Public domain)

$$I_P = I_{Q_{11}} \cdot (1-x)(1-y) + I_{Q_{12}} \cdot (1-x)(y) + I_{Q_{21}} \cdot (x)(1-y) + I_{Q_{22}} \cdot (x)(y)$$

Liegt P genau in der Mitte der vier Punkte, dann werden alle vier umliegende Punkte mit dem Faktor $\frac{1}{4}$ gewichtet.

Die Ergebnisse der bilinearen Interpolation sind erheblich besser als die des nearest neighbor-Verfahrens. Noch besser ist die bikubische Interpolation, die versucht, die Farbe von P mit noch mehr umliegenden Punkten anzunähern. Hier kommen kubische Splines zum Einsatz (vgl. IMP Mathematik Klasse 9).



Sonstiges

Bit-Ebenen / Steganographie

Einen weiteren interessanten Aspekt kann untersuchen, wenn man die Intensitäten als 8-Bit-Binärzahlen auffasst. Man kann das (Graustufen)-Bild dann in acht verschiedene Bit-Ebenen aufteilen. Möchte man z.B. die vierte Bitebene ($2^3 = 8$) haben, setzt man alle anderen Bits auf 0. Dies erreicht man am leichtesten mit einer Und-Verknüpfung:

$$I_{\text{neu}} = I_{\text{alt}} \text{ UND}_{\text{bitweise}} 8$$

	Dezimal	Binär							
I_{alt}	190	1	0	1	1	1	1	0	1
Bitebene 4	8	0	0	0	0	1	0	0	0
I_{neu}	8	0	0	0	0	1	0	0	0

Die niederwertigen Bitebenen sehen auf den ersten Blick komplett schwarz aus, da die maximale Helligkeit nur 1 (1. Bit-Ebene), 2 (2. Bit-Ebene) oder 4 (3. Bit-Ebene) beträgt. Daher muss man die Bilder in Schwarz-Weiß-Bilder umwandeln, damit man etwas erkennt.

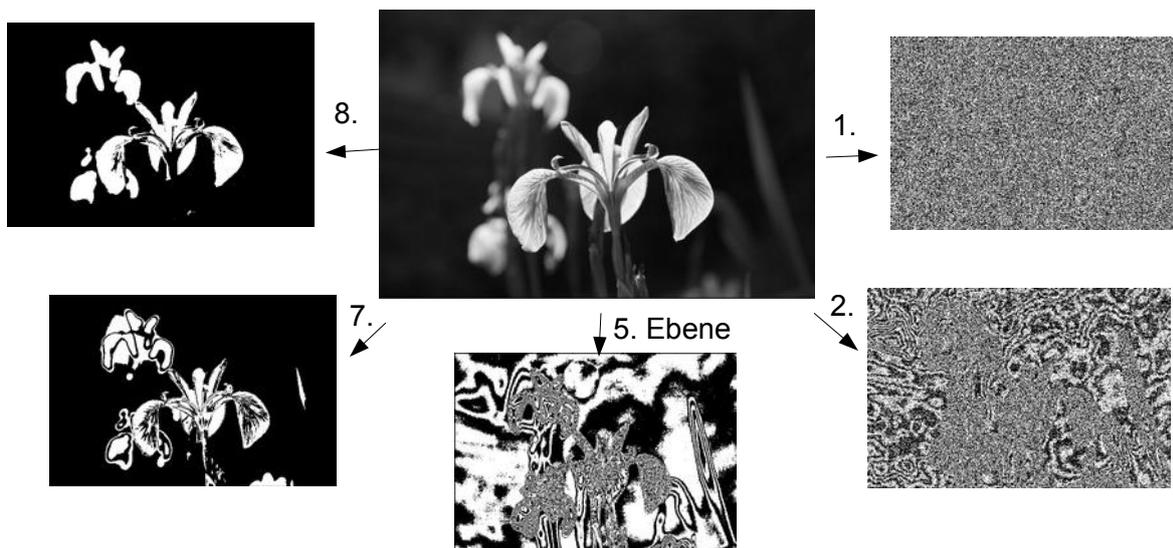
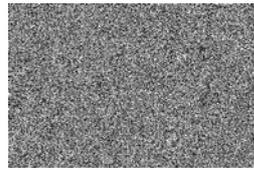


Abb. 28: Bitebenen eines Graustufenbildes (eigenes Werk)

Man erkennt, dass die höherwertigen Bitebenen bestimmen, ob eine Region des Bildes hell oder dunkel ist. Die niederwertigen Bitebenen beeinflussen nur die feinen Strukturen. Dabei trägt die 1. Bit-Ebene so wenig bei, dass ihr Einfluss mit dem bloßen Auge nicht wahrnehmbar ist (sie sieht oben ohnehin nur nach Bildrauschen aus).

Dies kann man ausnutzen, um in dem Bild eine geheime Botschaft zu verstecken, ohne dass auffällt, dass das Bild überhaupt eine Nachricht enthält. Dies bezeichnet man als **Steganographie**. Man kann zum Beispiel die 1. Bit-Ebene eines Bildes gegen ein anderes Schwarz-Weiß-Bild austauschen, das die geheime Nachricht enthält. Extrahiert man aus dem veränderten Bild die 1. Bit-Ebene, erhält man die Nachricht zurück.

Steganographie darf allerdings nicht mit Verschlüsselung verwechselt werden, da hier kein Schlüssel verwendet wird. Jeder, der das Verfahren kennt, kann die Nachricht wieder extrahieren. Man kann dieses Verfahren z.B. aber verwenden, um ein Wasserzeichen in ein Bild einzubauen.



Subtraktion



Addition



Extraktion
der 1. Bitebene



Pointillismus

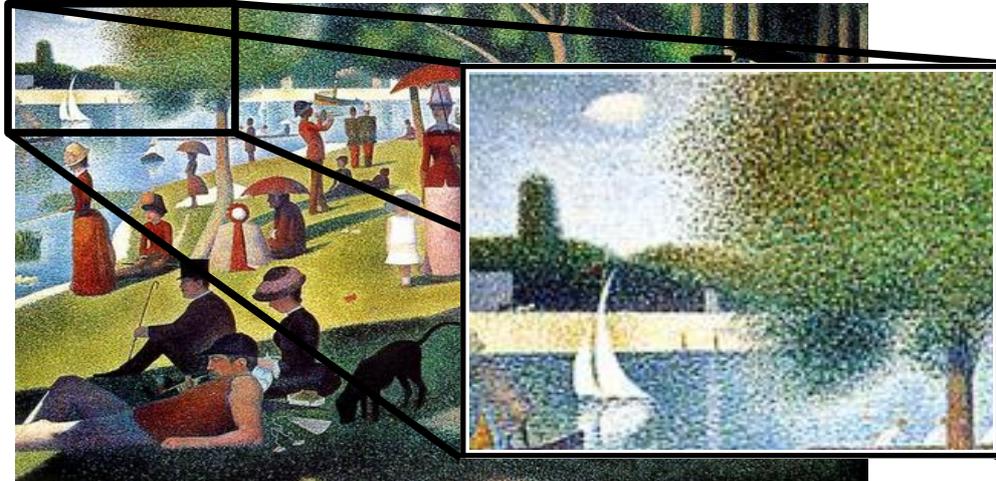


Abb. 29: Georges Seurat - Un dimanche après-midi à l'île de la Grande Jatte
Wikimedia Commons (Lizenz: Public Domain), URL:
https://commons.wikimedia.org/wiki/File:Georges_Seurat_-_Un_dimanche_apr%C3%A8s-midi_%C3%A0_l%27%C3%8ELe_de_la_Grande_Jatte.jpg

Pointillismus bezeichnet eine Stilrichtung in der Malerei. Sie hatte in den Jahren zwischen 1889 und 1910 ihre Blütezeit. Der Pointillismus wird dem Post-Impressionismus zugeordnet.

Typisch für den Pointillismus ist der streng geometrisch durchkomponierte, oft ornamental wirkende Bildaufbau. Die Farben werden dabei in kleinen Punkten auf die Leinwand gesetzt. Dabei nutzten pointillistische Maler nur reine Farben. Farbmischungen entstehen erst auf dem

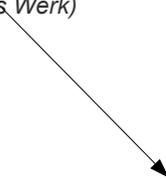


Malgrund selbst.

Man kann einen ähnlichen Effekt erreichen, wenn man in einem Bild zufällige Pixel auswählt und dann ausgefüllte Kreise einer bestimmten Größe an diese Stelle zeichnet.



Abb. 30: Tupfen und Stricheln (eigenes Werk)



Gleiches kann man statt mit Punkten auch mit kurzen schrägen Strichen machen. Auch hier erhält man schöne, künstlerische Effekte.

