



Histogramme

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 => unterbelichtet



Verteilung gleichmäßig (bis auf dunkle Ecke)



Viele sehr helle Pixel => überbelichtet

Abb. 1: Histogramme von verschiedenen belichteten Bildern (eigenes Werk)

Aufgabe

1. Öffne das Dokument *histogramme.ods*. Stelle den Zoom auf 100%, um die Zahlen gut lesen zu können. Stelle den Zoom auf 25% um den Bildausschnitt und das Histogramm besser sehen zu können. Beschreibe, was in der Spalte "Häufigkeit Original" gespeichert ist. Schaue dir dazu auch die Formel in den Zellen an. Beschreibe, wie die Werte bei einem unterbelichteten und wie bei einem überbelichteten Bild verteilt sind.
2. Erstelle eine Klasse "Histogramme". Dekлариere dort eine Methode `public int[] haeufigkeitsverteilung(Picture Originalbild)`, die ermittelt wie oft welcher Helligkeitswert im Bild vorkommt. Erzeuge dazu zunächst ein `int`-Array mit 256 Einträgen. Setze alle auf 0. Durchlaufe alle Pixel und berechne deren Helligkeit. Verwende für die Berechnung der Helligkeit eines Pixels eine der Varianten, die bei der Berechnung der Graustufenbilder zum Einsatz kamen. Erhöhe den entsprechenden Array-Eintrag um 1. Gib am Ende das Array zurück.

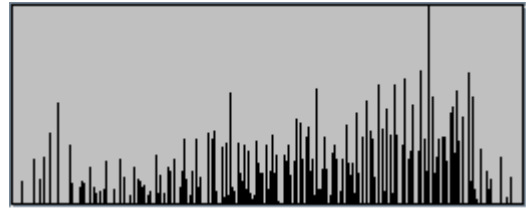


3. Deklareiere eine Methode `public Picture zeichneHistogramm(int[]`



`haeufigkeiten)`, die aus einer Häufigkeitsverteilung ein Histogramm, d.h. ein Bild mit einem Säulendiagramm wie in Calc, der Größe 256x100 erstellt.

Bestimme dazu zunächst das Maximum des Arrays. Erstelle dann ein Bild (Picture) der Größe 256x100 mit grauem Hintergrund. Zeichne darauf für jeden Array-Wert eine Linie. Die Länge der Linie gibt die Größe des Array-Eintrags an. Die Linie des größten Eintrags soll 100 Pixel lang sein. Verwende das ermittelte Maximum, um die Werte so zu normieren.



Teste deine Methode, indem du mit der Methode `haeufigkeitsverteilung` ein Bild untersuchst und das zurückgegebene Array in der Objektleiste speicherst. Lasse dann damit ein Histogramm zeichnen. Vergleiche die Histogramme von einem über- und einem unterbelichteten Bild.

Bildverbesserung durch Histogrammanpassung

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 recht gestaucht werden. Bei überbelichteten Bildern genau umgekehrt. Dafür gibt es viele verschiedene Möglichkeiten.

Aufgaben

4. Öffne das Dokument `histogramme.ods`. In der Spalte **neuer Wert** kann jeder alten Helligkeit (Intensität) ein neuer Wert zugewiesen werden. Versuche durch Verändern der Werte die Verteilung der Intensitäten zu verbessern. Teste auch, was passiert, wenn man dem Wert 0 den neuen Wert 255, der 1 den neuen Wert 254 usw. zuweist. Nutze dabei das Autoausfüllen von Calc (nicht abspeichern!!).

5. Implementiere eine Methode `public Picture histogrammAnpassen(Picture originalbild, int[] zuordnung)`, die für jedes Pixel jeder Farbkomponente mit Hilfe des Array `zuordnung` neue Werte zuordnet. Das Array `zuordnung` entspricht dabei der Spalte "neue Werte" des Calc-Dokuments.

Begründe, warum folgende Zeilen für die Rotkomponente das Gewünschte bewirken:

```
int rotalt = pixel[x][y].getRed();  
int rotneu = zuordnung[rotalt];
```

ERWEITERUNG: HISTOGRAMME

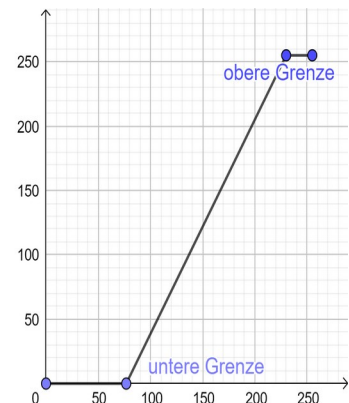


Lineare Histogrammanpassung:

Ein Möglichkeit, die Helligkeitsverteilung zu verbessern, besteht darin, alle Farbwerte bis zu einer unteren Grenze in Schwarz umzuwandeln. Oberhalb einer oberen Grenze wird alles zu Weiß. Dazwischen werden die alten Farbwerte (x-Werte) mit einer linearen Funktion in neue Farbwerte (y-Werte) umgerechnet.

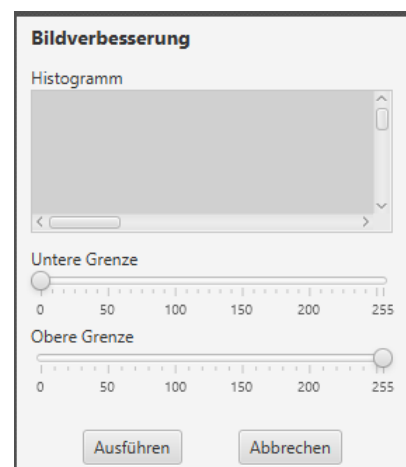
Aus der alten Intensität i_{alt} wird für die untere Grenze u und die obere Grenze o die neue Intensität i_{neu} also mit der folgenden abschnittsweise definierten Funktion berechnet:

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



Aufgaben

6. Öffne das Dokument `histogramme.ods` erneut. Lasse dir die Formel der Zelle BC7 anzeigen. Vergleiche mit der hier angegebenen Formel. Experimentiere mit der unteren und oberen Grenze (Zellen BA1 und BC1), um das Bild zu verbessern. Formuliere eine Regel, wie sich unterbelichtete bzw. überbelichtete Bilder verbessern lassen.
7. Implementiere eine Methode `public Picture lineareHistogrammanpassung (Picture originalbild, int unten, int oben)`. Diese soll ein Zuordnungsarray gemäß der oben formulierten Regel füllen und dann mit der Methode `histogrammAnpassen` die Änderungen durchführen. Erzeuge dazu zunächst ein `int`-Array mit 256 Einträgen. Fülle dieses Array. Dabei entspricht der Index der Arrayposition der alten Intensität und der Wert des Arrays der neuen Intensität. Rufe dann die Methode `histogrammAnpassen` mit diesem Array und dem OriginalBild auf und gib das Ergebnis zurück. Tipp: Wenn man `(int)` vor eine Kommazahl schreibt, werden störende Nachkommastellen einfach abgeschnitten.
8. Erstelle mit dem Gluon Scene Builder ein Optionsfenster mit einem `PictureViewer` für die Anzeige eines Histogramms und zwei `Slidern` für die obere und untere Grenze. Die Buttons `Ausführen` und `Abbrechen` sind standardmäßig dabei. Füge in der Haupt-GUI einen neuen Menüpunkt "lineare Histogrammanpassung" hinzu. Passe den Controller so an, dass bei Aufruf des Menüpunktes dieses Optionsfenster angezeigt wird. Das Histogramm des aktuellen Bildes soll berechnet und angezeigt werden. Beim Bewegen der Slider soll mit der Methode `lineareHistogrammanpassung` das Bild angepasst und angezeigt werden. Orientiere dich dabei an der Vorgehensweise der anderen Optionsfenster.

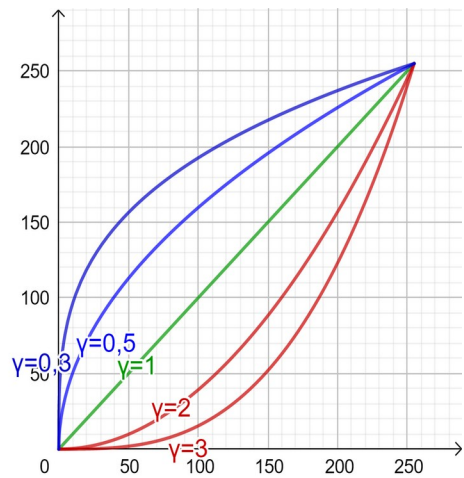




Gammakorrektur:

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.

Zur Berechnung werden die Werte zunächst auf maximal 1 normiert (durch 255 geteilt). Dadurch liegen die Werte im Bereich 0-1. Potenziert man dann mit Gamma, dann bleibt 0 bei 0 (da 0 hoch irgendwas immer 0 ergibt) und auch 1 bei 1 (gleiche Begründung). Nur die Werte dazwischen ändern sich. Danach wird wieder mit 255 multipliziert, um Intensitäten von 0-255 zu bekommen.



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

Aufgabe

9. Implementiere eine Methode `public Picture gammakorrektur(Picture originalbild, double gamma)`, die das Bild `originalbild` mit Hilfe der Gammakorrektur anpasst.

Erzeuge dazu zunächst ein `int`-Array mit 256 Einträgen. Fülle dieses Array. Dabei entspricht der Index der Arrayposition der alten Intensität und der Wert des Arrays der neuen Intensität. Rufe dann die Methode `histogrammAnpassen` mit diesem Array und dem `originalBild` auf und gib das Ergebnis zurück.

Tipp: Wenn man (`int`) vor eine Kommazahl schreibt, werden störende Nachkommastellen einfach abgeschnitten.

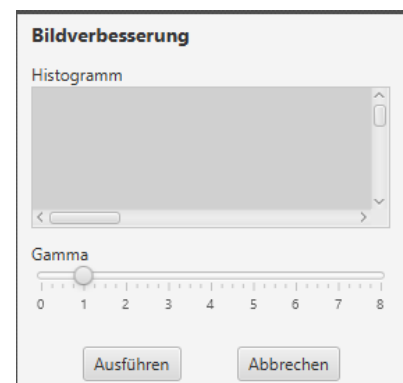
10. Erstelle mit dem Gluon Scene Builder ein Optionsfenster mit einem `PictureViewer` für die Anzeige eines Histogramms und einem Slider für den Gamma-Wert. Die Buttons `Ausführen` und `Abbrechen` sind standardmäßig dabei.

Füge in der Haupt-GUI einen neuen Menüpunkt "Gammakorrektur" hinzu.

- Passe den Controller so an, dass bei Aufruf des Menüpunktes dieses Optionsfenster angezeigt wird. Das Histogramm des aktuellen Bildes soll berechnet und angezeigt werden.

Beim Bewegen der Slider soll mit der Methode `lineareHistogrammanpassung` das Bild angepasst und angezeigt werden. Außerdem soll das Histogramm des neu berechneten Bildes erzeugt und im Histogrammfenster angezeigt werden.

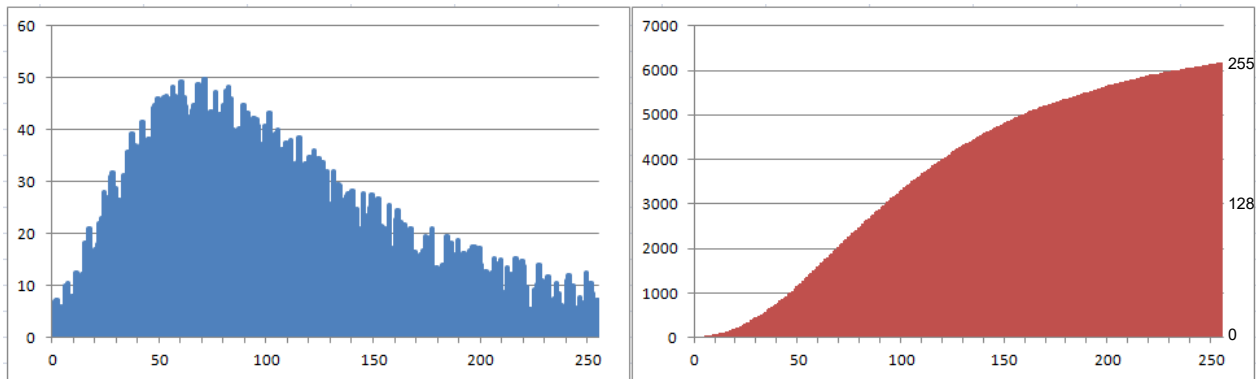
Orientiere dich dabei an der Vorgehensweise der anderen Optionsfenster.





Automatische Histogrammäquibilisierung:

In einem gut belichteten Bild verteilen sich die Helligkeiten relativ gleichmäßig. Dies kann man ausnutzen, um zu versuchen, das Histogramm dem einer Gleichverteilung anzunähern. 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 das gesamte Spektrum gleichmäßiger genutzt wird, indem manche Bereiche gedehnt und andere gestaucht werden.



Zur Berechnung der neuen Intensität für I_{neu} addiert man alle Häufigkeiten bis zu I_{alt} . Dadurch erhält man eine Verteilung wie im rechten Bild. Diese kann man nutzen, um die Zuordnung festzulegen. Teilt man durch die Gesamtzahl der Pixel (hier 6100) und multipliziert anschließend mit 255, verändert man die Skalierung so, dass die neuen Werte zwischen 0 und 255 liegen (rechte Skala).

Aufgabe

11. Bearbeite das Arbeitsblatt `alg_histogramm_ab.odt`.



12. Implementiere eine Methode `public Picture`



`automatischeHistogrammaequibilisation(Picture originalbild)`, die das Bild `originalbild` automatisch anpasst.

Bestimme die Gesamtzahl der Pixel aus der Bildgröße. Bestimme die Häufigkeitsverteilung, indem du die entsprechende Methode nutzt. Erzeuge ein `int`-Array `zuordnung` mit 256 Einträgen. Fülle dieses Array in einer `for`-Schleife. Summiere dazu die Häufigkeiten im Originalbild auf. Der Wert des Array entspricht dann dem wie oben beschriebenen normierten Wert der Summe. Rufe dann die Methode `histogrammAnpassen` mit diesem Array und dem Originalbild auf und gib das Ergebnis zurück.

Tipp: Wenn man (`int`) vor eine Kommazahl schreibt, werden störende Nachkommastellen einfach abgeschnitten.

13. Füge in der Haupt-GUI einen neuen Menüpunkt "automatische Histogrammäquibilisation" hinzu.



Passe den Controller so an, dass bei Aufruf des Menüpunktes die Anpassung automatisch durchgeführt und angezeigt wird.