

ANHANG

I. Meine Vorstellungen zum Thema Bildbearbeitung

Was ist eigentlich Farbe? Warum sehen wir die Dinge farbig?

Wie werden vom Computer Farben "erzeugt"? Warum können wir auf dem Bildschirm Farben sehen?

Hast du schon einmal Fotos am Computer nachbearbeitet? Wenn ja, hast du ihre Qualität verbessern können?

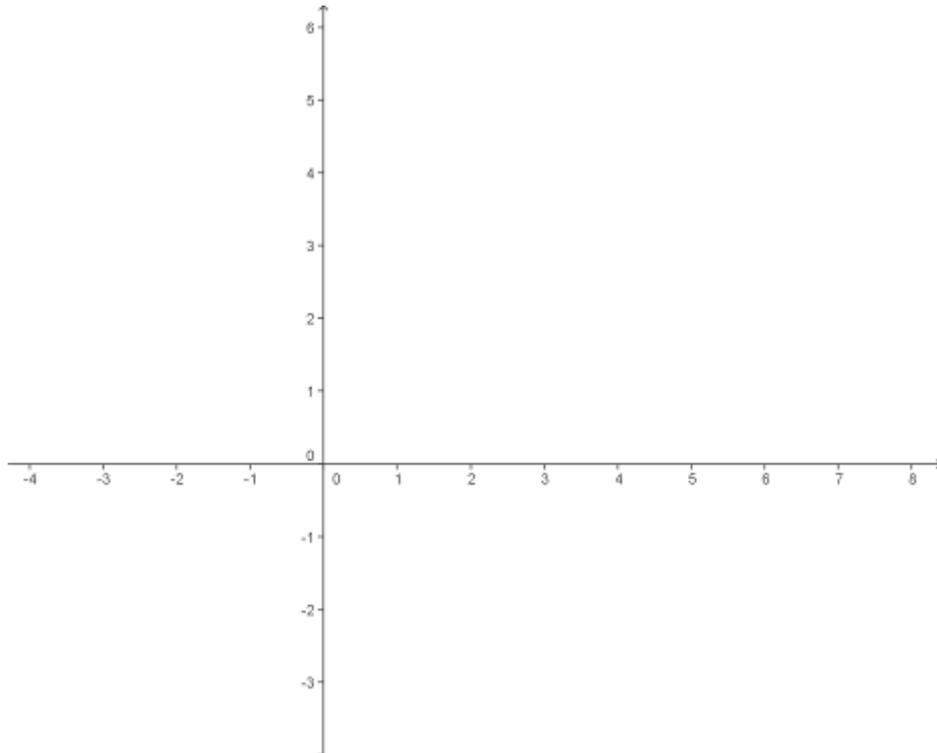
Weißt du, wie Statistik in der Bildbearbeitung angewendet wird?

Berechne bitte den Mittelwert der drei Zahlen 6, 5 und 10:

Was ist der Median dieser 3 Zahlen?

Kannst du erklären, was der Ausdruck "Varianz" bedeutet?

Weißt du, was ein Vektor ist? Wenn ja, zeichne 2 beliebige Vektoren a und b in das Koordinatensystem, und gib ihre Koordinatendarstellung an.



Vielleicht hast du ja schon einmal erlebt, wie Fotografen am Computer Passfotos nachbearbeiten. Wie, glaubst du, gehen Bildbearbeitungsprogramme vor, um z.B. fettig glänzende Hautstellen zu korrigieren?

Wie, glaubst du, schaffen es Bildbearbeitungsprogramme, verschwommene Fotos zu schärfen?

Was macht deiner Meinung gute Qualität bei einem Foto aus?

Weißt du, wie es ein Roboter schafft, ein Objekt zu erkennen (und zu bewegen), z.B. den Fußball beim Robocup (Fußballcup für Roboter)?

Wann ist deiner Meinung nach ein Bild "schön"?

II. Beobachtung Gruppenarbeit

Gruppe:	E-M ¹	Gruppenstruktur: Chef/in? Organisator/in?	Unterstützung benötigt? Bei welchen Problemen?
Essen & Trinken ♀			
Sport ♂			
Menschen & Tiere ♂			
Kleine Gegenstände ♀♂			
Emotion ♂			
Himmel ♀			

Sonstige Beobachtungen:

¹ E-M bedeutet: Die Beobachtungsperson soll ca. alle 10 Minuten für alle Gruppen eintragen, ob die Gruppenmitglieder gerade einzeln (E) oder miteinander (M) arbeiten.

III. Selbstbeobachtung Gruppenarbeit

Wenn du jetzt noch einmal daran denkst, wie ihr in eurer Gruppe gearbeitet habt...
zu wie viel Prozent ungefähr habt ihr einzeln gearbeitet, wie viel als Gruppe?

wie habt ihr euch die Arbeit in der Gruppe aufgeteilt?

gab es in eurer Gruppe eine/n, die/der die Verteilung der Arbeit und die Organisation in die Hand genommen hat?

haben alle ungefähr gleich viel gearbeitet, oder die einen mehr, die anderen weniger?

hat in eurer Gruppe größtenteils ein und die selbe Person die Arbeit am Computer übernommen?

Global gesehen, welche Note würdet ihr euch dafür geben, wie ihr in der Gruppe gearbeitet habt?

IV. Übungen mit Matlab

Demonstration des Programms

Ablauf:

- a. Die Lehrperson demonstriert Matlab am Lehrercomputer über den Beamer. Sie erklärt den Bildschirmaufbau, der nach dem Programmstart erscheint. Sie führt in der Kommandozeile einige Grundrechnungsoperationen vor. Im Matlab-Arbeitsplatz führt sie vor, wie Ergebnisse betrachtet werden können (Array Editor). Sie erklärt, dass Matlab nur numerisch rechnen kann, das heißt, es kann nicht Terme vereinfachen, oder Gleichungen lösen. Wenn wir Variable verwenden wollen, müssen wir ihnen immer einen Wert zuweisen. Wir müssen Variable allerdings nicht deklarieren, das macht Matlab automatisch. Dies zeigt die Lehrperson im Matlab-Arbeitsplatz (Class).

Für Matlab ist jede Variable eine Matrix, davon kommt auch der Name Matlab. Auch das Ergebnis einer Grundrechenoperation, die Variable „ans“, ist eine Matrix, eine 1 x 1 Matrix. Dies ist ebenfalls im Matlab-Arbeitsplatz zu sehen (Size).

- b. Eine Matrix können wir selbst anlegen und sie als Bild betrachten. Die Lehrperson legt eine Matrix bild an (`bild=zeros(80,100);`). Sie zeigt die Matrix im Matlab-Arbeitsplatz. bild ist eine 80 x 100 Matrix, wobei 80 die Anzahl der Zeilen ist, und 100 die Anzahl der Spalten. Dann zeigt sie bild als Bild an (`imshow(uint8(bild));` oder `image(uint8(bild));`). „Zeros“ bedeutet „Nullen“, das Bild ist also homogen schwarz.
- c. Wir können die Matrix auch verändern. Das bedeutet, wir ändern die Zahlen, die in der Matrix stehen. Wir schreiben in einen Teil der Matrix andere Zahlen hinein (`bild(1:50,:)=150;`) und zeigen die Matrix wieder als Bild an. Die Änderung der Zahlen können wir auch im Array Editor ausführen. Die Lehrperson demonstriert diese beiden möglichen Wege.
- d. Wir lesen nun eine Fotodatei in eine Matrix ein (`rose=imread('...[vollständiger Pfad]...\rose1.tif');`). Die Lehrperson demonstriert die Matrix rose im Matlab-Arbeitsplatz (Class) und im Array Editor. Die Matrix ist eine M x N x 3-Matrix, mit M als Anzahl der Zeilen und N als Anzahl der Spalten. Die Lehrperson zeigt rose auch als Bild an.
- e. Bis jetzt haben wir einzeilige Befehle in das Kommandofenster geschrieben. Wenn wir Abfolgen von mehreren Befehlen schreiben wollen - sogenannte Scripts - machen wir das im Matlab-Editor. Die Lehrperson öffnet den Editor (File -> New -> M-File) und speichert die neue M-Datei unter `muster.m`. Dann kann ein Script geschrieben werden, z. B.:

```
close all; %alle vorhandenen Bilder schließen
bild=zeros(80,100); %Matrix bild (80 x 100) anlegen
for zeile=1:80 %Schleife über alle Zeilen
    for spalte=1:100 %und alle Spalten
        if mod(zeile,5)==0 %falls Zeilennummer
            durch 5 teilbar,
```

```

        bild(zeile,spalte)=180;           %weise allen Pixeln
                                         %dieser Zeile den
                                         %Grauwert 180 zu
    end
end
end;
figure;
imshow(uint8(bild));                    %Bild zeigen

```

Die Lehrperson führt das Script aus (Klick auf die Taste mit dem blauen Pfeil nach unten, Run). Matlab ist so voreingestellt, dass es automatisch speichert.

Methoden:

Die Lehrperson demonstriert Matlab am Lehrercomputer über den Beamer. Am Beginn sollen die Schüler/innen sich so hinsetzen, dass alle gut auf die Leinwand sehen können. Ab Punkt b. sollen die Schüler/innen an die Computer gehen, das Programm starten und die demonstrierten Befehle eingeben. Die Schüler/innen arbeiten in ihren Kleingruppen und wechseln sich bei den Eingaben ab. Die Lehrperson schreibt die Eingaben auch an die Tafel. Bei Punkt e. wiederholt sie, dass das Schlüsselwort for eine Schleife einleitet. Die Matrix wird Zeile für Zeile und Spalte für Spalte durchlaufen, das heißt, jedes Pixel kommt an die Reihe. Die Lehrperson erklärt den Ausdruck mod (Divisionsrest) und den Ausdruck bild(zeile, spalte) (aktuelle Position im Bild).

Weitere Übungen

Ablauf:

Das Script muster.m (siehe oben) wird modifiziert, sodass statt der Streifen Punkte (&&), oder Kästchen (||) sichtbar werden. Eine weitere Möglichkeit ist es, eine Matrix rot zu erzeugen, mit

```

rot=zeros(80,100,3);
rot(:,:,1)=255;

```

Um rot als Bild anzeigen zu können, muss sie zuerst in den Datentyp uint8 (ganze Zahlen) umgewandelt werden, mit

```

rot=uint8(rot);

```

Die Anzeige als Bild erfolgt mit

```

figure;
imshow(rot);

```

Um die Farbe zu ändern, wird z. B. eine Matrix orange angelegt, mit

```

orange=rot;
orange(:,:,2)=180;

```

Danach kann das orange Bild mit grünen Streifen versehen werden, indem das Script muster.m adaptiert wird, z. B. mit:

```
for zeile=1:80
    if mod(zeile,5)~=0           %falls Zeilennummer nicht durch 5 teilbar,
        orange(zeile,:,1)=0;    %setze die Rotanteile aller Pixel
                                %dieser Zeile auf 0
    end
end;
end;
```

Method:

Die Lehrperson skizziert die Aufgaben als Problemstellungen an der Tafel (Ich habe: Ein rotes Bild - Ich möchte: Ein oranges Bild). Jede Aufgabenstellung wird zuerst im Plenum besprochen. Die Schüler/innen arbeiten in Kleingruppen an den Lösungen. Sie sollen ihre Lösungswege zuerst ins Heft schreiben, dann erst programmieren.

V. Projekt V: Fade Fotos? Korrigieren!

1. Lest die folgende Information und bearbeitet gemeinsam die gestellten Aufgaben

Manche Bilder wirken zu hell oder zu dunkel. Sie sehen blass und fad aus. Wir erkennen auch zu wenige Details. Ein Beispiel aus einer Fotografenwebseite²:



Überlegt gemeinsam, warum das Foto oben langweilig wirkt.

.....

.....

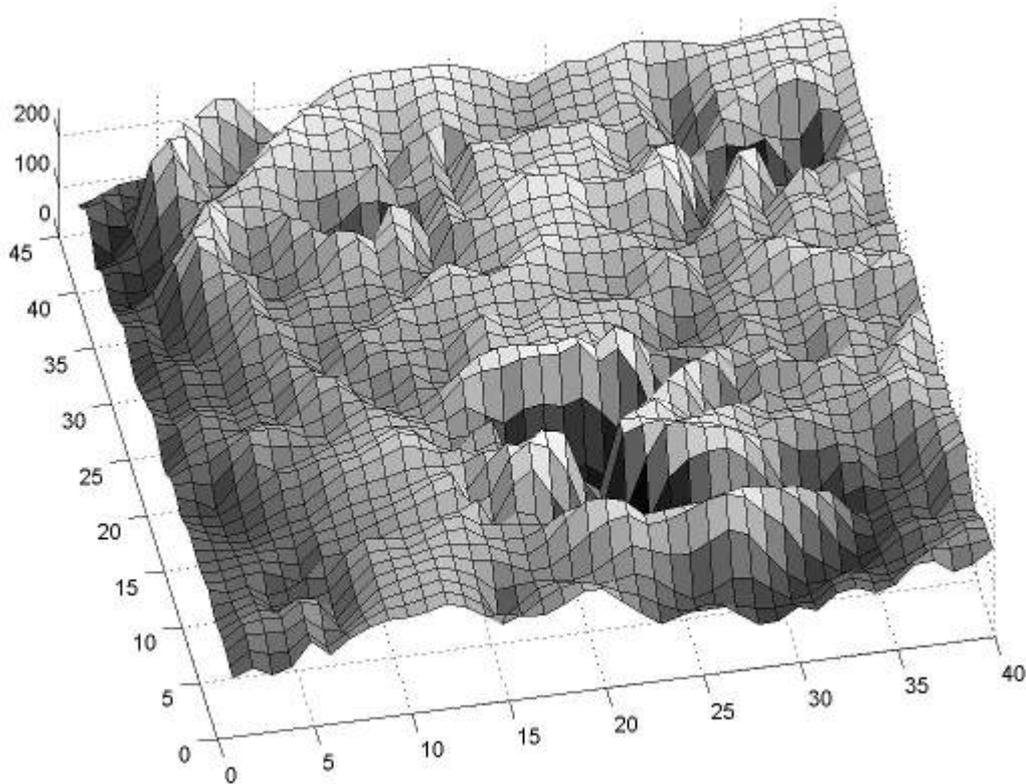
.....

Um zu verstehen, wie ein solches Bild interessanter gemacht werden kann, rufen wir uns wieder in Erinnerung, was ein Bild für den Computer ist:

Ein Bild wird vom Computer als Matrix aufgefasst. Eine Matrix ist eine Art Tabelle, in der aber nur gleichartige Werte Platz finden können. In einer Bildmatrix können z. B. positive ganze Zahlen im Bereich $[0, 255]$ erlaubt sein.

Wir können uns die Bildmatrix auch 3-dimensional, als Landschaft vorstellen: Die Zahlen der Matrix geben die Höhen der Gebirge in dieser Landschaft an. Im Bild auf der nächsten Seite seht ihr einen Löwenkopf als Landschaft. Weit heraus ragende Teile, die Gipfel der Landschaft, haben relativ große Zahlenwerte, wie das Ohr links oben. Das sind die hellen Bereiche des Bilds. Du kannst dir vorstellen, die Gipfel sind mit Schnee bedeckt.

² http://jmsg.org/content_tip-1667.php



Tiefe Teile der Landschaft, die Seen und Meere, sind dunkel. Sie werden durch einen kleinen Zahlenwert repräsentiert. Das sind z. B. die Augen und der dunkle Teil der Nase. Seht ihr sie?

In Matlab heißt der Datentyp, der Zahlen im Bereich $[0, 255]$ umfasst, „uint8“ (unsigned = positiv, integer = ganz, 8 bit Speicherplatz für die einzelne Zahl). Wenn wir mit einer Matrix in Matlab rechnen wollen, müssen wir sie erst vom Datentyp uint8 in den Datentyp double (hier sind Kommazahlen möglich) umwandeln.

Bildmatrizen enthalten in Matlab oft Zahlen im Bereich $[0, 255]$. Das ist übrigens auch beim Bild oben der Fall, was ihr an der z-Achse sehen könnt. Aber es sind auch andere Bereiche möglich. Der Bereich $[0, 1]$ z. B. wird ebenfalls verwendet, weil er manchmal praktischer zum Rechnen ist.

Aufgabe 1:

Die folgende Tabelle, wir nennen sie B , ist ein kleiner Ausschnitt aus einer Bildmatrix. Berechnet nun B_n , indem ihr alle Werte von B in den Bereich $[0, 1]$ bringt. Das Feld $B_n(1,1)$ ist schon ausgerechnet:

26	0	127
204	205	255

B

0,102		

B_n

Schreibt den Rechenweg hin, wie ihr B_n ausgerechnet habt:

B_n(1,2) =

B_n(1,3) =

B_n(2,1) = und so weiter...

Ob im Bereich [0, 1] liegend oder im Bereich [0, 255], die Zahl 0 bedeutet bei einem Schwarz-Weiß-Bild - dieses wird korrekt "Grauwertbild" genannt, da ja nicht nur Tief-schwarz und Reinweiß vorkommen -, Schwarz oder keine Farbe, das Maximum (255 oder 1) bedeutet Weiß.

Diese Matrix, in der die Zahlen enthalten sind, ist, genau wie eine Tabelle, aus Zeilen und Spalten aufgebaut. An den Kreuzungen von Zeile und Spalte - in einer Tabelle sind das die Zellen - sind die Grauwerte der einzelnen Pixel eingetragen:

Betrachtet noch einmal B. B(1,2) ist das Pixel in der 1. Zeile und 2. Spalte. Der Grauwert von B(1,2) ist 0, d.h. das Pixel erscheint schwarz. B(2,3) ist 255, d.h. das Pixel B(2,3) erscheint weiß.

Fotos können wir dunkler machen, indem wir alle Grauwerte mit einem Faktor kleiner als 1, z.B. mit 0,8, multiplizieren. Wie sich das auf unseren Bildausschnitt auswirkt, siehst du hier:

26	0	127
204	205	255

B

* 0,8 =

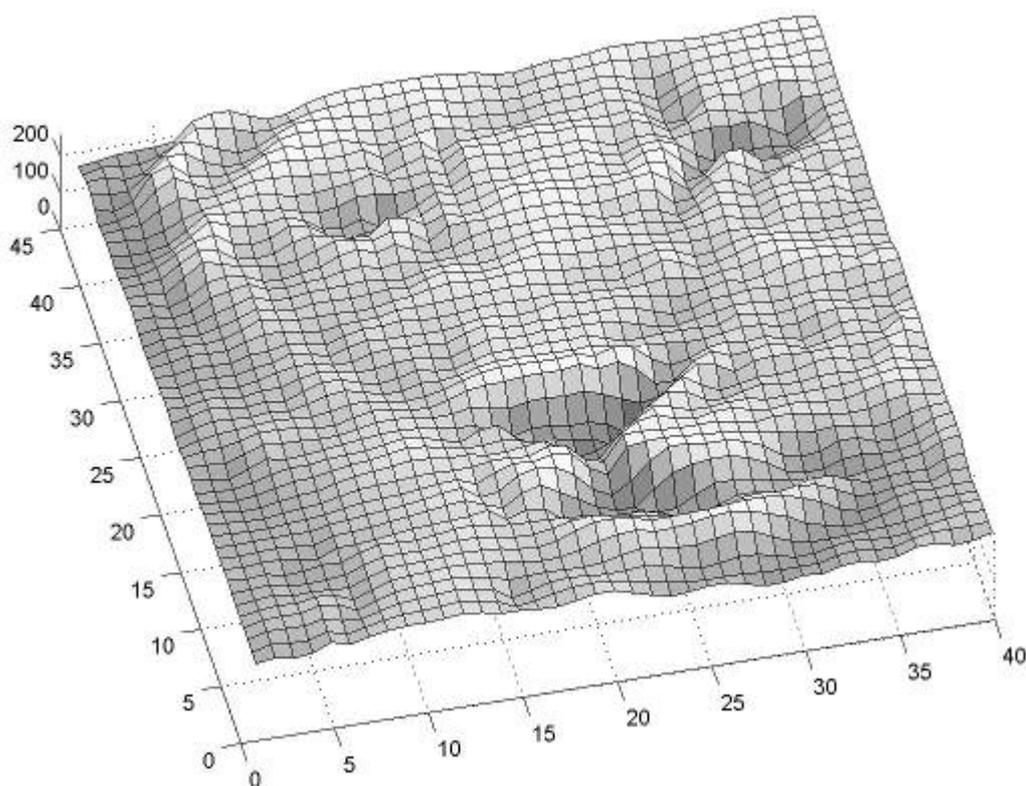
21	0	102
163	164	204

B'

Alle Grauwerte in B' sind kleiner geworden, das bedeutet, der Bildausschnitt erscheint dunkler.

Durch einfaches Vermindern der Grauwerte werden zu helle Bilder, wie unser Beispielfoto der Löwin, zwar dunkler, sie wirken aber immer noch langweilig. Das hat folgenden Grund: Viele Fotos schöpfen nicht alle Möglichkeiten des gesamten Helligkeitsbereichs aus. Wenn z. B. in einem überbelichteten Foto die meisten Pixel Werte zwischen 100 und 200 haben, hilft es nicht viel, von jedem Pixel 50 abzuziehen: Das Bild ist jetzt dunkler, aber genauso fad.

Ein fades Bild kannst du mit einer flachen, eintönigen Landschaft vergleichen. Die „Kopflandschaft“ der Löwin von Seite 1 sieht nach ihren wirklichen Grauwerten so aus:



Diese Landschaft ist doch wirklich flacher als die auf Seite 2, nicht wahr? Wie eine verschneite Hochebene.

Die Lösung heißt in diesem Fall **Histogrammdehnung**. Wir dehnen den Wertebereich des Bildes so, dass wir den gesamten möglichen Bereich von 0 - 255 überdecken. Unsere Bildpixel haben jetzt nicht mehr Werte nur zwischen 100 und 200, son-

dern wirklich zwischen 0 und 255. Dadurch wird der Kontrast erhöht, weil die Grauwerte weiter auseinander rücken. Die Landschaft wird abwechslungsreicher, zerklüfteter.

Bevor wir uns die Rechenoperationen anschauen, mit denen wir diese Dehnung ausführen werden, wollen wir uns genauer über die Helligkeitswerte unserer Bildmatrix klar werden. Diese Information liefert uns das *Bildhistogramm*.

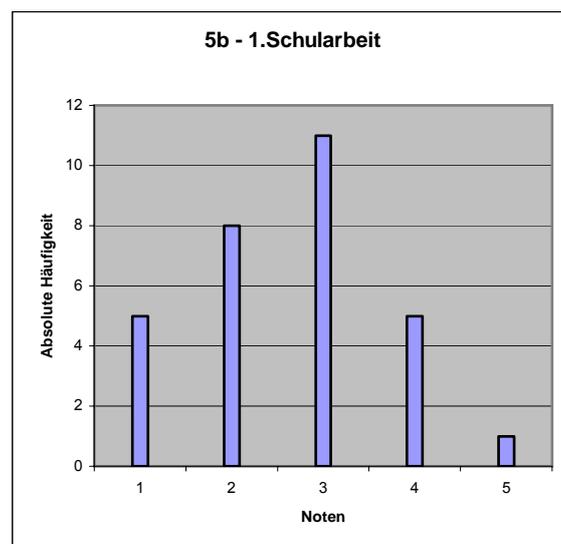
A) Das Histogramm

Ein Histogramm sieht so ähnlich aus wie ein Balkendiagramm in Excel. Auf der x-Achse werden die möglichen Werte - bei Bildern sind das die Helligkeitswerte 0 bis 255, bei Schularbeiten z.B. sind das die Noten 1 bis 5 - , aufsteigend geordnet, eingetragen. Auf der y-Achse können wir ablesen, wie oft jeder dieser Werte vorgekommen ist, also ihre relative oder absolute Häufigkeit.

Ein Beispiel aus der Schulwelt:

Von den 30 SchülerInnen der Klasse 5b hat ein Schüler auf die erste Mathematikschularbeit einen Fünfer geschrieben, aber - welch ein Glück für die anderen - 5 einen Einser, 8 einen Zweier, 11 einen Dreier, und 5 einen Vierer.

Bei diesem Beispiel tragen wir alle Noten, die bei dieser Schularbeit vorgekommen sind, also 5, 1, 2, 3, 4, aufsteigend geordnet auf der x-Achse auf. Darüber, in Form von Balken oder Linien, tragen wir auf, wie oft die Noten vorgekommen sind, also ihre absolute Häufigkeit. So sieht das Histogramm aus:



Histogramme werden gezeichnet, um die Verteilung von Daten zu veranschaulichen. Ihr bekommt, wenn ihr die Grafik ansieht, auf den ersten Blick einen Eindruck, wie die Schularbeit ausgefallen ist: Relativ gut, deutlich mehr Noten im Bereich von 1 - 2 als im Bereich 4 - 5. Die Note 3 kommt am häufigsten vor. Ganz wie ein Schularbeitsergebnis sein sollte!

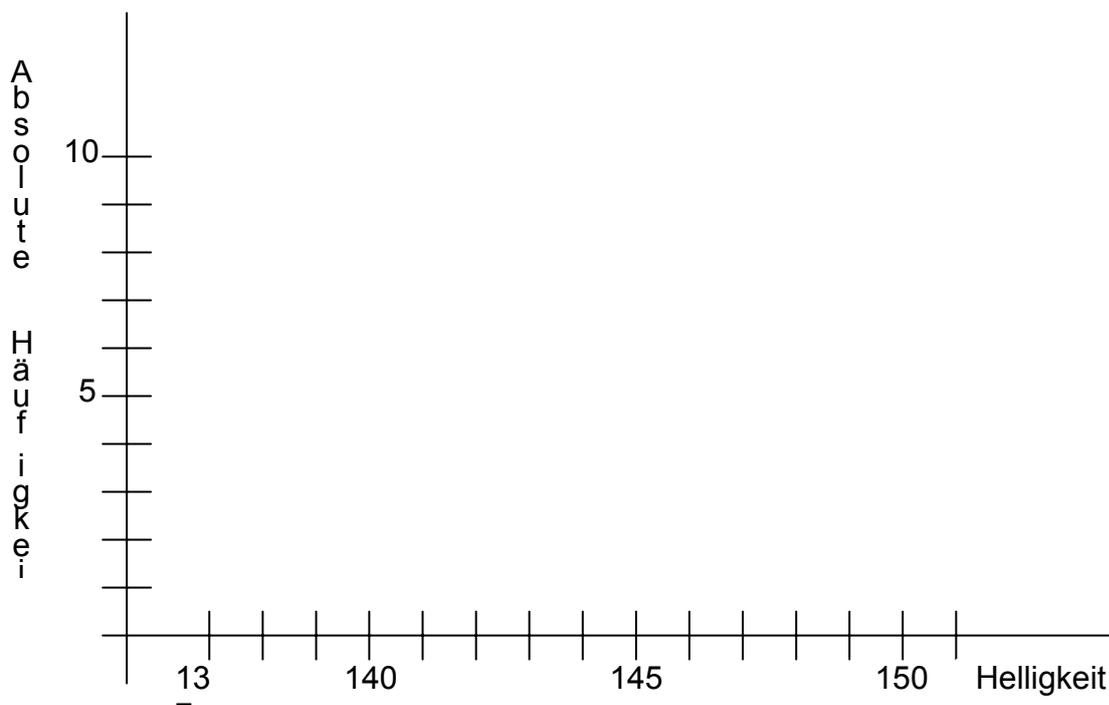
Aufgabe 2:

Zeichnet ein Histogramm von Helligkeitswerten!

Ein Bildausschnitt hat die folgenden Grauwerte:

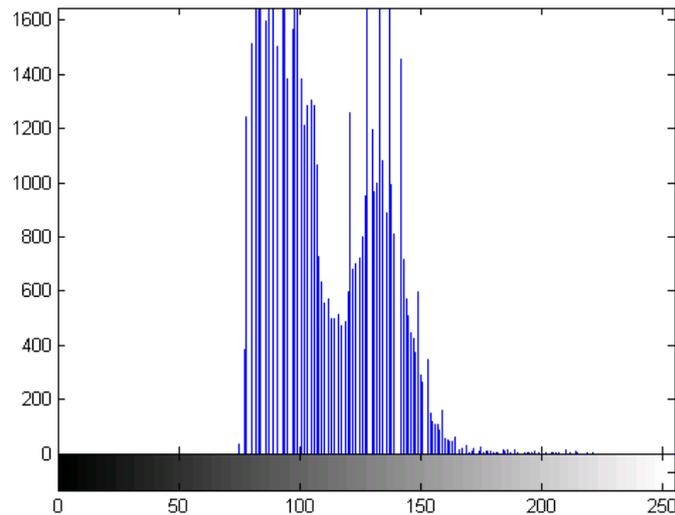
148	150	150	147	149
137	149	146	144	147
148	140	139	151	149
149	148	151	149	141
142	140	140	146	142
137	150	147	144	150
141	149	145	146	149
150	149	143	149	142

Zeichnet mit Bleistift und Lineal ein Histogramm dieser Grauwerte. Verwendet dazu die unten vorgegebenen Achsen:



Das ist jetzt das Histogramm eines kleinen Bildausschnitts (8 x 5 Pixel). In diesem Ausschnitt hattet ihr zufällig nur Grauwerte im Bereich [137,...,151]. Hätte das gesamte Bild aber nur Werte in einem ähnlich kleinen Bereich, so wäre es kontrastarm und würde eintönig wirken.

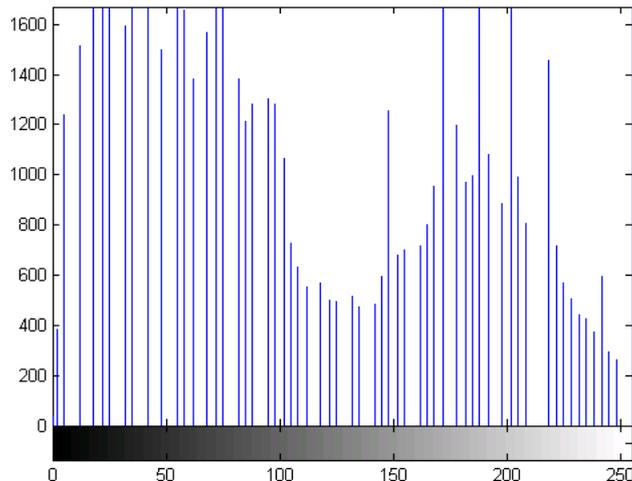
Das Histogramm eines eintönigen, undifferenzierten Bilds könnte etwa so aussehen:



In diesem Histogramm liegen die meisten Grauwerte im Bereich [75, 150]. Dieser Bereich ist nur 75 Werte breit, er beträgt knapp 30% des gesamten zur Verfügung stehenden Bereichs von 0 bis 255. Das Foto ist kontrastarm, wirkt also langweilig, eben wie unser Löwinnenfoto. Wir sehen allerdings auch, dass mehr Werte im Bereich unterhalb des Mittelwerts von 127,5 liegen als oberhalb. Das Bild ist also, im Gegensatz zur Löwin, eher dunkel.

B) Die Histogrammdehnung

Wenn nun die Balken des Histogramms auf Seite 6 gleichmäßig über den Bereich [0, 255] aufgeteilt werden - d.h. das Histogramm wird gedehnt, sieht das Histogramm des bearbeiteten Bilds etwa so aus:



Bei dieser Dehnung bleibt die Form des Histogramms gleich. Das bearbeitete Foto ist immer noch eher dunkel. Das Histogramm ist jetzt aber auf den gesamten Helligkeitsbereich, der zur Verfügung steht, ausgedehnt. Es entstehen keine neuen Grauwerte: Der Platz zwischen den Balken bleibt leer. Dadurch bekommt das bearbeitete Foto mehr Kontrast: Es wirkt weniger flach und fad.

Das Histogramm eines optimalen Fotos ist ausgewogen: Es füllt den gesamten zur Verfügung stehenden Bereich aus und ist weder links- noch rechtslastig, d.h. sein Schwerpunkt liegt ungefähr in der Mitte.

Aufgabe 3:

Probiert die Dehnung des Histogramms in Xnview und Matlab aus!

Verwendet als Beispiel das Grauwertbild loewin.jpg. So geht ihr vor:

Ihr öffnet in XnView das Foto. Unter dem Menüpunkt Bild -> Anpassen -> Stufen. Hier seht ihr das Histogramm. Aus seiner Form könnt ihr zwei Dinge erkennen: a) Das Histogramm füllt nur ca. 2/3 des Platzes aus, das ihm zur Verfügung stünde, d.h. das Bild ist kontrastarm. b) Das Histogramm ist "rechtslastig", sein Schwerpunkt liegt rechts vom Mittelwert. Das bedeutet, das Foto ist zu hell.

Zur Korrektur des Kontrasts stellt ihr Schwarzpunkt und Weißpunkt auf die Ränder des Histogramms und beobachtet in der Vorschau, was passiert: Der Bereich zwischen neuem Schwarzpunkt und neuem Weißpunkt wird auf den gesamten Wertebereich aufgedehnt, dadurch erscheint das Foto kontrastreicher. Zieht dann die beiden Punkte von links bzw. von rechts ein wenig in das Histogramm hinein. Dadurch wer-

den die äußersten Werte zwar abgeschnitten - d.h. Bildinformationen gehen verloren -, aber wenn der Verlust relativ klein ist, merkt der Betrachter es nicht. Je näher wir Schwarz- und Weißpunkt bringen, umso weiter wird das Histogramm gedehnt, und umso stärker werden die Kontraste, bis es nur mehr Schwarz und Weiß gibt.

Sucht nun einen für euer Empfinden passenden Schwarz- und Weißpunkt und notiert die Werte:

Schwarzpunkt:

Weißpunkt:

Speichert euer korrigiertes Foto als `loewin_hist1.jpg` in eurem Projektordner. Dann schaut ihr euch wieder unter Bild -> Anpassen -> Stufen das Histogramm des korrigierten Bilds an. Dieses sollte jetzt den gesamten Bereich `[0,255]` ausfüllen. Dieselbe Methode funktioniert übrigens auch bei Farbfotos.

Das war Histogrammkorrektur mit XnView; als nächstes probiert ihr dasselbe in Matlab aus: Um das Bildhistogramm anzuschauen und zu korrigieren, gibt es in Matlab zwei fertige Funktionen. Funktionen sind in der Informatik kleine Unterprogramme, die ein gewünschtes Zwischenergebnis liefern. Das Bildhistogramm könnt ihr mit der Funktion `imhist` berechnen, die Korrektur mit der Funktion `imadjust`. `imhist` funktioniert nur bei Bildmatrizen mit nur einer Lage, das heißt bei Grauwertbildern. Bei einer Farbmatrix müsst ihr also die Lagen, sprich die Rot-, Grün- und Blauanteile, trennen und euch deren Histogramme einzeln ansehen. `imadjust` funktioniert auch bei Farbbildern.

Ihr ladet das Grauwertbild `loewin.jpg`, indem ihr in das Kommandofenster schreibt:

```
loewin=imread('X:\...\Projekt\...\loewin.jpg'); und Enter drückt. Statt  
...\Projekt\... müsst ihr den genauen Pfad zur Bilddatei angeben!
```

Mit dieser Anweisung erzeugt ihr eine `200 x 299` Matrix `loewin`. Deren Histogramm lasst ihr mit

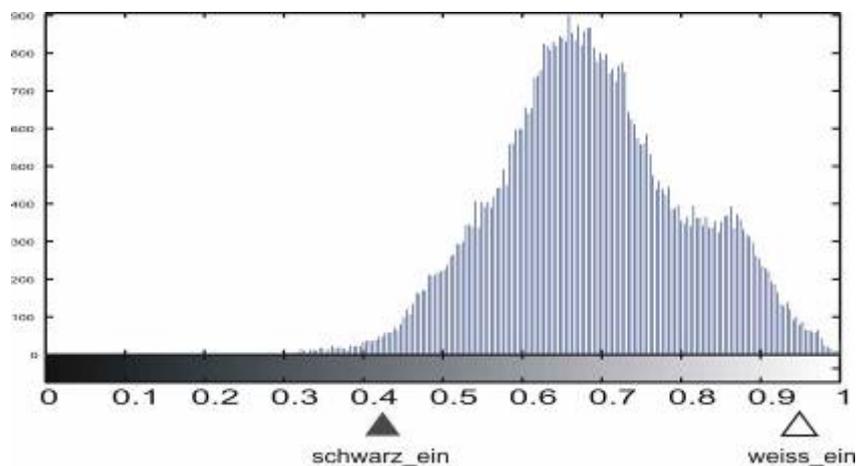
```
imhist(loewin); und Enter ausgeben..
```

In Matlab wird, wie in eurer eigenen Zeichnung, die x-Achse mit den möglichen Werten beschriftet, die y-Achse mit der absoluten Häufigkeit ihres Vorkommens. Die Form des Histogramms sollte jedoch, bis auf anzeigebedingte Unterschiede, gleich sein wie die des von Xnview erzeugten Histogramms.

In Matlab kann ein Histogramm mit der Funktion `imadjust` gedehnt werden. Lest zuerst, wie `imadjust` im Allgemeinen verwendet wird:

```
Bk = imadjust (B, [schwarz_ein, weiß_ein], [schwarz_aus, weiß_aus]);
```

Das heißt: `imadjust` erzeugt eine korrigierte Matrix B_k : Es dehnt (oder auch es staucht) dabei das Histogramm des Bilds B zwischen den Grenzen `schwarz_ein` und `weiß_ein` auf den Bereich zwischen `schwarz_aus` und `weiß_aus`. Also: Der Bereich in der ersten eckigen Klammer wird auf den Bereich in der zweiten eckigen Klammer gedehnt oder gestaucht. Dabei rechnet `imadjust` intern in einem Grauwertbereich $[0, 1]$. Es rechnet Bilder mit Grauwerten im Bereich $[0, 255]$ selbstständig auf $[0,1]$ um, das braucht ihr nicht zu machen.



Oben: Das Histogramm unseres Löwinnenfotos. Wir wollen es dehnen. Das heißt, wir spreizen den Bereich `[schwarz_ein, weiß_ein]` auf den gesamten möglichen Bereich $[0,1]$ auf. Unser zu dehnender Bereich ist genau der, den wir in Xnview zwischen Schwarzpunkt und Weißpunkt festgelegt haben.

Ihr verändert also die Programmzeile

```
Bk = imadjust(B, [schwarz_ein, weiß_ein], [schwarz_aus, weiß_aus]);
```

 indem ihr

- Statt B_k einen ausdrucksvollen Namen hinschreibt, z.B. `loewin_adjust`,
- Statt B den Namen der Bildmatrix, also `loewin`,
- In der ersten eckigen Klammer ersetzt ihr den Ausdruck `schwarz_ein` durch die Zahl, die ihr euch als Schwarzpunkt notiert habt, dividiert durch (= Schrägstrich) 255.
- Den Ausdruck `weiß_ein` ersetzt ihr durch euren Weißpunkt/255.

- In der zweiten eckigen Klammer tragt ihr statt `schwarz_aus` 0 ein, und statt `weiß_aus` 1.

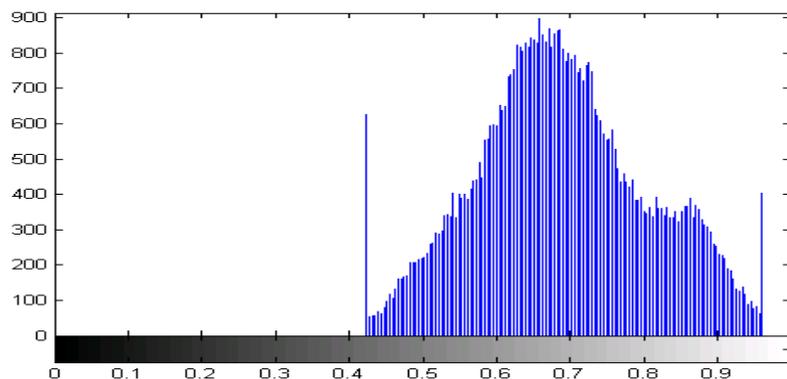
Die so angepasste Programmzeile schreibt ihr auch in das Kommandofenster und drückt Enter. Wenn es geklappt hat, lasst ihr euch mit der Zeile

```
imshow(loewin_adjust);
```

das korrigierte Foto anzeigen. Dieses sollte gleich aussehen wie das mit Xnview korrigierte.

Wie wird nun eine Histogrammdehnung gerechnet?

Das Histogramm des Bilds B, unseres Fotos, seht ihr auf Seite 9. `imadjust` berechnet vorerst ein Zwischenergebnis B': Alle Werte von B werden in den Bereich zwischen angegebenen Schwarz- und Weißpunkt [`schwarz_ein`,`weiß_ein`] gebracht. Dabei werden die Werte, die kleiner als `schwarz_ein` oder größer als `weiß_ein` sind, einfach durch `schwarz_ein` bzw. `weiß_ein` ersetzt. Alle übrigen Werte bleiben gleich.



In der obigen Grafik seht ihr das Histogramm der Matrix B'. B' hat keine Helligkeitswerte mehr, die kleiner als `schwarz_ein` = 0,42 oder größer als `weiß_ein` = 0,96 sind. Dass alle Helligkeitswerte außerhalb dieses Bereichs auf 0,42 bzw. 0,96 gesetzt worden sind, das seht ihr an den beiden längeren Linien an den Rändern des Histogramms.

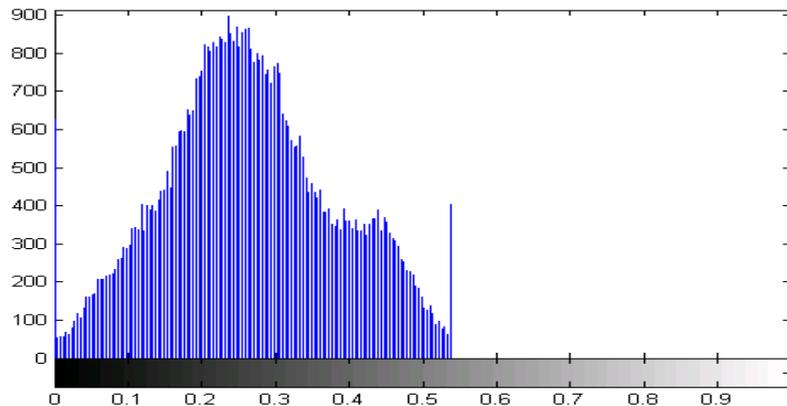
Die zentrale Anweisung für die Histogrammdehnung ist die folgende:

$$B_k = \frac{B' - \text{schwarz_ein}}{\text{weiss_ein} - \text{schwarz_ein}}, \text{ das bedeutet:}$$

`imadjust` berechnet das korrigierte Bild B_k , indem es

1. von jedem Helligkeitswert von B' den Wert des Schwarzpunkts abzieht.

Das bedeutet, es berechnet $(B' - \text{schwarz_ein})$. Der kleinste Helligkeitswert von $(B' - \text{schwarz_ein})$ ist 0. Das Histogramm von B' rückt dadurch ganz nach links, wie ihr in der Grafik unten seht:



2. $(B' - \text{schwarz_ein})$ durch $(\text{weiss_ein} - \text{schwarz_ein})$ dividiert.

$(\text{weiss_ein} - \text{schwarz_ein})$ ist der Abstand zwischen Schwarzpunkt und Weißpunkt. Dieser Abstand liegt zwischen 0 und 1, in unserem Beispiel, wie ihr in der Grafik oben sehen könnt, bei ca. 0,54. Eine Division durch eine Zahl zwischen 0 und 1 ist gleichbedeutend mit einer Multiplikation mit einer Zahl > 1 . In unserem Beispiel ist dies eine Multiplikation mit 1,85, da $1/0,54 = 1,85$. Durch diese Multiplikation wird das Histogramm auf den gesamten Bereich $[0,1]$ aufgedehnt. Das ist in unserem Beispiel für euch vielleicht nicht so leicht einzusehen, aber überlegt euch folgendes:

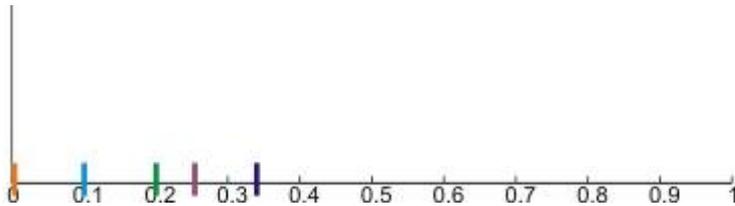
Wenn das Histogramm genau die Hälfte des zur Verfügung stehenden Bereichs einnehmen würde, dann würdet ihr durch die Zahl dividieren. Das wäre gleichbedeutend mit einer Multiplikation mit der Zahl

Wenn das Histogramm genau ein Drittel des zur Verfügung stehenden Bereichs einnehmen würde, dann würdet ihr durch die Zahl dividieren. Das wäre gleichbedeutend mit einer Multiplikation mit der Zahl

Überlegt euch weiters für das Bild, dessen Histogramm nur ein Drittel des Platzes ausfüllt:

Ihr hättet dann in der Bildmatrix $(B' - \text{schwarz_ein})$ nur Helligkeitswerte zwischen 0 und $1/3$. Dieser Bereich wird in der Grafik unten durch den orangen Strich bei 0 und den violetten Strich bei 0,33 begrenzt. Zur Bildkorrektur wird jeder Helligkeitswert,

der in (B' - schwarz_ein) vorkommt, mit der Zahl multipliziert, die ihr oben als letzte - hoffentlich richtig - eingefügt habt.



Führt diese Multiplikation an den farbig markierten Helligkeitswerten aus und zeichnet jedes Ergebnis - in der gleichen Farbe - in der Zahlengeraden oben ein! In welchem Zahlenbereich befinden sich also die Helligkeitswerte des korrigierten Bilds?

Aufgabe 4:

Korrigiert Farbbilder durch Histogrammdehnung!

Testet Farbbilder aus eurer Bildkategorie, ob sie mit Histogrammdehnung besser aussehen. Die Aufgabe könnt ihr in Xnview oder in Matlab machen. Speichert die zwei eurer Meinung nach besten Ergebnisse und dokumentiert ihre Daten unten:

Name der Bilddatei	[schwarz_ein, weiß_ein]		Anmerkungen

Beurteilt die korrigierten Bilder: Was hat sich optisch verändert? Sehen die Bilder interessanter aus? Könnt ihr mehr Details erkennen? Was ist euch aufgefallen?

.....

.....

.....

.....

2. Präsentation

Bereitet eine Präsentation vor, die ihr für eure Informatikgruppe halten werdet. Diese Präsentation soll ca. 10 Minuten lang sein:

- Ihr erklärt euren Mitschülern den theoretischen Hintergrund eurer Projektarbeit. Ihr geht auf die folgenden Punkte ein:
 - o Was ist ein Histogramm?
 - o Wie wird in XnView eine Histogrammdehnung durchgeführt, und wie wirkt sich diese auf ein Bild aus?
 - o Wie wird eine Histogrammdehnung gerechnet?

Dabei bearbeitet ihr einen der Punkte genauer. Zu diesem wählt ihr eine Aufgabe aus, die ihr gelöst habt und erklärt sie euren Kolleginnen und Kollegen so, dass es für sie nachvollziehbar ist.

- Ihr dürft (aber müsst nicht) zu diesem Thema auch weitere Informationen verwenden.
- Hebt hervor, was an eurem Projekt mit Mathematik zu tun hatte, und wozu euch hier Mathematik geholfen hat.

Achtet darauf, dass ihr bei der Präsentation wirklich von allen verstanden werdet. Damit eure Mitschüler/innen das, was ihr ihnen präsentiert, nicht sofort wieder vergessen, erstellt ihr ein Handout!

VI. Abschlussbefragung

Wir Lehrer/innen wollen unseren Unterricht verbessern. Deshalb bitte ich dich bei dieser Befragung um deine Mithilfe!

Allgemeines Fachwissen und -können

1. Ich kann mit einer Digitalkamera schöne Fotos machen.

stimmt genau eher ja eher nein überhaupt nicht

2. Während des Projekts habe ich besser mit einer Fotokamera umgehen gelernt.

stimmt genau eher ja eher nein überhaupt nicht

3. Ein Foto ist eine exakte Darstellung der Realität.

stimmt genau eher ja eher nein überhaupt nicht

4. Wenn ein Farbfoto unkomprimiert ist, und 1000 x 2000 Pixel Auflösung hat, dann ist sein Speicherplatzbedarf

2 MB 6 MB 600 KB

5. Während des Projekts habe ich Methoden kennen gelernt, um ein Foto besser oder interessanter zu machen.

stimmt genau eher ja eher nein überhaupt nicht

6. Bei manchen Methoden der Bildbearbeitung habe ich die Berechnung, die das Programm dabei ausführt, nachvollziehen können.

stimmt genau eher ja eher nein überhaupt nicht

7. Einige Berechnungen habe ich leider nicht verstanden.

stimmt genau eher ja eher nein überhaupt nicht

Spezielles Fachwissen (bei Wissensfragen können mehrere Möglichkeiten richtig sein!)

8. Ich kann nachvollziehen, durch welche Umrechnungen Helligkeit und Kontrast verändert werden.

stimmt genau eher ja eher nein überhaupt nicht

9. Das RGB-Farbmodell wird vor allem verwendet von

Druckern Monitoren Künstlern Digitalkameras

10. Nach einer Histogrammdehnung erscheint das Bild

heller kontrastreicher kontrastärmer

11. Differenzbilder werden verwendet

- bei ärztlichen Diagnosen
- in der Videokomprimierung
- um Bilder schärfer erscheinen zu lassen

Allgemeine Fragen zum Lernen und zur Zufriedenheit

12. Im Projekt habe ich mit eigenen Augen gesehen, wie sich Zahlenveränderungen auf das Aussehen eines Bildes auswirken.

stimmt genau eher ja eher nein überhaupt nicht

13. Ich habe viel mit den Leuten in meiner Gruppe zusammen gearbeitet.

stimmt genau eher ja eher nein überhaupt nicht

14. Wir haben oft die Lehrerin gefragt, weil uns etwas bei unseren Aufgaben nicht klar war.

stimmt genau eher ja eher nein überhaupt nicht

15. Die Arbeit mit Bildbearbeitung und ihren mathematischen Hintergründen hat mir gefallen.

stimmt genau eher ja eher nein überhaupt nicht

16. Bilder zu bearbeiten war lustig, aber die Mathematik dabei finde ich unwichtig.

stimmt genau eher ja eher nein überhaupt nicht

VII. Kurzfeedback

Projektname:

Wie war die heutige Projekteinheit für dich?

Um den Projektunterricht zu verbessern, ist deine Meinung wichtig:

Was hast du heute bei deiner Arbeit gut verstanden?

Was hast du nicht verstanden?

Was sollte an dem Projektteil, den ihr heute bearbeitet habt, verbessert werden?

Danke für deine Anregungen!

VIII. Fotos



Abbildung 1: Gruppenarbeit mit Beobachterin in Projektphase 2 (1)



Abbildung 2: Gruppenarbeit mit Beobachterin in Projektphase 2 (2)



Abbildung 3: Bildbearbeitung durch Histogrammdehnung (Schülerarbeit)