

Die Blätter dürfen in Gruppen von bis zu zwei Personen bearbeitet werden.
Abgabe **individuell** bis zum **19.12.2018, 23:59 Uhr** per Mail an den jeweiligen Tutor und Frederic Raber.
Als Betreff bitte `gdmi-[ROT|GRÜN|BLAU]-Matrikelnummer1-Matrikelnummer2-7` verwenden und
sowohl Namen als auch Matrikelnummern aller Gruppenmitglieder auf der Abgabe vermerken.

Geben Sie bei allen Aufgaben an, auf welche Quellen Sie sich beziehen.

Aufgabe 1: Bilddateien (4 Punkte)

Gegeben sei eine Bilddatei mit einer Auflösung von 1080×720 Pixeln.

1. Berechnen Sie, unter der Annahme, dass die Datei mit den folgenden Parametern unkomprimiert und ohne Farbpalette gespeichert wird, den benötigten Speicherplatz in KB (1 KB = 1000 Byte):
 - (a) Farbauflösung 8 Bit pro Farbe (RGB), 130 ppi.
 - (b) Farbauflösung 4 Bit pro Farbe (RGB), 130 ppi.
 - (c) Farbauflösung 4 Bit pro Farbe (RGB), 72 ppi.
 - (d) Welche Auswirkung haben ppi auf die Dateigröße?
2. Sie wollen eine Datei mit der Auflösung 1080×720 nun in der Größe $21\text{cm} \times 14.0\text{cm}$ ausdrucken. Wie viel ppi können ohne Skalierung des Bildes maximal erreicht werden?
(Anmerkung: 1 Zoll (Inch) = 2.54cm)
3. In welcher Größe (in cm) muss die Datei (1080×720) ausgedruckt werden, um eine horizontale und vertikale Auflösung von jeweils 300 ppi zu erreichen.

Geben Sie bei allen Aufgaben die Rechenwege an!

Aufgabe 2: Anti-Aliasing (4 Punkte)

Wir haben in der Vorlesung gesehen, dass Treppenstufeneffekte (Aliasing) durch unzureichende Abtaststraten zustande kommen können. In dieser Aufgabe versuchen wir nun eine Lösung zu entwickeln um gefundene Kanten zu glätten.

1. Entwerfen Sie einen einfachen Kantenglättungsalgorithmus, der auf ein Pixel angewendet wird und seine 8 Nachbarn dabei zu Hilfe nimmt. Die Farbe des betrachteten Pixel wird dabei auf die Durchschnittsfarbe aller 9 involvierten Pixel gesetzt. Aus Gründen der Einfachheit betrachten wir hierbei die Farbe jedes Pixels als einfachen Grauwert, der als Byte repräsentiert wird.

```
AntiAlias(  
  int hauptPixel, int links, int rechts,  
  int obenLinks, int obenMitte, int obenRechts,  
  int untenLinks, int untenMitte, int untenRechts)  
  ...  
  Gebe die neue Farbe des Pixels zurück
```

2. Wenden Sie ihren Kantenglättungsalgorithmus auf jedes Pixel des folgenden diskreten Bildes an. Beachten Sie dabei, dass die neue Farbe eines jeden Pixels lediglich in das Ergebnisbild geschrieben wird und für die aktuelle Berechnung der Kantenglättung die alten Eingabewerte benutzt werden. Farben von Pixeln außerhalb des Bildes werden mit 0 angesehen.

0	0	0	10	0
0	10	50	50	50
10	20	250	250	100
0	30	50	0	0
0	40	0	0	0

Aufgabe 3: Frequenzen und Schwingungen (5 Punkte)

Geben ist eine Schwingung der Form

$$s(t) = 3 \cdot \sin(2t + \pi).$$

1. Berechnen Sie Hoch- und Tiefpunkt der Schwingung (der Funktion) analytisch.
2. Berechnen Sie den Spitzenwert der Schwingung.
3. Berechnen Sie den Effektivwert der Schwingung für den Zeitraum $T = 2\pi$ nach der Formel

$$s_e = \sqrt{\frac{1}{T} \int_T s^2 dt}.$$

Die Lösungen werden nur gewertet, wenn auch alle Rechenschritte angegeben sind.

Aufgabe 4: Farbmodelle (3 Punkte)

1. Erklären Sie den Unterschied zwischen Additiven und Subtraktiven Farbmischungen und nennen Sie jeweils ein Farbmodell als Beispiel und einen Anwendungsbereich.
2. Erklären Sie den Unterschied zwischen HSV und HSL.
3. Zeichnen Sie eine Skizze für die Farbmodelle von HSV und HSL.

Aufgabe 5: Farbwerte (4 Punkte)

1. Wandeln Sie die beiden folgenden Werte aus dem RGB- in den HSV-Farbraum um:

(a) $(R,G,B) = (0,255,255)$

(b) $(R,G,B) = (42,5,5)$

2. Wandeln Sie die beiden folgenden Werte aus dem HSV- in den RGB-Farbraum um:

(a) $(H,S,V) = (180,0.5,0.75)$

(b) $(H,S,V) = (42,0.125,0.5)$

Gehen Sie dabei nach den Überföhrungsalgorithmen in den Abbildungen 1 und 2 (dargestellt in Pseudocode) vor und geben Sie alle Zwischenschritte an. Sie können Ihr Ergebnis z.B. mit GIMP verifizieren.

```

RGB2HSV (int R, int G, int B)
real Delta; real Min;
real H, S, V;
Min=kleinsten Wert von (R, G, B);
V = größter Wert von (R, G, B);
Delta = V - Min;
Wenn V==0
S = 0;
Sonst
S = Delta / V;
Wenn S==0
H = undefiniert
Sonst
Wenn R==V
H = 60 * (G - B) / Delta
Sonst
Wenn G==V
H = 120+60*(B-R)/Delta
Sonst
H = 240+60*(R-G)/Delta
Wenn H<0
H = H + 360.0t
V = V / 255

```

Abbildung 1: RGB nach HSV

```

HSV2RGB (real H, real S, real V)
real f; real p; real h2;
real q; real t;
int i;
V = 255 * V
Wenn S==0
Wenn H ist undefiniert
R = V;
G = V;
B = V;
Sonst
Fehler in der Farbspezifikation;
Sonst
Wenn H==360.0
H = 0;
h2 = H / 60.0;
i = größter ganzzahliger Wert
kleiner gleich h2;
f = h2 - i;
p = V * (1.0 - S);
q = V * (1.0 - (S * f));
t = V * (1.0 - (S * (1.0 - f)));
Wenn i==0
R = V;
G = t;
B = p;
Sonst
Wenn i==1
R = q;
G = V;
B = p;
Sonst
Wenn i==2
R = p;
G = V;
B = t;
Sonst
Wenn i==3
R = p;
G = q;
B = V;
Sonst
Wenn i==4
R = t;
G = p;
B = V;
Sonst
Wenn i==5
R = V;
G = p;
B = q;

```

Abbildung 2: HSV nach RGB