

### 3. Übungsblatt zur Vorlesung Computergraphik im WS 2017/18

Abgabe am Mittwoch, 15.11.2017 (12:00)

Besprechung am Montag, 20.11.2017 17:30 - 19:00 Uhr

#### Aufgabe 1 *Bildklasse [2 Punkte]*

Um geeignet mit Bildern arbeiten zu können, wird eine Bildklasse benötigt. Diese wurde bereits erstellt und besteht aus einer abstrakten Base-Klasse `image_base` und einer Template-Klasse `image`. Die Klassen sind so konzipiert, dass über das Template entschieden werden kann welcher Farbraum verwendet wird. Aktuell werden RGB, HSV, Graustufen und Schwarz-Weiß unterstützt.

In der Implementierung fehlt noch die Initialisierung und die Berechnung der Indices, um auf die gespeicherten Pixeldaten zuzugreifen. Diese Funktionen müssen von Ihnen implementiert werden. Anweisungen zu deren Funktionsweise finden Sie in den Kommentaren im Quellcode.

#### Aufgabe 2 *Konvertieren: RGB zu Graustufen [2 Punkte]*

In dieser Aufgabe sollen Sie ein RGB- in ein Graustufen-Bild umwandeln. Implementieren Sie dafür in der Datei `Aufgaben.cpp` zunächst das Laden, Konvertieren und Speichern der Bilder mit Hilfe der gegebenen Klassen `image_io` und `image_converter`. Die tatsächliche Konvertierung in der `image_converter`-Klasse muss jedoch noch implementiert werden. Für die Umwandlung in Graustufen, verwenden Sie die Luminanzgleichung aus der Vorlesung.

Zum Testen der Implementierung, starten Sie das erzeugte Programm von der Kommandozeile aus, mit den zwei Parametern für das Eingabebild (.ppm) und Ausgabebild (.pgm). Bei der Aufforderung zur Aufgabenwahl, wählen Sie die 1. Die Bilder können z.B. mit einem Bildbearbeitungsprogramm wie GIMP oder Adobe Photoshop geöffnet werden.

#### Aufgabe 3 *Konvertieren: Graustufen zu Schwarz-Weiß [1 Punkt]*

Nun da Sie ein Graustufen-Bild erzeugen können, soll dieses auch noch in Schwarz-Weiß umgewandelt werden. Implementieren Sie dafür in der Datei `Aufgaben.cpp` zunächst das Laden, Konvertieren und Speichern der Bilder mithilfe der gegebenen Klassen `image_io` und `image_converter`, analog zu Aufgabe 2. Die tatsächliche Konvertierung muss jedoch auch hier noch implementiert werden. Die Umwandlung erfolgt, indem alle Grauwerte mit Wert kleiner als 0,5 zu schwarz (0,0) umgewandelt werden, alle größeren Werte zu weiß (1,0).

Zum Testen der Implementierung starten Sie das erzeugte Programm von der Kommandozeile aus, mit den zwei Parametern für das Eingabebild (.pgm) und Ausgabebild (.pbm). Bei der Aufforderung zur Aufgabenwahl, wählen Sie die 2.

#### **Aufgabe 4**   *Konvertieren: RGB zu HSV und zurück [5 Punkte]*

Da es mehrere Farbräume gibt, sollen diese nun auch ineinander überführbar gemacht werden. Implementieren Sie dazu im `image_converter` die restlichen Funktionen, um RGB in HSV, sowie HSV in RGB, umzuwandeln. Beachten Sie, dass alle Werte immer zwischen 0,0 und 1,0 liegen sollen. Das heißt, dass auch der Winkel des H-Wertes auf den Bereich  $[0, 1]$  abgebildet werden muss. Da die Farbbilder nur im RGB-Format gespeichert werden können, implementieren Sie in der Datei `Aufgaben.cpp` das Laden des Bildes, die Umwandlung von RGB zu HSV und zurück zu RGB und speichern Sie das zwei Mal konvertierte Bild ab. Das soll Ihnen die Möglichkeit geben, den Erfolg der beiden Konvertierungsschritte zu bestätigen.

Zum Testen der Implementierung, starten Sie das erzeugte Programm von der Kommandozeile aus, mit den zwei Parametern für das Eingabebild (.ppm) und Ausgabebild (.ppm). Bei der Aufforderung zur Aufgabenwahl, wählen Sie die 3.