

2. Übungsblatt zur Vorlesung Computergraphik im WS 2025/26

Besprechung am Montag, Dienstag und Mittwoch, 17.11., 18.11 & 19.11.2025

Farbräume und Bildbearbeitung

Um die Bearbeitung der folgenden Aufgaben zu erleichtern, ist im gegebenen Programmskelett eine abstrakte Basisklasse `image_base` und eine Template-Klasse `image` gegeben. Der Typparameter der Template-Klasse `image` entscheidet über den verwendeten Farbraum einer `image`-Instanz. Der Code `cg::image<color_space_t::Gray> picture(WIDTH, HEIGHT)` instanziiert bspw. ein Graustufen-Bild. Aktuell werden die Farbräume RGB, HSV, Graustufen und Schwarz-Weiß unterstützt, die entsprechenden Enum-Werte sind in `ImageTraits.hpp` definiert.

Um ihre Implementierung der folgenden Aufgaben zu testen, starten Sie das (erfolgreich kompilierte) Programm von der Kommandozeile aus. Das Programm erwartet jeweils einen Parameter für das Ein- und Ausgabebild (Dateiendung: `.ppm`, `.pgm` oder `.pbm` für Farb-, Graustufen- oder Schwarz-Weiß-Bilder) und kann wie folgt aufgerufen werden:

Windows:

```
> .\ColorSpaces.exe <PathToImage>\ginkgo.ppm ginkgoGreyscale.pgm
```

Linux oder Mac OS:

```
$ ./ColorSpaces <PathToImage>/ginkgo.ppm ginkgoGreyscale.pgm
```

Die (erzeugten) Bilder können z.B. mit einem Bildbearbeitungsprogramm wie GIMP oder Adobe Photoshop geöffnet werden.

Aufgabe 1 *Konvertieren: RGB zu Graustufen [1 Votierpunkt]*

In dieser Aufgabe sollen Sie ein RGB- in ein Graustufen-Bild umwandeln.

- Implementieren Sie dafür in der Datei `ColorSpaces.cpp` zunächst das Laden, Konvertieren und Speichern der Bilder mit Hilfe der gegebenen Klassen `image_io` und `image_converter`.
- Implementieren Sie anschließend die tatsächliche Konvertierung in der `image_converter`-Klasse.

Verwenden Sie für die Umwandlung in Graustufen die Luminanzgleichung aus der Vorlesung.

Starten Sie nach erfolgreichem Kompilieren das erzeugte Programm mit den zwei Parametern für das Eingabebild (`.ppm`) und Ausgabebild (`.pgm`). Bei der Aufforderung zur Aufgabenwahl wählen Sie die 1.

Aufgabe 2 Konvertieren: Graustufen zu Schwarz-Weiß [1 Votierpunkt]

Nachdem Sie ein Graustufen-Bild erzeugen können, soll dieses auch noch in Schwarz-Weiß umgewandelt werden. Implementieren Sie dafür in der Datei `ColorSpaces.cpp` zunächst das Laden, Konvertieren und Speichern der Bilder mithilfe der gegebenen Klassen `image_io` und `image_converter`. Implementieren Sie anschließend die tatsächliche Konvertierung in der `image_converter`-Klasse. Die Umwandlung soll so erfolgen, dass alle Grauwerte mit einem Wert $< 0,5$ in schwarz (0,0) und alle Werte $\geq 0,5$ in weiß (1,0) umgewandelt werden.

Starten Sie nach erfolgreichem Kompilieren das erzeugte Programm mit den zwei Parametern für das Eingabebild (`.pgm`) und Ausgabebild (`.pbm`). Bei der Aufforderung zur Aufgabenwahl wählen Sie die 2.

Aufgabe 3 Konvertieren: RGB zu HSV und zurück [3 Votierpunkte]

Neben den technikgetriebenen RGB- und CMYK-Farbräumen existiert mit HSV ein Farbraum, welcher sich stärker an der menschlichen Farbwahrnehmung orientiert. Implementieren Sie, um diesen Farbraum nutzbar zu machen, die restlichen Funktionen der Klasse `image_converter` zur Umwandlung von:

- RGB in HSV
- HSV in RGB

Beachten Sie, dass alle Werte zwischen 0,0 und 1,0 liegen sollen. Das bedeutet auch, dass der Winkel des H-Wertes auf den Bereich $[0, 1]$ abgebildet werden muss.

Die Farbbilder können nur im RGB-Format gespeichert werden. Implementieren Sie deshalb in der Datei `ColorSpaces.cpp`

- das Laden des Bildes
- die Umwandlung von RGB zu HSV
- die Umwandlung zurück zu RGB

und speichern Sie das zwei Mal konvertierte Bild ab. Dies soll Ihnen die Möglichkeit geben, den Erfolg der beiden Konvertierungsschritte zu bestätigen.

Starten Sie nach erfolgreichem Kompilieren das erzeugte Programm mit den zwei Parametern für das Eingabebild (`.ppm`) und Ausgabebild (`.ppm`). Bei der Aufforderung zur Aufgabenwahl wählen Sie die 3.

Aufgabe 4 Bildbearbeitung im HSV-Farbraum [3 Votierpunkte]

Bildoperationen wie das Verschieben des Farbtons oder das Ändern der Sättigung sind im HSV-Farbraum einfach umzusetzen. In dieser Aufgabe soll als Beispiel hierfür aus dem gegebenen Bild `ginkgo.ppm` ein Bild mit Color-Key-Effekt erzeugt werden. Implementieren Sie dafür in der Datei `ColorSpaces.cpp` zunächst das Laden, Konvertieren von RGB nach HSV, die Rückkonvertierung und das Speichern der Bilder mithilfe der gegebenen Klassen `image_io` und `image_converter`. Manipulieren Sie das Bild mit Hilfe der Klasse `image_manipulation` im HSV-Farbraum, bevor Sie es zurück nach RGB konvertieren und speichern. Implementieren Sie anschließend die tatsächliche Manipulation in der Datei `ImageManipulation.cpp` entsprechend der Kommentare im Programmcode:

1. Rotieren des Farbwertes h um 30°

2. Setzen des Sättigungswertes s auf 0 für alle Pixel deren rotierter, normalisierter Farbwert h **außerhalb** des Intervalls $[50^\circ, 100^\circ]$ liegt.
3. Setzen des Helligkeitswertes v auf 80 % des vorherigen Wertes für alle Pixel deren rotierter, normalisierter Farbwert h **außerhalb** des Intervalls $[50^\circ, 100^\circ]$ liegt.
4. Setzen des Sättigungswertes s auf 90 % des vorherigen Wertes für alle Pixel deren rotierter, normalisierter Farbwert h **innerhalb** des Intervalls $[50^\circ, 100^\circ]$ liegt.
5. Setzen des Helligkeitswertes v auf 70 % des vorherigen Wertes für alle Pixel deren rotierter, normalisierter Farbwert h **innerhalb** des Intervalls $[50^\circ, 100^\circ]$ liegt.

Starten Sie nach erfolgreichem Kompilieren das erzeugte Programm mit den zwei Parametern für das Eingabebild (.ppm) und Ausgabebild (.ppm). Bei der Aufforderung zur Aufgabenwahl wählen Sie die 4. Abbildung 1 zeigt das erwartete resultierende Bild.



Abbildung 1: Modifiziertes Bild mit Color-Key-Effekt.

Quelle: Tomi Mäkitalo "Ginkgo biloba in autumn near Tokyo Station 秋の行幸通りのイチョウ並木 - panoramio", <https://creativecommons.org/licenses/by-sa/3.0/legalcode>

Aufgabe 5 *Perzeption und Farbe* [2 Votierpunkte]

1. Warum sind nachts alle Katzen grau?
2. Warum ist eine Cyan-ähnliche Schriftfarbe (RGB: 100, 255, 255) auf weißem Hintergrund eine ungeeignete Kombination? Begründen Sie Ihre Antwort mit Hilfe der Luminanzgleichung. Geben Sie Ihren Rechenweg an.

3. Weshalb gibt es Rot-Grün- und Blau-Gelb-Schwäche, aber nicht Rot-Blau-Schwäche?
4. Was sind Metamere? Nennen Sie verschiedene Ursachen für Metamerie.