

# Contents

<b>1</b>	<b>Einleitung</b>	<b>1</b>
1.1	was wird gemacht(gehört nicht ins Inhaltsverzeichnis)	1
1.2	grundliegende Idee	1
1.3	Erklärung Farbräume	1
<b>2</b>	<b>Code</b>	<b>1</b>
2.1	Farbräume	1
2.1.1	RGB	2
2.1.2	HSV	2
2.1.3	Anwendung im Code	3
2.1.4	Vergleich	3
2.2	Saturation/Value weg	4
2.2.1	Problem	4
2.2.2	Lösung	4
2.3	richtige Anzahl Felder pro Farbe	4
2.4	Wahrscheinlichkeit	4
2.4.1	Idee	4
2.4.2	logistische Funktion	4
<b>3</b>	<b>Schluss</b>	<b>4</b>
3.1	Schlussfolgerung	4
3.2	Reflexion	4
3.3	Ausblick	4

## 1 Einleitung

1.1 was wird gemacht(gehört nicht ins Inhaltsverzeichnis)

1.2 grundlegende Idee

1.3 Erklärung Farbräume

## 2 Code

### 2.1 Farbräume

Zuallererst musste festgelegt werden, wie die Farben der Steine aus dem Bild ausgelesen werden. Dafür muss man zuerst verstehen, wie Farben auf

einem Bild überhaupt dargestellt werden. Dies geschieht auf einem Computer mithilfe eines Farbraumes. Ein Farbraum ist eine fest definierte Anzahl an Farben, die oft mithilfe von drei Variablen beschrieben werden. Eine Bilddatei enthält somit nur Farben, die durch eine Kombination der drei Variablen erstellt werden kann.(QUELLENANGABE FARBRAUM) Diese Variablenwerte können dann ohne Probleme ausgelesen und miteinander verglichen werden. Zwei der gängigsten Farbräume, die hier miteinander verglichen werden, sind der RGB- und der HSV-Farbraum.

### 2.1.1 RGB

RGB ist der Standard-Farbraum, für die digitale Bildwiedergabe. Er setzt sich aus drei Werten für Rot, Grün, und Blau zusammen. Die Werte gehen jeweils von 0 bis 255, wobei die Farbe bei 0 nicht vorhanden ist, und bei 255 die Farbe mit voller Intensität leuchtet. (0, 0, 0) ist somit schwarz und (255, 255, 255) ist weiss. Der gesamte Farbraum lässt sich als Würfel in einem Koordinatensystem darstellen, wobei die Achsen jeweils die Intensität einer Farbe beschreiben. (BILD EINFÜGEN) Der RGB-Farbraum ist ein additiver Farbraum, was bedeutet, dass bei nichts angefangen wird, und je höher die Werte werden, desto mehr Farbe vorhanden ist. Deshalb wird RGB auch bei vielen Arten von Beleuchtung verwendet, wie Fernseher und Computerbildschirme, die ihre Pixel mit roten, grünen und blauen LEDs beleuchten. Auch der Mensch erkennt Farben auf diese Weise, da im Auge Rezeptoren vorhanden sind, die empfindlich auf die Wellenlängen von Rot, Grün und Blau sind.

### 2.1.2 HSV

Der HSV-Farbraum setzt sich aus den Werten "Hue", "Saturation" und "Value" zusammen. Hue ist ein Wert für den Farbton, und geht standardmässig von 0 bis 179. Ausserdem ist der Wertebereich von Hue kreisförmig, was heisst, dass 0 und 179 nebeneinander sind und es somit kein Anfang oder Ende gibt. "Saturation" beschreibt die Sättigung und Intensität der Farbe und geht von 0 bis 255, wobei 0 keine Sättigung und somit Weiss bedeutet und bei 255 die Farbe vollends vorhanden ist. "Value" ist ein Mass für die Helligkeit und reicht ebenfalls von 0 bis 255. Wenn der Value bei null ist, dann spielt es keine Rolle was der Farbton ist, und die Farbe wird schwarz. Je höher der Value dann geht, desto heller wird die Farbe. Der HSV-Farbraum wird häufig in Form eines Zylinders visualisiert.(BILD)

### 2.1.3 Anwendung im Code

```
1         col = 0
2         for m in range(20):
3             for n in range(20):
4                 col += \
5                     pics[i][coordinates[3 * i + j][k][0] - 10 + m][
6                         coordinates[3 * i + j][k][1] - 10 + n][1]
7         avgcol[3 * i + j][k][1] = int(col / 400)
```

Zuerst wurden für jedes Feld auf dem Würfel die RGB-Werte von 400 Pixeln ausgelesen, danach die Durchschnitte davon genommen und in einer Liste gespeichert. Anschliessend können die RGB-Werte noch in den HSV-Farbraum umgerechnet werden.

```
1         col = 200000
2         for k in range(6):
3             dist = 0
4             for l in range(3):
5                 dist += (avgcol[i][j][l] - avgcol[k][8][l]) ** 2
6             distances[i][j][k] = mt.sqrt(dist)
7             if dist < col:
8                 col = dist
9             tilecol[i][j] = colors[k]
```

Um eine Seite nun einer Mitte zuzuordnen, werden die Abstände der eigenen Farbwerte zu den Farbwerten einer Mitte berechnet und die Quadrate davon addiert. Dies wird mit allen Mitteln gemacht und schlussendlich wird die Seite derjenigen Mitte zugeordnet, bei der diese addierten Abstände am kleinsten waren.

### 2.1.4 Vergleich

Jetzt, wo die Farbräume und Methodik der Zuordnung klar sind, stellt sich die Frage, was die besten Ergebnisse liefert. Hierfür wurden fünf Ideen ausprobiert:

1. Die RGB-Werte verwenden
2. Die HSV-Werte verwenden
3. Nur Hue und Saturation verwenden
4. Nur Hue und Value verwenden
5. Nur Hue verwenden

## **2.2 Saturation/Value weg**

### **2.2.1 Problem**

### **2.2.2 Lösung**

## **2.3 richtige Anzahl Felder pro Farbe**

## **2.4 Wahrscheinlichkeit**

### **2.4.1 Idee**

### **2.4.2 logistische Funktion**

## **3 Schluss**

### **3.1 Schlussfolgerung**

### **3.2 Reflexion**

### **3.3 Ausblick**