

Projekt: Portrait als Baustein-Mosaik

Projektidee

Eine Firma, die bunte Bausteine für Kinder verkauft, bietet seit einiger Zeit in ihren Geschäften die Möglichkeit, ein Selbstportrait aufzunehmen und dieses als Mosaik aus Bausteinen zusammenzusetzen. Das Foto wird dazu in eine Vorlage aus ca. 4500 Bausteinen in den Farben Weiß, Hellgrau, Dunkelgrau und Schwarz sowie einer Hintergrundfarbe (hier Gelb) umgewandelt. Abbildung 1 zeigt ein Beispiel für eine Bauvorlage.

Die Firma behauptet, das Erstellen der Vorlage geschehe mit ein bisschen Technologie und ganz viel Magie. Mit Ihrem Wissen über den Aufbau von Bildern, der Codierung von Farben und der Implementierung von Algorithmen zur Transformation von Bildern sind Sie jedoch sicher in der Lage, diese Zauberei zu durchschauen und selbst ein Programm zu schreiben, das aus einem Foto eine entsprechende Vorlage erstellt.



Abbildung 1: Vorlage für das Erstellen eines Baustein-Mosaiks

Aufgabe 1: Haben Sie schon erste Ideen, wie der Zaubertrick funktioniert? Sammeln Sie algorithmische Ansätze für das Erstellen eines Bausteinmosaiks aus einem Portraitfoto. Überlegen Sie dazu, welche der Funktionen, die Sie bereits implementiert haben, hier ggf. hilfreich sind. Welche Voraussetzungen sollte das Originalbild erfüllen?

Umsetzung

Zunächst einmal müssen Hintergrund und Person voneinander unterschieden werden. Hierzu bietet es sich an, das Portrait vor einem einfarbigen Hintergrund aufzunehmen, der sich nach Möglichkeit farblich von der Kleidung, Augenfarbe usw. der Person unterscheidet. Ggf. können die Schattierungen des Hintergrunds, die ein Foto immer aufweisen wird, zunächst mithilfe der Zauberstab-Funktion ausgeglichen werden.

Für die Person wird eine Posterisation durchgeführt. Die Bildpunkte müssen in vier Helligkeitsintervalle unterteilt werden, um die Farben Schwarz, Dunkelgrau, Hellgrau und Weiß zuzuordnen.

Zusätzlich werden dabei die Bildpunkte vergrößert, also zum Beispiel ein Quadrat von 25, 49 oder 100 Bildpunkten immer in der gleichen Farbe gefärbt.

Aufgabe 2: Erstellen Sie ein Processing-Programm, das zu einem Portraitfoto die Vorlage für ein Baustein-Mosaik erzeugt.

Hinweise: Abbildung 3 zeigt einen stark vergrößerten Ausschnitt von 5x5 Pixeln aus einem Portraitfoto. Für jedes Pixel ist der RGB-Wert angegeben und es ist entsprechend gefärbt. Folgende Überlegungen können für die Implementierung hilfreich sein. Gehen Sie dabei davon aus, dass diese 25 Pixel zu einem Baustein zusammengefasst werden sollen.

- Überlegen Sie anhand des Beispiels, wie Sie die Helligkeit einer Gruppe von Pixeln definieren und bestimmen möchten. Betrachten Sie ergänzend dazu ein Foto in einem Bildbearbeitungsprogramm. Vergrößern Sie das Bild auf 2000 Prozent, so dass Sie die einzelnen Pixel gut erkennen können. Mithilfe des Farbauswahlwerkzeugs (Pipette) können Sie sich die RGB-Werte der einzelnen Pixel anzeigen lassen.
- Überlegen Sie, wie die Koordinaten der einzelnen Pixel zusammenhängen. Dabei können Sie z. B. von den Koordinaten (x0|y0) für das Pixel in der linken, oberen Ecke oder von den Koordinaten (xm|ym) für das Pixel in der Mitte ausgehen.
- Wie können Sie so durch die Reihung `pixels` iterieren, dass Sie jeweils eine Gruppe von Pixeln betrachten, die zu einem Baustein zusammengefasst werden sollen.

(252, 219, 196) (x0 y0)	(253, 221, 196)	(253, 221, 196)	(252, 219, 196)	(252, 220, 196)
(252, 218, 196)	(253, 220, 197)	(253, 220, 197)	(252, 218, 195)	(251, 219, 195)
(251, 218, 196)	(252, 218, 196)	(252, 219, 197) (xm, ym)	(252, 219, 196)	(252, 218, 195)
(252, 217, 196)	(251, 217, 195)	(252, 217, 195)	(253, 218, 196)	(251, 215, 193)
(252, 217, 196)	(251, 216, 194)	(251, 216, 194)	(252, 217, 195)	(251, 213, 191)

Abbildung 2: stark vergrößerter Ausschnitt aus einem Portraitfoto mit RGB-Werten der Pixel

Optimierung des Ergebnisses und Erweiterungen

Das Ergebnis ist nicht nur abhängig von Ihrem Algorithmus, sondern auch von Ihrem Ausgangsbild. Möglicherweise ist es daher sinnvoll, das Ausgangsbild zuvor mit einem Bildbearbeitungsprogramm oder den bereits selbst implementierten Funktionen vorzubereiten. Werden z. B. Helligkeit und/oder Kontrast verändert, fallen die Pixel ggf. in andere Helligkeitsintervalle.

Des Weiteren können Sie überprüfen, ob Sie eine passende Zuordnung der Helligkeitswerte zu den Bausteinfarben gewählt haben und ob die Größe der Bausteine in einem guten Verhältnis zur Größe des Bildes steht. Vielleicht möchten Sie Ihren Algorithmus auch variabel gestalten, so dass der Anwender die Größe und damit die Anzahl der Bausteine wählen kann. Abbildung 3 zeigt eine Vorlage, bei der die Höhe und Breite der Bausteine nur halb so groß ist, wie in Abbildung 1.

Um die einzelnen Bausteine in der Vorlage besser erkennen und tatsächlich danach bauen zu können, sind Bausteine mit einer Umrandung hilfreich. In Abbildung 4 wurden die einzelnen Bausteine daher grün umrandet.



Abbildung 3: Vorlage mit viermal so viel Bausteinen wie in Abbildung 1.

Weiterhin kann eine Ausgabe ergänzt werden, die den Anwender informiert, wie viele Bausteine pro Farbe benötigt werden.

Wenn Sie schon einmal einen Eindruck von dem Mosaik aus Bausteinen erhalten möchten, können Sie an der Position einer Gruppe aus Pixeln jeweils ein Foto eines Bausteins in der passenden Farbe ausgeben. Ein Beispiel zeigt Abbildung 5.



Abbildung 4: Umrandung der Bausteine zur besseren Unterscheidung der Bausteine.

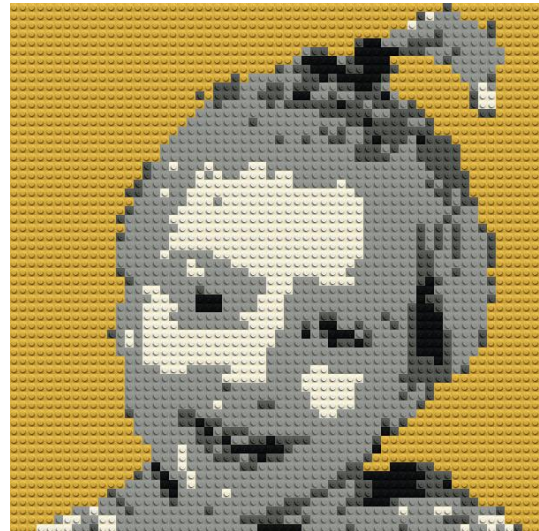


Abbildung 4: Mosaik aus Fotos von Bausteinen als Vorschau

Aufgabe 3: Erweitern und optimieren Sie Ihr Programm. Dazu können Sie die obigen Anregungen nutzen. Sicher haben Sie aber auch noch eigene Ideen.

Lizenz

Dieses Werk ist lizenziert unter einer [Creative Commons Namensnennung - Nicht-kommerziell - Weitergabe unter gleichen Bedingungen 4.0 International Lizenz](#). Sie erlaubt Bearbeitungen und Weiterverteilung des Werks unter Nennung meines Namens und unter gleichen Bedingungen, jedoch keinerlei kommerzielle Nutzung.

Bildnachweise: Die Abbildungen wurden von der Autorin selbst erstellt.