

Algorithmen zur Manipulation von Bildern

Neben dem Zeichnen verschiedener Formen gehört zu den typischen Funktionen von Bildbearbeitungsprogrammen, die Manipulation von Bildern, insbesondere Fotos, in Hinblick auf Helligkeit und Kontrast, das Verändern zu einem Graustufenbild, das Entfernen eines Rotstichs, das Entfernen oder Umfärben eines Bildbereichs usw. Ziel dieses Arbeitsblattes ist es, für einige dieser Funktionen geeignete Algorithmen zu entwerfen.

Grundidee

All den genannten Funktionen gemein ist, dass die RGB-Werte jedes einzelnen Pixels des Bildes betrachtet werden müssen, um diese dann mithilfe geeigneter Rechenoperationen entsprechend zu verändern.

Aufgabe 1: Diskutiert, in wie weit euer Algorithmus für das Erstellen eines Baustein-Mosaiks als Grundlage für andere Algorithmen zur Manipulation von Bildern verwendet werden kann. Welche Anpassungen sind notwendig?

Tatsächlich könnt ihr wesentliche Teile eures Algorithmus übernehmen. Für euer Baustein-Objekt habt ihr bereits die Teilprobleme gelöst, das Bühnenbild systematisch abzulaufen, den RGB-Wert an der aktuellen Position zu ermitteln und einen Abdruck zu hinterlassen. Die entsprechenden Teile des Algorithmus können für ein Pixel-Objekt, welches die Pixel eines Bildes auf andere Weise manipuliert, übernommen werden. Um den Algorithmus zu vervollständigen ergeben sich im Wesentlichen drei neue Teilprobleme, die gelöst werden müssen:

1. Wenn ein Foto z. B. in ein Graustufenbild umgewandelt oder die Helligkeit erhöht werden soll, möchten wir die Auflösung des Bildes erhalten. Das heißt, wir müssen tatsächlich jeden einzelnen Bildpunkt betrachten und das Kostüm des Pixel-Objektes auf die Größe eines einzelnen Pixels verkleinern.
2. Je nachdem, wie wir das Bild verändern möchten, können sich für die veränderten Bildpunkte beliebige RGB-Werte ergeben. Selbst bei der Umwandlung in ein Graustufenbild kommen 256 verschiedene Grautöne in Betracht. Daher können wir unmöglich für jeden möglichen RGB-Wert ein passendes Kostüm vorhalten.
3. Um den neuen RGB-Wert zu bestimmen, müssen wir uns zu der gewünschten Manipulation überlegen, wie wir den neuen RGB-Wert aus dem alten berechnen können.

Problem eins und zwei müssen wir nur einmalig lösen. Problem drei muss für jede gewünschte Manipulation neu betrachtet werden. Da das Verkleinern des Pixels dazu führt, dass es deutlich länger dauert, bis ein Bild vollständig bearbeitet wurde, kümmern wir uns zuerst um Problem zwei und drei. Als Beispiel für Problem drei schauen wir uns das Erzeugen eines Graustufenbildes an (s. Abbildung 1). Wenn wir erfolgreich ein grob gerastertes Graustufenbild erzeugt haben, können wir das Pixel-Kostüm entsprechend verkleinern.



Abbildung 1: Umwandlung eines Farbbildes in ein Graustufenbild

Erzeugen von Abdrücken mit beliebigen RGB-Werten am Beispiel Erzeugen eines Graustufenbildes

Aufgabe 2: Vorbereitung

- Erzeuge eine Kopie deines Snap!-Projektes für das Erstellen des Baustein-Mosaiks.
- Importiere, falls noch nicht geschehen, die Datei *Bildbearbeitung_Bloecke.xml* in dein Projekt.
- Importiere, das Bild, das du bearbeiten möchtest als Hintergrund für die Bühne. Achte darauf, dass die Größe der Bühne der Größe deines Bildes entspricht, damit die Blöcke zum Abfragen der RGB-Werte das richtige Ergebnis liefern.

Nach dem Importieren der Datei *Bildbearbeitung_Bloecke.xml* wird dir im Bereich *Aussehen* ganz unten der Block *faerbe Kostuem* angezeigt (S. Abbildung 2). In diesen Block kann ein

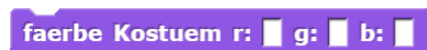


Abbildung 2: Block *faerbe Kostuem*

RGB-Wert eingetragen werden. Das aktuelle Kostüm wird dann vollständig in der entsprechenden Farbe eingefärbt. Dadurch benötigen wir für das Pixel-Objekt nur ein einziges Kostüm in der passenden Größe. Jedes Mal bevor wir einen Abdruck hinterlassen, färben wir es in der passenden Farbe ein. So können wir zur Laufzeit des Algorithmus zu jedem benötigten RGB-Wert ein passendes Kostüm erzeugen.

Zu einem Farbfoto das passende Graustufenbild zu erzeugen, bedeutet im Prinzip jeden einzelnen Bildpunkt auf seine Helligkeit zu reduzieren und den zur Helligkeit passenden Grauton auszuwählen. Die Idee für die Berechnung des passenden RGB-Wertes ist daher ganz ähnlich zum Erzeugen des Baustein-Mosaiks. Die Helligkeit eines Farbtons lässt sich unterschiedlich definieren. Eine Möglichkeit ist, die Helligkeit eines Bildpunktes als den Durchschnitt der Helligkeit des Rot-, des Grün- und des Blauwertes zu ermitteln. Wenn wir diesen durchschnittlichen Helligkeitswert für den Rot-, den Grün- und den Blauwert übernehmen, erhalten wir den passenden Grauton.

Tabelle 1 zeigt zwei Beispielrechnungen:

Farbiger Bildpunkt	Durchschnittliche Helligkeit	RGB-Wert des Grautons
R: 240, G: 123, B: 12	$(240 + 123 + 12) : 3 = 375 : 3 = 125$	R: 125, G: 125, B: 125
R: 60, G: 120, B: 150	$(60 + 120 + 150) : 3 = 330 : 3 = 110$	R: 110, G: 110, B: 110

Tabelle 1: Reduktion eines Farbtons auf den passenden Grauton

Aufgabe 3: Passe das Skript für das Pixel-Objekt so an, dass ein grob gerastertes Graustufenbild erzeugt wird. Verwende dazu den Baustein *faerbe Kostuem*.

Anpassen der Größe des Pixel-Objektes

Für ein Kostüm in der passenden Größe erstellen wir am besten mit einem Bildbearbeitungsprogramm ein Bild der Größe 1 x 1 Pixel. Die Farbe spielt in diesem Fall keine Rolle, da das Kostüm ja mithilfe des Bausteins *faerbe Kostuem* beim Ablaufen des Bildes passend gefärbt wird.

Aufgabe 4: Füge deinem Pixel-Objekt ein Kostüm der Größe eines einzelnen Pixels hinzu. Passe deinen Algorithmus zum systematischen Ablaufen der Bühne an die veränderte Größe des Kostüms an.

Tipp: Schalte vor dem Ausführen des Skriptes den Turbomodus ein.

Rekonstruktion weiterer Funktionen eines Bildbearbeitungsprogramms

Du hast nun ein Grundgerüst für Algorithmen zur Manipulation eines Bildes erstellt. Um weitere Funktionen eines Bildbearbeitungsprogramms zu implementieren, musst du lediglich die Rechenoperationen zum Ermitteln der neuen RGB-Werte anpassen.

Aufgabe 5: Kopiere dein Skript zum Erzeugen eines Graustufenbildes und verändere es so, dass alle Farben des Bühnenbildes invertiert, d. h. umgekehrt werden. Ein Beispiel für ein Originalbild mit dem zugehörigen invertierten Bild siehst du in Abbildung 3.



Abbildung 3: Original und Bild mit invertierten Farben

Hinweis: Wenn du keine Idee hast, welche Rechenoperationen hierzu notwendig sind, schau dir die Datei *AB2_Aufg5_Hinweis* an.

Aufgabe 6: Erstelle nun weitere Skripte, die das Bild auf unterschiedliche Weise verändern. Hier sind einige Ideen, dir fallen aber sicher auch noch andere Möglichkeiten ein:

- Verändern der Helligkeit
- Erhöhen des Kontrastes
- Entfernen eines Rotstichs
- weitere Varianten von Graustufenbildern
- Sepia-Effekt
- Schwarz-Weiß-Bild
- Posterisation
- ...

Hinweise: Überlege zunächst selbst, wie die RGB-Werte rechnerisch so verändert werden können, dass sich der gewünschte Effekt ergibt. Experimentiere dabei mit einem größeren Kostüm für dein Pixel-Objekt, damit dein Skript schneller ausgeführt werden kann.

Wenn du keine Idee hast, welche Rechenoperationen notwendig sind, kannst du ...

- ... verschiedene Rechnungen ausprobieren und dir anschauen, wie sie das Bild verändern.
- ... die Hinweise in den entsprechenden Dateien *AB2_Aufg6_Hinweis_...* zur Hilfe nehmen.

Dein Ergebnis muss nicht perfekt sein! Für manche Funktionen eines Bildbearbeitungsprogramms werden komplexe mathematische Funktionen verwendet, während man einen ähnlichen Effekt schon mit einfachen Rechenoperationen erreichen kann.

Für Fortgeschrittene

Manchmal soll nicht das gesamte Bild, sondern nur ein bestimmter Bereich des Bildes bearbeitet werden. Wenn beispielsweise bei der Aufnahme eines Fotos nur eine Ecke des Bildes nicht ausreichend belichtet wurde, so soll nur dieser Bereich aufgehellt werden.

Aufgabe 7: Erweitere dein Programm so, dass der Anwender die Möglichkeit hat, mithilfe der Maus zwei gegenüberliegende Eckpunkte eines Rechtecks auszuwählen. Die anschließend gewählte Funktion zur Veränderung des Bildes wird dann nur für diesen Bereich ausgeführt.

Im Bereich *Fühlen* findest du dazu hilfreiche Bausteine:

Maus x-Position

Maus y-Position

Maustaste gedrückt?

Abbildung 4: Bausteine zur Abfrage der Maus-Eigenschaften aus dem Bereich Fühlen

Zeichne zunächst das ausgewählte Rechteck auf der Bühne ein. Die dazu notwendigen Überlegungen werden dir helfen, das Pixel-Objekt anschließend innerhalb dieses Bereichs zu bewegen.

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.

Bildnachweis: Abbildung 1 und 3 wurden von der Autorin selbst erstellt.

Abbildung 2 und 4 wurden aus der Entwicklungsumgebung Snap! exportiert. Snap! wird von der University of California, Berkeley zur Verfügung gestellt: <https://snap.berkeley.edu>