

Simulation des Zauberstab-Werkzeugs

Um einen Gegenstand oder ein Portrait freizustellen, können mithilfe des Zauberstab-Werkzeugs alle Pixel in der Hintergrundfarbe ausgewählt und anschließend gelöscht werden. Alternativ kann den ausgewählten Pixeln eine neue Farbe zugewiesen werden. Abbildung 1 zeigt jeweils ein Beispiel. Auch diese Funktion können wir in einem Processing-Programm nachbauen.

Grundidee

Aufgabe 1:

- Beschreiben Sie möglichst genau, was das Zauberstab-Werkzeug leistet. Was bedeutet es, einen Bildpunkt „zu löschen“? Öffnen Sie dazu ggf. ein Foto in einem Bildbearbeitungsprogramm und wenden Sie das Zauberstab-Werkzeug an.
- Welche Gemeinsamkeiten und Unterschiede können Sie im Vergleich mit den Funktionen bzw. Algorithmen feststellen, die Sie bereits implementiert haben?

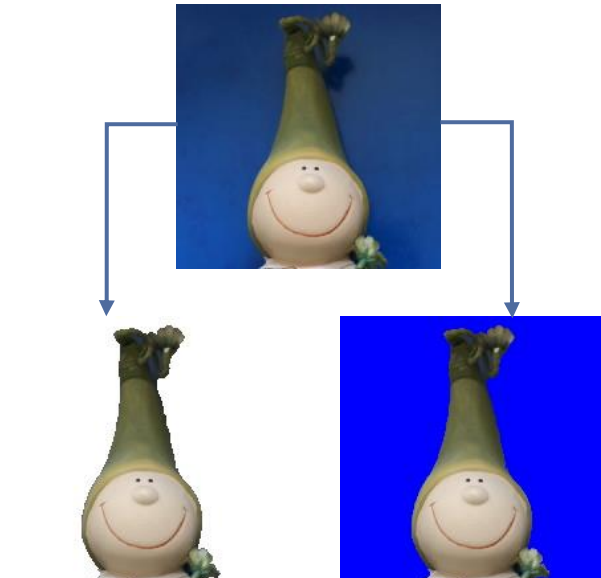


Abbildung 1: Anwendung der Zauberstab-Werkzeugs zum Freistellen bzw. Vereinheitlichen des Hintergrunds.

Im Gegensatz zu den anderen Algorithmen benötigen wir für das Zauberstab-Werkzeug zunächst einen Referenzwert, den der Anwender auswählt. Dazu kann er beispielsweise einen Bildpunkt der gewünschten Farbe im Bild mit der Maus anklicken. Bezogen auf diesen Referenzwert werden dann alle Bildpunkte mit einem ähnlichen Farbton gelöscht. Hier können wir zwei Vereinfachungen vornehmen: Anstatt die entsprechenden Pixel zunächst als Auswahl zu markieren, ersetzen wir jeden Bildpunkt, der zu dem ausgewählten Bildpunkt passt, direkt durch eine vorher festgelegte Farbe, z. B. Weiß. Außerdem führen wir die Ersetzung zunächst für alle Pixel des Bildes mit dem passenden Farbton durch. Wie wir die Funktion auf einen Auswahlbereich oder zusammenhängende Pixel beschränken können, überlegen wir später.

Mit diesen Vereinfachungen können wir eines der bereits erstellten Programme, z. B. zum Erzeugen eines Graustufenbildes, als Grundlage verwenden und für die Implementierung eines Zauberstab-Werkzeugs anpassen.

Auswahl des Referenzwertes

Um einen Referenzwert per Maus auswählen zu lassen, müssen wir den RGB-Wert des Bildes an der Position der Maus bestimmen. Die Position der Maus können wir den Systemvariablen `mouseX` und `mouseY` entnehmen. Um aus der Reihung `pixels` den entsprechenden Farbwert auszulesen, müssen wir uns überlegen, wie die Koordinaten der Maus mit den Indizes der eindimensionalen Reihung `pixels` zusammenhängen.

Das Koordinatensystem hat seinen Ursprung in der linken, oberen Ecke des Programmfensters. Von dort verläuft die x-Achse in positiver Richtung nach rechts und die y-Achse in positiver Richtung nach unten. In Abbildung 2 ist für jedes Pixel der Index in der eindimensionalen Reihung `pixels` eingetragen. Außen können die x- und die y-Koordinate des Pixels bzw. der Maus abgelesen werden.

		X - Koordinate				
		0	1	2	3	4
Y - Koordinate	0	0 = 0+0·5	1 = 1+0·5	2 = 2+0·5	3 = 3+0·5	4 = 4+0·5
	1	5 = 0+1·5	6 = 1+1·5	7 = 2+1·5	8 = 3+1·5	9 = 4+1·5
	2	10 = 0+2·5	11 = 1+2·5	12 = 2+2·5	13 = 3+2·5	14 = 4+2·5
	3	15	16	17	18	19
	4	20	21	22	23	24
	5	25	26	27	28	29

Abbildung 2: Veranschaulichung des Zusammenhangs zwischen den Indizes der Reihung `pixels` sowie der x- und y-Koordinate

Aufgabe 2: Erläutern Sie mithilfe von Abbildung 2, wie der Index eines Bildpunktes in der Reihung `pixels` anhand der x-Koordinate, der y-Koordinate und der Breite des Bildes bestimmt werden kann. Ergänzen Sie die Rechnungen für die untere Hälfte des Bildes und geben Sie eine allgemeine Gleichung an.

Aufgabe 3: Erstellen Sie ein Programm, das für jeden Bildpunkt, der mit der Maus angeklickt wird, den RGB-Wert ausgibt. Speichern Sie den RGB-Wert in einer globalen Variablen, um ihn später als Referenzwert für die Zauberstab-Funktion verwenden zu können.

Vergleichen von RGB-Werten

Um die Funktion des Zauberstab-Werkzeugs zu implementieren, muss für jedes Pixel des Bildes entschieden werden, ob der Farbton mit dem ausgewählten Referenzwert übereinstimmt. Daher müssen wir uns zunächst Gedanken darüber machen, unter welchen Bedingungen wir zwei Pixel als farblich übereinstimmend bezeichnen wollen.

Aufgabe 4:

- Wie können zwei RGB-Werte miteinander verglichen werden?
- Laden Sie in Ihr Programm aus Aufgabe 3 ein Foto, z. B. aus der beiliegenden Datei `Foto_Meer_600x800.jpg`. Wählen Sie mit der Maus verschiedene Bildpunkte aus, die Sie als farblich übereinstimmend bezeichnen würden und vergleichen Sie die RGB-Werte.
- Entscheiden Sie für die zwei RGB-Werte in Tabelle 1 jeweils, ob sie im Sinne des Zauberstab-Werkzeugs als farblich übereinstimmend bezeichnet werden sollten.













RGB-Wert des Referenzpunktes	RGB-Wert eines anderen Bildpunktes	Sind die RGB-Werte farblich übereinstimmend?
 (240, 40, 20)	 (240, 40, 20)	
 (240, 40, 20)	 (230, 45, 22)	
 (45, 155, 230)	 (155, 230, 45)	
 (100, 200, 200)	 (50, 100, 100)	
 (45, 155, 230)	 (65, 135, 250)	
 (150, 78, 10)	 (150, 78, 200)	

Tabelle 1: Vergleich von Farbtönen anhand ihrer RGB-Werte

- d) Die meisten Bildbearbeitungsprogramme bieten bei der Verwendung des Zauberstab-Werkzeugs die Möglichkeit, einen Toleranzwert festzulegen. Erläutern Sie die Bedeutung dieses Wertes für die Funktion des Zauberstab-Werkzeugs.

Aufgabe 5:

- a) Implementieren Sie basierend auf Ihren Erkenntnissen aus Aufgabe 4 eine Operation *vergleicheRGBWerte(rgb1: color, rgb2: color, toleranz: Ganzzahl): Wahrheitswert*. Die Operation erhält als Parameter zwei RGB-Werte, die miteinander verglichen werden sollen. Die Operation gibt *wahr* zurück, wenn die RGB-Werte als farblich übereinstimmend bezeichnet werden können, sonst *falsch*. Der Toleranzbereich, in dem RGB-Werte als farblich übereinstimmend gewertet werden, soll mithilfe des Parameters *toleranz* festgelegt werden. Legen Sie einen sinnvollen Wertebereich für den Parameter *toleranz* fest.
- b) Testen Sie Ihre Funktion, indem Sie den Anwender beispielsweise mit der rechten Maustaste einen Referenzwert auswählen lassen und für jeden Bildpunkt, der mit der linken Maustaste ausgewählt wird, ausgeben, ob die Bildpunkte farblich übereinstimmen. Dabei können Sie den Anwender zusätzlich einen Toleranzwert festlegen lassen.

Implementierung des Zauberstab-Werkzeugs

Die Vorbereitungen sind abgeschlossen. Ihnen stehen jetzt alle algorithmischen Konzepte zur Verfügung, um die Funktion des Zauberstab-Werkzeugs zu implementieren.

Aufgabe 6:

- a) Verwenden Sie Ihre Ergebnisse aus Aufgabe 3 und Aufgabe 5, um die Funktion des Zauberstab-Werkzeugs zu implementieren. Färben Sie alle Pixel des Bildes, die farblich mit dem ausgewählten Referenzpunkt übereinstimmen, weiß.
- b) Erweitern Sie Ihr Programm um die Möglichkeit, dass der Anwender den Toleranzbereich und/oder die Farbe, in der die ausgewählten Pixel gefärbt werden, bestimmen kann.

Häufig möchte man, dass das Zauberstab-Werkzeug nur auf einen Bereich des Bildes angewendet wird, um beispielsweise nur die Pixel des Hintergrundes umzufärben, nicht aber Pixel der gleichen Farbe, die zum abgebildeten Objekt gehören. Dazu könnte man den Anwender mithilfe der Maus einen rechteckigen Bereich aufziehen lassen, auf den die Bearbeitung des Bildes begrenzt wird. Alternativ können ausgehend vom ausgewählten Referenzpunkt nur die farblich übereinstimmenden Pixel ausgewählt werden, die mit dem Referenzpunkt direkt oder über andere farblich übereinstimmende Pixel verbunden sind. Abbildung 5 und 6 zeigen jeweils ein Beispiel.

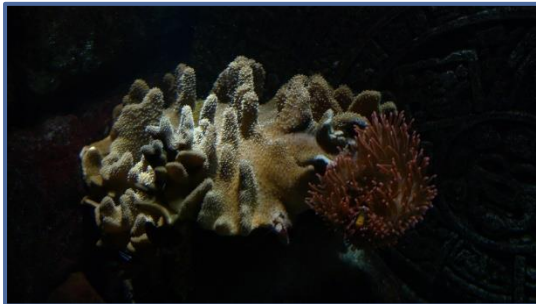


Abbildung 3: Original einer Koralle



Abbildung 4: Anwenden des Zauberstab-Werkzeugs auf das gesamte Bild

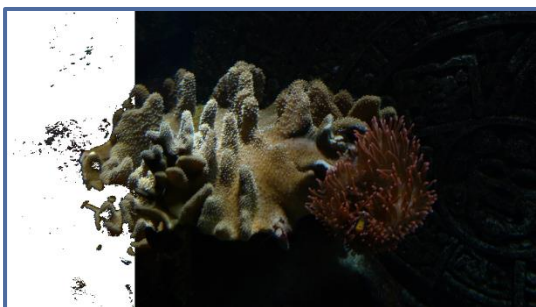


Abbildung 5: Anwenden des Zauberstab-Werkzeugs auf einen rechteckigen Auswahlbereich



Abbildung 6: Anwenden des Zauberstab-Werkzeugs auf einen zusammenhängenden Bereich

Beschränkung der Zauberstab-Werkzeugs auf einen rechteckigen Auswahlbereich

Betrachten wir zunächst die Variante, einen rechteckigen Auswahlbereich festzulegen. Abbildung 7 zeigt, wie ein Rechteck mithilfe von zwei gegenüberliegenden Ecken definiert werden kann.

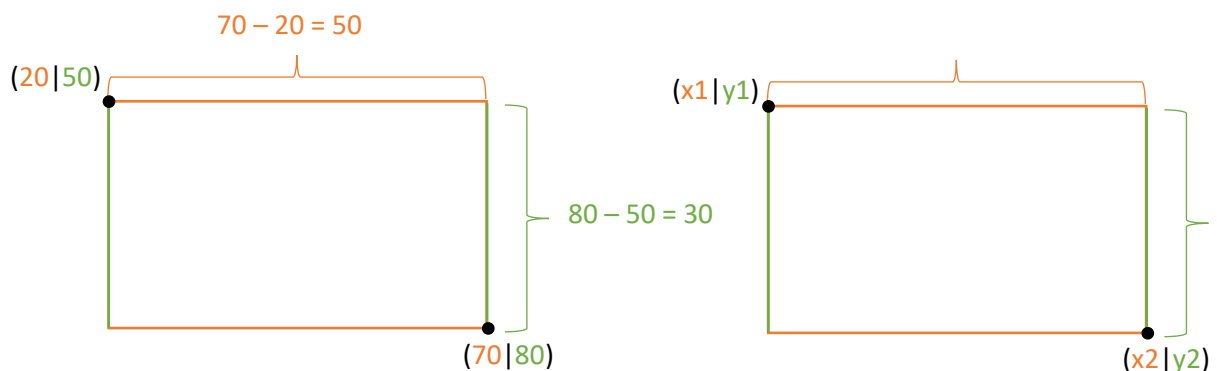


Abbildung 7: Berechnung der Höhe und Breite eines Rechtecks anhand der Koordinaten zweier gegenüberliegender Ecken

Aufgabe 7:

- a) Erläutern Sie mithilfe von Abbildung 7, warum ein Rechteck durch zwei gegenüberliegende Ecken eindeutig definiert ist.
- b) Geben Sie die Werte aller Parameter an, die der Operation `rect()` übergeben werden müssen, damit ein Rechteck mit den gegenüberliegenden Ecken (20|50) und (70|80) gezeichnet wird.
- c) Ergänzen Sie in Abbildung 7 die Berechnung der Breite und Höhe des Rechtecks für die allgemeinen Koordinaten (x1|y1) und (x2|y2).
- d) Entscheiden Sie für die zwei Punkte in Tabelle 2 jeweils, um welche Ecken eines Rechtecks es sich handelt. Erläutern Sie Ihr Vorgehen.
- e) Entwickeln Sie einen Algorithmus, der zu zwei beliebigen gegenüberliegenden Eckpunkten eines Rechtecks die Koordinaten des linken, oberen Eckpunktes und des rechten, unteren Eckpunktes sowie die Breite und Höhe des Rechtecks bestimmt.

Punkt A	Punkt B
(23, 87)	(49, 103)
(67, 193)	(40, 100)
(35, 100)	(65, 80)
(166, 290)	(33, 130)
(233, 40)	(120, 99)

Tabelle 2: Koordinaten der gegenüberliegenden Ecken eines Rechtecks

Aufgabe 8:

- a) Ergänzen Sie in Ihrem Programm aus Aufgabe 6 die Möglichkeit, ein Rechteck mit der Maus aufzuziehen. Das heißt, die erste Ecke des Rechtecks wird festgelegt, wenn die Maustaste gedrückt wird. Die zweite Ecke wird festgelegt, wenn die Maustaste wieder losgelassen wird.

Hinweise:

- Um das Aufziehen des Rechtecks von der Auswahl des Referenzpunktes zu unterscheiden, kann es hilfreich sein, die Funktionen zwei verschiedenen Maustasten zuzuordnen.
 - Um dem Anwender die Auswahl zu erleichtern, kann nach dem Herunterdrücken der Maustaste fortlaufend das Rechteck gezeichnet werden, das ausgewählt wird, wenn die Maustaste an der aktuellen Position losgelassen wird. Um dabei eine Ansammlung von Rechtecken zu vermeiden, kann zuvor mithilfe der Operation `updatePixels()` jedes Mal das Bild neu in das Fenster gezeichnet und damit das alte Rechteck überzeichnet werden.
 - Speichern Sie die Koordinaten der Eckpunkte des Rechtecks für Aufgabenteil b).
- b) Passen Sie Ihre Implementierung des Zauberstab-Werkzeugs so an, dass nur die Pixel im rechteckigen Auswahlbereich bearbeitet werden.

Hinweise:

- Da die Bildpunkte in der Reihung `pixels` bei der linken, oberen Ecke des Bildes beginnen und bei der rechten, unteren Ecke des Bildes enden, ist es sinnvoll die Koordinaten der gegenüberliegenden Ecken zunächst so zu sortieren, dass festgelegt ist, in welchen Variablen die Koordinaten der linken, oberen Ecke und in welchen Variablen die Koordinaten der rechten, unteren Ecke enthalten sind.
- Obwohl die Reihung `pixels` eindimensional ist, kann es für die Berücksichtigung des rechteckigen Auswahlbereichs hilfreich sein, zwei geschachtelte Zählschleifen zu verwenden. Während beispielsweise die äußere Zählschleife in den Grenzen der y-Koordinate arbeitet, kann die innere Schleife in den Grenzen der x-Koordinate arbeiten.
- Um die Funktion auch weiterhin auf das ganze Bild anwenden zu können, ist eine Reset-Funktion hilfreich, welche die Ecken des Auswahlbereichs auf die linke, obere und die rechte, untere Ecke des gesamten Bildes setzt.

Begrenzen des Zauberstab-Werkzeugs auf zusammenhängende Pixel

Für die Begrenzung des Zauberstab-Werkzeugs auf einen rechteckigen Auswahlbereich, haben wir uns an den x- und y- Koordinaten der Bildpunkte anstelle der Indizes der Reihung `pixels` orientiert. Diese Betrachtungsebene ist auch hilfreich, wenn wir die Bildpunkte, die mit einem Startpunkt zusammenhängen, ermitteln wollen. Zunächst wäre zu entscheiden, welche Pixel als zusammenhängend definiert werden. Ausgehend von dem blauen Pixel in Abbildung 8 können das nur die grünen Pixel oder zusätzlich die orangen Pixel sein.

Aufgabe 9: Geben Sie für die Pixel, die Sie als zusammenhängend mit dem blauen Startpixel bezeichnen möchten, die Koordinaten in Abhängigkeit von den Koordinaten x und y des blauen Startpixels an.

Aufgabe 10: Sammeln Sie algorithmische Ansätze, um ausgehend von einem Startpixel alle zusammenhängenden Pixel, die farblich mit dem Startpixel übereinstimmen, auszuwählen. Vergleichen und diskutieren Sie die Ansätze anschließend.

Aufgabe 11: Wählen Sie einen der Ansätze aus Aufgabe 10 aus, um die Begrenzung des Zauberstab-Werkzeugs auf zusammenhängende Pixel zu implementieren.

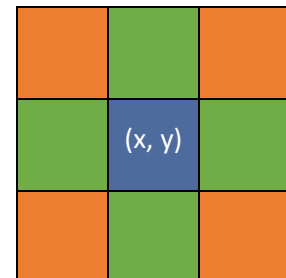


Abbildung 8: Nachbarn eines Pixels

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.