

# Methoden zur automatisierten Farbgestaltung von Webseiten aus Bildvorlagen

## Masterprojekt Zwischenabgabe II

Philipp Anders

20. März 2017

Ziel des Projekts ist die automatische Farbkonfiguration der Oberflächenelemente von Webseiten (Text, Buttons, Hintergründe, etc.) aus Bildvorlagen. Dabei wird das Problem in zwei Teilprobleme zerlegt: 1. Das Bilden einer Obermenge von Farben durch einen Algorithmus zur Color Palette Estimation (CPE). 2. Die Identifizierung von Farben für bestimmte Oberflächenelemente aus dieser Obermenge durch Lösung eines Constraintsystems. Für die CPE wird der ACoPa-Algorithmus von Petro [21] ausgewählt und implementiert, da dessen Ergebnis den aus Styleguides bekannten Farbpaletten-Definitionen in Form von Color Swatches ähnelt.

## 1 Einleitung

### 1.1 Problemmodellierung

Ziel dieser Arbeit ist die automatisierte Bestimmung der Farbwerte von Elementen in Webseiten. Der CSS-Standard zur Beschreibung der Formatierung von HTML-Dokumenten definiert mehr als 10 Eigenschaften zur farblichen Gestaltung, wobei einige spezifisch für bestimmte HTML-Elemente sind [32]. In dieser Arbeit werden die Farbeigenschaften von zwei verschiedenen Arten von Oberflächenkomponenten betrachtet:

1. **Texte:** Elemente mit transparentem Hintergrund (`background-color: transparent`) und definierter Vordergrundfarbe (`color`).

<h1>, <a>, etc.) betrachtet. Elemente, die visuell als Blöcke wahrgenommen werden (d.h. die Hintergrundfarbe "background-color" von <div>, <button> etc.) und (2) Es wird also die Farbbestimmung von **Blöcken** und **Texten** fokussiert. Rahmen, Textdekoration und andere färbbare Komponenten werden vernachlässigt.

Üblicherweise werden mehrere Elemente eines HTML-Dokuments auf die gleiche Farbe abgebildet. Beispielsweise können alle Links (Textfarbe) sowie Buttons (Hintergrundfarbe) in Blau dargestellt werden. Im Folgenden wird eine Menge von Elementen einer Webseite mit gleicher Farbabbildung als **Color Group**  $CG$  bezeichnet [siehe auch 16, 12].

Die Menge aller Color Groups einer Webseite wird als  $CGs$  formalisiert. Es gilt  $|CGs| = k$ . Eine Webseite mit  $k$  Color Groups tritt also mit  $k$  Farben in Erscheinung. Abbildung 1 veranschaulicht dies anhand einer exemplarischen Webseite für Musik-Streaming. (a) zeigt ein generisches Farbdesign, während (b) die sechs Color Groups separat hervorhebt.

Webdesigner verkleinern Suchraum zur Identifizierung geeigneter Farben für die Color Groups durch die Zusammenstellung einer sogenannten **Farbpalette** [16, 10, 26]. Dabei handelt es sich um eine Farbmeng  $P = \{c_1, c_2, \dots, c_n\}$  mit  $n$  Farben, wobei jedes  $c \in P$  als String von RGB Werten kodiert wird. Formal ist damit das Ziel dieser Arbeit die Ermittlung einer Abbildung  $CGs \rightarrow P$ . Die Abbildung muss injektiv sein, d.h. nicht jede Farbe der Farbpalette  $P$  muss verwendet werden, jedoch dürfen

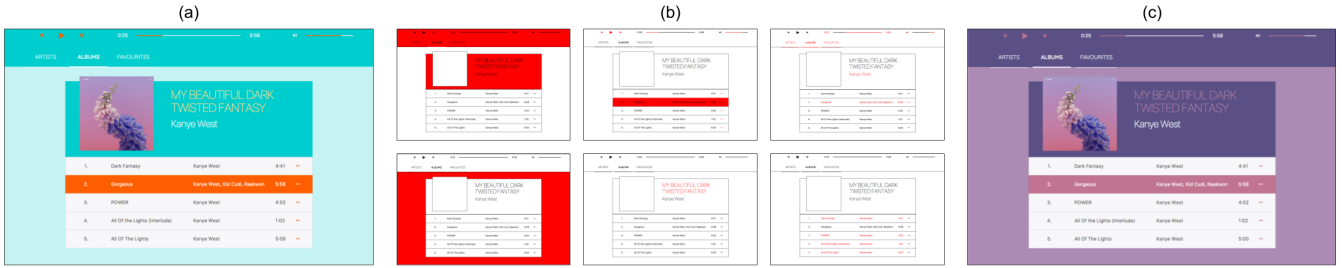


Abbildung 1: Beispiel der Farbgestaltung einer exemplarischen Weboberfläche für Musik-Streaming. (a) Generische Farbgestaltung ohne Anpassung an das Albumcover. (b) Separate Hervorhebung der sechs Color Groups. Elemente einer Color Groups sind jeweils rot dargestellt. (c) Beispiel einer Farbgestaltung mit Anpassung an das Album Cover.

mehrere Color Groups nicht gleich gefärbt werden. Es gilt  $k \leq n$ .

Entscheidendes Kriterium für die Zuordnung ist eine funktionale Gestaltung der Webseite. Im Rahmen dieser Arbeit wird hierunter eine Farbgestaltung verstanden, die Lesbarkeit und Benutzerführung gewährleistet. Eine intuitiv widersinnige Färbung ist beispielsweise ein roter Text auf orangem Hintergrund mit grauem Button. Weder ist der Text lesbar, noch wird die Aufmerksamkeit des Nutzers auf das Interaktionselement gelenkt.

Als Besonderheit dieser Arbeit soll die Farbpalette auf einer Bildvorlage basieren. So wird eine Harmonisierung des visuellen Eindrucks einer Weboberfläche und einer darin präsenten Grafik erreicht. Da die gewählte Farbgebung wesentlich für die vermittelte Atmosphäre einer Webseite ist [10], soll die Anpassung der Farbgebung an ein Motiv dessen Eindruck unterstützen. Da die Farbpalette somit auf den verarbeiteten Daten basiert, ist eine Berechnung zur Laufzeit möglich, wenn ein effizientes Suchverfahren umgesetzt wird. Ein Anwendungsbeispiel hierfür ist die farbliche Anpassung einer Seite für Musikstreaming an das gespielte Albumcover. Abbildung 1 (c) zeigt hierfür ein Beispiel.

## 1.2 Color Palette Estimation

Die Zusammenstellung einer Farbpalette aus einer Bildvorlage wird von Petro [21] als **Color Palette Estimation** (CPE) bezeichnet und als die Repräsentation eines Bildes mit einer minimalen Menge von Farben beschrieben. „Minimal“ bedeutet nach Auffassung der Autoren, dass redundante Farben reduziert

und die seltenen Farben der für die Wahrnehmung wichtigen Objekte erhalten bleiben. Formale Kriterien werden hierfür jedoch nicht geliefert. Abbildung 2 veranschaulicht diese intuitive Definition am Beispiel eines Bildes mit einem Marienkäfer, dessen Sichtbarkeit von der Wahl der Farbpalette abhängt.

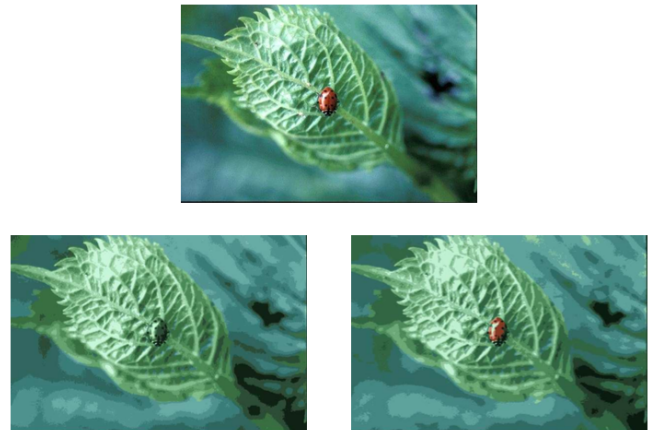


Abbildung 2: Beispiel für die Einfärbung eines Bildes mit unterschiedlichen Farbpaletten der Größe 12. Oben: Originalbild. Links: Farbpalette ohne rote Farbtöne. Rechts: Farbpalette mit roten Farbtönen, wodurch der Marienkäfer erkennbar ist (Quelle: [21])

Historisch geht die CPE aus der Farbquantisierung hervor, bei der die Farben von Grafiken aufgrund der damals zu kleinen Kapazität von Grafikpuffern vor deren Anzeige reduziert (Farbreduktion) und dann auf die reduzierte Farbpalette abgebildet werden mussten (Quantisierung) [24]. Aus diesem Kontext kommt das formale Kriterium der Summe des quadratischen Fehlers, welcher in diesem Anwendungsfall auch als *Recoloring Error* bezeichnet wird [11].

Da Grafikpuffer mittlerweile über ausreichend Kapazität verfügen, liegt die Anwendung der CPE in anderen Bereichen wie z.B. der farbbasierten Indizierung von Grafiken in Datenbanken oder der Zusammenstellung von Farbthemen zu Gestaltungszwecken. Hanrahan [11] zeigen, dass in letzterem Kontext der Recoloring Error keine geeignete Metrik zur Beurteilung der Güte einer Farbpalette in Bezug auf die Bildvorlage ist. Grund sind die menschlichen Wahrnehmungseigenschaften, wobei Bilder auf Komponenten- und nicht auf Pixelebene erfasst werden. Stattdessen werden eine Reihe anderer Metriken vorgestellt, die diesen Umstand berücksichtigen. Die Autoren zeigen zusätzlich empirisch, dass abhängig vom Individuum ein und dieselbe Farbpalette eines Bildes für unterschiedlich repräsentativ gehalten wird.

Dieser Befund hebt hervor, dass die Güte einer Farbpalette in Bezug auf die Bildvorlage subjektiv ist und vom Anwendungsbezug abhängt. Aus diesem Grund wird für die Beurteilung der zu ermittelnden Farbpalette keine objektive Bewertungsfunktion herangezogen. Stattdessen wird die Zweckmäßigkeit der Farbpalette in Hinblick auf die farbliche Gestaltung von Webseiten fokussiert. Das bedeutet, dass die aus der Bildvorlage hervorgehende Farbpalette eine Teilmenge an Farben beinhalten muss, mit welcher die automatisierte Farbgestaltung einer Webseite lösbar ist. Hierfür müssen Kriterien abgeleitet werden, indem Prinzipien für eine funktionale Gestaltung von Webseiten sowie Farbdefinitionen aus Style Guides analysiert werden. Davon ausgehend wird ein geeigneter Algorithmus zur Bestimmung einer solchen Farbpalette aus einer Bildvorlage ermittelt.

### 1.3 Einordnung und Literaturbesprechung

Die Arbeit ordnet sich in das Gebiet der automatisierten Farbgestaltung (Color Design Automation) ein. In diesem Bereich hat in den letzten Jahren Forschung in verschiedenen Anwendungsbezügen stattgefunden, wobei überwiegend Methoden des maschinellen Lernens zur Problemlösung eingesetzt wurden.

Hertzmann [14] haben 2011 eine Grundlage für datengetriebene Farbgestaltungssysteme gelegt, indem sie ein Regressionmodell zur Bewertung der Farbkompatibilität von 5 Farben entwickelt haben, d.h. die Bewertung der Farbharmonie von fünf-farbigen

Paletten. Hierfür wurde ein Training an den Datenbeständen von Farbpaletten-Communities wie z.B. Adobe Color<sup>1</sup> durchgeführt. Dabei hat sich unter anderem ergeben, dass die auf geometrischen Strukturen im Farbkreis beruhenden Modelle der klassischen Farbtheorie Farbharmonien nicht zufriedenstellend voraussagen können und in bestimmten Fällen sogar kontraproduktiv sind. Hierzu zählen zum Beispiel die Farbton-Schablonen von Itten [15] oder Birren [6]. Dies ist insofern bemerkenswert, als dass diese Modelle in zahlreichen Unterstützungswerkzeugen zur Farbgestaltung ihre Umsetzung finden. Die Grenzen des trainierten Modells liegen in der Bewertung der Harmonie von Farben mit unterschiedlicher räumlicher Ausprägung [16, 12]. Eine Matlab-Implementierung des Modells zur Überprüfung von Farbharmonien steht öffentlich zum Download bereit <sup>2</sup>.

Hanrahan [12] haben sich 2013 mit der Lösung einer grundlegenden Form eines Färbungsproblems auseinandergesetzt: Die Kolorierung von Mustern nach dem Prinzip "Malen nach Zahlen". Hierfür wurde ein probabilistisches Modell entwickelt, indem über 8000 von Künstlern entworfene Muster von der Seite <http://www.colourlovers.com/> ausgewertet wurden. Die Autoren stellen Kriterien zur Bestimmung von Hintergrund- und Vordergrundfarben sowie zur Kontrastberechnung vor, die für diese Arbeit genutzt werden können. Zur Lösungssuche wird ein *Factor Graph* genutzt, was ebenfalls ein Ansatz für die vorliegende Arbeit darstellt. Das Modell von Hertzmann [14] wird als externer Bestandteil des Graphen hinzugefügt, um eine globale Kompatibilität der eingesetzten Farben zu gewährleisten. Eine Anwendung, welche die Autoren vorschlagen, stellt die Umkehrung des Ziels dieser Arbeit dar: Die Anpassung eines Musters an das Farbschema einer Webseite.

Lou [16] haben sich 2016 mit der automatisierten Farbgestaltung von Webseiten auseinandergesetzt. Durch die Auswertung von 500 Webseiten wurde ein probabilistisches Modell in Form eines Optimierungsproblem mit mehreren Zielfunktionen entwickelt. Eine der Zielfunktionen gewährleistet ausreichenden Kontrast zwischen den Oberflächenelementen. Eine andere Zielfunktion passt die Farbgestaltung an ein Schlüsselwort an (z.B. „Business“ oder „Fresh“). Die letzte Zielfunktion wird durch das Modell von Hertz-

<sup>1</sup><https://color.adobe.com/de/explore/>

<sup>2</sup><http://www.dgp.toronto.edu/~donovan/color/>

mann [14] realisiert und gewährleistet Farbharmone. Die Optimierung wird durch eine lexikographische Strategie umgesetzt, bei welcher in Interaktion mit einem Gestalter die Zielfunktionen nacheinander angewendet werden. Die verwendeten Farbpaletten stammen von der Adobe Color Seite. Als alternatives Anwendungsbeispiel extrahieren die Autoren eine Farbpalette aus einer Grafik und nutzen diese als Grundlage zur Färbung der Webseite. Zur CPE wurde der K-Means Algorithmus verwendet. Im Gegensatz zum Ziel dieser Arbeit erfordert der vorgestellte Prozess jedoch eine Nutzerinteraktion aufgrund der lexikographischen Strategie bei der Optimierung. Somit handelt es sich bei der vorgestellten Lösung um ein Unterstützungswerkzeug und nicht um ein System zur vollautomatisierten Bestimmung einer Farbgestaltung.

Allebach [2] haben sich 2013 mit der automatisierten Farbgestaltung von Magazin-Covers auseinandergesetzt. Vergleichbar mit [16] wird auch hier die Optimierung des Farbkontrasts, der Farbharmone und der Farbsemantik verfolgt. Im Gegensatz zu den bisherigen Lösungen wird hier allerdings mit expliziten Regeln zur Bewertung von Farbharmone- [15] und Kontrasten anstatt mit Modellen gearbeitet, die sich aus Trainingsdaten ableiten. Über Flowcharts vermitteln die Autoren Lösungsprozeduren zur Suche geeigneter Schriftfarben und Stellen konkrete Grenzwerte vor, die für diese Arbeit übernommen werden können.

## 2 Problemlösungsansatz und Systemarchitektur

Ausgehend von der Literaturbesprechung wird ein Ansatz zur Problemlösung besprochen. Darauf aufbauend wird eine konkrete Architektur des zu implementierenden Systems entwickelt. Anschließend werden für die einzelnen Teilaufgaben der Architektur geeignete Methoden besprochen und ausgewählt.

### 2.1 Regelbasierter oder probabilistischer Ansatz

Lou [16] stellen für die automatisierte Farbgestaltung zwei grundlegende Herangehensweisen vor:

1. **Regelbasiert:** Beschreibt quantitative Modelle mit deterministischem Regelwerk. Die Arbeit von Allebach [2] stellt hierfür ein Beispiel dar. Durch die Analyse von Farbharmone-Modellen wurden Regeln für die automatisierte Gestaltung von Magazin-Covers abgeleitet.
2. **Datengetrieben:** Beschreibt Modelle, welche die Performanz möglicher Lösungen einer automatisierten Farbgestaltung auf Grundlage existierender Beispieldaten vorhersagt.

Die regelbasierten Modelle treffen feste Aussagen im Gegensatz zu probabilistischen Modellen, tendieren jedoch zur starken Vereinfachung des Problems und abstrahieren damit vom Anwendungsbezug. Im Gegensatz dazu stützen sich die Modelle des datengetriebenen Ansatzes auf reale Beispiele und tendieren daher zu robusteren Ergebnissen in der Anwendungsdomäne [16].

Da bei den probabilistischen Modellen alle potentiellen Lösungen bewertet und gegeneinander abgewogen werden müssen, sind hohe Laufzeiten möglich. Die bereits entwickelte Lösung von Lou [16] zur automatisierten Farbgestaltung von Webseiten benötigt selbst nach einer Optimierung des Suchverfahrens, bei der unwahrscheinliche Lösungen frühzeitig ausgeschlossen werden, 2 Stunden zur Konvergenz. Außerdem ist deren vorgestellter Prozess zur Anpassung der Farbgestaltung einer Webseite an eine Bildvorlage nicht im eigentlichen Sinne automatisiert: Einerseits erfordert die Verwendung des K-Means Algorithmus zur CPE die Eingabe der Farbzahl ( $k$ ) vom Gestalter, andererseits erfordert die lexikographische Strategie eine Nutzerinteraktion während des Optimierungsprozess.

Aus diesen Gründen wird sich diese Arbeit auf einen **regelbasierten** Ansatz **ohne Nutzerinteraktion** konzentrieren. Auch ohne die Auswertung großer Datenmengen existieren quantitative Modelle, um die Accessibility einer Webseite zu gewährleisten. Beispielsweise geben die *Web Content Accessibility Guidelines* [1] quantitative Grenzwerte für die Lesbarkeit von Texten an. Es sind Eigenschaften und Bedingungen zu erarbeiten, die für Farben einzelner Color Groups sowie zwischen diesen gelten sollen. Diese Einschränkungen werden im Folgenden als **Constraints** bezeichnet. Ein System von Constraints mit endlichem Wertebereich (in diesem Fall  $P$ ) wird als

**Constraint Satisfaction Problem** (CSP) bezeichnet [33]. Allebach [2] haben gezeigt, dass eine Lösung eines solchen Constraint-Systems beispielsweise über Faktor-Graphs möglich ist.

## 2.2 Layouts

Nicht jede Webseite hat die gleiche Anzahl Color Groups. Um die Komplexität des Constraint-Systems zu reduzieren, wird die Festlegung der Color Groups einer Webseite aus der Betrachtung entfernt. Dafür wird von einer Webseite als **Layout** abstrahiert. Für ein Layout wird als konstant angenommen, welche Color Groups mit welcher Semantik existieren. Beispielsweise definiert ein Layout eine Color Group, welche alle Buttons beinhaltet und deren Semantik *Interaktion* ist. Aus Sicht des Constraint-Systems werden keine Farben für einzelnen Buttons, sondern eine Interaktionsfarbe gesucht. Durch das Konzept der Color Groups wird somit von den betreffenden Elementen abstrahiert.

## 2.3 Aufteilung der Suchverfahren

Zur Lösungssuche mit einem regelbasierten Ansatz existieren zwei Herangehensweisen:

1. **Constraints-First:** Setze  $n = k$ . Hierbei wird die Lösungssuche zum Zeitpunkt der CPE verlagert. Unter Kenntnis der Color Groups ist hier der Suchraum das Histogramm der Bildvorlage. Dieser Ansatz wird unter anderem von Hertzmann [14] verfolgt. Sie verwenden ihr Regressionsmodell zur Bewertung der Farbharmonie, um eine alternative Optimierungsfunktionen zur Suche von  $P$  im Farbraum einer Bildvorlage zu formulieren.
2. **Constraints-Last:** Setze  $k \leq n$ .  $P$  wird unabhängig vom Layout ermittelt. Der Suchraum beschränkt sich dann auf die Elemente in  $P$ . Dieser Ansatz wird unter anderem von Obrador [19] bei der automatisierten Farbgestaltung von Dokumenten verfolgt. Nach der Bestimmung einer Farbpalette aus einer Bildvorlage werden sukzessive geeignete Farben aus  $P$  ausgewählt. Zuerst wird die Hintergrundfarbe des Dokuments komplementär zum Hintergrund des Bildes gewählt,

welche via Bildsegmentierung bestimmt wird. Daraufhin wird die Textfarbe unter Beachtung des Kontrastes aus  $P$  gewählt.

Für diese Arbeit wird die Herangehensweise **Constraints-Last** bevorzugt. Sie beinhaltet die Aufteilung des Problems in zwei getrennte Suchverfahren: Ein Suchverfahren für die Bestimmung von  $P$  und ein Suchverfahren zur Bestimmung der Abbildung  $CGs \rightarrow P$ . Dadurch wird eine Wiederverwendung einmal bestimmter Farbpaletten für verschiedene Layouts ermöglicht. Für die Arbeit bedeutet dies, dass eine Problemlösung anhand eines exemplarischen Layouts mit bestimmten Color Groups gezeigt wird, eine Übertragung auf andere Layouts jedoch möglich ist. Schlussendlich ermöglicht ein Cashing bereits berechneter Farbpaletten das Einsatzpotential der automatisierten Farbgestaltung zur Laufzeit, da nur noch eines des Suchverfahren durchgeführt werden muss.

Selbst unter den gegebenen Einschränkung des Suchraums durch die vorgeschaltete Ermittlung einer Farbpalette ist die Menge potentieller Lösungen mit  $\binom{n}{k}k!$  nach wie vor groß. Als Beispiel sei ein Layout mit 5 Colorgroups (z.B. Text, Texthintergrund, Buttons, Navigations-Hintergrundfarbe und Footer-Hintergrundfarbe) sowie eine Farbpalette mit 10 Farbe gegeben. Das bedeutet  $|CGs| = k = 5$  und  $|P| = n = 10$ , wodurch sich  $\binom{10}{5}5! = 30.240$  mögliche Kombinationen ergeben. Es ist ein effektives Suchverfahren zur Ermittlung von  $CGs \rightarrow P$  zu identifizieren. Der bereits vorgestellte, hierarchische Ansatz von Obrador [19] stellt hierfür eine Inspiration dar, bei welchem entscheidende Farbzusordnungen zuerst getroffen werden und dadurch der Suchraum für folgende Color Groups eingeschränkt wird (z.B. erst Hintergrund, dann Textfarbe).

## Zusammenfassung

Abbildung 3 fasst den Ablauf der automatisierten Farbgestaltung einer Webseite zusammen. Das zu entwickelnde System bildet eine Bildvorlage auf eine Farbpalette ab. Das Suchverfahren hierfür wird was als Color Palette Estimation (CPE) bezeichnet. Die entstandene Palette wird zum Suchraum möglicher Farben von Color Groups, welche durch ein Layout definiert werden. Da bei dieser Abbildung der

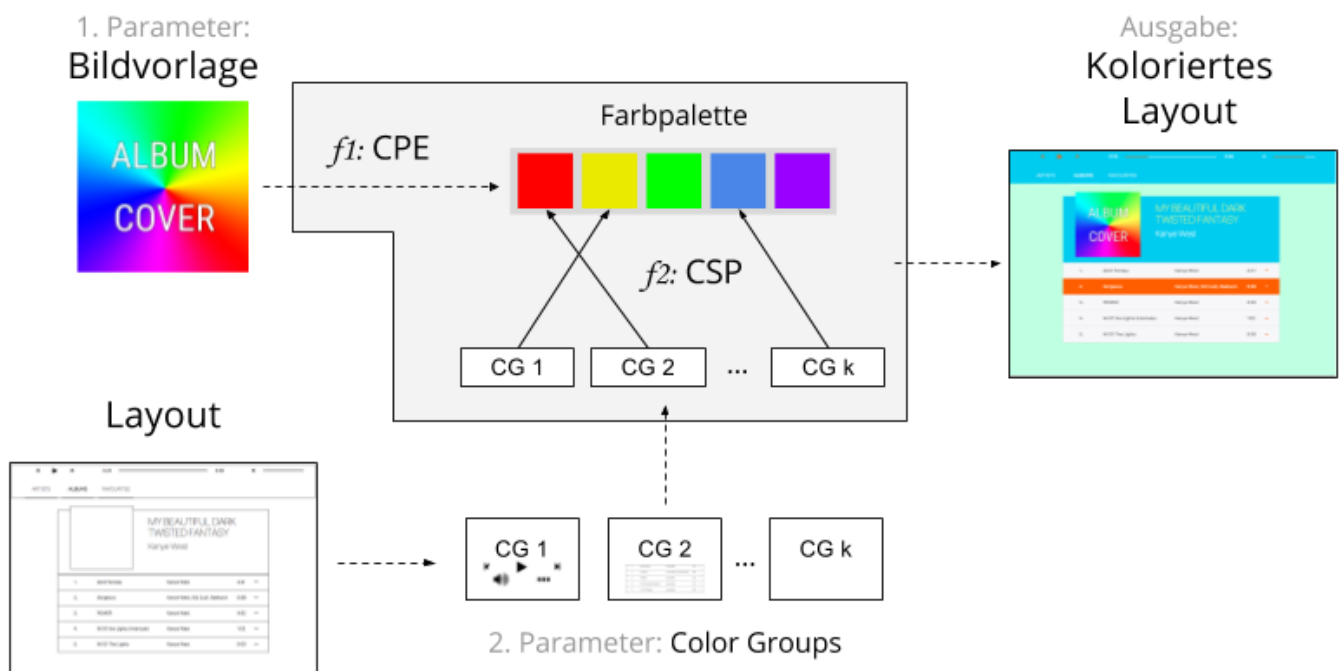


Abbildung 3: System-Architektur. Der grau unterlegte Kasten visualisiert den Bereich, für den algorithmische Lösungen gefunden werden müssen. Die Eingabeparameter sind eine Bildvorlage und eine Menge von Color Groups, welche von einem Layout definiert werden. Das System setzt zwei Suchverfahren um, welche mit  $f_1$  und  $f_2$  bezeichnet werden.  $f_1$  ermittelt zuerst eine Farbpalette aus der Bildvorlage (CPE).  $f_2$  ordnet Colour Groups Farben aus der Palette, in dem ein entsprechendes Constraint System gelöst wird. Nachdem eine Farabbildung gefunden wurde, wird das kolorierte Layout ausgegeben.

Wertebereich einer Color Group dementsprechend beschränkt ist, handelt es sich um ein Constraint Satisfaction Problem (CSP). Letztendlich wird eine kolorierte Variante des Layouts ausgegeben.

Im Folgenden sind konkrete Verfahren für die beiden Teilprobleme CPE und CSP zu implementieren. Ausgangspunkt hierfür sind Prinzipien für eine funktionale Gestaltung von Webseiten. Diese werden im Folgenden durch Webdesign-Literatur sowie die Analyse von Farbdefinitionen aus Style Guides ermittelt.

### 3 Farbgestaltung von Webseiten

Die Ziele der farblichen Gestaltung im Rahmen dieser Arbeit sind Lesbarkeit und Benutzerführung. Im Folgenden werden Gestaltungsprinzipien für diese beiden Ziele herausgearbeitet und als Grundlage für die weiteren Entscheidungen zur Problemlösung betrachtet.

#### 3.1 Lesbarkeit

Eine Auswertung von über 100.000 Webseiten hat ergeben, dass Text fast ausschließlich in Graustufen dargestellt wird [27]. Die populärsten Farben sind Schwarz, Dunkelgrau und Weiß. Hahn [10] begründet dies damit, dass Text schlechter lesbar wird, je farbiger er ist. Der Style Guide von Googles Material Design [8] hebt hervor, dass Aufgrund des Simultankontrasts die Verwendung von Grau als Textfarbe bei stark farbigem Hintergrund zu unerwünschten Effekten führt, wie Abbildung 4 veranschaulicht. Stattdessen wird in Abhängigkeit von der Luminanz des Hintergrunds die Verwendung von Schwarz bzw. Weiß mit einem Alpha-Wert (Transparenz) empfohlen. Konkret werden folgende Werte genannt:

1. Dunkle Schrift: `rgba(0, 0, 0, 0.87)`
2. Helle Schrift: `rgba(255, 255, 255, 1.0)`

Im Folgenden wird die Textfarbe mit Ausnahme von Fällen, in denen die Schrift aus stilistischen Gründen bunt erscheinen soll, auf diese Werte festgelegt. Es ist zu entscheiden, wann die dunkle und wann die helle Textfarbe verwendet wird. Hierfür geben die Web Content Accessibility Guidelines (WCAG) des

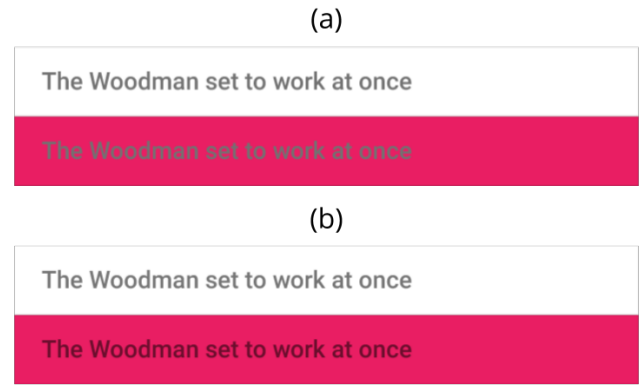


Abbildung 4: Empfohlene Textfarbe bei farbigem Hintergrund. (a) Textfarbe `rgb(114, 114, 114)`. Der Text ist aufgrund des Simultankontrasts unangenehmer zu lesen. (b) Textfarbe `rgba(0, 0, 0, 0.54)`, d.h. Schwarz mit einem Alphawert (Transparenz) von 54%. Die Textfarbe ergibt sich zu einer Schattierung der Hintergrundfarbe und vermeidet so den Simultankontrast. (Quelle: [8])

W3C konkrete Grenzwerte für Kontrastverhältnisse in Abhängigkeit von der Text- und Hintergrundfarbe sowie der Textgröße an. Die Formel lautet zur Berechnung des Kontrastverhältnisses  $L_c$  lautet [30]:

$$L_c(c_1, c_2) = \frac{L(c_1) + 0.05}{L(c_2) + 0.05}, \quad (1)$$

$$L(c_1) \geq L(c_2)$$

mit  $L_c \in [1, \dots, 22]$ , wobei  $L_c(\text{weiss}, \text{schwarz}) = 22$  und  $c_1 = c_2 : L_c(c_1, c_2) = 1$ . Die Berechnungsvorschrift der normalisierten relativen Luminanz einer Farbe  $L$  ist [31] zu entnehmen. Es gelten zwei verschiedene Grenzwertklassen [1]. Die Grenzwertklasse AA beschreibt die Minimalanforderung an lesbaren Text:

$$AA : L_c(c_1, c_2) \geq \begin{cases} 3.0, & \text{Schriftgr.} \geq 18pt \\ & \geq 14pt \wedge \text{fett} \\ 4.5, & \text{sonst} \end{cases} \quad (2)$$



Die Klasse AAA enthält höhere Grenzwerte und beschreibt verstärkte Lesbarkeit:

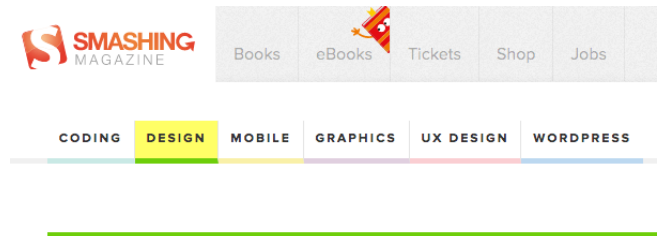
$$AAA : L_c(c_1, c_2) \geq \begin{cases} 4.5, & \text{Schriftgr.} \geq 18pt \vee \\ & \geq 14pt \wedge \text{fett} \\ 7, & \text{sonst} \end{cases} \quad (3)$$

### 3.2 Benutzerführung

Ein adäquater Farbeinsatz ermöglicht die Steuerung von Aufmerksamkeit und unterstützt die Bildung von Zusammenhängen [10]. Dieses Konzept wird auch als visuelle Hierarchy bezeichnet [5], bei welchem die Oberflächenelemente strukturiert und priorisiert werden. Hahn [10] unterteilt folgende Möglichkeiten zur Benutzerführung durch Farbeinsatz:

- **Themengruppen:** Bedeutet die farbliche Kodierung inhaltlich definierter Bereiche. Hierbei handelt es sich um eine vom Gestalter erfundene, subjektive Farbsystemlogik, die kein System, sondern eine Gestaltungsstruktur darstellt, die dem Anwender zu lernen aufgezwungen wird. Dies resultiert automatisch in Buntheit, welche Aufmerksamkeit erregt und somit kontraproduktiv für die Bildung einer visuellen Hierarchie ist. Aus diesen Gründen ist von dieser Variante abzuraten. Abbildung 5 verdeutlicht dies am Beispiel eines Online-Magazins.
- **Funktionsbereiche:** Bedeutet die farbliche Kodierung von Bereichen entsprechend ihrer Funktion. So sind beispielsweise Navigations-, Inhalts- und Servicebereich farblich differenzierbar. Diese Variante ist zu empfehlen, wenn nicht mehr als drei Farben verwendet werden und die betreffenden Bereiche gleichzeitig und somit im Bezug zueinander sichtbar sind. Abbildung 6 verdeutlicht dies am Beispiel einer farblichen Abgrenzung von Navigationsbereichen.
- **Funktionen und Zustände:** Bedeutet die farbliche Kodierung von funktionstragenden Elementen oder deren Funktionszustände. Beispielsweise sind anklickbare Elemente (z.B. Links und Buttons) durch eine einheitliche Farbe signalisierbar. Abbildung 7 zeigt hierfür ein Beispiel

Elemente mit destruktiver Funktion, z.B. Buttons zum Löschen oder Abbrechen einer Aktion, können einheitlich rot gefärbt werden. Wurde eine Farbe einmal mit einer Funktion belegt, sollte sie einheitlich verwendet werden. Beispielsweise würde ein nicht-anklickbares, blaues Element in der Oberfläche aus Abbildung 7 würde verwirren.



#### Category: Design

Abbildung 5: Farbkodierung von Themengruppen am Beispiel von [www.smashingmagazine.com/](http://www.smashingmagazine.com/). Es ist unklar, weshalb die gewählten Farben mit den jeweiligen Bereichen assoziiert werden.

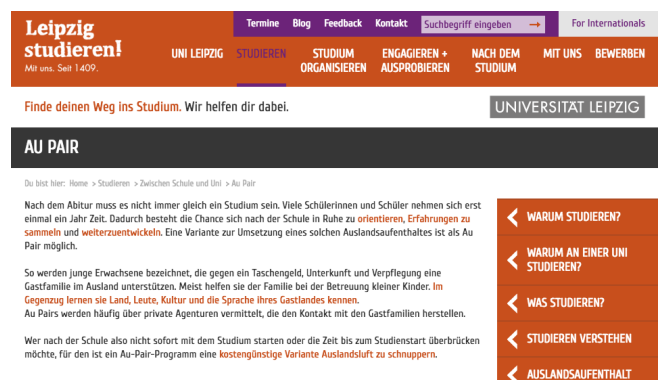


Abbildung 6: Farbkodierung von Funktionsbereichen am Beispiel von [www.leipzig-studieren.de](http://www.leipzig-studieren.de). Navigationsbereiche (orange) werden zueinander in Bezug gebracht und vom Inhalt (weiß) abgegrenzt.

### 3.3 Farbfunktionen

Aus Unterabschnitt 3.2 geht hervor, dass Elemente entsprechend ihrer Funktion gefärbt werden, nicht aus rein dekorativen Gründen oder zur thematischen Strukturierung. Aus diesen Gründen



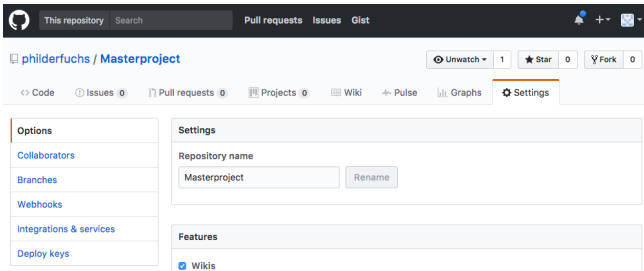


Abbildung 7: Farbkodierung von Funktionen am Beispiel von [www.github.com](https://www.github.com). Nur anklickbare Elemente sind blau, aktive Elemente werden zusätzlich braun markiert.

wird zwischen die Abbildung  $CGs \rightarrow P$  eine Abstraktionsschicht mit der Bezeichnung **Farbfunktionen**  $F$  geschoben, welche beschreibt, welche Funktion eine Color Group in einer Oberfläche erfüllt. In Anlehnung an [8, 7] wird exemplarisch  $F = \{\text{Primär, Sekundär, textInteraktion, Akzent, Text Hell, Text Dunkel, Hintergrund}\}$  definiert. Die einzelnen Farbfunktionen lauten wie folgt:

- **Primärfarbe:** Wird in Relation zu den anderen Farben am häufigsten eingesetzt und bildet so die farbliche Grundstimmung der Oberfläche [25].
- **Sekundärfarbe:** Unterstützt die Farbstimmung der Primärfarbe und orientiert sich dementsprechend an deren Farbton. Durch die Wahl einer unbunteren Farbe tritt sie in der visuellen Hierarchie stärker in den Hintergrund [5], eignet sich besser als Hintergrund für Fließtext und unterstützt durch den Qualitätskontrast die Wirkung der Primärfarbe [10].
- **Interaktionsfarbe:** Signalisiert Elemente, mit denen eine Interaktion möglich ist, z.B. klicken. Dies trifft nicht zwangsläufig auf Elemente zu, bei denen der Nutzer eine Interaktionsmöglichkeit aufgrund ihrer Funktionsgruppe bereits erwartet, wie z.B. in der Seitennavigation.
- **Akzent:** Signalisiert Elemente mit hoher Priorität in der visuellen Hierarchie. Dies ist erreichbar durch einen relativ seltenen (Quantitätskontrast, [10]) Einsatz einer bunten [5] Farbe.
- **Text Hell/Dunkel:** Farbe für Fließtext. Entsprechend Unterabschnitt 3.1 wird hierfür in

Abhängigkeit Hintergrund Weiß (Hell) bzw. Schwarz (Dunkel) mit Alpha-Kanal festgelegt.

- **Hintergrund:** Standardfarbe von Blöcken, die keiner der obigen Farbfunktionen entsprechen. Dies betrifft vorwiegend Blöcke mit Fließtext, so dass Textlesbarkeit zu beachten ist. Dementsprechend sind Graustufen zu bevorzugen [26], wobei entsprechend Unterabschnitt 3.1 ein ausreichender Luminanzkontrast zu gewährleisten ist. Darum wird die Hintergrundfarbe analog zur Textfarbe auf Weiß bzw. Schwarz festgelegt.

Die hier verwendeten Farbfunktionen sind exemplarisch. Abhängig von den individuellen Ansprüchen einer Webseite ist die Definition weiterer Farbfunktionen denkbar. Bei mobilen Anwendungen ist beispielsweise eine farbliche Differenzierung verschiedener Interaktionsformen sinnvoll (z.B. drücken und Wischen). Die Abbildung zwischen Color Groups und Farbfunktionen wird als die Funktion  $S$  definiert und im folgenden als **Farbschema** definiert. Es gilt:

$$S : F \rightarrow P \quad (4)$$

Diese Definition eines Farbschemas unterscheidet sich von der Verwendung des Begriffs in der Literatur [10]. In dieser ist ein Farbschema eine Charakterisierung einer Farbpalette. Es beschreibt, *wie viele* unterschiedliche Farbtöne eine Farbpalette enthält und in welcher *geometrischen Beziehung* diese zueinander stehender. Beispielsweise bezeichnet ein *triadisches* Farbschema eine Farbpalette mit drei Farbtönen, die sich in einem Abstand von  $120^\circ$  zueinander befinden. Genauer wird in *monochromatische, komplementäre (duale), triadische* und *tetraedische* Farbschemen unterschieden, welche Farbpaletten mit einem, zwei, drei oder vier verschiedenen Farbtönen repräsentieren. [3, 7, 8] empfehlen jedoch die Beschränkung auf höchstens 3 Farben im Webdesign, während Graustufen für die Darstellung von Text und Hintergründen dominieren. Dementsprechend werden die Begriffe **monochrom**, **dual** und **triadisch** für die Charakterisierung des Farbschemas als Farbabbildung in dieser Arbeit adaptiert.

Abbildung 8 zeigt hierfür Beispiele. Oben wird eine exemplarische *Farbpalette* abgebildet. Die *Farbfunktionen* entsprechen der in diesem Abschnitt definierte Menge  $F$ . Es ist zu erkennen, dass die Textfarben sowie die Hintergrundfarbe bereits pauschal definiert

werden. Damit realisiert jedes Farbschema im Rahmen dieser Arbeit einen **Bunt-Unbunt-Kontrast**, was nach [26] eine allgemeine Empfehlung für Webdesign darstellt. Die Abbildung zeigt weiterhin mögliche Abbildungen zwischen den Farbfunktionen und der Farbpalette unter *Farbschemata*. Ein monochromes Farbschema beschränkt sich auf einen Farbton in verschiedenen Schattierungen. Dadurch wird ein **Qualitätskontrast** gebildet, welcher die Wirkung der Farbe mit dem höchsten Buntheitswert erhöht. Die inverse Variante des monochromen Schemas veranschaulicht eine Variante mit allgemein dunkler Farbstimmung. Das duale Farbschema besitzt zwei Farbtöne. Da Interaktions- und Akzentfarben weniger häufig auftauchen als die Elemente der Primär- und Sekundärfarben wird hierdurch ein **Quantitätskontrast** gebildet, der die seltenere Farbe zusätzlich betont und so zur Bildung einer visuellen Hierarchie beiträgt. Das exemplarische tertiäre Farbschema bildet die Interaktions- und Akzentgruppen auf unterschiedliche Farben ab und differenziert diese so zusätzlich.

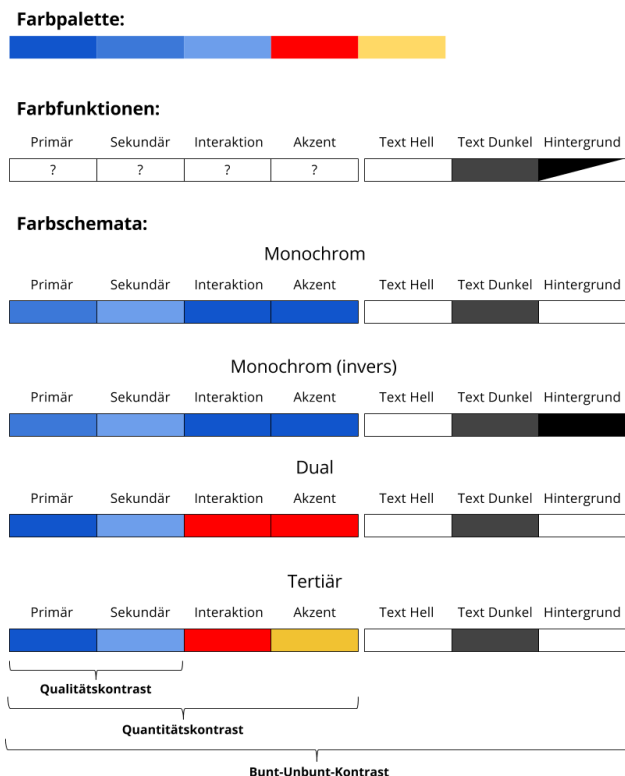


Abbildung 8: Beispiele für monochrome, duale und tertiäre Farbschemata. Der Farbeindruck der Seite wird durch Quantitäts-, Qualitäts- und Bunt-Unbunt-Kontraste bestimmt.

## Zusammenfassung

Zur Unterstützung der Benutzerführung werden in Webseiten Funktionsbereiche und Elemente mit einheitlicher Funktion bzw. Funktionszustand farblich kodiert. Dementsprechend werden für eine Webseite eine Menge an Farbfunktionen  $F$  definiert, wie z.B. die Interaktionsfarbe, welche klickbare Elemente signalisiert. Ein Farbschema  $S : F \rightarrow P$  ist eine Abbildung der Farbfunktionen auf die Farben einer Palette  $P$ . Eine Abbildung soll höchstens drei verschiedene Farbtöne enthalten. Für Fließtext sowie dessen Hintergrund werden standardmäßig Graustufen festgelegt. Zur Benutzerführung realisieren Webseiten somit einen Bunt-Unbunt- sowie Quantitätskontrast, welcher das zentrale Gestaltungsmittel des Screendesigns darstellt [10, 26].

## 4 Algorithmen zur Color Palette Estimation

Im Folgenden wird eine Algorithmus zur Lösung des Teilproblems der Ermittlung der Farbobermenge  $C_s$  gesucht. Hierzu findet eine Betrachtung von Typen vorhandener Algorithmen zur CPE statt. Abschließend wird ein geeigneter Algorithmus ausgewählt

### 4.1 Überblick

Grundlegend sind zwei Ansätze zur CPE zu unterscheiden:

1. **Histogramm-basiert:** Algorithmen, die nur auf dem Histogramm des Bildes arbeiten und somit die Positionsinformationen der Farben nicht beachten. Es handelt sich (bis auf Ausnahmen) um Clustering-Verfahren, die durch eine Partitionierung des Farbraums Gruppen ähnlicher Farben im Histogramm identifizieren.
2. **Bildsegmentierungs-basiert:** Algorithmen, die durch eine Segmentierung des Bildes zunächst zusammenhängende Komponenten identifizieren und für diese dann repräsentative Farben identifizieren.

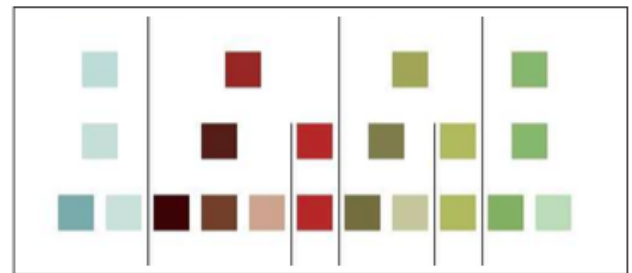
Bildsegmentierungs-basierte Algorithmen berücksichtigen die menschlichen Wahrnehmungseigenschaften auf Komponentenebene, führen aber durch die zusätzliche Betrachtung der Positionsinformation eine weitere Komplexitätsebene ein [11].

Uyyanonvarab [28] treffen eine Kategorisierung der Histogramm-basierten Verfahren in *hierarchisch* und *iterativ*. Hierarchisch arbeitende Algorithmen zur CPE werden auch als *Pre-Clustering Verfahren* bezeichnet, da sie vor dem Erreichen der (fest zu wählenden) Farbanzahl  $n$  mit mehr bzw. weniger Farben starten. Sie basieren auf der statistischen Analyse der Verteilung der Bildfarben im Farbraum. In diese Kategorie fallen *top-down* bzw. *bottom-up* Clustering-Algorithmen. Zu den Top-Down Verfahren zählen die in der Vergangenheit populären Raumunterteilungs-Algorithmen wie z.B. Mediancut [13] oder Octree[23]. Sie Zerteilen den Farbraum sukzessiv in disjunkte Teilräume und unterstellen den Clustern dabei eine Würfelform. Ergebnis der Verarbeitung ist ein Dendrogram, wobei die Blätter die Farben Farbpalette repräsentieren. Ein Schnitt des Dendograms entspricht einer Partitionierung des Raums, welche jedoch auch direkt durch die iterativ arbeitenden Algorithmen erreichbar ist [21]. Diese Verfahren werden darum auch als *partitionierend* [21] oder auch *Post-Clustering* [28] bezeichnet. Sie starten bereits mit der erforderlichen Anzahl Farben  $n$  und verbessern diese iterativ. Einige Methoden dieser Klasse verwenden den quadratischen Fehler, wie z.B. K-Means [17, 9] oder Fuzzy C-Means [4]. Andere analysieren das Histogramm auf dichte bzw. weniger dichte Regionen, wie z.B. Mean-Shift [18]. Eine detailliertere Vorstellung von Algorithmen zur CPE bietet [20].

Petro [21] kritisieren an den bisherigen Algorithmen, dass die Anzahl gesuchten Farben  $n$  zuvor bekannt sein muss, dass die Ergebnisse abhängig von der Initialisierung sind und dass Farben kleiner Bild-details im Sinne der Definition in Abschnitt 1.1 nur unzureichend repräsentiert werden, wie im Paper experimentell nachgewiesen wird. Aus diesem Grund stellen sie den **Automatic Color Palette (ACoPa)** Algorithmus vor, welcher durch die Analyse von Spitzen des Histogramms im HSI Raums eine Farbpalette erstellt und dabei deren Größe selbstständig bestimmt. Der Algorithmus ermittelt dabei zunächst die grundlegenden Farbtöne (Hue) des Bildes und schlüsselt diese daraufhin sukzessive nach deren Sättigungen (Saturation) und Schattierungen (Intensity)



(a)



(b)

Abbildung 9: CPE Ergebnis von ACoPa. (a) Originalbild "Peppers" (b) Hierarchische Farbpalette. Die unterste Ebene zeigt die finalen Farben. (Quelle: [21])

auf. Abbildung 9 veranschaulicht exemplarisch die hierarchische Arbeitsweise, bei der in jeder Ebene zusätzliche Sättigungen und Schattierungen der enthaltenen roten und grünen Farbtöne gebildet werden.

## Zusammenfassung und Wahl des Algorithmus zur CPE

Der Algorithmus zur CPE soll eine Obermenge  $C_s$  von Farben bilden, aus welcher im einem nachfolgenden Schritt eine Teilmenge von Farben  $C$  entsprechend ihrer Eignung für bestimmte Oberflächenelemente ausgewählt werden. Analog dazu werden Farbpaletten in Styleguides als Obermenge von Farben beschrieben, aus welcher der Designer eine Untermenge von Farben für die konkrete Oberfläche auswählt. Bestimmte Styleguides erweitern dabei das Farbpalettenkonzept um Color Swatches, bei welchen Farbtöne in zusätzliche Schattierungen aufgefächert werden. Dadurch hat der Designer eine größere Flexibilität beim Einsatz der Farbpalette.

Aus diesen Gründen wird der ACoPa Algorithmus von Petro [21] zur CPE gewählt. Da er Farbwerte automatisch in verschiedenen Sättigungen und Schattierungen ermittelt, initiiert er die Farbdefinition in Form von Color Swatches in Styleguides. Durch seine parameterfreie Arbeitsweise ermittelt er selbstständig die Anzahl repräsentativer Farben im Bild. Dadurch wird automatisch die erforderliche Obermenge zur Bildung der finalen Farbpalette bereitgestellt, wenn das Bild ausreichend viele Farben enthält. Das erzwingen eines großen Farbpalette mit anderen Clusteringverfahren, z.B. über einen pauschal großen K Parameter bei K-Means, führt hingegen unter Umständen zu einer Partitionierung des Farbraums, die nicht der Clusterstruktur des Histogramms entspricht.

## 5 ACoPa

Im Folgenden wird die grundlegende Arbeitsweise des ACoPa Algorithmus nach Petro [21] vorgestellt. Dabei werden die Herausforderungen, die bei der Implementierung aufgetreten sind, besprochen. Abschließend werden exemplarisch Anwendungsergebnisse präsentiert.

### 5.1 Konvertierung in den HSI-Raum

Zunächst wird das Histogramm in den HSI Farbraum  $\{(h, s, i) \mid 0 \leq h < 360 \wedge 0 \leq s, i \leq 1\}$  übertragen. Die Intensität eines Farbtons wird dabei in Polarkoordinaten via  $h$  und  $s$  angegeben, während die maximal mögliche Sättigung wiederum von der Intensität  $i$  abhängt. Zur Konvertierung vom RGB in HSI Raum wurden verschiedene Umrechnungsvorschriften erprobt. Die Umrechnung gemäß der ACoPa-Autoren lautet:

$$\begin{aligned} I &= \frac{R + G + B}{3} \\ S &= \sqrt{(R - I)^2 + (G - I)^2 + (B - I)^2} \\ H &= \arccos\left(\frac{(G - I) - (B - I)}{S\sqrt{2}}\right) \end{aligned} \quad (5)$$

Die Umrechnung gemäß eines Lehrbuchs für Farbbild-Verarbeitung [29] lautet hingegen:

$$\begin{aligned} I &= \frac{R + G + B}{3} \\ S &= 1 - \frac{\min(R, G, B)}{I} \\ H &= \arccos\left(\frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - B)}}\right) \end{aligned} \quad (6)$$

Abbildung 10 stellt die Umrechnungsergebnisse dem Referenz HSI Raum (R) gegenüber. Keine der Umrechnungsvorschriften führt zu einem Doppelkegel. Weder Formel 5 noch Formel 6 projiziert die Farben mit 100% Sättigung ( $s = 1$ ) in eine Ebene. Formel 5 führt lediglich zu einer Drehung und Stauchung des RGB-Würfels, Formel 6 führt zu einem nach unten geöffneten Kegel. Da schlussendlich keine Formel gefunden werden konnte, die zu einem korrekten Doppelkegel führt, wurde die Berechnung mit Formel 5 fortgeführt.

### 5.2 Histogramm-Segmentierung

Die Samples des Ausgangsbildes werden entlang der Hue-Werte sortiert. Das 1-dimensionale Hue-Histogramm  $h = (h_i)_{i=1..b}$  mit  $b$ -Bins wird gebildet. Gesucht wird nun eine Sequenz  $s = (s_i)_{i=1..k}$  mit  $1 = s_0 < s_1 < \dots < s_k = b$ , welche eine Segmentierung des Histogramms darstellt. Das Intervall  $[s_i, s_{i+1}]$  wird als Segment bezeichnet. Ziel ist, dass das Histogramm in den Bereichen  $[h_{s_i}, \dots, h_{s_{i+1}}]$ , eine „annähernd unimodale Verteilung aufweist“ [21]. Abbildung 11 zeigt das Prinzip an verschiedenen Beispielen.

Das Histogramm ist offensichtlich in jedem Segment unimodal, wenn  $s$  mit den Minima des Histogramms initialisiert wird. Es wird nun versucht, Elemente aus  $s$  zu entfernen, indem für  $\forall i = 1..k$  überprüft wird, ob  $h$  im Intervall  $[h_{s_{i-1}}, \dots, h_{s_{i+1}}]$  die „unimodale Hypothese“ erfüllt. Anschaulich bedeutet das die Verschmelzung benachbarter Segmente, so dass das neu entstandene Segment nach wie vor „annähernd unimodal ist“. Hierfür stellen die Autoren in einer separaten Veröffentlichung [22] einen parameterfreien statistischen Test vor, der  $h$  im betrachteten Intervall mit einem Referenz-Histogramm  $h^r$  vergleicht.  $h^r$  ist in  $[h_{s_{i-1}}^r, \dots, h_{s_{i+1}}^r]$  zunächst streng

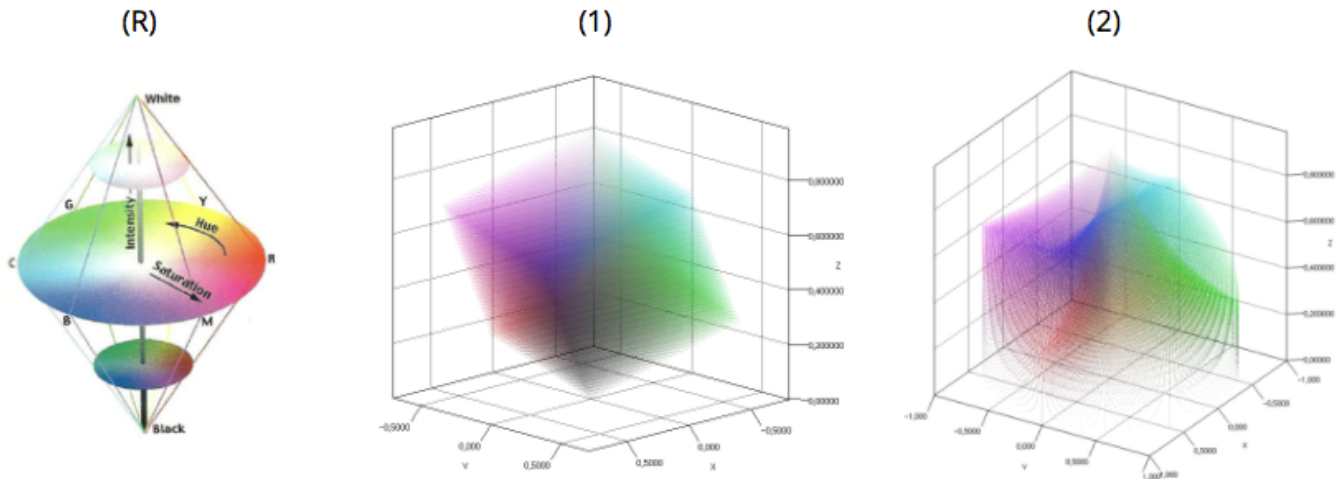


Abbildung 10: Gegenüberstellung von RGB zu HSI-Umrechnungsergebnisse. (a.) Referenz-HSI Raum. (b.) Umrechnung nach [21]. (c.) Umrechnung nach [29].

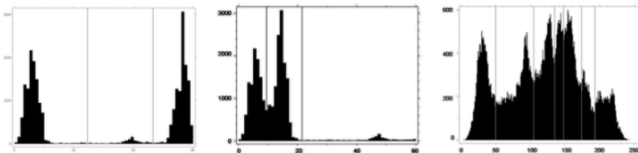


Abbildung 11: Beispiele der Segmentierung eines Histogramms in unimodale Abschnitte. (Quelle: [21])

monoton wachsend und danach streng monoton fallend und damit in jedem Fall unimodal. Das Referenz-Histogramm wird aus dem Original-Histogramm  $h$  durch Anwendung des Grenander-Operators gebildet. Die komplexen Details hierzu sind [21, 22] zu entnehmen. Da der parameterfreie Test verhältnismäßig aufwändig ist, wird in der eigenen Implementierung auf einen simplen T-Test zurückgegriffen. Dieser liefert ebenfalls befriedigende Ergebnisse, ist aber abhängig vom gewählten Signifikanzniveau.

Das Verfahren zur Histogramm-Segmentierung wird in [22] als **Fine-to-Coarse (FTC) Segmentation Algorithm** zusammengefasst. Zunächst wird  $s$  mit allen Minima des Histogramms initialisiert. Daraufhin werden so lange benachbarte Segmente durch Überprüfung der unimodalen Hypothese verschmolzen, bis keine Verschmelzung mehr möglich ist. Die Repräsentanten eines Segments werden durch Mittelung der Samples gebildet, die zum jeweiligen Segment gehören. Abbildung 12 zeigt dies an einem Beispiel.

### 5.3 Bildung der hierarchischen Farbpalette

Der ACoPa Algorithmus besteht aus einer hierarchischen Anwendung der Histogramm-Segmentierung. Dabei wird zuerst der  $h$ -, danach der  $s$ - und abschließend der  $i$ - Kanal segmentiert. Dabei werden in jedem Schritt die Samples der entstandenen Segmente separiert und die Histogramme der nächsten Ebene getrennt berechnet. Das Ergebnis ist eine hierarchische Farbpalette. Abbildung 13 zeigt dies am Beispiel der Covers aus Abbildung 12. Auf oberster Ebene (h) wurden die grundsätzlichen Farbtöne des Bildes identifiziert. Auf der zweiten Ebene werden die Farbtöne jeweils in unterschiedliche Sättigungen aufgeteilt, wenn nötig. Auf der dritten Ebene (i) werden von den Sättigungen zusätzlich Helligkeitsabstufungen gebildet.

Die letzte Ebene (i) bildet die Obermenge der Farben  $C_s$  für die weitere Verarbeitung. Petro [21] empfehlen zusätzlich, die erhaltenen Farben als Startpunkte für den K-Means Algorithmus zu verwenden. Abbildung 13 (b) zeigt, wie sich die Farben durch K-Means geändert haben. Es ist zu einem späteren Zeitpunkt zu entscheiden, welche der beiden Paletten für die weitere Verarbeitung geeigneter ist.

## 6 Ausblick

Als nächstes sind die Eigenschaften und Bedingungen der Farben zu ermitteln, die auf der Weboberfläche



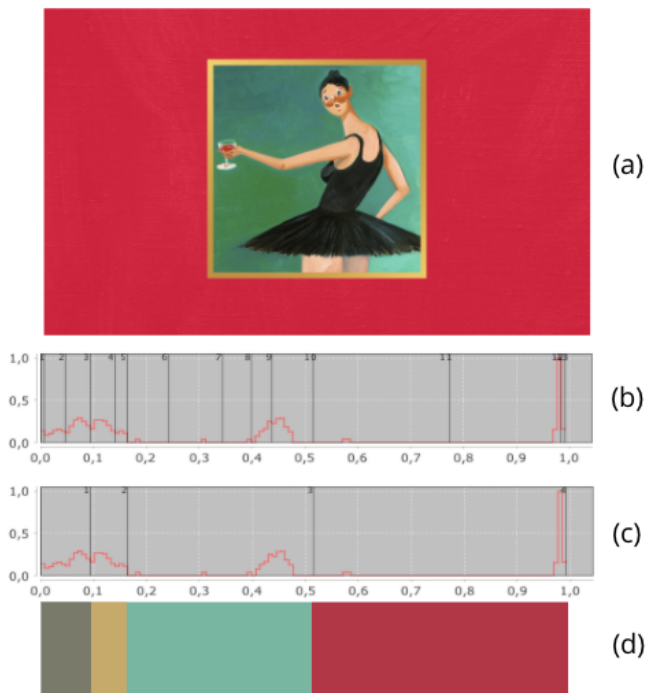


Abbildung 12: Beispiel für eine Segmentierung des Hue-Histogramms. (a) Ausgangsbild, ein Albumcover von Kanye West. (b) Hue-Histogramm (normalisiert), mit allen Minima als initiale Segmentierung. (c) Segmentierung nach Anwendung des FTC Algorithmus. (d) Farbmittelpunkte entsprechend der Samples der jeweiligen Segmente.

zum Einsatz kommen sollen. Die Anforderungen müssen in Constraints übersetzt werden. Daraufhin ist ein Algorithmus auszuwählen und zu implementieren, der aus der Farbpalette, die durch ACoPa ermittelt wurde, die passenden Farben herausfiltert. Abschließend werden die Ergebnisse an einer prototypischen Weboberfläche demonstriert.

## Literatur

- [1] Web content accessibility guidelines (wcag) 2.0. URL <https://www.w3.org/TR/WCAG/#visual-audio-contrast>.
- [2] Ali Jahanian & Jerry Liu & Qian Lin & Daniel R. Tretter & Eamonn O'Brien-Strain & Seungyon Lee & Nic Lyons & Jan P. Allebach. Automatic design of colors for magazine covers. In *SPIE Proceedings. Imaging & Printing in a Web 2.0 World IV.*, volume 8664, 2013.
- [3] Nick Babich. <https://www.smashingmagazine.com/2017/01/underestimated-power-color-mobile-app-design/>, jan 2017. URL <https://www.smashingmagazine.com/2017/01/underestimated-power-color-mobile-app-design/>.
- [4] J. Bezdek. Pattern recognition with fuzzy objective function algorithms. Kluwer Academic Publishers, 1981.
- [5] Alex Bigman. 6 principles of visual hierarchy for designers, 2014. URL <https://en.99designs.de/blog/tips/6-principles-of-visual-hierarchy/>.
- [6] Albert H. Munsell & F. Birren. *A grammar of color: a basic treatise on the color system of Albert H. Munsell*. Van Nostrand Reinhold Company, 1969.
- [7] Laura Elizabeth. A simple web developer's guide to color, apr 2016. URL <https://www.smashingmagazine.com/2016/04/web-developer-guide-color/>.
- [8] Google. Material design style - colors, 2016. URL <https://material.io/guidelines/style/color.html>.
- [9] Arthur R. Weks & G. Eric Hague. Color segmentation in the hsi color space using the k-means algorithm. In *Proceedings of SPIE*, volume 3026, 1997.
- [10] Martin Hahn. *Webdesign. Das Handbuch zur Webgestaltung*. Galileo Press, first edition, 2015.
- [11] S. Lin & P. Hanrahan. Modeling how people extract color themes from images. In *CHI Conference on Human Factors*, 2013.
- [12] Sharon Lin & Daniel Ritchie & Matthew Fisher & Pat Hanrahan. Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Transactions on Graphics*, 32(4), jul 2013.
- [13] Peter E. Hanford. Color image quantization for frame buffer displays. *Comp. Graph*, 16:297–307, 1982.



Abbildung 13: (a) Hierarchische Farbpalette des Covers aus Abbildung 12. (b) Farbpalette nach Anwendung von K-Means.

- [14] Peter O'Donovan & Aseem Agarwala & Aaron Hertzmann. Color compatibility from large datasets. In *ACM SIGGRAPH Conference*, page 63:1–63:12, 2011.
- [15] J. Itten. *The Elements Of Color*. Van Nostrand Reinhold Company, 1970.
- [16] Zhenyu Gu & Jian Lou. Data driven webpage color design. *Computer-Aided Design*, 77:46–59, aug 2016.
- [17] J. MacQueen. Some methods for classification & analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics & probability*, volume 1 of 14, California, USA, 1967.
- [18] D. Comaniciu & P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 5(24):603–619, 2002.
- [19] Pere Obrador. Automatic color scheme picker for document templates based on image analysis & dual problem. Technical report, Hewlett-Packard Laboratories, Palo Alto, CA, USA, 2006.
- [20] P. S. Periasamy. A probabilistic adaptive color reduction algorithm for single multiple images with & without transparency. Anna University, Faculty of Information & Communication Engineering, 2013.
- [21] J. Delon & A. Desolneux & J. L. Lisani & A. B. Petro. Automatic color palette. *Inverse Problems & Imaging*, 1(2):265–287, 2007.
- [22] J. Delon & A. Desolneux & J. L. Lisani & A. B. Petro. A nonparametric approach for histogram segmentation. 16(1):253–261, 2007.
- [23] M. Gervautz & W. Purgathofer. A simple method for color quantization: Octree quantization. *Graphic Gems. Academic Press*, page 287–293, 1990.
- [24] & S. K. M. Wong S. J. Wan & P. Prusinkiewicz. Variance-based color image quantization for frame buffer display. *Color Res. Appl.*, 15:52–58, 1990.
- [25] Marc Schenker. Designing with color contrast for improved usability, apr 2016. URL <https://www.awwwards.com/designing-with-color-contrast-for-improved-usability.html>.
- [26] Torsten Stapelkamp. *Web X.0. Erfolgreiches Webdesign und professionelle Webkonzepte*. Springer X.media.press, 2010.
- [27] Ranjitha Kumar & Arvind Satyanarayan & Cesar Torres & Maxine Lim & Salman Ahmad & Scott R. Klemmer & Jerry O. Talton. Webzeitgeist: Design mining the web. In *ACM Human Factors in Computing Systems (CHI)*, 2013.
- [28] K. Kanjanawanishkula & B. Uyyanonvarab. Novel fast color reduction algorithm for time-constrained applications. *Journal of Visual Communication & Image Representation*, 16:311–332, 2005.
- [29] K. N. Plataniotis & A. N. Venetsanopoulos. *Color Image Processing & Applications*. Springer Monograph, 2000.
- [30] W3C. Wcag contrast ratio, 2008. URL <https://www.w3.org/TR/2008/REC-WCAG20-20081211/Overview.html#contrast-ratiodef>.
- [31] W3C. Wcag relative luminance, 2008. URL <https://www.w3.org/TR/2008/REC-WCAG20-20081211/relative-luminance.xml>.
- [32] W3C. Css color module level 3, jun 2011. URL <https://www.w3.org/TR/css3-color/>.
- [33] Petra Hofstedt & Armin Wolf. *Einführung in die Constraint Programmierung*. Springer eXamen.press, 2007.