

Methoden zur automatisierten Farbgestaltung von Webseiten aus Bildvorlagen.

Masterprojekt

Philipp Anders

28. März 2017

Inhaltsverzeichnis

1	Einleitung	3
1.1	Problemmodellierung	3
1.2	Farbpaletten aus Bildvorlagen	4
1.3	Einordnung und Literaturbesprechung	5
2	Farbgestaltung von Webseiten	8
2.1	Lesbarkeit	8
2.2	Benutzerführung	9
2.3	Funktionsgruppen	10
2.4	Farbschemata	11
3	Problemlösungsverfahren und Systemarchitektur	13
3.1	Regelbasierter oder probabilistischer Ansatz	13
3.2	Layouts	13
3.3	Aufteilung der Suchverfahren	14
3.4	Systemarchitektur	15
4	Color Palette Estimation	17
4.1	Überblick	17
4.2	ACoPa	19
4.2.1	Konvertierung in den HSI-Raum	19
4.2.2	Histogramm-Segmentierung	20
4.2.3	Bildung der hierarchischen Farbpalette	21
4.3	Vorverarbeitung des Histogramms	21
5	Ausblick	23
	Literatur	23

1 Einleitung

Ziel dieser Arbeit ist die Untersuchung von Methoden zur automatisierten Bestimmung der Farbwerte von Oberflächenelementen in Webseiten, welche sich an einer Bildvorlage orientieren. Die Beantwortung dieser Fragestellung erfolgt durch den Vorschlag und die Umsetzung eines konkreten Verfahrens. Unterabschnitt 1.1 unternimmt eine grundlegende Modellierung des Färbungsproblems in Bezug auf Webseiten und legt den Rahmen der Zielstellung fest. Unterabschnitt 1.2 stellt das Teilproblem der Ermittlung repräsentativer Farben eines Bildes vor. Unterabschnitt 1.3 ordnet die Arbeit in den aktuellen Forschungskontext ein.

1.1 Problemmodellierung

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 [30]. In dieser Arbeit werden die Farbeigenschaften von zwei verschiedenen Arten von Oberflächenelementen $e = (c_{\text{text}}, c_{\text{background}}) \in C \times C$ betrachtet:

1. **Textelemente** e_{Text} : Elemente mit transparentem Hintergrund und definierter Vordergrundfarbe (z.B. `<h1>`, `<a>`, etc.).

Diese besitzen die CSS-Eigenschaften `{ background-color: transparent , color: e.ctext }`, wobei der Farbwert *transparent* als *rgba*(0,0,0,0) definiert wird. Dementsprechend gilt: $\text{type}(e) = \text{Text} \implies e.c_{\text{background}} = \text{transparent}$.

2. **Blockelement** e_{Block} : Elemente, die visuell als Blöcke wahrgenommen werden (z.B. `<div>`, `<button>`, etc.). Diese besitzen die CSS-Eigenschaften `{ background-color: e.cbackground , color: e.ctext }`.

Es wird also die Farbbestimmung von Vorder- und Hintergrundfarben fokussiert. Rahmen, Textdekoration und andere färbbare Eigenschaften werden vernachlässigt.

Üblicherweise werden mehrere Elemente eines HTML-Dokuments auf die gleiche Farbe abgebildet. Ein Beispiel ist die Darstellung aller Links ($e.c_{\text{text}}$) sowie Buttons ($e.c_{\text{background}}$) in Blau. Im Folgenden wird eine Liste von Elementen einer Webseite mit gleicher Farabbildung als **Color Group** $CG = (e_1, e_2, \dots)$ bezeichnet [siehe auch 11, 19]. Die Menge aller Color Groups einer Webseite heißt **Color Groups** $CGs = \{CG_1, \dots, CG_m\}$ mit $|CGs| = m$. Abbildung 1 veranschaulicht dies anhand einer exemplarischen Webseite für Musik-Streaming. (a) zeigt ein generisches Farbdesign, während (b) die sechs Color Groups separat hervorhebt.

TODO: Abbildung updaten

Webdesigner verkleinern den Suchraum zur Identifizierung geeigneter Farben für die Color Groups durch die Verwendung einer sogenannten **Farbpalette** [11, 12, 27]. Dabei handelt es sich um eine Farbmenge $P = \{c_1, c_2, \dots, c_n\}$ mit n Farben, wobei jedes $c \in P$



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.

als String von RGB Werten kodiert wird. **Die Farbgestaltung einer Webseite ist die Abbildung $f_{\text{coloration}} : CGs \rightarrow P$ mit folgender Definition:**

$$f_{\text{coloration}} : CGs \rightarrow P,$$

$$f_{\text{coloration}}(CG) = c \implies \forall e \in CG : \begin{cases} e.c_{\text{text}} = c, & \text{wenn } type(e) = \text{Text} \\ e.c_{\text{background}} = c, & \text{wenn } type(e) = \text{Block} \end{cases} \quad (1)$$

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

Die Farbpalette soll auf einer Bildvorlage basieren. So wird eine Harmonisierung des visuellen Eindrucks einer Weboberfläche und einer darin präsenten Grafik erreicht. Da die Farben einer Webseite wesentlich für deren vermittelte Atmosphäre sind [12], soll die Anpassung der Farbgebung an ein Motiv den Eindruck des Bildes unterstützen. Da die Farbpalette somit auf den verarbeiteten Daten basiert, ist eine Berechnung der Farbgestaltung einer Webseite zur Laufzeit möglich. Ein Anwendungsbeispiel hierfür ist die farbliche Anpassung einer Webanwendung für Musikstreaming an das gespielte Albumcover. Abbildung 1 (c) zeigt hierfür ein Beispiel. Die Ermittlung von P aus einer Grafik wird als Color Palette Estimation bezeichnet und im Folgenden besprochen.

1.2 Farbpaletten aus Bildvorlagen

Die Abbildung einer Bildvorlage I auf eine Farbpalette P wird von Delon et al. [6] als **Color Palette Estimation** (CPE) $f_{\text{CPE}} : I \rightarrow P$, 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.

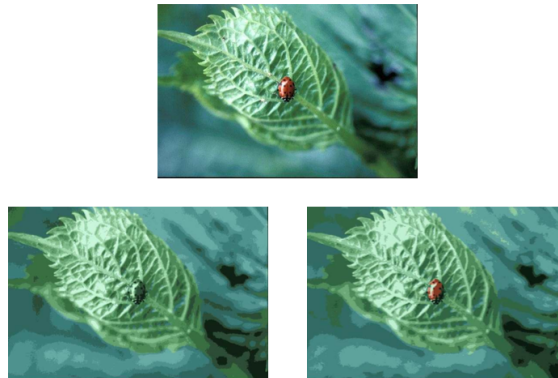


Abbildung 2: Beispiel für die Einfärbung eines Bildes mit unterschiedlichen Farbpaletten der Größe 12. Oben: Originalbild. Links: Farbpalette ohne rote Farbtöne. Rechts: Farbpalette mit roten Farbtönen, wodurch der Marienkäfer erkennbar ist (Quelle: [6])

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 wurden (Quantisierung) [32]. Aus diesem Kontext kommt das formale Kriterium der Summe des quadratischen Fehlers, welcher in diesem Anwendungsfall auch als *Recoloring Error* bezeichnet wird [18]. 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 Farbpaletten zu Gestaltungszwecken. Lin und Hanrahan [18] zeigen, dass in letzterem Kontext der Recoloring Error keine geeignete Metrik zur Beurteilung der Güte einer Farbpalette in Bezug auf die Bildvorlage ist. Dies beruht auf den 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 im Bezug auf die Bildvorlage als unterschiedlich repräsentativ bewertet wird.

Dieser Befund hebt hervor, dass die Güte der Abbildung $I \rightarrow P$ subjektiv ist und vom Anwendungsbezug abhängt. Aus diesem Grund wird für die Beurteilung der zu ermittelnden Farbpalette keine objektive Bewertungsfunktion herangezogen. Stattdessen wird die Auswahl eines Algorithmus zur CPE fokussiert, dessen resultierende Farbpalette zweckmäßig in Hinblick auf die farbliche Gestaltung von Webseiten ist. Hierfür werden in Abschnitt 2 Prinzipien für eine funktionale Gestaltung von Webseiten sowie Farbdefinitionen in Style Guides analysiert.

1.3 Einordnung und Literaturbesprechung

Diese 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 zum Einsatz kommen.

O'Donovan et al. [23] haben 2011 ein Regressionmodell zur Bewertung der Farbkompatibilität von bis zu fünf Farben entwickelt, d.h. zur Bewertung der Farbharmonie einer Farbpalette. Hierfür wurde ein Training an den Datenbeständen von Farbpaletten-Communities wie z.B. Adobe Color¹ durchgeführt. Dabei hat sich unter anderem ergeben, dass die auf geometrischen Strukturen im Farbkreis beruhenden Modelle der klassischen Farbentheorie Farbhamonien nicht zufriedenstellend voraussagen und in bestimmten Fällen sogar kontraproduktiv sind. Hierzu zählen zum Beispiel die Farbton-Schablonen von Itten [14] oder Munsell und Birren [22]. Die Grenzen des Modells liegen in der Bewertung der Harmonie von Farben mit unterschiedlicher räumlicher Ausprägung [11, 19]. Eine Matlab-Implementierung des Modells zur Überprüfung von Farbharmonien steht öffentlich zum Download bereit².

Lin et al. [19] 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 ausgewertet wurden³. Konkrete Färbungslösungen werden durch einen *Factor Graph* ermittelt, was ebenfalls einen Ansatz für die vorliegende Arbeit darstellt. Das Modell von O'Donovan et al. [23] 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.

Gu und Lou [11] 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 drei Zielfunktionen entwickelt. Zielfunktionen 1 gewährleistet einen ausreichenden Kontrast zwischen den Oberflächenelementen. Zielfunktion 2 passt die Farbgestaltung an ein Schlüsselwort an (z.B. „Business“ oder „Fresh“). Zielfunktion 3 wird durch das Modell von O'Donovan et al. [23] realisiert und gewährleistet Farbharmonie. Die Optimierung wird durch eine lexikographische Strategie umgesetzt, bei welcher in Interaktion mit einem Gestalter die Zielfunktionen nacheinander angewendet werden. Die Farben zur Seitenfärbung werden Farbpaletten von Adobe Color⁴ entnommen. Als alternatives Anwendungsbeispiel extrahieren die Autoren eine Farbpalette aus einer Grafik und nutzen diese als Grundlage zur Färbung, was dem Ziel der vorliegenden Arbeit entspricht. Zur CPE wird der K-Means Algorithmus verwendet. Somit ist der vorgestellte Prozess der Autoren im Gegensatz zum Ziel dieser Arbeit lediglich teilautomatisiert. Einerseits muss ein geeignetes k des K-Means Algorithmus vom Anwender ermittelt werden, andererseits erfordert die lexikographischen Strategie bei der Optimierung eine Nutzerinteraktion. Somit handelt es sich bei der Lösung um ein Unterstützungswerkzeug für Webdesigner und nicht um ein System zur vollautomatisierten Farbgestaltung einer Webseite.

Jahanian et al. [15] haben sich 2013 mit der automatisierten Farbgestaltung von Magazin-Covers auseinandergesetzt. Vergleichbar mit [11] wird ebenfalls die Optimierung des Farbkontrasts, der Farbharmonie und der Farbsemantik verfolgt. Im Gegensatz zu den bisherigen Lösungen wird allerdings mit expliziten Regeln zur Bewertung von Farbharmonien- [14] und Kontrasten anstatt mit Modellen gearbeitet, die sich aus Trainingsdaten ableiten. Über

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

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

³<http://www.colourlovers.com/>

⁴<https://color.adobe.com/de/explore/>

Flowcharts vermitteln die Autoren Lösungsprozeduren zur Suche geeigneter Schriftfarben unter Beachtung des Kontrasts, welche eine Anregung für die vorliegende Arbeit darstellen.



Abbildung 3: Die 40 populärsten Textfarben in Webseiten machen 70% aller verwendeten Textfarben aus. (Quelle: [17])

2 Farbgestaltung von Webseiten

Die Ziele der farblichen Gestaltung im Rahmen dieser Arbeit sind Lesbarkeit und Benutzerführung. Konkrete Kriterien für die Textlesbarkeit werden in Unterabschnitt 2.1 vorgestellt. In Unterabschnitt 2.2 wird der Begriff Benutzerführung erläutert und damit im Zusammenhang stehende Gestaltungsprinzipien im Webdesign herausgearbeitet. Davon ausgehend wird in Abschnitt 3 ein konkreter Vorschlag für eine Systemarchitektur zur automatisierten Farbgestaltung von Webseiten erarbeitet.

2.1 Lesbarkeit

Abbildung 3 zeigt, dass gemäß einer Auswertung von über 100.000 Webseiten Text in HTML-Dokumenten fast ausschließlich in Graustufen dargestellt wird [17]. Die populärsten Farben sind Schwarz, Dunkelgrau und Weiß. Hahn [12] begründet dies damit, dass Text schlechter lesbar wird, je bunter er ist. Der Style Guide von Googles Material Design [10] hebt hervor, dass Aufgrund des Simultankontrasts die Verwendung von Grau als Textfarbe bei Hintergrund mit hoher Buntheit zu unerwünschten Effekten führt, wie Abbildung 4 veranschaulicht. Stattdessen wird in Abhängigkeit von der Helligkeit 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)`

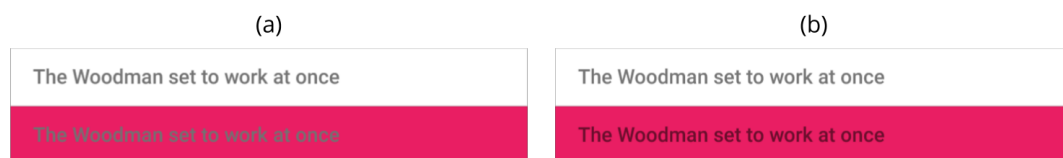


Abbildung 4: Empfohlene Textfarbe bei buntem Hintergrund. (a) Textfarbe `rgb(114, 114, 114)`. Der Text ist aufgrund des Simultankontrasts unangenehmer zu lesen. (b) Textfarbe `rgba(0, 0, 0, 0.54)`, d.h. Schwarz mit einem Alphawert von 54%. Die Textfarbe ergibt sich zu einer Verschwärzlichung der Hintergrundfarbe und vermeidet so den Simultankontrast. (Quelle: [10])

Im Folgenden wird der Farbwert `rgba(0, 0, 0, 0.87)` als *schwarz* und der Farbwert `rgba(255, 255, 255, 1.0)` als *weiss* bezeichnet.

Es ist zu entscheiden, wann *schwarz* und wann *wei* als Textfarbe verwendet wird. Hierfür geben die Web Content Accessibility Guidelines (WCAG) des W3C konkrete Grenzwerte

für Kontrastverhältnisse in Abhängigkeit von der Text-, der Hintergrundfarbe sowie der Textgröße an. Die Formel zur Berechnung des Kontrastverhältnisses L_c lautet [28]:

$$L_c(e.c_{\text{text}}, e.c_{\text{background}}) = \frac{L(e.c_{\text{text}}) + 0.05}{L(e.c_{\text{background}}) + 0.05}, \quad (2)$$

$$L(e.c_{\text{text}}) \geq L(e.c_{\text{background}})$$

mit $L_c \in [1, \dots, 22]$, wobei $L_c(\text{schwarz}, \text{weiss}) = 22$ und $e.c_{\text{text}} = e.c_{\text{background}} \implies L_c(e.c_{\text{text}}, e.c_{\text{background}}) = 1$ gilt. Die Berechnungsvorschrift der normalisierten relativen Luminanz $L()$ einer Farbe ist [29] zu entnehmen.

Es gibt zwei verschiedene Grenzwertklassen [1]. Die Grenzwertklasse *AA* beschreibt die Minimalanforderung an lesbaren Text:

$$AA : L_c(e.c_{\text{text}}, e.c_{\text{background}}) \geq \begin{cases} 3.0, & \text{size}(e) \geq 18\text{pt} \vee \text{size}(e) \geq 14\text{pt} \wedge \text{weight}(e) = \text{bold} \\ 4.5, & \text{sonst} \end{cases} \quad (3)$$

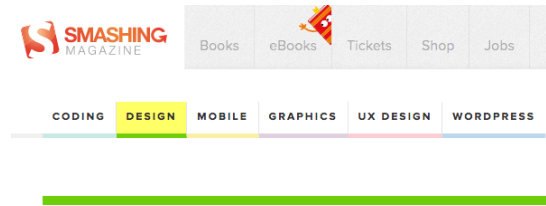
Die Klasse *AAA* beschreibt bessere Lesbarkeit durch die Verwendung höherer Grenzwerte:

$$AAA : L_c(c_1, c_2) \geq \begin{cases} 4.5, & \text{size}(e) \geq 18\text{pt} \vee \text{size}(e) \geq 14\text{pt} \wedge \text{weight}(e) = \text{bold} \\ 7.0, & \text{sonst} \end{cases} \quad (4)$$

2.2 Benutzerführung

Ein adäquater Farbeinsatz ermöglicht die Steuerung von Aufmerksamkeit und unterstützt die Bildung von Zusammenhängen [12]. Dieses Konzept wird auch als visuelle Hierarchy bezeichnet [4], bei welchem die Oberflächenelemente bei der Wahrnehmung strukturiert und priorisiert werden. Hahn [12] 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. Sie stellt kein System, sondern eine Gestaltungsstruktur dar, 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 Abgrenzung von Bereichen mit einheitlicher Funktion. Dies beinhaltet beispielsweise die farbliche Differenzierung von Navigations-, Inhalts- und Servicebereichen. 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 der farblichen Abgrenzung verschiedener Navigationsbereiche.
- **Funktionen und Zustände:** Bedeutet die farbliche Kodierung verschiedener Elemente mit einheitlicher Funktion. Beispielsweise sind alle anklickbaren Elemente (z.B.



Category: Design

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

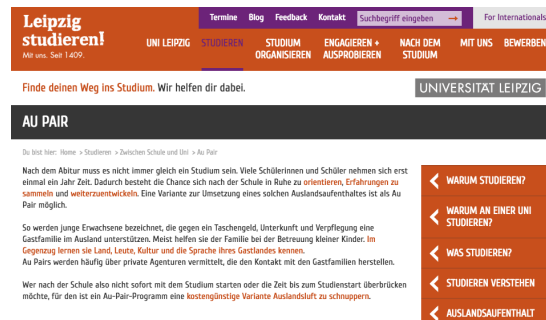


Abbildung 6: Farbkodierung von Funktionsbereichen am Beispiel von www.leipzig-studieren.de. Navigationsbereiche (orange) werden zueinander in Bezug gebracht und vom Inhalt (weiß) abgegrenzt.

Links und Buttons) einer Seite durch eine einheitliche Farbe signalisierbar. Abbildung 7 zeigt hierfür ein Beispiel. Bestimmte Wirkungen können zusätzlich farblich hervorgehoben werden, wie z.B. die rote Färbung von Buttons zum Löschen oder Abbrechen einer Aktion. Wurde eine Farbe einmal mit einer Funktion belegt, soll sie einheitlich verwendet werden. Beispielsweise würde ein nicht-anklickbares, blaues Element in der Oberfläche aus Abbildung 7 den Nutzer verwirren.

2.3 Funktionsgruppen

Aus Unterabschnitt 2.2 geht hervor, dass Farbeinsatz im Webdesign der Abgrenzung von Funktionsbereichen sowie der Kennzeichnung von Elementen mit einheitlicher Funktion

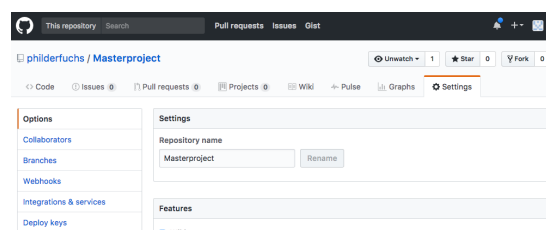


Abbildung 7: Farbkodierung von Funktionen am Beispiel von www.github.com. Nur anklickbare Elemente sind blau, aktive Elemente werden zusätzlich braun markiert.

dient. Aus diesen Gründen wird die Abbildung $f_{\text{coloration}} : CGs \rightarrow P$ durch eine Abstraktionsschicht erweitert, welche beschreibt, welche Funktion eine Color Group in einer Oberfläche erfüllt. In Anlehnung an [10, 8] wird exemplarisch die Menge der k Funktionsgruppen $FGs = \{\text{Primär, Sekundär, Akzent, Interaktion, Text (neutral), Hintergrund (neutral)}\}$ mit $|FGs| = k = 6$ definiert. Die Einzelnen Bedeutungen $\forall fg \in FGs$ lauten wie folgt:

- **Primärfarbe:** Wird in Relation zu den anderen Farben am häufigsten eingesetzt und bildet so die farbliche Grundstimmung der Oberfläche [26].
- **Sekundärfarbe:** Unterstützt die Farbstimmung der Primärfarbe und orientiert sich dementsprechend an deren Farbton. Durch die Wahl einer Farbe mit geringerer Buntheit tritt sie in der visuellen Hierarchie stärker in den Hintergrund [4] und unterstützt durch den Qualitätskontrast die Wirkung der Primärfarbe [12].
- **Interaktionsfarbe:** Signalisiert Elemente, mit denen eine Interaktion möglich ist, z.B. durch klicken. Dies trifft nicht zwangsläufig auf Elemente zu, bei denen der Nutzer ohnehin eine Interaktionsmöglichkeit aufgrund ihrer Funktionsgruppe bereits erwartet, wie z.B. in der Seitennavigation.
- **Akzent:** Signalisiert Elemente mit hoher Priorität in der visuellen Hierarchie. Dies wird durch den sparsamen Einsatz (Quantitätskontrast, [12]) einer Farbe mit hoher Buntheit [4] erreicht.
- **Text (neutral):** Farbe für Fließtext. Entsprechend Unterabschnitt 2.1 wird hierfür in Abhängigkeit vom Hintergrund *weiss* bzw. *schwarz* festgelegt.
- **Hintergrund (neutral):** Standardfarbe von Blöcken, die keiner der obigen Funktionen entsprechen. Dies betrifft vorwiegend Blöcke mit Fließtext, so dass Textlesbarkeit zu beachten ist. Dementsprechend sind Graustufen zu bevorzugen [27], wobei entsprechend Unterabschnitt 2.1 ein ausreichender Luminanzkontrast zur Textfarbe zu gewährleisten ist. Darum wird die neutrale Hintergrundfarbe analog zur Textfarbe auf *weiss* bzw. *schwarz* beschränkt.

Die hier verwendeten Funktionsgruppen sind exemplarisch. Abhängig von den individuellen Ansprüchen einer Webseite ist die Definition anderer Funktionsgruppen denkbar. Bei mobilen Anwendungen ist beispielsweise eine farbliche Differenzierung verschiedener Interaktionsformen sinnvoll (z.B. drücken und wischen).

2.4 Farbschemata

Die Abbildung zwischen den Funktionsgruppen $FGs = \{fg_1, \dots, fg_k\}$ und den tatsächlichen Farben einer Farbpalette $P = \{c_1, \dots, c_n\}$ wird als die Funktion $f_{\text{scheme}} : FGs \rightarrow P$ definiert und im Folgenden als **Farbschema** bezeichnet. Die Farbgestaltung einer Webseite ist damit die Abbildung eines Farbschemas auf eine Farbpalette $f_{\text{coloration}} : (CGs \rightarrow FGs) \rightarrow P$.

Diese Definition eines Farbschemas unterscheidet sich von der Verwendung des Begriffs in der Literatur [12]. In dieser bezeichnet ein Farbschema die Charakteristik einer Farbpalette. Es beschreibt, *wie viele* unterschiedliche Farbtöne eine Farbpalette enthält und in welcher *geometrischen Beziehung* diese zueinander im Farbkreis stehen. Beispielsweise bezeichnet ein *triadisches* Farbschema eine Farbpalette mit drei Farbtönen, die sich in einem Abstand von 120° zueinander befinden. Genauer wird in *monochromatische*, *komplementäre (duale)*, *triadische* und *tetraedische* Farbschemen unterschieden, welche Farbpaletten mit einem,

zwei, drei oder vier verschiedenen Farbtönen repräsentieren. [2, 8, 10] 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** als Charakterisierung der Funktion $f_{\text{coloration}}$ in dieser Arbeit adaptiert. Abbildung 8 zeigt hierfür Beispiele. Es wird eine exemplarische Farbpalette abgebildet. Die dargestellten Funktionsgruppen entsprechen der in Unterabschnitt 2.3 definierte Menge FGs . Es ist zu erkennen, dass die Textfarben sowie die Hintergrundfarbe bereits unabhängig von der Farbpalette vorgegeben werden. Damit realisiert jedes Farbschema im Rahmen dieser Arbeit einen **Bunt-Unbunt-Kontrast**, was nach [27] eine allgemeine Empfehlung für das Screendesign darstellt. Die Grafik zeigt weiterhin mögliche Abbildungen zwischen den Farbfunktionen und der Farbpalette unter „konkrete Farbschemata“. Für die neutrale Textfarbe stehen in Abhängigkeit vom jeweiligen Hintergrund im HTML-Dokument *weiss* bzw. *schwarz* zur Verfügung. Standardmäßig wird *weiss* für den neutralen Hintergrund angenommen. Die *dark*-Variante eines Farbschemas veranschaulicht eine Variante mit allgemein dunkler Farbstimmung durch die Verwendung eines dunklen Hintergrunds. Ein monochromes Farbschema beschränkt sich auf einen Farbton in verschiedenen Schattierungen. Dadurch wird ein **Qualitätskontrast** gebildet, welcher die Wirkung der Farbe mit der größten Buntheit erhöht. 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 selteneren Farben zusätzlich betont und so zur Bildung einer visuellen Hierarchie beiträgt. Das exemplarische tertiäre Farbschema bildet die Akzent- und Interaktionsgruppen auf unterschiedliche Farben ab und differenziert diese so zusätzlich.

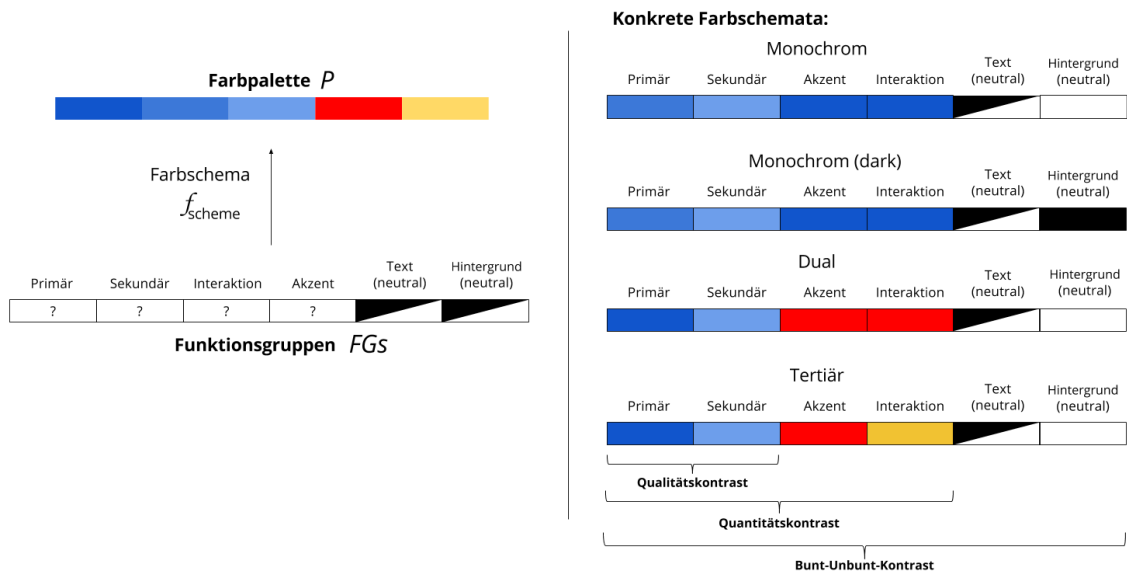


Abbildung 8: Beispiele für monochrome, duale und tertiäre Farbschemata. Der Farbeindruck der Seite wird durch Quantitäts-, Qualitäts- und Bunt-Unbunt-Kontraste bestimmt.

3 Problemlösungsverfahren und Systemarchitektur

Ausgehend von der Literaturbesprechung in Unterabschnitt 1.3 sowie den Schlussfolgerungen zur Farbgestaltung von Webseiten in Abschnitt 2 wird im Folgenden eine Methode zur Problemlösung besprochen. Darauf aufbauend wird in Unterabschnitt 3.4 die konkrete Architektur des zu implementierenden Systems vorgestellt.

3.1 Regelbasierter oder probabilistischer Ansatz

Gu und Lou [11] stellen für die automatisierte Farbgestaltung zwei grundlegende Herangehensweisen vor:

1. **Regelbasiert:** Beschreibt quantitative Modelle mit deterministischem Regelwerk. Die Arbeit von Jahanian et al. [15] zur automatisierten Färbung von Magazincovern stellt hierfür ein Beispiel dar. Durch die Analyse von Farbharmonie-Modellen wurden Regeln für Farbbeziehungen abgeleitet.
2. **Datengetrieben:** Beschreibt Modelle, welche die Performanz möglicher Lösungen einer automatisierten Farbgestaltung auf Grundlage existierender Beispieldaten vorhersagt.

Die regelbasierten Modelle treffen strickte Aussagen auf einem gewissen Abstraktionsgrad durch eine Vereinfachung des Problems bis zu einer Ebene, auf der Entscheidungen auf Grundlage weniger Parameter getroffen werden (z.B. durch die Angabe fester Grenzwerte). Im Gegensatz dazu stützen sich die Modelle des datengetriebenen Ansatzes auf reale Beispiele und tendieren daher zu robusteren Ergebnissen in der Anwendungsdomäne [11]. Durch die Vielfalt der Beispieldaten findet ein verstärkter Einsatz probabilistischer Modelle statt [siehe z.B. 11, 19].

Da bei probabilistischen Modellen alle potentiellen Lösungen bewertet und gegeneinander abgewogen werden müssen, sind hohe Laufzeiten möglich. Die bereits entwickelte Lösung von Gu und Lou [11] zur automatisierten Farbgestaltung von Webseiten benötigt selbst nach einer Optimierung des Suchverfahrens, bei der unwahrscheinliche Lösungen frühzeitig ausgeschlossen werden, bis zu 2 Stunden zur Konvergenz. Darüber hinaus 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 Farbanzahl (k) vom Gestalter, andererseits erfordert die lexikographische Strategie eine Nutzerinteraktion während des Optimierungsprozesses.

Aus diesen Gründen wird sich in dieser Arbeit für eine vollautomatisierte, regelbasierte Lösungsmethode entschieden. Auch ohne die Auswertung großer Datenmengen existieren quantitative Modelle zur Gewährleistung der Textlesbarkeit und Benutzerführung, wie z.B. die in 2.1 vorgestellten quantitativen Grenzwerte der *Web Content Accessibility Guidelines*.

3.2 Layouts

Ziel dieser Arbeit die Färbung einer konkreten Webseite. Das Parsing eines solchen HTML-Dokuments zur Erschließung aller $e \in CG$ sowie die Speicherung aller topologischen Informationen dieser Elemente innerhalb einer Seite führt jedoch eine zusätzliche Ebene der Komplexität in das zu lösende Problem ein. Der CSS-Standard sieht durch die Klassen-

Selektoren [31] ohnehin die Auszeichnung einheitlicher darzustellender Elemente bereits auf Ebene der Dokumentenbeschreibung vor. Dementsprechend wird von der konkreten HTML-Beschreibung einer Webseite als **Layout** abstrahiert. Ein Layout definiert:

1. *FGs*: Die Menge der k Funktionsgruppen *FGs*. Entsprechend Abschnitt 2 wird diese standardmäßig auf $FGs = \{\text{Primär, Sekundär, Akzent, Interaktion, Text (neutral), Hintergrund (neutral)}\}$ festgelegt.
2. *CGs*: Die Menge der Color Groups mit den enthaltenen Elementen $CGs = \{CG_1 = (e_1, e_2, \dots), \dots, CG_m = (e_1, e_2, \dots)\}$.
3. $CGs \rightarrow FGs$: Die Abbildung der Color Groups auf Funktionsgruppen. Beispielsweise definiert ein Layout selbst, welche Elementen interaktiv sind und somit zur Farbfunktion *Interaktion* gehören.

Somit wird die Komplexität der Suchverfahren reduziert, indem die Menge der Funktionsgruppen als konstant angesehen wird und die konkrete Topologie der Oberflächenelemente innerhalb des HTML-Dokuments im Verantwortungsbereich des Layouts liegt.

3.3 Aufteilung der Suchverfahren

Formale stellt sich die automatisierte Farbgestaltung einer Webseite aus einer Bildvorlage als die Abbildung $f_{\text{coloration}} : (CPs \rightarrow FGs) \rightarrow (I \rightarrow P)$ dar. In Unterabschnitt 3.2 wurde dargestellt, dass zur Lösungssuche die Ermittlung der Abbildung $CGs \rightarrow FGs$ als gegeben durch das Layout angesehen und somit aus der Betrachtung entfernt wird. Weiterhin existieren die Teilprobleme (1) $f_{CPE} : I \rightarrow P$, d.h. die Ermittlung von n repräsentativen Farben einer Bildvorlage sowie (2) Die Ermittlung des Farbschemas $f_{\text{scheme}} : FGs \rightarrow P$, d.h. die Auswahl von k Farben aus der Palette und deren Zuordnung zu den Funktionsgruppen.

In Unterabschnitt 3.4 wurde ein regelbasierter Ansatz zur Lösungssuche gewählt. Die Regeln werden im Folgenden als **Constraints** bezeichnet. Es existieren zwei Herangehensweisen:

1. **Constraints-First**: Setze $|FGs| = k = n = |P|$. P wird in Abhängigkeit vom Layout ermittelt. Hierbei wird die Ermittlung des Farbschemas zum Zeitpunkt der CPE verlagert. Unter Kenntnis der Funktionsgruppen mit deren gewünschten Eigenschaften ist der Suchraum das Histogramm der Bildvorlage. Dieser Ansatz wird unter anderem von O'Donovan et al. [23] 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 für das Farbschema beschränkt sich auf die Farben in P . Dieser Ansatz wird ebenfalls von [11] bei der datengetriebenen Färbung von Webseiten verfolgt.

Für diese Arbeit wird die Herangehensweise **Constraints-Last** bevorzugt, um Flexibilität in Bezug auf unterschiedliche Layouts zu gewährleisten. Für die Arbeit bedeutet dies, dass eine Problemlösung anhand eines exemplarischen Layouts mit einer definierten Menge an Funktionsgruppen veranschaulicht wird, eine Übertragung auf andere Layouts jedoch möglich ist.

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

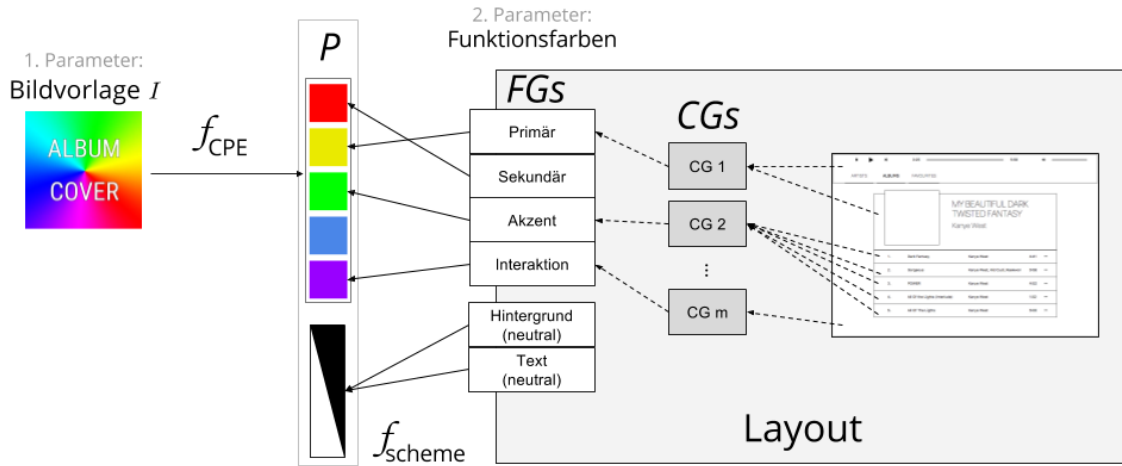


Abbildung 9: System-Architektur. Der grau unterlegte Kasten visualisiert den Bereich, für den algorithmische Lösungen gefunden werden müssen. Die Eingabeparameter sind eine Bildvorlage und eine Menge von Color Groups, welche von einem Layout definiert werden. Das System setzt zwei Suchverfahren um, welche 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 Farbabbildung gefunden wurde, wird das kolorierte Layout ausgegeben.

groß. Als Beispiel sei ein Layout mit 5 Funktionsgruppen sowie eine Farbpalette mit 10 Farben gegeben. Somit gilt $|FGs| = k = 5$ und $|P| = n = 10$, wodurch sich $\binom{10}{5} 5! = 30.240$ mögliche Kombinationen ergeben. Dementsprechend ist ein effektives Suchverfahren für die Ermittlung des Farbschemas zu ermitteln. Die von Lin et al. [19] verwendeten Factor Graphs stellen hierfür einen Ansatz dar.

3.4 Systemarchitektur

Abbildung 9 fasst die Systemarchitektur für die die automatisierte Farbgestaltung einer Webseite zusammen. Parameter 1 ist die Bildvorlage I , welche durch einen Algorithmus zur Color Palette Estimation f_{CPE} auf eine Menge repräsentativer Farben abgebildet wird. Aus Unterabschnitt 2.2 folgt, dass unabhängig von der Bildvorlage *weiss* und *schwarz* für den neutralen Text bzw. Hintergrund benötigt werden. Darum wird die Farbpalette entsprechend ergänzt: $P = f_{CPE}(I) \cup \{weiss, schwarz\}$.

Parameter 2 sind die Funktionsgruppen, welche in Unterabschnitt 2.3 festgelegt wurden. Durch ein Suchverfahren f_{scheme} werden Farben für $\forall fg \in FGs$ gesucht, d.h. es wird ein Farbschema gebildet. Hierfür wird ein Constraint System modelliert, welches durch die von Lin et al. [19] vorgestellten Factor Graphs realisiert wird.

Durch die Funktionsgruppen wird im Constraint System von der konkreten Topologie der Oberflächenelemente innerhalb der Webseite abstrahiert. Dies spiegelt sich darin wieder, dass für die Funktionsgruppe „Text (neutral)“ keine bestimmte Farbe durch das Constraint System festgelegt wird, sondern sowohl *weiss* als auch *schwarz* zur Auswahl gestellt werden. Darum wird auf der Layout-Ebene für Block-Elemente pauschal diejenige Textfarbe

festgelegt, welche unter Verwendung der Formel zur Berechnung des Kontrastverhältnisses L_c aus Unterabschnitt 2.1 den besseren Kontrast liefert:

$$\begin{aligned}
 & \forall FG \in FGs : \\
 & f_{\text{scheme}}(FG) = c \implies \forall e \in CG \rightarrow FG : \begin{cases} e = (c, \text{transparent}), & \text{wenn } type(e) = \text{Text} \\ e = (f_{\text{textcolor}}(c), c), & \text{wenn } type(e) = \text{Block} \end{cases} \\
 & \hline
 & f_{\text{textcolor}} : c \rightarrow c, f(c) = \begin{cases} \text{schwarz}, & \text{wenn } L_c(\text{schwarz}, c) \geq L_c(\text{weiss}, c) \\ \text{weiss}, & \text{sonst} \end{cases}
 \end{aligned} \tag{5}$$

Im Folgenden wird... [Vorschau auf die nächsten Kapitel](#)

4 Color Palette Estimation

In diesem Abschnitt wird eine Algorithmus zur Lösung des Teilproblems f_{CPE} ermittelt. Dies bedeutet die Festlegung eines Verfahrens zur Abbildung $I \rightarrow P$, wobei $P = (c_1, \dots, c_n)$ die Bildvorlage I mit einer minimalen Menge von Farben repräsentiert. Unterabschnitt 4.1 bietet einen Überblick über vorhandene Algorithmen zur CPE. Im Anschluss wird eine konkrete Methoden ausgewählt und in Unterabschnitt 4.2 erläutert.

4.1 Überblick

Grundlegend sind zwei Ansätze zur CPE zu unterscheiden:

1. **Histogramm-basiert:** Algorithmen, die von der räumliche Anordnung der Farben im Bild abstrahieren und P auf Grundlage der globalen Farbinformation bilden, d.h. des Histogramms. Da der Definitionsbereich eines Histogramm eine Teilmenge des Farbraums bildet, entspricht dieser Fall der unüberwachten Klassifizierung des Farbraums. Die Umsetzung erfolgt überwiegend durch Clustering-Verfahren, die Gruppen ähnlicher Farben im Bild identifizieren.
2. **Segmentierungs-basiert:** Algorithmen, die die räumliche Anordnung der Farben im Bild berücksichtigen. Durch eine Segmentierung des Bildes werden zusammenhängende Komponenten identifiziert, für welche im Folgenden repräsentative Farben ermittelt werden. Die menschlichen Wahrnehmungseigenschaften auf Komponentenebene werden auf diese Weise besser berücksichtigt, gleichzeitig wird durch die zusätzliche Betrachtung der Positionsinformation eine weitere Ebene der Komplexität eingeführt [18].

Morse et al. [21] kritisieren an den segmentierungs-basierten Verfahren, dass sie von einer akkuraten Bildsegmentierung abhängen, welche nicht für jede Bildstruktur gewährleistet ist (z.B. in stark texturierten Bereichen). Aus diesem Grund wird für die weitere Diskussion der Fokus auf die histogramm-basierten Verfahren gelegt.

Kanjanawanishkula und Uyyanonvarab [16] treffen eine Kategorisierung der histogramm-basierten Verfahren in *hierarchisch* und *iterativ*. Hierarchische Verfahren starten vor dem Erreichen der (fest zu wählenden) Farbanzahl n mit mehr (*bottom-up*) bzw. weniger (*top-down*) Clustern. Die Aufspaltung bzw. Vereinigung von Clustern basiert auf der statistischen Analyse der Verteilung der Bildfarben im Farbraum. Hierzu zählen unter anderem die in der Vergangenheit populären top-down Raumunterteilungs-Algorithmen wie z.B. Mediancut [13] oder Octree [9], welche den Farbraum sukzessiv in disjunkte Teilräume zerlegen und dabei den Clustern eine Würfelform unterstellen. Hierin begründet ebenfalls die Kritik an diesen Verfahren. Ergebnis der Verarbeitung ist ein Dendrogram, dessen Blätter die Farben der Farbpalette repräsentieren.

Ein Schnitt des Dendograms entspricht einer Partitionierung des Raums, was dem Ergebnis der iterative Verfahren entspricht. Sie starten bereits mit der erforderlichen Anzahl Clustern n und optimieren deren Position iterativ entsprechend einer Zielfunktion. Ein Beispiel ist die Minimierung des quadratischen Fehler, wie z.B. bei K-Means [20, 33] oder Fuzzy C-Means [3]. Andere Verfahren analysieren das Histogramm auf dichte bzw. weniger dichte Regionen, wie z.B. der Mean-Shift Algorithmus [5]. Die Kritik der iterativen Verfahren begründet sich in deren Abhängigkeit von den Startbedingungen [6].

Für eine detailliertere Vorstellung von Algorithmen zur CPE verweist der Autor auf Periasamy [24].

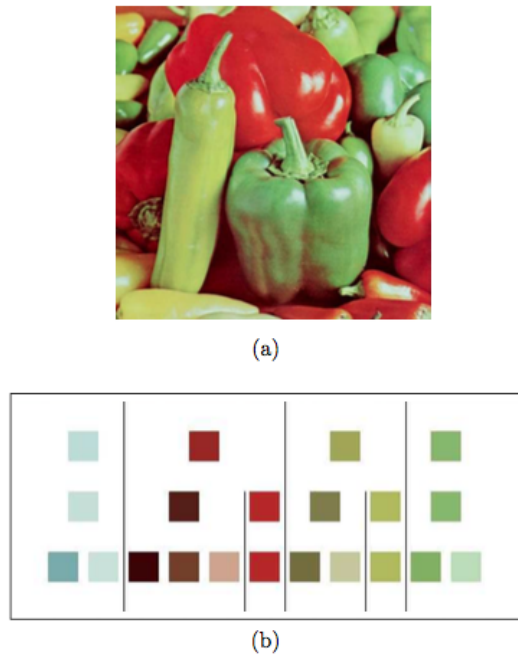


Abbildung 10: CPE Ergebnis von ACoPa. (a) Originalbild "Peppers". (b) Hierarchische Farbpalette. Die oberste Ebene identifiziert die wesentlichen Farbtöne. Ebene 2 bildet Sättigungsstufen. Ebene 3 schlüsselt das Ergebnis weiterhin in Helligkeitsstufen auf und zeigt die finalen Farben. (Quelle: [6])

Delon et al. [6] kritisieren an den bisherigen Algorithmen, dass die Anzahl n gesuchter Farben zuvor bekannt sein muss und dass Farben kleiner Bilddetails im Sinne der Definition aus 1.2 nur unzureichend repräsentiert werden. Dies begründet sich in der Tatsache, dass die bisherigen Verfahren eine Raumunterteilung vornehmen, ohne die Spezifika von Farbräumen zu beachten.

Demgegenüber stellen Delon et al. [6] und Morse et al. [21] zwei ähnliche, hierarchische Verfahren vor, die sich die Verwandtschaft der menschlichen Farbbeschreibung mit der Farbkodierung in zylindrischen Farbräumen zu Nutze machen. Sie basieren auf der hierarchischen Segmentierung eindimensionaler Histogramme. Die Algorithmen ermitteln zunächst die grundlegenden Farbtöne (Hue) der Bildvorlage, welche daraufhin sukzessive in Sättigungsstufen (Saturation) und diese wiederum in Helligkeitsstufen aufgespalten werden. Abbildung 10 veranschaulicht exemplarisch die hierarchische Arbeitsweise. Die Identifizierung verschiedener Farbtöne der Bildvorlage vereinfacht die Bildung *monochromer*, *dualer* und *triadischer* Farbschemen, welche in Unterabschnitt 2.4 vorgestellt wurden. Darum werden diese beiden Algorithmen für die weitere Diskussion ausgewählt.

Der von Delon et al. [6] vorgeschlagene Algorithmus lautet **Automatic Color Palette (ACoPa)** und verwendet den HSI-Farbraum. Er arbeitet parameterfrei und bestimmt n automatisch. Der Algorithmus von Morse et al. [21] verwendet den HLS-Farbraum und ist parameterabhängig, d.h. er benötigt Angaben in Bezug auf n . Die Autoren schlagen jedoch interessante Vorverarbeitungsschritte für die Histogramm-Bildung vor, welche die

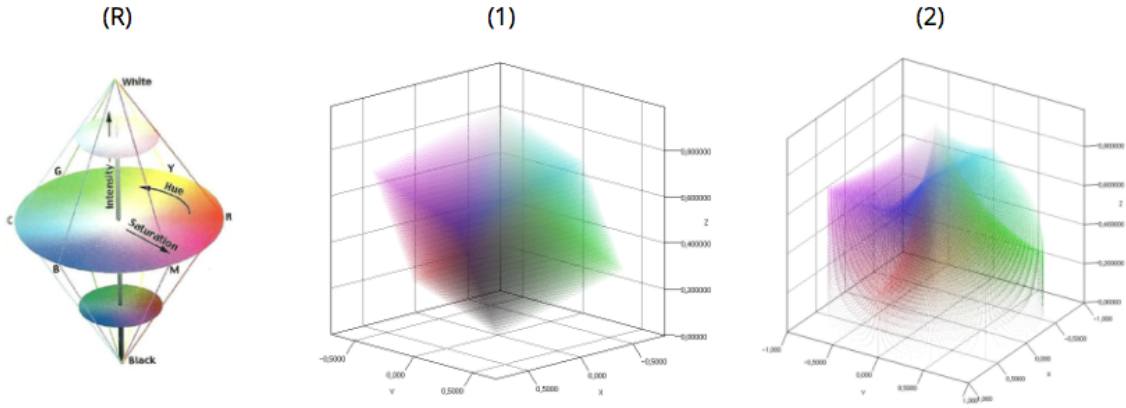


Abbildung 11: Gegenüberstellung von RGB zu HSI-Umrechnungsergebnisse. (a.) Referenz-HSI Raum. (b.) Umrechnung nach [6]. (c.) Umrechnung nach [25].

menschlichen Wahrnehmungseigenschaften stärker berücksichtigen und somit eine Brücke zu den segmentierungs-basierten Verfahren schlagen.

Um dem Ziel einer vollautomatisierten Methode zur Farbgestaltung von Webseiten aus Bildvorlagen gerecht zu werden, wird als Implementierungsgrundlage der ACoPa-Algorithmus gewählt, da er keine Nutzerinteraktion in Form der Bestimmung von n benötigt. Er wird im folgenden Unterabschnitt 4.2 vorgestellt. Daraufhin werden in Unterabschnitt 4.3 die Adaptionen dieses Algorithmus von Morse et al. [21] beschrieben.

4.2 ACoPa

Der ACoPa Algorithmus [6] basiert auf der Darstellung des Histogramms im HSI Raum. Die Konvertierung wird in Abschnitt 4.2.1 vorgestellt. Daraufhin werden Farben identifiziert, indem Segmentierungen der eindimensionalen H-, S- und I-Histogramme durchgeführt werden. Das zugehörige Verfahren wird als *Fine-to-Coarse Segmentation* bezeichnet und in Abschnitt 4.2.2 vorgestellt. Abschnitt 4.2.3 bespricht die letztendliche Bildung der Farbpalette und stellt Ergebnisse vor.

4.2.1 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:

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

Die Umrechnung gemäß eines Lehrbuchs für Farbbild-Verarbeitung [25] lautet hingegen:

$$\begin{aligned}
 I &= \frac{R + G + B}{3} \\
 S &= 1 - \frac{\min(R, G, B)}{I} \\
 H &= \arccos\left(\frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - B)}}\right)
 \end{aligned} \tag{7}$$

Abbildung 11 stellt die Umrechnungsergebnisse dem Referenz HSI Raum (R) gegenüber. Keine der Umrechnungsvorschriften führt zu einem Doppelkegel. Weder Formel 6 noch Formel 7 projiziert die Farben mit 100% Sättigung ($s = 1$) in eine Ebene. Formel 6 führt lediglich zu einer Drehung und Stauchung des RGB-Würfels, Formel 7 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 6 fortgeführt.

4.2.2 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“ [6]. Abbildung 12 zeigt das Prinzip an verschiedenen Beispielen.

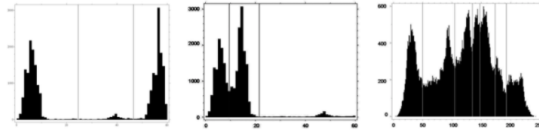


Abbildung 12: Beispiele der Segmentierung eines Histogramms in unimodale Abschnitte.
(Quelle: [6])

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 [7] 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 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 [6, 7] 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 [7] als **Fine-to-Coarse (FTC) Segmentation Algorithm** zusammengefasst. Zunächst wird s mit allen Minima des Histo-

gramms 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 13 zeigt dies an einem Beispiel.

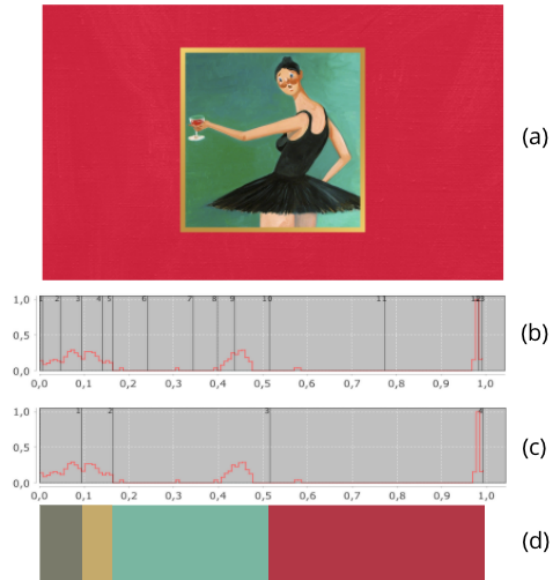


Abbildung 13: Beispiel für eine Segmentierung des Hue-Histogramms. (a) Ausgangsbild, ein Albumcover von Kanye West. (b) Hue-Histogramm (normalisiert), mit allen Minima als initiale Segmentierung. (c) Segmentierung nach Anwendung des FTC Algorithmus. (d) Farbmittelpunkte entsprechend der Samples der jeweiligen Segmente.

4.2.3 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 14 zeigt dies am Beispiel der Covers aus Abbildung 13. 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. Delon et al. [6] empfehlen zusätzlich, die erhaltenen Farben als Startpunkte für den K-Means Algorithmus zu verwenden. Abbildung 14 (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.

4.3 Vorverarbeitung des Histogramms

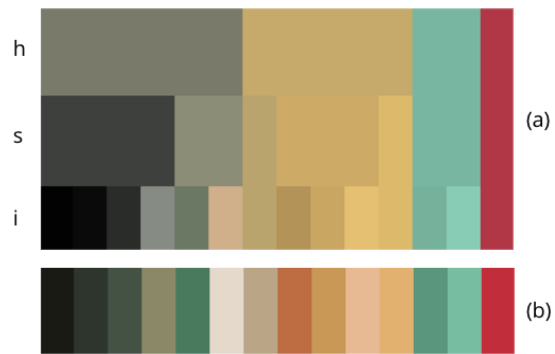


Abbildung 14: (a) Hierarchische Farbpalette des Covers aus Abbildung 13. (b) Farbpalette nach Anwendung von K-Means.

5 Ausblick

Als nächstes sind die Eigenschaften und Bedingungen der Farben zu ermitteln, die auf der Weboberfläche 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] 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/>.
- [3] J. Bezdek. Pattern recognition with fuzzy objective function algorithms. Kluwer Academic Publishers, 1981.
- [4] Alex Bigman. 6 principles of visual hierarchy for designers, 2014. URL <https://en.99designs.de/blog/tips/6-principles-of-visual-hierarchy/>.
- [5] D. Comaniciu und P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(24):603–619, 2002.
- [6] J. Delon, A. Desolneux, J. L. Lisani, und A. B. Petro. Automatic color palette. *Inverse Problems and Imaging*, 1(2):265–287, 2007.
- [7] J. Delon, A. Desolneux, J. L. Lisani, und A. B. Petro. A nonparametric approach for histogram segmentation. 16(1):253–261, 2007.
- [8] Laura Elizabeth. A simple web developer’s guide to color, apr 2016. URL <https://www.smashingmagazine.com/2016/04/web-developer-guide-color/>.
- [9] M. Gervautz und W. Purgathofer. A simple method for color quantization: Octree quantization. *Graphic Gems. Academic Press*, page 287–293, 1990.
- [10] Google. Material design style - colors, 2016. URL <https://material.io/guidelines/style/color.html>.
- [11] Zhenyu Gu und Jian Lou. Data driven webpage color design. *Computer-Aided Design*, 77:46–59, aug 2016.
- [12] Martin Hahn. *Webdesign. Das Handbuch zur Webgestaltung*. Galileo Press, first edition, 2015.
- [13] P. Heckbert. Color image quantization for frame buffer displays. *Comp. Graph*, 16:297–307, 1982.
- [14] J. Itten. *The Elements Of Color*. Van Nostrand Reinhold Company, 1970.
- [15] Ali Jahanian, Jerry Liu, Qian Lin, Daniel R. Tretter, Eamonn O’Brien-Strain, Seungyon Lee, Nic Lyons, und Jan P. Allebach. Automatic design of colors for magazine covers. In *SPIE Proceedings. Imaging and Printing in a Web 2.0 World IV.*, volume 8664, 2013.
- [16] K. Kanjanawanishkula und B. Uyyanonvarab. Novel fast color reduction algorithm for time-constrained applications. *Journal of Visual Communication and Image Representation*, 16:311–332, 2005.
- [17] Ranjitha Kumar, Arvind Satyanarayan, Cesar Torres, Maxine Lim, Salman Ahmad, Scott R. Klemmer, und Jerry O. Talton. Webzeitgeist: Design mining the web. In *ACM Human Factors in Computing Systems (CHI)*, 2013.
- [18] S. Lin und P. Hanrahan. Modeling how people extract color themes from images. In *CHI Conference on Human Factors*, 2013.

- [19] Sharon Lin, Daniel Ritchie, Matthew Fisher, und Pat Hanrahan. Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Transactions on Graphics*, 32(4), jul 2013.
- [20] J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1 of 14, California, USA, 1967.
- [21] Bryan S. Morse, Daniel Thornton, Qing Xia, und John Uibel. Image-based color schemes. In *IEEE International Conference on Image Processing (ICIP)*, volume 3, page 497–500, 2007.
- [22] Albert H. Munsell und F. Birren. *A grammar of color: a basic treatise on the color system of Albert H. Munsell*. Van Nostrand Reinhold Company, 1969.
- [23] Peter O'Donovan, Aseem Agarwala, und Aaron Hertzmann. Color compatibility from large datasets. In *ACM SIGGRAPH Conference*, page 63:1–63:12, 2011.
- [24] P. S. Periasamy. A probabilistic adaptive color reduction algorithm for single multiple images with and without transparency. Anna University, Faculty of Information and Communication Engineering, 2013.
- [25] K. N. Plataniotis und A. N. Venetsanopoulos. *Color Image Processing and Applications*. Springer Monograph, 2000.
- [26] Marc Schenker. Designing with color contrast for improved usability, apr 2016. URL <https://www.awwwards.com/designing-with-color-contrast-for-improved-usability.html>.
- [27] Torsten Stapelkamp. *Web X.0. Erfolgreiches Webdesign und professionelle Webkonzepte*. Springer X.media.press, 2010.
- [28] W3C. Wcag contrast ratio, 2008. URL <https://www.w3.org/TR/2008/REC-WCAG20-20081211/Overview.html#contrast-ratiodef>.
- [29] W3C. Wcag relative luminance, 2008. URL <https://www.w3.org/TR/2008/REC-WCAG20-20081211/relative-luminance.xml>.
- [30] W3C. Css color module level 3, jun 2011. URL <https://www.w3.org/TR/css3-color/>.
- [31] W3C. Css selectors level 3, sep 2011. URL <https://www.w3.org/TR/2011/REC-css3-selectors-20110929/>.
- [32] S. J. Wan, P. Prusinkiewicz, und S. K. M. Wong. Variance-based color image quantization for frame buffer display. *Color Res. Appl.*, 15:52–58, 1990.
- [33] Arthur R. Weks und G. Eric Hague. Color segmentation in the hsi color space using the k-means algorithm. In *Proceedings of SPIE*, volume 3026, 1997.