

Methoden zur automatischen Farbkonfiguration von Weboberflächen aus Bildvorlagen

Masterproject Zwischenabgabe I

Philipp Anders

2. 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 Delon et al. [5] ausgewählt und implementiert, da dessen Ergebnis den aus Styleguides bekannten Farbpaletten-Definitionen in Form von Color Swatches ähnelt.

1 Einleitung

1.1 Problemmodellierung

Der CSS-Standard zur Beschreibung des Erscheinungsbildes von Webseiten definiert mehr als 10 Eigenschaften zur farblichen Gestaltung von Texten, Hintergründen, Rahmen und anderen HTML-Elementen [26]. Ziel dieser Arbeit ist die automatisierte Bestimmung der Farbwerte von Elementen in Websites. Damit ordnet sich diese Arbeit in das Gebiet automatisierten Farbgestaltung (**Color Design Automation**) ein. Betrachtet werden all jene Oberflächenelemente, die visuell als Blöcke wahrgenommen werden (d.h. die Hintergrundfarbe "**background-color**" von `<div>`, `<button>`, `<p>` etc.) , sowie Texte (d.h. die Vordergrundfarbe "**color**" der Blöcke sowie von `<h1>`, `<a>`, etc.). Ziel ist also die Farbbestimmung von **Blöcken** und **Texten**, Rahmen und andere Textdekoration werden dabei ignoriert.

Üblicherweise werden mehrere Elemente eines HTML-Dokuments auf die gleichen Farben abgebildet. Beispielsweise können alle Links (Textfarbe) sowie Buttons (Hintergrundfarbe) die gleiche Farbe erhalten. Im Folgenden wird eine Menge von Elementen einer Webseite mit gleicher Farbabbildung als **Color Group** CG bezeichnet [siehe auch 9, 18]. Die Menge aller Color Groups einer Webseite wird als CGs formalisiert. Dabei gilt $|CGs| = k$, eine Webseite beinhaltet also k Color Groups. Abbildung 1 veranschaulicht dies anhand einer exemplarischen Web-

seite für Musikstreaming, welche sieben Color Groups beinhaltet.

Als vorbereitender Schritt wird der Suchraum zur Identifizierung geeigneter Farben durch die Zusammenstellung einer sogenannten **Farbpalette** verkleinert [9]. Dabei handelt es sich um eine Farbmenge $P = \{c_1, c_2, \dots, c_n\}$ 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 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 Accessibility (Lesbarkeit) und Usability (Benutzerfreundlichkeit) gewährleistet. Eine intuitiv widersinnige Färbung wäre 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 P auf einer Bildvorlage basieren. So wird eine Harmonisierung des visuellen Eindrucks einer Weboberfläche und der darin enthaltenen 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 ermittelt wird. Ein Anwendungsbeispiel hierfür ist die farbliche Anpassung einer Seite für Musikstreaming an das gespielte Albumcover. Abbildung 1 zeigt hierfür ein Gegenbeispiel: keine der Farben des Cover taucht in der umgebenen Weboberfläche auf.

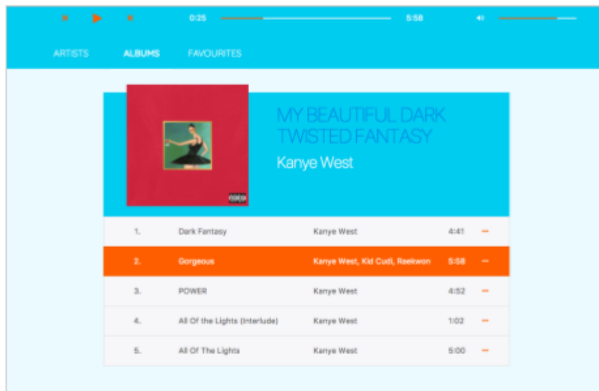


Abbildung 1: Beispiel der Farbgestaltung einer exemplarischen Weboberfläche für Musik-Streaming. Oben: Originales Farbdesign. Unten: Hervorhebung der sieben Color Groups. Elemente einer Color Group erhalten die gleiche Farbe. Blöcke sind umrahmt.

1.2 Color Palette Estimation

Die Zusammenstellung einer Farbpalette aus einer Bildvorlage wird von Delon et al. [5] als **Color Palette Estimation (CPE)** bezeichnet und als die Repräsentation eines Bildes mit einer minimalen Menge von Farben beschrieben. Diese ist dann minimal, wenn redundante Farben reduziert und die seltenen Farben der für die Wahrnehmung wichtigen Objekte erhalten bleiben. Formale Kriterien werden von den Autoren jedoch nicht geliefert. Abbildung 2 veranschaulicht diese intuitive Definition am Beispiel eines Bildes mit einem Marienkäfer, dessen Sichtbarkeit von der Wahl der Farbpalette abhängt.

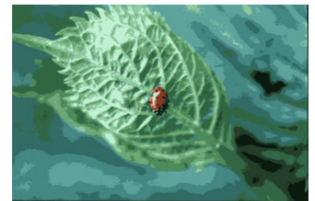
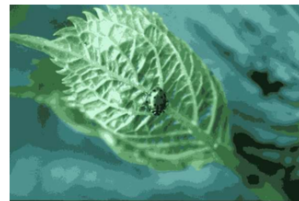


Abbildung 2: Beispiel für die Einfärbung eines Bildes mit unterschiedlichen Farbpaletten der Größe 12. Oben: Originalbild. Links: Farbpalette ohne rote Farbtöne. Rechts: Farbpalette mit roten Farbtönen, wodurch der Marienkäfer erkennbar ist (Quelle: [5])

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) [27]. Aus diesem Kontext kommt das formale Kriterium der Summe des quadratischen Fehlers, welcher in diesem Anwendungsfall auch als *Recoloring Error* bezeichnet wird [17].

Da Grafikpuffer mittlerweile über ausreichend Kapazität verfügen, liegt die Anwendung der CPE in anderen Bereichen wie z.B. der farbasierten Indizierung von Grafiken in Datenbanken oder der Zusammenstellung von Farbthemen zu Gestaltungszwecken. Lin and Hanrahan [17] zeigen, dass in diesem Kontext der Recoloring Error keine geeignete Metrik zur Beurteilung der Güte einer Farbpalette in Bezug auf das Ausgangsbild 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 das Ausgangsbild subjektiv ist und vom Anwendungsbezug abhängt. Aus diesem Grund wird für die Güte der zu ermittelnden Farbpalette C keine objektive Bewertungsfunktion herangezogen. Stattdessen wird die Farbpalette in Hinblick auf ihre Zweckmäßigkeit zur farblichen Gestaltung einer Webseite bewertet. Das heißt, die aus der Grafik hervorgehende Farbpalette muss eine Teilmenge an Farben beinhalten, mit welcher die

automatisierte Farbgestaltung einer Webseite lösbar ist. Hierfür müssen Kriterien abgeleitet, indem Prinzipien für eine funktionale Gestaltung von Webseiten sowie Farbdefinitionen aus Style Guides analysiert werden. Davon ausgehend wird eine geeignete Herangehensweise zur Bestimmung einer solchen Farbpalette aus einer Bildvorlage ermittelt.

2 Literaturbesprechung

Im Bereich der automatisierten Farbgestaltung hat in den letzten Jahren viel Forschung stattgefunden, wobei überwiegend Methoden des maschinellen Lernens zur Problemlösung eingesetzt wurden.

O'Donovan et al. [22] haben 2011 eine Grundlage für spätere Systeme zur automatisierten Farbgestaltung gelegt, indem sie ein Regressionsmodell zur Bewertung der Ästhetik von Farbpaletten der Größe 5 entwickelten. Hierfür wurde in 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 Farbtheorie nicht zur Vorhersage von Farbharmonie geeignet sind. Hierzu zählen zum Beispiel die Farbton-Schablonen von Itten [14] oder Munsell and Birren [20]. 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 im tatsächlichen Anwendungskontext, da die Auswirkungen der räumlichen Ausprägung von Farben nicht überprüft werden [9, 18]. Eine Matlab-Implementierung des Modells zur Überprüfung von Farbharmonien steht öffentlich zum Download bereit².

Lin et al. [18] haben sich 2013 mit einer Lösung für eine sehr grundlegende Form für ein 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. Eine vorgeschlagene Anwendung stellt die Umkehrung des Ziels dieser Arbeit dar: Die Anpassung eines Musters an das Farbschema einer Webseite. Die Autoren stellen Eigenschaften zur Bestimmung von Hintergrund- und Vordergrundfarben sowie zur Kontrastberechnung vor, die für diese Arbeit genutzt werden können. Zur Lösungssuche wird *Factor Graph* modelliert, welcher ebenfalls ein Ansatz für die vorliegende Arbeit darstellt. Das Modell von O'Donovan et al. [22] wird als externer Bestandteil

des Graphen hinzugefügt, um eine globale Kompatibilität der eingesetzten Farben zu gewährleisten.

Gu and Lou [9] haben sich 2016 mit der automatisierten Farbgestaltung von Webseiten. Durch die Auswertung von 500 Webseiten wurde ein probabilistisches Modell als Optimierungsproblem mit mehreren Zielen formuliert. Eine der Zielfunktionen gewährleistet ausreichenden Kontrast zwischen den Webelementen. Eine andere Zielfunktion passt die Farbgestaltung an ein Schlüsselwort an (z.B. *Business* oder *Fresh*). Die letzte Zielfunktion gewährleistet Farbharmonie, was wiederum durch das Modell von O'Donovan et al. [22] implementiert wird. Die Optimierung wird durch eine lexikographische Strategie realisiert, bei welcher in Interaktion mit einem Gestalter die Zielfunktionen nacheinander angewendet werden. In einer Anwendung extrahieren die Autoren ebenfalls 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 Gegenstand dieser Arbeit erfordert der vorgestellte Prozess jedoch eine Nutzerinteraktion aufgrund der lexikographischen Strategie. Somit handelt es sich eher um ein Unterstützungswerkzeug für Gestalter und kein System zur automatisierten Bestimmung einer Farbgestaltung zur Laufzeit. Die Autoren stellen eine Systemarchitektur vor, welche als Grundlage für die vorliegende Arbeit adaptierbar ist.

Jahanian et al. [15] haben sich 2013 mit der automatisierten Farbgestaltung von Magazin-Covers auseinander gesetzt. Vergleichbar mit [9] wird auch hier die Optimierung des Farbkontrasts, der Farbharmonie und der Farbsemantik verfolgt. Im Gegensatz zu den bisherigen Lösungen wird hier allerdings mit expliziten Modellen anstatt mit Trainingsdaten gearbeitet. Über Flowcharts vermitteln die Autoren Lösungsprozeduren zur Suche geeigneter Schriftfarben und Stellen konkrete Grenzwerte vor, die für diese Arbeit übernommen werden können.

3 Problemlösungsansatz und Systemarchitektur

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

Es ist ein Modell zu entwickeln, welches die Abbildung $CGs \rightarrow P$ beschreibt. Gu and Lou [9] stellen für die automatisierte Farbgestaltung zwei grundlegende Herangehensweisen vor:

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

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

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

Die regelbasierten Modelle treffen präzise Aussagen im Gegensatz zu probabilistischen Modellen, tendieren jedoch zur zu starken Abstraktion vom Anwendungskontext. Im Gegensatz dazu stützen sich die Modelle des datengetriebenen Ansatzes auf reale Daten und tendieren daher zu robusteren Ergebnissen in der Anwendungsdomäne [9]. Da bei den probabilistischen Modellen potentielle Lösungen bewertet und gegeneinander abgewogen werden müssen, sind hohe Laufzeiten möglich. Die bereits entwickelte Lösung von Gu and Lou [9] zur automatisierten Farbgestaltung von Websites benötigt selbst nach Optimierung des Suchverfahrens 2 Stunden zur Konvergenz. Außerdem ist deren vorgestellter Prozess zur Anpassung der Farbgestaltung einer Webseite an nicht im eigentlichen Sinne vollautomatisch: Einerseits erfordert die Verwendung des K-Means Algorithmus zur CPE die Bestimmung der Farbanzahl (k) vom Gestalter, andererseits erfordert die lexikographische Strategie einen Nutzerinteraktion während des Optimierungsprozess.

Aus diesen Gründen wird sich diese Arbeit auf einen **regelbasierten Ansatz ohne Nutzerinteraktion** konzentrieren, d.h. vollautomatisiert. Auch ohne die Auswertung großer Datenmengen existieren quantitative Modelle, um die Accessibility einer Webseite zur gewährleisten. Beispielsweise geben die *Web Content Accessibility Guidelines* [1] quantitative Grenzwerte für die Lesbarkeit von Text 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 [13]. Jahanian et al. [15] haben gezeigt, dass eine Lösung eines solchen Constraint-Systems beispielsweise über Faktor-Graphs möglich ist.

Nicht jede Webseite hat die gleiche Anzahl Color Groups. Um die Komplexität des Constraint-Systems zu reduzieren, wird auf die Feststellung der Color Groups einer Webseite verzichtet. Stattdessen wird von einer Webseite als Layout **abstrahiert**. Ein Layout definiert, 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 also von den betreffenden Elementen abstrahiert.

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 O'Donovan et al. [22] verfolgt. Sie verwenden ihr Regressionsmodell zur Bewertung der Farbharmenie, 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 Farben in P . Dieser Ansatz wird unter anderem von Obrador [21] 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 komplementär zum Hintergrund des Bildes gewählt, welche via Bildsegmentierung bestimmt wurde. 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 Caching bereits berechneter Farbpaletten das Einsatzpotential der automatisierten Farbgestaltung zur Laufzeit.

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. Für die exemplarischen Werte $|CGs| = k = 5$ und $|P| = n = 10$, d.h. 5 Colorgroups (z.B. Text, Texthintergrund, Buttons, Navigations-Hintergrundfarbe und Footer-Hintergrundfarbe) sowie eine Farbpalette mit 10 Farben, ergeben sich $\binom{10}{5}5! = 30.240$ mögliche Kombinationen. Es ist ein effektives Suchverfahren zur Ermittlung von $CGs \rightarrow P$ zu identifizieren. Die bereits vorgestellte hierarchische Ansatz von Obrador [21] stellt hierfür einen Ansatz dar, bei welchem wichtige Farbuordnungen 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 als Color Palette Estimation (CPE) bezeichnet. Die entstandene Palette wird zum Suchraum für die Color Groups, welche durch ein Layout definiert werden. Da bei dieser Abbildung der Wertebereich einer Color Group auf die Elemente der Farbpalette 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 ermitteln.

3.1 Farbpaletten in Style Guides

Style Guides zur Oberflächengestaltung beinhalten unter anderem Richtlinien für den Farbeinsatz. Dabei werden i.d.R. eine Obermenge von Farben definiert und mit Verwendungshinweisen verbunden. Einige Farben, wie z.B. die Logo-Farbe, stehen hierbei unter strikten Restriktionen. Davon abgesehen wählt der Designer eine Untermenge von Farben der Farbpalette aus dem Style Guide und legt selbst Einsatzregeln fest, wie z.B. die Farbe für Interaktionselemente.

Die Apple *iOS Human Interaction Guidelines* legen eine Farbpalette von 8 strahlenden Farben fest, die sich aufgrund ihrer Intensität ausschließlich für Interaktionselemente und ausgewählte Komponenten wie z.B. Statusleisten eignen [2]. Demgegenüber beschreibt Google im *Material Design Style Guide* Farben in Form eines Farbtons in abgestuften Schattierungen. Dieses Konzept wird als *Color Swatch* bezeichnet. Hierbei handelt es sich um eine Erweiterung der Farbpalette, die dem Designer mehr Spielraum bei der farblichen Komposition einräumt. Ein Beispiel hierfür zeigt Abbildung 4, auf welcher ein Auszug der Color Swatches und ein Einsatzbeispiel für den Umgang mit Schattierungen gezeigt wird. Ein anderes Beispiel hierfür zeigt das Corporate Design Handbuch der HTWK Leipzig. Der Farb-Guide stellt Color Swatches mit zwei bis drei Schattierungen bereit [12].

Das Konzept der Color Swatches wird in dieser Arbeit für die CPE bevorzugt, da über die Farbeigenschaften eines Bildes keine Annahmen getroffen werden können. Die Bereitstellung eines Farbtons in verschiedenen Schattierungen ermöglicht jedoch den flexiblen Einsatz von Farben, der bei Bildern mit wenig Farben notwendig ist. So

ist z.B. der selbe Farbton sowohl für Interaktionselement, oder in einer hellen Schattierung für den Hintergrund einsetzbar.

4 Algorithmen zur Color Palette Estimation

Im Folgenden wird ein 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 [17].

Kanjanawanishkula und Uyyanonvarab [16] 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 [11] oder Octree [7]. Sie zerteilen den Farbraum sukzessiv in disjunkte Teilräume und unterstellen den Clustern dabei eine Würfelform. Ergebnis der Verarbeitung ist ein Dendrogramm, wobei die Blätter die Farben der Farbpalette repräsentieren. Ein Schnitt des Dendrogramms entspricht einer Partitionierung des Raums, welche jedoch

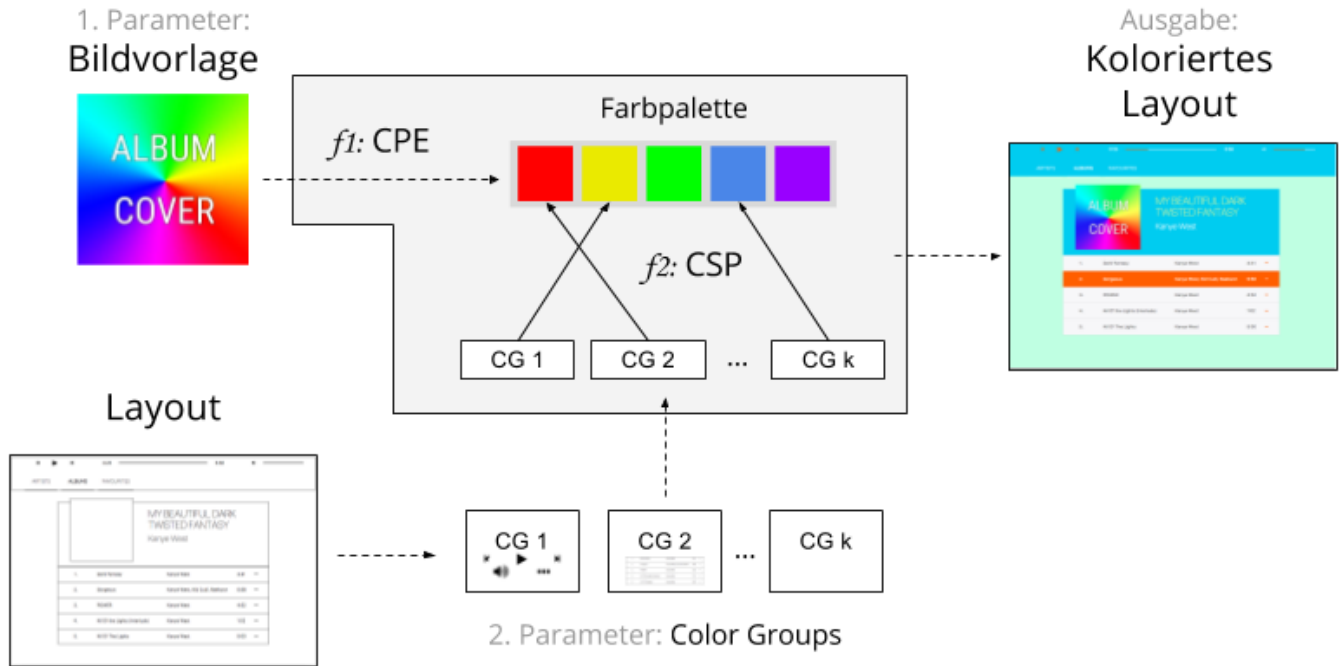


Abbildung 3: System-Architektur. Der grau unterlegte Kasten visualisiert den Bereich, für den algorithmische Lösungen gefunden werden müssen. Die Eingabeparameter sind eine Bildvorlage und eine Menge von Color Groups, welche von einem Layout definiert werden. Das System setzt zwei Suchverfahren um, welche 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 Farbabbildung gefunden wurde, wird das kolorierte Layout ausgegeben.

auch direkt durch die iterativ arbeitenden Algorithmen erreichbar ist [5]. Diese Verfahren werden darum auch als *partitionierend* [5] oder auch *Post-Clustering* [16] 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 [19, 28] oder Fuzzy C-Means [3]. Andere analysieren das Histogramm auf dichte bzw. weniger dichte Regionen, wie z.B. Mean-Shift [4]. Eine detailliertere Vorstellung von Algorithmen zur CPE bietet [23].

Delon et al. [5] 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 Bilddetails 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) auf. Abbildung 5 veranschaulicht exemplarisch die hierarchische Arbeitsweise, bei der in jeder Ebene zusätzliche Sättigun-

gen 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 Delon et al. [5] 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

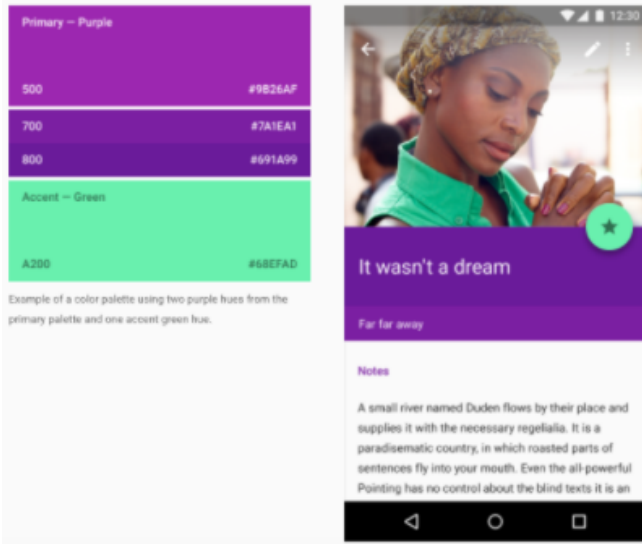


Abbildung 4: Auszug der Color Swatches in der Farbpalette des Google Material Designs (oben) mit Einsatzbeispiel (unten). (Quelle: [8])

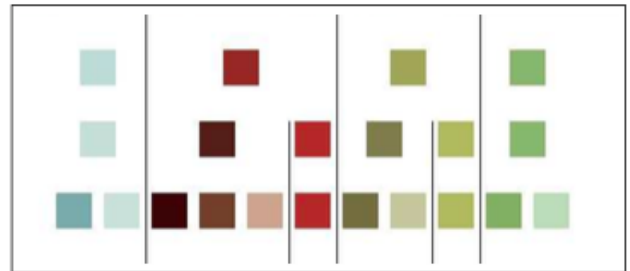
erforderliche Obermenge zur Bildung der finalen Farbpalette bereitgestellt, wenn das Bild ausreichend viele Farben enthält. Das erzwingen eines großen Farbpalette mit anderen Clusteringverfahren, z.B. über einen pauschal großen K Parameter bei K-Means, führt hingegen unter Umständen zu einer Partitionierung des Farbraums, die nicht der Clusterstruktur des Histogramms entspricht.

5 ACoPa

Im Folgenden wird die grundlegende Arbeitsweise des ACoPa Algorithmus nach Delon et al. [5] vorgestellt. Dabei werden die Herausforderungen, die bei der Implementierung aufgetreten sind, besprochen. Abschließend werden exemplarisch Anwendungsergebnisse präsentiert.



(a)



(b)

Abbildung 5: CPE Ergebnis von ACoPa. (a) Originalbild "Peppers" (b) Hierarchische Farbpalette. Die unterste Ebene zeigt die finalen Farben. (Quelle: [5])

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:

$$I = \frac{R + G + B}{3}$$

$$S = \sqrt{(R - I)^2 + (G - I)^2 + (B - I)^2}$$

$$H = \arccos\left(\frac{(G - I) - (B - I)}{S\sqrt{2}}\right) \quad (1)$$

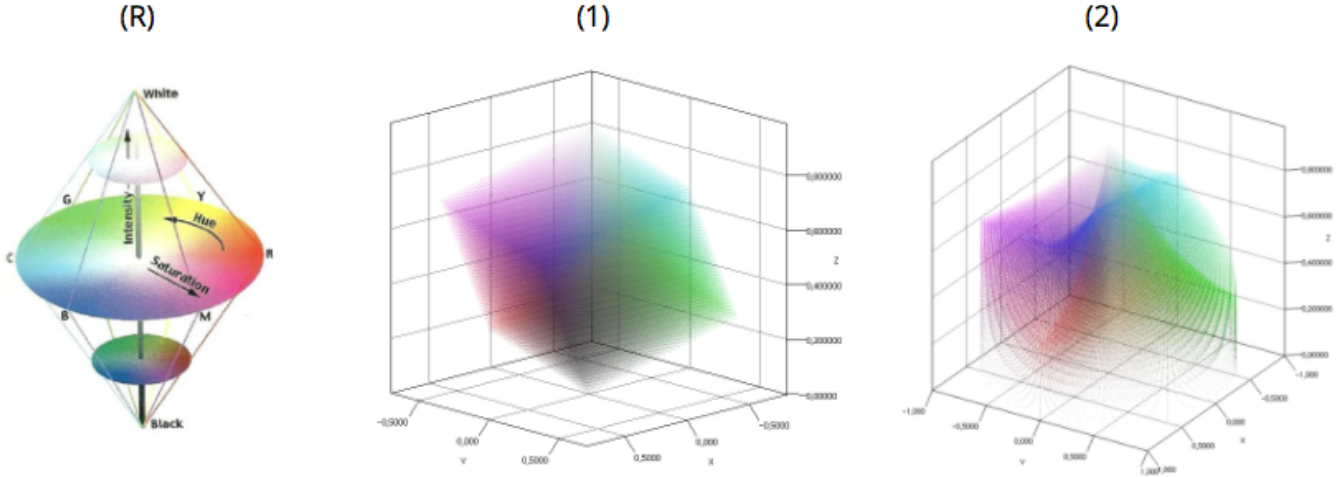


Abbildung 6: Gegenüberstellung von RGB zu HSI-Umrechnungsergebnisse. (a.) Referenz-HSI Raum. (b.) Umrechnung nach [5]. (c.) Umrechnung nach [24].

Die Umrechnung gemäß eines Lehrbuchs für Farbbild-Verarbeitung [24] lautet hingegen:

$$I = \frac{R + G + B}{3}$$

$$S = 1 - \frac{\min(R, G, B)}{I} \quad (2)$$

$$H = \arccos\left(\frac{\frac{1}{2}((R - G) + (R - B))}{\sqrt{(R - G)^2 + (R - B)(G - B)}}\right)$$

Abbildung 6 stellt die Umrechnungsergebnisse dem Referenz HSI Raum (R) gegenüber. Keine der Umrechnungsvorschriften führt zu einem Doppelkegel. Weder Formel 1 noch Formel 2 projiziert die Farben mit 100% Sättigung ($s = 1$) in eine Ebene. Formel 1 führt lediglich zu einer Drehung und Stauchung des RGB-Würfels, Formel 2 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 1 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“ [5]. Abbildung 7 zeigt das Prinzip an verschiedenen Beispielen.

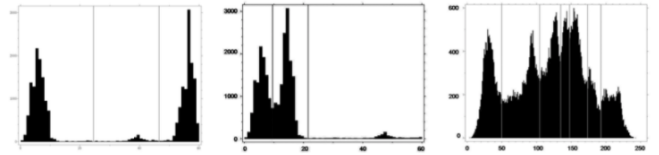


Abbildung 7: Beispiele der Segmentierung eines Histogramms in unimodale Abschnitte. (Quelle: [5])

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 [6] 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 [5, 6] 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 [6] 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 Ver-

schmelzung mehr möglich ist. Die Repräsentanten eines Segments werden durch Mittelung der Samples gebildet, die zum jeweiligen Segment gehören. Abbildung 8 zeigt dies an einem Beispiel.

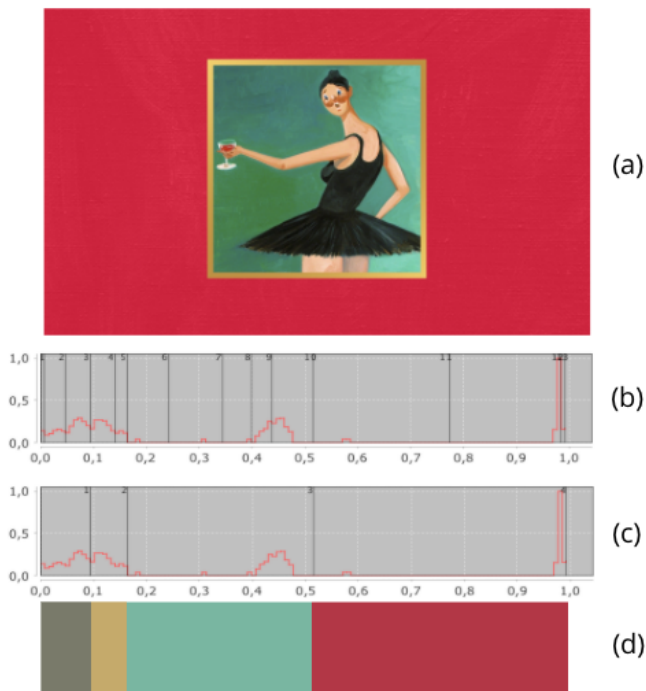


Abbildung 8: Beispiel für eine Segmentierung des Hue-Histogramms. (a) Ausgangsbild, ein Albumcover von Kanye West. (b) Hue-Histogramm (normalisiert), mit allen Minima als initiale Segmentierung. (c) Segmentierung nach Anwendung des FTC Algorithmus. (d) Farbmittelpunkte entsprechend der Samples der jeweiligen Segmente.

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 9 zeigt dies am Beispiel der Covers aus Abbildung 8. 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. Die Farbpalette spiegelt die in Abschnitt 3.1 definierten Color Swatches wieder. Delon et al. [5] empfehlen zusätzlich, die erhaltenen Far-

ben als Startpunkte für den K-Means Algorithmus zu verwenden. Abbildung 9 (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.



Abbildung 9: (a) Hierarchische Farbpalette des Covers aus Abbildung 8. (b) Farbpalette nach Anwendung von K-Means.

6 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] Apple. ios human interface guidelines - color, 2016. URL <https://developer.apple.com/ios/human-interface-guidelines/visual-design/color/>.
- [3] J. Bezdek. Pattern recognition with fuzzy objective function algorithms. Kluwer Academic Publishers, 1981.
- [4] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 5(24):603–619, 2002.
- [5] J. Delon, A. Desolneux, J. L. Lisani, and A. B. Petro. Automatic color palette. *Inverse Problems and Imaging*, 1(2): 265–287, 2007.
- [6] J. Delon, A. Desolneux, J. L. Lisani, and A. B. Petro. A nonparametric approach for histogram segmentation. 16(1): 253–261, 2007.

- [7] M. Gervautz and W. Purgathofer. A simple method for color quantization: Octree quantization. *Graphic Gems. Academic Press*, page 287–293, 1990.
- [8] Google. Material design style - colors, 2016. URL <https://material.io/guidelines/style/color.html>.
- [9] Zhenyu Gu and Jian Lou. Data driven webpage color design. *Computer-Aided Design*, 77:46–59, aug 2016.
- [10] Martin Hahn. *Webdesign. Das Handbuch zur Webgestaltung*. Galileo Press, first edition, 2015.
- [11] P. Heckbert. Color image quantization for frame buffer displays. *Comp. Graph*, 16:297–307, 1982.
- [12] Wirtschaft und Kultur Leipzig Hochschule für Wirtschaft, Technik. Corporate design handbuch - basiselemente - farben, 2016. URL <http://marketing.htwk-leipzig.de/de/corporate-design/handbuch/basiselemente/farben/>.
- [13] Petra Hofstedt and Armin Wolf. *Einführung in die Constraint Programmierung*. Springer eXamen.press, 2007.
- [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, and 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 and B. Uyyanonvarab. Novel fast color reduction algorithm for time-constrained applications. *Journal of Visual Communication and Image Representation*, 16: 311–332, 2005.
- [17] S. Lin and P. Hanrahan. Modeling how people extract color themes from images. In *CHI Conference on Human Factors*, 2013.
- [18] Sharon Lin, Daniel Ritchie, Matthew Fisher, and Pat Hanrahan. Probabilistic color-by-numbers: Suggesting pattern colorizations using factor graphs. *ACM Transactions on Graphics*, 32 (4), jul 2013.
- [19] 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.
- [20] Albert H. Munsell and F. Birren. *A grammar of color: a basic treatise on the color system of Albert H. Munsell*. Van Nostrand Reinhold Company, 1969.
- [21] Pere Obrador. Automatic color scheme picker for document templates based on image analysis and dual problem. Technical report, Hewlett-Packard Laboratories, Palo Alto, CA, USA, 2006.
- [22] Peter O’Donovan, Aseem Agarwala, and Aaron Hertzmann. Color compatibility from large datasets. In *ACM SIGGRAPH Conference*, page 63:1–63:12, 2011.
- [23] 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.
- [24] K. N. Plataniotis and A. N. Venetsanopoulos. *Color Image Processing and Applications*. Springer Monograph, 2000.
- [25] Carla Brambilla Silvia Zuffi, Giordano Beretta. A color selection tool for the readability of textual information on web pages. Technical report, Hewlett-Packard Laboratories, Palo Alto, CA, USA, 2005.
- [26] W3C. Css color module level 3, jun 2011. URL <https://www.w3.org/TR/css3-color/>.
- [27] S. J. Wan, P. Prusinkiewicz, and S. K. M. Wong. Variance-based color image quantization for frame buffer display. *Color Res. Appl.*, 15:52–58, 1990.
- [28] Arthur R. Weks and G. Eric Hague. Color segmentation in the hsi color space using the k-means algorithm. In *Proceedings of SPIE*, volume 3026, 1997.