

Methoden zur automatischen Farbkonfiguration von Weboberflächen aus Bildvorlagen

Masterproject Zwischenabgabe I

Philipp Anders

28. Februar 2017

Ziel des Projekts ist die automatische Farbkonfiguration der Oberflächenelemente von Websites (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. [4] ausgewählt und implementiert, da dessen Ergebnis den aus Styleguides bekannten Farbpaletten-Definitionen in Form von Color Swatches ähnelt.

1 Problemmodellierung

1.1 Problembeschreibung

Unter Farbkonfiguration von Websites wird die Zuordnung der Farben einer Farbpalette $C = \{c_1, c_2, \dots, c_n\}$ und einer festen Menge semantischer HTML-Oberflächenelemente $O = \{\text{Text, Hintergrund, Buttons, } \dots\}$ verstanden. Bei den Farben $c_{1..n}$ handelt es sich um 3-Tupel, wobei der Wertebereich der Komponenten abhängig vom gewählten Farbraum ist. Die Abbildung zwischen O und C ist surjektiv, d.h. jede Farbe der Farbpalette soll Verwendung finden, mehrere Oberflächenelemente können jedoch die gleiche Farbe besitzen.

Die Farbpalette soll sich dabei an einer Bildvorlage orientieren. Die Zusammenstellung einer Farbpalette aus einer Bildvorlage wird von Delon et al. [4] 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 1 veranschaulicht diese intuitive Definition am Beispiel eines Bildes mit einem Marienkäfer, dessen Sichtbarkeit von der Wahl der Farbpalette abhängt.

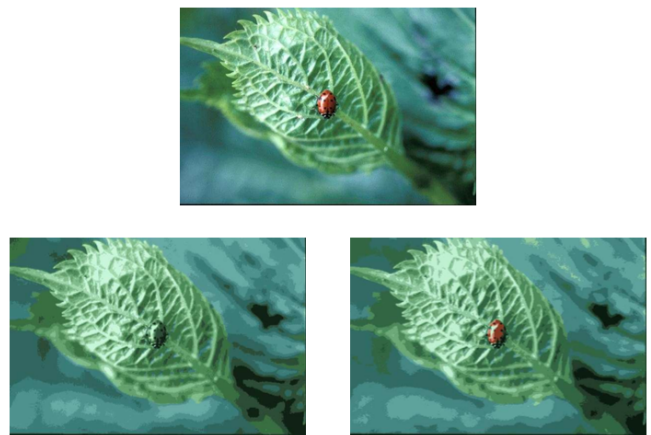


Abbildung 1: Beispiel für die Einfärbung eines Bildes mit unterschiedlichen Farbpaletten der Größe 12. Oben: Originalbild. Links: Farbpalette ohne rote Farbtöne. Rechts: Farbpalette mit roten Farbtönen, wodurch der Marienkäfer erkennbar ist (Quelle: [4])

Historisch geht die CPE aus der Farbquantisierung hervor, bei der die Farben von Grafiken aufgrund der früher zu kleinen Kapazität von Grafikpuffern vor deren Anzeige reduziert (Farbreduktion) und dann auf die reduzierte Farbpalette abgebildet werden mussten (Quantisierung) [18]. 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. Lin und Hanrahan [11] 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 exemplarisch ermittelt, wie Styleguides Farbpaletten für die Oberflächengestaltung beschreiben.

1.2 Farbpaletten in Styleguides

Styleguides 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 Styleguide 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 [1]. Demgegenüber beschreibt Google im *Material Design Styleguide* 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

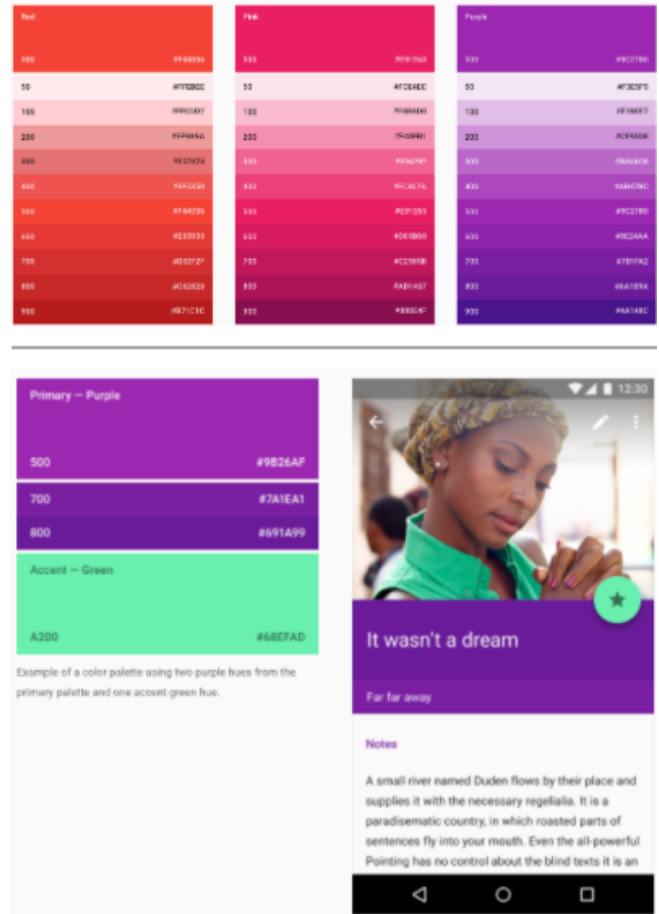


Abbildung 2: Auszug der Color Swatches in der Farbpalette des Google Material Designs (oben) mit Einsatzbeispiel (unten). (Quelle: [7])

Beispiel hierfür zeigt Abbildung 2, 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 [9].

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 Farbtone sowohl für Interaktionselemente, oder in einer hellen Schattierung für den Hintergrund einsetzbar.

1.3 Problemlösungsansatz

Die Durchführung einer CPE zur Ermittlung von $C = c_{1...n}$ reicht zur Problemlösung noch nicht aus. Außerdem

muss eine Zuordnung zwischen C und den Oberflächenelementen stattfinden. Hierbei ist zu beachten, dass die Farben je nach Einsatzzweck bestimmte Eigenschaften besitzen müssen. Beispielsweise kommen für Schaltelemente wie z.B. Buttons strahlende Farben in Frage. Das machen unter anderem die Standard-Buttonfarben der populären CSS-Frameworks Bootstrap¹, Foundation² oder Semantic-UI³ deutlich. Außerdem sind Bedingungen zu beachten, die zwischen Farben gelten sollen, wie z.B. ausreichender Kontrast zwischen Schrift und Hintergrund. Diese zu erfüllenden Eigenschaften und Bedingungen werden im Folgenden als **Constraints** bezeichnet.

Zur CPE in Verbindung mit Constraints werden im Folgenden zwei Herangehensweisen vorgestellt:

1. **Constraints-First:** Die Constraints werden schon während der Extraktion der Farbpalette berücksichtigt. Dieser Ansatz wird unter anderem von O'Donovan et al. [14] verfolgt. Durch Training an umfangreichen Datenbeständen von Farbpaletten-Communities wie z.B. Adobe Color⁴ wurde ein Modell zur Bewertung der Farbpaletten-Ästhetik entwickelt. Dieses Modell wird zur Formulierung einer alternativen Optimierungsfunktionen zur Suche von C im Farbraum verwendet, die neben der Minimierung des Recoloring Errors auch ästhetische Faktoren durch eine Bewertungsfunktion berücksichtigt.
2. **Constraints-Last:** Es wird zunächst eine Obermenge von Farben $C_s = \{c_1, \dots, c_m\}$ mit $n < m$ ermittelt. Aus dieser wird daraufhin die finale Farbpalette C durch Anwendung der Constraints ermittelt. Es gilt: $C \subseteq C_s$. Dieser Ansatz wird unter anderem von Obrador [13] verfolgt. Hierbei wird die Hintergrundfarbe eines Dokuments komplementär zur via Bildsegmentierung ermittelten Hintergrundfarbe eines Bildes gewählt. Daraufhin wird die Textfarbe unter Beachtung des Kontrastes aus C_s selektiert.

Im Rahmen des Projekts wird die zweite Herangehensweise (Constraints-Last) bevorzugt, da sie eine größere Flexibilität gewährleistet. Zum gegenwärtigen Zeitpunkt wird Angenommen, dass die Anforderungen an die Farbpalette in Bezug auf Anzahl und Constraints bis auf wenige, bereits untersuchte Sachverhalte wie z.B. der Schrift-Hintergrund-Kontrast [17], weniger durch wissenschaftliche Literatur als vielmehr durch Webdesign-Onlinemagazine erfasst werden. In diesem Zusammenhang ist mit wechselnden Anforderungen an die Farbpalette zu rechnen. Der Constraints-Last Ansatz ermöglicht die flexible Änderung der Constraints, ohne die Implementierung der CPE zu beeinflussen. Da Farbconstraints anwendungsabhängig sind, entkoppelt das Nachschalten

der Farbauswahl die Lösung vom Anwendungsbezug der Weboberflächen und erhöht die Wiederverwendbarkeit.

Dementsprechend wird folgender Problemlösungsansatz zur automatisierten Farbkonfiguration von Weboberflächen aus Bildvorlagen gewählt: 1. Ermittlung der Farbobermenge C_s durch einen Algorithmus zur CPE 2. Lösung eines Constraintsystems auf C_s , wodurch die Zuordnung der Oberflächenelemente zu den Farben der Farbpalette festgestellt wird. Die finale Farbpalette C als Teilmenge von C_s ergibt sich dadurch implizit als die Menge der Farben, die von den Oberflächenelementen verwendet werden.

2 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

2.1 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].

Kanjanawanishkula and Uyyanonvarab [10] 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 Ver-

¹<http://getbootstrap.com/css/#buttons>

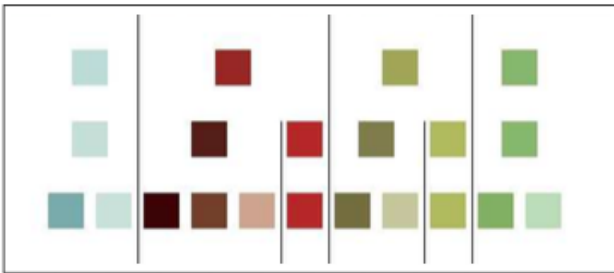
²<http://foundation.zurb.com/sites/docs/button.html>

³<http://semantic-ui.com/elements/button.html>

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



(a)



(b)

Abbildung 3: CPE Ergebnis von ACoPa. (a) Originalbild "Peppers"(b) Hierarchische Farbpalette. Die unterste Ebene zeigt die finalen Farben. (Quelle: [4])

teilung 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 [8] oder Octree[6]. 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 [4]. Diese Verfahren werden darum auch als *partitionierend* [4] oder auch *Post-Clustering* [10] 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 [12, 19] oder Fuzzy C-Means [2]. Andere analysieren das Histogramm auf dichte bzw. weniger dichte Regionen, wie z.B. Mean-Shift [3]. Eine detailliertere Vorstellung von Algorithmen zur CPE bietet [15].

Delon et al. [4] 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 reprä-

sentiert 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 3 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 Delon et al. [4] 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.

3 ACoPa

Im Folgenden wird die grundlegende Arbeitsweise des ACoPa Algorithmus nach Delon et al. [4] vorgestellt. Dabei werden die Herausforderungen, die bei der Implementierung aufgetreten sind, besprochen. Abschließend werden exemplarisch Anwendungsergebnisse präsentiert.

3.1 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:

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

Die Umrechnung gemäß eines Lehrbuchs für Farbbild-Verarbeitung [16] lautet hingegen:

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

Abbildung 4 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.

3.2 Histogramm-Segmentierung

Die Samples des Ausgangsbildes werden entlang der Hue-Werte sortiert. Das 1-dimensionale Hue-Histogramm $h = (h_i)_{i=1\dots b}$ mit b -Bins wird gebildet. Gesucht wird nun eine Sequenz $s = (s_i)_{i=1\dots k}$ 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“ [4]. Abbildung 5 zeigt das Prinzip an verschiedenen Beispielen.

Das Histogramm ist offensichtlich in jedem Segment unimodal, wenn s mit den Minima des Histogramms in-

itialisiert 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 [5] 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 [4, 5] 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 [5] 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 6 zeigt dies an einem Beispiel.

3.3 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 7 zeigt dies am Beispiel der Covers aus Abbildung 6. 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 1.2 definierten Color Swatches wieder. Delon et al. [4] empfehlen zusätzlich, die erhaltenen Farben als Startpunkte für den K-Means Algorithmus zu verwenden. Abbildung 7 (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.

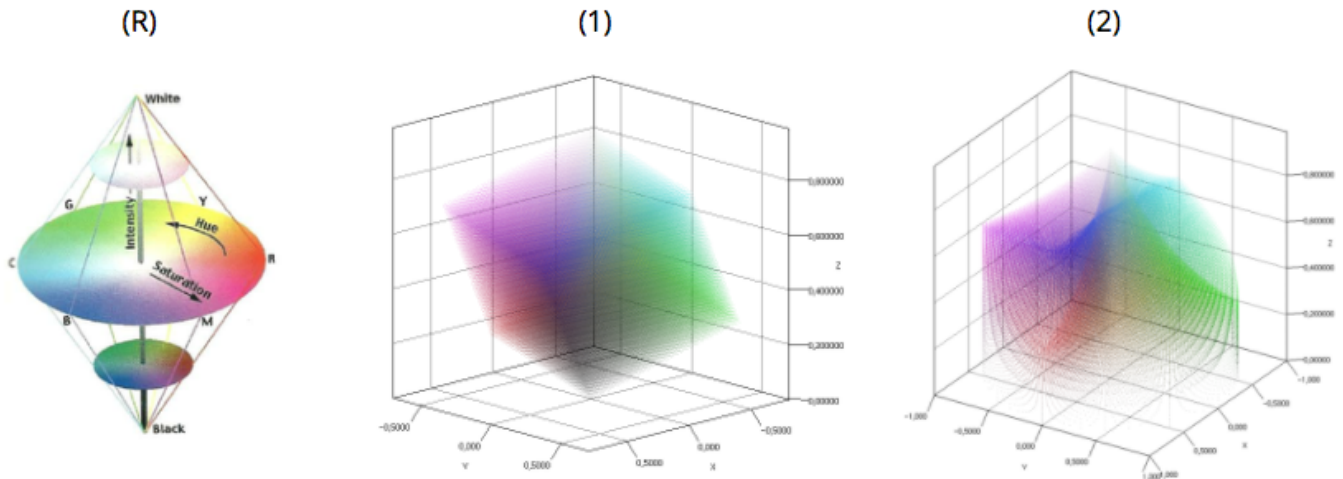


Abbildung 4: Gegenüberstellung von RGB zu HSI-Umrechnungsergebnisse. (a.) Referenz-HSI Raum. (b.) Umrechnung nach [4]. (c.) Umrechnung nach [16].

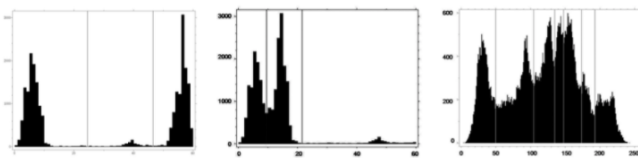


Abbildung 5: Beispiele der Segmentierung eines Histogramms in unimodale Abschnitte. (Quelle: [4])

4 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] Apple. ios human interface guidelines - color, 2016. URL <https://developer.apple.com/ios/human-interface-guidelines/visual-design/color/>.
- [2] J. Bezdek. Pattern recognition with fuzzy objective function algorithms. Kluwer Academic Publishers, 1981.
- [3] 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.
- [4] J. Delon, A. Desolneux, J. L. Lisani, and A. B. Petro. Automatic color palette. *Inverse Problems and Imaging*, 1(2): 265–287, 2007.

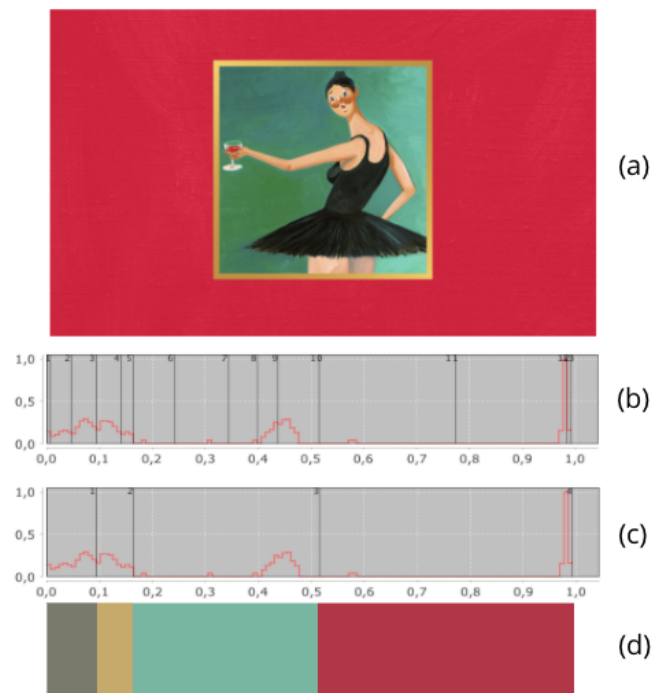


Abbildung 6: Beispiel für eine Segmentierung des Hue-Histogramms. (a) Ausgangsbild, ein Albumcover von Kanye West. (b) Hue-Histogramm (normalisiert), mit allen Minima als initiale Segmentierung. (c) Segmentierung nach Anwendung des FTC Algorithmus. (d) Farbmittelpunkte entsprechend der Samples der jeweiligen Segmente.



Abbildung 7: (a) Hierarchische Farbpalette des Covers aus Abbildung 6. (b) Farbpalette nach Anwendung von K-Means.

- [16] K. N. Plataniotis and A. N. Venetsanopoulos. *Color Image Processing and Applications*. Springer Monograph, 2000.
- [17] 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.
- [18] 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.
- [19] 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.
- [5] J. Delon, A. Desolneux, J. L. Lisani, and A. B. Petro. A nonparametric approach for histogram segmentation. 16(1): 253–261, 2007.
- [6] M. Gervautz and W. Purgathofer. A simple method for color quantization: Octree quantization. *Graphic Gems. Academic Press*, page 287–293, 1990.
- [7] Google. Material design style - colors, 2016. URL <https://material.io/guidelines/style/color.html>.
- [8] P. Heckbert. Color image quantization for frame buffer displays. *Comp. Graph*, 16:297–307, 1982.
- [9] 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/>.
- [10] 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.
- [11] S. Lin and P. Hanrahan. Modeling how people extract color themes from images. In *CHI Conference on Human Factors*, 2013.
- [12] 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.
- [13] 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.
- [14] Peter O'Donovan, Aseem Agarwala, and Aaron Hertzmann. Color compatibility from large datasets. In *ACM SIGGRAPH Conference*, page 63:1–63:12, 2011.
- [15] 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.