Die Mathematik der Sichtbarkeit

Wie ein unsichtbares Symbol in einer Notruf-App eine Gestaltungsregel messbar gemacht hat

Péter Kürtösi

Diplom-Maschinenbauingenieur

Entwickler der Notruf-App Segítség und der zugehörigen Wandkonsole

27. September 2026 · Werkstattbericht · Fassung 1.1

doi.org/10.5281/zenodo.22998702

Zusammenfassung

Diese Arbeit beginnt mit einem einzigen konkreten Fehler: Auf dem Startbildschirm einer Notruf-App war ein Symbol genau für die Menschen unsichtbar, die es am dringendsten gebraucht hätten. Gefunden hat den Fehler keine Maschine und auch nicht der Entwickler — gemeldet haben ihn sehbehinderte Nutzerinnen und Nutzer, die sagten, sie wollten die App verwenden, könnten sie aber nicht mit Sicherheit bedienen. Gemessen war der Fehler ein Kontrastverhältnis zwischen **1,00:1** und **1,17:1** dort, wo die Norm mindestens **3:1** verlangt — im Extremfall sind die beiden Farben rechnerisch nahezu gleich hell: das Verhältnis beträgt 1,00:1, das untere Ende der Skala.

Die Arbeit führt die Mathematik vor, gegen die sich in diesem Fall programmieren ließ: die Rückrechnung der sRGB-Codierung in lineares Licht, die Herkunft der Koeffizienten der relativen Leuchtdichte, die Kontrastformel der WCAG 2.1 samt dem darin enthaltenen Bildschirmschleier-Term und die Matrixform der Farbfehlsichtigkeits-Simulation. Jede Herleitung mündet an derselben Stelle: in einer Regel, die sich im Code erzwingen lässt.

Nach der Korrektur ergeben dieselben Messungen Werte zwischen **4,2:1** und **7,6:1**, und das Zeichen bleibt auch dann lesbar, wenn die betrachtende Person überhaupt keine Farbe sieht. Jede Zahl in dieser Arbeit ist gemessen oder hergeleitet; wo ein Modell oder eine Näherung verwendet wird, benennt der Text das ausdrücklich.

Schlagwörter: Barrierefreiheit · WCAG 2.1 · Kontrastverhältnis · relative Leuchtdichte · sRGB · Farbfehlsichtigkeit · Notruf-App · UX-Gestaltung · ältere Menschen · Sehbehinderung

Vorwort und Lesehinweis

> **Warum es diese Arbeit gibt**
>
> Dieser Text ist nicht am Schreibtisch entstanden. Er ist aus dem Fehler einer laufenden, tatsächlich benutzten Notruf-App hervorgegangen und wurde geschrieben, damit den nächsten Fehler dieser Art niemand mehr am eigenen Leib entdecken muss.

Die Erfahrung ist: Die Literatur zum Farbkontrast existiert in zwei Extremen. Das eine Extrem ist der Normtext — genau, aber er sagt nicht, **warum** darin steht, was darin steht. Das andere Extrem ist der Blogbeitrag aus der Gestaltungspraxis — lesbar, aber seine Formeln lassen sich auf nichts zurückführen, also kann man nicht darauf aufbauen.

Diese Arbeit zielt dazwischen. Jeder Abschnitt folgt derselben Struktur: **(1)** was wir beobachtet haben, **(2)** welche Mathematik dahintersteht, und **(3)** welche Regel sich daraus in Code schreiben lässt. Wo eine Aussage auf einer Messung beruht, steht auch die Messmethode dabei; wo auf einer Herleitung, dort die Herleitung.

> **Eine Anmerkung zum wissenschaftlichen Apparat**
>
> Die Formeln stammen nicht von uns: Jede einzelne kommt aus einer Norm oder einer begutachteten Veröffentlichung, und am Ende der Arbeit steht der zugehörige Nachweis. Was wir beigetragen haben, ist nicht die Mathematik, sondern der Fall: ein reales Produkt, mit den Rückmeldungen realer Nutzerinnen und Nutzer, und die Messungen, die wir an diesem Produkt durchgeführt haben.

Die Arbeit darf für Lehrzwecke verwendet werden. Die Abbildungen und die Zahlen darin werden sämtlich aus dem in Anhang B beschriebenen offenen Quelltext erzeugt — aus demselben Code, der auch die Zahlen im Fließtext rechnet. So kann eine Abbildung dem Text nicht widersprechen.

1. Der Fehler, der alles ausgelöst hat

Segítség („Hilfe”) ist eine kostenlose Notruf-App für ältere und genesende Menschen — mehr über die App unter **vigyazokrad.hu**. Ihr Startbildschirm besteht aus Kacheln, eine Kachel pro Person. Ein kleines Symbol am unteren Rand jeder Kachel sagt, was beim Antippen geschieht: Es wird angerufen, oder es geht eine Nachricht raus.

Auf Kacheln mit Foto war die Symbolfarbe einwandfrei. Über dem Foto liegt das Namensband als Schwarz mit 78 % Deckung, darauf war der blassgrüne Umschlag klar zu sehen — am Bildschirm gemessen 9,92:1. Diese Zahl lässt sich nicht aus Hexadezimalwerten ableiten, weil der Hintergrund dort das Foto selbst ist; deshalb haben wir sie gemessen und nicht gerechnet.

Nur hat nicht jede Kachel ein Foto. Wer kein Bild hinterlegt hat, bekommt eine gezeichnete Figur, darunter ein Namensband in einer Pastellfarbe — eine von sechs. Und dort verschwand dasselbe Symbol schlicht.

> **Der Satz, der alles ausgelöst hat**
>
> „Der Umschlag ist genau dasselbe Grün wie die Kachel dahinter. Das sieht kein Mensch.”

Gesagt hat das jemand, der nur auf das Telefon geschaut hat. Zuvor hatte aber bereits ein anderer Nutzer gemeldet — jemand, der selbst Schwierigkeiten mit Farben hat —, er finde überhaupt kein Symbol. Die

Antwort, die er bekam, lautete: „Da ist doch das kleine Symbol.” Beide hatten recht, und zwar nicht subjektiv. Die Messung zeigte eine Lage, die noch schlechter war als der Eindruck des Auges.

> **Was das bedeutet**
>
> Der gemeldete Fall — das blassgrüne Symbol auf dem grünen Namensband — misst **1,03:1**. Das ist kein „schwacher Kontrast”: Es bedeutet, dass die Leuchtdichte der beiden Farben praktisch gleich ist. Auf zwei der sechs Pastell-Hintergründe beträgt das Verhältnis auf zwei Nachkommastellen gerundet **1,00:1** — die gemessenen Werte sind **1,0010** und **1,0042**. Das ist das **untere Ende der Skala**: Ein Kontrastverhältnis kann nicht unter 1:1 fallen, denn dort sind die beiden Leuchtdichten bereits gleich. Wer es nicht gesehen hat, hat sich nichts eingebildet: Es war tatsächlich nichts da.

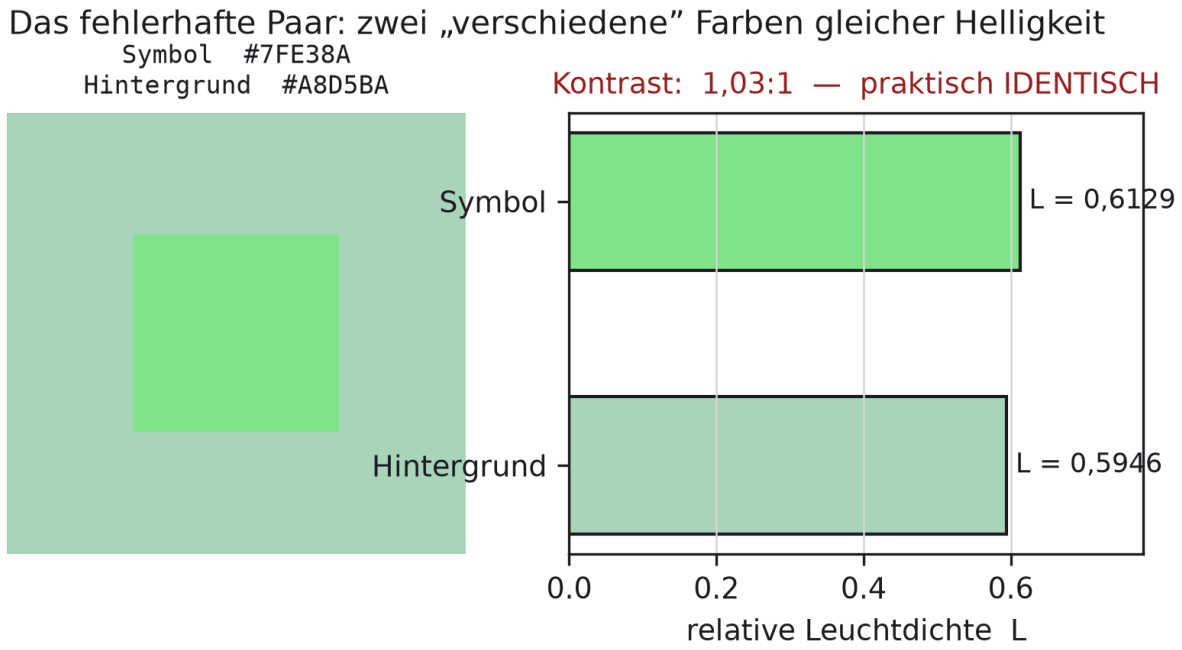


Abbildung 1. *Das fehlerhafte Paar. Links die beiden Farben so, wie sie am Bildschirm standen: das blassgrüne Symbol auf dem grünen Namensband. Rechts die relative Leuchtdichte der beiden Farben — praktisch identisch. Für das Auge sind die beiden Farben verschieden; in der Lichtmenge besteht zwischen ihnen kein Unterschied.*

2. Licht, Helligkeit, Farbton

Um den Fehler zu verstehen, müssen drei Begriffe getrennt werden, die die Alltagssprache vermischt.

— **Leuchtdichte** (luminance): wie viel Licht von der Fläche kommt. Eine physikalische Größe, messbar.

— **Helligkeit** (lightness): wie hell wir dieses Licht *wahrnehmen*. Nicht proportional zur Leuchtdichte — am dunklen Ende ist das Auge für Änderungen viel empfindlicher.

— **Farbton** (hue): ob wir etwas als rot, grün oder blau sehen. Das ergibt sich aus der Wellenlängenzusammensetzung des Lichts.

Die Kontrastformel der WCAG arbeitet ausschließlich mit der ersten Größe. Das ist kein Mangel, sondern eine Entscheidung: Der Leuchtdichte-Unterschied ist genau das Signal, das das visuelle System auch dann aufnimmt, wenn die Farbkanäle beeinträchtigt sind. Daraus folgt allerdings der wichtigste Satz dieser Arbeit, auf den wir in Abschnitt 8 zurückkommen: **Der Farbton kommt in der Formel nicht vor.**

Warum die drei Primärfarben nicht gleich gewichtet sind

Die Empfindlichkeit des Auges hängt stark von der Wellenlänge ab. Die Funktion $\bar{y}(\lambda)$ des CIE-Normalbeobachters von 1931 beschreibt genau das: Diese Kurve gibt an, wie viel „Helligkeit" aus Licht einer gegebenen Wellenlänge wird. Ihr Maximum liegt bei etwa 555 nm — im Grünen.

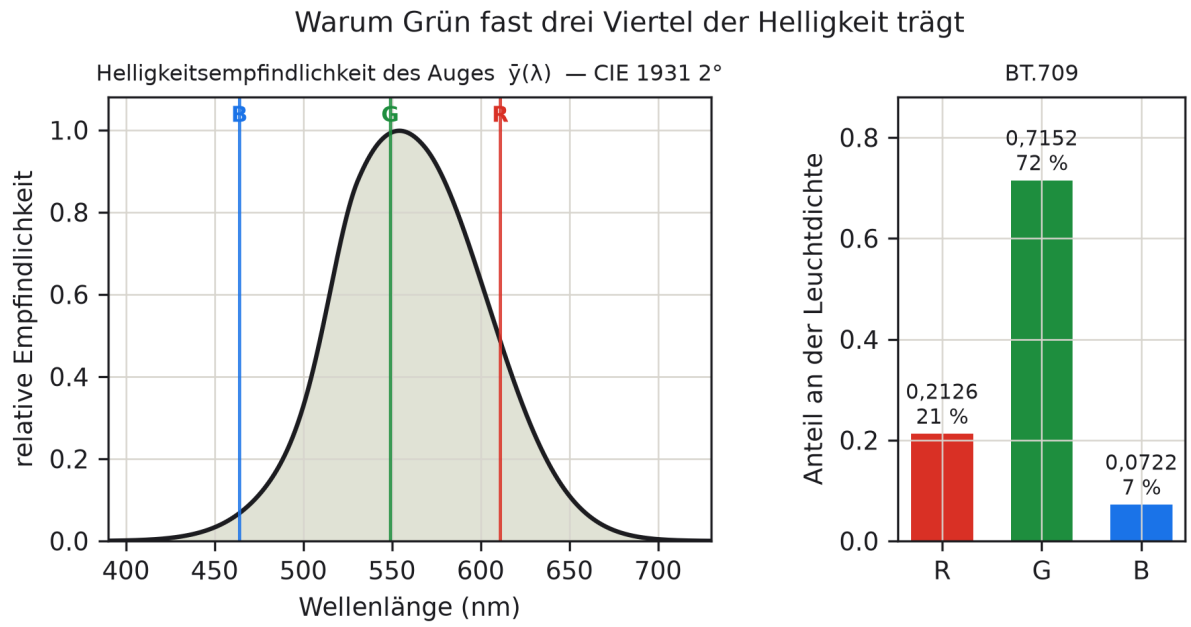


Abbildung 2. *Links die Helligkeitsempfindlichkeit des Auges in Abhängigkeit von der Wellenlänge (CIE 1931, 2°-Beobachter), mit der näherungsweisen dominanten Wellenlänge der drei sRGB-Primärfarben. Rechts die Leuchtdichte-Koeffizienten nach BT.709. Die grüne Primärfarbe sitzt fast auf dem Maximum der Kurve, die blaue dagegen an ihrem Fuß — deshalb steuert Grün* **72 %** *zur Helligkeit bei, Blau dagegen nur* **7 %**.

Das ist die Erklärung dafür, warum die drei Koeffizienten im nächsten Abschnitt so unterschiedlich ausfallen. Es ist weder Konvention noch Abstimmung nach Gefühl: Es ergibt sich aus dem Produkt der Empfindlichkeitskurve des Auges mit dem Spektrum der Primärfarben.

3. Die Rückrechnung der sRGB-Codierung

Auf einem Bildschirm ist #7FE38A drei Byte. Diese Bytes sind **keine** Lichtmengen: Die sRGB-Norm codiert nichtlinear. Die Form der Kurve folgt historisch der Antwortkurve von Kathodenstrahl-Bildschirmen, und es ist ein glücklicher Zufall, dass sie ungefähr das liefert, was auch das Auge braucht: Die 256 Stufen liegen dort dichter, wo das Auge für Unterschiede am empfindlichsten ist — am dunklen Ende. Bevor irgendetwas gerechnet werden kann, muss in lineares Licht zurückgerechnet werden.

$$s = c \,/\, 255$$

$$\text{lin}(s) = \begin{cases} s \,/\, 12{,}92 & \text{wenn } s \le 0{,}04045 \\ ((s + 0{,}055) \,/\, 1{,}055)^{2{,}4} & \text{sonst} \end{cases}$$

Warum es am dunklen Ende ein gerades Stück gibt

Die Formel ist deshalb keine einzelne Potenzfunktion, weil in der **Codier**-Richtung — wenn aus Licht ein Byte wird — die Steigung der Kurve nahe null über alle Grenzen wächst: Die Ableitung der Funktion *L^(1/2,4)* ist bei null unendlich. Das würde bei der Quantisierung und beim Rauschen Ärger machen, deshalb bekommt das dunkle Ende ein gerades Stück. Die Reihenfolge ist wichtig: Die Ableitung der **Rückrechen**-Richtung ist bei null im Gegenteil *null* — die beiden Richtungen lassen sich leicht verwechseln, deshalb lohnt es sich zu sagen, von welcher die Rede ist.

Warum sich nicht beide Konstanten aus einer Bedingung ergeben

Es läge nahe zu sagen, Knick und Steigung ergäben sich aus der Stetigkeit. So hält das jedoch nicht stand: Stetigkeit ist **eine** Gleichung, unbekannt sind aber **zwei** Größen — Knick und Steigung. Eine Gleichung legt zwei Zahlen nicht fest. Was sich sagen lässt, sind zwei verschiedene Anpassungen.

Stetige Anpassung. Wir verlangen nur, dass die *Werte* der beiden Zweige am Knick übereinstimmen:

$$s_0 / 12{,}92 \;=\; ((s_0 + 0{,}055) / 1{,}055)^{2{,}4}$$

Der Knick der Norm bei 0,04045 erfüllt das. Man muss es nicht glauben: Der Code in Anhang B rechnet beide Seiten aus, und der Unterschied beträgt **2,3·10⁻⁹** — an der Grenze der Gleitkommadarstellung also null.

Glatte Anpassung. Sie verlangt mehr: Auch die *Ableitungen* sollen übereinstimmen. Das sind bereits zwei Gleichungen, und es gibt eine geschlossene Lösung — die Steigung fällt heraus, und es bleibt *s0·2,4 = s0 + 0,055*, also

$$s_0 \;=\; 0{,}055 / (2{,}4 - 1) \;=\; 0{,}0392857$$

Die sRGB-Norm hat sich demnach für die **stetige** Anpassung entschieden, nicht für die glatte: Bei 0,04045 stimmen die Werte der beiden Zweige überein, ihre Steigungen weichen jedoch um **1,71 %** voneinander ab. Die Kurve ist stetig, aber nicht glatt — und das ist in Ordnung, denn der Knick ist für das Auge unsichtbar.

> **Eine Abweichung, die man kennen sollte**
>
> Im Text der WCAG 2.1 steht nicht 0,04045, sondern **0,03928** — und das sind genau die ersten fünf Nachkommastellen der *glatten* Lösung von oben. Die beiden Normen verwenden also zwei vertretbare Rundungen derselben Kurve. Wie viel das ausmacht, ist keine Frage der Meinung, sondern abzählbar: Von den 256 möglichen 8-Bit-Farbstufen fallen **0** zwischen die beiden Schwellen, und innerhalb dieses Bandes beträgt die größte Abweichung der Leuchtdichte der beiden Zweige **7,6·10⁻⁷**. Für keine einzige 8-Bit-Webfarbe liefern sie verschiedene Ergebnisse.

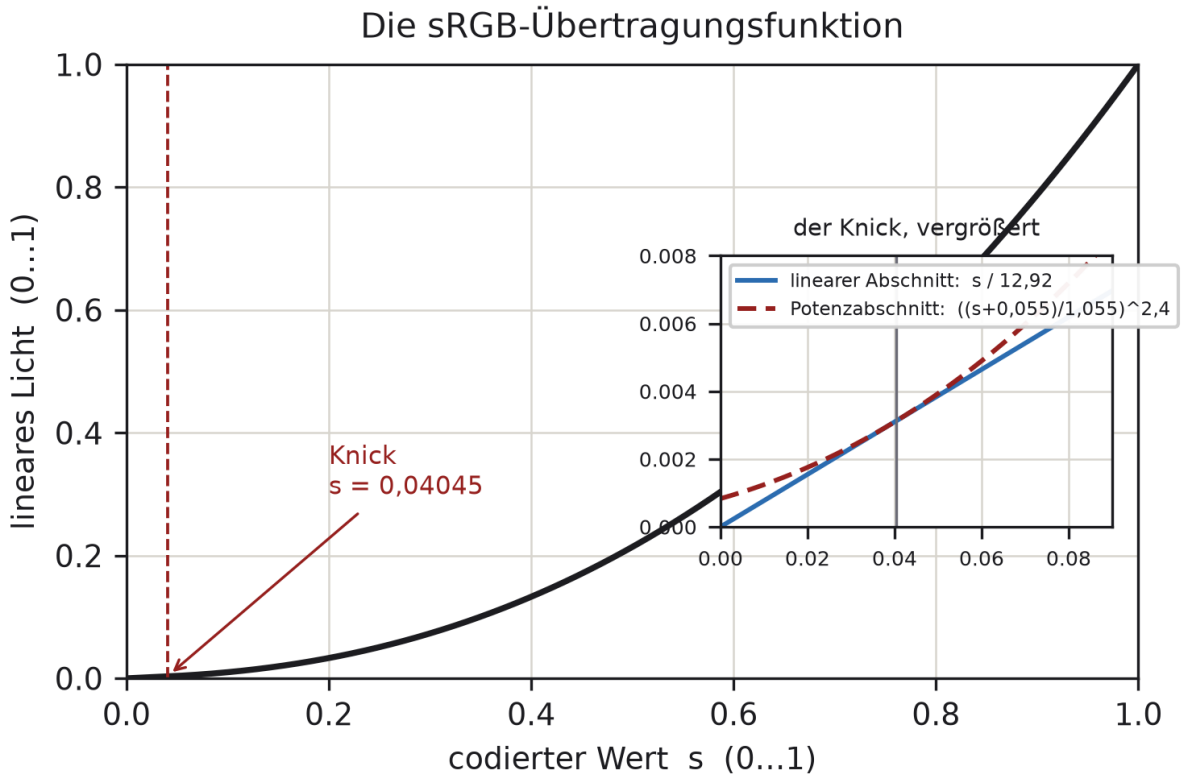


Abbildung 3. *Die sRGB-Übertragungsfunktion. Im vergrößerten Ausschnitt sind die beiden Zweige getrennt zu sehen: die blaue Gerade ist das lineare Stück, die rot gestrichelte Linie die Potenzfunktion. Am Knick stimmen die* **Werte** *der beiden Zweige überein — die Kurve ist stetig. Ihre Steigungen weichen dagegen um 1,71 % ab, sie ist also nicht glatt.*

> **Die praktische Folge**
> Wer direkt mit den Hexadezimalwerten rechnet — sie etwa einfach mittelt oder voneinander abzieht —, erhält kein Lichtverhältnis, sondern eine sinnlose Zahl. Die Codierung muss **immer** zurückgerechnet werden, bevor irgendetwas verglichen wird.

4. Die relative Leuchtdichte

Aus den drei linearen Kanälen wird eine einzige Zahl, die relative Leuchtdichte:

```
L  =  0,2126·R_lin  +  0,7152·G_lin  +  0,0722·B_lin
```

Die drei Koeffizienten stammen aus der Umrechnung der Primärfarben der Empfehlung ITU-R BT.709 und des Weißpunkts D65 in den CIE-XYZ-Raum — genauer aus der **Y-Zeile** dieser Umrechnungsmatrix. Im CIE-System ist die *Y*-Achse gerade die Leuchtdichte, abgestimmt auf die Empfindlichkeitsfunktion $\bar{y}(\lambda)$ des Auges.

Anders gesagt: Die drei Zahlen geben an, wie viel **Licht** jede Primärfarbe zum Weiß beiträgt. Sie drücken keine „Wichtigkeit" aus und sind keine ästhetische Gewichtung. Ihre Summe ist 1,0000, weil die relative Leuchtdichte von vollem Weiß definitionsgemäß 1 ist.

Kanal	Koeffizient	Anteil am Weiß
Rot (R)	0,2126	21 %
Grün (G)	0,7152	72 %
Blau (B)	0,0722	7 %

Tabelle 1. *Die Leuchtdichte-Koeffizienten nach BT.709. Grün allein trägt mehr Helligkeit als die beiden anderen zusammen: 3,4-mal so viel wie Rot und 9,9-mal so viel wie Blau.*

5. Das Kontrastverhältnis und der Bildschirmschleier

Der Kontrast zweier Farben ist nach WCAG 2.1 der Quotient ihrer Leuchtdichten, die hellere im Zähler:

```
CR  =  (L_heller + 0,05) / (L_dunkler + 0,05)
```

Was die 0,05 dort zu suchen hat

Der Term heißt **Bildschirmschleier** (flare). Die Oberfläche eines realen Displays reflektiert einen Teil des Raumlichts, deshalb hat das dargestellte „Schwarz" nie die Leuchtdichte null. Die 0,05 ist das von der Norm angenommene Maß dieses Streulichts, bezogen auf die Leuchtdichte des Weiß.

Daraus folgen unmittelbar die beiden Enden der Skala. Das kleinste Verhältnis ist offensichtlich 1:1, bei zwei Farben gleicher Leuchtdichte. Das größte liegt zwischen reinem Schwarz und reinem Weiß:

```
CR_max  =  (1,00 + 0,05) / (0,00 + 0,05)  =  21:1
```

Deshalb steht in jedem Text zur Barrierefreiheit 21:1 als obere Grenze. Es ist nicht zufällig eine runde Zahl: Sie folgt aus der Wahl des Schleier-Terms.

Die Schwellen

wofür	WCAG-Punkt	AA	AAA
normaler Text	1.4.3 / 1.4.6	4,5:1	7:1
großer Text (18 pt bzw. 14 pt fett ≈ 24 px bzw. 18,7 px)	1.4.3 / 1.4.6	3:1	4,5:1
Nicht-Text-Grafik, Bedienelement, Statusanzeige	1.4.11	3:1	—

Tabelle 2. *Die Kontrastschwellen der WCAG 2.1. Die Symbole auf den Kacheln gehören in die dritte Zeile: kein Text, aber die Oberfläche ist ohne sie nicht verständlich.*

Für die Symbole gilt also die Schwelle **3,0**. Die gemessenen **1,00–1,17** sind davon **33–39 %**.

6. Der Fall, gemessen

Die Farbe des Symbols war #7FE38A, ihre relative Leuchtdichte **0,6129**. Rechnet man das Kontrastverhältnis für jeden der sechs Pastell-Avatar-Hintergründe aus, ergibt sich folgendes Bild.

Avatar-Hintergrund	Hex	L	Kontrast zum Symbol	WCAG 1.4.11 (3:1)
Grün	#A8D5BA	0,5946	**1,03:1**	**NICHT ERFÜLLT**
Rosa	#F6C6D0	0,6454	**1,05:1**	**NICHT ERFÜLLT**
Gelb	#F2C879	0,6157	**1,00:1**	**NICHT ERFÜLLT**
Blau	#A9C7F0	0,5557	**1,09:1**	**NICHT ERFÜLLT**
Flieder	#C9B6E4	0,5147	**1,17:1**	**NICHT ERFÜLLT**
Türkis	#9BD8DE	0,6135	**1,00:1**	**NICHT ERFÜLLT**

Tabelle 3. *Kontrast des blassen Symbols (#7FE38A, L = 0,6129) auf den sechs Avatar-Hintergründen. Kein einziger Wert erreicht die Schwelle von 3:1.*

Der schlechteste Fall ist der Hintergrund **Türkis**: **1,0010:1** — **Gelb** liegt nur knapp darüber (**1,0042:1**), weshalb die Tabelle auf zwei Nachkommastellen für beide dasselbe zeigt. Der beste ist **Flieder** mit **1,17:1** — das heißt, **nicht ein einziger** der sechs erreicht die Schwelle von 3:1. Der gesamte Bereich bleibt unter **39 %** der Schwelle: Selbst der beste Fall liegt kaum über einem Drittel davon.

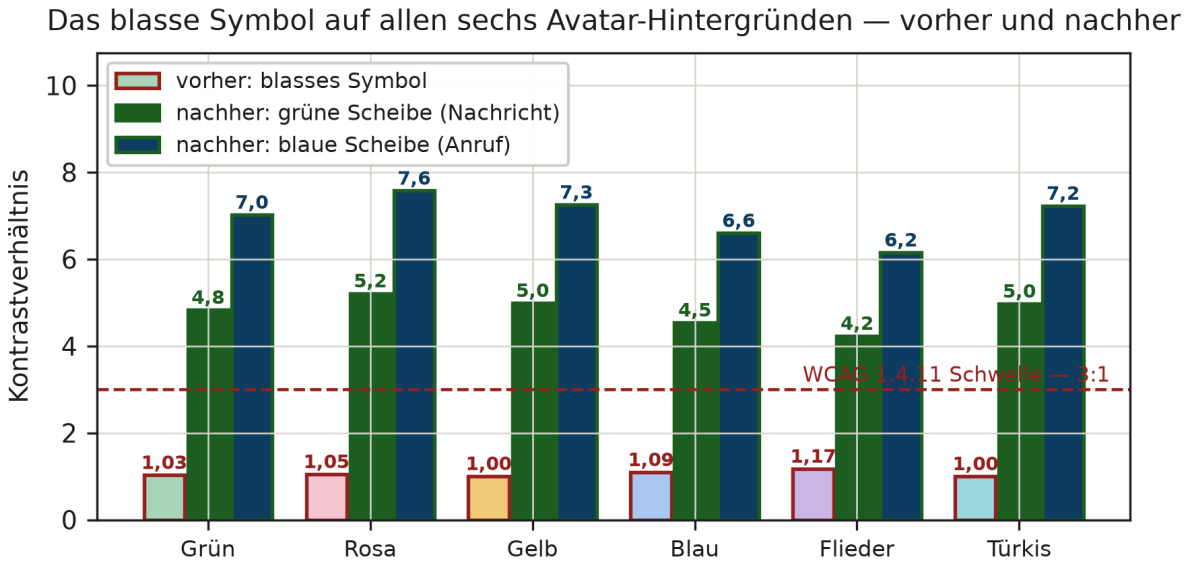


Abbildung 4. *Der Kontrast des blassen Symbols auf allen sechs Avatar-Hintergründen (roter Rahmen) und derselbe nach der Korrektur (grüner Rahmen). Die gestrichelte Linie ist die Schwelle von 3:1 nach WCAG 1.4.11. Vor der Korrektur erreicht sie keine einzige Säule; danach liegt jede darüber — hier mit der grünen Scheibe, der ungünstigeren der beiden.*

Die Messmethode

Die Hexadezimalwerte stammen aus dem Quelltext der Anwendung, die Kontraste rechnet die in den Abschnitten 3–5 beschriebene Formel. Die auf der Foto-Kachel gemessenen 9,92:1 sind dagegen eine *Bildschirmmessung*, denn dort ist der Hintergrund das Foto selbst — diese Zahl lässt sich nicht aus Hex-Werten herleiten, und wir haben es auch nicht getan.

7. Farbfehlsichtigkeit: der zweite Maßstab

Das Kontrastverhältnis allein sagt nichts darüber, was mit jemandem geschieht, der Farben anders sieht. Etwa **8 %** der Männer und etwa **0,4 %** der Frauen europäischer Abstammung haben eine Rot-Grün-Farbfehlsichtigkeit (Birch, 2012). Diese Zahl ist populationsabhängig: In chinesischen und japanischen Stichproben wurden bei Männern Anteile zwischen 4 % und 6,5 % gemessen. Die häufigste Form ist die Deuteranomalie.

Für die Simulation werden in der Praxis die von Machado, Oliveira und Fernandes (2009) veröffentlichten Matrizen verwendet — auf ihnen beruht auch die Sehsimulation mehrerer verbreiteter Entwickler- und Gestaltungswerkzeuge. Drei 3×3-Matrizen, nach Schweregrad skalierbar; hier werden die vollen Fälle (1,0) verwendet. Für dichromatische Fälle gibt es eine eigene Referenz, das Verfahren von Brettel, Viénot und Mollon, das im LMS-Raum projiziert.

> **Ein Detail, das oft falsch gemacht wird**
>
> Diese Matrizen sind für **lineares RGB** definiert. Wer sie direkt auf sRGB-Werte anwendet, kommt nahe heran — richtig ist es aber nicht. Die korrekte Kette lautet: **sRGB → linear → Matrix → sRGB**. Jede Simulation in dieser Arbeit ist so entstanden.

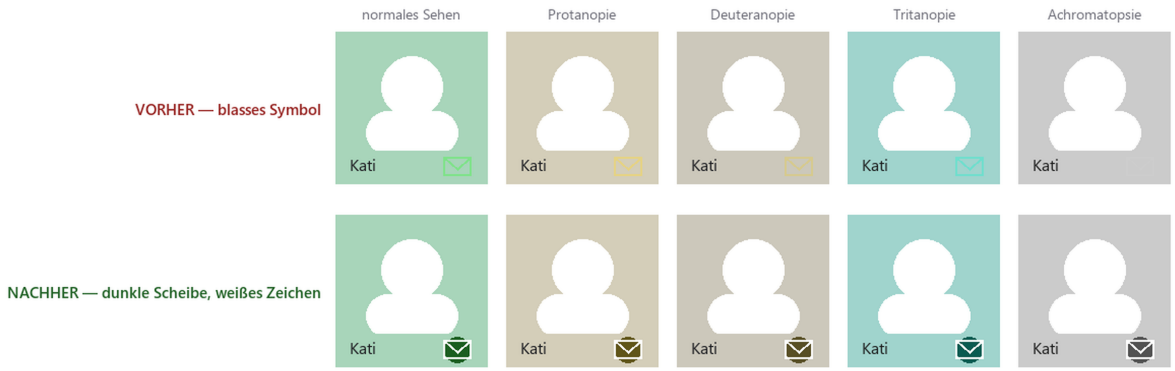


Abbildung 5. *Dieselben zwei Kacheln in fünf Sehweisen. In der oberen Reihe das blasse Symbol: mit normalem Sehen noch erkennbar, bei Achromatopsie* **vollständig verschwunden**. *In der unteren Reihe die Korrektur — dunkle Scheibe mit weißem Zeichen — in allen fünf Modi gleich gut lesbar.*

Bemerkenswert ist, dass sich die Kontrastzahlen zwischen den simulierten Modi kaum bewegen. Dafür gibt es einen messbaren Grund: Über die neun in dieser Arbeit vorkommenden Farben verschiebt die Simulation die relative Leuchtdichte um höchstens **0,041** — am meisten bei der Protanopie, am wenigsten bei der Tritanopie (0,014). Das ist allerdings kein allgemeines Gesetz: Die Matrizen sind *nicht* leuchtdichteerhaltend. Auf dieser Palette bleiben die Helligkeitsverhältnisse jedoch praktisch unverändert, und deshalb gilt hier: **Was über die Helligkeit ankommt, kommt auch bei farbfehlsichtigen Betrachtern an.**

8. Die Regel, die daraus folgt

> **Der wichtigste Satz dieser Arbeit**
>
> Der Farbton kommt in der Formel für die Leuchtdichte überhaupt nicht vor. Nur die Helligkeit. Deshalb können zwei Farben für das Auge deutlich unterscheidbar sein — im gemessenen Fall liegen ihre Farbtöne **59°** auseinander auf dem Farbkreis — und trotzdem ein Kontrastverhältnis von **1,00:1** haben. Mehr noch: Auf dem nahezu gegenüberliegenden Hintergrund **Rosa** (139°) sind es immer noch nur **1,05:1**.

Daraus folgt eine Regel, die sich tatsächlich in Code schreiben lässt:

> **Die Regel**
>
> Der Farbton darf niemals der einzige Unterschied zwischen zwei Dingen sein. Schaffe einen **Helligkeits**-Unterschied. Der Farbton darf zusätzliche Bedeutung tragen, aber er kann nicht der Träger der Bedeutung sein — denn für einen Teil der Betrachtenden ist er schlicht nicht vorhanden.

Das ist kein ästhetischer Ratschlag. Das ist die Form, in der die Mathematik der Abschnitte 2–7 benutzbar wird: eine Aussage, über die ein Test entscheiden kann.

9. Die Korrektur, gemessen

Die Korrektur ist genau der obigen Regel gefolgt. Das Symbol wurde **nicht** kräftiger grün — das wäre derselbe Fehler eine Stufe höher gewesen. Stattdessen bekam es eine eigene **dunkle Scheibe** mit einem weißen Zeichen darauf — und die Farbe dieser Scheibe trägt nun die Bedeutung weiter, die zuvor das Symbol selbst trug:

— **Nachricht**: grüne Scheibe, #1B5E20

— **Anruf**: blaue Scheibe, #0D3C61

— das Zeichen auf beiden: #FFFFFF — weiß

Die Messung nach der Korrektur:

Avatar-Hintergrund	Hex	grüne Scheibe	blaue Scheibe	WCAG 1.4.11 (3:1)
Grün	`#A8D5BA`	**4,83:1**	**7,02:1**	**ERFÜLLT**
Rosa	`#F6C6D0`	**5,21:1**	**7,57:1**	**ERFÜLLT**
Gelb	`#F2C879`	**4,99:1**	**7,25:1**	**ERFÜLLT**
Blau	`#A9C7F0`	**4,54:1**	**6,60:1**	**ERFÜLLT**
Flieder	`#C9B6E4`	**4,23:1**	**6,15:1**	**ERFÜLLT**
Türkis	`#9BD8DE`	**4,97:1**	**7,23:1**	**ERFÜLLT**

Tabelle 4. *Nach der Korrektur: Kontrast der grünen (`#1B5E20`, Nachricht) und der blauen (`#0D3C61`, Anruf) Scheibe auf denselben sechs Hintergründen.*

Der Kontrast zwischen Scheibe und Hintergrund liegt bei der **grünen** Scheibe zwischen **4,2:1** und **5,2:1**, bei der **blauen** zwischen **6,2:1** und **7,6:1** — also beim **1,4- bis 2,5-fachen** der Schwelle von 3:1. Das weiße Zeichen auf der Scheibe erreicht **7,87:1** (grün) bzw. **11,44:1** (blau) — beide über der Stufe AAA von 7:1.

> **Was das zeigt**
>
> Der Farbton ist nicht verschwunden: Die Scheibe ist grün bzw. blau und trägt die Bedeutung weiter. Geändert hat sich seine **Rolle**. Die Sichtbarkeit kommt nicht mehr vom Farbton, sondern vom **Helligkeitsunterschied** zwischen der dunklen Scheibe und dem Pastell-Hintergrund — und diesen Unterschied kann der Farbton nicht schlucken. Genau das ist die Regel aus Abschnitt 8, in Code geschrieben.

Und was noch wichtiger ist: Das Zeichen ist jetzt auch dann da, wenn die betrachtende Person überhaupt keine Farbe sieht. Das zeigt die untere Reihe von Abbildung 5.

10. Wie daraus ein Prüfschritt im Code wird

Eine Regel, die nichts überprüft, wird früher oder später verletzt. Die obige Mathematik lässt sich in wenigen Zeilen aufschreiben, und wenn sie aufgeschrieben ist, kann die kontinuierliche Integration sie bei jeder Änderung erneut abfragen.

```
function leuchtdichte(hex) {
  const v = [0, 2, 4].map(i => {
    const s = parseInt(hex.slice(i + 1, i + 3), 16) / 255;
    return s <= 0.04045 ? s / 12.92
                        : Math.pow((s + 0.055) / 1.055, 2.4);
  });
  return 0.2126 * v[0] + 0.7152 * v[1] + 0.0722 * v[2];
}

function kontrast(a, b) {
  const la = leuchtdichte(a), lb = leuchtdichte(b);
  return (Math.max(la, lb) + 0.05) / (Math.min(la, lb) + 0.05);
}
```

Der Prüfschritt ist danach eine einzige Zusicherung. Wichtig ist, dass die Schwelle *nicht* über den Code verstreut steht, sondern an einer Stelle, mit Namen:

```
import { strict as assert } from 'node:assert';

const SCHWELLE_UI = 3.0;   // WCAG 2.1 — 1.4.11

for (const hintergrund of AVATAR_FARBEN) {
  const cr = kontrast(SYMBOL_SCHEIBE, hintergrund);
  assert(cr >= SCHWELLE_UI,
         `${hintergrund}: ${cr.toFixed(2)}:1 < ${SCHWELLE_UI}:1`);
}
```

> **Was der Prüfschritt nicht löst**
>
> Der obige Test misst das Symbol auf dem *Namensband*, denn dort kennen wir beide Farben. Auf der Foto-Kachel kennen wir den Hintergrund nicht — dort *garantiert* die dunkle Scheibe die Lesbarkeit des Zeichens, nicht eine Zahl. Diesen Unterschied sollte man aussprechen, denn ein grüner Test erweckt leicht den Eindruck, er decke alles ab.

11. Wo die Formel der WCAG 2.1 irrt

Die Redlichkeit verlangt, es auszusprechen: Von der obigen Formel ist bekannt, dass sie ungenau ist. Am stärksten weicht sie auf dunklen Hintergründen und in den Mitteltönen von dem ab, was ein Mensch tatsächlich sieht. Der Grund ist der Term +0,05: Am dunklen Ende beherrscht **er** den Quotienten, nicht die tatsächliche Leuchtdichte.

Das ist kein grundsätzlicher Einwand, sondern nachrechenbar. Gegen eine weiße Fläche wächst das Verhältnis ohne Schleier über alle Grenzen, je näher man Schwarz kommt; die Formel der WCAG bleibt dagegen bei 21:1 stehen. Die beiden Kurven nähern sich nur langsam an: Bei Y = 0,05 ist das schleierfreie Verhältnis noch das **1,90-fache** des WCAG-Werts, und die Abweichung sinkt erst oberhalb von Y ≈ **0,19** unter 20 %. Am dunklen Ende glättet die Formel also im Wesentlichen genau den Bereich zu einer einzigen Zahl, in dem das Auge für Unterschiede am empfindlichsten ist.

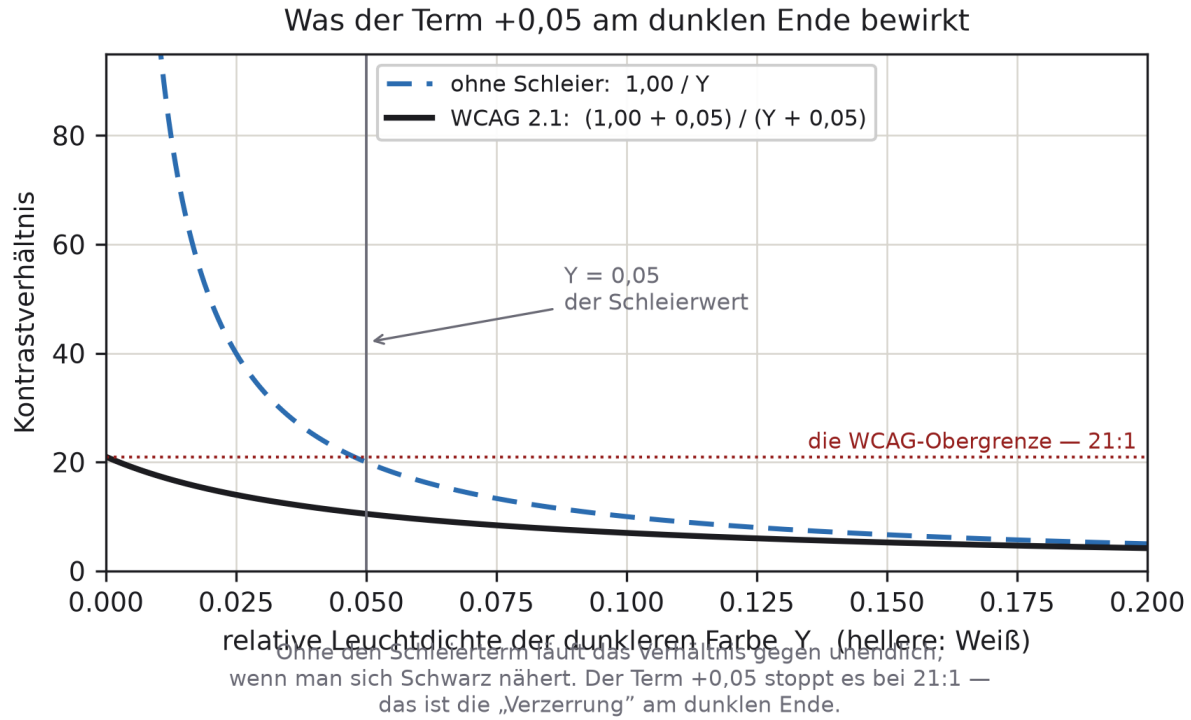


Abbildung 6. *Was der Schleier-Term bewirkt. Die hellere Farbe ist Weiß; auf der waagerechten Achse steht die Leuchtdichte der dunkleren Farbe. Ohne Schleier (blau gestrichelt) läuft das Verhältnis gegen unendlich, je näher man Schwarz kommt. Der Term +0,05* **stoppt es bei 21:1**. *Die beiden Kurven nähern sich langsam an: Bei Y = 0,05 beträgt die Abweichung noch das 1,90-fache, und erst oberhalb von Y ≈ 0,19 sinkt sie unter 20 % — die Formel glättet die Unterschiede also genau dort, wo das Auge am empfindlichsten ist.*

Die Nachfolgerin der Formel ist **APCA** (Accessible Perceptual Contrast Algorithm), die im Arbeitsentwurf der WCAG 3 auftaucht. APCA modelliert auch die Polarität — dunkel auf hell und hell auf dunkel sind nicht derselbe Fall — und liefert kein Verhältnis zurück, sondern einen *Lc*-Wert, etwa zwischen −108 und +106.

> **Die praktische Entscheidung**
>
> Normativ ist heute — zitierbar auf einer Produktseite, in einer Ausschreibung oder auf einer Konferenz — die WCAG 2.1. Also programmieren wir dagegen. APCA kennen wir, und wo es mehr sagt, hören wir hin; die Konformität sprechen wir jedoch in 2.1 aus.

Eine Anmerkung zu Abbildung 6: Sie zeigt bewusst *keine* APCA-Werte, sondern die genormte CIE-L*-Helligkeit. Die APCA-Formel haben wir nicht durch eigene Messung überprüft, deshalb rechnen wir für keinen einzigen Fall dieser Arbeit einen APCA-Wert aus — der oben genannte Lc-Bereich ist der veröffentlichte Wertebereich des Algorithmus, nicht unsere Messung. CIE L* ist dagegen genormt und belegt denselben Punkt.

12. Grenzen und weitere Arbeit

Auf die folgenden Fragen kennen wir die Antwort nicht. Es sind keine rhetorischen Fragen: Sie sind an einem laufenden Produkt aufgetaucht, und ihre Beantwortung würde ändern, wie wir gestalten.

1. **Reichen 3:1 auf einer Notruf-Oberfläche?** Die WCAG nennt ein Minimum, kein Optimum. Bei einer Notruftaste, die ein erschrockener, schlecht sehender älterer Mensch sucht — wo liegt die wirkliche Schwelle? Die Norm ist nicht für den Notfall gemacht worden.
2. **Was tun wir mit der Bedeutung der Farbe, wenn die Sichtbarkeit bereits die Helligkeit trägt?** Grün = Nachricht, Blau = Anruf bleibt bestehen. Hilft das noch jemandem, der die beiden nicht unterscheiden kann — oder verwirrt es eher?
3. **Nützt die Schweregrad-Skala in der Praxis?** Die Machado-Matrizen lassen sich in Schritten von 0,1 skalieren. Ist eine Deuteranomalie von 0,6 realistischer als eine vollständige Deuteranopie — oder sollte die Gestaltung immer auf den schlechtesten Fall zielen?
4. **Ist eine Pastellpalette überhaupt zu verteidigen?** Die sechs Avatar-Farben sind deshalb pastellig, weil das freundlich wirkt und dunklen Text trägt. Aber genau das hat das blasse Symbol unsichtbar gemacht. Gibt es eine Palette, die freundlich ist *und* Helligkeitsreserve für das lässt, was darauf kommt?

Von den Grenzen der Methode lohnt es sich, zwei eigens zu nennen. Erstens sind die Simulationsmatrizen Modelle, keine Personen: Das Sehen eines konkreten Menschen ersetzen sie nicht. Zweitens beruht die Arbeit auf Messungen an einem einzigen Produkt — die Ergebnisse weisen eine Richtung, sind aber keine statistische Stichprobe.

13. Zusammenfassung

Diesen Fehler hat keine Maschine gefunden. Kein einziger unserer Tests ist daran gescheitert — im Gegenteil: In der Umgebung, für die es entworfen war (über einem dunklen Foto), war das Symbol einwandfrei.

Ein Mensch hat auf das Telefon geschaut und es ausgesprochen. Ein anderer, der selbst Schwierigkeiten mit Farben hat, hatte schon früher gemeldet, er finde überhaupt kein Symbol — und bekam zur Antwort: „Da ist doch das kleine Symbol."

Er hatte recht, und zwar nicht subjektiv, sondern **mathematisch**. Dafür ist die Formel da — damit den nächsten Fehler niemand am eigenen Leib entdecken muss.

> **Die drei Sätze, die bleiben**
>
> **1.** Der Farbton kommt in der Kontrastformel nicht vor — nur die Helligkeit.
> **2.** Deshalb darf der Farbton nie der einzige Unterschied zwischen zwei Dingen sein.
> **3.** Was nichts überprüft, wird früher oder später verletzt — also mache aus der Regel einen Test.

Über den Autor

Péter Kürtösi ist Diplom-Maschinenbauingenieur und entwirft und baut seit fünfzehn Jahren Elektronik, schreibt Firmware und betreibt Systeme. Zu seinen Arbeiten gehören Klimasteuerungen, Thermostate, Prüfstände und Server; das meiste davon hat er selbst entworfen, gebaut und programmiert.

Die Anwendung *Segítség*, die den Gegenstand dieser Arbeit bildet, und die zugehörige Wandkonsole entstehen für ältere und genesende Menschen. Das System steht in zwei Pflegeheimen vor der Installation. Die Entwicklung begann nicht als geschäftlicher Auftrag: Sie ist aus der Erfahrung gewachsen, dass die Notfallmeldung dort am häufigsten fehlt, wo sie am nötigsten wäre. Über das System und die entstehenden Geräte: **vigyazokrad.hu**.

Kontakt und Zitation

Korrespondenz: Péter Kürtösi · peter.kurtosi@gmail.com
Zitiervorschlag: Kürtösi, P. (2026): *Die Mathematik der Sichtbarkeit*. Werkstattbericht, Fassung 1.1. doi.org/10.5281/zenodo.22998702
Diese Arbeit darf für Lehrzwecke unter Angabe der Quelle frei kopiert und zitiert werden.

Literatur

Keine der Formeln und keine der externen Angaben in dieser Arbeit stammen von uns: alle kommen aus den unten genannten Quellen.

[1] **W3C (2018)** — Web Content Accessibility Guidelines (WCAG) 2.1. W3C Recommendation, 5 June 2018. Insbesondere: 1.4.3 Contrast (Minimum), 1.4.6 Contrast (Enhanced), 1.4.11 Non-text Contrast. https://www.w3.org/TR/WCAG21/

[2] **IEC (1999)** — IEC 61966-2-1:1999 — Multimedia systems and equipment — Colour measurement and management — Part 2-1: Colour management — Default RGB colour space — sRGB.

[3] **ITU-R (2015)** — Recommendation ITU-R BT.709-6 — Parameter values for the HDTV standards for production and international programme exchange. International Telecommunication Union, Genf.

[4] **CIE (2004)** — CIE 15:2004 — Colorimetry, 3rd edition. Commission Internationale de l'Éclairage, Wien. (Definition des CIE-2°-Normalbeobachters von 1931 und von CIE L*.)

[5] **Birch (2012)** — Birch, J.: Worldwide prevalence of red-green color deficiency. Journal of the Optical Society of America A, 29(3), 313–320. doi:10.1364/JOSAA.29.000313

[6] **Machado, Oliveira & Fernandes (2009)** — Machado, G. M., Oliveira, M. M. & Fernandes, L. A. F.: A Physiologically-based Model for Simulation of Color Vision Deficiency. IEEE Transactions on Visualization and Computer Graphics, 15(6), 1291–1298.

[7] **Brettel, Viénot & Mollon (1997)** — Brettel, H., Viénot, F. & Mollon, J. D.: Computerized simulation of color appearance for dichromats. Journal of the Optical Society of America A, 14(10), 2647–2655.

[8] **Viénot, Brettel & Mollon (1999)** — Viénot, F., Brettel, H. & Mollon, J. D.: Digital video colourmaps for checking the legibility of displays by dichromats. Color Research & Application, 24(4), 243–252.

[9] **Wyman, Sloan & Shirley (2013)** — Wyman, C., Sloan, P.-P. & Shirley, P.: Simple Analytic Approximations to the CIE XYZ Color Matching Functions. Journal of Computer Graphics Techniques, 2(2), 1–11. (Die in den Abbildungen dieser Arbeit verwendete analytische Näherung.)

[10] **W3C (Arbeitsentwurf)** — WCAG 3 Working Draft und die Beschreibung von APCA (Accessible Perceptual Contrast Algorithm). https://www.w3.org/TR/wcag-3.0/ · https://git.apcacontrast.com/

[11] **Sharpe et al. (2005)** — Sharpe, L. T., Stockman, A., Jagla, W. & Jägle, H.: A luminous efficiency function, V*(λ), for daylight adaptation. Journal of Vision, 5(11), 948–968.

Anhang A — Referenzimplementierung

Der folgende Python-Code enthält die gesamte Mathematik dieser Arbeit: dieselben Routinen, die die Abbildungen und die Zahlen im Fließtext erzeugt haben, mit übersetzten Bezeichnern — das veröffentlichte Modul (szinmatek.py, siehe Anhang B) benennt sie auf Ungarisch. Es hat keine Abhängigkeiten.

```
import math

def srgb_linear(s):
    """Codierter sRGB-Kanal (0..1) -> lineares Licht."""
    if s <= 0.04045:
        return s / 12.92
    return ((s + 0.055) / 1.055) ** 2.4

def leuchtdichte(hexa):
    h = hexa.lstrip("#")
    r, g, b = (int(h[i:i+2], 16) / 255.0 for i in (0, 2, 4))
    return (0.2126 * srgb_linear(r)
            + 0.7152 * srgb_linear(g)
            + 0.0722 * srgb_linear(b))

def kontrast(a, b):
    la, lb = leuchtdichte(a), leuchtdichte(b)
    return (max(la, lb) + 0.05) / (min(la, lb) + 0.05)
```

Für die Simulation der Farbfehlsichtigkeit werden die von Machado und Mitarbeitern (2009) veröffentlichten Matrizen mit Schweregrad 1,0 benötigt, angewendet in linearem RGB und anschließend zurück nach sRGB codiert. Der Fall der Achromatopsie ist einfacher: Dort bleibt auf allen drei Kanälen nur die Leuchtdichte übrig.

Anhang B — Reproduzierbarkeit

Keine einzige Abbildung dieser Arbeit ist eine Zeichnung. Alle rechnen aus demselben Modul, das auch die Zahlen im Text liefert, und das Dokument lässt sich mit zwei Befehlen neu erzeugen:

```
python abrak.py de      # die sechs Abbildungen, 300 dpi
python pdf.py de        # die Arbeit
```

— szinmatek.py — die Mathematik: sRGB, Leuchtdichte, Kontrast, CIE L*, Farbabgleichfunktionen, Simulationsmatrizen

— abrak.py — die sechs Abbildungen, alle aus dem obigen Modul

— tartalom_de.py — der Text; an den Stellen der Zahlen stehen Marken, die der Erzeuger einsetzt

— pdf.py — der Satz

> **Warum so**
>
> Käme eine Zahl im Text aus einer anderen Quelle als die Zahl in der Abbildung, würden die beiden früher oder später auseinanderlaufen. In einem Lehrmaterial ist das das Schlimmste, was passieren kann: Die studierende Person kann nicht entscheiden, welcher sie glauben soll. So dagegen kann eine Abbildung dem Text nicht widersprechen.