The Mathematics of Visibility

How one invisible icon in an emergency-call application turned a design rule into something measurable

Péter Kürtösi

mechanical engineer (MSc)

developer of the Segítség emergency-call application and wall console

27 September 2026 · a paper from practice · version 1.1

doi.org/10.5281/zenodo.22998702

Abstract

This paper begins with a single concrete defect: on the home screen of an emergency-call application, an icon was invisible to exactly the people who most needed it. The defect was not found by a machine, nor by the developer — it was reported by visually impaired users who said they wanted to use the application but could not use it with confidence. Measured, the defect was a contrast ratio between **1.00:1** and **1.17:1** where the standard requires at least **3:1**; at the extreme the two colours are mathematically all but equal in lightness: the ratio is 1.00:1, the bottom of the scale.

The paper works through the mathematics that could be programmed against in this case: decoding sRGB back to linear light, the origin of the relative-luminance coefficients, the WCAG 2.1 contrast-ratio formula including the screen-flare term inside it, and the matrix form of colour-vision-deficiency simulation. Every derivation ends in the same place: a rule that can be enforced in code.

After the fix, the same measurements give values between **4.2:1** and **7.6:1**, and the mark remains legible even to a viewer who sees no colour at all. Every number in this paper is either measured or derived; wherever a model or an approximation is used, the text says so explicitly.

Keywords: accessibility · WCAG 2.1 · contrast ratio · relative luminance · sRGB · colour vision deficiency · emergency-call application · UX design · older adults · visual impairment

Foreword and how to read this paper

> **Why this paper exists**
> This was not written at a desk. It grew out of a defect in a working emergency-call application that is in people's hands, and it was written so that nobody has to discover the next such defect the hard way.

In our experience the literature on colour contrast lives at two extremes. At one end is the text of the standard: precise, but it does not tell you **why** it says what it says. At the other end is the designer's blog post: readable, but its formulas cannot be traced back to anything, so you cannot build on them.

This paper aims between the two. Every section has the same shape: **(1)** what we observed, **(2)** the mathematics behind it, and **(3)** the rule that can be written into code as a result. Where a claim rests on measurement, the method of measurement is given; where it rests on derivation, the derivation is given.

> **A note on the scientific apparatus**
> We did not invent the formulas: every one of them comes from a standard or a peer-reviewed publication, and each is referenced at the end of the paper. What we added is not the mathematics but the case: a real product, with real feedback from real users, and the measurements we took on that product.

This paper may be used for teaching. The figures, and the numbers inside them, are all generated from the open source described in Appendix B — from the same code that computes the numbers in the prose. A figure therefore cannot contradict the text.

1. The defect that started it

Segítség ("Help") is a free emergency-call application for elderly and convalescing people — more about it at **vigyazokrad.hu**. Its home screen is a grid of tiles, one person per tile. A small icon at the bottom of each tile tells you what will happen if you tap it: a call is placed, or a message is sent.

On tiles with a photograph the icon's colour was faultless. Over a photograph the name band is black at 78 % opacity, and the pale green envelope stood out clearly — measured on the screen at **9.92:1**. That number cannot be derived from hexadecimal values, because there the backdrop is the photograph itself; so we measured it rather than computing it.

But not every tile has a photograph. Anyone without a photo gets a drawn figure, with a pastel name band beneath it — one of six pastel colours. And there, the very same icon simply disappeared.

> **The sentence that started it**
> "The envelope is exactly the same green as the tile behind it. Nobody can see it."

That was said by someone who merely glanced at the phone. Before that, however, another user — one who has difficulty with colour himself — had already reported that he could not find *any* icon at all, and was told "the little icon is right there".

Both of them were right, and not subjectively. The measurement showed a situation worse than the eye suggested.

What this means

The reported case — the pale green icon on the green name band — measures **1.03:1**. This is not "low contrast": it means the two colours have practically identical luminance. On two of the six pastel backgrounds the ratio rounds to **1.00:1** at two decimals — the measured values are **1.0010** and **1.0042**. That is the **bottom of the scale**: a contrast ratio cannot fall below 1:1, because at that point the two luminances are already equal. The person who could not see it was not imagining things: there genuinely was nothing there.

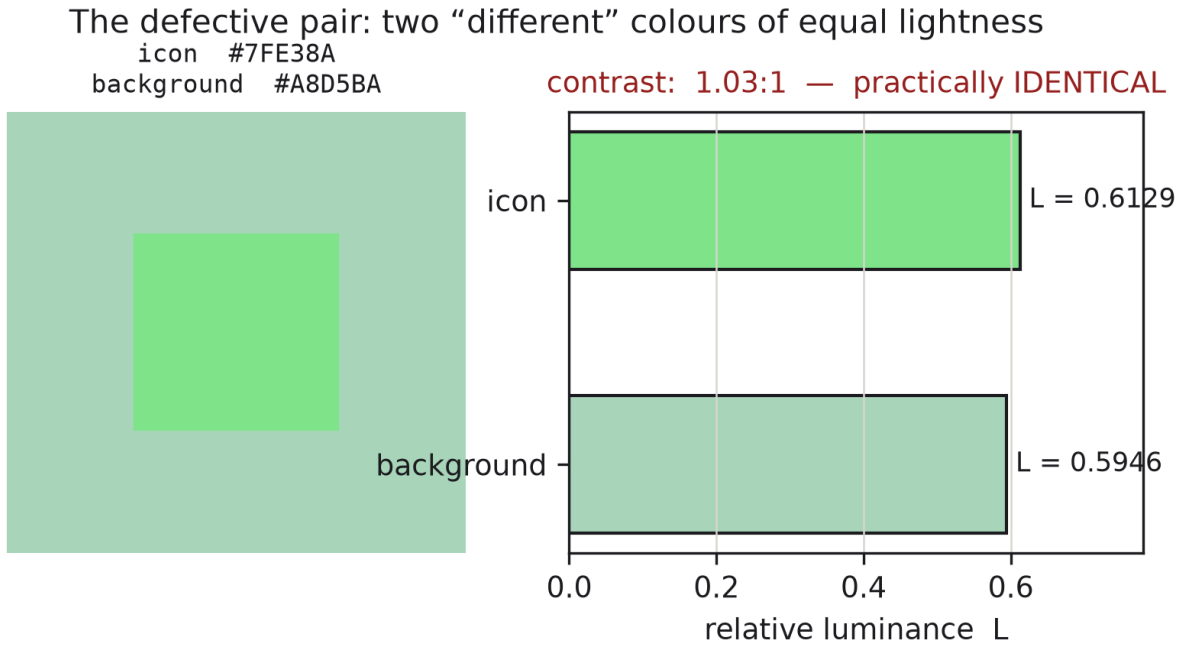


Figure 1. *The defective pair. On the left, the two colours as they stood on the screen: the pale green icon on the green name band. On the right, the relative luminance of each — practically identical. To the eye the two colours are different; in the quantity of light there is no difference between them.*

2. Light, lightness, hue

To understand the defect, three ideas that everyday language runs together have to be separated.

— **Luminance:** how much light comes off the surface. A physical quantity; measurable.

— **Lightness:** how much light we *perceive* it to be. Not proportional to luminance — the eye is far more sensitive to change at the dark end.

— **Hue:** whether we see it as red, green or blue. This comes from the wavelength composition of the light.

The WCAG contrast formula works with **the first of these only**. That is not an omission but a decision: a difference in luminance is the signal the visual system still receives even when the colour channels are impaired. From it follows the most important sentence in this paper, to which we return in section 8: **hue does not appear in the formula at all.**

Why the three primaries do not carry equal weight

The eye's sensitivity is strongly wavelength-dependent. The $\bar{y}(\lambda)$ function of the CIE 1931 standard observer describes exactly this: the curve tells you how much "lightness" results from light of a given wavelength. Its peak is around 555 nm — in the green.

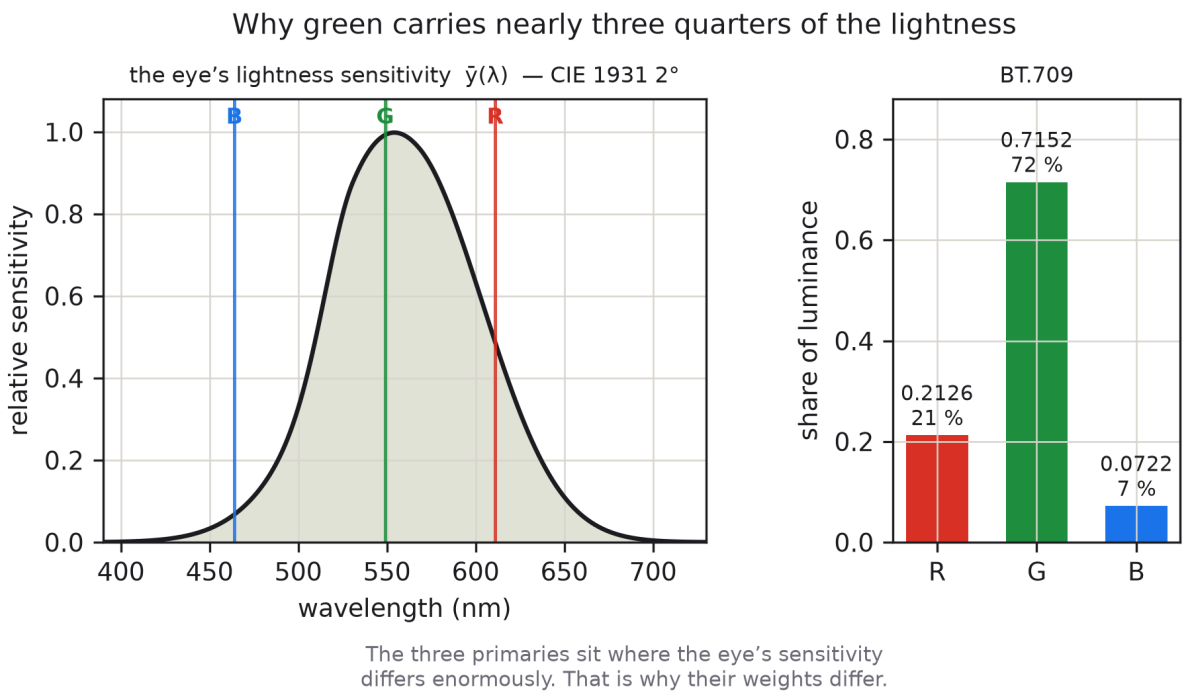


Figure 2. *On the left, the eye's lightness sensitivity as a function of wavelength (CIE 1931, 2° observer), with the approximate dominant wavelength of each of the three sRGB primaries. On the right, the BT.709 luminance coefficients. The green primary sits almost at the peak of the curve while the blue sits near its foot — which is why green contributes* **72 %** *of the lightness and blue only* **7 %**.

This is the explanation for why the three coefficients in the next section differ so much. It is neither convention nor tuning: it falls out of the product of the eye's sensitivity curve with the spectra of the primaries.

3. Decoding sRGB

On a screen, #7FE38A is three bytes. Those bytes are **not** quantities of light: the sRGB standard encodes non-linearly. The shape of the curve historically follows the response of cathode-ray displays, and it is a fortunate coincidence that this is roughly what the eye needs as well: the 256 steps are denser where the eye is most sensitive to difference — at the dark end. Before anything can be computed, they must be decoded back to linear light.

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

$$\mathrm{lin}(s) = \begin{cases} s \,/\, 12.92 & \text{if } s \le 0.04045 \\ ((s + 0.055) \,/\, 1.055)^{2.4} & \text{otherwise} \end{cases}$$

Why there is a straight segment at the dark end

The formula is not a single power function because in the **encoding** direction — when light becomes a byte — the slope of the curve grows without bound near zero: the derivative of $L^{1/2.4}$ at zero is infinite. That would cause trouble in quantisation and in noise, so the dark end is given a straight segment. The order matters: the derivative of the **decoding** direction at zero is, on the contrary, *zero*. The two directions are easy to confuse, so it is worth saying which one is meant.

Why both constants cannot follow from a single condition

It would be convenient to say that the knee and the slope follow from continuity. That does not hold up: continuity is **one** equation, while there are **two** unknowns — the knee and the slope. One equation does not determine two numbers. What can be stated is two different fits.

Continuous fit. We ask only that the *values* of the two branches agree at the knee:

$$s_0 / 12.92 \;=\; ((s_0 + 0.055) / 1.055)^{2.4}$$

The standard's knee of 0.04045 satisfies this. You do not have to take it on trust: the code in Appendix B computes both sides, and the difference is $\mathbf{2.3 \cdot 10^{-9}}$ — zero to the limit of floating-point representation.

Smooth fit. This asks for more: that the *derivatives* agree as well. That is two equations, and it has a closed-form solution — the slope cancels, leaving $s_0 \cdot 2.4 = s_0 + 0.055$, that is

$$s_0 \;=\; 0.055 / (2.4 - 1) \;=\; 0.0392857$$

The sRGB standard therefore chose the **continuous** fit, not the smooth one: at 0.04045 the values of the two branches agree, but their slopes differ by **1.71 %**. The curve is continuous but not smooth — and that is fine, because the kink is invisible to the eye.

> **A discrepancy worth knowing about**
>
> The text of WCAG 2.1 does not say 0.04045 but **0.03928** — and that is precisely the first five decimals of the *smooth* solution above. The two standards use two defensible roundings of the same curve. How much this matters is not a question of opinion but of counting: of the 256 possible 8-bit colour levels, **0** fall between the two thresholds, and within that band the largest difference between the two branches' luminance is $\mathbf{7.6 \cdot 10^{-7}}$. For no 8-bit web colour do they give different results.

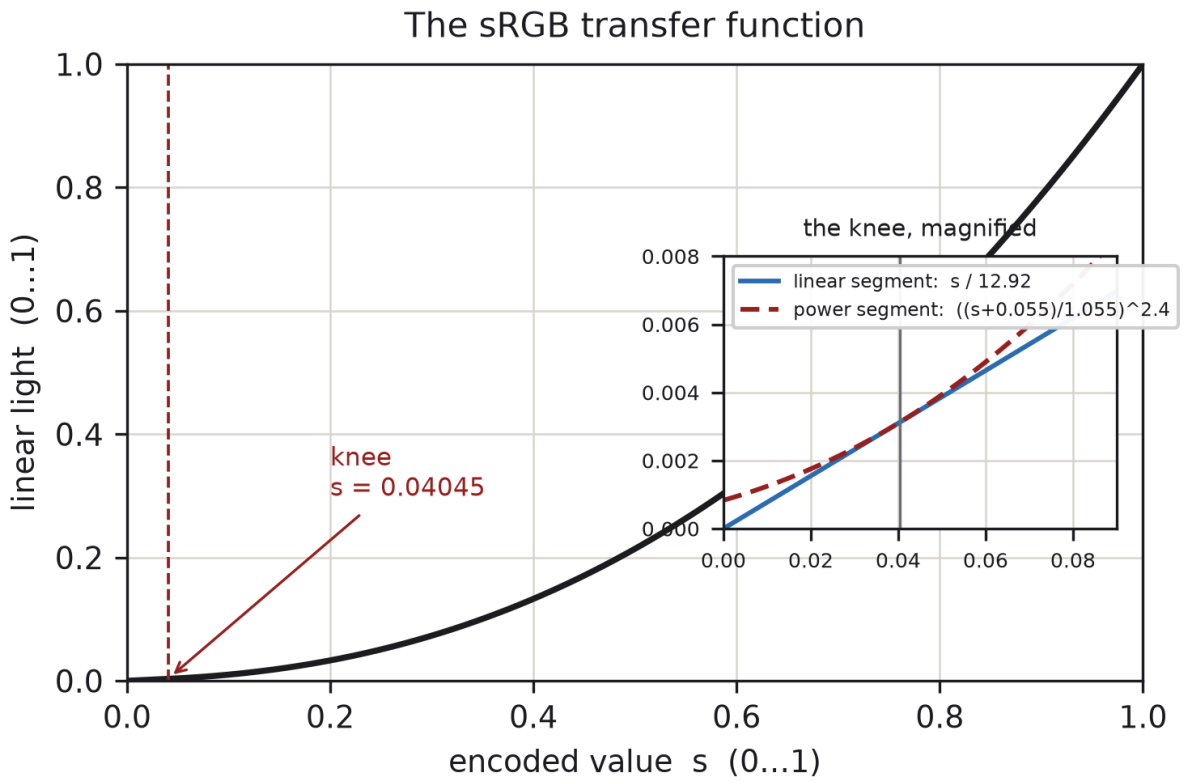


Figure 3. *The sRGB transfer function. In the magnified detail the two branches are separate: the blue straight line is the linear segment, the red dashed line the power function. At the knee the* **values** *of the two branches agree — the curve is continuous. Their slopes, however, differ by 1.71 %, so it is not smooth.*

> **The practical consequence**
>
> If someone computes directly with the hexadecimal values — averaging them, say, or subtracting one from another — the result is not a ratio of light but a meaningless number. The encoding must **always** be undone before anything is compared.

4. Relative luminance

The three linear channels collapse into a single number, the relative luminance:

```
L  =  0.2126·R_lin  +  0.7152·G_lin  +  0.0722·B_lin
```

The three coefficients come from the conversion of the ITU-R BT.709 primaries and the D65 white point into CIE XYZ — more precisely, from the **Y row** of that conversion matrix. In the CIE system the *Y* axis is luminance itself, aligned to the eye's $\bar{y}(\lambda)$ sensitivity function.

Put differently: the three numbers say how much **light** each primary contributes to white. They do not express "importance", and they are not an aesthetic weighting. They sum to 1.0000, because the relative luminance of full white is 1 by definition.

channel	coefficient	share of white
red (R)	0.2126	21 %
green (G)	0.7152	72 %
blue (B)	0.0722	7 %

Table 1. *The BT.709 luminance coefficients. Green alone carries more lightness than the other two combined: 3.4 times as much as red, and 9.9 times as much as blue.*

5. The contrast ratio and the screen flare

Under WCAG 2.1 the contrast of two colours is the ratio of their luminances, with the lighter one in the numerator:

```
CR  =  (L_lighter + 0.05) / (L_darker + 0.05)
```

What the 0.05 is doing there

The term is called **screen flare**. The surface of a real display reflects a little of the light in the room, so displayed "black" is never zero luminance. The 0.05 is the standard's assumed magnitude of that scattered light, as a proportion of the luminance of white.

The two ends of the scale follow directly. The smallest ratio is obviously 1:1, for two colours of equal luminance. The largest is between pure black and pure white:

```
CR_max  =  (1.00 + 0.05) / (0.00 + 0.05)  =  21:1
```

This is why 21:1 appears as the upper bound in every accessibility text. It is not a round number by accident: it follows from the choice of the flare term.

The thresholds

what it covers	WCAG criterion	AA	AAA
body text	1.4.3 / 1.4.6	4.5:1	7:1
large text (18 pt, or 14 pt bold ≈ 24 px, or 18.7 px)	1.4.3 / 1.4.6	3:1	4.5:1
non-text graphics, UI components, state indicators	1.4.11	3:1	—

Table 2. *The WCAG 2.1 contrast thresholds. The icons on the tiles belong in the third row: not text, but the interface cannot be understood without them.*

The threshold for the icons is therefore **3.0**. The measured **1.00–1.17** is **33–39 %** of that.

6. The case, measured

The icon colour was #7FE38A, with a relative luminance of **0.6129**. Computing the contrast ratio against each of the six pastel avatar backgrounds gives the following picture.

avatar background	hex	L	contrast with the icon	WCAG 1.4.11 (3:1)
green	#A8D5BA	0.5946	**1.03:1**	**FAIL**
pink	#F6C6D0	0.6454	**1.05:1**	**FAIL**
yellow	#F2C879	0.6157	**1.00:1**	**FAIL**
blue	#A9C7F0	0.5557	**1.09:1**	**FAIL**
lilac	#C9B6E4	0.5147	**1.17:1**	**FAIL**
turquoise	#9BD8DE	0.6135	**1.00:1**	**FAIL**

Table 3. *Contrast of the pale icon (#7FE38A, L = 0.6129) on the six avatar backgrounds. Not one value reaches the 3:1 threshold.*

The worst case is the **turquoise** background at **1.0010:1** — **yellow** is barely above it (**1.0042:1**), which is why the table shows the same figure for both at two decimals. The best is **lilac** at **1.17:1** — meaning that **not one** of the six reaches the 3:1 threshold. The entire range stays below **39 %** of the threshold: even the best case is barely more than a third of it.

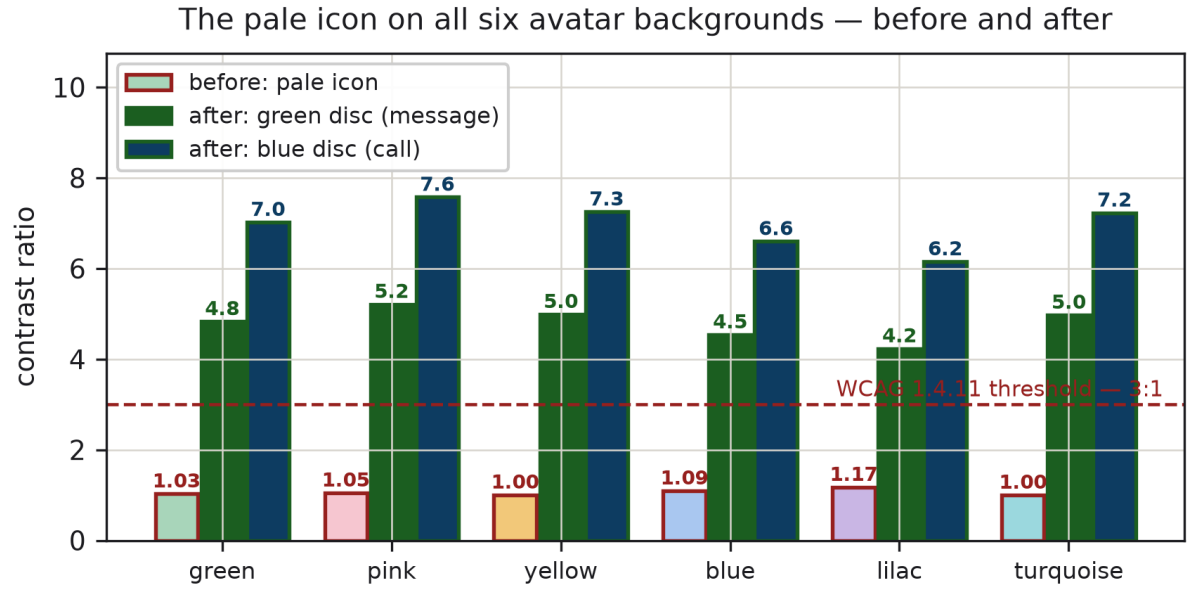


Figure 4. *The contrast of the pale icon on all six avatar backgrounds (red outline), and the same after the fix (green outline). The dashed line is the 3:1 threshold of WCAG 1.4.11. Before the fix not one bar reaches it; afterwards — shown here with the green disc, the worse of the two — every one is above it.*

> **How this was measured**
>
> The hexadecimal values come from the application's source code, and the contrasts are computed by the formula set out in sections 3–5. The 9.92:1 measured on a photographic tile, by contrast, is a *screen measurement*, because there the backdrop is the photograph itself — that number cannot be derived from hex values, and we did not pretend otherwise.

7. Colour vision deficiency: the second yardstick

The contrast ratio on its own says nothing about what happens to someone who sees colour differently. About **8 %** of men and about **0.4 %** of women of European descent have red–green colour vision deficiency (Birch, 2012). The figure is population-dependent: in Chinese and Japanese samples the rate measured among men is between 4 % and 6.5 %. The most common form is deuteranomaly.

In practice the matrices published by Machado, Oliveira and Fernandes (2009) are what people use for simulation — the vision simulation in several widely used developer and design tools is built on them. Three 3×3 matrices, scalable by severity; the full (1.0) cases are used here. Dichromatic cases have a reference of their own, the Brettel–Viénot–Mollon method, which projects in LMS space.

> **One detail that is often got wrong**
>
> These matrices are defined for **linear RGB**. Applying them straight to sRGB values gets you close — but it is not correct. The correct chain is **sRGB → linear → matrix → sRGB**. Every simulation in this paper was produced that way.

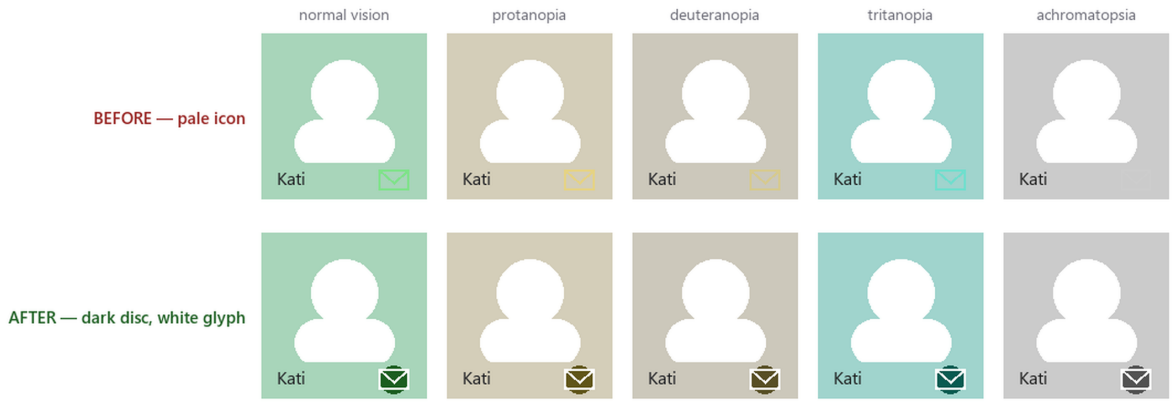


Figure 5. *The same two tiles under five vision modes. In the top row, the pale icon: still discernible in normal vision, it* **disappears entirely** *under achromatopsia. In the bottom row the fix — a dark disc with a white glyph — is equally legible in all five modes.*

It is worth noticing that the contrast figures barely move between the simulated modes. There is a measurable reason: across the nine colours used in this paper the simulation shifts relative luminance by at most **0.041** — most under protanopia, least under tritanopia (0.014). This is not a general law: the matrices are *not* luminance-preserving. On this palette, however, the lightness relations are effectively unchanged, which is why it holds here that **whatever passes on lightness also passes for a colour-deficient viewer**.

8. The rule that follows

> **The most important sentence in this paper**
>
> Hue does not appear in the luminance formula at all. Only lightness does. That is why two colours can be plainly distinguishable to the eye — in the measured case their hues are **59°** apart on the colour wheel — and still have a contrast ratio of **1.00:1**. More than that: on the nearly opposite **pink** background (139°) it is still only **1.05:1**.

From this follows a rule that can genuinely be written into code:

> **The rule**
> Never let hue be the only difference between two things. Create a **difference in lightness**. Hue may carry additional meaning, but it cannot be the carrier of the meaning — because for a share of your viewers it simply is not there.

This is not aesthetic advice. It is the form in which the mathematics of sections 2–7 becomes usable: a claim that a test can decide.

9. The fix, measured

The fix followed that rule precisely. The icon did **not** become a brighter green — that would have been the same mistake one notch higher. Instead it was given a **dark disc** of its own with a white glyph on it — and the colour of that disc now carries the meaning the icon itself used to carry:

— **message**: green disc, #1B5E20

— **call**: blue disc, #0D3C61

— the glyph on both is #FFFFFF — white

The measurement after the fix:

avatar background	hex	green disc	blue disc	WCAG 1.4.11 (3:1)
green	`#A8D5BA`	**4.83:1**	**7.02:1**	**PASS**
pink	`#F6C6D0`	**5.21:1**	**7.57:1**	**PASS**
yellow	`#F2C879`	**4.99:1**	**7.25:1**	**PASS**
blue	`#A9C7F0`	**4.54:1**	**6.60:1**	**PASS**
lilac	`#C9B6E4`	**4.23:1**	**6.15:1**	**PASS**
turquoise	`#9BD8DE`	**4.97:1**	**7.23:1**	**PASS**

Table 4. *After the fix: contrast of the green (`#1B5E20`, message) and blue (`#0D3C61`, call) discs on the same six backgrounds.*

The contrast between disc and background falls between **4.2:1** and **5.2:1** for the **green** disc and between **6.2:1** and **7.6:1** for the **blue** one — that is, **1.4–2.5 times** the 3:1 threshold. The white glyph on the disc reaches **7.87:1** (green) and **11.44:1** (blue) — both above the AAA level of 7:1.

> **What this shows**
> Hue did not disappear: the disc is green or blue, and it still carries the meaning. What changed is its **role**. Visibility no longer comes from hue but from the **lightness difference** between the dark disc and the pastel background — and hue cannot swallow that difference. This is exactly the rule of Section 8, written into code.

And, more importantly than that: the mark is now present even if the viewer sees no colour at all. The bottom row of Figure 5 shows this.

10. Turning this into a gate in code

A rule that nothing checks will eventually be broken. The mathematics above fits into a few lines, and once written down, continuous integration can ask the question again at every change.

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

The gate is then a single assertion. It matters that the threshold is *not* scattered through the code but stated once, with a name:

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

const UI_THRESHOLD = 3.0;   // WCAG 2.1 — 1.4.11

for (const background of AVATAR_COLOURS) {
  const cr = contrast(ICON_DISC, background);
  assert(cr >= UI_THRESHOLD,
         `${background}: ${cr.toFixed(2)}:1 < ${UI_THRESHOLD}:1`);
}
```

> **What the gate does not solve**
>
> The test above measures the icon on the *name band*, because there we know both colours. On a photographic tile we do not know the backdrop — there the legibility of the mark is *guaranteed* by the dark disc, not by a number. This distinction is worth stating out loud, because a green test easily creates the impression that it covers everything.

11. Where the WCAG 2.1 formula gets it wrong

Honesty requires saying it: the formula above is known to be inaccurate. It diverges most from what people actually see on dark backgrounds and in the midtones. The cause is the +0.05 term: at the dark end **it dominates the quotient**, not the actual luminance.

This is not an objection in principle; it can be computed. Against a white surface the ratio without flare grows without bound as black is approached, whereas the WCAG formula stops at 21:1. The two curves converge only slowly: at $Y = 0.05$ the flare-free ratio is still **1.90 times** the WCAG one, and the difference drops below 20 % only above $Y \approx \mathbf{0.19}$. At the dark end, then, the formula essentially flattens into a single number the very range in which the eye is most sensitive to difference.

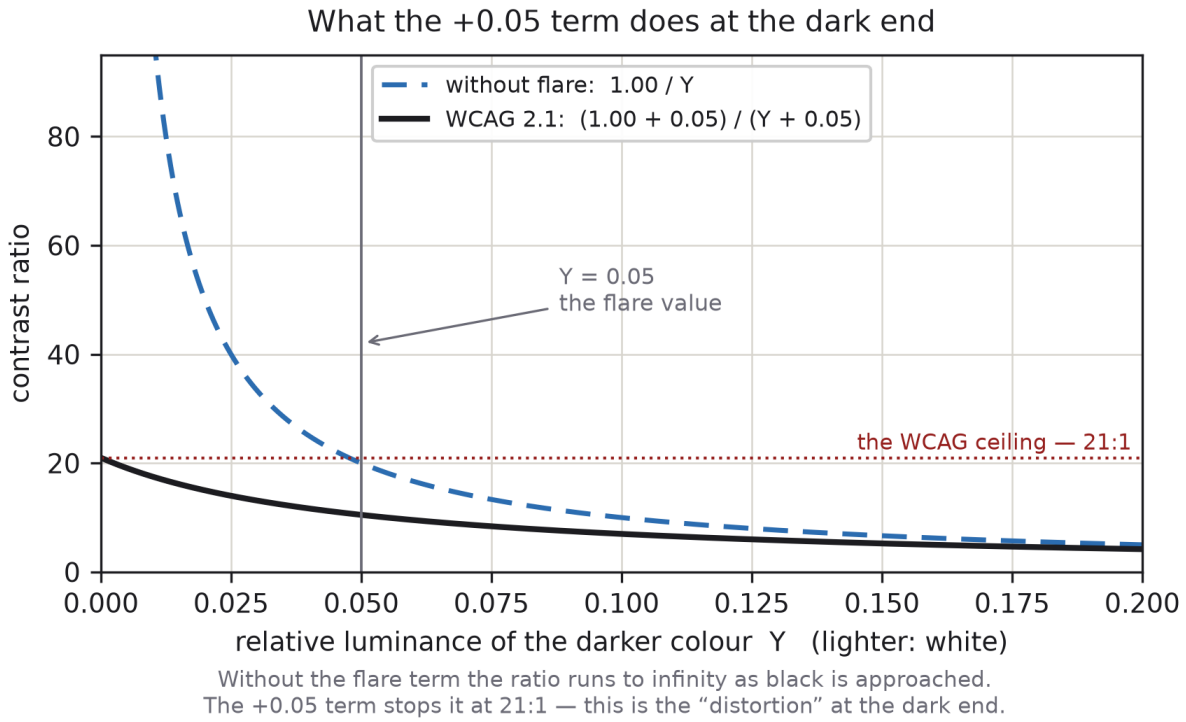


Figure 6. *What the flare term does. The lighter colour is white; the horizontal axis is the luminance of the darker colour. Without flare (blue dashed) the ratio runs to infinity as black is approached. The +0.05 term* **stops it at 21:1**. *The two curves converge slowly: at Y = 0.05 the difference is still 1.90-fold, and it drops below 20 % only above Y ≈ 0.19 — that is, the formula smooths away differences exactly where the eye is most sensitive.*

The successor to the formula is **APCA** (Accessible Perceptual Contrast Algorithm), which appears in the WCAG 3 working draft. APCA also models polarity — dark-on-light and light-on-dark are not the same case — and returns not a ratio but an *Lc* value roughly between −108 and +106.

> **The practical decision**
>
> What is normative today — what you can cite on a store listing, in a tender, or at a conference — is WCAG 2.1. So that is what we program against. We know APCA, and where it tells us more we listen; but conformance is stated in 2.1.

A note on Figure 6: it deliberately does *not* show APCA values. We have not verified the APCA formula by our own measurement, so we compute no APCA value for any case in this paper — the Lc range mentioned above is the algorithm's published range, not our measurement. CIE L*, by contrast, is a standard, and it proves the same point.

12. Limitations and further work

We do not know the answers to the following questions. They are not rhetorical: they arose on a working product, and their answers would change how we design.

1. **Is 3:1 enough on an emergency interface?** WCAG states a minimum, not an optimum. For an emergency button that a frightened, poorly-sighted, elderly person is hunting for — where is the real threshold? The standard was not written for emergency use.
2. **What should we do with the meaning of colour, now that lightness carries visibility?** Green = message, blue = call still holds. Does it still help someone who cannot tell the two apart — or does it actively confuse?

3. **Is the severity scale useful in practice?** The Machado matrices scale in steps of 0.1. Is a 0.6 deuteranomaly more realistic than full deuteranopia — or should design always target the worst case?
4. **Is the pastel palette defensible at all?** The six avatar colours are pastel because they feel friendly and carry dark text. But that is exactly what made a pale icon invisible. Is there a palette that is friendly *and* leaves lightness headroom for whatever is placed on it?

Two limitations of the method deserve to be stated separately. First, the simulation matrices are models, not people: they do not stand in for the vision of any particular individual. Second, this paper rests on measurements taken on a single product — the results indicate a direction, but they are not a statistical sample.

13. Conclusion

This defect was not found by a machine. No test of ours failed on it — indeed, in the environment it was designed for (over a dark photograph) the icon was flawless.

A person glanced at the phone and said so. Another, who has difficulty with colour himself, had already reported that he could not find any icon at all — and was told “the little icon is right there”.

He was right, and not subjectively: **mathematically**. That is what the formula is for — so that nobody has to discover the next one the hard way.

> **The three sentences that remain**
>
> **1.** Hue is not in the contrast formula — only lightness is.
> **2.** Therefore hue must never be the only difference between two things.
> **3.** What nothing checks will eventually break — so turn the rule into a test.

About the author

Péter Kürtösi is a mechanical engineer (MSc) who has been designing and building electronics, writing firmware and running systems for fifteen years. His work includes climate control, thermostats, test rigs and servers; most of it he designed, built and programmed himself.

The *Segítség* application that is the subject of this paper, and the wall console that goes with it, are built for elderly and convalescing people. The system is about to be installed in two care homes. The work did not begin as a commercial commission: it grew out of the observation that emergency signalling is most often missing exactly where it is most needed. More about the system and the devices under development: **vigyazokrad.hu**.

> **Contact and citation**
>
> Correspondence: Péter Kürtösi · peter.kurtosi@gmail.com
> Suggested citation: Kürtösi, P. (2026): *The Mathematics of Visibility*. A paper from practice, version 1.1. doi.org/10.5281/zenodo.22998702
> This paper may be freely copied and quoted for educational purposes with attribution.

References

None of the formulas or external data in this paper are our own: each comes from the sources below.

[1] **W3C (2018)** — Web Content Accessibility Guidelines (WCAG) 2.1. W3C Recommendation, 5 June 2018. In particular: 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, Geneva.

[4] **CIE (2004)** — CIE 15:2004 — Colorimetry, 3rd edition. Commission Internationale de l'Éclairage, Vienna. (Definitions of the CIE 1931 2° standard observer and of 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. (The analytic approximation used in the figures of this paper.)

[10] **W3C (working draft)** — WCAG 3 Working Draft and the description of 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.

Appendix A — reference implementation

The Python below contains the entire mathematics of this paper: the same routines that produced the figures and the numbers in the prose, with the identifiers translated — the published module (szinmatek.py, see Appendix B) names them in Hungarian. It has no dependencies.

```
import math

def srgb_to_linear(s):
    """Encoded sRGB channel (0..1) -> linear light."""
    if s <= 0.04045:
        return s / 12.92
    return ((s + 0.055) / 1.055) ** 2.4

def luminance(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_to_linear(r)
            + 0.7152 * srgb_to_linear(g)
            + 0.0722 * srgb_to_linear(b))

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

The colour-vision simulation additionally needs the severity-1.0 matrices published by Machado et al. (2009), applied in **linear** RGB and then re-encoded to sRGB. The achromatopsia case is simpler: there only the luminance survives, on all three channels.

Appendix B — reproducibility

Not one figure in this paper is a drawing. Each computes from the same module that supplies the numbers in the prose, and the whole document rebuilds with two commands:

```
python abrak.py en      # the six figures, 300 dpi
python pdf.py en        # the paper
```

— szinmatek.py — the mathematics: sRGB, luminance, contrast, CIE L*, colour matching functions, simulation matrices

— abrak.py — the six figures, all from the module above

— tartalom_en.py — the prose; markers stand where numbers belong, and the builder substitutes them

— pdf.py — the typesetting

> **Why it is done this way**
>
> If a number in the prose came from a different source than the number on the figure, they would drift apart sooner or later. In teaching material that is the worst thing that can happen: the student cannot tell which one to believe. Done this way, a figure cannot contradict the text.