Discussing Pixel Response Time Testing Methodologies

Andrew McDonald*

March 8, 2024

Abstract

Pixel response time measurements have had the same testing methodology applied for over two decades. While the standard method of testing is in line with other electronics industry testing and standards, in the specific application of pixel response times, the current standard is not fit for purpose. Both the tolerance values selected - 10% to 90% - and the lack of gamma correction means current response time figures are arbitrary and inaccurate to end-user experience. The lack of inclusion of overshoot, and overshoot time, is also a significant loss in the current standard's ability to accurately describe the panel's behaviour. In this paper I would like to propose several possible solutions to these problems, and discuss their possible benefits and drawbacks, alongside their implementation.

Introduction

Pixel response times are an important factor in explaining the visual experience of a monitor. As a basic definition, the pixel response time is how long any given pixel takes to change colours. This can be a large change, such as full black (RGB 0) to full white (RGB 255), or minor changes, such as Grey-to-grey. It can also include the individual sub-pixel response times, or the whole-pixel response time, depending on what is being tested. The way pixel response times have been tested has generally been laid out in standards documents, originally from VESA [1], although those same standards have now been transferred to the International Committee for Display Meteorology and their Information Display Measurements Standard [2]. This testing methodology has not changed substantially since their initial formation, and in this paper I would like to present arguments as to why I believe these standards are outdated and in need of renewal. I will present a selection of possible alternatives, and discuss their limitations and benefits.

As a brief introduction to myself and my qualifications to discuss this topic, I have been professionally reviewing technology - specifically focusing on the PC and gaming markets - for over a decade. I have, in particular, reviewed over 150 different displays and over 100 laptops as well. In late 2021 [3], I launched the Open Source Response Time Tool (OSRTT), which aimed to allow reviews like myself test displays with accuracy and ease. Through four total iterations, including the most recent 'Pro CS' model [4], I have collected and viewed data on hundreds of displays from reviewers around the world. In creating these tools, I have come to understand a great deal about pixel response times, the factors that affect the calculated figure, and how those figures then represent the end-user experience.

*Corresponding author: inbox@techeamgb.com
Published: March, 2024

Methodologies

1. VESA Standard

To be able to give a point of comparison, I would like to start with the existing VESA standard, now illustrated in IDMS V1.2 [2], as seen below:

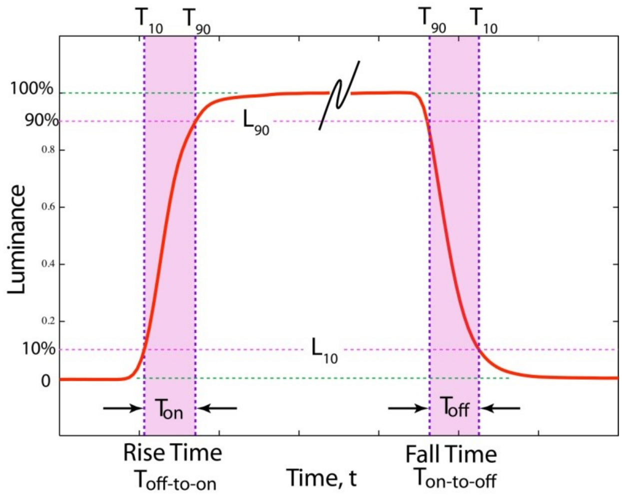

Figure 1: *Page 208, IDMS Version 1.2, 2023 , https://www.sid.org/Standards/ICDM.*

This graph, and the accompanying "Procedure" detail the process of capturing the prescribed response time figures. The instructions are to change the set point of the display from one shade to another, capturing the light level over time. Then, measure the time difference between 10% of the light level to 90% of the light level, by using the following formulae:

$$L_{range} = L_W - L_K$$

$$L_{10} = 0.1 * L_{range} + L_K$$

$$L_{90} = 0.9 * L_{range} + L_K$$

Where L_K is the light level at the darker shade, and L_W is the light level at the lighter shade, and repeat for the inverse (falling) transition. Specifically prescribed for the "Gray-to-Gray Response Time" [5], it gives an example of an equal "gray level set: 0,31,63,95,127,159,191,223,255" The above measurement procedure is to be repeated using this set, giving a matrix with $M(M-1)$ non-zero transitions. In the example case that is $9x8 = 72$.

This 10 to 90 percent tolerance is the main point of focus here, along with the location of the latter measurement position. This 10-90% measurement is the industry standard when describing electronics characteristics. Products such as the Melexis MLX75305 I used in the original Open Source Response Time Tool, for example, will often quote their rise time with 10 to 90 percent tolerances. This tolerance, described by Yang and Levine [6] as the choice for "underdamped" systems, is only one option they describe. Critically damped systems are said to use 5% to 95%, and overdamped systems use 0% to 100%. That is to say, these tolerances are not set in stone, in the book, Levine and Yang note these values are just the "common" choices.

Using tolerances on the whole is not without merit. Both due to inaccuracy of test equipment, and in an effort to provide a reasonable value that more closely reflects the end-user experience, including some tolerance to the top and bottom measurement points makes a lot of sense. As I will discuss below though, the choice of tolerance is quite important in more closely reflecting the end-user experience.

The other important factor that the above method doesn't cover, although the document itself does include [7], is that of "overshoot" and "undershoot". This is demonstrated in the chart below as L_{RO} and L_{FU}:

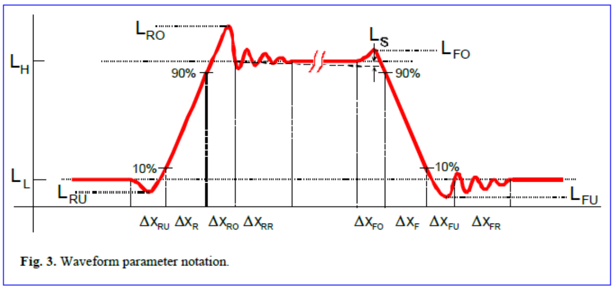

Figure 2: *Page 164, IDMS Version 1.2, 2023 , https://www.sid.org/Standards/ICDM.*

Overshoot is an artefact of monitor overdrive, a technology which purposefully sets the pixels to an exaggerated value [8] to make the pixels respond faster, then sets the pixel's target back to the actual desired level, ideally in time such that the pixel doesn't overshoot the target. Particularly bad overshoot creates an effect called "Inverse Ghosting" or "Corona Artifacts" [9], which as displayed below can be just as, if not more visually distracting than regular ghosting, which is created when the pixel response time exceeds the monitor's refresh rate, causing a copied "ghosted" frame to be visible at the same time as the most recent frame.

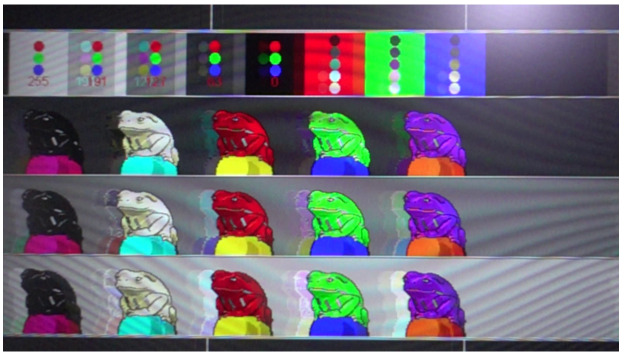

Figure 3: *Visual display of overshoot behaviour with Aperture Grille's Frog Pursuit test, Asus XG27ACS, Overdrive mode 19*

Overshoot, as described in the IDMS document is the distance between L_{RO} and L_H, or L_L and L_{FU}, normally represented as a percentage of the final light level. This is important, as we are describing the effects of a visual stimulus which impacts the end-user experience. The worse the overshoot is, regardless of the rise time itself, the more it can become a significant detractor to the visual experience.

Discussion of the VESA Standard The VESA Standard, as implemented originally in 1998, uses a somewhat rudimentary set of tolerances. Using the light level output seems reasonable at first, but as I will discuss below, when taking into account gamma curves, and specifically the human eye perception of light levels [10], this ends up being a wildly inaccurate method for testing. Equally, using the admittedly industry standard 10% to 90% tolerance is not accurate to end-user experience. Take the below RGB 0 to RGB 255 graph as an example:

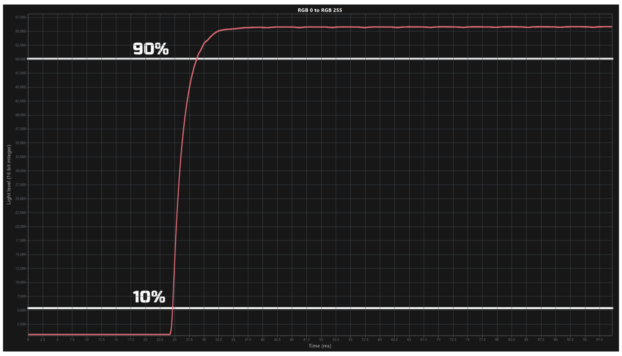

Figure 4: *Asus XG27ACS, RGB 0 to RGB 255 transition, OSRTT Pro.*

Given that the average final light level value is 55,750, and the average initial light level value is 675, and the VESA formula of $L_{Range} = L_W - L_K$, we get:

$$55,750 - 675 = 55,075$$

Calculating the 10% value with the VESA formula:

$$L_{10} = L_{Range} * 0.1 + L_K$$

we get:

$$55,075 * 0.1 + 675 = 6182.5$$

Calculating the 90% value with the VESA formula:

$$L_{90} = L_{Range} * 0.9 + L_K$$

we get:

$$55,075 * 0.9 + 675 = 50,242.5$$

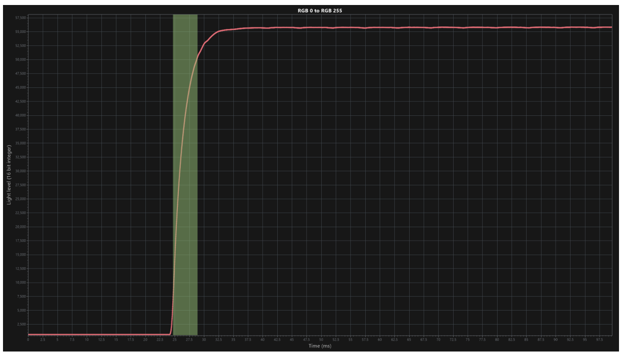

Figure 5: *Asus XG27ACS, RGB 0 to RGB 255 transition with block displaying response time, OSRTT Pro.*

Taking those positions, in this case (an Asus XG27ACS on Variable OD mode 1), we get a rise time of 4.1 milliseconds. However, using the gamma table OSRTT captured and calculated during the test, below are the two actual colours that we are cutting our timing off at:

Figure 6: *RGB 255 (Left), RGB 240 (Right)*

This is the top end 90% limit, where the target is RGB 255, but the cutoff stopped counting at RGB 240. This is a visually perceptible difference, although thanks to the gamma curve, not a truly significant one.

Taking those positions, in this case (an Asus XG27ACS on Variable OD mode 1), we get a rise time of 4.1 milliseconds. However, using the gamma table OSRTT captured and calculated during the test, below are the two actual colours that we are cutting our timing off at:

Figure 7: *RGB 0 (Left), RGB 92 (Right)*

The 10% limit though is incredibly significant. The target here is RGB 0 - full black - but with this measurement method, we are not starting to count (or more importantly for the reverse falling transition from RGB 255 to RGB 0, stopping the count) until RGB 92. That is 36% of the total colour space we are not including in our timing. To consider RGB 92 a reasonable point to decide a transition has started or finished, when RGB 0 was the true target, is frankly a ridiculous notion. It is with this premise I believe the VESA standard is not fit for purpose as the industry standard methodology for testing the pixel response time of displays.

Display manufacturers have an obvious incentive to maintain this standard, as especially for LCD panels, the latter part of a transition is by far the slowest part, and that slow section is exactly what the 10% to 90% tolerance excludes. As an example, the chart below is the same RGB 0 to RGB 255 transition from the Asus XG27ACS as shown above, but zoomed in to the slow part of the transition. The 90% light level endpoint - where VESA would have you stop counting the response time - is marked, as is the true transition complete time. That truly complete point is a full 7 milliseconds later than the suggested endpoint.

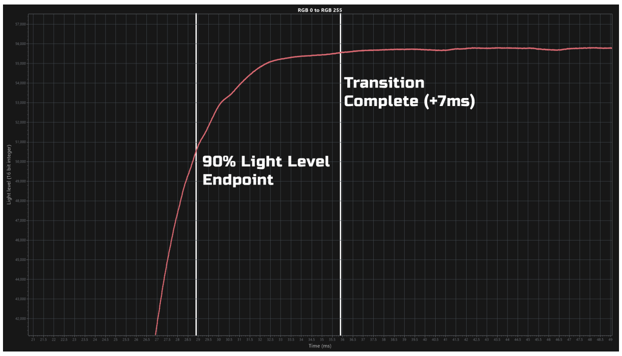

Figure 8: *Asus XG27ACS, RGB 0 to RGB 255 transition with end points marked, OSRTT Pro.*

This creates an incentive for monitor manufacturers to adhere to this less stringent standard, allowing them to quote incomplete figures, primarily in marketing materials in the hopes of convincing prospective buyers that their display is faster than it may be in real-world usage.

I also take issue with the point at which the endpoint is measured, at least in part. The VESA spec, now part of the IDMS document [7], does note other measurement points, namely XRR, which is the post-overshoot time taken, although it doesn't quote this as a measurement to be taken note of. I have taken to calling these the "initial response time" - referring to the traditional rise or fall time - and the "perceived response time" - referring to the post-overshoot time. As an example - even using the 10% to 90% light level tolerance - see the below chart:

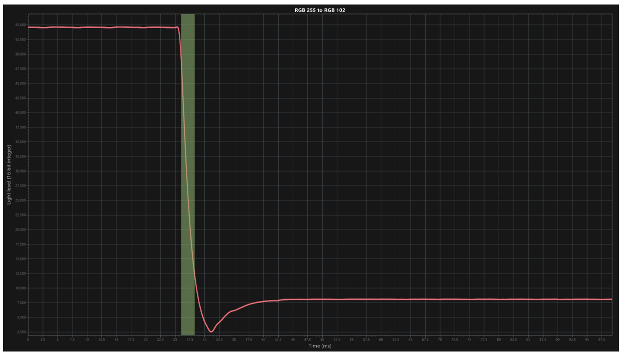

Figure 9: *Asus XG27ACS, RGB 255 to RGB 102 transition with fall time marked, OSRTT Pro.*

This is the same Asus XG27ACS monitor, although now on Variable OD mode 19. Using the rise (or in this case fall) time, this result only reports as 2.3 milliseconds, however we have undershoot that represents 45 RGB values too low, and a considerable recovery time. This is time that the frame has not fully rendered, but instead has inverted and become often more noticeably discomforting than if the panel took just as long, but without the undershoot.

Figure 10: *Asus XG27ACS, RGB 255 to RGB 102 transition with perceived response time marked, OSRTT Pro.*

Taking into account the same 10% light level tolerance, but including the overshoot time, we get a "perceived response time" of 5.8 milliseconds. This is a more accurate figure to what the end-user will see and experience, therefore having a secondary measurement of the "perceived" response time, I feel, is an important part of communicating the real-world experience of using the monitor.

2. Gamma Correction

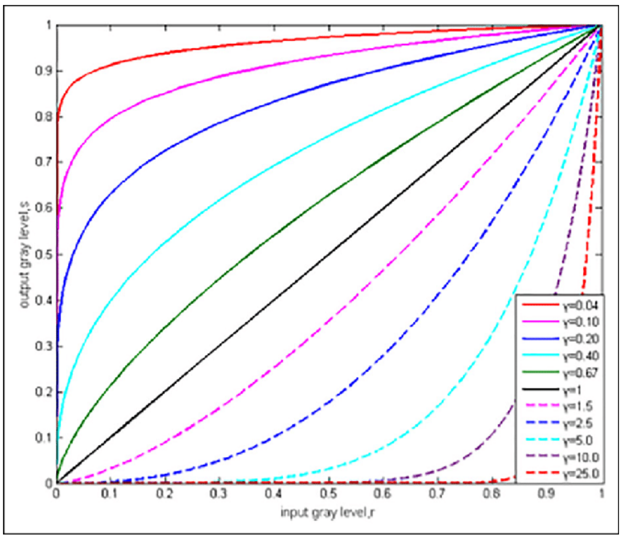

Figure 11: *Dong et al., 2018*

Gamma correction is the process of looking at the light level data, not as a standalone result, but viewing it with the context of how humans perceive differences in brightness and colour.

To quote Charles Poynton from "GammaFAQ" [11]:

> "Human vision has a nonlinear perceptual response to brightness: a source having a luminance only 18% of a reference luminance appears about half as bright."

This fact is why in the above section, despite only using 10% of the light level difference, on the brighter end there is fairly little perceptible difference between the target and actual RGB values, whereas on the darker side, 10% of the light level equates to 38% of the RGB values. This dichotomy is why many reviewers - such as Hardware Unboxed [12] - measure response times with a "Gamma Corrected" procedure. This difference comes from displays using a gamma curve. Gamma is generally described as

$$V_{out} = A * V_{in}^{\gamma}$$

although for the purposes of displays, it is more generally seen as

$$V_{out} = V_{in}^{\gamma}$$

with common values being: 1.8, 2.0, 2.2, 2.4 and 2.6. The higher the power factor, the more non-linear the gamma curve will be, and vice versa for lower power figures.

Gamma correction of response time data does add further steps to the testing process, as you need to measure a series of RGB values to build a table where you can compare light levels and RGB values when doing your response time calculations. OSRTT measures fifteen equal steps of 17 RGB values, then runs a natural spline interpolation function to interpolate the in-between RGB values worth of light levels. It also extrapolates upwards past RGB 255, up to RGB 306, to help catch and return results for monitors that manage to overshoot RGB 255.

Figure 12: *RGB 255 (Left), RGB 230 (Right)*

As an example, just using 10% of the RGB values instead of 10% of the light level change, we get a very similar top end target value, although as seen below, the low end result is much, much closer to the actual target.

Figure 13: *RGB 0 (Left), RGB 25 (Right)*

This means the response time start and end points more closely represent the actual user experience, and provide a more accurate figure for users to understand the quality of the product.

The primary discussion around gamma correction is what metric should be used for the tolerance. The primary two choices are using a fixed RGB value, versus using a percentage of the RGB value difference between the start and end points - much in the same way the LRange is calculated in the VESA Standard.

2.1. Percentage of RGB Range

Using a percentage of the RGB range of the transition seems like it is the logical follow-up to the original, non-gamma corrected, measurement style. It follows the exact same formulae:

$$L + range = L_W - L_K$$

$$L_{10} = 0.1 * L_{range} + L_K$$

$$L_{90} = 0.9 * L_{range} + L_K$$

although we swap the luminance values for the RGB values, so taking RGB 51 to RGB 255 as an example:

$$255 - 51 = 204$$

$$204 * 0.1 + 51 = 71.4$$

$$204 * 0.9 = 51 = 234.6$$

It maintains the same consistency relative to the transition, although at the cost of variability between tests. That is illustrated in the table below:

Table 1: *10% RGB level tolerance values (isolated)*

	0	51	102	153	204	255
0	0	5.1	10.2	15.3	20.4	25.5
51	5.1	0	5.1	10.2	15.3	20.4
102	10.2	5.1	0	5.1	10.2	15.3
153	15.3	10.2	5.1	0	5.1	10.2
204	20.4	15.3	10.2	5.1	0	5.1
255	25.5	20.4	15.3	10.2	5.1	0

As you can see, the larger the transition, the larger the RGB value difference during the transition, the larger the tolerance will be. These differences can be quite large, where in this table we have results that are up to five times larger than the smallest tolerances (5.1 versus 25.5). While this is an intrinsic property of this style of measurement, the effect can be lessened by changing the percentage used.

Table 2: *3% RGB level tolerance values (isolated)*

	0	51	102	153	204	255
0	0	1.53	3.06	4.59	6.12	7.65
51	1.53	0	1.53	3.06	4.59	6.12
102	3.06	1.53	0	1.53	3.06	4.59
153	4.59	3.06	1.53	0	1.53	3.06
204	6.12	4.59	3.06	1.53	0	1.53
255	7.65	6.12	4.59	3.06	1.53	0

In this table, the percentage chosen is 3%. This aligns with what Tim from Hardware Unboxed tests with [12], and clearly these tolerances are considerably more strict. With just 1.5 RGB values worth of tolerance in the lowest case, and only 7.65 RGB values on the largest end, this is an incredibly strict testing methodology. This has the benefit of including a considerable amount of the slow portion of the curve, although potentially at the cost of inconsistent and equally misleading data as it is difficult to argue that a single RGB value is something that can be easily distinguished between, especially when it comes to the inherently fast motion of a response time.

2.2. Rixed RGB Value

The alternative to using a percentage tolerance is to fix the tolerance at a given number of RGB values above or below the targets. This provides consistency between transitions, as every transition - regardless of the difference between the starting and ending colours - keeps the same tolerance value. This does essentially invert the variability, from transition to transition, to a variable amount of distance between points depending on the transition size. This method is also easier to compute, due to the lack of need to calculate what the tolerance values are for any given transition.

The two main options commonly used with the Open Source Response Time Tool are RGB 5 and RGB 10 tolerances, with the latter preferred by Simon from TFT Central [13], and the former preferred by myself (although I'm coming round to his reasoning).

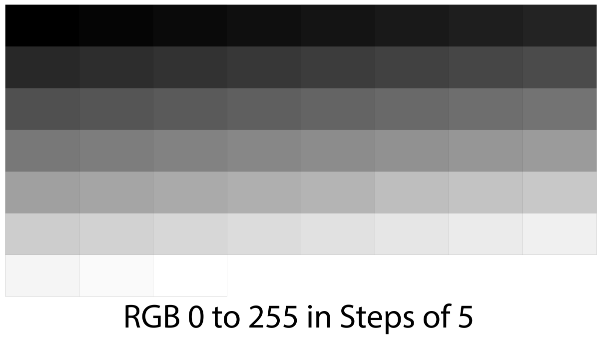

Figure 14: *RGB 0 to RGB 255 in steps of 5 RGB values*

To illustrate why these tolerances might be useful, the above graphic displays RGB 0 to RGB 255 in steps of 5 RGB values. While comparing, say, cell 2 row 1 with cell 2 row 2 is an obvious and easy to distinguish difference, comparing neighbouring cells is much more difficult.

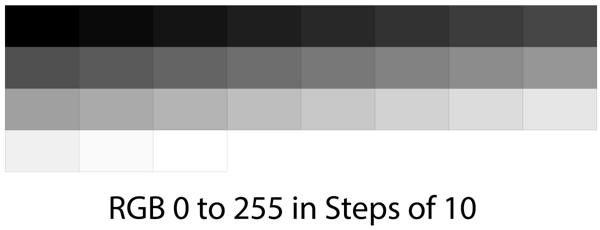

Figure 15: *RGB 0 to RGB 255 in steps of 10 RGB values*

Compare that to the above graphic which displays RGB 0 to RGB 255 in steps of 10 (save for the final RGB 5 step), where the differences between the two steps are much more perceptible. My view when creating the Open Source Response Time Tool was to create a stringent standard to help motivate the monitor industry to improve the display panels, and so choosing a target which is just outside easy perception made sense. However, throughout the process of testing displays with OSRTT and OSRTT Pro tools, I have come to understand the nuance of these results, and now more closely align with Simon from TFT Central's view that the RGB 10 tolerance is a better fit. When discussing this tolerance point via the lens of human perception, I feel it is important to remember the context that these transitions are measured in milliseconds, and while we may be able to perceive a slight difference between 10 RGB values after seconds of exposure, when discussing response times which are primarily relevant only between the refresh rate window, I believe a looser tolerance is still valid, and perhaps even preferred due to it's better accuracy and reliability.

2.3. Overshoot and Perceived Response Times

While overshoot is almost always reported alongside the response times, even from sources using the VESA Standard methodology, it is almost always reported as a percentage over the final light level. Take the following as an example:

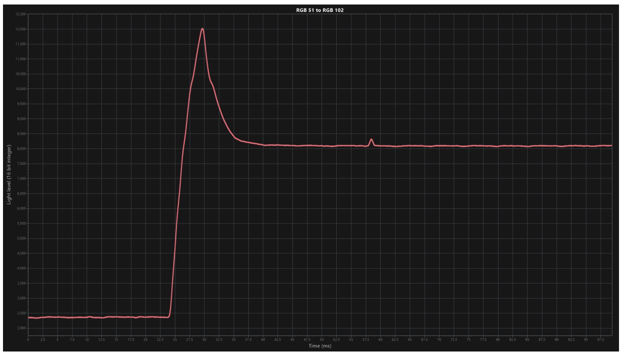

Figure 16: *Asus XG27ACS, RGB 51 to RGB 102 transition showing a large overshoot spike*

Here we have a pretty significant overshoot spike. The average target light level is around 8,100, and the peak spikes to 12,000. As a percentage, that translates to 48% higher than the target light level. Here, immediately, we run into the same problem when not gamma-correcting measurements. The final light level here was RGB 102, and this peak, if we use that 48% measurement, should have measured at around RGB 151. In practice, this peak equates to RGB 129, which only equates to 26.5% of the final RGB value. This proportionality inverts for undershoot, where visibly smaller dips equate to much larger perceived differences - but either way you look at it, it's clear that just reporting it as a percentage of the final light level is not an accurate methodology.

There are alternate ways to measure that overshoot percentage - with or without gamma correction - such as using the light level range as the comparator. Following the above example, that would be:

$$8,100 - 2,370 = 5730$$

$$12,000 - 2370 = 9630$$

$$9630 - 5730 = 3900$$

$$(3900/5730) * 100 = 68$$

In other words, 68% overshoot. This induces variability based on the size of the transition, as larger transitions with proportionally smaller overshoot waves - even those that display the same final amount of overshoot over the end light level - will effectively score 'better' than smaller transitions with the same amount of overshoot. I don't recommend this method of calculating overshoot.

The more stable solution is to both gamma correct the overshoot to normalise the response to what end-users will perceive, and to report the number of RGB values the transition over or undershot by. This provides a level of consistency across the suite of transitions, as a 10 RGB overshoot will look the same regardless of whether the transition started from RGB 0 or RGB 101, and whether the end light level is RGB 102, or RGB 204. This is the default in the OSRTT software - although all five options are made available should you prefer one of the other options.

One of the key reasons for highlighting the overshoot figure is that with the VESA Standard methodology, the time the panel takes to actually come to rest at the target light level is never taken into account or reported. I feel this is a significant omission, as while the initial response time (as in the first time the panel gets within the tolerance range of the end light level) is incredibly important, if the panel then takes another full frame with inverted colours, that is not what I would consider a "finished transition", and reporting it as such doesn't seem all too accurate.

To solve this problem, as mentioned above, I would like to propose an additional metric, the "perceived response time". This metric allows for the inclusion of the overshoot time, while still including the light level tolerance on either side of the end light level. As an example:

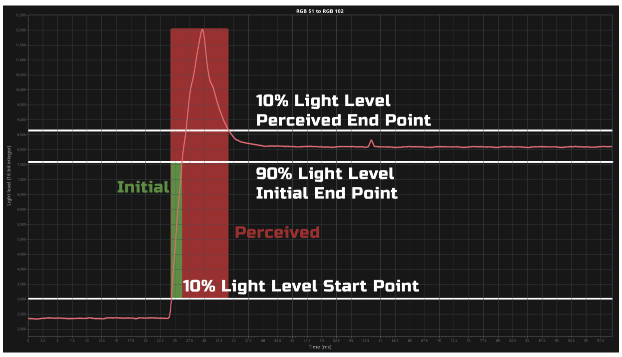

Figure 17: *Asus XG27ACS, RGB 51 to RGB 102 transition with response times marked*

Using the 10% of the light level tolerance as an example, you can see the initial response time is a tiny fraction of the perceived time, as both the generous tolerances and the considerable amount of overshoot impact the results. In this specific example, the initial response time is an impressive 1.8 milliseconds, while the perceived response time is an astonishing 9.7 milliseconds. That includes not counting the slowest part of the transition as it finally comes to rest. I would argue that the 9.7 millisecond figure is a more accurate choice when trying to describe the real-world behaviour of the panel, as the initial response time effectively becomes an arbitrary point to stop counting, rather than the logical conclusion drawn from the data and the real-world experience.

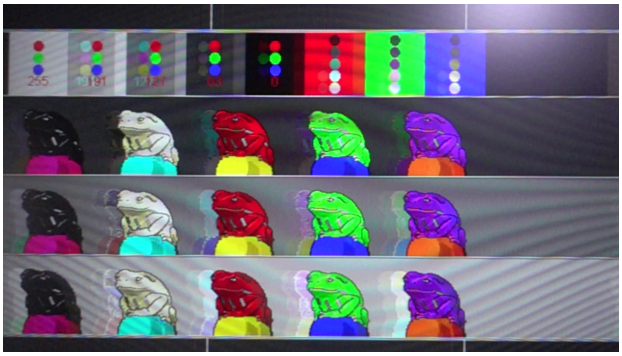

Figure 18: *Visual display of overshoot behaviour with Aperture Grille's Frog Pursuit test, Asus XG27ACS, Overdrive mode 19*

Just to emphasise this point, in the above image, the "initial response time" considers this a fully transitioned frame. A completed colour change. That is plainly not the case, and so including that overshoot time - either in place of, or alongside the initial response time - makes sense to provide the most accurate data to describe the quality and experience of using the panel.

Discussion

The limitations of the VESA Standard are plentiful. The overly loose tolerance, the lack of overshoot time reporting, the lack of gamma correction, it all adds to a less useful result that doesn't serve its purpose of accurately describing the experience of using the display. This has led to some negative consequences in the monitor market. The primary of which is the fact that when monitor manufacturers quote a response time, it is entirely separated from the real-world experience of using the monitor. Almost all 'gaming' marketed displays carry a "1ms GtG response time" claim, yet none meet that claim upon testing. A more stringent standard would lead to more accurate reporting of these figures, giving prospective buyers more useful information to make an informed purchasing decision.

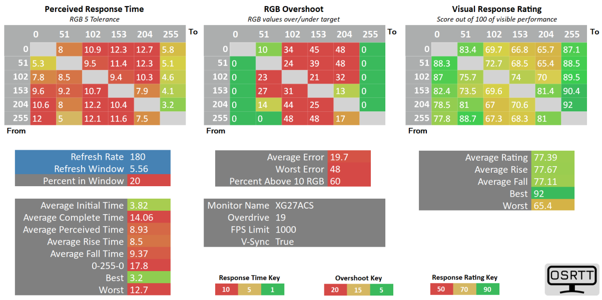

Figure 19: *OSRTT Result Heatmaps, Asus XG27ACS, Overdrive mode 19*

The more practical consequence of this standard has been what many in the review space call "marketing mode" overdrive settings. Often the maximum overdrive mode monitors ship with present the user with completely unusable experiences, purely so they can eke out a 'legitimate' 1 millisecond result, at the cost of horrendous overdrive and painfully long perceived response times. In the example above you can see the average initial response time is just 3.8 milliseconds, where the average perceived response time is well over double that at 8.9 milliseconds.

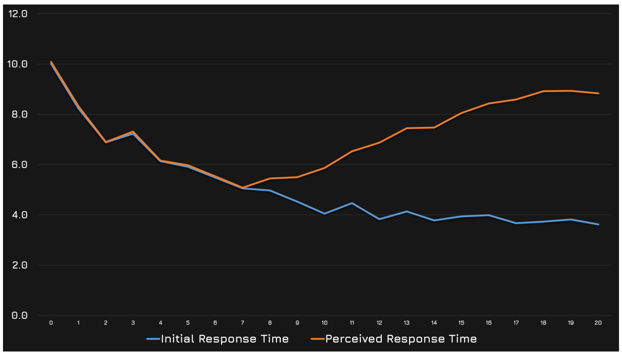

Figure 20: *OSRTT Response Time Results, Asus XG27ACS, Overdrive mode 0-20*

This is all of the average initial and perceived response time data from the Asus XG27ACS I collected for my review [14], and you can see the incongruence between the initial and perceived response times once the overdrive mode begins to produce significant overshoot. Asus claims this is a "1ms" panel [15], and with a cherry-picked methodology and result, you can get a 1 millisecond result, but it is in no way an accurate representation of the panel's real-world performance.

As for gamma correction, while it does add a significant amount of complexity to the test procedure, the accuracy it affords is a worthwhile tradeoff - especially when automated tools like OSRTT already support that function. The discussion around which tolerance style and value to use is still very much open, although in my experience I would suggest the fixed RGB 10 offset to be the one best suited to most closely match user experience while leaving enough room for noise in the raw data, giving the most reliable set of results.

It is worth noting that, in essence, what these figures - both the response time and overshoot results - aim to do is to describe the curve in as close to a single number as possible. As such, there are a number of other metrics we can calculate to describe that curve. I have implemented one such metric, called the "Visual Response Rating", and Tim from Hardware Unboxed has created what he calls "Cumulative Deviation" [12]. While I feel it is important to mention those here, it is outside the scope of this paper to discuss them in detail.

Conclusion

The VESA Standard, now found in the IDMS document [2], is outdated, and doesn't provide an accurate representation of a display's performance. The primary fix for this is to gamma correct the measurement points, although including the time spent in an overshoot or undershoot state may prove beneficial in characterising a panel's performance. Equally, reporting overshoot as RGB values above/below the target may be a more consistent format to describe that trait.

Acknowledgements

Simon from TFT Central - Simon was instrumental in the early testing, validation and cementing of methodologies for the Open Source Response Time Tool. I defer to his expertise in many matters regarding display testing, and I would recommend anyone interested in learning more about display to visit his website: https://tftcentral.co.uk/

Tim from Hardware Unboxed - Much of the early discussion of the methodologies came from a call with both Tim and Simon, and I have learned a great deal from his excellent work reviewing monitors. His monitor specific channel is well worth a subscription: https://www.youtube.com/@monitorsunboxed

Eric from Aperture Grille - His work on his own DIY response time tester was the onus for me to build OSRTT, and he was incredibly helpful in providing his code and advice at the fragile early stages. His work on the Unreal Engine projects, namely his "Frog Pursuit" project, has been an excellent tool in testing displays. His website is: https://www.aperturegrille.com/

OSRTT.COM - Of course, I can't go without linking to the OSRTT project's site, where you can purchase an Open Source Response Time Tool kit for your own testing. The units are hand-built at my desk, and are validated for accuracy before being shipped worldwide.

References

[1] Video Electronics Standards Association FPDM Task Group. "Flat Panel Display Measurements Standards v2.0". In: (2005). DOI: 10.55410/QUSG6326.

[2] The Society for Information Display. 2023. URL: https://www.sid.org/Standards/ICDM.

[3] TechteamGB. *I've been working on this for 6 months....* 2021. URL: https://www.youtube.com/watch?v=T5ZtN-U05fI.

[4] TechteamGB. *OSRTT Pro CS - My NEWEST Open Source Response Time Tool*. 2024. URL: https://youtu.be/e0kQ7uc3vTE.

[5] The Society for Information Display. *IDMS V1.2*. 2023, pp. 209, 210. URL: https://www.sid.org/Standards/ICDM.

[6] William S. Levine. "The Control Handbook". In: 1998, p. 158.

[7] The Society for Information Display. *IDMS V1.2*. 2023, p. 164. URL: https://www.sid.org/Standards/ICDM.

[8] Jun Wang, Kyeong yuk Min, and Jongwha Chong. "A high compression ratio image coding for frame memory reduction in LCD overdrive". In: (). URL: https://www.researchgate.net/publication/268438201_A_high_compression_ratio_image_coding_for_frame_memory_reduction_in_LCD_overdrive.

[9] Blur Busters. *LCD Motion Artifacts: Overdrive*. 2013. URL: https://blurbusters.com/faq/lcd-overdrive-artifacts/.

[10] Dong H et. "Secondary segmentation extracted algorithm based on image enhancement for intelligent identification systems." In: *International Journal of Distributed Sensor Networks* (2018).

[11] Charles Poynton. *GammaFAQ*. 1998. URL: https://poynton.ca/PDFs/GammaFAQ.pdf.

[12] Hardware Unboxed. *What Are Response Times? Overshoot? Ghosting? Monitor Testing Overhaul For 2021*. 2021. URL: https://www.youtube.com/watch?v=-Zmxl-Btpgk.

[13] TFT Central. *Response Time Testing*. 2021. URL: https://tftcentral.co.uk/articles/response_time_testing.

[14] TechteamGB. *Asus XG27ACS Review - Budget 1440p 180Hz 1ms Gaming Monitor*. 2024. URL: https://www.youtube.com/watch?v=dvNlCtiMBL0.

[15] Asus. *ROG Strix XG27ACS*. 2024. URL: https://rog.asus.com/monitors/27-to-31-5-inches/rog-strix-xg27acs/.