**OSRTT Guide – Understanding the Many Options, Deciding on Recommended Settings, and Ensuring Consistency Across Reviewers**Issue 1.1, last update 4 May 2022

**Introduction**

The OSRTT provides a wide range of options and features designed to test all elements of a display’s performance for pixel transitions and response times. There are many ways to analayze the data, and many possible ways you may wish to present the data to your readers and viewers depending on what you are trying to capture and cover. To ensure there is consistency across the industry as much as possible, and clear guidance for an average consumer trying to make sense of the data, it’s important that we try to retain some level of uniformity if we can in the way this data is presented.

This guide is to explain what the different options and measurements methods can do, how they might be interpreted and how ideally they should be presented to readers/viewers to achieve that consistent approach. It covers the “g2g” response time measurements as well as “overshoot” calculations.

The guide has been produced in conjunction with several established reviewers who already carry out this kind of testing, including **TFTCentral** **and Hardware Unboxed**. The aim is for it to represent an “industry standard” for this kind of display measurement if possible.

**TL;DR: Summary of Recommendations**

* Make sure you have consistency in your PC and display setup for this testing wherever possible (explained below)
* **When presenting G2G measurements please try and use the suggested names in this guide** – please do not bundle everything under the name “response time” without explanation of what you are presenting as there are many different approaches possible
* **Please state which method you are using if possible including any tolerance levels etc as it helps ensure results can be compared between reviewers**
* The OSRTT ‘recommended settings’ option can be left selected if you wish although it is probably useful to understand what it is capturing and what will be output in the results as explained below. This mode uses:
  + Gamma corrected response times
  + The “complete response time” (this is always captured in the data, no matter what options you select)
  + “Initial response time” (the bit before any overshoot – explained later) with a fixed RGB 5 offset tolerance level
  + “Visual response time” (including any overshoot in the figure – explained later) with a fixed RGB 5 offset tolerance level
  + Gamma corrected overshoot in RGB values
* If you instead want to instead use other settings away from the ‘recommended settings’ mode you can select “advanced settings” in the menu, but please take note of the guide below when selecting options, and when presenting data to your audience.

**PC and Display Setup**

1. Make sure the display you are measuring has been warmed up for a suitable amount of time, at least 30 minutes. Ideally you would run some dynamic content like a game, video or perhaps leave the full screen version of [this TestUFO.com](https://www.testufo.com/photo#photo=alien-invasion.png) test running if you want. The OSRTT software will warn you if it detects you might not have warmed up for long enough
2. Make sure that you have monitor features like blur reduction mode (which would strobe the backlight) or any local dimming *disabled*. These would cause variations in the light measurement and lead to errors in calculations.
3. Set your graphics card to the refresh rate you want to measure:
   1. For PC gaming usage it is best to set it to the maximum the screen will support. The software will allow you to measure the impact of VRR (Variable Refresh Rates – G-sync and FreeSync) where the frame rate drops below this, by limited the FPS settings, so you do not need to change the refresh rate setting for that.
   2. You may also want to test the screen at a fixed refresh rate for certain situations. For instance, you may want to set the graphics card refresh rate to a fixed 60Hz input, and disable G-sync / FreeSync to simulate the performance if you had a Blu-ray player or 60Hz games console connected. Likewise you could also set the refresh rate at 120Hz to capture the performance for a 120Hz input from modern consoles like the PS5 or Xbox Series X.
4. Position the device so it is measuring the centre of the screen as best you can – the actual sensor is located behind the purple light you can see on the front of the unit, so use that as a guide.
5. If the screen supports VRR (G-sync or FreeSync) then for PC gaming it is probably best to test with this enabled. Some screens show different performance with VRR turned on and off, so it’s probably best to measure with this on since most people will use this feature.

**Notes, warnings and comments:**

1. Displays that still feature PWM for backlight dimming (thankfully very rare nowadays) may create issues with the measurements since the brightness will fluctuate and this may mess up the calculations. It should be possible to still calculate response times but you may need to more closely validate or calculate the results using the ‘Graph View Template’ sheet. The software uses a moving average filter to reduce noise before processing the data. It’s adaptive based on how noisy the data is, although in some very niche cases it may fail to process, hence the need to check and validate your results.
2. On some displays where the transitions reach towards black, and the contrast ratio of the display is low, you may find the results hard to capture. You may need to exclude these from your results table if the figures look odd. You can also check the data in the ‘Graph View Template’ sheet if you want
3. Why not capture more transitions in the table? – after lots of experimentation and discussion we felt that more data was not necessarily better here. It over-complicated things for readers, added unnecessary time to testing, and created issues with measurements when transitions were very close together. We felt that the provided 30 transitions was the most appropriate sample set to use
4. Please remember these measurements and numbers can never fully replace the need for subjective assessment in games, movies and other fast motion content.

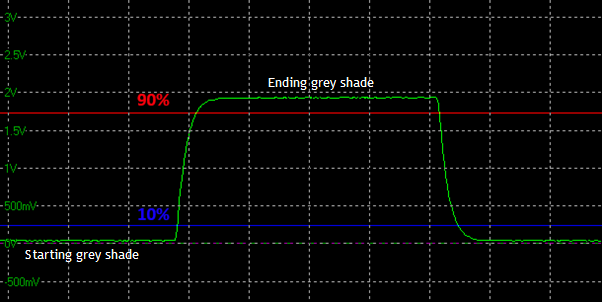
**What do we mean by display pixel “response time”? – it might not be as simple as you think**

First of all, we should consider what the term “response time” really means in the display market. It is not as simple as just saying *“it’s the time it takes for a pixel to change from one colour to another”.* That’s not actually what the term “response time” means strictly.

For over 20 years the definition of response time in the display market is actually, in simple terms: *“the time taken to switch between two colours, allowing for a small leeway of 10% either side”.* Put another way, *“It’s the time to go from close to the starting colour, to close to the end colour”.*

Actually we are measuring changes in grey shade of the pixels based on how much light they let through from the backlight (for LCD displays at least), from RGB 0 (black) to 255 (white). The colour filters are applied on top so from now on we will talk about this as changes in grey shade, which is where the term “grey to grey” (G2G / GtG) comes from in the market.

Represented on a graph measuring the change, it would look something like this:



Where the bottom green line of the graph is the darker grey shade it is changing FROM, and the top green line represents the lighter grey shade it is changing TO. The light from the display is picked up by a photosensor or similar device and converted to a voltage on the graph.

The time it takes to switch from the bottom green line (dark shade) to the top green line (light shade) would be the full transition time of the change, but actually the definition of the “response time”, as defined a long time ago and uniformly adopted by panel and display manufacturers, is the time it takes to change within a 10% tolerance of either shade. That’s represented by the 10% blue line close to the starting grey shade, and to the 90% red line which is close to the end grey shade. That’s what the term “response time” *officially* means in the display industry.

The reasons for this being adopted were:

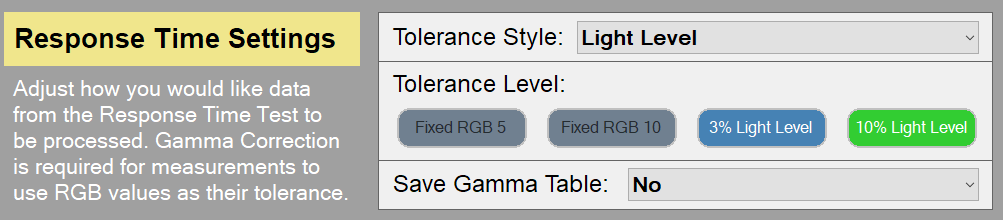
1. Because having a modest 10% tolerance level is widely adopted in various areas of electronics and measurement, so it seems to make sense to do the same here
2. It helps to some degree reflect a more realistic view of the time it takes to get “close enough” to the desired colour from a visual point of view. For instance, if you’re transitioning to black, there is a point where very dark grey is “close enough” to look visually indistinguishable and very similar. A small tolerance accounts for this leeway in what you see visually. This is a good thing for displays as after all, it’s all about what you see as a user.
3. It also helps avoid some complications with measurement methods, removing any noise or variation as you get close to certain shades, especially towards black.

**Let’s call this the “traditional response time” for reference.**

If you want to, you can measure using this method from OSRTT by

* Select the test settings menu > switch to advanced mode at the top > in response time settings section select tolerance style as “light level” > select “10% light level” option as shown below.

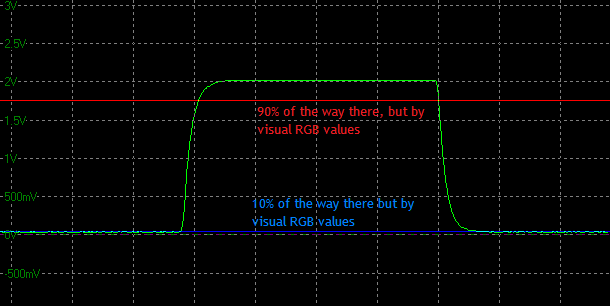
**We do not believe that this is the best way to present this measurement data any more, and would advise using the other methods discussed below. It is included for completeness and reference.**

****

**Gamma Correction provides a more accurate view of visual experience**

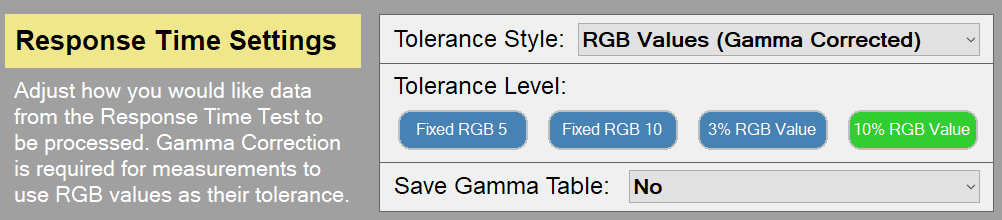
So that’s the definition of what “response time” actually means officially. In 2020 a video by [a5hun at Aperturegrille on YouTube](https://www.youtube.com/watch?v=MbZUgKpzTA0), and then later discussed at length by [Hardware Unboxed](https://www.youtube.com/watch?v=-Zmxl-Btpgk) and [TFTCentral](https://tftcentral.co.uk/articles/response_time_testing) highlighted a flaw with the traditional measurement method. This old method did not account for the impact of display gamma, and so the way the tolerance levels of 10% and 90% were used led to some issues.

You can refer back to the resource linked above for a lot more information, but a better method was suggested that would “gamma correct” the measurements and therefore provide a more realistic reflection of the performance from a visual point of view. The tolerance levels of 10 and 90% can still be kept if desired, but now the measurements are based on the actual visual RGB values instead of just the voltage that is converted from the light output. This removes the error with just referring to voltage scales on the data that were being used before in the “traditional response time method”.



This is useful to provide a more complete view of what you would see in real use and **we would recommend leaving “gamma corrected response times” enabled in the software**, unless you have some specific reason to measure using the old traditional method.

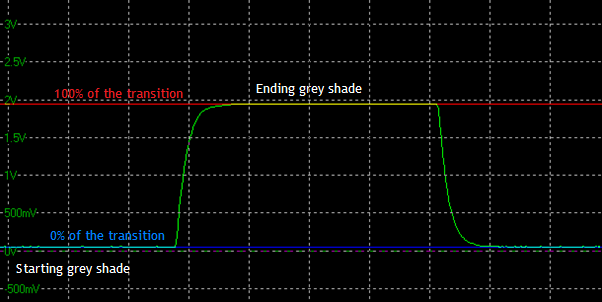
* Select the test settings menu > switch to advanced mode at the top > in response time settings section select tolerance style as “RGB values (Gamma corrected)l” > select “10% light level” option as shown below.



**From there the question becomes – what is a reasonable tolerance / offset level to use, if any, and which part of the response curve do you want to capture?**

**-----------------------------------------------------------------------------------  
Option 1 – Measure the Complete Response Curve (0 – 100%)  
Known as the “Complete Response Time”**

One option for the OSRTT tool is to consider the FULL transition time, the time it takes to fully change from one shade to another. This would be represented on the response curve graph as follows, measuring between the blue and the red lines without *any* tolerance baked in. It is the total time taken to fully change from the starting grey shade, to the target grey shade.



This may be favoured by some people as a measure of the full transition time – but we cannot call this the “response time” really as that’s not what we are measuring any more.

The recommendation is that if this data is to be used and presented, this should really be labelled as the **COMPLETE RESPONSE TIME**

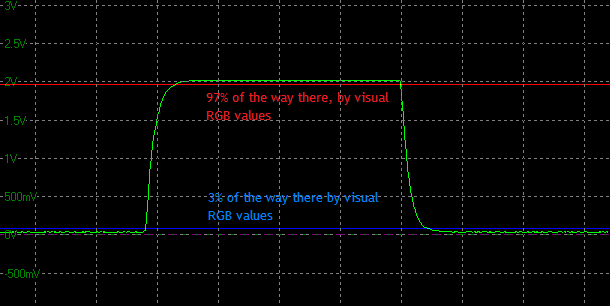
Some people may prefer this approach as a very strict measure of the total time it takes to switch from one shade fully to another. However, there are some limitations with this approach:

1. There is no consideration for when a transition has reached “close enough” to the target colour and so is visually indistinguishable. Is it fair to measure the curve to where it has to reach the final end shade if it got visually to the same thing much sooner? You will see how the response curves taper off when it gets close to the end points of the transition. The real question is how close does it need to get to visually represent the completion of the transition?
2. Some screens show a particularly long taper at the end of a transition as it finishes that last little bit. In some extreme cases if you were then measuring this “total transition time” it would extend the response time to maybe 20ms+. Is it fair to do that when it ctually reached 90% or even 95% of the way there, and looked visually the same?
3. **Note: Where overshoot is present** (discussed later) the “complete transition time” will incorporate this as well, as it measures the time it takes to get to the final RGB state AFTER the overshoot has happened. This will also add significant time to the G2G measurement. This is discussed more later, but it’s important to keep in mind for these measurements.

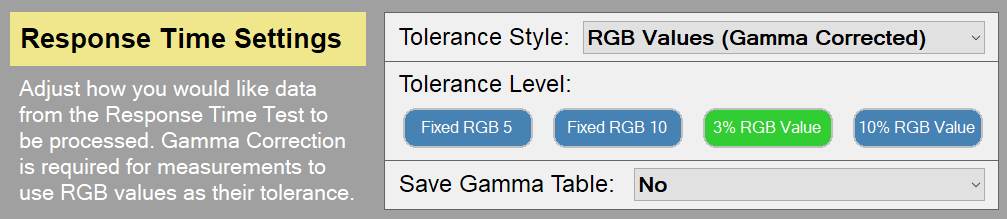
**There is perhaps merit and interest in capturing this “complete response time” for some users, but we also need to also keep in mind what we are ultimately trying to achieve here – and that is to provide a measurement that is representative of how the display appears visually to the user. As a result we do not recommend using the ‘total response time’ as it is overly strict and prone to causing unfair measurements.**

Note that if you do want to use this data, is is captured by default in the OSRTT output data CSV files as “**complete response time**”, you do not need to select that in the software, it will always be included

**-----------------------------------------------------------------------------------------------------------  
Option 2 – Using a tight tolerance level to capture nearly all the initial response time curve (3 – 97%) but excluding overshoot**At the time of writing in Mar 2022 this is the method currently used by Hardware Unboxed and discussed in [their video here](https://www.youtube.com/watch?v=-Zmxl-Btpgk&feature=youtu.be). The desire is to capture as much of the response curve as possible, but eliminating some of the problems discussed above with getting very close to the desired shade, removing the issue with the slow tapering off, and also ignoring the overshoot impact in the response time figure itself. Overshoot is captured as a separate table of data, it is not just ignored.



This can be selected within OSRTT using the following option:

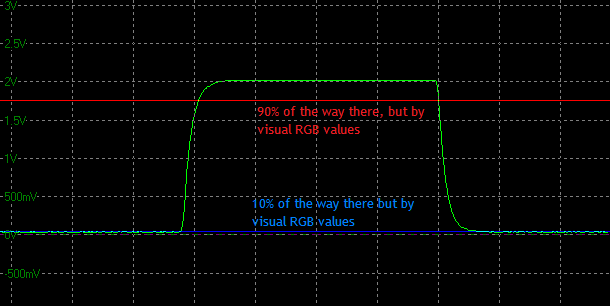


This avoids some of the issues in presenting the “complete response time” but also has some limitations of its own:

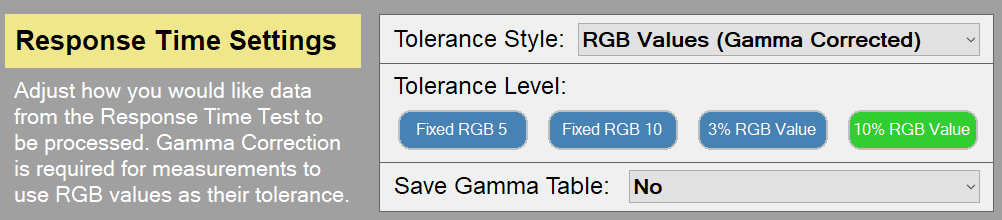
1. The % will result in the actual RGB tolerance being variable each time. For instance when measuring from 0 > 100 RGB as a transition, this would mean you’re really measuring between RGB 3 and RGB 97 values. But if you measure from 0 > 200 RGB you are measuring between RGB 6 and RGB 194 because the total range is wider. So on the one hand, it is being more strict and only allowing an RGB tolerance of 3, and in the other it’s measuring a slightly more relaxed tolerance of 6.   
     
   This doesn’t make a huge difference to the resulting measurements as really it’s trying to capture nearly the whole “total transition time” anyway, but it does cause some variance.
2. Like with the “total transition time” approach, it doesn’t necessarily consider the *visual* difference as well and whether a shade gets “close enough” sooner than the strict tolerance levels used, especially when measuring transitions close together. It may also in some cases capture portions of the slow tapering off in the grey shade, where it may be visually indistinguishable in practice.

**We do not recommend using a % tolerance level for the above reasons**

**----------------------------------------------------------------------------------------------------------  
Option 3 – Using the original tolerance levels but now with gamma correction  
(10 - 90%)**There is of course the option to stick with the original 10 – 90% tolerance levels that have been used in the market for 20+ years. It is worth including the gamma correction to more accurately capture the response times relative to what you see in RGB values. This removes some of the errors in sticking only to the voltage reading on the graphs, and allows you to account for display gamma.



The options should be selected as follows in the OSRTT software:



The limitations with this approach are:

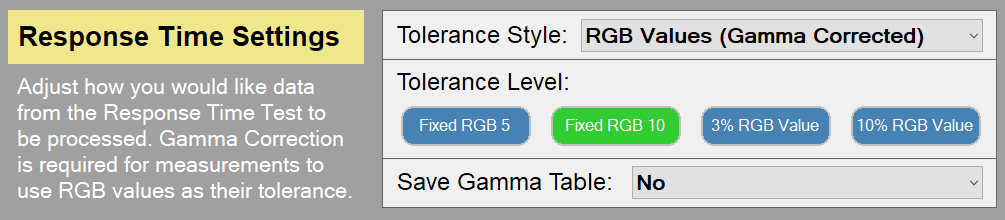
1. Like with the 3 – 97% tolerance levels discussed above, using any % will result in the actual RGB tolerance being variable each time. In this instance when measuring from 0 > 100 RGB, this would mean you’re really measuring between RGB 10 and RGB 90. But if you measure from 0 > 200 RGB you are measuring between RGB 20 and RGB 180 because the total range is wider. So on the one hand, it is being more strict and only allowing an RGB tolerance of 10, and in the other it’s measuring a more relaxed tolerance of 20.  
     
   Visually you are not being consistent in the tolerance allowed to get “close enough” to the end shade. This is far more variable when using 10/90% tolerance levels. If you are going to the lengths of correcting for gamma in the response times in order to provide a better indication of visual experience, this variation may not be ideal.

**We do not recommend using a % tolerance level for the above reason**

**-------------------------------------------------------------------------------------------------------  
Option 4 – Fixing an RGB tolerance level with gamma correction (+/- RGB 10)**This the approach favoured by TFTCentral in their measurements. Gamma correction is used to provide a more realistic measurement of what you see visually, accounting for display gamma and avoiding the pitfalls and errors from the old “traditional response time” method.

To overcome the issues with using a % discussed in the above options 2 and 3, and having variable RGB tolerance depending on the size of the range as a result, a fixed RGB tolerance level is selected. Based on lengthily visual experimentation and testing, an RGB balance of +/- 10 was deemed appropriate to get “close enough” to the desired grey shade from a visual point of view without being overly strict or too relaxed.

The [TFTCentral article](https://tftcentral.co.uk/articles/response_time_testing) explains the approach and how they reached this method in more detail. This can be selected in OSRTT using the following option:

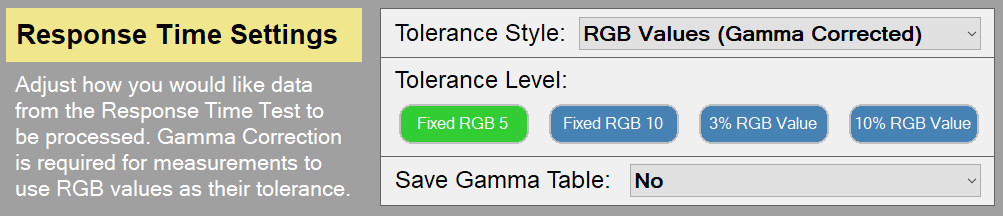


One possible concern with this approach is that for transitions that are close together (e.g. 0 – 51), there would not be much of the transition curve to measure, as you would be measuring from RGB 10 to RGB 41 in theory. This could still be considered “fair” visually since the two shades you are switching between are already very close together. Obviously if you were measuring something even closer like 0 > 20 RGB, you’d be measuring a range of 0!

You could also instead use…

**Option 5 – Fixing a tighter RGB tolerance level with gamma correction (+/- RGB 5)**Instead of using a tolerance of RGB 10, you could use a tighter RGB 5 tolerance. This captures more of the response time curve. This is the default option in the “recommended settings” although you may consider this overly strict from a visual point of view, you would have to decide whether you feel it is a better option than using RGB 10 as being “close enough” to the target grey shade..

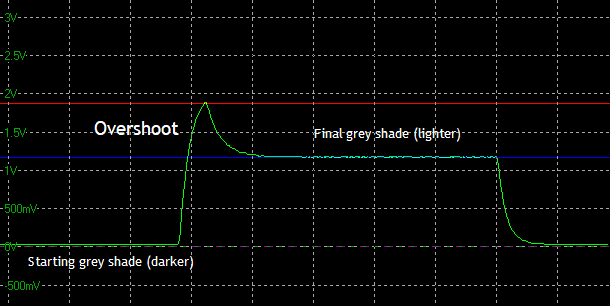
If you want to use a fixed RGB 5 tolerance you can select it in the software using the following options:



**We recommend using a fixed RGB tolerance level when measuring G2G response times, you can use either RGB 10 or RGB 5 depending on your preference. TFTCentral have settled with RGB 10 as explained in** [**their article here**](https://tftcentral.co.uk/articles/response_time_testing)**.**

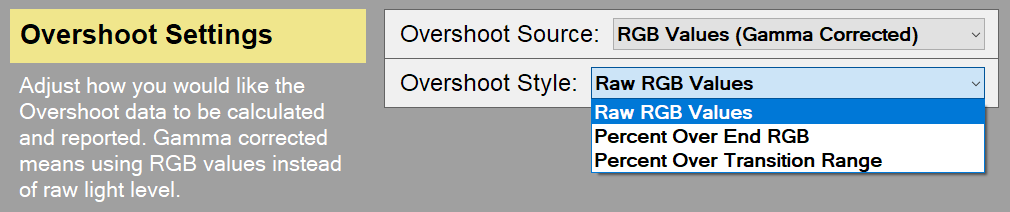
**Overshoot Calculations**

Above covers the pixel transition times and “G2G” figures. The other aspect that can be measured by OSRTT is “overshoot”. On a response time measurement graph the overshoot would be represented by the peak where the shade reaches beyond the desired grey shade, before it then drops back down to where it was supposed to be. This can also appear on the downward “fall time” as well at the end of the curve which would be “undershoot”.

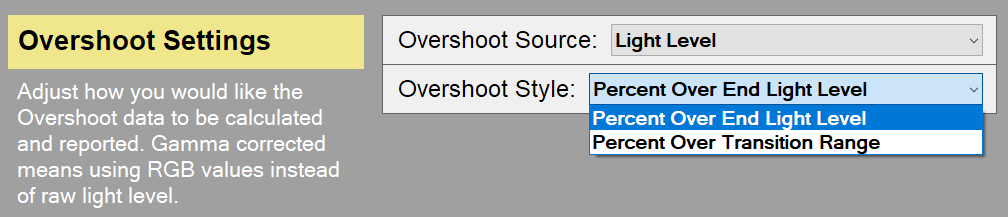


There are a few options included within the software for this:

1. **Gamma corrected (RGB values) overshoot** – **RECOMMENDED** - this will measure how many RGB values the pixel transition overshoots the desired shade. This is the recommended approach. Select RGB values (Gamma Corrected) > Raw RGB values in the settings as shown below.
2. **RGB value Percentage** – while not within the official “traditional response time” method defined in the industry 20+ years ago, the long standing approach has been to use a % to define how much the shade overshoots its target. This has always been based on light level readings, without gamma correction. Hopefully you can see why switching to RGB values and gamma correction is useful for these measurements.  
     
   If you really wanted you could do this as a % overshoot in RGB values as well, although it is not recommended. This can be selected as well within the software for completeness, but again has issues with variation. For instance if you measure 0 > 50 RGB and the shade overshoots by a fairly modest and hard to distinguish 10 RGB values, that would be a 20% overshoot and would be considered “high”. That difference of only 10 RGB values is not very visible in practice. If the transition being measured was 0 > 200 RGB instead and it overshoots by the same 10 RGB values, that would now only be 5% overshoot, even though it’s the exact same RGB difference. This is why the % method is flawed and creates issues with overshoot presentation.



The light level % can also be selected within the overshoot section if you really want to:



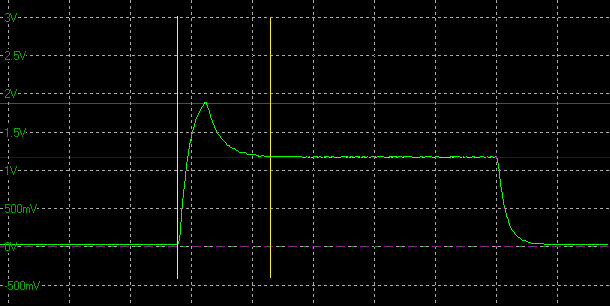
If you are experimenting with using % for overshoot there are two options available. This wil calculate the % depending on whether 1) you’re only considering how far beyond the end voltage value it goes. i.e only the target final voltage is considered along with the overshoot voltage, or 2) You’re considering the overshoot as a portion of the overall curve (end-start) setting. i.e. the range of the transition is considered in the calculation. Again these methods are not recommended for accurate overshoot calculations.

**The recommendation for a fairer reflection of overshoot is using the gamma corrected “RGB value overshoot” option.**

**A note about Measuring the G2G response times which includes overshoot**

One note that we mentioned earlier about the “Complete Response Time” (0 – 100%) is that it will incorporate the overshoot time as well. It measures the full time it takes to go from the starting grey shade, to the final grey shade AFTER any overshoot has taken place. It’s the time it takes to reach and settle on the grey shade it was trying to reach, not when it first reaches that shade before the overshoot appears. This basically captures the length of the overshoot in the response time figure as well.

For the “Complete Response Time” on the graph it would be measuring the time between the two yellow lines for example:

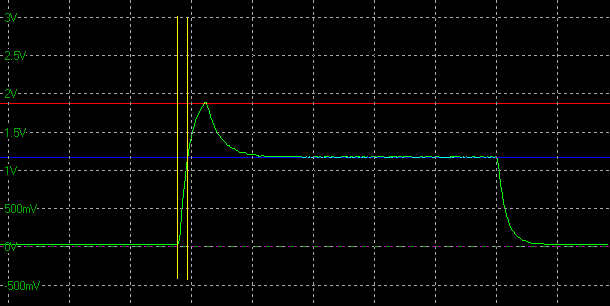


**---------------------------------------------------------------------  
Initial Response Time vs. Perceived Response Time**

In the calculation you will see two response times presented in your data – regardless of which other settings you have choosen. These are the “initial response time” and the “perceived response time”. These will both consider any overshoot differently.

**Initial Response Time**

This is the method that has been used in the market for a long time. The response time is measured on the part of the curve BEFORE the overshoot peak. It captures how long it takes to reach the desired shade, or to the desired tolerance level, but ignores the fact that the screen then overshoots this shade and then takes a while to reach back down. It would be capturing the time between the two yellow lines on the below graph:



The problem with this is that it does not then capture in that response time figure the fact that there is an overshoot, and in reality it will take quite a bit longer to reach back to the desired end grey shade. Is it fair to reward a screen with a very low response time G2G figure when the overshoot might take a long time to disappear?

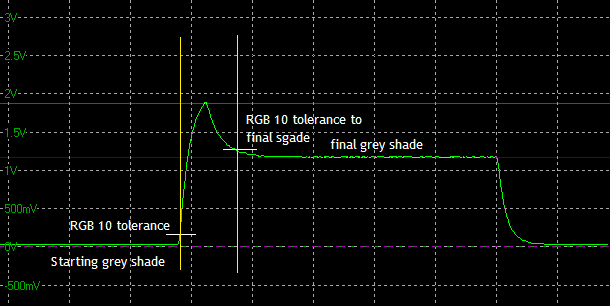
In this “initial response time” approach the overshoot is not forgotten, it is represented in a separate table showing the severity of the overshoot peak, preferably using gamma corrected RGB overshoot as discussed above.

It is still fine to present the “initial response time” in this way, but if overshoot is present this must be considered and presented alongside the G2G response times. This is the approach that has been used in the market for 20+ years so is the most widely recognised, and easiest to provide comparisons with over all the years’ worth of testing.

**Pereceived Response Time**

This is an alternative approach and is presented in the output data of any measurement. If overshoot is present it is captured in the G2G response time figure as well, representing the time it takes to settle back down to the final shade (or close to it when using a tolerance level) AFTER any overshoot has disappeared. It is the “perceived response time” as it considers the impact of the overshoot in the G2G figures. This may be a useful method to use if you wanted to show fewer data sets in your results, or were not including a normal overshoot table for instance.

On the graph it would look like this:



So keep in mind that if you are presenting the “**complete response time**” or the “**perceived response time**” data, and there is overshoot present, that will be captured in the G2G figure, and may explain why some are longer times.

If you use the “initial response time” figures, the G2G response time will be measured to where the shade is *first* reached, ignoring the fact there is then a large overshoot before it settles back to where it should be. An overshoot table should definitely be included along side the response time data if this method is used.

**Visual Response Rating  
Please note this is a work in progress and subject to change and improvements**

Especially when presenting just the “perceived response time” (which incorporates the overshoot as well as explained above), you lose a potentially important piece of information which is how quickly the display can move away from the initial colour - regardless of any overshoot. This is important as while the total transition time is incredibly key, from a visual perspective, you are likely to have a better viewing experience on a panel that is fast to reach it’s target colour, even if it misses it slightly, than a display that takes the same complete transition time but is just much slower to get there.

The “Visual Response Rating” has also been added as a potentially useful additional metric. It is a finite score rather than a direct measurement. The calculation is pretty simple, it’s: “100 – (Initial Response Time + Perceived Response Time)”. Since both metrics are using the same tolerance level, if a display doesn’t overshoot both times will be identical. This essentially rewards displays that are fast with a small amount of overshoot over displays that aren’t as fast even if they don’t overshoot at all – while still overall preferring ultra-fast, accurate monitors.

Some examples may help explain – these are theoretical rather than direct measurements:

|  |  |  |
| --- | --- | --- |
| Initial Response Time | Perceived Response Time | Visual Response Rating |
| 2.8ms | 2.8ms | 94.4 |
| 2.8ms | 5.6ms | 91.6 |
| 5.6ms | 5.6ms | 88.8 |
| 3.8ms | 9.1ms | 87.1 |
| 10.8ms | 10.8ms | 78.4 |

**Total System Latency Testing**

OSRTT also includes a handy measurement of the total system latency, in a similar way to tools like NVIDIA LDAT or NVIDIA Reflex Latency. More information about both tools is available [on TFTCentral here](https://tftcentral.co.uk/articles/nvidia_reflex).

It’s important to note that this is NOT capturing the monitor input lag on its own, which would be the lag of the display causing by the signal processing. That will be incorporated within the overall latency number somewhere, but this is basically capturing “click to photon” end to end latency. This can be influenced by many factors including graphics card, mouse, system settings, refresh rate, monitor etc. It’s a useful tool for measuring and comparing and optimising your end to end overall total system latency, but should NOT be used to represent the lag of the monitor (often referred to as “input lag”) as this cannot be isolated from within the measurement, and there are too many variables at play.