An Evaluation of an LCD Display With 240 Hz Frame Rate for Visual Psychophysics Experiments

Lin Shi: Faculty of Information Engineering and Automation,

Kunming University of Science and Technology, China

**Abstract**

Recently, a few LCD displays with 240 Hz frame rate have appeared on the market. He evaluated an LCD display with 240 Hz frame rate in terms of its temporal characteristics, progression between frames, and chromatic characteristics. The display showed accurate frame durations at millisecond level, gradual transition between adjacent frames, and acceptable chromatic characteristics.

LCD technology is constantly improving in ways that are beneficial to the needs of visual scientists. It used to be that LCD displays were not suitable for displaying objects in high-speed motion because of relatively slow updating speeds of LCD pixels. Recently, a few LCD displays with 240 Hz frame rate have appeared on the market, designed for computer game players. Here, he evaluates an LCD display (ROG PG258Q, ASUS) with 240 Hz frame rate in terms of its temporal characteristics, progression between frames, and chromatic characteristics. The display showed accurate frame durations at millisecond level and a gradual transition between adjacent frames, with no evidence of ghosting from one frame to the next. The full response spectra for the three-color channels are also presented.

Stimuli were controlled by web pages using HTML, JavaScript, and WebGL technology on a PC (Intel Core i7-6700 CPU, NVidia GeForce GTX 1060 GPU). The LCD display was connected to the graphics card through a DisplayPort cable of a PC running 64-bit Microsoft Windows 10 Home Edition (Chinese version). The web browser was Firefox (version number 53.0, 64-bit). OpenGL shader programs were used to make sure the stimuli were generated in real-time on the GPU and to make sure the stimuli were displayed reliably at the 240 Hz frame rate. The driver of the LCD display was set to a 1920x1080 resolution, 240 Hz frame rate, 32-bit color mode, and G-Sync was turned on (G-Sync is a sync technology developed by Nvidia aimed to eliminate screen tearing, and more details are available on the Nvidia web site).

**Temporal Characteristics:**

Temporal characteristics were measured by a light meter (LM03, Cambridge Research Systems Ltd., which is a demonstration product, provided freely at Asia Pacific Conference on Vision 2014) and a high-speed video camera (RX100-V, SONY). The LM03 light meter was designed for high temporal resolution but not necessarily accurate measurement of intensity. Precise measurement of output intensities to measure the gamma properties, for instance, is not the aim of this review. Temporal intensity variations of red, green, and blue channels and frame switching were measured as described later.

A sequence of frames was presented, with red, green, and blue separately increasing (in 240 steps) from 0 to 1 and then decreasing again, with a black frame (r ¼ 0, g ¼ 0, b ¼ 0) inserted between two continuous frames in order to separate continuous change clearly. The sample interval of the light meter was set as 500 microseconds and the monitor frame rate was set to 240 Hz. A subset of the data from the green channel measurements can be seen in Figure 1, which shows (a) the temporal intensity variation between the minimum value and the peak value was regular, sharp, and rapid; transitions from dark-to-light were slower than transitions from light-to-dark; and distance between adjacent peaks were in the range 8 to 8.5 ms which was consistent with frame duration 8.3 ms (two frames in 240 Hz frame rate). The variability may derive from aliasing between the sampling frequency (500 Hz) and the display frequency (240 Hz) rather than any actual imprecision in the frame rate.

Figure 1. Temporal intensity variations. Pane A shows a section of temporal intensity measurements while frames alternated green and black frame, with a decreasing intensity. Measurements of red and blue frames were similar with those of green frames. The x-axis shows the sample time and the y-axis shows the intensity measured by the light meter in arbitrary units. Pane B shows the histogram of inter-peak interval which indicated most intervals between adjacent peaks were 8.0 to 8.5 ms.

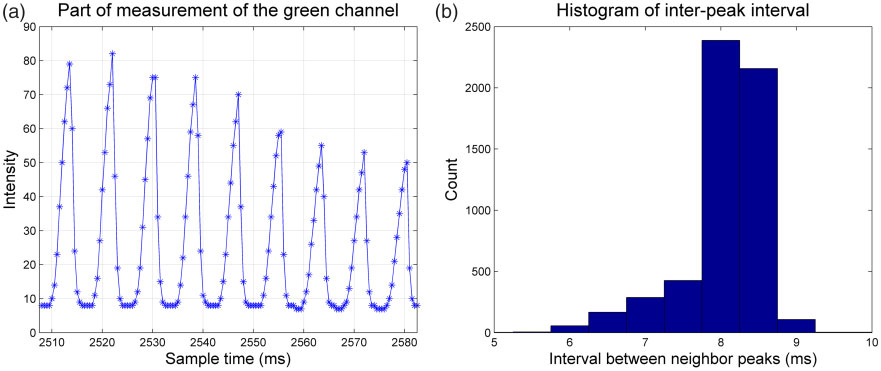


Figure 1. Temporal intensity variations

**Frame Switching Observed by a High-Speed Camera**

Animated numbers were displayed on a constant changing background. The number increased one at each frame. The frame rate was set as 240 Hz. The frame sequence was recorded by the high-speed video camera with a sample rate 1000 Hz. Figure 2 shows nine continuous frames captured by the camera, representing three display frames on the monitor (labeled 680, 681, and 682). Display Frame 681 corresponded to camera Frames 2, 3, 4, and 5 and display Frame 682 corresponded to camera Frames 6, 7, 8, and 9. Camera Frames 2, 3, 4, 6, 7, and 8 showed the gradual changes between two display frames as the LCD pixels switched. In Figure 3, we show the results of switching a larger pattern on the screen, with two adjacent frames showing a hexagonal checkerboard (camera Frame 1) and square checkerboard (camera Frame 6). Fusion between the hexagon and the square patterns was observed in white hexagons and white squares. Intermediate camera frames show the gradual transition between the images.

Figure 2. Continuous frames captured by the camera (sampling at 1 kHz) with the monitor updating at 240 Hz. Each displayed frame showed the frame number (680–682 here). Camera frames (numbered 1–9 here) showed the smooth transition from one display frame to the next, taking four frames to change, as expected. In camera Frames 1, 5, and 9, you can see that the frame has completely switched, with no evidence of ghosting from the previous frame.

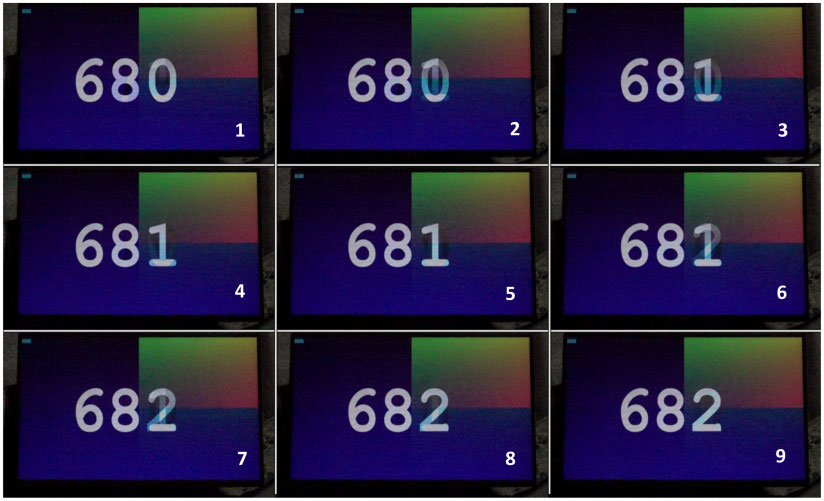
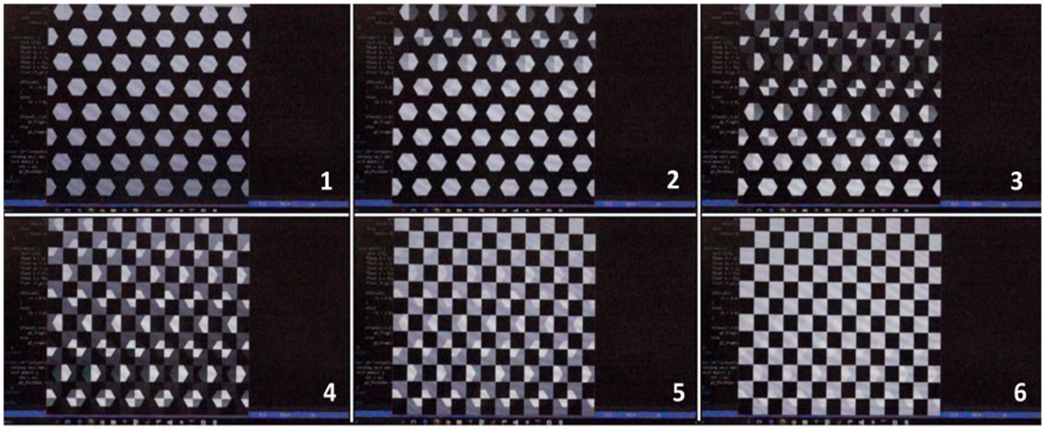


Figure 2. Continuous frames captured by the camera

Figure 3. Frame switching between hexagons and checkerboard. From left to right and from top to down, there were six continuous camera frames (1–6). Frame 1 corresponded with a hexagon display frame and frames from 2 to 6 show the transition to a checkerboard display frame. There was fusion between the hexagons and the squares in frames from 2 to 5.



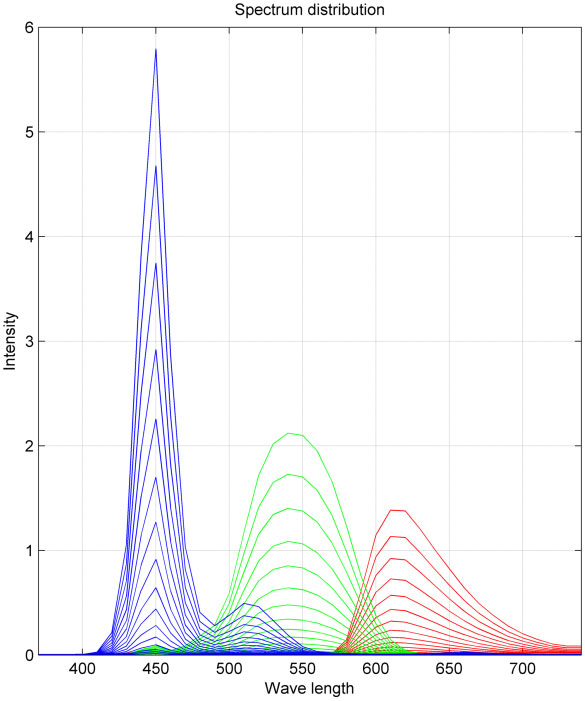


Figure 4. Spectrum of red, green, and blue channels

Figure 4. Spectrum of red, green, and blue channels. The r, g, and b values were varied from 0 to 1 in 16 steps separately and are shown as increasing lines in their corresponding color.

This 240 Hz LCD display has the following characteristics:

a) accurate frame durations at millisecond level;

b) gradual transition between adjacent frames;

c) acceptable chromatic characteristics.

**The moving target of visualization software for an increasingly complex world**

**Visualization** has evolved into a mature scientific field and it has also become widely accepted as a standard approach in diverse fields, including physics, life sciences, and business intelligence. However, despite its successful development, there are still many open research questions that require customized implementations in order to explore and establish concepts, and to perform experiments and take measurements. Many methods and tools have been developed and published but most are stand-alone prototypes and have not reached a mature state that can be used in a reliable manner by collaborating domain scientists or a wider audience.

The need to incorporate external technologies has become more urgent in recent years because the deployment environment for visualization software has become increasingly complex. In the past, visualization development frequently involved producing a stand-alone binary that could access data on a local file system and display it to a user’s monitor. The current requirements for visualization software are more challenging, where they may include delivery on the web as a service, accessing data in heterogeneous remote storage systems such as the cloud, working with data sets that exceed the RAM or even HDD capacity of a desktop computer.

Challenges. Incorporating external technology presents many challenges. In particular, we focus on the life span, future-proofing, interoperability, and costs. These challenges have been well understood by the software engineering community for several decades, but we think it necessary to reiterate and discuss the aspects that are most relevant for visualization software.

**Life span** is an important consideration for both the visualization software and the external technology that may be incorporated. The life span can vary greatly for visualization software: many projects might last for only a few years, whereas others continue for much longer. For example:

VMD (Visual Molecular Dynamics) was developed in the mid-1990s and it is still being extended.

VTK (The Visualization Toolkit) was initiated in the mid-1990s and two popular end-user applications were introduced in the early 2000s

These products are still undergoing active development. Similarly, external technology may have different life spans and various releases. OpenGL has been used extensively by our community, but it may be supplanted by Vulkan or Metal in the near future. In addition, many visualization software packages currently utilize Python V2.7, but this version will not be maintained after January 1, 2020. The basic concerns for visualization software in terms of the life span include:

1. incorporating an external technology that ends support,
2. making assumptions about external technology that might make change difficult in the future
3. being forced to allocate developer time updating to new versions of libraries.

**Interoperability** is another important challenge. In particular, a significant issue is the data formats that are often introduced by domain scientists without considering interoperability or visualization-friendly properties, such as their suitability for stream processing or parallelization. Another significant issue is the programming language employed, where some languages facilitate interoperability, such as Python and C, whereas others do not. Choosing a language naturally enables interoperability with some technologies, but makes adoption more difficult for others.

**Conclusion**. Making visualization software more usable would have benefits for both visualization and domain scientists. For domain scientists, the improved outcomes may include solving problems more rapidly (time savings) to discover more in the same period of time (increased knowledge). There are also additional benefits for visualization scientists. In particular, wider user adoption will lead to multiple positive outcomes, including increased funding, an increased reputation in the community, and an increased competitive advantage (for corporations). Moreover, making software easier to use requires less user support, which saves time and effort for visualization scientists. A sufficiently large user community can start to support itself (with Blogs, mailing lists, Wikis, etc.), which can save further time for visualization scientists. More usable software increases the likelihood of commercialization by a company or by industrious students who might attempt to bring their research to the marketplace. Finally, thorough evaluation of software by actual users is significantly simpler if the software offers a reasonable UX (**User Experience**).

Online resources:

1. An Evaluation of an LCD Display With 240 Hz Frame Rate for Visual Psychophysics Experiments: <https://journals.sagepub.com/doi/full/10.1177/2041669517736788>
2. The moving target of visualization software for an increasingly complex world:

<https://www.sciencedirect.com/science/article/pii/S0097849320300078>