

# VESA

**FPDM  
UPDATE  
May 19, 2005**

**VIDEO ELECTRONICS STANDARDS ASSOCIATION  
FPDM TASK GROUP**

---

## ***FLAT PANEL DISPLAY MEASUREMENTS STANDARD***

***Version 2.0***

---

### ***FPDM UPDATE FILE FPDMUPDT.PDF***

**ERROR & MODIFICATIONS LIST:** This is an ongoing listing of errors that have been caught to date and any available modifications or additions to either the document or to the images and patterns.

**ADDITIONS:** In anticipation of FPDM3, we wanted to get certain measurements and explanation into the hands of our users before FPDM3 will be issued. There will be many corrections and additions in FPDM3, but there was an industry need to get these particular ones out to you as soon as we could.

#### **NEW GLOSSARY ENTRIES:**

**saccade** – Saccades are the jerking motion of the eye as it follows some moving objects. Micro-saccades are tiny motions of the eye that maintains its imaging properties. Pronounced: sack-cod'.

**hold-type displays** – Display technologies in which the pixels when activated maintain their level (ideally, indefinitely) until readdressed to change to a different state. Many LCDs employ this kind of technology.

**impulse-type displays** – Display technologies in which the pixels are activated by a short pulse (or pulses) and return their rest state after the pulse is applied. Generally, the on-time of the pixels is short compared to the refresh period of the display. Many CRT displays employ this technology.

**interested parties** – We use the term “interested parties” throughout this document to refer to the display manufacturer, the original-equipment manufacturer (OEM), the display implementer, and the technical people who are involved in writing and measuring the specifications for a display purchase or implementation. It does not refer to others who were never involved in establishing the specifications for the display. Thus, FPDM measurements that are modified to accommodate an interested party would be a process that involves the manufacturer and implementer before the display is commercialized. The term “interested parties” would, therefore, not include a consumer who wants to subject the display to modified FPDM measurements in order to sue a supplier for a de-

iciency that he has detected. However, this does not preclude the use of the FPDM to determine if a display satisfies its claimed specifications if those specifications are based upon FPDM measurements, no matter who is involved.

**int( $x$ )** – Integer part of  $x$ . If  $x = 3.8$ , then  $\text{int}(x) = 3$ .

**frame, frame rate** – The frame rate is the frequency in Hz at which video information can be changed. If the display employs interlacing then this rate is called the **field rate** and several (normally two) fields are spoken of as composing one frame. Thus for some interlace technologies where two fields at 60 Hz create one frame, the field rate is 60 Hz where the frame rate is 30 Hz. This frame rate is not any sub-frame rate that may be at a higher frequency in order to make the display perform correctly or better. Frame rate or field rate refers to the rate at which information can be presented to the viewer. It is often between 59 Hz and 96 Hz. Some displays that have a frame rate of 60 Hz may run the display at 120 Hz in order to reverse the polarity on the pixel, but the information can only be changed at the 60 Hz frame rate. Some color sequential displays operate at 180 Hz, but the information is changed at the frame-rate of 60 Hz.

**native pixel array, native pixel format, native resolution (avoid)** – The largest pixel array available to present information on a display. The term generally refers to using all the pixels to present information without scaling the image. It is the highest resolution that the display can offer where each pixel can display the full range of colors. “Resolution” refers to the finest detail that the optical device (or eye) can see and should not be used in referring to pixel arrays. However, the term “resolution” used to describe the format is so ingrained in the display industry that we include it here for reference only. We would prefer that “pixel array” or “pixel format” be used instead. (See “pixel array” in FPDM2.) It is an optimal or preferred format for a non-pixelated display (e.g. CRTs).

**pixel array** – The array of pixels, usually rectangular, used to present information. Often people call this the display resolution.

**refresh, refresh rate, refresh period** – The refresh rate is the same as frame (or field) rate. It is the frequency at which information can be changed. Many technologies refresh the information on the screen at the frame rate. The refresh period is the inverse of the frame (or field) rate.

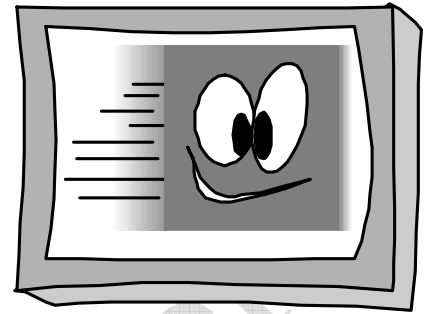


## 309 MOTION ARTIFACTS

The quantification of motion artifacts becomes important for motion video applications. A variety of motion artifacts exist, and we anticipate that this section will expand in time as more methods are developed to quantify the ability of displays to render moving objects. Motion artifacts are image distortions that may result from any video content of the display when that content goes into motion. They may be convoluted, complex, and have characteristics that may be manifested in a number of ways. In order to evaluate motion artifacts, their complexity must be reduced to a single motion condition in order to manifest the distortion in a controlled manner. For example, a block of one color moving against a background of another color could be a reduction of many possible moving conditions within a scene.

Some of these artifacts arise from the characteristics of the display device (its electronics combined with any inherent characteristics), and some arise from a combination of the human vision system with the display characteristics. There are a number of motion artifacts, some are smooth like blur and others are jerky like judder, and some have yet to be identified. For example, here is a list of some motion artifacts that have already been identified: motion blur, false contour generation, judder, dynamic chromatic (and gray-scale) aberrations, high-spatial-content detail loss, color break-up, and color smearing. More motion artifacts will be identified as research continues.

In keeping with the philosophy of this document (see FPDM2 p. 3), it is the intention of the originators of the FPDM to make the measurement methods as accessible to as many types of instrumentation as possible. This avoids unfairly promoting one type of instrumentation over another, unless there are sound metrological reasons for doing so. In this section, especially, a wide variety of instrumentation can be employed to obtain motion video measurements. We will always attempt to simplify the measurement as much as is reasonable in order for it to be accessible to as many types of measurement instrumentation as possible. However, a measurement that is simple may take much more time than using specialized instrumentation to make the measurement. Whatever method is used to obtain the final results to be reported must provide results in agreement with the measurement methods outlined herein. The intent is to have a procedure that will give equivalent test results that are independent of the instrumentation used for the test.



### A. SECTION OUTLINE

Here are our current selections:

309-1 Moving-Edge Blur

The following are place holders for FPDM3:

309-2 Box Motion Grayscale Blur

309-3 Moving-Line Contrast Degradation & Spreading

309-4 Wireframe Flickering

309-5 Dynamic Contrast of Moving Patterns

309-6 Motion-Induced Grayscale Aberrations

309-7 Motion-Induced Chromatic Aberrations

309-8 Dynamic False Contour Generation

309-9 Geometric Distortion of Moving Patterns

309-10 Color Breakup

Related to these sections are some tutorial considerations in the Discussion Section in the Appendix (A200). These sections will assist in calculating the gray levels and shades needed as well as understanding judder and blur from moving patterns. Here are the pertinent sections:

A229 Perceptively Equal Gray-Shade Intervals

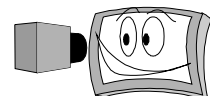
A230 Blur, Judder, & Smooth-Pursuit Eye Tracking

Also in the appendix additions to the A100 Metrology Section include:

A111 Array-Camera Considerations

Anticipated sections in the appendix associated with motion artifacts are:

A111 Array Camera Considerations — This needs to be expanded.





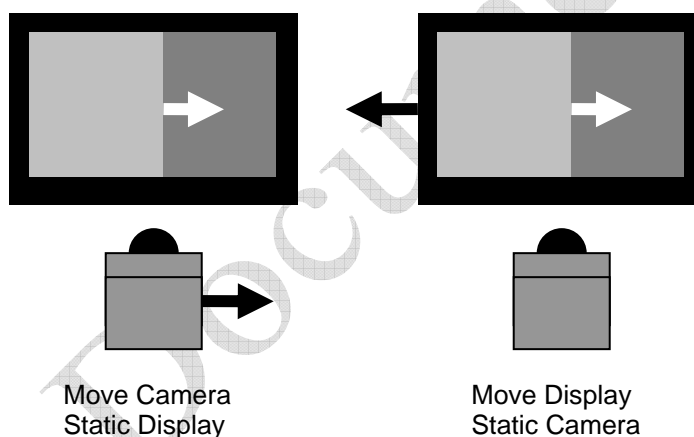
A116 Pursuit Camera Considerations  
 A117 High-Speed Pursuit Camera Simulation  
 A231 Moving Mirror Pursuit Camera  
 A232 Slit-Array Pursuit Camera

## B. INSTRUMENTATION

There are a number of comments that need to be made that can be carried throughout this Motion Artifacts Section. These can be types of instrumentation to be used, methods for evaluating the measurement results, and explanations that we prefer not to repeat in each measurement or that are common throughout this section. Unless stated otherwise, any of these instruments can be used for any of the measurement methods in this Motion Artifacts Section. Note: When using certain types of instrumentation (e.g., pursuit cameras) the obtained data may require filtration in order to provide an accurate representation of what the eye will see with some types of displays (e.g., impulse type of displays).

### 1. PURSUIT CAMERA SIMULATION OF SMOOTH-EYE PURSUIT

Ideally, the measurement system that is employed to characterize artifacts associated with moving patterns would closely mimic how the eye perceives them. Because of the complicated way the eye works, it can be very difficult to replicate such perception with machinery because of the eye's saccades and micro-saccades. However, it can be much easier to achieve smooth-pursuit eye tracking, where it is assumed that the eye smoothly tracks a moving object without saccades. In Fig. 1 we illustrate two pursuit camera systems to track moving objects—in this case a moving edge, for example. The edge moves from left to right and either the camera follows the edge or the display is moved opposite to the edge. There are other systems that can be used in a pursuit fashion to replicate or simulate these implementations, e.g., rotating cameras and rotating mirrors or optics, etc. Generally speaking, a triggering signal is used or created to synchronize the camera's view with the moving edge.



*Fig. 1. Two implementations of pursuit camera systems following a moving edge.*

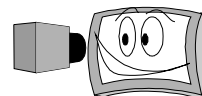
In the methods we describe, we attempt to use the simplest apparatus that is able to capture the necessary information to simulate smooth-pursuit eye tracking. In such apparatus we may employ photopic photodiodes or photopic photomultiplier tubes (PMTs). Usually, using the simplest apparatus requires least amount of expense for equipment but the greatest amount of time to acquire the data—that is certainly the case for many of these motion-artifact measurements. Pursuit camera systems require much less time but are more costly. A number of pursuit camera systems are available that will perform these measurements with ease. By describing simple measurements, we are making no attempt to discourage the use of pursuit-camera systems in any way.

### 2. FIXED OPTICAL DETECTORS

In a number of instances it may be possible to use fixed detectors to measure the characteristics of moving patterns without employing a pursuit camera.

### 3. HIGH-SPEED CAMERAS

High-speed cameras can be employed to simulate a pursuit camera provided the frames are properly overlapped, are sufficiently short, and sufficiently fill the frame period.





#### 4. SOFTWARE IMPLEMENTATION FOR QUANTITATIVE VISUAL ANALYSIS

There may be software implementation of these motion-artifact measurements that can provide numerical results that agree with the measurement apparatus.

### C. ANALYSIS OF MEASUREMENT RESULTS

#### 1. TRANSITIONS TIME QUANTIFICATION — 10 % – 90 %

One problem in measuring response times is the determination of threshold levels to characterize the transition. Whenever we have a temporal transition between two levels, a common standard engineering practice is to determine the time interval between two points in that transition. Typically the 10 % to 90 % points are used to characterize that time interval. We continue with that practice in these sections to be consistent with our previous measurement of non-moving patterns in 305-1 Response Time. It must be pointed out, however, that these 10 % – 90 % demarcations may have little to do with how the eye perceives the shades involved. Generally speaking for moving edges, finite response times that are significantly long to be perceived by the eye are seen as blurred edges.

#### 2. PARAMETERIZATION OF MOTION ARTIFACTS

The study of motion artifacts is a field that is still in flux. There can be many ways to parameterize any motion artifact. Which parameter or combinations of parameters that best quantifies the eye's perception of the artifact is not yet known. For example, associated with a moving object that is intended to have sharp edges, there are a number of parameters that can be used to characterize the degradation of that object while being viewed in motion. It has not yet been determined which parameters are the best to characterize the distortions. Thus, we will require a rather complete characterization of such motion artifacts until more research is done to isolate the most relevant parameters. For example, consider a moving block. If there is motion blur, we can measure the blur of the leading edge or the trailing edge. They can be reported separately, combined, or handled in some way to be determined in the future. As another example, consider line-spreading distortions arising from moving lines. Distortions that can result from this on some displays can change the luminance, change the color, and the line can spread in width.

### D. DIAGNOSTICS AND VALIDATION

Diagnostics are under investigation. The added complication of having to make measurements on moving patterns requires that attention be paid to how accurately the motion is being followed should that be the method that is employed as with a pursuit camera. Here we begin listing some of the diagnostics that might be used.

#### 1. CALIBRATED BLUR

In order to provide a means of calibrating a blur-measurement device, a transparent film or printed card could have a calibrated density gradient over a certain distance that is well-known. Moving that blur representation in front of the blur-measurement device can be used to diagnose the instrumentation by comparing the moving result with the static image.

#### 2. CALIBRATED LINE WIDTH

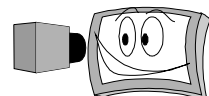
As with the calibrated blur, a calibrated blurred line (known width and level) could be similarly moved in front of the instrumentation.

#### 3. SPEED VERIFICATION

In many cases speed measurement capabilities of the instrumentation needs to be verified. One way to accomplish this would be to have a linear positioner move an object at a known rate to see if the camera instrumentation properly follows the object.

#### 4. VIDEO GENERATION VALIDATION

Whatever video generation system is providing the signal, we must be sure that the artifacts that we are seeing arise from the display and not the video generator. Such generators must be tested to have a smooth motion capability with no judder, breakup, tearing, or other temporal instabilities producing non-smooth motion. In the case where the video generator is included within the display system, such as a laptop computer, then there may be no way to determine if it is the display or the generator or the system is producing the artifact, other than comparisons of identical systems.



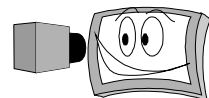


In general, the display should be operated in its native resolution or native pixel array; if that is not the case it must be agreed upon by all interested parties or will be specified clearly in any measurement procedure. If the native resolution is not used, then it is possible that the scaling engines can produce artifacts themselves.

OPTICAL

OPTICAL

Update Document

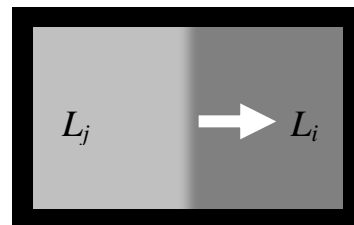




## 309-1 MOVING-EDGE BLUR

### (3.9.1 — Moving Edge Response, Moving Edge Response Time, Blur Edge Time, Motion Picture Response Time [in some literature], Blur Edge Width)

**DESCRIPTION:** We measure a blur width and response time that characterizes the blur that the eye sees when following an edge moving across the screen from left to right. Smooth-pursuit eye tracking is assumed to follow an edge of one luminance moving horizontally into another luminance in pixel increments (or jumps)  $\delta n$  for each refresh period  $\delta t$  (this is the information refresh rate and not a sub-refresh rate to make the display operate properly). The scroll-speed (or just speed) of the motion is  $u = \delta n / \delta t$  (units: px/s). A minimum of seven gray shades including black and white are used where the lightness intervals appear the same to the eye. However, more levels can be used if agreed upon by all interested parties. This measurement is particularly useful for display types that hold the image information for the duration of the refresh period. NOTE: this is a gray-scale measurement only.



**Fig. 1.** Moving edge blur between two levels.

**SETUP: Level Determination:** Given a knowledge of full-screen black  $L_K$  and full-screen white  $L_W$  (see 302-1 and 302-2 for such measurements), determine the five (5) intermediate gray levels required based upon equal lightness values (see A229 Perceptively Equal Gray-Shade Intervals for the method of calculation). This will give a total of seven (7) levels;  $i, j = 1, 2, 3, \dots, 7$  including white and black. When measuring the gray shades  $L_i$ , there can be noise in the determination of the luminance values, particularly when digitized traces are employed as shown in Fig. 2. For such cases, let  $\sigma_i$  be the standard deviation of the measurement of  $L_i$ . The standard deviation would be measured in the steady-state regions of the transition; that is, the standard deviation would not be measured in the vicinity of the transition if the levels are not flat.

**Pixel Increment (Jump) Region Determination:** The size  $\delta n$  (an integer in units of pixels [px]) of the pixel-increment (jump) region often ranges from 8 px to 16 px. It may be estimated by having the edge move across the screen in from  $\Delta t = 2$  s for common 4:3 aspect ratio screens up to  $\Delta t = 4$  s for wide screens with aspect ratios such as 16:9 or 16:10. Given the horizontal number of pixels  $N_H$ , the jump in pixels is

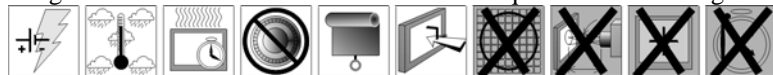
$$\delta n = \text{int}[N_H / (f \Delta t)]. \quad (1)$$

As an example, if we had a screen with  $N_H = 1024$  px with a refresh rate of  $f = 60$  Hz where we selected  $\Delta t = 2$  s, then  $\delta n = 17$  px. However, from experience we recommend that you start by using  $\delta n = 10$  px. Smaller screens, such as used in many hand-held devices, may require a smaller  $\delta n$ .

NOTE: (1) If there is any dependence of the result upon the size of the jump region, then several sizes should be used. (2) The frame rate  $f$  is the rate at which video information can be changed. It is not any sub-frame rate used to make the display function properly.

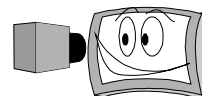
**Moving Pattern Generation:** Arrange to move an edge of luminance  $L_j$  from the far left of the screen on a background of  $L_i$ , where  $i \neq j$ , so that the edge of moves a discrete number of pixels (as determined above, e.g.,  $\delta n = 25$  px) to the right for each refresh of the screen until the entire screen is of shade  $L_j$ . Note that if you are using a computer to generate the moving pattern, be sure that the video generator in the computer is sufficiently fast to produce the moving pattern. If it is not, you will observe a jerkiness to the moving edge. If you must use such a generator, try to avoid measuring jump regions where any such jerkiness is observed.

**Standard Setup Conditions:** See Section 301 for any standard setup details. We don't confine our measurement to 500 px or more; the angular aperture and angular field of view is not required to be limited; only approximately center of screen is required; and because this is a temporal measurement simulating smooth-pursuit eye tracking, the integration time of the measurement must be equivalent to an integer multiple of the refresh period.



**PROCEDURE:** Because there are a number of ways that this measurement can be made, we describe the concept of the measurement and the results to be obtained. It is up to the implementers of any measurement apparatus to assure compliance with the intended result.

For any column of pixels, there is no reason to expect all the pixels in that column to activate at exactly the same time. However we will assume that they activate the same way once triggered to change. Thus, for example, a





scanning electrical activation may move from the top of the screen to the bottom, but each pixel in a column will exhibit the same response characteristic only at different times. If a narrow horizontal band of pixels is used in order to increase the amount of light, it must be determined that the luminance of the narrow band exhibits the same temporal nature as does a single row to within the uncertainty of the measurement. Given that the height of the band of pixels measured is sufficiently small so that the temporal performance of the luminance  $L_{ij}(n, t)$  is equivalent to that of a single row, we confine our attention to a jump region near the center of the screen:

$$c = \text{int} \left( \frac{N_H}{2\delta n} \right). \quad (1)$$

The luminance  $K(s)$  that the eye sees assuming smooth-pursuit eye tracking relative to its own on-screen coordinates is

$$K_{ij}(s) = \frac{1}{\delta t} \int_{c\delta t}^{(c+1)\delta t} L_{ij}([\text{int}(s + ut) + 1], t) dt, \quad (2)$$

where  $s$  is a continuous variable (non-integer) that defines the on-screen distance in units of pixels from the position of the eye-tracking point assuming smooth-eye pursuit of the edge: The pixel  $n$  is related to  $s$  by  $n = \text{int}(s + ut) + 1$ . See A230 Blur, Judder, & Smooth-Pursuit Eye Tracking for details of this calculation. The measurement of what the eye sees,  $K_{ij}(s)$ , are the data that need to be analyzed to provide the moving edge response time. The  $K_{ij}(s)$  may be obtained in a variety of ways that cannot be specified here.

**ANALYSIS: Case 1: No Overshoot or Undershoot:** For each eye-based  $i \neq j$  spatial transition  $K_{ij}(s)$  we determine the 10 % to 90 % blur-edge width  $b_{ij}$  (in units of pixels). The extended blur-edge width (in units of pixels)  $w_{ij} \geq 0$  is

$$w_{ij} = b_{ij}/0.8, \quad (1)$$

which extends the width to the 0 % to 100 % levels. The locations of the level intercepts are  $s_j$  and  $s_i$  so that  $w_{ij} = s_i - s_j$ . See Fig. 2. The total blur width  $W_{ij}$  associated with the  $i$ - $j$  transition is the combination of the leading extended blur-edge width  $w_{ij}$  and the trailing extended blur-edge width  $w_{ji}$ :

$$W_{ij} = w_{ij} + w_{ji}. \quad (2)$$

The moving-edge response time (in seconds) is the average of these widths divided by the average speed  $u$  of the edge:

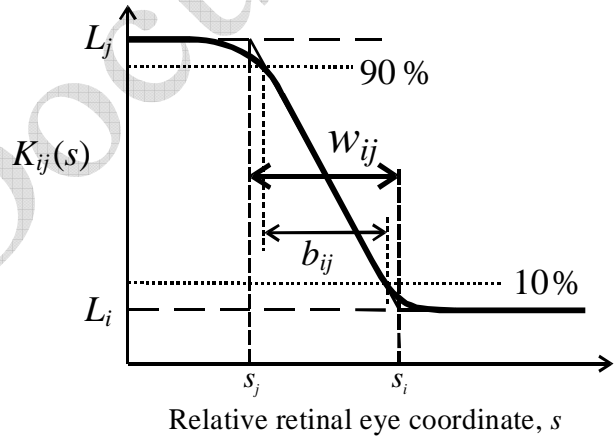
$$M = \frac{1}{uN} \sum_{i \neq j} w_{ij} \quad (3)$$

where  $N$  is the number of  $i \neq j$  transitions. For the case of seven luminance levels ( $i = 1, 2, \dots, 7; j = 1, 2, \dots, 7$ , and  $i \neq j$ ) there are  $7 \times 7 - 7 = 42$  transitions, or  $N = 42$ . For some applications, it may be useful to determine the maximum moving-edge response time:

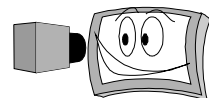
$$M_{\max} = \max(w_{ij})/u. \quad (4)$$

**Case 2: Overshoot and/or Undershoot:** In the event that an overshoot and/or an undershoot is present in  $K_{ij}$ , then additional parameters need to be measured to characterize the blur widths depending upon the sizes of the overshoot or undershoot. In all cases of overshoot and/or undershoot, the Case 1 measurement analysis will be made. There are several sub cases for each transition — see Figs. 3 and 4. Note that we consider an overshoot or undershoot is present when the overshoot or undershoot exceeds three standard deviations of the level values within a distance of an extended blur width of the transition region; that is, if any peak or valley is larger than  $3\sigma_i$  of  $L_i$  within a distance  $w_{ij}$  of  $s_j$  or  $s_i$ , then we have to subject the data to overshoot-undershoot analysis.

- Overshoot Only:** Measure the maximum luminance  $L_p$  and the distance  $p_{ij}$  (in pixels in relative retinal eye coordinates) between the peak and the lower level intercept  $s_i$ .
- Undershoot Only:** Measure the minimum luminance  $L_v$  and the distance  $p_{ij}$  (in pixels in relative retinal eye coordinates) between the upper level intercept  $s_j$  and the valley.



**Fig. 2.** Blur edge width  $b_{ij}$  and extended blur-edge width  $w_{ij}$  with level intercepts  $s_j$  and  $s_i$ .



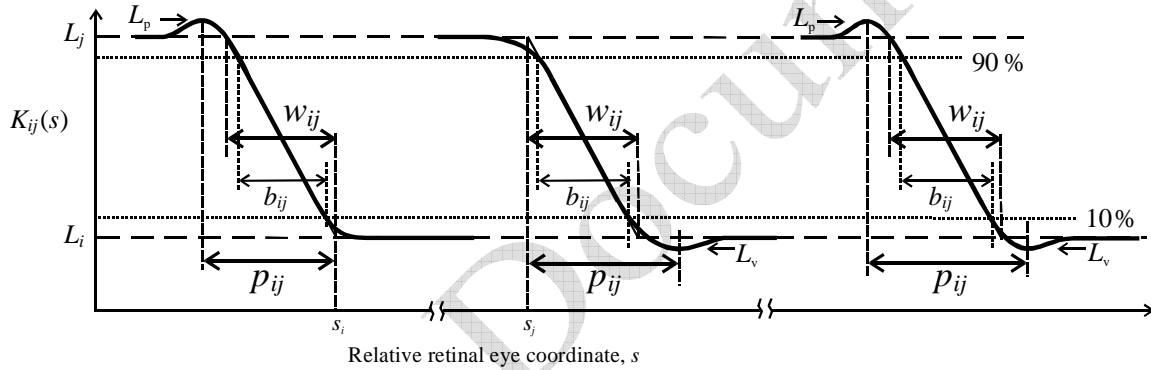


- c. **Both Overshoot and Undershoot:** Measure the maximum luminance  $L_p$  and minimum luminance  $L_v$  and the distance  $p_{ij}$  (in pixels in relative retinal eye coordinates) between the peak and the valley.
- d. **Overshoot Over 110 %:** Measure the expanded distance  $w'_{ij}$  (in pixels in relative retinal eye coordinates) between the 110 % intersection outside the transition and the lower-level intercept  $s_i$ .
- e. **Undershoot Below -10 %:** Measure the expanded distance  $w'_{ij}$  (in pixels in relative retinal eye coordinates) between the -10 % intersection outside the transition and the lower-level intercept  $s_j$ .
- f. **Overshoot Over 110% and Undershoot Below -10 %:** Measure the expanded distance  $w'_{ij}$  (in pixels in relative retinal eye coordinates) between the 110 % intersection outside the transition and the -10 % intersection outside the transition.

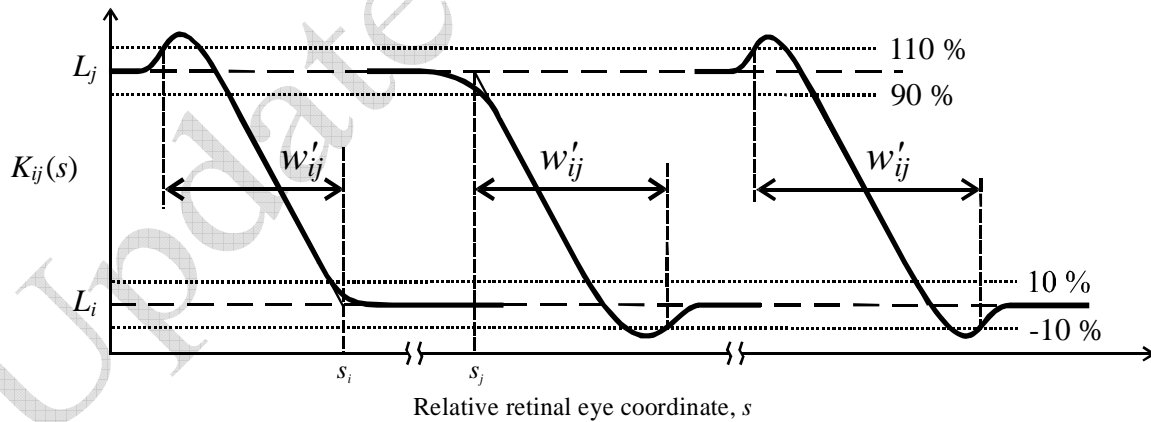
**Case 3: Scroll-Speed Dependence:** In the event that there is a scroll-speed  $u$  dependence on the resulting moving-edge response time  $M$  then measurement results for different jump-region widths  $\delta n$  must be obtained and analyzed appropriately as outlined in Case 1 and Case 2. For such cases, calculate the modified moving-edge response time  $M'$  as

$$M' = \frac{1}{N} \sum_{i \neq j} \frac{\partial w_{ij}}{\partial u}. \quad (5)$$

This amounts to plotting the extended-blur-edge width vs. the scroll speed  $u$  and extracting the slope  $\partial w_{ij}/\partial u$ .



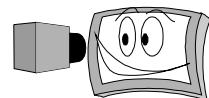
**Fig. 3.** Overshoot and undershoot showing additional measurement results  $L_p$ ,  $L_v$ , and  $p_{ij}$ .



**Fig. 4.** Overshoot greater than 110 % and undershoot lower than -10 % requiring the measurement of  $w'_{ij}$ .

**REPORTING:** Report the jump-region width  $\delta n$ , the information refresh rate  $f$ , the refresh period  $\delta t$  and the following data:

1. Report the levels  $L_i$  and their standard deviations  $\sigma_i$  if needed.
2. The extended blur-edge widths for all transitions, their average, their standard deviation, and their minimum and maximum;
3. The total extended blur-edge width  $W_{ij}$  for all transitions, their average, their standard deviation, and their minimum and maximum;



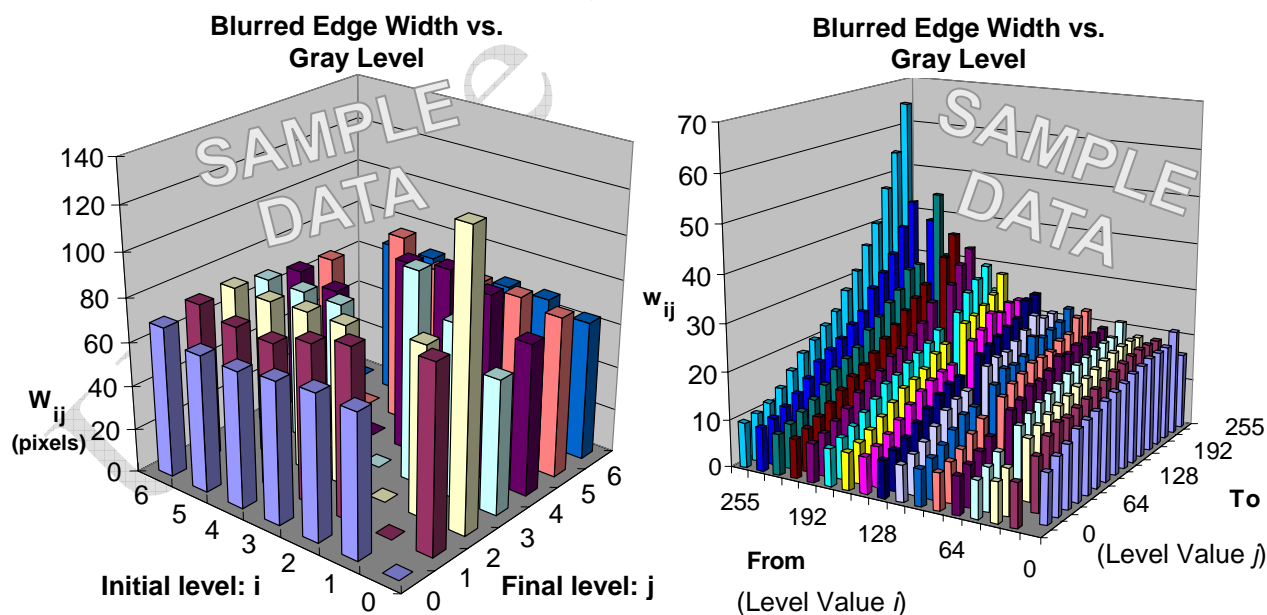


4. The moving-edge response time  $M$  in either milliseconds (ms) or seconds (s); and the maximum moving-edge response time  $M_{\max}$ .
5. If overshoots and undershoots exist in the  $K_{ij}$  then it will be necessary to report the overshoot and undershoot analysis data additionally (a–f above under Case 2).
6. If the display exhibits a scroll-speed dependence (Case 3) then report the above data for each jump-region width employed, and also report the modified moving-edge response time  $M'$ .

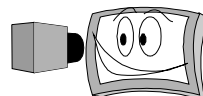
NOTE: Because sufficient research has not yet been done to determine which analysis results are most important to characterizing the motion blur, it is necessary to report all the above parameters. It is anticipated that fewer parameters will be identified in the future whereby the analysis and reporting load will be reduced.

**COMMENTS:** Please note:

1. There are a variety of apparatus and methods that can be used to provide these measurement results.
2. More luminance levels may be employed provided all interested parties agree.
3. **Why is a time result included?** Some have wondered why there is a time result included and not just distance measurements of blur. Scaling the blur width to the speed of the edge by dividing by  $u$  as in Eq. (2) is an attempt to lessen the dependence of the result upon the jump-region width  $\delta n$ . We could have specified a normalized blur width by dividing  $w_{ij}$  by the jump region width  $\delta n$  thereby obtaining a unitless result. Although this would have normalized the measurement, it would not have distinguished between displays using different video refresh rates.
4. **Blur widths and angles:** There may be reasons to report blur widths in terms of angles. Consider an optimally viewed display where 30 line-pairs-per-degree are resolvable to most people. That is equivalent to having one pixel per arc-minute. Thus, at this optimal viewing distance, a blur of width of  $w$  pixels also represents a blur of  $w$  arc-minutes.
5. **Static image blur:** Any incidental blur at a static transition edge is intended to be included in the blur measurement.
6. **Graphical reporting:** Some like to see a graphic form of reporting in addition to a tabular form. Here is an example:



**Fig. 5.** Two examples of 3D plot of data. It is not necessary to display the data in this manner, and there may be other types of plots that prove useful in addition. The left plot illustrates transitions between the required minimum of seven levels (labeled 0 through 6). The right plot illustrates transitions between 17 levels (labeled according to gray level for an eight-bit display). This measurement specifies that at least seven levels must be used, but it is possible to perform these measurements on all transitions (e.g., all 256 levels for an eight-bit grayscale).





**PLEASE NOTE:** The following are place holders for future motion-artifact measurements. Complementing these will be a set of diagnostics and tests to assure any apparatus is functioning properly. Additionally, guidelines for the employment of pursuit cameras, and other array devices will be offered.

## 309-2 Box Motion Grayscale Blur

### (3.9.2 — Horizontal Box Motion Blur)

**DESCRIPTION:** We measure the combined leading and trailing edge blur of a 100 px square box (typically) of one gray level moving from left to right across a background of a different gray level. Smooth-eye-pursuit tracking may be used, but other methods to quantify this result are anticipated. The box should be centered on the screen when the measurements of its horizontal blurred edges are made; it should also be centered vertically throughout its horizontal motion across the screen. Although some of the measurement results acquired here are also found in 309-1, it is anticipated that additional and different analysis will be included here. This measurement enables seeing both the leading and trailing edge at the same time.

## 309-3 Moving-Line Contrast Degradation & Spreading

### (3.9.3 — Moving Line Spreading)

**DESCRIPTION:** We measure the contrast and spreading of a line of one gray level moving horizontally from left to right across a background of a different gray level assuming smooth-eye-pursuit tracking of the line. We compare a static line (involving the same levels) with this moving line to determine a contrast degradation of the line relative to the background. The line width is a single pixel. (However, other line widths may be additionally employed if agreed to by all interested parties.) **NOTE:** The speed of the line must be 1 px/frame or more, preferable 5 px/frame. If the speed is slower, we migrate toward the case of Wireframe Flickering (309-4). See 309-5 Dynamic Contrast of Moving Patterns for an image-based determination of the degradation of the moving line.

## 309-4 Wireframe Flickering

### (3.9.4 — Moving Line Flicker)

**DESCRIPTION:** We measure the time-dependent flicker of a slowly moving line (generally the speed is 1 px/frame or slower, typically at a speed of 0.1 px/frame) having a minimum spacing between consecutive lines of at least 10 px. For motion artifacts for speeds of 1 px/frame or higher see 309-3 Moving-Line Contrast Degradation & Spreading.

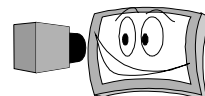
This metric arises from the need to quantify the visible flicker of wireframes and other line-based patterns as they are slowly moved across the screen of some displays. In extreme cases, an entire wireframe graphic may be seen as flashing on and off when it is slowly moved across the screen. Not only can there be flickering in luminance, but also color shifts.

## 309-5 Dynamic Contrast of Moving Patterns

### (3.9.5 — [no aliases at this time])

**DESCRIPTION:** We measure the dynamic contrast of moving patterns assuming smooth-pursuit eye tracking. Several types of patterns can be used.

The dynamic contrast of a moving image is based upon its static form. Suppose we have a static image. Generally we are dealing with only a small area of the screen. Let the relative location of the pixels associated with the static image be  $n_i$  and  $n_j$  in the  $(x, y)$  direction respectively for  $i = 1, 2, \dots, N_i$  and  $j = 1, 2, \dots, N_j$ , and let the luminance of each pixel be  $S_{ij}$  for the static image. Consider moving that pattern at a speed  $u$  (in px/s) [if  $u$  is a velocity, then there it will be defined by  $(u_x, u_y)$ ]. Assuming smooth-pursuit eye tracking where the moving image is precisely identified properly by the same relative coordinates  $(n_i, n_j)$ , let the luminance associated with each pixel in the moving image be  $M_{ij}$ . The dynamic contrast (based upon the definition of Michelson contrast) of the moving image is:





$$C_d = \frac{1}{N_i N_j} \sum_{i=1}^{N_i} \sum_{j=1}^{N_j} \left( 1 - \frac{|M_{ij} - S_{ij}|}{M_{ij} + S_{ij}} \right).$$

A number of patterns or images will be considered including a 100 px moving box (as employed in 309-2) and a single-pixel moving line (as in 309-3) of one gray level on a background of another gray level. The dynamic contrast ranges from zero to one—a perfect moving image exactly like the static image has a dynamic contrast of one. This lends itself to also expressing the dynamic contrast in percent by multiplying  $C_d$  by 100%.

## 309-6 Motion-Induced Grayscale Aberrations

### (3.9.6 — Dynamic Grayscale Aberrations)

**DESCRIPTION:** We measure non-blur motion artifacts assuming smooth-pursuit eye tracking of a moving pattern.

Blur may be thought of as a smooth transition between one level of gray of a simple pattern and another level of gray composing the background. However, there can be perturbations on this smooth transition that give rise to overshoot, undershoot, ripples, or other artifacts. This metric is based upon luminance measurements to be distinguished from smooth blurring.

## 309-7 Motion-Induced Chromatic Aberrations

### (3.9.7 — Dynamic Chromatic Aberrations)

**DESCRIPTION:** We measure non-blur colored motion artifacts assuming smooth-pursuit eye tracking of a moving pattern.

Blur for color may be thought of as a smooth transition between one color in the simple pattern and another color composing the background. However, there can be perturbations on this smooth transition that give rise to color shifts that deviate from what would be a smooth transition. This metric is based upon color measurement to be distinguished from smooth blurring of colors.

## 309-8 Dynamic False Contour Generation

### (3.9.8 — [no aliases at this time])

**DESCRIPTION:** We measure the contour distortions that may occur on boundaries or contours of images or patterns in motion. These anomalies might arise from super-frame-rate image generation (where the pixel information is updated faster than the frame rate); for example, these may be artifacts in images that are generated from sub-frame-period pulsing. This metric is to be distinguished from blur and other metrics introduced in this section.

## 309-9 Geometric Distortion of Moving Patterns

### (3.9.9 — [no aliases at this time])

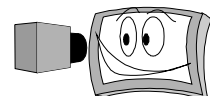
**DESCRIPTION:** We measure the distortions associated with an object in motion that geometrically differ from the object at rest besides blur and other artifacts already covered in this section.

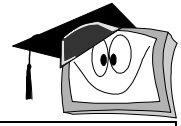
This metric is to be distinguished from blur and other metrics introduced in this section. It refers to this such as elongation of corners, indentations, flaring, visibility of new sub-geometric structures, rounding, and so forth.

## 309-10 Color Breakup

### (3.9.10 — [no aliases at this time])

**DESCRIPTION:** We measure motion artifacts associated with color generated from sequential frames, temporal dithering, such as frame-rate modulation, or other dynamics of screens that exhibit similar sequential behavior.





## A229 PERCEPTIVELY EQUAL GRAY-SHADE INTERVALS

An electronic display has a white luminance  $L_W$  and a black luminance  $L_K$ . We want to determine the luminances  $L_n$  for  $N$  perceptively equal gray shade intervals from black to white. Using the lightness metric of the CIE 1976 CIELUV and CIELAB color spaces the lightness  $L^*$  is

$$L^* = 116 \left( \frac{L}{L_W} \right)^{1/3} - 16, \quad (1)$$

$$\text{but } L^* = \left( \frac{29^3}{27} \right) \frac{L}{L_W}, \quad \text{for } \frac{L}{L_W} \leq \left( \frac{24}{116} \right)^3.$$

There is a lightness associated with the white and black screen:  $L_W^* = 100$ , and  $L_K^*$  is given by Eq. (1) with  $L = L_K$ . The lightness levels for  $N$  perceptively equal intervals above black ( $N+1$  levels in all) is

$$L_n^* = L_K^* + n \left( \frac{100 - L_K^*}{N} \right) \quad (2)$$

for  $n = 0, 1, 2, \dots, N$  giving a total of  $N+1$  levels including black ( $n = 0$ ). For example, if  $L_K = 0$  (a perfectly black screen ☺), then the lightnesses for  $N = 6$  intervals would be  $L_n^* = 0, 16.7, 33.3, 50, 66.7, 83.3, 100$ , providing seven levels.

Equation (2) provides the lightness values producing perceptually equal gray-shade intervals from black to white. The corresponding luminances of the display would be the inversion of Eq. (1) using the  $L_n^*$  values:

$$L_n = \left( \frac{L_n^* + 16}{116} \right)^3 L_W, \quad (3)$$

$$\text{but } L_n = \frac{L_n^* L_W}{(29^3/27)} \quad \text{for } \frac{L_n^*}{L_W^*} \leq \left( \frac{24}{116} \right)^3$$

For our example with a perfectly black screen, if  $L_K = 0$ , and for  $N = 6$  intervals, then the coefficients of  $L_W$  in the left side of Eq. (3) are: 0, 0.0223, 0.0769, 0.1842, 0.3619, 0.6279, 1; and if the luminance of white is  $L_W = 100 \text{ cd/m}^2$ , then the required luminances would be  $L_n = 0, 2.2, 7.7, 18.4, 36.2, 62.8, 100 \text{ cd/m}^2$  — for this example only.

The  $L_n$  are the luminances that we would need to reproduce with the screen gray shades selected as nearly as possible to have our desired perceptibly equal luminance intervals from black to white. The luminance of a screen is determined by the driving level  $V$  — the gray level — and the electro-optical transfer function (sometimes called “gamma”)  $L(V)$ . In practice, once we have the desired luminance levels  $L_n$ , we might adjust the driving levels  $V$  until we get the desired luminance dis-

played on the screen as closely as we can. To do this analytically, we would have to know the function form of  $L(V)$  and be able to invert it  $V(L)$  to obtain the desired driving levels  $V_n = V(L_n)$ . For discrete driving levels, the discrete level  $V_m$  that produces a luminance closest to  $L_n$  would be selected ( $m$  such that  $|L_m(V_m) - L_n|$  is minimum).

Because very few displays have a zero black luminance, we cannot provide a general table for all displays illustrating the levels needed for different  $N$  values. The gray levels (command levels) employed to provide equal lightness steps (perceptively equal gray-shade intervals) will depend upon the measurement of the black luminance, the white luminance, and the above analysis that depends upon the electro-optical transfer function as well. We provide an example below, but it is *only* an example. Please do not use these values. Each display can be very different and needs to be measured separately to determine the correct gray levels (command levels) to use to provide perceptively equal gray-shade steps from black to white.

**EXAMPLE ONLY:** For example, let's assume that the display has a “gamma” of 2.5, whereby the electro-optical transfer function could be expressed as (assuming  $V$  for black is zero)

$$L = aV^\gamma + L_K, \quad (4a)$$

$$\text{where } a = \frac{L_W - L_K}{V_W^\gamma}. \quad (4b)$$

Inverted, we have

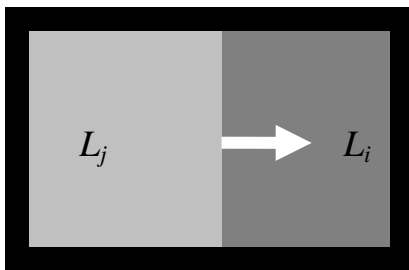
$$V = \left( \frac{L - L_K}{a} \right)^{1/\gamma}. \quad (5)$$

Assuming  $L_W = 100 \text{ cd/m}^2$ ,  $L_K = 0$ , and that  $V_W = 255$ , we obtain  $a = 9.6305 \times 10^{-5}$ , and the gray levels (command levels) rounded to the nearest integer are:  $V_n = 0, 56, 91, 139, 170, 212, 255$ . Note, these numbers are for this simple and ideal example ONLY.



## A230 BLUR, JUDDER, & SMOOTH-PURSUIT EYE TRACKING

We envision a vertical edge of an infinitely long block of one luminance  $L_j$  moving from left to right across a screen having a background luminance of  $L_i$ , where  $i \neq j$ . (See Fig. 1) We assume that for each refresh of the screen that this vertical edge moves (or jumps) a pixel increment of  $\delta n \geq 1$ ; each region of width  $\delta n$  will be called a jump region. We want to calculate what the eye sees using the simplest model we can. In this analysis, we will assume pixels that are 100 % filled; that is, we will assume that the pixels have no structure and are uniformly filling the surface area allocated to them.



**Fig. 1.** Moving edge of one luminance over another.

A number of parameters need to be defined to deal with motion artifacts. Here is the list of variables used:

$f$  = refresh rate (this is the frame rate for progressive-scan displays or the field rate for interlaced displays) in Hz:

$$f = 1/\delta t. \quad (1)$$

Note that  $f$  is the video refresh rate; that is,  $f$  is the rate at which information can be changed on the display surface. This refresh rate does not refer to any display framing rate that exceeds the rate at which information may be displayed. For example, a display may operate at 120 Hz in that it flips polarity at that rate, or it may operate at 180 Hz in a sequential mode, but in both cases, the video refresh rate is 60 Hz because the scene—the information—as viewed by the eye can only change at that slower rate.

$\delta t$  = frame (or field) time interval in seconds (s):

$$\delta t = 1/f. \quad (2)$$

This is also known as the video refresh period or simply refresh period.

$t$  = time in seconds from start of edge advancement:  
 $t = 0$  when the leading edge of the jump region is just to the left of the screen at the instant the leading edge is commanded to enter the screen area. For  $t > 0$  the edge has jumped into the

screen area at the left and the jump region begins to change (is activated) from the background. At  $t = 0$  is the beginning of the first frame.

$N_H$  = total integer number of pixels in the horizontal direction across the entire screen.  $N_H$  is an integer.

$n$  = pixel index (count or address) in the horizontal direction from  $n = 1$  at left to  $n = N_H$  at the right-most pixel;  $n$  is an integer.

$\delta n$  = pixel increment of advancement of the edge (jump in pixels) per screen refresh;  $\delta n$  is an integer.

$N_R$  = total number of full jumps across the screen:

$$N_R = \text{int}(N_H/\delta n); \quad (3)$$

$N_R$  is an integer.

$k$  = integer number indexing the jump regions from left to right—a counter:  $k = 1$  at the left side of the screen, and  $k = N_R$  for the last complete jump region at the right of the screen. The index  $k$  is a spatial index that is used to locate each jump region across the screen.

$t_k$  = time in seconds to the start of the activation of the  $k^{\text{th}}$  jump region

$$t_k = (k - 1)\delta t, \quad (4)$$

where  $t_k = 0$  for  $k = 1$ , the first jump region.

$u$  = edge average speed in px/s:

$$u = \delta n/\delta t. \quad (5)$$

If considered to be a velocity, it is directed toward the right.

$x'$  = non-integer distance from the left edge of the screen measured in units of pixels (*not* distance). The pixel  $n$  is related to  $x$  by

$$n = \text{int}(x') + 1, \quad (6)$$

where  $0 \leq x' < N_H$  is a continuous unit of measure in pixels and  $n$  is an integer count of the number of pixels from the left of the screen. For example, if we are considering a point at the center of the 12<sup>th</sup> pixel, then  $x' = 12.5$  px and  $n = 12$ . In terms of the actual distance  $x$  (in mm or m) from the left edge of the screen,  $x' = x/p$ , where  $p$  is the pixel pitch.

$n_p$  = pixel location of the edge for ideal or perfect (infinitely fast) transitions:

$$n_p = n_p(t) = \delta n \text{ int}(t/\delta t). \quad (7)$$

This is equivalent to identifying the farthest pixel (to the right) that is commanded (turned on, activated) to the new level in the jump region.



### SMOOTH-PURSUIT EYE TRACKING

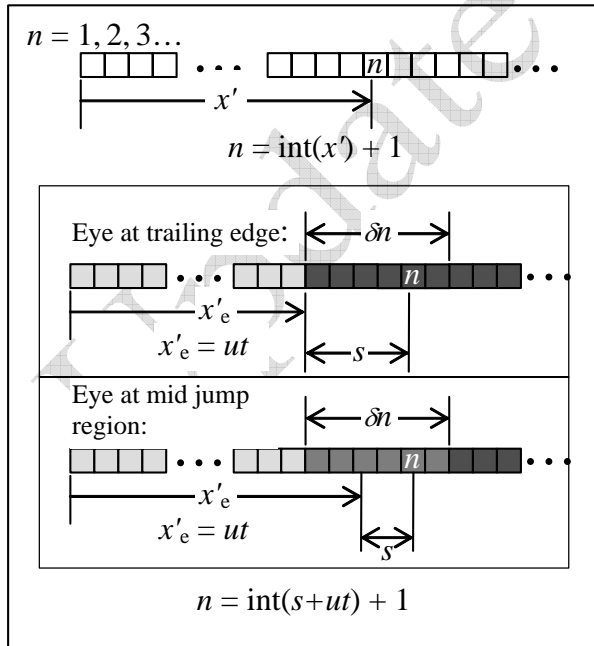
We now assume that the eye smoothly follows the trailing edge of the moving edge—smooth-pursuit eye tracking. This amounts to requiring the point of focus of the eye on the screen to move according to

$$x'_e = ut = \frac{\delta n}{\delta t} t, \quad (8)$$

which we will call the eye-tracking point—a continuous variable also in units of pixels that tells where the eye is looking as measured in units of pixels from the left of the screen. (The measure  $x'_e$  is exactly where on the screen the eye is looking in continuous units of pixels.) Relative to that eye-tracking point, we can think in terms of an on-screen relative retinal coordinate  $s$  that measures continuously in units of pixels from that eye-tracking point;

$$s = x' - x'_e, \quad (9)$$

which is simply the distance on the screen from the eye-tracking point measured in units of pixels. (To picture what  $s$  is, imagine a little  $x$ - $y$  coordinate system that is centered at the point where the eye is looking no matter where the eye looks—it moves around with the eye. The  $s$  coordinate is the horizontal position from the center of that little coordinate system in units of pixels along the  $x$ -axis or horizontal direction. This analysis is only concerned with the horizontal direction.) Combining these two equations, we can write a position on the screen in terms of the relative retinal coordinate and the time of observation since the start of the movement across the



**Fig. 2.** Continuous variable  $x'$  in units of pixels and the on-screen relative retinal coordinate  $s$  also in units of pixels.

screen:

$$x' = s + ut. \quad (10)$$

And we can then write the pixel count  $n$  in terms of the relative retinal coordinate and time as

$$n = \text{int}(s + ut) + 1, \quad (11)$$

which assumes smooth-pursuit eye tracking of the trailing (left-most) edge of the jump region. See Fig. 2.

### PERFECT TRANSITION VISUALIZATION

This section serves to illustrate how blur can arise because of smooth-pursuit eye tracking, although there may be no blur in the image on the screen. Let's confine our attention to the moving edge. At first we will consider that the transition between the two levels is perfect, that is, it is instantaneous, ideal. We will also consider the display to be on continuously; some call this a hold-type of display—where the luminance of a pixel (for this ideal case) will be essentially constant for the duration of the refresh period. Later we will incorporate temporal variations in the model.

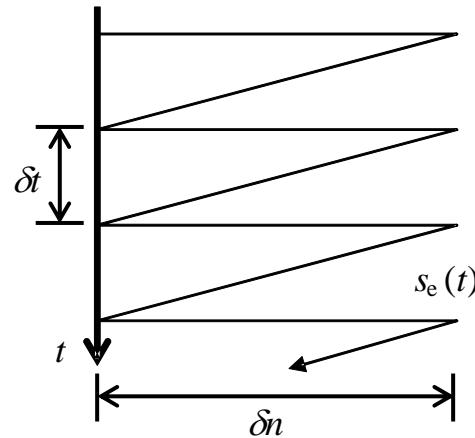
Consider the smooth-pursuit eye-tracking model where the eye tracks the motion without any jerkiness (no saccades). If the eye smoothly tracks the average position of the trailing edge of our advancing region, the pixel position of that tracking is [Eq. (8)]

$$x'_e = ut = t \delta n / \delta t. \quad (12)$$

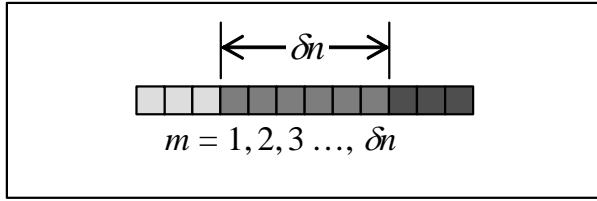
However, the edge is not moving smoothly, but moving along in jumps [according to Eq. (7)]:

$$n_p(t) = \delta n \text{int}(t / \delta t). \quad (13)$$

Because the eye is smoothly tracking the average position of the trailing edge, the position  $s_e$  of the edge as seen by the eye relative to its own moving coordinate system centered on the smooth-pursuit eye-tracking point is the difference between these quantities:



**Fig. 3.** Judder or blur arising from smooth-pursuit eye tracking of an ideal edge jumping across the screen in pixel increments of  $\delta n$ .



**Fig. 4.** Pixels within any jump region are labeled with the index  $m$ .

$$s_e(t) = n_p(t) - n_s(t), \quad (14)$$

which can be reduced to more basic quantities to give:

$$s_e(t) = \delta n [\text{int}(t/\delta t) - t/\delta t]. \quad (15)$$

This tracking gives rise to a sawtooth motion of the edge relative to the eye's gaze or tracking—see Fig. 3. If the refresh rate is slow enough a jerkiness is observed that is called judder. If the refresh rate is fast enough the edge appears to be blurred even though the transition between luminance levels is instantaneous. Keep in mind that the model we are discussing in this section only is for a hold type of a display where the pixels are illuminated throughout the refresh time and the transitions are perfect (instantaneous). The analysis that follows is general and does not require us to consider perfect transitions or even hold-type of displays. The following analysis will apply to impulsive displays (such as CRTs) as well as hold-type displays (such as LCDs).

### SMOOTH-PURSUIT EYE-TRACKING ASSUMING BLUR

We will now consider the case where we have a sufficiently fast refresh that we don't see judder, but we only see blur. We will consider a horizontal row of pixels or a narrow horizontal band of pixels and assume that all the pixels in any column  $n$  activate and perform the same way. Thus, we can write the luminance of that band (or row) as a function of pixel  $n$  and time  $t$ :

$$L_{ij} = L_{ij}(n, t). \quad (16)$$

Let's look at the edge near the center of the screen where we define

$$c = \text{int}\left(\frac{N_H}{2\delta n}\right) \quad (17)$$

to be the number of the beginning of a jump region just to the left of center or at the center. Because we are assuming blur, we can simply integrate the luminance  $L_{ij}$  for the edge transition near the center over a single refresh time period. However, because the eye-tracking point is not stationary, but moves across the jump region; we need to express  $n$  in terms of the eye-tracking coordinates in order to obtain what the eye sees  $K_{ij}(s)$  in terms of its own relative-retinal coordinates  $s$ . From Eq. (11) we have  $n$  in terms of  $s$  to obtain:

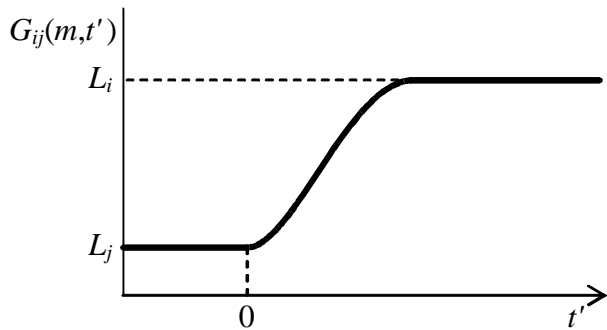
$$K_{ij}(s) = \frac{1}{\delta t} \int_{c\delta t}^{(c+1)\delta t} L_{ij}([\text{int}(s + ut) + 1], t) dt. \quad (18)$$

This provides us with the luminance as a function of continuous pixel position from the smooth-pursuit-eye-tracking point moving along with the edge motion at speed  $u$ . A pursuit camera that is moving with speed  $u$  and integrates for exactly one refresh period will obtain  $K_{ij}(s)$  directly (scaled appropriately in terms of  $s$  versus the camera pixels). Capturing an integer number of jump regions may be useful for noise reduction. If  $N$  jump regions are used, then the integral in Eq. (18) would be divided by  $N$  and the upper limit of integration would be  $(c + 1 + N)\delta t$ .

### MOVING EDGE SCREEN LUMINANCE

We now want to determine an expression for screen luminance  $L_{ij}(n, t)$  for an edge that moves in jumps based upon how the pixels change from one luminance  $L_i$  to a new luminance  $L_j$ . Once an expression for  $L_{ij}(n, t)$  is obtained, we may get some clues as to how many different ways it can be measured.

Within any jump region, we label the pixels with an index  $m = 1, 2, 3, \dots, \delta n$ . See Fig. 4. Consider any jump region. For each pixel  $n$  in the row of that jump region, suppose we know how the luminance changes for any transition  $i \neq j$  as the edge moves by that jump region; call this the transition luminance response  $G_{ij}(m, t')$ —see Fig. 5. Here,  $t'$  is the time as measured within any jump region. For this transition luminance response,  $G_{ij}(m, t')$ , suppose that the zero time,  $t' = 0$ , marks the beginning of the transition and is the same for all pixels within that jump region. What we now want to do is to write an expression for  $L_{ij}(n, t)$  based upon this understanding of how the jump region changes.



**Fig. 5.** Transition luminance response for each pixel  $m$  within a jump region.

We can write the luminance of the screen  $L_{ij}(n, t)$  in terms of  $G_{ij}(m, t')$  where we somehow confine the quantities  $m$  and  $t'$  to correctly describe the moving edge. To do this, we will introduce the sequencing factor

$$\text{int}\left(\frac{n-1}{\delta n}\right), \quad (19)$$



which provides an ordering of the jump regions. In fact the jump region index  $k$  can be defined by

$$k = \text{int}\left(\frac{n-1}{\delta n}\right) + 1. \quad (20)$$

The time of activation of the  $k^{\text{th}}$  jump region [Eq. (4)] now becomes

$$t_k = \delta t \text{int}\left(\frac{n-1}{\delta n}\right). \quad (21)$$

Table 1 illustrates how this sequencing factor functions as a way to order the jump regions. Essentially it tells us what jump region we are observing given any value of  $n$ . This sequencing factor will permit our regulation of the activities within the jump regions by using only the pixel position  $n$ , and it will permit us to write a comparatively simple expression for the screen luminance  $L_{ij}(n, t)$ .

**Table 1. Jump region ordering.**

$k$	Range of $n$	$\text{int}\left(\frac{n-1}{\delta n}\right)$
1	$1 \leq n \leq \delta n$	0
2	$\delta n + 1 \leq n \leq 2\delta n$	1
3	$2\delta n + 1 \leq n \leq 3\delta n$	2
...	...	...
$N_R = \text{int}(N_H/\delta n)$	$(N_R - 1)\delta n + 1 \leq n \leq N_R\delta n$	$(N_R - 1)$

We can now express the screen luminance  $L_{ij}(n, t)$  for the entire screen in terms of the transition luminance response  $G_{ij}(m, t')$  of a single jump region by carefully defining  $m$  and  $t'$  so that the screen is activated via a sequence of jump regions having the same response but at different times and places:

$$L_{ij}(n, t) = G_{ij}(m, t'), \quad (22)$$

where

$$m = n - \delta n \text{int}\left(\frac{n-1}{\delta n}\right), \quad (23)$$

and

$$t' = t - t_k = t - \delta t \text{int}\left(\frac{n-1}{\delta n}\right). \quad (24)$$

You will note the appearance of  $t_k$  as the expression after the minus sign. Thus  $t'$  remains less than zero until  $t > t_k$ . This is precisely what we want for the time-based motion of the edge moving in jumps. The jump regions activate sequentially. We can put this all together, but the expression is cumbersome and not particularly illuminating:

$$L_{ij}(n, t) = G_{ij}\left[\left[n - \delta n \text{int}\left(\frac{n-1}{\delta n}\right)\right], \left[t - \delta t \text{int}\left(\frac{n-1}{\delta n}\right)\right]\right]. \quad (25)$$

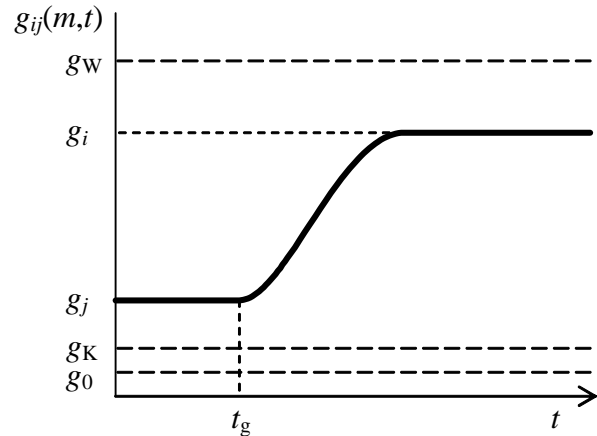
The term  $m$  recycles through each jump region; so it keeps track of where we are within any jump region no matter at which pixel  $n$  we are looking. The term  $t'$  activates the jump region at the appropriate time so that the edge moves across the screen in increments of  $\delta n$  for each refresh period  $\delta t$ . For times  $t' \leq 0$  then  $G_{ij}(m, t') = L_i$ ; and for long times,  $G_{ij}(m, \infty) = L_j$ .

In actuality, we rarely measure the luminance values  $G_{ij}(m, t')$  directly. We usually measure a voltage, a current, or obtain some detector pixel count (or level) in some sort of a digitized detector such as a CCD camera. Let  $g$  be what we actually measure, and assume it comes from a linear detector with a possible offset of  $g_0$  — see Fig. 6. We can associate  $g_W$  with the white luminance  $L_W$ ,  $g_K$  with black  $L_K$ ,  $g_i$  with  $L_i$ ,  $g_j$  with  $L_j$ , etc. The relationship between  $G$  and  $g$  is:

$$G_{ij}(m, t') = L_W \frac{g_{ij}(m, t' + t_g) - g_0}{g_W - g_0}. \quad (26)$$

Here the time scale of the recorded data  $g$  is shifted so that at  $t' = 0$  the transition for  $G_{ij}(m, t')$  begins. (We are also assuming that for no luminance,  $L = 0$ , then it must be that  $G = 0$ .)

What this analysis demonstrates is that if we can carefully measure the detailed time dependence of a jump region, then we can write the entire screen luminance  $L_{ij}(n, t)$  as a function of time. Once we have  $L_{ij}(n, t)$ , then we can use Eq. (18) to determine what the eye sees assuming smooth-pursuit eye tracking,  $K_{ij}(s)$ .



**Fig. 6.** Data obtained from linear detector to provide an indication of the luminance of the  $m^{\text{th}}$  pixel in a jump region as a function of time.



This page is intentionally left without information content.

DISCUSSIONS

Update Document

DISCUSSIONS