A gaze-contingent display to study contrast sensitivity under natural viewing conditions

Michael Dorr\textsuperscript{a} and Peter J. Bex\textsuperscript{a}

\textsuperscript{a}Schepens Eye Research Institute, Harvard Medical School, 20 Staniford St, Boston, USA

ABSTRACT

Contrast sensitivity has been extensively studied over the last decades and there are well-established models of early vision that were derived by presenting the visual system with synthetic stimuli such as sine-wave gratings near threshold contrasts. Natural scenes, however, contain a much wider distribution of orientations, spatial frequencies, and both luminance and contrast values. Furthermore, humans typically move their eyes two to three times per second under natural viewing conditions, but most laboratory experiments require subjects to maintain central fixation. We here describe a gaze-contingent display capable of performing real-time contrast modulations of video in retinal coordinates, thus allowing us to study contrast sensitivity when dynamically viewing dynamic scenes. Our system is based on a Laplacian pyramid for each frame that efficiently represents individual frequency bands. Each output pixel is then computed as a locally weighted sum of pyramid levels to introduce local contrast changes as a function of gaze. Our GPU implementation achieves real-time performance with more than 100 fps on high-resolution video (1920 by 1080 pixels) and a synthesis latency of only 1.5 ms. Psychophysical data show that contrast sensitivity is greatly decreased in natural videos and under dynamic viewing conditions. Synthetic stimuli therefore only poorly characterize natural vision.

Keywords: gaze-contingent display, contrast sensitivity function, natural scenes

1. INTRODUCTION

Contrast sensitivity functions (CSFs) describe the minimum contrast required for pattern detection and thus define the functionally critical boundary between images that are seen or not seen. CSFs are routinely employed in psychophysical, physiological, and clinical studies of vision because luminance variation (contrast) rather than luminance is encoded by early visual mechanisms\textsuperscript{1} in many species,\textsuperscript{2} whose receptive field profiles can be inferred by linear systems analysis.\textsuperscript{3} In the clinic, CSF deficits are diagnostic in many visual neuro-pathologies in patients with normal acuity or perimetry.\textsuperscript{4}

Current models of contrast sensitivity have been developed by presenting the visual system with synthetic stimuli such as sine-wave gratings at threshold contrasts on otherwise blank computer screens. While these studies have provided valuable insights into the visual system, it has been questioned how applicable these models are to natural vision,\textsuperscript{5} and the benefits of using more realistic rather than synthetic stimuli are currently under debate.\textsuperscript{6, 7}

Compared with traditional experimental stimuli used for measuring the CSF, natural scenes contain a much wider distribution of orientations, spatial frequencies, temporal frequencies, and both luminance and contrast values. Only recently, some work has begun to address how these factors influence contrast sensitivity when people view natural images.\textsuperscript{8}

A further limitation of classical paradigms is that subjects are usually required to maintain central fixation so that stimuli are displayed at a fixed eccentricity; under natural viewing conditions, however, humans typically move their eyes two to three times per second.\textsuperscript{9}

Recent technological advances in eye tracking and image processing speed make it now possible to study contrast sensitivity while an observer dynamically views dynamic scenes. In the following, we shall describe a

\begin{footnotesize}
\begin{itemize}
\item Further author information: (Send correspondence to M.D.)
M.D.: E-mail: michael.dorr@schepens.harvard.edu, Telephone: 1 617 912 2533
P.J.B.: E-mail: peter.bex@schepens.harvard.edu, Telephone: 1 617 912 2528
\end{itemize}
\end{footnotesize}
gaze-contingent display capable of performing real-time contrast modulations of high-resolution video in retinal coordinates. We will first briefly review Gaussian and Laplacian image multiresolution pyramids and present our algorithm for gaze-contingent synthesis of a Laplacian pyramid. Then, we shall describe our implementation on the Graphics Processing Unit (GPU) and discuss some performance considerations. Finally, we shall present empirical measurements that show that contrast sensitivity under natural conditions is very different from that usually measured in synthetic laboratory settings.

2. GAZE-CONTINGENT LAPLACIAN PYRAMID

Visual acuity is highest in the centre of the retina and falls off rapidly towards the periphery. To obtain detailed information from the whole visual scene, humans therefore move their eyes around about two to three times per second to sample the image with their high resolution fovea. Gaze-contingent displays exploit this property and render their contents as a function of gaze direction, which can be acquired in real time by an eye tracker. The first gaze-contingent displays masked parts of the visual field, for example to investigate the perceptual span in reading research or to simulate scotomata to study visual search strategies; other possible uses include changes to the scene during a saccade to study change blindness, trans-saccadic integration, or saccadic adaptation. Here, we describe a gaze-contingent display that bandpass-filters the video input in retinal coordinates. It is based on an algorithm by Perry and Geisler who simulated arbitrary visual fields by smoothly blending between differently low-pass-filtered versions of the input; these versions were efficiently represented on a Gaussian multiresolution pyramid. Even though a cutoff frequency can be specified for each pixel in retinal coordinates, the algorithm avoids the need to explicitly convolve individual pixels with filters of varying bandwidth, and the real-time application in a low-latency, gaze-contingent display thus became feasible. Instead of a spatial Gaussian pyramid, we now use a spatial Laplacian pyramid, which yields an efficient bandpass representation of the input, in order to specify weights for individual frequency bands. In principle, the possible use of a Laplacian had been mentioned already in the original paper; in practice, this modification significantly increases computational complexity for two reasons. First, computing the Laplacian is obviously more costly because computation of the Gaussian is one part of Laplacian pyramid analysis; for each downsampling operation, one additional upsampling operation and one subtraction at the higher resolution are necessary. A second performance issue arises because of the need to store signed intermediate results, i.e. the need of a sign bit without losing precision: in practice, this leads to at least a doubling of the data type width from 8 bits to 16 bits, so that required memory bandwidth is also doubled (plus an additional overhead because of the limited support of signed words in typical processor instruction sets). We overcome these computational constraints by an implementation on dedicated high-throughput graphics processing hardware and achieve more than 100 frames per second for high-resolution blu-ray video input.

2.1 The Laplacian pyramid

The Laplacian pyramid is one of the fundamental data structures in image processing. We denote the original image by \( I(x, y) \) with a width of \( W \) and a height of \( H \) pixels. We can now create a Gaussian pyramid that contains progressively low-pass filtered versions of \( I \) by defining \( I \) to be the “higest” level \( G_0(x, y) \); the other Gaussian levels \( G_k \), which have a resolution of \( W/2^k \) by \( H/2^k \) pixels, can then be computed iteratively by low-pass-filtering and downsampling:

\[
G_{k+1}(x, y) = \sum_{i=-c}^{c} \sum_{j=-c}^{c} w_{ij} G_k(2x + i, 2y + j),
\]

with a filtering kernel \( w \) of length \( 2c + 1 \) and coefficients \( w_1, ..., w_c \).

Because of the linearity of the Fourier transform, the subtraction of a Gaussian level \( G_{k-1} \) from level \( G_k \) will result in an image with only the high-frequency content of \( G_k \) remaining (which is a bandpass relative to \( G_0 \)); since \( G_{k-1} \) and \( G_k \) have different resolution, however, \( G_{k-1} \) needs to be upsampled first. This can be achieved by inserting zeros and a subsequent interpolation with a lowpass filter:

\[
\hat{G}_{k-1}(x, y) = \sum_{i=-c}^{c} \sum_{j=-c}^{c} w_{ij} G_k \left( \frac{x - i}{2}, \frac{y - j}{2} \right),
\]
where only those pixels of $G_k$ are included in the sums that exist for $G_k$, i.e. where $(x-i) \mod = 0, (y-j) \mod = 0$. Images of matching resolution can now be subtracted to obtain a bandpass representation:

$$L_k(x, y) = G_k(x, y) - \uparrow G_{k+1}(x, y).$$

The lowest level represents the DC component and is the same as on the Gaussian pyramid,

$$L_N = G_N.$$

Pyramid synthesis is straightforward and is achieved by reverting the analysis steps, i.e. by iteratively up-sampling and adding the Laplacian levels:

$$L'_k(x, y) = L_k(x, y) + \uparrow L'_{k+1}(x, y),$$

with $L'_0(x, y)$ a faithful reconstruction of the original image (if the $L_k$ remained unmodified); because of the missing neighbour for the DC component, we set

$$L'_N = L_N.$$

So far, however, we have only reconstructed the original image but not modified it. To locally change individual frequency bands, we introduce a weighting mask $\beta$ that is specified in retinal coordinates (as a function of gaze position $(g_x, g_y)$ and image coordinates):

$$L'_k(x, y) = \beta_k(g_x, g_y, x, y) \cdot L_k(x, y) + \uparrow L_{k+1}(x, y).$$

In locations where $\beta = 1$, the original input remains unchanged; a $\beta > 1$ leads to a contrast increase in frequency band $k$. In the experiment described in the following, we have also used a global $\beta = 0.75$ to reduce the overall contrast of the movie, such that local increases of image contrast did not lead to pixel overflows.

### 2.2 Implementation on the Graphics Processing Unit

We implemented all image processing algorithms on the Graphics Processing Unit using the “C for graphics” (Cg) programming language. GPUs typically have much higher memory bandwidth than CPUs and operate in a massively parallel fashion, executing small kernels on a per-pixel basis. In contrast to the much higher raw pixel throughput, however, GPU programming incurs an additional communication overhead to move data and program code from the CPU to the GPU via the system bus, which has implications for algorithmic design. For example, one consequence of this overhead is that one two-dimensional convolution operation may be faster than two one-dimensional convolution operations with a separable filter. We here used a 5x5 binomial filter that could be further reduced to a 4x4 filter by making use of the GPU’s hardware interpolation mechanism and specifying fractional pixel indices (or texels). This texel interpolation proved also useful for the weighting masks $\beta_k$: to reduce required memory bandwidth, these could be stored at lower resolution than the Laplacian levels $L_k$. In a CPU implementation, the necessary coordinate scaling to multiply two images of different resolution would be prohibitively expensive, whereas this scaling comes at no additional cost on the GPU. The GPU also supports a wider range of data types natively. The Laplacian levels $L_k$ represent differences of Gaussian levels $G_l$ and therefore require a sign bit in addition to the 8 bits of the video input; a quantization step to use only 7 bits would be computationally expensive. Signed words (16 bits) are only partially supported by the CPU instruction set, so that the fastest implementation uses full floats (32 bits) to represent the $L_k$. On the GPU, on the other hand, we could use a “half float” data type with one sign, five exponent bits, and 10 bits precision, so that memory requirements were reduced by 50% compared to the CPU implementation.

A further major difference between CPU and GPU is the different colour space. Video material is commonly stored in the $Y'CbCr$ colour space with one full-resolution luma channel and two colour channels at half resolution, and these channels can be processed independently on the CPU. The GPU, on the other hand, always operates on three colour channels simultaneously and uses the $RGB$ colour space, which requires twice the memory footprint (and bandwidth) as $Y'CbCr$. For our gaze-contingent display, it turned out that the necessary colour space
conversion and transfer of video frames to the GPU texture memory was a performance bottleneck; we therefore transferred the $Y'$, $C_b$, and $C_r$ channels to three individual textures and performed the colour space conversion and collation into one $RGB$ texture on the GPU. This reduced the parallelity of CPU and GPU operations, but also reduced the required transfer bandwidth by 50%.

The overall memory requirements were the same for both implementations because the effects of different bit depth and different colour space compensated each other.

2.3 Miscellaneous considerations

The critical performance criterion for a gaze-contingent display is the latency between an eye movement and a corresponding change on the display. Commercially available eye trackers now reach sampling rates of up to 2000 Hz and latencies of about 1–2 ms; as we shall see in the Results section, the filtering algorithms presented in this manuscript process a high-resolution video frame in about 4 ms. This makes the display device the slowest link in the chain (video decoding in our system takes about 8 ms per frame, but can be performed in parallel on a separate CPU core). Traditionally, vision scientists have used CRTs, which have well-understood temporal characteristics, and provide refresh rates of up to 200 Hz. At these high temporal resolutions, however, spatial resolution is severely limited. Because we strived to study vision under as naturalistic conditions as possible, a trade-off would have been necessary between a reasonably wide field-of-view and a sufficient spatial resolution per visual angle. Therefore, we evaluated alternative display options and TFT screens in particular. Driven by the 3D market, these have recently become available at refresh rates of up to 120 Hz and spatial resolution of up to 1920 by 1080 pixels (blu-ray standard), but the exact temporal characteristics are often difficult to determine because of internal digital post-processing, which can add significant delays.\footnote{We therefore measured the delay between a drawing command to the graphics card and its corresponding update on the screen with a photodiode and a USB-attached digital I/O acquisition board. Despite marketing claims that certain models offered “thru” modes without an additional input lag, we could not find a panel with Full HD resolution (1920 by 1080 pixels) that did not add at least one 60 Hz refresh cycle (16 ms) to input lag compared to a CRT. Thus, we used a ViewSonic VX2265wm TFT panel with slightly lower resolution (1680 by 1050 pixels) that was measured with only 4 ms additional latency compared to a CRT running at 120 Hz.}

A further potential issue when using TFTs for vision science experiments is their relatively low colour resolution. Even though TFT panels with 8 or even 10 bit grayscale resolution exist, they are unsuitable for our experiment because of their large input lag. All currently available panels with 120 Hz temporal resolution use only 6 bits to represent colour. However, as we shall see in the Results section, human contrast discrimination in complex moving scenes is even lower, so that we safely can use such displays for our gaze-contingent system.

Finally, one important implementation detail is the synchronization of time stamps between eye tracker and display PC. Initially, we stored gaze samples to disk only with the time stamps assigned to them by the eye tracker because they are spaced at seemingly perfect intervals of 1 ms (at 1000 Hz sampling frequency). However, all other events (e.g. stimuli on- and offsets) use time stamps based on the display PC clock, and both real time clocks are not synchronized. Even a small difference of as much as 0.005% in the two clocks led to large time
shifts over the course of long experiment sessions (e.g. 30 ms over 10 minutes). Therefore, we assigned time stamps to gaze samples also based on their arrival time at the display PC; because samples were collected from the network socket only once every refresh cycle (every 8 ms), absolute arrival time was used for the most recent sample and eye tracker time stamps were used to determined the temporal offset for previous samples.

3. EXPERIMENTAL SETUP

We used the gaze-contingent Laplacian pyramid to measure contrast sensitivity in the 1.5–3 cyc/deg frequency band on natural movies. Subjects were presented with video clips of about 10 minutes duration each from the BBC Planet Earth Series Blu-Ray Edition (for an example still shot, see Fig. 1). The videos originally had a spatial resolution of 1920 by 1080 pixels, but were cropped to the central 1680 by 1050 pixels to natively fit on a ViewSonic 3D VX2265wm TFT screen running at 120 Hz and with a maximum luminance of 260 cd/m²; at a viewing distance of 75 cm, the screen covered 35 by 22 deg of visual angle and the maximum resolution was 24 cyc/deg. Eye movements were recorded at 1000 Hz using an EyeLink 1000 system. Throughout the presentation, each video frame was decomposed into its frequency bands in real time on a pyramid with seven spatial scales. In order to create some headroom for the subsequent contrast modulations, all scales but the DC were multiplied with a factor of 0.75. At random intervals (1.5–2.5 s after a response), a 2 by 2 deg patch in the target frequency band was modulated by a factor $\alpha$ (one of 1.5, 3.0, 4.5, 6.0) for 600 ms. To avoid visible patch edges, this modulation was smoothly ramped on and off in time (one-sided Gaussians of s.d. 120 ms for the first and last 240 ms) and space (Gaussian of s.d. 1 by 1 deg). Target location was randomly chosen on one of the cardinal axes 2 deg away from the current gaze position and updated at 120 Hz. After target offset, a gaze-contingent cross at fixation prompted the subject to indicate the perceived target location (4AFC) and subsequently changed colour to give feedback on the accuracy of the response.

In an offline analysis step, the modified image patch (11 by 11 pixels on the fourth spatial scale) was extracted for each 8 ms refresh cycle throughout the 600 ms target presentation. Pedestal contrast was computed as the standard deviation of pixel intensities in the patch; these s.d.s were then averaged over the red, green, blue channels and all refreshes. Overall, we collected about 6000 trials that were binned into eight blocks, and psychometric functions were fitted to the relative contrast enhancement (ΔRMS, as a function of pedestal contrast) in each of the bins using the “psyphy” package for the R statistical software. Finally, the 50% correct target localization threshold was estimated for each bin; the commonly used 75% correct point could not be used because subjects did not reach that performance level for a wide range of pedestal contrasts.

4. RESULTS

4.1 Runtime measurements

Runtime measurements for both the CPU and the GPU implementation are shown in Fig. 2. The CPU implementation was evaluated with an Intel Core i7 920 CPU running at 2.66 GHz and used the OpenCV and the AMD Framework Performance libraries for highly optimized image processing routines; memory bandwidth in this system was measured to be about 12 GB/s. The GPU implementation was run on an NVIDIA GTX260 with 216 unified shader processors and a specified memory bandwidth of about 112 GB/s. Clearly, the GPU implementation is much faster (by a factor of more than five); the latency-critical synthesis step requires less than 1.5 ms, compared to more than 6 ms on the CPU (here, a further normalization step is required to convert the floating-point pyramid scales to the (0, 255) range for output). The most expensive computational step is the upload of video frames to the GPU texture memory (and the internal conversion from $Y'C_b'C_r$ to RGB colour space) with about 2 ms, but this step needs to be executed only once per video frame (i.e. at 24 frames per second) and is independent of the number of spatial scales. In practice, GPU performance is even slightly underestimated here because we inserted synchronization calls for greater timing accuracy; in release code, the GPU driver can parallelize some of the operations and achieve higher overall throughput.

With these performance numbers, the input lag measurements for the monitor (see above), and the eyetracking latency of 1.8 ms as specified by the manufacturer, we can estimate overall system latency as follows. Update of the screen buffer is synchronized to the vertical refresh of graphics card and begins immediately after a buffer swap (i.e. drawing of the previous frame). Typically, all image processing can be performed in 4 ms (in
24 out of 120, i.e. 80% of refresh cycles, texture upload and pyramid analysis are not necessary, so that image processing time then is only 1.5 ms) and thus before the next buffer swap (every 1/120 Hz = 8.3 ms); the TFT then takes up to 12.5 ms to physically display the buffer contents (in a top-to-bottom scanning fashion), so that the typical overall system latency sums to 18.2 ms for the central part of the screen and 22.4 ms for its bottom. However, logging time stamps for all buffer swaps revealed that occasionally single refresh cycles were skipped due to concurrent system activity, and the worst-case latency thus is 30.7 ms.

### 4.2 Contrast sensitivity measurements

Fig. 3 shows contrast discrimination thresholds for dynamic natural scenes when observers freely make eye movements. When analogous thresholds are measured under fixation and for isolated grating stimuli or static natural images, a characteristic “dipper”-shaped function is typically observed. Contrast increment detection thresholds are typically lowest when the pedestal (background) contrast is just above contrast detection threshold. On the assumption that discrimination thresholds are directly related to the slope of some underlying contrast response function, these data are generally modeled with the first derivative of a compressive function, e.g. Naka Rushton. Such compressive contrast response functions that are inferred from behavioural contrast discrimination data are similar to those observed in single unit recordings from neurones in early visual areas of primate brains. The present data increase approximately linearly with pedestal contrast, showing no evidence of a dipper at any contrast. This surprising result indicates that under natural viewing conditions, the underlying contrast response of the visual system may be relatively linear. In a previous study of contrast discrimination in static natural scenes with fixed gaze, we showed that dipper functions were less pronounced when spatial structure was aligned. This observation was attributed to contrast gain control from neural responses to spatial structure that is near in position and spatial frequency. The present results suggest that this local contrast gain control may be further weakened when either the images or the eyes are dynamic.
Figure 3. Contrast sensitivity under natural viewing conditions as a function of pedestal contrast, averaged over four observers. We fitted psychometric functions to eight bins of pedestal contrast ranges (left; shown here are only four bins) and estimated their 50% correct points (solid dots). The resulting curve of 50% thresholds as a function of pedestal contrast (right) is almost linear; in particular, there is no indication of the “dipper function” effect.

5. CONCLUSION

We here presented a gaze-contingent display that bandpass-filters video in retinal coordinates. Implemented on the Graphics Processing Unit, we achieve real-time performance on blu-ray high resolution videos.

Psychophysical data obtained with our gaze-contingent display show that contrast sensitivity is greatly decreased in natural videos and under dynamic viewing conditions. Synthetic stimuli therefore only poorly represent natural vision and tasks.

ACKNOWLEDGMENTS

Supported by NIH EY018664 and EY019281.

REFERENCES


