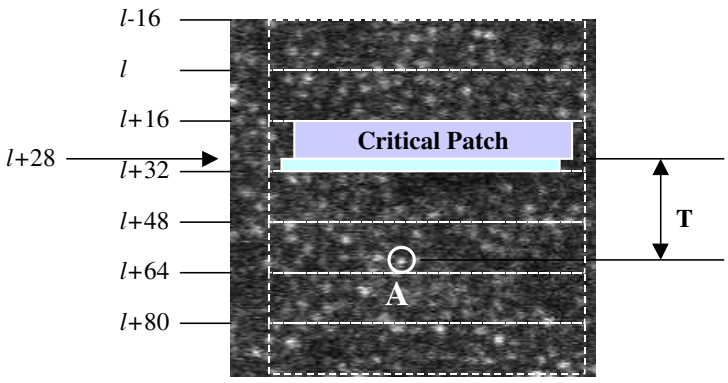


Fig. 7. An example of adjusting the buffering unit to the critical patch. 'A' is the target location.

In Fig. 7, the raw video is still buffered to the PC in every 16 lines, when there is no critical patch involved, e.g., [l-16, l-1] and [l, l + 15]. However, when the critical patch happens to have the last line at line l + 28, the FPGA will be programmed to buffer lines [l + 16, l + 28] to the PC immediately after line l + 28 is sampled by the A/D converter. Therefore, there is no need to move the critical patch back to lines [l, l + 15], as was the case for the multiple-board option (Fig. 5). The PC algorithm starts calculating the target location as soon as the PC receives lines [l + 16, l + 28] and this trims the prediction time to T exactly, instead of T + Ts. The sampling latency is therefore basically eliminated. This approach does add one more interrupt handling event in each frame, from the previous 32 interrupts to 33 interrupts, because the FPGA needs to buffer lines [l + 29, l + 31] to the PC separately. The total CPU usage for video sampling only on an Intel Core 2 Quad (Q6700 @ 2.66GHz) CPU is less than 5%.

(2) The second design objective was a common pixel clock for the D/A and A/D to eliminate misalignment of the input image and the target pattern due to PLL skews. This was simple to achieve since the same clock signal from the FPGA could be routed to both converter chips.

(3) The third design goal was to provide the buffering and control to allow the stimulus pattern to be preloaded into the FPGA buffer, and to be sequenced to the stimulus output channel with the correct timing to present it to the desired location in the raster under control of the PC software by merely uploading stimulus location coordinates for each frame. This represented a significant improvement over the multiple-board solution which required uploading all the pixels in the stimulus pattern raster for each frame to adjust the location of the stimulus. The encoding of stimulus pattern is simple if the stimulus size is 16x16 pixels or smaller, because there is only one pair of (x, y) coordinates to determine its location. It gets complicated with larger stimulus patterns. Large stimuli involve longer delivery times, during which eye motion can induce non-linear distortions that must be compensated as they occur. Hence, the algorithm needs to calculate a sequence of (x, y) coordinates for sequential patches of the stimulus pattern. For example, with a 180x180 pixel stimulus pattern, we calculate coordinates at lines 0, 32, 64, 96, 128, 160, 176. We then use these seven pairs of (x, y) to pre-warp the stimulus pattern and encode it to the two AOMs. We assume there is no intraline distortion because of the short duration of the horizontal sweep.

4. Results

We present five examples of stimulus delivery with 3-msec, 4-msec, 5-msec, 6-msec prediction times (latency) and 1 frame delay (33 msec), from a living retina. Figure 8

illustrates two live videos with prediction times 3 msec and 4 msec, and Fig. 9 illustrates another three live videos with prediction times 5 msec, 6 msec and a whole frame (33 msec). In all of the following examples, the stimulus is generated by modulating the imaging laser. As such, the stimulus gets encoded directly into the image. Switching the same modulation pattern to a second laser is trivial. Under typical operating conditions, the power of the second laser is generally not sufficient to be recorded into the image and so this simpler mode of operation is the most appropriate for illustration purposes.

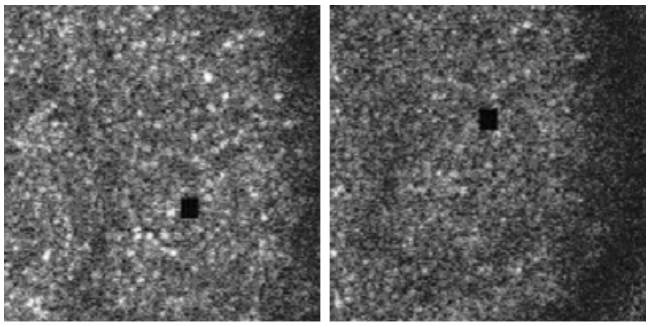


Fig. 8. Stimulus accuracy with different prediction times. (a) is with 3-msec latency, and (b) is with 4-msec latency (Media 1). The accompanying movies have been compressed to reduce file size and underrepresent the quality of the original AOSLO video.

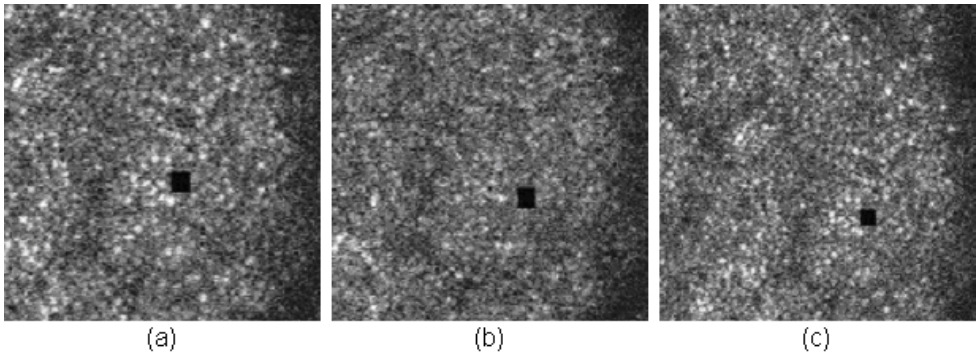


Fig. 9. Stimulus accuracy with different prediction times. (a) is with 5-msec latency, (b) is with 6-msec latency, and (c) is with one frame (33 msec) latency (Media 2). The accompanying movies have been compressed to reduce file size and underrepresent the quality of the original AOSLO video.

The RMS accuracy of the stimulus location is plotted in Fig. 10. It is worth noting that the results below are calculated from only one sample (600 frames of video) in each case.

#128571 - $15.00 USD Received 17 May 2010; revised 27 Jul 2010; accepted 28 Jul 2010; published 4 Aug 2010

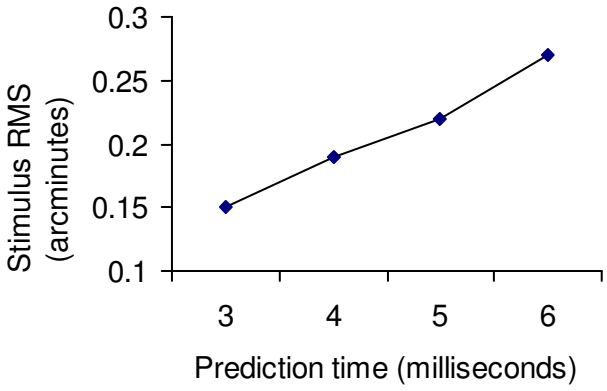


Fig. 10. RMS error of stimulus location versus prediction latency

We calculate the RMS error from the recorded raw video with standard cross correlation which is totally independent of the MSC algorithm, illustrated in Fig. 11 below.

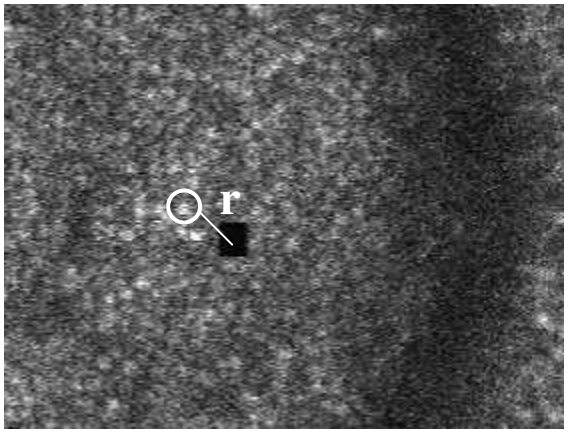


Fig. 11. Evaluation of stimulus accuracy

In Fig. 11, we use cross correlation to locate the stimulus (the black square) and a neighboring patch of cones, and measure how the distance **r** between them varies frame by frame. When the patch of cones is selected a) very close to the stimulus, and b) nearly in the same horizontal level as the stimulus, then the variation of **r** represents the accuracy of the stimulus placement. The size of the stimulus is 16x16 pixels, and the typical size of the cone is 9x9 pixels. A patch of 16x16 pixels is used to track the stimulus and a patch of 9X 9 pixels is used to track the cone.

With the benefit of high programmability of FPGAs, we can encode a large stimulus pattern to the D/A to control the two AOMs. Moreover, we can program the board to deliver animations to targeted retinal locations. Figure 12 shows an example where an animated letter "E" fades in at a targeted retinal location. The size of "E" spans multiple cones, and is large enough for the subject to resolve it. In the actual experiment, the stabilized "E" will fade from view, which is a common phenomena reported in the literature [37,38].

#128571 - $15.00 USD Received 17 May 2010; revised 27 Jul 2010; accepted 28 Jul 2010; published 4 Aug 2010

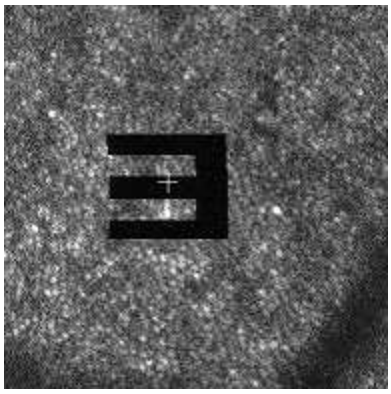


Fig. 12. Stabilized stimulus (video) on retina (Media 3)

We can also deliver a gray-scaled image on the retina, as illustrated in Fig. 13.

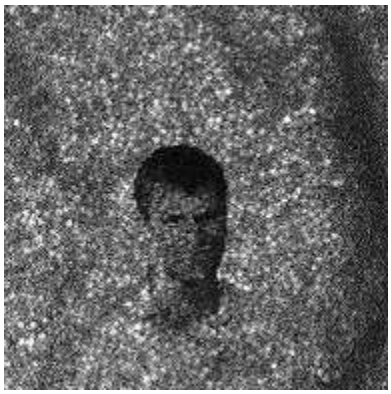


Fig. 13. Gray-scale stimulus on retina (Media 4)

5. Summary

1. With the integrated adapter solution, we have reduced the prediction time to 3 msec for small stimulus sizes (e.g. 16x16 pixels). The 3 msec provides a small "pad" over the original budget of 2 msec to allow for possible motion between the pre-critical patch and the critical patch, and to allow for the radius of the stimulus pattern. This case halves the previous prediction time needed for the multiple-board solution, and results in a reduction of the stabilization error from 0.27 arcmin (also 0.26 arcmin reported in [35]) to 0.15 arcmin.
2. The stimulus size can now be as large as the available buffer size on FPGA, which is currently 256x256 pixels, large enough to allow some motion within the 512 x 512 frame of the raw video. However, larger stimuli impose longer latencies, limited by the current speed of calculation of stimulus location and dewarping parameters. This may be mitigated by faster PC hardware or moving the computations to GPU hardware. The latter option is currently under investigation.

6. Discussion

To our knowledge, the tracking and stabilization is more accurate than any other method reported in the literature. Comparisons of the methods employed here vs other methods were described in the second paper of this series [35] and are summarized in Table 1.

#128571 - $15.00 USD Received 17 May 2010; revised 27 Jul 2010; accepted 28 Jul 2010; published 4 Aug 2010

Table 1. Comparison of other tracking and targeted stimulus/beam delivery methods.

Method	Tracking method	Tracking accuracy	Latency	Stabilization accuracy	Comments
AOSLO	Retinal image tracking	<0.1 arcmin	3 msec	0.15 arcmin	Gaze contingent stimulus projection. The stimulus is corrected with adaptive optics and can be as compact as a single cone.
Optical lever	Direct optical coupling	0.05 arcmin [39]	0 (optical)	0.38 arcminutes [40]	Stimulus is very precise but contact lens slippage will cause uncontrollable and unmonitorable shifts in stimulus position
Dual Purkinje (dPi) Eye Tracker with optical deflector [41]	Purkinje reflexes from cornea and lens	~1 arcminute [42]	6 msec	~1 arc minute (error is dominated by tracking accuracy)	
EyeRis™* [43]	dPi**	~1arcmin (dPi)	5-10 msec	~1 arcmin (tracking limited) [44]	Gaze contingent display.
MP1(Nidek, Japan)	Retinal image feature tracking	4.9 arcmin [45]	2.4 msec	Not reported	Gaze contingent display for single stimulus presentations (clinical visual threshhold measurements)
Physical Sciences Inc [46].	Retinal feature tracking	3 arcmin	<1 msec	3 minutes (tracking limited)	Used to optically stabilize a scanning raster on the retina to facilitate line scanning ophthalmoscope imaging.
Heidelberg Spectralis OCT (Heidleberg, Germany)	Retinal image feature tracking	Not reported	Not reported	Not reported	Used to stabilize the OCT b-scan at a fixed retinal location to facilitate scan averaging.

* This technique falls into a broad class of eye trackers coupled with gaze contingent displays. The EyeRIS system here has the best reported performance of any of the systems we found.
** Any tracking method can be used for this type of system but results from a dPi system are reported since they provide the best results.

Although the AOSLO has a performance advantage over other systems in many categories, there are important limits to the scope if its application. In general, all the systems with the exception of AOSLO are capable of measuring eye motions and controlling the stimulus or the display over a relatively large visual field. The following arguments explain why AOSLO tracking and stimulus presentation will only work for a limited range of eye motion. First, the method demands that the stimulus is placed within the confines of the scanning raster, which is typically between 1 X 1 and 2 X 2 degrees and never greater than 3 X 3 degrees. Second, tracking will begin to fail whenever the current frame starts to lose overlap with the reference frame by 50% or more. Third, the extent of the stimulus is limited by the FPGA buffer, whose current maximum limit is 256 X 256 pixels. With an AOSLO field size of 512 X 512 pixels, this further limits the range of eye motion for which an extended stimulus can be presented. As such, the tracking and stimulus delivery are practical mainly for an eye that is fixating.

7. Conclusion

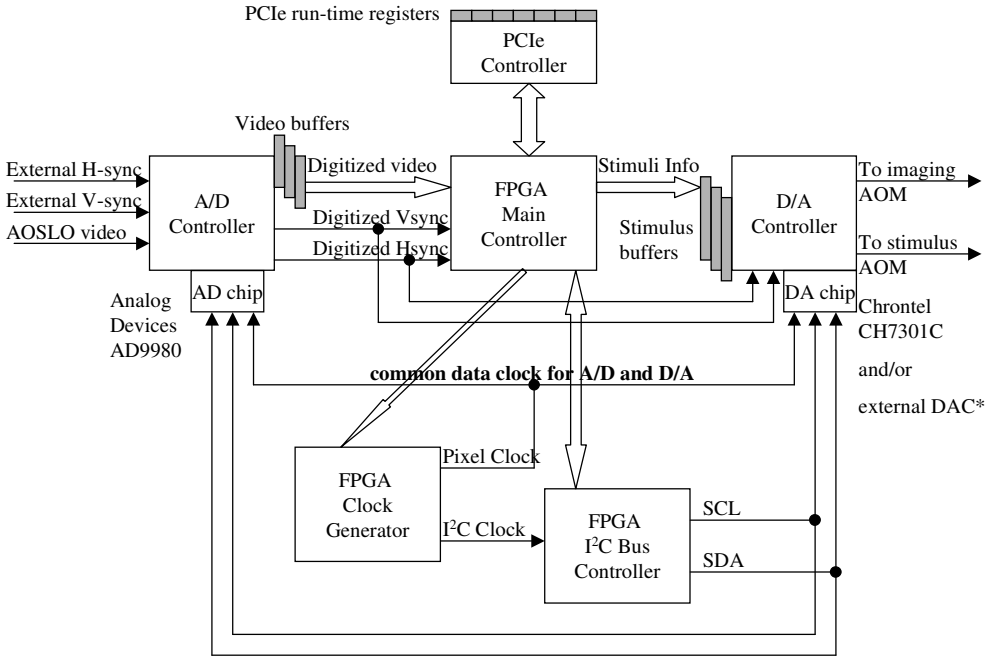
The FPGA solution to eye tracking and targeted stimulus delivery presented here represents a significant improvement in performance and reduction in cost over the previous solutions that have been implemented. These improvements will not only enhance ongoing research, but the ability to present larger stimuli to targeted locations broadens the scope of potential applications, which range from targeted delivery of therapeutic lasers, presentation of large

#128571 - $15.00 USD Received 17 May 2010; revised 27 Jul 2010; accepted 28 Jul 2010; published 4 Aug 2010

stimuli for receptive field measurements and basic psychophysics related to perception of stabilized and moving targets.

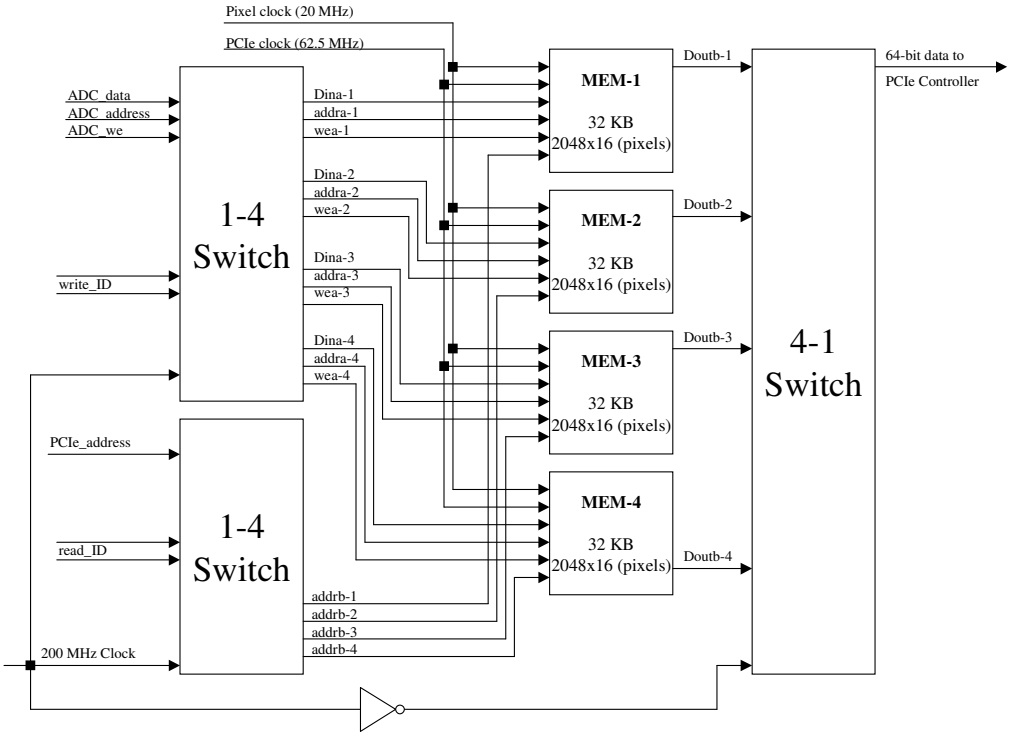
#128571 - $15.00 USD Received 17 May 2010; revised 27 Jul 2010; accepted 28 Jul 2010; published 4 Aug 2010

Appendix 1: A detailed block diagram of the FPGA applications



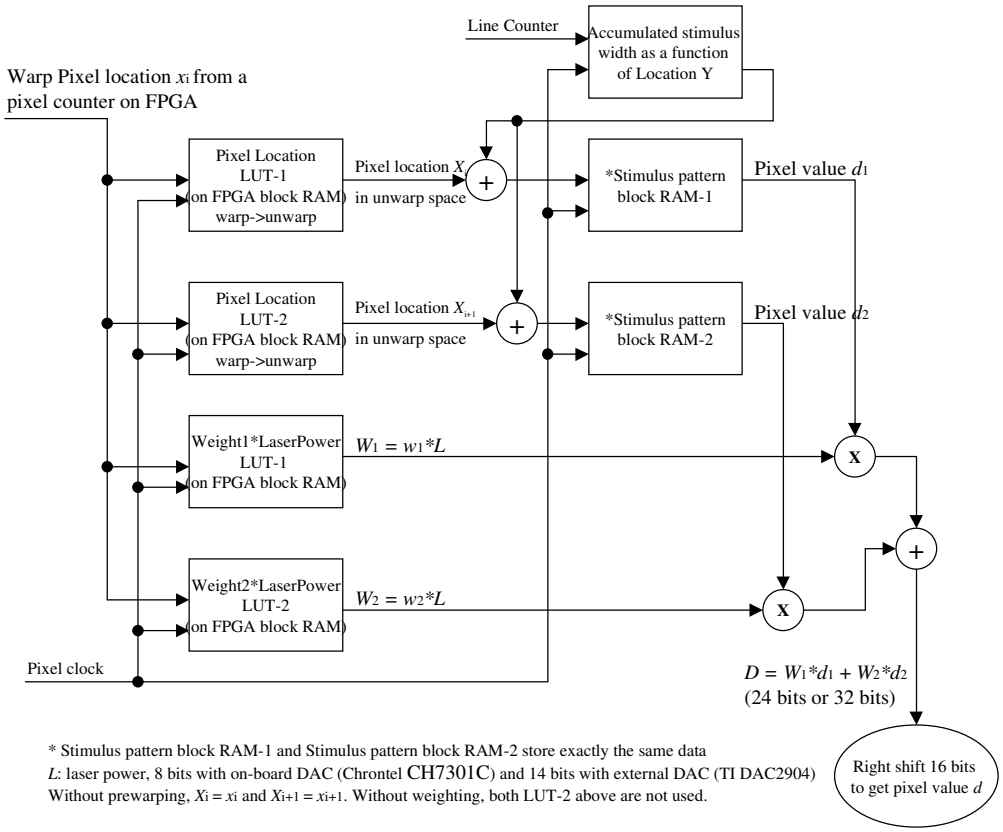
*: An example is to attach a TI's DAC2904-EVM module (14 bits) for high accuracy AOM control

Appendix 2: FPGA block RAM map for video buffers of the D/A controller



Appendix 3: FPGA block RAM map for stimulus buffer of the D/A controller

#128571 - $15.00 USD Received 17 May 2010; revised 27 Jul 2010; accepted 28 Jul 2010; published 4 Aug 2010



Acknowledgement

This work is funded by National Institutes of Health Bioengineering Research Partnership Grant EY014375 and by the National Science Foundation Science and Technology Center for Adaptive Optics, managed by the University of California at Santa Cruz under cooperative agreement AST-9876783.