Research Article | Vol. 15, No. 12 / 1 Dec 2024 / Biomedical Optics Express 6799
Biomedical Optics EXPRESS

Embedded CPU-GPU pupil tracking

BARTLOMIEJ KOWALSKI, XIAOJING HUANG, AND ALFREDO DUBRA*

Department of Ophthalmology, Stanford University, Palo Alto, CA 94303, USA
**adubra@stanford.edu*

Abstract: We explore camera-based pupil tracking using high-level programming in computing platforms with end-user discrete and integrated central processing units (CPUs) and graphics processing units (GPUs), seeking low calculation latencies previously achieved with specialized hardware and programming (Kowalski et al., [Biomed. Opt. Express **12**, 6496 (2021)]. Various desktop and embedded computers were tested, some with two operating systems, using the traditional sequential pupil tracking paradigm, in which the processing of the camera image only starts after it is fully downloaded to the computer. The pupil tracking was demonstrated using two Scheimpflug optical setups, telecentric in both image and object spaces, with different optical magnifications and nominal diffraction-limited performance over an ~18 mm full field of view illuminated with 940 nm light. Eye images from subjects with different iris and skin pigmentation captured at this wavelength suggest that the proposed pupil tracking does not suffer from ethnic bias. The optical axis of the setups is tilted at 45° to facilitate integration with other instruments without the need for beam splitting. Tracking with ~0.9-4.4 µm precision and safe light levels was demonstrated using two complementary metal-oxide-semiconductor cameras with global shutter, operating at 438 and 1,045 fps with an ~500 × 420 pixel region of interest (ROI), and at 633 and 1,897 fps with ~315 × 280 pixel ROI. For these image sizes, the desktop computers achieved calculation times as low as 0.5 ms, while low-cost embedded computers delivered calculation times in the 0.8-1.3 ms range.

1. Introduction

Pupil tracking can be defined as the estimation of the location, size and/or orientation of the pupil of the eye for monitoring changes in gaze and involuntary fixational movement (rotation and cyclotorsion) [1–3]. Pupil tracking has multiple applications, including computer graphical user interface design [4,5], interfacing human and computers [6–8], virtual and augmented reality [9–14], ophthalmic imaging [15,16] and vision testing [17,18]. Some of these applications require access to pupil data with minimal latency to take action in response to eye movement, such as, updating visual stimulation [19], controlling of surgical lasers [20,21], and field of view steering in retinal imaging [22,23]. Real time tracking and correction of involuntary eye movement can improve eye care, by mitigating distortion in retinal image captured with scanning ophthalmoscopes, in particular, for underserved populations, such as, children and subjects with nystagmus [24], the most extreme form of involuntary eye movement [1,25]. Here we describe a low latency pupil tracking approach for monitoring and, eventually, correcting physiological and pathological eye movement.

Pupil tracking using traditional digital cameras, that is, cameras in which their output is an array of pixel values proportional to the number of detected photons during an exposure period, consists of three potentially overlapping steps. First, the capture of an image of the front of eye and its surrounding area, which is then downloaded to a computer for estimating the desired pupil parameters. A paradigm in which these steps are sequentially executed has been demonstrated with a variety of optical setups, cameras, computing platforms, and image processing algorithms (e.g., [26–28]). In an alternative paradigm capable of delivering lower latencies, image processing

#541421 https://doi.org/10.1364/BOE.541421
Journal © 2024 Received 6 Sep 2024; revised 30 Oct 2024; accepted 5 Nov 2024; published 13 Nov 2024

Biomedical Optics EXPRESS

starts before all image pixel values have been downloaded to the computer. Unfortunately, this paradigm with partially overlapping download and processing cannot be easily implemented with commercially available cameras, often requiring expensive, highly specialized computing hardware and computer programming. We previously demonstrated such approach using a field-programmable gate array (FPGA) and a central processing unit (CPU) to deliver calculation latencies as low as 0.35 ms [29], with commercial off-the-shelf components totaling >$15,000, and an FPGA that required complex software development. Here we seek comparable latencies using the sequential pupil tracking paradigm with an order or magnitude lower cost using end-user GPUs, while also reducing software complexity by using a high level interpreted programming language for image processing (Python). The Nvidia (Santa Clara, CA, USA) GPUs used here, have been widely used in scientific applications, including pupil tracking using CUDA, which requires manual compiling and more advanced programming expertise than Python [30–32]. The key innovation here is the optimization of the software for computers with integrated CPU and GPU with shared RAM, a computer architecture that allows creating image processing pipelines that take advantage of both types of processors without data copying. The use of Python has the added benefit that the pupil tracking software can be deployed to multiple computing platforms without modification or needing a computer-dependent compilation process.

The rest of the manuscript is structured as follows. First, we describe the hardware in two pupil tracking setups, followed by descriptions of the pupil tracking algorithms and their optimization. Then, after a short section on the human subject participation, we review the results of various tests and experiments, concluding with a brief summary.

2. Hardware

Two pupil tracking optical setups with different magnifications were built for use with two cameras each and various computing platforms, all of which are described next.

2.1. Cameras

Two cameras with complementary metal-oxide-semiconductor (CMOS) sensors and USB3 interfaces were selected to capture monochrome images of the front of the eye. This interface is available on most current computing platforms and operating systems, therefore not requiring specialized electronics, such as a frame grabber, which add cost, latency, and complexity. The cameras were an acA640-750um by Basler AG (Ahrensburg, Germany) with 4.8 µm square pixels, and a BFS-U3-20S4M-C by Teledyne FLIR (Wilsonville, Oregon, USA) with 4.5 µm square pixels. These cameras were chosen for their relatively high frame rate, relatively low cost, and global shutter to avoid distortion due to eye movement.

2.2. Illumination

The front of the eye was incoherently illuminated with two light emitting diodes (LEDs). The selection of the LED central wavelength was guided by the following considerations: avoiding interference with psychophysical experiments or vision tests, avoiding potential crosstalk with the imaging wavelengths of most commercial ophthalmoscopes, minimizing the potential for retinal damage, reducing ethnic bias, and the availability of LEDs with the desired brightness per unit of solid angle. The first consideration excludes the visible range (400 to 700 nm), while the second limits the wavelength range to either between 900 and 1,000 nm or >1,150 nm, as most widely used optical coherence tomographers operate with central wavelengths around 850 and 1060 nm, and the fluorescent emission of indocyanine green (ICG), used in eye clinics for retinal fluorescence angiography, extends up to ~900 nm. Wavelengths longer than 1,150 nm were not considered due to the high cost of the SWIR cameras that would be required.

We then coarsely try to estimate light safety in the 900 to 1,000 nm range by calculating the product of the maximum permissible exposure (MPE) as a function of wavelength and the camera

Biomedical Optics EXPRESS

relative spectral sensitivity, as shown in Fig. 1. The left plot in this figure shows the retinal MPE for an exposure >10 s to a continuous wave point source focused on the retina, according to the American National Stanford for safe use of lasers ANSI Z136.1-2014 [33]. The plot in the middle panel shows the relative quantum efficiency of the camera sensors, which is the same for both cameras, and the plot on the right shows their product, which tells us that for wavelengths longer than 900 nm, the shorter the wavelength the better the light safety. This is a coarse estimation because it ignores the variation of iris reflectivity and back scattering properties across eyes.

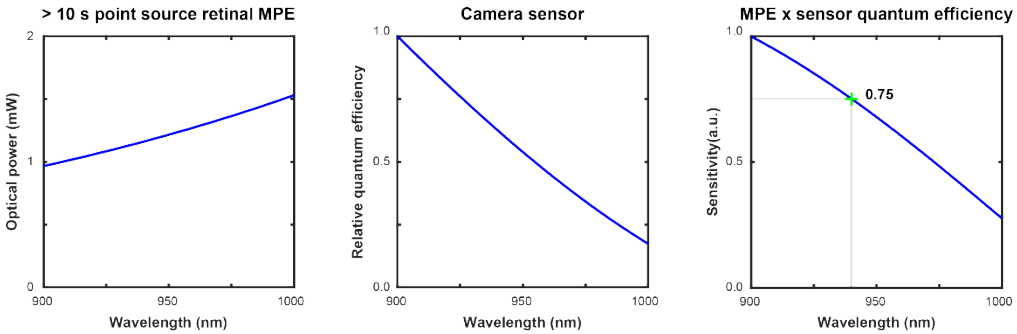


Fig. 1. Light safety considerations for pupil tracking in relation to camera sensitivity. The left plot shows the retinal maximum permissible exposure (MPE) as a function of wavelength according to the ANSI standard Z136.1-2014, when light from a point source is focused on the retina for >10 s. The middle plot shows the camera sensor relative quantum efficiency, and the right plot shows the product of the left and middle plots.

With this information, we chose 940 nm light due to the availability of LEDs with the necessary brightness, the almost negligible cone photoreceptor stimulation, with the photopic spectral luminous efficiency function being smaller than 10^{-6} [34], and despite the ~ 25% worse safety than 900 nm. This wavelength also appears to have a low potential for ethnic bias, which unfortunately, is all too common in the processing and analysis of images from human subjects (e.g., [35–37]), as illustrated by the images in Fig. 2. These images show little variation with iris and skin pigmentation and provide consistently high contrast between the iris and the pupil, unlike visible wavelengths.

Two LEDs (model LED940E, Thorlabs, Newton, NJ, USA) were positioned to the left and right of the optical axis of the pupil tracker imaging optics and tilted so that their maximal emission axes pointed to the center of the left and right halves of the field of view. This arrangement creates a more uniform intensity profile than if a single LED had been used. This off-axis illumination produces images in which the pupil of the eye appears dark with two vertically aligned corneal or scleral reflections [38,39].

The use of two LEDs ensures that their retinal irradiance is spread across two areas, with their centers separated by ~25° of visual angle, providing better light safety than a single LED. The maximum permissible exposure (MPE) for the continuous illumination of the retina, assuming the worst case scenario where the light entering the fully dilated pupil of the eye could be focused to a point for longer than 10 s, is 1.16 mW per LED (see Appendix A). When operated at their nominal 100 mA, each LED delivers ~0.28 mW to a circular area ~9 mm in diameter, which is larger than a pharmacologically dilated human pupil, placed at the eye's nominal position. If the operating current is increased beyond 100 mA, the optical power at the eye never exceeds ~0.63 mW with this output power decreasing by itself over a span of seconds due to LED overheating which, incidentally, serves as a desirable safety feature. A square metal plate with a circular hole was placed in front of the pupil tracker, to prevent the eye from accidentally getting close to the LEDs, as an additional light safety measure.

Biomedical Optics EXPRESS

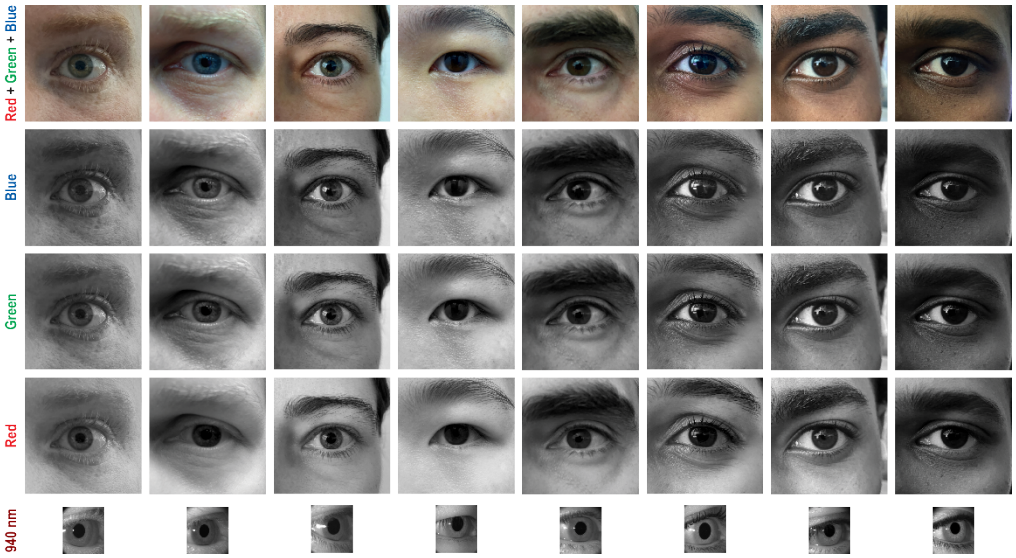


Fig. 2. Images captured with visible (top rows) and infrared (bottom row) light, showing that under 940 nm light there are no major contrast differences due to iris, hair or skin pigmentation.

2.3. Optical setup

Two optical setups were designed to relay a rectangular area ~18 × 21.6 mm onto the camera sensor, with two different magnifications. An 18 mm square field of view can contain a pupil with a diameter of an up to 8 mm within approximately ±25° of gaze changes, provided the head remains fixed in relation to the pupil tracker. An additional ~20% larger field of view along the horizontal axis is necessary for the pupil edge detection algorithm, which operates along the image horizontal axis and requires a few pixels to the left and right of the pupil edge. The optical setup was designed to operate at 45° angle of incidence with respect to the subject's face to facilitate integration with visual stimuli, vision testing and/or retinal imagers. In order to keep the entire field of view in focus, the camera sensor was tilted according to Scheimpflug's principle [40]. The optics were designed to achieve telecentricity in both object and image space to mitigate pupil image size changes due to axial head translation and/or camera focusing errors.

In order to facilitate switching between the two desired image magnifications, both optical setups have the same front portion. This was achieved by simultaneously optimizing both optical setups using the multi-configuration feature of the ray tracing software OpticStudio (Ansys Zemax, Kirkland, WA, USA). The ray tracing optimization aimed to minimize the default merit function (wavefront RMS), modified with custom operands to constrain the total system length (<300 mm), ensure adequate mechanical clearances, and maintain telecentricity (<0.3°). In the early design phases, all surface radii of curvature and airspaces were allowed to vary within constraints to maintain realistic lens shapes. These optimal lenses were then replaced with similar off-the-shelf lenses while maintaining diffraction limited performance at 940 nm. Once finalized, the airspaces between elements were adjusted for ease of assembly using off-the-shelf mechanical lens tubes, resulting in the values shown Table 1. One of the assembled optical setups is shown in Fig. 3 below, with complete bills of materials and mechanical drawings provided in Appendix B.

An interferometric optical filter that only transmits light in the 925-975 nm range was placed in the optical setup to ensure that only light from the illumination LEDs reaches the camera sensor, making the pupil tracking insensitive to room lighting.

Biomedical Optics EXPRESS

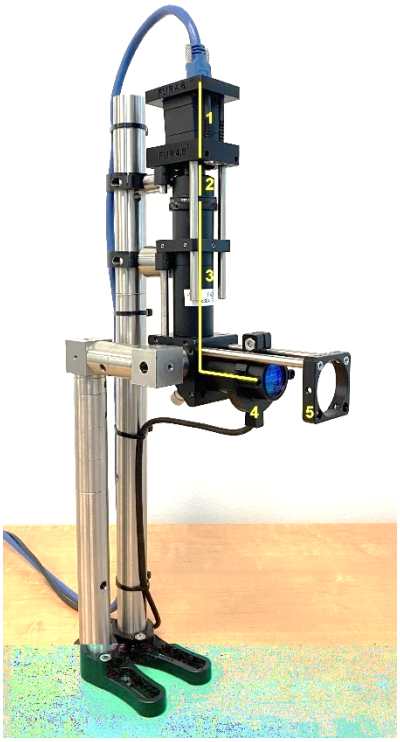


Fig. 3. Complete pupil tracker optical setup: 1) tilted FLIR camera; 2) and 3) interchangeable and fixed portions of the optical setup, respectively, 4) LED illumination and 5) mechanical mount to prevent accidental proximity of the eye to the LEDs.

Table 1. – Separation and tilt of optical elements for high frame rate and high resolution pupil trackers.

Element	Part # (manufacturer)	Separation (mm)	Tilt (°)
Eye		100.0	45.0
Lens 1	AC254-150-B (Thorlabs)	100.5	0.0
Lens 2	LF1822-B (Thorlabs)	40.9	0.0
Bandpass filter	84-792 (Edmund Optics)	36.0	0.0
Iris diaphragm	SM1D12C (Thorlabs)	8.7 / 8.0	0.0
Lens 3	LA1433-B / LBF254-075-B (Thorlabs)	3.4 / 22.8	0.0
Lens 4	LA1708-B (Thorlabs) / 49-326 (Edmund Optics)	8.2 / 27.0	0.0
Lens 5	AC127-025-B (Thorlabs) / -	17.8 / -	0.0
Camera	See main text		4.6 / 7.2

Biomedical Optics EXPRESS

The optomechanical components closest to the eye, common to both optical setups, include two lenses, a fold mirror, an interferometric bandpass filter, and an iris diaphragm (the system's aperture stop). Thus, to change magnifications, only the portion of the optical setup closest to the camera needs to be switched. Nominal optical distortion across the field of view is ~0.4%, when ignoring refraction at the ocular surfaces. Only points within ~1 mm of the field of view corners (at the eye) are vignetted, resulting in less than ~35% image intensity reduction, which is corrected by digital field-flattening of the pupil images.

2.4. Computing processing units

The pupil tracking algorithms were tested on four different computing platforms. Two desktops with discrete CPU and GPU, the first of which has an i9-13900 CPU by Intel (Santa Clara, CA, USA) with 128 GB of DDR5 random access memory (RAM) operating at 2,400 MHz, and an RTX A2000 GPU by Nvidia with 16 GB of RAM. The second desktop computer had a Ryzen 7 5800X CPU by AMD (Santa Clara, CA, USA) with 16GB of DDR4 RAM operating and an RTX 3070 Nvidia GPU with 8 GB of RAM. Both desktop computers were evaluated with the operating system Windows 11 (Microsoft, Santa Clara, CA, USA), and the AMD computer was also evaluated with Ubuntu Linux 22.04. Two Nvidia computers with integrated CPU and GPU that share RAM were also evaluated, seeking to reduce pupil tracking calculation latency by minimizing data copying. These computers were an AGX Xavier with an ARM Carmel CPU (v8.2) and a Volta GPU, and a Jetson Orin Nano with an ARM Cortex-A78AE CPU (v8.2), and an Ampere GPU. These devices, which cost ~$1,000 and $500, respectively, were evaluated using Ubuntu Linux 20.04 (Nvidia Jetpack 5.1) in the AGX Xavier and 22.04 (Nvidia Jetpack 6.0) in the Orin Nano.

3. Algorithms

The algorithms used here were conceived to detect the pupil of an eye in a field of view that does not include facial features beyond the eyelashes and eyebrows, as well as to cope with the pupil partially blocked by eyelids and eyelashes. The pupil position, size and orientation were determined by fitting an ellipse to the spatial coordinates of pixels considered to be at the pupil edge, using a previously described sequence of algorithms [29].

The sequence of algorithms executed in CPU or CPU-GPU is depicted in Fig. 4. The first algorithms mitigate camera sensor fixed pattern noise through background subtraction, correct for non-uniform illumination and vignetting through field flattening, reduce spurious edges through 1-dimensional (horizontal) filtering, and converted to a binary image through thresholding. Left pupil edge candidates are then found based on having a user-defined number of dark values to their left and a number of user-defined bright pixels to their right, while the reverse condition is used to find right pupil edges. After discarding left pupil edges without a matching right edge and vice versa, outlying pupil edge candidate pairs are also discarded based on a median fit to the more vertical of the two ellipse axes [29]. After repeating the median fit discarding a second time, an additional discarding of outliers is performed through clustering, before the least-squares ellipse fitting of the remaining pupil edge candidates.

No attempt was made to convert pupil coordinates to gaze angle, as this has been extensively addressed in the literature [41,42]. This conversion only requires a handful of multiplications with negligible execution time when compared with the hundreds of thousands required for pupil tracking itself.

All algorithms were implemented in Python 3.11, with the optimizations described next to reduce calculation time. The only code that was written in C++ was for acquiring camera images and passing them to the Python interpreter as Numpy arrays.

Biomedical Optics EXPRESS

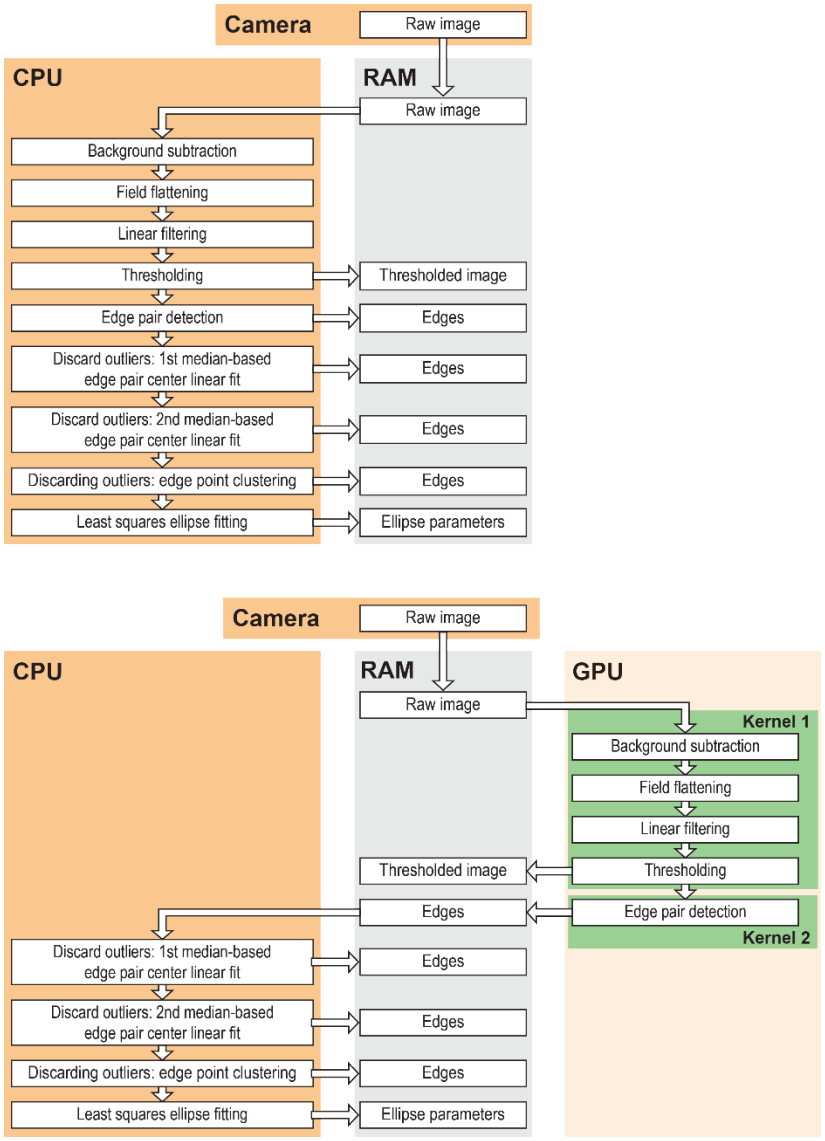


Fig. 4. Pupil tracking data processing flow through evaluated computing platforms. Note the algorithms executed in GPU have been implemented in only two kernels for faster execution. These diagrams can be compared with that on an FPGA-CPU pupil tracker (Fig. 2 of [29]).

3.1. CPU optimization

All mathematical operations executed in CPU were formulated, when possible, as element-by-element matrix operations to exploit optimizations in the mathematical software Numpy. All algorithms were evaluated with and without Numba decorators. Numba is a just-in-time (JIT) compiler that translates a subset of Python and Numpy functions into machine code [43]. All data buffers with fixed size were reused to reduce the need for dynamic memory allocation.

Biomedical Optics EXPRESS

3.2. GPU optimization

Only the naturally parallelizable algorithms were implemented and optimized for execution in GPU, including those that operate independently on single pixels (background subtraction, field flattening, thresholding), or on a pixel and its neighbors (filtering and edge detection). These algorithms were first implemented using CuPy, a library for GPU-accelerated mathematical computing with Python, which can be thought of as limited version of Numpy for GPUs. While having each operation contained within a separate CuPy kernel (i.e., a function that is executed simultaneously in multiple parallel threads) is the preferred approach for code clarity and code modularity, it is not always optimal in terms of execution time. This is because the launching of CuPy kernels requires performing some initial tasks, including input type checking and retrieval of the compiled kernel for the appropriate input data types. Because in the pupil tracking, the data types do not change over time, type checking is avoided by keeping handles to the necessary kernels. Additional kernel launching time is saved by consolidating the background subtraction, field flattening, filtering and thresholding into a single kernel. Although not guaranteed, this consolidation might also (non-deterministically) save some time by reducing the internally required synchronizations and data fetching from RAM into GPU.

3.3. CPU-GPU integration

Some computing platforms, such as those by Nvidia used here, have CPUs and GPUs integrated through physically shared RAM, which allows the bidirectional transfer of data between these processors without the need for copying. This was implemented by keeping raw images, processed images, background data, field flattening data and pupil edge candidates in pinned memory with pointers wrapped for both NumPy and CuPy arrays.

4. Human subjects

Nine adult subjects (5 male, 4 female) with no known ocular anterior segment pathology were recruited for this study, after obtaining informed written consent following the tenets of the Declaration of Helsinki and under a protocol approved by the institutional review board of Stanford University. The subjects were aligned and stabilized in front of the pupil tracker using a chin and forehead rest combination that is common in ophthalmic instruments.

5. Tests and results

5.1. Feasibility and repeatability

The four combinations of cameras and optical setups with different magnification were tested to evaluate whether pupil tracking could be achieved when capturing and streaming images at their maximum frame rate with safe light levels, and if so, with which precision (repeatability). Sequences of at least 600 images of a piece of paper with a 6 mm diameter black circle as a model pupil were recorded, together with the parameters of the fitted ellipses. These sequences were captured using three camera gains to test whether the electronic amplification of detected photon-electrons is beneficial or detrimental for pupil tracking, as we have previously shown that it is not necessary to use 256 grey levels for reliably tracking the pupil of the eye [29].

Both cameras were used in free run mode and set to their fast data readout mode, with the frame rates, gain and regions of interest shown in Fig. 5, together with the first image of each sequence. The text in the images also shows their approximate grey level range and the standard deviation of the estimated ellipse center. Under all experimental conditions, the pupil tracking precision is better (smaller) than 4.5 µm at the eye, which is approximately 1.4 arcmin of visual angle, confirming that precise pupil tracking can be achieved with safe light levels. The precision measured here is poorer than that shown in our previous work [29], due mostly to the use of cameras with different sensors, rather than the algorithms or the precision of the computing

Biomedical Optics EXPRESS

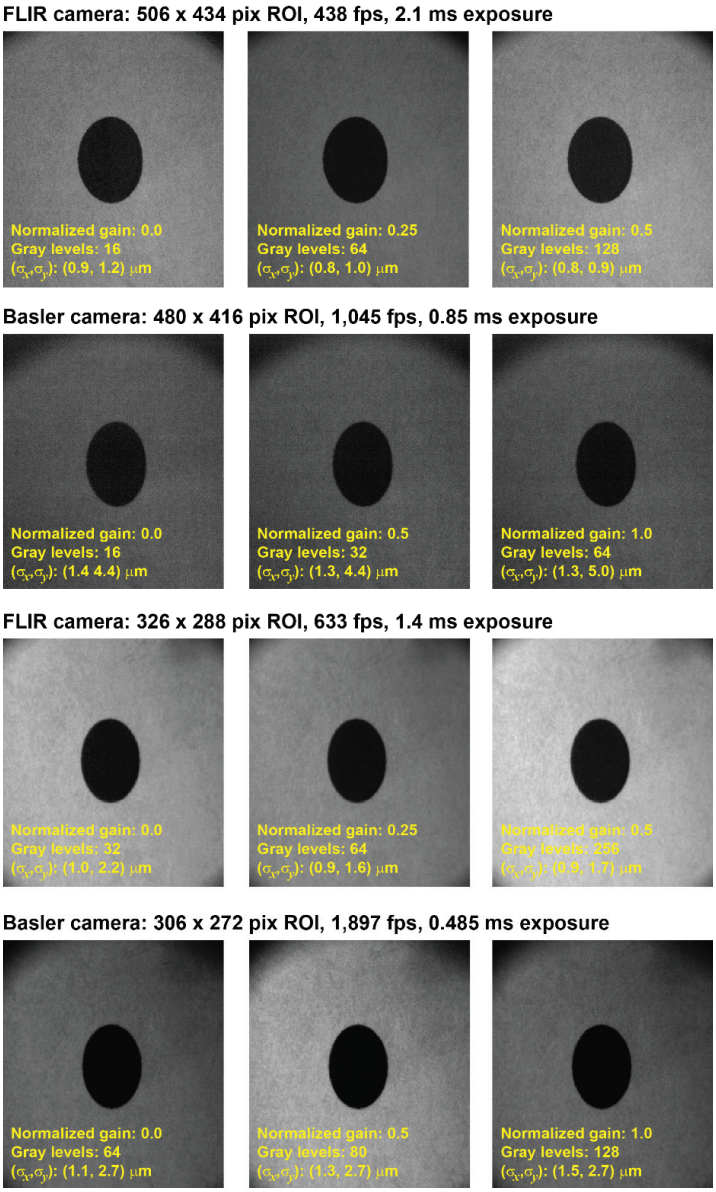


Fig. 5. Pupil tracker images (contrast-stretched for display purposes) of a piece of paper with a printed black circle (6 mm in diameter), captured with all four combinations of optical setups and cameras using less than 0.25 mW/LED through an 8 mm pupil, which is below the 1.16 mW/LED maximum permissible exposure assuming that the light from each LED is focused on the retina to a point, for over 10 s. The term gain here refers to the camera analog to digital converters, that has been normalized to between 0 and 1. "Gray levels" refers to the range of grey levels used by the pixels in the images, which have been contrasted here for display purposes. The Greek letter sigma denotes fitted ellipse center standard deviation along horizontal (x) and vertical (y) axis.

Biomedical Optics EXPRESS

Table 2. Pupil tracking algorithm timing across various computing platforms, operating systems and algorithm optimizations.

Python & NumPy | Numba | CuPy

	Operating system	Computing module	Bg+FF+filter+ threshold (us)	Edge detection (us)	Outlier discarding (us) Median	Outlier discarding (us) Clustering	Ellipse fitting (us)	Total (us)	Frame rate (Hz)
Image size: 306 × 272 pix	MS Windows	Intel CPU	637	2,005	90	31	15	2,778	360
			8,397	2,521	30	13	3	10,964	91
			637	2,005	30	13	3	2,688	372
		Intel CPU + Nvidia GPU	37	212	30	14	4	297	3,367
		AMD CPU	1,097	2,955	168	40	35	4,295	233
			8,313	3,405	28	15	4	11,765	85
			1,097	2,955	28	15	4	4,099	244
		AMD CPU + Nvidia GPU	45	443	28	16	5	537	1,862
	Linux	AMD CPU	729	1,673	143	34	21	2,600	385
			4,820	3,373	19	11	3	8,226	122
			729	1,673	19	11	3	2,435	411
		AMD CPU + Nvidia GPU	40	391	20	11	3	465	2,151
		Nvidia AGX CPU	1,345	6,832	743	145	128	9,193	109
			10,988	11,251	109	60	24	22,432	45
			1,345	6,832	109	60	24	8,370	119
		Nvidia AGX CPU + GPU	249	362	101	58	22	792	1,263
		Nvidia Nano CPU	1,717	5,700	501	94	72	8,084	124
			18,086	11,467	70	31	10	29,664	34
			1,717	5,700	70	31	10	7,528	133
		Nvidia Nano CPU + GPU	213	445	69	31	10	768	1,302
Image size: 480 × 416 pix	MS Windows	Intel CPU	1,875	4,299	88	45	15	6,322	158
			20,409	6,137	42	18	4	26,610	38
			1,875	4,299	42	18	4	6,238	160
		Intel CPU + Nvidia GPU	49	386	46	21	4	506	1,976
		AMD CPU	3,029	6,414	166	58	24	9,691	103
			19,900	8,088	42	20	5	28,055	36
			3,029	6,414	42	20	5	9,510	105
		AMD CPU + Nvidia GPU	82	1,365	40	50	5	1,542	649
	Linux	AMD CPU	1,724	3,236	151	49	22	5,182	193
			11,623	8,374	28	16	4	20,045	50
			1,724	3,236	28	16	4	5,008	200
		AMD CPU + Nvidia GPU	66	1,003	28	16	4	1,117	895
		Nvidia AGX CPU	3,049	11,965	722	204	128	16,068	62
			25,064	14,956	133	75	26	40,254	25
			3,049	11,965	133	75	26	15,248	66
		Nvidia AGX CPU + GPU	402	387	119	77	25	1,010	990
		Nvidia Nano CPU	4,029	10,819	527	138	73	15,586	64
			43,432	28,488	102	46	12	72,080	14
			4,029	10,819	102	46	12	15,008	67
		Nvidia Nano CPU + GPU	462	656	102	44	12	1,276	784

platform. Unfortunately, the previous cameras were not available with USB3 interface, and the ones used here represented a compromise between high frame rate, cost and sensitivity. Importantly, the small variation of precision with camera gain indicates that this parameter does not need to be fine-tuned, thus simplifying the pupil tracking operation.

5.2. Calculation timing

The pupil tracking algorithms were executed with the processor clocks set to their maximum value to both achieve minimum calculation times and minimize non-determinism. All algorithms

Biomedical Optics EXPRESS

were executed at least once before recording timing, to exclude the just-in-time compilation that takes place when Python, Numba or CuPy code is executed for the first time. The timing values, reported in Table 2, are averages across a minimum of 1,000 repetitions. Non-determinism in Linux was mitigated by allocating the pupil tracking application to pre-selected CPU cores from which the operating system was excluded.

A first examination of Table 2 reveals that most of the pupil tracking calculation time (>77%) is taken by the image processing and pupil edge detection algorithms, which execute 5-38 times faster in GPU than in CPU. In CPU, these algorithms execute faster using Python/Numpy than their Numba counterparts, while the edge outlier discarding, and ellipse fitting algorithms are 2-6 times faster with Numba. Therefore, CPU-only pupil tracking can achieve minimal calculation times by using Python/Numpy implementations for image processing and edge finding, and Numba implementations for the remaining algorithms. In computing platforms with both CPU and GPU, then CuPy and Numba implementations provide the lowest calculation latencies. The more powerful desktop CPUs and discrete GPUs can perform pupil tracking in as little as 0.5 ms, while the embedded units require 0.8-1.3 ms.

Interestingly, the AMD CPU, whether complemented by a GPU or not, deliver faster calculation times (>30%) using the Linux operating system, compared to Windows. Also, the more powerful desktop CPUs evaluated here delivered 2-3 times faster calculation times than the integrated ones, which is expected given their larger number of processor cores and faster internal clocks.

6. Summary

Camera-based pupil tracking was explored in computing platforms with both discrete and integrated CPUs and GPUs, seeking low calculation latencies. The implementation of the algorithms used for pupil tracking, originally developed for real-time FPGA processing of data streamed from a camera, were optimized for CPU, discrete CPU-GPU combination, and a new end-user integrated CPU-GPU computing platform using a high level interpreted programming language.

Two telecentric Scheimpflug optical setups were designed and built with commercial of-the-shelf optical and mechanical components for imaging the front of the eye using 940 nm light. It was shown that, at this wavelength, image contrast does not vary substantially with iris and skin pigmentation, indicating low potential for pupil tracking ethnic biases, when compared to visible light.

The algorithms used for this work were implemented in Python and only compiled by the Python or Numba just-in-time compilers at first execution. This makes the proposed pupil tracking software platform-independent, only limited by the availability of Numpy, Numba, CuPy and camera drivers for the desired computer platform and operating system. The only code that was written in C++ and compiled was the camera software, which returns the images as Numpy matrices.

Pupil tracking with a test target and safe light levels was demonstrated using two different CMOS cameras with USB3 interface and global shutter, operating at 438 and 1,045 fps when using an ~500×420 pix ROI, and at 633 and 1,897 fps when using an ~315×280 pix ROI. Despite the images of the target spanning only a fraction of the 8-bit image depth, the tracking precision for a 6 mm diameter model pupil was lower than 4.5 µm at the eye, reaching values as low as 0.9 µm depending on camera exposure.

Timing across various computing platforms showed that powerful desktop CPUs delivered calculation times as low as 0.5 ms, while the embedded units require 0.8-1.3 ms, which is reasonable given that the former have a larger number of processor cores (both in the CPUs and GPUs) and faster internal clocks. The relatively low calculation times on the embedded units, however, shows promise not only for pupil tracking, but also for other ophthalmic instrumentation given their relatively low cost, small dimensions and low power consumption. The timing

Biomedical Optics EXPRESS

data indicates that most of the pupil tracking calculation time is taken by the image processing and pupil edge detection algorithms, which execute many times faster in GPU than in CPU, confirming the value of the CPU-GPU combination for pupil tracking. An important timing observation was that the exact same computer can deliver substantially different pupil tracking calculation times (30%) depending on which operating system is used.

The programming required for this work illustrates how high-level mathematical libraries such as CuPy, can facilitate taking advantage of the highly parallel GPU architecture with minimal understanding of low-level hardware operation. Similarly, just-in-time compilers such as Numba allow optimization of non-parallelizable algorithms with minimal programming.

In summary, the CPU-GPU approach demonstrated here is well suited for tracking the pupil with precision comparable and/or better than that of current pupil trackers and with comparable or lower latency [29]. The precision and low latency achieved make the device suitable for improving eye care through real-time eye movement compensation in retinal imaging, retinal functional testing, retinal laser treatment and refractive surgery, in particular, in subjects with eye movement disorders, such as, nystagmus.

Appendix A: light safety calculations

The pupil tracker illumination reaches the iris of the eye, the sclera, the cornea, the crystalline lens, the retina and the skin around the eye. Therefore, maximum permissible exposures (MPEs) for these tissues were calculated using the American National Stanford for safe use of lasers ANSI Z136.1-2014 [33], assuming exposures to a continuous wave 940 nm light source for longer than 10 s.

According to this standard, the iris of the eye, the sclera and the skin around the eye are all considered skin, which for a 940 nm light has an MPE given by $0.2 \times 10^{0.002(940-700)}$ W/cm^2 (~0.6 W/cm^2). When the eye is at the nominal 100 mm from the pupil tracker, the illuminating LEDs deliver ~0.5 mW/cm^2 (measured), which is ~1,000 times lower than the skin MPE. For the cornea and crystalline lens, 0.5 mW/cm^2 is ~8,000 times below the MPE, which is 4.0 W/cm^2.

For the retina MPE calculation, we assumed the use of a head restrain, such as a bite bar or chin/forehead rest, intentional fixation, and light focused to a point. In reality, the LED light will only come to a focus on the retina whenever the refractive error of the eye is the inverse of the distance between the LED and the eye (10 D), and thus, for the vast majority of eyes, the LED light would be spread over an area substantially larger than what the standard considers a point (a circle 1.5 mrad of visual angle in diameter). Also, the LED emitting area is not a point, and thus, this alone will spread the LED light over an area larger than that assumed by the standard for a point source. For the conservative conditions assumed here, the retinal MPE is $10^{0.002(940-700)}$ mW/cm^2 times the area of a pupil 0.7 cm in diameter, which is 1.16 mW per LED, because when focused on the retina they would create images separated by ~25° of visual angle.

Appendix B: bills of materials and mechanical drawings

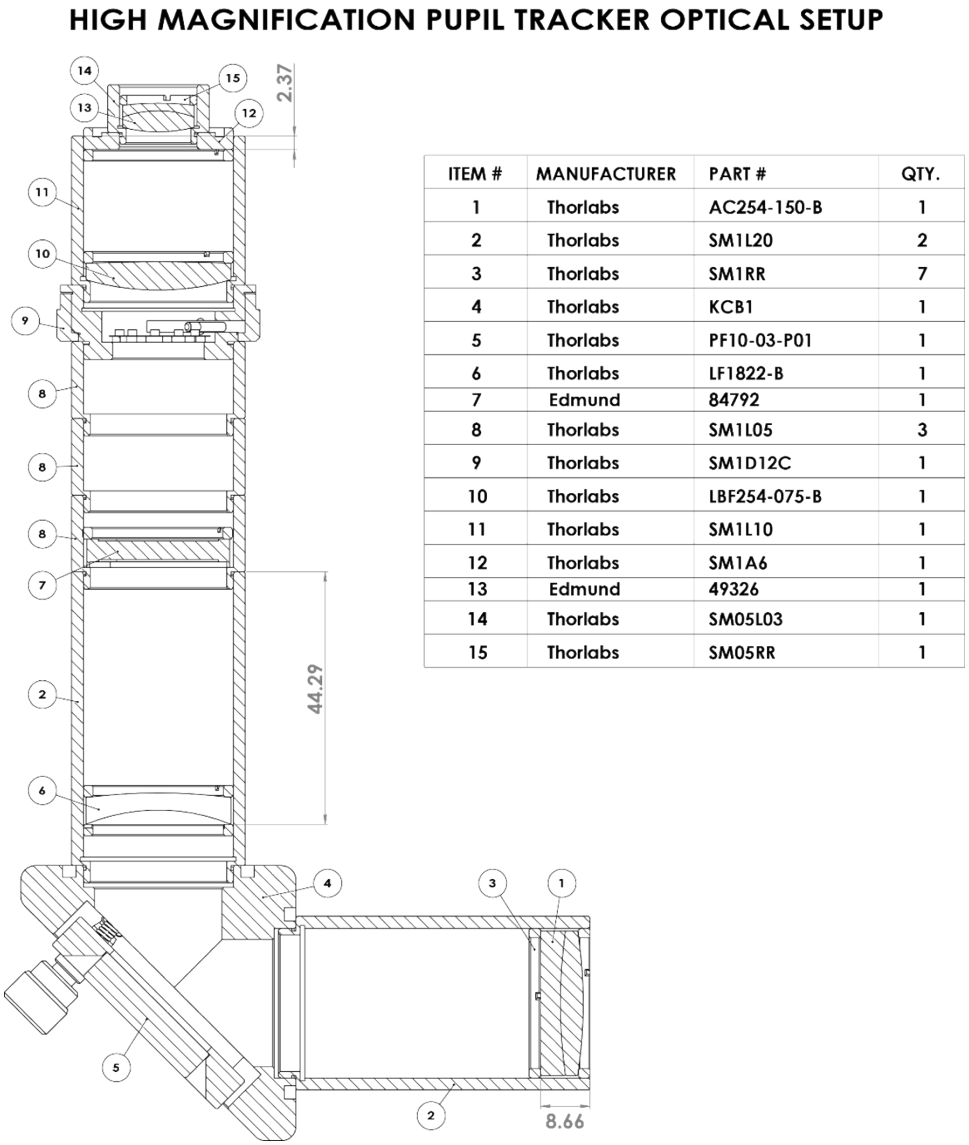
Imaging setup

Two imaging setups, with a common portion, relay an image of the eye onto a camera sensor, each with a different optical magnification. The assemblies and their components are shown in Fig. 6 and Fig. 7 below.

Camera mount

The pupil tracking cameras were secured by two 3D printed plates that clamped the front and back of the camera through three 8-32 or M4 screws and nuts. The clamped camera was attached and centered with respect to the optical setup by using four metal rods (Thorlabs, 6 mm cage system). Design file 1 and Design file 2 are the CAD drawings for mounted both cameras tilted

Biomedical Optics EXPRESS



ITEM #	MANUFACTURER	PART #	QTY.
1	Thorlabs	AC254-150-B	1
2	Thorlabs	SM1L20	2
3	Thorlabs	SM1RR	7
4	Thorlabs	KCB1	1
5	Thorlabs	PF10-03-P01	1
6	Thorlabs	LF1822-B	1
7	Edmund	84792	1
8	Thorlabs	SM1L05	3
9	Thorlabs	SM1D12C	1
10	Thorlabs	LBF254-075-B	1
11	Thorlabs	SM1L10	1
12	Thorlabs	SM1A6	1
13	Edmund	49326	1
14	Thorlabs	SM05L03	1
15	Thorlabs	SM05RR	1

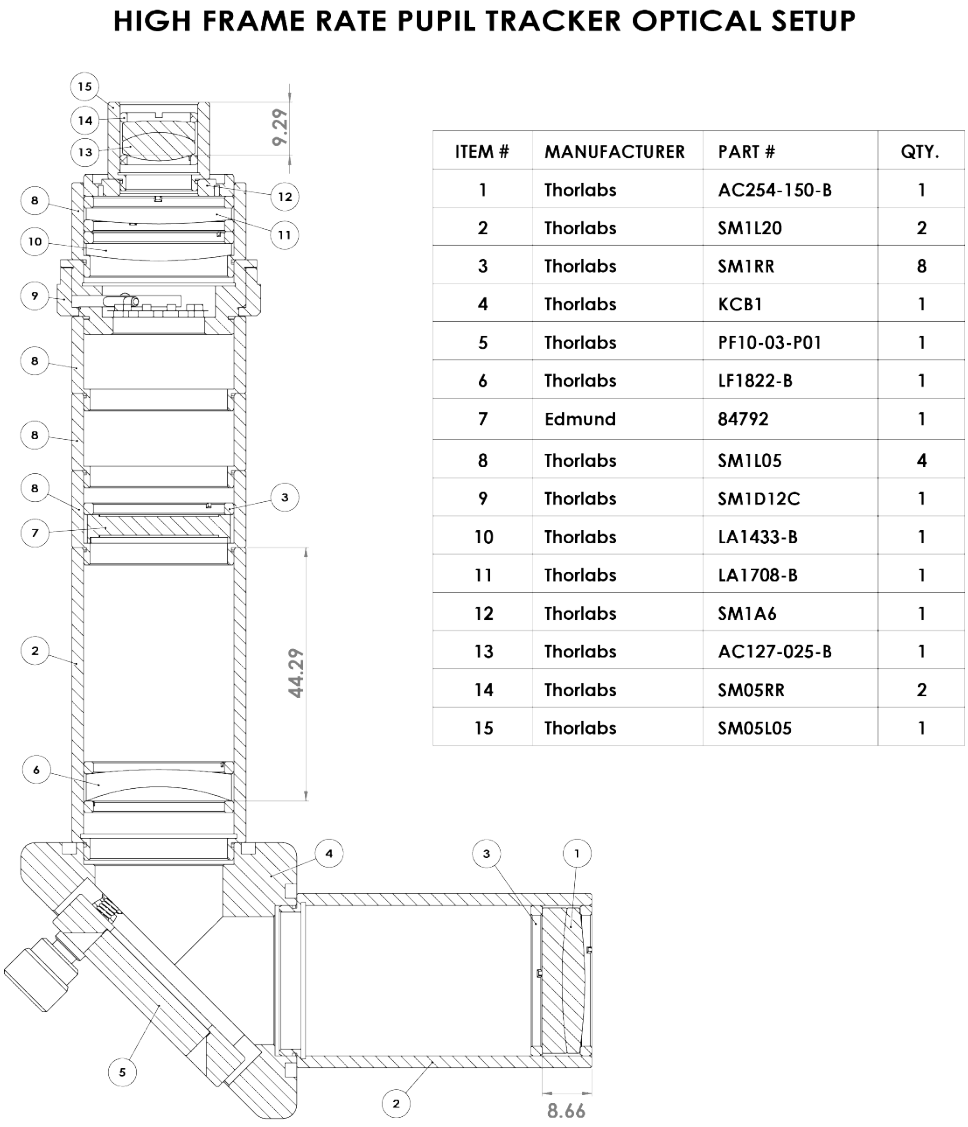
Fig. 6. Imaging portion of the high magnification pupil tracker setup and its components. Note the distances listed in the drawing can be measured with a caliper during the assembly process.

by 4.6°, for the high frame rate optical setup, while Design file 3 and Design file 4 are for the camera tilted by 7.24°, for the high magnification setup.

Illumination

The LEDs were mounted on the optical setup using a custom 3D-printed mount (see Fig. 8) that is clamped onto a Thorlabs SM1 lens tube by tightening an M4 or 8-32 screw opposite to a nut. The two LEDs were soldered in parallel to a Thorlabs cable (CAB-LEDD1) and plugged to a Thorlabs LEDD1B T-cube power supply, set to a maximum 200 mA current and continuous wave (CW) operation. In this way, each LED receives a maximum electrical current of ~100 mA,

Biomedical Optics EXPRESS



ITEM #	MANUFACTURER	PART #	QTY.
1	Thorlabs	AC254-150-B	1
2	Thorlabs	SM1L20	2
3	Thorlabs	SM1RR	8
4	Thorlabs	KCB1	1
5	Thorlabs	PF10-03-P01	1
6	Thorlabs	LF1822-B	1
7	Edmund	84792	1
8	Thorlabs	SM1L05	4
9	Thorlabs	SM1D12C	1
10	Thorlabs	LA1433-B	1
11	Thorlabs	LA1708-B	1
12	Thorlabs	SM1A6	1
13	Thorlabs	AC127-025-B	1
14	Thorlabs	SM05RR	2
15	Thorlabs	SM05L05	1

Fig. 7. High frame rate pupil tracker optical setup and its components. Note the distances listed in the drawing can be measured with a caliper during the assembly process.

which is the nominal operating current. CAD assembly file for the LED mount and its cover is provided in Design file 5 and Design file 6.

Centering and focusing aid

A custom 3D-printed aid was designed and made to center and focus the pupil tracking camera. This aid, shown in Fig. 9 (Design file 7 and Design file 8), consists of a flat target with a crosshair and an 18 by 18 mm square to facilitate camera focusing and centration, as well as field of view calibration. The target mount is a tube cut along its axis of symmetry to allow for the pupil tracking LED light to reach the target, while holding it at the nominal distance and centered with respect to the optical axis. Figure 10 below shows how this centering and focusing aid is mounted on the optical setup.

Biomedical Optics EXPRESS

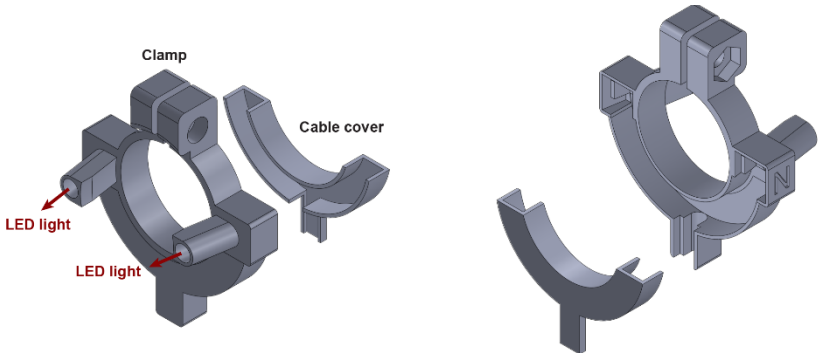


Fig. 8. Custom 3D-printed LED mount with cable cover and clamp for securing around Thorlabs SM1 lens tubes.

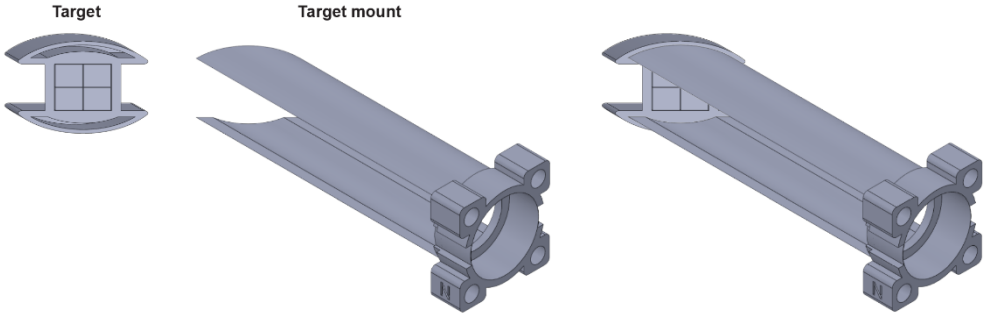


Fig. 9. Custom 3D-printed centering and focusing aid consisting of an 18 mm square target with a crosshair and a mount to place at the nominal distance from the pupil tracking optical setup.

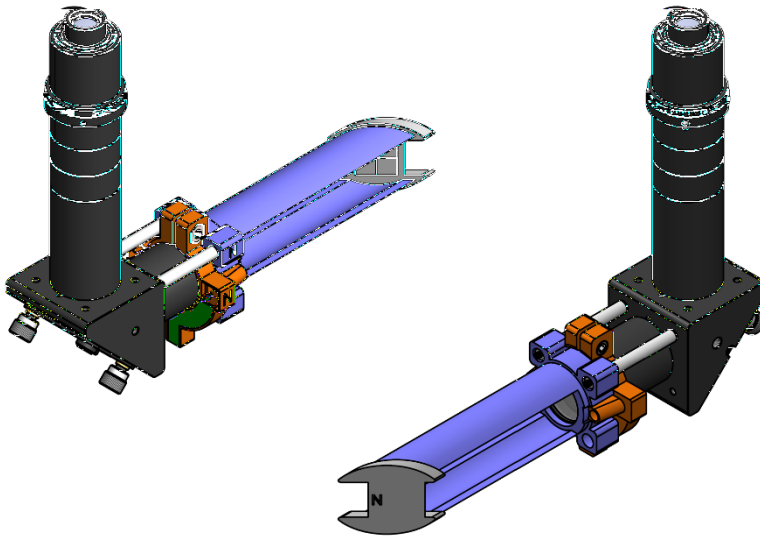


Fig. 10. Centering and focusing aid consisting of a target (light grey) and a mount (purple), which slides on the optical setup and is oriented using two metal alignment rods. This aid is directly adjacent to the LED mount (orange) with a back cable cover (green).

Biomedical Optics EXPRESS

Funding. National Institutes of Health (P30EY026877, R01EY031360, R01EY032147, R01EY032669); Research to Prevent Blindness (Departmental award).

Acknowledgments. We would like to thank David Buickians for the design of the camera mounts for the pupil tracking setups, Aubrey Hargrave for consenting the study participants and Gastón A. Ayubi for assembling the LED illumination circuits.

Disclosures. The authors declare no conflicts of interest.

Data availability. The raw de-identified data for this work may be obtained from the authors upon reasonable request with prior approval from Stanford's institutional review board (IRB) approval.

References

1. L. A. Riggs, J. C. Armington, and F. Ratliff, "Motions of the retinal image during fixation," J. Opt. Soc. Am. **44**(4), 315–321 (1954).
2. W. H. Hart Jr., *Adler's Physiology of the Eye: Clinical Application*, 9th ed. (Mosby-Year Book Inc., 1992).
3. N. Charman, "Optics of the Eye," in *Vision and Vision Optics*, 3rd ed., M. Bass, ed. (McGraw Hill, 2009).
4. H. O. Istance and P. A. Howarth, "Keeping an eye on your interface: The potential for eye-based control of graphical user interfaces (GUI's)," in *BCS HCI*, 1994), 195–209.
5. J. R. Bergstrom and A. Schall, *Eye Tracking in User Experience Design* (Elsevier, 2014).
6. A. Pasarica, R. G. Bozomitu, H. Costin, *et al.*, "Human-computer interface based on eye tracking with dwell time selection," in *2017 IEEE 23rd International Symposium for Design and Technology in Electronic Packaging (SIITME)*, 2017), 375–378.
7. M. Betke, J. Gips, and P. Fleming, "The Camera Mouse: visual tracking of body features to provide computer access for people with severe disabilities," IEEE Trans. Neural Syst. Rehabil. Eng. **10**(1), 1–10 (2002).
8. M. Adjouadi, A. Sesin, M. Ayala, *et al.*, "Remote eye gaze tracking system as a computer interface for persons with severe motor disability," in *Computers Helping People with Special Needs* (Springer Berlin Heidelberg, 2004), 761–769.
9. C. Jang, K. Bang, S. Moon, *et al.*, "Retinal 3D: augmented reality near-eye display via pupil-tracked light field projection on retina," ACM Trans. Graph. **36**(6), 1–13 (2017).
10. A. Yoshikaie, R. Ogawa, T. Imamura, *et al.*, "Full-color binocular retinal scan AR display with pupil tracking system," *SPIE AR | VR | MR* (SPIE 2023), Vol. 12449.
11. S. Lu, Y. P. Sanchez Perdomo, X. Jiang, *et al.*, "Integrating eye-tracking to augmented reality system for surgical training," Journal of Medical Systems **44**(11), 192 (2020).
12. L. J. Zheng, J. Mountstephens, and J. Teo, "Eye fixation versus pupil diameter as eye-tracking features for virtual reality emotion classification," in *2021 IEEE International Conference on Computing (ICOCO)*, 2021), 315–319.
13. sostel, "Eye tracking overview - Mixed Reality," (2023).
14. "Apple Vision Pro," Apple.
15. B. Sahin, F. Harms, B. Lamory, *et al.*, *A pupil tracking system for adaptive optics retinal imaging*, SPIE Photonics Europe (SPIE, 2008), Vol. 6991.
16. O. M. Carrasco-Zevallos, D. Nankivil, C. Viehland, *et al.*, "Pupil tracking for real-time motion corrected anterior segment optical coherence tomography," PLoS One **11**(8), e0162015 (2016).
17. P. N. Skonnikov and D. V. Trofimov, "Pupil visual tracking algorithms for automated static perimetry systems," Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci. **XLIV-2/W1-2021**, 195–199 (2021).
18. D. B. Henson and T. Emuh, "Monitoring vigilance during perimetry by using pupillography," Invest. Ophthalmol. Vis. Sci. **51**(7), 3540–3543 (2010).
19. V. Clay, P. König, and S. U. König, "Eye tracking in virtual reality," Journal of Eye Movement Research **12**(1) , (2019).
20. M. Mrochen, M. S. Eldine, M. Kaemmerer, *et al.*, "Improvement in photorefractive corneal laser surgery results using an active eye-tracking system," J. Cataract Refractive Surg. **27**(7), 1000–1006 (2001).
21. M. Bueeler and M. Mrochen, "Limitations of pupil tracking in refractive surgery: systematic error in determination of corneal locations," Journal of Refractive Surgery **20**(4), 371–378 (2004).
22. O. Carrasco-Zevallos, D. Nankivil, B. Keller, *et al.*, "Pupil tracking optical coherence tomography for precise control of pupil entry position," Biomed. Opt. Express **6**(9), 3405–3419 (2015).
23. S. Meimon, J. Jarosz, C. Petit, *et al.*, "Pupil motion analysis and tracking in ophthalmic systems equipped with wavefront sensing technology," Appl. Opt. **56**(9), D66–D71 (2017).
24. L. F. Dell'osso and R. B. Daroff, "Congenital nystagmus waveforms and foveation strategy," Doc. Ophthalmol. **39**(1), 155–182 (1975).
25. F. Ratliff and L. A. Riggs, "Involuntary motions of the eye during monocular fixation," J. Exp. Psychol. **40**(6), 687–701 (1950).
26. W. Fuhl, M. Tonsen, A. Bulling, *et al.*, "Pupil detection for head-mounted eye tracking in the wild: an evaluation of the state of the art," Machine Vision and Applications **27**(8), 1275–1288 (2016).
27. R. Rathnayake, N. Madhushan, A. Jeeva, *et al.*, "Current trends in human pupil localization: a review," IEEE Access **11**, 115836–115853 (2023).

Biomedical Optics EXPRESS

28. N. Min-Allah, F. Jan, and S. Alrashed, "Pupil detection schemes in human eye: a review," Multimedia Systems **27**(4), 753–777 (2021).
29. B. Kowalski, X. Huang, S. Steven, *et al.*, "Hybrid FPGA-CPU pupil tracker," Biomed. Opt. Express **12**(10), 6496–6513 (2021).
30. J. Mompeán, J. L. Aragón, P. Prieto, *et al.*, "GPU-Accelerated High-Speed Eye Pupil Tracking System," in *2015 27th International Symposium on Computer Architecture and High Performance Computing (SBAC-PAD)*, 2015), 17–24.
31. J. Mompeán, J. L. Aragón, P. M. Prieto, *et al.*, "Design of an accurate and high-speed binocular pupil tracking system based on GPGPUs," The Journal of Supercomputing **74**(5), 1836–1862 (2018).
32. J. B. Mulligan, "A GPU-accelerated software eye tracking system," in *Proceedings of the Symposium on Eye Tracking Research and Applications*, 2012), 265–268.
33. . ANSI, "American national standard for safe use of lasers ANSI Z136.1 - 2014," (2014).
34. J. J. Vos, "Colormetric and photometric properties of a 2 degree fundamental observer," Color Res. Appl. **3**(3), 125–128 (1978).
35. J. G. Cavazos, P. J. Phillips, C. D. Castillo, *et al.*, "Accuracy comparison across face recognition algorithms: where are we on measuring race bias?" IEEE Trans. Biom. Behav. Identity Sci. **3**(1), 101–111 (2021).
36. J. E. Kilbride and M. Yarczower, "Ethnic bias in the recognition of facial expressions," Journal of Nonverbal Behavior **8**(1), 27–41 (1983).
37. T. Xu, J. White, S. Kalkan, *et al.*, "Investigating bias and fairness in facial expression recognition," in *Computer Vision – ECCV 2020 Workshops* (Springer International Publishing, 2020), 506–523.
38. T. N. Cornsweet and H. D. Crane, "Accurate two-dimensional eye tracker using first and fourth Purkinje images," J. Opt. Soc. Am. **63**(8), 921–928 (1973).
39. J. Tabernero and P. Artal, "Lens oscillations in the human eye. Implications for post-saccadic suppression of vision," PLoS One **9**(4), e95764 (2014).
40. T. Scheimpflug, "Improved method and apparatus for the systematic alteration or distortion of plane pictures and images by means of lenses and mirrors for photography and for other purposes," GB patent **1196** (1904).
41. D. W. Hansen and Q. Ji, "In the eye of the beholder: A survey of models for eyes and gaze," 1 **32**(3), 478–500 (2010).
42. D. Su, Y. F. Li, and H. Chen, "Toward precise gaze estimation for mobile head-mounted gaze tracking systems," IEEE Trans. Ind. Inf. **15**(5), 2660–2672 (2019).
43. S. K. Lam, A. Pitrou, and S. Seibert, "Numba: a LLVM-based Python JIT compiler," in *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, (Association for Computing Machinery, Austin, Texas, 2015), Article 7, p. 1–6.