

# The Open Perimetry Initiative: A framework for cross-platform development for the new generation of portable perimeters

**Iván Marín-Franch**

Computational Optometry, Atarfe, Spain  
Southwest Eye Institute, Tavistock, UK



**Andrew Turpin**

School of Computing and Information Systems,  
University of Melbourne, Melbourne, Victoria, Australia



**Paul H. Artes**

Southwest Eye Institute, Tavistock, UK  
University of Plymouth, Plymouth, UK



**Luke X Chong**

School of Medicine (Optometry), Deakin University,  
Geelong, Australia



**Allison M. McKendrick**

Department of Optometry and Vision Sciences,  
University of Melbourne, Melbourne, Victoria, Australia



**Karam A. Alawa**

Departments of Neurology and Ophthalmology and  
Visual Sciences, University of Iowa, College of Medicine,  
Iowa City, Iowa, USA



**Michael Wall**

Departments of Neurology and Ophthalmology and  
Visual Sciences, University of Iowa, College of Medicine,  
Iowa City, Iowa, USA



The Open Perimetry Initiative was formed in 2010 with the aim of reducing barriers to clinical research with visual fields and perimetry. Our two principal tools are the Open Perimetry Interface (OPI) and the visualFields package with analytical tools. Both are fully open source. The OPI package contains a growing number of drivers for commercially available perimeters, head-mounted devices, and virtual reality headsets. The visualFields package contains tools for the analysis and visualization of visual field data, including methods to compute deviation values and probability maps. We introduce a new frontend, the opiApp, that provides tools for customization for visual field testing and can be used as a frontend to run the OPI. The app can be used on the Octopus 900 (Haag-Streit), the Compass (iCare), the AP 7000 (Kowa), and the IMO (CREWT) perimeters, with permission from the device manufacturers. The app can also be used on Android phones with virtual reality headsets via a new driver interface, the PhoneHMD, implemented on the OPI. The use of the tools provided by the OPI library is showcased with a custom static automated perimetry test for the full visual field (up to

50 degrees nasally and 80 degrees temporally) developed with the OPI driver for the Octopus 900 and using visualFields for statistical analysis. With more than 60 citations in clinical and translational science journals, this initiative has contributed significantly to expand research in perimetry. The continued support of researchers, clinicians, and industry are key in transforming perimetry research into an open science.

## Introduction

The advent of consumer electronics with high-quality display technology (Anthes, García-Hernández, Wiedemann, & Kranzlmüller, 2016) has paved the way for a new generation of portable devices for visual field testing. The transition from traditional projection perimetry to display-based perimetry requires rethinking and adaptation of conventional perimetry methods, and it also provides an opportunity to revise and improve.

Citation: Marín-Franch, I., Turpin, A., Artes, P. H., Chong, L. X., McKendrick, A. M., Alawa, K. A., & Wall, M. (2022). The Open Perimetry Initiative: A framework for cross-platform development for the new generation of portable perimeters. *Journal of Vision*, 22(5):1, 1–10, <https://doi.org/10.1167/jov.22.5.1>.



The Open Perimetry Initiative is an open-source project that started in 2010 with the goal of alleviating the difficulties of using commercial and experimental ophthalmic devices in vision research. The initiative has evolved beyond its original goal to include features that facilitate the development of new paradigms, standards, and good practices that exploit the technological advantages of portable devices. To maximize accessibility of novel perimetry methods and techniques, implementations using the Open Perimetry libraries should be an open source under the GNU General Public License. To avoid possible legal ramifications, permission from device manufacturers is required before the Open Perimetry Interface (OPI) code is used on their commercial instruments (Turpin, Artes, & McKendrick, 2012).

The key product of the Open Perimetry Initiative is the OPI (Turpin et al., 2012). The OPI not only provides drivers for an increasing number of devices, but it also sets standards and protocols for the implementation of custom visual field tests so that they can be run seamlessly on different instruments, with one implementation for many devices. Furthermore, the OPI can be run in simulation mode, so that new perimetric procedures can be implemented, debugged, and assessed before they are ported to the actual test device. The other key product of the Open Perimetry Initiative is the R (R Core Team, 2022) package `visualFields` (Marín-Franch & Swanson, 2013), a tool for the statistical analysis and visualization of perimetry results. Until now, the OPI and `visualFields` solutions have been developed largely independently from one another. However, the recently published `shiny` package (Chang, Cheng, Allaire, Xie, & McPherson, 2020; <https://shiny.rstudio.com>) makes it possible to develop cross-platform applications with frontends that integrate the OPI and `visualFields` software.

The purpose of this paper is to describe the recent advances in the OPI open-source library. A new interface, the PhoneHMD has been included for Android phones with headsets that are compatible with Google Cardboard. The `visualFields` package has been rewritten to include new interactive features and make it more compatible with the OPI drivers. A frontend app has also been developed that wraps the functionality provided by the OPI. This new frontend, the `opiApp`, consists of a series of tools to manage databases of study participants, generate custom perimetry grids, and run the same methods on different devices, including the Octopus 900, the AP-7000, and Compass perimeters, and headset devices such as the IMO and the Android phones. The use of these new tools is showcased with examples of published research and ongoing studies.

In addition to the development of novel paradigms, methods, and software, the optical properties of new

devices need to be characterized. This includes spatial and grey level resolution along with the effects of chromatic and achromatic aberrations. Further, it is necessary to develop adequate methods for display calibration and for compensating observers' refractive errors. These considerations are beyond the scope of this paper.

## Methods

The OPI has drivers for the Octopus 900 perimeter (Haag-Streit AG, Köniz, Switzerland), the Compass microperimeter (iCare, Finland), and the AP-7000 perimeter (Kowa, Torrance, CA, USA). New drivers have now been incorporated for the IMO (CREWT Medical Systems, Tokyo, Japan) and Android phones with headsets that are compatible with the Google Cardboard software development kit (Google Inc., Mountain View, CA, USA). Drivers for devices such as the VIVE Pro Eye (HTC Corporation, Taoyuan City, Taiwan) and AVA Advanced Vision Analyzer (Elisar, India) are being developed.

### OPI conventions and standards

The OPI implementations and drivers follow conventions and standards that not only accelerate software development, but also enable the creation of custom tests, perimetric algorithms, and procedures. These can all be used with different computer operating systems, programming languages, and with different commercial and experimental perimeters. The key OPI commands are described more fully elsewhere (Turpin et al., 2012) but are listed here for completeness:

- `opiInitialize()`: open connection with the perimeter and initialize it,
- `opiQueryDevice()`: get information about the perimeter,
- `opiSetBackground()`: set the background color, luminance, fixation point, etc.,
- `opiPresent()`: Present a stimulus and return observer's response, and
- `opiClose()`: close the connection with the perimeters.

There are three distinct types of visual field stimuli that can be presented with the `opiPresent()` command: static, temporal, and kinetic. The static type is used for static automated perimetry (Aulhorn & Harms, 1967; Fankhauser, Koch, & Roulier, 1972; Koch, Roulier, & Fankhauser, 1972). The temporal type is used to generate stimuli that vary over time and that are supported by the underlying hardware, such as a counterphase modulated sine-wave grating of

frequency-doubling perimetry (Kelly, 1966; Maddess & Henry, 1992) or the counterphase square-sine flicker stimulus presentation of contrast sensitivity perimetry (Swanson, Malinovsky, Dul, Malik, Torbit, Sutton, & Horner, 2014). The kinetic type can be used to present moving stimuli specified according to the nomenclature introduced by Hans Goldmann (Goldmann, 1999). The level of customization that the OPI offers depends on hardware limitations. For the PhoneHMD OPI, it is possible to define the dynamic range and contrast steps, the stimulus size, color, presentation time, and the response window. It is also possible to set the speed, temporal, and spatial properties of the stimuli.

The `opiPresent()` command returns the response of the observer (usually whether the response button was pressed or not after a stimulus presentation, along with the time between onset of stimulus presentation and response in milliseconds), and the  $x$  and  $y$  pupil position at the time the button was pressed in degrees of visual angle (if hardware allows). The accuracy and precision of all estimates depend on the underlying hardware.

It is possible to interact with the perimeter via the `opiPresent()` command and build a stimulus-response logic from which complex test procedures can be constructed. Several common procedures are already built into the OPI package:

- `MOCS()`: The method of constant stimulus or MOCS (Fechner, 1966),
- `fourTwo()`: The 4-2 dB staircase (Bebie, Fankhauser, & Spahr, 1976; Johnson, Chauhan, & Shapiro, 1992),
- `FT()`: The full threshold algorithm (Bebie et al., 1976; Johnson et al., 1992), and
- `ZEST()`: The Bayesian QUEST and ZEST algorithms (King-Smith, Grigsby, Vingrys, Benes, & Supowit, 1994; Turpin, McKendrick, Johnson, & Vingrys, 2003; Watson & Pelli, 1983).

The implementation of the aforementioned OPI commands and drivers are necessarily different for each different perimeter; however, the specific implementation can be selected with the command `opiChoose()`. Thus, once a specific perimetry driver — Octopus900, KowaAP7000, Compass, IMO, PhoneHMD, or other — has been selected, a dispatcher is set in place so that the same OPI commands listed earlier can be used without change with all supported hardware.

## The visualFields analytical tool

The `visualFields` package has undergone a major revision moving from version 0.6 (Marín-Franch & Swanson, 2013) to version 1.0.1 introduced here. Its core functionality is the same but the code has

been simplified, and a number of conventions have been adopted for clarity and simplicity. An effort has been made to improve its transparency and the reproducibility of its methods. For instance, the SUNY-IU dataset of healthy subjects that was used in the previous version to generate normative values has been incorporated into the package, `vfctrSunyiu24d2`, along with a function that generates the normative reference values. Thus, the command `nvgenerate(vfctrSunyiu24d2)` generates pointwise normative values and the command `nvgenerate(vfctrSunyiu24d2, method = "smooth")` generates the normative values used in the `visualFields` 0.6 (Marín-Franch & Swanson, 2013) using the smoothing techniques as those introduced by Heijl and colleagues for the Statpac 2 (Heijl, Lindgren, & Olsson, 1987; Heijl, Lindgren, Lindgren, Olsson, Asman, Myers, & Patella, 1991). Normative datasets, `vfctrIowaPC26` and `vfctrIowaPeri`, and reference values generated with the function `nvgenerate` for the custom tests used to study the advantages of exploring the full visual field are also included in the package (Marín-Franch, Artes, Chong, Turpin, & Wall, 2018; Wall, Lee, Wanzek, Chong, & Turpin, 2020; Wall, Lee, Wanzek, Zamba, Turpin, Chong, & Marín-Franch, 2019; Wall, Subramani, Chong, Galindo, Turpin, Kardon, Thurtell, Bailey, & Marín-Franch, 2019).

## The opiApp

The R package Shiny (Chang et al., 2020) was used to develop the `opiApp`, a graphical frontend that interacts with the OPI and can be used to run perimetry on conventional and custom grids of test locations. With the `opiApp`, it is possible to configure the device and perimetry settings, define the dynamic range of the dB scale and step size, obtain the luminance profile of Android devices (i.e. the correspondence between pixel value and physical luminance), manage patient datasets, and run static automated perimetry using ZEST, MOCS, 4-2 staircase, and Full-Threshold paradigms.

## Software and data availability

The release version of OPI can be found at <https://CRAN.R-project.org/package=OPI>. The release version of `visualFields` can be found at <https://CRAN.R-project.org/package=visualFields>. The development version of OPI can be found at <https://github.com/turpinandrew/OPI>. The development version of `visualFields` can be found at <https://github.com/imarinf/vf1>. The driver to run perimetry on Android phones with headsets that are compatible with Google Cardboard can be

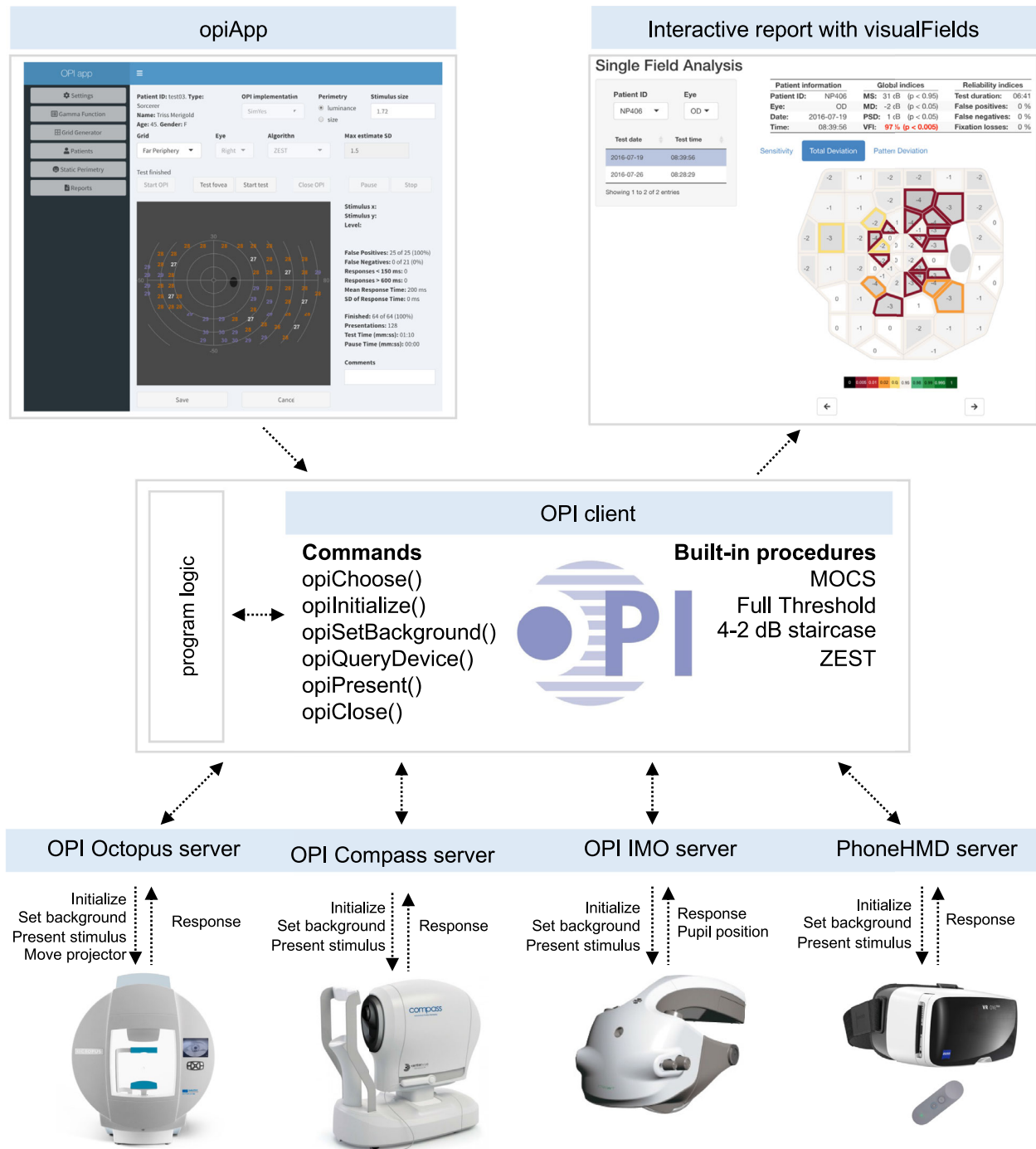


Figure 1. **Illustration of the OPI architecture.** Top left is a graphical interface generated in shiny for a program logic that can be run on any device at the bottom through the OPI client (center) as it dispatches commands via the OPI servers. Once a suitable dataset of healthy controls has been collected, statistical analyses as the one in the top right can be generated with the visualFields package.

found at <https://github.com/imarinftr/opiPhoneHMD>. The opiApp frontend for the OPI can be found at <https://github.com/imarinftr/opiApp>. Most datasets used in this paper can be found within the visualFields package. The dataset of healthy subjects for the full visual field can be found in the visualFields package as

well as in <https://www.sciencedirect.com/science/article/pii/S2352340918311570>. The applications developed based on the OPI and visualFields packages and presented in the paper, as well as the datasets are available from the corresponding author on reasonable request.

## Results

Figure 1 illustrates the OPI architecture. Once the hardware is selected, the R OPI client dispatches the commands to connect, disconnect, set background and other settings, and present stimuli to the corresponding OPI server (Octopus 900, AP-7000, Compass, IMO, PhoneHMD, etc.), which ultimately communicates the commands to the hardware. The server then waits for the hardware to send its state (machine initialized, background lit, pupil position, clicker pressed within a time response window after stimulus onset, etc.) and communicates the response to the OPI client. A frontend, as the opiApp shown in the top left of Figure 1, can be developed on top of a program logic to run conventional or custom visual field tests on both regular or irregular grids of test locations with the OPI built-in algorithms, such as Zippy Estimation of Thresholds (ZEST). It also has the capability to run other procedures, including a suprathreshold test due to Aulhorn (Aulhorn & Harms, 1967) or the binocular Esterman test (Esterman, 1982). The frontend and the backend are running in parallel separated processes. The frontend communicates actions for the backend to

process and the backend interfaces with the perimetry devices via the OPI. This way, the opiApp is replaceable by any other frontend developed in R, Java, Python, or any other suitable programming language.

The OPI was used to run a series of tests of the full visual field (Marín-Franch et al., 2018; Wall et al., 2020; Wall, Lee, et al., 2019; Wall, Subramani, et al., 2019). A publicly available dataset of 98 eyes of 98 healthy subjects (Marín-Franch et al., 2018) was used to derive normative values with the visualFields R package. Figure 2 shows a statistical analysis of the results for a specific full visual field, which consists of the combined analysis of two tests taken on the same day, one for the central visual field and another for the far periphery (Marín-Franch et al., 2018; Wall et al., 2020; Wall, Lee, et al., 2019; Wall, Subramani, et al., 2019). A dataset of healthy eyes (Marín-Franch et al., 2018) is incorporated in the visualFields version 1.0.1 for the central and peripheral tests, vfctrIowaPC26 and vfctrIowaPeri. A script that generates figures for all subjects in those datasets is provided as supplemental material with the name vfPlotFullField.r.

The visualFields package also offers tools to analyze longitudinal data, including pointwise linear regression, and the permutation of pointwise linear regression,

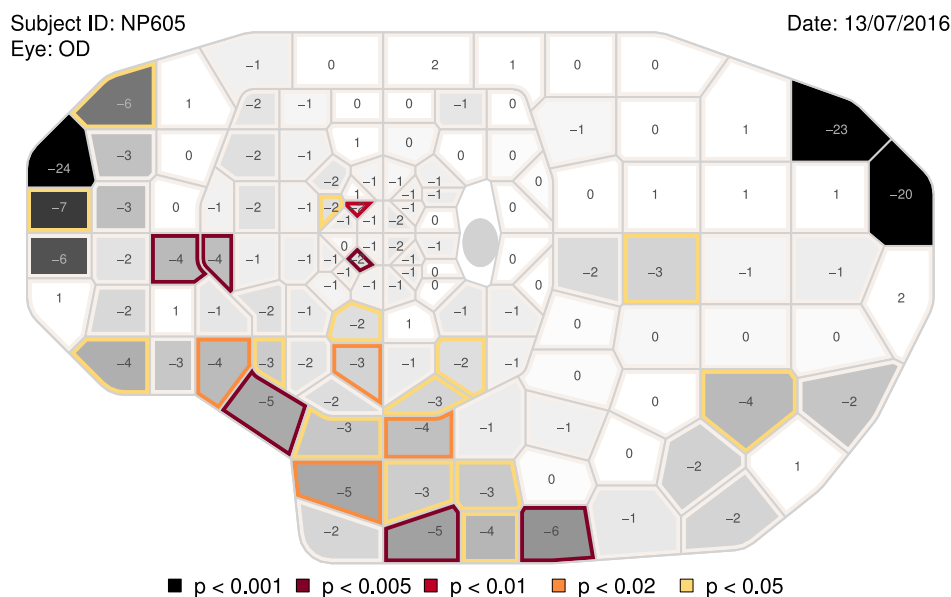


Figure 2. **Combined grayscale sensitivity and color-coded total-deviation map.** The representation of the full visual field results are composites of two tests taken on the same day: one spanning the central 26 degrees of the visual field and another from 26 degrees to 50 degrees nasally, to 80 degrees temporally, to 46 degrees superiorly, and to 50 degrees inferiorly. The values shown at each location are total deviations, departures in sensitivity from the mean normal sensitivities for age-matched controls. The background grayscale of each tile represents the estimated sensitivity at the corresponding visual field location, where darker means lower sensitivity. Tiles whose border is shown in color are significantly depressed according to the statistical analysis of the total-deviation map. The tiles involving each visual field location were obtained using Voronoi tessellation (Aurenhammer & Klein, 1999; Kucur, Holló, & Sznitman, 2018) to achieve an efficient representation for both highly irregular grids. Voronoi tessellations are a partitioning of a surface into regions so that the center of each cell is its mean (center of mass). Every point in a given Voronoi polygon is closer to its generating point than to any other cell.

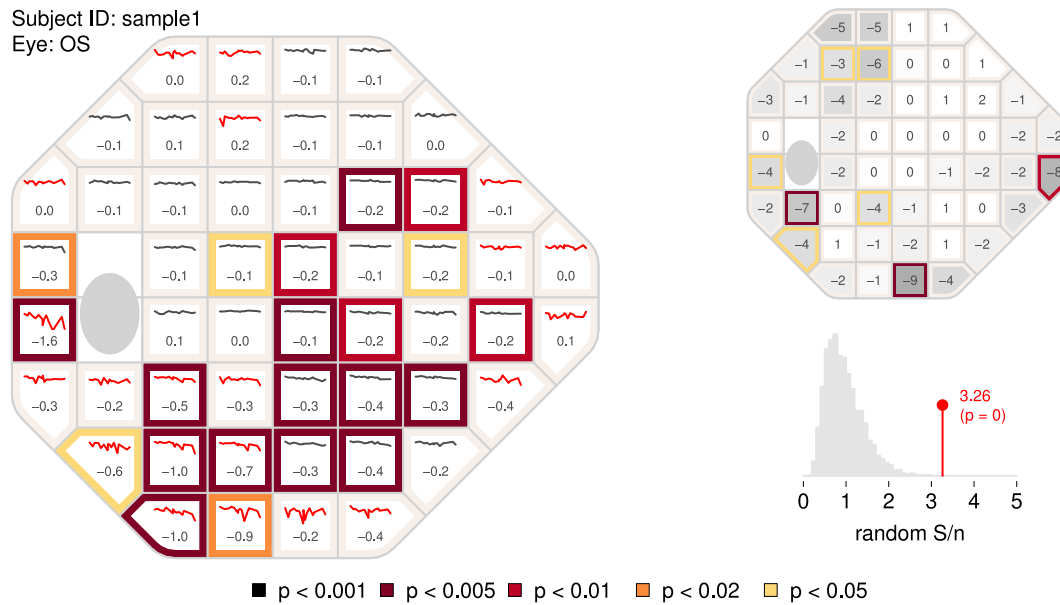


Figure 3. **The PoPLR analysis.** The left panel shows the slopes at each visual field location obtained with pointwise linear regression of total deviation values over time along with sparklines representing the values over the whole series. The colors at the border of the tiles categorize the  $p$  values of the one-tail  $t$ -test with the alternative that the slope is negative. To identify highly variable series of visual fields, the sparklines are shown in red if the median absolute deviation of the residuals from linear regression were greater than 2 dB. The top right graph is the combined grayscale sensitivity and total-deviation map of the baseline sensitivity values (intercept of pointwise regression on sensitivities). The bottom right function shows the histogram of random  $S/n$ , where  $n$  is the number (52 in this case) of regression analyses performed obtained by permuting the series as part of PoPLR analysis. The  $p$  value for the PoPLR analysis testing whether there is deterioration is shown next to the value of the observed  $S/n$  statistic.

or PoPLR (Marín-Franch, Artes, Turpin, & Racette, 2021; O’Leary, Chauhan, & Artes, 2012). Figure 3 shows a brief report (generated with the script `vfPoPLRAnalysis.r` provided as supplemental material) and with the `vfpgwSunnyiu24d2` dataset (Artes, O’Leary, Nicolela, Chauhan, & Crabb 2014), which was collected with the Humphrey Field Analyzer (Carl Zeiss Meditec, Inc., Dublin, CA, USA). The normative values to obtain total deviation values and probability maps were generated using the dataset, `vfctrSunnyiu24d2`, from a prospective longitudinal study conducted at Indiana University and State University of New York, SUNY (Marín-Franch & Swanson, 2013). The normative values were obtained with the command `nvgenerate(vfctrSunnyiu24d2)`.

The ZEST algorithm used to measure the visual field (Marín-Franch et al., 2018; Wall et al., 2020; Wall, Lee, et al., 2019; Wall, Subramani, et al., 2019) in Figure 2 can be invoked by the `opiApp`. This specific implementation has a bimodal prior probability mass function with one peak centered at 0 dB to model sensitivities of damaged locations and another peak that depends on sensitivity estimates in neighboring locations (Vingrys & Pianta, 1998) applied using a growth algorithm that defines the order at which locations at different regions of the visual field are tested. Supplementary Figure S1 illustrates the

growth algorithm for the 24-2 grid of test locations. A conventional 24-2 grid was used instead of the irregular one in Figure 2 for clarity of illustration. Supplementary Figure S2 shows the growth-algorithm setup for custom central and far periphery irregular grids of test locations.

Figure 4 shows a snapshot of the ZEST algorithm with a growth pattern algorithm to an irregular grid of test locations that corresponds to the central part in the full visual field tests (Marín-Franch et al., 2018; Wall et al., 2020; Wall, Lee, et al., 2019; Wall, Subramani, et al., 2019) with the `opiApp` running with the `PhoneHMD OPI`. The `opiPhone` server permits to present stimuli in either or both eyes. Likewise, the background and fixation targets can be set for the either or both eyes. The gamma function was obtained with the option `Gamma Function` provided by the `opiApp` frontend (see Supplementary Figure S3). Other irregular grids can be designed with the option `Grid Generator` (see Supplementary Figure S4).

## Discussion

Visual fields are among the most often conducted vision tests, second only after visual acuity. Thousands

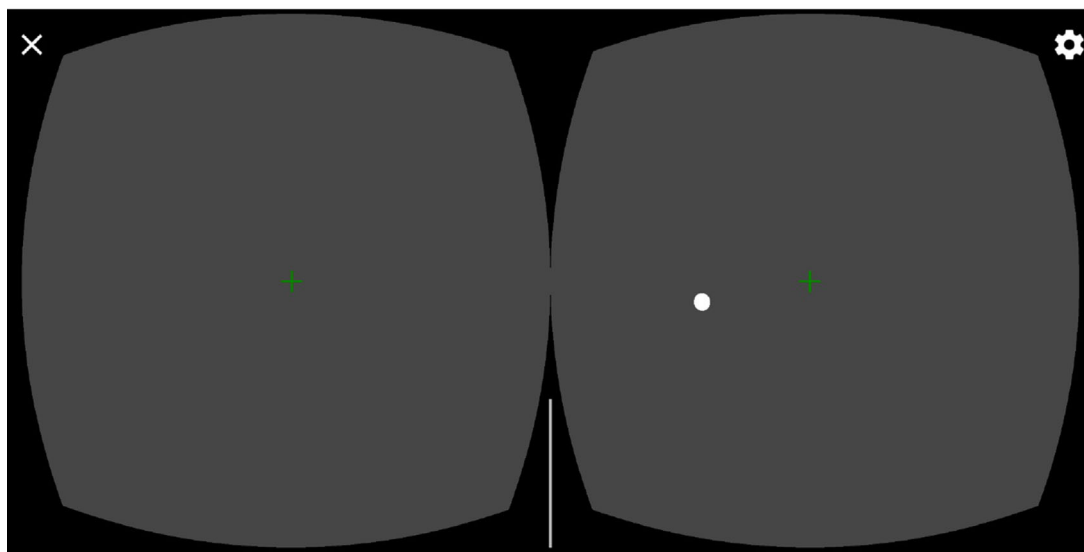
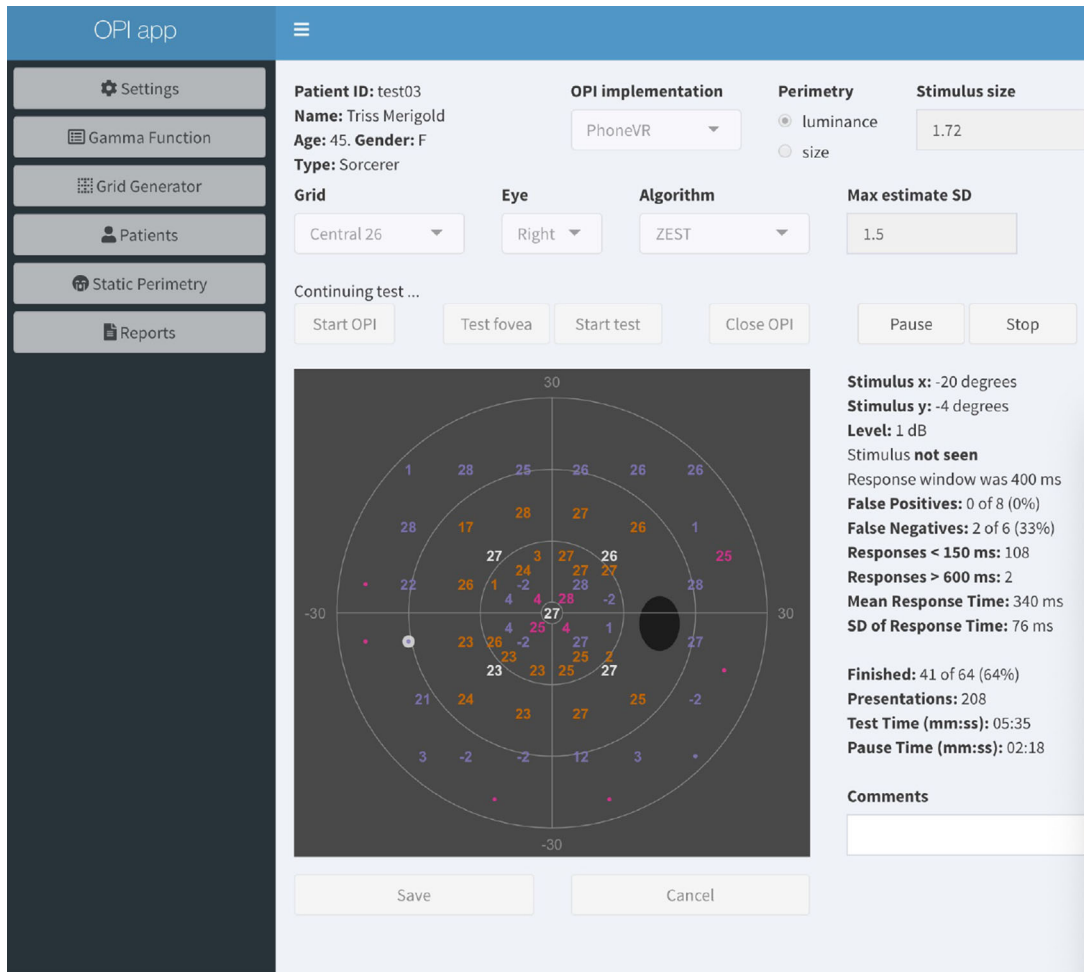


Figure 4. **The opiApp on the PhoneHMD OPI.** The ZEST algorithm for luminance (white-on-white) perimetry for a custom irregular grid of test locations is executed for a (fictitious) patient. The opiApp (top) sends commands to an Android Samsung Galaxy S9 phone (bottom) to generate white visual stimuli at different intensities and at different distances from the fixation point (green cross). At each presentation, the web app updates and shows the interim results of the test.

of tests are conducted every day. Although there have been many interesting advances in perimetry (Greenfield, Deiner, Nguyen, Wollstein, Damato, Backus, Wu, Schuman, & Ou, 2022; Jones, 2020; Prager, Kang, & Tanna, 2021; Prince, Thompson, Mwanza, Tolleson-Rinehart, & Budenz, 2021; Stapelfeldt, Kucur, Huber, Höhn, & Sznitman, 2021; Turpin, Myers, & McKendrick, 2016), the community pursuing clinical visual field research remains relatively small. To many newcomers, perimetry first appears as something of a black art, with a specialist terminology somewhat at odds with mainstream psychophysics.

In addition, whereas researchers have always shared resources, such as data and analysis routines with each other, this has traditionally been in an informal way, more open to those who are already part of an established research community than newcomers. More formal arrangements often come with bureaucratic and legal burdens. It has only been relatively recent that clinicians and scientists have begun to share datasets (Bryan, Vermeer, Eilers, Lemij, & Lesaffre, 2013; Marín-Franch et al., 2018; Montesano, Chen, Lu, Lee, & Lee, 2022; Swanson, Dul, Horner, & Malinovsky, 2016) and source code openly under the General Public License, such as the GNU GPL (Free Software Foundation, 2022) or the MIT license (Open Source Initiative, 2022). The founding aims of the Open Perimetry Initiative (Turpin et al., 2012) were to foster an unfettered exchange of ideas, tools, and data related to visual fields and perimetry, and thereby to reduce

the inefficiencies and inconveniences traditionally associated with research in perimetry.

Perimeters built on consumer hardware could make a real impact (Anthes et al., 2016) on several hitherto unsolved problems in clinical perimetry. For example, if patients could test themselves at home, visual field tests could potentially be performed much more frequently than with current office-based equipment. In turn, frequent testing could lead to a breakthrough in our ability to measure change over time. It is not hard to think of other examples where wider access to visual field tests could improve patient care or increase access to visual assessment in occupational or other relevant applied settings. Furthermore, reducing reliance on expensive perimetry specific hardware may accelerate the input from a new generation of vision scientists to the field of perimetry research. However, to make best use of consumer electronics hardware originally designed for other applications, visual field tests will need to be adapted, along with the statistical analyses and management of the resulting data.

By adhering to the open science principles — open source, open data, and open access — this work can be performed transparently and made as widely accessible as possible. Modern visual field research has many worthwhile challenges to be addressed, and we hope that the OPI will act as a catalyst to bring new minds into this domain, to facilitate collaboration between research groups across the globe (Figure 5), and between basic and clinical scientists and industry partners.

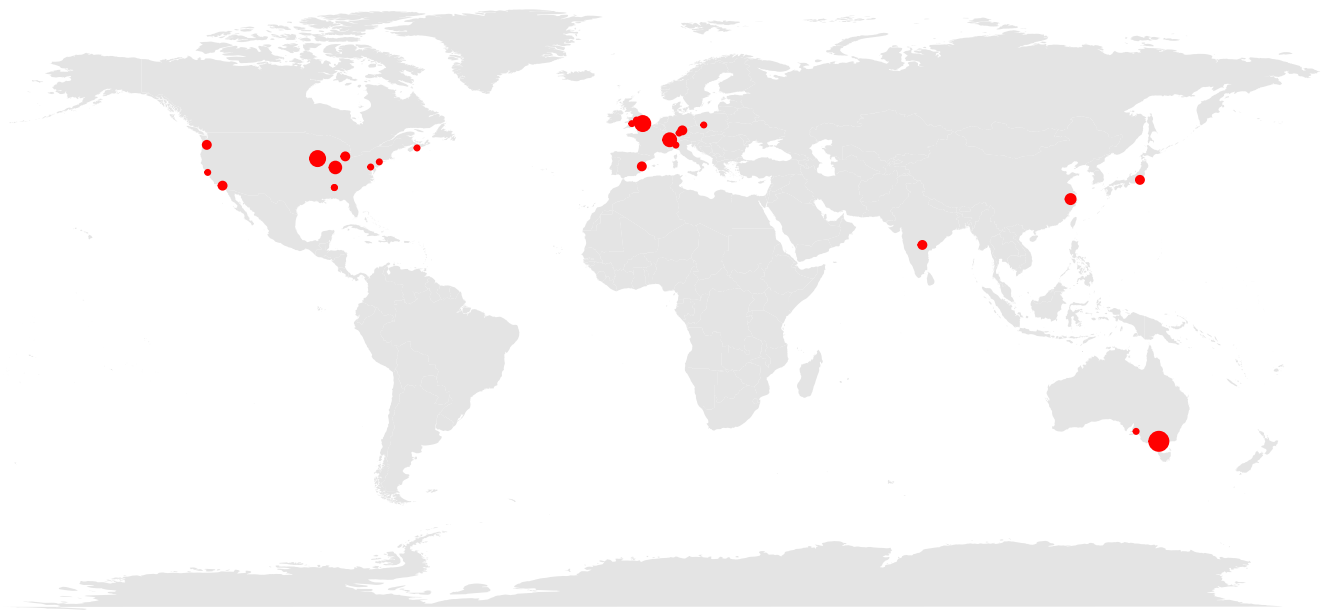


Figure 5. **Geographical map of citations to the OPI and the visualFields R package as listed in Scopus.** The red solid circles demarcate the cities of the first authors' affiliations. The sizes of the circles represent the number of citations from each city, up to three. As of February 2022, OPI and visualFields related publications received 66 peer-reviewed citations from 12 countries in 4 continents.



## Acknowledgments

The authors thank Jize Dong for writing an earlier version of the Java code for the OPI PhoneHMD server. The authors also thank Zachary Heinzman for reviewing a previous version of the manuscript.

Supported by Veterans Administration Merit Review (I01RX-001821-01A1) and Computational Optometry (Atarfe, Spain, URL: [www.optocom.es](http://www.optocom.es)).

Commercial relationships: none.

Corresponding author: Iván Marín-Franch.

Email: [imarinfr@optocom.es](mailto:imarinfr@optocom.es).

Address: Computational Optometry, Atarfe, Spain.

## References

- Anthes, C., García-Hernández, R. J., Wiedemann, M., & Kranzlmüller, D. (2016). State of the Art of Virtual Reality Technology. *IEEE Conference on Aerospace*, pp. 1–19.
- Artes, P. H., O’Leary, N., Nicolela, M. T., Chauhan, B. C., & Crabb, D. P. (2014). Visual Field Progression in Glaucoma: What Is the Specificity of the Guided Progression Analysis? *American Academy of Ophthalmology*, 121(10), 2023–2027.
- Aulhorn, E., & Harms, H. (1967). Early visual field defects in glaucoma. In W. Leydhecker (Ed.), *Glaucoma Tutzing Symposium*. Beckenham, UK: S. Karger, Ltd.
- Aurenhammer, F., & Klein, R. (1999). Voronoi Diagrams. In J. R. Sack, & J. Urrutia (Eds.), *Handbook of computational geometry* (pp. 201–290). North-Holland, UK: Elsevier.
- Bebie, H., Fankhauser, F., & Spahr, J. (1976). Static perimetry: strategies. *Acta Ophthalmologica*, 54(3), 325–338.
- Bryan, S. R., Vermeer, K. A., Eilers, P. H. C., Lemij, H. G., & Lesaffre, E. M. E. H. (2013). Robust and Censored Modeling and Prediction of Progression in Glaucomatous Visual Fields. *Investigative Ophthalmology and Visual Science*, 54(10), 6694–6700.
- Chang, W., Cheng, J., Allaire, J., Xie, Y., & McPherson, J. (2020). *shiny: Web Application Framework for R* (R package version 1.4.0.2), <https://cran.r-project.org/package=shiny>.
- Esterman, B. (1982). Functional Scoring of the Binocular Field. *Ophthalmology*, 98, 1226–1234.
- Fankhauser, F., Koch, P., & Roulier, A. (1972). On Automation of Perimetry. *Albrecht Von Graefes Arch Klin Exp Ophthalmol*, 184, 126–150.
- Fechner, G. T. (1966). *Elements of psychophysics*. New York, NY: Holt, Rinehart & Winston Publishing. [Original work published in German “Fechner, G. T., Elemente der Psychophysik, Breitkopf und Härtel, Leipzig, 1860”].
- Free Software Foundation. (2022). *GNU General Public Licence*, <https://www.gnu.org/licenses/gpl-3.0.en.html>.
- Goldmann, H. (1999). Fundamentals of exact perimetry. 1945. *Optometry and Vision Science*, 76(8), 599–604.
- Greenfield, J. A., Deiner, M., Nguyen, A., Wollstein, G., Damato, B., & Backus, B. T. et al. (2022). Virtual Reality Oculokinetic Perimetry Test Reproducibility and Relationship to Conventional Perimetry and OCT. *Ophthalmology Science*, 2(1), 100105.
- Heijl, A., Lindgren, G., Lindgren, A., Olsson, J., Asman, P., Myers, J. S., . . . Patella, M. (1991). Extended empirical statistical package for evaluation of single and multiple fields: Statpac 2. *Perimetry Update 1990/1991*.
- Heijl, A., Lindgren, G., & Olsson, J. (1987). A package for the statistical analysis of visual fields. *Documenta Ophthalmologica Proceedings Series*, 49, 153–168.
- Johnson, C. A., Chauhan, B. C., & Shapiro, L. R. (1992). Properties of Staircase Procedures for Estimating Thresholds in Automated Perimetry. *Investigative Ophthalmology and Visual Science*, 33(10), 2966–2974.
- Jones, P. E. (2020). An Open-source Static Threshold Perimetry Test Using Remote Eye-tracking (Eyecatcher): Description, Validation, and Preliminary Normative Data. *Translational Vision Science and Technology*, 9(8), 18:1–12.
- Kelly, D. H. (1966). Frequency Doubling in Visual Responses. *Journal of the Optical Society of America*, 56(11), 1628–1633.
- King-Smith, P. E., Grigsby, S. S., Vingrys, A. J., Benes, S. C., & Supowit, A. (1994). Efficient and Unbiased Modifications of the QUEST Threshold Method: Theory, Simulations, Experimental Evaluation and Practical Implementation. *Vision Research*, 34(7), 885–912.
- Koch, P., Roulier, A., & Fankhauser, F. (1972). Perimetry - the Information Theoretical Basis for its Automation. *Vision Research*, 12, 1619–1630.
- Kucur, Ş. S., Holló, G., & Sznitman, R. (2018). A deep learning approach to automatic detection of early glaucoma from visual fields. *PLoS One*, 13(11), e0206081.

- Maddess, T., & Henry, G. H. (1992). Performance of nonlinear visual units in ocular hypertension and glaucoma. *Clinical Vision Sciences*, 7(5), 371–383.
- Marín-Franch, I., Artes, P. H., Chong, L. X., Turpin, A., & Wall, M. (2018). Data obtained with an open-source static automated perimetry test of the full visual field in healthy adults. *Data in Brief*, 21, 75–82.
- Marín-Franch, I., Artes, P. H., Turpin, A., & Racette, L. (2021). Visual Field Progression in Glaucoma: Comparison Between PoPLR and ANSWERS. *Translational Vision Science and Technology*, 10(14), 13.
- Marín-Franch, I., & Swanson, W. H. (2013). The visualFields package: A tool for analysis and visualization of visual fields. *Journal of Vision*, 13(4), 1–12, 10.
- Montesano, G., Chen, A., Lu, R., Lee, C. S., & Lee, A. Y. (2022). UWHVF: A Real-World, Open Source Dataset of Perimetry Tests From the Humphrey Field Analyzer at the University of Washington. *Translational Vision Science and Technology*, 11(1), 1:1–8.
- O’Leary, N., Chauhan, B. C., & Artes, P. H. (2012). Visual field progression in glaucoma: estimating the overall significance of deterioration with permutation analyses of pointwise linear regression (PoPLR). *Investigative Ophthalmology and Visual Science*, 53(11), 6776–6784.
- Open Source Initiative. (2022). *The MIT License*, <https://opensource.org/licenses/MIT>.
- Prager, A. J., Kang, J. M., & Tanna, A. P. (2021). Advances in perimetry for glaucoma. *Current Opinion in Ophthalmology*, 32(2), 92–97.
- Prince, J., Thompson, A., Mwanza, J. C., Tolleson-Rinehart, S., & Budenz, D. L. (2021). Glaucoma Screening Using an iPad-Based Visual Field Test in a West African Population. *Ophthalmology Glaucoma*, <https://doi.org/10.1016/j.ogla.2021.09.002>.
- R Core Team. (2022). *R: A Language and Environment for Statistical Computing*, <https://www.r-project.org/>.
- Stapelfeldt, J., Kucur, Ş. S., Huber, N., Höhn, R., & Sznitman, R. (2021). Virtual Reality–Based and Conventional Visual Field Examination Comparison in Healthy and Glaucoma Patients. *Translational Vision Science and Technology*, 10(12), 10–1:20.
- Swanson, W. H., Dul, M. W., Horner, D. G., & Malinovsky, V. E. (2016). Contrast sensitivity perimetry data from adults free of eye disease. *Data in Brief*, 8, 654–658.
- Swanson, W. H., Malinovsky, V. E., Dul, M. W., Malik, R., Torbit, J. K., Sutton, B. M., . . . Horner, D. G. (2014). Contrast Sensitivity Perimetry and Clinical Measures of Glaucomatous Damage. *Optometry and Vision Science*, 91(11), 1302–1311.
- Turpin, A., Artes, P. H., & McKendrick, A. M. (2012). The Open Perimetry Interface: an enabling tool for clinical visual psychophysics. *Journal of Vision*, 12(11), 22.
- Turpin, A., McKendrick, A. M., Johnson, C. A., & Vingrys, A. J. (2003). Properties of Perimetric Threshold Estimates from Full Threshold, ZEST, and SITA-like Strategies, as Determined by Computer Simulation. *Investigative Ophthalmology and Visual Science*, 44(11), 4787–4795.
- Turpin, A., Myers, J. S., & McKendrick, A. M. (2016). Development of Visual Field Screening Procedures: A Case Study of the Octopus Perimeter. *Translational Vision Science & Technology*, 5(3): 3, 1–9.
- Vingrys, A. J., & Pianta, M. J. (1998). Developing a clinical probability density function for automated perimetry. *Australian and New Zealand Journal of Ophthalmology*, 26, S101–S103.
- Wall, M., Lee, E. J., Wanzek, R. J., Chong, L. X., & Turpin, A. (2020). Temporal Wedge Defects in Glaucoma. *Journal of Glaucoma*, 29(3), 191–197.
- Wall, M., Lee, E. J., Wanzek, R. J., Zamba, K. D., Turpin, A., Chong, L. X., . . . Marín-Franch, I. (2019). Threshold Automated Perimetry of the Full Visual Field in Patients with Glaucoma with Mild Visual Loss. *Journal of Glaucoma*, 28(11), 997–1005.
- Wall, M., Subramani, A., Chong, L. X., Galindo, R., Turpin, A., Kardon, R. H., . . . et al, . (2019). Threshold static automated perimetry of the full visual field in idiopathic intracranial hypertension. *Investigative Ophthalmology and Visual Science*, 60(6), 1898–1905.
- Watson, A. B., & Pelli, D. G. (1983). QUEST: A Bayesian adaptive psychometric method. *Perception & Psychophysics*, 33(2), 113–120.