Short communication

OrbitView: Eye movement visualization software

Simon Hong[a,*], Lance M. Optican[a], Edmond J. FitzGibbon[a], David S. Zee[b], Aasef G. Shaikh[c]

[a] National Eye Institute, NIH, Bethesda, MD, United States
[b] Department of Neurology, The Johns Hopkins University, Baltimore, MD, United States
[c] Department of Neurology, Case Western Reserve University, Cleveland, OH, United States

ARTICLE INFO

ABSTRACT

Measurement of eye movements often helps to diagnose ocular motor disorders in the clinic, and is also used as a research tool in ocular motor, vision and vestibular research. Eye movements, however, are usually recorded without simultaneous video recordings, making offline interpretation difficult. We developed a tool that converts the measured eye movement data into a three-dimensional (3D) movie of eye movements. Having useful functions such as slow-play, pause and exaggeration of the movements, this new software provides a research and teaching tool to aid interpretation of the recorded eye movements.

Published by Elsevier B.V.

1. Introduction

Measurement of eye movements is often used to analyze ocular motor behavior in clinical and research settings. For example, if the slow phases of a spontaneous nystagmus have a constant velocity, a vestibular disorder is suggested. On the other hand, an exponentially decaying drift towards the midline producing a gaze-evoked nystagmus suggests an impaired ("leaky") ocular motor neural integrator; also, a slow phase drift with an increasing velocity suggests an unstable neural integrator (Leigh and Zee, 2006). It is often difficult to discriminate, at the bedside, the time course of the drift of the slow phase. Similarly, when nystagmus is disconjugate, the phase relationship between the two eyes may be difficult to discern at the bedside. Thus, eye movements are usually recorded, converting the 3D rotations of the eye into 2D orientation vs. time displays. A tool that can convert the 3D positions of both eyes to emulate the actual eye movements would be helpful for clinical correlations and for teaching. There are computer applications that simulate eye movements (e.g., Buchberger et al., 2003) to illustrate the pattern of abnormal eye movements in diseases. To our knowledge, however, there is no simple software that can readily be used to visualize recorded eye movements in 3D.

Recognizing the need for a tool to visualize recorded eye movement data, we developed software that converts eye movement recordings into a more intuitive 3D movie. We are offering this software as a diagnostic, research and teaching tool to clinicians, scientists and students to display ocular motor behavior easily.

2. Methods and results

We developed the program in visual C++ to faithfully reproduce the eye movements with a temporal accuracy of ±1 ms or the highest temporal resolution of the acquired data (if acquired with less than 1000 Hz resolution). Data are automatically sub-sampled at an appropriate rate for display. The program accepts commonly used horizontal (H) vertical (V) and torsional (T) rotation angle components to describe the orientation of the two eyes at a given moment. Provision is also made for other data channels, such as head orientation.

2.1. Input data

The program reads data from a text file (e.g., Source1.txt), which contains the sampling rate (first line) followed by labels (second line) followed by movement data as column vectors (Fig. 1A). The program accepts up to 12 labels and corresponding columns of data.

Each label is one word. LeftH and Left-H are reserved names for the left eye horizontal rotation vector (not case sensitive). This convention applies to the other eye position labels (i.e., LeftV, Left-V, LeftT, Left-T, RightH, Right-H, RightV, Right-V, RightT, Right-T). The label cannot start with a number. The corresponding vectors are expressed in degrees. In the example shown in Fig. 1A, only

* Corresponding author at: Laboratory of Sensorimotor Research, NEI/NIH, 49 Convent Drive, Room 2A50, Bethesda, MD 20892, USA. Tel.: +1 301 496 9375; fax: +1 301 402 0511.
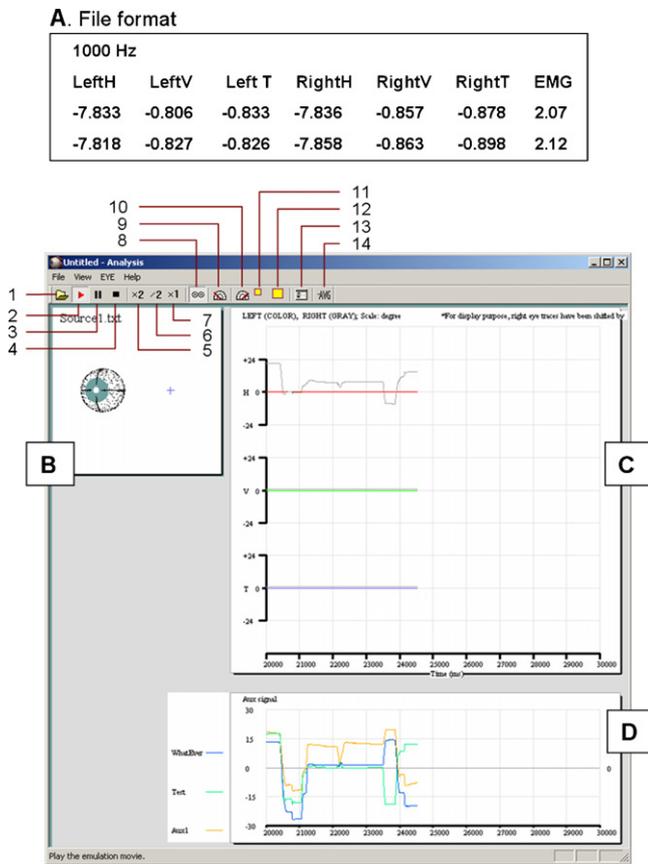E-mail address: hongy@nei.nih.gov (S. Hong).

A. File format

1000 Hz

LeftH	LeftV	Left T	RightH	RightV	RightT	EMG
-7.833	-0.806	-0.833	-7.836	-0.857	-0.878	2.07
-7.818	-0.827	-0.826	-7.858	-0.863	-0.898	2.12

Fig. 1. (A) The program reads data from a text file (e.g., Source1.txt). The text file starts with a sampling rate (1000 Hz in the figure) in the first line. The second line describes the labels for the vectors. Each label is one word. For example, LeftH or Left-H are reserved names for left eye horizontal vector (not case sensitive). Similar convention applies to other eye vector labels (i.e., LeftV, Left-V, LeftT, Left-T, RightH, Right-H, RightV, Right-V, RightT, Right-T). The column vectors are described from the third line. The vectors are expressed in degrees. The program accepts up to 12 labels and corresponding columns of vectors. The example shows 6 eye vectors and one extra vector. (B)–(D) The program has 14 icons and 3 figure panels. The 3D eyeball panel (B) shows the reconstructed eye movements in real time. Eye position trace panel (C) shows the horizontal (H), vertical (V), and torsional (T) positions of the eyes. Left eye position traces are in color, right eye position traces are in gray. The extra vector panel (D) shows the signals of the extra vectors. The ordinate axes of (C) and (D) are automatically adjusted to accommodate the amplitude of the signals. See text for details.

2 ms of movement data (12 elements = 6 vectors × 2 ms) are shown with a 1 kHz sampling rate. The spatial order in each column vector represents the temporal order (e.g., at $t = 1$, LeftH = −7.833; at $t = 2$, LeftH = −7.818). As long as the data elements are real numbers, the numeric format does not matter (i.e., the program is compatible with most integer and floating point formats). Note that any of the eye position vectors can be omitted, i.e., as long as there is at least one eye position vector, the data is playable. In addition to these eye vectors, up to six other vectors are also allowed for other measurements that accompany the eye movements, such as an electromyographic recording (EMG) as in Fig. 1A, or head movements. Each of these extra vectors should have a label and should be given in the file in a similar manner to the eye vectors, with the same sampling frequency. A label that is not one of the reserved names for the eye vectors will be regarded as the label of an extra vector. The traces of these extra vectors will appear in a separate graph (Fig. 1D), under the eye trace graph (Fig. 1C). These extra vectors are optional.

2.2. Coordinate systems

Rotations may be measured in many different coordinate systems. The two most commonly used in eye movement research are (1) Fick's nested axes (i.e., the axis for horizontal rotations is fixed in the head and carries the axis for vertical rotations that moves with the eye, and that axis carries the torsional axis) and (2) head fixed axes (i.e., quaternions, rotation vectors, or Euler–Rodrigues angles; Haslwanter, 1995). Many modern eye movement systems use rotation vectors, so we developed our software assuming that the input data was in that format. If data are stored in Fick coordinates, they can be easily converted to rotation vectors (cf. Eq. (25) in Haslwanter, 1995).

2.3. Visualization

To visualize the eye movements in real time 3D, the program uses numerous pixels $\{(x_i, y_i, z_i)| i = 1, 2, 3, \ldots\}$ arranged in the shape of an eyeball (Fig. 1B). By rotating these pixels using the horizontal, vertical, and torsional components (H, V, T) at time t, the program creates parallax differences that give a convincing 3D percept. Using the given rotation angles of H, V, T, the unit vector of axis of rotation (u_H, u_V, u_T) is generated as follows:

$$\begin{cases} u_H = (H/k)/\theta \\ u_V = (V/k)/\theta \\ u_T = (T/k)/\theta \end{cases} \quad (1)$$

where k of 57.2958 is a factor that converts the angle in degrees to radians; and θ is the amplitude of the vector:

$$\theta = \sqrt{H^2 + V^2 + T^2} \quad (2)$$

The rotation of pixels given initially as (x_i, y_i, z_i) was performed with the following matrix operation:

$$\begin{pmatrix} x'_i \\ y'_i \\ z'_i \end{pmatrix} = \begin{pmatrix} u_V^2 + (1-u_V^2)c & u_V u_T(1-c) - u_H s & u_V u_H(1-c) + u_T s \\ u_V u_T(1-c) + u_H s & u_T^2 + (1-u_T^2)c & u_T u_H(1-c) - u_V s \\ u_V u_H(1-c) - u_T s & u_T u_H(1-c) + u_V s & u_H^2 + (1-u_H^2)c \end{pmatrix} \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} \quad (3)$$

where $c = \cos\theta$ and $s = \sin\theta$. This rotation matrix (Goldstein, 1980) is useful when the following two factors are known: (1) the axis of rotation, given as a unit vector (u_H, u_V, u_T), and (2) a rotation angle θ along the axis.

2.4. User interface

Fig. 1(B)–(D) illustrates the user interface. Icon 1 is used to open the desired data file to be visualized. Once the data file is selected the emulation will play automatically. Icon 2 can be used to replay the emulation. Icon 3 is used to pause the emulation and icon 4 terminates the emulation before completion. Icons 5 and 6 are used to exaggerate or attenuate the amplitude of the eye movements, respectively. Icon 7 is used to reset the eye movement to its original amplitude. The default view of the emulated eye movements is the "examiner's view", i.e., as if the examiner is watching the eye movements of the subject from the front. Icon 8 can be used to reverse the viewing perspective from the "examiner's view" (Fig. 1B) to the "subject's view". Icons 9 and 10 are used to increase or decrease the speed at which the emulation is played (i.e., fast forward or slow motion). Icon 11 decreases the size of the window and the eyes, while icon 12 increases it. Icon 13 opens a 'dialog box' that accepts the *start* and *end* times to display when the user wants to see a specific period of the eye movement emulation. Icon 14 is used to subtract the average position of the eye movements. This

may be useful if the recording has an offset or when the user wants to see the movements of the eyes relative to mean eye position. The second half of the movie in Supplementary material 1 showing diagonal eye movements used this average-subtracted mode. This removed the bias of the eye position present throughout the recording period introduced by a search coil on the eye of the subject that became misaligned during the recording.

Fig. 1 shows an example in which the data file has a right eye vector and three extra vectors. Fig. 1C shows the 3D eye positions in degrees plotted along the *y*-axis and the time in seconds plotted on the *x*-axis. Colored traces represent left eye positions, while the corresponding gray traces depict the right eye positions. Red traces show horizontal, green traces vertical and blue traces torsional eye positions. The scale of the ordinate axes will automatically adjust to accommodate the scale of the data. In case the user wants a specific scale, it can be specified by opening the Dialog box (icon 13) and typing in the desired scale in the 3rd row.

2.5. Software validation

In order for the computer program to be useful, the calculated 3D eye movements should closely emulate the actual eye movements. To compare the emulated 3D eye movements and the actual eye movements, we measured eye positions around all three axes with a search coil system (Robinson, 1963) and also simultaneously video recorded the eye movements at 30 fps. During the experiment, the subject's head was stabilized with a chin and forehead support. Raw search coil signals were sampled at 1000 Hz with 12 bit resolution and filtered in hardware (90-Hz low-
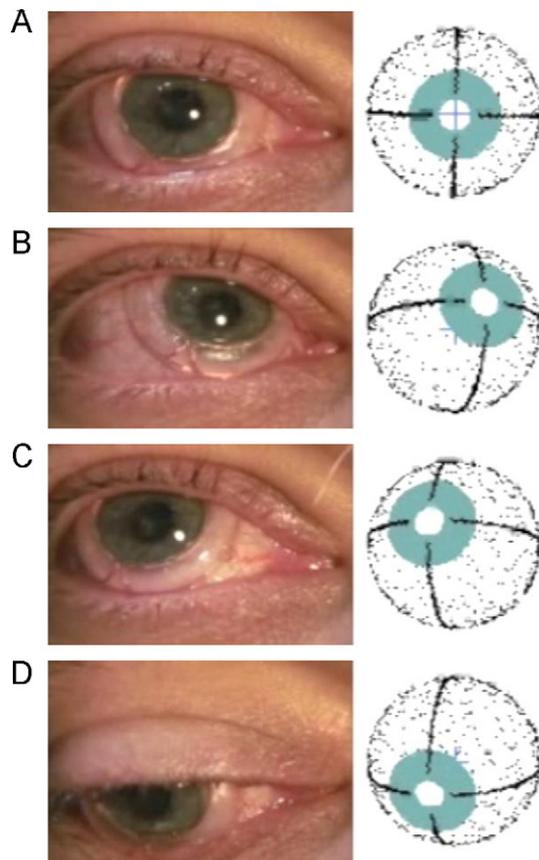

Fig. 2. Snapshots of actual and emulated eye positions. The photos of actual eye positions (A)–(D) and the corresponding emulated eye positions by the software (right column) demonstrates that the software could faithfully replicate the actual movements. Supplementary material 1 compares an actual video and an emulation of the subject's eye movements.

pass Butterworth). No off-line data processing was performed. The software-generated 3D eye positions closely matched those of the actual ones. Fig. 2 shows snapshots of the actual eye positions and the software-generated eye positions. Supplementary material 1 compares an actual video and an emulation of the subject's eye movements.

3. Discussion

There are some computer applications that simulate eye movements (e.g., Buchberger et al., 2003) to illustrate the pattern of abnormal ocular motor motility in neurological diseases, or tools to illustrate oculomotor coordinate systems (e.g., Schreiber and Schor, 2007). To our knowledge, however, there is no simple software that can readily be used to visualize data around all three axes of eye rotation from eye movement recordings, such as those acquired using scleral search coils. In the following we discuss some applications of our software.

Some clinical maneuvers alter abnormal eye movements, for example, eye closure increases the amplitudes of rapid saccadic oscillations such as ocular flutter (Shaikh et al., 2007, 2008), or reduces the amplitude of certain types of lower frequency oscillations such as pendular nystagmus (Nakada et al., 1988). The search coil technique allows eye movements to be measured under closed eyelids. However, it may be useful to visualize and compare the actual eye movement under closed eyelids with that when the eyes are open. A tool that converts 3D eye positions to emulate actual eye movements allows such a comparison.

There are situations in which temporally accurate measurement of movements is indispensable. As an example, the trajectories of certain types of eye movements (e.g., nystagmus secondary to a labyrinthine imbalance) is seen only when fixation is removed, but the observation requires an infrared video camera (for recording in complete darkness) or Frenzel goggles (to eliminate fixation). Also, accurate visualization of the trajectories of normal rapid eye movements, such as saccades or the vestibulo-ocular reflex (VOR) during high-speed head impulse testing (Hong et al., 2008), requires a specialized video recorder that can capture images at frequencies above the usual 30 fps. Our software eliminates the need for special equipment to visualize such data. The software will simply play back the reconstructed 3D eye movements from the recorded data; and this can be done by acquiring the data using the commonly used search coil technique. Using the "slow motion", and "exaggeration" functions, even the small and high frequency eye movements can be easily appreciated.

The software is freely available for download, see Supplementary materials 2 (software) and 3 (sample data).

Conflict of interest statement

The authors declare that they have no competing financial interests

Acknowledgments

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at doi:10.1016/j.jneumeth.2011.06.002.

References

Buchberger M, Kaltofen T, Priglinger S, Hoerantner R. Construction and application of an object-oriented computer model for simulating ocular positioning defects. Spektrum Augenh 2003;17:151–7.

Goldstein H. Classical mechanics. Reading, MA: Addison-Wesley; 1980.

Haslwanter T. Mathematics of three-dimensional eye rotations. Vision Res 1995;35:1727–39.

Hong S, Leigh RJ, Zee DS, Optican LM. Inferior olive hypertrophy and cerebellar learning are both needed to explain ocular oscillations in oculopalatal tremor. Prog Brain Res 2008;171:219–26.

Leigh RJ, Zee DS. The neurology of eye movements. 4th ed. USA: Oxford University Press; 2006.

Nakada T, Kwee IL, Seesaw nystagmus. Role of visuovestibular interaction in its pathogenesis. J Clin Neuroophthalmol 1988;8:171–7.

Robinson DA. A method of measuring eye movement using a scleral search coil in a magnetic field. IEEE Trans Biomed Eng 1963;10:137–45.

Schreiber KM, Schor CM. A virtual ophthalmotrope illustrating oculomotor coordinate systems and retinal projection geometry. J Vis 2007;7(4):1–14.

Shaikh AG, Miura K, Optican LM, Ramat S, Leigh RJ, Zee DS. A new familial disease of saccadic oscillations and limb tremor provides clues to mechanisms of common tremor disorders. Brain 2007;130:3020–31.

Shaikh AG, Ramat S, Optican LM, Miura K, Leigh RJ, Zee DS. Saccadic burst cell membrane dysfunction is responsible for saccadic oscillations. J Neuroophthalmol 2008;28:329–36.