Detecting repetitive pupil motion in nystagmus using first moment of pixel intensity and discrete period quadrature

Julia Foust[1] and George Stetten[1,2]

[1] Department of Bioengineering, University of Pittsburgh
[2] Robotics Institute, Carnegie Mellon University

Abstract. We present a novel method to determine frequency and phase of repetitive eye motion, using the widely available platforms of smartphones and tablets, requiring only the download of a free software app. Our purpose is to provide assistance to patients with nystagmus by presenting an image on the smart device's screen that moves in synchrony with the patient's periodic eye motion, thereby stabilizing it. The image may originate in real time from the device's back-facing camera, thus providing a stable view of the visual environment. The app would also gather eye motion data during activities of daily living to guide treatment and inform research into nystagmus and other diseases of the visual system. As opposed to most applications for tracking eye motion, ours is not based on first determining pupil location. Rather, our method simply identifies the central region of the face containing the eyes and then finds periodic variations in the first moment of pixel intensity within this region generated by movement of the iris and pupil relative to the visible portion of the sclera. We determine the frequency and phase of these variations using a novel method of time-frequency analysis we call *discrete period quadrature* (DPQ), which combines aspects of the traditional phase-locked loop and the discrete Fourier transform. Results are shown of applying the system to video image sequences that simulate the repetitive eye motions typical of nystagmus, reconstructed from still images of normal subjects gazing in predetermined directions.

Keywords: eye tracking, phase-locked loop, nystagmus.

1 Introduction

1.1 Current Treatments for Nystagmus

Nystagmus is a visual impairment characterized by involuntary, periodic eye motion. Approximately 785,000 individuals in the United States live with nystagmus [1], many of whom consequently suffer from oscillopsia, the perception of an unstable or moving visual field, which can cause difficulties with everyday tasks. Acquired nystagmus, developed later in life, tends to leave the patient with fewer adaptive skills, making them more likely candidates for treatment [2]. Current treatments for nystagmus include (1) corrective eyeglasses, often with only limited improvement, (2) medications, which provide irregular success and may have side effects, (3) invasive surgery, only applica-

ble in specific cases, and (4) rehabilitation therapy, typically requiring the use of expensive and cumbersome devices. A 2016 pilot study explored a novel method combining gaze-tracking with real-time visual feedback to stabilize retinal images for patients with downbeat (vertical) nystagmus [3]. Their system was shown to increase visual acuity by translating images viewed on a computer screen to compensate for the patient's periodic eye motion. It used sophisticated eye tracking equipment (EyeSeeCam®) mounted adjacent to the eyes with special lighting, making it expensive and intrusive. Nonetheless, the basic idea set the stage for the more practical system we are developing, building on the ubiquitous platforms of smartphones and tablet computers. Our system, which we call *StabilEyes*, employs a novel eye-tracking method, described here, which is tolerant of the lower resolution, greater variability in lighting conditions, and greater distance from the eyes inherent to the smart device cameras facing the user (the so called, "front-facing" camera).

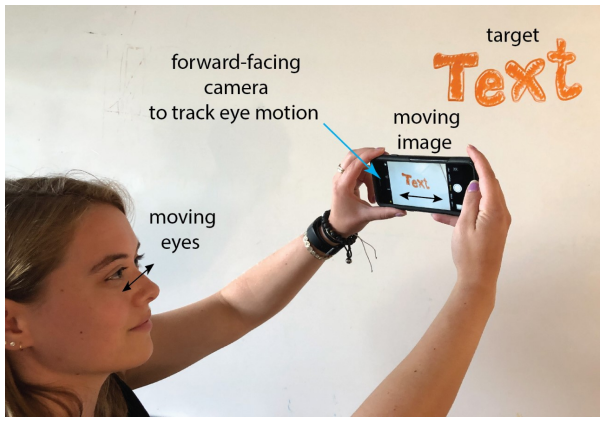

Fig. 1. Mock-up of proposed StabilEyes app on a smartphone, translating an image (in this case, of the word "Text" on a whiteboard) to compensate for periodic eye motion in nystagmus.

1.2 StabilEyes

Our goal is to develop a free software app to run on any smart device, to provide assistance for individuals with nystagmus. Our approach focuses on accessibility in daily life, leveraging technology that most people already have in their possession. The app, called StabilEyes, will utilize both the front- and back-facing cameras of the smart device to display a stable image for the user. Specifically, the front-facing camera will detect and track the user's eye motion while the back-facing camera will capture real-time images of the field of view in front of the device (see Fig. 1). On the screen of the device, those real-time images will be displayed and translated sinusoidally with the same frequency and phase as the user's periodic eye motion, thus cancelling much of the unwanted motion due to nystagmus. We expect to provide the user with the ability to adjust the amplitude of the sinusoid manually to optimize the perceived stability of the image at different distances between the screen and the patient's eyes. Nystagmus patients will thus be able to look "through" their phones and perceive a stable view of the world around them.

2 Eye Motion from First Moment of Intensity

2.1 Determining Eye Box Location

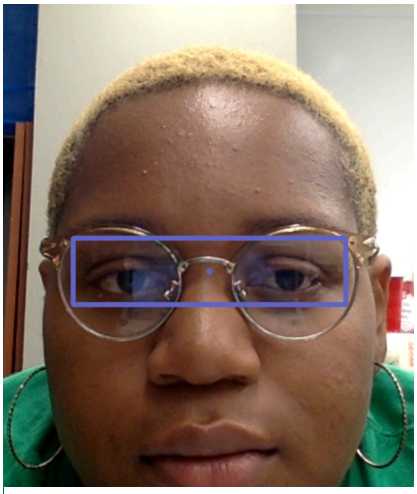

Fig. 2. Eye box around both eyes on a subject, in which the first moment of pixel intensity is computed to detect eye motion.

Detecting eye motion is typically accomplished via pupil tracking, with recent machine learning advancements increasing the potential for such techniques to be used effectively in daily life [4,5]. Compared to such techniques, we are at a unique advantage in our needs for eye motion detection – instead of locating the pupil in every frame to get absolute position, we simply need to detect the frequency and phase of the periodic motion of the eyes. To that end, we have developed a simple algorithm using the first moment of pixel intensity in a rectangular region containing both eyes. That region is found in the initial video frame using the *Haar Cascades* facial detection algorithm as implemented in the open-source computer vision library, OpenCV, to determine the location of all faces in the image [8]. The user's face is assumed to be the largest, thus allowing for other people to be present in the background. Given the location of the user's face, a rectangular region containing the eyes (eye box) is found based on an estimation of the average location of the eyes relative to the face (see Fig. 2). For subsequent frames, an optical flow method based on sum-absolute-difference (SAD) translates the eye box to keep it properly registered with the face, allowing for movement of the head relative to the camera.

2.2 Finding First Moment of Pixel Intensity with the Eye Box

To detect eye motion, the first moment M_x of the intensity $I(x,y)$ of pixels contained within the eye box B is calculated in each frame according to Eq. 1.

$$M_x = \frac{\sum_{(x,y)\in B} x \cdot I(x,y)}{\sum_{(x,y)\in B} I(x,y)} \tag{1}$$

As the iris and pupil move relative to the sclera (whites of the eyes), that motion is conveyed through the moment calculations. To evaluate this method, we developed "pseudo-nystagmus" image sequences in which periodic eye motion is simulated from a set of individual images over which predetermined gaze location varies horizontally. These sequences included individuals of various races and genders, as well as with and without glasses. The results from calculating M_x for eye boxes located in images from one such sequence vs. known gaze location are shown in Fig. 3. A linear best fit yielded a correlation coefficient of 0.8746.

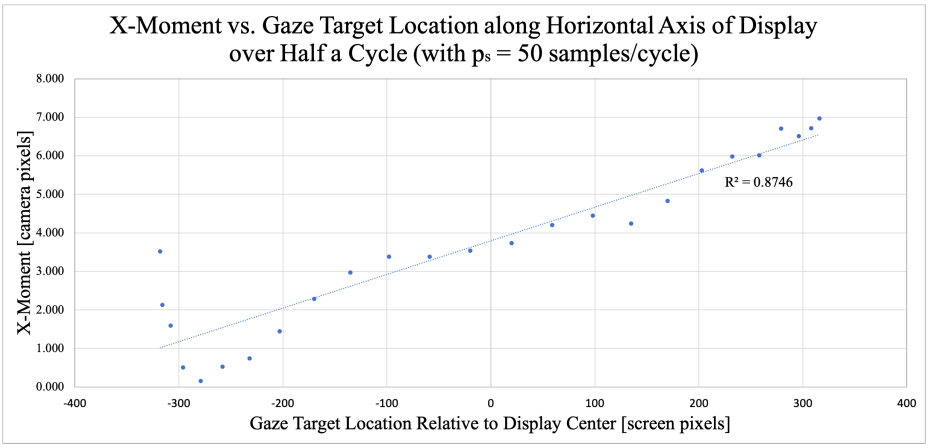

Fig. 3. Moment calculations compared to the known location of the gaze target for a pseudo-nystagmus image sequence.

3 Tracking of Frequency and Phase of First Moment

The variability in nystagmus eye motion and the unreliable quality of images of the eye captured by the front-facing camera in differing environments necessitate a robust method for tracking the frequency and phase of the periodic first moment signal just described. Moreover, the method must be able to follow changes in frequency with agility, preserving accurate phase from one sample of the signal to the next. An obvious candidate for this is the phase-locked loop.

3.1 Phase-Locked Loop

The phase-locked loop (PLL) is a classical control theory construct originally used in analog electronics for frequency modulation (FM) radio receivers and other applications that required tracking a signal with a rapidly changing frequency. It typically consists of an oscillator in the receiver whose frequency is controlled by comparing its own phase with the phase of the input signal. If the phase of the input signal gets ahead of the oscillator, the frequency of the oscillator is increased to compensate and stay in phase. The opposite occurs if the phase of the input signal falls behind. Thus, the oscillator is "phase-locked" to the input signal, and therefore maintains a frequency as close as possible to the frequency of the input signal. The process may be compared to that of a musician staying in sync with another musician whose tempo is varying beat-to-beat.

The concept of the PLL has been transferred to software algorithms by numerous researchers [6,7]. Preserved throughout are generally two modes operation: (1) Capture, in which the oscillator searches for and initially finds the periodicity in the input signal

and locks phase with it, and (2) Tracking, in which the oscillator remains in phase by adjusting its own frequency as described above.

We originally explored the PLL for capturing and tracking the frequency and determining the phase of the first moment signal, but we found that most PLL algorithms assume either a sinusoidal or square-wave input signal, achieved by filtering and/or thresholding the input signal. Although the eye motion of periodic nystagmus and its resulting first moment intensity signal may sometimes be roughly sinusoidal, they are often not symmetrical in terms of movement in one direction vs. the other. This leads to harmonics, which some PLL circuits ignore by correlating with a pair of sinusoids 90° apart. This "quadrature detection" guarantees detection of the signal with any phase, and determination of that phase. It is basically a Fourier transform at a single frequency, and it permits very rapid and accurate tracking of frequency by phase-locking the sinusoids to the input signal. However, it also leads to difficulties during the capture phase of operation, especially when the input signal's frequency differs significantly from that of the sinusoids.

These difficulties inspired us to broaden the concept of the PLL to include a spectrum of frequencies. A major advantage of utilizing a spectrum is in its wide capture range when first determining the frequency of the signal before tracking. However, the classical spectral techniques, such as the discrete Fourier transform (DFT), operate over a fixed time window for all frequencies in the spectrum. This limits both the accuracy of the frequency measurement and the agility with which a particular frequency can be tracked. In fact, the larger the time window, the greater the accuracy of frequency determination, but the slower the response to changes in frequency. This tradeoff led us to develop a different spectral approach, which we call *discrete period quadrature*, in which the time window varies with the particular frequency considered. The window at any given frequency, in fact, consists of a single period at that frequency.

3.2 Discrete Period Quadrature

We have developed discrete period quadrature (DPQ) based on concepts foundational to the DFT to better suit our needs in rapidly and accurately tracking the frequency and phase of a periodic signal. Similar to the DFT, DPQ acts on a discretized deterministic signal to yield a complex value describing each possible constituent frequency. Specifically, DPQ correlates the signal with exactly one cycle of a sine and cosine of each discrete period considered as a possible constituent. DPQ is thus distinguished from the DFT by using a window whose duration adapts to the given frequency, optimizing it as a method for time-frequency analysis, or in our case specifically, for rapid and accurate tracking of a periodic signal.

We denote DPQ operating on a discrete signal $s[n]$ as $Q[p,n]$, defined as

$$Q[p,n] = \frac{1}{p}\sum_{m=n}^{n-p+1} s[m]\left(\cos\left(\frac{2\pi(m-n)}{p}\right) + j\sin\left(\frac{2\pi(m-n)}{p}\right)\right), 2 \le p \le P, \quad (1)$$

where the sum operates on p samples preceding and including the sample at index n. For each integer period p between 2 and a maximum P, this yields the quadrature co-

variance as a complex number at the corresponding discrete frequency $\omega = 2\pi/p$ radians/sample. The magnitude $|Q[p,n]|$ and phase $\angle Q[p,n]$ play similar roles to the magnitude and phase in the DFT. Calculating these values for each p creates a magnitude spectrum and a phase spectrum for each sample in time. As each new sample is processed, a new spectrum is produced, and may be analyzed to determine the fundamental period of the signal at that point in time.

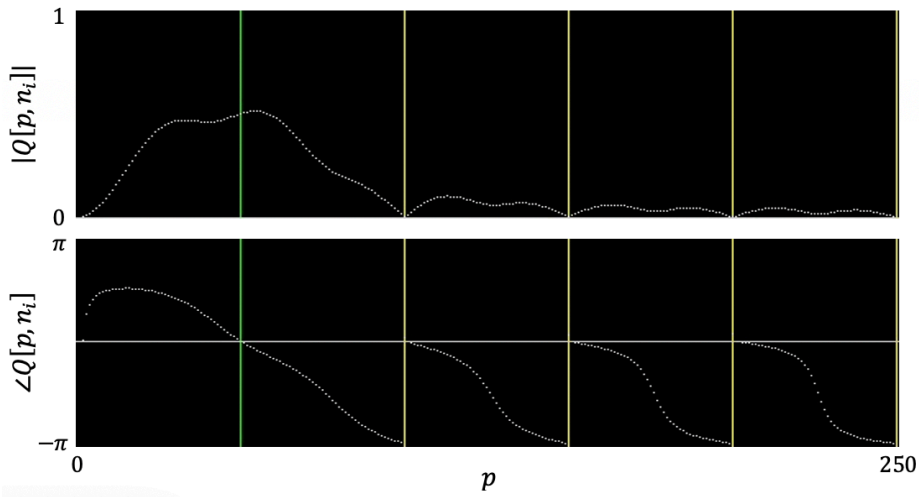

Fig. 4. Magnitude and phase components of the DPQ spectrum of a periodic signal for a particular sample n_i in time. The green vertical line marks the period of the signal (50 samples/cycle) with the yellow vertical lines located at its integer multiples of the signal period (i.e., subharmonics).

3.3 Interpreting Discrete Period Quadrature

Once the DPQ spectrum has been obtained, our goal is to interpret it to determine the predominant periodicity in a manner that is both robust to noise and quickly able to adapt to changes in the signal period. Ideally, the response should be phase-locked such that it tracks from one sample to the next.

Delta Phase. We identified several key features based on the DPQ with the potential to be used in tracking a deterministic signal. Ultimately, we determined that smoothness of phase change is optimal for inferring the correct signal period p_s. We know the correct rate of change of phase for each p to be equal to $2\pi/p$ radians/sample. For any particular values of p and n, the observed rate of change in phase from sample $n-1$ to sample n is

$$\angle Q[p,n] - \angle Q[p,n-1] \,, \tag{2}$$

and therefore the error in that phase change from the expected rate is

$$E[p,n] = \left[\angle Q[p,n] - \angle Q[p,n-1]\right] - \frac{2\pi}{p} . \tag{3}$$

The RMS error $E_{RMS}[p,n,k]$ of the rate of phase change over k *previous* samples is

$$E_{\mathrm{RMS}}[p,n,k] = \sqrt{\sum_{m=n}^{n-k+1} \frac{E[p,m]^2}{k}}, k \geq 1 \tag{4}$$

By comparing $E_{\mathrm{RMS}}[p,n,k]$ for every p at a particular point in time we should see a minimum at the correct period (for a given k),

$$p_s[n,k] = \arg\min_p E_{\mathrm{RMS}}[p,n,k] . \tag{5}$$

This feature was, in fact, first noticed in a series of magnitude and phase spectra produced over time. As the program implementing DPQ runs on an input signal, the magnitude and phase spectra are displayed and updated with each subsequent sample processed. One such pair of spectra are shown in Fig. 4. Observing the resulting spectra in real time, a smooth downward motion of the phase value through time was witnessed at the fundamental period (visualized as the green vertical line in Fig. 4), as compared to erratic behavior of time-varying phase elsewhere. Such behavior is, of course, expected – by definition, phase progresses smoothly with time at a signal's fundamental period.

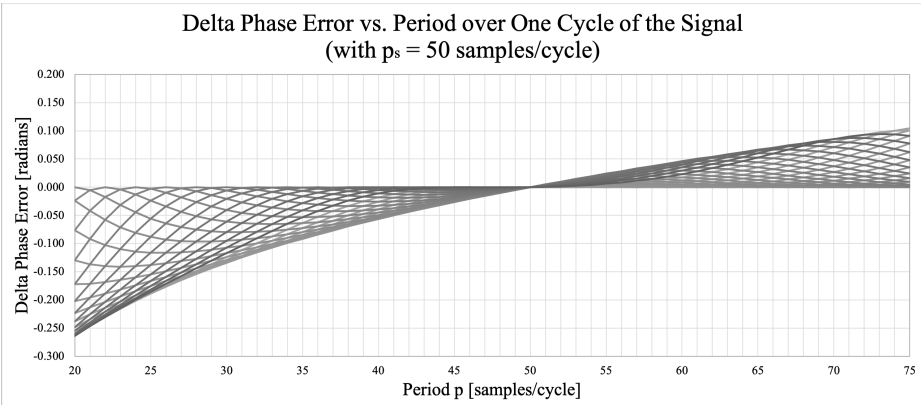

Fig. 5. Delta Phase error $E[p,n]$ at various values of p. Each plotted line represents a different sample in time n from one full cycle of the pure sinusoidal input signal.

The efficacy of this *Delta Phase* feature is further evidenced by Fig. 5, which shows the value of $E[p,n]$ as a function of p superimposed for every sample n from one cycle of the signal period, for a signal whose period is 50 samples/cycle. At p = 50 samples/cycle, the delta phase error $E[50,n]$ = 0 for all n. The progression of phase calculated at the signal period is not only smooth but also consistent with the expected

phase value, meaning that its RMS error over time will be zero as well. All other periods have nonzero delta phase error values at one or more samples in time, which causes their RMS error to be larger than zero.

3.4 Comparing Delta Phase with other Methods of Interpreting DPQ

We compared the use of the *Delta Phase* feature to three alternate features we had developed previously, with respect to their ability to infer signal period: *Peak*, *Subharmonic*, and *Variance*. We describe each here and include them in Figs. 7 and 8.

Peak. Given a sinusoidal signal with a period p_s, we expect $|Q[p,n]|$ to have a maximum at $p \cong p_s$ because the signal would correlate best with sine and cosine at the correct period. In Fig. 4, the green vertical line denotes the location of p_s. However, as portrayed by the magnitude spectrum for the particular time sample shown, there exists some fluctuation of the peak around the location of p_s. This is due to the truncation of the input signal when correlating with periods near to p_s, as the correlation may increase or decrease depending on the phase of the input signal and the particular section that is truncated.

Subharmonic. We further observe a minimum correlation $|Q[p,n]|$ (ideally zero) at subharmonics $p \cong kp_s$, $k = 2, 3, 4,....$, shown in Fig. 4 as yellow vertical lines. Presumably, this occurs because at these exact multiples of the period there are no truncation errors, and thus no correlation against truncated periods of the input signal to pollute the expected zero correlation. In our implementation of this feature, we use the expected minimum magnitude at the subharmonic $p = 2p_s$ to infer p_s.

Variance. Again due to truncation error, we also expect the variance of $|Q[p,n]|$ over time to have local minima at p_s and at subharmonics $p \cong kp_s$, $k = 2,3,4 ....$ as well. Compared to all other values of p, the truncation error is minimized (or, ideally, nonexistent) at the signal period and its subharmonics, and thus $|Q[p,n]|$ should be most consistent with lowest variance through time at these values of p.

To compare the Peak, Subharmonic, and Variance features with the Delta Phase feature, we tested their performance in two situations: (1) changing signal period at various rates, as shown in Fig. 6, and (2) adding different amounts of Gaussian noise to the signal, as shown in Fig. 7. As shown in these two experimental results, Delta Phase and Subharmonic are similarly superior to the other two in tracking at all rates of change for period, but Delta Phase is superior to all others in the presence of noise at all levels tested.

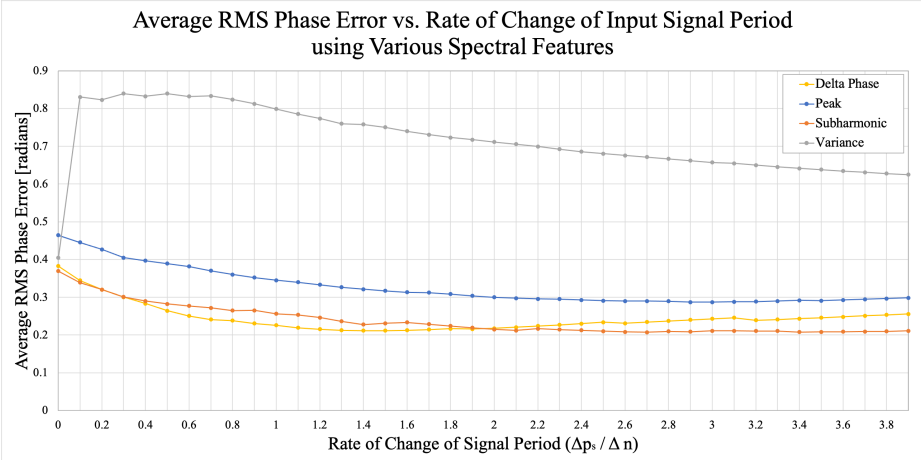

Fig. 6. RMS error between the signal phase and inferred phase during various rates of change of signal period. Each data point was averaged over the result of using a signal with $2 \leq p_s \leq 62$. Delta Phase and Subharmonic features perform similarly and are more accurate (i.e., produce the least amount of error) at all rates of change than Peak and Variance features.

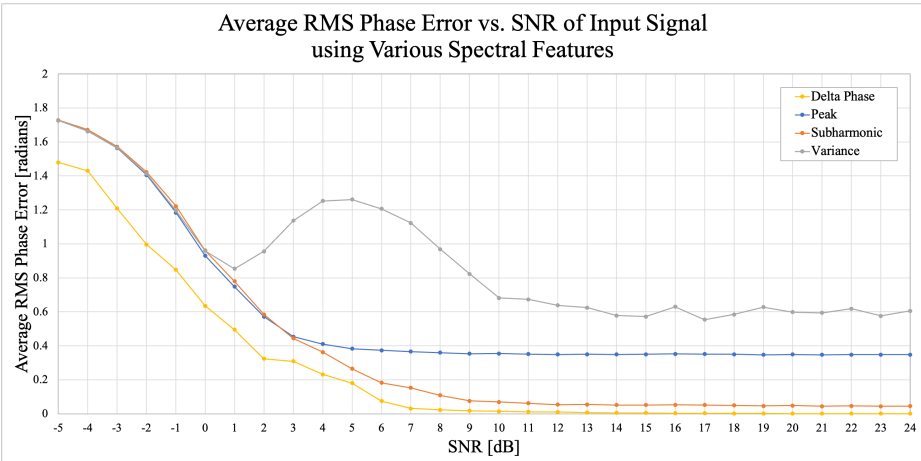

Fig. 7. RMS error between the signal phase and inferred phase with various signal-to-noise ratios (SNRs) from various levels of Gaussian noise added to a pure sinusoidal signal. Each data point was averaged over the result of using a signal with $2 \leq p_s \leq 62$. The Delta Phase feature outperforms Subharmonic, Peak, and Variance features.

4 Demonstration of Full System on Image Sequence

We demonstrated the full system by constructing a pseudo-nystagmus image sequence with sinusoidally varying gaze location and applying the methods described above to find the first moment in the x direction and then determine its frequency and phase

using the DPQ Delta Phase feature to infer gaze location. The results are shown in Fig. 8. The phase produced by DPQ is clearly locked to that of the input signal. As already mentioned, our system does not yield amplitude, and thus the inferred gaze location is displayed with a normalized amplitude of 1. In the eventual StabilEyes App, amplitude will be adjusted manually by the user to best match the translations of the displayed image to the corresponding periodic eye motion, given a particular distance between the screen and the patient's eyes.

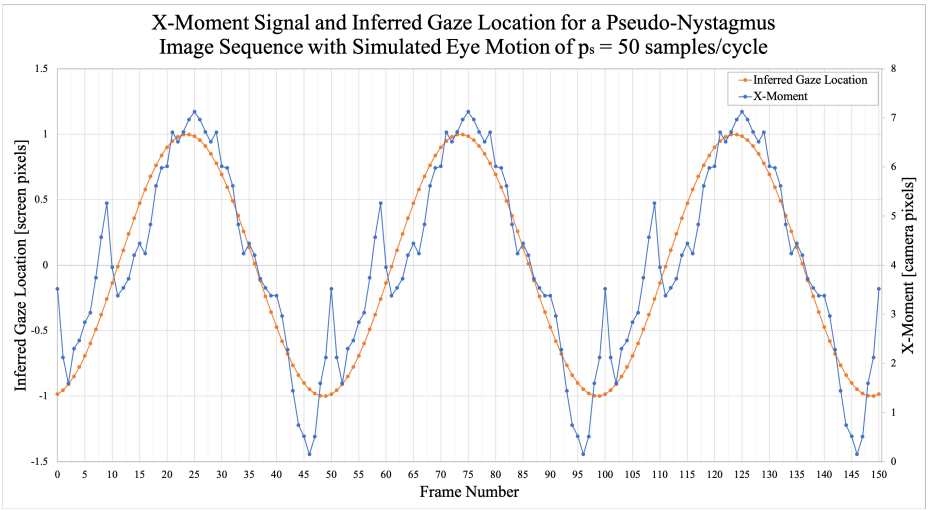

Fig. 8. X-Moment (blue) and corresponding Inferred Gaze Location (red) with period and phase determined by DPQ (and with amplitude normalized), for one series of pseudo-nystagmus images simulating horizontal sinusoidal eye motion.

4.1 Computation Speed

The programs described above were written in C++ using the OpenCV library and run on an Apple MacBook Pro. Without much effort so far at optimizing for speed, a framerate of 21 frames per second has been achieved for processing the pseudo-nystagmus image sequences. Eventually, the process would also need to include live video capture and image display translation, and it would be running on a slower smartphone or tablet computer platform. However, significant improvements are expected with greater attention paid to efficiency in developing the eventual deployed system.

5 Discussion and Future Work

We have developed a method for detecting periodic eye motion and determining the frequency and phase of that eye motion. We plan to deploy this method in a free software app for smartphones and tablet computers to aid nystagmus patients in their everyday lives. The technology is intended not only as an assistive device, but also as a tool for diagnostic assessment and for further research on nystagmus.

We are presently implementing StabilEyes on an Apple iPad and iPhone, and plan to begin testing on patients at the University of Pittsburgh Medical Center Department of Ophthalmology, once restrictions due to the COVID-19 pandemic are lifted.

By developing a method for determining periodic eye motion that can run on smartphones and tablets, we have leveraged these ubiquitous platforms to increase accessibility of our technology to a wide range of nystagmus patients. However, since our method yields only frequency and phase, the resulting compensatory translation can only be sinusoidal, and thus only match the complex periodic eye motion to a first approximation. We are presently exploring ways of expanding our present method to permit more exact matching to non-sinusoidal periodic eye motion. We are also considering ways to provide optimal amplitudes for the compensatory image translation automatically, rather than by manual adjustment, over a range distances from the screen to the patient's eyes.

Finally, we are exploring other applications for DPQ that require accurate phase tracking. For example, we are investigating its use in analyzing temporal correlation of regions in functional magnetic resonance imaging (fMRI) sequences of the brain acquired during the resting state. We have implemented a version of DPQ in Python using the Google Colab online environment, for convenient demonstration and testing by other researchers who may find applications for DPQ in tracking the phase and frequency of any periodic signal [9].

Acknowledgements

Funded by a Brackenridge Fellowship, a University Honors College Fellowship, and a grant from the Center for Medical Innovation, all at the University of Pittsburgh.

References

1. Sarvananthan, N., Surendran, M., Roberts, E. O., Jain, S., Thomas, S., Shah, N., Proudlock, F. A., Thompson, J. R., McLean, R. J., Degg, C., Woodruff, G., Gottlob, I.: The prevalence of nystagmus: the Leicestershire nystagmus survey. Investigative Ophthalmology & Visual Science, 50(11), 5201-5206 (2009).
2. Abadi, R.: Mechanisms underlying nystagmus. Journal of the Royal Society of Medicine 95(5), 231-234 (2002).
3. Teufel, J., Bardins, S., Spiegel, R., Kremmyda, O., Schneider, E., Strupp, M., Kalla, R.: Real-time computer based visual feedback improves visual acuity in downbeat nystagmus – a pilot study. Journal of NeuroEngineering and Rehabilitation, 13(1), (2016).
4. Zhang, X., Sugano, Y., Fritz, M., Bulling, A.: MPIIGaze: Real-world dataset and deep appearance-based gaze estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence 41(1), 162-175 (2019).
5. Krafka, K., Khosla, A., Kellnhofer, P., Kannan, H., Bhandarkar, S., Matusik, W., Torralba, A.: Eye tracking for everyone. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2176-2184. IEEE, Las Vegas, NV, USA (2016).
6. Kratyuk, V., Hanumolu, P. K., Moon, U., Mayaram , K.: A design procedure for all-digital phase-locked loops based on a charge-pump phase-locked-loop analogy. IEEE Transactions on Circuits and Systems II: Express Briefs, 54(3), 247-251 (2007).

7. Xiu, L., Li, W., Meiners, J., Padakanti, R.: A novel all-digital PLL with software adaptive filter. IEEE Journal of Solid-State Circuits, 39(3), 476-483 (2004).
8. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), pp. I-I. IEEE, Kauai, HI, USA (2001).
9. Foust, J: Discrete Period Quadrature, a Google Colab demo, https://tinyurl.com/y4n4eaje