EE 367/368 Project Proposal

Real-time Gazed-Controlled Digital Page Turning System

Yao Chen (yaochen1), Zhenzhi Xia (zxia3), Aida Zhumabekova (adazoom)

Introduction

For musicians, turning music sheets by hand while playing an instrument has always posed an unpleasant challenge. The musician often has to stop playing in the middle of the composition to turn the page, therefore a human page turner is commonly being used during a performance. However, during the personal practicing time musicians are left to deal with this problem on their own. A pianist for example, will usually either sacrifice several notes at the end of each page or spend considerable amount of time practicing quickly turning the page so that the music flow is not affected. Although there are a fair amount of digital page turning applications using bluetooth connected foot pedal or music note detection available on the market, such devices don't provide musicians with enough autonomy that they may occupy extra foot motion while a pianist needs to pedal the music or, which is the more common downfall, the note detection algorithm will turn the page way too early causing a lot of frustration for musician. Therefore, we would like to provide a more accessible page turning solution to free musicians' hands and feet and let them devote wholeheartedly to playing music during their practice sessions and even performance. We will do so by providing a software that tracks their eyeball movement simply by using the webcam and turns pages automatically when it receives the indication to do so. We chose the webcam as our medium for this solution because many music players today heavily rely on the electronic music sheets and almost every electronic device has a webcam attached.

Related work

As a Human-Computer Interaction system, page turning system consists of a real-time image acquisition system incorporated with the hardware of electronic device, an eye tracking system, and a feedback system for page movement. The core of the page turning system is to track the page turner's eye position e.g. gaze, distance and its orientation in real time. Tremendous work has been done in this field [1-5]. Traditional approach detects eyes position on the image based on the intensity or color distribution, which requires a large amount of training data to accurately distinguish purpils from different objects. Eigen-face detection[5] could be also extended to the eigen-eye detection. To ease the computational burden of pre-processing the training data, Zhiwei Zhu et al. proposed a methodology for real-time eye detection and tracking under various lighting conditions by combining the bright-pupil effect and the conventional object detection techniques. Morimoto et al. took advantage of pupil-corneal reflection technique for gaze tracking.

Approach

Hardware preparation

- Use a logitech webcam instead of laptop's native webcam for the first stage to live stream the player's head image, the streamed images will be processed to contain most information of head and eyeball movement, and gaze duration.

Dataset

- We would use training images of eyeball images available online to start the eye-tracking algorithm
- The final dataset would be retrieved from the live stream of head and eyeball movement of the person sitting in front of the device in real time, the distance to the tablet might vary due to different purposes.

Calibration

- We will infer the eye-gaze locations a user on a page in real time by using image processing techniques we learned in the class. We will decompose the live video feed into frames and apply `vision.CascadeObjectDetector` to locate and track eyes.
- To further extract irises from within the eye region we will use image processing to detect circles in the image region. This will allow us to compute the delta of iris movement from the video feed stream.
- We will specify corner regions on each page which will act as "hot corners" to signal the page turning. The user will then activate the "hot corner" by staring directly at it for a specified short period of time.
- We will infer the point of eye gaze intersection with the page by treating the gaze as a ray coming directly from the iris and calculating where it intersects with the plane of the page region within the application window.

Milestones and timeline

Week	Task
1	Set up the live video feed routing
2	Enable eyes + iris detection on the single frame
3	Process the stream of the frames in real time and track user's iris movement
4	Add "hot corners" and enable page turning
5	Optimization and debugging

Caveats

- The iris tracking can possibly be problematic because the natural iris movement is very quick and chaotic. We will focus on the implementation that relies on the user to maintain the stare at one of the specified "hot corners"
- The real time processing can become another challenge since that would imply considerable time constraints under which we have to produce a reliable output. As a possible solution we can perform our computation on GPU or use parallelization processing to speed up the biggest bottlenecks.

References

[1] Zhu, Zhiwei, Kikuo Fujimura, and Qiang Ji. "Real-time eye detection and tracking under various light conditions." *Proceedings of the 2002 symposium on Eye tracking research & applications*. ACM, 2002
[2] Morimoto, Carlos H., and Marcio RM Mimica. "Eye gaze tracking techniques for interactive applications." Computer vision and image understanding 98.1 (2005): 4-24.
[3] Bellamy, Rachel KE, et al. "Correcting systematic calibration errors in eye tracking data." U.S. Patent No. 9,782,069. 10 Oct. 2017.
[4] Nguyen, Phi Bang, et al. "Method for calibration free gaze tracking using low cost camera." U.S. Patent Application No. 14/389,783.
[5] Turk, Matthew, and Alex Pentland. "Eigenfaces for recognition." *Journal of cognitive neuroscience* 3.1 (1991): 71-86