Mobile Eye Tracking
- Expanding rapidly
- Implemented widely, such as in advertising, medicine, and scientific research
- Eye movements can reliably indicate certain thought processes

Current methods
- Time-consuming and onerous
- Require researchers to manually inspect each frame of the video
- to determine fixation points

Goals of the project
- Develop imaging algorithms capable of detecting objects in a dynamically changing scene.
- Improve existing computer vision packages to develop tools to automatically identify objects in an environment.

What have we done?
- A toolkit to help researcher to process Eye tracking data
- A fast and robust object identification algorithm
- A system analyzing the result

Multiple Object Identification

Original SIFT with Brute-Force matching
SIFT feature with Flann-Based matching
Accurately detect multiple objects and feature points

What have we done?
- A toolkit to help researcher to process Eye tracking data
- A fast and robust object identification algorithm
- A system analyzing the result

Multiple Object Identification

Original SIFT with Brute-Force matching
SIFT feature with Flann-Based matching
Accurately detect multiple objects and feature points

Blackout Method – Multiple Occurrences Object Detection

Erratic boundaries: multiple object occurrences
Black-out method to improve object detection
Accurately detect multiple occurrences of objects
Equipment and Data Collection

- SMI Eye Tracking Glasses were used
- Data generated are videos with subject gaze points overlaid
- Created image processing software using OpenCV, Python, and Python libraries
- OpenCV gave us access to many computer vision algorithms
- Algorithms used to analyze eye tracking videos include Scale Invariant Feature Transform (SIFT) and Fast Approximate Nearest Neighbor Search Library (FLANN)

Multiple Processing

- Runtime for a 40s video is around 90 minutes
- Divide video into separate CPU processes
- Reduced runtime proportional to number of processes

Graphical User Interface

- Create a robust, interactive GUI for easier use
- Select multiple reference images and processing methods
- Drag and select reference image from the GUI interface

Future Projects

- Create a structure to hold metadata about each frame
- Use previous frames as templates for subsequent frames
- Create a lightweight, initial processing run to mark frame differences to determine discrepancies between frames and optimize processing

Contact: dy46@duke.edu, xc77@duke.edu, db240@duke.edu, njs28@duke.edu, jaw69@duke.edu

https://bit.ly/2ajvu0W

https://github.com/dy46/Eye-Tracking