Real-Time Eye Gaze Estimation for Dynamic Head Movement

Amirmahdi Aboutalebi
University of Genova
Amir.abootalebi2001@gmail.com

***Abstract*—This report presents an approach to real-time eyes gaze estimation using Mediapipe's face mesh model and dynamic calibration. The primary objective is to account for head movements, ensuring accurate gaze estimating on a calibrated display area. The system incorporates calibration by locking gaze positions at the corners of the screen and dynamically adjust for face positioning and scaling. Experimental results demonstrate robust tracking accuracy within the defined calibration area.**

I. Introduction

Eye gaze estimation plays a vital role in human-computer interaction, enabling applications such as assistive technologies, gaze-controlled interfaces, and from pandemic cheating online exam tracking. This report address the challenge by dynamically calibrating the model, ensuring robust gaze tracking by compensating for changes in face position and scale. The novelty lies in using relative positional offsets between facial landmarks and calibration boundaries to achieve robustness under variable head movements.

II. Methodology

A. System Design

The proposed method leverages Mediapipe's Face Mesh solution to detect key facial landmarks. By focusing on the left eye landmarks (It will possible for both the eyes but in this report only the left eye will be discussed), the system identifies the gaze position and dynamically calibrates it to predefined screen regions. Key steps include:

1) Detecting facial landmarks using Mediapipe's Face Mesh. (Figure 1)
2) Defining a exact left gaze landmark position. (Figure 2)
3) Defining a bounding rectangle around the left eye using specific landmarks (e.g., 33, 159, 155, and 144) (Figure 3).
4) Calibrating by locking gaze points at the screen's top-left (Figure 6) and bottom-right (Figure 7) corners.
5) Dynamically resizing the calibration boundaries based on relative offsets to maintain accuracy under head movements.
6) Scaling the relative gaze displacement from the calibration box to match the physical screen dimensions.

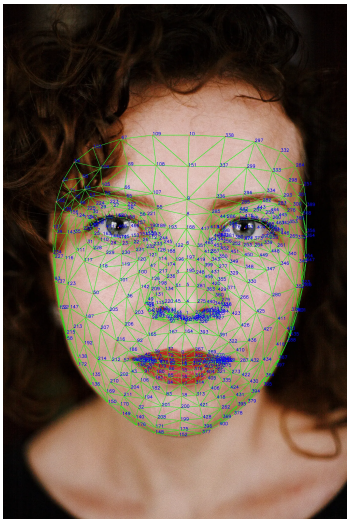

Fig. 1. Visualization of landmarks from MediaPipe Face Mesh used for gaze estimation. Key points around the eyes, eyebrows, and facial contours are highlighted, forming the basis for accurate tracking and estimation of gaze direction in computer vision applications.

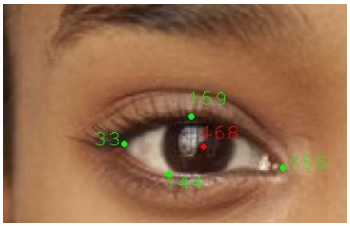

Fig. 2. landmark 468 is for the left eye gaze, and those other four landmarks cover the left eye (used to build eye boundary around the left eye)

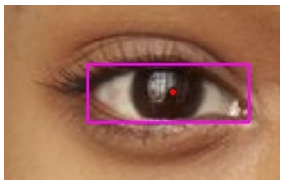

Fig. 3. With a given landmarks approximately considering a box boundary around the left eye

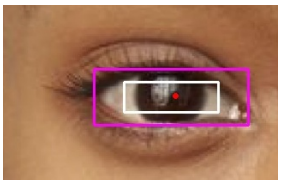

Fig. 4. Considering a boundary around top left point of screen and bottom right as a callibration boundary

B. Calibration Process

Calibration is a crucial step that ensures accurate mapping of gaze positions. Instead of directly storing screen gaze coordinates during calibration, the system calculates and saves the offset between the calibration points and the detected eye

boundary. The steps are as follows: (Figure 5)

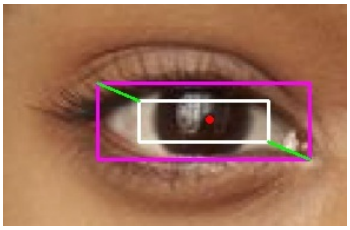

Fig. 5. Instead of saving specific point that are static, we save the distance of top left point of the eye boundary and calibration box and the same for bottom right, that can be adjusted during head adjustment of the users. In this way when the eye detected the dynamic calibration will be stimated for ongoing position

1) The user is instructed to look at the top-left corner of the screen. The system computes and stores the distance between the top-left corner of the eye boundary and the gaze point within the calibration rectangle.
2) Similarly, the user is instructed to look at the bottom-right corner of the screen. The system records the offset between the eye boundary's bottom-right corner and the gaze point.
3) These offset values allow the system to dynamically reconstruct the calibration rectangle even when the user's head position or orientation changes.

With this approach, the calibration boundary adapts dynamically by factoring in changes in face scale and orientation relative to the camera.

C. Gaze Tracking and Scaling

Once the calibration is complete, the system tracks the gaze movement within the adjusted calibration boundaries. Instead of detecting an absolute gaze position on the screen, the gaze movement relative to the calibration boundary is calculated. This displacement is then scaled to the physical screen dimensions:

- Calculate the change in gaze position within the calibration rectangle.
- Multiply this relative change by the ratio of screen dimensions to the calibration boundary dimensions. For example, if the calibration box dimensions are 192×108, and the screen dimensions are 1920×1080, a relative displacement of 2 pixels corresponds to a 20-pixel movement on the screen.

This ensures the gaze tracking remains accurate regardless of dynamic changes in user positioning.

III. Results

Experimental results were collected by tracking eye movements on a recorded video that was attached to the project and displaying them on a scaled grid. Figures 10 and 13 visualize the tracked gaze points during runtime.

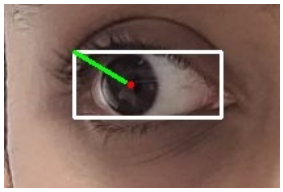

Fig. 6. Calibration at Top-Left Corner: "The user gazes at the top-left corner of the screen for calibration, with the eye boundary box dynamically linked to the upper-left calibration boundary."

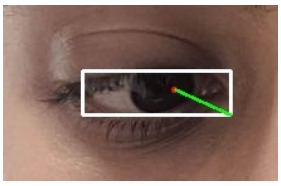

Fig. 7. Calibration at Bottom-Right Corner: "The user looks at the bottom-right corner of the screen for calibration, establishing the lower-right calibration boundary relative to the eye boundary box."

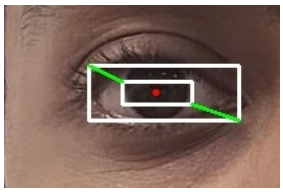

Fig. 8. Gaze Position at the Center of the Screen: "An example illustrating the user's gaze at the center of the screen, mapped accurately within the defined calibration box after successful dynamic scaling."

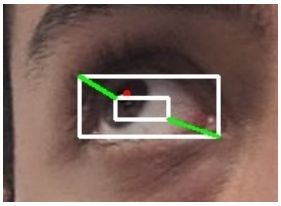

Fig. 9. Gaze Outside the Screen Calibration Boundary: "An example demonstrating the user's gaze positioned outside the screen's calibration boundary, with tracking indicating the point is beyond the predefined limits."

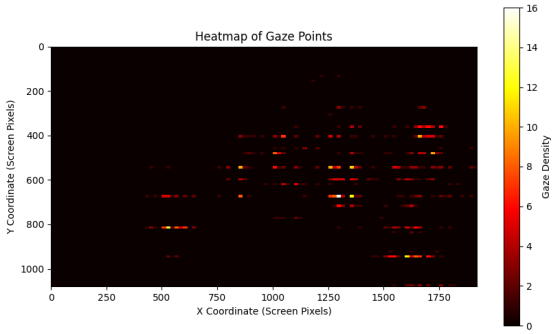

Fig. 10. Figure shows the heat map of eye movements

IV. Discussion

A. Challenges and Limitations in the Current Approach

The performance of the proposed system is heavily dependent on the accuracy of the MediaPipe Face Mesh solution. Under optimal conditions, MediaPipe reliably detects key facial landmarks; however, real-world applications often introduce several challenges that degrade performance:

1) **Lighting Conditions:** Variations in lighting, such as low-light or high-glare environments, can significantly impact landmark detection accuracy.

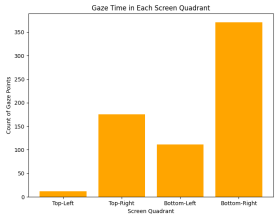

Fig. 11. Screen has been splitted into 4 parts, Top-left, Top-right, Bottom-Left, Bottom-right, this bar chart disply the movements of eyes during the screen in each parts.

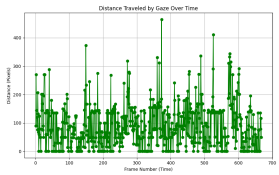

Fig. 12. Distances which has been traveled by gaze over time stemp.

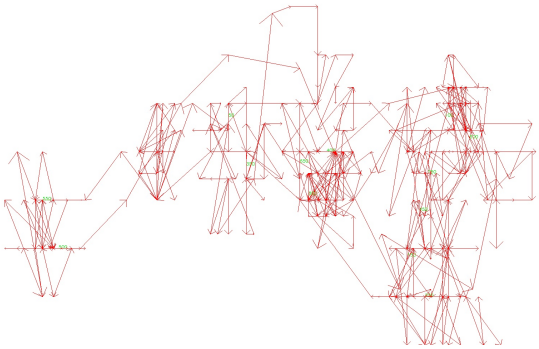

Fig. 13. The figure illustrates the tracked eye movement trajectory on the screen using red arrowed lines. Each arrow represents the direction and order of gaze shifts, providing insights into the user's visual attention and interaction patterns.

2) **Camera Quality:** Cameras with low resolution, compression artifacts, or limited dynamic range may result in unreliable facial landmark tracking.
3) **Dynamic Movements:** User movements, such as rapid head rotations or changes in posture, as well as facial occlusions (e.g., glasses or hair), can cause MediaPipe to fail in detecting key landmarks.

These challenges collectively hinder robust gaze tracking, especially when used in diverse environments with varying user conditions.

B. Proposed Solutions for Robust Gaze Estimation

To address these challenges, several enhancements can be explored:

Improved Landmark Detection: One solution involves combining landmark detection across multiple frames using temporal smoothing techniques such as Kalman filters. This approach would stabilize tracking and reduce the effect of missing detections caused by rapid head movements or occlusions. Moreover, integrating 3D depth estimation could assist in understanding head pose, enabling the system to compensate for user rotation angles.

Lighting and Camera Adjustments: Improvements in preprocessing can mitigate issues caused by poor lighting or low camera quality. Techniques such as histogram equalization, adaptive contrast enhancement, or the use of high-dynamic-range cameras can enhance the input image quality. Additionally, extending the training dataset of MediaPipe with images captured under extreme lighting conditions will help improve its robustness.

V. Future Work

A. Toward Automation: Using CNN and Advanced Learning Techniques

Although the proposed method achieves robust tracking through dynamic calibration, an automated solution could further reduce user involvement and improve scalability for real-world applications. CNNs, due to their ability to learn spatial relationships in image data, offer a promising direction for predicting gaze positions without explicit calibration.

A key improvement involves automating the calibration step by leveraging CNNs. By dynamically determining the user's distance from the screen based on the inter-eye distance, the system can infer scaling parameters to map gaze points onto the display. Gathering data across varying screen sizes, eye positions, and camera settings will enable the training of robust predictive models for this purpose.

To address the limitation of labeled data, self-supervised learning approaches could extract relevant features from large unlabeled datasets. Techniques like contrastive learning enable models to distinguish between different gaze orientations by focusing on feature representation from related gaze patterns. Once pre-trained using a large corpus of unlabeled data, these models can be fine-tuned using limited labeled data for highly accurate gaze prediction.

Semi-supervised learning could bridge the gap between labeled and unlabeled data, utilizing both to improve the model's performance. By combining labeled gaze data from dynamic calibration with extensive unlabeled data, the system can generalize better across diverse users and settings. These advanced techniques, when integrated, will help automate gaze estimation with minimal user involvement, reduce error rates, and eliminate the need for manual calibration.