

Main Program Issues and Solutions Log

1. Problem: Efficiency issues with video processing.

Solution: I chose to process only one frame per second instead of every frame, greatly improving efficiency.

2. Problem: An error in the code logic caused every frame to be printed out.

Solution: I adjusted the frame processing logic to ensure only one process and output per second.

3. Problem: When changing the color of the crosshair, the original code only changed the outer ring, and not the center cross shape.

Solution: I designed a new function to change the color of the crosshair, making sure the entire crosshair (including the center) changes color.

4. Problem: An error occurred when trying to change the color of the crosshair because the color values were incorrectly interpreted as strings instead of integers.

Solution: I modified the storage format of the colors to ensure they are integer tuples, and then updated the code accordingly.

5. Problem: When changing the color of the crosshair, the transparent areas were also mistakenly altered.

Solution: I more accurately defined a mask that only selects those pixels close to black and not transparent for alteration.

6. Problem: Some black residual edges when changing the crosshair color to white.

Solution: I fine-tuned the color change function by appropriately adjusting the threshold to ensure all required pixels are changed.

7. Problem: Accuracy issues in extracting text from video frames.

Solution: I used the Tesseract OCR library to extract text information from video frames. To improve accuracy, I also discussed possibly needing to preprocess the video frames, like adjusting contrast and brightness, but later found that using the original frames worked well.

8. Problem: How to accurately extract direction and tilt information from OCR output.

Solution: I used regular expressions to precisely match and extract the needed text segments. This ensured I could accurately get data from complex OCR outputs.

9. Problem: How to determine the phone's orientation (front, back, left, right) based on extracted direction data.

Solution: I designed a function to calculate the angle difference between the direction extracted from OCR and the user-input direction, using this difference to determine the phone's orientation.

10. Problem: How to determine if video frames were shot during the day or at night.

Solution: I developed a function to calculate the average brightness of the video frames and used a predefined brightness threshold to decide if it's day or night.

11. Problem: How to determine if the phone is facing up or down.

Solution: I judged based on the tilt information extracted from the OCR output. If the tilt is within a certain range, we consider the phone as facing up, otherwise facing down.

12. Problem: Sometimes logic errors or undefined variables occur when attempting to modify the code.

Solution: I reviewed the code step by step, ensuring all variables are properly defined, all functions are correctly called, and the logical flow is continuous.