

RELAZIONE DI PROGETTO

Pupil tracking

Componenti del gruppo:

- *Lazzari Matteo*

Indice

1	Introduzione	2
2	Analisi dei requisiti	3
3	Progettazione	4
4	Implementazione	7
5	Risultati ottenuti	11
6	Conclusioni e sviluppi futuri	12

1 Introduzione

Il progetto ha lo scopo di realizzare un sistema realtime per il tracciamento della pupilla per il movimento del puntatore del mouse sullo schermo, mediante l'utilizzo di linguaggio c++ e libreria OpenCV. Uno dei principali punti è quello di effettuare tutti i calcoli in realtime, mediante l'utilizzo di una telecamera che andrà ad ottenere il video della scena che sarà analizzato.

Il tracking della pupilla può essere applicato a diversi settori, quali militare e automotive.

Uno dei requisiti fondamentali è stato quello di rendere il sistema il più reattivo e veloce possibile, andando a minimizzare la latenza che si ottiene dall'ottenimento del frame del video all'azione finale da eseguire.

2 Analisi dei requisiti

Durante la fase di studio del progetto, sono stati analizzati quali possano essere i requisiti del progetto e quali sono i vincoli che sono stati imposti durante la fase di sviluppo.

I vari requisiti sono elencati qui sotto.

Requisiti funzionali

Questi requisiti descrivono le funzionalità specifiche che il sistema informatico deve implementare. Questi requisiti specificano ciò che il sistema deve fare, come deve comportarsi e quali azioni deve consentire agli utenti.

- L'utente deve avere la possibilità di muovere il cursore del mouse tramite il solo movimento dell'occhio;
- L'utente deve avere la possibilità di utilizzare il sistema dopo previa calibrazione.

Requisiti non funzionali

Questi requisiti rappresentano i criteri che definiscono la qualità del sistema, ma non riguardano specificamente le funzionalità.

- Il sistema deve essere il più veloce possibile nella computazione, per non aggiungere latenza tra il frame attuale e il risultato della computazione;
- L'interfaccia grafica deve essere il più semplice possibile e guidare l'utente nell'utilizzo.

Vincoli

Sono stati definiti dei vincoli al progetto, per semplificarne lo sviluppo.

- Il sistema presenta dei parametri di configurazione manuale che devono essere impostati manualmente a seconda della luce nella stanza;
- Il sistema deve essere utilizzato in presenza di luce con intensità costante.

3 Progettazione

Nella fase di progettazione, sono stati sviluppati 3 prototipi con i relativi pro e contro, per verificare quale fosse il miglior approccio e utilizzo nei vari casi. Ognuno di essi presenta la stessa logica di base, ma verranno applicate metodologie diverse per quanto riguarda l'individuazione della posizione dell'occhio. Questi saranno poi discussi nel dettaglio nella sezione di Implementazione.

Durate la fase di progettazione, sono stati individuati alcuni metodi di cattura dell'occhio che serviranno per delimitare la sezione dell'occhio rispetto al resto dell'immagine. In particolare, sono stati identificati 3 punti principali, che coincidono con gli approcci applicati nei vari prototipi.

- Identificazione dell'occhio mediante machine learning;
- Identificazione dell'occhio tramite window;
- Identificazione dell'occhio tramite approccio ibrido (automatico e manuale).

Essendo che la camera potrebbe presentare problemi di noise, dovuto da fattori interni o esterni, è stata eseguita una calibrazione iniziale, oltre all'utilizzo della soppressione dei neri, per far sì che la camera abbia l'immagine più coerente con la realtà, riducendo al minimo noise o altri errori.

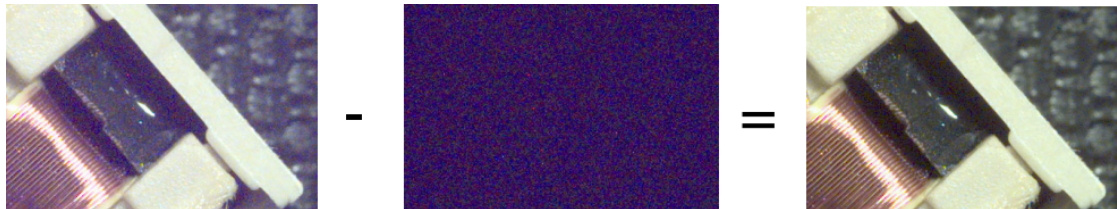


Figura 1: Esempio di come avviene la soppressione dei neri.

Al termine della fase di analisi per la riduzione del noise/errori, sono state analizzate diverse tecniche per ottenere la pupilla dall'immagine. Qui sotto sono elencati i principali metodi che sono stati trovati, con i relativi pro e contro.

- Utilizzo di algoritmi di machine learning per l'identificazione della pupilla.
In particolare, sul web sono già presenti reti pre-trained per questo scopo,

quindi si tratta solo dell'inserirle all'interno del progetto e adeguarle. Il problema principale è che un algoritmo del genere porta a rallentamenti notevoli del sistema, andando ad aggiungere una notevole latenza tra l'ottenimento del frame e l'effettiva terminazione della computazione, andando comunque ad ottenere un buon risultato (possibili miss da parte dell'algoritmo, ma offre un eccellente risultato finale);

- Utilizzo di algoritmi di edge detection per identificare il cerchio della pupilla e utilizzare poi la sua posizione per il tracciamento su schermo. Questo approccio porta ad ottime performance, in quanto non sono presenti algoritmi di machine learning, quindi si tratta di puri calcoli matematici e logica, andando a fornire le giuste performance. Inoltre, lavorando con una window di piccole dimensioni, non si è impattati da problemi che possono insorgere nel calcolo degli edge su frame di grandi dimensioni. Tuttavia, senza un'adeguata impostazione dei parametri per identificare al meglio la pupilla, si ha la possibilità di ottenere dei falsi-positivi;
- Utilizzare un approccio ibrido, ovvero utilizzare algoritmi di edge detection per identificare i bordi, per poi utilizzare algoritmi di machine learning per andare ad effettuare una stima sulle forme e il centro di essi, andando così ad ottenere la posizione della pupilla. Come nel caso iniziale, anche questo approccio porta ad un incremento della latenza considerevole nell'elaborazione.

Infine, è stato analizzato un metodo efficace per determinare la posizione dell'occhio rispetto al resto della zona di interesse, in quanto andare direttamente ad effettuare l'identificazione della pupilla sull'immagine non avrebbe portato ad alcun risultato. Perciò, sono state analizzate 2 principali metodologie.

- Utilizzo di algoritmi di machine learning per identificare la zona occupata. Come discusso precedentemente, per quanto si abbia un'alta affidabilità, si avrà un alta latenza nella computazione;
- Utilizzo di tecniche morfologiche per aumentare il contrasto tra aree chiare/scure. In questo modo, l'occhio risulterà in notevole contrasto con tutta la pelle circostante (che, essendo monocolora e più chiara, risulterà schiarita).

In questo modo si riesce a delineare facilmente l'occhio rispetto alla pelle circostante.

4 Implementazione

La fase di implementazione è stata preceduta da una fase di ricerca. In questa fase, si sono delineati (a grandi linee), quali sono le tecnologie principali utilizzate attualmente in tecnologie attualmente presenti nel mercato. Da quel che si è trovato, le principali tecnologie sono:

- Utilizzo di apparecchiature personalizzate con una camera montata che punta in direzione dell'occhio. La telecamera si troverà ad una piccola distanza dall'occhio, quindi si avrà una buona risoluzione dell'occhio. È possibile anche utilizzare di camere IR con lo stesso meccanismo per semplificare l'identificazione della pupilla;
- Utilizzo di algoritmi di ML per l'identificazione dell'occhio da un frame. Questo algoritmo deve avere un'elevata accuracy per portare a risultati attendibili.

Durante la fase di implementazione sono state sviluppate 3 versioni, ognuna per ottenere i migliori risultati.

Approccio con identificazione automatica dell'occhio

Nella prima versione del progetto è stata utilizzata un algoritmo di machine learning, con una rete pre-trained, per l'identificare la posizione dell'occhio ed estrarne un ROI. Questo poi sarà analizzato tramite algoritmi che sono descritti nella versione successiva del progetto.

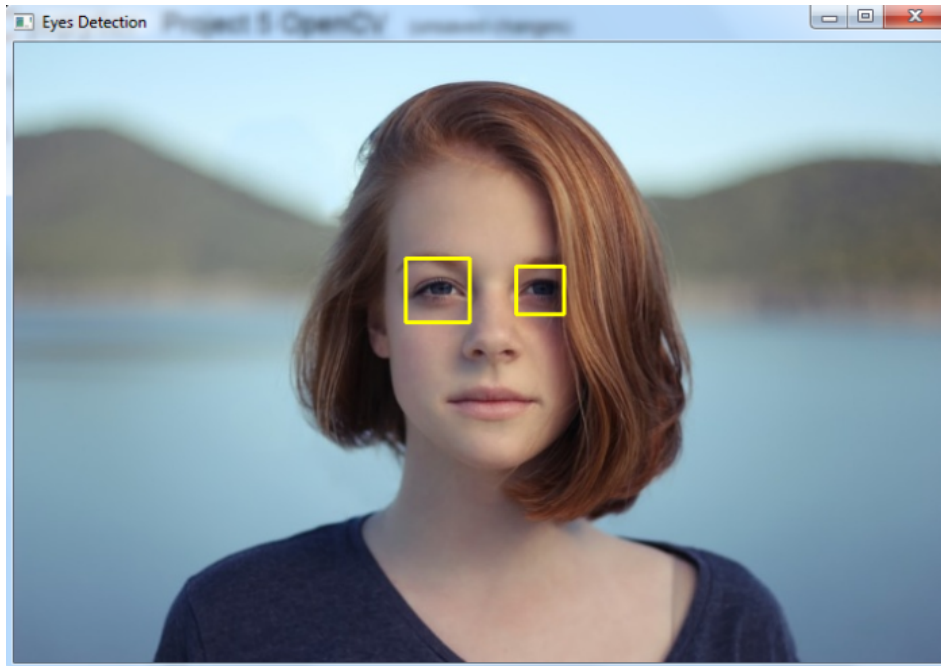


Figura 2: Esempio del risultato finale tramite l'utilizzo della rete per l'identificazione dell'occhio.

I risultati dell'algoritmo hanno portato a ottimi risultati nell'identificazione dell'occhio, fino ad un 90% delle volte, ma sono stati riscontrati grossi problemi riguardanti le performance generali. Infatti si è notato un decremento delle performance generali del 50%, portando il sistema ad un notevole frame delay. Inoltre, oltre ad aver diminuito le performance, il problema principale si è riscontrato con l'identificazione dell'occhio. Infatti, l'algoritmo ricerca sempre il match migliore per l'occhio. Questo porta ad avere l'identificazione dell'occhio in posizioni diverse del frame, facendo sì che lo stesso occhio, porti ad avere posizioni diverse nel frame, andando così a modificare la posizione della pupilla e rendendone inutile il tracciamento.

Approccio con window impostata dall'utente

Differente approccio è quello adottato successivamente, che si avvale nell'impostare una window sull'occhio direttamente da parte dell'utente. Questo approccio è sicuramente meno dinamico rispetto all'approccio che identifica automaticamente

l'occhio, ma sicuramente porta stabilità nel tracciamento della pupilla.

Il sistema si basa su diversi step.

- Il sistema prende in input i vari parametri necessari per funzionare, quali risoluzione dello schermo, black noise...
- Il sistema chiede all'utente di posizionare la window in modo che copra un'occhio dell'utente, andando poi anche ad impostare i parametri in modo da avere una corretta visualizzazione;
- Verrà chiesto di effettuare una calibrazione della posizione della pupilla in ognuno dei 4 bordi del display;
- Infine, terminati gli step successivi, la pupilla verrà tracciata e si avrà anche uno spostamento del mouse.

Principalmente, il sistema esegue questi seguenti calcoli per ottenere la pupilla e tracciarne lo spostamento.

- Il sistema permetterà all'utente di posizionare una window sull'occhio;
- Si andrà, se inserito, a cancellare il black noise dal frame ottenuto;
- Verrà ritagliata la window selezionata dall'immagine e trasformerà l'immagine in grayscale;
- Verrà fatta selezionare all'utente la threshold e la saturazione per avere la miglior separazione tra pupilla/ciglio;
- Il sistema andrà a calcolare i bordi ottenuti, eliminando successivamente quelli che toccano il bordo esterno e selezionando solamente il contorno che si trova più in basso di tutti (che coinciderà con la pupilla);
- Verrà effettuata dall'utente una calibrazione dei 4 punti dello schermo (sinistra, destra, sopra e sotto);
- Il sistema, terminata questa fase di calibrazione, andrà a spostare il mouse in base allo spostamento della pupilla.

Questo sistema si avvale dell'utilizzo di un aumento del contrasto tra bianchi e neri, nella fase di identificazione dell'occhio. Infatti, essendo la pupilla scura, andando a moltiplicare per una costante si otterrà che i bianchi diventano più bianchi e i neri più scuri. In questo modo è possibile eliminare tutta la parte di pelle in quanto risulterà sicuramente più chiara rispetto alla parte dell'occhio.

Un punto importante di questo algoritmo riguarda il sopracciglio. Essendo scuro, risulterà preso nel calcolo della pupilla. Per questo è importante fare sì che il bordo sopra della window impostata dall'utente tocchi il sopracciglio. In questo modo, l'algoritmo andrà ad eliminare quel segmento rendendo tracciata solo la pupilla.

Questo sistema risulta di elevata velocità e realtime, non andando ad aggiungere nessun delay percepibile, sicuramente però limitandone la flessibilità nell'utilizzo. Questo sarà alla fine l'approccio utilizzato nel sistema finale.

Approccio ibrido

Infine, per aumentare la flessibilità dell'approccio con window, è stato provato l'utilizzo dell'algoritmo di identificazione dell'occhio con una window impostata dall'utente. In questo caso, la window risulta essere più grande e permissiva nel movimento della testa, in quanto l'algoritmo andrà ad identificare l'occhio. In questo modo si ha una riduzione nel calcolo dell'algoritmo in quanto la superficie da analizzare è inferiore. Tuttavia, il risultato non è migliore dell'approccio 1, se non per le performance.

5 Risultati ottenuti

L'approccio finale per l'ottenimento della soluzione migliore è quello basato su una window impostata dall'utente, che risulta essere quello attualmente sviluppato per il sistema.

La soluzione finale del progetto ha portato ad ottimi risultati a livello di performance, non andando ad impattare in alcun modo sulle prestazioni del sistema e rendendo il framerate fluido.

Per quanto riguarda il tracciamento della pupilla, il sistema è in grado di tracciare perfettamente lo spostamento. Il problema principale è che non sempre viene tracciata solo la pupilla, ma anche le aree scure attorno. Considerando la luce costante, questi neri non andranno ad influire sul risultato finale, in quando vengono tracciati costantemente come se fossero parte della pupilla, rendendo il tracciamento consistente (non valido in caso di luce non costante).

Il problema principale riguarda lo spostamento del mouse. In particolare, lo spostamento della pupilla avviene in un'area di piccole dimensioni (più distanza si ha dalla camera, minore sarà lo spostamento percepito dalla camera). Per questo, lo spostamento del mouse sullo schermo dipende interamente dalla distanza camera/occhio (aumentando lo zoom si ottiene un risultato migliore). Maggiore la distanza, maggiore sarà la grandezza dello spostamento del mouse.

6 Conclusioni e sviluppi futuri

Il progetto ha portato ad una conoscenza più approfondita nell'utilizzo della libreria OpenCV, approfondendo principalmente lo studio di come ridurre il noise dato dalla camera.

In particolare, il progetto finale ha soddisfatto i requisiti posti inizialmente, lasciando comunque margine per future migliorie, quali possibile separazione della pupilla mediante forma o una miglior separazione tra la pupilla e tutti i tratti neri che circondano l'occhio (ciglia, sopracciglia, zone scure...).