**Summary**

1. Introduzione all’esperimento (Abstract)

Obiettivo: trovare le differenze di guida principali tra piloti professionisti e non, soprattutto riguardo come approcciano i movimenti della testa e degli occhi in prossimità delle curve.

1. Descrizione dell’esperimento

Utilizzo di Eye Tracking System for Motorsport

1. Descrizione dei possibili algoritmi usati per l’esperimento
2. Risultati finali

1)

Sette piloti professionisti sono stati confrontati con dieci piloti amatoriali, con l'obiettivo di ottenere il tempo sul giro più veloce**. Lo studio ha esaminato le abilità percettive e cognitive di piloti automobilistici confrontandole con piloti non professionisti attraverso l'uso di un simulatore di guida.**

Una grande quantità di ricerca è stata dedicata allo studio degli effetti degli stimoli visivi sul controllo dello sterzo. **Land e Lee hanno formulato un modello che descrive come i piloti amatoriali (cioè non automobilisti professionisti) dirigono il loro veicolo**. Questi autori hanno scoperto che i piloti concentrano il loro sguardo principalmente sul punto di tangenza (definito come il punto in cui il margine stradale interno cambia direzione) e hanno illustrato una relazione geometrica tra la posizione del punto di tangenza, la curvatura della curva e l'input necessario per lo sterzo.

Il punto di tangenza in un momento specifico coincide con il punto di massima curvatura di una curva. Il punto di massima curvatura è una posizione fissa sul margine stradale interno di una curva ed è strettamente correlato alla traiettoria di guida che il conducente segue durante la curva.

Altri modelli suggeriscono che i conducenti controllano lo sterzo mediante informazioni sul flusso ottico e che dirigono lo sguardo sul loro percorso futuro, approssimativamente da 1 a 2 secondi davanti al veicolo. Questi risultati corrispondono a altri studi che mostrano che gli esseri umani utilizzano il flusso ottico per percepire la loro direzione.

In particolare, **questa ricerca ha anche dimostrato che i conducenti esperti dirigono il loro sguardo più avanti e mostrano una maggiore varianza dello sguardo rispetto ai principianti, il che può essere spiegato dal fatto che ispezionano attivamente l'ambiente circostante. Infine, è stato riscontrato che i conducenti esperti fanno meno affidamento sulla visione foveale e più sulla visione periferica per il controllo dello sterzo.**

2)

**Per misurare e studiare questi tipi di dati si è usato un sistema di eye tracking.**

Questo sistema è stato montato su un casco da corsa per monitorare lo sguardo del pilota in tempo reale. **Un sistema di telecamere doppie è stato utilizzato per catturare sia l'immagine dell'occhio che la vista frontale. L'immagine dell'occhio è stata elaborata utilizzando OpenCV-Python per determinare le coordinate dello sguardo. Inoltre, la posizione dello sguardo è stata mappata nella vista frontale per determinare la linea di vista del pilota.**

Per quanto riguarda il sensore della telecamera, sono state scelte due telecamere OV2640 per il loro piccolo volume, che hanno una risoluzione massima di 1632 x 1232. Inoltre, ogni modulo della telecamera è dotato di un filtro infrarosso per evitare che la luce infrarossa proveniente dall'ambiente influenzi la qualità dell'immagine. Tuttavia, il filtro infrarosso è stato rimosso sulla telecamera di tracciamento degli occhi, in modo che fosse possibile utilizzare un LED infrarosso per illuminare l'occhio. In questo modo, la telecamera di tracciamento degli occhi è stata in grado di catturare un'immagine chiara dell'occhio in ambienti bui.

A computer screen shot of a helmet

Description automatically generated

3)

- Algoritmo di dilatazione ed erosione

**Una volta che l’immagine dell’occhio è stata catturata dalla telecamera viene elaborata mediante un filtro mediano per rimuovere i punti di disturbo**. A differenza dei filtri lineari, come il filtro di media, il filtro mediano non usa una combinazione ponderata di pixel nell'area di filtraggio, ma calcola il valore mediano dei pixel. Il processo di filtraggio mediano coinvolge la scansione di un'immagine con una finestra mobile (o kernel) e la sostituzione del valore di ciascun pixel con il valore mediano dei pixel all'interno della finestra. **Questo tipo di filtro è particolarmente efficace nel ridurre il rumore nelle immagini senza introdurre troppa sfocatura.**

Successivamente, le immagini degli occhi sono state trasformate in una figura HSV, che ha tre canali: H, S e V. Il valore di soglia di ciascun canale doveva essere regolato manualmente per creare una maschera per l'immagine dell'occhio**. Inoltre, è stato applicato anche un algoritmo di dilatazione ed erosione per ottimizzare la maschera.**

L'algoritmo di dilatazione ed erosione è utilizzato per la manipolazione di immagini al fine di migliorare o ridurre alcune caratteristiche di oggetti nell'immagine. Questi sono processi di base nell'ambito della morfologia matematica, una branca dell'elaborazione delle immagini che coinvolge la manipolazione di forme geometriche.

Ecco una breve spiegazione di ciascun passo:

1. **Filtro Mediano :** 
   * **Descrizione:** è un filtro di smoothing utilizzato per ridurre il rumore nelle immagini, solitamente utilizzato come pre-processing prima di effettuare altre operazioni
   * **Come funziona:** realizza un’operazione non lineare sui pixel appartenenti alla maschera: i valori vengono ordinati e poi calcolato il valore che si trova nella posizione centrale dell’ordinamento (valore mediano)
   * **Utilizzo:** il filtro mediano è usato per ridurre la presenza di rumore nelle immagini ed è usato grazie alla sua capacità di non sfocare troppo

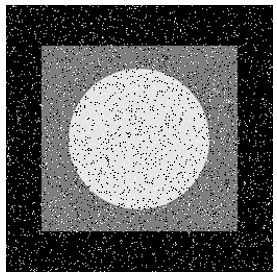
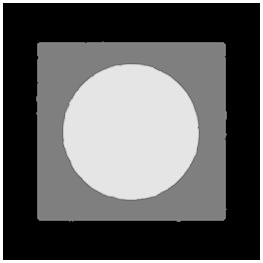
****

immagine rumorosa smoothing tramite filtro mediano

1. **Dilatazione:**
   * **Descrizione:** La dilatazione è un'operazione che allarga l'area di oggetti bianchi in un'immagine binaria.
   * **Come funziona:** Si applica un elemento strutturante (un piccolo kernel, spesso un quadrato o un cerchio) all'immagine. Per ogni posizione del kernel, se anche solo un pixel è bianco, tutti i pixel sottostanti nel risultato diventano bianchi.
   * **Utilizzo:** La dilatazione è spesso utilizzata per unire regioni vicine di pixel bianchi e per riempire buchi in oggetti.

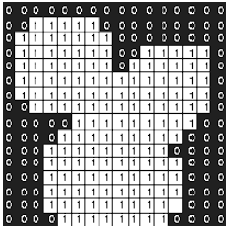
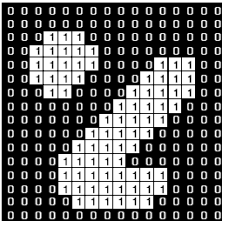


immagine di base Immagine dilatata con un filtro quadrato 3x3

1. **Erosione:**
   * **Descrizione:** L'erosione è un'operazione che restringe l'area degli oggetti bianchi in un'immagine binaria.
   * **Come funziona:** Simile alla dilatazione, si applica un elemento strutturante all'immagine. Tuttavia, in questo caso, tutti i pixel sottostanti diventano bianchi solo se tutti i pixel nel kernel sono bianchi.
   * **Utilizzo:** L'erosione è spesso utilizzata per separare regioni di oggetti vicini e per eliminare dettagli non desiderati.

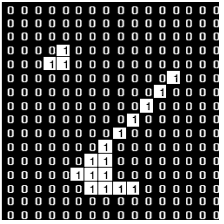
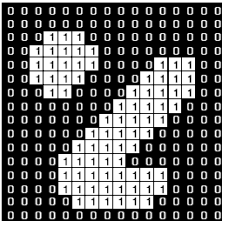


immagine di base Immagine erosa con un filtro quadrato 3x3

Combinando queste due operazioni in questo ordine ( prima dilatazione poi erosione) e usando lo stesso kernel otterremo un operazione detta di Closing che serve per chiudere eventuali buchi interni nelle aree bianche(foreground).

Al contrario se avessimo usato queste due operazioni in ordine inverso avremmo ottenuto l’operatore di Opening che ha il compito di preservare il più possibile le regioni di forma simile a quelle dell’elemento strutturante e di eliminare quelle differenti, è quindi un filtro di **smoothing**, il cui effetto è determinato dalla forma e dalle dimensioni del kernel

Per definire solo l'area della pupilla e dell'iride in nero e il resto in bianco, si può utilizzare una combinazione di dilatazione ed erosione in modo da ottenere un'immagine binaria che evidenzi solo l'area desiderata.

Codice :

import cv2

import numpy as np

# Leggi l'immagine dell'occhio

eye\_image = cv2.imread('occhio.png', cv2.IMREAD\_GRAYSCALE)

# Applica una soglia per ottenere un'immagine binaria

\_, thresholded\_eye = cv2.threshold(eye\_image, 30, 255, cv2.THRESH\_BINARY)

# Definisci il kernel per la dilatazione ed erosione

kernel = np.ones((5, 5), np.uint8)

# Dilatazione

dilated\_eye = cv2.dilate(thresholded\_eye, kernel, iterations=1)

# Erosione

eroded\_eye = cv2.erode(dilated\_eye, kernel, iterations=1)

# Ora hai un'immagine in cui solo l'area della pupilla e dell'iride è bianca e il resto è nero

**Spiegazione del codice:**

1. **Lettura dell’immagine dell’occhio**: *eye\_image = cv2.imread('occhio.png', cv2.IMREAD\_GRAYSCALE)*.

L'immagine dell'occhio viene letta dal file 'occhio.png' e convertita in scala di grigi utilizzando cv2.IMREAD\_GRAYSCALE.

1. **Soglia binaria :** *\_, thresholded\_eye = cv2.threshold(eye\_image, 30, 255, cv2.THRESH\_BINARY).*

Viene applicata una soglia binaria all'immagine dell'occhio. I pixel con intensità superiore a 30 vengono impostati a 255 (bianco), mentre quelli inferiori o uguali a 30 vengono impostati a 0 (nero). L'immagine risultante è binaria.

1. **Dilatazione:** *dilated\_eye = cv2.dilate(thresholded\_eye, kernel, iterations=1).*

Viene eseguita un'operazione di dilatazione sull'immagine binaria precedentemente ottenuta. La dilatazione espande le regioni bianche nell'immagine. Il kernel utilizzato è una matrice 5x5 di elementi tutti uguali a 1.

1. **Erosione :** *eroded\_eye = cv2.erode(dilated\_eye, kernel, iterations=1).*

Successivamente, viene eseguita un'operazione di erosione sull'immagine dilatata. L'erosione ha l'effetto di restringere le regioni bianche nell'immagine. Anche in questo caso, il kernel è una matrice 5x5 di elementi tutti uguali a 1.

Nel codice sopra, **thresholded\_eye** è l'immagine binaria risultante dalla soglia. La dilatazione è applicata per espandere l'area bianca, e poi l'erosione è applicata per ridurre eventuali zone bianche aggiuntive. Il risultato (**eroded\_eye**) sarà un'immagine binaria in cui solo l'area della pupilla e dell'iride è bianca e tutto il resto è nero. Puoi regolare i parametri della soglia e del kernel in base alle tue esigenze specifiche.

A screenshot of a computer

Description automatically generated

**Dopo questi processi, solo l'area dell'iride e della pupilla è stata contrassegnata in nero e tutte le altre aree sono diventate bianche. Le coordinate del punto di sguardo potevano quindi essere ottenute calcolando i contorni della maschera**.

**Il passo successivo è la calibrazione,** in cui quattro punti delle coordinate dello sguardo devono essere mappati con quattro coordinate relative nella vista frontale**. Con quattro coordinate su ciascuna immagine, è stata calcolata la matrice di trasformazione di proiezione, e quindi le altre coordinate dell'immagine dell'occhio potevano essere abbinate alla vista frontale. Dopo tutti questi passaggi, il casco di tracciamento degli occhi è in grado di identificare in tempo reale dove una persona sta guardando.**

4)

In conclusione, **l’analisi dei movimenti oculari ha evidenziato una strategia di sguardo più variabile per i piloti automobilistici rispetto ai non piloti. I conducenti non professionisti hanno seguito una strategia di tracciamento del punto di tangenza per tutta la curva, mentre i piloti professionisti hanno spostato lo sguardo in relazione al punto di tangenza**. **Questi risultati si integrano con quelli di uno studio condotto su una pista automobilistica reale, il quale ha mostrato che un pilota automobilistico dirige il suo sguardo nelle vicinanze del punto di tangenza anziché direttamente su di esso.** Contrariamente ai modelli di controllo visivo dello sterzo, nei quali sono utilizzati punti di riferimento (ad esempio, il punto di tangenza) per guidare l'input dello sterzo, il nostro studio mostra che i piloti automobilistici modificano il loro sguardo durante le curve, probabilmente per verificare il percorso e anticipare future azioni di controllo. Inoltre, i conducenti non professionisti potrebbero semplicemente guardare dove vogliono andare, mentre i piloti automobilistici potrebbero indirizzare il loro sguardo a informazioni rilevanti per la guida.