



دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و
کامپیوتر



طراحی و ساخت سیستم تعقیب کننده ی نگاه

پایان نامه برای دریافت درجه کارشناسی
در رشته مهندسی برق گرایش کنترل

محمدحسین کاشانی نیا

شماره دانشجویی

۸۱۰۱۹۷۵۶۰

استاد راهنما:

سرکار خانم دکتر فریبا بهرامی

شهریورماه ۱۴۰۱

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

تعهدنامه اصالت اثر

باسمه تعالی

اینجانب محمدحسین کاشانی نیا تأیید می کنم که مطالب مندرج در این پایان نامه حاصل تلاش اینجانب است و به دستاوردهای پژوهشی دیگران که در این نوشته از آنها استفاده شده است مطابق مقررات ارجاع گردیده است. این پایان نامه قبلاً برای احراز هیچ مدرک هم سطح یا بالاتر ارائه نشده است.

کلیه حقوق مادی و معنوی این اثر متعلق به دانشکده فنی دانشگاه تهران می باشد.

نام و نام خانوادگی دانشجو : محمدحسین کاشانی نیا

امضای دانشجو :



چکیده^۱

تعقیب چشم^۲ به معنای تشخیص حرکات و موقعیت چشم و به دست آوردن نقطه ی نگاه شخص است. در مقابل تعقیب نگاه^۳ به بررسی همزمان حرکات و موقعیت چشم و سر می‌پردازد و نقطه ی نگاه شخص را به دست می‌آورد. روش های متعدد و گوناگونی برای مشاهده و به دست آوردن حرکات چشم در زمینه های تحقیقاتی و تجاری مورد استفاده قرار گرفته است. این پروژه به پیاده سازی یک سیستم تعقیب نگاه با استفاده از دوربین می‌پردازد. سیستم ساخته شده برای تعقیب نگاه بیماران همزمان با تست و ثبت حرکات دست توسط روبات موجود در آزمایشگاه کنترل حرکت و علوم اعصاب محاسباتی به کار گرفته خواهد شد. برای تخمین نقطه ی نگاه، ابتدا با استفاده از کتابخانه ی mediapipe و زبان برنامه نویسی پایتون صورت و چشمهای شخص تشخیص داده شده، سپس با به دست آوردن مکان مردمک ها و همچنین مکان و جهت سر، و با محاسبات هندسی نقطه ی نگاه به دست می‌آید. توانایی محاسبه ی نقطه ی نگاه به صورت بی درنگ^۴ با دقت متوسط حدود ۳ سانتی متر با فاصله ی حدود ۴۰ سانتی متری از صفحه و کیفیت تصویر 640 × 480 از ویژگی های این سیستم است که با بالا رفتن کیفیت تصویر و دستگاه اجرا کننده ی برنامه بهبود بهبود خواهد یافت.

کلمات کلیدی: تعقیب چشم، تعقیب نگاه، mediapipe، پردازش تصویر، وبکم، زبان پایتون، نشانه گذاری

صورت

¹ Abstract

² Eye Tracking

³ Gaze Tracking

⁴ Real time

فهرست مطالب

فصل ۱: مقدمه و بیان مساله	۱
فصل ۲: مرور کارهای انجام شده در این زمینه	۳
۲-۱- مقدمه	۳
۲-۲- روش الکترواکولوگرافی	۴
۲-۳- روش سیم پیچ جستجوی صلبیه ای	۵
۲-۴- اکولوگرافی مادون قرمز	۷
۲-۵- روش های مبتنی بر پردازش تصویر	۸
۲-۵-۱- مکان یابی صورت	۸
۲-۵-۲- مکان یابی چشم	۹
۲-۵-۳- مکان یابی مردمک	۹
۲-۶- کاربردهای تعقیب چشم و تعقیب نگاه	۹
۲-۷- جمع بندی	۱۲
فصل ۳: پیاده سازی	۱۳
۳-۱- مقدمه	۱۳
۳-۲- انتخاب محیط توسعه	۱۳
۳-۳- روش haar آبشاری	۱۴
۳-۴- کتابخانه dlib	۱۶
۳-۵- کتابخانه mediapipe	۱۷
۳-۶- تشخیص مکان عنبریه و زاویه ی چشم	۱۹
۳-۶-۱- میزان باز بودن چشم و پلک زدن	۱۹

۲۰	۳-۶-۲- مکان و زاویه ی افقی.....
۳۱	۳-۶-۳- مکان و زاویه ی عمودی.....
۳۲	۳-۷- تشخیص موقعیت و جهت گیری سر.....
۳۵	۳-۸- به دست آوردن مکان نگاه با ترکیب اطلاعات چشم ها و سر.....
۳۶	۳-۹- کالیبراسیون.....
۳۶	۳-۱۰- ارزیابی.....
۳۷	۳-۱۱- ساختار کلی اجرای برنامه ی نوشته شده.....
۳۸	فصل ۴: جمع بندی، نتیجه گیری و پیشنهاد ها.....
۳۹	مراجع.....
۴۱	پیوست الف: آدرس مخازن و فایل ها.....

فهرست شکل‌ها

- شکل (1-1) : روبات موجود در آزمایشگاه کنترل حرکت و علوم اعصاب محاسباتی..... ۱
- شکل (2-1) : قرارگیری الکترودهای پوستی برای ثبت EOG [4, 1]..... ۵
- شکل (2-2) : قرارگیری سیم پیچ روی چشم [1]..... ۵
- شکل (2-3) : وضعیت سیم پیچ و چشم در میدان‌های مغناطیسی برای تشخیص حرکات چشم [1]..... ۶
- شکل (2-4) : لنز دارای سیم پیچ برای اندازه‌گیری دو بعدی، لنز دارای سیم پیچ برای اندازه‌گیری سه بعدی [1]..... ۶
- شکل (2-5) : اندازه‌گیری حرکت افقی چشم با استفاده از روش IROG [1]..... ۷
- شکل (2-6) : مراحل تعقیب چشم یا نگاه به روش VOG [1]..... ۸
- شکل (2-7) : تعداد مقالات در سال برای بازه ی ۱۹۶۸ تا ۲۰۱۸. جستجو شده با عبارت "eye TOPIC: tracking" OR "eye-tracking" OR "eyetracking" در شبکه ی دانش [2]..... ۱۰
- شکل (2-8) : درصد موضوعات مختلف بین مقالات پیدا شده در شکل (۷-۲) [2]..... ۱۱
- شکل (3-1) : چند نمونه از ویژگی‌های haar. جمع پیکسل‌های داخل مستطیل‌های سیاه از جمع پیکسل‌های داخل مستطیل‌های سفید کم میشود [10, 11]..... ۱۵
- شکل (3-2) : روند اجرای مراحل برای دسته بندی کننده ی آبشاری [12, 13]..... ۱۵
- شکل (3-3) : نقاط تشخیص داده شده روی صورت توسط کتابخانه ی dlib..... ۱۷
- شکل (3-4) : روند تشخیص صورت و تخمین نقاط روی صورت توسط Mediapipe [19]..... ۱۸
- شکل (3-5) : نقاط تشخیص داده شده روی صورت توسط کتابخانه ی mediapipe..... ۱۸
- شکل (3-6) : محاسبه ی نسبت چشم برای تشخیص پلک زدن..... ۲۰
- شکل (3-7) : خط بین نقاط دور چشم در کتابخانه های mediapipe و dlib..... ۲۱
- شکل (3-8) : تبدیل به تصویر خاکستری و اعمال ماسک (راست)، اعمال حد آستانه و رسم خط محاسبه ی میانگین وزن دار (خاکستری) و نقطه ی به دست آمده (سفید مشخص شده با دایره ی قرمز) (چپ)..... ۲۲
- شکل (3-9) : جابجایی نقطه ی سفید با جابجایی چشم..... ۲۲
- شکل (3-10) : نمودار مکان نسبی مرکز عنبیه به دست آمده به روش آستانه گذاری (نگاه به وسط، راست، چپ و سپس وسط)..... ۲۲
- شکل (3-11) : نمودار نقاط شکل (۳-۱۰) با اعمال فیلتر میانگین ۱۰ نقطه ای (نگاه به وسط، راست، چپ و سپس وسط)..... ۲۳
- شکل (3-12) : با نگاه به بالا، خط واصل دو گوشه ی چشم عنبیه را قطع نمیکند..... ۲۴
- شکل (3-13) : خط بسته ی حاصل از نقاط دور چشم (قرمز)، کوچکترین دایره شامل چهار نقطه اطراف عنبیه (سبز) و مرکز عنبیه و مردمک (بنفش)..... ۲۴
- شکل (3-14) : نقاط سه بعدی صورت که توسط mediapipe تخمین زده شده اند..... ۲۵
- شکل (3-15) : خط واصل گوشه ی چشم و مرکز عنبیه (بنفش)، خط واصل دو گوشه ی چشم (خاکستری) و تصویر نقطه ی مرکز عنبیه روی خط واصل بین دو چشم (قرمز)..... ۲۵
- شکل (3-16) : خط واصل گوشه های دو چشم (سبز)، تصویر مرکز عنبیه ها روی امتداد خط واصل گوشه های دو چشم (بنفش)، فاصله ی افقی مرکز عنبیه ها (قرمز) و فاصله ی عمودی مرکز عنبیه ها (آبی)..... ۲۶

- شکل (3-17) : مکان افقی محاسبه شده برای هر کدام از چشم ها، ابتدا به مرکز صفحه نگاه شده و سپس به ترتیب سمت راست و چپ صفحه. و در نهایت دوباره به مرکز صفحه نگاه شده ۲۶
- شکل (3-18) : زمان نمونه های سیگنال ذخیره شده ۲۷
- شکل (3-19) : فاصله ی زمانی بین نمونه ها ۲۸
- شکل (3-20) : سیگنال در حوزه ی زمان و طیف آن در حوزه ی فرکانس ۲۸
- شکل (3-21) : نتیجه ی اعمال فیلتر IIR و FIR طراحی شده روی سیگنال ۲۹
- شکل (3-22) : نتیجه ی اعمال میانگین ۱۰ نقطه ای و فیلتر IIR روی سیگنال ۳۰
- شکل (3-23) : نتیجه ی اعمال میانگین ۱۰ نقطه ای و ترکیب آن با فیلتر IIR ۳۰
- شکل (3-24) : مکان عمودی محاسبه شده برای هر کدام از چشم ها، ابتدا به مرکز صفحه نگاه شده و سپس به ترتیب بالا و پایین صفحه. و در نهایت دوباره به مرکز صفحه نگاه شده ۳۲
- شکل (3-25) : نقاط انتخاب شده برای حل مسئله ی PnP [8] ۳۳
- شکل (3-26) : زوایای محاسبه شده برای سر [8] ۳۳
- شکل (3-27) : دستگاه اندازه گیری دقت این روش با استفاده از آردوینو و Vive [8] ۳۳
- شکل (3-28) : نتیجه ی کد و جهت پیدا شده برای سر (مشخص شده با پیکان قرمز روی بینی شخص) [8] ... ۳۴
- شکل (3-29) : با جابجا شدن مکان سر (ضربدر سفید) زاویه نسبت به دوربین تغییر کرده (فاصله ی ضربدر سفید تا ضربدر بنفش کوچک) اما جبران زاویه، آن را به جای درستش برگردانده (ضربدر بنفش بزرگ که روی ضربدر سفید بودن آن زاویه ی صفر را نشان میدهد) ۳۴
- شکل (3-30) : با زاویه گرفتن سر (دور شدن ضربدر بنفش از ضربدر سفید)، مکان سر محاسبه شده قدری تغییر کرده (ضربدر سفید کوچک جابجا شده) اما جبران مکان، آن را سر جایش نگه داشته است (ضربدر سفید بزرگ حرکت نکرده است) ۳۴
- شکل (3-31) : ماشین حالت برنامه ی نوشته شده ۳۷

فهرست جدول‌ها

- جدول (۱-۲) : اندازه گیری های پر کاربرد از تعقیب نگاه در تحقیقات [6] ۱۱
- جدول (۱-۳) : شماره نقاط انتخاب شده برای اندازه گیری نسبت چشم ها برای هر کدام از کتابخانه ها ۲۰
- جدول (۲-۳) : نقاط دور چشم برای هر کدام از کتابخانه ها ۲۱

فهرست علائم اختصاری

EOG	Electrooculography
VOG	Video-Oculography
EEG	Electroencephalogram
AC	Alternating current
IROG	Infrared Occulography
NIR	Near Infrared
UDP	User Datagram Protocol
GPU	Graphical processing unit
IIR	Infinite Impulse response
FIR	Finite Impulse response
PnP	Perspective n points

فصل ۱: مقدمه و بیان مساله

در آزمایشگاه کنترل حرکت و علوم اعصاب محاسباتی (در دانشکده مهندسی برق و کامپیوتر) بازوی روبات ۵ لینکه موازی عمودی برای آنالیز حرکت دست وجود دارد (شکل (۱-۱)). مسئله ی هماهنگی حرکت چشم (نگاه) و دست برای کارهای تحلیل حرکت و اصول توانبخشی مهم است. بنابراین در این پروژه هدف طراحی یک سیستم دنبال کننده ی نگاه^۱ است که روی سیستم روباتیکی مذکور متصل شود و داده های نگاه را به صورت هم زمان^۲ با داده های حرکت دست - که بازوی ربات را حرکت می دهد - ثبت کند. همچنین اساتید دانشکده توانبخشی علوم پزشکی ایران اعلام کرده اند که نیاز به ثبت داده های نگاه در کنار حرکات دست برای درمان و آنالیز حرکات افراد بیمار پس از سکته ی مغزی و کلا بیمارانی با مشکلات حرکتی با منشاء زوال عصبی^۳ که دچار اختلال در سیستم عصبی حرکتی هستند، دارند. مجموعه ربات ۵ لینکی مجهز به سیستم تعقیب نگاه قرار است در این مرکز مورد استفاده و تحلیل از دید بالینی قرار بگیرد.



شکل (۱-۱): روبات موجود در آزمایشگاه کنترل حرکت و علوم اعصاب محاسباتی

¹ Gaze Tracker

² Synchronous

³ neurodegenerative

از بین روش های موجود، روش تعقیب نگاه با استفاده از پردازش تصویر با دوربین ثابت (غیر متصل به سر) انتخاب شد تا شخص آزادی عمل داشته باشد و تاثیری روی آزمایش گذاشته نشود. برای این کار با استفاده از کتابخانه ی mediapipe نقاطی روی صورت تخمین زده می شوند و با استفاده از نقاط دور چشم و عنبیه، وضعیت نسبی چشم ها نسبت به سر تعیین می شود. همچنین با داشتن مختصات سه بعدی نقاط جهت و مکان سر تعیین می شود و با ترکیب این اطلاعات با وضعیت چشم ها، نقطه ی نگاه محاسبه می شود. در این پایان نامه، ابتدا در فصل ۲: به بررسی تاریخچه و کارهای انجام شده در زمینه ی تعقیب چشم و تعقیب نگاه، و روش های مختلف انجام آن می پردازیم؛ سپس در فصل ۳: روند و نحوه ی پیاده سازی آن بیان خواهد شد و در نهایت، در فصل ۴: جمع بندی کار انجام شده و پیشنهادهایی برای بهبود این سیستم و استفاده از آن ارائه می شود.

کدهای پروژه در مخزن^۱ github پروژه که لینک آن داخل بخش پیوست وجود دارد موجود هستند.

¹ Repository

فصل ۲: مرور کارهای انجام شده در این زمینه

۱-۲- مقدمه

تاریخچه ی تعقیب چشم به قرن ۱۸ بر میگردد. در ۱۷۹۲ ولز^۱ از پس دید^۲ (تصویر باقی مانده روی شبکه پس از نگاه به نور) برای توصیف حرکات چشم استفاده کرد. در ۱۸۷۹ جاوال^۳ و در ۱۸۹۱ لامار^۴ با جفت کردن مکانیکی چشم و گوش با نوار لاستیکی به حرکات قابل شنیدن چشم دست یافتند [1] و در ۱۸۹۸ دلابار^۵ با اتصال مکانیکی خودکار به قالب گچی حلقه ای قرار داده شده روی قرنیه، موفق به ثبت حرکات چشم شد [2]. داج^۶ و کلاین^۷ در ۱۹۰۱ از یک روش عکس برداری و بازتاب نور برای اندازه گیری حرکات افقی چشم استفاده کردند [1] و در ۱۹۳۵، باسول^۸ روشی برای دریافت انعکاس پرتوی نور از قرنیه روی فیلم به وسیله ی منشورها ابداع کرد [2]. در ۱۹۳۹ جانگ^۹ با استفاده از الکترواکولوگرافی^{۱۰} حرکات عمودی و افقی چشم را به صورت همزمان ثبت کرد. این روش برای اولین بار امکان اندازه گیری بی درنگ^{۱۱} حرکات چشم به وسیله ی کامپیوترهای آنالوگ را نشان داد [1]. یربس^{۱۲} در ۱۹۶۷ لنزی قرنیه ای ساخت که به وسیله ی مکش متصل می شد، و از آن در تحقیقاتش استفاده کرد [2].

در دهه ی ۱۹۸۰، کامپیوترها به قدری توانمند شدند که امکان اندازه گیری بی درنگ حرکات چشم را با استفاده از روش های مبتنی بر ویدیو^{۱۳} برای ارتباط انسان و کامپیوتر ایجاد کردند. از دهه ۱۹۹۰ تا به حال رشد ثابتی در استفاده از تعقیب کننده های نگاه وجود داشته و ارزان تر شدنشان، باعث گسترش استفاده از

¹ Wells

² Afer image (ghost image)

³ Javal

⁴ Lamare

⁵ Delabarre

⁶ Dodge

⁷ Cline

⁸ Buswell

⁹ Jung

¹⁰ Electrooculography (EOG)

¹¹ Real time

¹² Yarbus

¹³ Video-Occulography (VOG)

آنها و دسترسی عمومی تر به امکاناتشان شده است [1].

روش های مختلفی برای تعقیب چشم وجود دارد که می تواند از آنها با ثابت نگه داشتن سر مکان نگاه را محاسبه کرد. همچنین در بعضی روش ها می توان با در نظر گرفتن مکان و زاویه ی سر کار تعقیب نگاه را انجام داد. در این فصل به بررسی این روش ها می پردازیم.

۲-۲- روش الکترواکولوگرافی

چشم به عنوان یک دو قطبی الکتریکی عمل می کند، به این صورت که بخش قدامی آن دارای پتانسیل مثبت و بخش خلفی دارای پتانسیل منفی است. الکترواکولوگرافی یا EOG، روشی است برای اندازه گیری این پتانسیل. در این روش الکترودها راست و چپ یا بالا و پایین چشم ها قرار می گیرند، اگر مرکز چشم به سمت یکی از الکترودها حرکت کند، آن الکتروود پتانسیل مثبت و الکتروود مقابل پتانسیل منفی می بیند. با در نظر گرفتن پتانسیل استراحت به عنوان صفر، اختلاف پتانسیل الکتریکی بین دو الکتروود را می توان به عنوان حرکت چشم در راستای الکترودها در نظر گرفت. از این روش در چشم پزشکی و سینما (برای ضبط حرکات چشم برای جلوه های ویژه) استفاده می شود [3]. سیگنال EOG ناشی از عوامل مختلفی مثل حرکت کره ی چشم، حرکت پلک، منابع دیگر مثل EEG^۱، مکان الکترودها، حرکت سر و ... می باشد؛ بنابراین لازم است مقدار پتانسیل استراحت تعیین شود. همچنین برای حل این مشکل از یک تقویت کننده ی تفاضلی AC با بهره ی بالا استفاده می شود [1].

پتانسیل ایستای چشم در تاریکی کم شده و بعد از ۸ تا ۱۲ دقیقه به حداقل خود می رسد، همچنین در روشنایی زیاد می شود و در ۱۰ دقیقه به حداکثر خود می رسد. این روش توسط الوین مارگ^۲ در ۱۹۵۱ بیان و نامگذاری شد و جفری آردن^۳ اولین کاربرد بالینی آن را ایجاد کرد. دامنه ی تغییر ولتاژ در این روش حدود ۵۰ تا ۳۵۰۰ میکرو ولت است (به ازای هر درجه جابجایی چشم حدود ۱۴ تا ۲۰ میکرو ولت تغییر ولتاژ وجود دارد). در این روش الکتروود دیگری به عنوان مرجع پتانسیل می تواند در مرکز پیشانی قرار داده شود [1, 4].

¹ Electroencephalogram

² Elwin Marg

³ Geoffrey Arden



شکل (۱-۲) : قرارگیری الکترودهای پوستی برای ثبت EOG [1, 4]

همانطور که بیان شد، می توان از اختلاف پتانسیل به دست آمده از این روش حرکت چشم را تقریب زد، اما این فقط یک تخمین از وضعیت چشم است زیرا بافت های اطراف چشم دارای ساختاری نامنظم هستند. همچنین بین EOG و حرکت چشم تنها تا زاویه ی 30° به صورت عمودی و 50° به صورت افقی رابطه ی خطی برقرار است [1, 5]. یکی از تعقیب کننده های چشم تجاری BIOPAC MP 45/100 می باشد [1].

۲-۳- روش سیم پیچ جستجوی صلبیه ای^۱

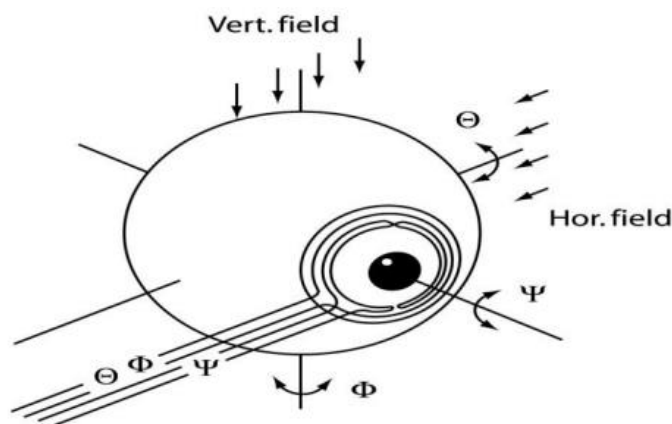
در این روش یک سیم پیچ نازک که داخل یک لنز تعبیه شده، پس از بی حسی چشم روی آن قرار می گیرد. دو سیم پیچ در دو طرف سر میدان های مغناطیسی ایجاد می کنند. این میدان مغناطیسی باعث می شود با حرکت چشم و در نتیجه حرکت سیم پیچ قرار گرفته روی چشم، در آن ولتاژ القا شود که با اندازه گیری این ولتاژ و انجام محاسبات لازم، می توان حرکات دقیق چشم را شناسایی کرد [1].



شکل (۲-۲) : قرارگیری سیم پیچ روی چشم [1]

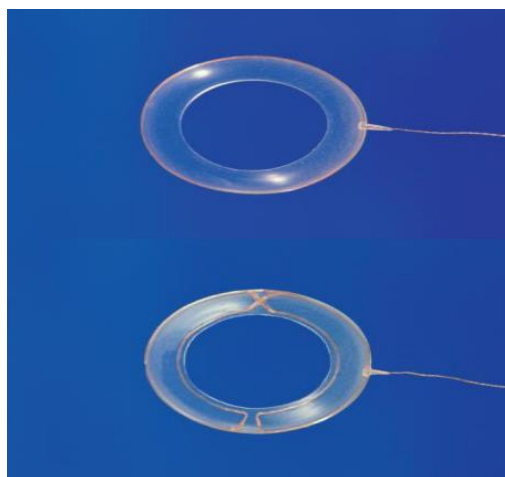
¹ Scleral Search Coil Method

با اضافه کردن یک مجموعه سیم پیچ دیگر اطراف سر و ایجاد میدان مغناطیسی عمودی، حرکات عمودی چشم هم باعث ایجاد سیگنال در سیم پیچ روی چشم می شوند، به وسیله ی ابزار الکترونیکی مناسب می توان سیگنال مربوط به حرکات عمودی را از سیگنال مربوط به حرکات افقی جدا کرد.



شکل (۳-۲) : وضعیت سیم پیچ و چشم در میدان های مغناطیسی برای تشخیص حرکات چشم [1]

مزیت بزرگ این روش دقت و سرعت بالای آن است. به همین دلیل در تحقیقات پزشکی و روانشناسی از آن استفاده می شود. دقت این روش به قدری زیاد است که می تواند حرکات بسیار ریز چشم مثل میکروساکاد^۱ را نیز شناسایی کند. اما این روش عیب بزرگی نیز دارد و آن ایجاد ناراحتی با قرار گیری سیم پیچ روی چشم است، به همین دلیل در کاربرد های بالینی به ندرت از این روش استفاده می شود. با اضافه کردن یک حلقه ی دیگر روی چشم حتی می توان حرکت چشم حول محور نگاهش^۲ (که در شکل (۳-۲) با Ψ نمایش داده شده) را نیز آشکار کرد [1].



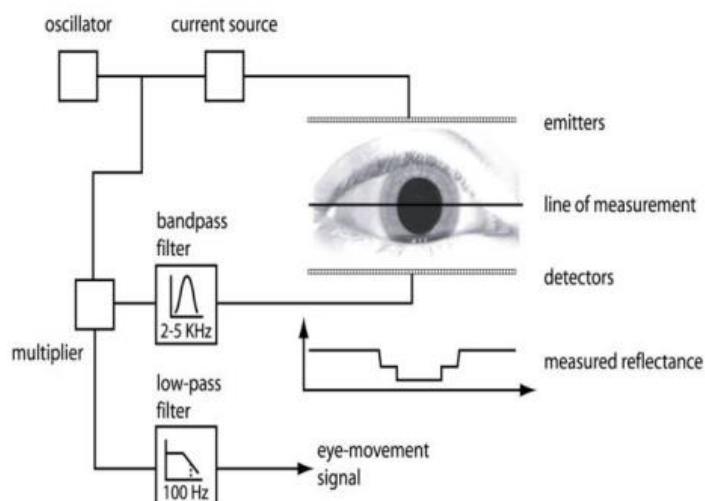
شکل (۴-۲) : لنز دارای سیم پیچ برای اندازه گیری دو بعدی، لنز دارای سیم پیچ برای اندازه گیری سه بعدی [1]

¹ Microsaccades

² Torsional Movements

۴-۲- اکولوگرافی مادون قرمز^۱

با تاباندن یک منبع نور ثابت، مقدار نور بازتاب شده از چشم به یک آشکارساز به موقعیت چشم بستگی دارد. در این روش تعدادی منبع نور مادون قرمز به چشم تابانده می شوند و گیرنده ها در جای مناسب قرار میگیرند تا بازتاب را دریافت کنند. این سیستم برای تعقیب مرز بین صلبیه و عنبیه^۲ یا مرز بین عنبیه و مردمک ساخته شده است. استفاده از نور مادون قرمز به دلیل نامرئی بودن این نور برای انسان، در دید او اثری نمی گذارد و همچنین چون گیرنده های مادون قرمز به نور مرئی حساس نیستند، نور زمینه ای که در محیط وجود دارد در اندازه گیری بی اثر است [1].



شکل (۵-۲): اندازه گیری حرکت افقی چشم با استفاده از روش IROG [1]

دقت این روش در اندازه گیری حرکت افقی چشم از مرتبه ی 0.1° است و حدود بازه ی $\pm 15^\circ$ تا $\pm 40^\circ$ بسته به سیستم طراحی شده قابل اندازه گیری است. اندازه گیری حرکت عمودی چشم به دلیل مزاحمت پلک ها و پنهان شدن مرزهای عنبیه زیر آن ها بسیار سخت تر است [1].

¹ Infrared Oculography (IROG)

² Limbus (The boundary between sclera and iris)

۵-۲- روش های مبتنی بر پردازش تصویر

در این روش ها که به VOG^1 معروف هستند، تخمین نگاه از روی تصویر به دست آمده از یک دوربین انجام می شود. این تخمین می تواند فقط زاویه ی چشم را نسبت به سر به دست بیاورد (برای سیستم های نصب شده روی سر) یا زاویه ی چشم ها را نسبت به محیط محاسبه کند. روش اول دقت بیشتری دارد و در مطالعات حرکت چشم استفاده می شود. در حالی که در روش دوم می توان نقطه ی نگاه روی یک صفحه را به دست آورد. مراحل تعقیب چشم یا نگاه به وسیله ی پردازش تصویر در شکل (۲-۶) نمایش داده شده اند [1].



شکل (۲-۶) مراحل تعقیب چشم یا نگاه به روش VOG [1]

برای دریافت تصویر می توان از انواع دوربین ها (دوربین/وبکم/دوربین مادون قرمز/دوربین NIR^2 ...) استفاده کرد. همچنین بسته به تکنولوژی مورد استفاده می توان از چند دوربین استفاده شود [1]. از تعقیب کننده های چشم به این روش می توان به ERICA و Tobii 1750 اشاره کرد [1].

۱-۵-۲- مکان یابی صورت

مکان یابی صورت به معنای تشخیص صورت در یک تصویر و به دست آوردن مکان آن در تصویر می باشد. روش های مختلفی برای این کار وجود دارد، مثل روش بر مبنای رنگ پوست، روش مبتنی بر پایگاه داده^۳، روش مبتنی بر ویژگی های صورت، روش مبتنی بر بافت، روش مبتنی بر تطبیق الگو، روش مبتنی بر ظاهر^۴، روش مبتنی بر ویژگی های شبه Haar^۵، و ... [1].

¹ Video oculography

² Near Infrared

³ Knowledge based

⁴ Appearance based

⁵ Haar like features

۲-۵-۲- مکان یابی چشم

پس از مکان یابی صورت مکان یابی چشم انجام می شود. به طور کلی دو روش برای مکان یابی چشم ها می توان در نظر گرفت؛ روش کل نگرانه^۱ که با استفاده از نمایش های سراسری به پیدا کردن چشم ها می پردازد، و روش انتزاعی^۲ که با استخراج و محاسبه ی ویژگی های مجزا و تشخیص الگو چشم ها را مکان یابی می کند. روش های مکان یابی چشم به طور کلی اعمال فیلتر کالمن^۳، مدل تصویرسازی خاکستری، تطبیق الگو، ویژگی های هندسی چشمه ها، تابع تصویرسازی^۴، تشخیص گوشه ها یا ترکیبی از این روش ها هستند [1].

۲-۵-۳- مکان یابی مردمک

روش های مکان یابی مردمک به این ترتیب هستند: الگوریتم تابع توزیع تجمعی^۵، الگوریتم تابع تصویرسازی^۶، تحلیل لبه ها، تصویرسازی صحیح^۷ و مدل گوسی، تطبیق الگو/ویژگی های شکل عنبیه، تشخیص گوشه ی هریس^۸ و تخمین انحنای ایزوفوت^۹ ها. همچنین برای تعقیب چشم مبتنی بر تصویر از روش های تعقیب مرز عنبیه و صلبیه، روش پورکنژ^{۱۰} و هاپلوسکوپ فاندوس^{۱۱} نیز استفاده می شود [1]. همچنین در بعضی کاربردها با تاباندن نور مادون قرمز به چشم و اندازه گیری مکان بازتاب نور از قرنیه نسبت به مردمک، کار تعقیب چشم را انجام می دهند [2].

۲-۶- کاربردهای تعقیب چشم و تعقیب نگاه

اولین استفاده از تعقیب کننده های نگاه در سال ۱۹۴۷ برای پیدا کردن بهترین مکان برای کنترل های

¹ holistic

² abstractive

³ Kallman

⁴ Projection function

⁵ Cumulative distribution function (CDF) algorithm

⁶ Projection function (PF)

⁷ Integral projection

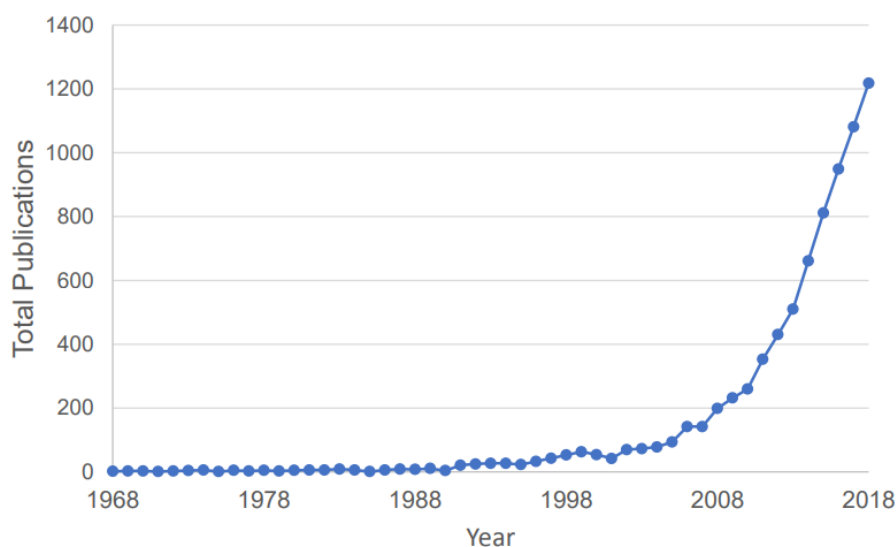
⁸ Harris corner detector

⁹ Isophote

¹⁰ Purkinje

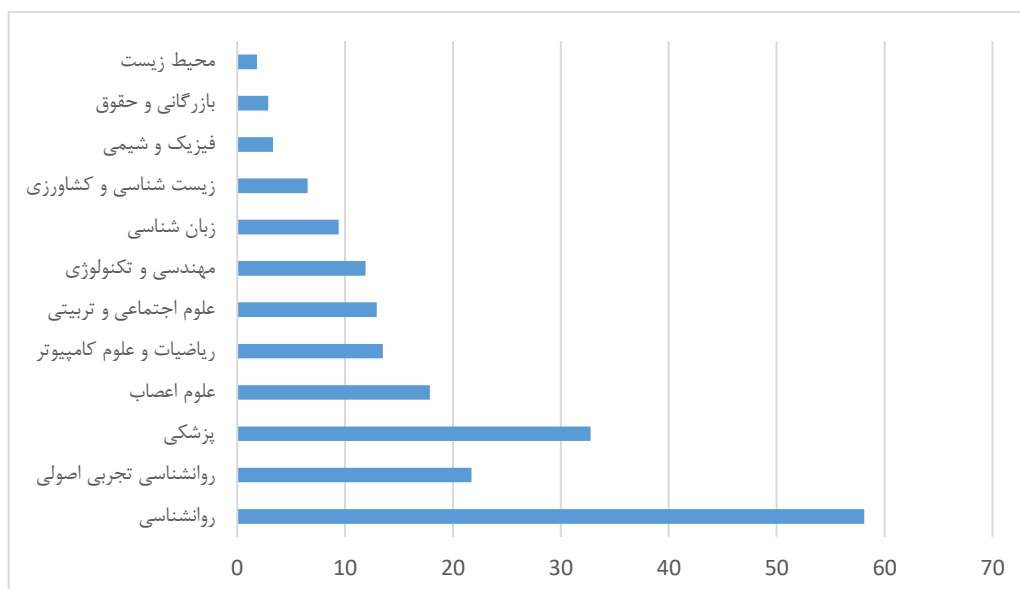
¹¹ Fundus Haploscope

اتاقک هواپیما توسط نیروی هوایی آمریکا انجام شد [1]. با پیشرفت این تکنولوژی، دسترسی به آن افزایش و هزینه ی آن کاهش یافته و همانطور که در شکل (۷-۲) و شکل (۸-۲) قابل مشاهده است، استفاده از آن در تحقیقات افزایش چشمگیری پیدا کرده [2]، به طوری که امروزه شاهد استفاده ی وسیع آن در کاربردهای مختلف هستیم؛ از جمله رابط انسان و کامپیوتر، بازی های ویدیویی، توان بخشی، شبیه سازی رانندگی، تحقیقات بازاریابی و تبلیغات، تشخیص خستگی، علوم شناختی، تحقیقات قابل استفاده بودن، تحقیقات پزشکی، برهمکنش نگاه، دستیار رانندگی، تایپ با چشم و ... [1].



شکل (۷-۲) تعداد مقالات در سال برای بازه ی ۱۹۶۸ تا ۲۰۱۸. جستجو شده با عبارت
 TOPIC: ("eye tracking" OR "eye-tracking" OR "eyetracking")
 در شبکه ی دانش^۱ [2]

¹ Web of science



شکل (۲-۸) : درصد موضوعات مختلف بین مقالات پیدا شده در شکل (۲-۷) [2]

همچنین، تعقیب نگاه در تحقیقات یادگیری و پروسه های شناختی مورد استقبال دانشمندان واقع شده است. در تحقیقات مختلف، اندازه گیری های مختلفی از تعقیب نگاه انجام می شود که در جدول (۲-۱) نمونه هایی از اندازه گیری های پر استفاده آمده است [6].

جدول (۲-۱) : اندازه گیری های پر کاربرد از تعقیب نگاه در تحقیقات [6]

	ترکیبی	ساکاد ^۱	تثبیت ^۲
زمانی	مدت کل مطالعه	مدت ساکاد	مدت کلی تثبیت
	زمان اولین دور		مدت نگاه
	زمان دوباره خواندن		میانگین مدت تثبیت
			مدت اولین تثبیت
			زمان اولین تثبیت
			مدت تثبیت دوباره
			نسبت مدت تثبیت
مکانی	الگوی مسیر خواندن	طول ساکاد	مکان تثبیت
			دنباله ی تثبیت ها
تعداد		تعداد ساکادها	تعداد تثبیت ها
		تعداد حرکات بین خواندن	تعداد میانگین تثبیت ها
			تعداد تثبیت های دوباره
			احتمال تعداد تثبیت ها

¹ Saccade

² Fixation

۷-۲- جمع بندی

از بین روش های گفته شده، نمی توان از روش هایی استفاده کرد که نیازمند نصب الکتروود یا گذاشتن لنز هستند یا به هر طریقی مانع آزادی بیمار می شوند، زیرا در آزمایش تاثیر خواهد داشت. بنابراین روش انتخاب شده، روش مبتنی بر پردازش تصویر است که با یک دوربین ثابت انجام می شود (به دلیل گفته شده نمی توان این دوربین را روی سر قرار داد). گرچه این روش به طور متوسط دقت پایین تری نسبت به سایر روش ها دارد، اما ارزان تر و برای استفاده ساده تر است (تنها به یک دوربین و یک برنامه نیاز است که نقطه ی نگاه را مستقیما محاسبه می کنند). همچنین می توان با بالا بردن کیفیت و سرعت دوربین و قدرت پردازشی دستگاه، دقت و سرعت سیستم را افزایش داد و باعث بهبود کارایی آن شد.

فصل ۳: پیاده سازی

۱-۳- مقدمه

سیستم خواسته شده باید برای بیماران استفاده شود و برای نتیجه ی بهتر، در انجام آزمایش بیمار باید آزاد باشد. بنابراین نمی توان از روش هایی استفاده کرد که برای اندازه گیری نیاز به نصب الکتروود یا گذاشتن لنز دارند. در نتیجه از پردازش تصویر برای تعقیب چشم و نگاه استفاده می کنیم، و به دلیل گفته شده (آزادی بیمار هنگام آزمایش) دوربین نمی تواند روی سر بیمار نصب شود، پس سیستم علاوه بر تعقیب چشم، باید موقعیت و زوایای سر را نیز به دست بیاورد تا با ترکیب آنها نقطه ی نگاه را محاسبه کند. این انتخاب همچنین باعث ارزان شدن و نصب ساده تر این سیستم می شود چون تنها یک دوربین (یا حتی وبکم) برای این کار کافی است.

در روش حل مسئله، ما ابتدا به تعقیب نگاه با فرض ثابت بودن سر پرداخته ایم، به این منظور روش های مختلفی پیدا و آزمایش کردیم که در بخش های بعد بررسی خواهند شد، و در نهایت بهترین روش که استفاده از کتابخانه ی mediapipe پایتون بود را برگزیدیم.

در نهایت با استفاده از همین کتابخانه زاویه و مکان سر نیز به دست آمده و به ترکیب زوایا و مکان های چشم و سر برای محاسبه ی نقطه ی نهایی نگاه پرداختیم. برای پردازش تصویر در این پروژه از کتابخانه ی opencv استفاده می شود.

۲-۳- انتخاب محیط توسعه

گزینه های مختلفی برای پیاده سازی این سیستم وجود داشت، مثل MATLAB، C++، C# و پایتون. سیستم باید طوری ساخته می شد که امکان اجرا روی دستگاه های مختلف را داشته باشد بنابراین MATLAB گزینه ی خوبی نبود. نرم افزار نوشته شده برای خود روبات به زبان C# است و مطلوب بود که این سیستم هم به همین زبان نوشته شود تا بتوان آن را با نرم افزار ترکیب کرد، اما

کتابخانه ی OpenCV که از معروف ترین کتابخانه های پردازش تصویر است امکان اجرای مستقیم روی C# را نداشت. به دلیل سریع بودن حرکات چشم، سرعت سیستم اهمیت پیدا می کند و باید بالا باشد، با توجه به [7] سرعت C++ از پایتون بیشتر است اما در عمل به دلیل پیشنیازهای لازم برای نصب OpenCV روی C++ و مشکلات ارتباط برقرار کردن با درایور دوربین، دریافت تصویر از دوربین در پایتون بسیار سریع تر از C++ انجام می شد. در نهایت با توجه به موارد گفته شده، پایتون برای اجرای این پروژه انتخاب شد که مزیت های دیگری مثل امکان استفاده از کتابخانه های پیشرفته ی آن به خصوص mediapipe – که بخش اصلی کار با آن انجام شد – و سرعت توسعه ی بیشتر را به همراه داشت.

برای ارتباط برنامه ی نوشته شده با نرم افزار روبات می توان مشابه [8] از پروتکل هایی نظیر UDP¹ استفاده کرد. همچنین می توان برای بالا رفتن سرعت، نسخه ی نهایی پروژه ی انجام شده را با C++ و یک سیستم مناسب پیاده کرد، یا حتی با استفاده از wrapper آن را مستقیماً در نرم افزار روبات به زبان C# قرار داد.

۳-۳- روش haar آبشاری^۲

یکی از روش های تشخیص اشیا در تصاویر استفاده از ویژگی های haar^۳ و روش haar آبشاری است. این روش توسط پاول ویولا^۴ و مایکل جونز^۵ در [9] در سال ۲۰۰۱ معرفی شد. این روش یکی از روش های یادگیری ماشین است که در آن با استفاده از تعداد زیادی تصویر مثبت و منفی (دارای شیء مورد نظر و فاقد آن) یک تابع آبشاری آموزش داده می شود [10, 11].

یک ویژگی haar از در نظر گرفتن مستطیل هایی اطراف نقطه ی مورد نظر (مطابق شکل (۳-۱)) و جمع مقدار پیکسل ها در هر مستطیل و در نهایت کم کردنشان از یکدیگر به دست می آید. برای مثال در تشخیص صورت، نواحی چشم از نواحی گونه تیره تر هستند؛ در نتیجه یک ویژگی haar برای تشخیص صورت دو مستطیل روی هم در منطقه ی چشم و گونه می باشد [12].

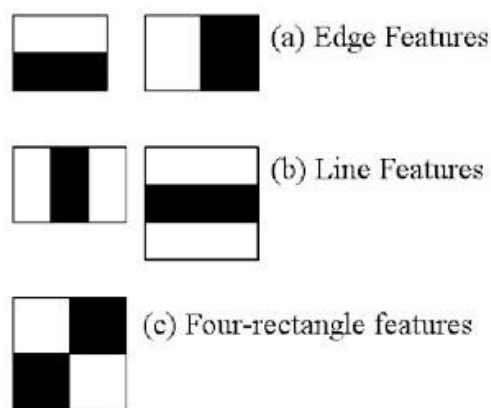
¹ User Datagram Protocol

² Haar cascade

³ Haar features

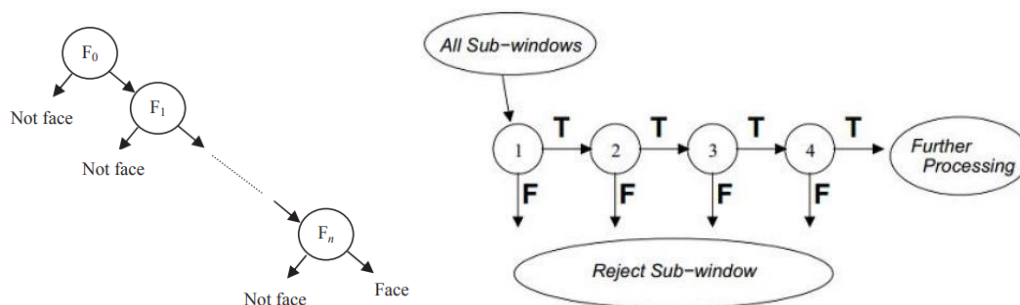
⁴ Paul Viola

⁵ Michael Jones



شکل (۳-۱) چند نمونه از ویژگی های haar. جمع پیکسل های داخل مستطیل های سیاه از جمع پیکسل های داخل مستطیل های سفید کم می شود [10, 11]

یک دسته بندی کننده ی آبشاری^۱ از چند مرحله^۲ تشکیل می شود که هر مرحله از یادگیرنده های ضعیف تشکیل می شود. مرحله ها به دنبال هم قرار میگیرند تا سیستم نهایی تشکیل شود. برای این که این سیستم به درستی کار کند، هر مرحله باید خطای منفی^۳ کمی داشته باشد [12] آبشاری بودن این روش به این معناست. به این صورت اگر برای مثال هر مرحله ۹۹.۹٪ چهره ها را تشخیص دهد و تعداد زیادی (برای مثال ۵۰٪) از موارد دیگر را نیز به عنوان چهره شناسایی کند؛ با قرار دادن ۲۰ مرحله به صورت آبشاری مطابق شکل (۳-۲) می توان به نرخ تشخیص غلط (False Positive) برابر $0.0001\% \approx 0.5^{20}$ و نرخ تشخیص $98\% \approx 0.999^{20}$ رسید [13].



شکل (۳-۲) : روند اجرای مراحل برای دسته بندی کننده ی آبشاری [12, 13]

با استفاده از این توابع مدل هایی برای تشخیص اشیای مختلف (از جمله صورت، چشم و ...) آموزش می بینند و در قالب فایل های xml ذخیره می شوند (لینک مخزن این نمونه فایل ها در پیوست قرار داده شده). با استفاده از کتابخانه ی OpenCV و این فایل ها می توان تشخیص چهره

¹ Cascade Classifier

² Stage

³ False Negative

و چشم را با این روش انجام داد [14].

در ابتدای پروژه از این روش برای تشخیص چهره و چشم استفاده شد، ولی گرچه این روش با دقت خوبی می تواند در تشخیص چهره و چشم موفق باشد، اما به دلیل نیاز به حرکت این فیلترها برای استخراج ویژگی های haar روی تصویر توان پردازشی زیادی نیاز دارد. به همین دلیل باعث کندی سیستم می شود. همچنین هنگام تشخیص چشم، مرزها را به درستی و ثابت تشخیص نمی دهد (در صورتی که برای استخراج مکان مردمک نسبت به صورت به تصویر نسبتا ثابتی از چشم نیاز داریم) و حتی برخی اوقات چندین پنجره را روی هم به عنوان چشم تشخیص می دهد که مطلوب نیست.

۴-۳- کتابخانه dlib

Dlib یک کتابخانه ی مدرن برای C++ است که الگوریتم های یادگیری ماشین و امکاناتی برای نوشتن برنامه های پیچیده برای حل مسائل دنیای واقعی را شامل می شود [15]. این کتابخانه با استفاده از ترکیب ویژگی های هیستوگرام گرادیان جهت دار^۱ و دسته بندی کننده ی خطی^۲، یک هرم تصویر^۳ و شمای تشخیص پنجره ی متغیر^۴ صورت را تشخیص می دهد [16] و به کمک درخت های رگرسیون^۵ ۶۸ نقطه^۶ ی صورت را شناسایی می کند [17]. داده های مدل آموزش دیده برای این کار در یک فایل dat ذخیره شده اند که در برنامه می توانند مورد استفاده قرار گیرند (لینک این فایل در پیوست قرار داده شده است).

¹ Histogram of oriented gradients (HOG) features

² Linear classifier

³ Image pyramid

⁴ Sliding window detection schema

⁵ Regression trees

⁶ landmark



شکل (۳-۳) : نقاط تشخیص داده شده روی صورت توسط کتابخانه dlib

با این روش دور چشم ها به سادگی به دست می آید و می توان با داشتن تصویر چشم ها، به بررسی مکان نسبی مردمک پرداخت. همچنین با داشتن این نقاط می توان زاویه ی سر را نیز به دست آورد و در محاسبه ی نقطه ی نگاه از آن استفاده کرد. نحوه ی این محاسبات و کارهای انجام شده با این کتابخانه در بخش های بعد با جزئیات بیان خواهد شد.

زمان انجام پردازش و به دست آوردن این نقاط روی صورت برای تصویر با ابعاد 640×480 حدود ۱۶۰ میلی ثانیه بود و سرعت ۶ فریم بر ثانیه را به ما میداد. گرچه با کاهش ابعاد تصویر برای تشخیص چهره و سپس پیدا کردن نقاط روی صورت روی تصویر با کیفیت اصلی زمان پردازش به حدود ۶۰ میلی ثانیه کاهش یافت و سرعت به ۱۶ فریم بر ثانیه رسید، اما بازهم سیستم سرعت مورد نظر را نداشت و برای کاربرد این پروژه کند بود.

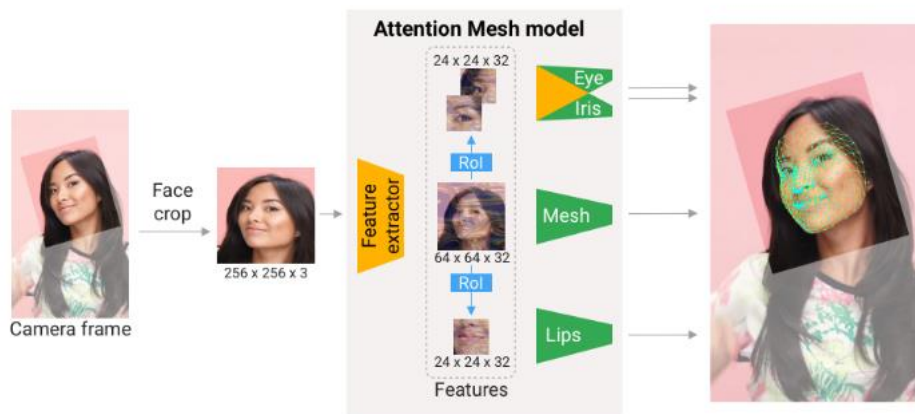
۳-۵- کتابخانه mediapipe

Mediapipe کتابخانه ای برای محیط های توسعه ی مختلف است که برای مسائل مربوط به پخش زنده و استریمینگ^۱، راه حل هایی مبتنی بر یادگیری ماشین ارائه می دهد. بعضی از این مسائل عبارتند از: تشخیص صورت، شبکه ی صورت، تشخیص عنبیه، دستها و ... [18]. بخش شبکه سازی صورت این کتابخانه ۴۶۸ نقطه را روی صورت به صورت سه بعدی تخمین می زند. این کار با استفاده از دو شبکه ی عصبی عمیق بی درنگ^۲ انجام می شود، یکی روی کل تصویر اعمال می شود و مکان صورت را پیدا می کند و دیگری مدل سه بعدی نقاط روی صورت را به وسیله ی رگرسیون

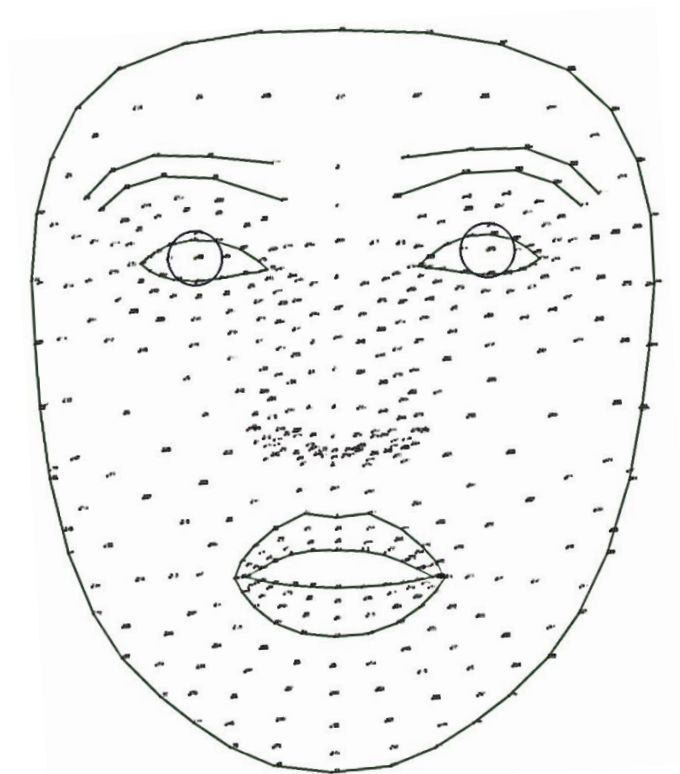
¹ Streaming

² Real time

تخمین می‌زند [19]. همچنین این کتابخانه امکان مکان یابی عنبیه را نیز به واسطه ی آموزش دیدن با حدود ۵۰ هزار عکس دارد [20].



شکل (۳-۴) روند تشخیص صورت و تخمین نقاط روی صورت توسط Mediapipe [19]



شکل (۳-۵) : نقاط تشخیص داده شده روی صورت توسط کتابخانه ی mediapipe

به دلیل استفاده از شبکه های عصبی و سازگاری با GPU، سرعت پردازش و تخمین نقاط صورت با این کتابخانه بسیار بالاست، به طوری که دریافت تصویر و تخمین این نقاط روی سیستم مشابه تست شده برای کتابخانه dlib، روی تصویر با ابعاد 640×480 تنها ۵۰ میلی ثانیه زمان

می‌برد که به وسیله ی آن می‌توان با یک سیستم و وبکم عادی به سرعت ۲۰ فریم بر ثانیه رسید. بنابراین امکان رسیدن به سرعت های بالاتر روی سیستم و دوربین پیشرفته تر به سادگی وجود خواهد داشت.

مزیت های دیگر (علاوه بر سرعت بسیار بیشتر) استفاده از این کتابخانه در مقایسه با dlib عبارتند از: تعداد نقاط بسیار بیشتر، سه بعدی بودن نقاط، وجود نقاط اطراف و مرکز عنبیه، نامتقارن بودن نقاط (در کتابخانه dlib نقاط نسبت به محور عمودی وسط صورت متقارن بودند، بنابراین برای مثال امکان تشخیص بسته شدن هر چشم به صورت جداگانه وجود نداشت) و دقت بیشتر

۳-۶- تشخیص مکان عنبیه و زاویه ی چشم

در ابتدا با فرض ثابت بودن سر به بررسی مکان نگاه با توجه به مکان عنبیه پرداخته شد. روش های متفاوتی برای این کار آزمایش شد که در ادامه بیان می‌شود.

۳-۶-۱- میزان باز بودن چشم و پلک زدن

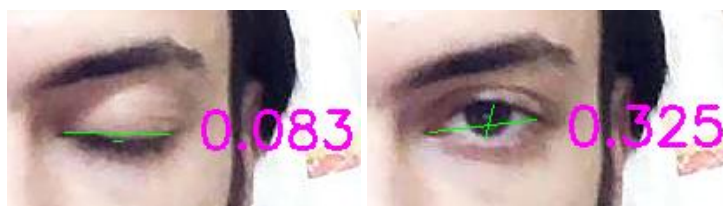
تشخیص باز یا بسته بودن چشم بسیار مهم است، زیرا در حالت بسته بودن چشم باید محاسبه ی نقطه ی نگاه متوقف شود و بسته بودن چشم اعلام شود، در غیر این صورت نتیجه نامطلوب و غیر واقعی خواهد بود (در حالی که شخص چشمش بسته است جایی به عنوان نقطه ی نگاه تعیین می‌شود!).

این کار به طور معمول با اندازه گیری نسبت چشم (تقسیم اندازه ی عمودی بر افقی یا برعکس) انجام می‌شود. به این صورت اگر این نسبت از عددی عبور کرد بسته بودن چشم اعلام می‌شود [21, 22, 23, 24].

جدول (۳-۱) : شماره نقاط انتخاب شده برای اندازه گیری نسبت چشم ها برای هر کدام از کتابخانه ها

mediapipe	dlib	نقطه
۳۳	۳۶	سمت چپ چشم چپ
۱۳۳	۳۹	سمت راست چشم چپ
۱۵۹	نقطه وسط ۳۷ و ۳۸	بالای چشم چپ
۱۴۵	نقطه وسط ۴۰ و ۴۱	پایین چشم چپ
۳۶۲	۴۲	سمت چپ چشم راست
۲۶۳	۴۵	سمت راست چشم راست
۳۸۶	نقطه وسط ۴۳ و ۴۴	بالای چشم راست
۳۷۴	نقطه وسط ۴۶ و ۴۷	پایین چشم راست

خط افقی برای هر چشم بین نقاط راست و چپ، و خط عمودی بین نقاط بالا و پایین در نظر گرفته می شود. نسبت طول خط افقی به طول خط عمودی را به عنوان معیار در نظر می گیریم. آستانه ی بسته شدن چشم با سعی و خطا حدود ۰.۱ تا ۰.۱۸ به دست آمد، ولی برای هر شخص و هر چشم باید در مرحله ی کالیبراسیون جداگانه محاسبه شود.



شکل (۳-۶) : محاسبه ی نسبت چشم برای تشخیص پلک زدن

این کار ابتدا با کتابخانه ی dlib انجام شد ولی با اضافه شدن کتابخانه ی mediapipe به پروژه، از این کتابخانه استفاده شد. به این صورت امکان تشخیص بسته شدن هر چشم به صورت جداگانه نیز فراهم شد (همان طور که گفته شد نقاط چشم در کتابخانه ی dlib باهم باز و بسته می شدند).

۲-۶-۳- مکان و زاویه ی افقی

در مرحله ی اول مکان چشم ها در تصاویر به کمک نقاط به دست آمده روی صورت به دست آمدند.

جدول (۳-۲) : نقاط دور چشم برای هر کدام از کتابخانه ها

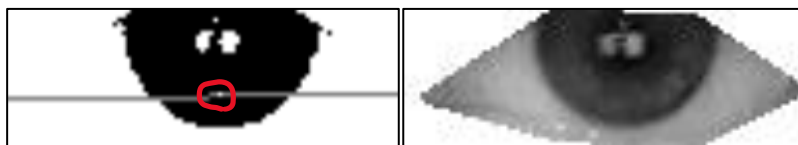
mediapipe	dlib	چشم
۳۶۲، ۳۸۲، ۳۸۱، ۳۸۰، ۳۷۴، ۳۷۳، ۳۹۰، ۳۴۹، ۲۶۳، ۴۶۶، ۳۸۸، ۳۸۷، ۳۸۶، ۳۸۵، ۳۸۴، ۳۹۸	۴۲ تا ۴۷	چشم راست
۱۳۳، ۱۵۵، ۱۵۴، ۱۵۳، ۱۴۵، ۱۴۴، ۱۶۳، ۷، ۳۳، ۲۴۶، ۱۶۱، ۱۶۰، ۱۵۹، ۱۵۸، ۱۵۷، ۱۷۳	۳۶ تا ۴۱	چشم چپ



شکل (۳-۷) : خط بین نقاط دور چشم در کتابخانه های mediapipe و dlib

روشی که در ابتدا برای مکان یابی افقی عنبیه مورد استفاده قرار گرفت، استفاده از تیره بودن عنبیه و روشن بودن صلبیه بود. به این صورت که با استفاده از تصویر خاکستری شده، نقاطی که شدت آنها از آستانه ای کمتر بود انتخاب می شدند و به عنوان عنبیه در نظر گرفته می شدند. این روش در [21, 22, 24] مورد استفاده قرار گرفته اما با تقسیم افقی چشم به دو یا سه ناحیه و شمردن تعداد پیکسل های پایین تر از حد آستانه در هر ناحیه، مشخص می کردند عنبیه در کدام ناحیه قرار دارد و در نهایت تنها بیان می کردند نگاه به راست، چپ یا وسط است. اما در این پروژه، ما می خواهیم مکان چشم را به صورت پیوسته به دست بیاوریم.

برای این کار، ابتدا با استفاده از نقاط دور چشم، کل پیکسل های غیر از چشم در تصویر سفید می شوند (تا در محاسبات خللی وارد نکنند). سپس با اعمال حد آستانه، نقاط عنبیه به صورت تقریبی به دست می آیند. برای به دست آوردن مرکز عنبیه و مردمک، روی خط افقی چشم (خط واصل بین دو گوشه ی چشم) میانگین وزن دار گرفته می شود، به این صورت که وزن نقاط با شدت پایین تر از حد آستانه ۱ و وزن سایر نقاط صفر در نظر گرفته می شود؛ با این روش مرکز نقاط با شدت کمتر از حد آستانه به دست می آید که تقریباً برابر مرکز مردمک و عنبیه است.



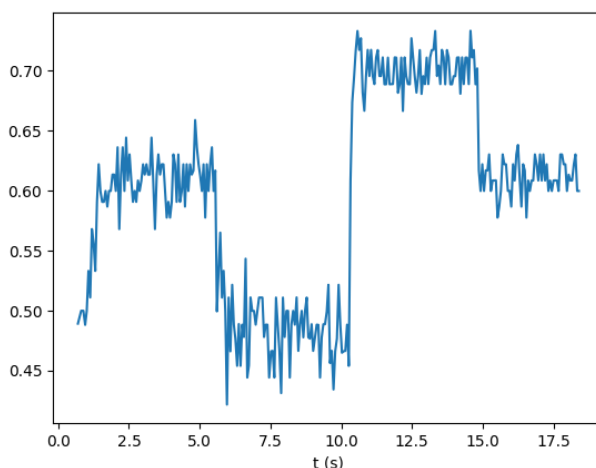
شکل (۳-۸) : تبدیل به تصویر خاکستری و اعمال ماسک (راست)، اعمال حد آستانه و رسم خط محاسبه ی میانگین وزن دار (خاکستری) و نقطه ی به دست آمده (سفید مشخص شده با دایره ی قرمز) (چپ)



شکل (۳-۹) جابجایی نقطه ی سفید با جابجایی چشم

برای اندازه گیری مکان نسبی مردمک، فاصله ی نقطه ی به دست آمده تا گوشه چشم بر طول چشم (فاصله ی دو گوشه ی چشم) تقسیم می شود. به این صورت چون با عقب و جلو شدن شخص، هر دو فاصله به یک نسبت کوچک و بزرگ می شوند و مقدار به دست آمده از تقسیم این دو طول ثابت می ماند.

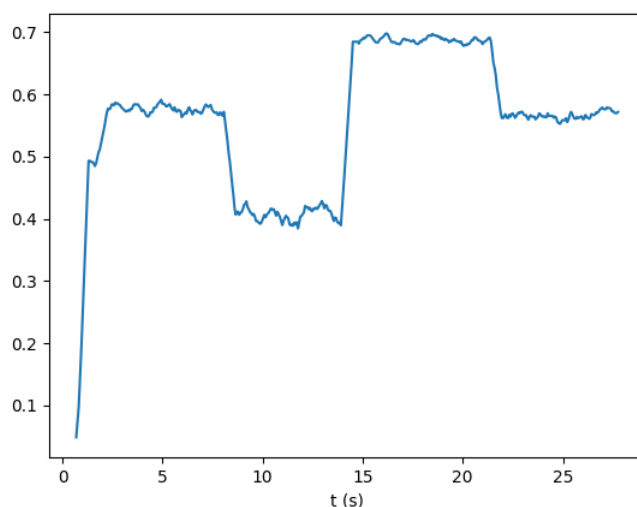
تصویر به دست آمده از آستانه گذاری در این روش به دلیل نویز پرش های زیادی دارد، که باعث می شود پیکسل های سیاه و سفید مدام در تغییر باشند و این موضوع باعث ایجاد نویز روی مکان نقطه ی به دست آمده می باشد. منبع دیگر این نویز، لرزش نقاط تخمین زده شده روی صورت است، با این لرزش طول خط واصل دو گوشه ی چشم و خط واصل مرکز مردمک و گوشه ی چشم در حد چند پیکسل تغییر دارد.



شکل (۳-۱۰) : نمودار مکان نسبی مرکز عنیه به دست آمده به روش آستانه گذاری (نگاه به وسط، راست، چپ و سپس وسط)

با میانگین گیری این نقاط در طول زمان، می توان این نویز را کاهش داد. اما همچنان نویز

وجود خواهد داشت و علاوه بر آن، سیستم دارای تاخیر می شود.



شکل (۱۱-۳) : نمودار نقاط شکل (۳-۱۰) با اعمال فیلتر میانگین ۱۰ نقطه ای (نگاه به وسط، راست، چپ و سپس وسط)

می توان این نویز را با روش های دیگری نیز کاهش داد، از جمله اعمال فیلتر مکانی (مثل تار کردن گوسی)^۱ روی تصویر برای کاهش پرش ها، استفاده از تصویر خاکستری^۲ به جای تصویر آستانه گذاری شده (به این صورت پیکسل ها فقط سیاه یا سفید نیستند و بازه ای بین صفر تا ۲۵۵ را می توانند بگیرند، و این باعث کاهش طول هر پیکسل می شود. به این صورت وقتی رنگ یک پیکسل می خواهد تغییر کند، به جایی یک پرش ۲۵۵ واحدی، یک پرش کوچک تر خواهد داشت)

ایراد دیگر این روش، وجود سایه گوشه ی چشم است که می تواند باعث کشیده شده ی نقطه ی میانگین به دست آمده به سمت گوشه ی چشم شود. همچنین به دلیل نویز پیکسل هایی از صلبیه در لحظاتی سیاه می شوند که با توجه به کم بودن تعداد پیکسل چشم، اثر قابل توجهی روی میانگین می گذارد. این نقاط و سیاهی های دیگر مثل سایه افتادن ها را می توان با کوچک کردن^۳ و سپس بزرگ کردن^۴ بخش های سیاه تصویر حذف کرد (به این صورت در مرحله ی کوچک کردن بخشهای با مساحت کم از بین می روند و با دوباره بزرگ کردن بخش های باقی مانده به ابعاد قبلی خودشان باز می گردند و دورشان نیز نرم تر می شود). همچنین می توان مشابه [25] منحنی بسته^۵

¹ Gaussian Blur

² Grayscale

³ erode

⁴ dilate

⁵ Contour

ی دور بخش های سیاه تصویر را به دست آورد و از بین آنها منحنی با بیشترین مساحت را انتخاب کرد.

این روش همچنین حساسیت بالایی به شدت و زاویه ی نور (به دلیل ایجاد سایه) و رنگ چشم دارد، به این ترتیب سیستم پیاده سازی شده با این روش هر بار و برای هر شخص نیاز به کالیبراسیون و پیدا کردن آستانه ی مناسب برای آن شرایط خواهد داشت. به علاوه، با نگاه به بالا خط واصل دو چشم عنبیه را قطع نخواهد کرد (شکل (۳-۱۲)) و محاسبه مرکز عنبیه به روش گفته شده امکان پذیر نخواهد بود (می توان این مشکل را نیز با میانگین گیری همه ی نقاط تصویر به جای فقط یک خط پس از حذف نویز حل کرد).

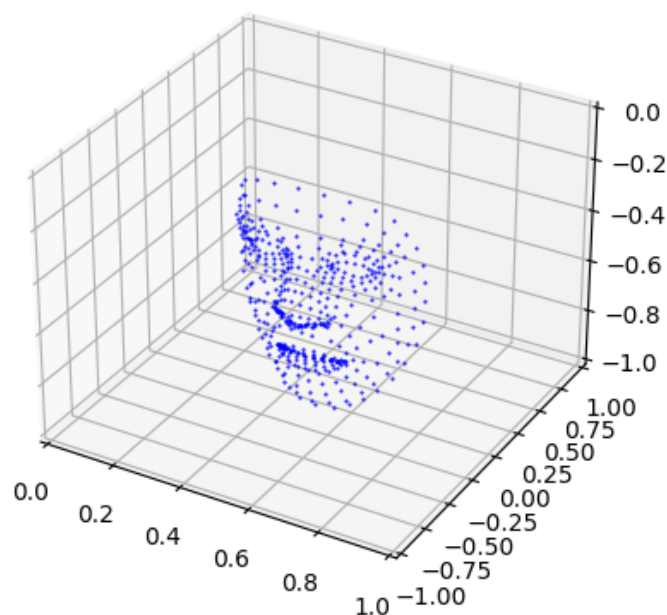


شکل (۳-۱۲) : با نگاه به بالا، خط واصل دو گوشه ی چشم عنبیه را قطع نمی کند

پس از رسیدن به کتابخانه ی mediapipe روش بهتر و سریع تری برای مکان یابی عنبیه پیدا شد و به همین دلیل، نیازی به پیاده سازی راه حل های گفته شده وجود نداشت. کتابخانه ی mediapipe علاوه بر ۴۶۸ نقطه ی صورت، می تواند چهار نقطه دور هر عنبیه و نقطه ی مرکز هر کدام را تخمین بزند (این نقاط در شکل (۳-۵) قابل مشاهده هستند). به علاوه، نقاط به دست آمده روی صورت در سه بعد به دست می آیند.

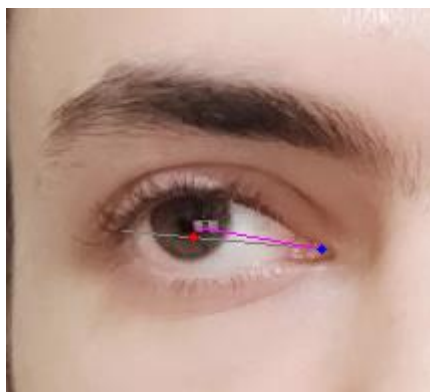


شکل (۳-۱۳) : خط بسته ی حاصل از نقاط دور چشم (قرمز)، کوچکترین دایره شامل چهار نقطه اطراف عنبیه (سبز) و مرکز عنبیه و مردمک (بنفش)



شکل (۳-۱۴) : نقاط سه بعدی صورت که توسط mediapipe تخمین زده شده اند

با داشتن مرکز عنبیه (یا به دست آوردن آن با میانگین گیری از نقاط اطراف عنبیه) می توان مکان آن را نسبت به سر به دست آورد. برای این کار می توان فاصله ی این نقطه از گوشه ی چشم را بر طول خط واصل دو گوشه ی چشم تقسیم کرد [26]. اما با استفاده از این روش، بالا و پایین رفتن چشم در این اندازه گیری تاثیرگذار است. بنابراین تصویر این نقطه را روی خط واصل دو گوشه ی چشم به دست می آوریم و فاصله ی این نقطه ی به دست آمده را تا گوشه ی چشم در نظر میگیریم.



شکل (۳-۱۵) : خط واصل گوشه ی چشم و مرکز عنبیه (بنفش)، خط واصل دو گوشه ی چشم (خاکستری) و تصویر نقطه ی مرکز عنبیه روی خط واصل بین دو چشم (قرمز)

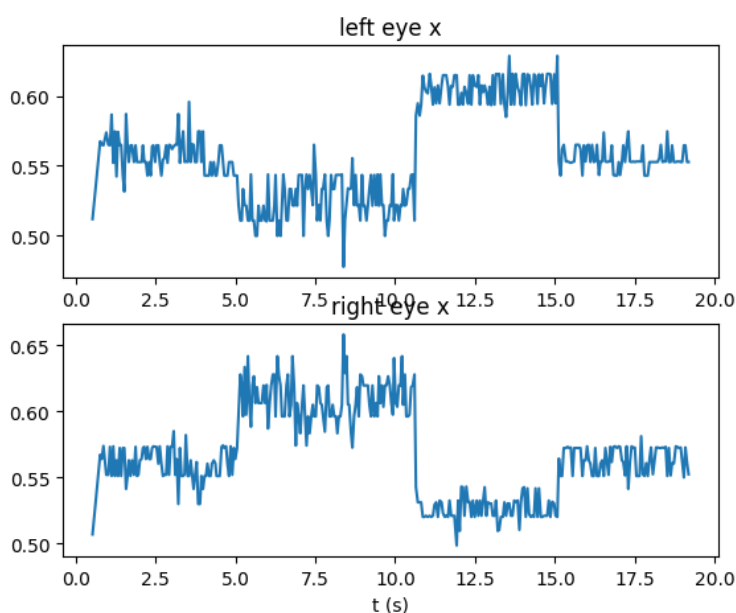
با حرکت چشم، پلک نیز به دنبال آن حرکت میکند. این حرکت باعث تغییر نقاط دور چشم و در نتیجه جابجایی خط واصل بین دو گوشه ی چشم می شود. بنابراین این خط برای اندازه گیری مکان عنبیه مناسب نیست. روش دیگری که به کار گرفته شد استفاده از خط واصل بین گوشه های

دو چشم بود. این خط نسبت به سر تقریباً ثابت می ماند و می تواند معیار خوبی باشد. بنابراین تصویر مرکز عنبیه روی امتداد این خط محاسبه شد.



شکل (۱۶-۳) : خط واصل گوشه های دو چشم (سبز)، تصویر مرکز عنبیه ها روی امتداد خط واصل گوشه های دو چشم (بنفش)، فاصله ی افقی مرکز عنبیه ها (قرمز) و فاصله ی عمودی مرکز عنبیه ها (آبی)

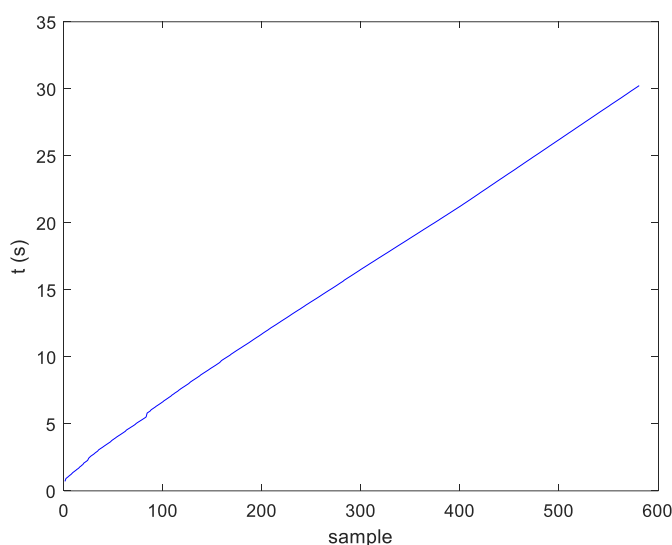
با تقسیم فاصله ی تصویر مرکز عنبیه روی امتداد خط واصل گوشه های دو چشم تا گوشه ی چشم، بر فاصله ی گوشه های چشم (برای از بین بردن اثر فاصله) معیار مناسبی از مکان افقی نسبی عنبیه به دست می آید.



شکل (۱۷-۳) : مکان افقی محاسبه شده برای هر کدام از چشم ها، ابتدا به مرکز صفحه نگاه شده و سپس به ترتیب سمت راست و چپ صفحه. و در نهایت دوباره به مرکز صفحه نگاه شده

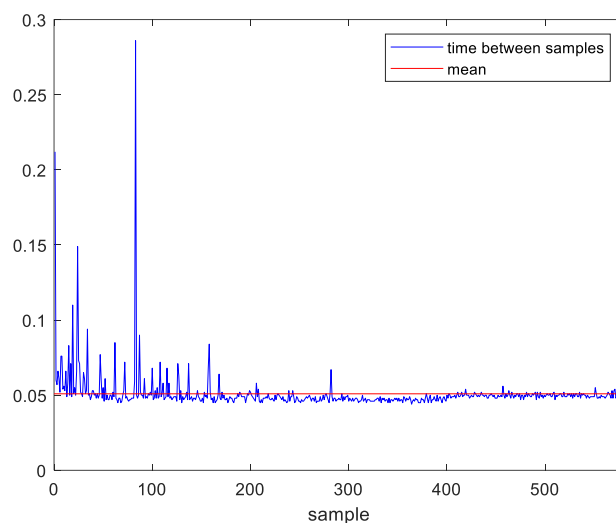
با توجه به شکل (۱۷-۳) ، طبق انتظار با نگاه به راست فاصله ی مرکز مردمک چشم راست از گوشه ی آن زیاد و این فاصله برای چشم چپ کم شده است، و هنگام نگاه به چپ برعکس این حالت پیش آمده است. بنابراین نگاه به نقاط مختلف صفحه قابل مشاهده است و می توان از این

اندازه گیری، نقطه ی نگاه را تخمین زد. اما همان طور که مشاهده می شود این اندازه گیری هم نویز قابل توجهی دارد که هنگام محاسبه ی نقطه ی نگاه (که در آن این مقدار به دست آمده در ضریب بزرگی ضرب می شود) این نویز نیز بسیار بزرگ می شود. علت این نویز لرزش نقاطی است که mediapipe روی صورت تخمین می زند. برای حذف این نویز نیاز به اعمال فیلتر داریم. برای طراحی فیلتر ابتدا یک نمونه از خروجی محاسبه ی مکان افقی عنبیه ذخیره، و سپس بازه های زمانی بین نمونه ها بررسی شد.



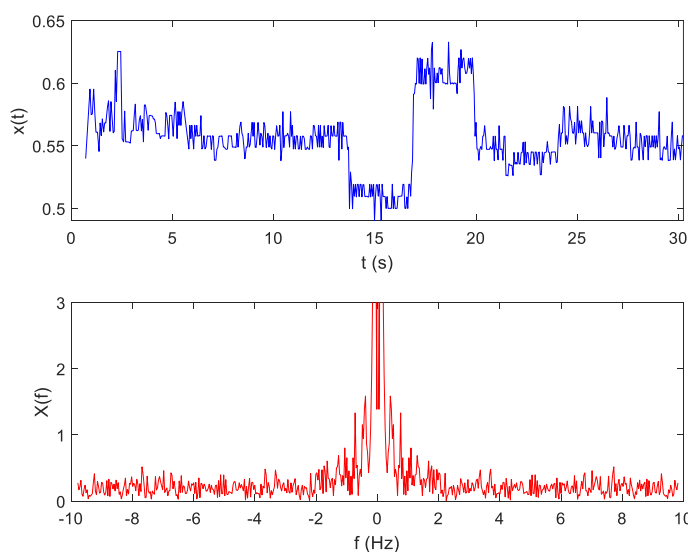
شکل (۳-۱۸) : زمان نمونه های سیگنال ذخیره شده

همانطور که در شکل (۳-۱۸) مشاهده می شود، بازه های زمانی تقریباً ثابت هستند و می توان این سیگنال را یک سیگنال نمونه برداری شده ی تک نرخی در نظر گرفت. میانگین فاصله ی زمانی بین نمونه ها را به عنوان تناوب نمونه برداری در نظر می گیریم.



شکل (۱۹-۳): فاصله ی زمانی بین نمونه ها

به این صورت تناوب نمونه برداری ۵۰.۹ میلی ثانیه و در نتیجه فرکانس نمونه برداری ۱۹.۶۴۲۳ هرتز به دست می آید که طراحی فیلتر دیجیتال را با آن انجام می دهیم.



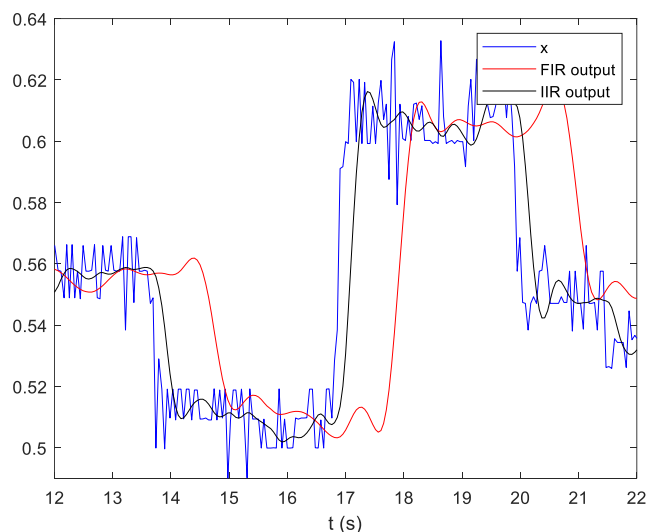
شکل (۲۰-۳): سیگنال در حوزه ی زمان و طیف آن در حوزه ی فرکانس

با توجه به طیف سیگنال در شکل (۳-۲۰)، به نظر می رسد تا فرکانس ۲ هرتز برای حرکت چشم کافی باشد و از بعد از آن نویز شروع شود. با بازی کردن با مقادیر دامنه و فرکانس قطع، یک فیلتر IIR و یک فیلتر FIR به کمک نرم افزار متلب طراحی شد (روابط ۳-۱ و ۳-۲).

$$G_{FIR}(z) = 0.0011422 + 0.0051294z^{-1} + 0.008849z^{-2} + 0.011549z^{-3} + 0.012542z^{-4} + 0.011337z^{-5} + 0.0077619z^{-6} + 0.0020454z^{-7} + -0.0051582z^{-8} + -0.012812z^{-9} + -0.019604z^{-10} + -0.024115z^{-11} +$$

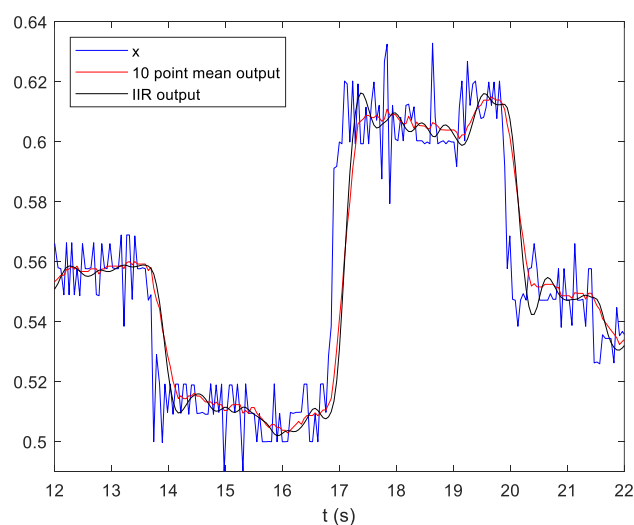
$$\begin{aligned}
 & -0.02502z^{-12} + -0.021293z^{-13} + -0.012392z^{-14} + 0.0016148z^{-15} + \\
 & 0.019999z^{-16} + 0.041412z^{-17} + 0.064017z^{-18} + 0.085696z^{-19} + \\
 & 0.1043z^{-20} + 0.11791z^{-21} + 0.1251z^{-22} + 0.1251z^{-23} + 0.11791z^{-24} + \\
 & 0.1043z^{-25} + 0.085696z^{-26} + 0.064017z^{-27} + 0.041412z^{-28} + \\
 & 0.019999z^{-29} + 0.0016148z^{-30} + -0.012392z^{-31} + -0.021293z^{-32} + \\
 & -0.02502z^{-33} + -0.024115z^{-34} + -0.019604z^{-35} + -0.012812z^{-36} + \\
 & -0.0051582z^{-37} + 0.0020454z^{-38} + 0.0077619z^{-39} + 0.011337z^{-40} + \\
 & 0.012542z^{-41} + 0.011549z^{-42} + 0.008849z^{-43} + 0.0051294z^{-44} + \\
 & 0.0011422z^{-45}
 \end{aligned}
 \quad \text{(رابطه ۱-۳)}$$

$$G_{IIR}(z) = \frac{0.0084 + 0.0252z^{-1} + 0.0252z^{-2} + 0.0084z^{-3}}{1 - 2.0727z^{-1} + 1.5292z^{-2} - 0.3892z^{-3}}
 \quad \text{(رابطه ۲-۳)}$$



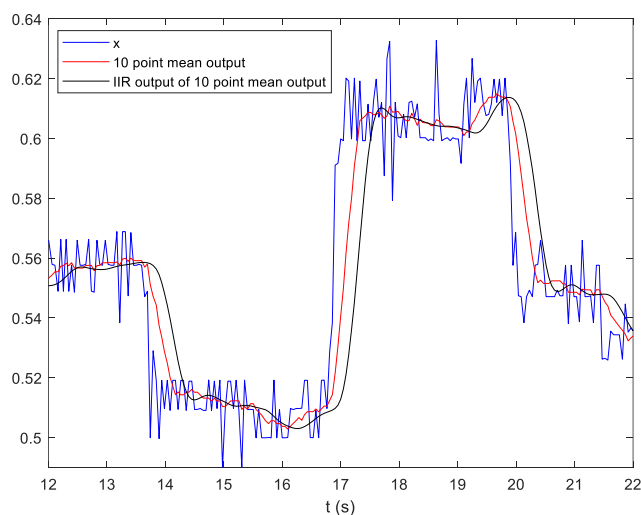
شکل (۳-۲۱) : نتیجه ی اعمال فیلتر FIR و IIR طراحی شده روی سیگنال

فیلتر FIR به دلیل داشتن جملات زیاد، تاخیر زیادی ایجاد می کند و نتیجه اش هم مشابه نتیجه ی فیلتر IIR است. بنابراین استفاده از آن مطلوب نیست.



شکل (۳-۲۲) : نتیجه ی اعمال میانگین ۱۰ نقطه ای و فیلتر IIR روی سیگنال

گرچه اعمال فیلتر IIR طراحی شده نویز را حذف می کند، ولی باعث ایجاد فراجهش^۱ در هنگام تغییرات می شود و در بخش های نسبتاً ثابت سیگنال هم موجک^۲ ایجاد می کند که مطلوب نیست. در مقابل میانگین گیری این ایرادات را ندارد ولی به مقدار کافی خروجی نرمی تولید نمی کند. بنابراین برای رسیدن به خروجی مطلوب تر از ترکیب این دو فیلتر باهم استفاده می کنیم.



شکل (۳-۲۳) : نتیجه ی اعمال میانگین ۱۰ نقطه ای و ترکیب آن با فیلتر IIR

همانطور که در شکل (۳-۲۳) مشاهده می شود با اعمال هر دوی این فیلترها خروجی نرم و با موجک کمی به دست می آید.

¹ Overshoot

² Ripple

برای استفاده از این فیلتر داخل کد پایتون، یک کلاس^۱ فیلتر نوشته شد. با دادن ضرایب صورت و مخرج تابع تبدیل فیلتر به این کلاس، می توان از فیلتر استفاده کرد. این کلاس به این صورت کار می کند که به تعداد مرتبه ی فیلتر ورودی ها و خروجی های قبلی اش را نگه می دارد؛ با گرفتن ورودی جدید، خروجی جدیدش را با ضرب کردن ضرایب صورت در ورودی های قبلی و کم کردن ضرب ضرایب مخرج در خروجی های قبلی طبق معادله تفاضلی فیلتر (رابطه ۳-۳) محاسبه می کند.

$$G(z) = \frac{b_0 + b_1 z^{-1} + \dots}{1 + a_1 z^{-1} + \dots} \Rightarrow$$

$$y[n] = b_0 x[n] + b_1 x[n-1] + \dots - a_1 y[n-1] - \dots \quad (\text{رابطه ۳-۳})$$

۳-۶-۳- مکان و زاویه ی عمودی

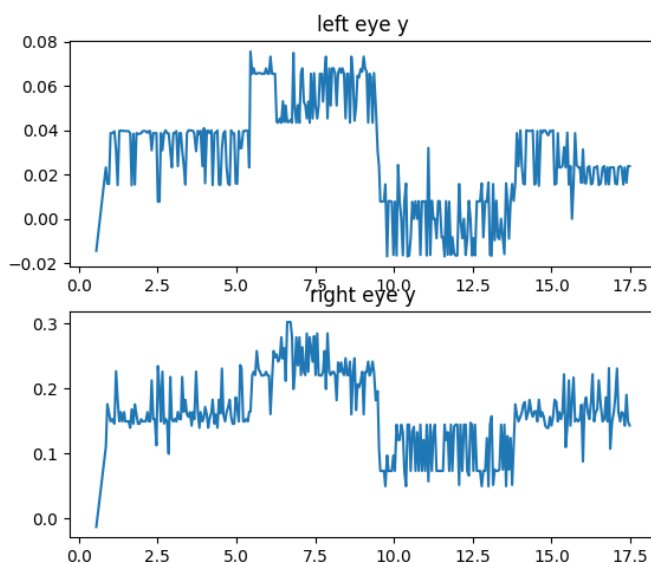
با توجه به بازه ی کمتر حرکت چشم در راستای عمودی، اندازه گیری این حرکت سخت تر از اندازه گیری حرکت افقی است. همچنین کمتر بودن تعداد پیکسل ها در راستای عمودی چشم باعث می شود که لرزش یکسان نقاط، نویز بیشتری روی اندازه گیری مکان عمودی عنبیه نسبت به مکان افقی عنبیه ایجاد کند.

پلک بالا تقریباً همزمان با عنبیه به صورت غیر ارادی بالا و پایین می رود. با توجه به این موضوع روش ابتدایی اندازه گیری مکان عمودی عنبیه استفاده از نسبت چشم بود (مشابه آن چه که برای تشخیص پلک زدن مورد استفاده قرار گرفت). اما ایراد این روش این است که فرد ممکن است در هنگام آزمایش خودش چشمش را بازتر یا بسته تر (مثلاً برای تمرکز چشمانش را جمع کند) کند، و این مورد روی نسبت اندازه گیری شده از چشم تاثیر می گذارد. بنابراین روشی که در نهایت برای این اندازه گیری به کار برده شد، استفاده از فاصله ی مرکز عنبیه تا امتداد خط واصل بین دو چشم بود شکل (۳-۱۶).

این فاصله با استفاده از ضرب خارجی محاسبه می شود، بنابراین با پایین تر رفتن مرکز عنبیه نسبت به امتداد خط واصل بین دو چشم، مقدار منفی محاسبه می شود (اگر فقط فاصله اندازه گیری

¹ Class

می شد این اتفاق نمی افتاد و فقط مقادیر مثبت داشتیم که درست نبود).



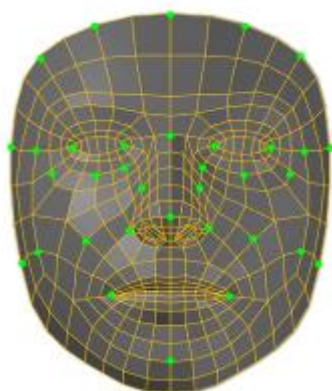
شکل (۲۴-۳) : مکان عمودی محاسبه شده برای هر کدام از چشم ها، ابتدا به مرکز صفحه نگاه شده و سپس به ترتیب بالا و پایین صفحه. و در نهایت دوباره به مرکز صفحه نگاه شده

با توجه به این که نویز مشابه نویز مکان افقی است، همان فیلتر طراحی شده بای مکان افقی روی مکان عمودی هم اعمال شد.

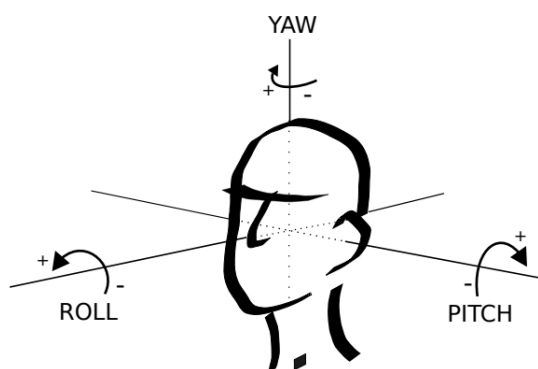
۳-۷- تشخیص موقعیت و جهت گیری سر

با توجه به سه بعدی بودن نقاطی که mediapipe روی صورت قرار می دهد، می توان جهت و موقعیت سه بعدی سر را به دست آورد. برای این کار از روش گفته شده در [8] استفاده شد (لینک مخزن گیتهاب مربوط به این منبع در پیوست قرار داده شده است).
برای به دست آوردن جهت سر، با در نظر گرفتن تعدادی از نقاط روی صورت مسئله ی PNP^۱ حل می شود و به این صورت، زوایای اوپلر مربوط به سر به دست می آیند.

¹ Perspective n points



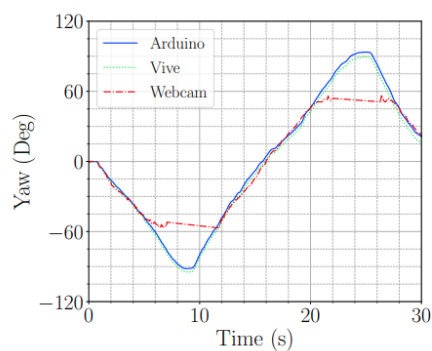
شکل (۳-۲۵) : نقاط انتخاب شده برای حل مسئله ی PnP [8]



شکل (۳-۲۶) : زوایای محاسبه شده برای سر [8]

حل مسئله ی PnP علاوه بر بردار دوران، بردار جابجایی را نیز به ما می دهد که بیان گر مکان

سر در صفحه است.



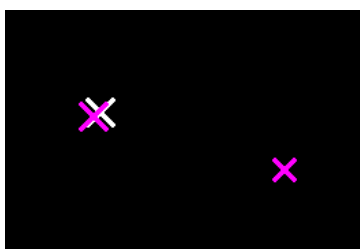
شکل (۳-۲۷) : دستگاه اندازه گیری دقت این روش با استفاده از آردوینو و Vive [8]



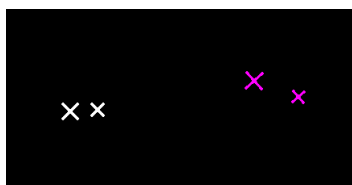
شکل (۳-۲۸) : نتیجه ی کد و جهت پیدا شده برای سر (مشخص شده با پیکان قرمز روی بینی شخص) [8]

مکان نقطه ی به دست آمده برای سر با این روش جلوی سر قرار دارد و با چرخش سر در مکان ثابت حرکت می کند. همچنین با ثابت بودن زاویه ی سر و حرکت نسبت به دوربین، چون زاویه ی سر نسبت به دوربین تغییر می کند مقدار زاویه ی محاسبه شده نیز عوض می شود. این دو مورد برای استفاده ی درست از زوایا و مکان سر باید اصلاح شوند.

برای اصلاح مکان هنگام چرخش، با توجه به قرار گرفتن نقطه روی صورت، باید ضربی از تانژانت زاویه ی سر از آن برای جبران کم شود. این ضرب در مرحله ای از کالیبراسیون به دست می آید. برای اصلاح زاویه ی سر هنگام حرکت نیز مقدار \tan^{-1} فاصله ی افقی از دوربین نسبت به فاصله ی عمودی از دوربین باید از آن زاویه کم شود.



شکل (۳-۲۹) : با جابجا شدن مکان سر (ضربدر سفید) زاویه نسبت به دوربین تغییر کرده (فاصله ی ضربدر سفید تا ضربدر بنفش کوچک) اما جبران زاویه، آن را به جای درستش برگردانده (ضربدر بنفش بزرگ که روی ضربدر سفید بودن آن زاویه ی صفر را نشان می دهد)



شکل (۳-۳۰) : با زاویه گرفتن سر (دور شدن ضربدر بنفش از ضربدر سفید)، مکان سر محاسبه شده قدری تغییر کرده (ضربدر سفید کوچک جابجا شده) اما جبران مکان، آن را سر جایش نگه داشته است (ضربدر سفید بزرگ حرکت نکرده است)

۸-۳- به دست آوردن مکان نگاه با ترکیب اطلاعات چشم ها و سر

اگر سر را ثابت فرض کنیم، رابطه ی مکان محاسبه شده ی مردمک ها (x) با نقطه ی نگاه (X) در هر محور به صورت رابطه ۳-۴ خواهد بود.

$$X = A \tan(\sin^{-1}(Bx + C)) + D \quad (\text{رابطه ۳-۴})$$

با در نظر گرفتن مکان و جهت سر، محاسبه ی مکان نگاه بسیار پیچیده خواهد شد. یک راه به دست آوردن نقطه ی نگاه در این شرایط، استفاده از مسئله ی سینماتیک مستقیم^۱ می باشد. در این صورت به سیستم سر و هر کدام از چشم ها به عنوان یک ربات سری نگاه می شود و با داشتن طول بازوها و زوایای بین آنها نقطه ی نهایی^۲ محاسبه می شود.

اما ایراد این روش کالیبراسیون بسیار سخت آن است، به دلیل وجود روابط مثلثاتی مدل غیر خطی است و نمی توان آن را به روش هایی مثل حداقل مربعات مانده ها کالیبره کرد. بنابراین سیستم حول نقطه ی صفر خطی سازی می شود و فرض می شود همه ی پارامتر های محاسبه شده با نقطه ی نگاه رابطه ی خطی دارند. با توجه به این که در کاربرد این پروژه شخص در مکانی ثابت می شود، این فرض معقول است.

زاویه ی شخص نسبت به دوربین و نسبت به صفحه ای که به آن نگاه می کند در محاسبات موثر است. این زاویه داشتن باعث از بین رفتن استقلال بین محورهای افقی و عمودی می شود، بنابراین در محاسبه ی هر یک از محورها رابطه ی خطی با محور دیگر نیز در نظر گرفته می شود. به علاوه، کروی بودن چشم قابل صرف نظر نیست (مردمک روی یک کره در حال حرکت است که انحنای آن نسبت به حرکتش قابل مقایسه است) بنابراین عبارات درجه ۳ برای مکان محاسبه شده ی مردمک را نیز وارد محاسبات می کنیم تا به واقعیت شبیه تر شود.

با توجه به این که با هر کدام از چشم ها به تنهایی می توان نقطه ی نگاه را محاسبه کرد، این محاسبه برای هر چشم به تنهایی انجام می شود و در نهایت نقطه ی بین دو نقطه ی به دست آمده به عنوان نقطه ی نگاه در نظر گرفته می شود. از آنجا که چشم ها متقارن عمل می کنند. اگر خطایی در مدل وجود داشته باشد به صورت متقارن روی نقطه ی نگاه هر چشم اثر می گذارد و باعث می شود نقطه ی میانگین آنها ثابت بماند.

¹ Forward kinematic problem

² End effector

۹-۳- کالیبراسیون

فرایند کالیبراسیون در چند مرحله انجام می‌شود. ابتدا با نشان دادن زوایا و مکان سر خواسته می‌شود که شخص سرش را طوری قرار دهد که همه ی آنها صفر شوند. سپس نقطه ای نشان داده می‌شود که فرد با چرخاندن سر به سمت آن زاویه ی ۳۰ درجه بگیرد، این کار هم در راستای عمودی و هم افقی انجام می‌شود تا ضرایب مناسب برای اصلاح مکان سر به دست بیایند.

در مرحله ی بعد چند نقطه نشان داده می‌شود که شخص سرش را روبروی آنها قرار دهد، با این کار ضرایب نسبت حرکت سر در تصویر به حرکت نگاه در صفحه به دست می‌آید.

سپس، یک نقطه در وسط صفحه و یک شبکه ی ۳ در ۴ نقاط به شخص نمایش داده می‌شود تا آنها را با حرکت چشم و بدون حرکت سر دنبال کند. با استفاده از داده های جمع شده در این بخش مسئله ی حداقل مربعات مانده ها برای پیدا کردن بهترین ضرایب برای توصیف بهترین تابع بین پارامترها و نقطه ی نگاه حل می‌شود.

در انتها از شخص خواسته می‌شود تا چشمش را ببندد تا آستانه ی بسته بودن چشم محاسبه شود. این آستانه طوری تنظیم می‌شود که فاصله اش تا بسته ترین حالت چشم در مراحل قبل ۴ برابر فاصله اش تا حالت بسته ی چشم باشد.

۱۰-۳- ارزیابی

ارزیابی دقت عملکرد سیستم در دو حالت استاتیک و دینامیک می‌تواند انجام شود. در حالت استاتیک یک شبکه از نقاط روی صفحه نمایش داده می‌شود و هنگام دیدن هر کدام از نقاط روی صفحه توسط شخص، میانگین فاصله ی نقطه ی نگاه محاسبه شده از آن نقطه به عنوان خطا حساب می‌شود. با انجام این کار برای همه ی نقاط می‌توان خطا را در نقاط مختلف صفحه به دست آورد و آن را به شکل یک رویه رسم نمود.

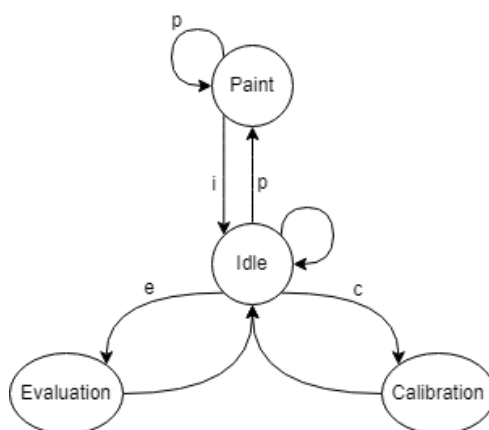
در ارزیابی دینامیک، یک نقطه روی صفحه طبق مسیری که به آن داده می‌شود حرکت می‌کند و شخص باید با چشم آن را دنبال کند. در نهایت مسیر حرکت جسم و مسیر حرکت نقطه ی نگاه در کنار هم بررسی می‌شوند (می‌توان عواملی مانند تاخیر و فاصله ی نقطه ی محاسبه شده

به عنوان نقطه ی نگاه تا مکان نقطه ی متحرک روی صفحه که به آن نگاه شده را به عنوان معیار در نظر گرفت).

ایراد این شیوه ی ارزیابی این است که ممکن است شخص دقیقاً به نقطه ای که برای او تعیین شده نگاه نکند. برای ارزیابی دقیق تر باید از ابزارهای دیگر تعقیب چشم که دقت مشخصی دارند (مشابه مواردی که در فصل ۲: بیان شد) استفاده شود.

۱۱-۳- ساختار کلی اجرای برنامه ی نوشته شده

برنامه ی نوشته شده در چهار حالت آماده به کار^۱، کالیبراسیون، ارزیابی و رسم اجرا می شود. از حالت آماده به کار با زدن کلید مربوط روی صفحه کلید می توان به حالت مورد نظر رفت.



شکل (۳-۳۱) : ماشین حالت برنامه ی نوشته شده

در حالت رسم مسیر حرکت نگاه روی صفحه باقی می ماند و می توان با نگاه در آن به رسم اشکال مختلف پرداخت!

^۱ idle

فصل ۴: جمع بندی، نتیجه گیری و

پیشنهادهای

در این پروژه به پیاده سازی یک سیستم تعقیب کننده ی نگاه پرداخته شد که با استفاده از یک دوربین، نقطه ی نگاه شخص را تخمین می زند. موارد بیان شده در فصل های قبل همگی با استفاده از تصاویر 640×480 به دست آمده بودند، استفاده از تصویر با کیفیت تر و دوربین سریع تر و سیستم قوی تر می تواند باعث بهبود کیفیت و سرعت تعقیب نگاه شود.

علت اصلی نویز موجود در سیستم، لرزش نقاط تخمین زده شده توسط mediapipe روی صورت می باشد. علاوه بر فیلتری که برای کاهش نویز اعمال شد، افزایش کیفیت تصویر و بهبود نور محیط می تواند باعث بهبود تصویر صورت شود و به این صورت، mediapipe با خطای کمتری می تواند صورت را شناسایی، و نقاط را روی آن جایگذاری کند. به علاوه، اعمال فیلتر روی مکان نقاط روی صورت نیز می تواند باعث کاهش نویز شود، زیرا با این کار لرزش نقاط کاهش پیدا می کند و نویز حاصل از این لرزش ها در محاسبات روی هم تجمیع نمی شوند.

موارد بسیاری برای بهبود این سیستم تخمین نگاه وجود دارد. محاسبه ی مکان عنیه می تواند در مختصات قطبی انجام شود، به این صورت مکان عنیه نسبت به مرکز چشم در مختصات کروی با رابطه ی ساده تری به دست می آید و با توجه به زاویه های به دست آمده در این مختصات، با رابطه ای دقیق تر و ساده تر می توان نقطه ی نگاه را محاسبه کرد. با بررسی دقیق تر آناتومی چشم، می توان روابط مدلسازی شده را به دنیای واقعی نزدیک تر کرد. همچنین، دقیق تر کردن مدل ساختار دوربین در محاسبات نیز می تواند باعث بهبود مدل محاسباتی شود. به جای محاسبه ی موقعیت و زاویه ی سر، با توجه به نقاط سه بعدی تولید شده توسط mediapipe می توان موقعیت هر چشم را در فضا محاسبه کرد و با در نظر گرفتن زاویه ی هر چشم، نقطه ی نگاه هر چشم را به دست آورد و نقطه ی بین این دو نقطه ی به دست آمده را به عنوان نقطه ی نگاه معرفی کرد.

مراجع

- [1] H. Singh and J. Singh, "Human Eye Tracking and Related Issues: A Review," *International Journal of Scientific and Research Publications*, vol. 2, no. 9, 2012.
- [2] B. T. Carter and S. G. Luke, "Best practices in eye tracking research," *International Journal of Psychophysiology*, 2020.
- [3] "Electrooculography," [Online]. Available: <https://en.wikipedia.org/wiki/Electrooculography>.
- [4] D. J. Creel, "The electrooculogram," in *Handbook of Clinical Neurology*, 2019, pp. 495-499.
- [5] ا. ا. گلو و س. راحتی قوچانی, "آشکارسازی جهت حرکت چشم از روی سیگنال EOG بوسیله شبکه های عصبی مصنوعی," در چهاردهمین کنفرانس ملی سالانه انجمن کامپیوتر ایران, تهران, ۱۳۷۸.
- [6] M.-L. Lai, M.-J. Tsai, F.-Y. Yang, C.-Y. Hsu, T.-C. Liu, S. W.-Y. Lee, M.-H. Lee, G.-L. Chiou, J.-C. Liang and C.-C. Tsai, "A review of using eye-tracking technology in exploring learning," *Educational Research Review*, vol. 10, pp. 90-115, 2013.
- [7] "Does performance differ between Python or C++ coding of OpenCV?," [Online]. Available: <https://stackoverflow.com/questions/13432800/does-performance-differ-between-python-or-c-coding-of-opencv>.
- [8] D. R. Carvalho, W. D. Fonseca, J. Hollebon, P. H. Mareze and F. M. Fazi, "Head tracker using webcam for auralization," in *inter noise*, Washington, DC, 2021.
- [9] P. Viola and M. Jones, "Rapid Object Detection using a Boosted Cascade of Simple Features," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Kauai, 2001.
- [10] "Cascade Classifier," [Online]. Available: https://docs.opencv.org/3.4/db/d28/tutorial_cascade_classifier.html.
- [11] A. Mittal, "Haar Cascades, Explained," 21 December 2020. [Online]. Available: <https://medium.com/analytics-vidhya/haar-cascades-explained-38210e57970d>.
- [12] S. Soo, "Object detection using Haar-cascade".
- [13] L. Cuimei, Q. Zhiliang, J. Nan and W. Jianhua, "Human face detection algorithm via Haar cascade classifier combined with three," in *2017 IEEE 13th International Conference on Electronic Measurement & Instruments*, 2017.
- [14] "LEARN OPENCV C++ in 4 HOURS | Including 3x Projects | Computer Vision," [Online]. Available: <https://www.youtube.com/watch?v=2FYm3GOonhk>.
- [15] "Dlib C++ Library," [Online]. Available: <http://dlib.net/>.
- [16] "Face detector," [Online]. Available: http://dlib.net/face_detector.py.html.
- [17] "Face landmark detection," [Online]. Available: http://dlib.net/face_landmark_detection.py.html.
- [18] "Mediapipe," Google, [Online]. Available: <https://google.github.io/mediapipe/>.

-
- [19] "Mediapipe Face Mesh," Google, [Online]. Available: https://google.github.io/mediapipe/solutions/face_mesh.html.
- [20] A. Vakunov and D. Lagun, "MediaPipe Iris: Real-time Iris Tracking & Depth Estimation," Google, 6 August 2020. [Online]. Available: <https://ai.googleblog.com/2020/08/mediapipe-iris-real-time-iris-tracking.html>.
- [21] Pysource, "Gaze controlled keyboard with Opencv and Python," [Online]. Available: <https://www.youtube.com/playlist?list=PL6Yc5OUgcoTlvHb5OfFLUJ90ofBuoU5g8>.
- [22] AiPhile, "Eyes Tracking Mediapipe," [Online]. Available: <https://www.youtube.com/playlist?list=PLJ958Ls6nowUwRXHUcFwZy2CT0naMULR3>.
- [23] "Eye Blink Counter using OpenCV Python | Computer Vision," [Online]. Available: <https://www.youtube.com/watch?v=-TVUwH1PgBs>.
- [24] AiPhile, "Eyes Tracking Opencv python," [Online]. Available: <https://www.youtube.com/watch?v=obKG1SXp76Y>.
- [25] Pysource, "Eye motion tracking - Opencv with Python," [Online]. Available: <https://www.youtube.com/watch?v=kdbdZFT9NQI>.
- [26] AiPhile, "Iris Position Estimation Python part 2| MediaPipe | Opencv 2022," [Online]. Available: <https://www.youtube.com/watch?v=8CIxfcbGU3s>.

پیوست الف: آدرس مخازن و فایل ها

- مخزن^۱ github پروژه:

https://github.com/MHKashaniN/gaze_tracker

- مخزن مدل های آموزش دیده ی haarcascade:

<https://github.com/opencv/opencv/tree/master/data/haarcascades>

- مدل تشخیص الگوی ۶۸ نقطه ای صورت برای استفاده در تشخیص چهره ی کتابخانه ی dlib:

http://dlib.net/files/shape_predictor_68_face_landmarks.dat.bz2

- مخزن github کد تشخیص جهت و مکان سر [8]:

<https://github.com/eac-ufsm/internoise2021-headtracker>

¹ Repository

Abstract:

Eye tracking means measuring movements and position of the eyes to calculate the gaze point. Meanwhile gaze tracking examines both eyes and head movements simultaneously to obtain the point of gaze. Many methods in academic and industrial projects have been in use for observing and calculating eye movements. In this project a gaze tracker system is made using a camera. This system is usable for tracking patients' gaze synchronous with that of their hand movements recorded by motor control and computational neuroscience lab's robot. To estimate the point of gaze, the face and eyes are being located first using mediapipe library and python, followed by locating the pupils and head position and orientation used for geometrical calculations to calculate the point of gaze. This system can achieve real-time calculation of gaze point and about 3cm accuracy for a system with 640×480 picture resolution and 40cm distance from camera, which can be improved by using high resolution camera and more powerful system for running the program.

Keywords: Eye tracking, Gaze tracking, Mediapipe, Image processing, Webcam, Python language, Facial landmarking



University of Tehran



College of Engineering

School of Electrical and Computer Engineering

Design and Develop of a gaze tracking system

A thesis submitted to the Undergraduate Studies Office

In partial fulfillment of the requirements for

The degree of Electrical Engineering

By:

MohammadHossein Kashaninia

Std. ID:

810197560

Supervisor:

Dr. Fariba Bahrami