پردیس علوم
دانشکده ریاضی، آمار و علوم کامپیوتر

پیاده سازی نرم افزاری دوربین های با سنسور دید دینامیک

نگارنده

نیلوفر هوشیاری پور

استاد راهنما: دکتر محمد گنج تابش

پایان‌نامه برای دریافت درجه کارشناسی
در رشته علوم کامپیوتر

تیر ماه ۱۳۹۶

چکیده

در این پروژه هدف، پیاده سازی نرم افزاری دوربین های با سنسور دید دینامیک [1] است. مشخصه ی مهم و متفاوت این دوربین ها از سایر دوربین ها ، مصرف حافظه ی کمتر و در نتیجه زمان کمتر برای نمایش تصویر است . در اصل ایده ی اصلی این دوربین ها از نحوه ی کارکرد سیستم بینایی انسان گرفته شده است . این دوربین ها همانند چشم انسان حساس به تغییرات هستند و مانند سایر دوربین ها همه ی فریم ها را ذخیره نمی کنند . در چشم انسان در بخش شبکیه چند دسته سلول وجود دارد که وقتی تصویری از محیط توسط چشم دیده می شود ، نور هایی از محیط که به چشم میرسد در شبکیه توسط این سلول ها جذب می شود و هر سلول بسته به وظیفه اش (یکسری حساس به نور ، یکسری حساس به مرکز و ...) اطلاعاتی از محیط بدست می آورند ، در نهایت این اطلاعات به هسته های خمیده ی جانبی [2] می روند و از آنجا به بخش بینایی مغز می روند .

این دوربین ها همانند مرحله ی اولیه ی پردازش تصویر در چشم انسان به لبه ها حساس اند و در نمایش تصویر گرفته شده از این دوربین ها لبه های هر شی به صورت واضح دیده می شود . از کاربرد های این دوربین ها می توان به تعقیب اشیا (برای مثال خواندن پلاک ماشین در حال حرکت و ...) و بررسی حرکت سریع ذرات اشاره کرد .

[1] dynamic vision sensor cameras

[2] lateral geniculate nucleus

فهرست مطالب

فصل ۱

ساختار مغز و چشم

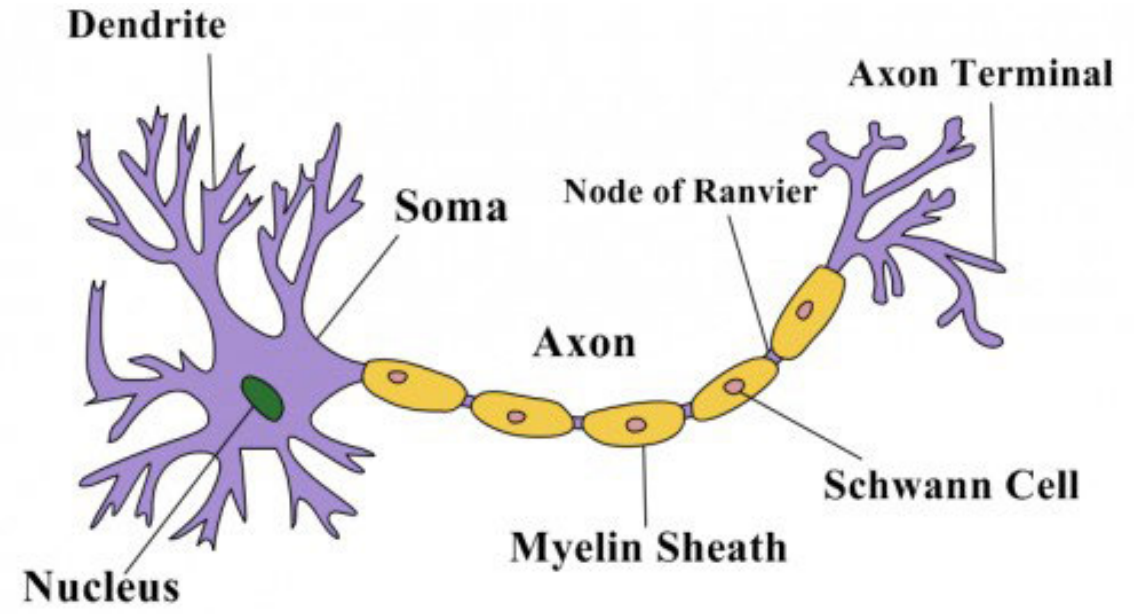
در این بخش به توضیح مختصر ساختار قسمت های مختلف مغز و عملکرد آن ها می پردازیم، سپس به شرح چشم و سلول های دریافت کننده ی اطلاعات می پردازیم . در انتها به سراغ قشر بینایی مغز انسان می‌رویم و مسیر پردازش اطلاعات از چشم تا لایه های بالایی مغز را بررسی می‌کنیم.

۱.۱ ساختار مغز

مغز ، نخاع و اعصاب محیطی از سلول های عصبی میکروسکوپی به نام نورون(شکل۱.۱) ساخته شده‌اند که به صورت یک شبکه ی بسیار پیچیده با هم در ارتباط هستند و پیام های عصبی را به صورت امواج الکتریکی منتقل می‌کنند [۱].

هر نورون از سه قسمت اصلی تشکیل شده است . تنه ، داندریت و آکسون .نخستین قسمت تنه است که حاوی هسته و سایر ساختارهایی است که معمولا در سلول ها یافت می‌شوند. بدنه ی این قسمت سوما نام دارد . دومین قسمت نورون از زواید کوتاه و چند شاخه ای تشکیل می‌شود که از تنه ی سلول بیرون زده‌اند و داندریت خوانده می‌شوند .سومین قسمت ، زایده ی دراز و باریکی است که آکسون نام دارد . در این قسمت آکسون پیام‌های الکتریکی عصب را در طول مسیر خود انتقال می‌دهد و هر نورون را به نورون دیگر یا به یکی از ماهیچه‌ها متصل می‌کند .

سلول های عصبی پیام خود را از طریق سیناپس ، که یک ساختار زیستی در پایانه‌ی آکسون است ، به داندریت یک نورون دیگر یا سلول ماهیچه ای یا یک غده می‌فرستد و در سیناپس نورون ها به یکدیگر یا به اندام های بدن متصل نمی‌شوند ، بلکه در فاصله ی بسیار کمی از هم قرار می‌گیرند و پیام های الکتریکی از طریق آزاد شدن مواد شیمیایی از یک نورون به نورون بعدی منتقل می‌شوند تا در نهایت پیام ها مسیر مورد انتظار را در سیستم عصبی طی کنند .در حالت عادی پتانسیل الکتریکی درون نورون بین -40 و-90 میلی ولت است که به آن پتانسیل استراحت می‌گویند . وقتی نورون تحریک می‌شود ، پتانسیل الکتریکی داخل نورون بسته به محرک می‌تواند بالاتر یا پایین تر برود . اگر این پتانسیل به اندازه ی کافی منفی تر شود (مثلا اگر به -200 برسد نورون اسپایک می زند .) به



شکل ۱.۱: سلول نورون

سطحی به نام آستانه ی تحریک می‌رسد که باعث فعال شدن پتانسیل عمل می‌شود .در این هنگام با سرازیر شدن یون های سدیم به ذرون سلول پتانسیل الکتریکی درون نورون ناگهان به 20تا 50 میلی ولت می رسد. پتانسیل عمل تنها چند میلی‌ثانیه درام دارد و پس از آن با خروج یون های پتاسیم از سلول، بار الکتریکی آن دوباره منفی شده و سلول دوباره به حالت استراحت برمی‌گردد. به این ترتیب اطلاعات در طول نورون به شکل تکانه ی شیمیایی - الکتریکی حرکت می‌کند و با سرعتی معادل 3 تا 300 کیلومتر در ساعت از مناطق داندریتی به سوی انتهای آکسون می‌رود .

دستگاه عصبی با استفاده از نورون ها هدایت تحریکات و پیام های عصبی در بدن را به عهده دارد . اطلاعات مورد نیاز از اندام های مختلف بدن را حس کرده و به مغز می‌فرستد و در مقابل فرمان های مغزی را به اندام های بدن منتقل می‌کند .

دستگاه عصبی مرکزی قسمتی از دستگاه عصبی است که در درون محفظه ای استخوانی به نام استخوان جمجمه و ستون فقرات قرار گرفته است و شامل مغز و نخاع می‌باشد . دستگاه عصبی مرکزی به همراه دستگاه عصبی محیطی، نقش بنیادینی در کنترل رفتار انسان دارد . دستگاه عصبی مرکزی پردازش اطلاعات و محاسبه ی حرکت مناسب در پاسخ به دریافت ورودی را بر عهده دارد . بزرگترین بخش مغز مخ نام دارد و متخصصان هر قسمت از آن را بسته به وظایفی که انجام می‌دهند تقسیم بندی کرده‌اند : لوب پس سری — لوب گیج گاهی — لوب جداری — لوب پیشانی .

از بین این بخش ها لوب پس سری یا قشر بینایی ، مرکز پردازش اطلاعات دیداری در مغز است و آسیب رسیدن به این ناحیه منجر به اختلالات بینایی و یا نابینایی کامل می‌شود .

۲.۱ ساختار دستگاه بینایی

چشم انسان عضو اصلی ایجاد تصویر از دنیای واقعی برای مغز هستند . توانایی انسان در پردازش داده های حسی و تشخیص الگو موضوعی است که توجه محققان عصب شناسی ،هوش مصنوعی و رباتیک را به خود جلب کرده است . انسان در تشخیص چهره ها ، اشیا و صدا ها بلادرنگ عمل می کند و صحت این تشخیص تقریبا از تمام برنامه های رایانه ای موجود بیشتر است . این خصوصیت سبب شده است که مغز انسان به عنوان مدلی عالی برای تشخیص الگو مطرح شود .

۱.۲.۱ درک تصویر در شبکیه

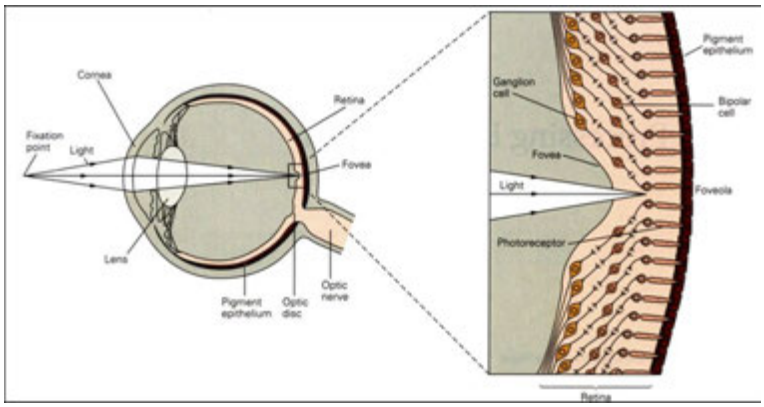
ادراک بینایی از شبکیه آغاز می شود و در دو مرحله صورت می‌گیرد. نوری که وارد قرنیه می‌شود طرحی را در پشت چشم ایجاد می‌کند و در آنجا سیگنال های نوری توسط اندام تخصص یافته ی حسی به سیگنال های الکتریکی تبدیل می‌شوند. سپس این سیگنال ها از طریق عصب بینایی به نواحی بالاتر در مغز فرستاده می‌شوند تا پردازش بیشتر مورد نیاز برای ادراک بینایی در آن نواحی صورت گیرد .

شبکیه به دلایل متعدد مورد بررسی های دقیق قرار می‌گیرد . نخست اینکه به طور کلی برای درک انتقال حسی مفید است و گیرنده های نوری در شبکیه چه بسا شناخته شده ترین سلول های حسی هستند . دوم اینکه ، بر خلاف سایر ساختارهای حسی ، مثل حلزون گوش یا گیرنده های حسی- پیکری در پوست ، شبکیه یک عضو محیطی نیست ، بلکه بخشی از سیستم عصبی مرکزی محسوب می‌شود ، و سازماندهی پیام رسانهای شیمیایی آن به دیگر ساختارهای عصبی مرکزی شباهت دارد . در عین حال ، شبکیه در مقایسه با نواحی دیگر مغز ساختمان ساده تری دارد .

چشم ها به گونه ای طراحی شده‌اند که تصویر را با کمترین خطا بر روی شبکیه متمرکز کنند . نور توسط عدسی و قرنیه متمرکز می‌شود و سپس قبل از رسیدن به گیرنده های نوری در شبکیه از مایع زجاجیه که حفره ی چشم را پر کرده است ، عبور می‌کند .

شبکیه در جلوی بافت پوششی رنگدانه دار که پشت چشم را مفروش می‌کند ، قرار گرفته است . سلول ها در بافت پوششی رنگدانه دار که پر از رنگدانه های سیاه هستند ، هر نوری را که به شبکیه برخورد نکرده است را جذب می‌کنند . علاوه بر این در ناحیه ی دید محیطی (فووا) از شبکیه ، جسم سلولی نورون های نخستین شبکیه به اطراف تغییر جهت داده‌اند و این مساله به گیرنده های نوری این توانایی را می‌بخشد که تصویرهای بینایی را با کمترین تغییر شکل دریافت کنند . این تغییر جهت در مرکز فووا به بیشترین مقدار خود می‌رسد .

شبکیه شامل ناحیه ی دیگری نیز هست که صفحه ی بینایی نامیده می‌شود و اعصاب بینایی ، شبکیه را از این نقطه ترک می‌کنند .صفحه ی بینایی منطقه‌ای است که هیچ گونه گیرنده ی نوری ندارد و بنابراین نقطه ی کور میدان بینایی محسوب می‌شود .



شکل ۲.۱: تصویری از شبکیه ی چشم

-سلول های دریافت کننده ی نور

دو نوع گیرنده‌ی نوری در شبکیه وجود دارد : استوانه ای و مخروطی . سلول های مخروطی توانایی دید در روز را به انسان می دهند و سلول های استوانه ای بینایی در شب را ایجاد می‌کنند .سلول های مخروطی به طور کلی نقش مهم تری در انجام وظایف بینایی داشته و بهتر از سلول های استوانه ای عمل می‌کنند(به جز شناسایی تحریکات نور ضعیف). دقت بینایی منتقل شده توسط سلول های مخروطی از دقت بینایی که توسط سلول‌های استوانه ای منتقل می شود بیشتر است و سلول‌های مخروطی تفکیک بهتری از تغییرات سریع تصویر فراهم می‌کنند .این سلول ها دید رنگی را نیز منتقل می‌کنند .سیستم سلول‌های استوانه ای در برابر نور حساسیت بیشتری از سیستم مخروطی دارد، اما این سیستم فاقد رنگ است .

- سلول های جمع آوری کننده ی اطلاعات دریافتی چشم(گانگلیون)

خروجی شبکیه از طریق سلول‌های گانگلیون انتقال می‌یابد . بین گیرنده‌های نوری و سلول‌های گانگلیونی سه گروه از نورون های بینابینی قرار دارند که عبارت اند از:

سلول‌های دو قطبی ، سلول‌های افقی ،سلول‌های آماکرین .این سلول‌ها سیگنال های حاصل از چندین گیرنده ی نوری را چنان با هم ترکیب می‌کنند که پاسخ‌های الکتریکی برخاسته از سلول های گانگلیونی به طور دقیق به طرح های فضایی و زمانی نوری ، که شبکیه را تحریک می‌کنند ، وابسته شوند .در اکثر سلول‌های گانگلیونی ،نواحی گیرنده ای به دو بخش تقسیم می‌شود :

ناحیه ی حلقوی مرکزی که مرکز ناحیه ی گیرنده نام دارد و مناطق باقی مانده که محیط ناحیه ی گیرنده نامیده می‌شود.بر اساس پاسخ های سلول های گانگلیونی به نقطه ی نورانی کوچک که بر روی مرکز ناحیه ی گیرنده ای انداخته می‌شود ، می‌توان دو دسته از سلول‌های گانگلیونی را شناسایی کرد :سلول‌های میان روشن و سلول‌های میان خاموش .

سلول های گانگلیونی میان روشن هنگامی که نور به مرکز ناحیه‌ی گیرنده ای آنها تابانده می شود ،تحریک می‌شوند ، در این سلول ها تابش نور به محیط ، سلول را مهار می کند . این روند در سلول های میان خاموش برعکس است ، این سلول ها به وسیله ی نوری که به مرکز ناحیه‌ی گیرنده ای آن ها تابانده می شود، مهار می‌شوند . یعنی این سلول‌ها زمانی تحریک می شوند که نقطه ی نوری موجود در مرکز آن ها خاموش شود . همچنین نور زمانی سلول میان خاموش را تحریک می کند که مستقیما به محیط ناحیه ی گیرنده تابانده شود .

همه ی سلول های گانگلیونی دارای ناحیه ی گیرنده ، با سازماندهی مرکزی-محیطی نیستند . به عنوان مثال ، شماری از سلول ها گانگلیونی به تغییرات روشنایی کلی در میدان بینایی پاسخ می دهند و در کنترل بازتابش های مردمک بسیار مهم هستند.

سلول‌های گانگلیونی میان روشن و میان خاموش دو روش موازی برای پردازش اطلاعات بینایی فراهم می‌کنند . به علاوه اندازه ی ناحیه ی گیرنده ی آن‌ها در عرض شبکیه متفاوت است . در ناحیه ی فووا و در محلی که دقت بینایی در بالاترین میزان است ، نواحی گیرنده ای کوچک هستند و در نواحی ای که دقت کم است ، نواحی گیرنده بزرگ هستند .

فصل ۲

نحوه ی پردازش اطلاعات در چشم

در این بخش به بررسی مختصر چگونگی انتقال اطلاعات بینایی از طریق سلول های شبکیه ابتدا به بخش هسته های خمیده ی جانبی و سپس به بخش بینایی در مغز می‌پردازیم .

۱.۲ ارسال اطلاعات از طریق شبکه ای از نورون های بینابینی

هر دسته از نورون های موضعی شبکیه (افقی ، دو قطبی و آماکرین) نقش خاصی را در شکل دادن به سیگنال های گیرنده ی نوری که از طریق شبکیه منتقل می شوند ، ایفا می‌کنند . نقش نورون های بینابینی شبکیه را می توان به بهترین صورت از طریق تمرکز بر سلول های دوقطبی توضیح داد، چون این سلول ها مستقیم ترین مسیر را در بین گیرنده ها و سلول های گانگلیونی ایجاد می کنند . سلول های دو قطبی ، به صورت مستقیم و غیر مستقیم، سیگنال ها را از سلول های دریافت کننده ی نور به سلول های گانگلیون انتقال می‌دهند .از این رو دو قطبی نامیده می شوند که می توانند با سلول های مخروطی و استوانه ای(ولی نه هر دو) اتصال برقرار کنند .

آن ها همچنین از سلول های افقی نیز ورودی می‌پذیرند . سلول های افقی به تنظیم و یکپارچه سازی ورودی خود که از چندین سلول دریافت کننده ی نور به آن ها می رسد ، می پردازد . سلول های افقی با کارکرد خور مسوولیت عملکرد مناسب در نور زیاد و کم را به عهده دارند .

سلول های آماکرین نیز نورون های مهاری هستند که از طریق داندریت های خود با سلول های گانگلیون و یا سلول های دو قطبی ارتباط برقرار می‌کنند .

۲.۲ هسته های خمیده ی جانبی

خروجی سلول های گانگلیون وارد تالاموس و هسته ی مرکزی آن می‌شوند. یکی از کارکرد های تالاموس هماهنگی بین گیرنده های حسی و قشر مغز می‌باشد . هسته های تالامیک موجود در این

شکل ۱.۲: خروجی پردازش سلول های گانگلیون در شبکیه[۱]

ناحیه همانند یک رله داده ها را به نواحی مختلف مغز ارسال می‌کنند .اطلاعات ارسال شده از سلول های گانگلیون توسط این رله های هوشمند به قشر اولیه ی بینایی که در لوب پس سری وجود دارد ارسال می‌شود .

۳.۲ مسیر بینایی

میدان دید (شکل۲.۲) از دو قسمت تشکیل شده است :

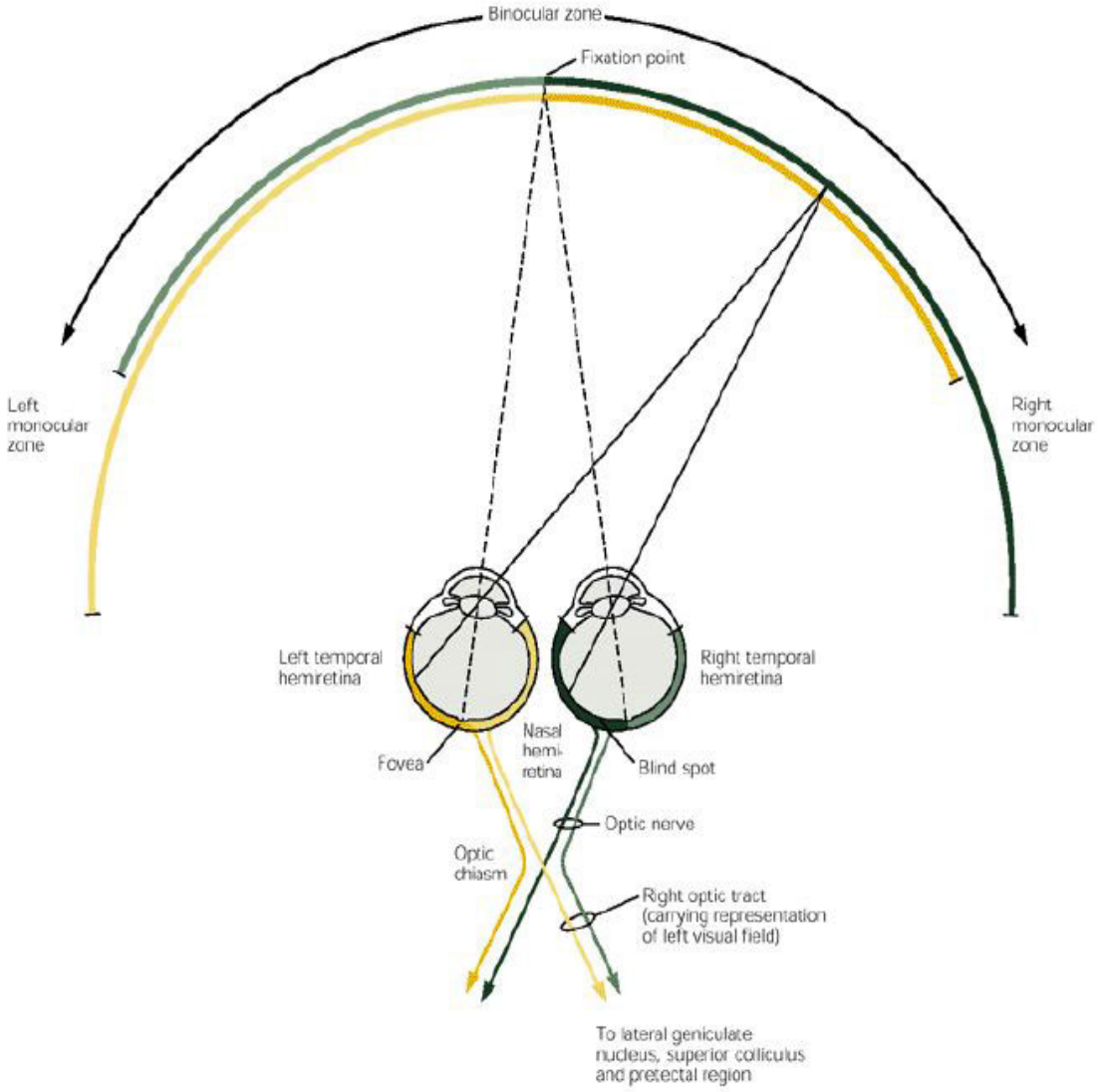
۱.منطقه ی دید دو چشمی : منطقه ای که نور منتشره از آن به هر دو چشم می‌رسد .

۲.منطقه ی دید یک چشمی : قسمت هایی از اطراف میدان بینایی که نور از آنها فقط به یک چشم می‌رسد .

شبکیه نیز دارای دو قسمت است : جانب بینی[۱] و جانب گیجگاهی[۲] . وقتی جسمی در منطقه ی یک چشمی راست قرار می گیرد شعاع نور از شی به نیمه ی نازال شبکیه ی چشم راست می‌رسد .اگر شی در منطقه‌ی دید دو چشمی و در طرف راست خط تقارن بدن قرار بگیرد شعاع نور از چشم به نیمه ی نازال شبکیه چشم راست و منطقه ی نیمه تمپورال چشم می‌رسد .

ورودی هایی که به نیمه ی نازال شبکیه می‌رسند ،پس از تقاطع به مرکز پردازش داده های حسی(تالاموس) سمت مقابل و سپس به قشر مغز طرف مقابل می‌روند ، اما ورودی های رسیده بهه منطقه ی نیمه تمپورال بدون تقاطع به تالاموس و سپس به قشر مغز طرف خود می‌روند .چشم ها به دو نیمکره ی مغز اطلاعات می‌فرستند . سازمان بندی بینایی بر اساس چشم راست و چپ نیست ، بلکه بر اساس میدان بینایی است که دو نیمه ی چپ و راست دارد که هر نیمه میدان بینایی در قشر سمت مخالف خود پردازش می‌شود .

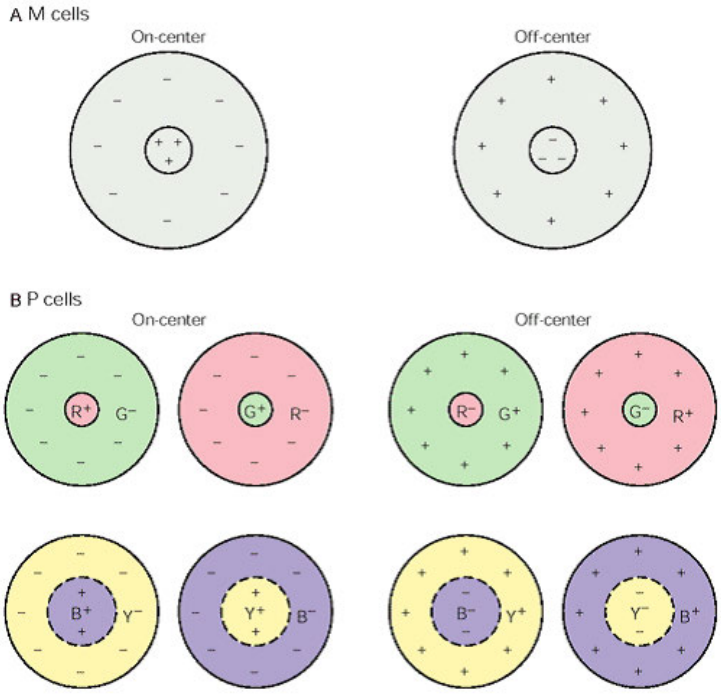
[۱] Nasal
[۲] Temporal



شکل ۲.۲: دو منطقه ی دید چشمی میدان بینایی

۴.۲ انتقال سیگنال از چشم به LGN

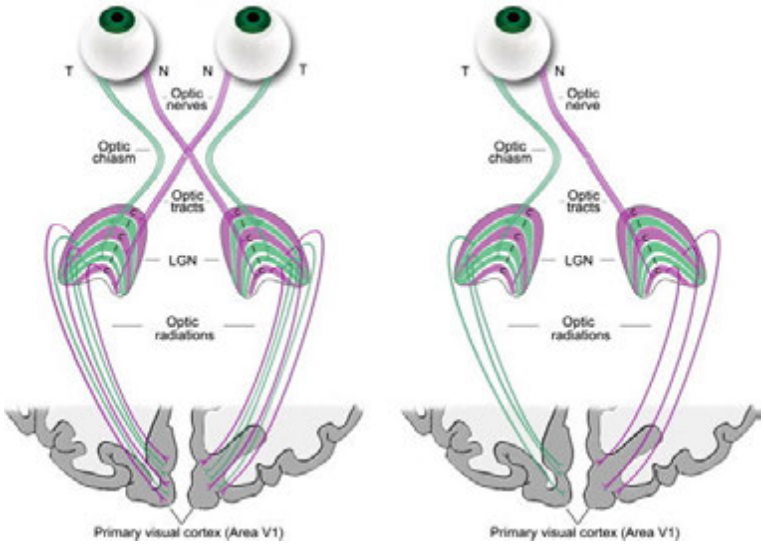
حدود 20 نوع سلول گانگلیونی شناسایی شده که کار های مختلف را انجام داده و هر کدام یک بعد و جنبه ی متفاوتی از اطلاعات بینایی را منتقل می‌کنند [۱]. دو نوع از مهم‌ترین این سلول‌ها که مورد بررسی قرار می گیرند سلول های نوع M از magno به معنی اندازه بزرگ و سلول های نوع P از parvo به معنی کوچک هستند(شکل۳.۲). مشاهده خواهد شد که در هسته های تالاموسی که مخصوص دریافت اطلاعات از شبکیه هستند نیز لایه های M و P را داریم. سلول های نوع P از نظر اندازه کوچک هستند و اطلاعات خود را به لایه های parvo هسته تالاموسی (LGN) میفرستند. M cell ها نیز سلول های بزرگی هستند و این سلول ها اطلاعات خود را به لایه ی magnocellular هسته ی تالاموسی ویژه ی بینایی می‌فرستند.



شکل ۳.۲: Mcell ، Pcell

این مسیر تا کورتکس حفظ می شود یعنی همواره یک مسیر سریع و جدید وجود دارد و یک مسیر کند و قدیمی که این مسیر تا کورتکس حفظ می شود.یکی از تفاوت های سلول های نوع P و سلول های نوع M در ناحیه ی دریافت اطلاعات آن هاست.سلول های نوع P اندازه ی کوچکتری

دارند بنابراین میدان دید [۳]کوچکتری نیز دارند؛ سرعت انتقال اطالعات در آکسون های سلول های P نسبت به سلول های M خیلی آهسته تر است؛ سلول های P به طور ویژه به محرک های رنگی پاسخ می دهند و همچنین اطلاعاتی که می فرستند حالت پایدار و پیوسته دارد در حالی که سلول های M بیشتر ناپایدار هستند؛ سلول های M نسبت به سلول های P بیشتر حساس هستند و بنابراین میتوانند در انتقال اطلاعات سیاه و سفید یا اطلاعات در نور کم یا اطلاعاتی که خیلی دقیق نیستند دخیل باشند.



شکل ۴.۲: قشر بصری اولیه

هسته ی خمیده ی جانبی به عنوان دروازه ای برای پیام های بینایی عمل می کند. این هسته(شکل۴.۲) به زیر لایه های متعدد تقسیم شده که تفاوت اطلاعات دریافتی از شبکیه را حفظ می کند ، به شکلی که اطلاعات به صورت دقیق و نقطه به نقطه به زیر لایه های این هسته منتقل می شوند. بنابراین تفاوتی که میان اطلاعات در ورود به شبکیه ، سیستم انتقالی آنها (مسیر سریع - مسیر کند) وجود داشت در سطح این هسته نیز حفظ می شود . این هسته دارای ۶ لایه است ، لایه های ۲،۳،۵ در هر سمت اطلاعات مربوط به نیمه ی تمپورال همان سمت را دریافت می کند و لایه های ۱،۴،۶ در هر سمت اطلاعات نیمه ی نازال شبکیه در سمت مقابل را دریافت می کند .

۵.۲ قشر بینایی مغز

غشای بینایی به نواحی مختلفی تقسیم شده است . V۱ و V۲ بزرگترین ناحیه ها هستند [۲]. اطلاعات رسیده به V۱ از تالاموس در دو مسیر جداگانه و به طور مستقل پردازش می گردند :

[۳]Receptive Field

۱.مسیر تشخیص چیستی شی : کار این مسیر ، شناسایی ویژگی های اشیا (رنگ ، شکل و ...) است .
۲.مسیر تشخیص مکان شی : کار این مسیر ، شناسایی مشخصات فضایی صحنه (جهت حرکت صحنه و ...) است .

۱.۵.۲ ناحیه ی V۱

شبکیه تصویر بینایی را به پیکسل های متفاوت درخشندگی تجزیه می کند . نورون های قشر بینایی باید اطلاعات تجزیه شده را بازسازی کنند . اولین قدم بازسازی در قشر V۱ است . سلول های این ناحیه به خطوط یا لبه های محرک پاسخ می دهند . یعنی هر سلول به جهت خاصی از محرک و در محل خاصی از میدان بینایی پاسخ می دهد . به این سلول ها ، سلول های گزینش پذیر به جهت می گویند .

پس برای بازسازی تصویر ، اولین قدم این است که سیستم ، لبه های متفاوت جسم در زوایای مختلف را شناسایی کند . به همین دلیل است که سیستم بینایی ما یک سیستم شناساگر لبه است . یعنی شناسایی جسم بر اساس لبه های آن جسم است و این اولین قسمت مراحل شناسایی یک جسم است . دو نوع سلول در قشر V۱ شامل نورون های ساده و پیچیده کار شناسایی جهت لبه های محرک را ایفا می کنند .

۱.سلول های ساده : این سلول ها قادر به تشخیص خط در یک زاویه ، اندازه و محل مشخص هستند . در اصل این سلول ها تنها به یک باریکه ی نور در یک جهت خاص پاسخ می دهند . مثلا اگر در میدان دریافت کنندگی یک تک سلول ساده که به خط ۹۰ درجه پاسخ می دهد ، خطی ۰ درجه قرار بگیرد ، سلول به هیچ وجه فعال نخواهد شد .

۲.سلول های پیچیده :این سلول ها تنها حساس به زاویه اند و مانند سلول های ساده حساس به اندازه و محل محرک نیستند . ورودی این سلول ها از ترکیب خروجی های چند سلول ساده در یک زاویه ی ثابت اما در اندازه و محل های مختلف بوجود می‌آید .

هر نورون در قشر بینایی دارای ناحیه ی دریافت است.ناحیه ی دریافت معادل قسمتی از تصویر ورودی است که تغییر در خصوصیات آن، فعالیت نورون را تحت تاثیر قرار می‌دهد. در ناحیه ی نورون،V۱ ها دارای ناحیه ی دریافت کوچکی هستند که اندازه‌ی آن در نقطه‌ی مرکزی دید حدود یک درجه می باشد. ناحیه‌ی دریافت در تصویر محلی می باشد و هر نورون قسمت خاصی را از طریق ناحیه‌ی دریافت خود مشاهده می‌کند.این سلول ها محرک های یک نقطه از شبکیه را در ناحیه ی دریافت خود دریافت می کنند.همچنین سلول های ساده، مناطق وادارنده و بازدارنده در ناحیه ی دریافت خود دارند که این مناطق بزرگتر از مناطق سلول های گانگلیون می‌باشد. نحوه ی قرار گرفتن این نورون ها در قشر بینایی اولیه بدین صورت است که نورون های با جهت مشابه در کنار یکدیگر قرار گرفته‌اند.

۲.۵.۲ ناحیه ی V۲ و V۴

دومین ناحیه از سلسله مراتب پردازش ،ناحیه‌ی V۲ می‌باشد. نورون های این ناحیه، ورودی های خود را از نورون های ناحیه‌ی V۱ دریافت می‌کنند.مسیر های رنگ،شکل و حرکت از V۱ در V۲ ادامه پیدا می کند.نورون های مسیر شکل درV۲ نسبت به جهت انتخابی هستند و نیمی از آن ها به انتهای لبه ها و خطوط پاسخ می‌دهند.

ناحیه ی بعدی V۴ است. این ناحیه در مجاورت IT قرار داردو نورون های آن خصوصیات پیچیده ای از تصویر ها را تشخیص می‌دهند. این ناحیه مرکز پردازش رنگ نیز به شمار می‌آید.

۳.۵.۲ ناحیه ی IT

قطعه‌ی مجانبی قشر مغز، آخرین ناحیه‌ای که داده های محرک های تصویری در مغز را پردازش می‌کند. این ناحیه ورودی خود را از V۲ و V۴ دریافت می‌کند و در شناسایی اشیا و تمایز آن‌ها درگیر است. نورون های این ناحیه همواره مورد توجه عصب شناسان بوده است و مطالعات فراوانی در این زمینه انجام شده است.

این ناحیه به دو بخش اصلی تقسیم می‌شود: IT عقبی و IT جلویی .

در IT عقبی اکثر نورون‌ها با ترکیب ساده‌ی ویژگی‌ها مانند خطوط یا حلقه ها در اندازه، جهت و رنگ های متفاوت فعال می‌شوند.

نورون های این ناحیه سلول های اصلی نامیده می‌شوند. ولی نورون های ناحیه ی IT جلویی به ویژگی های پیچیده تری نیاز دارند . این نورون ها که سلول های جزیی نامیده می شوند، ورودی خود را از نورون های ناحیه ی IT عقبی می گیرند.

نکته‌ی مهم در مطالعه ی نورون های IT استفاده از مجموعه‌ی محرکی است که بتواند تمام فرضیه های احتمالی در مورد خصوصیات این نورون ها را شامل شود.در مطالعه ای که در سال1995 انجام شد، مجموعه‌ی بسیار زیادی از اشیای گوناگون برای تحریک نورون های IT استفاده شد. سپس هر شکلی که پاسخ قوی از نورون استخراج می‌کرد، در یک فرایند ساده سازی به اجزای تشکیل دهنده‌ی خود تجزیه می‌شد. این ساده سازی تا هنگامی که پاسخ نورون افت قابل ملاحظه ای نداشت، ادامه می یافت. بدین ترتیب، محرک هایی برای هر نورون پیدا می شدکه پاسخ نسبتا قوی در نورون ایجاد می‌کردند و تا حد امکان ساده بودند.

فصل ۳

دوربین های با سنسور دید دینامیک

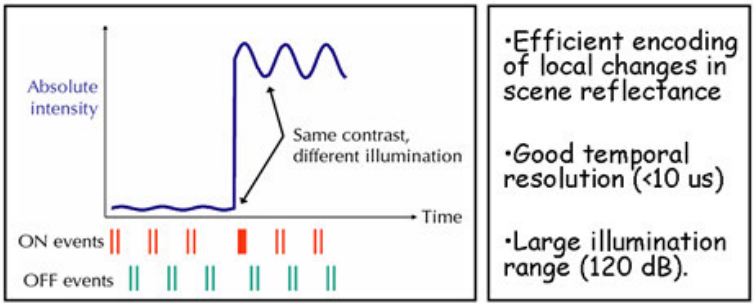
دوربین های معمولی محیط فیلم برداری را به صورت سری ای از فریم ها می بینند .این فریم‌های پشت سر هم اطلاعات بی‌مصرف و اضافی زیادی دارند که در نتیجه باعث می‌شود حافظه‌ی زیادی را به اشغال در بیاورد و همین طور محاسبات زیاد تر شده و زمان زیادی را هم هدر می‌دهند. به علاوه برای هر فریم زمان یکسانی روی همه ی پیکسل هایش صرف می‌شود که این موضوع برای صحنه هایی که بسیار تاریک یا بسیار روشن هستند مشکل ایجاد می‌کند.

دوربین های DVS این مشکلات را ،با تکنولوژی‌ای که همانند شبکیه در چشم انسان کار می کند ، حل می‌کنند . به جای فرستادن همه ی اطلاعات در هر فریم تنها آن پیکسل هایی فرستاده می‌شوند که تغییراتی در مقادیرشان رخ داده است .در نتیجه به جای جریانی از فریم ها ، جریانی از تغییرات مورد نگهداری و بررسی قرار می‌گیرد [۳].

۱.۳ بررسی کارکرد دوربین های DVS

کارکرد این دوربین ها این گونه است که در آن پیکسل ها به رویدادهایی با زمان‌بندی دقیق به صورت کنتراست موقت واکنش نشان می‌دهند . تغییرات صحنه یا تغییرات یک شی با بازتابش و روشنایی ثابت باعث تغییرات نسبی در شدت نور می شود . بنابراین پیکسلها به طور ذاتی به روشنایی صحنه منجر می شوند و به طور مستقیم تغییرات بازتابی صحنه را رمزگذاری می کنند.

رویداد ها به طور غیر همزمان و تقریبا به صورت بلافاصله روی یک گذرگاه خارج می شوند ، در نتیجه دقت زمانی بالاتری را نسبت به نرخ فریم ها ،در حالتی که انتقال ها از طریق فریم ها است، دارند .

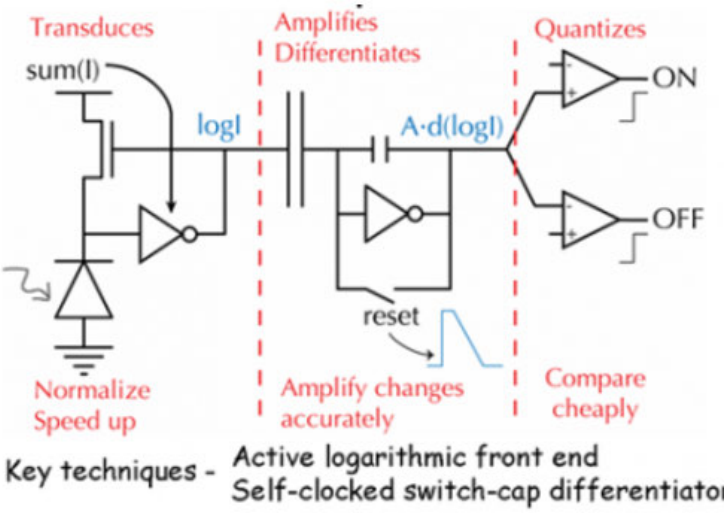


شکل ۱.۳ : واکنش در هنگام تغییرات در شدت نور

۲.۳ بررسی تکنولوژیکی دوربین های DVS

چهار نوآوری کلیدی در این دوربین ها عبارت است از : طراحی پیکسل ها ، ژنراتورهای تعویض دیجیتال روی تراشه ، پیاده سازی بسیار کاربردی USB۲ و نرم افزار پردازش jAER .

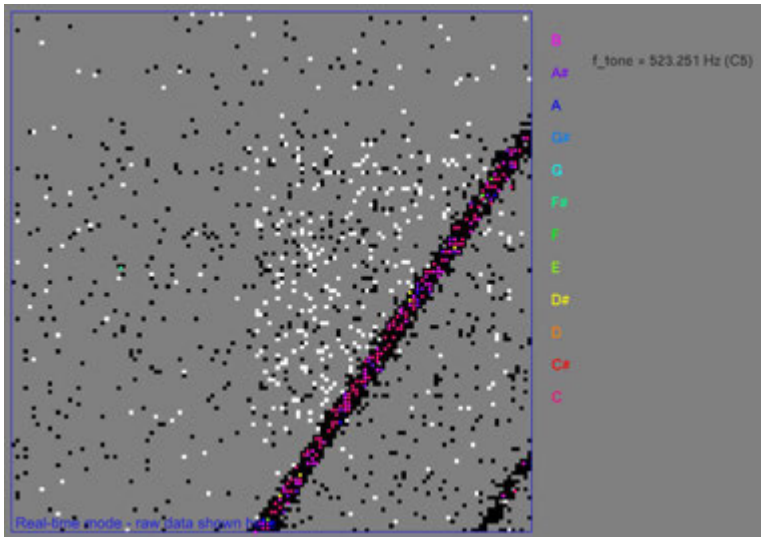
پیکسل ها از گیرنده های نوری زمان – پیوسته استفاده می کنند(در واقع از گیرنده های نوری تطابق پذیر در چشم الهام گرفته است) ، که توسط تفکیک کننده تعویض خازن خودکار بهنگام دقیق کنترل می شوند (که این ها هم از ستون تقویت کننده ای که در پالس تصویر برداری دو قطبی استفاده می شود الهام گرفته شده است). مهم ترین ایده ی ابتکاری در این پیکسل ها ایده ی خود زمان بندی تعویض لغزشی و خود بیشگی گیرنده های نوری است .



شکل ۲.۳ : پیاده سازی سخت افزاری دوربین های DVS

۳.۳ کاربرد های دوربین های DVS

از کاربرد های این دوربین ها می توان به موضوع به تصویر در آوردن ارتعاشات اشاره کرد ، هدف این است که ارتعاشات اجسام را بدون برخورد یا تماس با آنها تصویر برداری کنیم . که برای این کار از این دوربین ها استفاده کرده اند که با خاصیت تاخیر میکرو ثانیه ای هستند(شکل۳.۳).



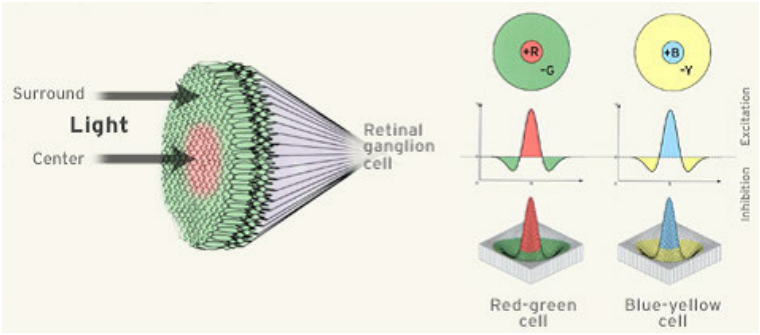
شکل ۳.۳: تصویری از ارتعاشات یک جسم توسط دوربین های DVS

از کاربرد های دیگر این دوربین ها می توان در تحقیقات اختلالات خواب ، روباتیک موبایل ، پیگیری سرعت ذرات سیال ، تصویر برداری سریع در روشنایی بد .

فصل ۴

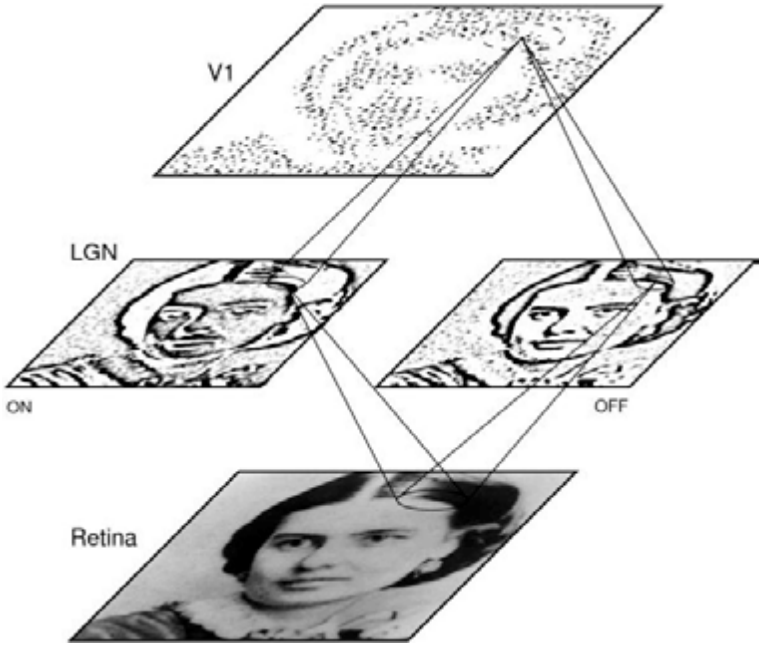
پیاده سازی نرم افزاری دوربین های با سنسور دید دینامیک

برای پیاده سازی نرم افزاری این دوربین ها علاوه بر آشنایی با نحوه ی کارکرد چشم به برسی فیلتر گاوسی می پردازیم ، در اصل هنگامی که گیرنده های نوری اطلاعات را به سلول های گانگلیون می دهند این سلول ها بسته به نوعشان ، همانطور که در بخش های قبل گفته شد :
سلول های گانگلیونی P و M ، واکنششان به صورت تابع گاوسی خواهد بود (شکل۱.۴) .



شکل ۱.۴ : واکنش سلول های گانگلیون

خروجی این سلول ها در اصل به صورت تفاضل تابع های گاوسی ایجاد شده از آن ها به LGN فرستاده می شود(شکل۳.۴) .
در نتیجه با توجه به (شکل۳.۴) می بینیم که در LGN داده ها شکلی از تصویر اولیه اند که لبه ها در آن ها مورد تمرکز قرار گرفته است .
در اصل در دوربین های DVS نیز از همین ایده استفاده شده است ، هدف این پروژه هم پیاده سازی این دوربین ها است . در بخش بعد به توضیح مختصری دربارهی نحوه ی کارکرد فیلتر گاوسی



شکل ۲.۴: cortex visual to retina

و DOG[1] می پردازیم .

۱.۴ فیلتر گاوسی و DOG

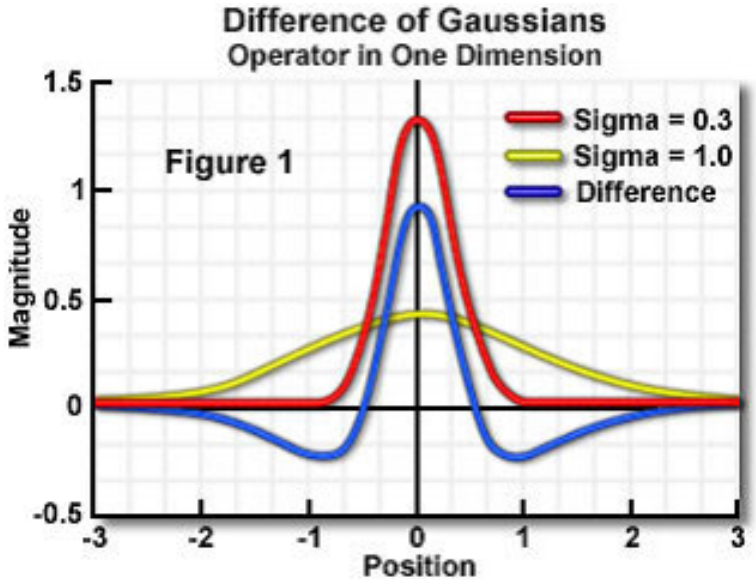
فیلتر گاوسی :

فیلتر گاوسی یک نوع فیلتر میانگین وزندار میباشد که مولفه هاي ماتریس ماسک آن از تابع گاوس اقتباس میگردد.تابع گاوسی در علوم احتمال، آمار و هوش مصنوعی و به ویژه در توزیع نرمال، استفاده فراوان دارد.

در ماسک فیلتر گاوسی، بیشترین ارزش به پیکسل اصلی(مرکزي) داده میشود و پیکسل هاي همسایه وزنی متناسب با فاصله اشان تا پیکسل مرکزي به خود اختصاص میدهند. هرچه فاصله دورتر میشود مقدار وزن نیز کوچکتر میشود.

از دلایلی که از فیلتر گاوسین استفاده می کنیم حذف noise است که باعث جلوگیری از اشتباه در تشخیص لبه ها می شود . این نکته اي است که باعث میشود لبه ها و مرزها بهتر حفظ گردد.منظور از لبه ، تغییرات ناگهانی محلی در روشنایی تصویر است و توجه شود که تغییرات ناشي از نویز لبه نیستند .

[1] difference of gaussian



شکل ۳.۴: gaussian of difference

در ریاضیات، تابع گاوسی (نام گذاری شده به نام کارل فریدریش گاوس)، تابعی است به شکل نمایی که به صورت زیر تعریف می شود(تابع گوسی دومتغیره به صورت زیر میباشد) :

$$G(X,Y) = \frac{e^{\frac{x^2+y^2}{2\delta^2}}}{2\pi\delta^2} \qquad (۱.۴)$$

که δ یک عدد حقیقی مثبت است. این تابع یک تابع پیوسته میباشد و در همه جا مثبت است. تابع گوسی ماکزیمم خود را در (۰،۰) اختیار میکند و در بینهایت به صفر میل میکند. هرچه مقدارδبزرگتر باشد، آنگاه تابع با سرعت کمتري به صفر میل میکند.

فیلتر گوسی از تابع گوسی ساخته میشود. تابع گوسی یک تابع پیوسته است، اما فیلتر ماهیتی گسسته دارد. لذا یک تقریب گسسته از تابع گوسی مورد استفاده قرار میگیرد. رویکردهاي متفاوتی براي ساخت این تقریب گسسته درنظر گرفته میشود که ماهیتا با یکدیگر معادلند. در شکل (شکل۴.۴) نمونه ای از اعمال فیلتر گاوسی را می بینیم .

در ماتریس ماسک فیلتر گاوسی با پارامترδ ،بهتر است که اندازه ماتریس ماسک برابر زیر باشد ، که درآن K سایز ماتریس است .

چرا که در این حالت اندازه تمامی مولفه ها به اندازه کافی بزرگ است و لذا فیلتر کردن با این ماتریس در تمامی مولفه ها تاثیرگذار خواهد بود. اگر اندازه ماتریس ماسک بزرگتر از مقدار پیشنهادي فوق باشد، آنگاه مولفه هاي کناري ماتریس بسیار کوچک هستند و یا در فرایند فیلتر کردن تاثیر بسیار ناچیزي دارند. در این شرایط فقط حجم محاسبات بالا میرود و همچنین کناره هاي تصویر به خاطر

شکل ۴.۴: فیلتر گاوسین

لایه گزاري بیشتر به هم میریزد .

$$2K + 1 = 2[3\delta] \tag{۲.۴}$$

حال به بررسی تفاضل دو فیلتر گاوسی(DOG)می پردازیم :
هدف از اعمال این فیلتر پیدا کردن لبه ها در تصویر است .
الگوهـاي تشـخیص لبـه شـامل سـه مرحلـه ي فیلترینـگ، مشتق گیري و تشخیص می باشند. در مرحله ي فیلترینگ، تصویر از یک فیلتر به منظور حذف نویز گذرانده می شود. در مرحله ي مشتق گیري، موقعیت هـایی در تصـویر بـا تغییـرات شدت نویز زیاد برجسته می گردند. سرانجام، در مرحله ي تشخیص با آستانه گیري از نقـاطی کـه در مرحلـه ي مشـتق گیـري برجسـته شـده انـد نقـاط لبـه تعیـین می گردند.
در اینجا ما برای قسمت فیلترینگ از فیلتر گاوسی استفاده می کنیم و در مرحله ی مشتق گیری اندازه گیری تغییرات می تواند از طریق مشتق گیری هم انجام شود که بیشترین تغییرات به معنای ماکزیمم شدن مشتق اول و صفر شدن مشتق دوم است ، بردار گرادیان حداکثر نرخ تغییرات روشنایی تعیین می کند اما با توجه به فرمول مشتق در حالت گسسته و در حالت دو بعدی از تفاضل متناهی به عنوان یک تخمینی از مشتق استفاده می کنیم .

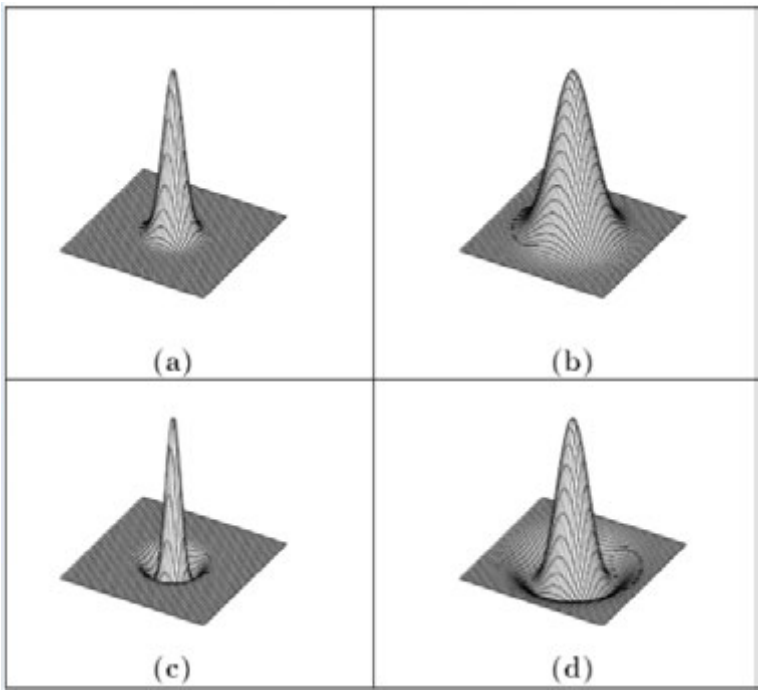
$$\nabla f = [\frac{\nabla f}{\nabla x}, \frac{\nabla f}{\nabla y}] \tag{۳.۴}$$

برای این کار می آییم از یک تصویر دو تا تصویر با اعمال فیلتر گاوسین با دو تنظیم متفاوت(اندازه ی ماتریس و مقدار سیگما) ایجاد می کنیم ، تفاضل این دو تصویر ، تصویری است که لبه های تصویر

اوله را نشان می دهد .در اصل DOG تفاضل دو ماسک گوسین با δ کاملاً متفاوت با تصویر است .در اصل ما برای کشف لبه ها از روش تفاضل دو تابع گاوسی استفاده کرده ایم .

روش هاي تشخيص لبه را می توان به سه رده دسته بندي نمود:

- اولین رده شامل عملگرهاي گرادیان ۳×۳ و ۵×۵ است که شامل عملگرهاي پرویت، روبرت، سابل و لاپلاسین می باشند.
- دومین رده، شامل عملگرهایی است که شکل سطح را در بر می گیرند. عملگرهاي Hueckel ، Hartly و Haralick به این رده تعلق دارند.
- نهایتاً، سومین دسته از مشتقات گاوسی استفاده می نمایند.



شکل ۵.۴: (c)DOG ، (b) gaussian filter sigma = ۲، (a) gaussian filter sigma = ۱
(d) DOG sigma=۱، sigma=۲

در شکل (۶.۴) نمونه ای از اعمال فیلتر DOG بر روی تصویری که در شکل (۴.۴) دیدیم را می بینیم .

شکل ۶.۴: نمونه ای از اعمال DOG

۲.۴ چگونگی پیاده سازی نرم افزاری DVS camera

برای پیاده سازی نرم افزاری این دوربین ها ، ابتدا ویدیوی اولیه که توسط یک دوربین معمولی (مکان دوربین ثابت است) به عنوان ورودی به برنامه داده می‌شود . سپس روی هر فریم فیلتر گاوسین با دو تنظیم مختلف (این تنظیم ها توسط کاربر انجام می‌شود ، ابتدا یک مقدار برای اندازهی ماتریس گاوسین داده می‌شود و سپس دو مقدار سیگمای متفاوت برای ساختن دو فیاتر گاوسین متفاوت داده می‌شود .) اعمال می شود و تفاضل این دو تصویر جدید را محاسبه می کند . در نتیجه برای هر فریم بعد از اعمال DOG ، می آییم تفاضلش را با فریم قبلی (که DOG رویش اعمال شده بوده است) محاسبه می‌کنیم و خروجی را که خود یک فریم است ، به عنوان یک فریم ویدیو ی نهایی نمایش می‌دهیم . برای جلوگیری از ایجاد noise و پرش های تصویر در ویدیوی نهایی در هرمرحله نرمال سازی انجام می‌دهیم ، به این معنا که ماکسیمم فریم فعلی را با میانگینی از ماکسیمم ها فریم ها قبلی میانگینی وزن دار می‌گیریم و در نهایت نرمال سازی را با این مقدار ماکسیمم انجام می‌دهیم ، این قسمت در کد متلبی که در شکل (۷.۴) نشان داده شده است دیده می‌شود .

در شکل زیر قسمتی از کد(بدون قسمت (UI که در متلب نوشته شده است را می بینید که تمام مراحل گفته شده در بالا روی ویدیو ی ورودی اعمال شده است . برنامه اصلی به زبان c++ نوشته شده است و قسمت رابط کاربری آن نیز با زبان csharp پیاده سازی شده است که در قسمت پیوست کدشان قرار داده شده است .

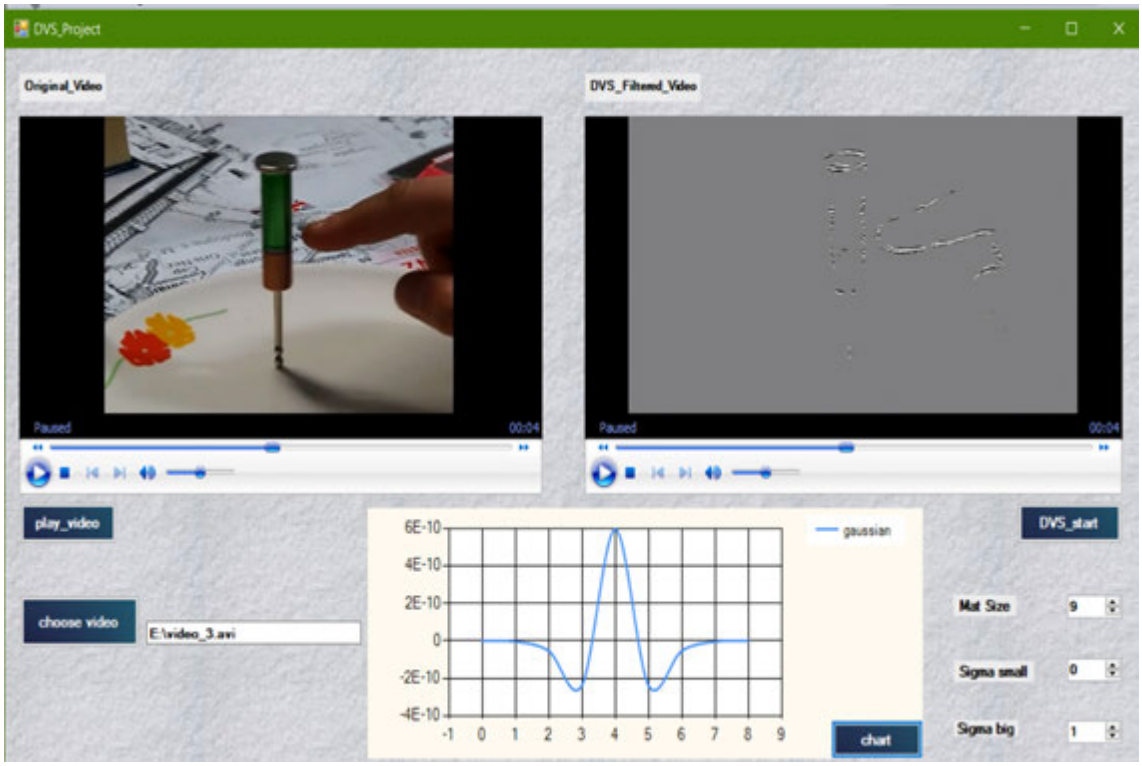
```
inputVideo = VideoReader(fullfile(workingDir,inputFile));
outputVideo = VideoWriter(fullfile(workingDir,
['out_',inputFile]));
open(outputVideo)
مقدار دهی پارامتر ها برای اعمال فیلتر گاوسین
gaussian1 = fspecial('Gaussian', 10, .3);
gaussian2 = fspecial('Gaussian', 10, 1);
dog = gaussian1 - gaussian2;
preFrame = rgb2gray(readFrame(inputVideo));
while hasFrame(inputVideo)
   vidFrame = rgb2gray(readFrame(inputVideo));
    اعمال فیلتر تفاضل دو فیلتر گاوسین ایجاد شده
    dogFilterImagePre = conv2(double(preFrame), dog, 'same');
    dogFilterImageCur = conv2(double(vidFrame), dog, 'same');
    %res = imabsdiff(dogFilterImagePre, dogFilterImageCur);
    res = (dogFilterImagePre - dogFilterImageCur)
    نرمال سازی با میانگینی از مقادیر ماکزیمم فریمهای قبلی
    max_new = max(max(abs(res)));
    max_avg = (max_new*0.1 + max_old*0.9);
    max_old = max_avg;
    res = (0.5/max_avg) * res;
    res = res + 0.5;
    res(res > 1) = 1;
    res(res < 0) = 0;
    تبدیل مقادیر خروجی به سه رنگ سیاه،سفید،خاکستری
    res(res < 0.55 & res > 0.45)= 0.5;
    res(res<=0.45) = 0;
    res(res>=0.55)= 1;
    preFrame = vidFrame;
    writeVideo(outputVideo,res);
end
close(outputVideo)
```


شکل ۷.۴: کد متلب

۳.۴ خروجی نرم افزار

در این بخش به رابط کاربری برنامه‌ای که پیاده سازی کرده‌ام اشاره ای می‌کنیم .

همان طور که در شکل(۸.۴) مشاهده می‌کنید، هنگامی که برنامه شروع یه اجرا می‌کند این صفحه ایجاد می‌شود . برای انتخاب ویدیو ابتدا گزینه ی choose video را انتخاب می کنیم و با رفتن به مسیری که در آن ویدیو را ذخیره کرده‌ایم انتخابش می‌کنیم . برای نمایش ویدیو ی اولیه از مدیا پلیر سمت چپ استفاده شده است ، که با زدن گزینه‌ی play video ویدیو شروع به پخش شدن می‌کند. سپس برای تبدیل آن به ویدیویی که یک دوربین DVS ایجاد می‌کند ، ابتدا باید به پارامتر هایی که در قسمت پایین سمت راست رابط کاربری هستند مقدار بدهیم . این پارامتر ها عبارت اند از : mat size،sigma small،sigma big . این پارامترها به ترتیب بیانگر سایز ماتریسی که میخواهد فیلتر گاوسی را اعمال کند و سیگمای کوچک و سیگمای بزرگ دو مقداری هستند که دو فیلتر را از هم متفاوت می‌کنند تا بتوان DOG را محاسبه کرد ،این سیگما ها در اصل میزان تاثیر گذاری پیکسل های اطراف را تعیین می‌کنند . پس از مقدار دهی به پارامترها ، با زدن گزینه ی start DVS این مقادیر به برنامه ی اصلی داده می شود و با اعمال آن‌ها و ساختن مقادیر تابع گاوسی ایجاد شده این مقادیر به یک فایل ریخته می شود و در نهایت ویدیو ی نهایی تولید می شود و در مدیا پلیر سمت راست نمایش داده می شود . و با زدن گزینه ی chart ، مقادیر تولید شدخ برای نمایش تابع گاوسی از فایل گفته شده خوانده می‌شوند و نمودار تابع گاوسی ایجاد شده نشان داده می‌شود .



شکل ۸.۴: رابط کاربری برنامه

کتاب‌نامه

[۱] BASIC VISION an introduction to visual perception،authors : Robert Snowden Peter Thompson Tom Troscianko School of Psychology Department of Psychology School of Experimental Psychology Cardiff University University of York University of Bristol

[۲] SEEING A Computational Approch to Biological Vision Second Edition authors: John P.Frisby and James V.Stone

[۳] https://inilabs.com/products/dynamic-vision-sensors/

فصل ۵

پیوست(کد اصلی و کد قسمت رابط کاربری)

```
..
DVS_Project .....................................
........

#include "stdafx.h"
#include "opencv2\highgui\highgui.hpp"
#include "opencv2\imgproc\imgproc.hpp"
#include "opencv\cv.h"
#include "opencv2\features2d\features2d.hpp"
#include <opencv2\opencv.hpp>
#include <iostream>

using namespace cv;
using namespace std;

int main()//int argc, char *argv[])
{

	int ksize;
	float sigmaSmall;
	float sigmaBig;

	//Input information from
DVS_UI ..............................
	string strValues = string(argv[1]);
	ksize = atoi(strValues.data());
	int iPos = strValues.find(',');
	strValues.erase(0, iPos + 1);
	sigmaSmall = atof(strValues.data());
	iPos = strValues.find(',');
	strValues.erase(0, iPos + 1);
	sigmaBig = atof(strValues.data());
	iPos = strValues.find(',');
	strValues.erase(0, iPos + 1);

	string video_name = strValues;
```

```
	//reading a video
	cout << "video name : " << video_name << endl;
	cout << "K size      : " << ksize << endl;
	cout << "Sigma small : " << sigmaSmall << endl;
	cout << "Sigma big   : " << sigmaBig << endl;
	cout << "----------------------------" << endl;

	VideoCapture cap(video_name); // open the video file for
	reading
	if (!cap.isOpened())  // if not success, exit program
	{
		cout << "Cannot open the video file" << endl;
		return -1;
	}

	double fps = cap.get(CV_CAP_PROP_FPS);

	cout << "Frame per seconds : " << fps << endl;

//	namedWindow("MyVideo", CV_WINDOW_AUTOSIZE);


	Size S = Size((int)cap.get(CV_CAP_PROP_FRAME_WIDTH),
		(int)cap.get(CV_CAP_PROP_FRAME_HEIGHT));
	//int codec = CV_FOURCC('M', 'J', 'P', 'G');
	int ex = static_cast<int>(cap.get(CV_CAP_PROP_FOURCC));
	VideoWriter out("e:\\output34.avi", CV_FOURCC('W', 'M', 'V',
'2'), fps, S, false);   //CV_FOURCC('M','P','4','2')
CV_FOURCC('W', 'M', 'V', '2')


	int fsize = cap.get(CV_CAP_PROP_FRAME_COUNT);
	cout << fsize;
```

```
	//DOG set up ..........................................

	int intSigmaBig = 70;
	int intMaxSigmaBig = 120;

	int intSigmaSmall = 60;
	int intMaxSigmaSmall = 120;
	//float sigmaBig =  8;//intSigmaBig / 10.0f;
	//float sigmaSmall = 0.3;//intSigmaSmall / 100.0f;

	// sigma = 0.3*((ksize-1)*0.5 - 1) + 0.8
	//int ksize = 51;// 9;// ceilf((sigmaBig - 0.8f) / 0.3f) * 2
+ 3;
	//int ksize = 10;
	cv::Mat gauBig = getGaussianKernel(ksize, sigmaBig, CV_32F);
	cv::Mat gauSmall = getGaussianKernel(ksize, sigmaSmall, CV_
32F);

	cv::Mat DoG = gauSmall - gauBig;

	FILE *pFile = fopen("e:\\dataDoG.txt", "wt");
	if (pFile)
	{
		INT32 *dataDoG = new INT32 [ksize];
		memcpy(dataDoG, DoG.data, ksize * 4);
		for (int i = 0; i < ksize; i++)
			fprintf(pFile, "%d\n", dataDoG[i]);
		fclose(pFile);
		delete [] dataDoG;
	}
```

```
	//start main work .....................................
	Mat pre_frame_rgb ,pre_frame_gray;
	double max_cur = 0;
	double max_avg = 0;
	double max_pre = 127;
	int i = 0;
	while (i < fsize - 1)
	{
		Mat cur_frame_rgb;
		Mat cur_frame_gray;
		cap >> cur_frame_rgb;
		cvtColor(cur_frame_rgb, cur_frame_gray, CV_RGB2GRAY);
		if (i == 0)
			pre_frame_gray = cur_frame_gray;
		//DOG filter ......................
		Mat filt_pre_frame;
		filter2D(pre_frame_gray, filt_pre_frame, -1, DoG,
Point(-1, -1),0, BORDER_DEFAULT);
		Mat filt_cur_frame;
		filter2D(cur_frame_gray, filt_cur_frame, -1, DoG,
Point(-1, -1),0, BORDER_DEFAULT);
		Mat sub_fil_frame;
		subtract(filt_cur_frame/2, filt_pre_frame/2,
sub_fil_frame);
		sub_fil_frame  = sub_fil_frame + 127;
		max_cur = *max_element(sub_fil_frame.begin<uchar>(),
sub_fil_frame.end<uchar>());
		 max_avg = (max_cur*0.1 + max_pre*0.9);
		 max_pre = max_avg;
		 if (max_avg < max_cur)
			max_avg = max_cur;
		 sub_fil_frame = 255 * sub_fil_frame / max_avg;
		 sub_fil_frame = (sub_fil_frame / 64) * 64;
		 sub_fil_frame.convertTo(sub_fil_frame,CV_32FC1);
```

```
            //reading frame
values ..................................................
.
            for (size_t y=0; y < sub_fil_frame.rows ; y++)
            {
                 for (size_t x = 0; x < sub_fil_frame.cols; x++)
                 {
                      if (sub_fil_frame.at<float>(y, x) > 60.0 &&
sub_fil_frame.at<float>(y, x) < 180.0)
                           sub_fil_frame.at<float>(y, x) = 127.0;
                 }
            }


            sub_fil_frame.convertTo(sub_fil_frame, CV_8UC1);
           out << sub_fil_frame;
           pre_frame_gray = cur_frame_gray;

//         imshow("frames", sub_fil_frame);
           //cvReleaseMat(sub_fil_frame);

           if (waitKey(30) == 27) //wait for 'esc' key press for
30 ms. If 'esc' key is pressed, break loop
           {
                cout << "esc key is pressed by user" << endl;
                break;
           }

           i++;
           //cout << i << endl;
      }
      ReleaseCapture();
```

```
namespace DVS_UI
{
    public class DVS_param
    {
        int ksize;
        double sigmasmall;
        double sigmabig ;
        string video;
        Process myProcess = new Process();
        public void set_ksize(int k)
            ksize = k;
        public void set_sigmabig(double b)
            sigmabig = b;
        public void set_sigmasmall(double s)
            sigmasmall = s;
        public void set_video(string v)
            video = v;
        public void start_dvs()
        {
            //calling the dvs_project in c++ .................
             myProcess.StartInfo.FileName =
"C:/Users/niloofar/Documents/Visual Studio
2013/Projects/DVS_Opencv/Debug/DVS_Opencv.exe"; // Note the
absolute path
               //--- values
                string video_name = video;
                string val1 = ksize.ToString();
                string val2 = sigmasmall.ToString();
                string val3 = sigmabig.ToString();
                myProcess.StartInfo.Arguments = val1 + "," + val2
+ "," + val3 + "," + video_name;
                // This will actually run the process.
                myProcess.Start();
                myProcess.WaitForExit();
                 // Waits here for the process to exit.
            }
        }
    }
}
```

```
namespace DVS_UI
{

    public partial class DVS_UI : Form
    {
        int ksize;
        double sigmas;
        double sigmab;
        string videoname;
        DVS_param dvsparam = new DVS_param();
        public DVS_UI()
        {
            InitializeComponent();
        }
        private void MatSizeInput_ValueChanged(object sender,
          EventArgs e)
        {
            ksize = Convert.ToInt32(MatSizeInput.Text);
            dvsparam.set_ksize(ksize);
        }
        private void SigmaBigInput_ValueChanged(object sender,
          EventArgs e)
        {
            sigmab = float.Parse(SigmaBigInput.Text);
            dvsparam.set_sigmabig(sigmab);
        }
        private void SigmaSmallInput_ValueChanged(object sender,
          EventArgs e)
        {
            sigmas = float.Parse(SigmaSmallInput.Text);
            dvsparam.set_sigmasmall(sigmas);
        }
```

```
// start to do
  dvs_project.............................
private void DVS_start_Click(object sender, EventArgs e)
{
    if (MatSizeInput.Text != "" && SigmaSmallInput.Text !
  = "" && SigmaBigInput.Text != "")
    {
        dvsparam.start_dvs();
        axWindowsMediaPlayer1.URL = "e:\\output34.avi";
        axWindowsMediaPlayer1.Ctlcontrols.play();
    }
    else
        MessageBox.Show("please enter parameters !!", "
  null parameter ", MessageBoxButtons.OK,
  MessageBoxIcon.Error);
}

//play the
  video ..........................................
private void play_video_Click(object sender, EventArgs e)
{
    axWindowsMediaPlayer2.URL = video_name_input.Text;
    axWindowsMediaPlayer2.Ctlcontrols.play();
}

private void video_name_input_func(object sender,
  EventArgs e)
{
    videoname = video_name_input.Text;
    dvsparam.set_video(videoname);
    axWindowsMediaPlayer2.URL = video_name_input.Text;
    axWindowsMediaPlayer2.Ctlcontrols.play();
}
```

```
private void choose_video_Click(object sender, EventArgs
  e)
{
    OpenFileDialog openfiledialoge = new
  OpenFileDialog();
    if(openfiledialoge.ShowDialog() ==
  System.Windows.Forms.DialogResult.OK)
    {
        video_name_input.Text = openfiledialoge.FileName;
    }
}

private void button3_Click(object sender, EventArgs e)
{

    int counter = 0;
    string line;

    // Read the file and display it line by line.
    System.IO.StreamReader file =
       new System.IO.StreamReader("e:\\dataDoG.txt");
    while ((line = file.ReadLine()) != null)
    {


  this.chart1.Series["gaussian"].Points.AddXY(counter,flo
  at.Parse(line)/1000000000.0);

        //Console.WriteLine(line);
        counter++;
    }

    file.Close();

}
```