

بسم الله الرحمن الرحيم

شرکت مهندسی نرم افزاری هلو

گزارش الگوریتم تغییر موقعیت قرار گیری دوربین ها

کاری از امیر علی نسیمی

فهرست

مقدمه ۲

الگوریتم های استفاده شده ۲

الگوریتم اول ۴

مقدمه ۴

بررسی میزان تغییرات بین دو پیکسل ۴

نگاه وزن دار به جهت تغییرات ۶

نتایج ۸

الگوریتم دوم ۱۰

مقدمه ۱۰

روش محاسبه تغییرات ۱۰

نتایج ۱۰

محدودیت های مربوط به الگوریتم برتر ۱۱

مقدمه

مسئله Multiple-View Image در مسئله مورد نظر بسیار حائز اهمیت بوده است. با توجه به این موضوع هدف این است که در صورت تغییر ناخواسته و ناگهانی دوربین، کلیه میز و صندلی های تعیین شده توسط کاربر دوباره بر اساس حالت اول، با تغییرات دوربین تطبیق داده شود. در این خصوص الگوریتم های زیادی مورد استفاده قرار گرفته است که به دو مورد کلی از آنها که دارای بیشترین دقت بوده اند پرداخته خواهد شد. سپس در بخش های بعدی به بررسی محدودیت های مربوط به الگوریتم برتر پرداخته خواهد شد.

الگوریتم های استفاده شده

با توجه به این مسئله، دو الگوریتم برای بررسی تعریف شده است. در هر دو الگوریتم، نقطه ای مشترک در صحنه انتخاب می شود. این نقطه در دو تصویر گرفته شده از دو دوربین نیز موجود است. یکی از تصاویر به عنوان تصویر مرجع انتخاب خواهد شد لذا بر این اساس مختصات نقطه مشترک در این حالت، امکان تعیین میزان تغییرات در دو تصویر خواهد بود.

با توجه به اینکه تعیین این نقطه مشترک زمان بر بوده و نیاز به آموزش مدل های شبکه عصبی می باشد لذا جهت متعهد بودن به زمان بندی تعیین شده برای الگوریتم ها، یک فرد به عنوان نقطه ثابت در نظر گرفته شده است. برای مثال دو تصویر زیر از دو زاویه مختلف از دوربین گرفته شده است – جهت مشاهده میزان تغییر زاویه، به موقعیت جارختی در دو تصویر توجه شود:



الگوریتم اول

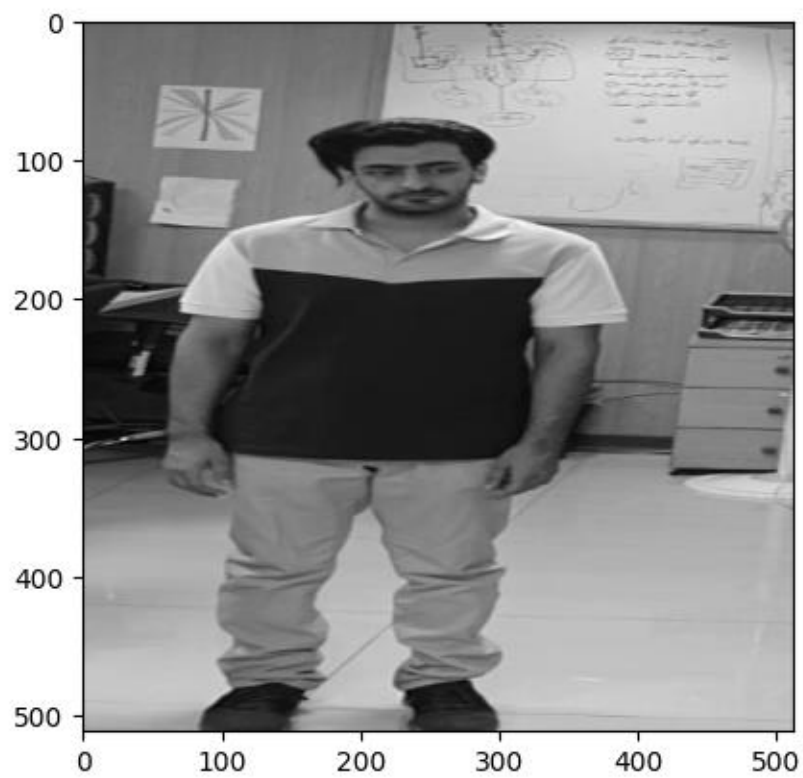
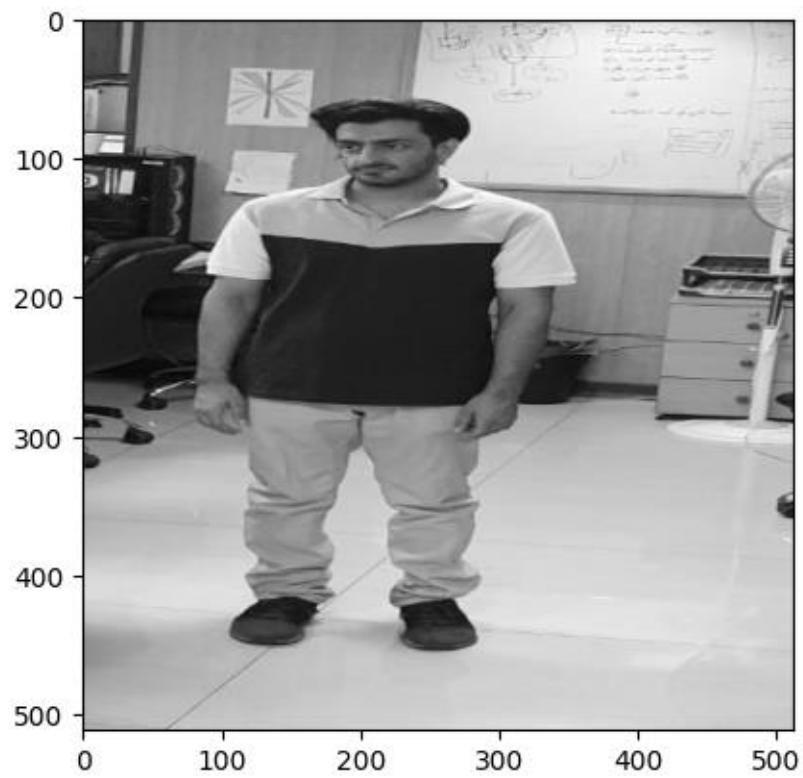
مقدمه

هدف از این الگوریتم استفاده از optical flow جهت تعیین میزان تغییرات پیکسل ها - یا در حالاتی پنجره از پیکسل ها - بین دو تصویر می باشد. این الگوریتم از چند بخش تشکیل شده است که توضیح در خصوص هر بخش به اختصار به شرح زیر می باشد.

بررسی میزان تغییرات بین دو پیکسل

جهت بررسی این مسئله به نقطه مشترک بین دو تصویر باید نظر کرد؛ دقت شود که برای انجام این مسئله، ابتدا با روش های هیستوگرامی و استخراج ویژگی، تصاویر مربوط به نقطه مشترک باید تعیین شود.

- الگوریتم Optical Flow: یک تکنیک بینایی ماشین است که برای تخمین حرکت اشیاء یا پیکسل ها بین چندین فریم متوالی در یک توالی ویدیویی استفاده می شود. ایده اصلی این الگوریتم پیگیری جابجایی شدت پیکسل ها یا نقاط ویژه کلیدی از یک فریم به فریم بعدی است. این الگوریتم فرض می کند که پیکسل های مجاور در یک فریم تصویر شباهت حرکت دارند. الگوریتم میزان و جهت این حرکت را محاسبه می کند.
- cv2.goodFeaturesToTrack: یک تابع در OpenCV است که برای شناسایی نقاط ویژه "خوب" در تصویر استفاده می شود. این تابع تصویری با سطوح خاکستری به عنوان ورودی می گیرد و مختصات نقاط ویژه کلیدی که برای پیگیری مناسب هستند را برمی گرداند. این تابع از الگوریتم شناسایی گوشه ها استفاده می کند که به دنبال مناطق با تغییر شیب یا تغییر شدت در یک تصویر می گردد. این مناطق احتمالاً ممیز و قابل پیگیری هستند. با توجه به این مسئله تصاویر باید بدین صورت باشند:



- **RANSAC:** یک روش آماری محکم برای تخمین پارامترهای مدل از مجموعه‌ای از نقاط داده با حضور قابل توجهی از داده‌های خارجی استفاده می‌شود. در خصوص **optical flow**، RANSAC می‌تواند برای حذف نقاط خارجی از مجموعه نقاط پیگیری شده استفاده شود، تا فقط تطابق‌های قابل اطمینان در نظر گرفته شوند. در این خصوص، الگوریتم مذکور به صورت تکراری یک زیرمجموعه تصادفی از نقاط داده (یک نمونه حداقلی) را برای جلب یک مدل (مثلاً مدل حرکت) انتخاب کرده و سپس ارزیابی می‌کند که چقدر مدل به تمام مجموعه داده می‌خورد. نقاط خارجی (نقاطی که به خوبی با مدل مطابقت ندارند) شناسایی و حذف می‌شوند و فرآیند تکرار می‌شود تا مدل بهبود یابد. همانطور که ذکر شد استفاده از این الگوریتم بسیار مناسب بوده زیرا با حذف تطابق‌های نویزی یا نادرست، دقت تخمین حرکت را افزایش می‌دهد.

نگاه وزن دار به جهت تغییرات

بخش بسیار مهمی است. هدف این است که محاسبه میزان تغییرات بصورت وزن دار انجام شود. قدم‌های مربوط به این الگوریتم:

- **تابع هدف:** این تابع یک مجموعه وزن را به عنوان ورودی می‌گیرد و یک مقدار اسکالر را محاسبه می‌کند که باید به کمینه رسانده شود. هدف از این تابع یافتن وزن‌هایی است که نرم L_2 (نرم اقلیدوسی) تغییر وزن داده شده را به حداقل برسانند. به عبارت دیگر، هدف از آن یافتن وزن‌هایی است که تغییر وزن به حداقل برسانند.

- **مقداردهی اولیه وزن‌ها:** تابع با مقداردهی اولیه وزن‌ها به ۰.۵ شروع می‌شود، که به معنای اولیه برابری وزن‌های تغییرات x و y است.

- **محاسبه آستانه پویا:** یک آستانه پویا بر اساس مقدار زاویه تغییر محاسبه می‌شود. این آستانه با ضرب یک مقدار پیش فرض (یک پارامتر تعیین شده توسط کاربر، با مقدار پیش فرض ۰.۱) با مقدار زاویه تغییر ضرب می‌شود. ایده این آستانه پویا این است که وزن‌ها را بسته به مقدار مشاهده شده تغییر کند. تغییرات کوچک آستانه کوچک‌تری خواهند داشت و تغییرات بزرگ آستانه بزرگ‌تری خواهند داشت.

- **تنظیم وزن بر اساس آستانه پویا:** کد چک می‌کند که آیا مقدار زاویه تغییر کمتر از آستانه پویا است یا خیر. اگر بله باشد، این نشان می‌دهد که یک تغییر نسبتاً کوچک وجود دارد. در چنین مواردی، کد جزو تعیین‌کننده (x, y) تغییر وزن را با استفاده از ماکزیمم گیری شناسایی می‌کند. سپس وزن بیشتری (0.8) به جزو تعیین‌کننده اختصاص می‌دهد در حالی که وزن جزو غیرتعیین‌کننده را در مقدار اولیه‌اش (0.5) نگه می‌دارد. این تنظیم باعث می‌شود که جهت جزو تعیین‌کننده در محاسبه تغییر بیشتری تأثیرگذار باشد.
- **محدودیت‌ها:** بهینه‌سازی تحت محدودیتی انجام می‌شود که تضمین می‌کند که مجموع وزن‌ها همیشه برابر ۱.۰ باشد. این محدودیت توازن وزن‌ها را حفظ می‌کند، به این معنا که همیشه مجموع وزن‌ها به ۱.۰ می‌رسد.
- **بهینه‌سازی – کمینه سازی:** در نهایت، کد از تابع **minimize** برای یافتن وزن‌های بهینه که کمینه کردن تابع هدف را انجام دهند در حالی که محدودیت را ارضا می‌کنند، استفاده می‌کند. نتیجه این عملیات مجموعه‌ای از وزن‌ها است که بهترین ترکیب را برای به حداقل رساندن تغییر وزن تعیین می‌کند.

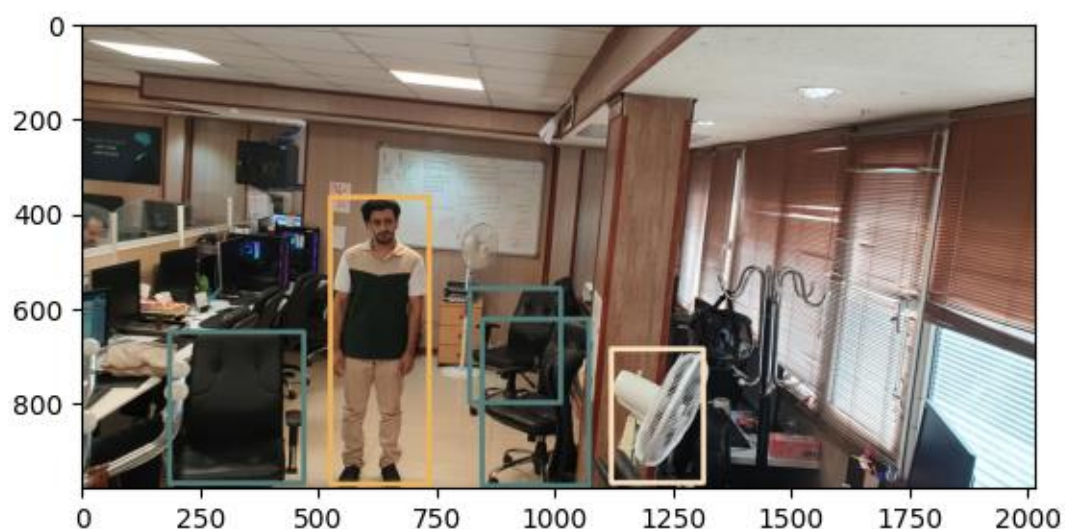
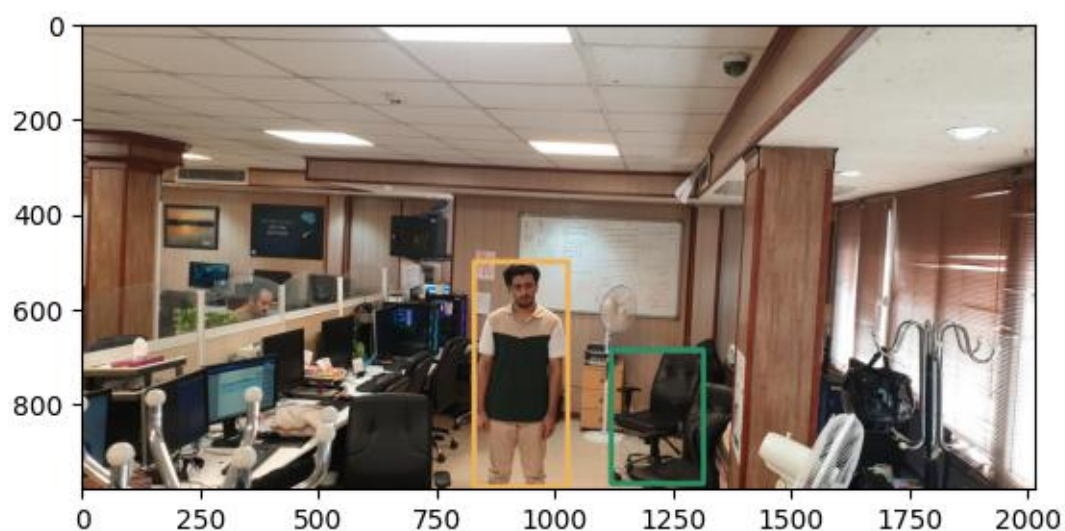
تأثیر استفاده از این مسئله:

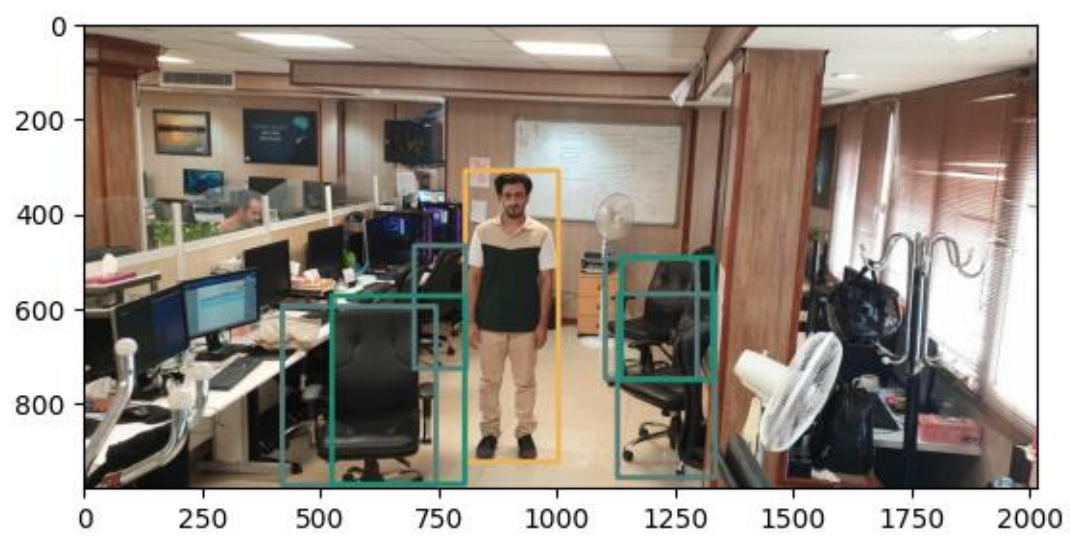
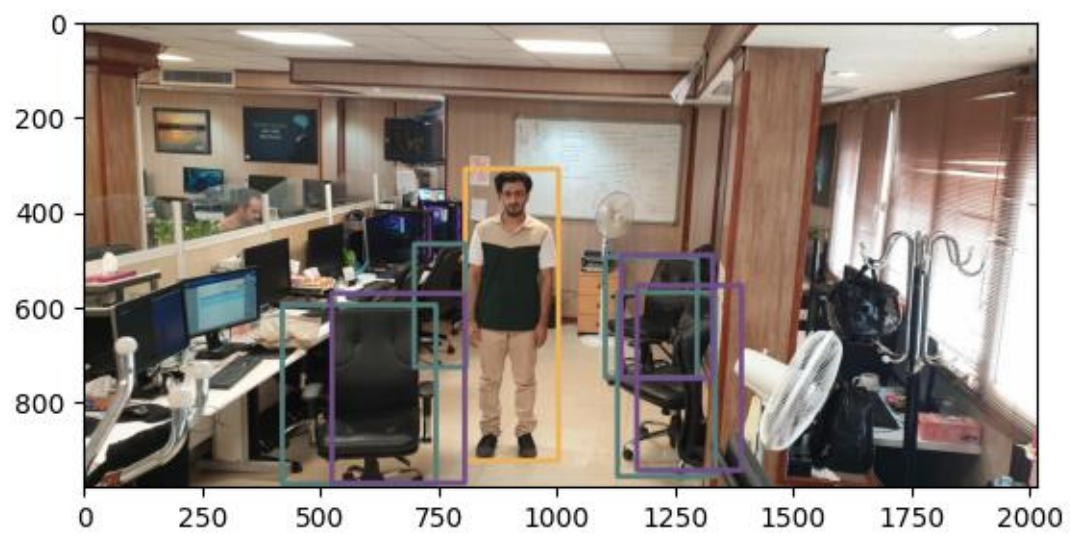
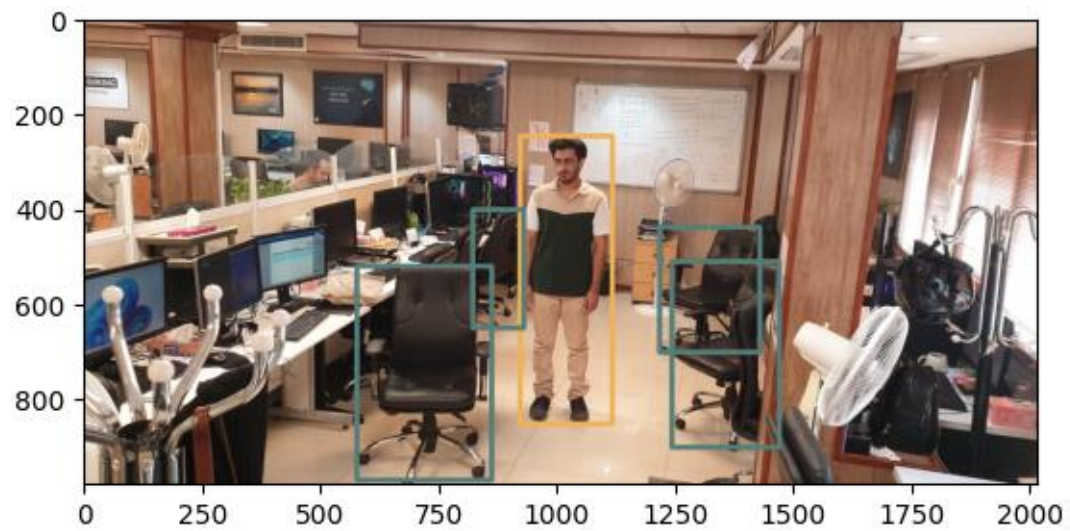
- تغییر وزن به تنظیم: با تنظیم وزن‌ها بر اساس مقدار تغییر مشاهده شده، تابع به صورت انعطاف‌پذیر اهمیت هر جزء (x, y) از تغییر وزن را تعیین می‌کند. این قابلیت امکان می‌دهد تا الگوریتم در مواجهه با تغییرات کوچک، دقت بیشتری داشته باشد.
- آستانه‌گذاری: مکانیزم آستانه‌گذاری پویا کمک می‌کند تا از تمرکز بیش از حد بر روی یک جزو برای تغییرات بزرگ جلوگیری شود و اطمینان حاصل شود که هر دو جزو x و y به طور معناداری در محاسبه نقش می‌آفرینند.
- ترکیب بهینه: فرآیند بهینه‌سازی به دنبال یافتن وزن‌های بهینه برای به حداقل رساندن تغییر وزن به موجب تابع هدف است. این به این معناست که تابع ترکیبی از وزن‌ها را تشخیص می‌دهد که بهترین ترکیب را برای جنبه تغییر وزن ارائه می‌دهد.

به طور خلاصه، این مورد نقش مهمی در افزایش دقت تشخیص تغییر زاویه دوربین ایفا کرده که با تنظیم وزن‌ها بر اساس مقدار مشاهده شده تغییر وزن به صورت پویا، این تنظیم بهینه به تحقق می‌پیوندد و کمک می‌کند تا تغییر محاسبه شده واقعی حرکت دوربین در موارد مختلف را به دقت بازتاب دهد.

نتایج

در این الگوریتم یکسری از نتایج مشهود است که به پیوست ضمیمه شده است. این روش نتیجه خوبی به همراه نداشت.





الگوریتم دوم

مقدمه

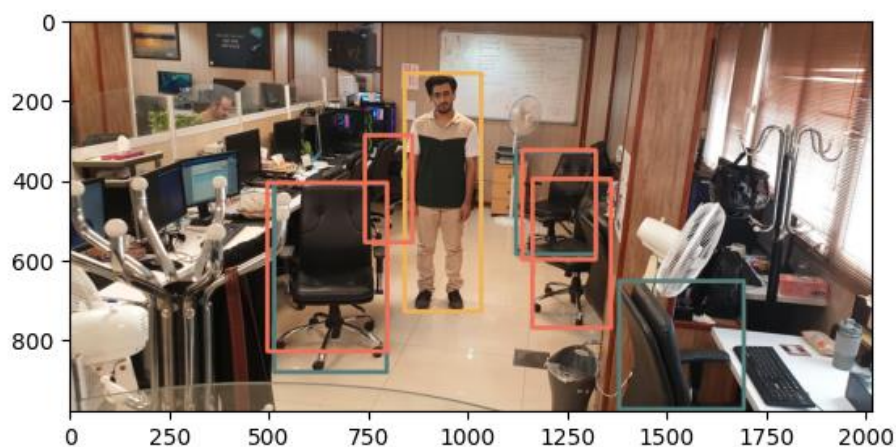
در این نگاه میزان تغییرات بطور مستقیم محاسبه خواهد شد. نیاز به محاسبه طولانی نبوده و سرعت بسیار بالا در این خصوص وجود دارد.

روش محاسبه تغییرات

در این روش به شرح زیر عمل می شود:

- برای دو نقطه x و y که برای دو نقطه مشترک در دو فریم وجود دارد - بر خلاف الگوریتم قبلی - میزان تغییرات خود فریم محاسبه می شود. با توجه به این مسئله بطور مستقیم فاصله اقلیدسی بین دو مرکز محاسبه می شود.
- جهت تعیین جهت، هر تغییر بر خلاف آن جهت با توجه به ضرب یک و منفی یک حاصل می شود.
- در نهایت تغییرات با تغییرات قبلی جمع می شود.

نتایج



محدودیت های مربوط به الگوریتم برتر

الگوریتم دوم بهتر است. با توجه به این مسئله، نکات زیر حائز اهمیت می باشند:

- تغییرات دوربین نباید از ۲۰ درجه بیشتر باشد
- تنها یک نقطه اشتراکی در تصویر باید وجود داشته باشد.
- ابعاد اصلی bounding box های از پیش تعریف شده در این الگوریتم تغییری نخواهند کرد.
- سرعت اجرای این الگوریتم بر بستر cpu حدود یک صدم میلی ثانیه می باشد.