



## ردیابی ابربی درنگی اهداف همزمان متحرک در تصاویر ویدئویی

محمد رضا نجفی<sup>۱\*</sup>، امین اله مه‌آبادی<sup>۲</sup>

\*نویسنده مسئول، دریافت: ۹۹/۰۷/۲۶، بازنگری: ۹۹/۰۸/۳۰، پذیرش: ۹۹/۰۹/۱۳

<sup>۱</sup>گروه مهندسی کامپیوتر و فناوری اطلاعات، دانشکده فنی مهندسی، دانشگاه شاهد، تهران، ایران

### چکیده

اکنون نیاز به پردازش ابربی درنگ و استفاده از الگوریتم‌های موازی، سریع و کارا در ردیابی همزمان اشیاء برای پردازش سریع داده‌های عظیم تصویری اجتناب‌ناپذیر است. در این مقاله روش موازی مبتنی بر توصیف‌کننده‌های باینری محلی برای ردیابی همزمان ۵۰ شیء با سرعت بی‌درنگ ارائه می‌شود. روش پیشنهادی ضمن مدل‌سازی چندمنحی ردیابی و تشخیص اشیاء با استفاده از مدل FREAK در تصاویر ویدئویی، عمل ردیابی همزمان بی‌درنگ برخط را انجام می‌دهد. نتایج آزمایش‌های تجربی در تصاویر با ۳۳۰ فریم بر ثانیه نشان می‌دهد که این روش ضمن برخورداری از سرعت زیاد و ردیابی همزمان اشیاء با سناریوهای مختلف در مقایسه با چهار الگوریتم مهم توصیف‌کننده، دارای بهترین کارایی و عملکرد است. سیستم ردیابی سریال مبتنی بر توصیف‌کننده FREAK، قادر به ردیابی یک شیء با سرعت ۶۰ فریم بر ثانیه است درحالی‌که در ردیابی موازی پیشنهادی با سرعت ۳۳۰ فریم بر ثانیه و تا ۵.۵ برابر سریع‌تر ردیابی می‌کند. همچنین این ردیاب قادر به ردیابی بی‌درنگ در حالت سریال حداکثر تا ۷ شیء همزمان و در حالت موازی قادر به ردیابی ابربی درنگ تا ۵۰ شیء است.

**کلمات کلیدی:** ردیابی همزمان اشیاء، الگوریتم‌های موازی، پردازش تصاویر ویدئویی، توصیف‌کننده‌های باینری، داده‌های عظیم تصویری.

### ۱- مقدمه

شیء متحرک در هر فریم ویدئویی است [۱]. ردیابی اهداف متحرک در طیف مرئی<sup>۷</sup> و نامرئی<sup>۸</sup> در دسته‌بندی‌های مختلفی توسعه یافته است. روش‌هایی هم وجود دارند که بر اساس موقعیت و اطلاعات جغرافیایی هدف، عمل ردیابی آن را انجام می‌دهند. در طیف مرئی که بیشتر مرتبط با موضوع سامانه‌های ردیابی مبتنی بر بینایی است غالباً از دوربین ویدئویی استفاده می‌شود. اما در طیف نامرئی، حسگرهای رادار<sup>۹</sup> و لیدار<sup>۱۰</sup> به کار می‌رود. ردیابی هدف در سامانه‌هایی که وظایف کنترل و مراقبت و راهنمایی را به عهده دارند از بخش‌های ضروری سامانه محسوب می‌شود. گوشی‌های همراه و ابزارهای قابل حمل<sup>۱۱</sup> بخش قابل ملاحظه‌ای از سیستم‌های تصویری امروزی را تشکیل می‌دهند. نمی‌توان از الگوریتم‌های پیچیده بر روی این سیستم‌ها براحتی استفاده کرد. زیرا این سیستم‌ها منابع لازم برای این کار را ندارند و میزان توان مصرفی پردازنده مسأله بسیار مهمی است. این موضوع سبب شده که الگوریتم‌هایی ابداع شوند که دارای پیچیدگی محاسباتی کمی باشند و به‌صورت بهینه عمل کنند تا در اجرا توان زیادی در پردازنده مصرف نکنند. در بینایی ماشین نیز الگوریتم‌هایی مبتنی بر الگوهای محلی باینری<sup>۱۲</sup> برای این گونه ابزار ابداع شده که از نظر محاسباتی دارای پیچیدگی زیادی نیستند و قابل اجرا بر روی پردازنده‌های با توان مصرفی پایین هستند [۳]. اگر از آن‌ها در پردازنده‌های مرسوم استفاده شود نسبت به الگوریتم‌های مشابه، دارای عملکرد

در دنیای کنونی سرعت پیشرفت فناوری بسیار زیاد است و حجم اطلاعاتی که در این دوره زمانی تولید می‌شود، نسبت به دوره‌های مشابه در دهه‌های قبل، بسیار بیشتر است. حجم عظیمی از اطلاعاتی که اکنون تولید می‌شوند، اطلاعات چندرسانه‌ای<sup>۱</sup> دیجیتال است و توسط دوربین‌های ویدئویی دیجیتال تولید و منتقل می‌شوند. هزینه دوربین‌های ویدئویی و فضاهای ذخیره‌سازی دیجیتال به‌سرعت در حال کاهش است و هم‌اکنون در میزان اطلاعات ویدئویی ذخیره شده در جهان، افزایش چشمگیری مشاهده می‌شود. اکنون در هر دقیقه ۱۰۰ ساعت فایل ویدئویی به سایت یوتیوب<sup>۲</sup> بارگذاری می‌شود [۱]. لذا تقاضای روزافزون پردازش خودکار و سریع این حجم عظیم از اطلاعات ویدئویی در سراسر جهان وجود دارد. تحلیل خودکار اطلاعات تصاویر دیجیتال و تصاویر ویدئویی بصورت خودکار، در تشخیص و تحلیل حرکت اشیاء از مباحث مهم است که در کاربردهای متعددی استفاده می‌شود. از جمله کاربردهای تحلیل و تشخیص حرکت، ردیابی اشیاء پرنده و یکی از اساسی‌ترین فرایندهای درک و پردازش اطلاعات ویدئویی است. ردیابی<sup>۴</sup> در اصطلاح لغوی به معنی تعقیب و دنبال کردن هدف یا شیء است و فرایند ردیابی بصری شیء<sup>۵</sup>، فرایند یافتن مکان و تنظیمات پویای<sup>۶</sup> یک یا چند

کارایی سیستم است. معیارهای این ارزیابی عبارت از (۱) مقایسه صحت مکان‌های ردیابی شده توسط اطلاعات واقعی و با معیار، (۲) مقایسه بهبود سرعت ردیابی در دو حالت پیاده‌سازی و اجرا در CPU<sup>۳۱</sup> و GPU<sup>۳۲</sup>، و (۳) مقایسه عملکرد توصیف-کننده‌ها با عملکرد الگوریتم SIFT<sup>۳۳</sup> به‌عنوان معیار بررسی است.

### ۳-۱ الگوریتم‌های ردیابی

از الگوریتم‌های مبتنی بر الگوهای محلی باینری می‌توان در عملیات ردیابی اشیاء استفاده کرد [۳]. در این صورت از نظر سرعت اجرای الگوریتم ردیابی، نسبت به ردیاب‌های مبتنی بر سایر الگوریتم‌ها، عملکرد برنامه سریع‌تر است [۴]. اما در فرایند ردیابی اشیاء متحرک علاوه بر این که سرعت ردیابی حائز اهمیت است دقت ردیابی نیز باید بالا باشد. در بعضی کاربردها همانند رابط‌های کاربری بین انسان و ماشین<sup>۳۴</sup>، سیستم باید قادر باشد همزمان<sup>۳۵</sup> و به‌صورت بی‌درنگ<sup>۳۶</sup> با دقت بالا چند شیء را ردیابی کند. این عوامل دلیل آن است که در طراحی سیستم ردیابی همزمان چند شیء مصالحه<sup>۳۷</sup> بین سرعت، دقت، تعداد اهداف همزمان و میزان پیچیدگی الگوریتم صورت می‌گیرد و سرعت یا دقت فدای<sup>۳۸</sup> دیگری می‌شود (شکل ۱).

با استفاده از محاسبات موازی<sup>۳۹</sup> می‌توان هر دو عامل دقت و سرعت را با همدیگر بهبود داد. محاسبات موازی در کاربردهای محاسباتی، ابزاری قدرتمند و از علل اصلی توسعه سریع پردازنده‌های گرافیکی است. اکنون استفاده از آنها در گوشی‌های همراه و سایر ابزار دنیای مدرن ضروری است. این پردازنده‌ها، شامل تعداد زیادی هسته محاسباتی<sup>۴۰</sup> با نخ‌های کنترلی<sup>۴۱</sup> فراوان هستند و نسبت به پردازنده‌های تک‌هسته‌ای، قدرت پردازشی بسیار بیشتری را برای برنامه‌نویس فراهم می‌کنند. اکنون در بسیاری از برنامه‌های کاربردی برای بررسی همزمان چندین متغیر از این پردازشگرها استفاده می‌شود [۵]. استفاده از پردازنده‌های با هسته‌های زیاد<sup>۴۲</sup> در زمینه کاربردهای مربوط به ردیابی، دارای پتانسیل فراوانی می‌باشد که سرعت اجرای بسیاری از الگوریتم‌های متداول افزایش داده است [۶]. با استفاده از این پردازنده‌ها و تکنیک‌های محاسبات موازی در الگوریتم‌های مربوط به ردیابی اشیاء متحرک، می‌توان از قدرت و مزایای پردازش موازی بهره‌مند گردید بدون این که دقت فرایند ردیابی کاهش یابد. سرعت اجرای الگوریتم‌های موجود به‌نحو قابل ملاحظه‌ای بهبود می‌یابد. چند شیء متحرک با سرعت‌های بالا حرکتی به‌راحتی ردیابی شوند و عملیات ردیابی چند هدفه با سرعت و دقت بالا انجام می‌شود. لذا تسریع و موازی‌سازی الگوریتم‌های قابل اعتماد موجود از کارهای مهم و در دست تحقیق است.

هدف این مقاله ارائه الگوریتم سریع و موازی ردیابی همزمان اشیاء متحرک با استفاده از معماری چندهسته‌ای پردازشگرهای گرافیکی جهت دستیابی به کارایی زیاد، سرعت بالا و مقیاس‌پذیر است. پیاده‌سازی الگوریتم ردیابی مبتنی بر استخراج ویژگی‌های باینری محلی<sup>۴۳</sup> اشیاء هدف برای ردیابی می‌باشد که در دو بخش استخراج ویژگی‌های پردازنده‌های گرافیکی استفاده شده است. معماری موازی کارا جهت ردیابی بهینه دارای سرعت زیاد پیاده‌سازی و ارائه شده است.

### ۴-۱ روش پیشنهادی ردیابی

سیستم ردیابی همزمان چند شیء در شکل ۲ بر اساس توصیف‌کننده محلی باینری است و پیاده‌سازی بر روی پردازنده گرافیکی صورت خواهد گرفت. الگوریتم را در پنج بخش تعریف شبکه جستجو، توصیف نقاط کلیدی، امتیازدهی نقاط کلیدی، انتخاب نقاط کلیدی برنده و سازگاری با مقیاس نشان می‌دهد. موارد تعریف شبکه جستجو، دیکشنری توصیف‌کننده‌ها، توصیف نقاط نامزد برای مقایسه و تطبیق بر اساس توصیف‌کننده محلی باینری در ادامه بیان می‌گردد. عملیات

بهتری هستند و از زمان اجرای کمتری برخوردارند. همچنین باید برای پردازش داده‌های عظیم تصویری<sup>۴۴</sup> آینده جهت پردازش سریع فریم‌های زیاد ورودی به سمت پردازش ابربی‌درنگ<sup>۴۵</sup> الگوریتم‌های موازی با پیاده‌سازی توزیعی و موازی چندهسته‌ای<sup>۴۶</sup>، چندپردازندگی<sup>۴۷</sup> در قالب پردازش ابری<sup>۴۸</sup> و مدل‌های عامل‌گرا<sup>۴۹</sup> با عامل‌های هوشمند<sup>۵۰</sup> حرکت کنیم [۴-۶].



شکل ۱: ردیابی بی‌درنگ اهداف همزمان متحرک.

### ۱-۱ ردیابی همزمان

عبارت ردیابی تصویری به عمل دنبال کردن همزمان یک یا چند شیء<sup>۵۱</sup> متحرک در یک صحنه گفته می‌شود [۷]. شیء می‌تواند هر موجودیت موردنظر در صحنه ردیابی باشد. شیء متحرک در هر فریم در بخشی از تصویر قرار دارد که باید شناسایی و تعیین موقعیت شود. نحوه شناسایی آن وابسته به فضای ردیابی و کاربرد موردنظر دارد. معمولاً در محیط پویا، پس زمینه<sup>۵۲</sup> و اشیاء تغییر می‌کنند. حل این مسأله بدون وجود هیچ قید و شرط در عمل سخت است. می‌توان مجموعه‌ای از فرضیات را طرح کرد تا مسأله قابل حل شود. با افزایش تعداد فرضیات، حل مسأله سریع‌تر و آسان‌تر است ولی درصد خطای تشخیص و ردیابی بیشتر می‌شود. بعضی از فرضیات برای حل مسئله ردیابی شیء عبارت از حرکت نرم و آهسته<sup>۵۳</sup> شیء بدون هیچ نوع تغییر ناگهانی<sup>۵۴</sup>، عدم وجود تغییرات زیاد در پس‌زمینه، تغییرات آهسته و تدریجی<sup>۵۵</sup> در ظاهر شیء<sup>۵۶</sup>، ثبات وضعیت دوربین، محدودیت تعداد و اندازه اشیاء، و عدم وجود جفت شدن و انسداد<sup>۵۷</sup> در تصاویر اشیاء است.

#### ۲-۱ چالش‌های ردیابی

هنوز چالش‌های زیاد در مسئله ردیابی اشیاء وجود دارد که عبارت از حرکت ناگهانی شیء، وجود نویز تصویر، تغییرات روشنایی صحنه، تغییر الگوی ظاهری و شکل اشیاء، تغییرات صحنه، جفت شدن شیء با شیء و شیء با صحنه، ساختارهای غیر صلب و غیر حکم<sup>۵۸</sup>، حرکت دوربین، پردازش توزیعی و پردازش ابربی‌درنگ است. هر الگوریتم جدید تشخیص و ردیابی شیء متحرک<sup>۵۹</sup> باید این مشکلات را با مجموعه پیش‌فرض‌هایی در کاربرد خود در نظر بگیرد تا بتواند به فرایند ردیابی شیء موردنظر خود بپردازد. به‌عنوان مثال الگوریتم‌هایی که در زمینه حمل‌ونقل عمومی ارائه می‌شوند از مجموعه ساختارهای مبتنی بر ظاهر و شکل<sup>۶۰</sup> استفاده می‌کنند [۸]. الگوریتم‌هایی که برای ردیابی انسان عرضه می‌شوند از ساختارهای دیگری بهره می‌برند [۹]. در زمینه ردیابی اشیاء متحرک، عامل سرعت ردیابی از چالش‌های اساسی به شمار می‌رود که در ردیابی همزمان چند شیء، در میزان کارایی سیستم بسیار تأثیر گذار است [۱۰].

چالش‌های متعددی در زمینه طراحی و کاربرد الگوریتم‌های ردیابی بصری اشیاء وجود دارد. این چالش‌ها عبارت از (۱) دقت الگوریتم در شناسایی درست و صحیح شیء، (۲) مقاومت الگوریتم در مقابل تغییرات مکان شیء مانند جابه‌جایی و حرکت آن، (۳) مقاومت الگوریتم در مقابل تغییرات وضعیت شیء مانند دوران و بزرگ و کوچک شدن آن یا دور و نزدیک شدن به دوربین، (۴) هم‌زمانی ردیابی اشیاء متحرک، (۵) سرعت پردازش فایل ویدئویی برحسب فریم بر ثانیه، (۶) سرعت ردیابی هر شیء برحسب فریم بر ثانیه، (۷) سرعت ردیابی چند شیء برحسب فریم بر ثانیه، (۸) حداکثر اهداف قابل ردیابی بی‌درنگ همزمان و (۹)

ردیابی در سه مرحله ساخت توصیفگر، استخراج ویژگی و مکان‌یابی انجام می‌شود. در اولین مرحله، توصیفگرهای باینری محلی اشیاء موجود در تصویر استخراج می‌شوند.

## ۱-۶ ساختار مقاله

در بخش ۲ کارهای مرتبط در ردیابی هم‌زمان اشیاء تشریح می‌شود و در بخش ۳ پیش‌نیاز ردیابی و روش پیشنهادی ارائه می‌شود. بخش ۴ به نتایج آزمایش‌های تجربی اشاره می‌کند و نهایتاً بخش ۵ به نتیجه‌گیری می‌پردازد.

## ۲- کارهای مرتبط

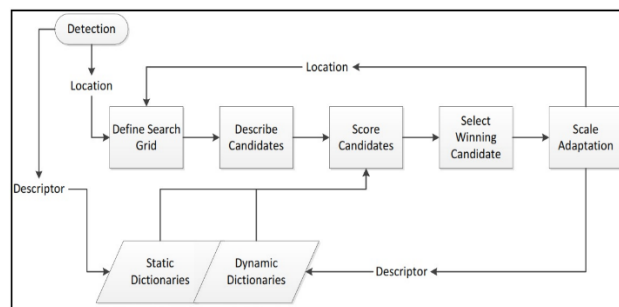
روش‌های متعددی برای ردیابی بصری بر اساس الگوی تشخیص و سپس ردیابی وجود دارد که به سیستم اجازه دهند هر نوع شیء به صورت تصادفی و فقط با مقداردهی اولیه تعقیب کنند. عیب اصلی این روش‌ها اتکا به مدل‌سازی شیء مورد نظر برای تعقیب هدف است. محدودیت دیگر آن‌ها محاسبات زیاد و پیچیده آنها است که مستلزم مصرف حافظه زیاد می‌باشد. این عیوب مانع از کاربرد آنها برای محیط‌های دارای منابع محدود مانند ابزارهای موبایل می‌شود که مناسب پیاده‌سازی یک سیستم پیچیده و عملیاتی نیست. برای ارائه یک راهکار مناسب، نقش توصیف‌کننده‌ها برای ردیابی اشیاء را بیان می‌کنیم. سپس به معرفی و بررسی تعدادی از الگوریتم‌های مهم توصیف‌کننده می‌پردازیم و الگوریتم‌های ردیابی بر اساس این توصیف‌کننده‌ها را بیان می‌کنیم.

## ۲-۱ توصیف‌کننده‌ها

تمامی برنامه‌های کاربردی دارای بخش اصلی و تعیین‌کننده مربوط به استخراج توصیفگر خاص و انتخاب توصیف‌کننده مناسب هستند. سپس شناسایی اشیاء و مرحله ردیابی اشیاء صورت می‌گیرد (شکل ۳). انتخاب الگوریتم توصیف‌کننده مورد نظر متکی بر کارایی مطلوب مورد انتظار است. توصیف‌کننده‌ها را می‌توان به دو دسته باینری و ممیز شناور دسته‌بندی کرد. توصیف‌کننده SIFT توصیف‌کننده ممیز شناور و توصیف‌کننده‌های ORB، BRIEF، BRISK و FREAK باینری هستند. مقالات زیادی از ویژگی SIFT برای ردیابی استفاده کرده‌اند [۱۱-۱۲]. توصیف‌کننده SIFT به‌عنوان معیار<sup>۴۶</sup> برای مقایسه میزان صحت و دقت ردیابی شیء مورد نظر مبتنی بر توصیف‌کننده‌های محلی باینری همانند FREAK<sup>۴۷</sup> است. دقت این توصیف‌کننده‌ها در برابر سطح معیار<sup>۴۸</sup> مکان هدف مورد ردیابی و محک قرار گرفته و ارزیابی می‌شود. از مهمترین ملاحظات تصمیم‌گیری قرارداد کد نقاط تصویر در اختیار توصیف‌کننده است.

یکی از توصیف‌کننده‌ها در این زمینه تشخیص‌دهنده گوشه هریس است [۱۳]. با استفاده از این تشخیص‌دهنده، گوشه‌های یک شیء خاص در تصویر یافته و شرایطی که آن گوشه‌های دارای آنها می‌باشند به صورت عددی ارائه و بیان می‌شوند. بر اساس سطح پذیرش و قبولی این اعداد می‌توان گوشه‌ها را بررسی و متناسب کرد. این عمل توسط یک نسخه ویرایش‌شده از تشخیص‌دهنده گوشه Moravec کامل می‌شود [۱۴]. این تشخیص‌دهنده، پنجره‌های محلی یک تصویر را بررسی و میانگین شدت روشنایی آنها را در حین شیفت آنها محاسبه می‌کند. بعداً توسط هریس و همکاران تغییر کرد و بسیار قوی‌تر و کارا تر شد. اولین تغییر آنها استفاده از یک بسط آنالیزی به‌جای مجموعه کوچک به‌منظور پوشش تمام شیفت‌های کوچک است. اکنون آن نقاط مورد نظر را به‌خوبی می‌توان تشخیص داد و مکانیابی کرد و به‌خوبی توصیف نمود.

الگوریتم SIFT [۱۵] یکی از الگوریتم‌های پیشرو در زمینه توصیف‌کننده‌های محلی و مهم‌ترین و قوی‌ترین توصیف‌کننده است [۱۷] که به‌عنوان معیار ارزیابی و محک در نظر گرفته می‌شود. الگوریتم SIFT بر اساس تفاضل گاوسی<sup>۴۹</sup> و مبتنی بر DoG عمل می‌کند. تصاویر گاوسی توسط اجرای دو هسته<sup>۵۰</sup> یک‌بعدی بر روی تصویر بوجود می‌آیند که در مقابل تغییرات مقیاس، دوران و شدت روشنایی مقاوم هستند. این توصیف‌کننده به دلیل فیلتر گاوسی، نسبت به تغییرات نویز نیز مقاوم است و ابزار با دقت بالا در مقایسه یک شیء یا نقطه بین تصاویر مختلف را فراهم



شکل ۲: بلاک دیگرام عمومی روش ردیابی پیشنهادی.



شکل ۳: بستر ردیابی اشیاء مبتنی بر استخراج ویژگی‌های کلیدی.

در مرحله دوم ویژگی‌های استخراجی با یک سری از ویژگی‌های مرجع مقایسه و ضرابی به آنها اضافه می‌شود. در مرحله سوم نیز بر اساس نتایج حاصل از مقایسه ویژگی‌ها، مکان اشیاء مورد ردیابی در فریم مورد نظر، مشخص می‌گردد. در انتها دیکشنری‌ها به‌روزرسانی می‌شوند و ردیابی در فریم بعدی ادامه می‌یابد. البته به دلیل آن که روش پیشنهادی مبتنی بر توصیف‌کننده باینری محلی است برای حل مشکلات محلی بودن و تأخیر جیت<sup>۴۴</sup>، متغیرهای وزن‌دار گاوسی به‌عنوان پناهی به تابع امتیازدهی اضافه می‌شود تا حرکت آزادانه شیء در مقابل تغییر شکل را تضمین کند. همچنین باید برای یافتن شیء سریع در بیرون از ناحیه جستجو شناسی وجود داشته باشد. بعلاوه باید برای سازگاری با تغییرات مقیاس شیء و قاب جستجو، تکنیکی اعمال شود تا نتیجه‌ای مطابق توصیفگر اصلی حاصل شود و روش مقاومی ارائه گردد.

## ۱-۵ نوآوری

در ردیابی اشیاء متحرک به‌وسیله الگوریتم مبتنی بر استخراج ویژگی‌های باینری محلی، یک پیاده‌سازی و یک معماری موازی ردیابی هم‌زمان چند شیء ارائه شده است. الگوریتم بکار گرفته شده در ردیابی چندین هدف، به سرعت‌های بیشتر از بی‌درنگ دست پیدا کرده باشد. به طور خلاصه نوآوری‌های این مقاله عبارت از (۱) ارائه و پیاده‌سازی یک روش موازی و مقیاس‌پذیر برای استخراج ویژگی‌های باینری محلی؛ (۲) طراحی الگوریتم موازی پرسرعت و مقیاس‌پذیر برای ردیابی دقیق یک هدف و ردیابی هم‌زمان چندین هدف متحرک بصری؛ (۳) پیاده‌سازی الگوریتم موازی به‌صورت خط لوله<sup>۴۵</sup> بر روی پردازنده گرافیکی و تقسیم‌کار بین بخش‌های مختلف موجود در خط لوله؛ (۴) افزایش سرعت پردازش از بی‌درنگ به ابربی‌درنگ یعنی چندین برابر بی‌درنگ؛ (۵) ردیابی هم‌زمان ۷۰ شیء یا هدف متحرک با کارایی بالا؛ (۶) روش سریع پردازشی ردیابی ابربی‌درنگ با افزایش پردازش به حدود ۳۳۰ فریم بر ثانیه است. همچنین سه محصول نرم‌افزاری استخراج شده و تعدادی از آنها به پایگاه‌های عمومی دریافت نرم‌افزارها بارشده است [۳۶]. این محصول‌ها، نرم‌افزار استخراج ویژگی‌های باینری محلی به‌صورت موازی و با سرعت بالا مبتنی بر پردازنده‌های گرافیکی CUDA، نرم‌افزار ردیابی شیء متحرک به‌صورت سریال و نرم‌افزار ردیابی هم‌زمان شیء و یا اشیاء متحرک به‌صورت موازی مبتنی بر معماری پردازنده‌های گرافیکی CUDA هستند.

FAST استفاده می‌کند. تفاوت اصلی دیگر BRISK با ORB یا BRIEF این است که BRISK متکی بر الگوی جستجوی تصادفی نیست و از الگوی ساخت‌یافته منظم برای نمونه‌برداری استفاده می‌کند. گرچه از نرم برای کاهش تأثیر نویز بر روی توصیف‌کننده استفاده می‌کند. این توصیف‌کننده مانند ORB و BRIEF از طریق فاصله همینگ مقایسه می‌شود و لذا از زمان مقایسه بیشتری برخوردار نیست.

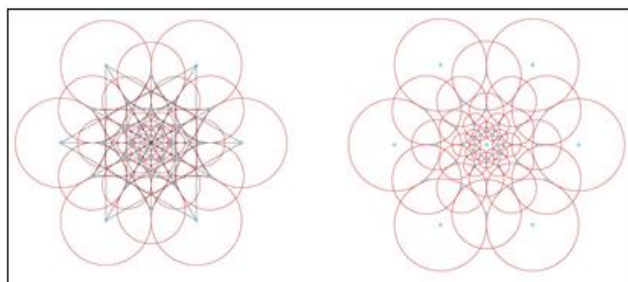
الگوریتم FREAK از کارهای جدید است که دارای ویژگی‌های مشترک با دیگر توصیف‌کننده‌ها می‌باشد [۱۷]. مانند BRISK از تشخیص‌دهنده AGAST برای تولید نقاط کلیدی استفاده می‌کند. از الگوی نمونه برداری ثابت به جای توزیع تصادفی برای نمونه‌برداری بهره می‌برد که مشابه ORB، تعداد زیادی جفت‌های احتمالی را از طریق یک الگوریتم یادگیری بازآرایی می‌کند. فرایند اصلی این الگوریتم با فرایند الگوریتم ORB دارای تفاوت است. این الگوریتم برای سنجش میزان فاصله توصیف‌کننده‌ها از فاصله همینگ استفاده می‌کند. روش ویژه آن در تطبیق دارای تفاوت‌هایی است که این الگوریتم ابتکاری را برجسته‌تر می‌کند. الگوریتم FREAK برای برخورداری از قدرت تأثیر بیشتر ایده‌های جدیدی با الهام از پردازش تصویر در چشم انسان به‌عنوان روش جست‌وجو ارائه کرد (شکل ۴ الف)) که در آن دایره آبی نقطه نمونه‌برداری و دایره قرمز همان نرم است. به‌منظور انتخاب ترکیب معتبر از میان مقایسه‌ها، از روش مشابه ORB بهره می‌برد که در آن تمامی ترکیبات ممکن بررسی می‌شوند. بخش‌نهایی FREAK اعمال تغییرناپذیری با دوران معادل الگوریتم BRISK است با این تفاوت که در این الگوریتم برای جهت‌یابی، به‌جای استفاده از نقاط تولید توصیف‌کننده از الگوی نمونه‌برداری متقارن استفاده می‌کند (شکل ۴ ب)). الگوریتم FREAK با ترکیب دو تسریع در سرعت حاصل از تطبیق و محاسبات خود به الگوریتم کارآمدی تبدیل شد.

الگوریتم SIFT از پیچیدگی محاسباتی بالایی برخوردار و مبتنی بر اعداد ممیز شناور است لذا از آن نمی‌توان در کاربردهای بی‌درنگ استفاده کرد. الگوریتم BRIEF از نظر محاسباتی دارای پیچیدگی اندکی است. الگوریتم ORB ترکیبی از نسخه‌های تغییریافته الگوریتم‌های BRIEF و FAST است. الگوریتم‌های BRISK و FREAK از الگوی نمونه‌برداری مشترک استفاده می‌کنند که به دلیل برخورداری از دقت بیشتر و سرعت بهتر نسبت به نسخه مرجع الگوریتم‌های BRIEF و ORB، در کاربردهای بی‌درنگ استفاده می‌شوند. الگوریتم FREAK به دلیل استفاده از نقاط کمتر در تعیین زاویه، از الگوریتم BRISK ساده‌تر و دارای سرعت اجرای بیشتر است. گرچه این توصیف‌کننده برخلاف الگوریتم BRISK دارای آشکارساز نقاط کلیدی نیست. خلاصه مقایسه این الگوریتم‌های مهم نسبت به نوع، اندازه توصیف‌کننده و مقاومت آنها نسبت به چالش‌های مقیاس، دوران و روشنایی سراسری در جدول ۱ بیان شده و نشان می‌دهد که الگوریتم FREAK نسبت به تمام چالش‌ها مقاوم است.

## ۲-۲ الگوریتم‌های ردیابی اشیاء

ردیابی در ساده‌ترین حالت عبارت از شناسایی یک ناحیه در تصویر و سپس جستجوی مکرر آن ناحیه و تطابق در تصاویر متوالی برای پیگیری حضور اشیاء است که به معیارهای تطابق نیاز دارد. تلاش‌های زیادی انجام گرفته تا یافتن و تطبیق اشیاء موردنظر در بخش‌های تصویر به‌صورت صحیح و مناسب انجام پذیرد. روش‌های مهم میانگین انتقال<sup>۵۴</sup>، ردیابی نقطه‌ای<sup>۵۵</sup>، قالب محور<sup>۵۶</sup> و جداسازی پیش‌زمینه و پس‌زمینه<sup>۵۷</sup> وجود دارد. ردیابی نقطه‌ای عمل ردیابی یک شیء را از طریق تطابق بین نقاط در یک دنباله مدل می‌کند و در دو دسته قطعی<sup>۵۸</sup> [۲۴] و یا احتمالاتی<sup>۵۹</sup> [۲۵] قرار می‌گیرد. میانگین انتقال الگوریتم مناسبی است که برای ردیابی سازگاری یافت [۲۶-۲۷]. این الگوریتم در زمینه کاربردهای نهفته<sup>۶۰</sup> با بهینه‌سازی‌های مرتبط با GPU [۲۸] مورد استفاده قرار گرفت. تئوری آن شکل-دادن مقدار بیشینه در یک تابع چگالی است. روش‌های قالب‌محور، قالبی می-

می‌کند [۱۶-۱۵]. مشکل عمده الگوریتم SIFT سرعت محدودکننده آن در ردیابی بی‌درنگ است که بیشتر ناشی از محاسبات مبتنی بر پایه ممیز شناور است.



شکل ۴: الگوی نمونه‌برداری و جهت‌یابی FREAK [۱۷].  
(الف) الگوی نمونه‌برداری (ب) الگوی جهت‌یابی

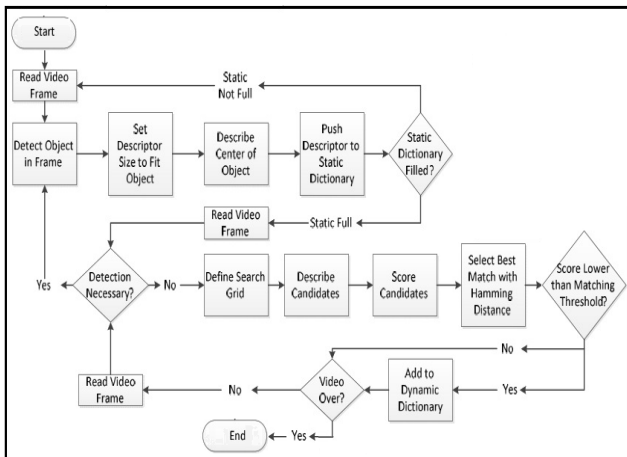
جدول ۱: مشخصات و معیارهای مقاومت توصیف‌کننده‌های مهم.

توصیف‌کننده	نوع	اندازه مرسوم توصیف‌کننده	معیارهای مقاومت
SIFT	اعشاری	۱۲۸ بعد	مقیاس، دوران و روشنایی سراسری
BRIEF	باینری	۲۵۶ بیتی	روشنایی سراسری
ORB	باینری	۲۵۶ بیتی	دوران و روشنایی سراسری
BRISK	باینری	۵۱۲ بیتی	مقیاس، دوران و روشنایی سراسری
FREAK	باینری	۵۱۲ بیتی	مقیاس، دوران و روشنایی سراسری

برای حل مشکل سرعت الگوریتم SIFT تلاش‌های زیادی انجام گرفت. الگوریتم SURF<sup>۶۱</sup> یک توصیف‌کننده کاملاً جدید آن است [۱۸]. همچنین الگوریتم [۱۹] که یک نسخه تغییریافته از الگوریتم SIFT برای موبایل‌ها است. معرفی ویژگی‌های محلی باینری به تولید توصیف‌کننده‌های باینری منجر شد. این توصیف‌کننده‌ها با استفاده از مجموعه مقایسه کننده‌های شدت روشنایی یک رشته باینری تولید می‌کنند. این رشته حاصل از الحاق نتایج تمامی مقایسه‌ها است و این امر به توصیفگرهای باینری اجازه می‌دهد که از طریق فاصله همینگ مقایسه شوند. این روش مقایسه و امتیازدهی، می‌تواند بهبود قابل‌ملاحظه سرعت نسبت به فاصله مرسوم اقلیدسی و یا هر الگوریتم خوشه‌بندی مرسوم را فراهم کند. الگوریتم BRIEF<sup>۶۲</sup> به‌عنوان روش مستقیم مولد رشته باینری ارائه شد [۲۰]. این روش با بررسی یک بخش از تصویر و انجام مقایسه‌های میزان شدت روشنایی بر روی مجموعه‌ای از پیکسل‌های خاص انجام می‌شود. اگر نتیجه هر مقایسه به‌عنوان یک-بیت در نظر گرفته شود الحاق این مقادیر به‌یکدیگر منجر به ساخت عنصر توصیفگر می‌شود. متأسفانه الگوریتم BRIEF به‌تنهایی هیچ جهت و یا مقاومت‌پذیری در برابر مقیاس‌پذیری ندارد زیرا فقط شامل یک‌سری مقایسه ساده است. این توصیف‌کننده متکی بر تشخیص‌دهنده خارجی است که در مقابل تغییرات جهت و مقیاس بسیار مقاوم و تغییرناپذیر باشد.

BRIEF نسبت به دوران ضعف دارد که این ضعف با الگوریتم BRIEF جهت‌دار به نام oFAST که تغییریافته الگوریتم FAST است بهبود می‌یابد [۲۱-۲۲]. کاهش تمایزپذیری توصیفگر منجر به کاهش توانایی یافتن تطابق‌های درست در بین توصیفگرها و نقاط می‌شود. به‌منظور دوام BRIEF جهت‌دار، الگوریتم یادگیری برای انتخاب تست‌های باینری با بیشترین تغییرات و کمترین همبستگی مشترک به نام rBRIEF ارائه گردید. با ترکیب الگوریتم‌های oFAST و rBRIEF الگوریتم ORB طراحی شد. الگوریتم BRISK<sup>۶۳</sup> [۲۳] برخلاف ORB، بهبودیافته روش جدیدی است. برای تولید نقاط کلیدی، مشابه با ORB و متکی به پیاده‌سازی FAST است که از نسخه بهبودیافته FAST به نام AGAST برای بهبود سرعت بهبود استفاده می‌کند. این الگوریتم برخلاف ORB برای مرتب‌سازی نقاط کلیدی‌اش متکی به امتیاز گوشه‌های هسین نیست و برای یافتن نقاط قابل‌نگهداری از کسب بیشترین امتیاز

الگوریتم با مجموعه‌ای از قالب‌ها که نماینده تصویر هستند داده‌رسانی می‌شود که با استفاده از این دیکشنری فرایند ردیابی شیء صورت می‌گیرد.



شکل ۵: دیاگرام بلاکی ردیابی پیشنهادی اشیاء مبتنی بر توصیف‌کننده باینری محلی.

### ۳- روش پیشنهادی

در این بخش ابتدا ردیابی مبتنی بر توصیف‌کننده‌های باینری محلی بیان خواهد شد. سپس ردیابی سریال مبتنی بر توصیف‌کننده مورد بحث قرار می‌گیرد و عملکرد توصیف‌کننده‌های متفاوت بررسی می‌شود. در نهایت الگوریتم ردیابی موازی پیاده‌سازی شده بر روی پردازنده‌های گرافیکی<sup>۶۱</sup> بررسی می‌گردد.

### ۳-۱- ردیابی مبنایی سریال

سیستم ردیابی بر پایه توانایی تطبیق توصیف‌کننده‌های باینری طراحی شده تا با سرعت بی‌درنگ فرآیند ردیابی صورت گیرد. باید بستری مناسب برای پردازش توصیف‌کننده‌ها طراحی شود تا قادر به تطبیق و ردیابی باشند. دیاگرام بلاکی ردیابی دارای بخش‌های مجزا و در شکل ۵ مشاهده می‌شود. بخش‌های بالایی دیاگرام، برای تشخیص و یافتن هدف است که معمولاً فقط یکبار در تصویر ویدئویی مورد بررسی، اجرا می‌شود. لذا در هر جریان ویدئویی فقط با یک فریم تصویر مورد ردیابی اجرا شود. بخش‌های پایین شکل ۵ مربوط به ردیابی در تمامی فریم‌ها است. به محض کامل شدن دیکشنری ایستا، مکان شیء موردنظر را مشخص و شروع به ردیابی می‌کند. برای ردیابی، شبکه جستجو در اطراف آن محل تعیین می‌شود. سپس نقاط درون شبکه توصیف با نقاط داخل دیکشنری تطبیق می‌گردد. نقاط نامزد دارای پایین‌ترین امتیاز در فاصله همینگ به‌عنوان نقاط نظیر و مورد تطبیق پذیرفته می‌شوند و این فرآیند تکرار می‌شود. الگوریتم ردیابی در بخش‌های تعریف شبکه جستجو، توصیف نقاط کلیدی، امتیازدهی نقاط کلیدی، انتخاب نقاط کلیدی برنده و سازگاری با مقیاس انجام می‌شود. البته تغییراتی برای حل مشکل محلی بودن توصیف‌کننده و رفع تأخیر جیتر، انجام شده تا حرکت آزادانه شیء تضمین شود؛ برای یافتن شیء سریع در بیرون از ناحیه جستجو شانس وجود آید؛ برای سازگاری با تغییرات مقیاس شیء و قاب جستجو نتیجه‌ای مطابق توصیفگر اصلی حاصل شود؛ و روش مقاومی ارائه گردد. این عملیات در دو فاز آشکارسازی و ردیابی مطابق الگوریتم ۱ برای آشکارسازی شیء و الگوریتم ۲ برای ردیابی شیء انجام می‌شود.

الگوریتم ۱: گام‌های فاز آشکارسازی ردیابی توصیف‌کننده باینری محلی.

- | Input: Video Frame  |
|---|
| 1. Find desired <i>Object</i> .   |
| 2. Resize the <i>Descriptor</i> to match the <i>Object Size</i> .           |
| 3. Describe the <i>Object</i> from <i>Center Point</i> .                    |
| 4. Add the <i>Descriptor</i> to the appropriate <i>Dictionary Element</i> . |
| 5. Repeat until <i>Frame Number</i> exceeds the <i>Dictionary Size</i> .    |

سازند که نماینده شیء و بخشی از تصویر است و حاوی شیء می‌باشد. با استفاده از اطلاعات نهفته در این قالب، هر شیء در فریم‌های متوالی ویدئویی تعقیب می‌شود. روش‌های مختلفی از ردیابی‌های مبتنی بر قالب در طی سالیان متوالی بوجود آمدند. قالب‌ها می‌توانند با استفاده از شدت روشنایی پیکسل‌ها پیاده‌سازی شوند [۲۹-۳۱]. با این حال می‌توان قالب‌ها را با استفاده از توصیف‌کننده‌ها نیز ساخت که در آن صورت این قالب‌ها در برابر مقیاس و دوران مقاومت خواهند داشت [۳۲]. از روش‌های جدید جداسازی پیش‌زمینه و پس‌زمینه به‌عنوان مساله‌های طبقه‌بندی یاد می‌شود. با جداسازی پس‌زمینه از پیش‌زمینه، تغییرات کوچک در شیء می‌تواند کنترل شود. یعنی به‌آرامی به طبقه‌بندی‌کننده پیش‌زمینه اضافه می‌شوند و تغییرات کوچک شیء همواره به طبقه‌بندی پیش‌زمینه نزدیک‌تر و از پس‌زمینه دورتر باشد.

اهمیت ردیابی به‌صورت مؤثر و قوی منجر به خلق روش‌های مختلف زیادی شد. یکی از نواحی اصلی تحقیقات، تحلیل آماری به‌عنوان روش مؤثر ردیابی است. در این زمینه الگوریتم CAM SHIFT<sup>۶۱</sup> توجه زیادی را به خود جلب کرد [۲۷]. این الگوریتم شبیه الگوریتم MEAN SHIFT است و بر اساس توزیع‌های احتمالاتی عمل می‌کند و سعی بر یافتن نقطه ماکزیمم توزیع دارد. پیچیدگی الگوریتم CAM SHIFT از مرتبه  $\alpha N^2$  است که در آن  $N$  ابعاد تصویر می‌باشد. به این دلیل میزان کاهش سرعت حاصل از استفاده الگوریتم CAM SHIFT برای تصاویر داده‌های عظیم امروزی بسیار زیاد است. روش محبوب دیگر با نتایج ردیابی عالی روش MIL<sup>۶۲</sup> یا روش یادگیری چندتمثیلی است [۳۳]. این ردیاب بر اساس تشخیص شیء و مبتنی بر تحلیل‌های آماری است. این الگوریتم به‌علت استفاده از نمونه‌های مختلف در فرایند یادگیری، از دقت بالایی در ردیابی برخوردار است. در آن چگونگی تعریف شیء می‌تواند منجر به ابهام و به اشتباه افتادن تشخیص‌دهنده شود. به‌روزرسانی مستمر مدل شیء در ردیابی MIL این امکان را فراهم می‌آورد تا بدون هرگونه نظارت ضروری، تغییر طبیعت هر شیء را کنترل و مدیریت کند. این ویژگی آن را برای به‌عنوان روش ردیابی قوی و ایده‌آل مطرح می‌سازد هرچند که پیچیده است ولی قابلیت توسعه برای افزایش سرعت را دارد.

الگوریتم آشکارساز یادگیری ردیابی TL<sup>۶۳</sup> [۱۰] یک الگوریتم ردیابی و تشخیص‌دهنده است و بین دو قسمت ردیابی و تشخیص آن تعاملات وجود دارد. نکته جالب در سه مؤلفه اصلی ردیابی، یادگیری و شناسایی آن است که در فرایند پردازش تصویر کاری طولانی و زمان‌بر می‌باشد. شیء موردنظر در هر فریم تعقیب می‌شود. با توجه به اطلاعات به‌دست آمده از ویژگی‌های محلی هدف، فرآیند یادگیری انجام و در صورت‌نیاز به اصلاح پارامترهای آموزش، اقدام لازم صورت می‌گیرد. این الگوریتم به‌دلیل استفاده از مفاهیم آموزش‌ماشینی، دارای دقت مناسب است و برای ردیابی‌های طولانی مدت استفاده می‌شود. برای تعقیب شیء از بخش‌های محلی تصویر استفاده می‌کند و کل تصویر مورد ارزیابی قرار نمی‌گیرد. می‌توان نتیجه گرفت با توصیف محلی تصویر و استفاده از توصیف‌کننده محلی به‌جای توصیف‌کننده سراسری<sup>۶۴</sup>، این الگوریتم در بازشناسی مورد استفاده قرار می‌گیرد و در ردیابی اشیاء بصری کاربرد ندارد. الگوریتم SURF [۳۴] نمونه‌عالی از توصیف‌کننده‌ای است که برای ردیابی، بازشناسی پیوسته و مداوم شیء<sup>۶۵</sup> از آن استفاده می‌شود و از نکات اصلی آن جداسازی توصیف‌کننده ویژگی‌ها از آشکارساز نقاط می‌باشد. این الگوریتم از آشکارساز نقطه‌ای بهره می‌برد ولی مرحله توصیف ویژگی‌های آن نقاط که معمولاً سایر الگوریتم‌ها انجام می‌دهند را حذف می‌کند.

توصیف‌کننده SURF فقط به‌عنوان یک ابزار بازشناسی شیء مورد استفاده قرار می‌گیرد. لذا این توصیف‌کننده‌ها می‌توانند برای بازشناسی مکرر یک شیء مورد استفاده قرار گیرند و برای ردیابی کاربرد ندارند. در الگوریتم DT<sup>۶۶</sup> از روش DML [۳۲] و استفاده از توصیف‌کننده‌ها در تنظیمات دیکشنری و سایر تکنیک‌ها استفاده شده است. الگوریتم این ردیاب از ITML<sup>۶۸</sup> استفاده می‌کند. دیکشنری این

جلوگیری اثر رانش و جفت‌شدگی استفاده از حد آستانه است که ورود به مجموعه دیکشنری را کنترل کند. با محاسبه فاصله همینگ  $dH(c_i, t_i)$  میزان امتیاز طبق رابطه (۱) محاسبه می‌شود که در آن  $c_i$  کاندیدای موردنظر در شبکه جستجو و  $t_i$  الگوی بی‌تی در دیکشنری ایستا است. مقایسه این مقدار با حد آستانه منجر به ایجاد مجموعه دیکشنری جدید می‌شود.

$$S_H^i = \operatorname{argmin}_{(t_i \in D)} \{d_H(c_i, t_j)\} \quad (1)$$

$$D = \begin{cases} t_0 = C_{win} \text{ and } t_1 \dots t_n = t_0 \dots t_{n-1} & \text{if } S_H^{win} \leq \text{Threshold} \\ D & \text{if } S_H^{win} > \text{Threshold} \end{cases} \quad (2)$$

با استفاده از رابطه (۲) می‌توان جلوی مشکل توقف ردیابی به دلیل جفت‌شدگی را گرفت. زیرا فاصله همینگ شیء جفت شده بسیار بیشتر از مقداری است که به عنوان مقدار تطبیق صحیح در نظر گرفته شده است. این رابطه از مشکل رانش نیز جلوگیری می‌کند زیرا هر کاندید انتخابی که با شیء اصلی بسیار متفاوت باشد و فاصله همینگ زیادی داشته باشد به دیکشنری وارد نخواهد شد. مقایسه کننده مبتنی بر FLANN در حالت پیش فرض ویژگی‌های ورودی را با نرم L2 پردازش و مقایسه می‌کند که به صورت عادی نمی‌توان از آن برای مقایسه توصیفگرهای باینری استفاده کرد. لذا از مقایسه کننده‌های سازگار با فاصله همینگ استفاده می‌شود.

**ناحیه مورد جستجو:** منطقه جستجو از مهم‌ترین بخش‌ها از نظر کارایی و بهره‌وری الگوریتم ردیابی است. ممکن است دقت توسط یکتایی توصیف‌کننده تعیین شود ولی بدون یک منطقه جستجوی دارای اندازه مناسب، هر شیء سریع ممکن است یافت نشود و عملیات ردیابی متوقف گردد. لذا ناحیه جستجو به صورت سلسله‌مراتبی تصور و مورد بررسی قرار می‌گیرد. از تخمین مکان شیء استفاده نشده و با حد آستانه جلوی تطبیق‌های غلط گرفته می‌شود. افزایش منطقه جستجو می‌تواند هزینه زیادی بر سرعت الگوریتم تحمیل کند. بنابراین بدون تعریف منطقه جستجوی مناسب، به دلیل امتناع از جستجو در فاصله به اندازه کافی دور ممکن است ردیاب هدف را از دست بدهد. برعکس اگر فضای جستجو بزرگ باشد سرعت به پایین‌تر از سرعت بی‌درنگ نزول و ممکن است بعضی فریم‌ها نادیده گرفته شوند لذا شیء هدف از دست برود.

راه‌حلی مبتنی بر جستجوی نرم به سخت (ریز به درشت) برای اجرای کارآمد ردیاب در زمان ردیابی اشیاء سریع توسعه داده شد. نتیجه آن شبکه‌ای از نقاط موردنظر برای جستجو است که در مرکز این شبکه، نقاط با فاصله کم از یکدیگر (نرم و دارای فاصله اندک) بررسی می‌شوند و در نواحی خارجی شبکه، با فاصله‌ای در بین نقاط جستجو (درشت و دارای فاصله بیشتر) انجام می‌گیرد. اساساً انتظار نداریم که شیء در هر فریم به ناحیه خیلی دور حرکت کند. لذا جستجوی دقیق و ظریف لازم است. برای در نظر گرفتن جنبش و حرکت‌های سریع با جابه‌جایی زیاد، تعریف ناحیه جستجوی بزرگ با نمونه‌های زیاد، مؤثر نیست. بنابراین منطقه جستجو در دو ناحیه ظریف و درشت، ظریف دارای نقاط با فواصل کم و درشت دارای نقاط با فواصل زیاد تعریف می‌شود. ناحیه ظریف یک ناحیه کوچک در اطراف مرکز ناحیه جستجو است. ناحیه درشت از یک فاصله معین از نقطه مرکزی ناحیه جستجو شروع می‌شود. به سادگی طول بین نقاط در ناحیه جستجو افزایش می‌یابد. ناحیه جستجو به صورت مجموعه دلتا تعریف و در محل مرکز ناحیه جستجو بکار می‌رود. این مقادیر به محض شروع ردیابی طبق روابط (۳)، (۴) و (۵) تعریف می‌شوند که در آنها  $r$  شعاع ناحیه جستجو،  $\alpha$  ثابت گسترش مقیاس،  $i$  و  $z$  متغیرهای حلقه هستند. با استفاده از این روابط می‌توان ناحیه جستجو را به گونه‌ای تعمیم داد که بدون کاهش نرخ فریم طیف وسیعی از سرعت حرکت را کنترل کنیم.

$$\Delta_{i,j} = (i + \sigma_{x,i} + \sigma_y) \quad (3)$$

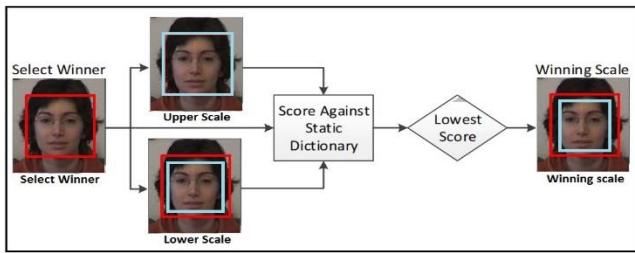
الگوریتم ۲: گام‌های فاز ردیابی ردیاب توصیف‌کننده باینری محلی.

Input: Video Frame
1. Create all Keypoints from Search Area.
2. Describe all Keypoints.
3. Compute Hamming Distance of each describe Keypoint to each Dictionary.
a. Keep the Track of the Minnum Score.
b. Add constant Value to the Score of Dynamic Dictionary to bias towards the Static Dictionary.
4. Compare the Winning Descriptor against all Descriptors.
a. If the Winning Descriptor is above the matching threshold Goto Step 1.
5. Update th Location of the Object with the Location of the Wining Descriptor.
6. Check the Upper and the Lower Scales.
a. If needed Change the Scale.
7. Monitor the Location of the Wining Descriptor.
8. Goto Step 1.

الگوریتم ۱ در ابتدا شیء موردنظر را انتخاب می‌کند. اندازه توصیف‌کننده را مطابق اندازه شیء تغییر می‌دهد. شیء موردنظر در مرکز ناحیه توصیف می‌شود. توصیف‌کننده را به دیکشنری مناسب اضافه می‌کند. کلیه عملیات ذکر شده را تا شماره فریم از اندازه دیکشنری بیشتر شود تکرار می‌کند. الگوریتم ۲ کلیه نقاط کلیدی را از ناحیه جستجو استخراج می‌کند. تمامی نقاط کلیدی را توصیف می‌کند. برای هر دیکشنری، فاصله همینگ از هر نقطه کلیدی توصیف‌شده را محاسبه می‌کند، کمینه امتیازها ثبت می‌شود و مقدار ثابتی به آن امتیاز در دیکشنری پویا برای گرایش به دیکشنری استاتیک اضافه می‌گردد. توصیف‌کننده برنده با تمام دیکشنری‌ها مقایسه می‌شود و اگر از مقدار آستانه بیشتر باشد دوباره به مرحله استخراج نقاط کلیدی بیان‌شده می‌رود. در غیر این صورت محل آن شیء را با محل این کاندیدای برنده تغییر می‌دهد. مقیاس موجود را با مقیاس پایین‌تر و بالاتر مقایسه می‌کند و در صورت لزوم مقیاس تغییر داده می‌شود. موقعیت فعلی را نمایش می‌دهد و دوباره کلیه عملیات بیان‌شده را از سر می‌گیرد. البته برای وضوح بیشتر در ادامه، مفاد تعریف شبکه جستجو، دیکشنری توصیف‌کننده‌ها، توصیف نقاط نامزد برای مقایسه و تطبیق بر اساس توصیف‌کننده محلی باینری بیان می‌گردد.

**دیکشنری توصیف‌کننده‌ها:** دیکشنری مسئول نگهداری مجموعه توصیف‌کننده‌هایی است که با بررسی‌های دقیق ایجاد شده‌اند تا خصوصیات شیء به وضوح بیان شود [۴۵]. فرض آن که در ابتدای جریان ویدئویی، شیء موردنظر دارای تغییرات محدود می‌باشد لذا از چند تصویر ابتدایی برای ساخت دیکشنری استفاده و مقادیر اولیه شده است. مسئله مهم در زمان شروع و تنظیم دیکشنری و نحوه نگهداری توصیف‌کننده‌ها در جریان ویدئویی آن که اغلب اشیاء در طول زمان دقیقاً به صورت همان شکل اولیه باقی نمی‌مانند. احتمالاً شیء مورد ردیابی به عقب و یا جلو حرکت می‌کند در نتیجه از دوربین دور و یا به آن نزدیک می‌شود (تغییر در مقیاس و اندازه شیء)، دوران می‌یابد، دچار جفت‌شدگی می‌شود، و یا شدت روشنایی آن تغییر می‌کند. این چالش‌ها را می‌توان با پرکردن دیکشنری ایستا توسط نماهای مختلف شیء، تا حدودی خنثی کرد. این روش کامل نیست زیرا حرکت شیء به احتمال زیاد در چند فریم اول تغییر زیادی نخواهد داشت. دیکشنری ایستا می‌تواند ورودی‌های جدید در ویدئو را بپذیرد. آشکارسازی شیء فرآیندی کند و اجتناب از آن برای افزایش سرعت پردازش ردیابی مفید است. به منظور مقاومت در برابر این چالش، دیکشنری به دو مجموعه مجزای ایستا و پویا تقسیم می‌شود. دیکشنری ایستا شامل توصیف‌کننده‌ها و دیکشنری پویا حاوی جدیدترین تطابق‌های یافت‌شده است. با به‌روزرسانی دیکشنری پویا در هر فریم می‌توان بر تغییرات مقیاس و دوران غلبه کرد. اقدامات نهایی برای اعتماد به این الگوریتم، نحوه تعامل با رانش‌ها و جفت‌شدگی‌های شیء در تصویر است. برای

محاسباتی (شکل ۷) است. این روش حتی زمانی که شیء دوران یافته و دستخوش برخی تغییرات شده، کارا است.



شکل ۷: نمونه تطبیق تغییر مقیاس در هر فریم.

### ۳-۲ ردیابی پیشنهادی موازی

برای پیاده‌سازی سیستم ردیابی موازی از امکانات موجود در پردازنده‌های کارت گرافیکی بهره‌برداری و از بستر CUDA برای برنامه‌نویسی تراشه‌های NVIDIA استفاده شده و با معماری خط لوله پنج‌مرحله‌ای، الگوریتم موردنظر در GPU پیاده‌سازی شده است. پنج مرحله که آنها را هسته ۷۰ می‌نامیم عبارت از هسته تصویر انتگرالی، هسته مولد توصیفگرها، هسته امتیازدهی، هسته بهترین تطبیق دیکشنری و هسته بهترین نامزد موردنظر است (شکل ۸). هسته تولید تصویر انتگرالی، تنها برای توصیف‌کننده‌هایی مانند FREAK و BRISK که در آنها برای تسریع و بهبود سرعت در انجام عملیات میانگین‌گیری از تصاویر انتگرالی استفاده می‌شود بکار می‌رود. توصیف‌کننده‌های BRIEF از هسته تولید تصویر انتگرالی استفاده نمی‌کنند. بعد از ساخت توصیف‌کننده‌های موردنظر برای تمام نقاط کاندیدا هسته امتیازدهی اجرا می‌شوند و تمامی نقاط کاندیدای موجود دیکشنری امتیازدهی می‌شوند. بهترین توصیف‌کننده موجود دیکشنری برای هر نامزد پیدا می‌شود و بهترین موارد یافت‌شده به‌عنوان توصیفگرهای متناظر بهترین کاندیدا به‌عنوان برنده فرض می‌شود. بعد مشخص شدن برنده، مکان برنده از GPU به CPU منتقل می‌شود. البته بیشتر وظایف محاسباتی بر عهده GPU است و CPU وظیفه ورود و خروج نتایج را دارد. در پیاده‌سازی نکات زیادی مانند ساختار و تنظیمات نخ‌های کنترلی وجود دارد که بیان جزئیات در راستای این مقاله نمی‌گنجد.

Image Transferred Into GPU	Integral Image Kernel	Descriptors Generation Kernel	Scoring Kernel	Dictionary Selection Kernel	Candidate Selection Kernel	Object Location Transferred Off GPU
----------------------------	-----------------------	-------------------------------	----------------	-----------------------------	----------------------------	-------------------------------------

شکل ۸: خط لوله انجام الگوریتم موازی.

### ۴- نتایج آزمایش‌های تجربی

ما نتایج آزمایش‌های تجربی خود را در سه مرحله و به این شرح ارائه می‌دهیم. ابتدا تنظیمات آزمایشات، ابزار، و مجموعه‌های داده‌ای ارائه و ساختار آنها معرفی می‌گردد. سپس معیارهای ارزیابی آزمایش‌ها و الگوریتم معرفی و در نهایت نتایج تجربی آزمایشات ارائه و تحلیل می‌شود.

#### ۴-۱ تنظیمات آزمایش‌ها

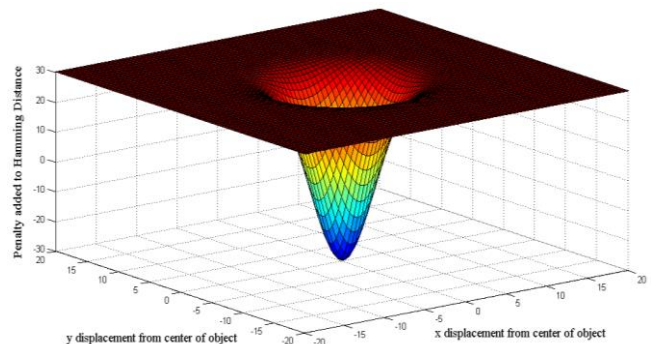
الگوریتم ردیابی سریال با زبان C++ پیاده‌سازی و صحت عملکرد آن در ویدئوهای مختلف بررسی شد. پس از اطمینان از صحت عملکرد الگوریتم ردیابی در بستر نرم‌افزاری CUDA و زبان C++ بر روی پردازنده گرافیکی به‌صورت موازی و با معماری خط لوله پیاده‌سازی شد. در نهایت مقادیر بهینه پارامترها جهت ویدئوهای مختلف و اشیاء موردردیابی بدست آمدند. سیستم در دو بخش سخت-افزاری و نرم‌افزاری پیاده‌سازی شد. در بخش نرم‌افزاری، برای پیاده‌سازی‌های

$$\sigma_x = \begin{cases} 0 & \text{if } \frac{-r}{2} < i < \frac{r}{2} \\ \alpha \left( i - \frac{r}{2} \right) & \text{if } i \geq \frac{r}{2} \\ \alpha \left( i - \frac{r}{2} \right) & \text{if } i \leq \frac{-r}{2} \end{cases} \quad (4)$$

$$\sigma_x = \begin{cases} 0 & \text{if } \frac{-r}{2} < j < \frac{r}{2} \\ \alpha \left( i - \frac{r}{2} \right) & \text{if } j \geq \frac{r}{2} \\ \alpha \left( i - \frac{r}{2} \right) & \text{if } j \leq \frac{-r}{2} \end{cases} \quad (5)$$

**محدودیت‌های محلی:** نگرانی اصلی در ناحیه جستجو این که توصیف‌کننده برای تطبیق مناسب به اندازه کافی متمایز نباشد. برای حل آن و کاهش اثر بی-ثباتی از یک محدودیت مکانی برای توقف تطبیق‌های نادرست استفاده شد [۴]. با این فرض که شیء موردنظر بین فریم‌های متوالی دارای پرش‌هایی بزرگ نباشد که توسط سری از وزن‌های گاوسی به‌عنوان جریمه به تابع امتیازدهی انجام می‌شوند. این کار به شیء اجازه حرکت آزادانه می‌دهد و می‌تواند دچار تغییر شکل شود. این روش برای متمایل کردن مکان شیء به مرکز شعاع ناحیه جستجو به‌صورت رابطه (۶) است.  $x$  و  $y$  در این رابطه گاوسی مختصات نقطه موردنظر نسبت به مختصات شیء موردردیابی است. هر چه نقطه موردنظر به مرکز ناحیه جستجو نزدیکتر باشد شانس انتخاب آن بیشتر است.

$$weight(x, y) = scaling\ Factor \times \left( 0.5 - e^{-\frac{x^2+y^2}{2\sigma^2}} \right) \quad (6)$$



شکل ۶: ضرایب تابع گاوسی برای محدودیت‌های مکانی.

آرایه‌ای از وزن‌ها به‌عنوان جریمه به بخش‌های مشخصی از ناحیه جستجو اضافه می‌کنیم. درحقیقت نقطه مرکزی دارای کمترین فاصله همینگ می‌شود. تابع گاوسی شکل ۶ شعاع جستجوی را برای ۲۰ پیکسل نشان می‌دهد که با تنظیم  $\sigma^2 = \frac{radius}{3}$  انجام می‌شود. در تنظیم ضرایب گاوسی باید مقدار فاکتور مقیاس مناسب تنظیم شود تا برای یافتن شیء در نقاط بیرونی ناحیه جستجو، شانس وجود داشته باشد. اگر جریمه زیاد باشد ردیاب در حرکت سریع شیء دچار تأخیر جیتر می‌شود و اگر جریمه کم باشد ردیاب موارد نادرست را به اشتباه درست تلقی می‌کند. تعیین بهترین مقدار  $\sigma^2$  برای ضرایب گاوسی مهم و مقدار خیلی بالای آن باعث تأخیر ردیاب در ردیابی اشیاء سریع و از دست‌دادن آنها می‌شود.

**سازگاری با مقیاس و اندازه شیء:** مقیاس همان تغییرات اندازه قابی است که شیء را ردیابی می‌کند. هرچند که ردیاب بعضی از تأثیرات مقیاس‌بندی را با دیکشنری پویا حل می‌کند ولی کامل نیست. اگر اندازه توصیف‌کننده در طول زمان براساس اندازه‌شیء تغییر کند مفید است. روش سازگاری با اندازه و مقیاس شیء را بررسی می‌کند که راه‌حل آن متکی بر تطبیق توصیف‌کننده‌ها است. در هر فریم نامزد دارای کمترین امتیاز و فاصله انتخاب می‌شود. در مقیاس برنده، توصیف‌کننده‌های جدید با مقیاس بالاتر و مقیاس پایین‌تر از توصیف‌کننده کنونی ساخته و با دیکشنری‌های ایستا مقایسه می‌شوند. مقیاس دارای فاصله و امتیاز کمتر از برنده کنونی برای فریم‌های آینده انتخاب می‌شود و باعث افزایش کارایی

است که توانایی عالی ردیاب را در مقابل تغییرات نشان می‌دهد. David و Dollar ویژگی‌های مشابهی با دوربین متحرک دارند. ویدئو Surfer دارای تصاویر ضعیف است و جهت کنترل تغییرات ردیابی، مؤثرترین ویدئو است. مشخصات و چالش‌های داده‌ها در ستون‌های جدول ۲ موجود و به ترتیب عبارت از حرکت دوربین<sup>۸۱</sup>، تغییر روشنایی<sup>۸۲</sup>، دوران خارج قاب<sup>۸۳</sup>، بهم‌ریختگی زمینه<sup>۸۴</sup>، تغییر مقیاس<sup>۸۵</sup>، انسداد درازمدت<sup>۸۶</sup> و ازدحام عوامل<sup>۸۷</sup> است.

#### ۴-۴ معیارهای ارزیابی

معیارهای اصلی ارزیابی توصیف‌کننده‌ها و ردیابی عبارت از صحت<sup>۸۸</sup>، دقت<sup>۸۹</sup>، تمایز<sup>۹۰</sup>، سرعت<sup>۹۱</sup> و مقاومت<sup>۹۲</sup> هستند. بدون مداخله انسان در بهترین تنظیم پارامترها هیچ وضعیت ایده‌آل ردیابی وجود ندارد. صحت مهم‌ترین پارامتر برای آزمون است. ردیاب می‌تواند بسیار سریع باشد ولی بدون دقت و صحت کافی در ثبت واقعی موقعیت شیء یک نمایش ساده جریان ویدیویی است.

**صحت و دقت:** از ساده‌ترین راه‌های اندازه‌گیری دقت ردیابی آن که تمام مکان‌های شیء ردیابی شده به‌عنوان خروجی تولید و سپس فاصله آنها با حقیقت پایه، در طول زمان ترسیم شود. میزان درجه نزدیکی یک اندازه به مقدار واقعی خودش است که برای هر مجموعه از پارامترها، فاصله میانگین از حقیقت پایه مشاهده و اندازه‌گیری می‌شود. پارامترهای مورد بررسی شعاع جستجو، حد آستانه تطبیق و  $\sigma$  ضرایب گاوسی خواهند بود. برای کمی سازی دقت الگوریتم ردیابی از روابط (۷) و (۸) استفاده می‌شود.

$$Precision = \frac{\# \text{ frames within } 20 \text{ pixels around ground truth}}{\# \text{ all frames}} \times 100 \quad (7)$$

$$Precision = \sum \text{frames} (\text{distance from ground truth})^2 / \# \text{ all frames} \quad (8)$$

**تمایز:** تمایز از عوامل مهم ارزیابی توصیف‌کننده‌ها است زیرا برای تطابق صحیح و درست باید منحصر به فرد باشند. به بیان دیگر دو توصیف‌کننده از دو نقطه متمایز شیء نباید دارای فاصله کمی باشند که منجر به تطبیق اشتباه آن دو شود. متأسفانه بررسی این مورد به‌آسانی بررسی میزان دقت و درستی نیست زیرا حقیقت پایه هیچ اطلاعاتی از توصیف‌کننده ندارد. گرچه فاصله همینگ معیاری از اندازه اولیه اینکه یک توصیف‌کننده خاص چقدر متمایز است را ارائه می‌دهد که به‌جای محاسبه فاصله‌ها و بررسی پایین‌ترین فاصله، تمام فواصل به خروجی ارسال و در یک نمودار سه‌بعدی ترسیم می‌شود که فاصله بین نقاط کلیدی متفاوت را نشان می‌دهد. الگوی ایده‌آل الگویی است که در آن برای نقاط نزدیک‌تر به مرکز فاصله کمتری وجود دارد. با افزایش فاصله مکانی فاصله همینگ نیز افزایش می‌یابد. اگر فقط یک بیت از دو نقطه کلیدی مجاور متفاوت باشد با احتمال فراوان این دو توصیف‌کننده به‌اندازه کافی متمایز نیستند لذا هرچه این نرخ دارای فاصله بیشتر باشد توصیف‌کننده در میان هم‌رتبه‌اش از یکتایی بهتر برخوردار است. لذا با مقیاس اندازه‌های توصیف‌کننده وجود نقطه شکست مبین معتبر نبودن توصیف‌کننده برای ردیابی است.

**سرعت سریال:** سرعت مشاهده آماری از نرخ است که ردیاب یک جریان ویدئویی را بر اساس فریم بر ثانیه پردازش می‌کند. معمولاً نتایج با معنی ردیاب بدون نمایش شیء در صفحه نمایش اجرا می‌شود که این نمایش خود عامل محدودکننده برای بدترین زمان‌بندی است. آزمون سرعت در تشخیص شیء با حداکثر دقت صورت می‌گیرد. سپس شعاع جستجو برای حداکثر و حداقل سرعت تعیین می‌شود تا در رابطه پیچیدگی روش برای جمع‌آوری تعداد زیاد نقاط کلیدی کسب اطلاعات شود و زمان راه‌اندازی و زمان توصیف به‌دست می‌آید. این کار وضعیتی را تعیین می‌کند که در آن وضعیت دیگر میزان و تعداد نقاط کلیدی از نظر زمانی بر عملکرد ردیابی تأثیر قابل‌توجهی ندارد. آزمون بعدی بررسی اثر

الگوریتم‌های پردازش تصویر، کتابخانه OpenCV نسخه ۲.۴.۱۱ مورد استفاده قرار گرفت و فایل‌های اجرایی آن کتابخانه با استفاده از نرم‌افزار CMake و کتابخانه‌های BBT<sup>۹۱</sup>، IPP<sup>۹۲</sup>، MKL<sup>۹۳</sup> و CUDA کامپایل شد. الگوریتم‌های توصیف‌کننده‌های باینری محلی با پیاده‌سازی‌های الگوریتم‌های استخراج باینری موجود در این کتابخانه برنامه‌سازی شد. برنامه‌نویسی با محیط یکپارچه Microsoft Visual Studio Ultimate 2013 SP2 صورت گرفت. برای بررسی جزئیات تصاویر، افزونه ImageWatch اضافه و برای کار با پردازنده گرافیکی، کیت توسعه نرم‌افزاری شرکت NVIDIA نصب و جمع‌شد. به‌منظور یافتن گلوگاه کد مربوط به کارت گرافیکی، از ابزار NVIDIA Nsight Visual Studio Edition نسخه ۴.۷ در سیستم عامل Microsoft Windows 8.1 Enterprise X64 استفاده شد. درایور کارت گرافیک از نسخه ۷.۵ NVIDIA است.

مشخصات سخت‌افزار عبارت از پردازنده Intel Core i7 4510U با تکنولوژی ۲۲ نانومتر، دارای ۴ پردازنده منطقی<sup>۹۶</sup> با فرکانس کاری ۲.۸ گیگاهرتز و سرعت گذرگاه ۱۰۰ مگاهرتزی و حافظه Cache به میزان ۴ مگابایت است. حافظه داخلی ۶ گیگابایت DDR3 و دارای فرکانس ۸۰۰ مگاهرتز است. کارت گرافیکی NVIDIA GEFORCE 840M دارای چیپ GM108M با معماری ماکسول<sup>۹۷</sup> و ظرفیت محاسباتی ۵.۰ و ۳ چند پردازنده<sup>۹۹</sup> ۱۲۸ هسته‌ای (در مجموع ۳۸۴ هسته محاسباتی) با نرخ کلاک ۱۱۲۴ مگاهرتز است. هر پردازنده قادر به اجرای ۲۰۴۸ نخ می‌باشد. بیشترین تعداد نخ هر بلاک این کارت گرافیکی برابر با ۱۰۲۴ نخ است. تعداد رجیسترهای هر بلاک برابر با ۶۵۵۳۶ رجیستر و حافظه مشترک بین نخ‌های هر بلاک، به میزان ۴۸ کیلوبایت است. حافظه ثابت کارت گرافیکی ۶۴ کیلوبایت. حافظه کارت گرافیکی ۴ گیگاهرتز DDR3 با فرکانس ۸۵۰ مگاهرتز عمل می‌کند. حداکثر اندازه بافت دوبعدی برابر  $65536 \times 65536$  است.

جدول ۲: چالش‌های موجود در مجموعه داده‌های ویدئویی آزمایشات.

ازدحام عوامل	انسداد طولانی	تغییر مقیاس	خرابی زمینه	دوران خارج قاب	تغییر روشنایی	حرکت دوربین	مجموعه داده
		●	●	●			Cliffbar
	●				●		Coke Can
		●			●	●	David
							Dollar
●	●	●		●			Girl
	●					●	Occluded face
				●			Surfer
					●		Sylvester
	●		●		●		Tiger1
	●		●		●		Tiger2
							Twinnings

#### ۴-۲ روش‌ها علمی مورد مقایسه

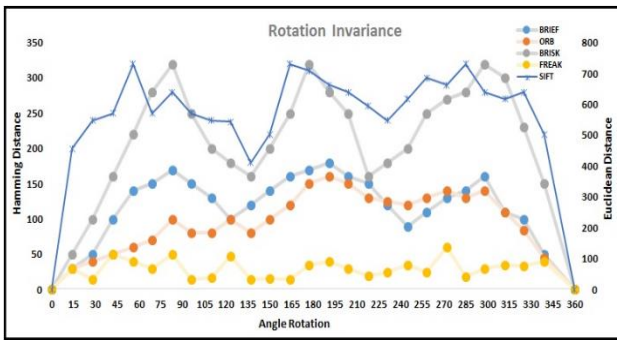
برای ارزیابی الگوریتم‌های موازی پیشنهادی، الگوریتم‌های مهم ORB، BRIEF، BRISK، SIFT و FREAK پیاده‌سازی شده است. پیاده‌سازی موازی الگوریتم بر روی الگوریتم‌های BRISK، FREAK و BRIEF مبتنی بر باینری محلی با استفاده از معماری CUDA صورت گرفته و الگوریتم پیشنهادی ردیابی هم‌زمان چند شیء متحرک دارای سرعت ابربی‌درنگ است.

#### ۴-۳ مجموعه داده‌ها

مجموعه داده‌های مهم ویدئویی مورد آزمایش عبارت از Cliffbar، Coke Can، Occluded Face، Dollar، David، Girl، Tiger1، Tiger2، Sylvester، Surfer و Twinnings که شش داده آن‌ها دارای حقیقت پایه<sup>۹۰</sup> هستند و نرخ فریم بر ثانیه آنها ۱۵، ۲۲۹، ۵۸۱، ۶۰۸ و ۷۰۹ است [۳۷]. بالاترین نرخ فریم در ویدئوی Cliffbar

#### ۴-۵ نتایج آزمایش‌ها

با توجه به تناسب ساختار معماری، الگوریتم و مقیاس‌پذیری روش مورد بررسی نتایج آزمایشات بر اساس معیارهای بی‌درنگی، مقاومت و سرعت ارائه می‌شود. البته بیان نتایج الگوریتم‌های مهم به دلیل اعتبارسنجی برای الگوریتم موازی ارائه شده است. مطالعات نشان می‌دهد که برای بسط و توسعه مدل‌های ردیابی جهت پوشش داده‌های عظیم تصویری نیازمند مدل‌های موازی جدید و هوشمندی عامل‌ها هستیم. معیار سرعت زیاد در کنار دقت بالا از معیارهای مهم ردیابی می‌باشد و کاهش پیچیدگی پیاده‌سازی الگوریتم‌ها در کنار توزیع پردازش‌ها در ارائه عملی این مدل‌ها لازم است.



شکل ۹: بررسی میزان مقاومت توصیف‌کننده‌ها نسبت به دوران.

**بی‌درنگی:** جدول ۳ ارائه سرعت‌هایی است که برای ردیابی محاسبه شده‌اند. روشن است SIFT هرگز برای سیستم بی‌درنگ معتبر نیست ولی تمام توصیف‌کننده‌های باینری واحدهای توصیفی با بهره‌وری بالا هستند. BRIEF بیشترین سرعت را دارد در صورتی که FREAK در شعاع‌های بیشتر به‌عنوان بهترین الگوریتم رتبه‌بندی می‌شود. جدول ۴ زمان ساخت توصیف‌کننده برای اندازه‌های مختلف را نشان می‌دهد. در این آزمایش بر روی یک مجموعه از نقاط کلیدی در اندازه‌هایی که کنترل شده بودند، بررسی‌ها انجام شد. چیزی که مجدداً در اینجا نیز خودش را نشان می‌دهد، این است که SIFT برای کاربردهای بی‌درنگ مناسب نیست. ORB در میان تمامی توصیف‌کننده‌ها، تنها موردی است که برخلاف انتظار خوب عمل نکرد. این امر ناشی از انتخاب روش مبتنی بر جبر خطی این الگوریتم برای یافتن جهت است. سه الگوریتم BRIEF، BRISK و FREAK رفتار قابل‌پیش‌بینی دارند و برای تصاویر با رزولوشن‌های متفاوت حتی تصاویر HD خوب عمل می‌کنند.

**مقاومت:** باوجودآنکه تمام تلاش‌های در جهت و مقاوم‌سازی نسبت به همه چالش‌ها ردیابی در بخش‌های یک روش مربوط است بخش اعظمی از آن مربوط به مقاومت توصیف‌کننده‌ها می‌باشد. برای آزمایش میزان مقاومت در برابر دوران، یک شیء دلخواه تعریف و توصیف می‌شود. اگر تصویر حول آن شیء دوران یافته در مکان باقی بماند فقط یک توصیف و مقایسه برای هر زاویه جدید انجام می‌شود. یک شیء برای تمام آزمایش‌های دوران و مقیاس استفاده شده است. تمام توصیف‌کننده‌ها بر روی این تصویر تمایزپذیری بالایی را از خود نشان دادند. این بدین معنی است که همگی آنها باید قادرند به‌خوبی آن را توصیف و تطبیق کنند. شکل ۹ تغییرناپذیری توصیف‌کننده‌های متفاوت را نسبت به دوران نشان می‌دهد. نکته‌ای که در سرعت باید در نظر گرفته شود این است که فاصله اقلیدسی برای الگوریتم SIFT در محور عمودی سمت راست در نظر گرفته شده زیرا فاصله توصیف‌کننده‌های این الگوریتم از نوع اقلیدسی و فاصله توصیف‌کننده‌های باینری از نوع فاصله همینگ است. از مشاهده نمودار الگوریتم SIFT مشخص می‌شود که نمودار آن نسبتاً مشابه با نمودار توصیف‌کننده‌های باینری است. به‌روشنی معلوم است که SIFT نسبت به دوران تغییرناپذیر نیست.

سرعت بر توصیف‌کننده‌ها با اندازه‌های متفاوت مانند طول باینری بزرگ و کوچک است. در تئوری نباید تفاوتی در سرعت برای یک توصیف‌کننده باینری بزرگ و یک توصیف‌کننده باینری کوچک وجود داشته باشد زیرا هر دوی آنها به‌سادگی فقط یک‌سری مقایسه‌های نقطه‌ای هستند. این مورد باید بررسی گردد تا مشخص شود که چقدر طول می‌کشد تا توصیف‌کننده ساخته شود. همچنین مطمئن شویم که الگوریتم استخراج توصیف‌کننده‌ها در عمل مانند تئوری عمل می‌کنند.

جدول ۳: سرعت ردیابی به فریم بر ثانیه با شعاع‌های مختلف جستجوی.

شعاع جستجو	SIFT	BRIEF	ORB	BRISK	FREAK
۱۰	۱,۱۰	۴۰۰,۰۰	۱۰۵,۰۴	۱۷۳,۳۱	۱۴۷,۹۳
۱۲	۰,۷۸	۲۹۸,۵۱	۷۵,۳۰	۱۴۰,۰۶	۱۳۷,۱۷
۱۴	۰,۵۸	۲۳۲,۰۲	۵۷,۱۱	۱۱۸,۲۰	۱۲۹,۳۷
۱۶	۰,۴۵	۱۸۴,۸۴	۴۴,۱۷	۱۰۱,۴۲	۱۱۷,۷۹
۱۸	۰,۳۶	۱۵۱,۰۶	۳۵,۱۹	۸۹,۳۷	۱۱۳,۳۸
۲۰	۰,۲۰	۱۲۴,۳۸	۲۸,۶۰	۷۶,۲۲	۱۰۳,۳۱
۲۲	۰,۲۶	۱۰۵,۰۶	۲۴,۰۵	۶۸,۲۱	۹۲,۱۷
۲۴	۰,۲۲	۸۹,۵۳	۲۰,۳۹	۵۷,۲۷	۸۲,۷۸
۲۶	۰,۱۸	۷۸,۸۶	۱۷,۳۷	۴۹,۸۳	۷۴,۰۷
۲۸	۰,۱۶	۶۵,۹۲	۱۴,۸۹	۴۵,۳۹	۶۸,۴۵
۳۰	۰,۱۴	۵۸,۷۵	۱۳,۱۳	۴۰,۳۴	۶۱,۰۱

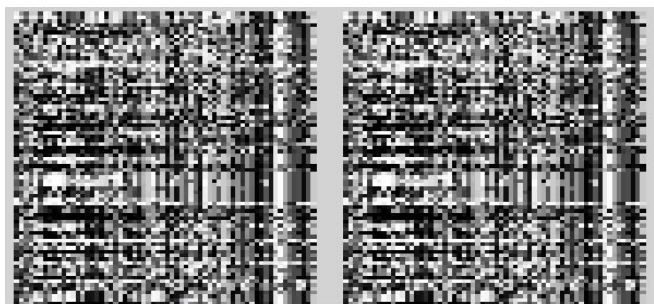
جدول ۴: زمان ساخت توصیف‌کننده به میلی‌ثانیه با نقاط کلیدی متفاوت.

نقاط کلیدی	SIFT	BRIEF	ORB	BRISK	FREAK
۴,۰۰۰	۴,۱۵۳	۰,۰۰۴	۰,۰۰۵	۱,۰۶۳	۱,۸۰۶
۸,۰۰۰	۴,۲۶۶	۰,۰۰۴	۰,۰۰۵	۱,۰۸۹	۱,۷۹۴
۱۶,۰۰۰	۴,۵۴۴	۰,۰۰۴	۰,۰۰۵	۱,۰۷۶	۱,۸۱۱
۳۲,۰۰۰	۶,۲۲۱	۰,۰۰۴	۰,۰۰۶	۱,۰۷۶	۱,۸۱۳
۶۴,۰۰۰	۸,۳۲۹	۰,۰۰۴	۰,۰۱۳	۱,۰۸۵	۱,۷۲۱
۱۲۸,۰۰۰	۹,۶۶۰	۰,۰۰۴	۰,۰۴۰	۱,۰۹۳	۱,۷۱۱
۲۵۶,۰۰۰	۹,۹۱۵	۰,۰۰۴	۰,۱۴۲	۱,۰۶۵	۱,۷۲۸

**سرعت موازی:** برای تنظیمات الگوریتم ردیابی موازی از پارامترهای عمومی نتایج الگوریتم سریال استفاده می‌کنیم. برای تحلیل سرعت اجرای این الگوریتم با معماری موازی، آن را با سه پارامتر میزان بهبود سرعت در انتقال تصویر به حافظه، استخراج ویژگی‌های توصیف‌کننده‌ها و تعداد اشیاء در ردیابی مورد بررسی قرار می‌دهیم. پارامتر میزان بهبود سرعت حاصل از قسمت انتقال عکس به حافظه GPU و استخراج ویژگی‌های مربوط به توصیف‌کننده‌ها می‌باشد. پارامتر میزان بهبود سرعت به نرخ پردازش فریم‌های ویدئویی است که الگوریتم موازی در ردیابی یک شیء، تا چه نرخ و سرعتی می‌تواند فریم‌های ویدئویی را پردازش و شیء موردنظر را ردیابی کند. پارامتر تعداد اشیایی یعنی الگوریتم موازی می‌تواند در سرعت بی‌درنگ عملیات ردیابی را بر روی آنها اجرا کند که با افزایش تعداد اشیاء مورد ردیابی میزان بهبود سرعت تا چه میزان است.

**مقاومت:** مقاوم بودن از سخت‌ترین معیارهای سنجش و یک مفهوم انتزاعی است. هدف اصلی این معیار تعیین دقیق این موضوع است که قبل از استفاده از توصیف‌کننده، تعیین میزان مقاومت نسبت به اعوجاج شیء، چرخش یا مانند آن است که با استفاده از تصاویر حقیقت‌پایه مشخص می‌شود. به‌محض یافتن شیء موردنظر برشی از تصویر در نظر گرفته می‌شود. از کتابخانه OpenCV برای آزمون مقاومت استفاده می‌گردد.

BRIEF برای تعیین مقیاس عملیات تطبیق اکتاو انجام نمی‌دهد. تمام توصیف-کننده‌های دیگر را شکست داده است. ORB عملکردی مشابه با BRIEF دارد ولی در مقیاس کمی بدتر از آن عمل می‌کند. BRISK از توصیف‌کننده‌های BRIEF و ORB دارای بیشترین فاصله از دیکشنری است. خطای آن تقریباً ۲ برابر خطای BRIEF و ORB است لذا نسبت به تغییرات مقیاس حساس‌تر و دارای نرخ خطای بالا است. FREAK به‌آسانی تغییرات ناشی از دوران را حل می‌کند اما در مقیاس-بندی با فاکتور ۱,۳ شکست می‌خورد. وقتی تغییرات ناشی از مقیاس در معادلات آن وارد می‌شوند خیلی سریع شروع به تطبیق اشتباه می‌کند که به‌وسیله دیکشنری پویا تا حدودی پوشش داده می‌شود ولی ممکن است شیء را از دست بدهد.



شکل ۱۲: ویژگی‌های استخراجی دو الگوریتم سریال و موازی (الف) الگوریتم موازی (ب) الگوریتم سریال

جدول ۵: بهبود سرعت در استخراج ویژگی‌ها

توصیف‌کننده	زمان استخراج با CPU	زمان استخراج با GPU	بهبود سرعت
BRIEF32	۱۱۷.۸۱	۷.۳۰	۹.۲
BRIEF64	۲۲۳.۸۳	۱۴.۳۲	۱۱.۷
BRISK	۱۳۳.۱۶	۳۸.۵	۳.۵
FREAK	۱۵۰.۷	۱۵.۱	۱۰

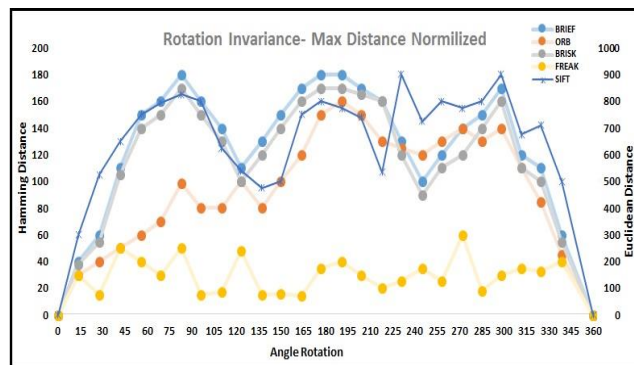
جدول ۶: سرعت ردیابی تک‌شیء بر حسب فریم بر ثانیه

توصیف‌کننده	سرعت ردیابی با CPU	سرعت ردیابی با GPU	بهبود سرعت
BRIEF32	۲۶۰.۹	۲۵۵۶.۸۲	۹.۸
BRIEF64	۱۲۸.۱	۱۶۷۸.۱۱	۱۳.۱
BRISK	۲۲۸.۲	۸۸۹.۹۸	۳.۹
FREAK	۶۱.۰۱	۳۳۵.۵۵	۵.۵

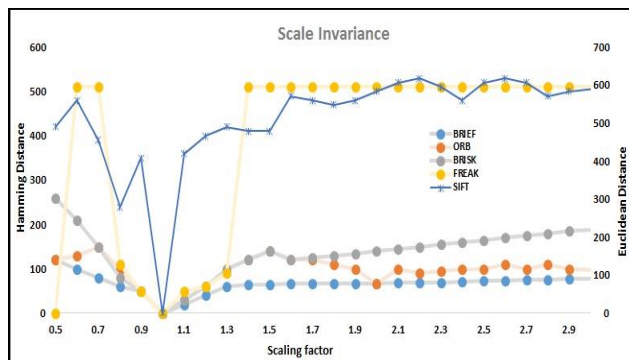
هر دو توصیف‌کننده BRIEF و BRISK در مقابله با تغییرات مقیاس عالی عمل می‌کنند و میزان خطایشان کاهش یافته است. BRIEF الگوی نمونه‌برداری خود را فقط به سمت مقیاس جدید حرکت داده و در تئوری هر بار نقاط مشابهی را نمونه‌برداری می‌کند. توصیف‌کننده ORB از اعمال تغییرات مقیاس تأثیری نپذیرفته است. الگوریتم FREAK در آزمایش‌های دوران و مقیاس به‌خوبی عمل می‌کند و بالاترین میزان دقت را دارد.

**اعتبارسنجی:** مقایسه خروجی دو الگوریتم سریال و موازی در دو بخش استخراج ویژگی‌های باینری و محاسبه فاصله‌همینگ صورت می‌گیرد. استخراج ویژگی‌های باینری در دو حالت سریال و موازی است. شکل ۱۲ ویژگی‌های باینری استخراجی توسط الگوریتم FREAK را در دو حالت سریال و موازی نشان می‌دهد. از شکل مشخص است که این ویژگی‌ها مشابه هستند و تفاوتی ندارند. در هر بایت از حافظه ۸ بیت از توصیف‌کننده‌ها ذخیره شده و هر تغییر رنگ در شکل (الف) و (ب) نشان‌دهنده یک بایت از اطلاعات توصیف‌کننده‌ها است. در CPU و GPU، به دلیل انجام محاسبه فاصله‌همینگ از طریق XOR بیتی بین بیت‌های حافظه، تفاوتی

با دقت در شکل متوجه می‌شویم که با حرکت به سمت توصیف‌کننده‌های باینری، اتفاقات جالبی رخ داده است. اولین نکته آنکه BRIEF عملاً الگویی مشابه با بسیاری از توصیف‌کننده‌های نامتغیر با دوران را دارد. این الگوریتم در میان توصیف‌کننده‌های باینری در زوایای دوران کم شیء دارای فاصله همینگ نسبتاً زیادی می‌باشد ولی در بعضی زوایای خاص چرخشی حایز فاصله کمتری است. به نظر می‌رسد که الگوی نمونه‌برداری تصادفی آن به الگوریتم کمک کرده است که بهتر عمل کند. الگوریتم ORB مشابه الگوریتم BRIEF عمل می‌کند اما ناحیه خاصی وجود دارد که BRIEF واقعاً دارای خطای کمتری است. روش جهت‌یابی ORB قادر نیست در تمامی حالات به‌خوبی عمل کند.



شکل ۱۰: مقاومت به دوران با مقایسه خطای توصیف‌کننده‌های ۳۲ و ۶۴ بیتی.



شکل ۱۱: مقاومت به دوران و مقیاس در توصیف‌کننده‌های مهم.

با توجه به الگوریتم‌های BRISK و FREAK روشن است که نتایج هر دوی آنها به شدت با نتایج BRIEF و ORB متفاوت می‌باشد. BRISK مقدار عظیمی خطا دارد که حتی از خطای BRIEF بزرگ‌تر است. حلقه‌های دایره‌ای BRISK نسبت به دوران حساس و متغیر می‌باشد لذا برای کاهش تأثیرات دوران روش خوبی نیست. در زوایای تقریبی ۱۳۵ درجه و ۲۷۰ درجه بیشترین سطح افت فاصله را دارد. مسلماً این مقدار خطا به آن است که تعداد بیت‌های تولیدی BRISK به اندازه ۲ برابر BRIEF و ORB است. شکل ۱۰ تأثیر نرمالیزه فاصله همینگ را نشان می‌دهد. توصیف‌کننده‌های ۶۴ بیتی و ۳۲ بیتی فاصله ماکزیمم مشابهی دارند. لذا BRISK از نظر عملکرد قابل‌مقایسه با ORB و BRIEF است. اگرچه FREAK مانند BRISK الگوی نمونه‌برداری مبتنی بر حلقه دارد ولی از کمترین میزان خطا برخوردار است. این امر مبین توانایی FREAK به‌عنوان توصیف‌کننده عالی و مقاوم به می‌باشد. روش FREAK برای مقابله با دوران با روش BRISK مشابه است ولی به‌جای استفاده از صدها جفت در محاسبه گردادیان محلی تنها از ۴۵ مورد استفاده می‌کند. مسئله مهم دیگر مقاومت به تغییر مقیاس در ردیابی پیوسته یک شیء است. شکل ۱۱ نتایج بررسی مقاومت به تغییر مقیاس را نشان می‌دهد. نتایج SIFT نشان از عمل خوب دارد و به نظر می‌رسد که با فاصله نسبتاً ثابت مستقر شده است. الگوریتم BRIEF علی‌رغم سادگی‌اش، بسیار مقاوم به تغییرات مقیاس است.

## ۵- نتیجه‌گیری

ردیابی اشیای متحرک از مسائل پرکاربرد بینایی ماشین است. الگوریتم مبتنی بر توصیف‌کننده‌های باینری محلی می‌تواند جایگزین عالی برای ردیاب‌های پیشرفته و مدرن در دستگاه‌های کوچک مانند عینک‌های هوشمند ردیاب و دارای سرعت بالا باشد. روش ردیابی پیشنهادی با دو دیکشنری ایستا و پویا وضعیت ردیابی شیء را تحت نظر می‌گیرد. در ناحیه جستجو به دنبال نزدیک‌ترین مورد مشابه است و برای غلبه بر تغییرات اندازه شیء با مقیاس بالاتر و پایین‌تر مقایسه می‌کند. همچنین در مقابل تغییرات اندازه و مقیاس شیء و دوران آن به خوبی مقاومت می‌کند و دارای سرعت اجرای ابربی‌درنگ است. با توجه به دستیابی به بهترین نتایج در توصیف‌کننده FREAK، این الگوریتم به صورت موازی مقاوم‌سازی و پیاده‌سازی شد. بهبود سرعت آن در الگوریتم‌های متفاوت با میانگین ۸ برابر از ۴ تا ۱۳ برابر به دست آمد که اجازه افزایش تعداد اشیاء در سرعت‌های ۳۰۰ تا ۳۳۰ فریم بر ثانیه را داد. با تنظیمات نخبه در الگوریتم موازی خاصیت مقیاس‌پذیری با تعداد ۳۰ تا ۱۴۰ شیء همزمان را بی‌درنگ فراهم کرد. در حالت سریال قادر به ردیابی یک شیء با سرعت ۶۰ فریم بر ثانیه است و در حالت موازی تا ۵٫۵ برابر سریع‌تر با سرعت ۳۳۰ فریم بر ثانیه به صورت ابربی‌درنگ ردیابی شد. در حالت سریال تا ۷ شیء را به صورت بی‌درنگ و همزمان ردیابی شد در حالی که حالت موازی بر اساس همان توصیف‌کننده قادر به ردیابی بی‌درنگ تا ۵۰ شیء همزمان گردید. ارزیابی‌ها نشان داد که با توجه به افزایش تعداد اهداف در ردیابی داده‌های عظیم تصویری برای مقیاس‌پذیری الگوریتم ردیابی استفاده از معماری موازی عامل‌گرا با عامل‌های هوشمند اجتناب‌ناپذیر است که کارهای آینده ما در جهت توسعه روش‌های مقاوم موجود برای مقیاس‌پذیر شدن با اشیاء بسیار زیاد خواهد بود.

## ۶- مراجع

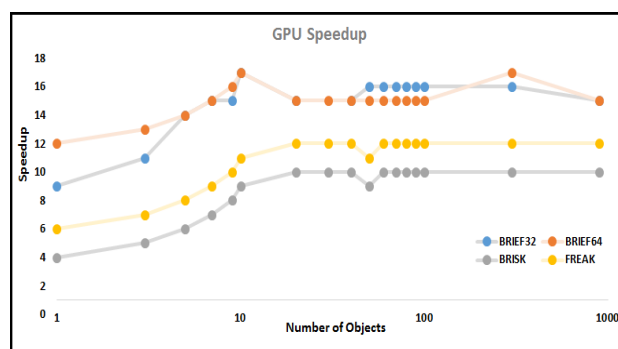
- [1] YouTube Processes 100 Hours of Video Each Minute. Available: <http://thenextweb.com/google/2013/05/19/youtube-100-million-hours-per-minute/>
- [2] T. Acharya and A. K. Ray, *Image processing: principles and applications*: John Wiley & Sons, 2005.
- [3] S. X. Guo, J. X. Chen, and B. Peng, "Research of Object Tracking Algorithm Based on BRISK," in *Advanced Materials Research*, pp. 1496-1501, 2014.
- [4] B. Minnehan, H. Spang, and A. E. Savakis, "Robust and Efficient Tracker using Dictionary of Binary Descriptors and Locality Constraints," in *Advances in Visual Computing*, Springer, pp. 589-598, 2014.
- [5] K. Chen, C. Lin, S. Zhong, and L. Guo, "A Parallel SRM Feature Extraction Algorithm for Steganalysis Based on GPU Architecture," in *Parallel Architectures, Algorithms and Programming (PAAP), 2014 Sixth International Symposium on*, pp. 178-182, 2014.
- [6] S. Heymann, K. Müller, A. Smolic, B. Froehlich, and T. Wiegand, "SIFT implementation and optimization for general-purpose GPU," in *WSCG'2007 International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision*, 2007.
- [7] D. A. Forsyth and J. Ponce, *Computer Vision: A Modern Approach*, pp. 88-101, 2003.
- [8] Y. Yuan, S. Emmanuel, W. Lin, and Y. Fang, "Visual object tracking based on appearance model selection," in *Multimedia and Expo Workshops (ICMEW), IEEE International Conference on*, pp. 1-4, 2013.
- [9] T. Hachaj and M. R. Ogiela, "Rule-based approach to recognizing human body poses and gestures in real time," *Multimedia Systems*, vol. 20, pp. 81-99, 2014.
- [10] Z. Kalal, K. Mikolajczyk, and J. Matas, "Tracking-learning-detection," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, pp. 1409-1422, 2012.

در اجرای دستور دو پردازنده وجود ندارد و فاصله‌های همینگ محاسبه شده یکسان است. ولی در محاسبه محدودیت‌های مکانی، اگر از دقت مضاعف استفاده شود در چند رقم انتهایی ممیز خروجی تفاوت‌هایی وجود دارد زیرا معماری پردازنده‌ها در محاسبات اعشاری ۶۴ بیتی یکسان نیستند [۳۵]. این تفاوت به دلیل ناچیز بودن تفاوت محاسبات در مقابل میزان تأثیر فاصله همینگ، تأثیری در تغییر برنده از نامزدهای شبکه جستجو ندارد و نتیجه ردیابی مکان شیء تغییر نمی‌کند. لذا برای دو الگوریتم تغییر در ساختار محاسبات داده نشده و مقایسه سرعت عادلانه است.

**سرعت الگوریتم‌ها:** اولین پارامتر مورد بررسی بهبود سرعت در زمان صرف شده برای استخراج ویژگی‌ها است. در جدول ۵ بهبود سرعت الگوریتم استخراج ویژگی‌های در GPU مشاهده می‌شود. بهبود سرعت در ویژگی‌های ۶۴ بیتی نسبت به ویژگی‌های ۳۲ بیتی بیشتر است. الگوریتم FREAK در حالت ۱۰۲۴ بیتی میزان بهبود سرعت ۱۵ برابر می‌باشد. در جدول ۶ سرعت پردازش بر حسب فریم بر ثانیه را برای ردیابی یک شیء در دو پیاده‌سازی بر CPU و GPU آمده است.

جدول ۷: ردیابی چند شیء با سرعت بی‌درنگ

توصیف‌کننده	اهداف ردیابی با CPU	اهداف ردیابی با GPU	میزان افزایش
BRIEF32	۸	۱۴۰	۱۷٫۵
BRIEF64	۴	۷۰	۱۷٫۵
BRISK	۳	۳۰	۱۰٫۵
FREAK	۷	۵۰	۷٫۱



شکل ۱۳: مقیاس‌پذیری افزایش اشیاء و تسریع ردیابی همزمان.

میزان بهبود سرعت در توصیف‌کننده‌های ۶۴ بیتی بیشتر است از توصیف‌کننده‌های ۳۲ بیتی است. بهبود سرعت توصیف‌کننده ۵۱۲ بیتی الگوریتم BRIEF نسبت به توصیف‌کننده ۲۵۶ بیتی آن به میزان ۳٫۳ برابر بیشتر می‌باشد. مفاد جدول ۷ مبین آن است پردازش ردیابی بر حسب تعداد اشیاء می‌تواند با سرعت بی‌درنگ انجام شود. تعداد اشیاء قابل‌ردیابی در GPU نسبت به CPU از ۷ تا ۱۷ برابر است. در شکل ۱۳ افزایش سرعت ردیابی با افزایش تعداد اشیاء مبین مقیاس‌پذیری الگوریتم در پردازنده گرافیکی است. بیشترین مقیاس قابل تحمل با توجه به منابع پردازنده گرافیکی هنگامی است که تعداد اشیاء مورد ردیابی به ۱۰۰ یعنی در نهایت سرعت برسد. تمام نخ‌های پردازنده اشباع و منبع آزادی در پردازنده گرافیکی وجود ندارد. در الگوریتم BRIEF64 از کمترین افزایش سرعت با ۷ هدف در CPU و ۵۰ هدف در GPU حداقل ۷ برابر تا بیشترین میزان بهبود سرعت ردیابی همزمان با ۸ هدف در CPU و ۱۴۰ هدف در GPU حداکثر حدود ۱۷ برابر است. این اندازه‌گیری‌ها نشان می‌دهد که با افزایش تعداد اهداف و نیاز بهبود سرعت برای ردیابی مقیاس‌پذیر در داده‌های عظیم تصویری لزوم استفاده از معماری موازی عامل‌گرا و منابع سخت‌افزاری مناسب اجتناب‌ناپذیر است.

- [33] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, pp. 1619-1632, 2011.
- [34] D.-N. Ta, W.-C. Chen, N. Gelfand, and K. Pulli, "Surfrac: Efficient tracking and continuous object recognition using local feature descriptors," in *Computer Vision and Pattern Recognition, IEEE Conference on*, pp. 2937-2944, 2009.
- [35] N. Whitehead and A. Fit-Florea, "Precision & performance: Floating point and IEEE 754 compliance for NVIDIA GPUs," *rn (A+B)*, vol. 21, pp. 1-1874919424, 2011.
- [36] [HTTPS://github.com/MohammadRezaNajafi/](https://github.com/MohammadRezaNajafi/).
- [37] [http://cvlab.hanyang.ac.kr/tracker\\_benchmark/datasets.html](http://cvlab.hanyang.ac.kr/tracker_benchmark/datasets.html).
- [11] S.-W. Ha and Y.-H. Moon, "Multiple object tracking using SIFT features and location matching," *International Journal of Smart Home*, vol. 5, pp. 17-26, 2011.
- [12] H. Zhou, Y. Yuan, and C. Shi, "Object tracking using SIFT features and mean shift," *Computer vision and image understanding*, vol. 113, pp. 345-352, 2009.
- [13] C. Harris and M. Stephens, "A combined corner and edge detector," in *Alvey vision conference*, p. 50, 1988.
- [14] H. P. Moravec, *Obstacle avoidance and navigation in the real world by a seeing robot rover*, STANFORD UNIV CA DEPT OF COMPUTER SCIENCE, 1980.
- [15] D. G. Lowe, "Object recognition from local scale-invariant features," in *Computer vision, The proceedings of the seventh IEEE international conference on*, pp. 1150-1157, 1999.
- [16] J. Xing and Z. Miao, "An improved algorithm on image stitching based on SIFT features," in *Innovative Computing, Information and Control, ICIC'07. Second International Conference on*, pp. 453-453, 2007.
- [17] W. He, et al. "Spatiotemporal local compact binary pattern for background subtraction in complex scenes." *Multimedia Tools and Applications*, vol.78, no.22, pp. 31415-31439, 2019.
- [18] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *Computer vision-ECCV 2006*, pp. 404-417, 2006.
- [19] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 16, pp. 355-368, 2010.
- [20] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *Computer Vision-ECCV*, Springer, pp. 778-792, 2010.
- [21] E. Rosten and T. Drummond, "Fusing points and lines for high performance tracking," in *Computer Vision, ICCV Tenth IEEE International Conference on*, pp. 1508-1515, 2005.
- [22] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Computer Vision-ECCV*, Springer, pp. 430-443, 2006.
- [23] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Computer Vision (ICCV), IEEE International Conference on*, pp. 2548-2555, 2011.
- [24] C. J. Veenman, M. J. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, pp. 54-72, 2001.
- [25] T. J. Brodia and R. Chellappa, "Estimation of object motion parameters from noisy images," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, pp. 90-99, 1986.
- [26] M. J. Deilamani and R. N. Asli, "Moving object tracking based on mean shift algorithm and features fusion," in *Artificial Intelligence and Signal Processing (AISP), International Symposium on*, pp. 48-53, 2011.
- [27] G. R. Bradski, "Real time face and object tracking as a component of a perceptual user interface," in *Applications of Computer Vision, Fourth IEEE Workshop on*, pp. 214-219, 1998.
- [28] J. Zhang, S. Luo, and X. Liu, "Weighted Mean Shift Object Tracking Implemented on GPU for Embedded Systems," in *Proceedings of the 2012 International Conference on Control Engineering and Communication Technology*, pp. 982-985, 2012.
- [29] G. D. Hager and P. N. Belhumeur, "Efficient region tracking with parametric models of geometry and illumination," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, pp. 1025-1039, 1998.
- [30] I. Matthews, T. Ishikawa, and S. Baker, "The template update problem," *IEEE transactions on pattern analysis and machine intelligence*, vol. 26, pp. 810-815, 2004.
- [31] N. Alt, S. Hinterstoisser, and N. Navab, "Rapid selection of reliable templates for visual tracking," in *Computer Vision and Pattern Recognition (CVPR), IEEE Conference on*, pp. 1355-1362, 2010.
- [32] G. Tsagkatakis and A. Savakis, "Online distance metric learning for object tracking," *Circuits and Systems for Video Technology IEEE Transactions on*, vol. 21, pp. 1810-1821, 2011.

محمد رضا نجفی کارشناس مهندسی کامپیوتر سخت-

افزار و کارشناس ارشد معماری کامپیوتر خود را از

دانشگاه شاهد اخذ نموده است. تحقیقات مورد علاقه

نامبرده پردازش تصویر و الگوریتم‌های موازی است.



آدرس پست الکترونیکی ایشان عبارت است:

mohammadreza.najafi1400@gmail.com

امین‌اله مه‌آبادی تحصیلات خود را در رشته

مهندسی برق سخت‌افزار و معماری کامپیوتر به انجام

رسانده و استادیار گروه مهندسی کامپیوتر و فناوری

اطلاعات دانشگاه شاهد است. تحقیقات مورد علاقه

نامبرده پردازش داده‌های عظیم تصویری، سیستم-



های بی‌درنگ نهفته، پردازش موازی و ابزارهای هوشمند است.

آدرس پست الکترونیکی ایشان عبارت است:

mahabadi@shahed.ac.ir

- 1 Multimedia
- 2 YouTube
- 3 Upload
- 4 Tracking
- 5 Visual Object Tracking (VOT)
- 6 Dynamic Configuration
- 7 Visible Spectrum
- 8 Invisible
- 9 Radio Detection And Ranging (RADAR)
- 10 Light Detection And Ranging (LIDAR)
- 11 Mobile and Portable devices
- 12 Local Binary Patterns
- 13 Big Image Data (BAD)
- 14 Hyper Real-time
- 15 Multi-core
- 16 Multi-processor
- 17 Multi-machine
- 18 Fog Computing
- 19 Agent-oriented
- 20 Intelligent Agent
- 21 Object
- 22 Background
- 23 Smooth
- 24 Abrupt Changes
- 25 Gradual
- 26 Appearance of Object
- 27 Occlusion
- 28 Non-rigid
- 29 Moving Object Detection (MOD)

---

30 Shape  
31 Central Processing Unit (CPU)  
32 Graphic Processing Unit (GPU)  
33 Scale Invariant Feature Transform (SIFT)  
34 Human Computer Interface (HCI)  
35 Simultaneously  
36 Real Time  
37 Tradeoffs  
38 Sacrifice  
39 Parallel Computing  
40 Processor Core  
41 Multi-thread  
42 Many-core Processor  
43 Local Binary Features (LBF)  
44 Jitter  
45 Pipeline  
46 Benchmark  
47 Fast Retina Keypoints (FRK)  
48 Ground Truth (GT)  
49 Difference of Gaussian  
50 Kernel  
51 Speedup Robust Features (SRF)  
52 Binary Robust Independent Elementary Features (BRIEF)  
53 Binary Robust Independent Scalable Keypoints (BRISK)  
54 Mean Shift  
55 Moving Points  
56 Template  
57 Foreground Background Discrimination  
58 Deterministic  
59 Probabilistic  
60 Embedded  
61 Continuously Adaptive Mean Shift (CAMS)  
62 Multiple Instance Learning (MIL)  
63 Tracking Learning Detection (TLD)  
64 Global Descriptor  
65 Continuous Object Recognition (COR)  
66 Dictionary Tracking (DT)  
67 Distance Metric Learning (DML)  
68 Information Theory Metric Learning (ITML)  
69 GPU  
70 Kernel  
71 Threading Building Blocks  
72 Integrated Performance Primitives  
73 Math Kernel Library  
74 Integrated Development Environment  
75 Software Development Kit  
76 Logical Processor  
77 Maxwell Architecture  
78 Compute Capability  
79 Multiprocessor  
80 Ground Truth  
81 Camer Motion  
82 Illumination Change  
83 Ou-of-Plane Rotation  
84 Background Clutter  
85 Scaling Chage  
86 Long-Term Occlusion  
87 Presence of Distractors  
88 Accuracy  
89 Precision  
90 Distinctness  
91 Speed  
92 Robustness

# Hyper Real-time Tracking of Simultaneous Moving Targets in Video Images

Mohamad Reza Najafi, Aminollah Mahabadi

Department of Computer Engineering, Shahed University, Tehran, Iran

---

## Abstract

Need for cloud computing and the use of parallel, fast and efficient algorithms in simultaneous objects tracking for fast processing of big image data is inevitable. This paper presents a parallel method based on local binary descriptors for concurrent tracking of 50 objects at real-time speed. The proposed method performs simultaneous real-time online tracking while using multi-threading and object detection applying the FREAK model in video images. The experimental results with 330 frames per second show that this method has the best efficiency and performance while having high speed and simultaneous tracking of objects with different scenarios in comparison with four important classical descriptive algorithms. The FREAK descriptor-based serial tracking system is capable of tracking an object at 60 frames per second, while in the proposed parallel tracking it tracks up to 330 frames per second and up to 5.5 times faster. Also, this tracker is able to real-time track up to 7 simultaneous objects in serial mode and in parallel mode, it is able to track up to 50 objects in hyper real-time.

**Keywords:** Concurrent Tracking of Objects, Parallel Algorithms, Video Image Processing, Binary Descriptors, Big Image Data.