

# توسعه روشی برای بررسی الگوریتم‌های تعیین موقعیت ربات پرنده با استفاده از سیستم ناوبری اینرشیا و عوارض نقطه‌ای و خطی تصاویر

محمد مهدی عباس‌پور<sup>۱\*</sup>، علی حسینی نوه<sup>۲</sup>

<sup>۱</sup> دانشجوی کارشناسی ارشد مهندسی نقشه برداری گرایش فتوگرامتری - دانشکده مهندسی نقشه برداری - دانشگاه

صنعتی خواجه نصیرالدین طوسی

mohammadmahdi.abbaspour@gmail.com

<sup>۲</sup> استادیار گروه فتوگرامتری و سنجش از دور - دانشکده مهندسی نقشه برداری - دانشگاه صنعتی خواجه نصیرالدین

طوسی

hosseininaveh@kntu.ac.ir

(تاریخ دریافت: مهر ماه ۱۳۹۹، تاریخ تصویب: اردیبهشت ماه ۱۴۰۲)

## چکیده

با پیشرفت روز افزون فناوری، ربات‌های پرنده در بسیاری از کاربردها مانند نقشه‌برداری، بازرسی و ایمنی، حمل کالا و عملیات‌های نظامی مورد استفاده قرار می‌گیرند. جهت ناوبری این پرنده‌ها، انجام عملیات تعیین موقعیت به صورت آنی، ضروری است. در محیط‌های خارج از ساختمان این کار اغلب با استفاده از سیستم‌های تعیین موقعیت ماهواره‌ای انجام می‌شود. اما در محیط‌های داخل ساختمان امواج سیستم‌های تعیین موقعیت ماهواره‌ای قابل دریافت نیستند. بنابراین، جهت ناوبری ربات‌های پرنده در داخل ساختمان باید از سایر روش‌های تعیین موقعیت مانند استفاده از دوربین‌های تصویربرداری بهره جست. همچنین، امکان استفاده از داده‌های حسگرهای اینرشیا به عنوان مکمل داده‌های تصویری وجود دارد. در فضای داخلی به علت کمبود بافت و مشکلات مربوط به روشنایی، می‌توان علاوه بر عوارض نقطه‌ای تصویر از عوارض خطوط تصویری نیز استفاده کرد. با توجه به مطالب ذکر شده، محققان الگوریتم‌های بسیاری جهت تعیین موقعیت ربات‌های پرنده ارائه داده‌اند. اما برای مقایسه این الگوریتم‌ها صرفاً به بررسی خطای کلی تعیین موقعیت و زمان اجرای الگوریتم بسنده شده و تحلیل و بررسی در مورد دلایل افزایش یا کاهش دقت هندسی یا علت شکست خوردن الگوریتم‌ها در محیط‌های پر چالش، کمتر مورد توجه قرار گرفته‌است. در این پژوهش، عملیات تعیین موقعیت پهپاد EuRoC با استفاده از تکنیک‌های منتخب، یعنی الگوریتم‌های PL-SVO و ORB-SLAM 2 و vins fusion انجام گرفته و نتایج آن‌ها تحلیل و مقایسه شده‌است. همچنین، دلایل موفقیت یا عدم موفقیت الگوریتم‌ها در عملیات تعیین موقعیت ربات در شرایط مختلف، با استناد به اطلاعات فیزیکی ربات و خواص رادیومتریکی تصاویر بیان گردیده‌است. در بین الگوریتم‌های ذکر شده، vins fusion با میزان RMSE کمتر از یک متر برای دو مورد از تست‌ها و حدود ۱/۲ متر برای تست دیگر بالاترین دقت را به خود اختصاص داده‌است. همچنین، الگوریتم ORB-SLAM 2 نیز در بخش‌هایی از پرواز که موفق به تخمین موقعیت ربات پرنده شده، RMSE کمتر از یک متر دارد.

**واژگان کلیدی:** انجمن علمی مهندسی نقشه‌برداری و ژئوماتیک ایران، فصلنامه علمی پژوهشی علوم و فنون نقشه‌برداری، راهنمای

نگارش نسخه داوری، قلم نازنین ۱۱. (حداکثر ۶ کلمه)

\* نویسنده رابط

## ۱- مقدمه

دریافت اطلاعات با استفاده از حسگرهای مختلف، از محیط، مسئله و چالشی مهم در علم رباتیک می‌باشد. یکی از کاربردهای مهم این حسگرها تعیین موقعیت ربات است که اولین مرحله عملیات ناوبری اتوماتیک محسوب می‌شود. از بین حسگرهای مختلف، دوربین‌های اخذ تصاویر دیجیتال و ویدئو به دلیل هزینه مناسب، سرعت بالای اخذ اطلاعات، ارائه داده‌های غنی و قابلیت انعطاف زیاد، بسیار مورد استفاده قرار می‌گیرند. مسئله تعیین موقعیت نسبی دوربین و بازسازی ساختار سه بعدی عوارض در دنیای بینایی ماشین با نام SFM<sup>۱</sup> شناخته می‌شود [۱، ۲]. در فرآیند SFM پارامترهای موقعیتی و وضعیتی دوربین و ساختار سه بعدی عوارض هر دو با هم استخراج می‌شوند و در نهایت خروجی‌ها با انجام یک سری محاسبات سرشکنی دسته اشعه<sup>۲</sup> بهبود می‌یابند [۳]. Visual odometry فرآیندی است که حرکت یک عامل (انسان، ربات، وسیله نقلیه و...) را با استفاده از ورودی‌های تصویری یک یا چند دوربین متصل به آن تخمین می‌زند [۴]. بنابراین می‌توان گفت Visual odometry حالت خاصی از SFM است به طوریکه تمرکز آن بیشتر روی تخمین پارامترهای موقعیتی و وضعیتی دوربین می‌باشد و این کار را به صورت آنی و هم‌زمان با اخذ فریم‌های جدید انجام می‌دهد. همچنین در این روش می‌توان به صورت غیر آنی با استفاده از سرشکنی دسته اشعه، تعیین موقعیت دوربین در طول مسیر حرکت را بهبود بخشید [۴]. به طور عمومی می‌توان گفت تخمین حرکت دوربین از طریق ردیابی عوارض تصویر با استفاده از هندسه فرض و آزمون Visual odometry نامیده می‌شود [۵]. فرآیند Visual SLAM نیز همانند Visual odometry با استفاده از داده‌های تصویری مسیر حرکت دوربین را تخمین می‌زند. در مجموع SLAM<sup>۳</sup> راهکاری است که با استفاده از آن، یک ربات می‌تواند در محیطی ناشناخته و با موقعیتی نامعلوم، نقشه محیط اطراف را تهیه و موقعیت خود را در آن بدست آورد [۶].

در صورتی که تکنیک SLAM با استفاده از ابزار لایدار انجام گیرد Lidar SLAM نامیده می‌شود و در

صورتی که با تلفیق داده‌های تصویری و داده‌های لایدار انجام گیرد به آن Visual-Lidar SLAM می‌گویند. از این روش‌ها برای ناوبری اتوماتیک ربات‌ها و وسایل نقلیه استفاده می‌شود [۷-۹].

یکی دیگر از مسائل پیش‌رو در این زمینه، تعیین موقعیت ربات به وسیله داده‌های تصویری در فضای داخل ساختمان می‌باشد. با توجه به عدم دسترسی به امواج سیستم‌های تعیین موقعیت ماهواره‌ای در محیط‌های داخلی، اهمیت استفاده از سنسورهای دیگر مانند دوربین‌های تصویر برداری جهت انجام این کار دو چندان می‌شود.

مسئله تخمین حرکت در وسایل نقلیه با استفاده از ورودی‌های تصویری اولین بار توسط Moravec در سال ۱۹۸۰ مطرح شد [۱۰]. سپس این موضوع توسط محققان دیگر ادامه یافته و توسعه داده شده‌است و هم اکنون نیز یکی از زمینه‌های تحقیقاتی پژوهشگران علوم رباتیک، بینایی ماشین و فتوگرامتری می‌باشد. سرعت پردازش داده‌ها و همچنین دقت و صحت تخمین مسیر حرکت ربات به صورت آنی و غیر آنی یکی از چالش‌های اصلی این حوزه است. به طوری که سالانه تحقیقات زیادی در جهت بهبود سرعت و دقت تعیین موقعیت ربات‌ها با استفاده از داده‌های تصویری انجام می‌گیرد [۱۱]. در اغلب این تحقیقات دقت تعیین موقعیت ربات در کل مسیر و همچنین سرعت اجرای الگوریتم به عنوان معیار برتری برای روش‌های مختلف در نظر گرفته شده‌است. اما بررسی دلایل به دست آمدن دقت هندسی مشخص، در محیط‌های مختلف با شرایط متفاوت و همچنین دلایل شکست الگوریتم‌ها در طول مسیر حرکت ربات، کم‌تر مورد توجه واقع شده‌است.

هدف پژوهش حاضر تعیین موقعیت ربات پرنده در محیط داخل ساختمان، با استفاده از چندین الگوریتم Visual odometry و Visual SLAM و بررسی و مقایسه دقت و کارایی الگوریتم‌ها می‌باشد. در این تحقیق علاوه بر بررسی نقاط قوت و ضعف روش‌های مختلف تعیین موقعیت در فضای داخل ساختمان و انتخاب الگوریتم‌های مناسب جهت انجام این کار، دلایل وجود خطای قابل توجه در برخی روش‌ها در مواجهه با شرایط مختلف ذکر شده و موفقیت یا عدم موفقیت الگوریتم‌ها در انجام عملیات تعیین موقعیت در بخش‌های مختلف مسیر پرواز، با استفاده از اطلاعات و پارامترهای فیزیکی ربات و همچنین خواص و ویژگی‌های تصاویر اخذ شده تحلیل می‌گردد.

<sup>۱</sup> Structure From Motion

<sup>۲</sup> Bundle Adjustment

<sup>۳</sup> Simultaneous Localization And Mapping

## ۲-۱- الگوریتم‌های مستقیم Visual-SLAM

در روش مستقیم Visual-SLAM، تصاویر اخذ شده به صورت متوالی با هم مقایسه شده و با بهینه کردن خطای فتومتریک درجات خاکستری پیکسل‌های آن، موقعیت ایستگاه اخذ تصاویر نسبت به هم به دست می‌آید. از روش‌های معروف Visual SLAM مستقیم می‌توان به DTAM<sup>۷</sup> [۱۵] و LSD SLAM<sup>۸</sup> [۱۶] و DPPTAM<sup>۹</sup> [۱۷] اشاره کرد. این روش‌ها به دلیل تناظریابی متراکم تصاویر، بیشتر برای تخمین موقعیت عامل به همراه تهیه مدل سه بعدی متراکم، مورد استفاده قرار می‌گیرند. لذا در صورتی که هدف صرفاً به دست آوردن موقعیت ربات و تخمین مسیر حرکت آن باشد، به دلیل حجم بالای محاسبات گزینه مناسبی نیستند.

## ۲-۲- الگوریتم‌های غیر مستقیم Visual-SLAM

در روش غیر مستقیم ابتدا عوارض کلیدی تصاویر مانند نقاط گوشه یا حباب‌های تصویری استخراج شده و برای هر یک توصیفگری تعریف می‌شود. به این ترتیب می‌توان همان عوارض را در تصاویر متوالی بعدی یافت. سپس با استفاده از عوارض مشترک استخراج شده بین تصاویر، آن‌ها با یکدیگر توجیه نسبی شده و موقعیت سه بعدی عوارض بدست آمده، با استفاده از روش تقاطع محاسبه گردیده و ابر نقطه سه بعدی از آن‌ها بدست می‌آید. بدین ترتیب می‌توان با مشاهده مختصات تصویری عوارض و همچنین با آگاهی از مختصات سه بعدی آن‌ها نسبت به محاسبه پارامترهای موقعیتی و وضعیتی دوربین در هنگام اخذ تصاویر اقدام کرد. تکنیک‌های غیر مستقیم Visual SLAM اغلب از عوارض نقاط بارز تصویری استفاده می‌کنند. تکنیک‌هایی مانند PTAM<sup>۱۰</sup> [۱۸]، DT SLAM<sup>۱۱</sup> [۱۹]، ORB-SLAM [۲۰] و ORB-SLAM 2 [۲۱] از روش‌های برتر این گروه از تکنیک‌ها هستند.

برای این منظور ابتدا تحقیقات انجام شده در این حوزه ذکر شده و الگوریتم‌های منتخب از لحاظ تئوری مورد بررسی قرار می‌گیرند. همچنین نقاط قوت تعیین موقعیت با استفاده از الگوریتم‌های منتخب در محیط‌های داخل ساختمان بیان می‌گردد. در ادامه نحوه بررسی و معیارهای مقایسه و تحلیل خروجی‌های روش‌های تعیین موقعیت ربات بیان شده و سپس چگونگی پیاده‌سازی الگوریتم‌ها در ربات پرنده شرح داده می‌شود. در قسمت آخر نیز نتایج بدست آمده مورد بحث و بررسی قرار گرفته و پیشنهاداتی برای پژوهش‌های آتی در این حوزه مطرح می‌گردد.

## ۲-۲- مروری بر تحقیقات گذشته

تحقیقات بسیاری برای تهیه نقشه و تعیین موقعیت عامل متحرک، با استفاده از داده‌های لایدار در محیط‌های متفاوت جهت کاربردهای مختلف انجام شده‌اند. برای مثال در سال ۲۰۱۴ ژانگ<sup>۱</sup> و سانجیو سینگ<sup>۲</sup> در پژوهشی اقدام به تعیین موقعیت و نقشه برداری از محیط اطراف با استفاده از داده لایدار کردند [۱۲]. اما سنسورهای لایدار با وجود ارائه دقت مکانی مطلوب، گران قیمت بوده و مصرف انرژی بالایی دارند. همچنین این سنسورها با چالش‌های جدی به دلیل عدم امکان اخذ داده از روی سطوح شیشه‌ای و شفاف مواجه هستند [۱۳]. به همین دلیل در این پژوهش استفاده از آن‌ها برای تعیین موقعیت ربات مد نظر نمی‌باشد. از طرفی روش‌های تعیین موقعیت عامل و تهیه مدل سه بعدی با استفاده از دوربین‌های تصویر برداری به دو روش Visual SLAM و Visual odometry تقسیم می‌شود. تکنیک Visual SLAM نیز به نوبه خود به دو روش مستقیم و غیر مستقیم انجام می‌گیرد [۱۴].

تحقیقات آکادمیک در حوزه Visual SLAM با کار پیتر چسمن<sup>۳</sup> و جیم کرولی<sup>۴</sup> و هو دورانت ویت<sup>۵</sup> که در کنفرانس IEEE Robotics and Automation سال ۱۹۸۶ در سانفرانسیسکو ارائه گردید، آغاز شد. در حوزه Visual odometry نیز کار آکادمیک با مقاله داوید نیستر<sup>۶</sup> با عنوان Visual odometry در سال ۲۰۰۴ رسماً شروع شد [۵].

<sup>۷</sup> Dense Tracking And Mapping

<sup>۸</sup> Large Scale Direct SLAM

<sup>۹</sup> Dense Piecewise Parallel Tracking And Mapping

<sup>۱۰</sup> Parallel Tracking And Mapping

<sup>۱۱</sup> Deferred Triangulation SLAM

<sup>۱</sup> Ji Zhang

<sup>۲</sup> Sanjiv Singh

<sup>۳</sup> Peter Cheeseman

<sup>۴</sup> Jim Crowley

<sup>۵</sup> Hugh Durrant-Whyte

<sup>۶</sup> David Nister

جهت در برخی روش‌ها از ترکیب اطلاعات تصویری و داده‌های اینرشیا برای تعیین موقعیت ربات و تخمین مسیر حرکت آن استفاده شده‌است. به عنوان مثال الگوریتم vins fusion اطلاعات حسگر اینرشیا را با اطلاعات موقعیت و وضعیت استخراج شده از تصاویر برای به دست آوردن موقعیت ربات ترکیب می‌کند [۲۹].

### ۳- الگوریتم‌های منتخب

در ادامه الگوریتم‌های منتخب تعیین موقعیت ربات در فضای داخلی و دلایل انتخاب‌شان بررسی شده و خلاصه‌ای از مبانی نظری آن‌ها بیان می‌گردد.

#### ۳-۱- الگوریتم PL-SVO

الگوریتم SVO الگوریتمی کارا و سریع در بین الگوریتم‌های مختلف Visual odometry می‌باشد که برای تعیین موقعیت و وضعیت دوربین تصویربرداری و همچنین ساختن مدلی نیمه متراکم از محیط اطراف، توسعه داده شد. در این الگوریتم دو فرآیند تعیین موقعیت و وضعیت دوربین و نقشه‌برداری از محیط اطراف به صورت موازی انجام شده و هر یک بر نتایج نهایی دیگری تأثیر گذار است. رویکرد این روش حالت ارتباط داده سه بعدی به دو بعدی بوده<sup>۳</sup> و با استفاده از یک فیلتر عمق اقدام به بازسازی موقعیت و وضعیت دوربین در حین اخذ تصویر و نیز محاسبه موقعیت نقاط منتخب محیط اطراف می‌کند. این روش بر روی یک پهپاد کوچک پیاده سازی شده و سرعت بالایی را ارائه داده‌است. به طوری که با استفاده از برد پهپاد به سرعت پردازش ۵۵ فریم در ثانیه و با استفاده از یک کامپیوتر شخصی به سرعت ۳۰۰ فریم بر ثانیه رسیده‌است [۲۲].

یکی از ویرایش‌های الگوریتم SVO با عنوان PL-SVO، در سال ۲۰۱۶ توسط روبن گومز اوجدا<sup>۴</sup> و همکارانش ارائه شد. این الگوریتم با توجه به استفاده از عوارض خطوط تصویری علاوه بر عوارض نقاط تصویری، برای استفاده در فضای داخلی مناسب بوده و در این محیط دقت بیشتری نسبت به الگوریتم SVO به دست آورده‌است [۲۵]. دلیل استفاده از عوارض خطوط

همچنین کریستین فرستر<sup>۵</sup> و همکارانش در سال ۲۰۱۴ یک الگوریتم Visual odometry ارائه دادند که ترکیبی از روش مستقیم و غیر مستقیم بود [۲۲]. این روش که SVO<sup>۶</sup> نام گرفت در سال‌های بعد، بسیار مورد توجه محققان قرار گرفته و چندین ویرایش از آن معرفی شد [۲۳-۲۶].

### ۳-۲- الگوریتم‌های مبتنی بر ترکیب ویژگی‌های

#### نقاط و خطوط بارز تصویری

در فضاهای داخل ساختمان معمولاً به دلیل کمبود بافت و یکنواخت بودن رنگ سطوح، ویژگی نقاط کمتری قابل استخراج هستند. بنابراین می‌توان از عوارض خطوط به همراه عوارض نقاط در اجرای تکنیک‌های فوق بهره گرفت. الگوریتم‌های PL-SVO<sup>۷</sup> [۲۵] و PL-StVO<sup>۸</sup> [۲۶] ویرایش‌هایی از الگوریتم SVO هستند که از این روش جهت تعیین موقعیت و تهیه مدل سه بعدی استفاده می‌کنند. همچنین الگوریتم‌های PL-SLAM<sup>۹</sup> [۲۷] و PL-SLAM (stereo)<sup>۱۰</sup> [۲۸] نیز از ترکیب عوارض نقاط و خطوط برای اجرای تکنیک SLAM بهره می‌گیرند.

### ۳-۴- الگوریتم‌های مبتنی بر تلفیق داده‌های

#### تصویری و اینرشیا

اطلاعات حسگر اینرشیا به دلیل مستقل بودن از شرایط محیطی از جمله روشنایی و جنس سطوح عوارض محیط اطراف و سر پوشیده یا باز بودن فضای مد نظر، می‌تواند بسیار ارزشمند باشد. این حسگرها ارزان قیمت و سبک بوده و مصرف انرژی پایینی دارند. مشکل اساسی حسگرهای اینرشیا خطای تجمعی بیش از حد در انجام عملیات تعیین موقعیت است. به همین علت استفاده از این حسگرها به تنهایی کاربردی نیست اما می‌توان از آن‌ها به عنوان مکمل برای دیگر حسگرهای مورد استفاده در ربات به خصوص در هنگام عدم دسترسی امواج سیستم‌های تعیین موقعیت ماهواره‌ای، بهره گرفت [۱۳]. به همین

<sup>۳</sup> Christian Forster

<sup>۴</sup> Semi direct Visual Odometry

<sup>۵</sup> Point and Line-Semi direct Visual Odometry

<sup>۶</sup> Point and Line-Semi direct Stereo Visual Odometry

<sup>۷</sup> Point and Line-Simultaneous Localization And Mapping

<sup>۸</sup> Point and Line-Simultaneous Localization And Mapping

(Stereo)

<sup>۹</sup> 3D-2D data association

<sup>۱۰</sup> Ruben Gomez-Ojeda

تصویری در محیط‌های داخل ساختمان، وجود داشتن خطوط بارز و مشخص در این فضاها می‌باشد که با اپراتورهای تشخیص لبه و خط، قابل استخراج هستند. همین امر این امکان را به وجود می‌آورد که به وسیله عملگرهای توصیف خطوط، تناظر یابی تصاویر انجام شده و موقعیت دوربین با استفاده از نقاط ابتدا و انتهای آن‌ها محاسبه گردد. این خطوط با استفاده از یک تکنیک استخراج قطعات خطوط از تصاویر، به نام LSD<sup>۱</sup> به دست می‌آیند. LSD نسبت به روش‌های سنتی استخراج خطوط مانند Hough transform سریع‌تر و دقیق‌تر بوده و نیازی به تنظیم پارامترهای ورودی ندارد [۳۰]. با استفاده از روش ذکر شده برای استخراج خطوط و سپس استفاده از خطوط استخراج شده برای تخمین حرکت دوربین، میزان دقت تعیین موقعیت ربات نسبت به روش SVO افزایش می‌یابد. البته از لحاظ زمان اجرا، الگوریتم PL-SVO دارای سرعت پایین‌تری نسبت به الگوریتم SVO است [۲۵].

### ۳-۲- الگوریتم ORB-SLAM 2

از بین الگوریتم‌های غیر مستقیم Visual SLAM از عوارض نقطه‌ای استفاده می‌کنند، الگوریتم ORB-SLAM یکی از کاراترین، سریع‌ترین و دقیق‌ترین روش‌ها به شمار می‌رود. همچنین این الگوریتم در محیط‌های داخلی و خارجی پیاده‌سازی شده و در هر دو محیط قابلیت اجرا دارد. این الگوریتم با استفاده از استخراجگر چند مقیاسه FAST و توصیفگر ORB، هم زمان اقدام به تولید مدل سه بعدی محیط اطراف و تعیین موقعیت و وضعیت دوربین تصویربرداری در محیط‌های بزرگ مقیاس و کوچک مقیاس داخلی و خارجی می‌کند. الگوریتم ORB، یک توصیفگر نقاط بارز در تصاویر است که بر مبنای عملگر BRIEF توسعه داده شده و سرعت بسیار بالایی را ارائه می‌دهد. این الگوریتم نقاط استخراج شده را با استفاده از بردار ۲۵۶ مولفه‌ای دودویی، با سرعت بسیار بالا نسبت به توصیفگرهای سنتی، توصیف می‌کند تا در مرحله تطبیق تصاویر بتوان از توصیفگرها جهت تناظریابی نقاط متناظر استفاده کرد [۳۱]. در روش ORB-SLAM سه

فرآیند به طور موازی انجام می‌شوند. فرآیند اول ردیابی عوارض استخراج شده‌است که با استفاده از آن می‌توان موقعیت و وضعیت دوربین را به دست آورد. دومین فرآیند، تولید مدل سه بعدی محیط اطراف است که با استفاده از فریم‌های کلیدی اخذ شده صورت می‌گیرد و سومین فرآیند اصلی، تشخیص بسته شدن حلقه برای انجام سرشکنی و تعدیل خطاهای تجمعی به وسیله سرشکنی دسته اشعه و فرآیند بهینه سازی با استفاده از گراف ضروری<sup>۲</sup> می‌باشد. به این ترتیب می‌توان در صورتی که ربات یک مسیر بسته را طی کرد، با استفاده از تشخیص بسته بودن مسیر، عملیات سرشکنی کلی را با دقت بالا انجام داد [۲۰]. همچنین الگوریتم ORB-SLAM 2 بر اساس الگوریتم ORB-SLAM برای استفاده دوربین‌های تکی، استریو و RGB-D توسعه داده شد [۲۱]. در مورد دوربین‌های استریو، علاوه بر مراحل ذکر شده مرحله‌ای نیز جهت انجام عملیات تناظریابی استریو انجام می‌گیرد. مزیت روش استریو، حل ابهام مقیاس با استفاده از فاصله ثابت بین دوربین چپ و دوربین راست می‌باشد. البته این روش در محیط‌هایی قابل مقایسه با اندازه فاصله بین دوربین چپ و راست مفید بوده و در محیط‌های با ابعاد بزرگ‌تر کارایی خود را از دست می‌دهد.

### ۳-۳- الگوریتم vins fusion

الگوریتم vins fusion نیز بر پایه الگوریتم ORB-SLAM بنا نهاده شده‌است که با استفاده از موقعیت یابی به روش ORB-SLAM 2 و تلفیق آن با اطلاعات حسگرهای اینرشیا<sup>۳</sup> در محیط‌های داخلی و اطلاعات حسگرهای اینرشیا و داده‌های گیرنده امواج سیستم‌های تعیین موقعیت ماهواره‌ای<sup>۴</sup> در محیط‌های خارجی، اقدام به تعیین موقعیت وسیله نقلیه بدون سرنشین می‌کند [۲۹]. با توجه به توضیحات فوق الذکر، الگوریتم‌های PL-SVO و ORB-SLAM 2 و vins fusion به عنوان الگوریتم‌های منتخب جهت تعیین موقعیت ربات پرنده در این پژوهش در نظر گرفته می‌شوند.

<sup>۳</sup> Essential graph optimization

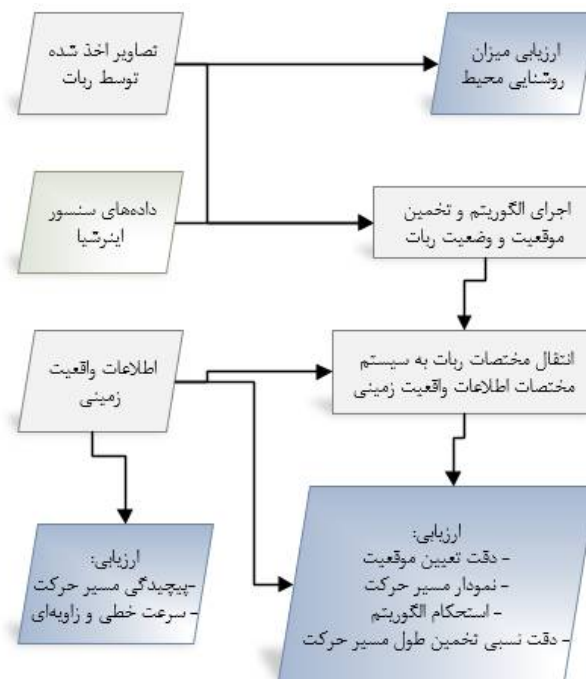
<sup>۴</sup> Inertial Navigation System

<sup>۱</sup> Global Positioning System

<sup>۲</sup> Line Segment Detector

#### ۴- روش پیشنهادی برای ارزیابی دقت و کارایی الگوریتم‌ها در شرایط مختلف

در این قسمت روش پیشنهادی برای ارزیابی الگوریتم‌های منتخب و همچنین بررسی شرایط محیطی و پارامترهای فیزیکی ربات در حین اخذ داده بیان می‌شود. به این ترتیب می‌توان الگوریتم‌ها را ارزیابی و مقایسه کرده و دلیل موفقیت یا عدم موفقیت آن‌ها را با توجه به شرایط اخذ داده تحلیل کرد.



شکل ۱- فلوچارت روش پیشنهادی جهت ارزیابی دقت و کارایی الگوریتم‌های تعیین موقعیت ربات بر پایه داده‌های تصویری و تحلیل و بررسی تاثیر شرایط نوری محیط و نحوه حرکت ربات بر آن.

خام اخذ شده توسط ربات بررسی شده و سرعت خطی و زاویه‌ای و پیچیدگی مسیر حرکت ربات نیز از داده‌های واقعیت زمینی محاسبه می‌شود. در ادامه هر یک از مراحل بیان شده به طور کامل، همراه با ذکر روابط ریاضی، شرح داده شده‌است.

#### ۴-۱- معیارهای بررسی دقت و کارایی الگوریتم‌ها

معیارهای پیشنهادی برای بررسی دقت و کارایی الگوریتم‌ها عبارتند از استحکام الگوریتم، دقت تعیین موقعیت و دقت نسبی تخمین طول مسیر حرکت که در ادامه هر کدام از آن‌ها به صورت مجزا توضیح داده می‌شود.

ابتدا داده‌های تصویری و داده‌های حسگر اینرشیا به عنوان ورودی به الگوریتم تعیین موقعیت معرفی می‌شوند. سپس الگوریتم، مختصات ربات در طول مسیر حرکت را به دست می‌آورد و پس از تبدیل مختصات، دقت تعیین موقعیت از مقایسه مختصات محاسبه شده با مختصات داده‌های واقعیت زمینی<sup>۱</sup> تعیین می‌شود. از طرفی با استفاده از زمان ثبت شده برای مختصات به دست آمده، استحکام الگوریتم و با استفاده از طول مسیر محاسبه شده و طول مسیر واقعی دقت تخمین طول مسیر حرکت به دست می‌آید. همچنین میزان روشنایی تصاویر، از تصاویر

<sup>۲</sup> Ground Truth

در زمان ناوبری اتوماتیک ربات، تعیین موقعیت آن امری ضروری می‌باشد. به همین دلیل مهم‌ترین مسئله استحکام الگوریتم تعیین موقعیت است که نشان دهنده محاسبه مختصات ربات در طول مسیر پرواز می‌باشد. در صورت استحکام بالای الگوریتم، دقت تعیین موقعیت توسط آن در رتبه بعدی اهمیت قرار دارد.

#### ۴-۱-۱- استحکام الگوریتم

اولین مسئله جهت ارزیابی الگوریتم‌های منتخب، استحکام و کارایی الگوریتم می‌باشد. برای بدست آوردن این معیار قسمت‌هایی از مسیر حرکت ربات که مختصاتشان توسط الگوریتم محاسبه نشده‌است، بررسی می‌شود. الگوریتم‌های تعیین موقعیت با توجه به شرایط محیط و نحوه حرکت ربات در محیط، همیشه توانایی محاسبه مختصات آن را ندارند. از این رو هر الگوریتم در بخش‌های بیشتری از مسیر حرکت ربات، مختصات آن را از دست بدهد، از لحاظ استحکام و کارایی ضعیف‌تر می‌باشد. در این پژوهش با استفاده از زمان ثبت شده برای مختصات محاسبه شده ربات در طول مسیر، این معیار مورد بررسی قرار گرفته‌است.

#### ۴-۱-۲- دقت تعیین موقعیت

مسئله بعدی دقت تعیین موقعیت ربات می‌باشد. معیاری که برای بررسی خطای تعیین موقعیت الگوریتم‌ها استفاده شده RMSE است. جهت بررسی میزان دقت عملیات موقعیت یابی با استفاده از معیار RMSE بایستی مختصات به دست آمده توسط الگوریتم‌های تعیین موقعیت با استفاده از یک تبدیل متشابه سه بعدی به سیستم مختصات داده‌های واقعیت زمینی منتقل شود. رابطه ۱ فرمول مربوط به تبدیل متشابه سه بعدی را نشان می‌دهد.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \lambda \times R(roll, pitch, yaw) \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} Tx \\ Ty \\ Tz \end{bmatrix} \quad (1)$$

در رابطه فوق X و Y و Z مختصات نقاط در سیستم مختصات مربوط به اطلاعات واقعیت زمینی و x و y و z مختصات اولیه نقاط، مربوط به موقعیت ربات در طول مسیر می‌باشد که توسط الگوریتم تخمین زده شده‌است. R نشان

دهنده ماتریس دوران سه بعدی و ماتریس T نشان دهنده پارامترهای انتقال و  $\lambda$  معرف ضریب مقیاس است.

محاسبه مقدار RMSE با استفاده از روابط ۲ و ۳ امکان پذیر می‌باشد.

$$dr = \sqrt{dx^2 + dy^2 + dz^2} \quad (2)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^n dr_i^2}{n}} \quad (3)$$

در روابط فوق dx و dy و dz اختلاف مختصات بین موقعیت محاسبه شده توسط الگوریتم و مختصات مربوط به واقعیت زمینی است و n تعداد کل نقاطی می‌باشد که در طول مسیر پرواز توسط الگوریتم محاسبه شده‌است. برای یافتن موقعیت متناظر هر نقطه محاسبه شده توسط الگوریتم از بین موقعیت‌های ثبت شده واقعیت زمینی، از زمان ثبت شده برای هر نقطه استفاده می‌شود. کمیت زمان در مجموعه داده مربوط به موقعیت محاسبه شده توسط الگوریتم و مجموعه اطلاعات واقعیت زمینی با فرمت یکسان و دقت بالا که دارای واحد ثانیه و نانو ثانیه می‌باشد، ثبت شده‌است و می‌توان برای این منظور از آن استفاده کرد.

با استفاده از تبدیل متشابه سه بعدی و انتقال مختصات محاسبه شده به سیستم مختصات واقعیت زمینی، می‌توان نمودار مسیر حرکت ربات که توسط الگوریتم تخمین زده شده را در کنار مسیر حرکت آن که با استفاده از اطلاعات واقعیت زمینی به دست می‌آید ترسیم و از لحاظ بصری دقت الگوریتم را مورد ارزیابی قرار داد. همچنین می‌توان معیار RMSE را فقط برای قسمت‌هایی از مسیر بدست آورد که در آن الگوریتم موفق به تخمین موقعیت ربات شده‌است. همانطور که در بخش قبل گفته شد، به دلیل شرایط روشنایی محیط یا نحوه حرکت ربات در بعضی از قسمت‌های مسیر، الگوریتم‌های تعیین موقعیت در محاسبه موقعیت ربات شکست خورده و مختصات آن را از دست می‌دهند. در این پژوهش علاوه بر محاسبه RMSE کل مسیر حرکت ربات، این معیار صرفاً برای قسمت‌هایی از مسیر حرکت ربات که در آن الگوریتم موفق عمل کرده نیز محاسبه می‌شود. این کار با استفاده از مقایسه زمان ثبت شده به ازای مختصات به دست آمده توسط الگوریتم و زمان ثبت شده به ازای اطلاعات واقعیت زمینی انجام

می‌گیرد. به این ترتیب می‌توان دقت الگوریتم‌ها را فارغ از قسمت‌هایی از مسیر که به دلیل شرایط محیطی یا شرایط حرکت ربات شکست خورده‌اند، بررسی کرد.

#### ۴-۱-۳- دقت نسبی تخمین طول مسیر حرکت

معیار بعدی برای بررسی و مقایسه دقت الگوریتم‌ها درصد خطای نسبی تخمین طول مسیر حرکت ربات می‌باشد. برای بدست آوردن این معیار از رابطه ۴ استفاده می‌شود.

$$\times 100E = \frac{|Dg - Dc|}{Dg} \quad (4)$$

در رابطه بالا  $Dg$  نشان دهنده طول مسیر طی شده واقعی و  $Dc$  نشان دهنده طول مسیر محاسبه شده است. طول مسیر حرکت ربات با استفاده از مجموع طول پاره خط‌هایی که بین دو موقعیت متوالی ربات قرار دارند، تعیین می‌شود. طول واقعی از داده‌های واقعیت زمینی، و طول به دست آمده توسط الگوریتم، از مختصات که الگوریتم قبل از اعمال تبدیل سه بعدی به دست آورده، محاسبه می‌شود.

#### ۴-۲- معیارهای بررسی شرایط روشنایی محیط و نحوه حرکت ربات

معیار بررسی شرایط روشنایی محیط با استفاده از بررسی تصاویر و معیارهای بررسی نحوه حرکت ربات با استفاده از اطلاعات واقعیت زمینی به دست می‌آیند. به این ترتیب می‌توان عملکرد الگوریتم‌های منتخب را در شرایط محیطی مختلف و در انواع حرکات ربات بررسی کرد. در ادامه هر دو گروه از معیارها شرح داده می‌شوند.

#### ۴-۲-۱- روشنایی محیط

برای بررسی شرایط روشنایی فضای حرکت ربات، می‌توان از درجات خاکستری تصاویر استفاده کرد. برای این منظور می‌توان میانگین درجات خاکستری پیکسل‌های تصاویر را در هر لحظه مشخص، محاسبه کرده و میزان روشنایی تصاویر اخذ شده توسط ربات در طول مسیر حرکت را ارزیابی کرد. در مورد الگوریتم‌هایی که از تصاویر دوربین استریو استفاده می‌کنند بایستی میانگین درجات

خاکستری پیکسل‌های مربوط به هر جفت تصویر استریو محاسبه شود.

#### ۴-۲-۲- نحوه حرکت ربات

نحوه حرکت ربات با دو پارامتر (۱) سرعت خطی و سرعت زاویه‌ای و (۲) پیچیدگی مسیر حرکت ربات ارزیابی می‌شود.

- سرعت خطی و زاویه‌ای:

جهت بررسی نحوه حرکت ربات از اطلاعات واقعیت زمینی از جمله سرعت خطی حرکت ربات و سرعت زاویه‌ای ربات در هر لحظه استفاده می‌گردد. به این ترتیب می‌توان اثر سرعت ربات را بر روی عملکرد و دقت الگوریتم‌های تعیین موقعیت تحلیل و بررسی کرد.

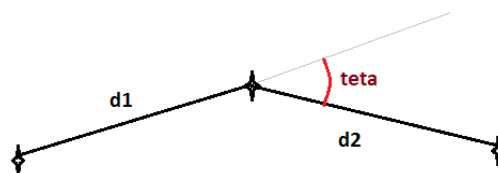
- پیچیدگی مسیر حرکت:

فارغ از سرعت خطی و زاویه‌ای ربات در هر لحظه، شرایط هندسی مسیر حرکت آن در نحوه عملکرد الگوریتم‌ها موثر است. روش‌های تعیین موقعیت دیداری اغلب در مورد تخمین مسیرهای مستقیم و مسیرهای با پیچیدگی زیاد کارکرد و دقت متفاوتی دارند. لذا بررسی شرایط هندسی مسیر حرکت ربات نیز از اهمیت زیادی برخوردار است. برای نشان دادن این موضوع، در این پژوهش میانگین زوایای مسیر حرکت ربات محاسبه شده است. در رابطه ۵ نحوه محاسبه زوایای مسیر حرکت بیان شده است که در نهایت با محاسبه میانگین این مقادیر می‌توان معیار مد نظر را بدست آورد.

$$\theta = |\cos^{-1} \frac{d1 \cdot d2}{|d1| \times |d2|}| \quad (5)$$

در رابطه فوق،  $\theta$  زاویه بین دو بخش مستقیم از مسیر پرواز، و  $d1$  و  $d2$  بردارهای مربوط به این دو بخش مستقیم هستند. زاویه محاسبه شده در صفحه‌ای قرار دارد که دو بردار بر روی آن قرار گرفته‌اند. هر بخش مستقیم مسیر، خط ما بین دو نقطه متوالی از مسیر است که موقعیت آن‌ها از اطلاعات واقعیت زمینی استخراج شده است. در شکل ۲ زاویه  $\theta$  نمایش داده می‌شود.



شکل ۲- بردارهای  $d_1$  و  $d_2$  و زاویه  $\theta$ 

## ۵- پیاده سازی و ارزیابی نتایج

در این قسمت ابتدا داده‌های مورد استفاده شرح داده شده و سپس در مورد چگونگی پیاده سازی الگوریتم‌ها بر روی ربات پرنده و نتایج آن بحث می‌شود. پس از پیاده سازی الگوریتم‌های منتخب بر روی ربات پرنده، ابتدا معیارهای استحکام الگوریتم‌ها و دقت تعیین موقعیت

### ۵-۱- داده مورد استفاده

داده‌های EuRoC چندین مجموعه داده آماده جهت ارزیابی الگوریتم‌های مختلف تعیین موقعیت و مدل سازی سه بعدی در محیط‌های داخلی می‌باشد.

جدول ۱- پارامترهای کالیبراسیون دوربین مورد استفاده در پروژه EuRoC

مقدار	پارامتر کالیبراسیون	مقدار	پارامتر کالیبراسیون	مقدار	پارامتر کالیبراسیون
۰	k4	۰/۰۷۳۹۵۹۰۷	k1	۳۶۷/۲۱۵	cx
۴۵۸/۶۵۴	fx	۰/۰۰۰۱۹۳۵۹	k2	۲۴۸/۳۷۵	cy
۴۵۷/۲۹۶	fy	۰/۰۰۰۰۱۷۶۲	k3	-۰/۲۸۳۴۰۸۱۱	k0

جدول ۲- اطلاعات مربوط به داده‌های EuRoC

ویژگی	سرعت زاویه‌ای متوسط (رادیان بر ثانیه)	سرعت خطی متوسط (متر بر ثانیه)	زمان پرواز (ثانیه)	طول مسیر پرواز (متر)	شماره و نام پرواز
بافت خوب، صحنه روشن	۰/۲۲	۰/۴۴	۱۸۲	۸۰/۶	شماره ۱- آسان
حرکت سریع، صحنه روشن	۰/۲۹	۰/۹۹	۱۳۲	۱۳۰/۹	شماره ۳- متوسط
حرکت سریع، صحنه تاریک	۰/۲۱	۰/۸۸	۱۱۱	۹۷/۶	شماره ۵- دشوار

کالیبراسیون دوربین در جدول شماره ۱ به طور کامل ذکر شده‌است. سه سری داده شماره ۱ (آسان)، شماره ۳ (متوسط) و شماره ۵ (دشوار) از بین این پنج داده جهت استفاده در این پژوهش انتخاب شده‌اند که اطلاعات تکمیلی آن‌ها به صورت تقریبی در جدول ۲ ذکر شده‌است. مسیر پرواز اول و سوم و پنجم به عنوان پروازهای منتخب برای این تحقیق در شکل ۵ نمایش داده شده‌است. این مسیرها از داده‌های واقعیت زمینی استخراج شده و دارای دقت ۱ میلی متر هستند.

مجموعه تهیه‌شده جهت آزمایش الگوریتم‌ها و روش‌های تعیین موقعیت ربات در محیط داخلی، با استفاده از یک ربات پرنده کوچک که مجهز به یک جفت دوربین استریو global shutter از نوع MT9V034 دارای سرعت ۲۰ فریم بر ثانیه و ابعاد سنسور ۷۵۲ در ۴۸۰ پیکسل و یک حسگر اینرشیا (IMU) <sup>۱</sup> (ممز<sup>۲</sup> دارای سرعت ۲۰۰ هرتز و یک سیستم ردیابی لیزری برای تهیه داده‌های واقعیت زمینی می‌باشد، در ۵ سری با استفاده از پنج نوبت پرواز مجزا برداشت شده‌است [۳۲]. پارامترهای

<sup>۱</sup> Inertial Measurement Unit<sup>۲</sup> MEMS

گردید. این الگوریتم بر روی هر سه داده منتخب EuRoC پیاده سازی شد. اما در مورد داده شماره ۵ به دلیل سرعت بالای پهپاد هنگام اخذ تصاویر و همچنین صحنه تاریک، الگوریتم PL-SVO به طور کلی نا موفق عمل کرده و موقعیت ربات پرنده را ارائه نداد. به این ترتیب سرعت پایین الگوریتم PL-SVO در این مورد مشهود بوده و نقطه ضعف این روش محسوب می‌گردد.

الگوریتم ORB-SLAM 2 تحت سیستم عامل ROS<sup>۳</sup> که بر روی سیستم عامل ubuntu 14.04 نصب شده بود، با استفاده از کتابخانه OpenCV نسخه ۳ و کتابخانه Eigen نسخه ۳/۳ پیاده سازی و خروجی‌های آن از پوشه مربوط به الگوریتم در سیستم عامل ubuntu استخراج گردید. این الگوریتم نیز بر روی هر سه سری داده منتخب EuRoC اجرا شده و در هر سه مورد موقعیت ربات پرنده را ارائه داد.

الگوریتم vins fusion نیز بر روی سیستم عامل ROS که بر روی سیستم عامل ubuntu 18.04 نصب شده بود با کمک کتابخانه Ceres Solver اجرا شده و خروجی‌های آن از طریق ثبت داده‌های مربوط به تاپیک‌های سیستم عامل ROS استخراج گردید. این الگوریتم دارای عملکرد مشابه با الگوریتم ORB-SLAM 2 می‌باشد. تفاوت عمده آن با الگوریتم ORB-SLAM 2 قابلیت استفاده از حسگر اینرشیا است. به همین علت این الگوریتم بدون از دست دادن موقعیت ربات در طول مسیر، بر روی هر سه مجموعه داده ۱ و ۳ و ۵ پیاده سازی شد.

### ۳-۵- مقایسه و ارزیابی نتایج پیاده سازی الگوریتم‌های منتخب بر روی داده‌های EuRoC

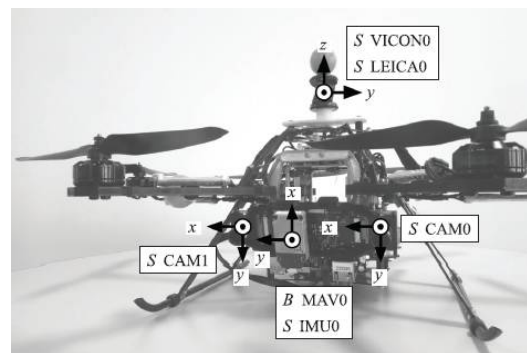
در ادامه با استفاده از معیارهای پیشنهاد شده در بخش‌های قبلی، الگوریتم‌های تعیین موقعیت ربات مورد مقایسه و ارزیابی قرار گرفته و نتایج پیاده سازی آن‌ها بر روی داده‌های EuRoC تحلیل و بررسی می‌شود.

#### ۳-۵-۱- استحکام الگوریتم‌ها

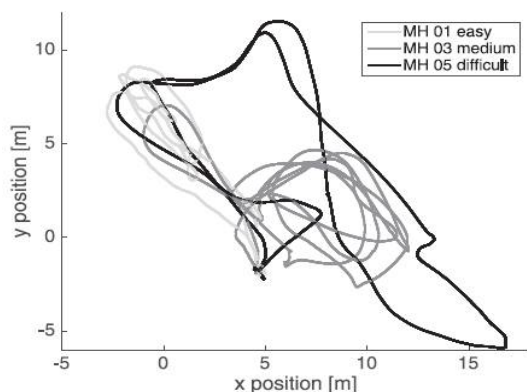
ابتدا علل شکست الگوریتم‌های منتخب در طول مسیر پرواز بررسی می‌شود. در شکل ۶ نمودار زمان و تعداد



شکل ۳- فضای داخلی پرواز پهپاد EuRoC [۳۲]



شکل ۴- پهپاد پروژه EuRoC و حسگرهای متصل به آن [۳۲]



شکل ۵- مسیر پرواز اول و سوم و پنجم پهپاد EuRoC از منظر بالا [۳۲]

### ۳-۵-۲- پیاده سازی الگوریتم‌های منتخب بر روی داده‌های EuRoC

الگوریتم PL-SVO بر روی سیستم عامل ubuntu 14.04 و با استفاده از کتابخانه OpenCV<sup>۱</sup> نسخه ۳ و برنامه MRPT<sup>۲</sup> اجرا شده و نتایج آن از فایل موجود در پوشه مربوط به الگوریتم در سیستم عامل ubuntu استخراج

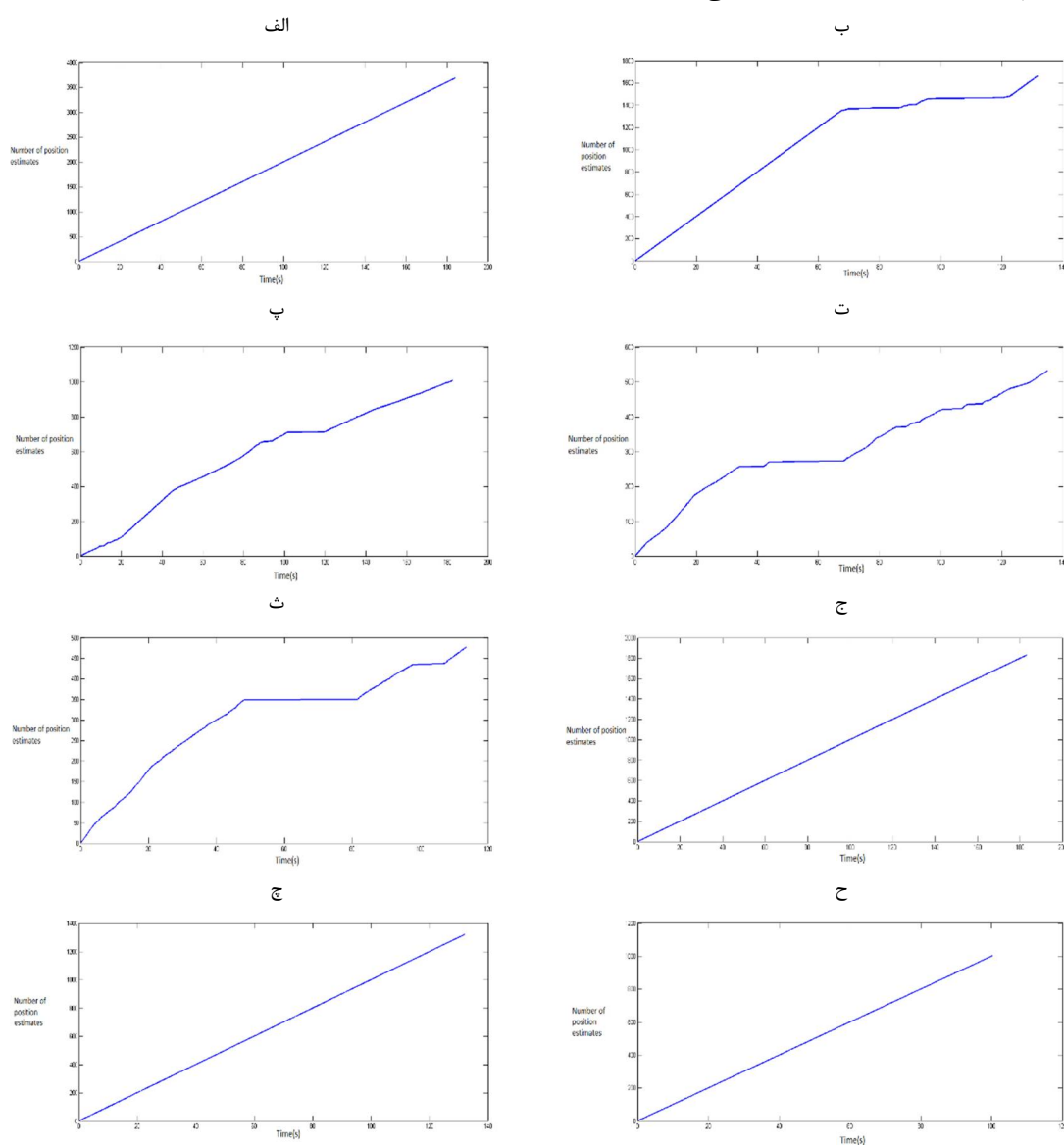
<sup>۱</sup> Open Computer Vision

<sup>۲</sup> Mobile Robot Programming Toolkit

<sup>۳</sup> Robot Operating System

موقعیت‌های محاسبه شده برای ربات به صورت تجمعی نمایش داده شده‌است. حالت خطی شیب دار این نمودار نشان دهنده نرخ ثابت محاسبه موقعیت ربات نسبت به زمان می‌باشد. در نتیجه می‌توان گفت در هیچ قسمتی از مسیر، الگوریتم موقعیت ربات را از دست نداده و مختصات سه بعدی ربات به صورت پیوسته تخمین زده شده‌است. اما در صورتی که نمودار نزدیک به حالت افقی شود، می‌توان نتیجه گرفت که الگوریتم در آن بازه زمانی موقعیت جدیدی برای ربات محاسبه نکرده‌است. با توجه به شکل ۶ (قسمت الف) ملاحظه می‌شود که الگوریتم PL-SVO در پرواز شماره ۱ (آسان) به طور کلی اجرا شده

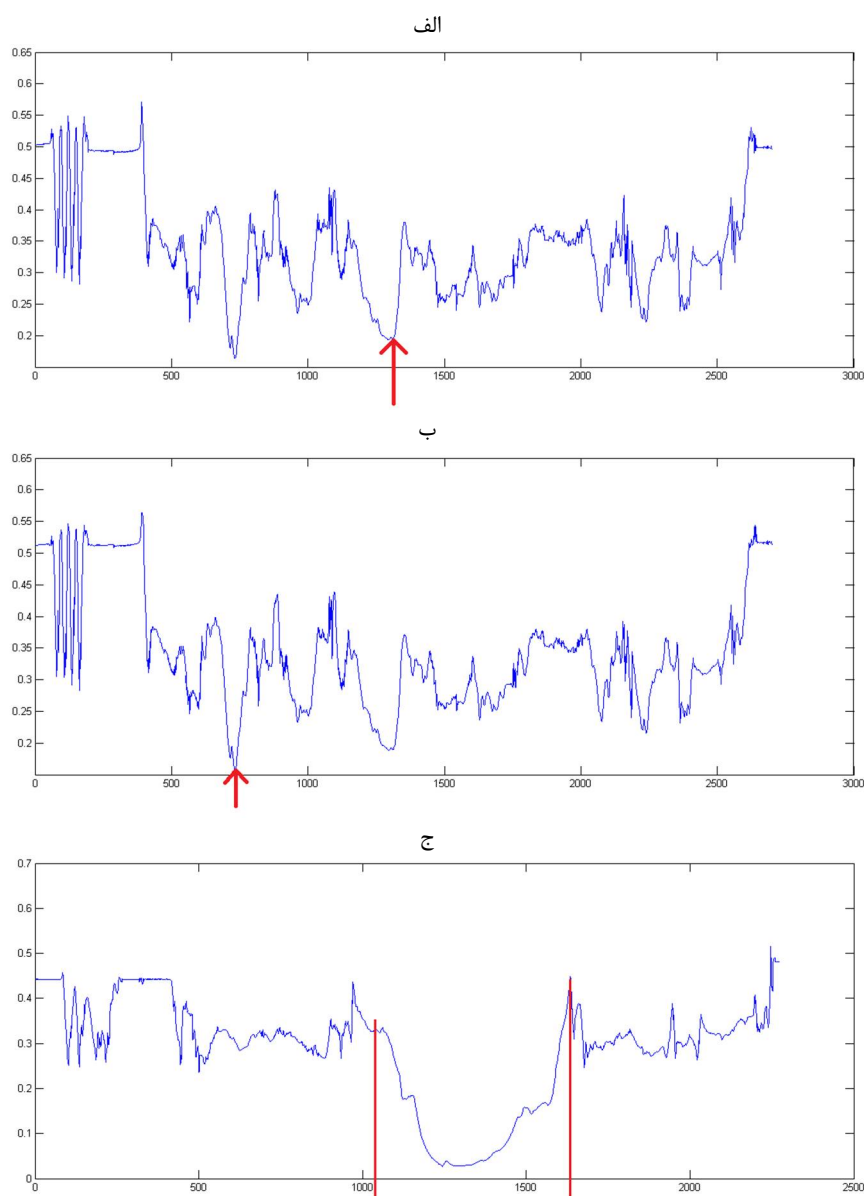
اما در پرواز شماره ۳ (متوسط) بخشی از مسیر را از دست داده‌است (شکل ۶ قسمت ب). الگوریتم ORB-SLAM 2 هم در مجموعه داده شماره ۱ (آسان)، به جز قسمت‌های کوچکی از مسیر، موفق عمل کرده و موقعیت ربات را تخمین زده‌است (شکل ۶ قسمت پ). اما در پرواز شماره ۳ (متوسط)، الگوریتم در زمان قابل توجهی از پرواز شکست خورده و موقعیت ربات پرنده را از دست داده‌است (شکل ۶ قسمت ت). بنابراین ابتدا علت شکست الگوریتم‌های PL-SVO و ORB-SLAM 2 در پرواز شماره ۳ (متوسط) بررسی می‌شود.



شکل ۶- نمودار تعداد موقعیت‌های محاسبه شده توسط الگوریتم‌های تعیین موقعیت، به صورت تجمعی نسبت به زمان. الف: PL-SVO به ازای پرواز شماره یک- ب: PL-SVO به ازای پرواز شماره سه- پ: ORB-SLAM 2 به ازای پرواز شماره یک- ت: ORB-SLAM 2 به ازای پرواز شماره سه- ث: ORB-SLAM 2 به ازای پرواز شماره پنج- ج: vins fusion به ازای پرواز شماره یک- چ: vins fusion به ازای پرواز شماره سه- ح: vins fusion به ازای پرواز شماره پنج

در مورد الگوریتم PL-SVO با توجه به شکل ۶ (قسمت ب) در پرواز شماره ۳ از ثانیه ۷۰ تا ۱۲۰ پس از شروع پرواز، موقعیت پرنده از دست رفته‌است. در این مدت سرعت زاویه‌ای پرنده تغییر چندانی نکرده‌است. اما میانگین سرعت خطی پرنده در این بازه زمانی برابر ۱/۱۲۷ متر بر ثانیه می‌باشد. در حالی که میانگین سرعت خطی ربات پرنده در کل پرواز شماره ۳ برابر ۰/۹۹ متر بر ثانیه است. بنابراین سرعت خطی ربات پرنده در بازه زمانی که موقعیت آن در الگوریتم PL-SVO از دست رفته‌است،

حدود ۱۴ درصد بیشتر از میانگین سرعت ربات پرنده در کل مسیر می‌باشد. برای الگوریتم ORB-SALM 2 نیز به ازای داده‌های مربوط به پرواز شماره ۳ (شکل ۶ قسمت ت)، از ثانیه ۳۵ تا ثانیه ۷۰ پس از شروع پرواز موقعیت ربات از دست رفته‌است. در این بازه زمانی هم سرعت زاویه‌ای تفاوت چندانی با مقدار میانگین ندارد اما میانگین سرعت خطی برابر با ۱/۴۳۵ متر بر ثانیه است که ۴۵ درصد بیشتر از میانگین سرعت خطی کل مسیر می‌باشد.



شکل ۷- الف: نمودار میانگین درجات خاکستری تصاویر مربوط به دوربین سمت راست برای پرواز شماره ۳ (علامت قرمز رنگ: میانگین درجه خاکستری تصویر مربوط به ثانیه ۷۰) - ب: نمودار میانگین درجات خاکستری تصاویر مربوط به دوربین‌های سمت راست و چپ برای پرواز شماره ۳ (علامت قرمز رنگ: میانگین درجه خاکستری زوج تصویر مربوط به ثانیه ۳۵) - ج: نمودار میانگین درجات خاکستری تصاویر مربوط به دوربین‌های سمت راست و چپ برای پرواز شماره ۵ (علامت قرمز رنگ: میانگین درجه خاکستری زوج تصویر مربوط به بازه ثانیه ۵۰ تا ثانیه ۸۰)

بنابراین هر دو الگوریتم در زمان‌هایی شکست خورده‌اند که در آن میانگین سرعت خطی ربات بیشتر از میانگین سرعت خطی ربات در کل مسیر است. اما هر کدام از الگوریتم‌ها در بازه متفاوتی، موقعیت ربات را از دست داده‌اند. برای بررسی بیشتر علت این مورد، سطح روشنایی تصاویر بایستی بررسی شود. در شکل ۷ نمودار روشنایی تصاویر در پرواز شماره ۳ و ۵ نمایش داده شده‌است. محور افقی، شماره تصاویر و محور عمودی شدت درجه خاکستری میانگین هر تصویر در مقیاس صفر (تاریک) تا یک (روشن) می‌باشد. البته برای الگوریتم PL-SVO میانگین درجات خاکستری تصاویر مربوط به دوربین سمت راست محاسبه شده‌اند.

اما برای الگوریتم ORB-SLAM 2 که از جفت دوربین استریو استفاده کرده‌است، میانگین درجات خاکستری تصاویر مربوط به هر دو دوربین سمت چپ و راست محاسبه شده‌است. علاوه بر این با توجه به نرخ تصویر برداری نسبت به زمان (۲۰ فریم بر ثانیه) و زمان کل پرواز، درجات خاکستری تصاویر مربوط به ثانیه ۷۰ پس از شروع پرواز (که الگوریتم PL-SVO از آن زمان موقعیت ربات را از دست داده‌است) و ثانیه ۳۵ پس از شروع پرواز (که الگوریتم ORB-SLAM 2 از آن زمان موقعیت ربات را از دست داده‌است) در شکل ۷ (قسمت الف و قسمت ب) مشخص شده‌است.

با توجه به نمودارهای شکل ۷ (قسمت الف و قسمت ب) مشاهده می‌شود که در ابتدای از دست رفتن موقعیت ربات در هر دو الگوریتم PL-SVO و ORB-SLAM 2 تصاویر دارای درجات خاکستری بسیار پایین بوده‌اند. به این معنی که ربات پرنده در فضای تاریک قرار گرفته و الگوریتم‌ها موفق به استخراج عوارض تصاویر و تطبیق آن‌ها با یکدیگر نشده‌اند. پس از قرار گرفتن ربات در فضای تاریک، الگوریتم‌ها به مدت چند ده ثانیه توانایی بازیابی موقعیت ربات را نداشته و مسیر را از دست داده‌اند. البته سرعت خطی بالا در این بازه‌های زمانی هم می‌تواند در این امر تاثیرگذار باشد.

در مورد پرواز شماره ۵ به دلیل سرعت بالا و تاریکی صحنه، الگوریتم PL-SVO به طور کلی موفق به محاسبه موقعیت ربات پرنده نشد. اما الگوریتم ORB-SLAM 2 مسیر حرکت ربات را به صورت ناقص تخمین زد. با توجه به شکل ۶ (قسمت ث) این الگوریتم در زمان‌های ۵۰ ثانیه تا

۸۰ ثانیه بعد از شروع پرواز و همچنین ۹۵ تا حدود ۱۰۵ ثانیه پس از شروع پرواز، موقعیت ربات پرنده را از دست داده‌است. در این بازه‌های زمانی نیز میانگین سرعت زاویه‌ای ربات تفاوت چندانی با میانگین کل نداشته اما میانگین سرعت خطی ربات بیشتر از میانگین کل می‌باشد. در بازه ۵۰ تا ۸۰ ثانیه پس از شروع پرواز میانگین سرعت خطی برابر ۰/۹۹۹ متر بر ثانیه است که حدود ۱۳ درصد بیشتر از میانگین کل می‌باشد. همچنین در بازه ۹۵ تا ۱۰۵ ثانیه از شروع پرواز نیز میانگین سرعت خطی برابر ۱/۰۱۴ متر بر ثانیه است که ۱۵ درصد از میانگین کلی بیشتر می‌باشد. بنابراین، در هر دو بازه شکست الگوریتم، سرعت خطی نسبت به میانگین کل مسیر بیشتر بوده‌است. اما در این مورد نیز دلیل اصلی از دست رفتن موقعیت ربات پرنده، مسئله تاریک بودن فضای اخذ تصاویر می‌باشد. در شکل شماره ۷ (قسمت ج) میانگین درجات خاکستری تصاویر در بازه ۵۰ تا ۸۰ ثانیه پس از شروع پرواز که مهم‌ترین قسمت از دست رفته موقعیت ربات می‌باشد، نشان داده شده‌است.

با توجه به شکل ۷ (قسمت ج) ملاحظه می‌شود که در بازه ۵۰ ثانیه تا ۸۰ ثانیه پس از شروع پرواز که الگوریتم ORB-SLAM 2 در آن شکست خورده‌است، میانگین درجات خاکستری تصاویر بسیار پایین می‌باشد. این امر نشان می‌دهد که در این بازه زمانی، ربات پرنده در محلی تاریک پرواز می‌کرده و بدین سبب الگوریتم موفق به استخراج عوارض تصویری نشده و شکست خورده‌است.

الگوریتم‌های غیر مستقیم تعیین موقعیت مبتنی بر تصاویر، پس از استخراج عوارض تصویری و تطبیق آن‌ها با یکدیگر، علاوه بر بدست آوردن موقعیت سه بعدی عوارض استخراج شده، اقدام به محاسبه موقعیت مراکز تصاویر اخذ شده می‌کنند. به دلیل اتصال دوربین‌های تصویر برداری به ربات پرنده، موقعیت مراکز تصویر می‌تواند با اعمال یک مقدار مشخص انتقال، به عنوان موقعیت ربات در نظر گرفته شود. این مقدار انتقال در قالب تبدیل متشابه سه بعدی در کنار سایر پارامترها و در داخل ماتریس انتقال به همراه مقدار کلی انتقال بین دو سیستم، اعمال شده و موقعیت ربات به دست می‌آید. اما در صورت تاریکی تصاویر یا پایین بودن مقدار کنتراست تصاویر، عوارض تصویری قابل استخراج نبوده و الگوریتم شکست می‌خورد. به همین جهت روشی برای حرکت ربات، برای الگوریتم‌های مبتنی بر تصاویر بسیار مهم

می‌باشد. این الگوریتم‌ها حساسیت پایینی به جنس و دما و فشار محیط اطراف دارند. برای نمونه قابلیت اجرا در محیط زیر آب و فضای خارج از جو و حتی بر روی سطح سایر سیارات برای این الگوریتم‌ها قابل تصور است. اما در صورت عدم روشنایی صحنه یا عدم وجود بافت مناسب و ویژگی‌های بارز تصویری، با مشکل جدی مواجه می‌شوند.

به این ترتیب می‌توان نتیجه گرفت که الگوریتم‌های تعیین موقعیت مبتنی بر تصاویر با وجود استحکام بالا در مقابل پارامترهای محیطی، در برابر شرایط نوری فضای حرکت ربات، استحکام کمی داشته و در مقابل این پارامتر حساسیت زیادی نشان می‌دهند.

الگوریتم vins fusion به دلیل استفاده از حسگر اینرشیا در کنار دوربین تصویر برداری در هنگام قرار گرفتن ربات در محیط‌های تاریک، موقعیت آن را از دست نداده و ادامه مسیر ربات را تخمین می‌زند. با توجه به شکل ۶ (قسمت ج، چ، ح) ملاحظه می‌شود که هر سه نمودار مربوط به الگوریتم vins fusion به صورت خطی با شیب ثابت هستند. یعنی نرخ محاسبه موقعیت‌های جدید برای ربات نسبت به زمان، در الگوریتم vins fusion یکسان است. این امر نشان می‌دهد که الگوریتم به صورت پیوسته موقعیت ربات پرنده را تخمین زده‌است. بنابراین وابسته نبودن الگوریتم تعیین موقعیت، صرفاً به داده‌های تصویری می‌تواند سبب استحکام آن در برابر شرایط نوری محیط باشد.

### ۵-۳-۲- دقت تعیین موقعیت الگوریتم‌ها

همان طور که گفته شد پس از انتقال مختصات محاسبه شده برای ربات، به سیستم مختصات اطلاعات واقعیت زمینی با استفاده از تبدیل متشابه سه بعدی، می‌توان از اطلاعات واقعیت زمینی به عنوان مرجع استفاده کرده و دقت تعیین موقعیت الگوریتم‌های مختلف را با استفاده از آن مورد ارزیابی قرار داد. در شکل ۸ نمودار مربوط به تخمین مسیر حرکت ربات توسط الگوریتم‌های منتخب به همراه نمودار مسیر استخراج شده از اطلاعات واقعیت زمینی نشان داده شده‌است. به این ترتیب می‌توان دقت الگوریتم‌ها را به صورت بصری ارزیابی و مقایسه کرد. علاوه بر آن، مقدار RMSE بدست آمده برای الگوریتم‌های مختلف به ازای پروازهای ۱ و ۳ و ۵ نیز در همین سیستم مختصات محاسبه شده‌است. نمودار مربوط به الگوریتم PL-SVO به ازای داده شماره ۵ به دلیل شکست الگوریتم و عدم اجرای

آن بر روی این سری داده، در شکل ۸ موجود نمی‌باشد. همانطور که در شکل ۸ (قسمت ج، چ، ح) ملاحظه می‌شود الگوریتم vins fusion دارای دقت بهتری در هر سه پرواز ۱ و ۳ و ۵ می‌باشد. البته از مقایسه خطوط قرمز و آبی نمودارهای مربوط به الگوریتم PL-SVO و ORB-SLAM 2 در پروازهای ۳ و ۵ می‌توان دریافت که الگوریتم‌ها در قسمتی از مسیر پرواز شکست خورده و موقعیت ربات را محاسبه نکرده‌اند. این واقعیت از نمودارهای مربوط به شکل ۶ نیز قابل دریافت است. همچنین با توجه به این نمودارها می‌توان نتیجه گرفت که الگوریتم ORB-SLAM 2 در قسمت‌هایی که موفق عمل کرده و موقعیت ربات را به دست آورده‌است، دقت بالایی دارد.

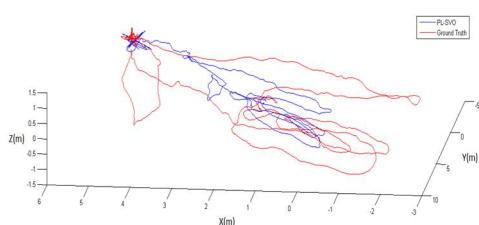
به عنوان نمونه، در شکل ۸ (قسمت ث) نمودار مربوط به الگوریتم ORB-SLAM 2 به ازای داده‌های پرواز شماره ۵ نمایش داده شده‌است. با توجه به شکل مشاهده می‌شود که دقت این الگوریتم در مواقعی که موفق به تخمین موقعیت ربات شده‌است بسیار بالا بوده و می‌توان گفت فارغ از قسمت‌های شکست خورده عملکرد بهتری نسبت به الگوریتم vins fusion دارد.

با توجه به مطالب عنوان شده، برای مقایسه الگوریتم‌ها، در جدول ۳ مقدار معیار RMSE برای الگوریتم‌های PL-SVO به ازای تمام مسیر پرواز و به ازای بازه‌هایی که الگوریتم موفق به تخمین موقعیت شده‌است و ORB-SLAM 2 به ازای تمام مسیر پرواز و به ازای بازه‌هایی که الگوریتم موفق به تخمین موقعیت شده‌است و vins fusion به ازای تمام مسیر پرواز، برای هر سه مجموعه داده نوشته شده‌است. با توجه به جدول ۳ نیز ملاحظه می‌شود که به ازای تمام مسیر پرواز، الگوریتم vins fusion بهترین دقت را در تعیین موقعیت ربات پرنده دارد. اما اگر صرفاً قسمتی از مسیر پرواز که الگوریتم در آن شکست نخورده‌است در نظر گرفته شود، الگوریتم ORB-SLAM 2 دقت بیشتری دارد. در شکل ۹ نمودار مربوط به اطلاعات جدول ۳ نمایش داده شده‌است. بایستی توجه شود که RMSE الگوریتم PL-SVO پس از اعمال مقیاس محاسبه شده‌است. مقیاس مسیر به دست آمده برای الگوریتم PL-SVO به دلیل اینکه این الگوریتم از تصاویر دوربین تکی استفاده می‌کند، مقدار واقعی نبوده و نیاز است که طی تبدیل متشابه سه بعدی ضریب مقیاس مناسب اعمال شده و مختصات مربوط به مسیر

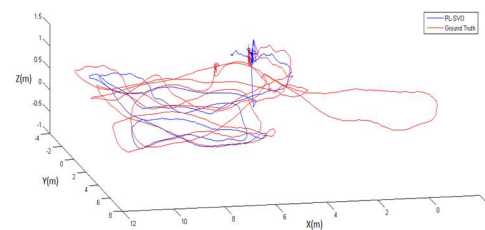
حرکت ربات، در فضای متریک محاسبه گردد. برای درک موضوع اثر ضریب مقیاس، در شکل ۱۰ نمودار مربوط به الگوریتم PL-SVO به ازای پرواز شماره ۳ قبل از اعمال مقیاس، با نمودار مربوط به اطلاعات واقعیت زمینی

مقایسه شده است. با مشاهده شکل ۱۰ می توان دریافت که ابهام مقیاس می تواند اثرات بسیار زیادی بر روی مختصات به دست آمده از عملیات تعیین موقعیت با استفاده از داده های تک دوربین داشته باشد.

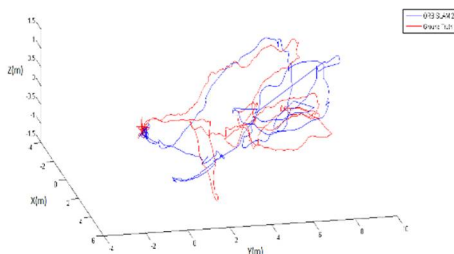
الف



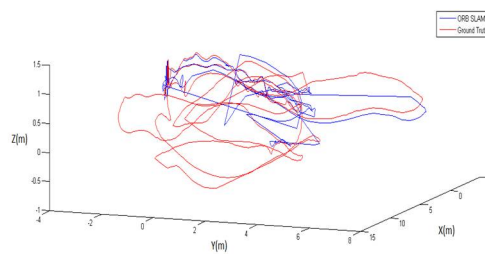
ب



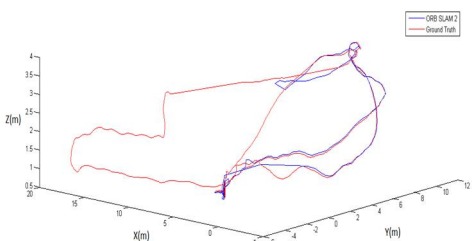
پ



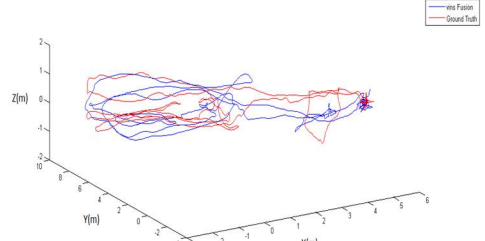
ت



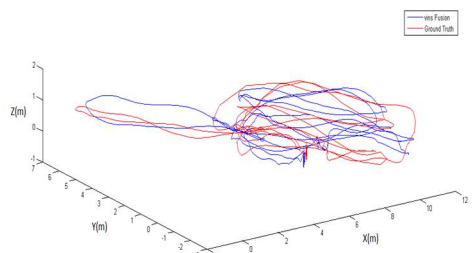
ث



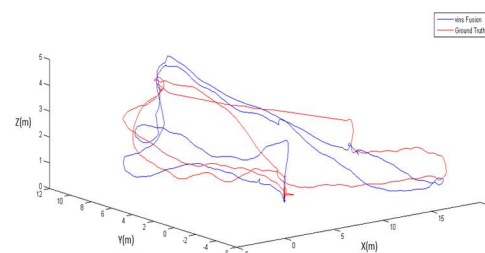
ج



چ



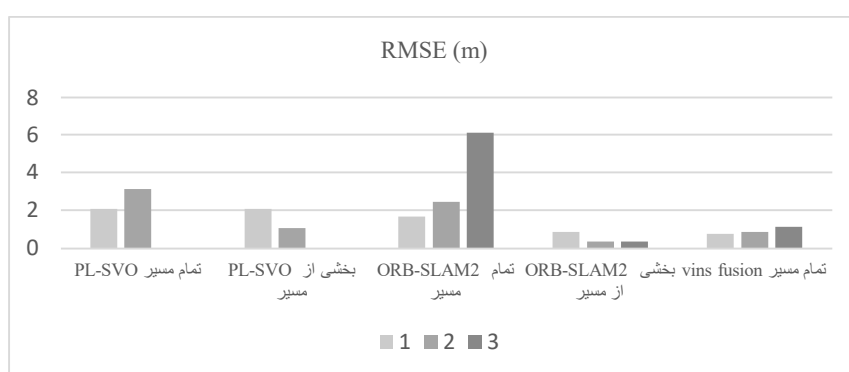
ح



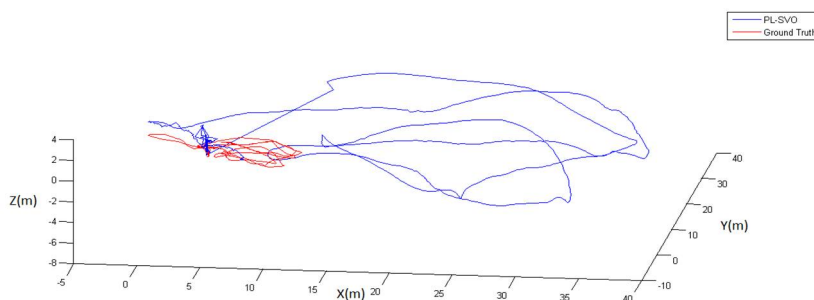
شکل ۸- نمودار مسیر محاسبه شده برای ربات با استفاده از الگوریتم های منتخب (آبی) و نمودار مسیر استخراج شده ربات با استفاده از اطلاعات واقعیت زمینی (قرمز) در حالت سه بعدی. الف: PL-SVO به ازای پرواز شماره یک- ب: PL-SVO به ازای پرواز شماره سه- پ: ORB-SLAM 2 به ازای پرواز شماره یک- ت: ORB-SLAM 2 به ازای پرواز شماره سه- ث: ORB-SLAM 2 به ازای پرواز شماره پنج- ج: vins fusion به ازای پرواز شماره یک- چ: vins fusion به ازای پرواز شماره سه- ح: vins fusion به ازای پرواز شماره پنج

جدول ۳- مقدار RMSE برای الگوریتم‌های منتخب به ازای پروازهای ۱ و ۳ و ۵

پرواز شماره ۵ (دشوار)	پرواز شماره ۳ (متوسط)	پرواز شماره ۱ (آسان)	الگوریتم تعیین موقعیت
-	۳/۱۱۷ متر	۲/۰۹ متر	PL-SVO (تمام مسیر پرواز)
-	۱/۰۴ متر	۲/۰۹ متر	PL-SVO (بخشی از مسیر پرواز که الگوریتم موفق بوده)
۶/۱۲۸ متر	۲/۴۵۷ متر	۱/۶۷۹ متر	ORB-SLAM2 (تمام مسیر پرواز)
۰/۳۵۴ متر	۰/۳۵۵ متر	۰/۸۶۶ متر	ORB-SLAM2 (بخشی از مسیر پرواز که الگوریتم موفق بوده)
۱/۱۲ متر	۰/۸۳۹ متر	۰/۷۴۷ متر	vins fusion (تمام مسیر پرواز)



شکل ۹- نمودار میله‌ای مقدار RMSE برای الگوریتم‌های منتخب به ازای پروازهای ۱ و ۳ و ۵



شکل ۱۰- نمودار مسیر به دست آمده برای پرواز شماره ۳ با استفاده از الگوریتم PL-SVO (آبی) در مقایسه با واقعیت زمینی (قرمز) قبل از اعمال مقیاس

چندانی بین سه پرواز وجود ندارد. از لحاظ روشنایی نیز پرواز شماره ۱ وضعیت بهتری نسبت به پرواز شماره ۳ و ۵ دارد (فضایی با روشنایی بیشتر). اما با این حال در مواقعی که دو الگوریتم فوق الذکر موفق به تخمین موقعیت ربات شده‌اند، دقت در پرواز شماره ۱ کم‌تر از پرواز شماره ۳ و ۵ می‌باشد. دلیل این امر، حرکت ربات در خطوط نسبتاً مستقیم در پروازهای شماره ۳ و ۵ تشخیص داده شده‌است. در پرواز شماره ۱، به دلیل چرخش‌های زیاد ربات پرنده،

با مشاهده جدول ۳ و شکل ۹ می‌توان مسئله مهم دیگری را در نظر گرفته و بررسی کرد. الگوریتم PL-SVO و الگوریتم ORB-SLAM2 در بخش‌هایی از پرواز ۱ (آسان) که موفق به تخمین مسیر حرکت ربات شده‌اند دقت کم‌تری نسبت به بخش‌هایی از سایر پروازها (متوسط و دشوار) که موفق به تخمین موقعیت ربات شده‌اند را دارا می‌باشند. پرواز شماره ۱ سرعت خطی کم‌تری نسبت به پروازهای ۳ و ۵ دارد. در مورد سرعت زاویه‌ای تفاوت



دقت تعیین موقعیت کاهش یافته‌است. برای نشان دادن این موضوع میانگین زوایای مسیر حرکت ربات در هر سه پرواز با استفاده از رابطه ۵ محاسبه شده‌است. به این ترتیب مقدار این معیار برای پرواز شماره ۱ (آسان)  $24/17$  درجه، برای پرواز شماره ۳ (متوسط)  $20/6$  درجه و برای پرواز شماره ۵ (دشوار)  $20/06$  درجه بدست آمد.

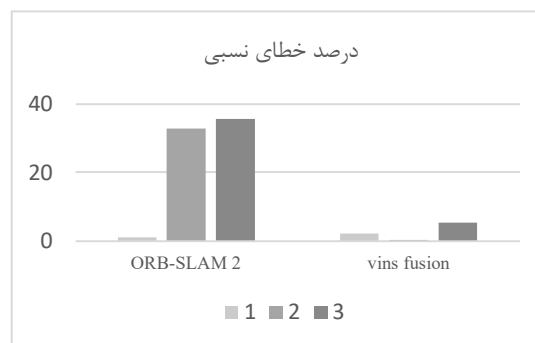
همانطور که بیان شد معیار زاویه نشان دهنده میزان چرخش ربات در طول پرواز و پیچیدگی مسیر حرکت آن می‌باشد. با توجه به اینکه این معیار برای پرواز شماره ۱ مقدار بیشتری دارد می‌توان گفت ربات پرنده در این پرواز چرخش بیشتری داشته و همین امر سبب شده دقت تخمین موقعیت الگوریتم‌های ORB-SLAM 2 و PL-SVO هنگام عملکرد موفق، در پرواز شماره ۱ کمتر از سایر پروازها باشد.

### ۵-۳-۳- دقت نسبی تخمین طول مسیر حرکت توسط الگوریتم‌ها

در ادامه دقت تخمین طول مسیر پرواز ربات، توسط الگوریتم‌های ORB-SLAM 2 و vins fusion (الگوریتم‌هایی که از دوربین استریو استفاده می‌کنند و ابهام مقیاس ندارند) با یکدیگر مقایسه می‌شود.

جدول ۴- درصد خطای نسبی تخمین طول مسیر پرواز توسط الگوریتم‌های ORB-SLAM 2 و vins fusion

پرواز شماره ۵ (دشوار)	پرواز شماره ۳ (متوسط)	پرواز شماره ۱ (آسان)	الگوریتم تعیین موقعیت
۳۵/۵۲	۳۲/۸۷	۱/۰۶	ORB-SLAM 2
۵/۳۷	۰/۲۶	۲/۱۲	vins fusion



شکل ۱۱- نمودار میله‌ای درصد خطای نسبی تخمین طول مسیر پرواز توسط الگوریتم‌های ORB-SLAM 2 و vins fusion

برای این منظور درصد خطای نسبی تخمین طول پرواز شماره ۱ و ۳ و ۵ توسط دو الگوریتم ذکر شده با استفاده از رابطه ۴ و اطلاعات واقعیت زمینی محاسبه و در جدول شماره ۴ و نمودار مربوط به شکل ۱۱ بیان شده‌است. با توجه به جدول ۴ و نمودار مربوط به شکل ۱۱ به جز پرواز شماره ۱ که خطای الگوریتم vins fusion مقدار ناچیزی بیشتر از خطای الگوریتم ORB-SLAM 2 می‌باشد، در دیگر موارد خطای الگوریتم ORB-SLAM 2 بسیار بیشتر از خطای الگوریتم vins fusion است. دلیل عمده این مسئله نیز از دست رفتن موقعیت ربات در بازه‌های زمانی طولانی در پروازهای ۳ و ۵ توسط الگوریتم ORB-SLAM 2 است. به این ترتیب الگوریتم vins fusion به دلیل تخمین موقعیت ربات در تمام طول مسیر، خطای نسبی کمتری در محاسبه طول مسیر پرواز نسبت به الگوریتم ORB-SLAM 2 دارد.

### ۶- بحث و نتیجه‌گیری

هدف اصلی این پژوهش تحلیل و بررسی عملکرد الگوریتم‌های مبتنی بر تصاویر در تعیین موقعیت ربات پرنده در یک محیط داخلی پر چالش بود. همچنین برای مقایسه دقت و استحکام و شناخت نقاط قوت و ضعف این الگوریتم‌ها روشی جدید ارائه گردید.

در مجموع می‌توان گفت که دقت الگوریتم ORB-SLAM 2 به شرط از دست ندادن موقعیت ربات، بیشترین مقدار را در بین سه الگوریتم منتخب دارد. الگوریتم vins fusion در رتبه بعدی قرار می‌گیرد و الگوریتم PL-SVO با وجود عملکرد مناسب در هنگام حرکت آهسته ربات در فضای داخلی، دقت و سرعت کمتری نسبت به دو الگوریتم دیگر دارد. دقت تعیین موقعیت الگوریتم ORB-SLAM 2 نسبت به الگوریتم vins-fusion بیشتر می‌باشد اما با توجه به اینکه الگوریتم ORB-SLAM 2 صرفاً متکی به داده‌های تصویری بوده و در فضای تاریک، موقعیت ربات را از دست می‌دهد، لذا در مواردی که ربات در محیط با روشنایی کم یا بافت تصویری ضعیف قرار می‌گیرد، الگوریتم vins fusion مناسب تر بوده و می‌تواند موقعیت ربات را به خوبی تخمین بزند. به این ترتیب می‌توان نتیجه گرفت که در صورت وجود روشنایی کافی و بافت مناسب صحنه، استفاده از داده‌های حسگر اینرشیا و ترکیب آن با داده‌های تصویری جهت تعیین موقعیت، می‌تواند دقت را کاهش

## ۷- پیشنهادات

برای پژوهش‌های آتی توسعه الگوریتمی که توانایی اجرای عملیات تعیین موقعیت در محیط‌های روشن با استفاده از دوربین تصویربرداری و در محیط‌های تاریک توسط دوربین تصویربرداری به همراه حسگر اینرشیا را داشته باشد پیشنهاد می‌گردد. به این ترتیب که الگوریتم به صورت پیش فرض فقط از داده‌های تصویری استفاده کند اما با محاسبه میانگین درجات خاکستری تصاویر، در هنگام تاریک شدن صحنه، روند اجرای عملیات تعیین موقعیت را تغییر داده و با ادغام اطلاعات سنسور اینرشیا با داده‌های تصویری مانع از دست رفتن موقعیت ربات شود.

دهد اما در صورت عدم روشنایی مناسب یا بافت مطلوب، حسگر اینرشیا می‌تواند مانع از دست رفتن موقعیت ربات شده و دقت و استحکام عملیات تعیین موقعیت ربات را افزایش دهد. در ضمن سرعت خطی زیاد ربات می‌تواند برای الگوریتم‌های تعیین موقعیت به خصوص الگوریتم PL-SVO مشکل ساز بوده و باعث از دست رفتن موقعیت ربات یا بازیابی دیرتر آن شود. در مورد تخمین طول مسیر حرکت ربات نیز می‌توان گفت در صورتی که تمام مسیر حرکت توسط الگوریتم تعیین موقعیت، تخمین زده شده و هیچ قسمتی از آن از دست نرود، دقت مناسب خواهد بود و در غیر این صورت محاسبه طول مسیر با خطای فاحشی انجام می‌گیرد.

## مراجع

- [۱] H. C. Longuet-Higgins, "A computer algorithm for reconstructing a scene from two projections," *Nature*, vol. 293, pp. 133-135, 1981.
- [۲] C. G. Harris and J. Pike, "3D positional integration from image sequences," *Image and Vision Computing*, vol. 6, pp. ۸۷-۹۰, ۱۹۸۸.
- [۳] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, *et al.*, "Building rome on a cloudless day," in *European Conference on Computer Vision*, 2010, pp. 368-381.
- [۴] F. Fraundorfer and D. Scaramuzza, "Visual odometry: Part i: The first 30 years and fundamentals," *IEEE Robotics and Automation Magazine*, vol. 18, pp. 80-92, 2011.
- [۵] D. Nistér, O. Naroditsky, and J. Bergen, "Visual odometry," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, 2004, pp. 1-1.
- [۶] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE robotics & automation magazine*, vol. 13, pp. 99-110, 2006.
- [۷] B. Gao, H. Lang, and J. Ren, "Stereo Visual SLAM for Autonomous Vehicles: A Review," in *2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2020, pp. 1316-1322.
- [۸] C. Debeunne and D. Vivet, "A review of visual-LiDAR fusion based simultaneous localization and mapping," *Sensors*, vol. 20, p. 2068, 2020.
- [۹] A. Pfrunder, P. V. Borges, A. R. Romero, G. Catt, and A. Elfes, "Real-time autonomous ground vehicle navigation in heterogeneous environments using a 3D LiDAR," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 2601-2608.
- [۱۰] H. P. Moravec, "Obstacle avoidance and navigation in the real world by a seeing robot rover," Stanford Univ CA Dept of Computer Science 1980.
- [۱۱] K. Yousif, A. Bab-Hadiashar, and R. Hoseinnezhad, "An overview to visual odometry and visual SLAM: Applications to mobile robotics," *Intelligent Industrial Systems*, vol. 1, pp. 289-311, 2015.
- [۱۲] J. Zhang and S. Singh, "LOAM: Lidar Odometry and Mapping in Real-time," in *Robotics: Science and Systems*, 2014.
- [۱۳] M. O. Aqel, M. H. Marhaban, M. I. Saripan, and N. B. Ismail, "Review of visual odometry: types, approaches, challenges, and applications," *Springerplus*, vol. 5, p. 1897, 2016.
- [۱۴] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual SLAM algorithms: a survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, pp. 1-11, 2017.
- [۱۵] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison, "DTAM: Dense tracking and mapping in real-time," in *Computer Vision (ICCV), 2011 IEEE International Conference on*, 2011, pp. 2320-2327.
- [۱۶] J. Engel, T. Schöps, and D. Cremers, "LSD-SLAM: Large-scale direct monocular SLAM," in *European Conference on Computer Vision*, 2014, pp. 834-849.
- [۱۷] A. Concha Belenguer and J. Civera Sancho, "DPPTAM: Dense piecewise planar tracking and mapping from a monocular sequence," in *Proc. IEEE/RSJ Int. Conf. Intell. Rob. Syst.*, 2015.

- [۱۸] G. Klein and D. Murray, "Parallel tracking and mapping for small AR workspaces," in *Mixed and Augmented Reality, 2007. ISMAR 2007. 6th IEEE and ACM International Symposium on*, 2007, pp. 225-234.
- [۱۹] C. D. Herrera, K. Kim, J. Kannala, K. Pulli, and J. Heikkilä, "DT-SLAM: deferred triangulation for robust SLAM," in *3D Vision (3DV), 2014 2nd International Conference on*, 2014, pp. 609-616.
- [۲۰] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE transactions on robotics*, vol. 31, pp. 1147-1163, 2015.
- [۲۱] R. Mur-Artal and J. D. Tardos, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, pp. 1255-1262, 2017.
- [۲۲] C. Forster, M. Pizzoli, and D. Scaramuzza, "SVO: Fast semi-direct monocular visual odometry," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, 2014, pp. 15-22.
- [۲۳] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza, "SVO: Semidirect visual odometry for monocular and multicamera systems," *IEEE Transactions on Robotics*, vol. 33, pp. 249-265, 2016.
- [۲۴] L. Heng and B. Choi, "Semi-direct visual odometry for a fisheye-stereo camera," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4077-4084.
- [۲۵] R. Gomez-Ojeda, J. Briaes, and J. Gonzalez-Jimenez, "PL-SVO: Semi-direct Monocular Visual Odometry by combining points and line segments," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2016, pp. 4211-4216.
- [۲۶] R. Gomez-Ojeda and J. Gonzalez-Jimenez, "Robust stereo visual odometry through a probabilistic combination of points and line segments," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2521-2526.
- [۲۷] A. Pumarola, A. Vakhitov, A. Agudo, A. Sanfeliu, and F. Moreno-Noguer, "PL-SLAM: Real-time monocular visual SLAM with points and lines," in *2017 IEEE international conference on robotics and automation (ICRA)*, 2017, pp. 4503-4508.
- [۲۸] R. Gomez-Ojeda, F.-A. Moreno, D. Zuñiga-Noël, D. Scaramuzza, and J. Gonzalez-Jimenez, "Pl-slam: a stereo slam system through the combination of points and line segments," *IEEE Transactions on Robotics*, 2019.
- [۲۹] X. Feng, W. Hao, and B.-f. FANG, "Research on Unmanned Vehicle Positioning Technology Based on Multi-sensor Fusion," *DEStech Transactions on Computer Science and Engineering*, 2018.
- [۳۰] R. G. Von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, pp. 722-732, 2008.
- [۳۱] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An efficient alternative to SIFT or SURF," in *2011 International conference on computer vision*, 2011, pp. 2564-2571.
- [۳۲] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, *et al.*, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, vol. 35, pp. 1157-1163, 2016.