

واقعیت افزوده سریع بدون SLAM

امین رضا غلامی، بهروز نصیحت کن و محسن سریانی

چکیده

الگوریتم‌های واقعیت افزوده برای پیدا کردن محل نقاط شی مجازی در هر قاب، از روش ردیابی و مکان یابی دوربین استفاده می‌کنند. در این روش‌ها ابتدا پارامترهای داخلی و خارجی دوربین تعیین می‌شوند و سپس با استفاده از آن‌ها محل شی مجازی در هر قاب مشخص می‌شود. این روش‌ها باید پارامترهای داخلی دوربین را از قبل داشته باشند یا اینکه پارامترهای داخلی دوربین در کل ویدیو ثابت بمانند. در روش ارائه شده نیازی به تخمین پارامترهای دوربین نیست و محل شی مجازی به صورت مستقیم تعیین می‌گردد. برای این کار الگوریتم پیشنهادی در ابتدا ماتریس بنیادی بین قاب‌های کلیدی و قاب جاری را محاسبه می‌کند و سپس محل نقاط مجازی در هر قاب را با استفاده از تقاطع خطوط اپیپولار بدست می‌آورد. محاسبه ماتریس بنیادی به دلیل وجود نقاط پرت دارای خطا است. استفاده از روش‌هایی مانند رنسک برای حذف نقاط پرت باعث زمان بر بودن الگوریتم می‌شود. در این مقاله که تاکید بر افزایش سرعت الگوریتم پیشنهادی دارد، با استفاده از داده‌های قاب قبلی نقاط پرت را حذف می‌کنیم و یک روش سریع برای پیدا کردن ماتریس بنیادی در هر قاب ارائه می‌دهیم. روش ارائه شده برای محاسبه ماتریس بنیادی، تا ۵ برابر الگوریتم قبلی ما را سریعتر کرد.

کلید واژه‌ها

واقعیت افزوده، ماتریس بنیادی، شی مجازی

۱ - مقدمه

سه بعدی ردیابی شده و پس از آن محاسبه ماتریس دوران و انتقال است. در ادامه و پس از محاسبه این ماتریس‌ها، محل شی مجازی در قاب جدید بدست می‌آید.

هدف اصلی الگوریتم‌های واقعیت افزوده، پیدا کردن محل نقاط یک شی مجازی در هر قاب از ویدیوی تهیه شده توسط دوربین است. با داشتن محل دوبعدی هر نقطه از شی مجازی می‌توان تصویر آن را رسم کرد. در روش ارائه شده قبلی [۱]، ما ماتریس بنیادی را به اجزای سازنده آن، ماتریس‌های دوران، انتقال و پارامترهای داخلی دوربین، تجزیه نمی‌کنیم و به صورت مستقیم از آن برای محاسبه محل نقاط شی مجازی استفاده می‌کنیم. برای این منظور، ابتدا ماتریس‌های بنیادی بین قاب جاری و قاب‌های کلیدی را محاسبه می‌کنیم و سپس با استفاده از این ماتریس‌های بنیادی، محل نقاط شی مجازی را محاسبه می‌کنیم. بر خلاف روش‌های قبلی، روش ارائه شده ما نیازی به محاسبه ماتریس‌های دوران و انتقال برای محاسبه محل شی مجازی ندارد. همچنین روش ارائه شده نیازی به دانستن پارامترهای داخلی دوربین ندارد و زمانی که

ماتریس بنیادی بین نقاط دو قاب که از یک صحنه گرفته شده اند، یک رابطه ایجاد می‌کند. این ماتریس با استفاده از پارامترهای داخلی و خارجی مانند ماتریس دوران و انتقال قابل محاسبه است. تعدادی از روش‌های واقعیت افزوده ابتدا ماتریس بنیادی را محاسبه می‌کنند و سپس با استفاده از آن ماتریس دوران و انتقال را بدست می‌آورند. یک چارچوب رایج برای واقعیت افزوده، تشکیل نقاط

این مقاله در اردیبهشت‌ماه ۱۴۰۴ دریافت شد؛ در مردادماه بازنگری و در شهریورماه پذیرفته گردید.

^۱ دانشجوی دکترای هوش مصنوعی، دانشگاه علم و صنعت ایران
رایانامه: am_gholami@comp.iust.ac.ir

^۲ استادیار دانشکده مهندسی کامپیوتر، دانشگاه خواجه نصیرالدین طوسی
رایانامه: nasihatkon@kntu.ac.ir

^۳ دانشیار دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران
رایانامه: soryani@iust.ac.ir

روش کلین زمانی که دوربین در یک نقطه بایستد و فقط حرکت زاویه‌ای داشته باشد کارایی خود را از دست می‌دهد و نتایج خوبی ندارد. روش گوگلیتز و همکاران هنگامی که دوربین فقط حرکت زاویه‌ای دارد نیز به خوبی عمل می‌کند [۲]. آنها روش مبتنی بر قاب کلیدی و روش پانوراما را ترکیب کردند. الگوریتم، بسته به اینکه دوربین فقط حرکت زاویه‌ای دارد و یا اینکه جابجایی هم دارد یکی از دو مدل را انتخاب می‌کند و توانایی این را دارد که حرکت فقط زاویه‌ای دوربین را هم پشتیبانی کند. سامانه ارائه شده گوگلیتز، تبدیل بین دو قاب را با ماتریس ضروری E یا یک ماتریس هموگرافی H مدل می‌کند. زمانی که دوربین حرکت می‌کند این ماتریس‌ها به صورت کامل حرکات دوربین را مدل می‌کنند. به عبارت دیگر تمام حرکات دوربین با یکی از این دو، مدل سازی می‌شوند. آنها برای انتخاب مدل از معیار اطلاعات مقاوم هندسی (Geometric Robust Information Criterion) GRIC استفاده می‌کنند [۸].

روش ارائه شده توسط مرآتال و همکاران [۹] نیز این دو نخ را دارد. آن‌ها از نقاط ویژگی ORB (Oriented FAST and Rotated BRIEF) استفاده کردند و روش ORBSLAM را ارائه دادند. مرآتال و همکاران نقاط ویژگی ORB را به دلیل سرعت محاسبه و انطباق و همچنین یکسان بودن در دیدهای مختلف انتخاب کردند. این نقاط ویژگی باعث افزایش سرعت الگوریتم آن‌ها شد به صورتی که حتی بدون استفاده از GPU هم روش آنها برخط است [۱۰]. البته روش مورد استفاده آن‌ها مختص این نقاط ویژگی نیست.

زمانی که کیفیت تصویر پایین باشد، ردیابی نقاط ویژگی با خطای زیادی همراه است و باعث کاهش عملکرد سامانه‌های واقعیت افزوده می‌شود. شو و همکاران [۱۱] برای رفع این مشکل از خطوط استفاده کرده‌اند. هدف اصلی الگوریتم آن‌ها، رفع مشکل دقت پایین مکان‌یابی در محیط‌های با بافت ضعیف و نور کم است، جایی که استخراج ویژگی‌های نقطه‌ای ناکافی است. برای این منظور، ویژگی‌های خطی به سامانه افزوده شده‌اند و همچنین آن‌ها با حذف خطوط بی‌فایده، بار محاسباتی را کاهش داده‌اند.

در بسیاری از برنامه‌های کاربردی واقعیت افزوده، نیازی به نگهداری مداوم نقشه کامل محیط نیست؛ به جای آن می‌توان فقط در مجاورت شیء مجازی نقشه را نگه داشت. الگوریتم SLAM نسبت به Visual Odometry (VO) دقت بالاتری دارد، ولی برای تلفن‌های همراه سنگین و پرهزینه است. سوامی و همکارانش [۱۲] یک روش تطبیقی برای تخمین موقعیت در برنامه‌های واقعیت افزوده تلفن همراه ارائه می‌دهند که بین این دو رویکرد جابه‌جا می‌شود تا هزینه محاسباتی و حافظه مورد نیاز کاهش یابد، بدون آنکه کیفیت تخمین کم شود.

نادر محمود و همکاران یک سامانه واقعیت افزوده برای نشان دادن مدل آناتومی بیمار بر روی بدن او ارائه کرده‌اند [۱۳]. کار آن‌ها دو مرحله اصلی دارد؛ ابتدا با استفاده از تصاویر CT یک مدل سه

در حین اجرا پارامترهای داخلی دوربین مانند فاصله کانونی دوربین تغییر کند، نیز به خوبی کار می‌کند. گلوگاه محاسباتی روش قبل، محاسبه ماتریس‌های بنیادی است. زیرا تعداد نقاط پرت ردیابی شده زیاد است و این نقاط با روش رنسک حذف می‌شوند. هر چه تعداد نقاط پرت زیادتر باشد، به محاسبات بیشتری نیاز داریم. در روش جدید ارائه شده در این مقاله با استفاده از اطلاعات ماتریس بنیادی قبلی، یک روش سریع برای محاسبه ماتریس بنیادی قاب جدید ارائه شده است. همچنین تعداد مراحل الگوریتم رنسک با استفاده از تخمین تعداد نقاط پرت بهینه شده است.

۲- کارهای قبلی

یک روش رایج برای افزودن اشیاء به ویدیو، کمک گرفتن از روش مکان‌یابی و نقشه‌برداری هم‌زمان (SLAM) است. در این روش ابتدا ماتریس‌های دوران و انتقال محاسبه می‌شوند و سپس با استفاده از این ماتریس‌ها و همچنین پارامترهای داخلی دوربین، شیء مجازی به تصاویر اضافه می‌شود. روش‌های مکان‌یابی و نقشه‌برداری هم‌زمان دارای دو نخ هستند: نخ ردیابی (Tracking) و نخ نقشه‌برداری (Mapping) [۲]، [۳]. نخ ردیابی نقاط ویژگی موجود در تصاویر را استخراج و ردیابی می‌کند. در بخش نقشه‌برداری ساختار محیط مثل یک ابر نقاط از ویژگی‌های موجود در تصاویر ساخته می‌شود. این کار فقط در قاب‌های کلیدی انجام می‌شود و برای بقیه قاب‌ها لازم به اجرا نیست. در نخ ردیابی، با استفاده از این مدل سه بعدی ساخته شده و نقاط متناظر ردیابی شده در قاب‌های مختلف، ماتریس‌های دوران و انتقال برای هر قاب، محاسبه می‌شوند. این کار با سرعت بالاتری و برای تمام قاب‌ها انجام می‌شود. انتخاب قاب کلیدی باعث افزایش سرعت این الگوریتم‌ها شده است. کلین و ماری جز اولین نفراتی بودند که این روش را ارائه کردند [۴]، [۵] و [۶]. آنها یک سامانه برای ردیابی و نقشه‌برداری نقاط با هدف ایجاد واقعیت مجازی در محیط‌های کوچک ارایه کردند. روش ارائه شده نیاز به مدل قبلی ندارد و در یک محیط با اشیاء ناشناخته کار می‌کند.

یکی از چالش‌های سامانه‌های واقعیت افزوده، حرکت سریع دوربین و جابه‌جایی زیاد آن است. روش ارائه شده توسط لیو و همکاران در حرکات سریع دوربین نیز به خوبی کار می‌کند [۷]. ویژگی اصلی کار آن‌ها ارائه یک روش جدید ردیابی نقاط ویژگی بر اساس هموگرافی چندگانه است که در برابر حرکات سریع دوربین مقاوم است. لیو در ردیابی نقاط از ۳ نوع ماتریس هموگرافی استفاده می‌کند؛ هموگرافی‌های سراسری، محلی و مرتبط با یک صفحه خاص. ماتریس‌های هموگرافی سراسری، رابطه بین قاب‌های کلیدی و قاب جاری را شامل می‌شوند. هموگرافی محلی ماتریس‌های هموگرافی بین قاب‌های اخیر (غیر کلیدی) و قاب جاری را دارد. هموگرافی خاص همانند هموگرافی محلی است اما برای تعدادی صفحه مشخص در فضای سه بعدی استخراج شده است.

در شکل ۱ مراحل الگوریتم نشان داده شده است. در ابتدا مختصات دوبعدی نقاط شی مجازی به همراه عمق نسبی آنها در چند قاب کلیدی ابتدایی مقداردهی اولیه می‌شوند. در مرحله بعد نقاط ویژگی در قاب‌های متوالی ردیابی می‌شوند. برای انتخاب قاب‌های کلیدی مانند کار قبل، فاصله قاب جاری با قاب کلیدی قبلی و تعداد نقاط ویژگی ردیابی شده، در نظر گرفته می‌شود [۱]. سپس با استفاده از نقاط ردیابی شده بین قاب جاری و قاب‌های کلیدی ابتدا ماتریس هموگرافی و سپس ماتریس بنیادی محاسبه می‌شوند. با استفاده از ماتریس‌های بنیادی و محل نقاط شی مجازی در قاب‌های کلیدی، محل نقاط شی مجازی در قاب جاری محاسبه می‌شود. پس از پیدا کردن محل نقاط شی مجازی، برای درج نقاط نیاز به دانستن عمق نسبی است تا بتوان تعیین کرد که کدام صفحات از شی مجازی دیده می‌شوند و کدام صفحات از دید کاربر پنهان هستند. برای این کار در ابتدای کار عمق نقاط شی مجازی به عنوان ورودی به الگوریتم داده می‌شود و سپس الگوریتم در هر قاب با کمک عمق نسبی نقاط در قاب‌های قبلی، عمق نسبی جدید را محاسبه می‌کند.

در این مقاله قسمت "محاسبه ماتریس بنیادی بین قاب جاری و قاب‌های کلیدی" و "محاسبه ماتریس هموگرافی بین قاب جاری و قاب‌های کلیدی" با روشی جدید که سرعت بالاتری دارد، اجرا شده است. کلیات روش ارائه شده، در شکل (۲) آمده است که در بخش بعد توضیح می‌دهیم.

۳-۱- محاسبه سریع ماتریس بنیادی

در روش ارائه شده، نقاط ویژگی شی توماسی (بهینه سازی شده نقاط ویژگی هریس) shi-tomasi در قاب‌های کلیدی پیدا می‌شوند و سپس در قاب‌های بعدی ردیابی می‌شوند. برای ردیابی نقاط از الگوریتم لوکاس کانادی Lucas-Kanade استفاده می‌کنیم. در ادامه، باید ماتریس بنیادی بین قاب جاری و هر قاب کلیدی را با استفاده از این نقاط ردیابی شده، محاسبه کنیم.

در بسیاری از ویدیوها از جمله مجموعه داده مورد استفاده ما، یک سطح صاف مانند یک میز یا یک قالی وجود دارد که نقاط ویژگی زیادی روی آن قرار دارند. نسبت زیاد نقاط روی این سطح صاف نسبت به تعداد کل نقاط، باعث پایین آمدن دقت محاسبه ماتریس بنیادی می‌شود. بنابراین ابتدا ما یک ماتریس هموگرافی محاسبه می‌کنیم و سپس با اطلاعات این ماتریس هموگرافی و نقاط بیرون این صفحه، ماتریس بنیادی محاسبه می‌شود [۱].

در این مقاله، برای افزایش سرعت و دقت، ماتریس هموگرافی و ماتریس بنیادی محاسبه شده در قاب قبلی نیز به عنوان ورودی این بخش در نظر گرفته می‌شوند. مراحل انجام کار برای محاسبه ماتریس بنیادی در ادامه آمده است. ماتریس‌های F_t و F_{t-1} به ترتیب ماتریس بنیادی یک قاب کلیدی با قاب قبلی و ماتریس بنیادی قاب کلیدی با قاب فعلی هستند. همچنین ماتریس‌های H_t و H_{t-1} به ترتیب ماتریس‌های هموگرافی سطح صاف پیدا

بعدی از آناتومی بدن بیمار می‌سازند، سپس موقعیت دوربین را در هر لحظه پیدا می‌کنند و تصویر مدل را بر روی آن منطبق می‌کنند. روش ارائه شده توسط آن‌ها در محیط کوچک کار می‌کند و بر اساس روش ارائه شده توسط کلین است. آن‌ها از توصیفگرهای ORB و نقاط ویژگی FAST استفاده می‌کنند. مقدار دهی اولیه سامانه با استفاده از یک فیلتر کالمن توسعه یافته انجام می‌شود.

یکی از ویژگی‌های اصلی سامانه‌های ردیابی و نقشه برداری، نوع حرکت دوربین و هندسه محیطی است که آن‌ها پشتیبانی می‌کنند. بعضی از آن‌ها در محیط‌های دارای صفحه به خوبی کار می‌کنند [۳] و یا اینکه ممکن است فرض کنند که دوربین تنها حول یک نقطه می‌چرخد [۱۴]، [۱۵]. تعدادی از این سامانه‌ها ادعا می‌کنند که می‌توانند مستقل از هندسه محیط و نوع حرکت دوربین کار کنند [۲]، اما بسیاری از آنها نیاز به پارامترهای داخلی دوربین دارند.

روش ارائه شده نیازی به محاسبه پارامترهای داخلی ندارد. همچنین برای پیدا کردن محل شی مجازی پارامترهای خارجی دوران و محل دوربین را در هر قاب محاسبه نمی‌کند، بلکه با استفاده از اطلاعات ماتریس‌های بنیادی، محل دوبعدی نقاط شی مجازی را در هر قاب به صورت مستقیم محاسبه می‌کند.

۳- روش ارائه شده

در سامانه‌های واقعیت افزوده، باید محل هر نقطه از شی مجازی در هر قاب محاسبه شود. در روش ارائه شده با استفاده از اطلاعات ماتریس‌های بنیادی بین قاب جاری و قاب‌های کلیدی، مکان این نقطه محاسبه می‌شود [۱]. برای این کار از مختصات دوبعدی نقاط شی مجازی در قاب‌های کلیدی استفاده می‌شود. ابتدا باید ماتریس‌های بنیادی بین قاب جدید و قاب‌های کلیدی قبلی محاسبه شوند. با استفاده از نقاط ویژگی ردیابی شده با قاب‌های کلیدی، این ماتریس‌ها بدست می‌آیند. به صورت تجربی از ۶ قاب کلیدی استفاده می‌کنیم. فرض کنید که نقاط x_1, x_2, \dots, x_6 محل یک نقطه از شی مجازی در ۶ قاب کلیدی قبل باشند و همچنین F_1, F_2, \dots, F_6 ماتریس‌های بنیادی متناظر با این قاب‌های کلیدی و قاب جاری باشند. خط اپیپولار l_i متناظر با قاب کلیدی i ام از رابطه:

$$l_i = F_i^T x_i \quad (1)$$

محاسبه می‌شود. محل هر نقطه شی مجازی در قاب جاری برابر با محل تقاطع خطوط اپیپولار متناظر با قاب‌های کلیدی است. همانطور که در روش قبلی گفته شده، چون این خطوط دقیقاً در یک نقطه برخورد نمی‌کنند، نقطه‌ای با حداقل جمع فاصله‌ها از همه خطوط به عنوان جواب در نظر گرفته می‌شود. اگر $dist(x, l)$ برابر با فاصله نقطه x از خط l باشد، نقطه بهینه برابر است با:

$$x_{opt} = \operatorname{argmin}_x \sum_{i=1}^n dist(x, l_i) \quad (2)$$

در مرحله سوم با استفاده از ماتریس بنیادی قبلی F_{t-1} مقدار $e_{i,t}^F$ را برای نقطه i ام در قاب t با استفاده از رابطه زیر محاسبه می‌کنیم:

$$e_{i,t}^F = \max(\text{dist}(I_{i,t}, x_{i,t}), \text{dist}(I'_{i,t}, x_{i,k})) \quad (6)$$

که در آن نقاط $x_{i,t}$ و $x_{i,k}$ به ترتیب محل یک نقطه ردیابی شده در قاب کلیدی و قاب جاری هستند و خطوط $I_{i,t}$ و $I'_{i,t}$ ، خط‌های اپیپولار متناظر با آنها هستند که با استفاده از ماتریس بنیادی قبلی F_{t-1} محاسبه شده‌اند. تابع dist فاصله نقطه از خط را نشان می‌دهد. اگر این مقدار خطا از یک حد آستانه بیشتر بود، این نقطه را از محاسبه ماتریس بنیادی جدید حذف می‌کنیم.

در مرحله چهارم با استفاده از رابطه $H^T F + F^T H = 0$ شش معادله برای محاسبه ماتریس F بدست می‌آید که ۵ تای آنها مستقل هستند. همین طور هر دو نقطه ویژگی متناظر که برای محاسبه H استفاده نشده‌اند، نیز یک معادله برای محاسبه ماتریس بنیادی به ما می‌دهند. اگر تعداد این زوج نقاط n عدد باشد، در مجموع $6 + n$ معادله برای محاسبه F داریم و با استفاده از این معادلات ماتریس بنیادی جدید را محاسبه می‌کنیم. برای حذف نقاط پرت باقی مانده دوباره از رنسک استفاده می‌کنیم.

۲-۳- تعداد مراحل اجرای الگوریتم رنسک

به دلیل اینکه در مراحل ۱ و ۳ نقاط پرت را حذف کرده‌ایم، هنگام محاسبه ماتریس‌های هموگرافی و بنیادی با استفاده از رنسک، نیاز به تعداد مراحل کمی داریم و الگوریتم سریع به نتیجه می‌رسد.

تعداد مراحل محاسبه ماتریس بنیادی با استفاده از رابطه زیر بدست می‌آید [۱۶]:

$$N = \log(1 - p) / \log(1 - (1 - \epsilon)^s) \quad (7)$$

که در آن N تعداد مراحل مورد نیاز برای یافتن جواب با احتمال درستی $p = 0.99$ است. همچنین ϵ برابر است با احتمال اینکه یک نقطه جزء نقاط پرت باشد. به دلیل عدم تغییر زیاد در دو قاب متوالی، برای تخمین ϵ از داده قاب قبلی استفاده می‌کنیم. به عبارت دیگر مقدار آن را برابر با درصد تعداد نقاط پرت در قاب قبلی در نظر می‌گیریم. همچنین s برابر با حداقل تعداد نقاط متناظر لازم برای محاسبه ماتریس بنیادی یا ماتریس هموگرافی است. این مقدار برای محاسبه ماتریس هموگرافی برابر با ۴ و برای محاسبه

شده قاب کلیدی با قاب قبلی و قاب فعلی هستند. $x_{i,t}$ برابر با نقطه ویژگی i ام ردیابی شده در قاب t ام است.

به دلیل اینکه تغییرات در دو قاب متوالی زیاد نیست، مقدار ماتریس‌های H_t و H_{t-1} تقریباً یکسان است. بنابراین در ابتدا مقدار H_t را با H_{t-1} تقریب می‌زنیم و با این تقریب تعدادی از نقاط پرت و خارج از صفحه را حذف می‌کنیم و سپس با نقاط باقیمانده ماتریس H_t را به صورت دقیق‌تر محاسبه می‌کنیم. این کار در مراحل ۱ و ۲ از شکل ۲ انجام می‌شود. برای ماتریس‌های بنیادی نیز به همین صورت عمل می‌کنیم و ابتدا با تقریب مقدار F_t با F_{t-1} نقاط پرت را حذف می‌کنیم و سپس با نقاط باقی‌مانده مقدار F_t را بدست می‌آوریم. محاسبه این ماتریس در مراحل ۳ و ۴ از شکل ۲ انجام می‌شود. مراحل ۱ تا ۴ در ادامه توضیح داده شده‌اند.

در مرحله اول با استفاده از ماتریس هموگرافی قبلی، نقاطی که قطعاً در این قاب روی یک صفحه نیستند، حذف می‌شوند. فرض کنید نقاط $x_{i,t}$ و $x_{i,k}$ به ترتیب محل یک نقطه ردیابی شده در قاب کلیدی و قاب جاری باشند. ابتدا $y_{i,k}$ تصویر منتقل شده نقطه $x_{i,k}$ از قاب کلیدی به قاب جاری با استفاده از ماتریس هموگرافی H_{t-1} را با روابط زیر محاسبه می‌کنیم:

$$y'_{i,k} = H_{t-1} * x_{i,k} \quad (3)$$

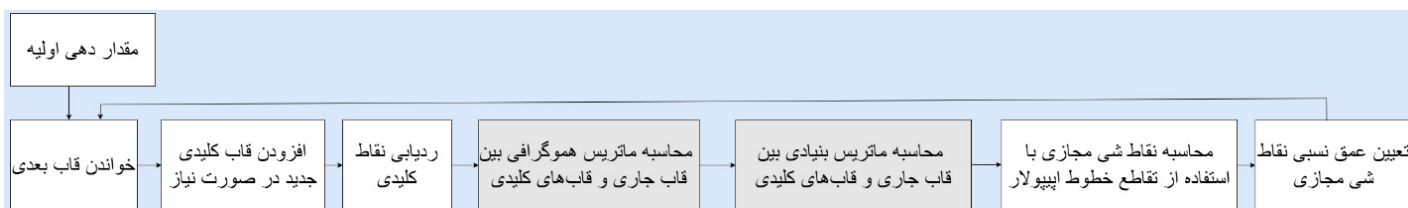
$$y_{i,k} = \frac{y'_{i,k}}{y'_{i,k,3}} \quad (4)$$

در رابطه اخیر $y'_{i,k,3}$ برابر با عنصر سوم بردار $y'_{i,k}$ است. مقدار خطای $e_{i,t}$ برای نقطه ردیابی شده i ام در قاب t را با استفاده از رابطه زیر محاسبه می‌کنیم:

$$e_{i,t} = \|y_{i,k} - x_{i,t}\| \quad (5)$$

اگر $e_{i,t}$ محاسبه شده از یک حد آستانه بیشتر بود، این نقطه را از محاسبه ماتریس هموگرافی حذف می‌کنیم. چون H_t دقیقاً برابر با H_{t-1} نیست، تعدادی از نقاط پرت و خارج از صفحه باقی می‌مانند. اما تعداد آن‌ها کم می‌شود. همچنین نقاط درست باقی می‌مانند و حذف نمی‌شوند.

در مرحله دوم با استفاده از نقاط باقی مانده، ماتریس هموگرافی جدید H_t بین قاب جدید و قاب کلیدی را محاسبه می‌کنیم. برای حذف نقاط پرت باقی مانده از الگوریتم رنسک کمک می‌گیریم.



شکل ۱: مراحل الگوریتم ارائه شده برای افزودن شی مجازی به ویدیو در کار قبلی [۱]. در روش ارائه شده در این مقاله، دو قسمت "محاسبه ماتریس بنیادی بین قاب جاری و قاب‌های کلیدی" و "محاسبه ماتریس هموگرافی بین قاب جاری و قاب کلیدی" بهبود یافته‌اند و با سرعت بالاتری محاسبه می‌شود.



شکل ۲- روش ارائه شده برای تسریع محاسبه ماتریس بنیادی

و اشیاء دیگر مشابه قرار دارند. نتایج این آزمایش را در شکل ۷ می‌توانید ببینید.

۴-۱- مقایسه با روش ORB-SLAM

روش ارائه شده برای افزایش سرعت، کاهش دقت آن نداشت و دقت آن مانند روش قبلی بود. برای اندازه‌گیری دقت، ابتدا در هر قاب محل نقاط شیء مجازی محاسبه می‌شود و سپس جذر میانگین مربعات خطای این نقاط به عنوان معیار خطا در نظر گرفته می‌شود. محاسبه خطا برای روش ارائه شده و الگوریتم ORB-SLAM انجام شد و در شکل ۳ نمایش داده شده است. میانگین خطای روش ارائه شده ۱۱/۶ پیکسل و روش ORB-SLAM ۹/۱ پیکسل است.

همانطور که در کار قبلی [۱] بحث شد، روش ارائه شده با وجود ۱۱ درجه آزادی و نداشتن هیچ فرضی برای پارامترهای داخلی دوربین، دقتی نزدیک به روش ORB-SLAM دارد. روش ORB-SLAM شش درجه آزادی دارد و پارامترهای داخلی دوربین به عنوان ورودی مورد نیاز است.

۴-۲- مقایسه سرعت اجرا

در روش ارائه شده قبلی، در مرحله محاسبه ماتریس‌های بنیادی برای حذف نقاط پرت از روش رنسک استفاده می‌شد. این بخش زمان زیادی نیاز داشت. به صورت میانگین زمان اجرای کل برای افزودن شیء مجازی به هر قاب از مجموعه داده Freiburg2-desk برابر با ۱۰۹۷ میلی ثانیه بود. در روش جدید برای قاب‌های کلیدی همانند روش قبل عمل می‌کنیم بنابراین زمان اجرا برای این قاب‌ها همان ۱۰۹۷ میلی ثانیه است اما برای قاب‌های غیرکلیدی با روش پیشنهادی، زمان اجرا به ۱۶۷ میلی ثانیه بهبود یافت. و در مجموع در روش جدید، میانگین زمان اجرای کل الگوریتم برای هر قاب (کلیدی و غیرکلیدی)، از ۱۰۹۷ به ۲۱۴ میلی ثانیه رسید. بنابراین سرعت در قاب‌های غیرکلیدی ۶/۵ برابر و در کل قاب‌ها ۵ برابر افزایش یافته است. جزئیات در جدول ۲ نشان داده شده است.

در آزمایش دیگری برای اضافه کردن مکعب به تصاویر مجموعه داده IUST-DESK زمان اجرا برای قاب‌های غیر کلیدی از ۱۰۸ میلی ثانیه به ۳۲ میلی ثانیه و در مجموع، میانگین زمان اجرای کل الگوریتم از ۱۰۸ میلی ثانیه به ۳۶ میلی ثانیه برای هر قاب کاهش یافت. افزایش سرعت برای قاب‌های غیرکلیدی ۳/۴ برابر و برای کل قاب‌ها ۳ برابر شده است. جزئیات زمان اجرا در جدول ۳ نمایش داده شده است.

ماتریس بنیادی در الگوریتم ۸ نقطه برابر با ۸ است. با روش ارائه شده برای محاسبه ماتریس بنیادی، $s = 3$ است، زیرا از اطلاعات ماتریس هموگرافی نیز استفاده می‌شود.

استفاده از ماتریس هموگرافی علاوه بر بالا بردن دقت محاسبه ماتریس‌های بنیادی، باعث کاهش تعداد مراحل مورد نیاز برای الگوریتم رنسک می‌شود. تعداد مراحل الگوریتم رنسک با توجه به تعداد نقاط پرت و اندازه نمونه s در جدول ۱ آمده است. با محاسبه مستقیم ماتریس بنیادی با الگوریتم ۷ نقطه یا ۸ نقطه نیاز به تعداد مراحل بیشتری نسبت به روش پیشنهادی است. مثلاً با داشتن ۳۰ درصد نقاط پرت، در الگوریتم ۸ نقطه با پارامتر $s = 8$ ، نیاز به ۷۸ مرحله اجرا است اما در روش ارائه شده، محاسبه ماتریس هموگرافی ($s = 4$) نیاز به ۱۷ مرحله و سپس محاسبه ماتریس بنیادی با اطلاعات ماتریس هموگرافی و ۳ نقطه جدید، ($s = 3$) نیاز به ۱۱ مرحله دارد. یعنی در مجموع روش ارائه شده با ۲۸ مرحله در الگوریتم رنسک نسبت به ۷۸ مرحله الگوریتم ۸ نقطه، سرعت بالاتری دارد.

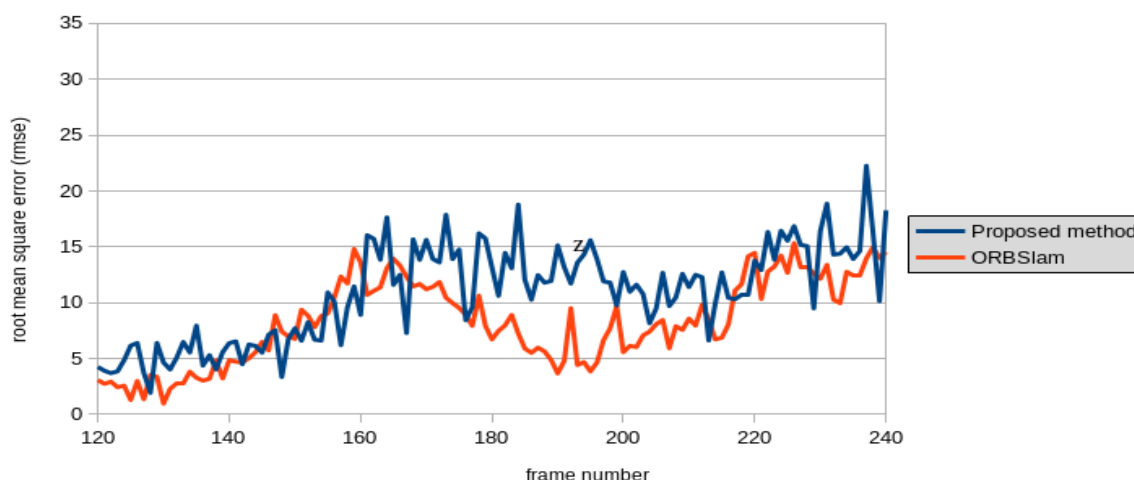
جدول ۱ تعداد مراحل مورد نیاز برای اجرای الگوریتم رنسک

حداقل تعداد نقاط	احتمال پرت بودن یک نقطه متناظر			
s	50%	40%	30%	25%
3	35	19	11	9
4	72	34	17	13
7	588	163	54	33
8	1177	272	78	44

۴- نتایج

برای بررسی عملکرد روش ارائه شده از مجموعه داده Freiburg2-desk استفاده شد. این مجموعه داده با استفاده از یک دوربین گرفته شده است. محیط این مجموعه داده، دو میز با اشیایی مثل صفحه نمایش، صفحه کلید، و تلفن بر روی آن است. روش ارائه شده بر روی این مجموعه داده اجرا شد و یک مکعب رنگی به تصاویر اضافه شد. نتیجه نهایی را می‌توانید در شکل ۴ ببینید. همچنین برای بررسی بیشتر، اشیایی پیچیده تر با تعداد نقاط بیشتر به همین مجموعه داده اضافه شد. در شکل ۵ یک بیست وجهی منظم و در شکل ۶ یک قوری به تصاویر مجموعه داده اضافه شده است.

مجموعه داده دوم که روش پیشنهادی بر روی آن آزمایش شد، IUST-DESK است. این مجموعه داده توسط تیم ما با استفاده از دوربین موبایل تهیه شده است. محیط تهیه آن یک میز بزرگ است که روی آن لپتاپ، صفحه نمایش، صفحه کلید، کتاب، تلفن



شکل ۳- خطای روش ارائه شده در هر قاب برای مجموعه داده Freiburg2-desk با رنگ آبی و برای روش ORB-SLAM با رنگ قرمز نمایش داده شده است [۱].

۳-۴- نتیجه‌گیری و کارهای آینده

در این مقاله، روشی برای افزایش سرعت محاسبه ماتریس بنیادی در کاربرد افزودن اشیاء مجازی به ویدیو ارائه شد. در این روش با استفاده از ماتریس بنیادی محاسبه شده در قاب‌های قبلی، نقاط پرت حذف شدند و این کار موجب افزایش سرعت محاسبه ماتریس بنیادی شد. این روش جدید در الگوریتم ارائه شده قبلی برای افزودن اشیاء مجازی به تصاویر به کار برده شد. با حذف اولیه نقاط پرت، تعداد قدم‌های لازم برای اجرای الگوریتم رنسک در محاسبه ماتریس بنیادی کاهش یافت. روش ارائه شده باعث افزایش تا ۵ برابری سرعت الگوریتم قبلی شد در حالی که دقت افزودن اشیاء ثابت ماند.

در ادامه می‌توان به جای استفاده از ماتریس بنیادی قاب قبلی، یک روش برای پیش‌بینی ماتریس بنیادی قاب فعلی از داده‌های قبلی استفاده کرد. با این کار تعداد نقاط پرت بیشتری را می‌توان حذف کرد و الگوریتم بهتر عمل می‌کند.

در روش ارائه شده، محاسبه هرکدام از ماتریس‌های بنیادی می‌تواند به صورت موازی انجام شود و همچنین در هنگام محاسبه یک ماتریس بنیادی، می‌توان از پردازشگرهای موازی استفاده کرد. ترکیب روش پیشنهادی ما با استفاده از پردازشگرهای موازی سرعت اجرای الگوریتم را بهبود بیشتری می‌دهد.

جدول ۲: مقایسه متوسط زمان اجرای کل الگوریتم برای هر قاب در روش پیشنهادی با روش پایه [۱] در مجموعه داده Freiburg2-desk

روش	زمان قاب‌های کلیدی (ms)	زمان قاب‌های غیر کلیدی (ms)	میانگین زمان برای تمام قاب‌ها (ms)
روش پایه [۱]	1097	1097	1097
روش پیشنهادی	1097	167	214

جدول ۳: مقایسه متوسط زمان اجرای کل الگوریتم برای هر قاب در روش پیشنهادی با روش پایه [۱] در مجموعه داده IUSTR-DESK

روش	زمان قاب‌های کلیدی (ms)	زمان قاب‌های غیر کلیدی (ms)	میانگین زمان برای تمام قاب‌ها (ms)
روش پایه [۱]	108	108	108
روش پیشنهادی	108	32	36

روش ارائه شده به صورت بنیادین با روش‌های مبتنی بر SLAM تفاوت دارد. این روش مزایای متعددی از جمله عدم نیاز به بازسازی سه‌بعدی نقاط صحنه، بی‌نیازی از پارامترهای داخلی دوربین و حفظ عملکرد مناسب حتی با تغییر پارامترهای دوربین، دارد. علیرغم این مزایا، دقت روش ارائه شده با SLAM قابل مقایسه است. در این مقاله سرعت اجرای روش پایه بهبود داده شد اما هنوز در مقایسه با روش‌های SLAM سرعت پایین تری دارد. در جدول ۴ سرعت اجرای روش ORB-SLAM3 و روش پیشنهادی و همچنین روش مقاله پایه در مجموعه داده Freiburg2-desk مقایسه شده است.

جدول ۴: مقایسه سرعت اجرای روش پیشنهادی با روش ORB-SLAM3 و همچنین میزان بهبود آن نسبت به روش پایه [۱]

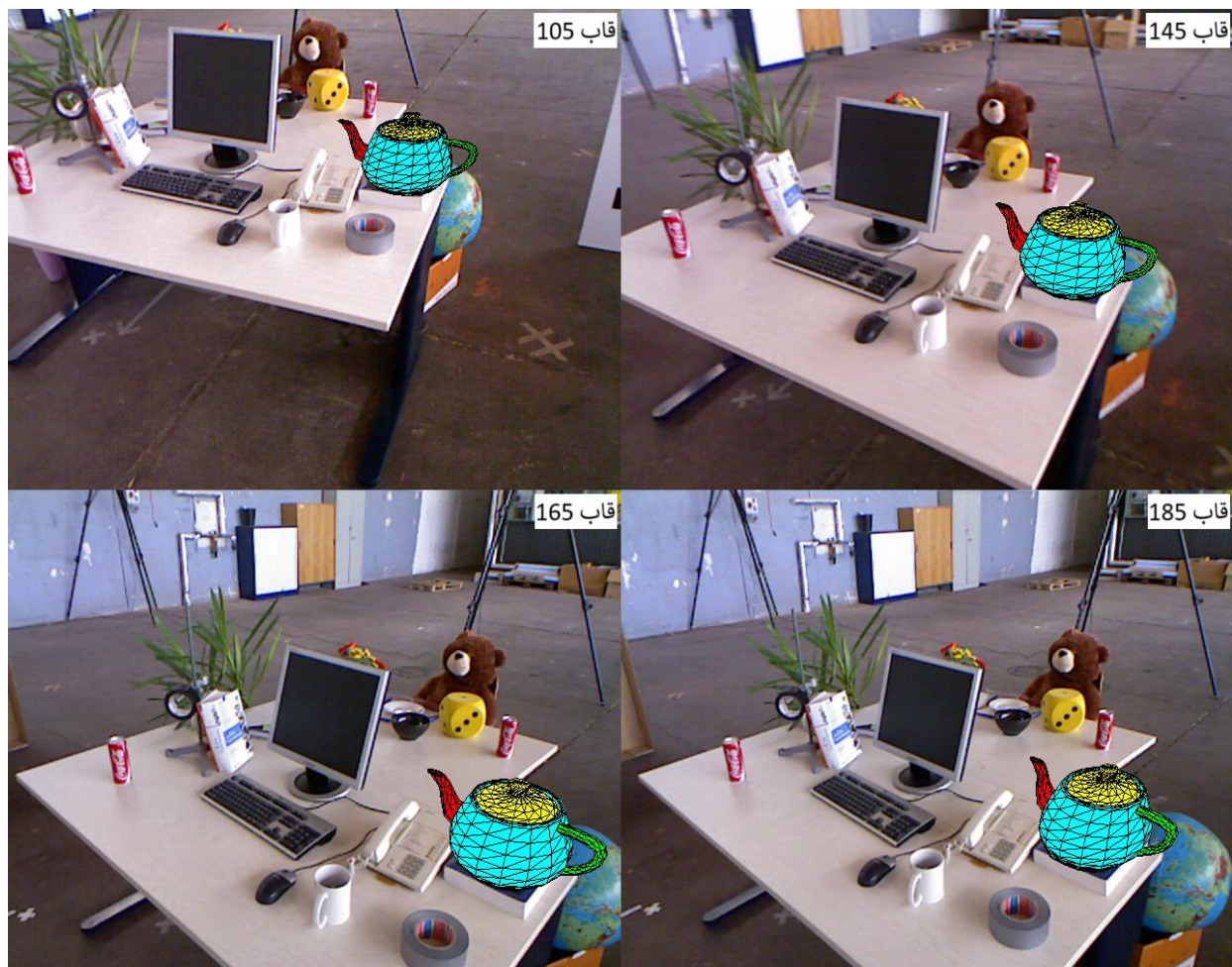
روش	تعداد قاب‌های اجرا شده	زمان کل اجرا (ms)	میانگین زمان اجرا برای هر قاب (ms)
ORB_SLAM3	120	6840	57
روش پایه [۱]	120	131640	1097
روش پیشنهادی	120	25680	214



شکل ۴: نتیجه اجرای روش ارائه شده، یک مکعب به تصاویر مجموعه داده Freiburg2-desk اضافه شده است.



شکل ۵: با استفاده از روش ارائه شده یک بیست وجهی به تصاویر مجموعه داده Freiburg2-desk اضافه شده است.



شکل ۷: با استفاده از روش ارائه شده یک قوری به تصاویر مجموعه داده Freiburg2-desk اضافه شده است.



شکل ۶: نتیجه افزوده شدن یک مکعب به مجموعه داده IUST-DESK با روش ارائه شده.

مراجع

- [13] N. Mahmoud, Ó. G. Grasa, S. A. Nicolau, C. Doignon, L. Soler, J. Marescaux, et al., *On-patient see-through augmented reality based on visual SLAM*, *International Journal of Computer Assisted Radiology and Surgery*, 2017.
- [14] DiVerdi S, Wither J, Hollerer T. *All around the map: Online spherical panorama construction*. *Computers & Graphics*, 2009.
- [15] Wagner D, Mulloni A, Langlotz T, Schmalstieg D. *Real-time panoramic mapping and tracking on mobile phones*. *IEEE virtual reality conference (VR)*, 2010.
- [16] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision: Cambridge university press*, 2003.
- [1] Gholami, A., Nasihatkon, B., & Soryani, M. *Augmented reality without SLAM*. *Multimedia Tools and Applications*, 2024.
- [2] Gauglitz S, Sweeney C, Ventura J, Turk M, Hollerer T. *Live tracking and mapping from both general and rotation-only camera motion*. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2012.
- [3] Pirchheim C, Reitmayr G. *Homography-based planar mapping and tracking for mobile phones*. *10th IEEE International Symposium on Mixed and Augmented Reality*, 2011.
- [4] Klein G, Murray D. *Parallel tracking and mapping for small AR workspaces*. *6th IEEE and ACM international symposium on mixed and augmented Reality*, 2007.
- [5] Klein G, Murray D. *Improving the agility of keyframe-based SLAM*. *Computer Vision—ECCV: 10th European Conference on Computer Vision*, 2008.
- [6] Klein G, Murray D. *Parallel tracking and mapping on a camera phone*. *8th IEEE International Symposium on Mixed and Augmented Reality*, 2009.
- [7] Liu H, Zhang G, Bao H. *Robust keyframe-based monocular SLAM for augmented reality*. *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2016.
- [8] P. H. S. Torr, *Bayesian model estimation and selection for epipolar geometry and generic manifold fitting*, *International Journal of Computer Vision*, 2002.
- [9] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, *ORB-SLAM: a versatile and accurate monocular SLAM system*, *IEEE Transactions on Robotics*, 2015.
- [10] Rabbi I, Ullah S. *A survey on augmented reality challenges and tracking*. *Acta graphica: znanstveni casopis za tiskarstvo i graficke komunikacije*, 2013.
- [11] Xu, J., Fang, Y., Gao, W., *Robot visual inertial SLAM algorithm based on point-line features*, *7th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, 2024.
- [12] Swamy S.M.K, Han Q., *APPEAR: Adaptive Pose Estimation for Mobile Augmented Reality*, *IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, 2024.



بهروز نصیحت‌کن تحصیلات کارشناسی و کارشناسی ارشد خود را در رشته مهندسی کامپیوتر به ترتیب در سال‌های ۱۳۸۴ و ۱۳۸۷ در دانشگاه شیراز به پایان رساند. وی در سال ۱۳۹۲ موفق به دریافت مدرک دکترای خود از دانشگاه ملی استرالیا شد. رساله دکتری او بر تعمیم نظریه بازسازی افکنشی در بینایی کامپیوتر با هدف طراحی و تحلیل الگوریتم‌های بازسازی سه‌بعدی متمرکز بود. او هم‌اکنون استادیار دانشگاه صنعتی خواجه نصیرالدین طوسی در تهران است. زمینه‌های پژوهشی مورد علاقه او شامل بینایی ماشین، هندسه چنددیدگاهی، سامانه‌های پیشرفته کمک‌راننده و رانندگی خودران می‌شود.



محسن سربانی در سال ۱۳۵۹ مدرک کارشناسی خود را در رشته مهندسی برق - الکترونیک از دانشگاه علم و صنعت ایران اخذ کرد. او سپس به ترتیب در سال‌های ۱۳۶۶ و ۱۳۶۹ مدارک کارشناسی ارشد و دکتری خود را از دانشکده مهندسی برق و کامپیوتر دانشگاه هریوت-وات

در شهر ادینبورو در اسکاتلند با تخصص‌های "فنون دیجیتال" و "پردازش تصویر" دریافت نمود. در سال ۱۳۶۹ و پس از فراغت از تحصیل، در دانشکده فنی دانشگاه مازندران به عنوان عضو هیات علمی مشغول به کار گردید. در سال ۱۳۸۱ به گروه هوش مصنوعی و رباتیک دانشکده مهندسی کامپیوتر دانشگاه علم و صنعت ایران پیوست و در حال حاضر در این گروه با مرتبه دانشیاری مشغول به کار است. زمینه‌های پژوهشی مورد علاقه وی شامل پردازش و تحلیل تصاویر، بینایی ماشین و کاربردهای هوش مصنوعی در این مباحث است.