

## ارایه راهکاری نوین در توالی و همترازی تصاویر به منظور موزاییک بندی تصاویر

محمد قاسمی<sup>۱</sup>، نوشین قادری<sup>۲</sup>، مریم تقی زاده دهکردی<sup>۳</sup>

<sup>۱</sup>کارشناسی ارشد برق الکترونیک، دانشگاه شهرکرد

<sup>۲</sup>استادیار دانشکده برق و الکترونیک، دانشگاه شهرکرد، ghaderi.nooshin@eng.sku.ac.ir

<sup>۳</sup>استادیار دانشکده برق و الکترونیک، دانشگاه شهرکرد

### چکیده

در این مقاله الگوریتمی بهبود یافته برای موزاییک سازی خودکار تصاویر هوایی و ارتقای کیفیت تصاویر حاصل از آن به همراه توالی بین تصاویر ارایه شده است. با استفاده از این الگوریتم زمان پردازش در مورد تصاویر با حجم بالا به میزان قابل توجهی کاهش می یابد. در این الگوریتم ابتدا تصاویر، تقسیم بندی شده و به هر پنجره، درجه احتمالی متناظر آن اختصاص داده می شود. در نهایت، با رتبه بندی احتمالات اختصاص داده شده به هر پنجره، از روند تکراری برای انتخاب نقاط جلوگیری می شود. نتایج ارایه شده در این مقاله نشان می دهد که با استفاده از الگوریتم فوق، موزاییک بندی تصاویر با حجم بالا با دقت و سرعت مناسبتری نسبت به الگوریتم های موجود انجام می پذیرد.

### کلیدواژه

موزاییک بندی تصاویر، نقاط کلیدی، ویژگی هندسی، نقاط پرت، نقاط متناظر

### مقدمه

وسیع و تفکیک پذیری بالا است. در این روشها، با محاسبه پارامترهای تبدیل با استفاده از نقاط کنترلی در عکس مربوطه یکی از تصاویر به عنوان تصویر پایه انتخاب می شود و هندسه تصویر دوم به تصویر پایه تبدیل خواهد شد. در نهایت می بایست مشکلاتی از قبیل حالت روح مانند و اختلاف نورها را نیز برطرف کرد تا کیفیت تصویر موزاییکی با درصد قابل ملاحظه ای، ارتقا یابد [۳].

تکنولوژی دوخت تصویر مبتنی بر الگوریتم SIFT به تدریج تبدیل به بحث داغی در زمینه پردازش تصویر شده است. این تکنولوژی تمرکز بر بهبود دقت تطابق تصویر و کاهش زمان تطبیق تصویر برای حفظ صفحه نمایش در زمان واقعی و صحت ادراک بصری انسان دارد [۴].

بایر و همکاران الگوریتم SURF را بر مبنای الگوریتم SIFT پیشنهاد کردند [۵]. این الگوریتم تا حد زیادی سرعت استخراج ویژگی را افزایش می دهد. اما در دوخت تصاویر چند گانه می تواند انحرافات بزرگ تولید کند.

در [۶] با استفاده از روشهای تبدیل موجک و فشرده سازی به سرعت بالاتری در SIFT دست یافته است. این در حالی است که این روش فقط بر روی تصاویر حجم پایین اعمال شده و نتایج گزارش شده است.

موزاییک بندی تصویر، ترکیب تصاویر هم پوشانی شده ای است که تصویر مخلوط شده کمترین انسداد را در ناحیه مخلوط شده داشته باشد در حالی که ظاهر کلی حفظ شود [۱]. ترکیب تصاویر باید به گونه ای انجام شود که محدوده هایی که در دور نواحی مشترک پدید می آیند، از بین رفته و تصویر موزاییک بندی شده حاصل، کمترین انحراف ممکن را از تصاویر تشکیل دهنده داشته باشد. موزاییک بندی تصاویر یکی از موضوعات تحقیقاتی مهم در زمینه های پردازش تصاویر، بنیایی ماشین و گرافیک کامپیوتری می باشد که هم اکنون کاربردهای فراوانی در فشرده سازی تصاویر و نمایه گذاری آنها دارد [۲،۳]. بازسازی ساختار صحنه سه بعدی و نقشه برداری سه بعدی محیط با استفاده از تصاویر مختلف که به کاربر اجازه می دهد تا محیط احاطه کننده را با تصاویر واقعی ببیند نمونه هایی از کار در این حوزه های تحقیقاتی هستند [۳].

اخیرا با توجه به ارسال تصاویر ماهواره ها به زمین، نیاز به موزاییکی کردن تصاویر در نقشه برداری های هوایی، بیش از پیش افزایش پیدا کرده است.

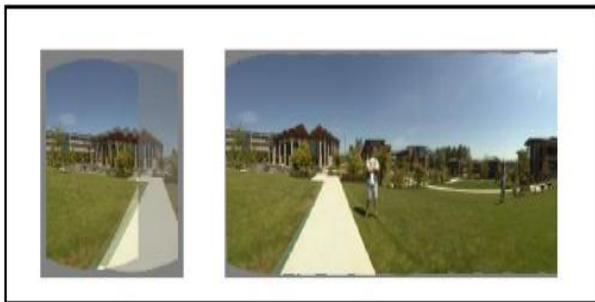
روش های مختلفی برای موزاییک سازی تصاویر ارایه شده است [۳-۶] که هدف نهایی در آنها، ایجاد تصویری با میدان دید

تصویر می‌باشد. روش‌هایی که از این روش مقایسه پیکسل به پیکسل استفاده می‌کنند را روش‌های مستقیم نام نهاده‌اند.

برای استفاده از این مدل روش‌ها، باید ابتدا یک تابع خطای مناسب تعریف شود تا بتوان به وسیله آن، پیکسل‌های تصاویر را با یکدیگر مقایسه کرد. مرحله بعدی انتخاب روشی مناسب برای جستجو است که به وسیله آن فضای حالات ممکن به طور بهینه بررسی گردد. روش ساده برای جستجو، آزمایش کردن تمام حالات ممکن است که در عمل سرعت انجام کار را بسیار پایین می‌آورد. استفاده از روش سلسله مراتبی درشت به ریز که براساس هرم تصاویر به دست آورده می‌شود، یکی از راه‌حل‌هایی است که برای سرعت دادن به انجام همترازی استفاده می‌گردد [۷]. در این حالت باید حرکت انتقالی محاسبه گردد. برای پوشش حرکت انتقالی، مقدار  $\delta t = (\delta t_x, \delta t_y)$  را باید تخمین زد که این کار با حداقل کردن تابع خطای زیر که اختلاف مقادیر شدت بین دو تصویر را نشان می‌دهد انجام می‌شود.

$$E(\delta t) = \sum_i [I_1(X'_i + \delta t) - I_0(X_i)]^2 \quad (1)$$

که  $X_i = (x_i, y_i)$  و  $X'_i = (x_i + t_x, y_i + t_y)$  نقاط متناظر در دو تصویر می‌باشند و  $t = (t_x, t_y)$  انتقال حرکتی سراسری صحنه است که برای تمام پیکسل‌های تصویر به طور مشابه انجام می‌شود. شکل ۱ قسمتی از موزاییک پانورامای استوانه‌ای که به وسیله این روش همترازی انتقال ساخته شده است را نشان می‌دهد.



(ب)

(الف)

شکل ۱: (الف) نگاشت دو تصویر به مختصات استوانه‌ای. (ب) نتیجه حاصل از اتصال تصاویر نگاشت شده به مختصات استوانه‌ای [۷].

### ترکیب تصاویر

پس از منطبق کردن تصاویر نسبت به هم می‌بایست سطحی که می‌خواهیم تصویر موزاییک شده نهایی را بر روی آن تصویر کنیم، مشخص گردد. این سطوح می‌توانند شامل صحنه، استوانه و کره باشند. در صورتی که تعداد اندک باشند معمولاً یکی از تصاویر را

در این مقاله، با ارایه یک الگوریتم جدید، زمان و دقت مناسبی برای پردازش تصاویر حجم بالا به دست آمده است. در این الگوریتم، برای کاهش زمان مورد نیاز، تلاش می‌شود که نقاط پرت حذف شده و در صد زیادی از نقاط منطبق باقی بماند. به این منظور ابتدا تصاویر، تقسیم بندی شده و به هر پنجره، درجه احتمالی متناظر آن اختصاص داده می‌شود. در نهایت، با رتبه بندی احتمالات اختصاص داده شده به هر پنجره، از روند تکراری برای انتخاب نقاط جلوگیری می‌شود. نتایج ارایه شده در این مقاله نشان می‌دهد که با استفاده از الگوریتم فوق، موزاییک بندی تصاویر با حجم بالا با دقت و سرعت مناسبتری نسبت به الگوریتم‌های موجود انجام می‌پذیرد.

نحوه تنظیم این مقاله به صورت زیر می‌باشد. به دنبال این بخش، در بخش دوم مراحل مختلف موزاییک سازی تصویر بیان می‌شود. در بخش سوم استخراج ویژگی‌ها توسط آشکارسازها و در بخش چهارم روش اجماع اتفاقی نمونه برای حذف نقاط پرت ارائه شده است. در بخش پنجم الگوریتم ارایه شده برای موزاییک سازی توضیح داده شده است. نتایج آزمایشی در بخش ششم و نتیجه‌گیری نیز در بخش هفتم آورده شده است.

### مراحل مختلف موزاییک سازی تصاویر

الگوریتم‌های موزاییک سازی تصاویر به صورت کلی دارای چهار مرحله هستند که عبارتند از: ۱- محاسبه مدل هندسی ۲- منطبق کردن تصاویر ۳- ترکیب تصاویر ۴- تصحیح بهبود تباین (کنتراست). در ذیل به طور خلاصه به هر یک از این مراحل اشاره می‌شود.

#### محاسبه مدل هندسی

برای اینکه بتوانیم تصاویر را همتراز کنیم ابتدا می‌بایست یک مدل ریاضی برای نگاشت پیکسل‌های یک عکس به پیکسل‌های عکس دیگر تعریف کنیم. مدل‌های حرکتی پارامتری مختلفی برای این مسئله ارایه شده‌اند. این مدل‌ها از تبدیل‌های ساده دو بعدی تا مدل‌های پرسپکتیو مسطح، دوران سه بعدی دوربین، اعوجاج لنز و نگاشت به سطوح غیر مسطح مانند استوانه و کره تشکیل شده‌اند.

#### منطبق کردن تصاویر

وقتی مدل مناسب برای تشریح همترازی بین جفت تصویر انتخاب شد، باید روشی را برای تخمین مدل به کار ببریم. یکی از این روش‌ها شیفت دادن یا چرخاندن یک تصویر نسبت به تصویر دیگر و اندازه‌گیری میزان تاپید از طرف پیکسل‌های دو

هدف مرحله تصحیح بهبود تباین (کنتراست)، جبران سازی تغییرات شدت روشنایی موجود در تصاویر مختلف است که ناشی از تصویربرداری در زمان‌های مختلف می‌باشد. هر کدام از این مشکلات را با روش خاصی می‌توان بهبود داد. اگر تغییرات شدت روشنایی وجود نداشته باشد، در این حالت باید اختلاف شدت روشنایی بین دو تصویر در ناحیه مشترک، صفر باشد. در صورت صفر نبودن اختلاف شدت روشنایی‌های بین هر دو تصویر در ناحیه مشترک، باید اثر آن جبران شود [۹]. برای رفع اختلاف منظر همچنین می‌توان بجای استفاده از ماتریس هموگرافی از ماتریس پروجکشن استفاده کرد که دارای خطای کمتری است. هر چند برای استفاده از این امر (استفاده از عمق نقاط) می‌بایست از تطابق استریو چگال استفاده کرد که بار محاسباتی بسیار زیادی را در بر خواهد داشت.

### استخراج ویژگی

برای به دست آوردن نقاط کلیدی تصویر و ویژگی‌های آن روش‌های متفاوتی ارائه شده است. بدست آوردن این ویژگی‌ها در تصاویری که در شرایط نوری و در زاویه‌ها و مقیاس‌های متفاوتی گرفته می‌شوند دارای اهمیت بالایی است [۱۰]. از جمله این روش‌ها که مورد توجه است، استفاده از الگوریتم HARRIS و SIFT می‌باشد که آن‌ها را مورد بررسی قرار می‌دهیم.

### الگوریتم HARRIS

یکی از روش‌های استخراج ویژگی استفاده از الگوریتم HARRIS [۱۱] می‌باشد.

آشکار ساز هریس، اگرچه در برابر تغییرات مقیاس مقاوم نیست، ولی کارایی بالایی را در مقایسه با دیگر آشکار سازها دارد. برای آشکار سازی نقاط هریس از ماتریس ممان دوم (M) استفاده می‌شود [۱۲].

$$M = \sigma_D^2 g(\sigma_I) \times \begin{bmatrix} f_x^2(x, \sigma_D) & f_x(x, \sigma_D) f_y(x, \sigma_D) \\ f_x(x, \sigma_D) f_y(x, \sigma_D) & f_y^2(x, \sigma_D) \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \quad (3)$$

$$f_x(x, \sigma_D) = \frac{\partial}{\partial x} g(\sigma_D) * f(x), \quad (4)$$

$$f_y(x, \sigma_D) = \frac{\partial}{\partial y} g(\sigma_D) * f(x)$$

$$g(\sigma) = \frac{1}{2\pi\sigma^2} \quad (5)$$

در این رابطه مشتق‌های محلی تصویر  $(f_x, f_y)$  با کرنل گاوسی مقیاس  $\sigma_D$  (مقیاس مشتق‌گیری) محاسبه می‌شود سپس با استفاده از پنجره گاوسین به سائز  $\sigma_I$  (مقیاس انتگرال‌گیری) میانگین‌گیری می‌شود. مقادیر ویژه این ماتریس، تغییرات را در

به عنوان مرجع انتخاب می‌کنند و سپس بقیه تصاویر را بر روی مختصات آن تصویر می‌پیچند. نتیجه این ترکیب را گاهی اوقات پانورامای تخت می‌نامند. اما هنگامی که صحنه بزرگ باشد معمولاً نمی‌توان از پانورامای تخت استفاده کرد زیرا در این حالت می‌بایست پیکسل‌های گوشه‌ی تصاویر کشیده شوند که مطلوب نیستند. در این حالت‌ها به طور معمول از ساختار استوانه‌ای یا کروی استفاده می‌شود [۸].

برای اینکه تصویر یک پارچه و قابل قبولی را ارائه دهیم می‌بایست تصمیم بگیریم که کدام پیکسل‌های تصاویر ورودی را می‌خواهیم استفاده کنیم و تعیین کنیم که وزن دهی پیکسل‌ها به چه صورت قرار است باشند. برای این کار می‌توان از دو روش میانگین‌گیری و میانگین‌گیری وزن دار استفاده کرد. در روش اول از تمام شدت روشنایی‌هایی که در یک موقعیت بر روی نواحی مشترک قرار دارند میانگین گرفته می‌شود. این روش ساده و سریع است ولی ممکن است نتایج خیلی خوبی نداشته باشد. روش دیگر برای این کار میانگین وزن دار از پیکسل‌هاست [۸].

$$C(x) = \sum_K \frac{W_K(x) I_K(x)}{\sum_K W_K(x)} \quad (2)$$

که در آن W مقدار دودویی صفر یا یک را می‌گیرد که معتبر یا نامعتبر بودن پیکسل را مشخص می‌کند و I شدت نور پیکسل بعد از نگاشت بر روی سطح ترکیبی می‌باشد. در روش دوم وزن دادن بر اساس فاصله از موقعیت‌های نواحی مشترک تصاویر است. در نتیجه اگر یک پیکسل دور از نقاط مشترک بین دو تصویر باشد، وزن کمتری خواهد داشت حال آنکه پیکسل‌های نزدیک به نقاط مشترک، وزن بیشتری خواهند داشت. دقت این روش بیشتر از روش اول است و ما از آن در الگوریتم پیشنهادی استفاده کرده‌ایم. در صورتی که اشیاء متحرکی در صحنه داشته باشیم یک فیلتر میانه می‌تواند نتیجه بهتری را بدست آورد. همچنین استفاده از احتمال کمینه می‌تواند کپی‌هایی از اشیاء را بدست آورد.

### تصحیح بهبود تباین (کنتراست)

وقتی که جهت‌ها و فاصله کانونی دوربین‌ها را مشخص می‌کنیم، ممکن است تصاویر بدست آمده باز هم به درستی همتراز نشده باشند و تصاویر موزاییک شده مات یا مانند روح شده باشند. دلایل ایجاد این حالت‌ها می‌تواند درست مدل نکردن اعوجاج‌ها یا اختلاف منظر سه بعدی (عدم دوران درست دوربین حول مرکز آن) یا حرکت کوچک اشیاء مانند حرکت موجی شکل پرچم و یا حرکات بزرگتر مانند جابه‌جایی اشیاء بزرگ یا حرکت آدم‌ها باشد که در بدترین حالت می‌تواند اضافه یا حذف شدن یک شیء به تصویر باشد [۹].

کانولوشن تصویر دو بعدی  $I(x,y)$  با گاوسین  $g(0;\sigma)$  معمولاً با اعمال جداگانه دو فیلتر گاوسین یک بعدی انجام می‌شود. هر کانولوشن یک بعدی به وسیله گسسته شده با اندازه ثابت انجام می‌گیرد که این اندازه اغلب یک عدد فرد است که از  $3\sigma$  بزرگتر می‌باشد. برای فیلترهای گاوسین  $g(0;\sigma_1)$  و  $g(0;\sigma_2)$  رابطه زیر برقرار می‌باشد:

$$g(0;\sigma_1) * g(0;\sigma_2) = g(0;\sqrt{\sigma_1^2 + \sigma_2^2}) \quad (7)$$

می‌توان این کار را برای  $\sigma$ های بزرگ به صورت بازگشتی انجام داد تا در محاسبات صرفه‌جویی شود.

روش‌های شناسایی ویژگی‌های تغییر ناپذیر نسبت به مقیاس معمولاً کارایی مناسبی در کاربردهای زمان واقعی<sup>۴</sup> ندارند. استفاده از هرم تصویر می‌تواند این زمان را در حد قابل توجهی کاهش دهد [۴].

این الگوریتم نسبت به دوران، تغییرات روشنایی، تغییر موقعیت و تغییر شکل افاین پایدار است اما به خاطر حجم محاسبات بالایی که دارد از سرعت پایین برخوردار است.

### روش اجماع اتفاقی نمونه<sup>۵</sup>

الگوریتم RANSAC [۱۴] الگوریتمی قوی برای تطابق مدل‌ها است. این روش قابلیت تفسیر داده‌هایی که حاوی درصد بالایی از خطاهای فاحش هستند را دارد. در واقع این روش یک راهکار تکرار پذیر برای بدست آوردن مدل با بیشترین نمونه ممکن از روی مجموعه نقاط است. این الگوریتم برای محدوده وسیعی از مسائل تخمین پارامتر مدل‌ها در بینایی کامپیوتر به کار می‌رود، از جمله این مسائل، مسئله تطابق ویژگی‌ها، کشف و رجیستر کردن مشخصه‌های هندسی و غیره می‌باشد.

در این الگوریتم ابتدا نمونه‌ها به طور یکنواخت و شانسی از بین مجموعه داده‌های ورودی انتخاب می‌شوند. هر داده، شانس یکسانی برای انتخاب شدن دارد. در این مرحله نقاطی که خطای آن‌ها نسبت به مقداری که مدل به ما می‌دهد از یک مقدار مشخص کمتر باشد را نقاط درون نهاد<sup>۶</sup> می‌نامیم و نقاطی که میزان خطای آن‌ها از این مقدار بیشتر باشد نقاط برون نهاد یا پرت<sup>۷</sup> می‌نامیم.

دو جهت متعامد نقطه ارائه می‌کند. بنابراین در پیکسل‌هایی که این مقادیر بزرگ باشند می‌توان محل اتصال یا گوشه‌ها را پیدا کرد. به این خاطر که محاسبه مقادیر ویژه پیچیده است می‌توان از رابطه زیر که هزینه کمتری دارد ماکزیمم محلی (R) را تعیین کرد:

$$R = \det(M) - \alpha \times \text{trace}^2(M) = (ac - b^2) - \alpha(a + c)^2 \quad (6)$$

در این رابطه  $\alpha$  ضریب تنظیم است که به صورت تجربی مقدار آن در حدود  $0.06$  انتخاب می‌شود. بنابراین با تعیین R، بیشینه‌های محلی به عنوان گوشه‌های هریس تعیین می‌شود. الگوریتم هریس هنگامی که تغییرات روشنایی وجود داشته باشد به خوبی نقاط کلیدی را مکان یابی می‌کند اما زمانی که تصاویر در زاویه دید متفاوتی گرفته شده باشند و تغییراتی مانند تغییرات مستوی وجود داشته باشد چندان کار ساز نمی‌باشد.

### الگوریتم SIFT<sup>۱</sup>

الگوریتم SIFT یکی از الگوریتم‌های استخراج ویژگی‌های کلیدی در تصویر است که ویژگی‌های محلی تصویر را استخراج و توصیف می‌کند. این الگوریتم که در سال ۱۹۹۹ توسط دیوید لائو ارائه شد [۴] کاربردهای فراوانی در زمینه شناسایی اشیاء، هدایت ربات، بازیابی تصاویر، مدل سازی سه بعدی و تشخیص حرکت دارد. این ویژگی نسبت به انتقال شیء در تصویر، چرخش و تغییر مقیاس آن تغییر ناپذیر است و همچنین نسبت به تغییرات ملایم تبدیل تجسم سه بعدی<sup>۲</sup> مقاوم می‌باشد.

برای دست یابی به تغییر ناپذیری نقاط کلیدی نسبت به مقیاس، باید آشکارساز مورد نظر در فضا مقیاس<sup>۳</sup> تصویر عمل کند. ابتدا تصویر توسط مقیاس‌های متفاوت از تابع گاوسی فیلتر شده و سپس مشتق گاوسین تصویر فیلتر شده محاسبه می‌شود. فضای مقیاس تصویر از بازنمایی‌های مختلف یک تصویر در سطوح دقت مختلف حاصل می‌شود. تصویر اولیه در هر دقت، از کانولوشن آن تصویر با یک کرنل مشخص مثلاً تابع گاوسی به دست می‌آید [۱۳و۴].

کانوالو کردن یک تصویر به صورت افزایشی به وسیله فیلتر گاوسین  $g(0;\sigma)$  با  $\sigma$  کوچک از لحاظ محاسباتی کم هزینه‌تر از یک کانولوشن تنها با مقدار  $\sigma$  بزرگ می‌باشد.

<sup>۵</sup> RANdom SAMple Consensus  
<sup>۶</sup> Inlier  
<sup>۷</sup> Outlier

<sup>۱</sup> Scale Invariant Feature Transformation  
<sup>۲</sup> Perspective  
<sup>۳</sup> Scale-Space  
<sup>۴</sup> Real time

آوردن مدل مناسب حداقل  $n$  داده مورد نیاز باشد و یک مجموعه شامل  $p > n$  داده داشته باشیم. در این صورت الگوریتم RANSAC به صورت زیر می‌باشد.

۱. به صورت تصادفی  $n$  نقطه از مجموعه داده انتخاب کن.
۲. پارامترهای مدل را به وسیله این  $n$  نقطه انتخاب کن.
۳. مجموعه نقاط معتبر  $S_i$  را با توجه به مدل  $M$  مورد نظر مشخص کن. این مجموعه داده  $S_i$  مجموعه‌ای هستند که روی مدل محاسبه شده اجماع دارند.
۴. اگر تعداد اعضای مجموعه اجماع  $S_i$  از یک حد آستانه  $T$  بزرگتر بود از این اعضا برای تخمین مدل استفاده کن و حلقه را پایان بده.

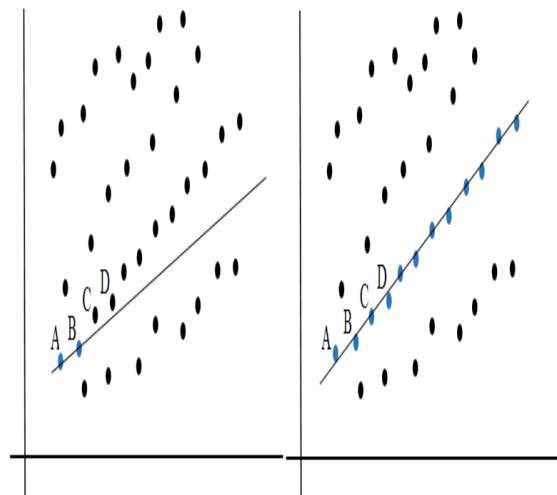
اگر تعداد تکرار حلقه از  $N$  کوچکتر بود برو به مرحله ۱ و در غیر اینصورت بزرگترین مجموعه  $S_i$  را انتخاب و با استفاده از اعضای آن، پارامترهای مدل را محاسبه کن و حلقه را پایان بده.

### روش پیشنهادی

در این مقاله روشی مناسب برای ترکیب و منطبق کردن تصاویر که اعوجاج‌های ناشی از ایجاد تصویرهای موزاییکی را کم می‌کند، ارائه شده است. در روش پیشنهادی برای شناسایی نقاط پرت و توالی به جای استفاده از ویژگی‌های محلی از ویژگی‌های هندسی نقاط استفاده می‌شود. به این مفهوم که بجای اینکه فقط به محل نقاط در تصاویر دقت شود به ویژگی‌های هندسی از جمله چرخش و مقیاس و ... نیز توجه گردد. اگر به عنوان مثال نقاطی در لبه‌ها به دست آمده و نقاط متناظر آن فقط در لبه‌های تصاویر دیگر جستجو شود، احتمال اشتباه در نظر گرفتن نقاط متناظر بالا خواهد رفت. اما اگر علاوه بر توجه به محل قرار گرفتن نقاط تصاویر، به ویژگی‌های هندسی از جمله چرخش و یا جهت آن نیز توجه شود و نتایج آن به عنوان نقاط متناظر در دیگر تصاویر بررسی شود، درصد خطا به شدت پایین خواهد آمد. براساس ویژگی‌های استخراج شده میانگین این ویژگی‌ها طبق رابطه (۲) به ازای تمام نقاط به دست می‌آید. در این مرحله برای جلوگیری از بالا رفتن زمان اجرای الگوریتم، به جای استفاده مستقیم از الگوریتم RANSAC، به منظور حذف نقاط پرت، به تقسیم بندی تصاویر می‌پردازیم و با درجه بندی و آستانه گذاری تصاویر، سعی در پیدا کردن توالی تصاویر می‌پردازیم. همچنین با استفاده از روش‌های ارائه شده، بهترین همترازی را بر روی تصاویر انجام می‌دهیم به طوری که سرعت و دقت را از دست ندهیم. در ادامه مراحل الگوریتم پیشنهادی مورد بررسی قرار می‌گیرد.

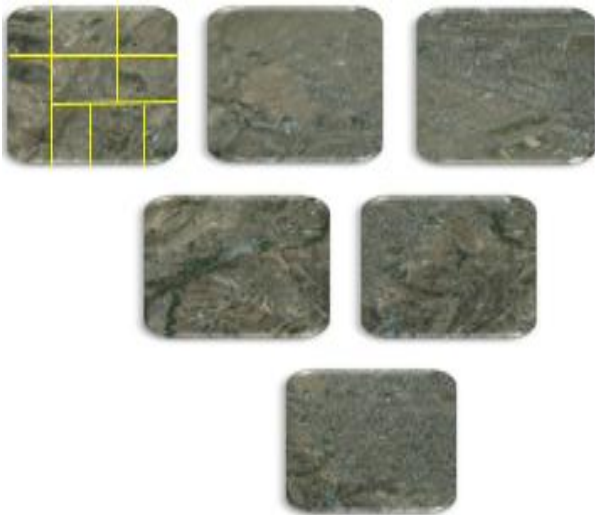
این روش یک روش غیر قطعی است که میزان دقت تخمین‌ها عموماً با تکرار بیش‌تر الگوریتم به منظور پیدا کردن مدل مناسب حاصل می‌گردد. داده‌های ورودی ممکن است چندین مدل مختلف را تایید کنند. به این صورت که در مرحله اول با تعدادی از نقاط انتخابی، مدلی را مشخص می‌کند و در مرحله‌های بعد، دوباره با استفاده از نقاط قبلی و نقاط جدید این روند را تکرار می‌کند که از این لحاظ سرعت الگوریتم به خصوص برای تصاویر با حجم بالا به شدت پایین می‌آید. الگوریتم پیشنهادی روشی برای جبران این سرعت پیشنهاد داده است که در قسمت مربوطه توضیحات آن آورده شده است.

پس از چندین مرحله تکرار حالتی که بیشترین دامنه را به ما می‌دهد به عنوان خروجی در نظر گرفته می‌شود. یک مثال ساده از چگونگی انتساب مدل به مجموعه‌ای از نقاط، انتساب یک خط در فضای دو بعدی می‌باشد که در شکل ۲ قابل مشاهده است. ابتدا با استفاده از نقاط  $A$  و  $B$  یک تخمین ابتدایی از پارامترهای مدل (خطی که این مجموعه نقاط ساخته‌اند) زده شده است که با توجه به آستانه، نقاط  $C$  و  $D$  از آن حمایت کرده‌اند و تخمین جدیدی زده می‌شود. سپس به وسیله بهینه سازی مدل براساس چهار نقطه  $A, B, C$  و  $D$  پارامترهای دقیق‌تری برای مدل به دست آورده شده است. همین روند برای بقیه نقاط نیز تکرار می‌شود و مدل‌های مختلف از آن استخراج می‌گردد.



شکل ۲: خط منتسب به مجموعه نقاط با روش RANSAC

روش RANSAC یک حد آستانه به منظور مقدار تحمل خطا و همچنین مشخص کننده حجمی که تمام نقاط سازگار باید در آن قرار بگیرند را در نظر می‌گیرد. در هنگام وجود تعداد زیادی نقاط پرت، تعداد مدل‌های مشخص کننده تخمین مناسب افزایش پیدا می‌کند. با استفاده از متغیر آستانه، تعداد پاسخ قابل قبول برای تعیین مدل کاهش می‌یابد. فرض کنید برای بدست



شکل ۳: بخش بندی تصویر

برای هر بخش پنجره‌هایی با ابعاد مشخص در نظر می‌گیریم و میانگین هیستوگرام نقاط مرتبط در هر پنجره را طبق رابطه ۲ بدست می‌آوریم. آنگاه طبق رابطه ۸ تابع چگالی هسته‌ای با استفاده از مشاهده‌های  $x_1, \dots, x_n$  از میانگین هیستوگرام پیکسل‌های تصویر بدست خواهد آمد.

$$p_r(x_t) = \frac{1}{N} \sum_{i=1}^N k_\sigma(x_t - x_i) \quad (4)$$

در این حالت براساس مشاهدات انجام شده تا زمان جاری و مقدار پیکسل در حالت جاری، تابع چگالی به عنوان احتمال تطابق پیکسل با پیکسل پنجره هدف در نظر گرفته می‌شود.

در رابطه  $k_\sigma, \lambda$  تابع هسته‌ای می‌باشد که پهنای باند آن  $\sigma$  است. در واقع مقدار پیکسل در فریم جاری با رفتار پیکسل مبتنی بر  $N$  مشاهده اخیر سنجیده می‌شود و در صورت عدم تطابق به عنوان نقطه پرت در نظر گرفته می‌شود.

احتمالات را برای همه پنجره‌ها نسبت به هم به دست می‌آوریم. احتمالات هر پنجره با پنجره‌های دیگر را درجه بندی می‌کنیم. ماکزیمم احتمال اشتراک برای هر پنجره با ضریب یک در نظر گرفته می‌شود و پایین‌ترین احتمال اشتراک با ضریب صفر و مابقی آن‌ها از صفر تا یک مقداردهی می‌شود. در این صورت تاثیر آن دسته از احتمالاتی که داری مقادیر پایین هستند در احتمال نهایی کمتر می‌گردد.

در این الگوریتم ابعاد پنجره می‌تواند متفاوت باشد. با بزرگ در نظر گرفتن پنجره، از حجم محاسبات کمتری برخوردار خواهیم بود و در نتیجه در زمان کمتری محاسبات انجام می‌شود ولی باید این را هم در نظر گرفت که با بزرگتر در نظر گرفتن پنجره‌ها، دقت الگوریتم پایین خواهد آمد. در این آزمایش ما طول پنجره‌ها را براساس هیستوگرام پیکسل‌های تصویر در نظر گرفتیم. به این

### محاسبه مدل هندسی بین دو تصویر

برای محاسبه مدل هندسی بین دو تصویر ابتدا باید نقاط کلیدی در تصویر مشخص شده و سپس نقاط تطبیقی آنها محاسبه شوند. برای استخراج نقاط کلیدی می‌توان از گوشه‌های تصویر استفاده کرد که نقاط تطبیقی آنها با دقت بالایی قابل محاسبه است [۱۵]. برای تطبیق نیز می‌توان از روش‌هایی مثل همبستگی نرمالیزه یا روش هرم استفاده کرد. اما نکته‌ای که باید به آن توجه کرد این است که نقاط تطبیقی بعضی از گوشه‌ها به علت تغییر شرایط تصویر برداری و تغییرات سه بعدی تصویر و وجود نقاط پرت ممکن است به صورت درست محاسبه نشوند. بنابراین باید از روش‌های آماری مناسبی برای محاسبه مدل هندسی بین دو تصویر استفاده کرد تا مدل به صورت دقیق محاسبه شود. همانطور که گفته شد یکی از روش‌هایی که می‌توان برای حذف نقاط پرت در نظر گرفت استفاده از الگوریتم RANSAC می‌باشد ولی به خاطر روند تکراری این الگوریتم، زمان زیادی را از دست خواهیم داد. پس به طریقی دیگر عمل می‌کنیم تا هم این زمان را جبران کنیم و هم دقت را از دست ندهیم. توضیحات این روش در بخش بعد آورده شده است.

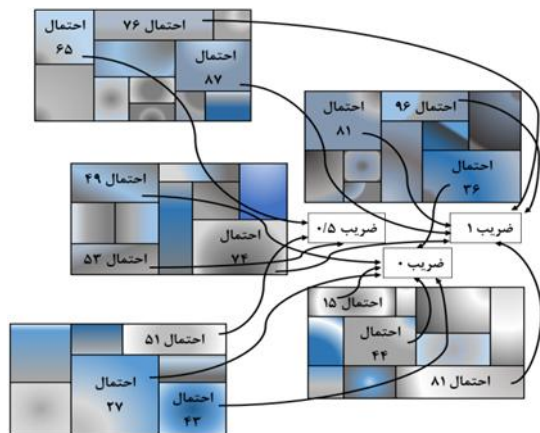
### استخراج ویژگی هندسی نقاط مرتبط و یافتن توالی عکس‌ها

بعد از مشخص شدن نقاط مرتبط باید به ازای هر زوج نقطه، مشخصات هندسی آن‌ها مشخص شود که این مشخصات هندسی به عنوان مثال می‌تواند میزان چرخش و نسبت مقیاس باشد. انتظار می‌رود که اختلاف چرخش همه نقاط به یک میزان باشد. بنابراین نقاطی که اختلاف چرخش آن‌ها با بقیه متفاوت است به اشتباه با هم مرتبط شده‌اند و چون زاویه دید دوربین نسبت به همه نقاط به یک نسبت تغییر می‌کند بنابراین باید نسبت مقیاس تمام نقاط به یک نسبت تغییر نماید.

اما در صورتی که اختلاف چرخش همه نقاط به یک میزان نباشد این روش دچار خطا می‌شود پس روش دیگری را در نظر می‌گیریم، به این صورت که تصویر را مانند شکل ۳ به بخش‌های کوچکتر تقسیم می‌کنیم.

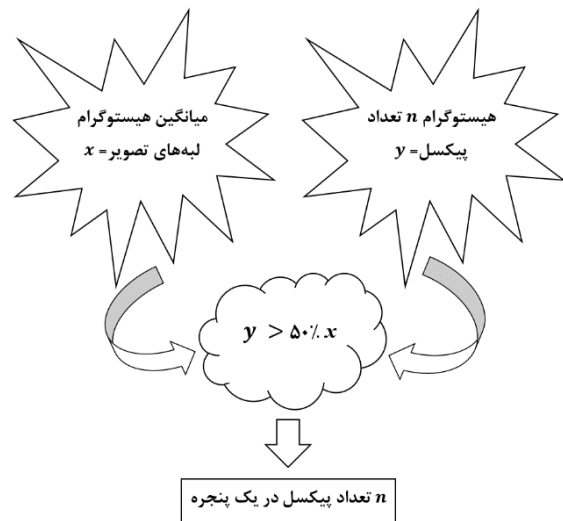
عمل نمی کند و ما فقط به خاطر مقایسه روی یک پنجره آن را اعمال کردیم. اما در جدول ۲ به خاطر اینکه روی یک پنجره با محدود کردن نقاط انتخابی الگوریتم را اعمال کردیم، درصد خطا کمتر شده است و نسبت به احتمالات بدست آمده، از زمان مناسب تری برخوردار شده است. به همین خاطر می توان کل تصویر را به همین شکل تقسیم بندی کرد و روی نقاطی که درصد این احتمالات بالا می باشد الگوریتم RANSAC را اعمال کرد چرا که در این صورت RANSAC محدود خواهد شد و روی نقاط تکراری زمان را از دست نخواهد داد و در نتیجه نتایج آن از هر لحاظ بهتر خواهد بود.

هرچند احتمالات پایین تر از ۷۰ درصد، شانس کمی در نتیجه دارد اما به دلیل اینکه ممکن است در انتخاب ابعاد پنجره دچار اشتباه شده باشیم، احتمالات بین ۵۰ تا ۷۰ را با ضریب ۰/۵ و احتمالات پایین تر از ۵۰ درصد را با ضریب صفر در نظر می گیریم (شکل ۵). بعد از میانگین گیری اگر در هر تصویر مقادیر به دست آمده بالاتر از ۵۰ درصد باشد متوالی در نظر گرفته می شود. آنگاه الگوریتم را جهت همترازی با نزدیک ترین همسایه ادامه می دهیم.



شکل ۵: رتبه بندی نواحی مختلف در الگوریتم پیشنهادی

مفهوم که ابتدا میانگین هیستوگرام لبه‌های تیز تصویر را به دست می آوریم، آنگاه به ازای هر تعداد پیکسلی که هیستوگرام آن‌ها از ۵۰ درصد میانگین هیستوگرام لبه‌ها بالاتر بود یک پنجره در نظر خواهیم گرفت. در این صورت حتی اگر تصاویر شلوغ باشد و یا شدت روشنایی در تصاویر زیاد باشد با این روش هیستوگرام دچار نوسان نخواهد شد (شکل ۴).



شکل ۴: طریقه به دست آوردن ابعاد پنجره

در جدول ۱ و ۲ می‌توانید احتمالات اشتراک نواحی مختلف را مشاهده کنید که برای حالت‌های مختلف به دست آمده است. لازم به ذکر است که این احتمالات در جدول ۱ برآیند احتمالات نقاط صحیح نسبت به نقاط پرت می‌باشد و در جدول ۲ در واقع برآیند احتمالات پنجره‌های مختلف به دست آمده از هیستوگرام رنگی و تابع چگالی هسته‌ای به منظور انتخاب نقاط صحیح با استفاده از الگوریتم پیشنهادی است.

مقادیر جداول ۱ و ۲ را برای پنجره ای با ابعاد  $40 \times 40$  مورد مقایسه قرار داده ایم. الگوریتم RANSAC در نواحی مختلف به صورت تصادفی احتمالات را به دست می آورد به همین خاطر همان طور که در جدول ۱ می بینید در بعضی از قسمت ها مقدار احتمال کمی را به دست آورده است و همچنین به خاطر عمل تکرار پذیری آن زمان بیشتری را صرف کرده است. هر چند که به طوری عادی الگوریتم RANSAC بر روی بخشی از تصویر

جدول ۱: محاسبه احتمالات با طول پنجره ۴۰×۴۰ با استفاده از الگوریتم RANSAC

| نواحی اشتراک | نمونه برداری ۱۰۰ در ۹۰۰ پیکسل |             | نمونه برداری ۳۰۰ در ۹۰۰ پیکسل |             | نمونه برداری ۵۰۰ در ۹۰۰ پیکسل |             |
|--------------|-------------------------------|-------------|-------------------------------|-------------|-------------------------------|-------------|
|              | احتمال(درصد)                  | زمان(ثانیه) | احتمال(درصد)                  | زمان(ثانیه) | احتمال(درصد)                  | زمان(ثانیه) |
| ۲.۱,۱.۳      | ۳۶/۲۸                         | ۰/۲۸        | ۳۵/۲۳                         | ۰/۹۶        | ۳۱/۱۸                         | ۱/۳۵۲       |
| ۲.۶,۱.۵      | ۴۲/۳۴                         | ۰/۳۸        | ۴۵/۲۳                         | ۰/۶۹        | ۳۶/۲۵                         | ۱/۸۶۹       |
| ۲.۴,۱.۵      | ۵۹/۳۴۸                        | ۰/۳۳        | ۶۳/۵۴                         | ۰/۸۹        | ۶۸/۳۵                         | ۱/۲۵        |
| ۲.۵,۱.۶      | ۷۱/۵۶۷                        | ۲/۶۵        | ۶۶/۲۳                         | ۲/۳۶        | ۱۲/۳۸                         | ۰/۵۳        |
| ۲.۸,۱.۸      | ۶۱/۴۱                         | ۰/۴۵        | ۵۱/۲۶                         | ۱/۸۶        | ۶۵/۷۴                         | ۱/۲۹        |
| ۳.۳,۲.۳      | ۳۹/۳۱                         | ۱/۴۸        | ۴۷/۳۶                         | ۱/۸۸        | ۵۶/۰۱                         | ۱/۶۸۹       |
| ۴.۱,۳.۳      | ۵۳/۳۹                         | ۰/۸۹        | ۴۸/۶۳                         | ۱/۲۴        | ۵۹/۶۳                         | ۱/۹۹        |
| ۴.۲,۳.۵      | ۳۶/۲۸                         | ۰/۹۶        | ۲۳/۴۵                         | ۰/۸۴        | ۴۶/۲۶                         | ۱/۲۹        |

جدول ۲: محاسبه احتمالات با طول پنجره ۴۰×۴۰ با استفاده از الگوریتم پیشنهادی

| نواحی اشتراک | نمونه برداری ۱۰۰ در ۹۰۰ پیکسل |             | نمونه برداری ۳۰۰ در ۹۰۰ پیکسل |             | نمونه برداری ۵۰۰ در ۹۰۰ پیکسل |             |
|--------------|-------------------------------|-------------|-------------------------------|-------------|-------------------------------|-------------|
|              | احتمال(درصد)                  | زمان(ثانیه) | احتمال(درصد)                  | زمان(ثانیه) | احتمال(درصد)                  | زمان(ثانیه) |
| ۲.۱,۱.۳      | ۷۳/۲۸                         | ۱/۲۶        | ۷۰/۸۸                         | ۰/۷۶        | ۹۶/۳۶                         | ۰/۵۶        |
| ۲.۶,۱.۵      | ۶۷/۳۶                         | ۰/۳۲        | ۸۱/۱۲                         | ۰/۳۴        | ۷۸/۰۶                         | ۰/۵۹        |
| ۲.۴,۱.۵      | ۴۶/۵۴                         | ۰/۶۵        | ۷۳/۶۷                         | ۰/۵۶        | ۹۶/۲۷                         | ۱/۳۶۳       |
| ۲.۵,۱.۶      | ۸۵/۲۶۳                        | ۰/۵۴        | ۸۳/۱۱                         | ۱/۱۶        | ۴۳/۵۸                         | ۰/۹۳        |
| ۲.۸,۱.۸      | ۴۸/۱۳                         | ۰/۵۵        | ۶۵/۶۷                         | ۱/۶۶        | ۷۶/۴۱                         | ۱/۰۹        |
| ۳.۳,۲.۳      | ۸۷/۴۷                         | ۰/۷۵        | ۹۲/۸۲                         | ۰/۵۳        | ۸۶/۷۰                         | ۰/۹۸۳       |
| ۴.۱,۳.۳      | ۸۳/۵۷                         | ۰/۶۷        | ۷۶/۱۳                         | ۱/۱۰        | ۶۲/۳۵                         | ۱/۱۱        |
| ۴.۲,۳.۵      | ۳۸/۹۵                         | ۰/۵۷        | ۵۶/۳۶                         | ۰/۶۷        | ۴۸/۴۶                         | ۰/۹۱        |

- احتمالات بالاتر از ۷۰٪ با ضریب یک، بین ۵۰٪ تا ۷۰٪ با ضریب نیم و احتمالات زیر ۵۰٪ با ضریب صفر در نظر گرفته می شود. بعد از میانگین گیری اگر در هر تصویر مقادیر به دست آمده بالاتر از ۵۰٪ باشد متوالی در نظر گرفته می شود در غیر این صورت به مرحله ۴ بر می گردیم.

۷- یافتن تبدیل بین تصاویر و همتراز کردن آنها

۸- اتصال تصاویر به یکدیگر و ساخت یک تصویر با زاویه دید وسیع.

که هرکدام از مراحل فوق با توجه به بخش بندی شدن تصاویر و محاسبات مربوط به احتمالات در هر پنجره منحصر به الگوریتم پیشنهادی می باشد و موجب افزایش سرعت پردازش و حفظ دقت می گردد.

### ارزیابی روش پیشنهادی

برای مقایسه نتیجه پیاده سازی الگوریتم ارائه شده با الگوریتم‌های رایج مطرح شده در زمینه موزاییک سازی تصاویر، تصویر حاصل از اعمال این الگوریتم را با تصویر حاصل از اعمال یکی از الگوریتم‌های معمول در این زمینه مقایسه می کنیم. روش پیشنهادی، نقاط کلیدی را برای محاسبه مدل هندسی با دقت بالایی محاسبه می کند.

شکل‌های ۶ و ۷ به ترتیب نقاط کلیدی بدست آمده از الگوریتم HARRIS و الگوریتم پیشنهادی را نشان می دهد.

همانطور که در شکل ۷ مشاهده می کنیم الگوریتم ارائه شده، تعداد قابل توجهی از این نقاط را که دقت انطباق کمتری دارند حذف می کند.

حال جهت تطبیق دادن تصاویر، نقاط کلیدی دو تصویر را یکبار با الگوریتم SIFT و بار دیگر با الگوریتم پیشنهادی در حضور تغییرات روشنایی در دو تصویر، تطبیق دادیم. در صورتی که تصاویر از حجم پایینی برخوردار باشند (در حدود کیلو بایت) هر دو الگوریتم به خوبی عمل می کند و می توانیم نتایج حاصل از آن را در جدول ۳ مشاهده کنیم.

به منظور سرعت بخشی به این مرحله، آن دسته از پنجره‌هایی که احتمال اشتراک نواحی آن در مراحل قبل، بالاتر از دیگر نواحی به دست آمده است را در نظر می گیریم. در این صورت ابتدا فاصله‌ی نیمی از مولفه‌های نقاط داخل پنجره‌ی اول در تصویر اول را طبق رابطه ۹ نسبت به نقاط متناظر در پنجره‌ی اول از تصویر دوم، محاسبه کرده و در صورتی که این فاصله از یک حد آستانه کوچکتر باشد، فاصله‌ی نقاط بعدی نیز محاسبه می گردد. همین روند را برای دیگر پنجره‌ها نیز ادامه می دهیم تا در نهایت همسایه‌های نزدیک به هم پیدا شوند.

$$N_{ij} = \sum_{m=1}^M f(x_{pi}(m) - x_{qj}(m)) \quad (9)$$

که در آن تابع  $f(x)$  به صورت زیر تعریف می شود:

$$f(x) = \begin{cases} 1 & \text{if } x \neq 0 \\ 0 & \text{if } x = 0 \end{cases} \quad (10)$$

در این رابطه  $M$  تعداد مولفه‌های نقاط انتخابی،  $x_{pi}$  بردار  $i$ ام مربوط به مولفه‌ی نقطه کلیدی  $i$ ام از پنجره‌ی  $p$ ام در تصویر اول،  $x_{qj}$  بردار  $j$ ام مربوط به مولفه‌ی نقطه کلیدی  $j$ ام از پنجره‌ی  $q$ ام در تصویر دوم و  $N$  فاصله‌ی بردارها است. در این صورت اگر بردارهای همترازی تحت شرایط گفته شده، به درستی در پنجره‌های مختلف رسم گردند، می توان به بهترین انطباق دست پیدا کرد.

با توجه به توضیحات فوق می توان الگوریتم روش پیشنهادی را به صورت زیر خلاصه کرد.

۱- دریافت تصاویر و اجرای پردازش های اولیه

۲- استخراج نقاط کلیدی و مشخص شدن نقاط مرتبط

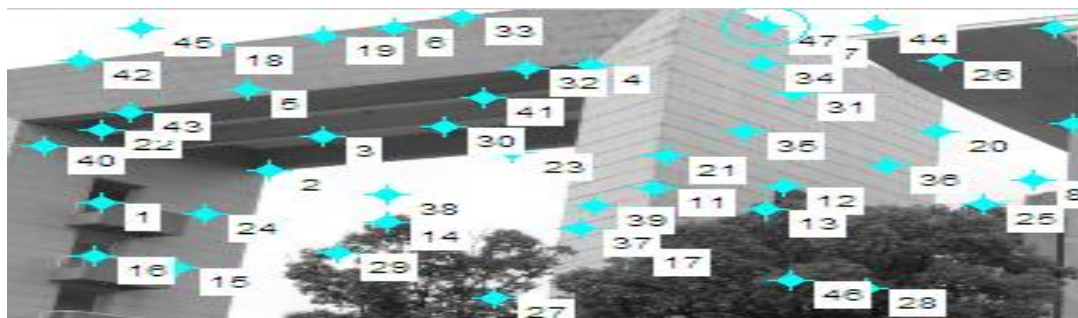
۳- بخش بندی تصاویر

۴- رتبه بندی پنجره ها و حذف نقاط پرت

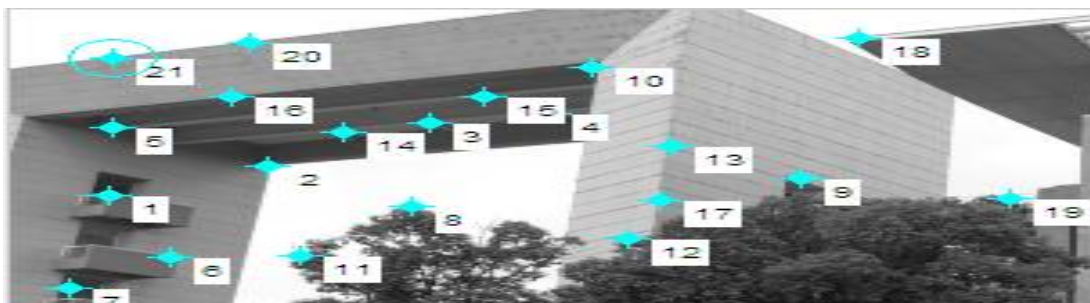
- ماکزیمم احتمال با ضریب یک ، پایین ترین احتمال با ضریب صفر و مابقی از صفر تا یک مقداردهی می شود

۵- یافتن نواحی مشترک بین تصاویر

۶- یافتن توالی بین عکس ها و شماره گذاری



شکل ۶: استخراج نقاط کلیدی با استفاده از الگوریتم HARRIS



شکل ۷: استخراج نقاط کلیدی با استفاده از الگوریتم پیشنهادی

جدول ۳: مقایسه بین روش موجود و روش پیشنهادی روی تصاویر با حجم محدود

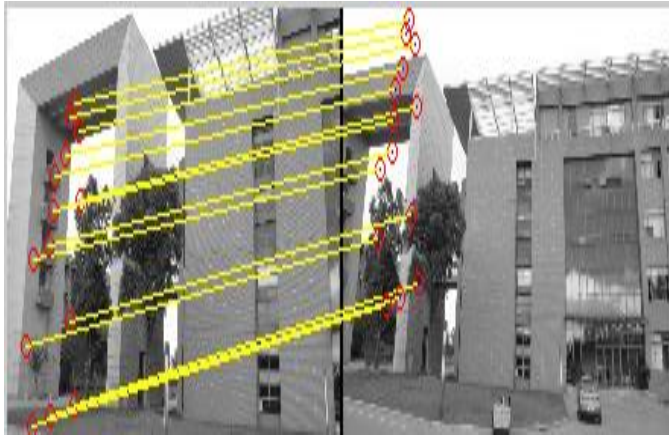
| ابعاد پنجره               | ۱۰×۱۰ | ۱۵×۱۵ | ۲۰×۲۰ | ۴۰×۴۰ |
|---------------------------|-------|-------|-------|-------|
| زمان روش SIFT (ثانیه)     | ۰/۶۴  | ۱/۱۳  | ۲/۰۶  | ۲/۵۷  |
| زمان روش پیشنهادی (ثانیه) | ۱/۰۱  | ۱/۹۷  | ۲/۱۱  | ۲/۲۴  |

جدول ۴: مقایسه زمانی بین روش موجود و روش پیشنهادی روی تصاویر با حجم بالا

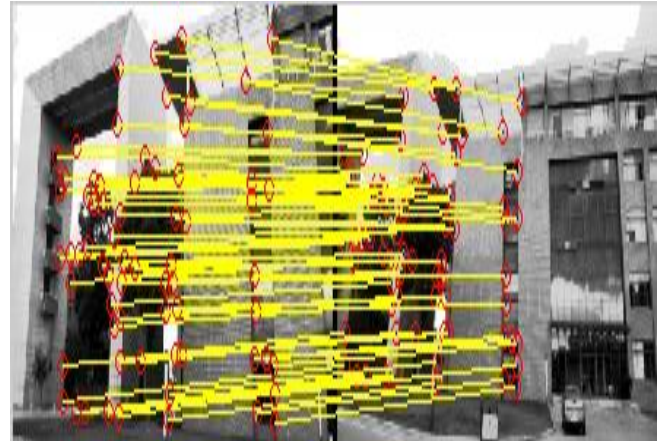
| ابعاد پنجره               | ۱۰×۱۰ | ۱۵×۱۵ | ۲۰×۲۰ | ۴۰×۴۰ |
|---------------------------|-------|-------|-------|-------|
| زمان روش SIFT (ثانیه)     | ۳/۷۶  | ۷/۶۸  | ۹/۳۶  | ۱۴/۴۶ |
| زمان روش پیشنهادی (ثانیه) | ۲/۰۳  | ۲/۸۶  | ۳/۱۲  | ۵/۸۶  |

همانطور که از جدول ۴ مشاهده می‌کنید زمان پردازش در روش پیشنهادی نسبت به روش SIFT بهبود قابل توجهی پیدا کرده است. به طوری که زمان در روش پیشنهادی بیش از دو برابر کاهش یافته است. میزان کاهش این زمان در پنجره‌های بزرگتر می‌تواند به هفت برابر هم برسد. برای بررسی همترازی و تطبیق تصاویر، مقایسه‌ای بین روش [۱۶] و روش پیشنهادی در شکل ۸ در صورتی که تعداد نقاط محدود باشد، آورده شده است.

همانطور که مشاهده می‌کنید در صورتی که ابعاد تصاویر کم باشد، روش SIFT در پنجره‌های کوچک‌تر به دلیل پایین‌تر بودن حجم محاسبات، از زمان بهتری نسبت به روش پیشنهادی برخوردار است اما با بالا رفتن ابعاد پنجره تعداد پیکسل‌ها هم بالا رفته و محاسبات SIFT نیز بیشتر می‌شود در نتیجه روش پیشنهادی رفته رفته بهتر از قبل می‌شود. حال ابعاد تصاویر را بالا می‌بریم (در حدود مگا بایت) و با شرایط قبلی نتایج را در جدول ۴ می‌توانیم ببینیم.



ب) نتیجه روش [۱۶] به ازای ۸۰ نقطه اول با حذف نقاط پرت

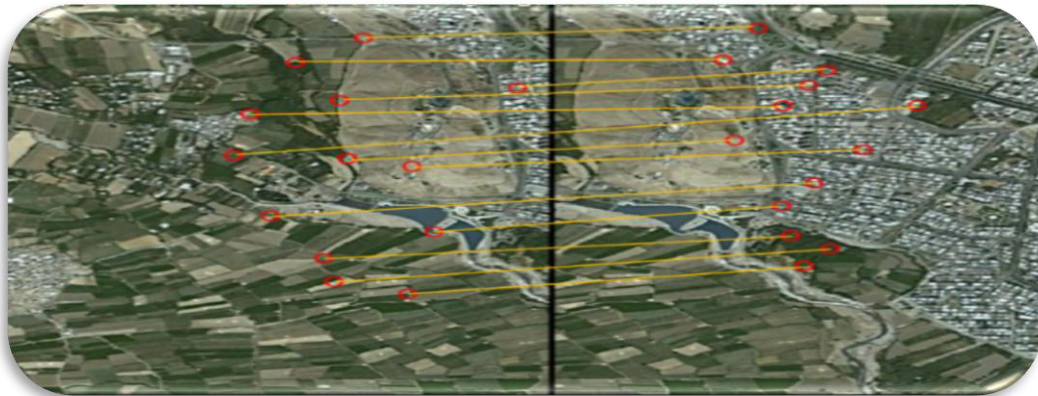


الف) نتیجه الگوریتم پیشنهادی به ازای ۸۰ نقطه اول با حذف نقاط پرت

شکل ۸: مقایسه روش پیشنهادی و روش ارایه شده در [۱۶]

تصاویر هوایی نیز اعمال می‌کنیم تا نتایج حاصل از بردار همترازی مشخص گردد. شکل‌های ۹ و ۱۰ نقاط کلیدی و همترازی بدست آمده از الگوریتم پیشنهادی و الگوریتم معمول در این زمینه را نشان می‌دهد.

همانطور که انتظار می‌رفت با استفاده از الگوریتم پیشنهادی تطبیق بهتری در دو تصویر ایجاد شد در صورتی که در شکل (۹ب) علاوه بر حذف نقاط اصلی، تطبیق اشتباهی نیز صورت گرفته است. ولی از آنجایی که هدف اصلی ما کار بر روی تصاویر هوایی با ابعاد بالا می‌باشد، این نتایج را با شرایط قبلی بر روی



شکل ۹: نتیجه الگوریتم SIFT با پارامترهای  $S=3$  و  $T=0/06$  به ازای ۴۰ نقطه اول با حذف نقاط پرت

همترازی را رسم می‌کنیم. در هر مرحله اگر خطایی رخ دهد با تغییر ابعاد پنجره و پارامترهای الگوریتم، این خطا را رفع می‌کنیم. اکنون اگر دو عکس را تطبیق دهیم به شکل ۱۰ خواهیم رسید.

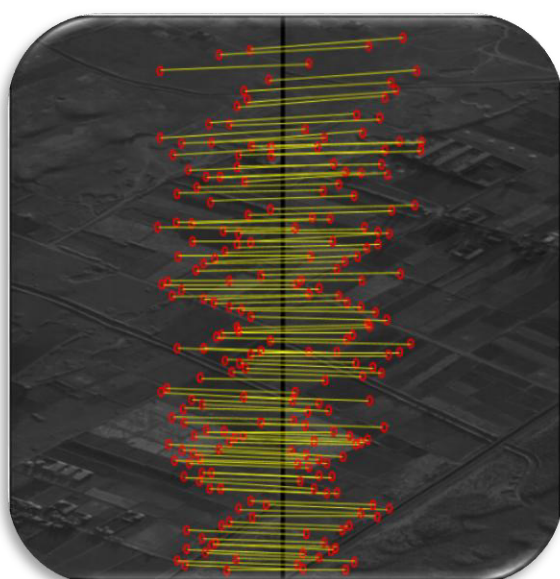
همانطور که ملاحظه می‌کنید وقتی تصاویر هوایی با حجم بالا داشته باشیم الگوریتم SIFT دچار خطا می‌شود. با به کار گیری الگوریتم پیشنهادی و به دست آوردن پارامترهای لازم، ابتدا نقاط متناظر در نواحی مشترک را پیدا می‌کنیم و سپس بردارهای



شکل ۱۰: نتیجه الگوریتم پیشنهادی به ازای ۴۰ نقطه اول با حذف نقاط پرت

هنگامی که ابعاد پنجره را کوچک در نظر بگیریم از لحاظ زمان بهتر از الگوریتم پیشنهادی عمل می‌کند ولی با افزایش ابعاد پنجره به خاطر حجم محاسبات بالای آن، زمان را از دست خواهد داد.

به منظور اطمینان از صحت الگوریتم، نتایج را با تصویر دیگری که در آن علاوه بر تغییرات روشنایی و تفاوت مقیاس، نسبت افزایش مقیاس در راستای افقی و عمودی برابر نیست، بررسی می‌کنیم.



شکل ۱۱: نتیجه الگوریتم پیشنهادی

شکل ۱۱ نتیجه همترازی با روش پیشنهادی است که بعد از تصحیح اختلاف چرخش و مقیاس به دست آمده است. به منظور بررسی دقیق‌تر، جدول ۶ مقایسه‌ای در شرایط مختلف را نشان می‌دهد.

همانطور که انتظار می‌رفت با استفاده از الگوریتم پیشنهادی تطبیق بهتری در دو تصویر ایجاد شد ولی در شکل ۹ علاوه بر حذف نقاط اصلی، تطبیق اشتباهی نیز صورت گرفته است. جزییات بیشتر در جدول ۵ آورده شده است.

جدول ۵: مقایسه نتایج حاصل از اجرای الگوریتم‌ها

| ردیف | معیار                              | الگوریتم پیشنهادی | الگوریتم HARRIS | الگوریتم SIFT |
|------|------------------------------------|-------------------|-----------------|---------------|
| ۱    | تعداد نقاط کلیدی در تصویر سمت چپ   | ۱۶۳۸              | ۱۹۴۵            | ۱۸۳۴          |
| ۲    | تعداد نقاط کلیدی در تصویر سمت راست | ۱۸۴۶              | ۲۱۲۶            | ۲۰۲۱          |
| ۳    | تعداد نقاط پرت در ۲۰ نقطه اول      | ۰                 | ۶               | ۴             |
| ۴    | تعداد نقاط پرت در ۴۰ نقطه اول      | ۳                 | ۱۳              | ۹             |
| ۵    | تعداد نقاط پرت در ۵۰ نقطه اول      | ۸                 | ۲۳              | ۱۵            |

همانطور که مشاهده می‌کنید هنگامی که تصاویری با شرایطی که گفته شد، داشته باشیم الگوریتم‌های HARRIS و SIFT نسبت به الگوریتم پیشنهادی دچار خطا می‌شود ولی الگوریتم پیشنهادی با اعمال پارامترهای هندسی امتیاز کمتری به نقاط پرت می‌دهد در نتیجه دارای دقت بالاتری خواهد بود. البته لازم به ذکر است الگوریتم SIFT در تعداد نقاط بالاتر نسبت به حالتی که نقاط کمتری در الگوریتم داشته باشد از دقت بالاتری برخوردار خواهد بود. در واقع با افزایش پیکسل‌ها، رفته رفته خطاهای خود را جبران می‌کند و به دقت قابل قبولی در همترازی می‌رسد و به نسبت الگوریتم پیشنهادی تفاوت چندانی نخواهد داشت. همچنین در تصاویر حجم پایین در بحث همترازی

جدول ۶: مقایسه نتایج حاصل از اجرای الگوریتم‌ها در شرایط مختلف

| زمان اجرا در روش پیشنهادی (ثانیه) | زمان اجرا در روش HARRIS (ثانیه) | زمان اجرا در روش SIFT (ثانیه) | چرخش (درجه) | مقیاس | تصویر راست (پیکسل×پیکسل) | تصویر چپ (پیکسل×پیکسل) |
|-----------------------------------|---------------------------------|-------------------------------|-------------|-------|--------------------------|------------------------|
| ۱۵/۴۳                             | ۲۱/۸۸                           | ۳۹/۶۵                         | +۲۲         | ۳     | ۱۳۰۶۲×۱۶۸۳               | ۱۱۷۱×۱۳۶۵۱             |
| ۱۲/۶۲                             | ۶/۹۱                            | ۱۹/۴۱                         | -۱۷         | ۴     | ۱۲۹۴×۱۶۲۷                | ۲۲۰۱×۳۲۶۸              |
| ۳۵/۰۵                             | ۲۹/۲۲                           | ۴۴/۲۶                         | +۴۵         | ۷     | ۴۰۹۶×۵۸۴۱                | ۳۱۲۶×۶۳۸۱              |

همچون ناوبری مناسب‌تر است. در نتیجه جهت بررسی با این الگوریتم، مقایسه‌ای با مقاله [۱۷] در شرایطی که علاوه بر تغییرات روشنایی و تفاوت مقیاس، نسبت افزایش مقیاس در راستای افقی و عمودی نیز برابر نیست انجام می‌دهیم. این شرایط در شکل ۱۳ مشخص شده است.



شکل ۱۳: تصاویر انتخابی در الگوریتم‌ها

در این جدول چرخش در جهت عقربه‌های ساعت را مثبت و چرخش در خلاف جهت عقربه‌های ساعت را منفی در نظر گرفته- ایم و ارقام مربوط به مقیاس، در واقع اختلاف مقیاس دو تصویر نسبت به هم می‌باشد. همانطور که مشخص است زمان اجرای الگوریتم به شرایط مختلفی وابسته است ولی آنچه واضح است این است که الگوریتم پیشنهادی نسبت به روش SIFT از سرعت بهتری برخوردار است. دلیل آن این است که با به دست آوردن احتمالات و اختصاص درجه به هر نقطه، بسیاری از نقاط اشتباه حذف می‌گردد و الگوریتم با کم شدن نقاط پرت و تطبیق اشتباه، زمان کمتری را صرف نتیجه نهایی می‌کند. البته لازم به ذکر است روش HARRIS تحت این شرایط نسبت به دو روش دیگر از دقت پایینی برخوردار است و در این جدول فقط مقایسه‌ای از لحاظ سرعت صورت گرفته است. تصویر اتصال یافته دو تصویر قبلی در شکل ۱۲ نشان داده شده است.



شکل ۱۲: تصویر اتصال یافته

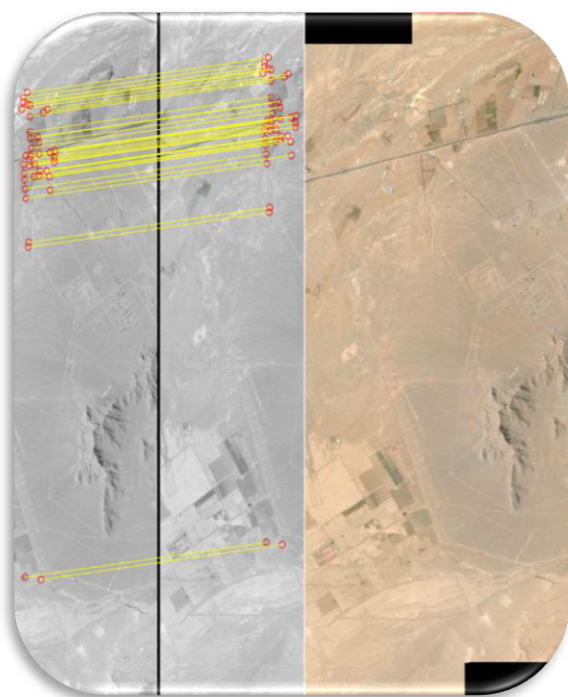
به طور کلی الگوریتم SIFT در کاربردهای عملی از کارایی بالایی برخوردار است و استفاده گسترده‌ای نیز از آن می‌شود ولی همانطور که در مطالب پیشین توضیح داده شد از لحاظ سرعت مورد پسند ما نمی‌باشد. همان طور که گفته شد، الگوریتم SURF<sup>۸</sup> شکل توسعه یافته‌ی الگوریتم SIFT از لحاظ سرعت می- می‌باشد که به دلیل سرعت بهتر برای کاربردهای بلادرنگ

<sup>۸</sup> Speeded Robust Features

جدول ۷: مقایسه نتایج حاصل از اجرای الگوریتم‌ها در شرایط مختلف

| تصویر چپ<br>(پیکسل×پیکسل) | تصویر راست<br>(پیکسل×پیکسل) | مقیاس | تعداد<br>انطباقات<br>صحیح در<br>روش<br>SIFT | تعداد<br>انطباقات<br>صحیح در<br>روش<br>SURF | تعداد<br>انطباقات<br>صحیح در<br>روش<br>پیشنهادی | زمان اجرا در<br>روش<br>SIFT<br>(ثانیه) | زمان اجرا در<br>روش<br>SURF<br>(ثانیه) | زمان اجرا در<br>روش<br>پیشنهادی<br>(ثانیه) |
|---------------------------|-----------------------------|-------|---|---|---|--|--|--|
| ۱۴۶۴×۱۸۲۸                 | ۱۷۳۴×۱۹۴۱                   | ۲     | ۱۰۱   | ۳۹  | ۱۰۶   | ۴۸/۲۱                                  | ۲۷/۶۵                                  | ۳۶/۷۲                                      |
| ۲۴۸۱×۳۷۶۵                 | ۱۵۳۱×۱۶۲۶                   | ۴     | ۷۲  | ۱۸  | ۷۹  | ۲۵/۱۴                                  | ۱۹/۷۶                                  | ۲۱/۰۲                                      |
| ۳۴۹۱×۵۲۶۱                 | ۴۹۶۷×۴۸۷۲                   | ۸     | ۴۷  | ۰   | ۳۸  | ۸۴/۳۲                                  | ۲۹/۲۵                                  | ۷۱/۶۹                                      |

مختلف در ادامه خواهد آمد. تصویر موزاییک شده نهایی با استفاده از الگوریتم پیشنهادی در شکل ۱۴ آورده شده است.



شکل ۱۴: نتیجه الگوریتم پیشنهادی بر روی شکل ۱۳

جهت نتیجه گیری کلی، بررسی‌های انجام شده روی ۵۰۰ تصویر حجم بالا را در قالب نمودار ارائه می‌دهیم. لازم به ذکر است هدف این مقاله بررسی روی تصاویر با مقیاس نه چندان بالا است ولی جهت مقایسه کلی در مقیاس بالا نیز مقایسه‌ای شده است.

ایده اصلی الگوریتم SURF آنالیز تصویر با استفاده از تصاویر انتگرالی است که با استفاده از آن مجموع پیکسل‌های درون یک مستطیل با اندازه دلخواه تنها با انجام سه جمع انجام می‌گیرد. همانطور که از جدول ۷ مشخص است روش SURF از زمان بهتری برخوردار است ولی در تصاویر حجم بالا از دقت کافی برخوردار نیست و این دلیل برتری الگوریتم پیشنهادی و الگوریتم SIFT نسبت به آن می‌باشد. در روش [۱۷] یک رویکرد جدید برای پیش بینی آستانه اولیه ماتریس Hessian ارائه کرده است که باعث کاهش زمان در استخراج نقاط متناظر می‌شود. اما نکته‌ای که وجود دارد این است که یک رابطه غیر خطی تحت تاثیر صحنه‌های مختلف بین نرخ رزولوشن و آستانه وجود دارد و با افزایش آستانه، بی ثباتی بردار توصیفگر ویژگی نقاط نیز افزایش می‌یابد. همچنین در این مقاله برای موزاییک کردن تصاویر از نگاشت تصاویر بر روی استوانه استفاده کرده است و همچنین با کم کردن تعداد بردارهای متصل شده از نقاط متناظر سعی در کم کردن زمان نهایی کرده است. اما ساخت تصاویر موزاییک شده به این روش محدودیت‌هایی دارد، اول اینکه این روش تنها در حالت‌های ساده‌ای از تغییرات تصاویر خوب عمل می‌کند. دوم اینکه اگرچه می‌توان یک تصویر دو بعدی را برای زوایای شناخته شده به مختصات استوانه‌ای منتقل کرد ولی نمونه برداری ضعیف در قطب‌های شمالی و جنوبی (بالا و پایین صحنه) منجر به خطاهای بزرگ در رجیستر کردن تصاویر می‌گردد. سوم اینکه این روش احتیاج دارد که فاصله کانونی را بدانیم. اگرچه فاصله کانونی در آزمایشگاه می‌تواند به دقت محاسبه شود اما تخمین فاصله کانونی لنز به وسیله رجیستر کردن دو یا چند تصویر نمی‌تواند دقیق باشد.

با توجه به جدول ۷ در مقیاس‌های بالاتر، روش SIFT بهتر از روش پیشنهادی عمل می‌کند که دلایل آن همان مطالبی است که قبلاً ذکر شده است و توضیحات بیشتر تحت نمودارهای

در ازای بالا رفتن زمان محاسبات حاصل می‌گردد و این موضوع را می‌توان در شکل ۱۷ مشاهده کرد.

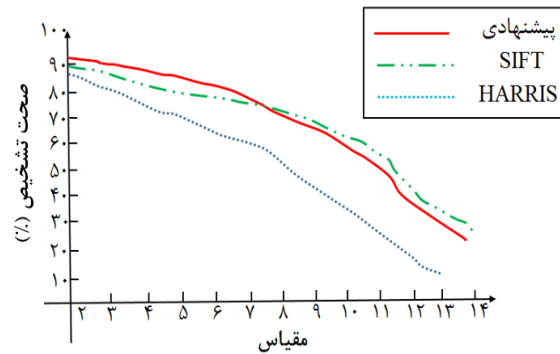
از آنجایی که روش HARRIS در برابر مقیاس چندان مقاوم نیست در نتیجه از دقت کافی نیز برخوردار نخواهد بود و با وجود اینکه با بالا رفتن مقیاس، دقت الگوریتم پیشنهادی نیز کاهش می‌یابد ولی این کاهش دقت در روش پیشنهادی نسبتاً کمتر از روش HARRIS بوده و می‌توان نتیجه گرفت روش پیشنهادی نسبت به روش HARRIS دارای عملکرد بهتری است.

شکل ۱۶ زمان اجرای روش SIFT و زمان اجرای روش پیشنهادی را نشان می‌دهد و بیان کننده این است که سرعت اجرای روش پیشنهادی در هر حالت از روش SIFT بهتر است و این ناشی از محدود کردن نقاط انتخابی در الگوریتم پیشنهادی و رتبه بندی آن‌ها است بنابراین در اجرای روش پیشنهادی زمان کمتری صرف محاسبه نقاط اشتباه می‌شود اما در روش SIFT، اضافه شدن نقاط بیشتر، منجر به محاسبات سنگین در سطح‌های مختلف می‌شود که باعث پایین آمدن سرعت آن می‌گردد.

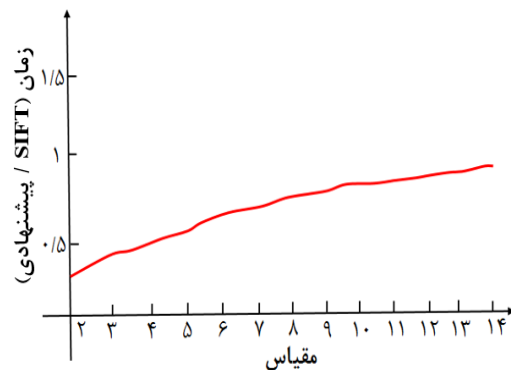
شکل ۱۷ محاسبه ی نسبت زمان اجرای روش پیشنهادی به زمان اجرا در روش HARRIS را نشان می‌دهد. این نمودار بیان می‌کند در صورتی که چرخش وجود نداشته باشد تا مقیاس حدود ۵، روش پیشنهادی زمان اجرای کمتری دارد ولی با بالا رفتن مقیاس، سرعت روش پیشنهادی کمتری می‌شود و این به دلیل بررسی تعداد پیکسل‌های بیشتر در الگوریتم پیشنهادی است؛ اما این کاهش سرعت در مقابل روش HARRIS با توجه به کمتر بودن دقت الگوریتم HARRIS نسبت به الگوریتم پیشنهادی قابل چشم پوشی است.

## نتیجه گیری

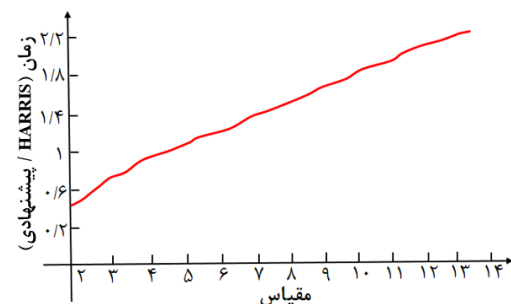
مسئله استخراج نقاط مرتبط و موزاییک بندی تصاویر از جمله مسائل چالش برانگیز در بینایی ماشین می‌باشد که هنوز توجه بسیاری از دانشمندان را به خود معطوف کرده است. در این مقاله الگوریتم جدیدی برای موزاییک بندی ارائه شد. الگوریتم ارائه شده باعث ارتقای کیفیت تصویر نهایی در موزاییک سازی تصاویر گردید. از آنجا که تمرکز عمده ی این مقاله بر استفاده از روش موزاییک بندی برای تصاویر هوایی و در جهت مصارف نظامی می‌باشد بنابراین سرعت پردازش الگوریتم بسیار حائز اهمیت است. در این الگوریتم با ارائه تکنیک های جدیدی از جمله تقسیم



شکل ۱۵: مقایسه صحت تشخیص روش‌های مختلف



شکل ۱۶: نسبت محاسبه زمان روش پیشنهادی به زمان روش SIFT



شکل ۱۷: محاسبه ی نسبت زمان اجرای روش پیشنهادی به زمان HARRIS

نمودار نشان داده شده در شکل ۱۶ نمایانگر مقایسه صحت روش SIFT و HARRIS با روش پیشنهادی است. همانطور که مشخص است تا مقیاس حدود ۷، روش پیشنهادی بهتر عمل می‌کند اما با بالا رفتن مقیاس، روش SIFT عملکرد بهتری پیدا می‌کند. هر چند در مقیاس‌های بالاتر، روش SIFT نسبت به روش پیشنهادی نتیجه بهتری حاصل می‌کند ولی همانطور که در شکل مشخص است، اختلاف کمی بین آن‌ها وجود دارد و نشان دهنده ی قابل قبول بودن روش پیشنهادی در مقیاس‌های بالا می‌باشد. همانطور که توضیح داده شد، روش SIFT دارای حجم محاسبات سنگینی است و در نتیجه عملکرد بهتر روش SIFT در مقیاس‌های بالاتر

[12] Tinne, T. and Mikolajczyk, K., "Local invariant feature detectors: a survey", Foundations and Trends. In Computer Graphics and Vision, Vol. 3, no. 3, pp 177-280, 2008.

[13] Lowe, D., "Distinctive image features from scale-invariant keypoints", Int. J. Computer. Vision, 60, 2, 91-110, 2004.

[14] Fischler, M. A., Bolles, RC., "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image analysis and Automated Cartography", Communications of the ACM, 24, 381-395, 1981.

[15] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 59-73, Aug. 2007.

[16] Jain, D.K.; Saxena, G.; Singh, V.K., "Image mosaicing using corner technique" in International Conference on Communication System and Network Technologies ,pp 79-84,2012.

[17] Cai, C., Wang, P., & Liang, Y. H. "Fast image stitching based on improved SURF" In Computer Supported Cooperative Work in Design (CSCWD), 2016 IEEE 20th International Conference on, pp. 411-416, 2016.

بندی تصاویر و درجه بندی پنجره های مختلف، نیز محدود کردن انتخاب نقاط متناظر و رتبه بندی احتمالات به دست آمده ، سرعت پردازش در دقت قابل قبولی برای تصاویر با حجم بالا افزایش پیدا کرد.

#### منابع

[1] Jiaya Jia, Member, IEEE, and Chi-Keung Tang, "Image Stitching Using Structure Deformation" *IEEE transaction on pattern analysis and machine intelligence* vol. 30, no. 4, APRIL 2008.

[2] J. Jia and C.-K. Tang, "Image stitching using structure deformation," *IEEE Trans. PAMI*, vol. 30, no. 4, pp. 617-631, Apr. 2008.

[3] D. J. Holtkamp and A. A. Goshtasby, "Precision registration and mosaicking of multicamera images," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 10, pp. 3446-3455, Oct. 2009.

[4] Lowe, D.G., "Object Recognition from Local Scale-Invariant Feature", *IEEE*, 2, 1150-1157, 1999.

[5] Bay H, Ess A, Tuytelaars T "Surf: speeded up robust features. *Comput Vis Image Underst*" vol.110, no. 3, pp. 346-359, 2006.

[6] X. Xie, Y. Xu, Q. Liu, F. Hu, T. Cai, N. Jiang, H. Xiong, "A study on fast SIFT image mosaic algorithm based on compressed sensing and wavelet transform," *J Ambient Intell Human Comput*, vol. 6 ,pp. 835-843, 2015.

[7] R. Szeliski, "Image alignment and stitching: A tutorial," in *Handbook of Mathematical Models in Computer Vision*. New York: Springer-Verlag, pp. 273-292, 2005.

[8] Szeliski. R. "computer vision: Algorithms and application" . springer-velage-New York Inc, 2010.

[9] X. Fang, B. Luo, H. Zhao, J. Tang, S. Zhai, "New multi-resolution image stitching with local and global alignment", *IET Computer Vision*, vol. 4, n. 4, p. 231-246, 2010.

[10] Schmid, C. Mohr, R. and Bauckhage, C., "Evaluation of interest point detectors", *International Journal of Computer Vision*, Vol. 37, No. 2, pp. 151-172, 2008.

[11] Harris, C and Stephens, M., "A combined corner and edge detector", *Proceeding 4th Alvey Vision Conference*, pp. 147-151, UK, 1988.