Featuresنسيم فاني

|  |  |  |
| --- | --- | --- |
| اطلاعات گزارش |  | چکیده |
| **تاریخ:** 4**/1/99** |  | ابتدا تخمين هندسي براي حالات مختلف را بررسي مي­كنيم.  با الگوريتم گوشه ياب هريس، گوشه هاي يك تصوير را مي­يابيم. |
| **واژگان كليدي:**  گوشه ياب هريس  ويژگي  geometric transformation |  |

1-مقدمه[[1]](#footnote-1)

ویژگی، یک مشخصه بصری از تصویر است که میتواند از رنگ، بافت، شکل یا لبه‌های تصویر استخراج شود. در واقع، استخراج ویژگی، فرایند تبدیل مقدارهای خام پیکسل‌های یک تصویر به اطلاعات مفید و معنادار است. در گذشته، طراحی این فرایند توسط متخصصان این حوزه انجام می­شد و باید ویژگی‌ها را به صورت دستی تعریف و استخراج می‌کردیم ولی امروزه با کمک شبکه‌های عصبی و یادگیری عمیق، این عمل به صورت خودکار در لایه‌های کانولوشنی شبکه‌های عصبی انجام می­شود. از ویژگی‌ها در وظایفی مثل بخش‌بندی (segmentation) یا تشخیص اشیاء (object recognition) استفاده می­شود.

ویژگی‌های استخراج شده از تصاویر، در قالب مقادیر عددی نشان داده می­شوند و معمولا نسبت به تصویر اصلی، ابعاد بسیار کمتری دارند.

به صورت کلی، ویژگی‌های استخراج شده از تصاویر به دو دسته تقسیم می­شوند: سراسری و محلی

۱. ویژگی های سراسری

این ویژگی‌ها را می­توانیم از تصویر خام، در گام اول پردازش تصویر استخراج کنیم و معمولا نسبت به مواردی مانند تغییر زاویه نور پایدار نیستند. در بازیابی تصویر از این ویژگی‌ها برای «شناسایی اشیاء» ( بررسی حضور یا عدم حضور یک شی خاص در تصویر) استفاده می­شود. البته صرفا استفاده از این ویژگی‌ها برای دریافت محتوا و معنای یک تصویر کافی نیست.

۲. ویژگی های محلی

بعد از استخراج ویژگی‌های سراسری و شناسایی قسمتی از تصویر که احتمال حضور یک شی در اون وجود دارد، تشکیل کادرهای Region of Interest= RoI، با تقسیم آن بخش‌ها، به بلوک‌هایی کوچکتر، به بررسی جزئی‌تر ویژگی‌ها در هر یک از بلوک‌ها می‌پردازیم تا ماهیت آن شی تشخیص داده شود. استخراج این ویژگی‌ها، برای توصیف محتوای تصویر ضروری است.

نقش ویژگی‌ها در شناسایی و تشخیص اشیاء تصویر

در پردازش تصاویر، ابتدا ویژگی‌های سطح پایین (low-level features) موجود در تصویر مانند رنگ، بافت و شکل شناسایی می­شوند. پس از شناسایی بخشی از تصویر، که احتمالا یک شی در آن قرار دارد، تشخیص هویت آن شی انجام می‌شود تا نام آن مشخص شود. ویژگی‌های استخراج شده از تصاویر، می‌توانند سطح پایین یا سطح بالا باشند. ویژگی‌های سطح بالا(high-level features) به کمک ویژگی‌های سطح پایین کشف می­شوند.

در تشخیص اشیاء، هدف طبقه‌بندی اشیاء موجود در تصویر، در مجموعه‌ای از کلاس‌های از پیش تعیین شده است. هرکدام از کلاس‌ها، مجموعه‌ای از ویژگی‌های مختص به خود دارند که به طبقه‌بندی اشیاء متعلق به این کلاس‌ها کمک می­کند. برای مشخص کردن اینکه چه چیزهایی در یک تصویر و در کجای آن قرار دارند، از مدل‌های یادگیری ماشین استفاده می‌شود.

2-توضيحات تكنيكال

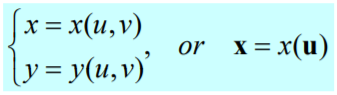
Estimate Geometry:

در ابتدا بايد مفهوم geometric transformation را درك كنيم.

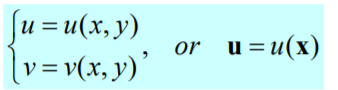
با geometric transformation ، موقعیت پیکسل های یک تصویر را اصلاح می کنیم ، اما رنگ آن­ها را بدون تغییر نگه می داریم.

اگر (u، v) مختصات تصویر را در تصویر اصلی و (x، y) را در یک تصویر تغییر شکل یافته (یا تاب خورده) نشان دهد.، برای ارتباط پیکسل های مربوطه در استفاده از یک جفت تابع استفاده می کنیم:

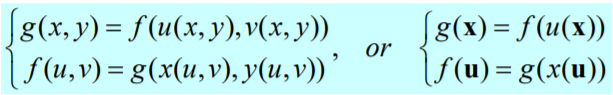
Forward mapping:



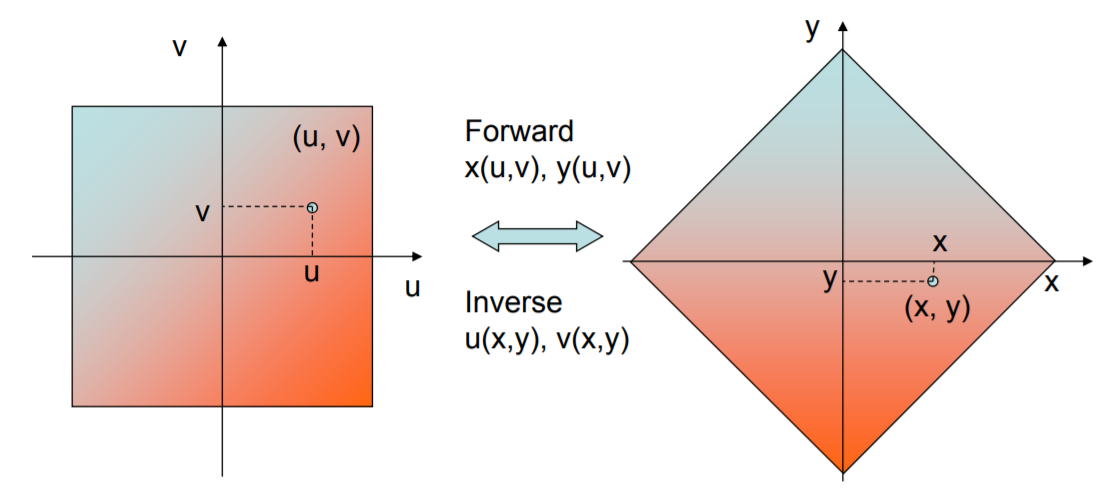
Inverse mapping:



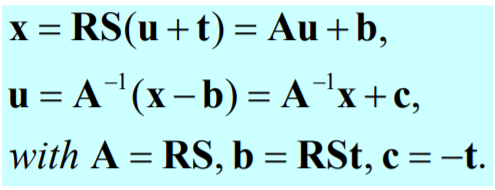
اگر f (u,v) یا f (u) نشانگر تصویر اصلی و g (x,y) یا g (x) تصویر تغییر شکل یافته باشد. آن­ها به اين صورت با هم مرتبط می شوند:



به عنوان مثال:



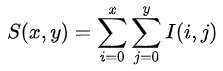
geometric transformation تركيبي از translation، scaling و rotation با فرم كلي زير است:



براي اينكه بتوانيم هندسه يك تصوير را تخمين بزنيم نياز داريم كه مقادير اين تابع را پيدا كنيم.

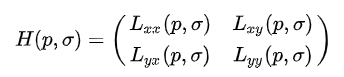
براي اين منظور از نقاط ويژگي استفاده مي­كنيم. ابتدا نقاط ويژگي را در هر دو تصوير پيدا مي­كنيم. براي اين منظور الگوريتم­هاي زيادي وجود دارد كه ما در اينجا از الگوريتم surf استفاده مي­كنيم.

SURF از فیلترهای مربع شکل به عنوان تقریب صاف کننده گاوسی استفاده می کند. (رویکرد SIFT از فیلترهای آبشار برای تشخیص نقاط مشخصه غیرقابل تغییر در مقیاس استفاده می کند ، جایی که اختلاف Gaussians (DoG) بر روی تصاویر تغییر یافته به صورت تدریجی محاسبه می شود.) اگر از تصویر انتگرال استفاده شود ، فیلتر کردن تصویر با یک مربع بسیار سریعتر است:



مجموع تصویر اصلی درون یک مستطیل را می توان با استفاده از تصویر انتگرال به سرعت ارزیابی کرد و نیاز به ارزیابی در چهار گوشه مستطیل دارد.

SURF از یک blob detector بر اساس ماتریس هسیان برای یافتن نقاط مورد علاقه استفاده می کند. تعیین کننده ماتریس هسیان به عنوان معیاری برای تغییر محلی در اطراف نقطه استفاده می شود و نقاطی انتخاب می شوند که این تعیین کننده حداکثر باشد.

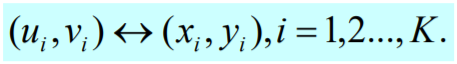


سپس اين نقاط را بايد ديگر match كنيم. با مقایسه توصیفگرهای بدست آمده از تصاویر مختلف ، می توان جفت های منطبق را پیدا کرد. در نهايت براي نقاطي كه با هم match شده اند يك تابع تبديل مطابق آنچه گفته شد مي­يابيم اين ماتريس­ها را در كنار هم قرار داده و ماتريس تبديل را به دست مي­آوريم.

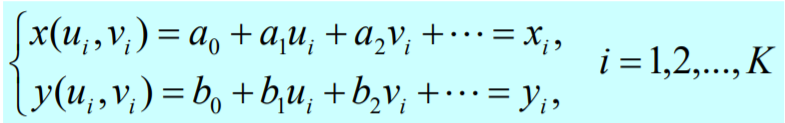
به كمك ماتريس تبديل حاصل مي­توانيم تصوير تغيير يافته را به تصوير اوليه مپ كنيم.

شرح مراحل:

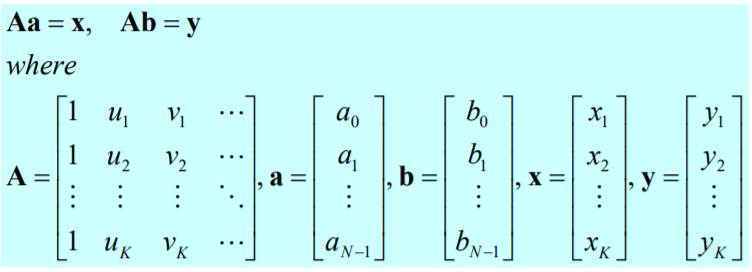
1. پيدا نقاط correspond بين دو تصوير K ≥N



1. پيدا كردن ضرايب ai ، bi ،... (i=0,..,N-1)



براي حل اين معادله معادله را به معادله ماتريسي تبديل مي­كنيم:



اگر k=N و ماتريس A يك ماتريس non-singular است:



اگر k>N از زوش least square استفاده مي­كنيم:



در غير اينصورت بايد نقاط correspond بيشتري پيدا كنيم!

در متلب از توابع detectSURFFeatures ، extractFeatures و matchFeatures استفاده كرده­ايم.

Corner Detection:

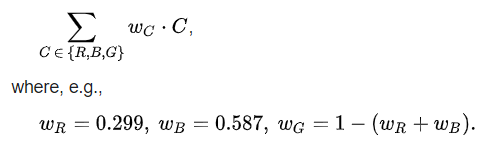
معمولاً الگوریتم آشکارساز گوشه هریس را می­توان به پنج مرحله تقسیم کرد:

1. Color to grayscale
2. Spatial derivative calculation
3. Structure tensor setup
4. Harris response calculation
5. Non-maximum suppression

Color to grayscale:

اگر از آشکارساز گوشه هریس در یک تصویر رنگی استفاده کنیم ، اولین قدم تبدیل آن به یک تصویر در مقیاس خاکستری است که باعث افزایش سرعت پردازش می شود.

مقدار پیکسل مقیاس خاکستری را می توان به عنوان مجموع­های وزن­دار از مقادیر R ، B و G تصویر رنگ محاسبه کرد :



Spatial derivative calculation:

بعد، می خواهیم Ix(x,y) و Iy(x,y) را محاسبه كنيم.

Structure tensor setup:

با Ix(x,y) و Iy(x,y) مي­توانيم structure tensor M را بسازيم.

Harris response calculation:

در این مرحله ، با استفاده تقریب زير ، کوچکترین مقدار ویژه تنسور سازه را محاسبه می کنیم:

براي x<<y :



با trace:

یکی دیگر از محاسبات معمول پاسخ هریس به شرح زیر است:



Non-maximum suppression:

به منظور انتخاب مقادیر بهینه برای نشان دادن گوشه ها ، local maxima را به عنوان گوشه های داخل پنجره پیدا می کنیم که یک فیلتر 3 در 3 است.

3-شکل‏ها، جدول‏ها و روابط (فرمول‏ها)

تصوير 1 ، attack 1

فيچرهاي match شده:



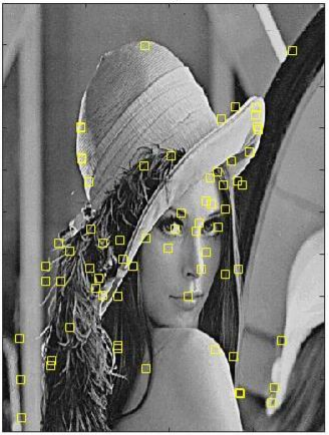


تصوير نتيجه:



تصوير2، attack1

فيچرهاي match شده:



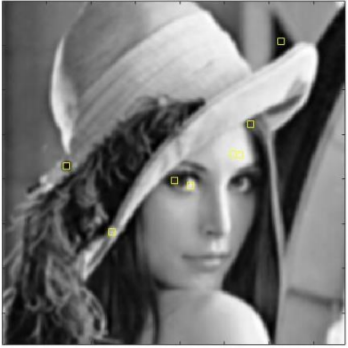


تصوير نتيجه:



تصوير3، attack1

فيچرهاي match شده:





تصوير نتيجه:



تصوير4، attack1

فيچرهاي match شده:





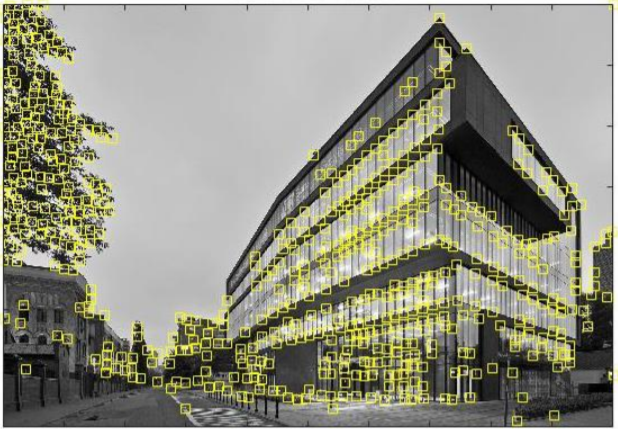
تصوير نتيجه:



نتايج:

|  |  |  |  |
| --- | --- | --- | --- |
| MP | MSE | SSIM |  |
| 14 | 17.2 | 0.9 | 1 |
| 68 | 617.3 | 0.6 | 2 |
| 14 | 403 | 0.6 | 3 |
| 28 | 63.1 | 0.9 | 4 |
| 31 | 275.15 | 0.75 | mean |
| 22.11 | 247.8 | 0.15 | std |

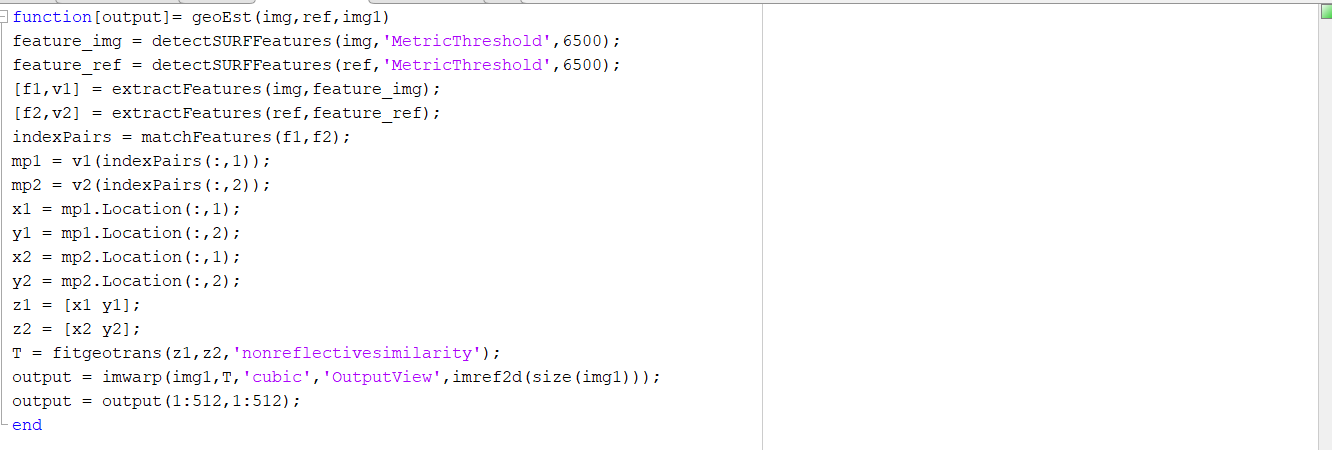
گوشه يابي:



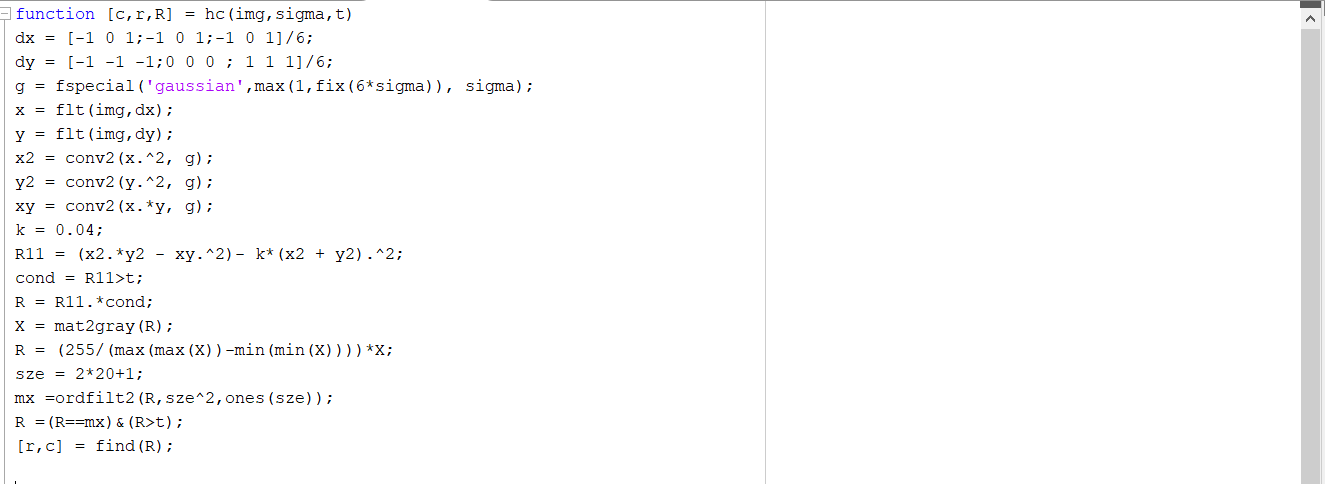
**4- کدها:**

تخمين هندسي:





گوشه ياب:





**مراجع**

* https://eeweb.engineering.nyu.edu/~yao/EL5123/lecture12\_ImageWarping.pdf
* https://virgool.io/@za.haghgu/%D9%88%DB%8C%DA%98%DA%AF%DB%8C%D9%87%D8%A7%DB%8C-%D8%AA%D8%B5%D9%88%DB%8C%D8%B1-ifwdtdf3s3pq
* https://en.wikipedia.org/wiki/Harris\_Corner\_Detector

1. [↑](#footnote-ref-1)