

دانشگاه صنعتی امیر کبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر

تمرین پنجم درس بینایی ماشین

دکتر صفابخش

غلامرضا دار ۴۰۰۱۳۱۰۱۸

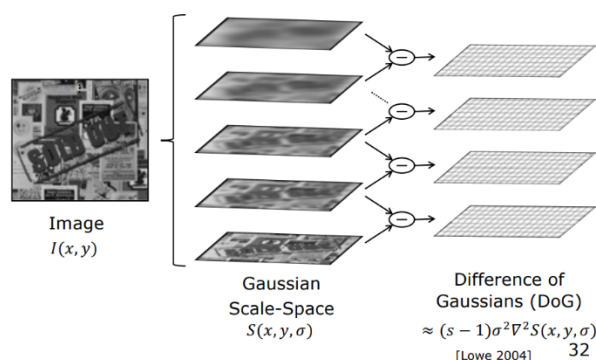
بهار ۱۴۰۱

فهرست مطالب

۳	(۱)
۶	(۲)
۱۲	(۳)

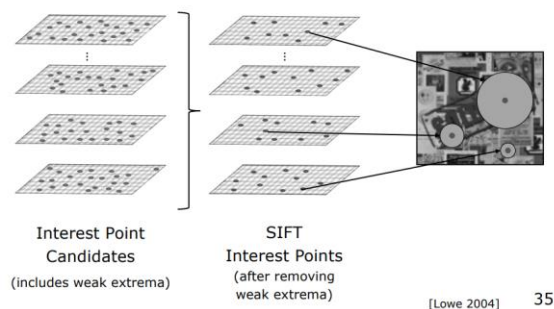
از مشکلات عمده‌ای که موقع استخراج ویژگی در تصویر به وجود می‌آید، استقلال این ویژگی‌ها از چرخش و مقیاس (اندازه) است. یکی از تکنیک‌هایی که در مقاله **SIFT** برای مقابله با این مشکلات استفاده شد، تشخیص ویژگی‌ها در مجموعه‌ای از تصاویر که به کمک فیلترهایی با اندازه مختلف، پردازش شده اند بود. تصاویری که بسیار Blur شده اند، ویژگی‌هایی با مقیاس کوچک را از دست داده اند و اکثراً شامل ویژگی‌های عمده (Coarse) هستند و تصاویری که کمتر Blur شده اند دارای ویژگی‌های جزئی (Fine) بیشتری هستند. در این مقاله به عنوان تخمین اعمال فیلتر Laplacian of Gaussian، از Difference of Gaussians استفاده شده. در تصویر زیر مشاهده می‌کنید که تصویر ورودی در مقیاس‌های مختلفی Blur شده است. تفاضل دو به دوی این تصاویر، DoG ها را می‌سازد.

Extracting SIFT Interest Points

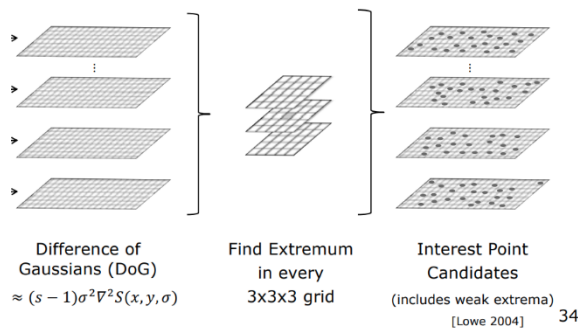


در ادامه، در این تصاویر، با کمک روش‌هایی مانند Non-Maximal Suppression به دنبال یافتن بیشینه‌های محلی می‌گردیم. این بیشینه‌های محلی، یک سه تایی (x,y,σ) هستند. این سه تایی‌ها مشخص می‌کنند، در کدام پیکسل از تصویر، و در کدام مقیاس، یک فیچر مناسب وجود دارد. پس از حذف ویژگی‌های ضعیف، تعدادی ویژگی داریم که نسبت به اندازه حساس نیستند.

Extracting SIFT Interest Points

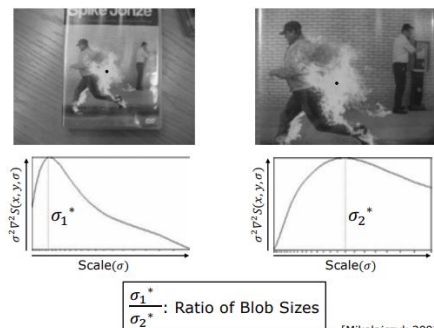


Extracting SIFT Interest Points



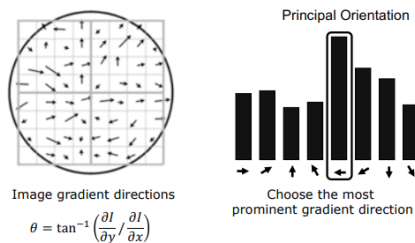
یکی دیگر از تکنیک‌هایی که برای استقلال از مقیاس مورد استفاده قرار گرفته، در تصویر زیر مشهود است. یک نقطه مشخص از کتاب زیر در دو تصویر با مقیاس‌های مختلف تشخیص داده شده است. به دلیل مقیاس مختلف دو تصویر، ریسپانس برای تصویر اول در مقیاس σ_1^* به مقدار بیشینه رسیده است و برای تصویر دوم در σ_2^* . چون این دو مقدار در مرحله تشخیص ویژگی مشخص می‌شوند، می‌توانیم با تقسیم کردن این دو مقدار، اثر مقیاس را در مرحله Matching خنثی کنیم.

SIFT Scale Invariance



Computing the Principal Orientation

Use the histogram of gradient directions



41

در نهایت برای استقلال از چرخش، از هیستوگرام گرادیان‌های موجود در ناحیه تشخیص داده شده به عنوان ویژگی استفاده می‌شود. پس از تشخیص ویژگی، در ناحیه‌ای که به کمک σ^* آن ویژگی مشخص می‌شود، هیستوگرام جهت‌های گرادیان‌های آن ناحیه محاسبه می‌شود و به عنوان ویژگی در یک بردار 128 بعدی قرار می‌گیرد. این تکنیک در مرحله Matching، استقلال از چرخش را به همراه دارد.

SIFT Rotation Invariance

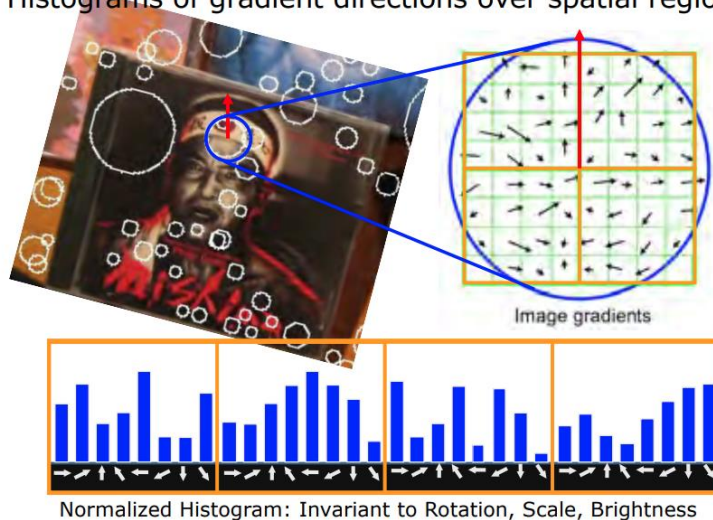
Use the principal orientation to undo rotation



برای استقلال از سطح روشنایی (مثال: شرایط نوری متفاوت هنگام تصویربرداری) هیستوگرام مقادیر پیکسل‌ها نیز در نظر گرفته می‌شود. در نهایت برای هر ویژگی تشخیص داده شده، مشابه تصویر زیر، یک بردار ۱۲۸ بعدی از ترکیب موارد ذکر شده ساخته می‌شود. این بردار descriptor ویژگی تشخیص داده شده در مرحله قبل است.

SIFT Descriptor

Histograms of gradient directions over spatial regions



45

مرحله آخر، Matching این ویژگی‌ها در دو تصویر است. چون در انتهای مرحله قبل به یک بردار ۱۲۸ بعدی از اعداد به ازای هر ویژگی رسیدیم، می‌توان با معیارهای فاصله مانند L2 Norm و ... میزان شباهت هر دو بردار را سنجید. اگر این شباهت از یک مقدار آستانه بیشتر بود، Match رخ داده است.

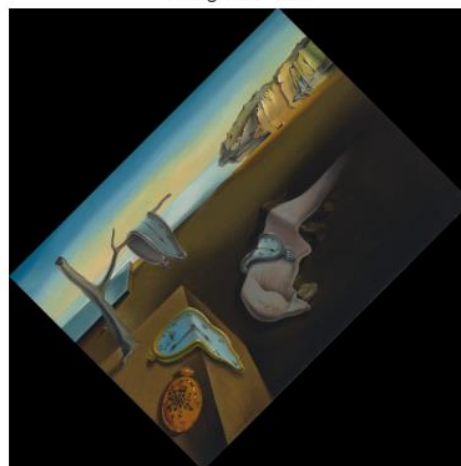
(۲)

در این بخش ابتدا طبق گفته سوال، تصویر ورودی را می چرخانیم و سپس با کمک SIFT سعی می کنیم تصویر چرخانده شده را به حالت اولیه برگردانیم. برای از دست رفتن پیکسل ها و در نتیجه بازگردانی ۱۰۰ درصد در ادامه سوال، تصاویر را قبل از چرخش مقداری Pad می دهیم.

Image Original

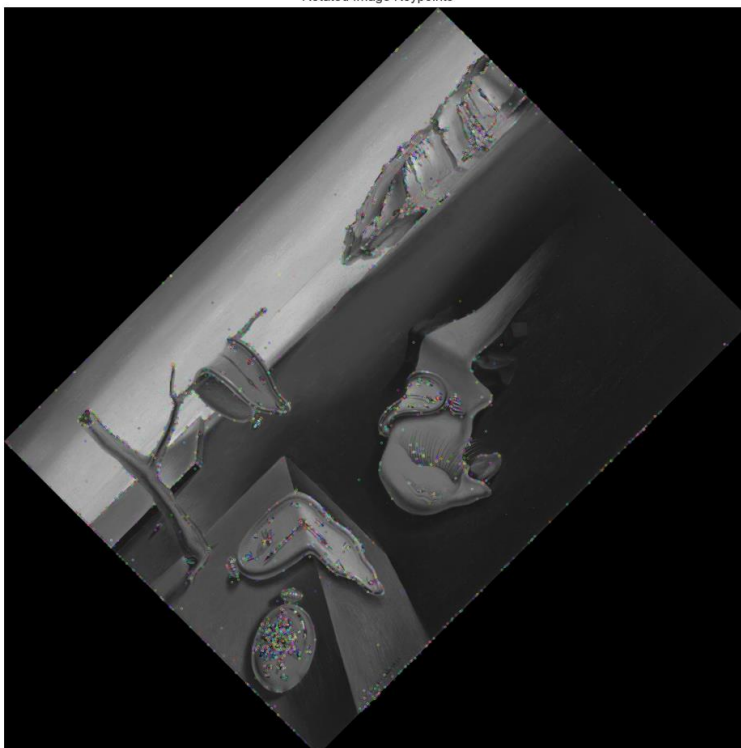


Image Rotated



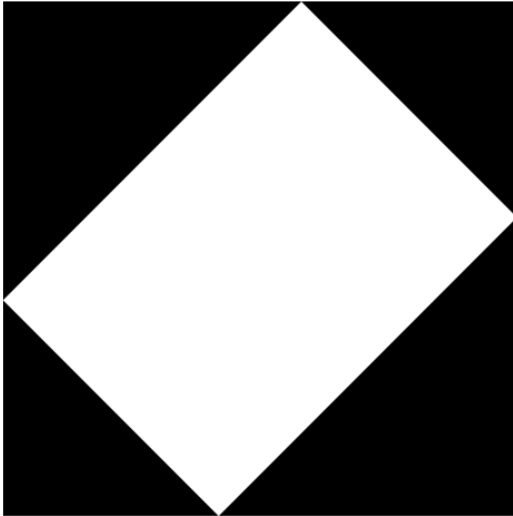
یکی از مشکلاتی که به وجود می آید انتخاب نقاط کلیدی در مرز تصویر است. این مرز لبه بسیار خوبی است ولی نباید به عنوان ویژگی در نظر گرفته شود زیرا صرفاً در اثر چرخش تصویر بوجود آمده است.

Rotated Image Keypoints

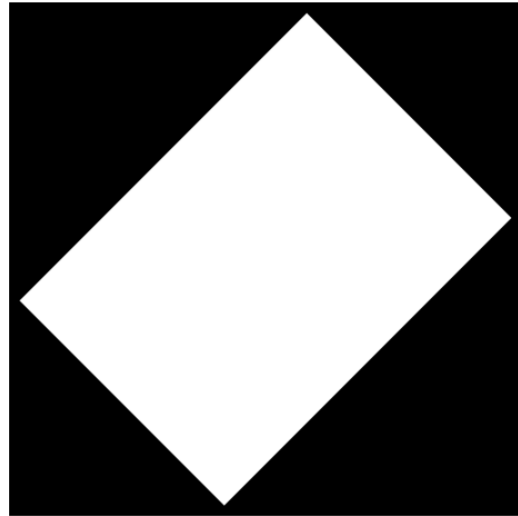


برای حل این مشکل از یک ماسک استفاده می کنیم.

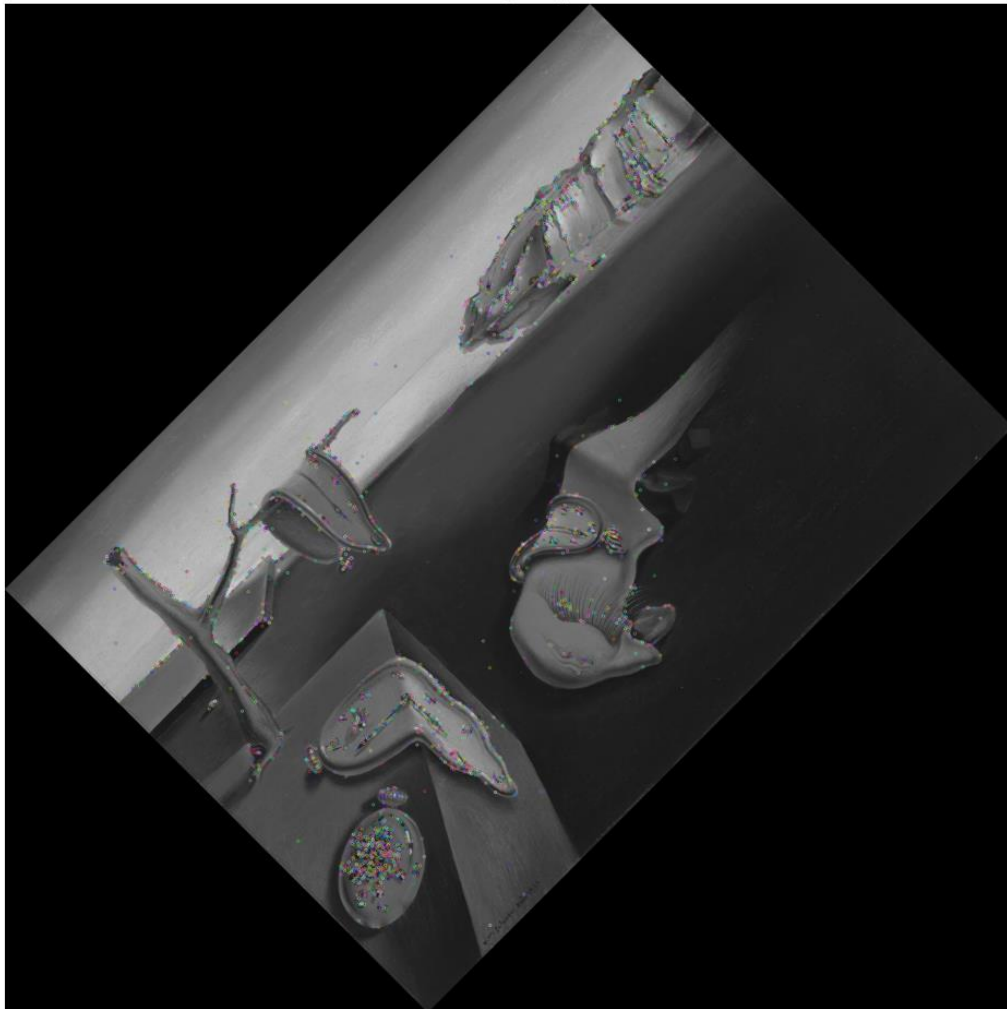
Rotated Mask



Eroded Mask

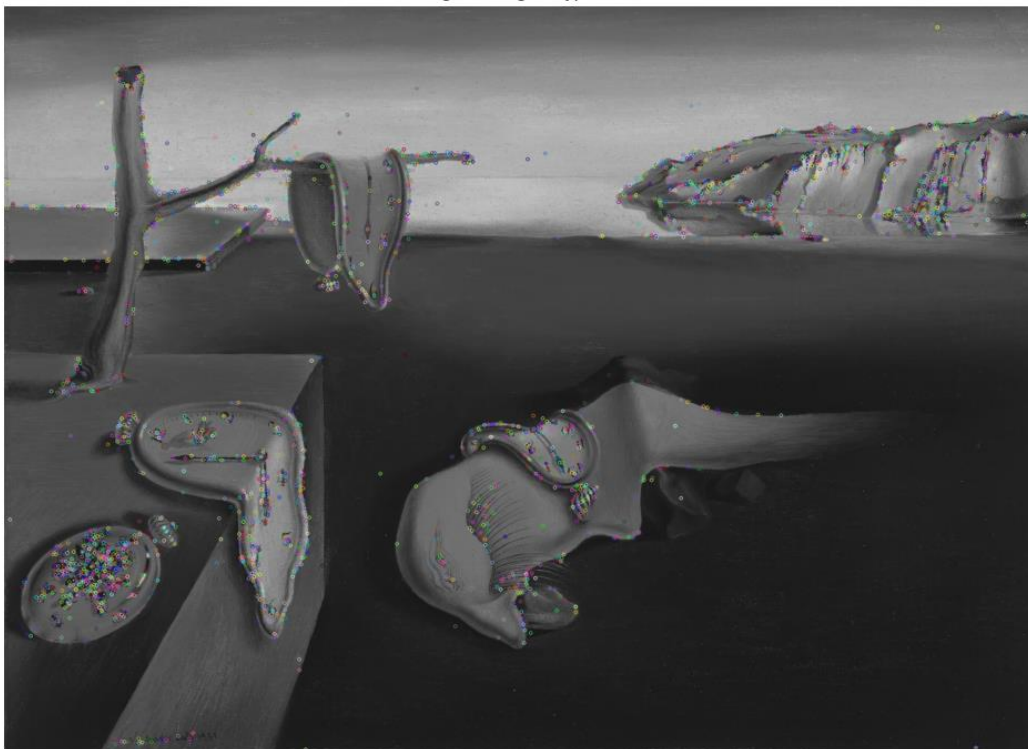


Rotated Image Keypoints



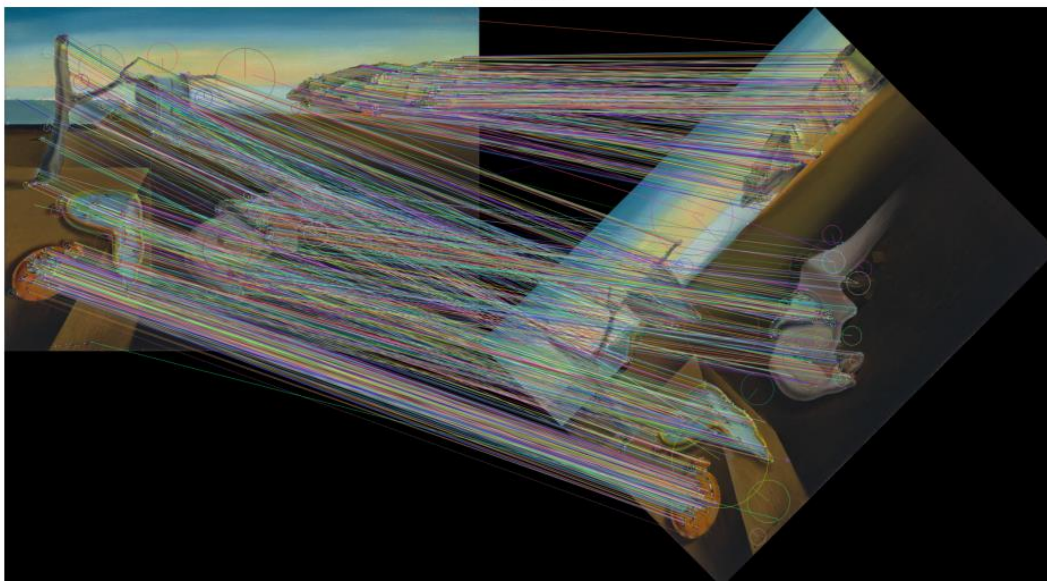
استخراج نقاط کلیدی برای تصویر اصلی

Original Image Keypoints



تطبیق ویژگی

Matches



محاسبه ماتریس دوران و اعمال Warp به تصویر چرخیده

Image Original



Image Rotated (30 degrees)



Image Original



Image Warped



همانطور که مشاهده می‌شود، تصویر پس از اجرای الگوریتم به تصویر اصلی بازگشت. زمان اجرای این الگوریتم حدود ۹۸۰ میلی ثانیه با در نظر گرفتن زمان لازم برای چرخش تصاویر اما منهای زمان لازم برای نمایش تصاویر است.

در صفحه بعد می‌توانید نتیجه الگوریتم را به ازای زوایای ۹۰ و ۱۶۵ نیز مشاهده کنید.

۹۰ درجه:

Matches

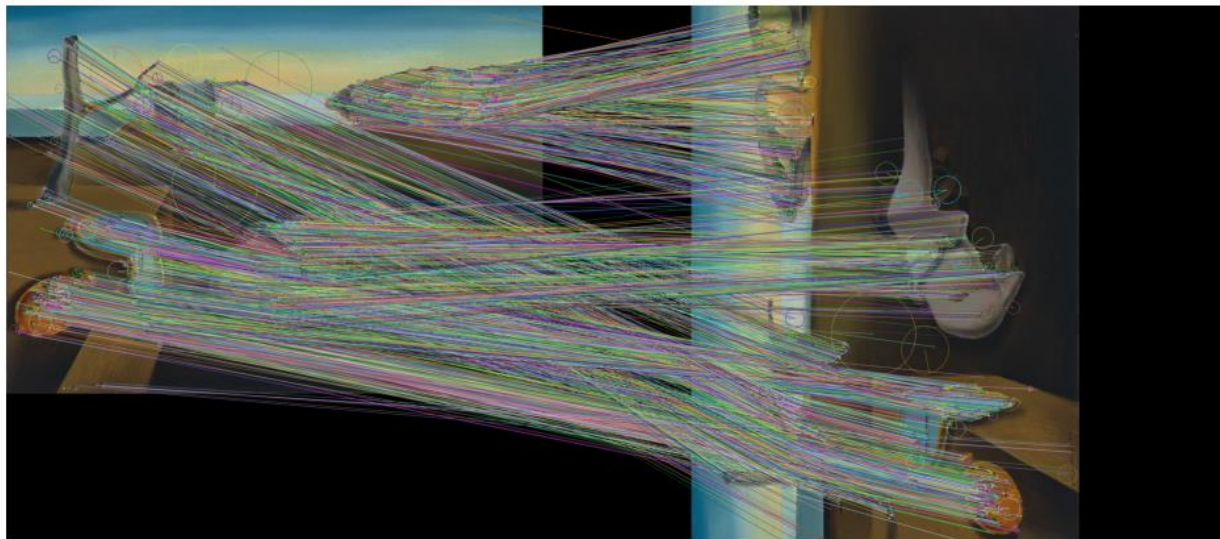


Image Original



Image Rotated (90 degrees)



Image Original



Image Warped



۱۶۵ درجه:

Matches

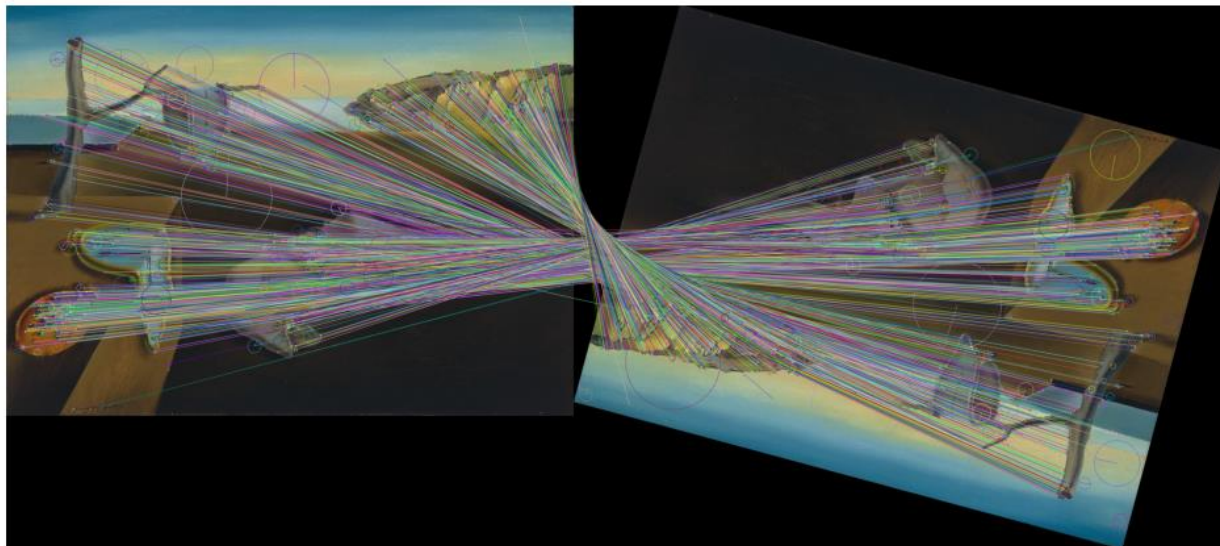


Image Original



Image Rotated (165 degrees)



Image Original



Image Warped



(۳)

نتایج اجرای الگوریتم **FREAK** بر روی تصویر را به ازای زوایای ۳۰، ۹۰ و ۱۶۵ درجه را در ادامه مشاهده می کنید.

۳۰ درجه:

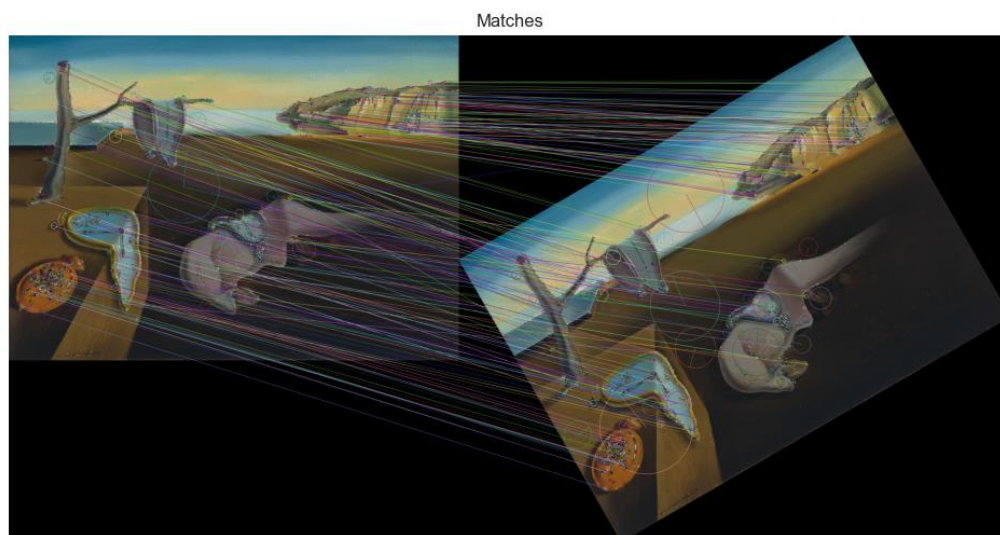


Image Original



Image Rotated (30 degrees)



Image Original



Image Warped



۹۰ درجه:

Matches

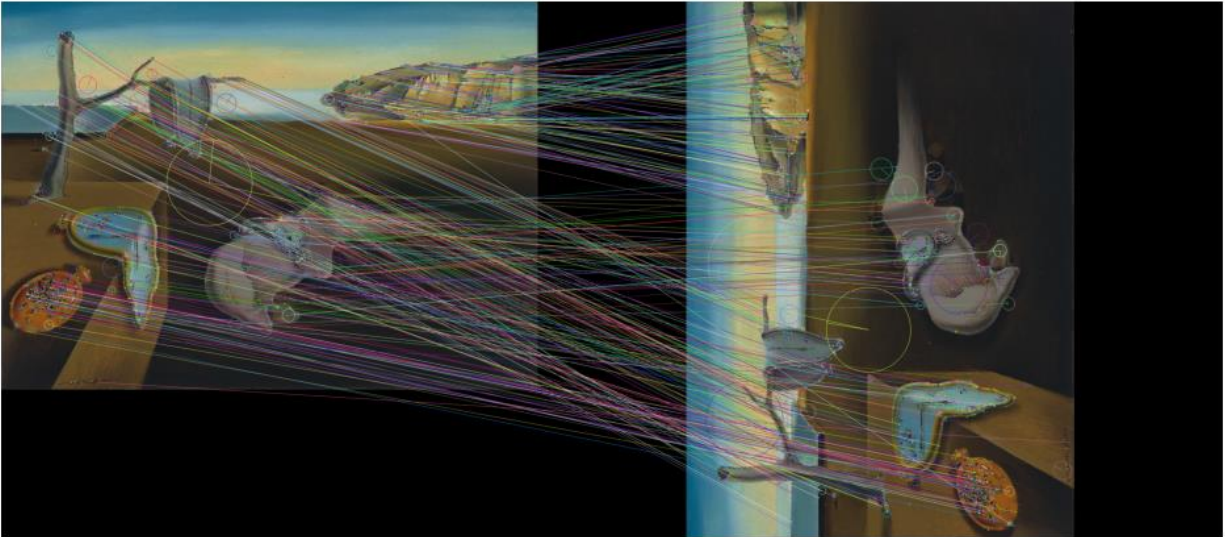


Image Original



Image Rotated (90 degrees)



Image Original



Image Warped



۱۶۵ درجه:

Matches

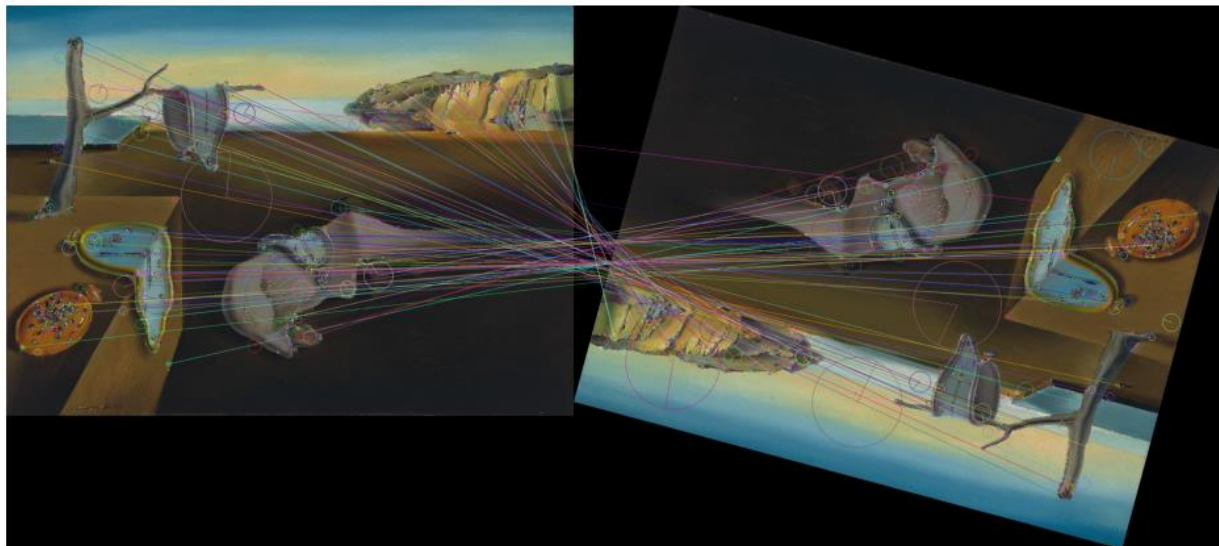


Image Original



Image Rotated (165 degrees)



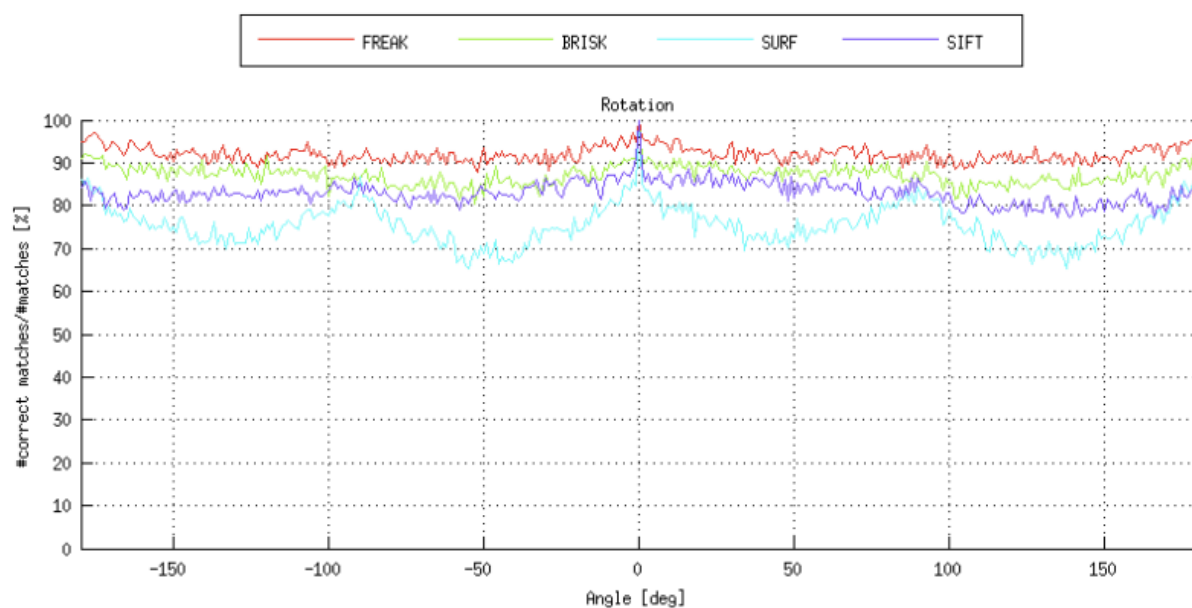
Image Original



Image Warped



طبق آزمایش‌های انجام شده هر دو روش در تشخیص ویژگی‌ها و بازگردانی تصویر چرخش یافته به حالت اولیه عملکرد عالی داشتند. اگر به مقاله FREAK نیز نگاه کنیم، این نتیجه مشاهده می‌شود.



برای مقایسه زمان اجرای این دو الگوریتم، زمان اجرای مربوط به بخش‌های مختلف الگوریتم را خروجی گرفتیم.

(۱۰۷۰ میلی ثانیه برای روش FREAK در مقابل ۱۸۵۰ میلی ثانیه برای روش SIFT)

```
> 0.64 ms : Grayscale conversion
> 12.05 ms : Rotation
> 1.44 ms : Grayscale conversion
> 5.70 ms : Mask Creation
> 5.61 ms : Mask Erosion
> 316.95 ms : Keypoint 1 @(1440, 1992) Extraction
> 657.79 ms : Keypoint 2 @(2243, 2445) Extraction
>> kp1: 2707
>> kp2: 2729
> 51.90 ms : Keypoint Descriptor Extraction
> 11.69 ms : Keypoint Matching
> 0.25 ms : Ratio Test
> 1.09 ms : Homography Estimation
> 4.19 ms : Image Warping
CPU times: total: 2.86 s
Wall time: 1.07 s
```

```
> 0.64 ms : Grayscale conversion
> 9.89 ms : Rotation
> 1.18 ms : Grayscale conversion
> 5.48 ms : Mask Creation
> 5.67 ms : Mask Erosion
> 325.42 ms : Keypoint 1 @(1440, 1992) Extraction
> 645.63 ms : Keypoint 2 @(2243, 2445) Extraction
>> kp1: 2707
>> kp2: 2729
> 823.89 ms : Keypoint Descriptor Extraction
> 17.36 ms : Keypoint Matching
> 0.28 ms : Ratio Test
> 1.79 ms : Homography Estimation
> 4.52 ms : Image Warping
CPU times: total: 3.5 s
Wall time: 1.85 s
```

تصویر سمت راست: زمان اجرای الگوریتم SIFT، تصویر سمت چپ: زمان اجرای الگوریتم FREAK

همچنین به دلیل اینکه الگوریتم FREAK، مرحله Keypoint Extraction را دارا نیست، برای هر دو آزمایش از بخش استخراج نقاط کلیدی SIFT استفاده شد که یکی از قسمت‌های پرهزینه کل روش است. در ادامه این بخش را با تکنیک استخراج نقطه کلیدی الگوریتم FAST جایگزین کردیم (برای هر دو روش) و نتایج به شدت متفاوتی به دست آمد.

(۹۵ میلی ثانیه برای روش FREAK در مقابل ۲۳۸ میلی ثانیه برای روش SIFT)

<pre> > 0.56 ms : Grayscale conversion > 10.84 ms : Rotation > 1.00 ms : Grayscale conversion > 5.34 ms : Mask Creation > 6.06 ms : Mask Erosion > 1.39 ms : Keypoint 1 @(1440, 1992) Extraction > 1.88 ms : Keypoint 2 @(2243, 2445) Extraction >> kp1: 3128 >> kp2: 2035 > 49.38 ms : Keypoint Descriptor Extraction > 10.05 ms : Keypoint Matching > 0.27 ms : Ratio Test > 1.02 ms : Homography Estimation > 4.95 ms : Image Warping CPU times: total: 531 ms Wall time: 95.7 ms </pre>	<pre> > 0.61 ms : Grayscale conversion > 10.25 ms : Rotation > 1.27 ms : Grayscale conversion > 5.53 ms : Mask Creation > 5.66 ms : Mask Erosion > 1.39 ms : Keypoint 1 @(1440, 1992) Extraction > 2.01 ms : Keypoint 2 @(2243, 2445) Extraction >> kp1: 3128 >> kp2: 2035 > 183.43 ms : Keypoint Descriptor Extraction > 16.03 ms : Keypoint Matching > 0.21 ms : Ratio Test > 1.18 ms : Homography Estimation > 5.22 ms : Image Warping CPU times: total: 1.08 s Wall time: 238 ms </pre>
---	---

تصویر سمت راست: زمان اجرای الگوریتم SIFT، تصویر سمت چپ: زمان اجرای الگوریتم FREAK

همان‌طور که مشاهده می‌شود، بخش Keypoint Description در روش FREAK بیش از ۳ برابر سریع‌تر عمل کرده و بخش Keypoint Matching به دلیل استفاده از فاصله Hamming نیز اندکی تسریع داشته است.

منابع:

[Introduction to SIFT \(Scale-Invariant Feature Transform\)](#)

[A. Alahi, R. Ortiz and P. Vandergheynst, "FREAK: Fast Retina Keypoint," 2012 IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 510-517, doi: 10.1109/CVPR.2012.6247715.](#)

<https://fpcv.cs.columbia.edu/Monographs>

https://docs.opencv.org/4.x/df/d74/classcv_1_1FastFeatureDetector.html