

توضیح تکلیف سوم

آریا ادیبی ۹۲۱۱۰۴۷۶

نحوه‌ی قرارگیری فایل‌ها

فایل‌های مربوط به هر پرسش در پوشه‌ی با شماره‌ی همان پرسش قرار گرفته‌اند. فرض شده است که در هر کدام از این پوشه‌ها تصویرهای مورد نیاز اولیه قرار داده شده‌اند.

پرسش ۱:

این ۲ تصویر نتیجه‌ی این کد است.



(a) Frequency Domain



(b) Spatial Domain

انتخاب عکس:

همان طور که مقاله نیز اشاره شده بود، من چند نکته را راجع به انتخاب عکس رعایت کردم. نکات از این قراراند:

- عکسی که باید از دور دیده شود خود مدلی هموار مانند دارد که وقتی آن را در فرکانس پایین در نظر می‌گیریم جزئیات زیادی حذف شده به نظر نیاید.
- عکسی که باید از نزدیک دیده شود شامل خط‌های زیادی است که باعث می‌شود که وقتی در فرکانس بالا آن را در نظر می‌گیریم نتیجه‌ی خوبی به دست آید و اصل شکلی‌تر نمایان شود.
- رنگ هر دو شکل در تضاد یک دیگر نبوده و باعث می‌شود بهتر با هم ادغام گردند.
- صورت کلی ۲ شکل تقریباً یک چیز است و باعث می‌شود که ادغام بهتری به دست آید.

توضیح کد:

کد بخش بخش شده است که هر بخش را جداگانه توضیح می‌دهم.

۱. **inputs and initializations**:

در این بخش یک سری مقدمات برای اجرا انجام می‌شود و همچنین ۲ عکس نیز خوانده می‌شود.

۲. **align**:

در این بخش دو تصویر به طور مناسبی روی هم قرار گرفته می‌شوند و نتیجه نیز نشان داده می‌شود. قرار دادن این ۲ تصویر توسط تابع `align_2im` صورت می‌گیرد که نحوه کار آن از این قرار است.

از کاربر درخواست می‌شود که ۲ نقطه روی تصویر اول داده شده انتخاب کند سپس از او درخواست می‌شود که در تصویر دوم متناظر آن ۲ نقطه را نیز انتخاب کند. این تابع این ۲ تصویر را طوری جابه‌جا، تغییر اندازه و دوران می‌دهد که این نقاط متناظر روی هم بیافتند.

این کار اینگونه انجام می‌شود که ابتدا مرکز ۲ تصویر را با اضافه کردن پیکسل ۰ به اول یا آخر تصویرها روی هم قرار می‌دهد. سپس با تغییر اندازه دادن عکسی باعث می‌شود که فاصله‌ی ۲ نقطه در هر ۲ عکس برابر شود. بعد از آن عکس اول را دوران داده به طوری که این ۲ خط متناظر موازی شوند. در آخر نیز با بریدن بخشی از عکس‌ها باعث می‌شود که این ۲ خط متناظر روی هم بیافتند.

۳. **find and show DFTs**:

در این قسمت این ۲ عکس به فضای فوریه می‌رود (با جابه‌جا کردن، فرکانس‌های کم را در مرکز قرار می‌دهم) و سپس نتایج نشان داده می‌شود. برای نمایش نتایج از تابع `log` استفاده شده تا تصویر روشن‌تر شده و تصویر قابل فهم‌تری (اثرات را بتوان راحت دید) از نتیجه را بتوان دید.

۴. **find and show high pass and low pass filtered images**:

در این قسمت دقیقاً همانند توضیح پرسش عمل شده است. ۲ فیلتر گوسی یکی برای فرکانس‌های کم و دیگری برای فرکانس‌های زیاد و با در نظر گرفتن `cutoff` به دست می‌آید. این فیلترها در لایه‌های تصویرهای مورد نظر طبق قضیه موجود در درس درایه به درایه در هم ضرب می‌شوند و نتیجه‌ی هر یک به دست می‌آید.

همچنین در اینجا فیلترها نیز نشان داده می‌شوند.

۵. **create the hybrid image in frequency domain**:

در این جا به سادگی با + این ۲ تصویر با هم ترکیب شده و پس از بریدن بخشی از آن نمایش داده می‌شود.

۶. **create the hybrid image in spatial domain**:

استفاده از دامنه‌ی فرکانس‌ها می‌سازیم. نحوه‌ی ساخت مشابه آن چیزی است که در صورت پرسش ذکر شده است. تنها در آخر برای بهتر شدن نتیجه هر یک از این ۲ عکس حاصل را در عددی ضرب و سپس جمع می‌کنم و با بریدن بخشی از آن، تصویر نهایی را نمایش می‌دهم.

پرسش ۲:

نتیجه‌ی به دست آمده در این پرسش در شکل زیر نمایان است.

(من کدی نوشته بودم که نوع دگرگونی را نیز تشخیص می‌داد ولی متأسفانه به دلیل نامعلومی درست کار نمی‌کرد و از آن جایی که در پرسش این موضوع اجباری نبود آن را حذف کردم.)

در این کد از \ در Matlab استفاده شده است. اگر $X = A \setminus B$ را در Matlab بنویسیم X ماتریسی خواهد بود که اگر در A ضرب شود B حاصل گردد. این دقیقاً همان چیزی است که نیاز داشتیم.

دقت کنید که با توجه به عکس‌ها، همه‌ی آن‌ها توازی خط‌ها را جز عکس چهارم رعایت کرده‌اند. از این رو تمامی عکس‌ها جز عکس چهارم از نوع Affine هستند و عکس چهارم از نوع Projective است. از این روی برای ۳ عکس اول ۳ نقطه و برای عکس چهارم ۴ نقطه کافی خواهد بود.

نقاط را همانند ماتریسی که در درس توضیح داده شده وارد ماتریس مکنیم و با عمل \ ماتریس تبدیل را به دست می‌آوریم. تمامی ماتریس‌های به دست آمده چک می‌شوند و با اثر دادن در تصویر اصلی تصویر حاصل نشان داده می‌شود. همانطور که در شکل نیز مشخص است نوع هر تبدیل ذکر شده است. به دست آوردن نوع تبدیل کار بسیار آسانی

است. در تبدیل‌های Affine سطر آخر باید 0 0 1 باشد که همان طور که قبلاً نیز حدس زده شده بود جز ماتریس آخر بقیه این خاصیت را دارند. برای این که بفهمیم ماتریس Simi-larity است یا نه کافیه با $M(2, 1)/M(1, 1)$ ، $\tan(\theta)$ را به دست آوریم (دقت کنید که فرض می‌کنیم Similarity است اگر نباشد در آخر متوجه می‌شویم). سپس با آن

```
T1:
Similarity:
    1.8665    -0.1779    601.0839
    0.7173     0.4645    10.4807
         0         0         1.0000

T2:
Affine:
    1.0e+03 *
    0.0009    -0.0004    1.1349
    0.0014     0.0006   -0.0198
         0         0     0.0010

T3:
Affine:
    1.0002     0.1988     4.3868
    0.3033     0.9989     3.2905
         0         0     1.0000

T4:
Projective:
    1.0027     0.2000   -0.0618
    0.3014     1.0024   -2.8503
    0.0001     0.0002     1.0000

fx >>
```

شکل ۲

$\sin(\theta)$ و $\cos(\theta)$ را حساب می‌کنیم که به کمک آن‌ها a ، b به دست می‌آید اگر در به دست آوردن a و b دیدیم که خاصیت مورد نظر رعایت نشده است پس ماتریس ما Similarity نیست در غیر این صورت است چرا که متغیرهای آن را به دست آورده‌ایم (دقت کنید که t_x و t_y خاصیت خاصی ندارند و تأیید کننده تشخیص ما نیستند). با عملیات ساده‌ی ریاضی مشابه می‌توان بقیه تبدیل‌ها را نیز تشخیص داد. دقت کنید که ماتریس سوم به تبدیل Shear بسیار نزدیک است. تنها مشکل t_x و t_y است که ناصفراند. در واقع آن را می‌توان حاصل یک Shear و یک Translation در نظر گرفت.