



بسمه تعالی

دانشگاه صنعتی شریف

دانشکده علوم کامپیوتر

Image processing – دکتر مصطفی کمالی

محمد حسین مسلمی – 97102463

HW2 report

### سوال سه – Homography and Image Warping

در ابتدا مختصات چهار گوشه از سه کتاب را به صورت دستی پیدا میکنیم ، روش های مختلفی برای اینکار وجود دارد یک روش استفاده از `cv2.EVENT_LBUTTONDOWN` است که هرگاه موس کلیک کرد تابعی مختصات کلیک شده روی عکس را نشان دهد ، ولی این روش مقداری خطای انسانی میتواند داشته باشد که با دقت خوبی گوشه را نتوان کلیک کرد ، برای این کار من از `paint` استفاده کردم که عکس را با `paint` باز کرده و در گوشه هر کتاب تا جای ممکن زوم کرده و مختصات آن گوشه را یادداشت کردم که حاصل برای هر سه کتاب در سه آرایه `pnt1,pnt2,pnt3` قرار دارد .

ترتیب قرار گیری نقاط در هر یک از `pnt` ها به اینصورت است :

[down\_left , down\_right , up\_right , up\_left]

الزاما نباید این ترتیب را رعایت کرد ولی باید همین ترتیب را در قرار دهی گوشه های مقصد هم رعایت کرد . حال یک تابع `apply_homography` داریم که با ورودی گرفتن عکس اصلی و نقاط گوشه های مورد نظر به ما ماتریس هوموگرافی متناظر و عکس `warp` شده متناظر را میدهد حال این تابع را توصیف میکنیم.

در این تابع ابتدا طول و عرض عکس نهایی را بدست میآوریم ، به اینصورت که میانگین فاصله دو نقطه پایینی و بالایی میشود عرض و میانگین فاصله دو نقطه کناری میشود طول. البته این دو فاصله برای دو دسته نقطه دو طرف تقریبا یکی میشود ولی با میانگین گیری مقداری جزئی دقت بهتر میشود .

حال میخواهیم نقاط مبدا را که در `pnt` است به مقصد ببریم به اینصورت که گوشه بالا چپ در `(0,0)` قرار میگیرد و گوشه پایین راست در `(w,h)` قرار میگیرد ، با این اطلاعات و با استفاده از تابع `cv2.findHomography` ماتریس هوموگرافی بدست میآید ، سپس از این ماتریس معکوس میگیریم که در ادامه میگوییم فایده آن چیست . حال به جای اینکه عکس اصلی را به عکس نهایی ببریم ، برای هر نقطه از عکس خروجی که در ابتدا خالی است بررسی میکنیم که مقدار آن از کدام

نقطه از عکس اصلی باید بیاید ، برای اینکار معکوس هوموگرافی را در سمت چپ مختصات همگن نقطه تصویر خروجی ضرب میکنیم و ماتریس حاصل که سه درایه دارد را همگن میکنیم ، یعنی درایه هایش را تقسیم بر درایه سوم میکنیم ، که حاصل میشود مختصات نقطه مورد نظر در تصویر اصلی ، سپس مقدار پیکسل در عکس خروجی را مساوی با مقدار پیکسل در تصویر اصلی که مختصاتش بدست آمد قرار میدهیم و این کار را برای همه نقاط تصویر خروجی انجام میدهیم.

در نهایت حاصل را ذخیره میکنیم و مقدار ماتریس هوموگرافی برای سه کتاب *graph theory* و *DFT* و *image science* به ترتیب به صورت زیر است :

Homography for book1 is :

```
[[ -3.51972175e-01  9.51001742e-01  3.56541045e+01]
 [ -9.51243321e-01 -3.34220626e-01  7.03380162e+02]
 [ -7.54217427e-06  6.11404243e-05  1]]
```

Homography for book2 is :

```
[[ -9.61270197e-01 -1.66101836e-01  4.72022542e+02]
 [  1.43496744e-01 -9.41697385e-01  6.45708444e+02]
 [ -1.80011164e-04  1.99191273e-05  1]]
```

Homography for book3 is :

```
[[ -8.87915696e-01  5.63428232e-01  1.76476932e+02]
 [ -5.64115000e-01 -8.71814092e-01  1.30254154e+03]
 [  1.45402478e-06  5.39189843e-05  1]]
```