

# توضیح تکلیف اول

آریا ادیبی ۹۲۱۱۰۴۷۶

## نحوه‌ی قرارگیری فایل‌ها

فایل‌های مربوط به پرسش اول در پوشه‌ی ۱ و پرسش دوم در پوشه‌ی ۲ قرار داده شده‌اند.  
در پوشه‌ی ۱:

- یک پوشه به نام Images قرار داده شده است که عکس‌های اصلی در آن قرار دارند.
  - در پوشه‌های ۱ تا ۳ کد و تصویر اصلاح شده‌ی عکس‌ها به ترتیب قرار دارد.
- در پوشه‌ی ۲:  
کدها و تصویرها در این پوشه قرار دارند.

**نکته:** نحوه‌ی کار تابع‌های مورد استفاده و درستی آن‌ها در انتهای این توضیحات خواهد بود؛  
مگر این که مخصوص آن پرسش بوده که در همان پرسش توضیح داده خواهد شد.

## پرسش ۱:

### تصویر ۱:

ابتدا کل قسمت‌های تیره‌ی تصویر را با استفاده از تابع لگاریتمی<sup>۱</sup> روشن می‌کنم. سپس نقاطی در صفحه‌ی ۲ بعدی را در نظر می‌گیرم که در آن نقطه‌ها شدت رنگ برای هر ۳ رنگ RGB کمتر از ۲۵ باشد. روی هر ۳ رنگ در این نقطه‌ها تابع توانی<sup>۲</sup> با توان کمتر از ۱ (برای روشن‌سازی بیشتر) را اجرا می‌کنم.

کار روشن‌سازی به اتمام رسید. برای این که نتیجه بهتر شود از ۲ فیلتر زیر استفاده می‌کنیم. ابتدا فیلتر wiener2 را روی هر رنگ اجرا می‌کنیم که برخی از noise‌ها را بگیرد (در مورد این فیلتر در انتها صحبت می‌کنم). تصویر حاصل هنوز حالت زبری را دارد، علاوه بر آن noise‌های رنگی نیز مشاهده می‌شود. برای برطرف کردن این موضوع تصویر را با در نظر گرفتن هر ۳ بعد به صورت همزمان بلور می‌کنیم که برای ضریب دهی از تابع گوس استفاده می‌کنیم. به این ترتیب به تصویر نهایی می‌رسیم.

---

<sup>۱</sup>Log Transformation

<sup>۲</sup>Power-Law Transformation

## تصویر ۲:

ایده‌ی اصلی این تصویر کاملاً مشابه تصویر ۱ است بنابر این در جا تنها به نکات خاص این تصویر توجه می‌کنیم.

اگر به صورت ساده تصویر را روشن کنیم مشکلاتی به وجود می‌آید که با تغییر میزان روشنایی حل نمی‌شود. ۳ مشکل این چینی از همه بیشتر قابل دیدن است، این ۳ مشکل از این قراراند:

۱. در گوشه‌ی بالا سمت راست تصویر، در دیوار آجری سوراخ‌هایی وجود دارد که تا میزان روشنایی را بالا نبریم به طرز ناپسندی تیره دیده می‌شوند.

۲. اگر کمی روشنایی را افزایش دهیم شاهد پدید آمدن مرزهای سفید اطراف برگ‌های درخت‌های جلو خواهیم شد که تصویر را خیلی غیر واقعی می‌کند. از طرفی درخت‌های جلو هنوز بسیار تیره هستند.

۳. در گوشه‌ی پایین سمت چپ، درخت‌هایی وجود دارند که به قسمت بالایی آن‌ها آفتاب خورده و روشن‌اند ولی قسمت پایینی آن‌ها تیره است. اگر قسمت‌های تیره را روشن کنیم بدون در نظر گرفتن این قسمت‌های روشن تصویر بد می‌شود. این اثر در بازتاب نور از شیشه‌ی این قسمت هم مشهود است.

برای حل این مشکلات من راه‌کار زیر را پیش گرفتیم: ۳ تا mask درست می‌کنم که بتوانم گوشه‌ی بالا سمت راست، گوشه‌ی پایین سمت چپ و بقیه‌ی تصویر را جداگانه دستکاری کنم. برای سادگی نام این ۳ قسمت را به ترتیب قسمت‌های ۱ تا ۳ بگذارید.

قسمت ۱ را روشن می‌کنم. قسمت ۳ را به همان میزان روشن می‌کنم اما برای حل مشکل اشاره شده به جای این که تمام قسمت‌های تاریک را بگیرم آن‌هایی را می‌گیرم که خاصیت زیر را دارند. اگر آن pixel وسط یک یک ماتریس  $7 \times 7$  در نظر بگیریم، شمار قابل توجهی از همسایه‌هایش تاریک باشند. روشن است که با این روش مشکل گفته شده دیگر ایجاد نمی‌شود.

قبل از روشن کردن قسمت ۲م این قسمت را با ضرایب گوسی بلور می‌کنم. دو قسمت قبل را با تابع لگاریتمی روشن کرم، این قسمت را با روشنایی کمتر و با استفاده از تابع توانی روشن می‌کنم چرا که رفتار آن ملایم‌تر است.

این‌گونه این تصویر ساخته می‌شود. البته قسمت درخت‌های قسمت پایین چپ با این که خیلی بهتر شده به خاطر بلور کردن و کمتر روشن کردن ولی هنوز می‌توانست بهتر شود. ولی به دلیل وقت کم این کار را انجام ندادم.

## تصویر ۳:

در این قسمت ایده‌ی جدیدی نسبت به ۲ تصویر قبل انجام ندادم. به طور خلاصه کل قسمت‌های تیره را روشن کردم، قسمت‌های تیره‌تر را بیشتر روشن کردم.

۲ فیلتر wiener2 و بلور با ضرایب گوسی را برای نتیجه‌ی بهتر اعمال کردم.

## پرسش ۲:

### تصویر ۱:

در ابتدا تصویر بریده شده تا کادر سفید (و در مرز سیاه) آن حذف شود، تا در ارزیابی ما مشکل ایجاد نکند. سپس با به ۳ قسمت مساوی تقسیم کردن تصویر ۳ رنگ RGB را به دست می آوریم. حال همان طور که در صورت پرسش توضیح داده شده جست و جوی کامل<sup>۳</sup> در جابهجایی به اندازه ی بیشینه ۲۰ pixel را انجام می دهیم و بهترین نتیجه را ابتدا برای R و G به دست می آوریم. سپس با همین روش بهترین جایگاه برای B را با نظر گرفتن هر دوی R و G را نیز به دست می آوریم. قسمت های مشترک این ۳ را می گیریم و بقیه را کنار می اندازیم. حاصل عکس ما خواهد بود. برای مقایسه نتایج از Sum of squared differences استفاده می کنیم با این تفاوت که تقسیم بر شمار pixel های شامل هم می کنیم که معیار بهتری برای زمانی است که سطح اشتراک متفاوت می شود. در انتها نیز اگر بهتر باشد کمی دیگر از تصویر را می برم.

### تصویر ۲:

در این جا دقیقاً همانند تصویر ۱ عمل می کنیم. تنها تفاوت اینجاست که مشکل رنگ اطراف تصویر بعد از الگوریتم با یک بریدن ساده به طور کامل حل نمی شود. ۲ قسمت بسیار بنفش در ۲ قسمت راست و چپ تصویر قرار می گیر که خیلی داخل آمده. برای حل این مشکل با استفاده از طراحی یک mask تنها دو نوار نازک از راست و چپ را می گیریم. چون در این قسمت تقریباً همه چیز سفید است جز همین ۲ قسمت بنفش به مقدار سبز هر جا سبزش کم است اضافه می کنیم (که بنفش را به سفید می برد). سپس یک فیلتر میانه هم می زنم. این گونه قسمت های زیادی از بنفش از بین می رود. در انتها بریدن نهایی را انجام می دهیم.

### تصویر ۳:

در اینجا همان طور که اشاره شده از روش هرم استفاده می کنیم. در این توضیحات تنها به جاهایی که با توضیحات تصویر ۱ متفاوت است اشاره می کنیم. من از هرم گوسی<sup>۴</sup> در این روش استفاده کردم. به این صورت عمل می کنم که ۳ بار متوالی تصویر را بلور با ضرایب گوسی می کنم و سپس اندازه ی تصویر را به روش نزدیکترین همسایه<sup>۵</sup> از هر طرف نصف می کنم (همان حذف سطر و ستون ساده). حال در این اندازه بهترین جواب را به دست می آورم و قسمت مشترک بین ۳ رنگ می گیرم و بقیه را کنار می گذارم. حال اندازه ی هر کدام را در هر طرف ۲ برابر می کنم (با استفاده از روش bicubic) این کار را تکرار می کنم تا ۳ بار ۲ برابر کرده باشم و تصویر حاصل می شود. نکاتی در مورد نحوه ی بررسی:

<sup>۳</sup>exhaustive search

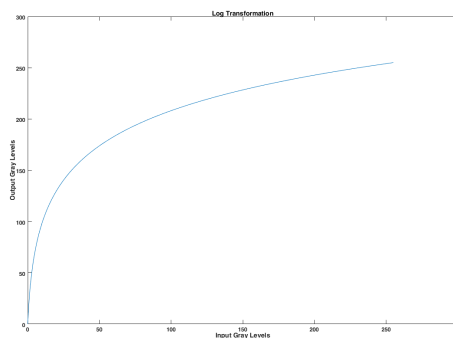
<sup>۴</sup>Gaussian Pyramid

<sup>۵</sup>nearest neighbor

- در هر مرحله که می‌گردم یک منطقه  $25 \times 25$  را با گام‌های ۵ تایی می‌گردم. اینگونه محدوده‌ی وسیع‌تری را بررسی کرده‌ام بعد از پیدا کردن بهترین جایگاه یک محدوده‌ی  $8 \times 8$  را نیز به صورت دقیق می‌گردم که جواب بهتر شود. این مرحله‌ی آخر اختیاری است و زمان بر. در سیستم من با وجود این مرحله تقریب ۷۰ ثانیه طول می‌کشد ولی بون آن ۳۰ ثانیه. تفاوت ایجاد شده با وجود این مرحله بسیار اندک است بنابراین اگر سرعت برای شما مهم‌تر است می‌توانید آن را حذف کنید. کافیست یک { در جایگاه مشخص شده اضافه کنید.
- وقتی شمار pixelها از حدی بیشتر باشد همه‌ی آنها برای مقایسه و نمره دهی در نظر گرفته نمی‌شوند و نسبت به این شمار برخی انتخاب می‌شوند و آنها تصمیم گیرنده‌اند. این کار را من با کوچک کردن به روش نزدیک‌ترین همسایه<sup>۶</sup> انجام می‌دهم.

## توضیح تابع‌ها:

### تابع لگاریتمی:



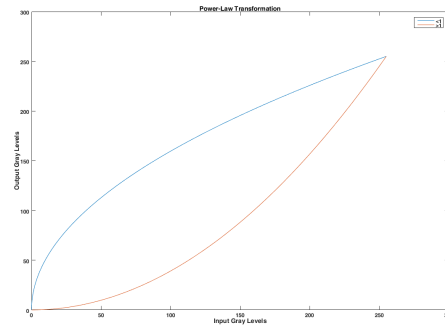
به سادگی دیده می‌شود که مقادیر سیاه روشن می‌شوند ولی مقادیر روشن‌تر کمتر روشن می‌شوند.

### تابع توانی:

برای توان‌های کمتر از ۱ به سادگی دیده می‌شود که مقادیر سیاه روشن می‌شوند ولی مقادیر روشن‌تر کمتر روشن می‌شوند و برای توان‌های بزرگتر برعکس.

---

<sup>۶</sup>nearest neighbor



### wiener2:

توضیح مطلب: wiener2 lowpass-filters a grayscale image that has been de-graded by constant power additive noise. wiener2 uses a pixelwise adaptive Wiener method based on statistics estimated from a local neighborhood of each pixel.