



تمرین کامپیوتری سوم

سیستم‌های عامل - پاییز 1400

طراحی چندریسه‌ای

دانشکده مهندسی برق و کامپیوتر

مسئولان تمرین:

مهلت تحویل:

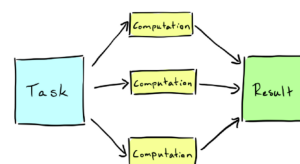
استاد:

میینا شاه‌بنده، محمدامین باقرشاهی

تا ساعت 23:59 روز یکشنبه 28 آذر

دکتر مهدی کارگهی

مقدمه



هدف از این تمرین آشنایی شما با مفاهیم اولیه طراحی چندریسه‌ای¹ یک مسئله است. در این

تمرین شما به اعمال فیلترهایی روی تصاویر می‌پردازید. این تصاویر در فرمت 24 بیتی بیت‌مپ²

هستند و کد نحوه خواندن این تصاویر به شما داده شده و شما باید اعمال فیلترها روی این تصاویر را در دو حالت سریال و موازی پیاده‌سازی کنید.

شرح تمرین



در این تمرین شما به اعمال فیلترهایی روی تصاویر می‌پردازید و پس از انجام مراحل، نتیجه که تصویری

تغییر یافته است، به دست می‌آید. در ابتدا برنامه شما اقدام به خواندن تصویر ورودی کرده و مقادیر

سه کانال رنگی پیکسل‌های آن را در حافظه خود ذخیره می‌کند. پس از استخراج این اطلاعات،

برنامه اقدام به اعمال مرحله به مرحله فیلترهای مورد نظر می‌کند. در این تمرین شما به دو روش این مسئله را پیاده‌سازی می‌کنید.

¹ Muti-Threaded Design

² Bitmap

خواندن تصویر

کد این قسمت در فایل با نام `readImg.cpp` در کنار این پرونده به شما ارائه شده و شما باید این کد را تکمیل کنید. مقدار عددی هر پیکسل از تصویر در حالت RGB (مقادیر سه کانال رنگی قرمز، سبز و آبی) را باید در ساختمان داده‌ی دلخواه خود ذخیره کنید. از این مقادیر در مراحل بعدی برای ایجاد تغییر در تصویر استفاده خواهید کرد. همچنین این مقادیر در بازه 0 تا 255 هستند. تصویر نمونه‌ای که به شما داده شده به شکل زیر است:

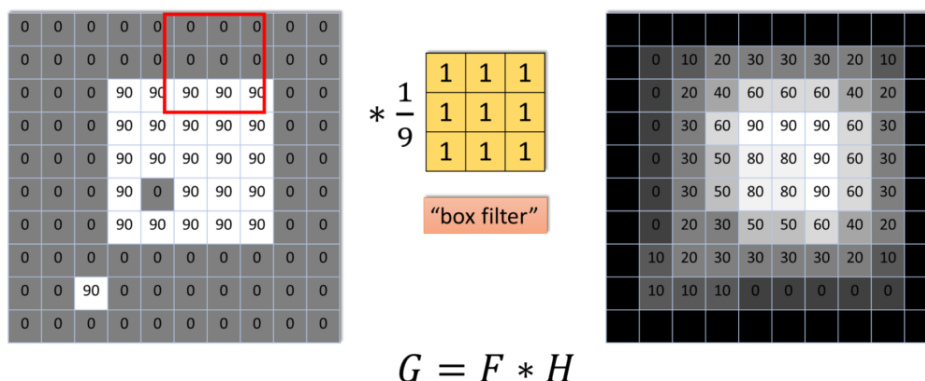


فیلتر Smoothing

این فیلتر تصویر را تار³ می‌کند. برای اعمال این فیلتر، باید مقدار هر کانال رنگی هر پیکسل را برابر با میانگین مقدار همان کانال خود و همسایه‌های آن پیکسل قرار دهید. در واقع برای هر پیکسل، پنجره‌ای 3 در 3 داریم که از مقادیر آن میانگین گرفته می‌شود و برای پیکسل وسطی قرار داده می‌شود.

³ Blur

$$F(x, y) * H(u, v) = G(x, y)$$



حاصل این مرحله به شکل زیر است:



فیلتر Sepia

این فیلتر حالتی قدیمی به تصاویر می دهد. برای اعمال این فیلتر روی هر پیکسل، از روابط زیر استفاده کنید:

$$\text{outputRed} = (\text{inputRed} * 0.393) + (\text{inputGreen} * 0.769) + (\text{inputBlue} * 0.189)$$

$$\text{outputGreen} = (\text{inputRed} * 0.349) + (\text{inputGreen} * 0.686) + (\text{inputBlue} * 0.168)$$

$$\text{outputBlue} = (\text{inputRed} * 0.272) + (\text{inputGreen} * 0.534) + (\text{inputBlue} * 0.131)$$

حاصل این مرحله (با انجام مراحل قبل) به شکل زیر است:



فیلتر میانگین کلی

این فیلتر تصاویر را به حالت شسته شده⁴ در می آورد. برای اعمال این فیلتر، ابتدا باید میانگین مقدار تمام پیکسل های تصویر را برای هر سه کانال رنگی بدست آورده و سپس مقدار هر کانال رنگی هر پیکسل را به حاصل رابطه زیر تغییر دهید:

$$output_{channel} = input_{channel} * 0.4 + mean_{channel} * 0.6$$

حاصل این مرحله (با انجام مراحل قبل) به شکل زیر است:



⁴ Washed out

افزودن ضربدر روی تصویر

در این مرحله ضربدری روی تصویر ایجاد می شود. برای انجام این مرحله، باید سه ردیف پیکسل از صفحه که روی قطر تصویر قرار می گیرند (ردیفی که دقیقا روی قطر است به همراه دو ردیف دو طرف آن) را به رنگ سفید درآورید. مقدار متناظر با رنگ سفید برابر با حداکثر مقدار ممکن برای هر پیکسل است.

حاصل این مرحله (با انجام مراحل قبل) به شکل زیر است:



پیاده سازی سری⁵

در این بخش از تمرین شما به پیاده سازی سری برنامه خواسته شده می پردازید. سعی کنید در این بخش از تمرین بهترین پیاده سازی که می توانید را از لحاظ زمان اجرا انجام دهید؛ زیرا که برای مقایسه عملکرد پیاده سازی چندریسه ای با سری، باید حالت سری در حالت بهینه پیاده سازی شده باشد. پس از این مرحله اعمالی را که بیشترین زمان اجرا را به خود اختصاص داده اند شناسایی کنید.

پیاده سازی چندریسه ای

در این بخش از تمرین به موازی سازی اعمال صورت گرفته در توابعی که در بخش قبل به عنوان Hotspot⁶ از آن ها یاد کردید می پردازید. خواندن ورودی و ذخیره سازی آن در حافظه از اعمال زمان گیر در بسیاری از برنامه هاست که احتمالا از توابع مربوط به آنها (درکنار سایر توابع) به عنوان Hotspot های برنامه یاد کرده اید. برای موازی سازی این بخش می توانید خواندن و ذخیره سازی مقادیر پیکسل های تصویر و اعمال فیلتر روی آنها را توسط چندین ریشه انجام دهید. بهترین ترکیب تعداد ریشه ها، نحوه تقسیم داده ها و

⁵ Serial

⁶ توابعی که در برنامه تان بیشترین زمان اجرا را به خود اختصاص می دهند.

مکانیزم های همگام سازی ریسه ها را باید بدست آورده و انتخاب های خود را توجیه کنید. در انتها، میزان تسریع پیاده سازی چندریسه ای به پیاده سازی سری را از رابطه زیر بدست آورده و طبق قالب خروجی که در ادامه آمده است، گزارش کنید:

$$speedup = \frac{serial\ execution\ time}{parallel\ execution\ time}$$

- **دقت کنید** که خروجی برنامه چندریسه ای شما باید عیناً مطابق با خروجی برنامه سری شما باشد.
- **توجه شود** که این بخش از تمرین باید به صورت **چندریسه ای** پیاده سازی گردد و سایر پیاده سازی ها قابل قبول نیست.
- **دقت شود** برای موازی سازی پروژه تنها مجاز به استفاده از کتابخانه **PThread** هستید و استفاده از کتابخانه های دیگر (بجز کتابخانه های پایه زبان C++) مجاز نیست.
- نام فایل اجرایی شما در هر دو حالت سری و موازی باید ImageFilters.out باشد.

ورودی و خروجی برنامه

برنامه شما باید نام فایل تصویر ورودی را از خط فرمان دریافت کند. نمونه اجرای برنامه با فرض اینکه تصویر ورودی با نام ut.bmp در کنار فایل اجرایی شما قرار گرفته است در زیر آمده است.

نمونه اجرا

```
./ImageFilters.out ut.bmp
```

قالب و نمونه خروجی این اجرای برنامه در زیر آمده است که در آن باید مقدار speedup تا دو رقم اعشار گزارش شود. همچنین علاوه بر خروجی خط فرمان، تصویر خروجی برنامه شما باید در فایلی با نام output.bmp در کنار فایل اجرایی شما نوشته شود و صحت آن بررسی می شود.

قالب خروجی

```
Speedup: <speedup_value>
```

نمونه خروجی

```
Speedup: 2.12
```

توجه: نام فایل خروجی برنامه و متن قالب خروجی را از اینجا کپی نکنید و آن را تایپ کنید!

نکات تکمیلی

- تمام خروجی‌های برنامه را در جریان خروجی استاندارد⁷ چاپ کنید.
- تضمین می‌شود که ورودی‌هایی که به برنامه شما داده می‌شود صحیح هستند و نیازی به بررسی صحت ورودی توسط برنامه شما نیست.
- طراحی درست، کارایی⁸ برنامه و شکستن برنامه به بخش‌های کوچکتر تأثیر زیادی در نمره‌ی تمرین دارد.

نحوه‌ی تحویل

- **دقت کنید** که فایل آپلودی شما با نام OS_CA3_<SID>.zip حتماً باید شامل دو پوشه⁹ مجزا باشد که در یک پوشه پیاده‌سازی سری و در پوشه دیگر پیاده‌سازی موازی آورده شده است. دقت کنید که فایل zip شما شامل فولدر بیرونی نباشد و مستقیماً پس از unzip کردن آن، دو پوشه پیاده‌سازی سریال و موازی شما بدست آید. تصویر ورودی و خروجی را در فایل آپلودی خود قرار ندهید.

○ برای مثال، نمونه فایل مورد قبول در زیر آمده است:

OS_CA3_810198999.zip

```
|— parallel
|   |— main.cpp
|   |— makefile
|— serial
|   |— main.cpp
|   |— makefile
```

- برنامه شما باید در سیستم عامل لینوکس و با مترجم g++ با استاندارد c++11 ترجمه و در زمان معقول برای ورودی‌های آزمون اجرا شود.
- **دقت کنید** که پروژه شما باید دارای Makefile باشد. همچنین در Makefile خود مشخص کنید که از استاندارد c++11 استفاده می‌کنید.

⁷ Standard Output Stream

⁸ Performance

⁹ Directory

- نام فایل اجرایی شما که در کنار Makefile خود ساخته می‌شود باید ImageFilters.out باشد.
- نکته‌هایی که در جلسه توجیهی تمرین گفته می‌شود و یا در فروم‌های مربوطه مطرح می‌شود بخشی از تمرین هستند؛ بنابراین به آن‌ها توجه داشته باشید.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت کشف تقلب مطابق قوانین درس با آن برخورد خواهد شد.