

آرین حدادی ۸۱۰۱۹۶۴۴۸

ایمان مرادی ۸۱۰۱۹۶۵۶۰

گزارش تمرین کامپیوتری سوم برنامه نویسی موازی

سوال اول

برای این بخش ابتدا عکس های داده شده به صورت دستی crop شدند تا ابعاد آن بر ۱۶ بخش پذیر باشند زیرا در SIMD حداکثر ۱۶ پیکسل را به صورت همزمان میتوان خواند. تصویر را به صورت grayscale با استفاده از کتابخانه opencv میخوانیم. در مرحله serial روی سطر ها و ستون ها for زده و در هر پیکسل مقدار قدر مطلق تفاضل پیکسل متناظر در تصویر اول و پیکسل متناظر در تصویر دوم را قرار می دهیم. اما در مرحله parallel این بار روی سطر ها و ستون ها باز هم for میزنیم ولی هر بار ۱۶ پیکسل از سطری که در آن هستیم را از هر دو تصویر میخوانیم.

توجه شود که چون به پوینتر های از جنس `_m128i` کار میکنیم با هر بار ++ کردن روی پوینتر به اندازه سایز این متغیر یعنی ۱۲۸ بیت یا همان ۱۶ بایت به عدد آن افزوده می شود. لذا ماکسیمم مقداری که `col` که متغیری است که در `for` روی ستون های عکس از آن استفاده میکنیم تا `NCOLS/16` خواهد رفت. برای محاسبه محل خواندن و نوشتن pixel ها هم از عبارت `row * NCOLS/16 + col` استفاده میکنیم.

برای محاسبه قدر مطلق تفاضل مقدار خوانده شده را به صورت saturation از هم کم میکنیم. یکبار پیکسل تصویر اول را از پیکسل تصویر دوم کم میکنیم و یکبار برعکس. چون به صورت saturation از هم کم کرده ایم یکی از این دو مثبت و دیگری صفر خواهد شد. لذا خروجی این دو را با هم OR میکنیم.

```
output = _mm_or_si128(_mm_subs_epu8(firstImagePixelsSet, secondImagePixelsSet),  
                      _mm_subs_epu8(secondImagePixelsSet, firstImagePixelsSet));
```

سپس خروجی بالا را در محل متناظر در تصویر خروجی می نویسیم. به این شکل نتیجه نهایی بدست می آید. (شکل در صفحه بعد)



میزان تسریع اکثرا بیت 4 تا 6 بود.

```
Run Time For Serial Calculation for Q1 is 492 microseconds.  
Run Time For Parallel Calculation for Q1 is 111 microseconds.  
Q1 Speedup = 4.43243
```

سوال دوم

برای این بخش ابتدا عکس های داده شده به صورت دستی crop شدند تا ابعاد آن ها مضربی از ۱۶ باشد زیرا در SIMD حداکثر ۱۶ پیکسل را به صورت همزمان می توان خواند.

در مرحله پیاده سازی ابتدا تصاویر را به صورت grayscale میخوانیم. برای بخش serial که صرفا کافیت تصویر را خوانده و در خروجی ریخت و سپس روی ابعاد تصویر کوچکتر for زد و هر کدام از پیکسل های آن را در نیم ضرب کرده و با پیکسل خروجی جمع کرده و با عدد ۲۵۵ مقدار آن را مینیمم گرفت (به صورت saturation جمع زد)

برای مرحله parallel هم در ابتدا باز خروجی را از روی تصویر اول که تصویر بزرگتر است میخوانیم. توجه شود که چون تصویر نهایی همان تصویر اول که تصویر بزرگتر است به همراه تصویر دوم در گوشه ای از آن است هم در مرحله parallel و هم در مرحله serial ابتدا برای خروجی تصویر اول را میخوانیم و سپس تغییراتی که لازم است را روی همان خروجی اعمال میکنیم. و نیازی به نوشتن بقیه پیکسل ها که ثابت است به صورت دوباره نیست.

سپس روی ابعاد تصویر کوچکتر for میزنیم.

توجه شود که چون به پوینتر های از جنس __m128i کار میکنیم با هر بار ++ کردن روی پوینتر به اندازه سائز این متغیر یعنی ۱۲۸ بیت یا همان ۱۶ بایت به عدد آن افزوده می شود. لذا ما کسیم مقداری که col که متغیری است که در for روی ستون های عکس از آن استفاده میکنیم تا NCOLS/16 خواهد رفت. برای محاسبه محل خواندن و نوشتن pixel ها هم از عبارت row * NCOLS/16 + col استفاده میکنیم.

در هر مرحله for به تعداد ۱۶ پیکسل که چون هر پیکسل ۸ بیت است میشود ۱۲۸ بیت و کل فضای متغیر __m128i پر می شود. حال این پیکسل ها را باید بر ۲ تقسیم کنیم. برای این کار یک واحد به راست کل متغیر ۱۲۸ بیتی را شیفت می دهیم و سپس با یک متغیر به نام mask که در هر ۸ بیتش یک 0x7f نوشته شده است AND میکنیم. زیرا در این متغیر mask در هر ۸ بیت تمام بیت ها جز بیت سمت چپ ۱ هستند و با AND کردن بیت سمت چپ هر بسته ۸ بیتی برابر ۰ می شود و بقیه همانطور که هستند باقی می مانند. این متغیر به این شکل تعریف شده است:

```
__m128i mask = _mm_set1_epi8(0x7f);
```

به این شکل هم AND را انجام می دهیم:

```
in = _mm_and_si128(in, mask);
```

به این صورت بیت سمت چپ هر ۸ بیتی از این ۱۲۸ بیت برابر صفر می شود.

دلیل انجام این عملیات آن است که دستوری برای شیفت دادنی بسته های ۸ بیتی نداریم و حداقل بسته برای شیفت دادن که دستوری برای آن وجود دارد ۱۶ بیتی است. سپس مقدار شیفت داده شده را به صورت SATURATION جمع کرده و در خروجی می نویسیم.

پس از اتمام محاسبات دو تصویر را نمایش می دهیم. (تصویر در صفحه بعد)



میزان تسریع اکثرا بین ۸ تا ۱۰ بود.

```
Run Time For Serial Calculation Q2 is 344 microseconds.  
Run Time For Parallel Calculation Q2 is 38 microseconds.  
Q2 Speedup = 9.05263
```