پروژه اول برنامه نویسی موازی

کسرا نوربخش ۸۱۰۱۰۰۲۳۰

شهنام فیضیان 810100197

مقدمه

در این پروژه در 2 سوال پردازش آرایه و همچنین در 2 سوال پردازش تصویر و ویدعو صورت گرفت که ابتدا پیاده سازی بهینه ای در حالت سریال انجام دادیم و سپس پیاده سازی موازی بهینه. همچنین در بخش مربوط به هر سوال میزان speedup(که با استفاده از کتابخانه ipp صورت گرفت) و همچنین خروجی، گزارش شد.

Image Blending

ابتدا به مشاهده خروجی و میزان تسریع میپردازیم:

عکس 1: خروجی سریال(بالا) خروجی موازی(پایین)

```
Serial Run time = 8244498 cycles
Parallel Run time = 1219789 cycles
Speedup = 6.75895
```

عکس 2: میزان تسریع سوال اول

در هر دو حالت، یک ROI(Region Of Interest) تعریف شد که در این سوال قسمت بالا، چپ بود و سپس پردازش بر روی این ناحیه صورت گرفت، هم چنین یک پارامتر به نام alpha داریم که مشخص کننده ضریب ترکیب 2 عکس با یکدیگر است. برای بهینه بودن بخش سریال، دسترسی به عناصر آرایه را به صورت پوینتری، به عنوان مثال: const Vec3b* frontRow = frontROI.ptr<Vec3b>(y); انجام دادیم. پردازش را در حالت سریال برای هر پیکسل و روی هر 3 کانال آن انجام دادیم. ابتدا برای به دست آوردن نحوه ذخیره سازی تصاویر توسط OpenCV از دستور type(). کمک گرفتیم که عدد 16 را به ما برگرداند که با توجه به جدول شماره 1، یعنی هر پیکسل از 3 کانال به ترتیب: B, G, R تشکیل شده اند و مقادیر هر کانال، 8 بیت بدون غلامت می باشند. هم در بخش موازی و هم در بخش سریال نیاز بود تا بعد از پردازش، مقادیر به دست آمده محدود به بازه 0 تا 255 شوند(clamping). در تمامی کد های موازی نوشته شده، دقت به عمل آمد تا دستور های زمان بر به مانند: __m128 alpha_val = _mm_set1_ps(alpha);، خارج حلقه استفاده شوند. نحوه پیاده سازی بخش موازی به این صورت است که ابتدا به صورت موازی، 4 کانال به 4 کانال، لوگو را با alpha ضرب می کنیم و بعد، لوگو پردازش شده و عکس پس زمینه را، 16 کانال به 16 کانال، جمع می کنیم.

280

For debugging purposes in case you want to look up a raw `Mat::type` in a debugger:

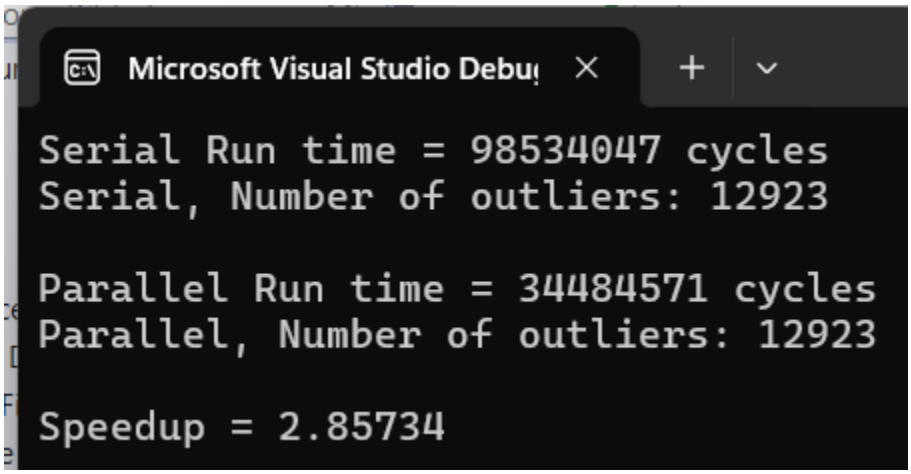
	C1	C2	C3	C4	C(5)	C(6)	C(7)	C(8)
CV_8U	0	8	16	24	32	40	48	56
CV_8S	1	9	17	25	33	41	49	57
CV_16U	2	10	18	26	34	42	50	58
CV_16S	3	11	19	27	35	43	51	59
CV_32S	4	12	20	28	36	44	52	60
CV_32F	5	13	21	29	37	45	53	61
CV_64F	6	14	22	30	38	46	54	62

So for example, if type = 30 then OpenCV data type is `CV_64FC4`. If type = 50 then the OpenCV data type is `CV_16UC(7)`.

جدول 1: تایپ های مختلف تصاویر

Outlier

ابتدا به مشاهده میزان تسریع میپردازیم:

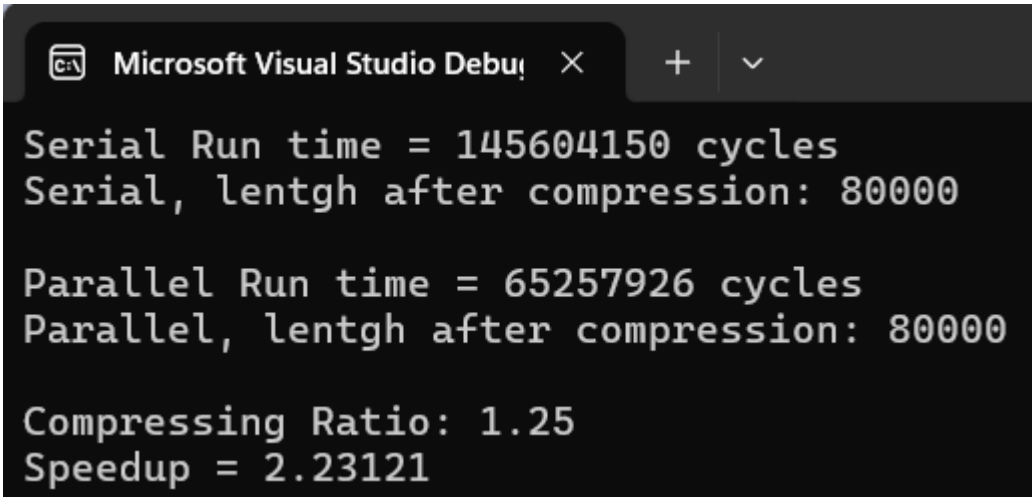


عکس 3: میزان تسریع سوال دوم

برای پیاده سازی این بخش، توابع سریال و موازی میانگین گیر و محاسبه کننده انحراف از معیار نوشتیم و به خاطر تطابق نسخه سریال با موازی و بهینه بودن آن، از تکنیک loop unrolling بهره گرفتیم. در ابتدا چون حد z-score برابر با 2.5 بود و توزیع داده های تصادفی، به صورت یکنواخت بین 0 و 1 صورت می گرفت، outlier ای نداشتیم بنابراین با توزیع نرمال داده های تصادفی را تولید کردیم.

Compression

ابتدا به مشاهده خروجی و میزان تسریع میپردازیم:



عکس 4: میزان تسریع سوال سوم

برای ران کردن برنامه یک رشته پایه و همچنین تعداد تکرار این رشته پایه باید به عنوان command argument به برنامه داده شود. برای حل این مسئله 16 کاراکتر داخل رجیسترهای وکتوری لود می کنیم. سپس آنها را یک بایت به سمت راست شیفت می دهیم و در وکتور دیگری ذخیره می کنیم. حال دو وکتور را با یکدیگر مقایسه می کنیم. نتیجه این مقایسه به ما می گوید که هر

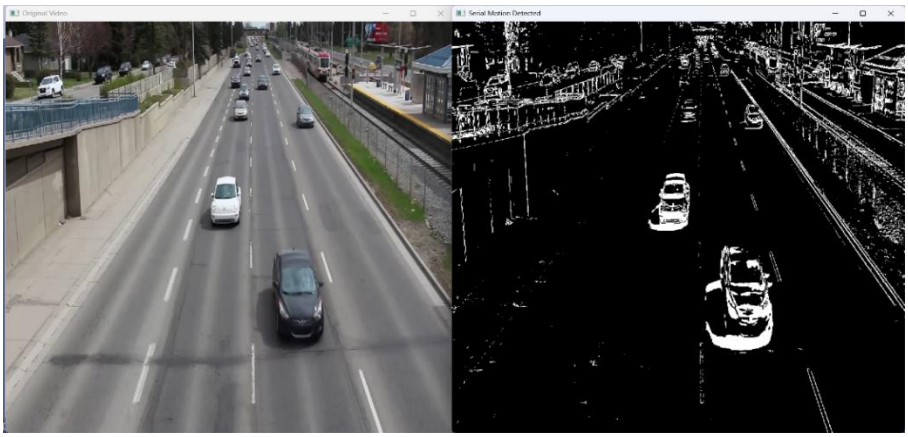
کاراکتر با کاراکتر سمت راستی خود یکی است یا خیر، بنابراین بعد از این مقایسه صرفا با شمارش یک های متوالی می توانیم تعداد تکرار هر کاراکتر به صورت پشت سر هم را پیدا کنیم.

توابع کمکی نوشته شده:

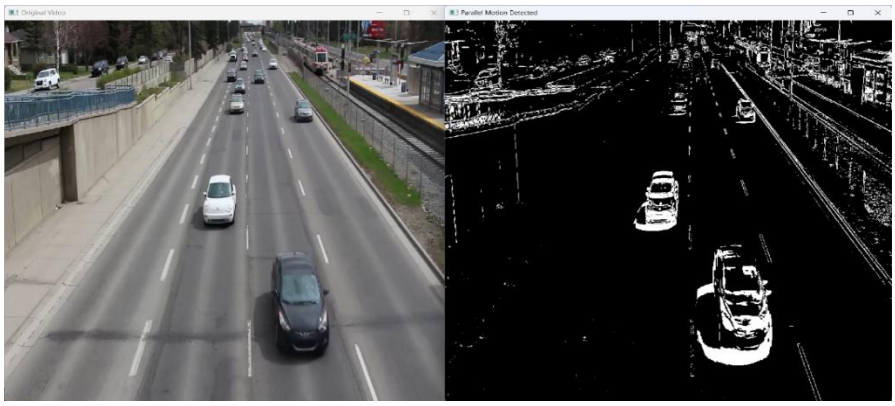
1. **تابع generate_sample:**
   - برای تولید یک رشته نمونه تکراری از یک رشته پایه به تعداد مشخصی استفاده می‌شود.
2. **تابع write_report:**
   - گزارشی از نتایج اجرای دو الگوریتم سریالی و موازی شامل زمان اجرا، طول رشته فشرده‌شده و نسبت فشرده‌سازی را چاپ می‌کند.
   - همچنین، سرعت نسبی دو روش (سرعت بیشتر نسخه موازی در مقایسه با سریالی) را محاسبه می‌کند.

Motion Detection

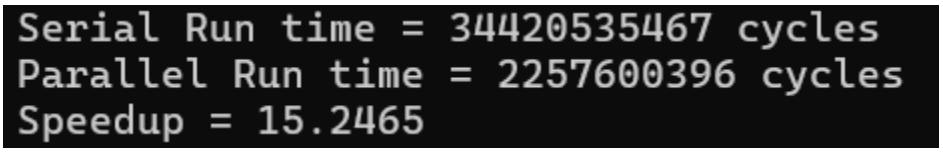
ابتدا به مشاهده خروجی و میزان تسریع میپردازیم:



عکس 5: خروجی سریال



عکس 6: خروجی موازی



عکس 7: میزان تسریع سوال چهارم

هم در پیاده سازی موازی و هم در پیاده سازی سریال، یک threshold قرار دادیم تا باعث تشخیص بهتر حرکت شود. در این سوال به دلیل ارتباط زیاد با حافظه، مدت زمان نسخه سریال و موازی را فقط در بخش پردازشی محاسبه کردیم. (در هر دو حالت با استفاده از کلید q می توان برنامه را خاتمه داد.) پردازش موازی این سوال، 16 پیکسل به 16 پیکسل انجام شد.