Bicubic 테이블 분석

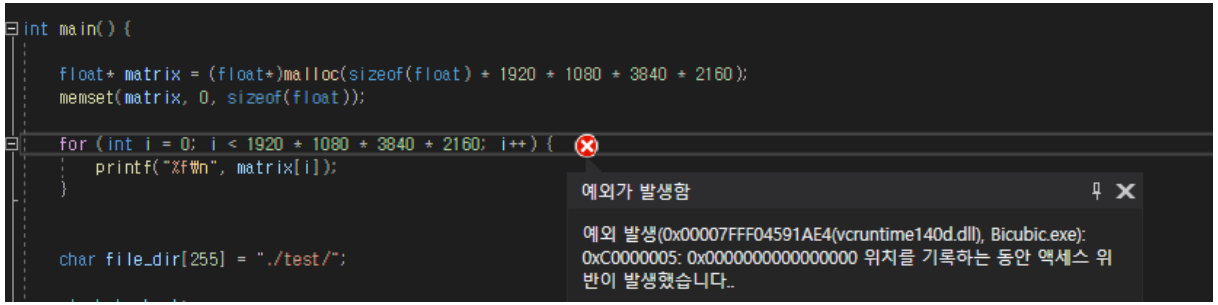
염지현

SM, Origin 방법 기반 시간 측정표

SM, Origin 방법 기반 시간 측정표

1) MM

			Python	C언어	
				no CUDA	with CUDA
	MM		메모리 터짐	메모리 터짐	메모리 터짐
연산 시간	ORIGIN	Padding(이미지 패딩 추가)	0.018	0.201	0.021
		내적	1330	9.502	0.007
		합	1330	9.523	0.028
	SM	SM (matrix 생성)	493.05	10.3	
		SM (matrix 곱)	2336.22	0.91	
		합	2829.27	11.2	



〈C언어〉

〈파이썬〉

SM, Origin 방법 기반 시간 측정표

1) MM - matrix 용량 한계

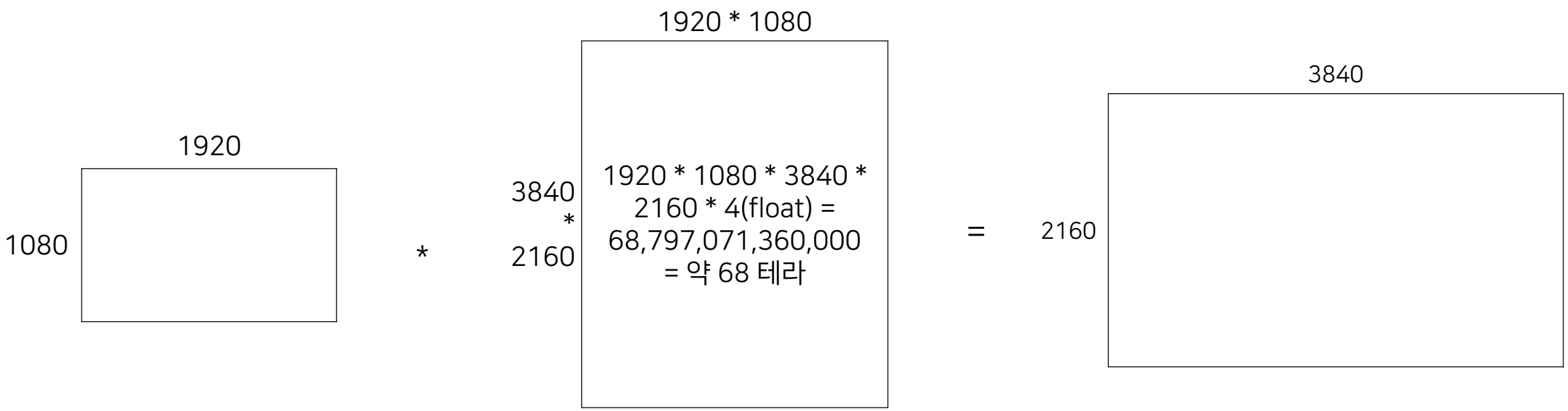
```
int main() {

    float* matrix = (float*)malloc(sizeof(float) * 1920 * 1080 * 3840 * 2160);
    memset(matrix, 0, sizeof(float));

    for (int i = 0; i < 1920 * 1080 * 3840 * 2160; i++) {
        printf("%f\n", matrix[i]);
    }


    char file_dir[255] = "./test/";
```

예외가 발생함

예외 발생(0x00007FFF04591AE4(vcruntime140d.dll), Bicubic.exe): 0xC0000005: 0x0000000000000000 위치를 기록하는 동안 액세스 위반이 발생했습니다..

Kernel Restarting

The kernel appears to have died. It will restart automatically.



SM, Origin 방법 기반 시간 측정표

1) MM - matrix 용량 한계

```
int main() {

    float* matrix = (float*)malloc(sizeof(float) * 1920 * 1080 * 3840 * 2160);
    memset(matrix, 0, sizeof(float));

    for (int i = 0; i < 1920 * 1080 * 3840 * 2160; i++) {
        printf("%f\n", matrix[i]);
    }


    char file_dir[255] = "./test/";
```

예외가 발생함

예외 발생(0x00007FFF04591AE4(vcruntime140d.dll), Bicubic.exe): 0xC0000005: 0x0000000000000000 위치를 기록하는 동안 액세스 위반이 발생했습니다.

Kernel Restarting

The kernel appears to have died. It will restart automatically.

메모리 부족

1920 * 1080

1920

1080

*

3840

*

2160

1920 * 1080 * 3840 * 2160 * 4(float) = 68,797,071,360,000 = 약 68 테라

=

2160

3840

SM, Origin 방법 기반 시간 측정표

2) ORIGIN

			Python	C언어	
				no CUDA	with CUDA
	MM		메모리 터짐	메모리 터짐	메모리 터짐
연산 시간	ORIGIN	Padding(이미지 패딩 추가)	0.018	0.201	0.021
		내적	1330	9.502	0.007
		합	1330	9.523	0.028
	SM	SM (matrix 생성)	493.05	10.3	
		SM (matrix 곱)	2336.22	0.91	
		합	2829.27	11.2	

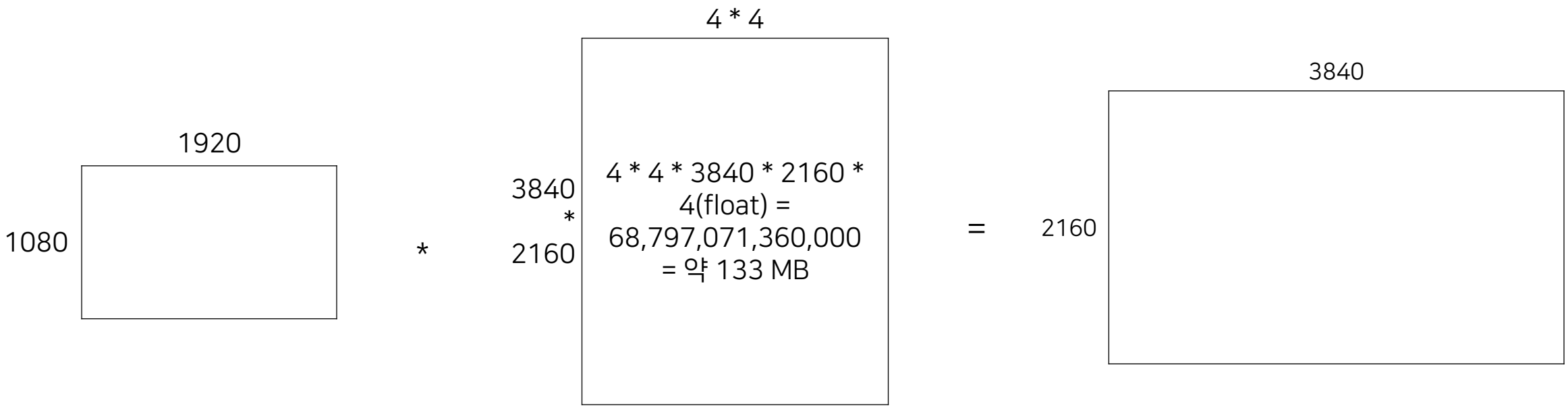
```
mat_l = np.matrix([[u(x1,a),u(x2,a),u(x3,a),u(x4,a)]])
mat_m = np.matrix([[img[int(y-y1),int(x-x1),c],img[int(y-y2),int(x-x1),c],img[int(y+y3),int(x-x1),c],img[int(y+y4),int(x-x1),c]],
                   [img[int(y-y1),int(x-x2),c],img[int(y-y2),int(x-x2),c],img[int(y+y3),int(x-x2),c],img[int(y+y4),int(x-x2),c]],
                   [img[int(y-y1),int(x+x3),c],img[int(y-y2),int(x+x3),c],img[int(y+y3),int(x+x3),c],img[int(y+y4),int(x+x3),c]],
                   [img[int(y-y1),int(x+x4),c],img[int(y-y2),int(x+x4),c],img[int(y+y3),int(x+x4),c],img[int(y+y4),int(x+x4),c]]])
mat_r = np.matrix([[u(y1,a)],[u(y2,a)],[u(y3,a)],[u(y4,a)]])
dst[j, i, c] = np.dot(np.dot(mat_l, mat_m),mat_r)
```

x, y 좌표를 기반으로 함수 u에 의해 weight 값 출력 → weight 값과 픽셀 값 내적 → 한 픽셀에 해당하는 값 구하기

SM, Origin 방법 기반 시간 측정표

3) SM

			Python	C언어	
				no CUDA	with CUDA
연산 시간	MM		메모리 터짐	메모리 터짐	메모리 터짐
	ORIGIN	Padding(이미지 패딩 추가)	0.018	0.018	0.201
		내적	1330	1330	9.502
		합	1330	1330	9.523
	SM	SM (matrix 생성)	493.05	10.3	
		SM (matrix 곱)	2336.22	0.91	
		합	2829.27	11.2	



Bicubic 기반의 dataset 결과 PSNR 측정표

Bicubic 기반의 dataset 결과 PSNR 측정표

1) ORIGIN ver

Bicubic origin method	PSNR
Museum	33.48
CafeInterior	32.04
PolyTown	35.05
Village	35.4

Bicubic 기반의 dataset 결과 PSNR 측정표

2) SMM ver → 인덱스 오류로 인해 weight 값이 미묘하게 달라 PSNR이 약 10 정도 낮게 나옴(수정 필요)