

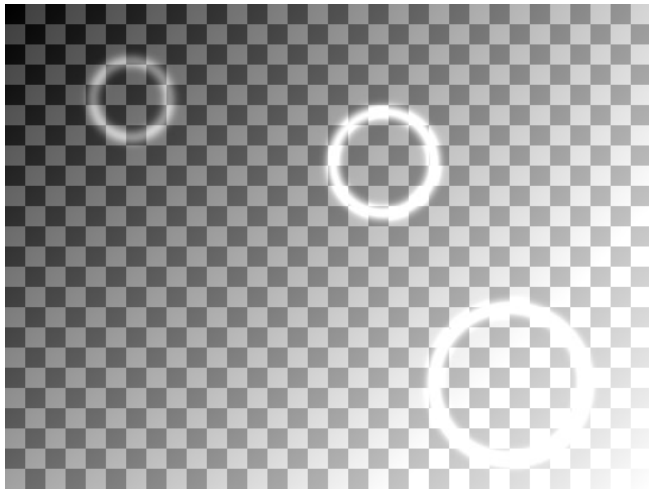
HPE² LAB

병렬프로그래밍 기말프로젝트

Canny Edge Detector
[OpenMP]

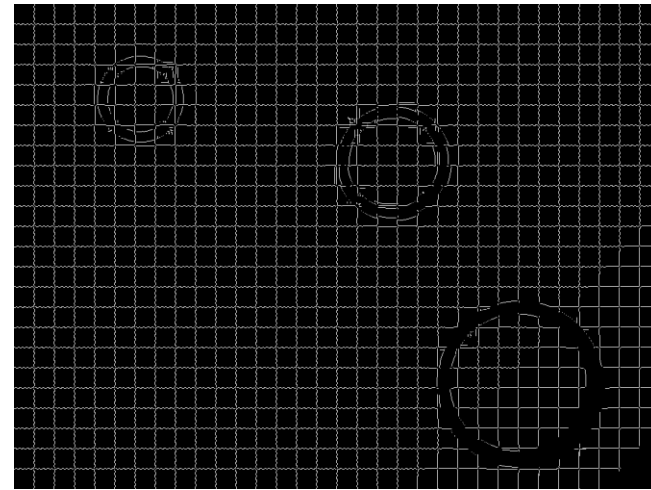
Canny Edge Detector

컴퓨터 상에서 이미지 내의 Edge를 인식하기 위해 구현된 알고리즘
OpenCV에서는 Canny() 함수를 통해 이 기능을 제공하고 있으며,
사용 시에는 다음과 같이 이미지 내의 Edge만 출력된다.



Before

Canny()



After

Canny Edge Detector Pipeline

이번 과제의 핵심이 되는 Serial Code로,
파이프라인 구조의 Canny Edge Detector를 C++로 구현함

[프로그램 진행흐름]



(1) Gaussian Filter(`student_kernel.hpp/gaussian`)

노이즈를 제거하기 위한 필터로 5×5 가우시안 필터링과 동일함
경계는 Clamp에 의해 처리됨

Clamp?

특정 값이 주어진 범위를 벗어나지 못하게 하는 보조 함수
Canny Edge Detector에서는 경계 처리를 진행할 때 나오는 패턴

(2) Sobel Filter (`student_kernel.hpp/sobel_grad`)

수직(x)와 수평(y) 방향의 미분값을 구해 기울기를 계산함
Edge의 변화량을 숫자로 만든 단계

미분된 값을 크기와 방향으로 정리하며
방향은 0/45/90/135 도 로 단순화

(3) Non-Maximum Suppression(NMS)

비최대 억제 단계로 Sobel Filter를 거쳐 픽셀 값이 미분 되는 경우, 출력 값에 해당하는 이미지의 Edge 주변이 두껍게 번지듯 나오므로

1. 픽셀 Edge 방향이 어디를 향하는 지 확인하여
2. 해당 방향으로 한 칸 앞/뒤 픽셀과 값을 비교한 후
3. 현재 픽셀이 가장 큰 경우 살리고, 아닌 경우 0으로 지워버리는 것

위 세 단계를 통해 얇게 정리된 Edge 맵을 얻을 수 있게 된다.

(4) auto/double threshold

auto_threshold

비최대 억제까지 끝난 이미지의 전체 통계에서 실제로 사용할 임계값을 계산해주는 함수

double_threshold

결정된 임계값을 기준으로 각 픽셀의 Edge 라벨링을 진행
이미지에서 Edge에 해당할 것 같은 위치를 저장하는 과정

(5) Hysteresis edge tracking

Canny Edge Detector의 마지막 단계로,
double threshold 처리 이후 최종 Edge를 확정하는 단계

라벨링 된 픽셀 중 Edge의 특징이 강한 픽셀과 붙어있는 경우만
값을 유지하고 나머지는 값을 버려 Output.pgm을 생성함

기말프로젝트 목표

Serial Code로 구현된 Canny Edge Detector Pipeline 프로그램을
OpenMP를 통해 병렬화 하여 프로그램의 성능 향상

[채점 항목]

1. **Scalability** : 쓰레드의 수가 증가하면 성능도 더 향상되는지
2. **Execution Time** : 프로그램 실행 시간
3. **Correctness** : 프로그램의 출력물이 정상적인지

기말프로젝트 예시

```
Input: input.pgm (1024x768)
Threads: 2
Reference time: 126.802 ms
Student time: 95.9307 ms
PASS: PASS (Mismatch 0.0120799 %)
```

```
Input: input.pgm (1024x768)
Threads: 4
Reference time: 127.669 ms
Student time: 51.3213 ms
PASS: PASS (Mismatch 0.0120799 %)
```

```
Input: input.pgm (1024x768)
Threads: 8
Reference time: 129.635 ms
Student time: 41.6392 ms
PASS: PASS (Mismatch 0.0120799 %)
```

적절한 병렬화를 적용한 경우

Scalability가 잘 나타나며

쓰레드 수에 관계없이 결과물이 정상적으로 생성되는 것을 확인할 수 있음

기말프로젝트 코드

컴파일

```
g++ -fopenmp -march=native main.cpp -o canny
```

실행

```
./canny [입력 파일명] [출력 파일명] [쓰레드 수]
```

기말프로젝트 유의사항

학생이 수정가능한 코드는 '**student_kernel.hpp**' 코드만 수정 가능
시스템과 관련된 호출을 이용해 향상 시킨 경우는 **무효 처리**
따라서 절대 사용하지 않는 것을 추천함

[유의사항]

1. 모든 학생의 코드는 동일하게 주어진 main.cpp 에서 평가됨
2. 제공된 샘플 PGM 이미지와 다른 이미지로 채점될 예정임
3. 코드 유사도가 높은 경우, 부정행위로 간주될 수 있음