

KANGWON NATIONAL UNIVERSITY

컴퓨터비전 실습

실습7 | LOG

실습과제 이루리 내 제출

CVMIPALAB @ KNU

실습 7-1 | LOG

문제

주어진 코드를 활용하여 “bucks.jpg” 파일을 흑백으로 읽은 뒤, Laplacian of Gaussian 알고리즘을 구현하세요.

요구 결과

아래 이미지와 같이 LOG를 적용한 영상을 화면에 표시 후 “bucks_zerocrossing.bmp”로 저장합니다.

저장된 영상과 “.cpp” 총 두 개의 파일을 압축하여 제출합니다.

제출 관련 공지

※ 압축파일 최상위에 폴더를 만들지 말고, 파일만 그대로 압축해 주시기 바랍니다.

※ 실습이 여러 개인 경우 한 압축 파일에 모두 압축하여 주시기 바랍니다.



실습 7-1 | LOG

설명자료

LOG::CreateMask 함수에 σ 크기의 LOG 필터를 구합니다.

알고리즘 3-2 Marr-Hildreth 에지 검출(LOG 필터 사용)

입력: 영상 $f(j,i)$, $0 \leq j \leq M-1$, $0 \leq i \leq N-1$, 가우시안의 표준편차 σ

출력: 에지 영상 $b(j,i)$, $0 \leq j \leq M-1$, $0 \leq i \leq N-1$ // 에지는 1, 비에지는 0인 이진 영상

- 1 σ 크기의 LOG 필터를 입력 영상 f 에 적용한다.
- 2 결과 영상에서 영교차를 찾아 에지로 설정하고, 나머지는 비에지로 설정한다.

$$\nabla^2 G(y,x) = \left(\frac{y^2 + x^2 - 2\sigma^2}{\sigma^4} \right) G(y,x)$$

// G(y,x) 만들기

```
for (int dy = -halfSize; dy <= halfSize; ++dy)
for (int dx = -halfSize; dx <= halfSize; ++dx)
{
    // (y^2+x^2-2sigma^2) / sigma^4
    // G(y, x)
    double numerator = exp(-((dy * dy + dx * dx) / sigmaSquare));
    double denominator = 3.14 * sigmaSquare;
    mask[halfSize + dy][halfSize + dx] *= numerator / denominator;
}
```

$$\longrightarrow G(y,x) = \frac{1}{2\pi\sigma^2} e^{-\frac{y^2 + x^2}{2\sigma^2}}$$

실습 7-1 | LOG

설명자료

구한 LOG 필터를 LOG::FindZeroCrossing과 LOG::CheckSign을 이용하여 구현합니다.

알고리즘 3-2 Marr-Hildreth 에지 검출(LOG 필터 사용)

입력: 영상 $f(j,i)$, $0 \leq j \leq M-1$, $0 \leq i \leq N-1$, 가우시안의 표준편차 σ

출력: 에지 영상 $b(j,i)$, $0 \leq j \leq M-1$, $0 \leq i \leq N-1$ // 에지는 1, 비에지는 0인 이진 영상

- 1 σ 크기의 LOG 필터를 입력 영상 f 에 적용한다.
- 2 결과 영상에서 영교차를 찾아 에지로 설정하고, 나머지는 비에지로 설정한다.

```
int LOG::CheckSign(const float& value)
{
    /**
     * value의 부호를 결정하는 코드를 작성하세요.
     * 값이 0인 경우도, 0으로 return하도록 하세요.
     */
    // ** 지금부터 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **

    // ** 여기까지 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
}
```

```
void LOG::FindZeroCrossing(cv::InputArray& input_image, cv::OutputArray& output_image)
{
    output_image.create(input_image.size(), CV_8UC1);

    cv::Mat input_mat = input_image.getMat();
    cv::Mat output_mat = output_image.getMat();
    output_mat.setTo(cv::Scalar(0));

    /**
     * 영교차 알고리즘을 구현하세요.
     * 주어진 n0와 같이, n1~n7까지를 구현하고
     * 부호를 확인하여 cnt값을 증가시킵니다.
     * cnt값의 임계치를 2로 두고, 임계치 이상인 경우 영상 output_mat에 255를 할당하세요.
     */
    // ** 지금부터 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
    for (int y = 1; y < input_mat.rows - 1; ++y)
    {
        for (int x = 1; x < input_mat.cols - 1; ++x)
        {
            int cnt = 0;
            float n0 = input_mat.at<float>(y, x + 1);

            // ... (rest of the code for finding zero crossings) ...

        }
    }
    // ** 여기까지 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **
}
```

실습 7-1 | LOG

설명자료

1. 여덟 개의 이웃 중에 마주보는 동-서, 남-북, 북동-남서, 북서-남동의 화소 쌍 네 개를 조사한다. 그들 중 두 개 이상이 서로 다른 부호를 가진다.
2. 부호가 다른 쌍의 값 차이가 임계값을 넘는다.

이번 실습에서는 1번만 설계

n_5	n_6	n_7
n_4	p	n_0
n_3	n_2	n_1



$$n_0! = n_4$$

$$n_1! = n_5$$

$$n_2! = n_6$$

$$n_3! = n_7$$



참인 조건문이 2개 이상인 경우 에지