



KANGWON NATIONAL UNIVERSITY

컴퓨터비전 실습

10주차 | SIFT

CVMIP LAB @ KNU

실습 10 | SIFT 기술자

문제

주어진 코드를 활용하여 "cliff.bmp" 파일을 흑백으로 읽은 뒤,
SIFT 기술자 알고리즘을 구현하여 결과 이미지를 출력하세요.

요구 결과

SIFT 기술자 알고리즘을 구현하고 키포인트 결과 이미지와 첫번째 기술자 출력값을 캡처합니다.
저장된 이미지 파일과 ".cpp" 총 두 개의 파일을 압축하여 제출합니다.

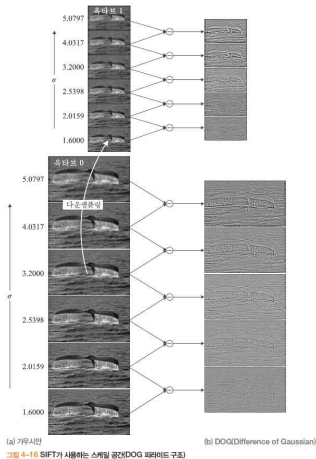
제출 관련 공지

- ※ 파일만 그대로 압축해 주시기 바랍니다.
- ※ 실습이 여러 개인 경우 한 압축 파일에 모두 압축하여 주시기 바랍니다.

실습 10 | SIFT 기술자

이미지에서 크기와 회전에 대해 불변인 특징들을
추출할 수 있는 방법

1. 가우시안을 통해서 이미지를 다운샘플링하고 크기를 줄입니다.
2. Difference of Gaussian을 하여 최대, 최소 특징을 구합니다. 여기서는 그것을 키포인트라고 부릅니다.
3. 여기서 나온 키포인트들 주변의 gradient의 크기와 방향을 계산합니다.



실습 10 | SIFT 기술자

```
void SIFT_c::MakeOctaveImg(cv::Mat& org)
{
    // 첫번째 영상은 스무딩인 적용되었다고 가정함
    Octave oct;
    oct.octave = octaveVec.size();
    org.copyTo(oct.gauImage[0]);

    // 이외의 이미지에 해당하는 시그마 만큼 가우시안 적용
    /*
    CV_EXPORTS_W void GaussianBlur( InputArray src, OutputArray dst, Size ksize,
                                   double sigmaX, double sigmaY = 0,
                                   int borderType = BORDER_DEFAULT );
    src - 원본 영상(org)
    dst - 결과 영상(oct.gauImage array)
    ksize - 커널 사이즈
    sigmaX - sigma(sig array)
    */
    for(int i = 1; i < 6; ++i)
    {
        // 첫번째 이미지 이외의 이미지에 해당하는 시그마 만큼 가우시안 적용하시오.
        // 줄음표에 들어가야할 코드를 입력하시오.
        cv::GaussianBlur(org, oct.gauImage[i], cv::Size(), ?);
    }

    octaveVec.push_back(oct);
}
```

```
void SIFT_c::DOG()
{
    Octave &oct = octaveVec[octaveVec.size()-1];
    for(int i = 1; i < 6; ++i)
    {
        // oct.gauImage array를 이용하여 oct.DOG array에 DOG를 생성하시오.
        // 줄음표에 들어가야할 코드를 입력하시오.
        oct.DOG[i-1] = oct.gauImage[?] - oct.gauImage[?];
    }
}
```

실습 10 | SIFT 기술자

설명자료

알고리즘 6-1 SIFT 기술자 추출

입력 : 입력 영상 f 에서 검출된 키포인트 집합 $p_i = (y_i, x_i, \sigma_i)$, $1 \leq i \leq n$ // 4.4.3절의 알고리즘으로 추출

출력 : 기술자가 추가된 키포인트 집합 $p_i = (y_i, x_i, \sigma_i, \theta_i, \mathbf{x}_i)$, $1 \leq i \leq m$

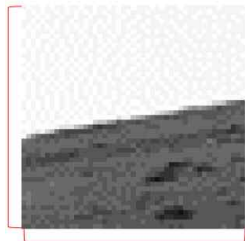
```
1  for( $i=1$  to  $n$ ) {  
2       $p_i$ 의 지배적인 방향  $\theta_i$ 를 계산한다. // 이때 하나의 키포인트가 여러 개로 나뉠 수 있음  
3       $p_i$ 의 특징 벡터  $\mathbf{x}_i$ 를 계산한다.  
4       $\mathbf{x}_i$ 를 정규화한다.  
5  }
```

실습 10 | SIFT 기술자

설명자료

1단계: Dominant Orientation 찾기

→ 영상에서 D_x, D_y 를 계산합니다.



$1.5 * 3 * \text{key.sigma}$

-1	0	1
----	---	---

-1	0	1
----	---	---

^T

D_x

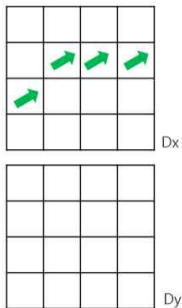
D_y

실습 10 | SIFT 기술자

설명자료

1단계: Dominant Orientation 찾기

→ D_x, D_y 를 이용하여 에지의 방향과 크기를 계산합니다.



$$\text{Edge} = \arctan\left(\frac{dy}{dx}\right) + 90^\circ$$

$$\text{Magnitude} = \sqrt{d_y^2 + d_x^2}$$

0	0	0	0
0	48	43	40
45	0	0	0
0	0	0	0

Edge

0	0	0	0
0	100	90	80
120	0	0	0
0	0	0	0

Magnitude

실습 10 | SIFT 기술자

설명자료

1단계: Dominant Orientation 찾기

→ 아래 함수를 이용하여 Edge의 방향과 크기를 계산하세요.

```
CV_EXPORTS_W void cartToPolar(InputArray x, InputArray y,  
                             OutputArray magnitude, OutputArray angle,  
                             bool angleInDegrees = false);
```

x : dx

y : dy

magnitude : 에지 크기

angle : 에지 방향

angleInDegrees : 각도로 출력할 것인가(실습에서 True로 처리)

실습 10 | SIFT 기술자

설명자료 1단계: Dominant Orientation 찾기

```
// Octave와 Sigma에 맞는 영상으로 처리
int signeldx = sift_GetSigneldx(key, sign);
const cv::Mat& img = sift_octaveVec[key.octave].gslImage[signeldx];
for (int h = -radius; h <= radius; ++h)
{
    int y = key.y + h;
    if (y < 0 || y >= img.rows - 1)
        continue;
    for (int w = -radius; w <= radius; w++)
    {
        int x = key.x + w;
        if (x < 0 || x >= img.cols - 1)
            continue;

        // 픽셀값 dx와 dy를 계산하십시오.
        // 볼륨표에 들어가야할 코드를 입력하십시오.
        float dx = (float)(?);
        float dy = (float)(?);

        cv::Mat Mag(cv::Size(1, 1), CV_32FC1); // 예지 크기
        cv::Mat Angle(cv::Size(1, 1), CV_32FC1); // 예지 방향

        /*
         * CV_EXPORTS_W void cartToPolar(InputArray x, InputArray y,
         *                               OutputArray magnitude, OutputArray angle,
         *                               bool angleInDegrees = false)
         *
         * x : dx
         * y : dy
         * magnitude : 예지 크기
         * angle : 예지 방향
         */
        // 예지 방향과 예지 크기를 계산하십시오. (위의 함수를 활용)
        // ** 지금부터 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **

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

        orient.push_back(Angle.at<double>(0));
        mag.push_back(Mag.at<double>(0));
    }
}
```

실습 10. SIFT 기술자

설명자료

2단계: SIFT 기술자 추출하기

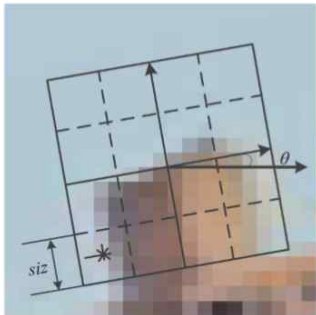


그림 6-4 SIFT 기술자 추출을 위한 좌표계와 4x4 블록



Dominant 방향으로 윈도우를 기울인 후
에지 히스토그램을 계산

실습 10. SIFT 기술자

설명자료

2단계: SIFT 기술자 추출하기

→ 복합 동차 행렬을 이용하여 영상을 중점 회전합니다.

표 2-1 기하 변환을 위한 동차 행렬

변환	동차 행렬 \hat{H}	설명
이동	$T(t_y, t_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_y & t_x & 1 \end{pmatrix}$	y방향으로 t_y , x방향으로 t_x 만큼 이동
회전	$R(\theta) = \begin{pmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$	원점을 중심으로 시계방향으로 θ 만큼 회전
크기	$S(s_y, s_x) = \begin{pmatrix} s_y & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & 1 \end{pmatrix}$	y방향으로 s_y , x방향으로 s_x 만큼 확대
가울임	$Sh_y(h_y) = \begin{pmatrix} 1 & 0 & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, Sh_x(h_x) = \begin{pmatrix} 1 & h_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	Sh_y : y방향으로 h_y 만큼 가울임 Sh_x : x방향으로 h_x 만큼 가울임

실습 10. SIFT 기술자

설명자료

2단계: SIFT 기술자 추출하기

→ 아래 함수를 이용하여 복합 동차 행렬을 계산합니다.

```
CV_EXPORTS_W Mat getRotationMatrix2D( Point2f center, double angle, double scale );
```

center : 중점 - cv::Point로 넘김 ex) cv::Point(centerX, centerY)

angle : 회전 방향

scale : 스케일(실습에서는 1)

실습 10. SIFT 기술자

설명자료

2단계: SIFT 기술자 추출하기

```
CV_EXPORTS_W Mat getRotationMatrix2D( Point2f center, double angle, double scale );
// center : 중심 cv::Point2f로 넘김 ex) cv::Point2f(centerX, centerY)
// angle : 회전 방향
// scale : 스케일 (실습에서는 1)
*/
cv::Mat m = cv::getRotationMatrix2D(cv::Point2f( img.cols / 2, img.rows / 2 ), orientation, 1);

for (int h = beginPoint.y; h < endPoint.y; h += siz)
{
    for (int w = beginPoint.x; w < endPoint.x; w += siz)
    {
        std::vector<float> orient;
        std::vector<float> mag;
        for (int h2 = h; h2 < h + siz; h2++)
        {
            for (int w2 = w; w2 < w + siz; w2++)
            {
                cv::Mat orgIdx(cv::Size(1, 2), CV_32FC1);

                orgIdx.at<float>(0) = h2;
                orgIdx.at<float>(1) = w2;

                // 복합 동차 행렬 계산
                int rotationIdY = int(orgIdx.at<float>(0) * m.at<double>(0) + orgIdx.at<float>(1) * m.at<double>(1) + m.at<double>(2));
                int rotationIdX = int(orgIdx.at<float>(0) * m.at<double>(3) + orgIdx.at<float>(1) * m.at<double>(4) + m.at<double>(5));

                if (rotationIdX > 0 && rotationIdX < img.cols - 1
                    && rotationIdY > 0 && rotationIdY < img.rows - 1)
                {
                    // 픽셀별 dx와 dy를 계산하십시오. (rotationIdY와 rotationIdX를 이용하세요.)
                    float dx = (float)(img.at<uchar>(rotationIdY, rotationIdX + 1) - img.at<uchar>(rotationIdY, rotationIdX - 1));
                    float dy = (float)(img.at<uchar>(rotationIdY + 1, rotationIdX) - img.at<uchar>(rotationIdY - 1, rotationIdX));

                    cv::Mat Mag(cv::Size(1, 1), CV_32FC1); // 에지 크기
                    cv::Mat Angle(cv::Size(1, 1), CV_32FC1); // 에지 방향

                    /*
                     * CV_EXPORTS_W void cartToPolar(InputArray x, InputArray y,
                     * OutputArray magnitude, OutputArray angle,
                     * bool angleInDegrees = false)
                     * x : dx
                     * y : dy
                     * magnitude : 에지 크기
                     * angle : 에지 방향
                     */
                    // ** 지금까지 코드를 작성하세요. 이 줄은 지우시면 안 됩니다 **

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

실습 10. SIFT 기술자

결과영상

