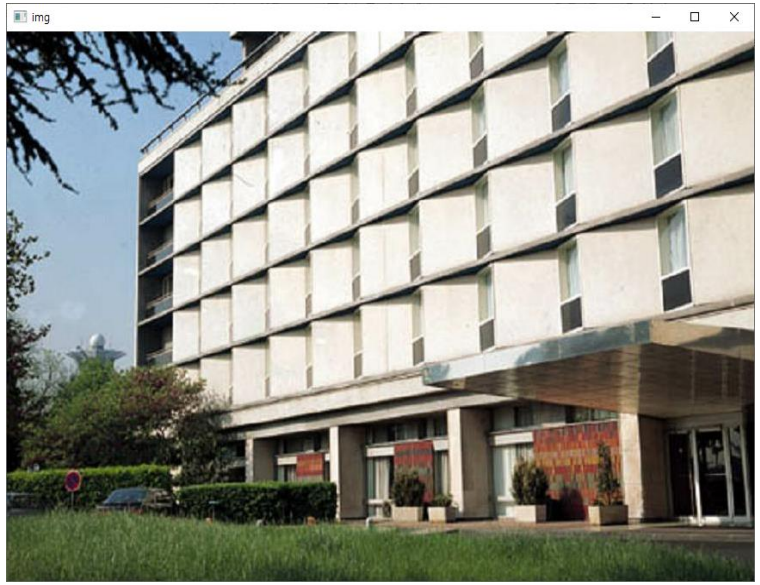
영상처리 실제 - 12주차 과제

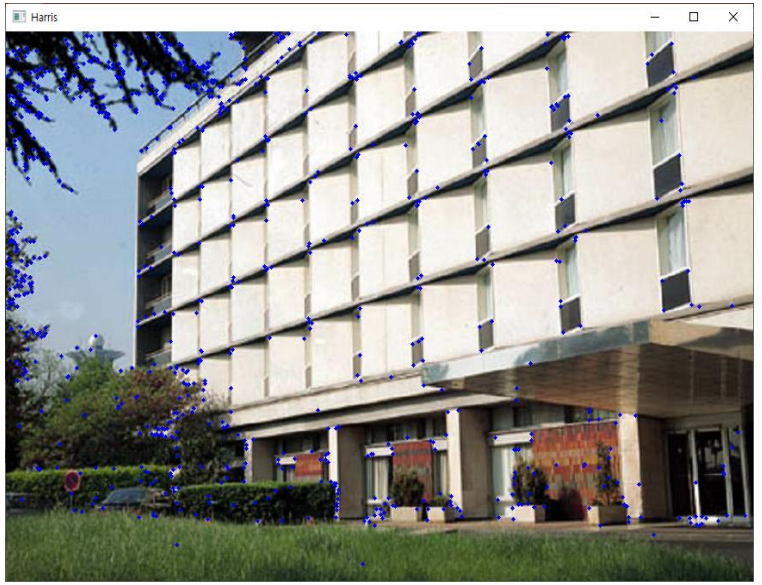
2023254009 최현동

: 17 – 특징추출(2) – HW1

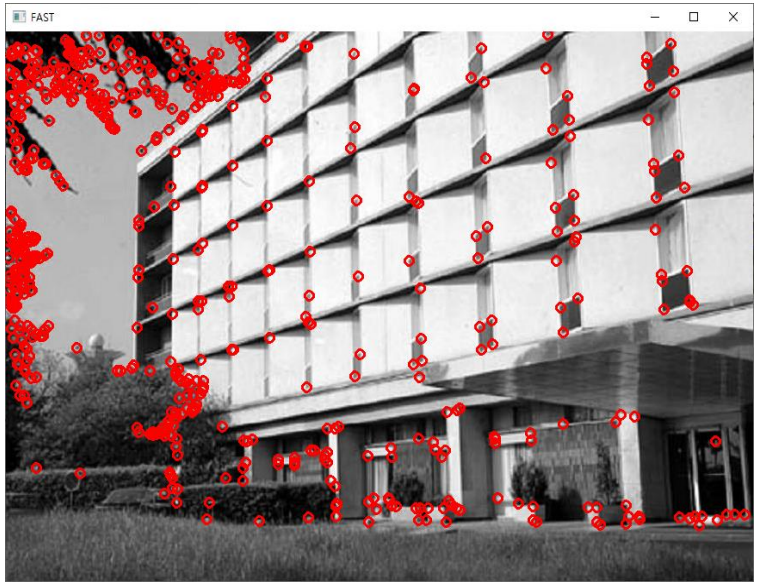
```cpp
//17 - 특징추출(2) - HW1
#if 1
Mat img = imread("D:\\999.Image\\building.jpg", IMREAD_COLOR);
if (img.empty())
{
    cout << "can not open " << endl;
    return -1;
}

//Harris
Mat Harris_gray;
Mat Harris_img;
Mat Harris_Result_img;
int blockSize = 4;
int apertureSize = 3;
double k = 0.04;
int  thresh = 20;

imshow("img", img);

cvtColor(img, Harris_gray, COLOR_BGR2GRAY);

cornerHarris(Harris_gray, Harris_img, blockSize, apertureSize, k);  // OpenCV 제공 함수
Harris_Result_img = draw_coner(Harris_img, img.clone(), thresh);
imshow("Harris", Harris_Result_img);

//FAST
vector<KeyPoint> keypoints_FAST;
Mat FAST_gray;
Mat FAST_Result_img;
cvtColor(img, FAST_gray, COLOR_BGR2GRAY);
FAST(FAST_gray, keypoints_FAST, 60, true);

cvtColor(FAST_gray, FAST_Result_img, COLOR_GRAY2BGR);

for (KeyPoint kp : keypoints_FAST)
{
    Point pt(cvRound(kp.pt.x), cvRound(kp.pt.y));
    circle(FAST_Result_img, pt, 5, Scalar(0, 0, 255), 2);
}
imshow("FAST", FAST_Result_img);

//SIFT
Mat SIFT_gray;
Mat SIFT_desc;
Mat SIFT_Result_img;
Ptr<Feature2D> feature_SIFT = SIFT::create();
vector<KeyPoint> keypoints_SIFT;

cvtColor(img, SIFT_gray, COLOR_BGR2GRAY);
feature_SIFT->detect(SIFT_gray, keypoints_SIFT);
feature_SIFT->compute(SIFT_gray, keypoints_SIFT, SIFT_desc);
drawKeypoints(SIFT_gray, keypoints_SIFT, SIFT_Result_img, Scalar::all(-1), DrawMatchesFlags::DRAW_RICH_KEYPOINTS);
imshow("SIFT", SIFT_Result_img);

//SURF
Mat SURF_gray;
Mat SURF_desc;
Mat SURF_Result_img;
Ptr<Feature2D> feature_SURF = xfeatures2d::SURF::create();
vector<KeyPoint> keypoints_SURF;
cvtColor(img, SURF_gray, COLOR_BGR2GRAY);
feature_SURF->detect(SURF_gray, keypoints_SURF);
feature_SURF->compute(SURF_gray, keypoints_SURF, SURF_desc);
drawKeypoints(SURF_gray, keypoints_SURF, SURF_Result_img, Scalar::all(-1), DrawMatchesFlags::DRAW_RICH_KEYPOINTS);
imshow("SURF", SURF_Result_img);

//ORB
Mat ORB_gray;
Mat ORB_desc;
Mat ORB_Result_img;
Ptr<Feature2D> feature_ORB = ORB::create();
vector<KeyPoint> keypoints_ORB;
cvtColor(img, ORB_gray, COLOR_BGR2GRAY);

feature_ORB->detect(ORB_gray, keypoints_ORB);
feature_ORB->compute(ORB_gray, keypoints_ORB, ORB_desc);

drawKeypoints(ORB_gray, keypoints_ORB, ORB_Result_img, Scalar::all(-1), DrawMatchesFlags::DRAW_RICH_KEYPOINTS);
imshow("ORB", ORB_Result_img);

waitKey();
#endif
```



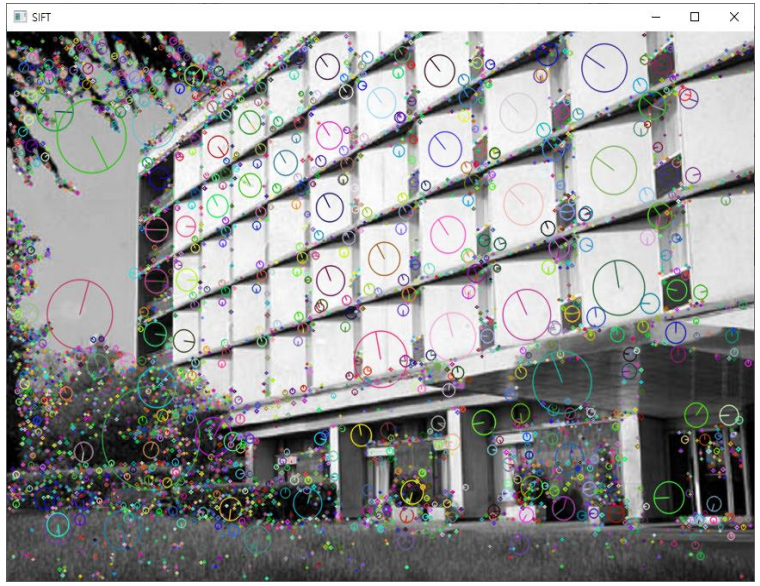
< source img>



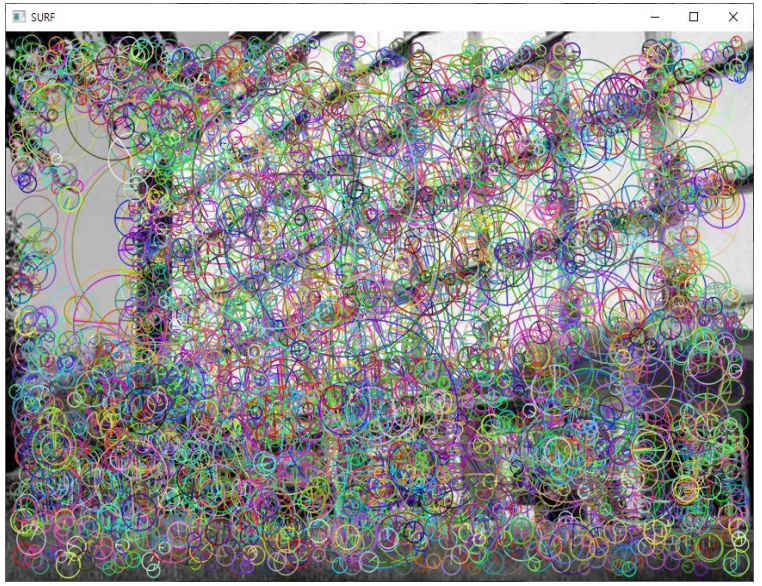
< Harris >



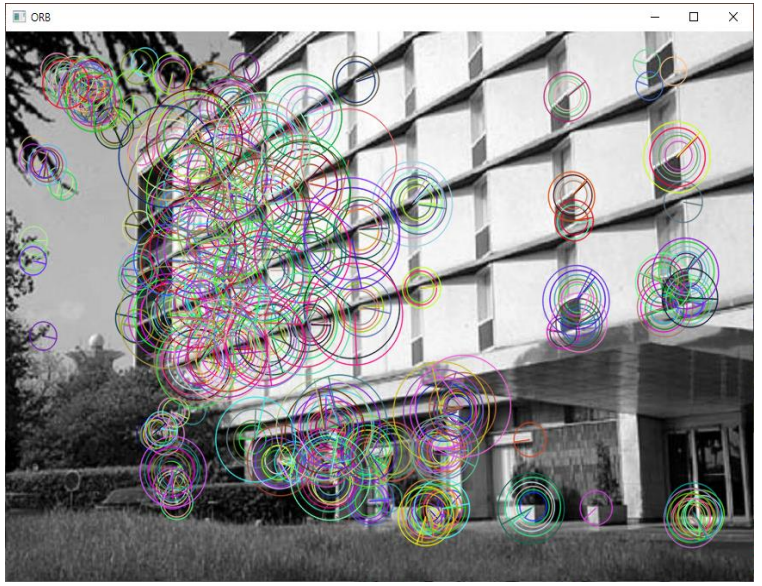
< FAST >



< SIFT >



< SURF >



< ORB >