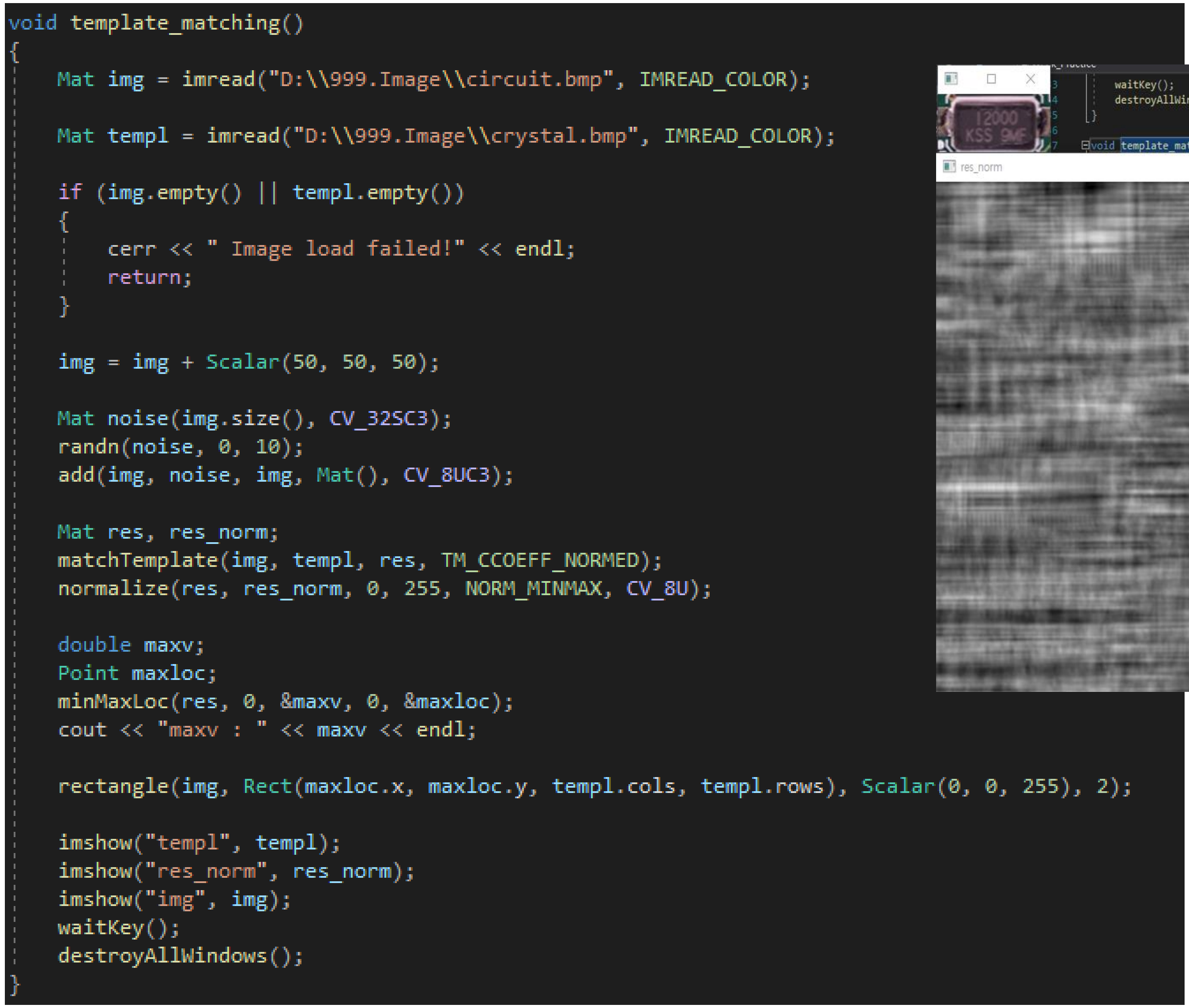
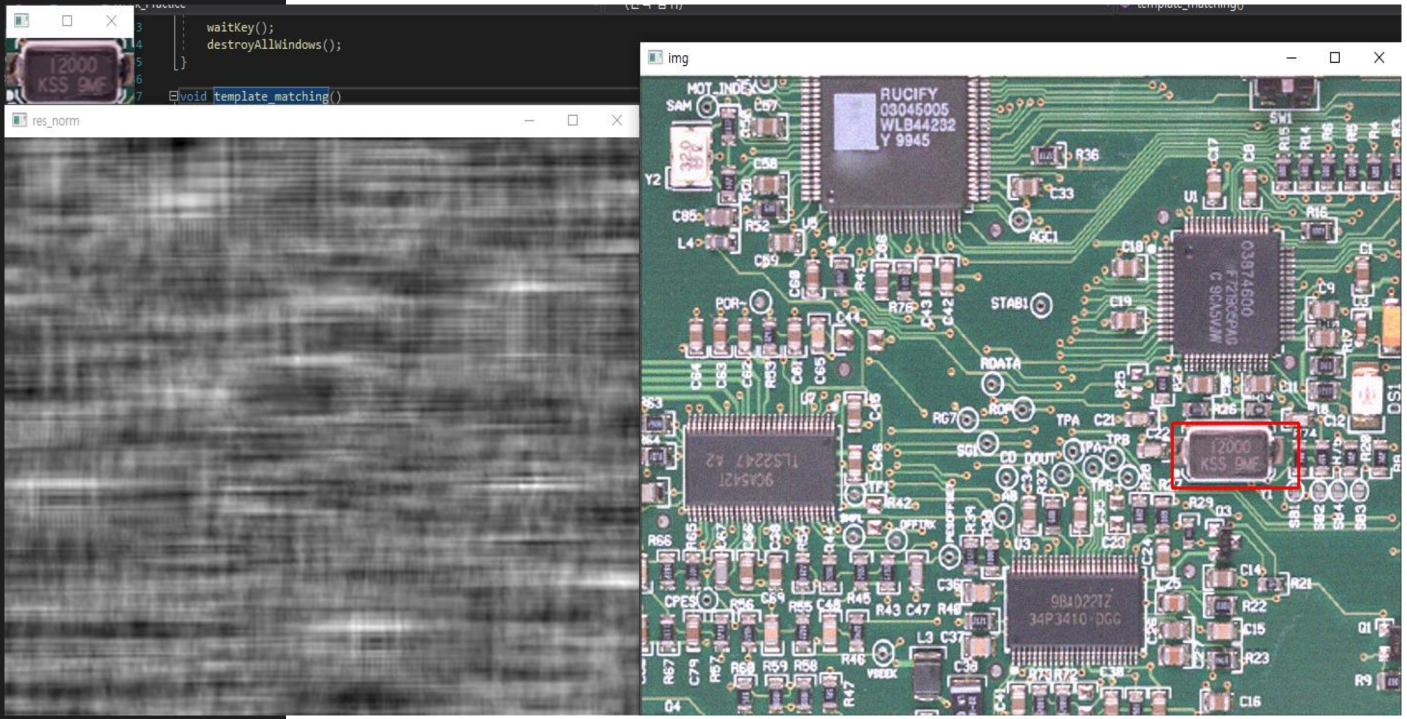
영상처리 실제 - 13주차 실습

2023254009 최현동

: 18.특징매칭 – p.12

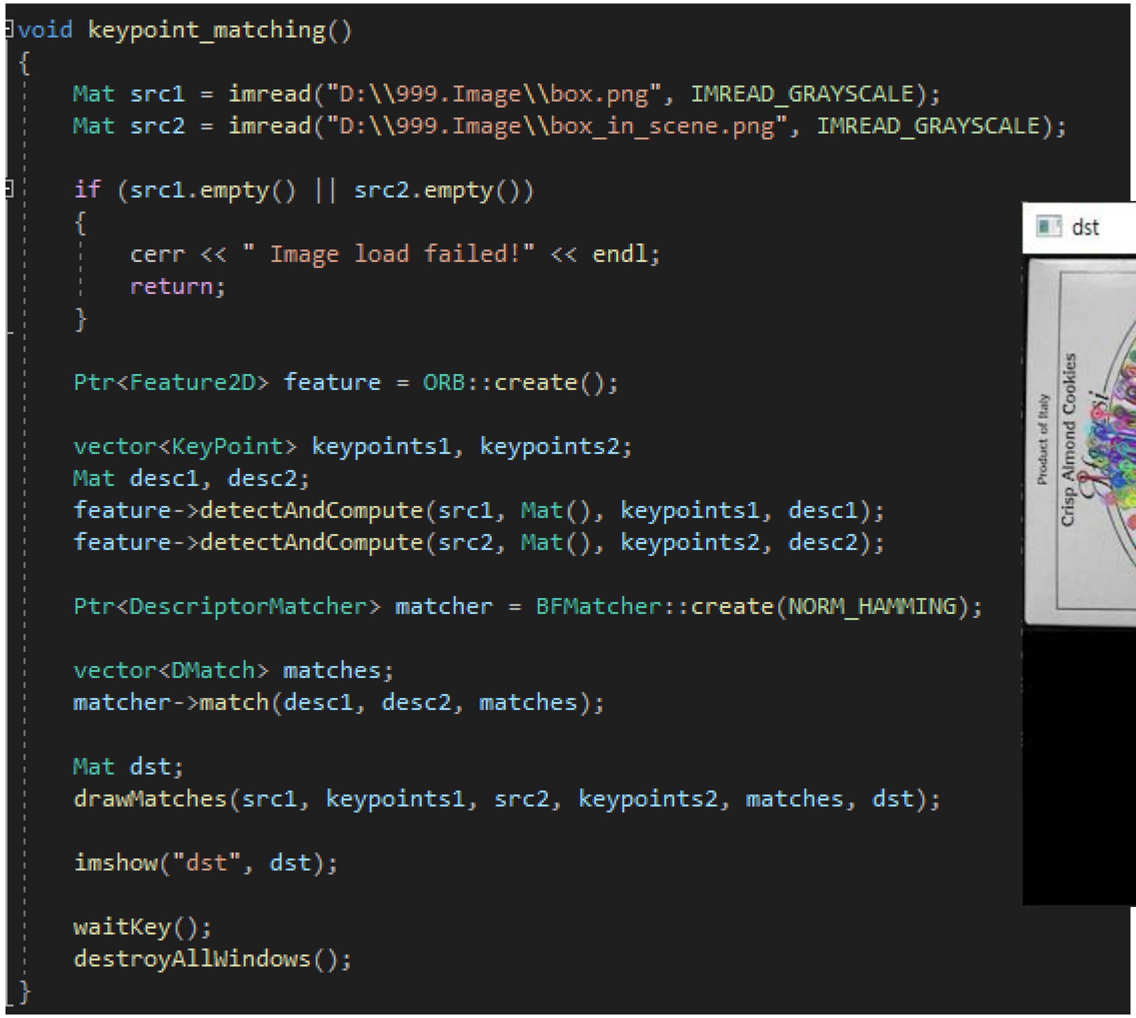


```cpp
void template_matching()
{
    Mat img = imread("D:\\999.Image\\circuit.bmp", IMREAD_COLOR);

    Mat templ = imread("D:\\999.Image\\crystal.bmp", IMREAD_COLOR);

    if (img.empty() || templ.empty())
    {
        cerr << " Image load failed!" << endl;
        return;
    }

    img = img + Scalar(50, 50, 50);

    Mat noise(img.size(), CV_32SC3);
    randn(noise, 0, 10);
    add(img, noise, img, Mat(), CV_8UC3);

    Mat res, res_norm;
    matchTemplate(img, templ, res, TM_CCOEFF_NORMED);
    normalize(res, res_norm, 0, 255, NORM_MINMAX, CV_8U);

    double maxv;
    Point maxloc;
    minMaxLoc(res, 0, &maxv, 0, &maxloc);
    cout << "maxv : " << maxv << endl;

    rectangle(img, Rect(maxloc.x, maxloc.y, templ.cols, templ.rows), Scalar(0, 0, 255), 2);

    imshow("templ", templ);
    imshow("res_norm", res_norm);
    imshow("img", img);
    waitKey();
    destroyAllWindows();
}
```

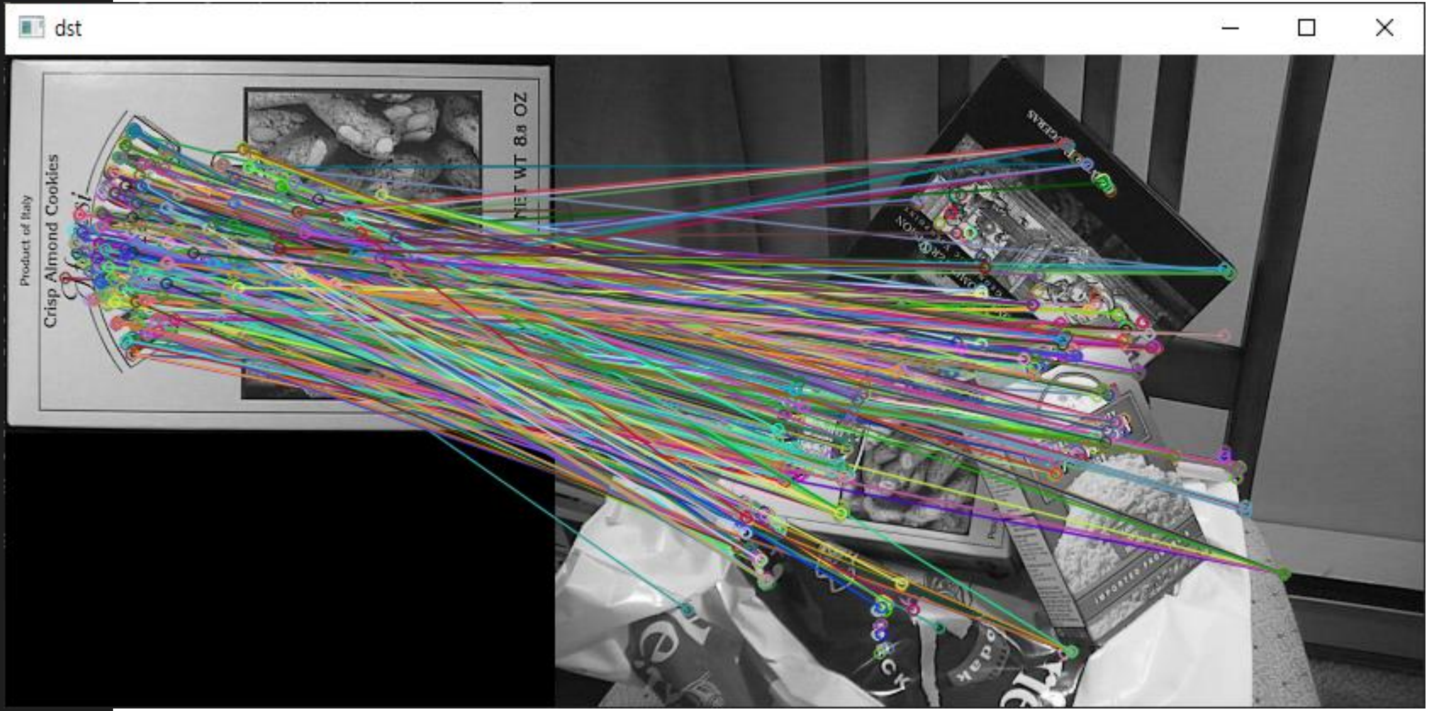


```cpp
    waitKey();
    destroyAllWindows();
}

void template_matching()
```

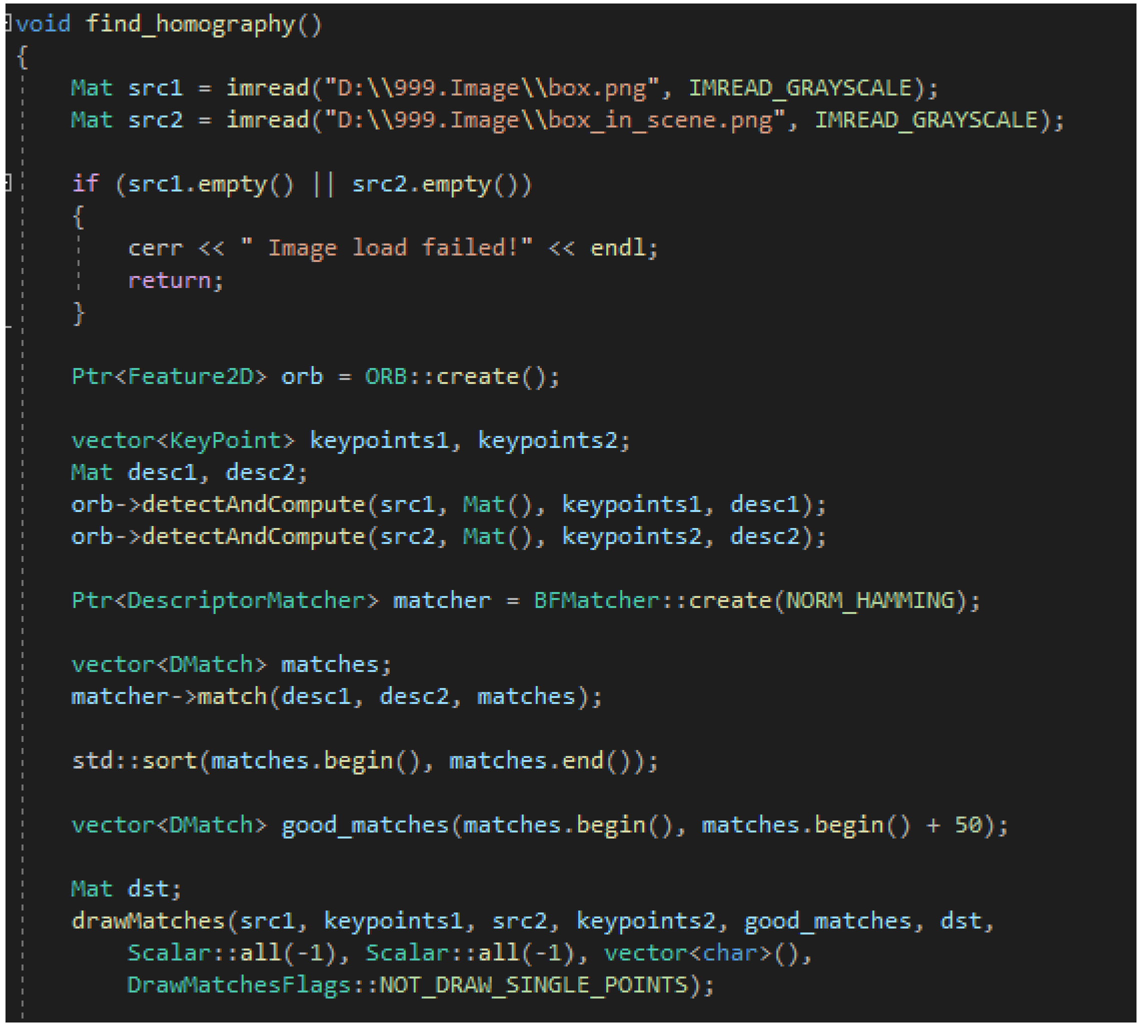
: 18.특징매칭 – p.29



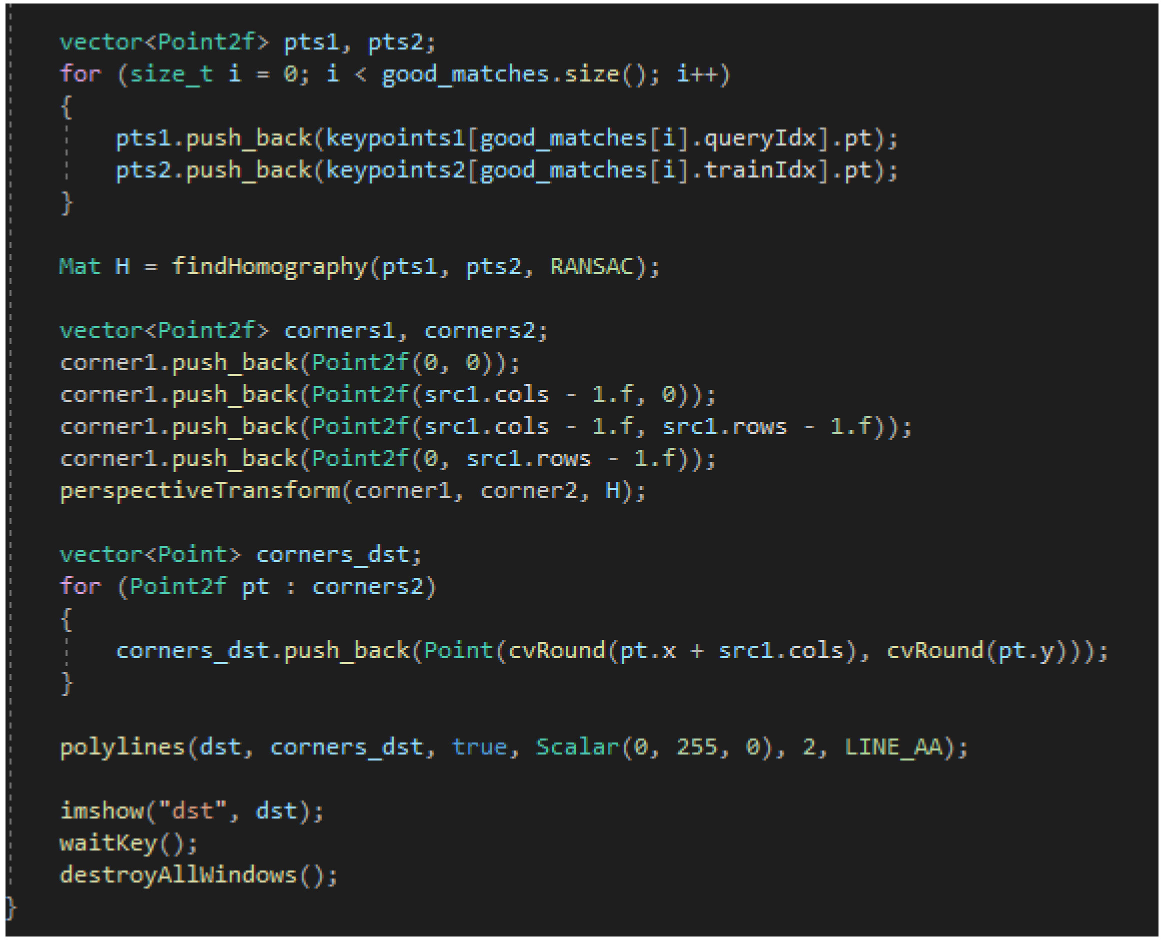
```
void keypoint_matching()
{
    Mat src1 = imread("D:\\999.Image\\box.png", IMREAD_GRAYSCALE);
    Mat src2 = imread("D:\\999.Image\\box_in_scene.png", IMREAD_GRAYSCALE);

    if (src1.empty() || src2.empty())
    {
        cerr << " Image load failed!" << endl;
        return;
    }

    Ptr<Feature2D> feature = ORB::create();

    vector<KeyPoint> keypoints1, keypoints2;
    Mat desc1, desc2;
    feature->detectAndCompute(src1, Mat(), keypoints1, desc1);
    feature->detectAndCompute(src2, Mat(), keypoints2, desc2);

    Ptr<DescriptorMatcher> matcher = BFMatcher::create(NORM_HAMMING);

    vector<DMatch> matches;
    matcher->match(desc1, desc2, matches);

    Mat dst;
    drawMatches(src1, keypoints1, src2, keypoints2, matches, dst);

    imshow("dst", dst);

    waitKey();
    destroyAllWindows();
}
```



: 18.특징매칭 – p.39 ~ 41

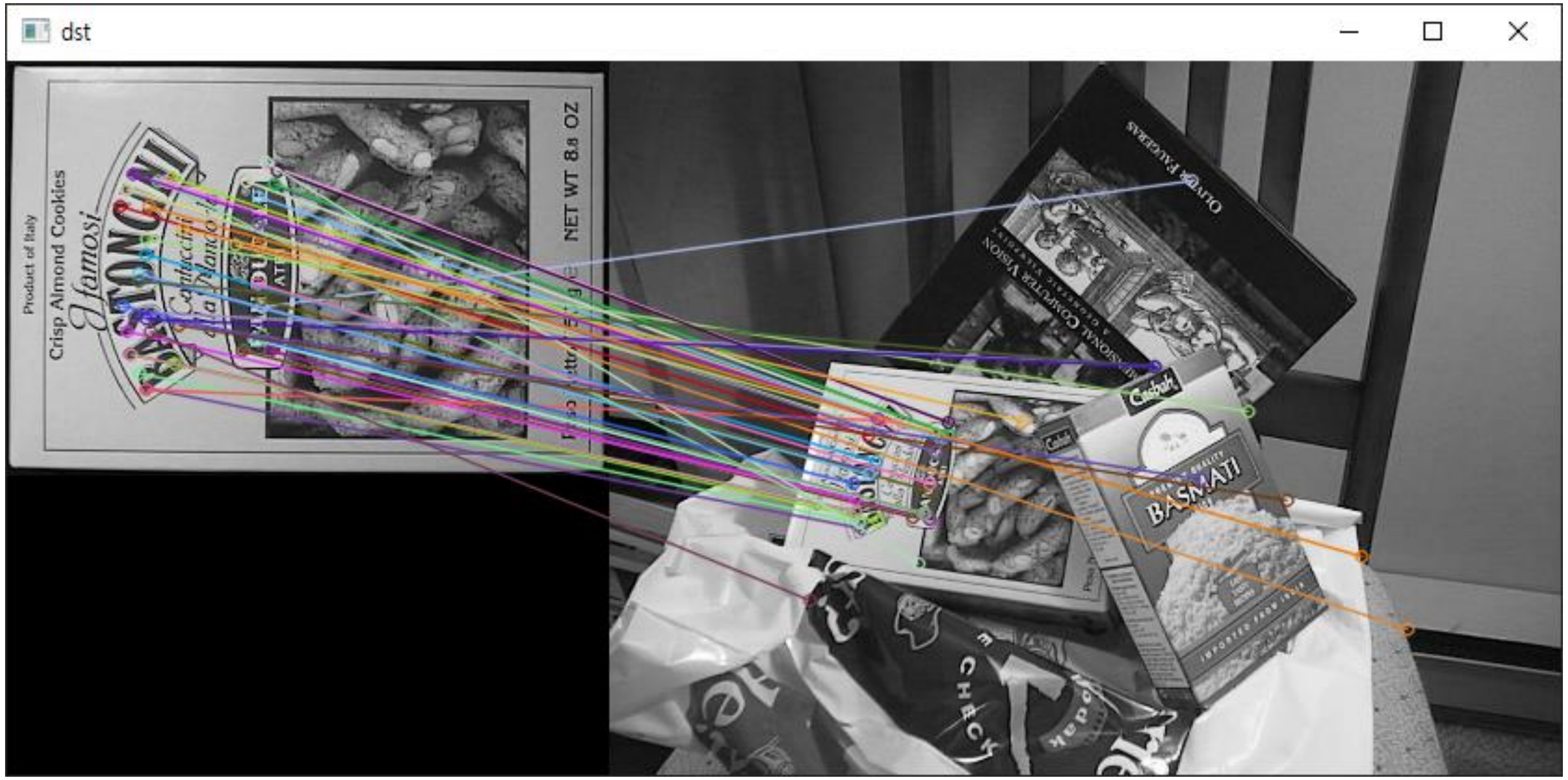


```
void find_homography()
{
    Mat src1 = imread("D:\\999.Image\\box.png", IMREAD_GRAYSCALE);
    Mat src2 = imread("D:\\999.Image\\box_in_scene.png", IMREAD_GRAYSCALE);

    if (src1.empty() || src2.empty())
    {
        cerr << " Image load failed!" << endl;
        return;
    }

    Ptr<Feature2D> orb = ORB::create();

    vector<KeyPoint> keypoints1, keypoints2;
    Mat desc1, desc2;
    orb->detectAndCompute(src1, Mat(), keypoints1, desc1);
    orb->detectAndCompute(src2, Mat(), keypoints2, desc2);

    Ptr<DescriptorMatcher> matcher = BFMatcher::create(NORM_HAMMING);

    vector<DMatch> matches;
    matcher->match(desc1, desc2, matches);

    std::sort(matches.begin(), matches.end());

    vector<DMatch> good_matches(matches.begin(), matches.begin() + 50);

    Mat dst;
    drawMatches(src1, keypoints1, src2, keypoints2, good_matches, dst,
        Scalar::all(-1), Scalar::all(-1), vector<char>(),
        DrawMatchesFlags::NOT_DRAW_SINGLE_POINTS);
```



```
    vector<Point2f> pts1, pts2;
    for (size_t i = 0; i < good_matches.size(); i++)
    {
        pts1.push_back(keypoints1[good_matches[i].queryIdx].pt);
        pts2.push_back(keypoints2[good_matches[i].trainIdx].pt);
    }

    Mat H = findHomography(pts1, pts2, RANSAC);

    vector<Point2f> corners1, corners2;
    corner1.push_back(Point2f(0, 0));
    corner1.push_back(Point2f(src1.cols - 1.f, 0));
    corner1.push_back(Point2f(src1.cols - 1.f, src1.rows - 1.f));
    corner1.push_back(Point2f(0, src1.rows - 1.f));
    perspectiveTransform(corner1, corner2, H);

    vector<Point> corners_dst;
    for (Point2f pt : corners2)
    {
        corners_dst.push_back(Point(cvRound(pt.x + src1.cols), cvRound(pt.y)));
    }

    polylines(dst, corners_dst, true, Scalar(0, 255, 0), 2, LINE_AA);

    imshow("dst", dst);
    waitKey();
    destroyAllWindows();
}
```

: 18.특징매칭 – p.39 ~ 41


dst
Product of Italy
Crisp Almond Cookies
Bastongini
I famosi
Cantuccini
NET WT 8.8 OZ
BASMATI
Olivier Faugeras
Computer Vision