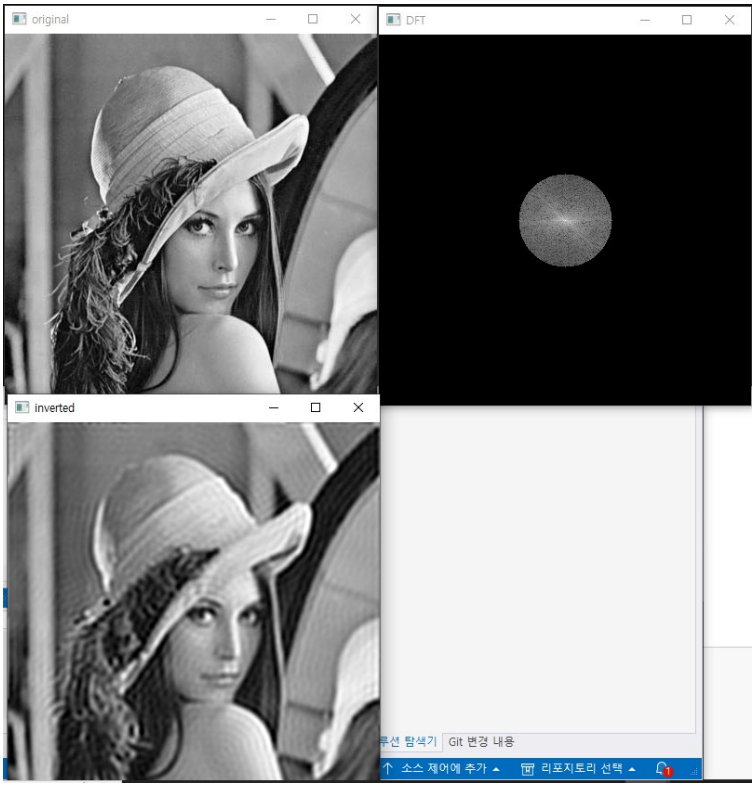
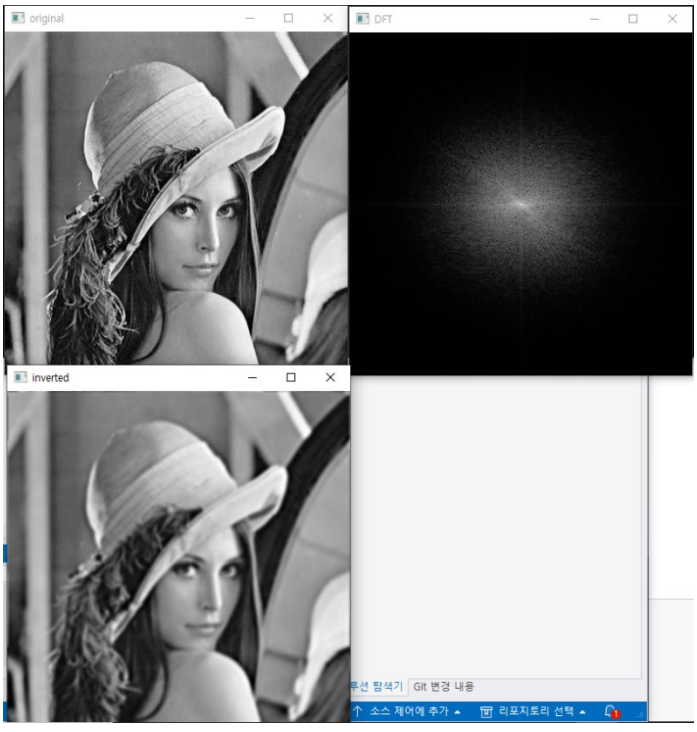
14-주파수영역 처리.pdf의 페이지 18page

```cpp
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;

void displayDFT(Mat& src) { ... }

void shuffleDFT(Mat& src) { ... }

// 원형 필터를 만든다.
Mat getFilter(Size size)
{
    Mat filter(size, CV_32FC2, Vec2f(0, 0));
    circle(filter, size / 2, 50, Vec2f(1, 1), -1);
    return filter;
}
int main()
{
    Mat src = imread("C:/Users/Chan's Victus/Documents/class/Project/image/lenna.jp
    Mat src_float;
    imshow("original", src);
    // 그레이스케일 영상을 실수 영상으로 변환한다.
    src.convertTo(src_float, CV_32FC1, 1.0 / 255.0);
    Mat dft_image;
    dft(src_float, dft_image, DFT_COMPLEX_OUTPUT);
    shuffleDFT(dft_image);
    Mat lowpass = getFilter(dft_image.size());
    Mat result;
    // 원형 필터와 DFT 영상을 서로 곱한다.
    multiply(dft_image, lowpass, result);
    displayDFT(result);
    Mat inverted_image;
    shuffleDFT(result);
    idft(result, inverted_image, DFT_SCALE | DFT_REAL_OUTPUT);
    imshow("inverted", inverted_image);
    waitKey(0);
    return 1;
}
```

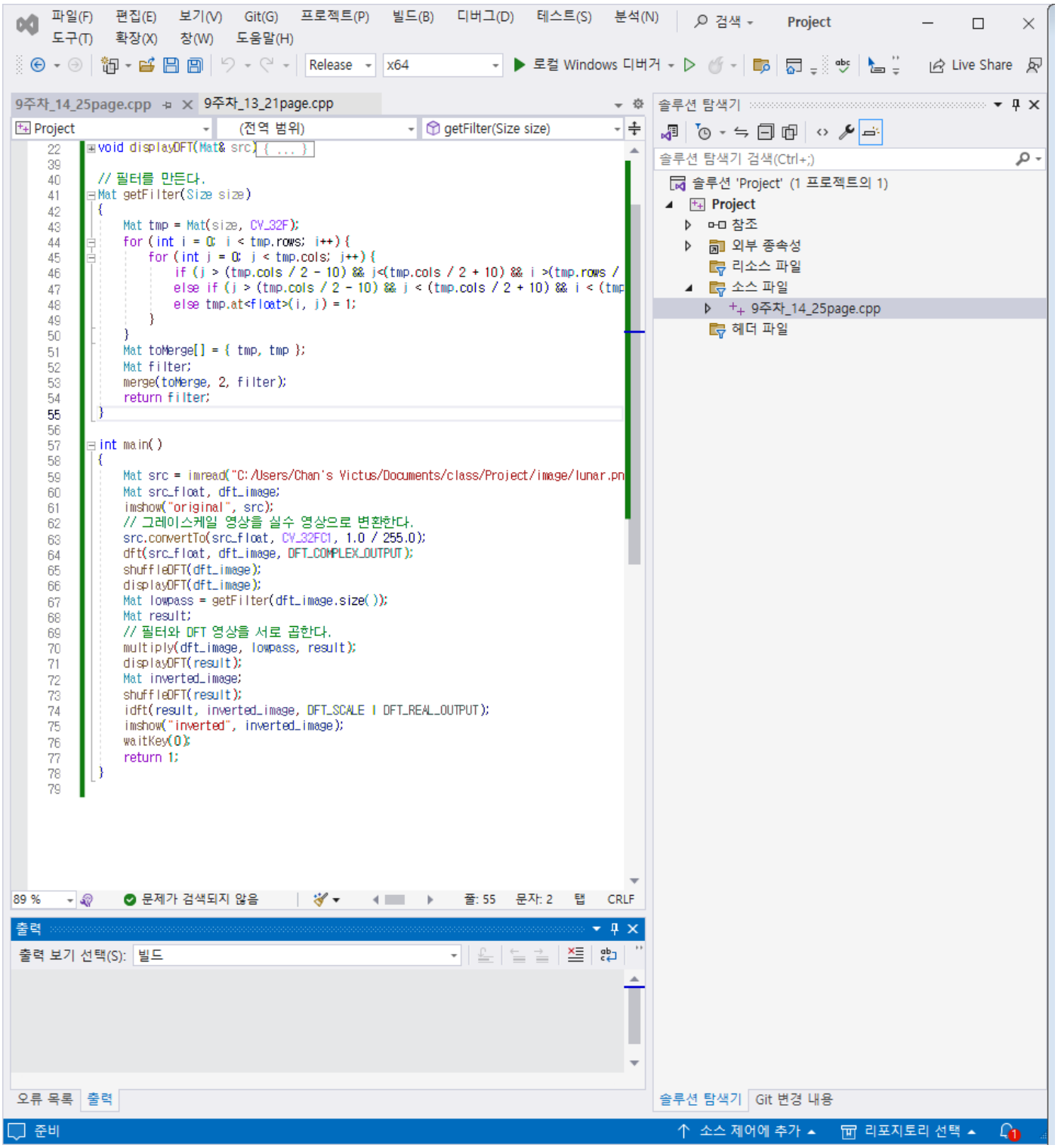


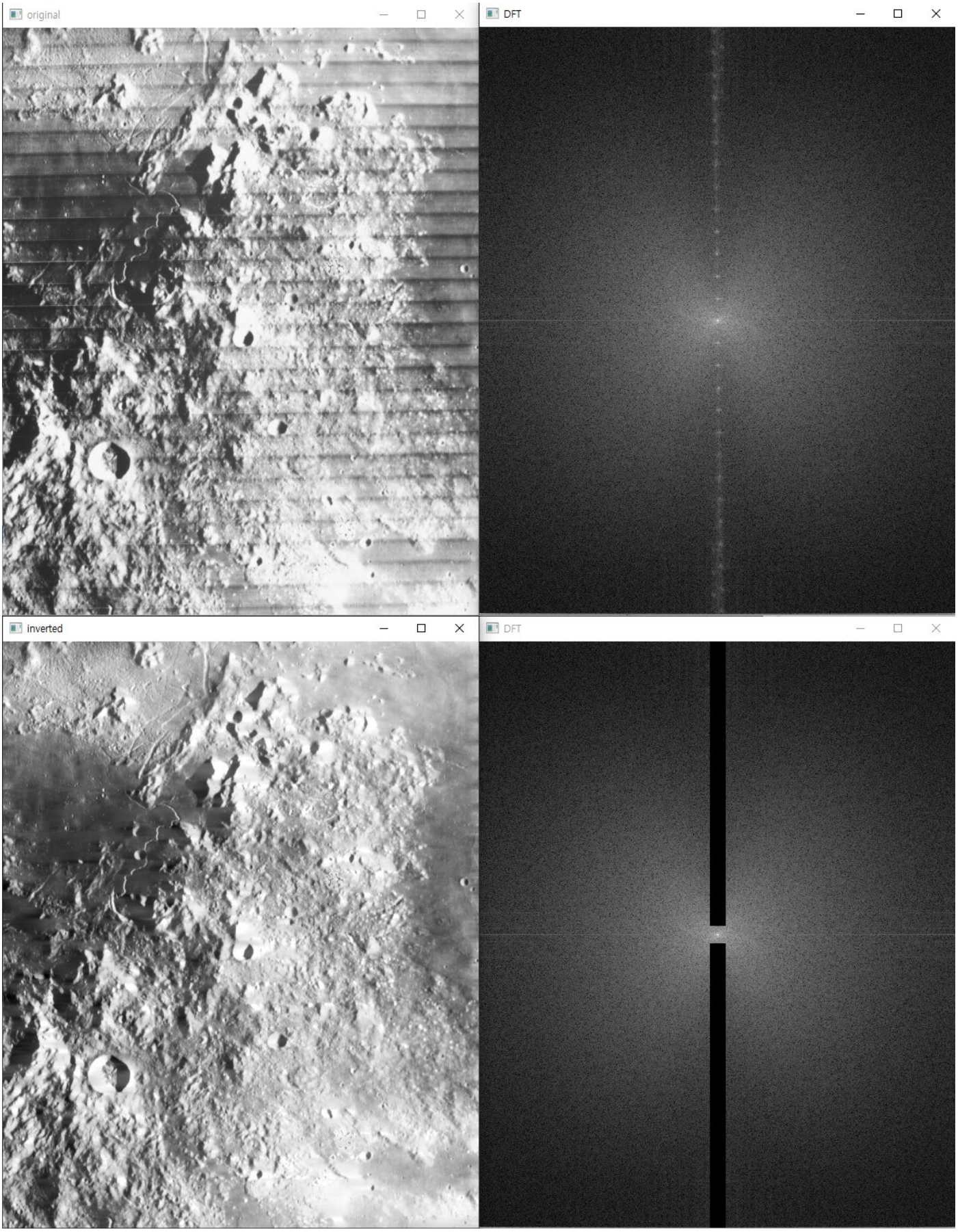
14-주파수영역 처리.pdf의 페이지 21page

```
void shuffleDFT(Mat& src) { ... }

void displayDFT(Mat& src) { ... }

// 버터워쓰 필터를 만든다.
Mat getFilter(Size size)
{
    Mat tmp = Mat(size, CV_32F);
    Point center = Point(tmp.rows / 2, tmp.cols / 2);
    double radius;
    double D = 50;
    double n = 2;
    for (int i = 0; i < tmp.rows; i++) {
        for (int j = 0; j < tmp.cols; j++) {
            radius = (double)sqrt(pow((i - center.x), 2.0) + pow((double)(j - cente
            tmp.at<float>(i, j) = (float)(1 / (1 + pow((double)(radius / D), (doubl
        }
    }
    Mat toMerge[] = { tmp, tmp };
    Mat filter;
    merge(toMerge, 2, filter);
    return filter;
}

int main()
{
    Mat src = imread("C:/Users/Chan's Victus/Documents/class/Project/image/lenna.jp
    Mat src_float;
    imshow("original", src);
    // 그레이스케일 영상을 실수 영상으로 변환한다.
    src.convertTo(src_float, CV_32FC1, 1.0 / 255.0);
    Mat dft_image;
    dft(src_float, dft_image, DFT_COMPLEX_OUTPUT);
    shuffleDFT(dft_image);
    Mat lowpass = getFilter(dft_image.size());
    Mat result;
    // 원형 필터와 DFT 영상을 서로 곱한다.
    multiply(dft_image, lowpass, result);
    displayDFT(result);
    Mat inverted_image;
    shuffleDFT(result);
    idft(result, inverted_image, DFT_SCALE | DFT_REAL_OUTPUT);
    imshow("inverted", inverted_image);
    waitKey(0);
    return 1;
}
```

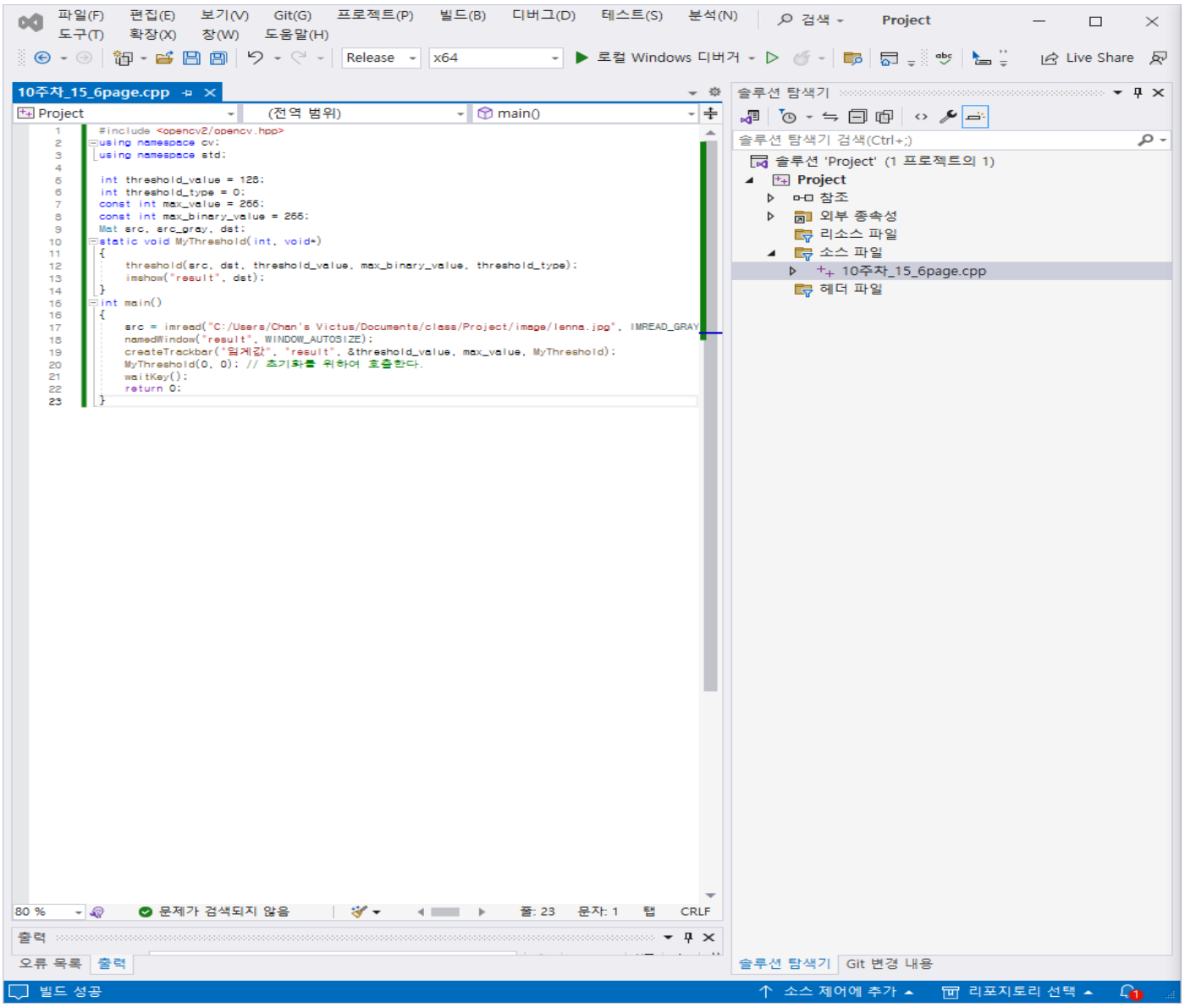


14-주파수영역 처리.pdf의 페이지 25page




original
DFT
inverted
DFT

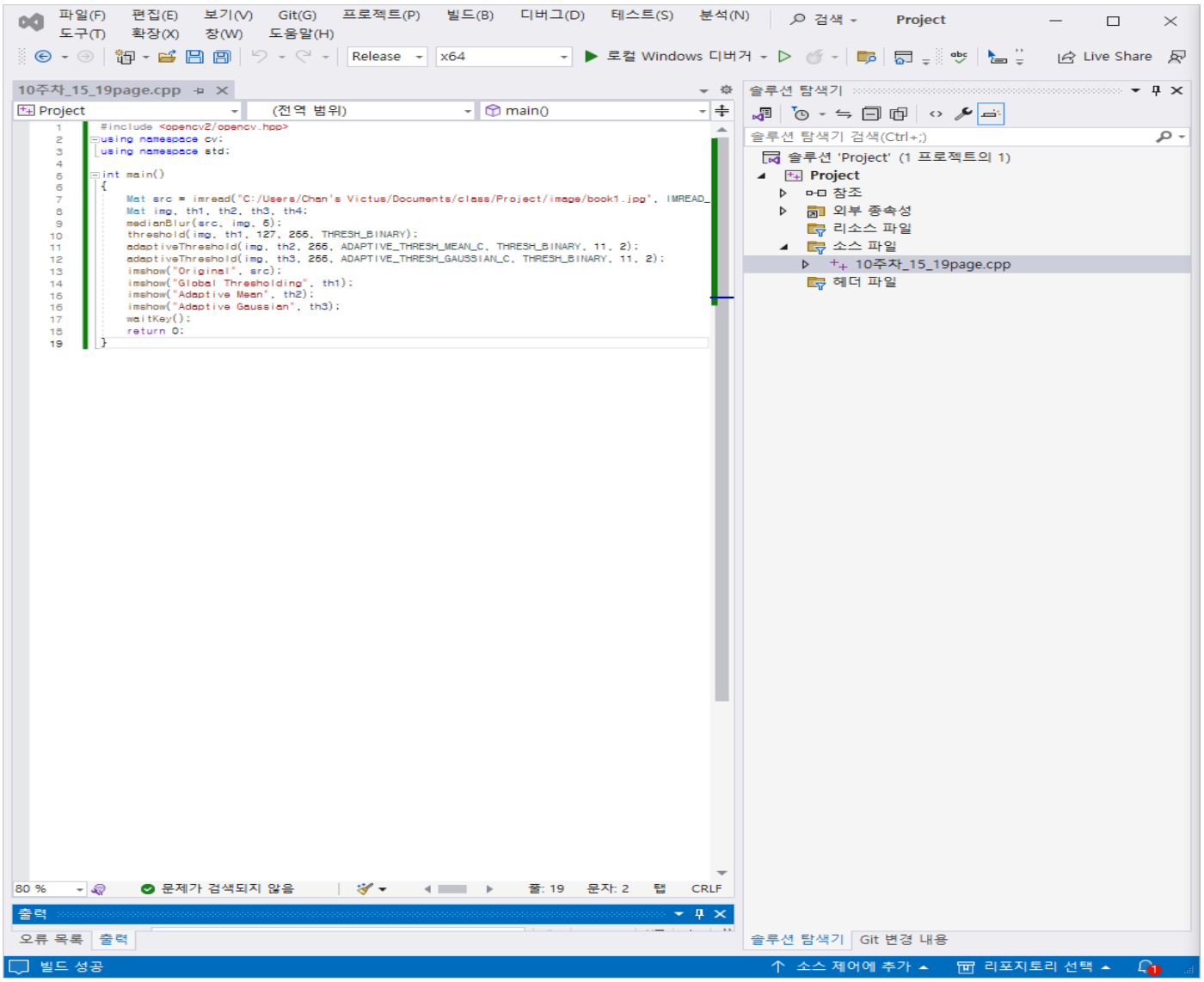
15-영상분할.pdf 6page

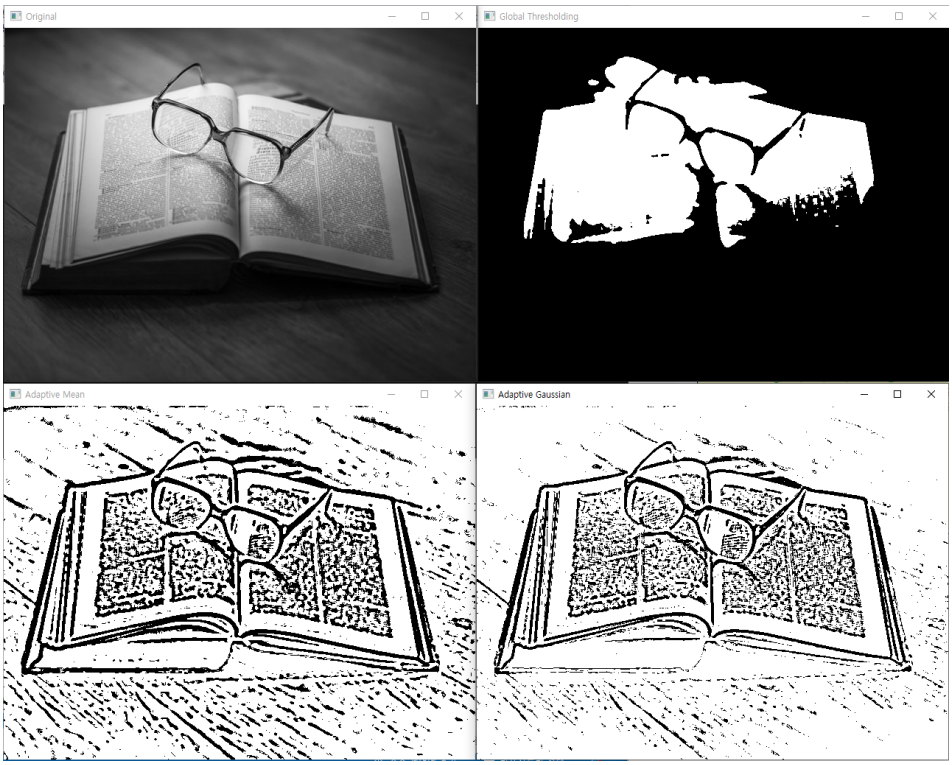


15-영상분할.pdf 13page

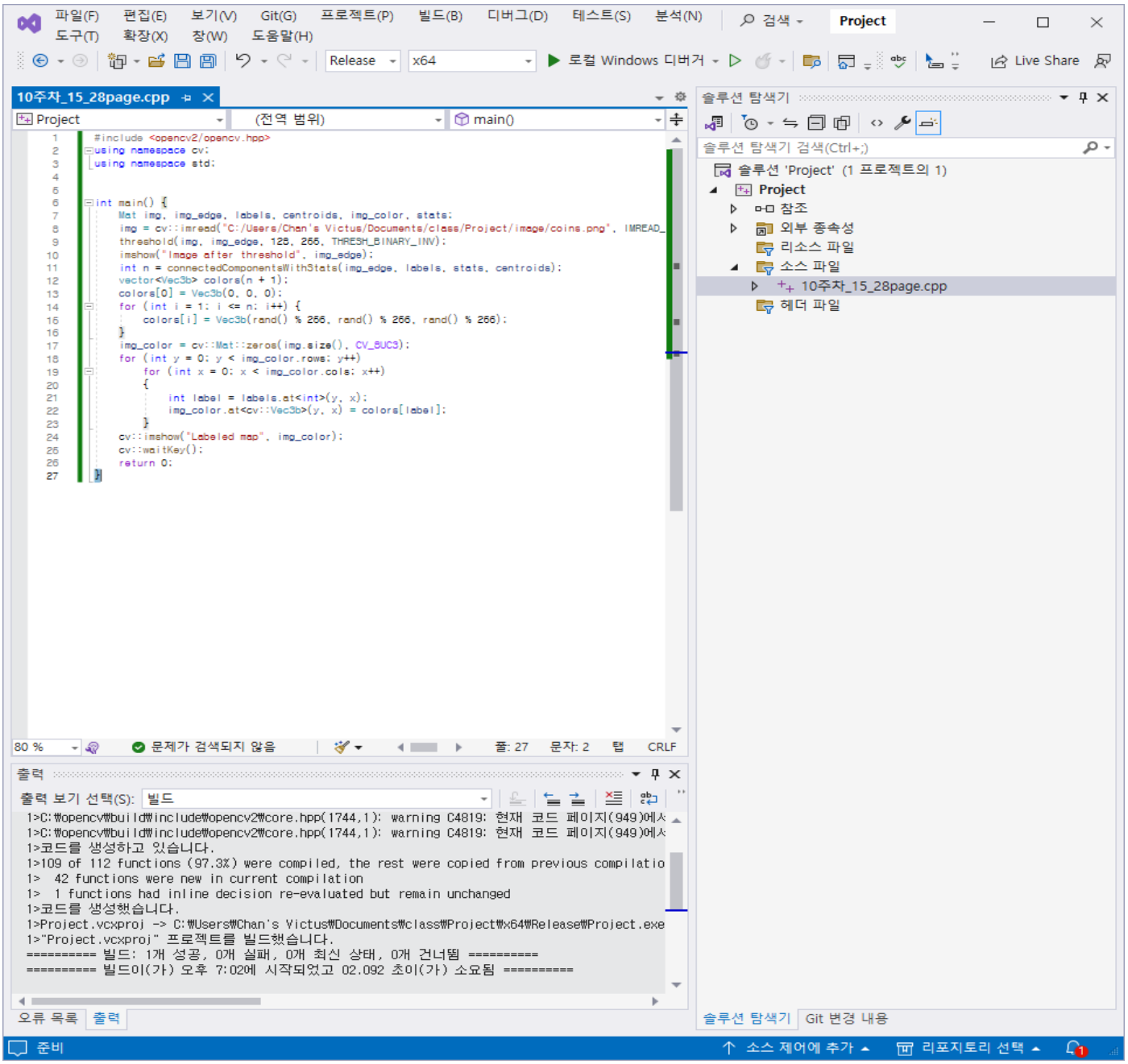
```cpp
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;

int main()
{
    Mat src = imread("C:/Users/Chan's Victus/Documents/class/Project/image/lenna.jpg", IMREAD_
    Mat blur, th1, th2, th3, th4;
    threshold(src, th1, 127, 255, THRESH_BINARY);
    threshold(src, th2, 0, 255, THRESH_BINARY | THRESH_OTSU);
    Size size = Size(5, 5);
    GaussianBlur(src, blur, size, 0);
    threshold(blur, th3, 0, 255, THRESH_BINARY | THRESH_OTSU);
    imshow("Original", src);
    imshow("Global", th1);
    imshow("Ostu", th2);
    imshow("Ostu after Blurring", th3);
    waitKey();
    return 0;
}
```

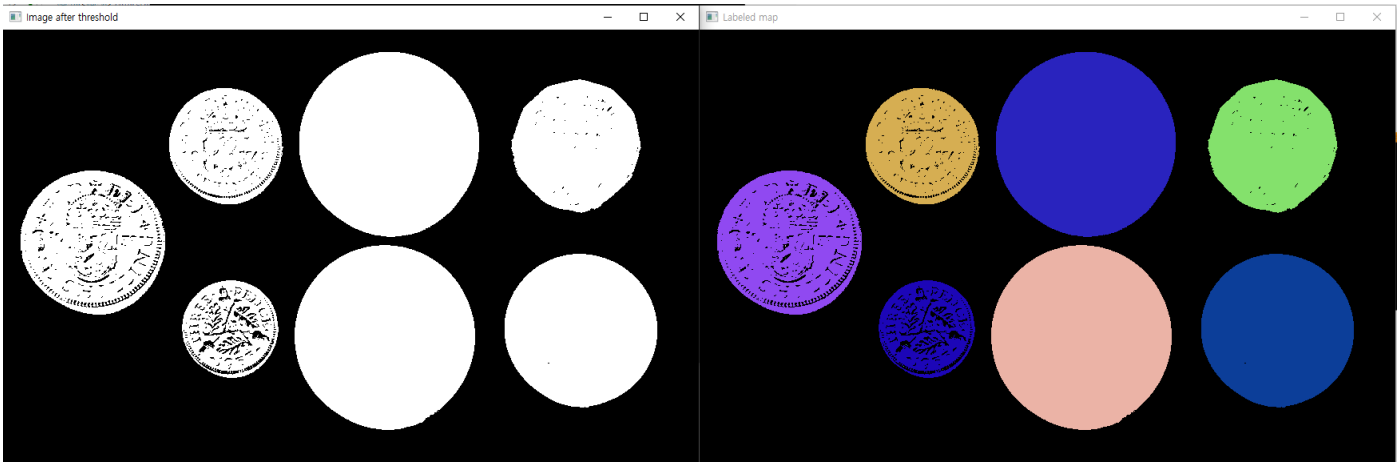
15-영상분할.pdf 19page





15-영상분할.pdf 28page

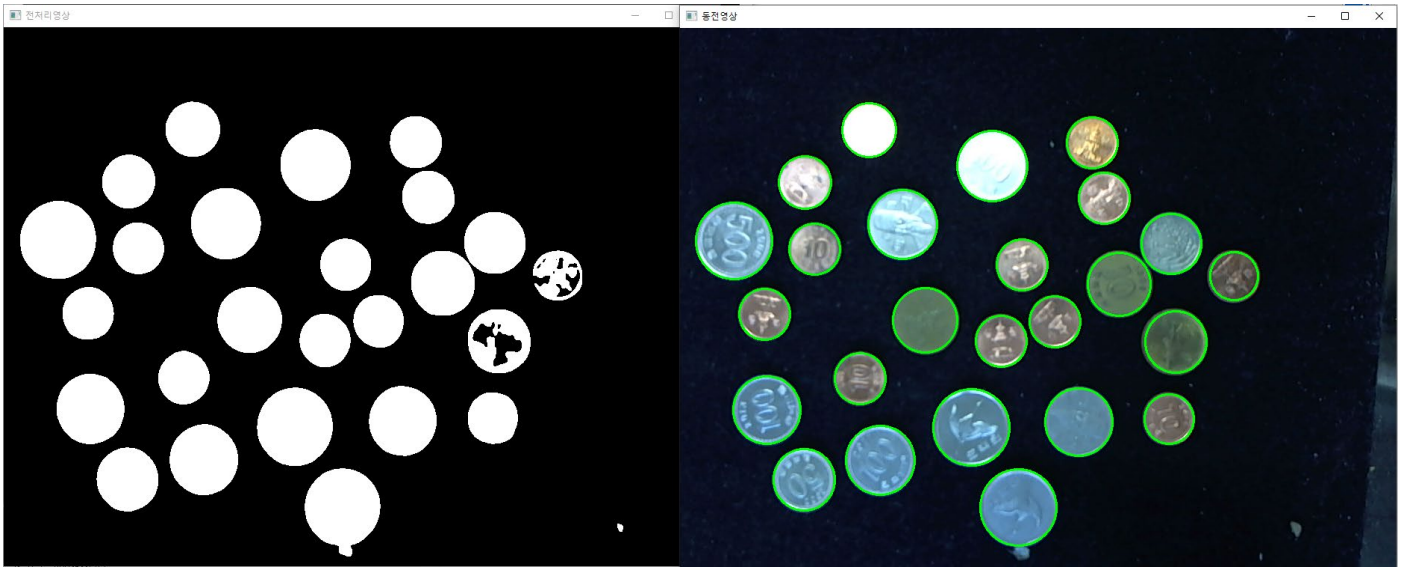




15-영상분할.pdf 34page

```cpp
#include <opencv2/opencv.hpp>
using namespace cv;
using namespace std;

Mat preprocessing(Mat img)
{
    Mat gray, th_img;
    cvtColor(img, gray, COLOR_BGR2GRAY); // 명암도 변환
    GaussianBlur(gray, gray, Size (7, 7), 2, 2); // 블러링
    threshold(gray, th_img, 130, 255, THRESH_BINARY | THRESH_OTSU); // 이진화
    morphologyEx(th_img, th_img, MORPH_OPEN, Mat(), Point(-1, -1), 1); // 열림연산
    return th_img;
}

vector<RotatedRect> find_coins(Mat img)
{
    vector<vector<Point> > contours;
    findContours(img.clone(), contours, RETR_EXTERNAL, CHAIN_APPROX_SIMPLE);

    vector<RotatedRect> circles;
    for (int i = 0; i< (int)contours.size(); i++)
    {
        RotatedRect mr = minAreaRect(contours[i]);
        mr.angle = (mr.size.width + mr.size.height) / 4.0f;

        if (mr.angle > 18) circles.push_back(mr);
    }
return circles;
}

int main()
{
    int coin_no = 20;
    String frame = format("C:/Users/Chan's Victus/Documents/class/Project/image/coin
    Mat image = imread(frame, 1);
    CV_Assert(image.data);
    Mat th_img = preprocessing(image);
    vector<RotatedRect > circles = find_coins(th_img);

    for (int i = 0; i < circles.size(); i++) {
        float radius = circles[i].angle; // 동전객체 반지름
        circle(image, circles[i].center, radius, Scalar(0, 255, 0), 2); // 동전 표시
    }
    imshow("전처리영상", th_img);
    imshow("동전영상", image);
    waitKey();
    return 0;
}
```

```
1>  0 functions were new in current compilation
1>  0 functions had inline decision re-evaluated but remain unchanged
1>코드를 생성했습니다.
1>Project.vcxproj -> C:\Users\Chan's Victus\Documents\class\Project\x64\Release\Project.exe
1>"Project.vcxproj" 프로젝트를 빌드했습니다.
========== 빌드: 1개 성공, 0개 실패, 0개 최신 상태, 0개 건너뜀 ==========
========== 빌드이(가) 오후 7:49에 시작되었고 01.817 초이(가) 소요됨 ==========
```



15-영상분할.pdf 39page