Special Lecture for Computer Science – COSE490

Digital Image Processing :: Harris Corner Detector

Due date : 2022-12-11

고려대학교 컴퓨터학과 2017320108

고재영

개발 환경 : Matlab Desktop

과제 만기일 : 2022-12-11

제출 날짜 : 2022-12-02

최종 제출: 2022-12-02

재제출 사유: none

[0] – 과제 설명

- In this assignment, you will implement the Harris corner detector.

- As we discussed in Lecture 21, you need to build a Harris matrix and compute R value.

- Once you implement the Harris corner detector, then you should visualize the results as described below.

Visualizing R values, Visualizing corner points

- Run the experiment with various combination of k and R threshold on different testing images and discuss the result in the report.

[1] – harris_corner.m

먼저 matlab으로 구현한 harris_corner.m에 대한 코드 설명을 하고자 한다. 아래 내용은 harris_corner.m에 대한 전체 코드이고, 코드의 주요 영역에 대해 설명하고자 한다.

```
%
% Skeleton code for COSE490 Fall 2022 Assignment 4
%
% Won-Ki Jeong (wkjeong@korea.ac.kr)
%

clear all;
close all;

%
% Loading input image
%
I=imread('building-600by600.tif');
%I=imread('checkerboard-noisy2.tif');
Img=double(I(:,:,1));

%
% ToDo: Compute R
%

% Compute gradients
[Ix Iy] = gradient(Img);
% Apply Gaussian smoothing
sigma = 2;
IxIx = imgaussfilt(Ix.*Ix, sigma);
IxIy = imgaussfilt(Ix.*Iy, sigma);
IyIy = imgaussfilt(Iy.*Iy, sigma);
% input image size: height and width
[img_h, img_w] = size(Img);
% build matrix R with k value
R = zeros(size(Img));
k = 0.05;

% Compute Harris matrix H and corner response function R
for i=1:img_h
    for j=1:img_w
        H = [IxIx(i,j) IxIy(i,j); IxIy(i,j) IyIy(i,j)];
        valueR = det(H) - (k*((trace(H)).^2));
        R(i,j) = valueR;
    end
end

%
% Example of collecting points and plot them
%
% (10,1), (15,2), (20,3)
%
%{
location = [10 15 20; 1 2 3]'
```

```
points = cornerPoints(location)
plot(points)
%}

%
% ToDo: Visualize R values using jet colormap
%

% normalize values into [0, 255] for indexing
minval = min(R, [], 'all');
indexR = R - minval;
maxval = max(indexR, [], 'all');
indexR = indexR./maxval;
indexR = indexR.*255;
indexR = round(indexR);

% with preprocessed values, build jet colormap
visualR = ind2rgb(indexR, colormap(jet));
imshow(visualR);
caxis([min(R, [], 'all') max(R, [], 'all')]);
colorbar;
title("Visualize R with Jet Colormap");

%
% ToDo: Threshold R & Collect Local Maximum Points
%

% Set threshold value
threshold = max(R, [], 'all')/10;
% local max points set : initial with (1,1) to avoid empty set during concat
locSet = [1 1];
for i=1:img_h
    for j=1:img_w
        if R(i,j) > threshold
            temp = [j i];
            locSet = cat(1, locSet, temp);
        end
    end
end
% check initial (1,1) meets threshold : if not, drop it
if R(1, 1) <= threshold
    locSet = locSet(2:end, :);
end
% Finally, collect corner points
points = cornerPoints(locSet);


%
% Visualize corner points over the input image
%

figure, imshow(I)

hold on

plot(points)

hold off
```

<1> R 계산하기

이미지로부터 gradient를 계산해낸 후, Gaussian filtering을 통한 smoothing을 진행한다. 그 결과로 얻어낸 Smoothed gradients들을 통해 Harris Matrix H를 계산하고, Harris Matrix로부터 corner response function R을 계산한다. Harris Matrix로부터 eigen value를 구하는 것은 상당한 계산량으로 소모적인 작업이기 때문에, 수업시간과 pdf에 지시된 내용처럼 matrix H의 determinant와 trace를 이용하여 구했다.

<2> 제트 컬러맵을 이용한 R 시각화

<1>번에서 R을 계산한 작업을 수행한 결과로, 우리의 목적은 여러 region에 대해서 flat region인지, edge 또는 corner인지 탐지를 하기 위함이다. 이를 위해 컬러맵을 통한 시각화를 진행하는데 특히 jet 컬러맵으로 무지개 색상을 표현하도록 제약사항이 주어져 있기 때문에 이를 기반으로 구현했다. matlab에서 인덱스 값은 양의 정수로 제약되는 사항이 존재하기 때문에, R값을 기반으로 index range에 들어오도록 일종의 값에 대한 normalization의 전처리 작업으로 indexR을 만들었다. 컬러맵에서 컬러바의 상한 하한의 경우, 실제 계산된 R값의 최솟값과 최댓값을 이용하여 설정했다.

<3> R에 대한 thresholding과 Local maximum point 수집, 그리고 corner point 시각화

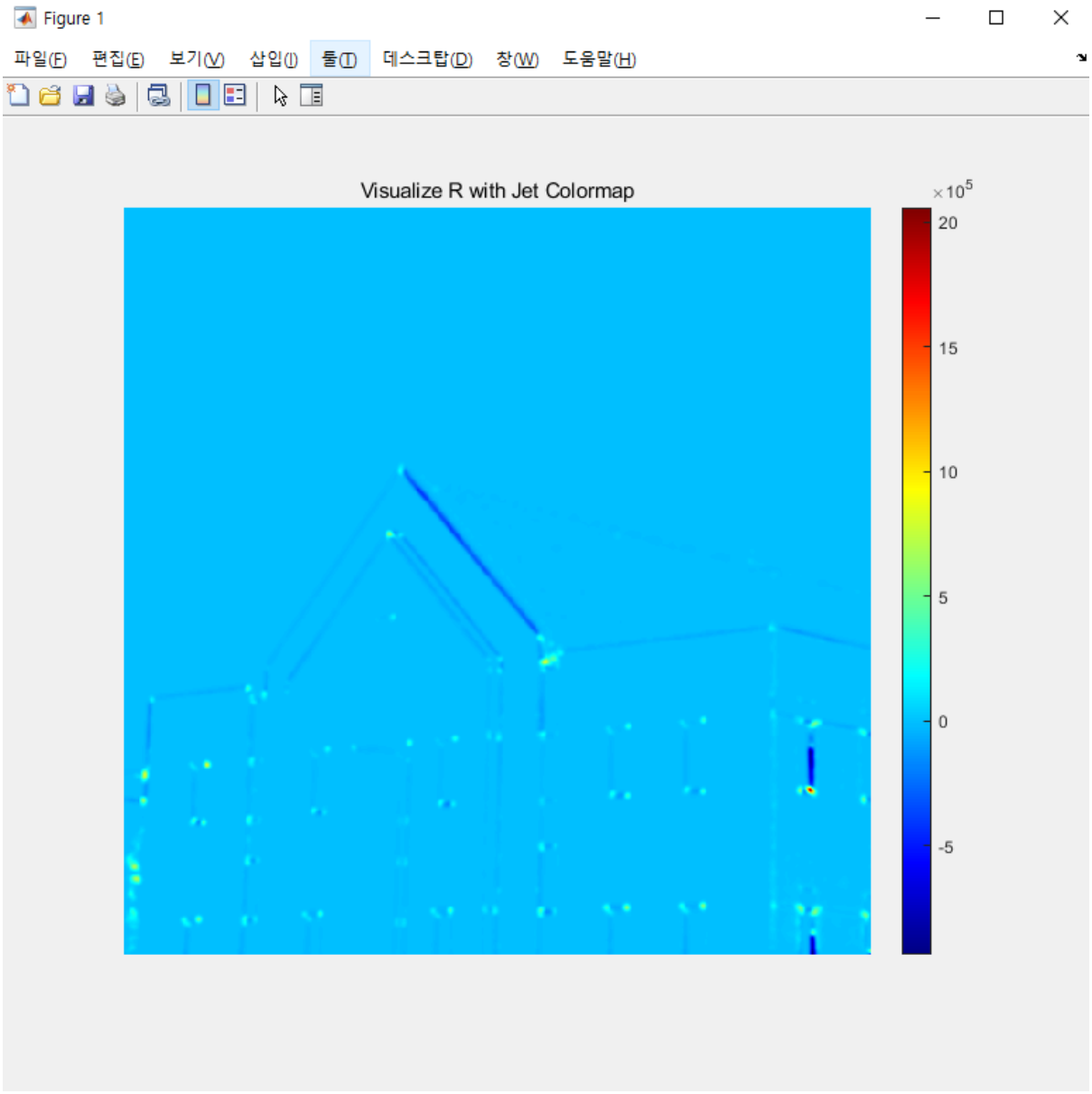
계산된 R 값에 대해서 특정 높은 양의 값의 threshold보다 높은 값에 대해 local maxima로 검출할 수 있다. Corner points를 수집하는 방식은 예제 코드에 기반하여 작성했다. 다만, locSet을 [1 1]로 초기화한 후 concatenation을 진행했는데, matlab에선 empty 배열에 대해 다루기 어렵기 때문에 cat을 쉽게 사용할 수 있도록 이처럼 초기화했다. 물론, 이렇게 초기화한 만큼, 실제로 (1,1) 점이 threshold보다 큰 지 재수정의 작업을 거쳤다.

[2] – 이미지 결과와 해석

[1]에서 설명한 것처럼 구현한 harris_corner.m을 통해 그 결과 이미지들에 대한 이야기를 할 것이다.

<1> 'building-600by600.tif' 이미지

원본 이미지는 건축물에 대한 600x600 사이즈의 이미지이다. 아래는 다양한 k값으로 계산된 R을 Jet colormap으로 시각화한 이미지들이다.



K = 0.05

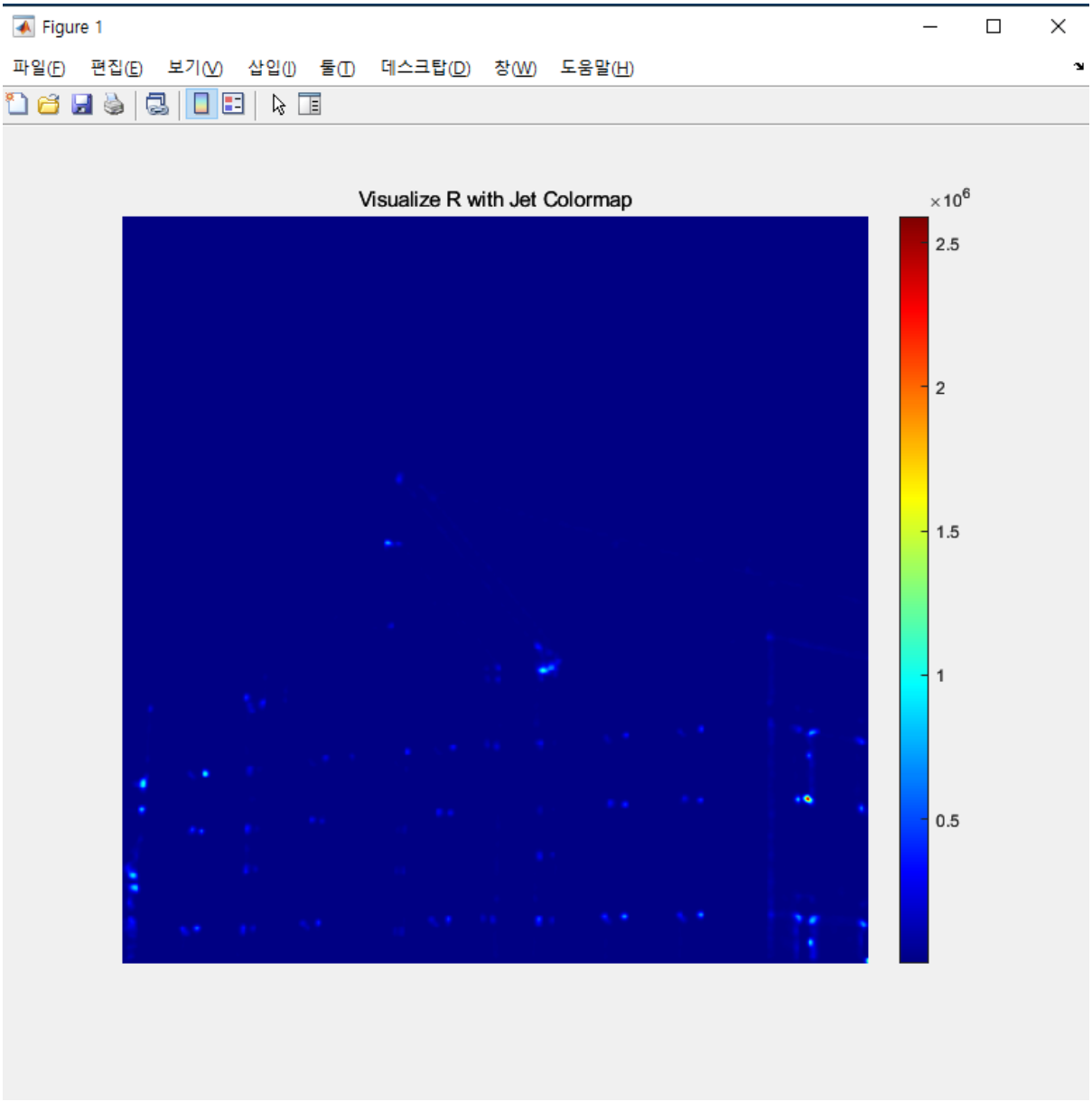


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Visualize R with Jet Colormap
×10^6
2.5
2
1.5
1
0.5

K = 0.0001

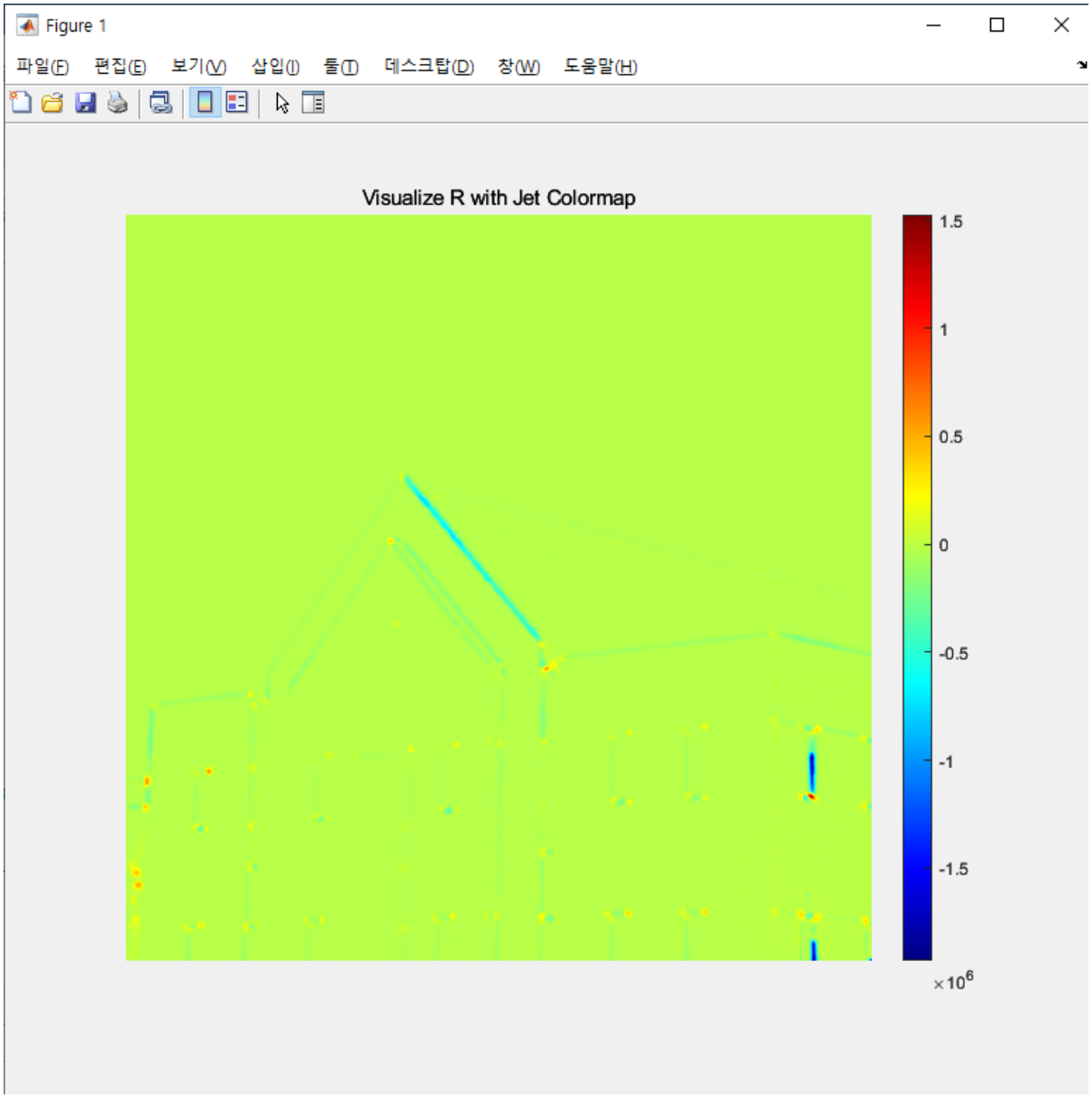
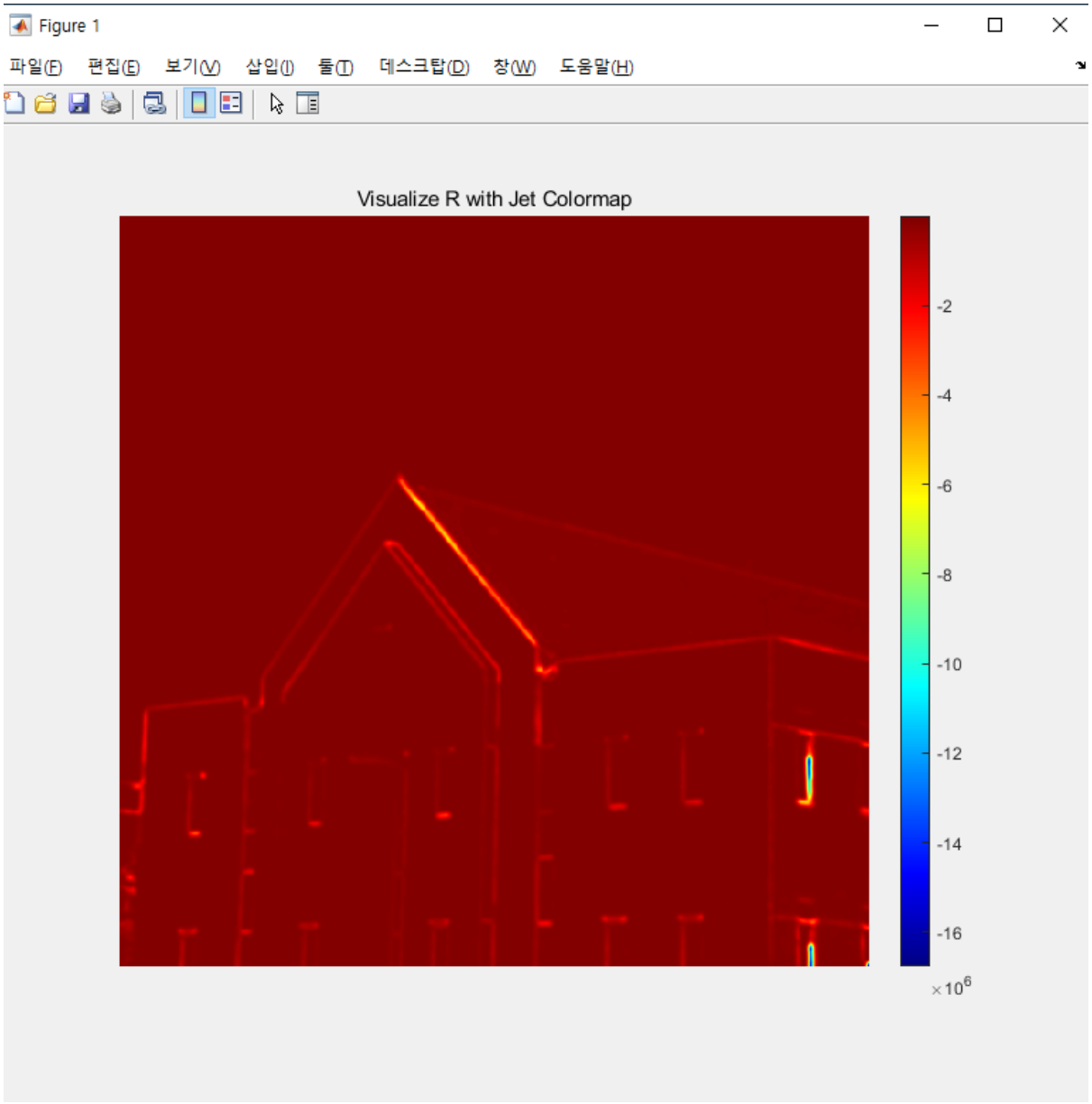


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Visualize R with Jet Colormap
1.5
1
0.5
0
-0.5
-1
-1.5
×10^6

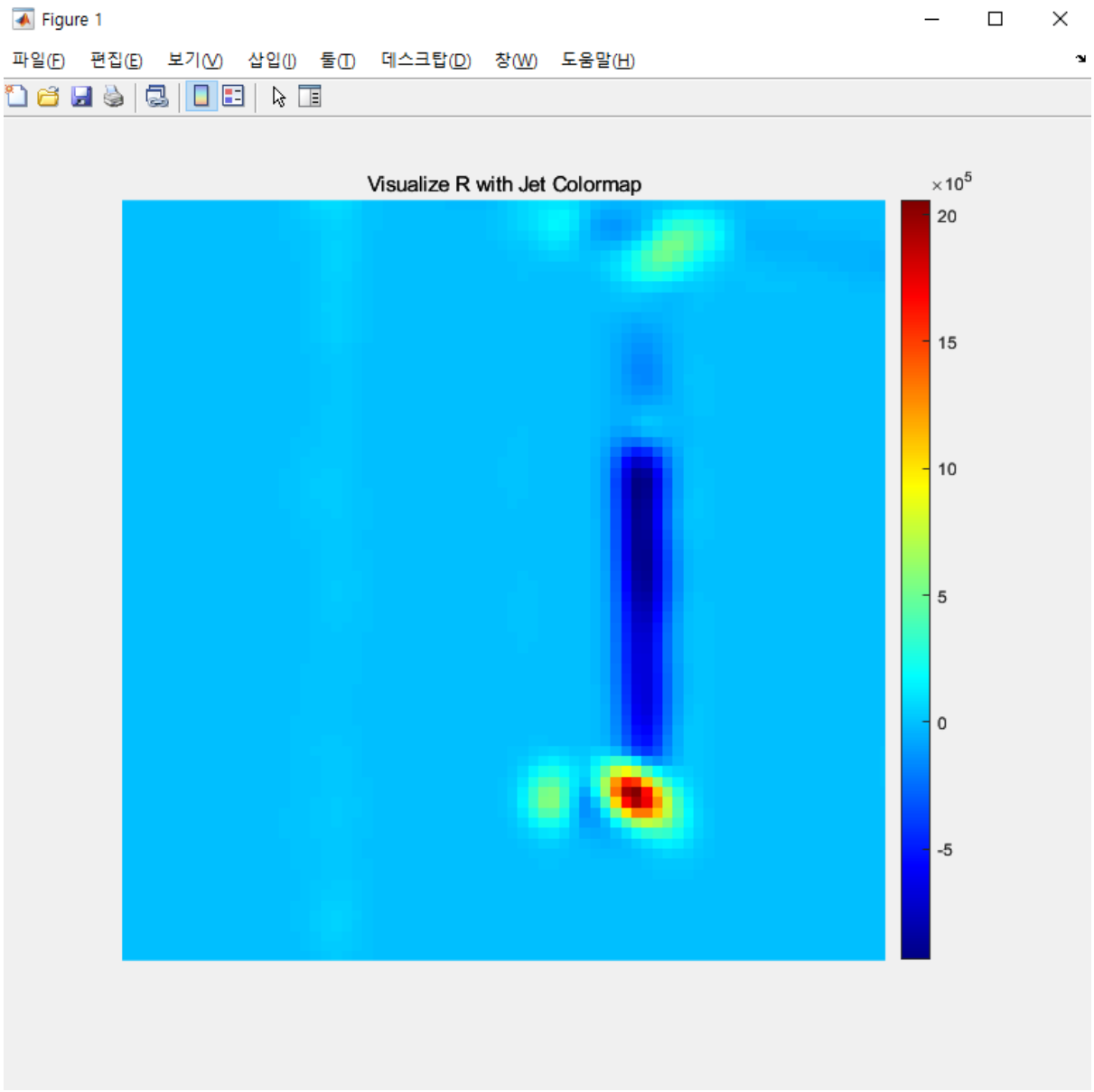
K = 0.1



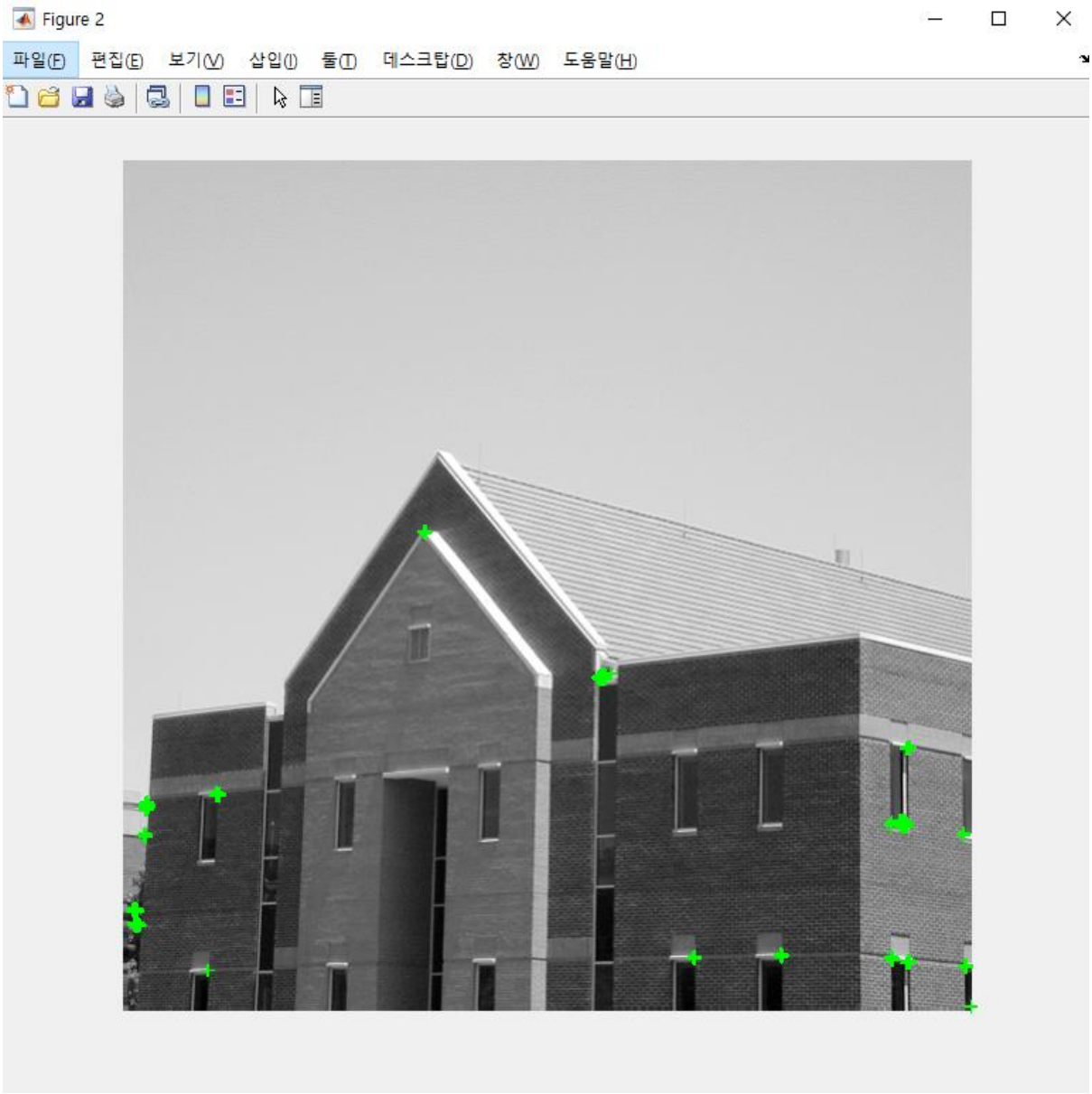
K = 0.75

R에서 중요한 점은, corner에 대해선 높은 양의 값, elongated edge에 대해선 낮은 음의 값을 가진다는 점이다. 따라서 이러한 특징적인 corner와 edge가 상반되게 관찰할 수 있게 아무래도 flat region은 적당한 중간점을 가져야 이를 관찰하기 편할 것이므로, 위 이미지들 중에선 관찰에 용이한 것은 k = 0.05 와 k = 0.1이 적절해 보인다고 할 수 있다.

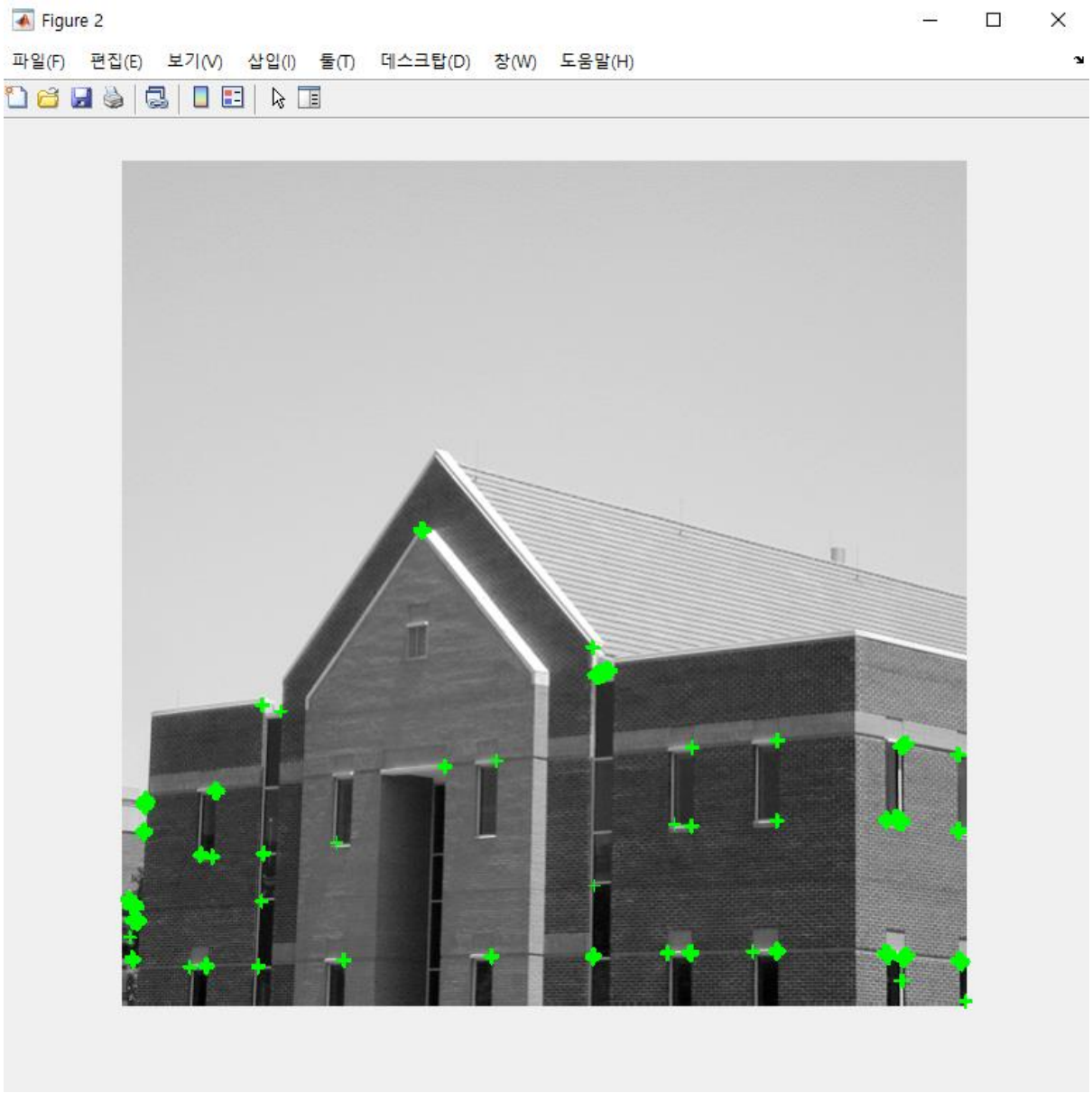
특히나 k = 0.05일 때가 pdf의 예시와 상당히 유사하기 때문에 추가로 k = 0.05 일 때 특정 지역을 zoom in 한 이미지 결과도 첨부한다.



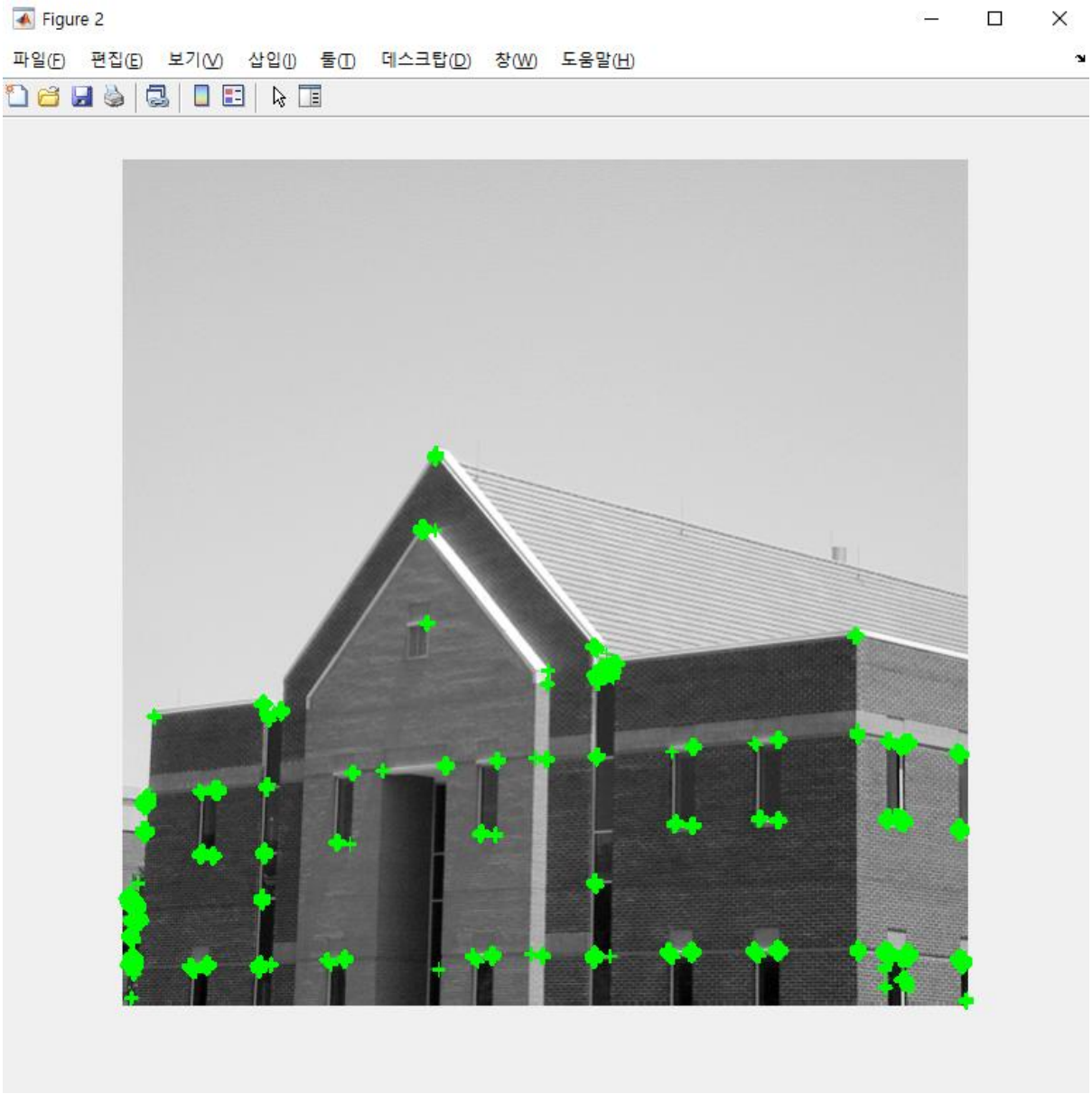
이제 아래는 k = 0.05로 고정시킨 후, threshold 값에 따라 corner detect에 관한 포인트 마커에 대한 이미지 결과이다. Threshold를 물론 양의 정수 값으로 하나 하나 대입해가며 찾는 것도 방법이지만, 필자의 경우에는 threshold를 전체 R 값 중 최대값에 대해서 몇 배수인지를 바꿔가는 식으로 적용했다. 아래가 그 결과들이다.



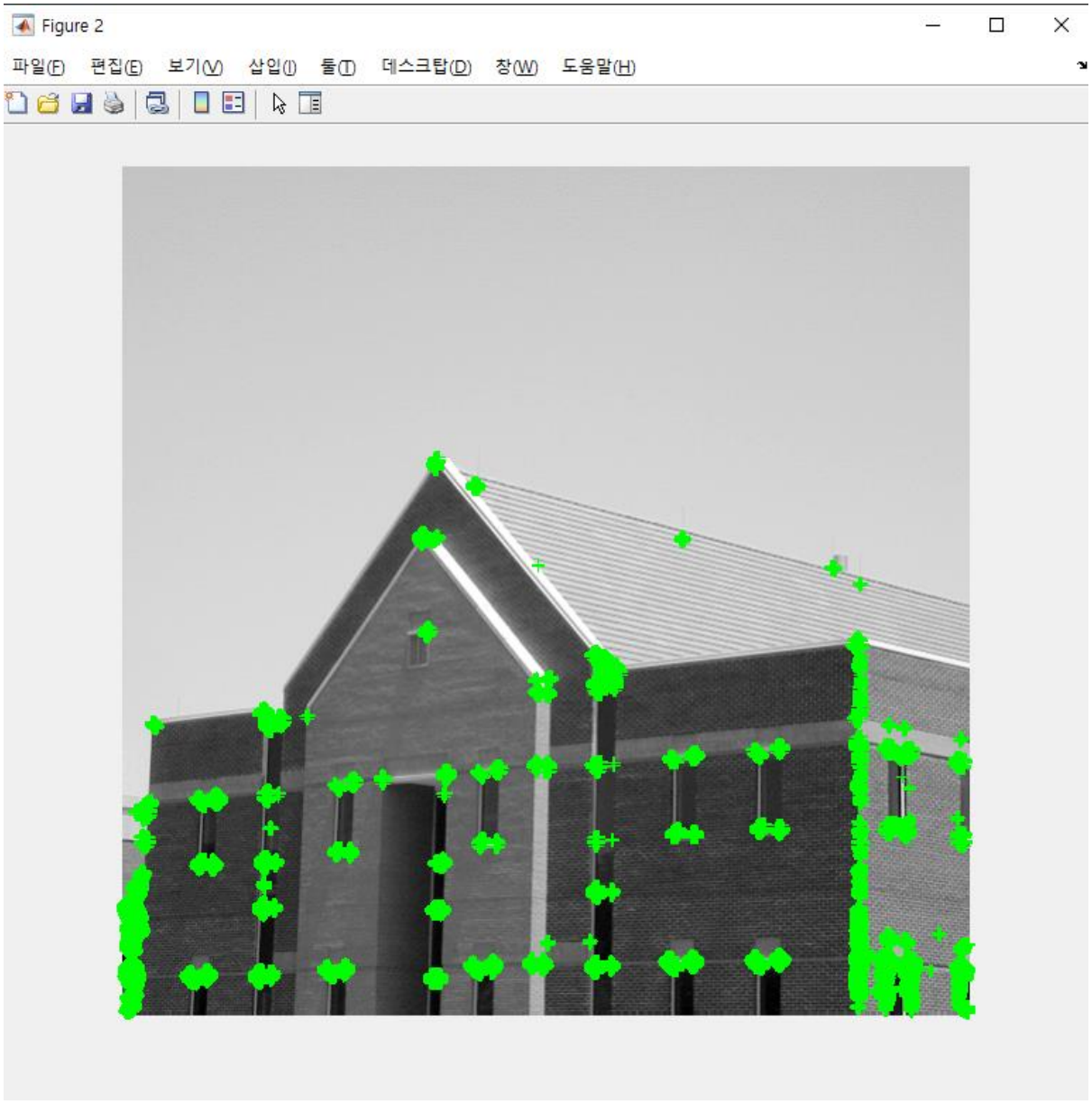

```
threshold = max(R, [], 'all')/5;
```




```
threshold = max(R, [], 'all')/10;
```




```
threshold = max(R, [], 'all')/25;
```

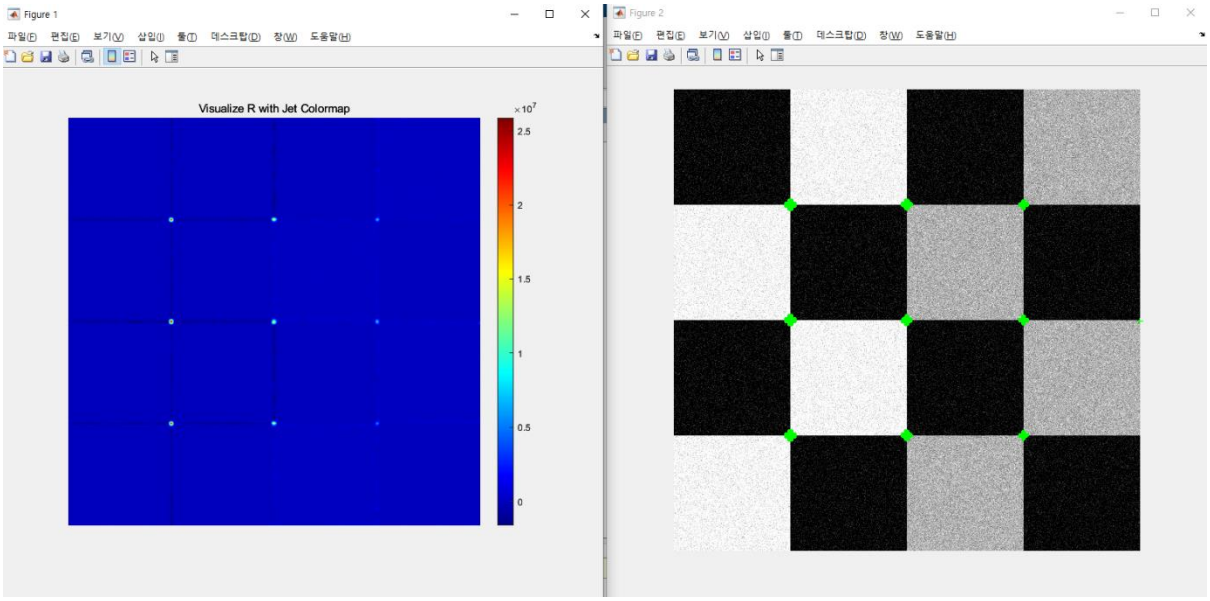



```
threshold = max(R, [], 'all')/100;
```

threshold값이 너무 클 경우, 극히 일부의 corner에 대해서만 detection이 이루어진 모습이다. 그에 반해, 그 값이 상대적으로 작아서 더 많은 점들을 통과시킬 경우는 과도하게 찍히는 것을 볼 수 있다.

<2> ‘checkerboard-noisy2.tif’ 이미지

내용적인 부분의 설명은 위에서 진행했으므로, 값에 대한 이미지만 첨부한다.



K = 0.05, `threshold = max(R, [], 'all')/10;`

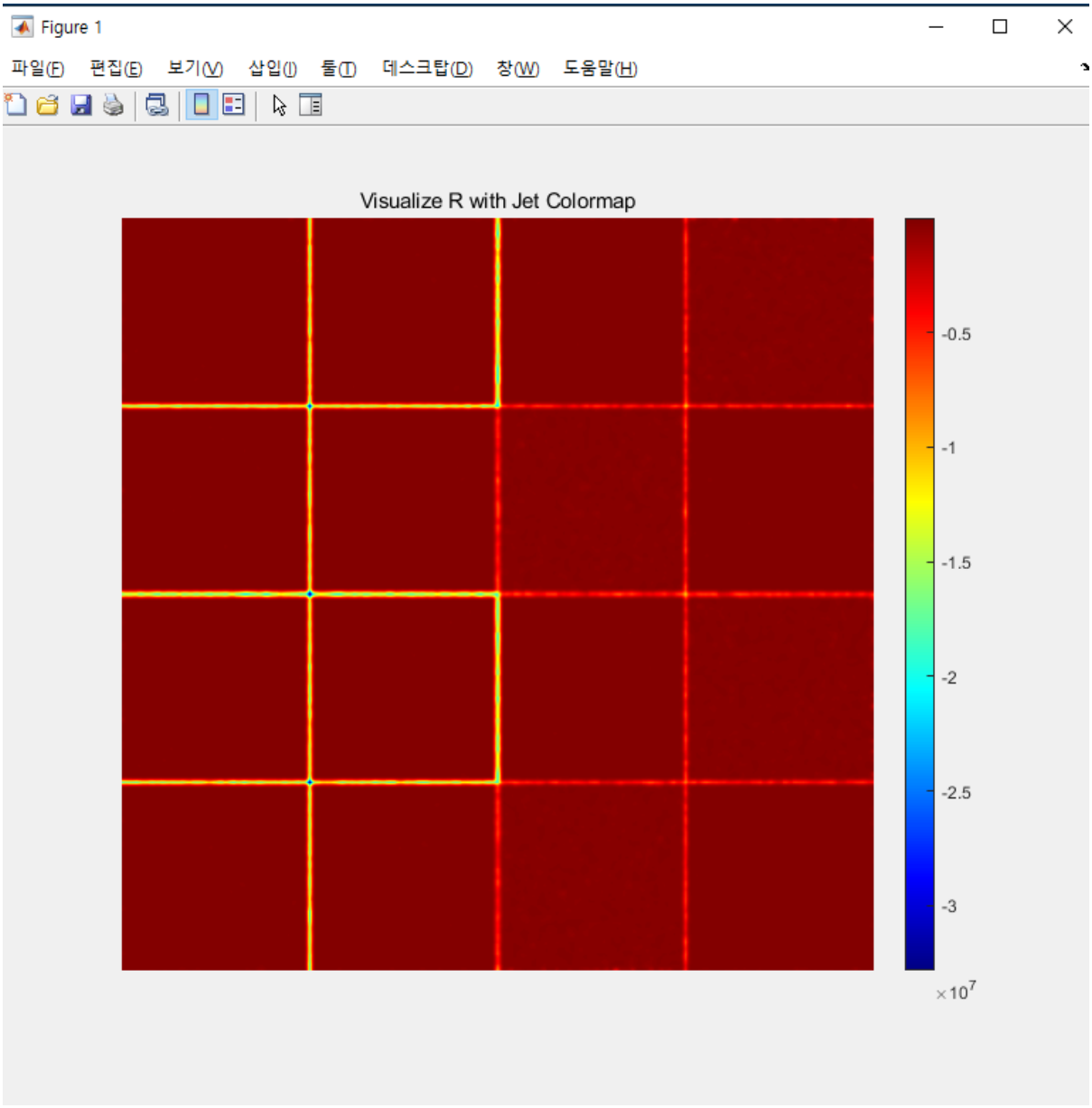
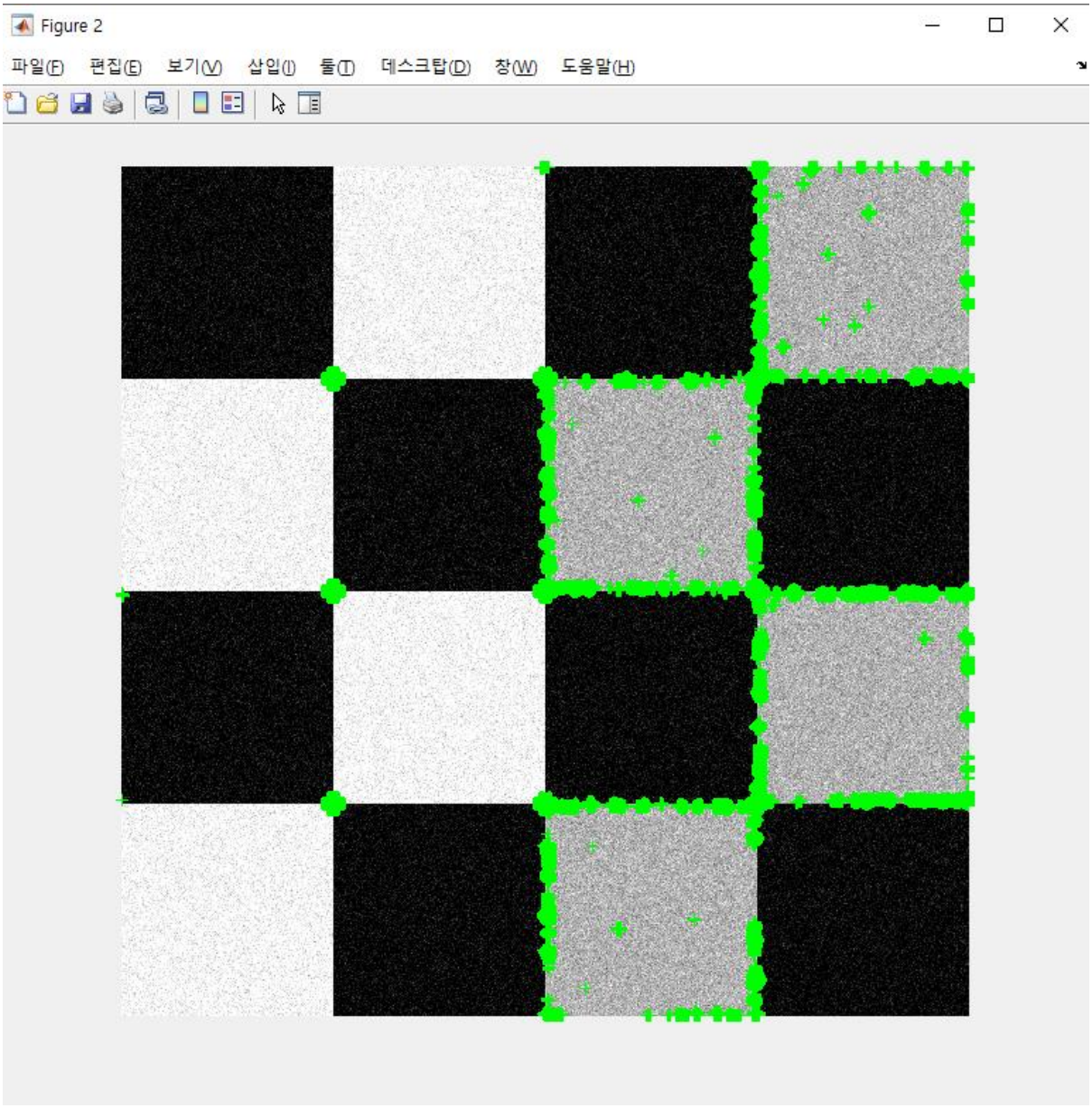
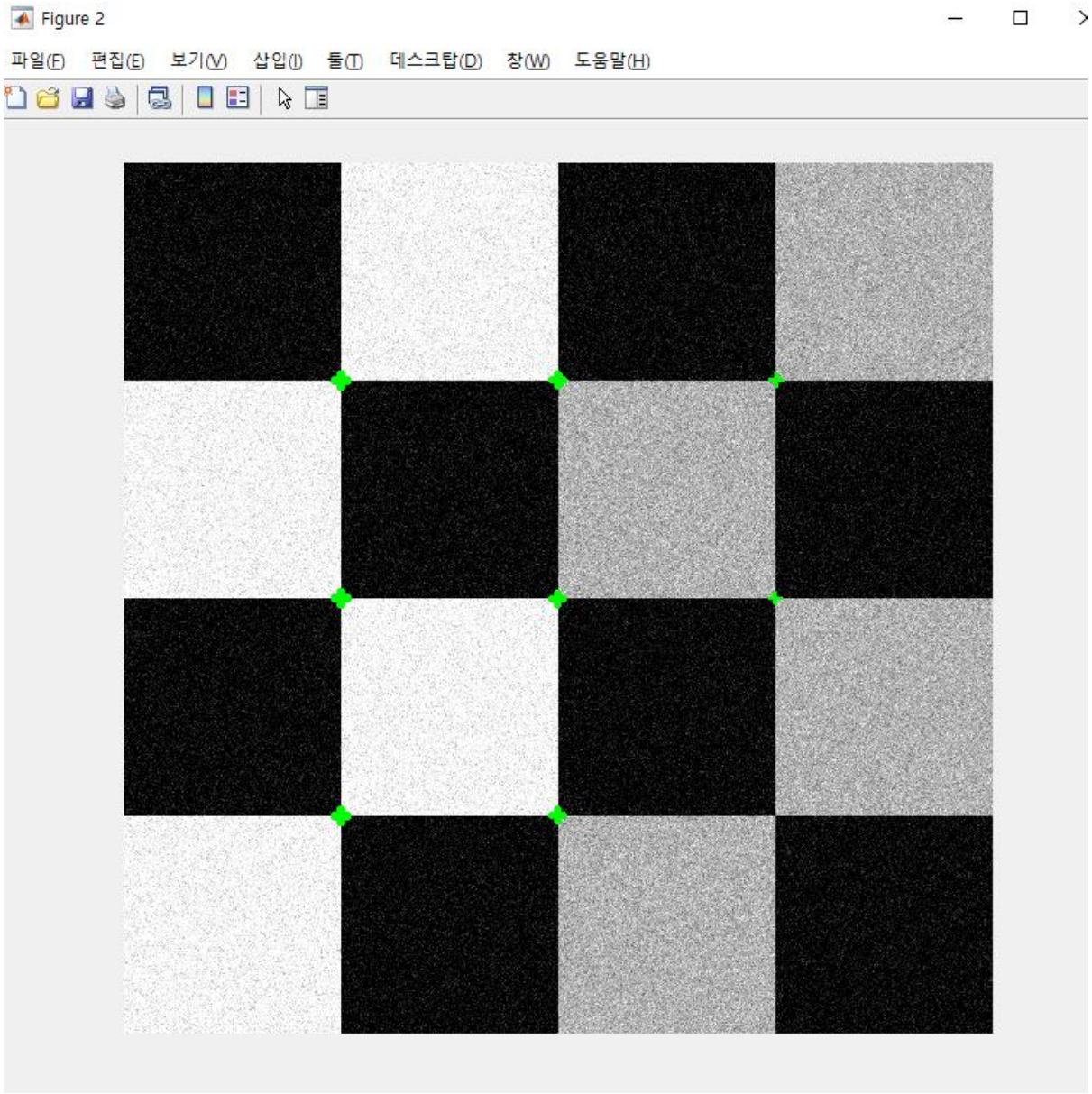


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Visualize R with Jet Colormap
-0.5
-1
-1.5
-2
-2.5
-3
×10^7

K = 0.5



K = 0.05, `threshold = max(R, [], 'all')/100;`



K = 0.05, `threshold = max(R, [], 'all')/4;`

<3> 'ArseneWenger.png' 이미지

예제를 제외하고 적당한 커스텀 사진을 이용하고자 했으나 이전 과제에서 사용했던 전 아스날 감독 아르센 벵거에 대한 회색조 이미지를 이용했다.


```
I=imread('ArseneWenger.png');
```

위 두 예제를 진행하면서, 어느 정도 fine tuning 된 값이 k = 0.05, `threshold = max(R, [], 'all')/10;` 라고 생각했기 때문에 해당 값으로 적용한 결과이다.

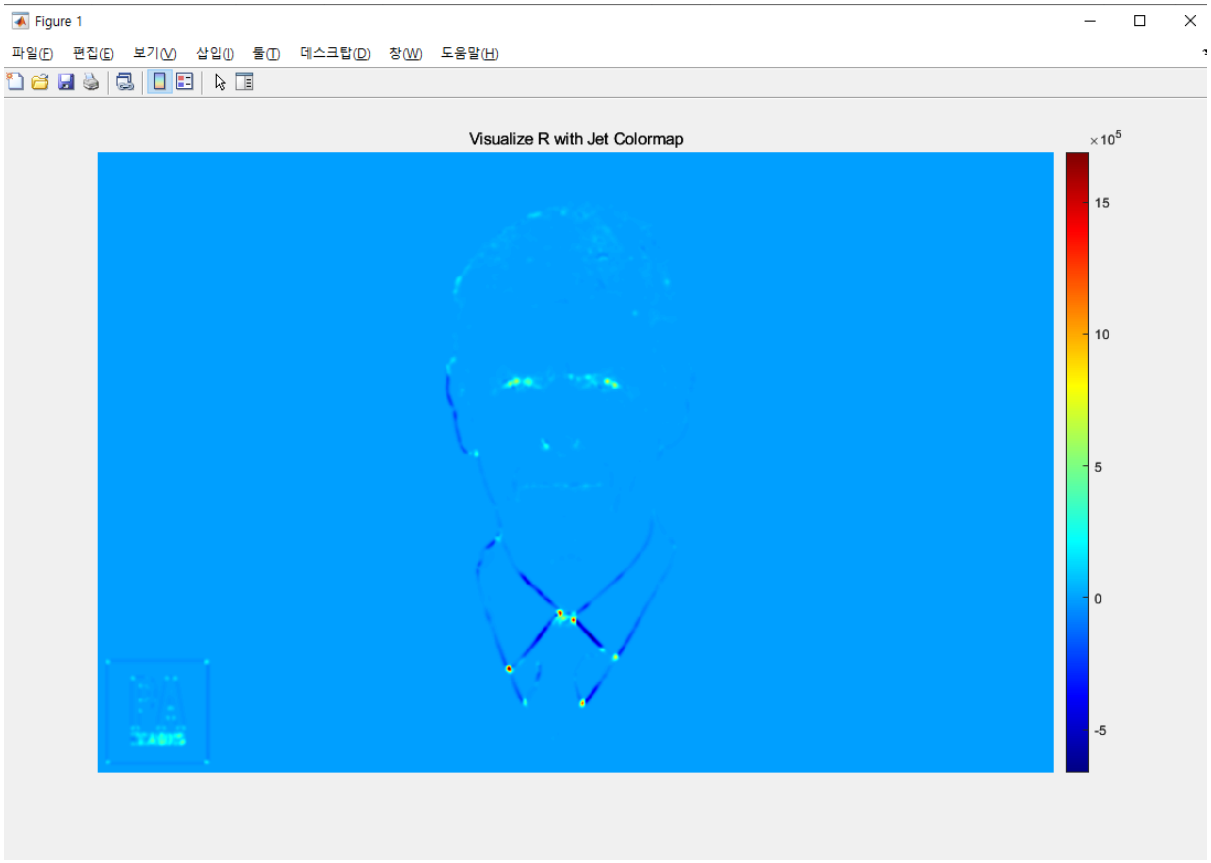


Figure 1
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
Visualize R with Jet Colormap
×10^5
15
10
5
0
-5

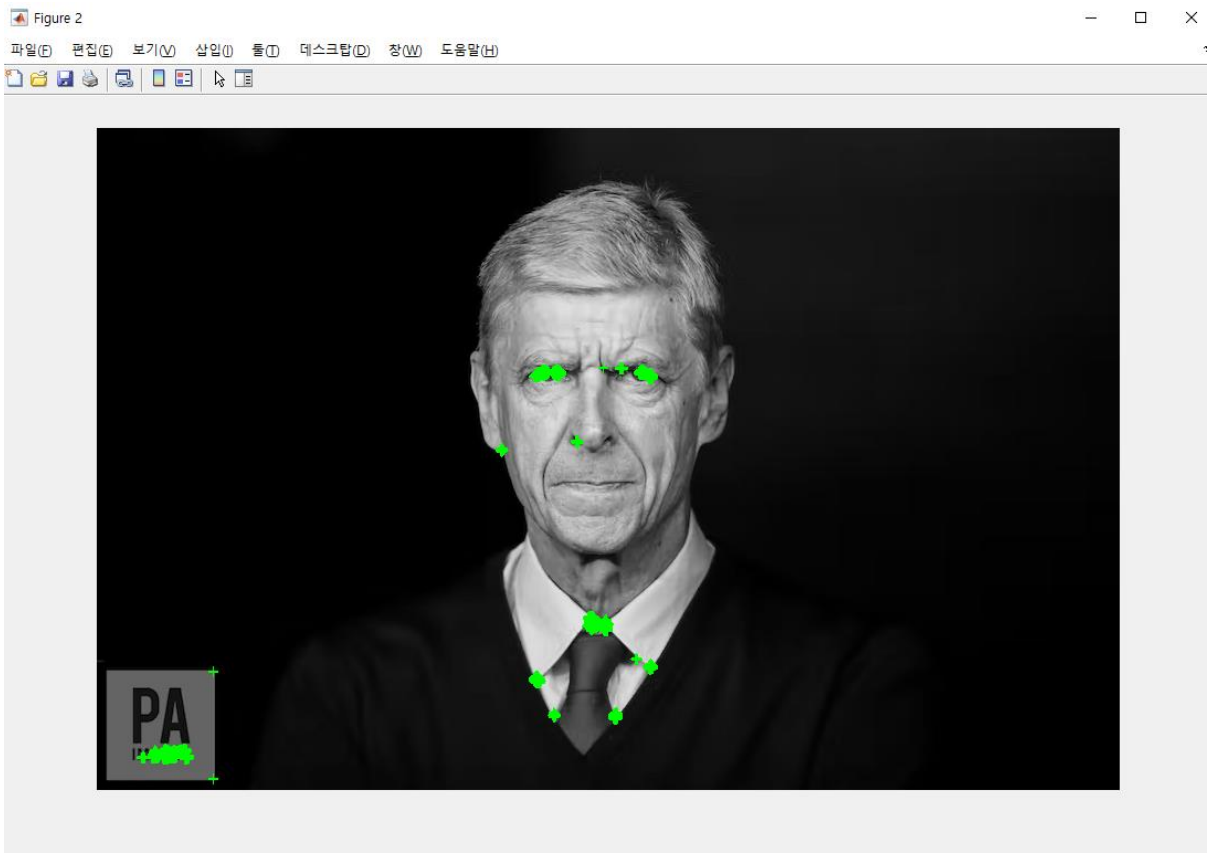


Figure 2
파일(F) 편집(E) 보기(V) 삽입(I) 툴(T) 데스크탑(D) 창(W) 도움말(H)
PA