중간 프로젝트

3차원 영상 시스템

2018265103 이형주

fixed_VCR.m (원본)

```
clc;
clear all;
close all;
% Integral Imaging system parameters settting
% High resolution (The number of pixels for each elemental image)
%Nx = 3072;
%Ny = 2048;
% Medium resolution (The number of pixels for each elemental image)
Nx = 1536;
Ny = 1024;
f = 105;        % Focal length of lens
p = 2;          % pitch between elemental images
cx = 36;        % Sensor size in x direction
cy = 24;        % Sensor size in y direction
Lx = 10;        % The number of elemental images in x direction
Ly = 10;        % The number of elemental images in y direction
for z = 200:10:500 % Reconstruction depth
    z
    % Shifting pixels for each elemental image
    shx = round(Nx*f*p/(cx*z));
    shy = round(Ny*f*p/(cy*z));
    % Reconstructed 3D image matrix
    recon = zeros(Ny + (Ly-1)*shy, Nx + (Lx-1)*shx, 3);
    % Overlapping matrix
    recon_over = zeros(Ny + (Ly-1)*shy, Nx + (Lx-1)*shx, 3);
    % Superposition of elemental images atreconstruction depth
    for k1 = 1:Ly
        for k2 = 1:Lx
            %a = imread(['D: \office_com_backup\mjcho\Lecture_Note\학부수업\1
            a = imread(['D: \office_com_backup\mjcho\Lecture_Note\학부수업\1학
            recon((k1-1)*shy + 1:(k1-1)*shy + Ny, (k2-1)*shx + 1:(k2-1)*shx
            recon_over((k1-1)*shy + 1:(k1-1)*shy + Ny,(k2-1)*shx + 1:(k2-1)
        end
    end
    % Averaging
    recon = recon./recon_over;
    % Save reconstructed 3D image
    %imwrite(recon, ['D: \office_com_backup\mjcho\Lecture_Note\학부수업\1학기 
    imwrite(recon, ['D: \office_com_backup\mjcho\Lecture_Note\학부수업\1학기 \
end
```

fixed_VCR.m (수정본)

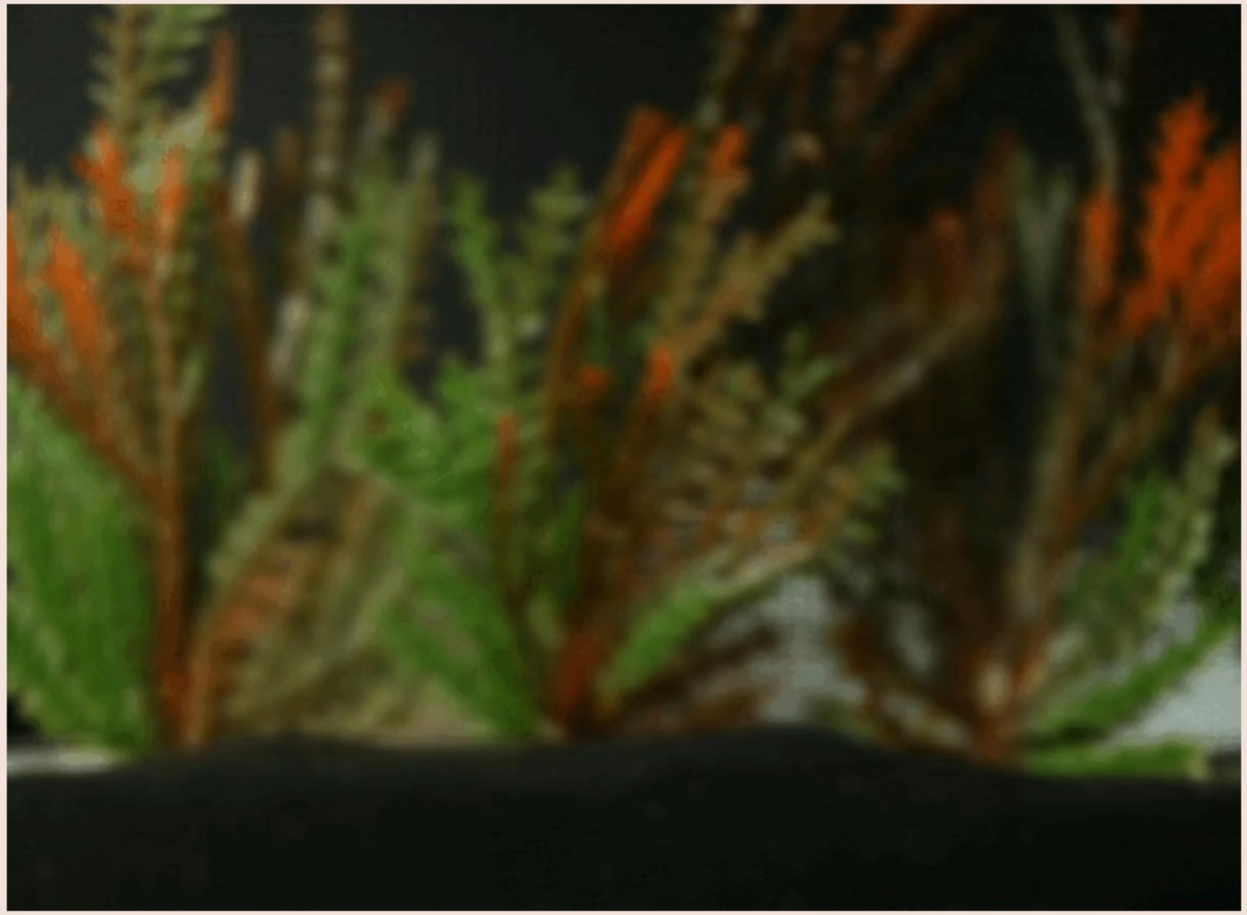
```
clc;
clear all;
close all;
% Integral Imaging system parameters settting

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% High resolution (The number of pixels for each elemental image)
%Nx = 3072;
%Ny = 2048;
% Medium resolution (The number of pixels for each elemental image)
%Nx = 1536;      %이미지의 픽셀값
%Ny = 1024;
%Three_Characters (The number of pixels for each elemental image)
Nx = 3008;
Ny = 2000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

f = 105;        % Focal length of lens
p = 2;          % pitch between elemental images
cx = 36;        % Sensor size in x direction
cy = 24;        % Sensor size in y direction
Lx = 10;        % The number of elemental images in x direction
Ly = 10;        % The number of elemental images in y direction

% Reconstruction depth
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%for depth = 200:5:500 % grass_and_car는 200:5:500
for depth = 400:5:1000 % Three_Chracters는 400~1000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    depth
    % Shifting pixels for each elemental image
    shx = round(Nx*f*p/(cx*depth));
    shy = round(Ny*f*p/(cy*depth));
    % Reconstructed 3D image matrix
    recon = zeros(Ny + (Ly-1)*shy, Nx + (Lx-1)*shx, 3); %겹쳐진 이미지의 전체 없이를 계산하는 식
    % Overlapping matrix
    recon_over = zeros(Ny + (Ly-1)*shy, Nx + (Lx-1)*shx, 3);
    % Superposition of elemental images at reconstruction depth
    for k1 = 1:Ly
        for k2 = 1:Lx

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %image = imread(['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/high_EIs/high_',num2str((k1-1)*L
            %image = imread(['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/medium_EIs/medium_',num2str((k1-
            image = imread(['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/Three_Characters/EI_', num2str((k
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            %image를 중첩하여 채움
            recon((k1-1)*shy + 1:(k1-1)*shy + Ny, (k2-1)*shx + 1:(k2-1)*shx + Nx, :) = recon((k1-1)*shy + 1:(k1-1)*
            %같은 방식으로 1을 중첩하여 채움
            recon_over((k1-1)*shy + 1:(k1-1)*shy + Ny,(k2-1)*shx + 1:(k2-1)*shx + Nx, :) = recon_over((k1-1)*shy +
        end
    end
    % Averaging
    recon = recon./recon_over; %요소간의 나눗셈 ./

    % Save reconstructed 3D image
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %imwrite(recon, ['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/high_results_fixed/recon_fixed_high_z_',
    %imwrite(recon, ['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/medium_results_fixed/recon_fixed_medium_
    imwrite(recon, ['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/Three_Characters_fixed/recon_fixed_Three_
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
```

3 character에
대해서도 실행

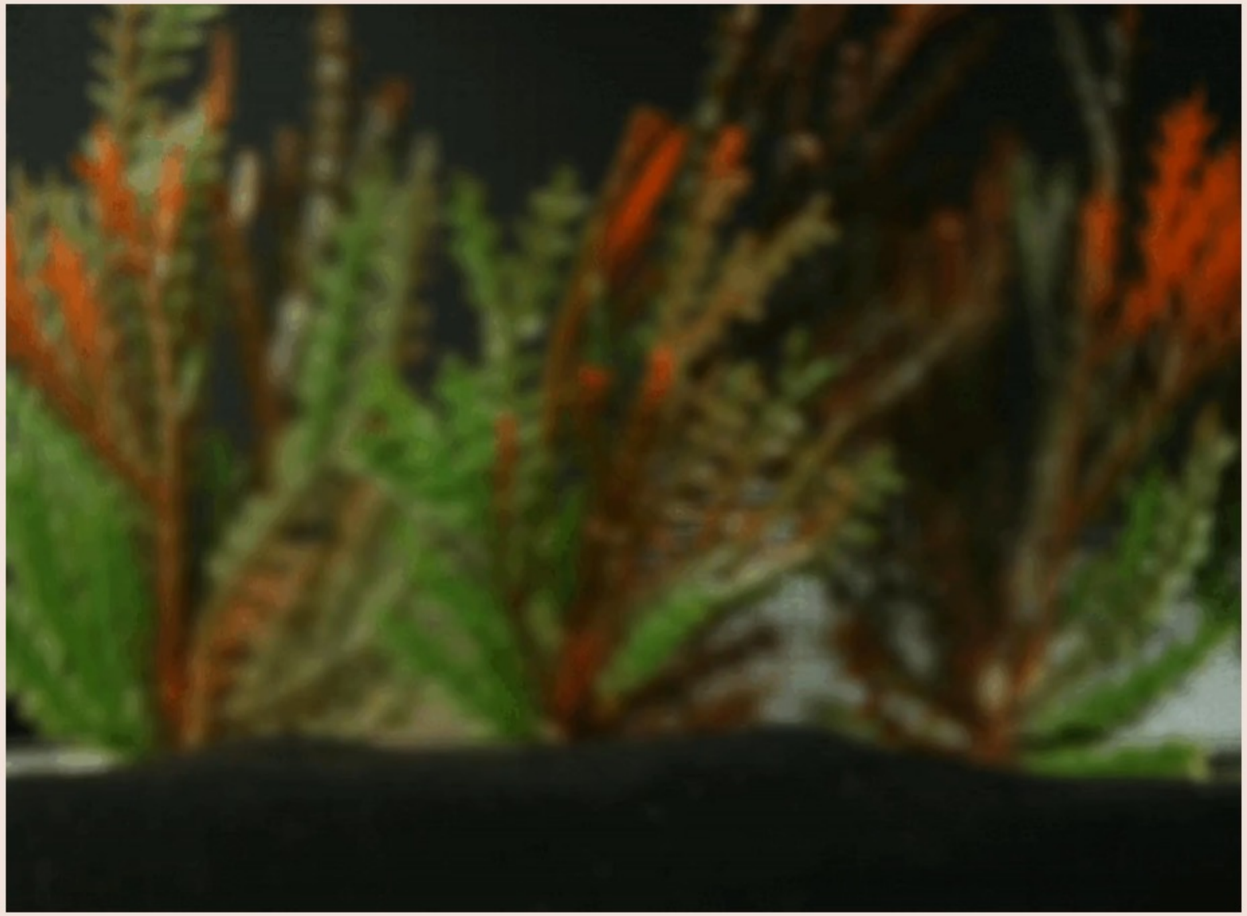
변수 이름 재설정

for문 범위 변경

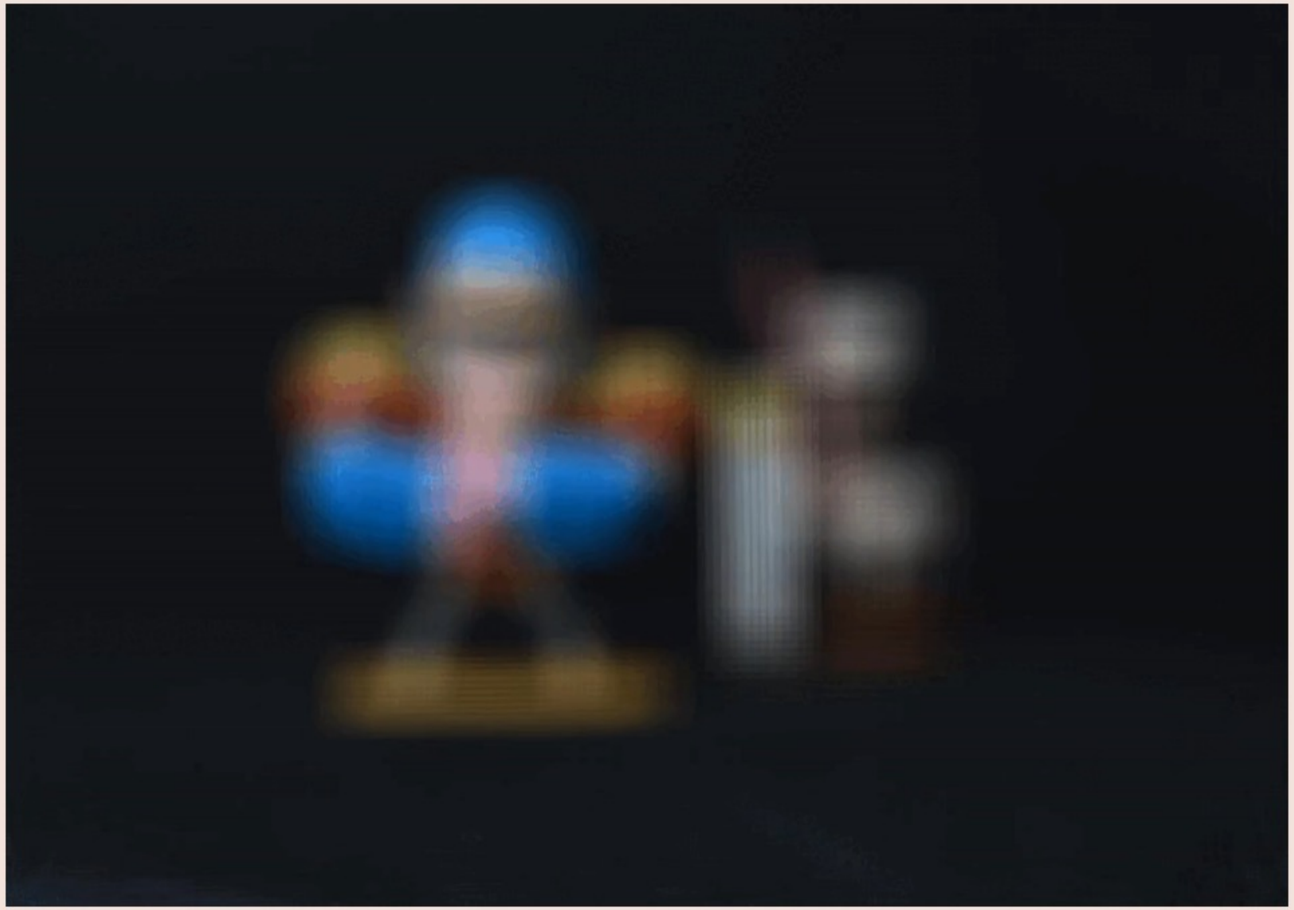
medium_results_fixed.mp4



high_results_fixed.mp4



Three_Characters_fixed.mp4



nonuniform_VCR.m (원본)

```
clc;
clear all;
close all;
% Integral Imaging system parameters settting
% High resolution (The number of pixels for each elemental image)
Nx = 3072;
Ny = 2048;
% Medium resolution (The number of pixels for each elemental image)
%Nx = 1536;
%Ny = 1024;
f = 105;
p = 2;
cx = 36;
cy = 24;
Lx = 10;
direction
Ly = 10;
direction
for z = 200:10:500 % Reconstruction depth
    z
    % Shifting pixels for each elemental image
    shx = round(Nx*f*p/(cx*z).*[0:Lx-1]);
    shy = round(Ny*f*p/(cy*z).*[0:Ly-1]);
    % Reconstructed 3D image matrix
     recon = zeros(Ny + max(shy(:)), Nx + max(shx(:)),3);
    % Overlapping matrix
    recon_over = zeros(Ny + max(shy(:)), Nx + max(shx(:)), 3);
    % Superposition of elemental images at reconstruction depth
    for k1 = 1:Ly
        for k2 = 1:Lx
            a = imread(['D: \office_com_backup\mjcho\Lecture_Note\학부수업\1학기 \3D_I
            %a = imread(['D: \office_com_backup\mjcho\Lecture_Note\학부수업\1학기 \3D_
            recon(shy(k1) + 1:shy(k1) + Ny, shx(k2) +1:shx(k2) + Nx, :) = recon(shy
            recon_over(shy(k1) + 1:shy(k1) + Ny, shx(k2) + 1:shx(k2) + Nx, :) = rec
        end
    end
    % Averaging
    recon = recon./recon_over;
    % Save reconstructed 3D image
    imwrite(recon, ['D: \office_com_backup\mjcho\Lecture_Note\학부수업\1학기 \3D_Imagi
    %imwrite(recon, ['D: \office_com_backup\mjcho\Lecture_Note\학부수업\1학기 \3D_Imag
end
```

nonuniform_VCR.m (수정본)

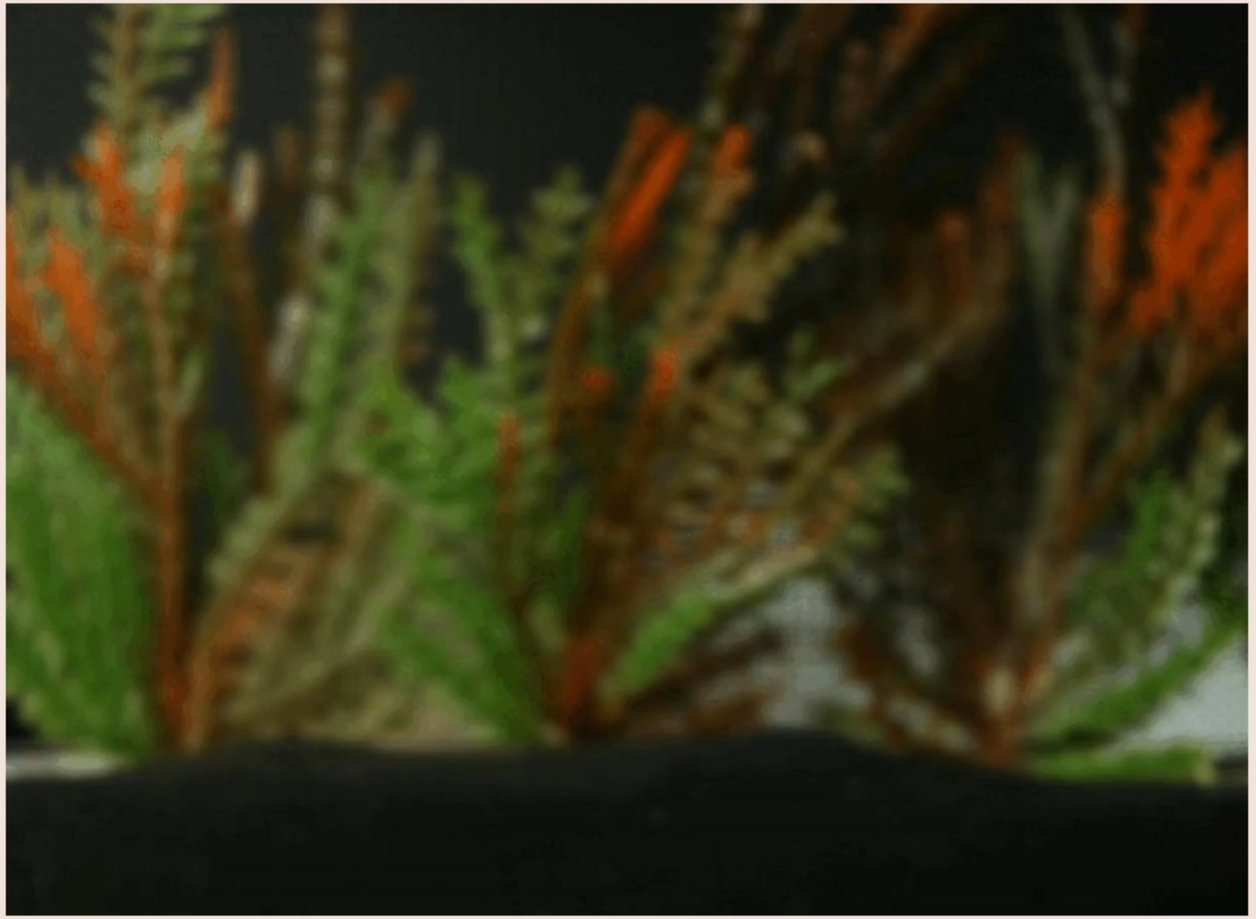
```
clc;
clear all;
close all;
% Integral Imaging system parameters settting

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% High resolution (The number of pixels for each elemental image)
%Nx = 3072;
%Ny = 2048;
% Medium resolution (The number of pixels for each elemental image)
Nx = 1536;
Ny = 1024;
%Three_Characters (The number of pixels for each elemental image)
%Nx = 3008;
%Ny = 2000;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

f = 105;        % Focal length of lens
p = 2;          % pitch between elemental images
cx = 36;        % Sensor size in x direction
cy = 24;        % Sensor size in y direction
Lx = 10;        % The number of elemental images in x direction
Ly = 10;        % The number of elemental images in y direction

% Reconstruction depth
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for depth = 205:10:500 % grass_and_car = 200:5:500
%for depth = 400:5:1000 % Three_Chracters = 400:5:1000
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

    depth
    % Shifting pixels for each elemental image
    shx = round(Nx*f*p/(cx*depth).*[0:Lx-1]);
    shy = round(Ny*f*p/(cy*depth).*[0:Lx-1]);
    % Reconstructed 3D image matrix
    recon = zeros(Ny + max(shy(:)), Nx + max(shx(:)),3);
    % Overlapping matrix
    recon_over = zeros(Ny + max(shy(:)), Nx + max(shx(:)), 3);
    % Superposition of elemental images at reconstruction depth
    for k1 = 1:Ly
        for k2 = 1:Lx

            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
            %image = imread(['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/high_EIs/high_', num2str((k1-1)*
            image = imread(['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/medium_EIs/medium_', num2str((k1-
            %image = imread(['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/Three_Characters/EI_', num2str((
            %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

            %image를 중첩하여 채움
            recon(shy(k1) + 1:shy(k1) + Ny, shx(k2) + 1:shx(k2) + Nx, :) = recon(shy(k1) + 1:shy(k1) + Ny, shx(k2)
            %같은 방식으로 1을 중첩하여 채움
            recon_over(shy(k1) + 1:shy(k1) + Ny, shx(k2) + 1:shx(k2) + Nx, :) = recon_over(shy(k1) + 1:shy(k1) + N
        end
    end
    % Averaging
    recon = recon./recon_over; %요소간의 나눗셈 ./

    % Save reconstructed 3D image
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    %imwrite(recon, ['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/high_results_nonuniform/recon_nonuniform
    imwrite(recon, ['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/medium_results_nonuniform/recon_nonunifor
    %imwrite(recon, ['/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/Three_Characters_nonuniform/recon_nonuni
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
end
```

3 character에
대해서도 실행

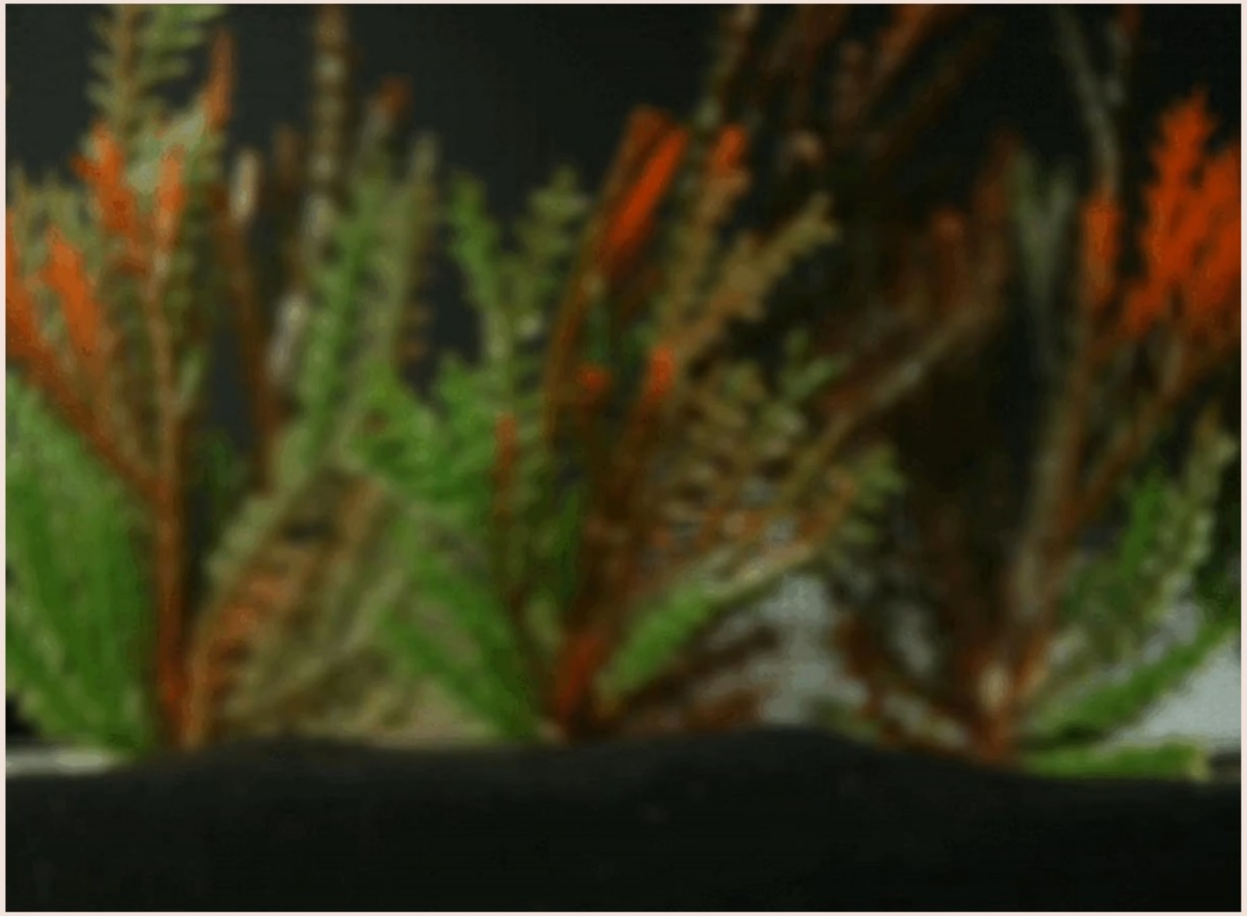
변수 이름 재설정

for문 범위 변경

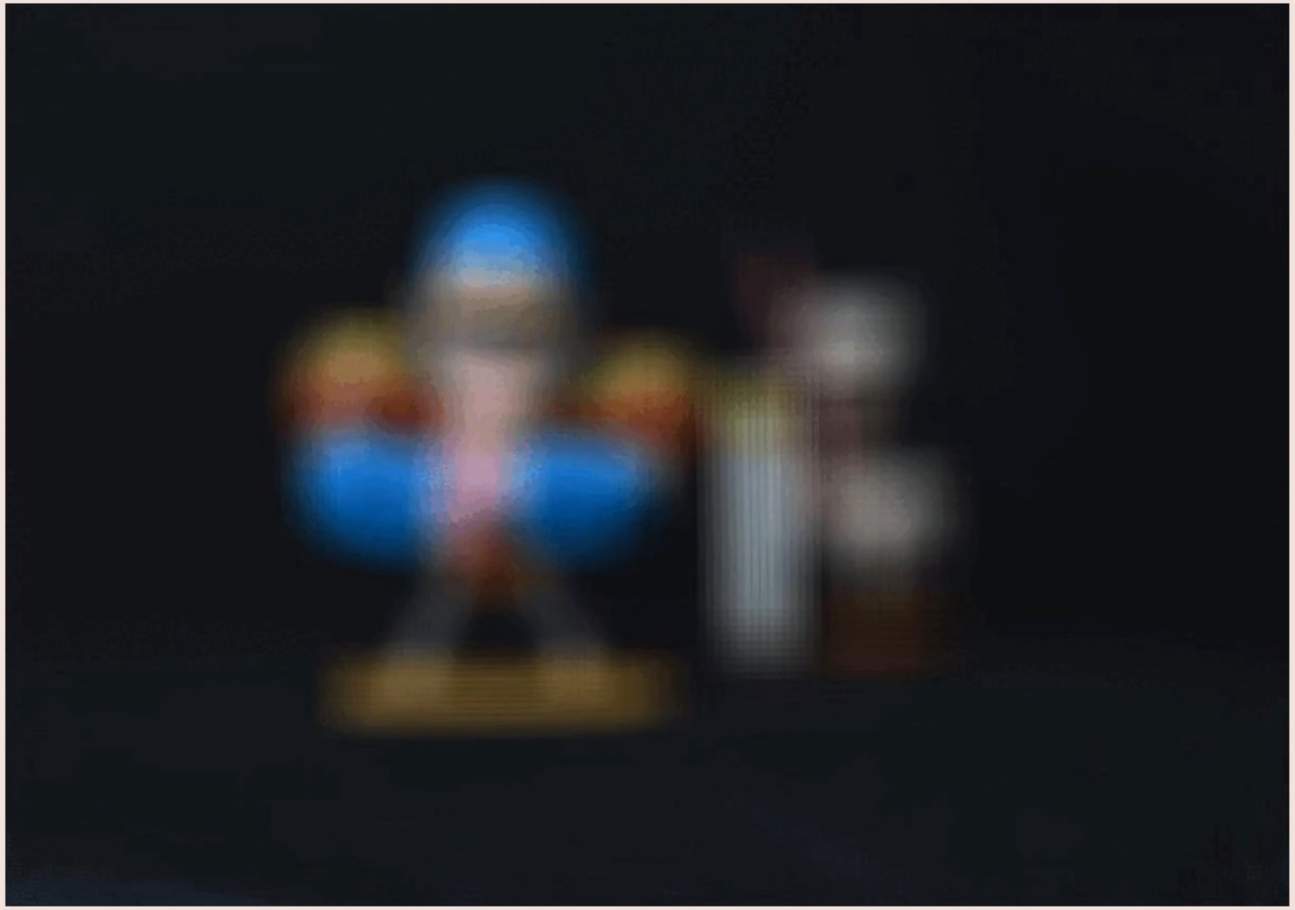
medium_results_nonuniform.mp4



high_results_nonuniform.mp4



Three_Characters_nonuniform.mp4



ex_test.m (원본)

```
clc;
clear all;
close all;
PSR_t = zeros(7,3);
a = im2double(rgb2gray(imread('/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/lena.jpeg')));   % Reference Image
b = im2double(rgb2gray(imread('/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/lena.jpeg')));   % False Image
[v h d] = size(a);
[v1 h1 d1] = size(b);
% Zero-Padding
a1 = [zeros(v, 256), a, zeros(v, 256)];
a1 = [zeros(256, h+512); a1; zeros(256, h+512)];
b1 = [zeros(v1, 256), b, zeros(v1, 256)];
b1 = [zeros(floor(v/2), h1+512); b1; zeros(floor(v/2), h1+512)];
% Fourier Transform
aff1 = fft2(a1);
bff1 = fft2(b1);
% Linear Correlation Filter
LCFa = abs(fftshift(ifft2(aff1.*conj(aff1))));       % autocorrelation
LCF = abs(fftshift(ifft2(bff1.*conj(aff1))));       % cross-correlation
max_L = max(LCFa(:));   % Maximum of autocorrelation
figure, mesh(LCF./max_L);
axis([0 1500 0 1700 0 1]);  % Axis (IMPORTANT)
% Phase Only Filter
POFa = abs(fftshift(ifft2(exp(1i.*(angle(aff1) - angle(aff1))))));       % autocorrelation
POF = abs(fftshift(ifft2(exp(1i.*(angle(bff1) - angle(aff1))))));       % cross-correlation
max_P = max(POFa(:));   % Maximum of autocorrelation
figure, mesh(POF./max_P);
axis([0 1500 0 1700 0 1]);  % Axis (IMPORTANT)
% Nonlinear Correlation Filter
k = 0.3;
NLFa = abs(fftshift(ifft2(((abs(aff1).*abs(aff1)).^k).*exp(1i.*(angle(aff1) - angle(aff1))))));
NLF = abs(fftshift(ifft2(((abs(bff1).*abs(aff1)).^k).*exp(1i.*(angle(bff1) - angle(aff1))))));
max_N = max(NLFa(:));   % Maximum of autocorrelation
figure, mesh(NLF./max_N);
axis([0 1500 0 1700 0 1]);  % Axis (IMPORTANT)
```

ex_test.m (수정본)

```
clc;
clear all;
close all;

referenceImage = im2double(rgb2gray(imread('/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/h
inputImage = im2double(rgb2gray(imread('/Users/zoocasso/Desktop/2022 - 1학기/3차원영상처리/image/high_
[v, h] = size(referenceImage);
[v1, h1] = size(inputImage);

if(rem(v,2) == 1)
    v = v - 1;
end
if(rem(h,2) == 1)
    h = h - 1;
end

referenceImage = imresize(referenceImage,[v,h]);

if(rem(v1,2) == 1)
    v1 = v1 - 1;
end
if(rem(h1,2) == 1)
    h1 = h1 - 1;
end

inputImage = imresize(inputImage,[v1,h1]);

% Zero-Padding
referenceImageZeroPadding = padarray(referenceImage,[floor((v1-v)/2),floor((h1-h)/2)],0,'both');
figure, imshow(referenceImageZeroPadding)
size(referenceImageZeroPadding)
figure, imshow(inputImage)
size(inputImage)

% Fourier Transform
fourierReferenceImage = fft2(referenceImageZeroPadding);
fourierInputImage = fft2(inputImage);
```

inputImage, referenceImage의 이미지 픽셀 값 조건문으로 짝수설정

변수 이름 재설정

padarray() 사용

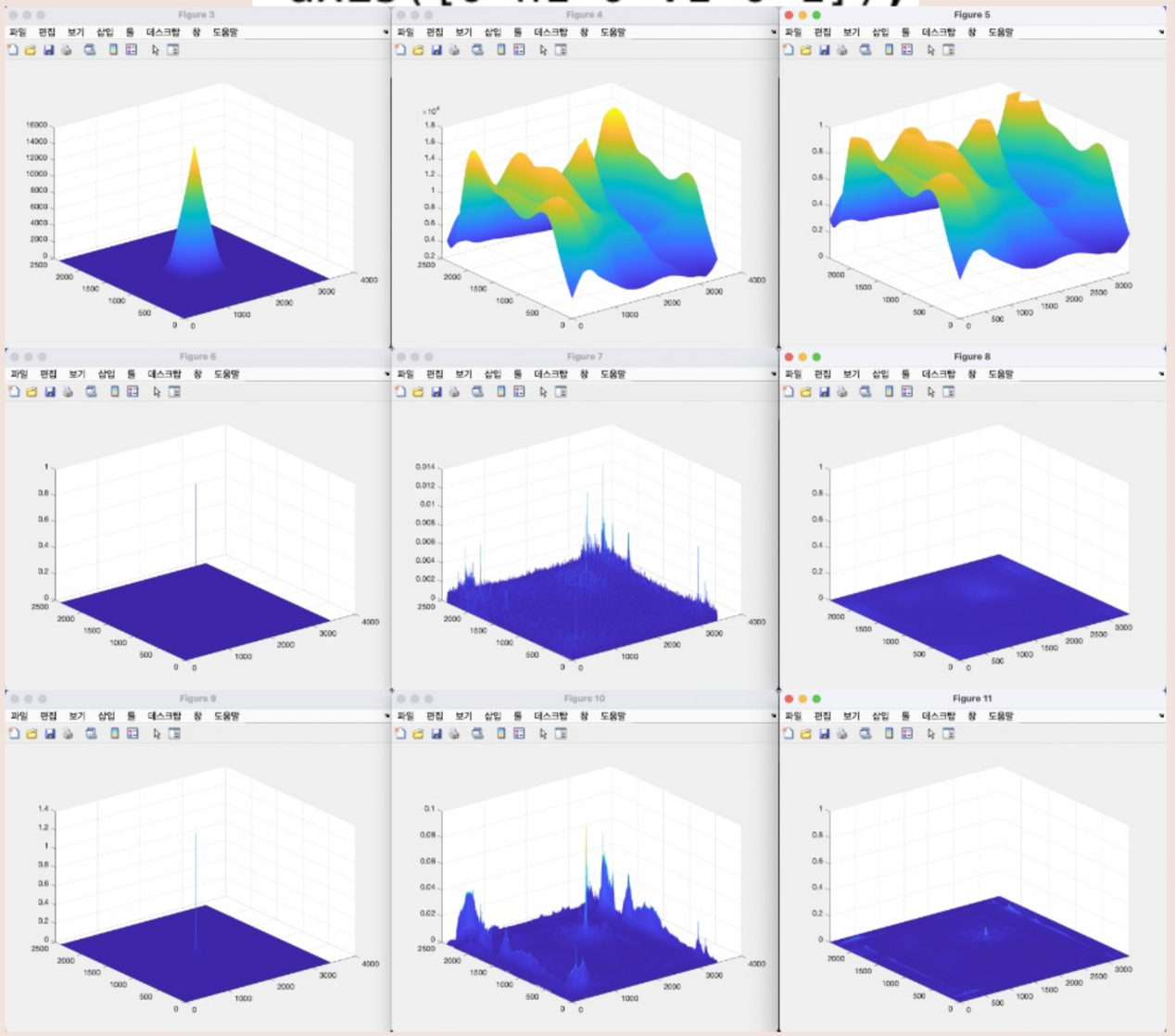
ex_test.m (수정본)

```
% Linear Correlation Filter
LCF_referenceImage = abs(fftshift(ifft2(fourierReferenceImage.*conj(fourierReferenceImage))));       % autocorrelation
LCF_inputImage = abs(fftshift(ifft2(fourierInputImage.*conj(fourierReferenceImage))));       % cross-correlation
max_LCF = max(LCF_referenceImage(:));   % Maximum of autocorrelation
figure, mesh(LCF_referenceImage);
figure, mesh(LCF_inputImage);
figure, mesh(LCF_inputImage./max_LCF);
max_LCF
axis([300 h1-300 300 v1-300]);  % Axis (IMPORTANT)

% Phase Only Filter
POF_referenceImage = abs(fftshift(ifft2(exp(1i.*(angle(fourierReferenceImage) - angle(fourierReferenceImage))))));
POF_inputImage = abs(fftshift(ifft2(exp(1i.*(angle(fourierInputImage) - angle(fourierReferenceImage))))));       % cros
max_POF = max(POF_referenceImage(:));   % Maximum of autocorrelation
figure, mesh(POF_referenceImage);
figure, mesh(POF_inputImage);
figure, mesh(POF_inputImage./max_POF);
max_POF
axis([300 h1-300 300 v1-300]);  % Axis (IMPORTANT)

% Nonlinear Correlation Filter
k = 0.3;
NLF_referenceImage = abs(fftshift(ifft2(((abs(fourierReferenceImage).*abs(fourierReferenceImage)).^k).*exp(1i.*(angle(f
NLF_inputImage = abs(fftshift(ifft2(((abs(fourierInputImage).*abs(fourierReferenceImage)).^k).*exp(1i.*(angle(fourierIn
max_NLF = max(NLF_referenceImage(:));   % Maximum of autocorrelation
figure, mesh(NLF_referenceImage);
figure, mesh(NLF_inputImage);
figure, mesh(NLF_inputImage./max_NLF);
max_NLF
axis([300 h1-300 300 v1-300]);  % Axis (IMPORTANT)
```

axis()인자를 이미지 픽셀을 의미하는 변수로 선언하여 여러 입력 이미지에 대하여 성립

axis() 수정 전

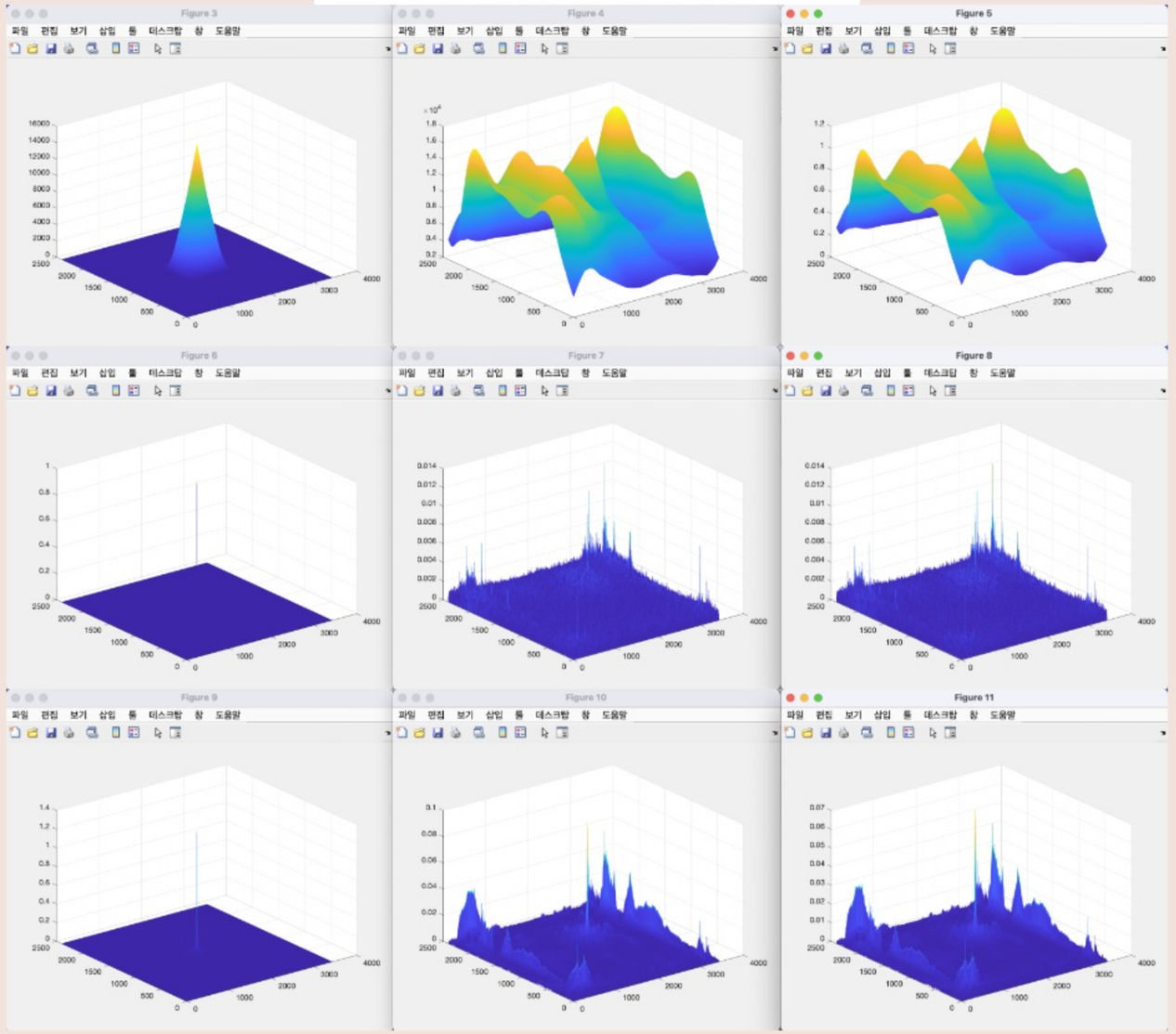
```
axis([0 h1 0 v1 0 1]);
```



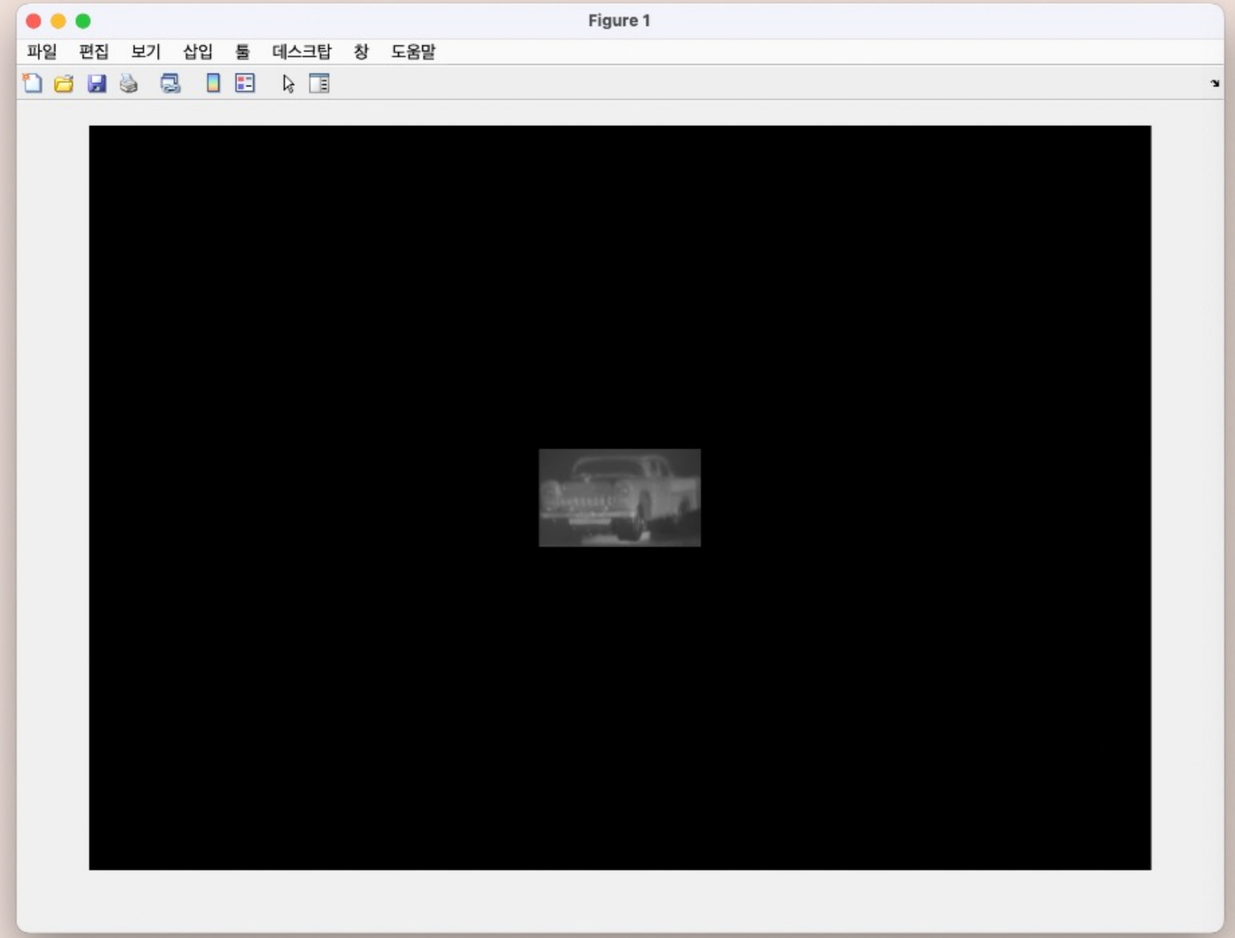
(high_nonuniform_455mm + high_target)

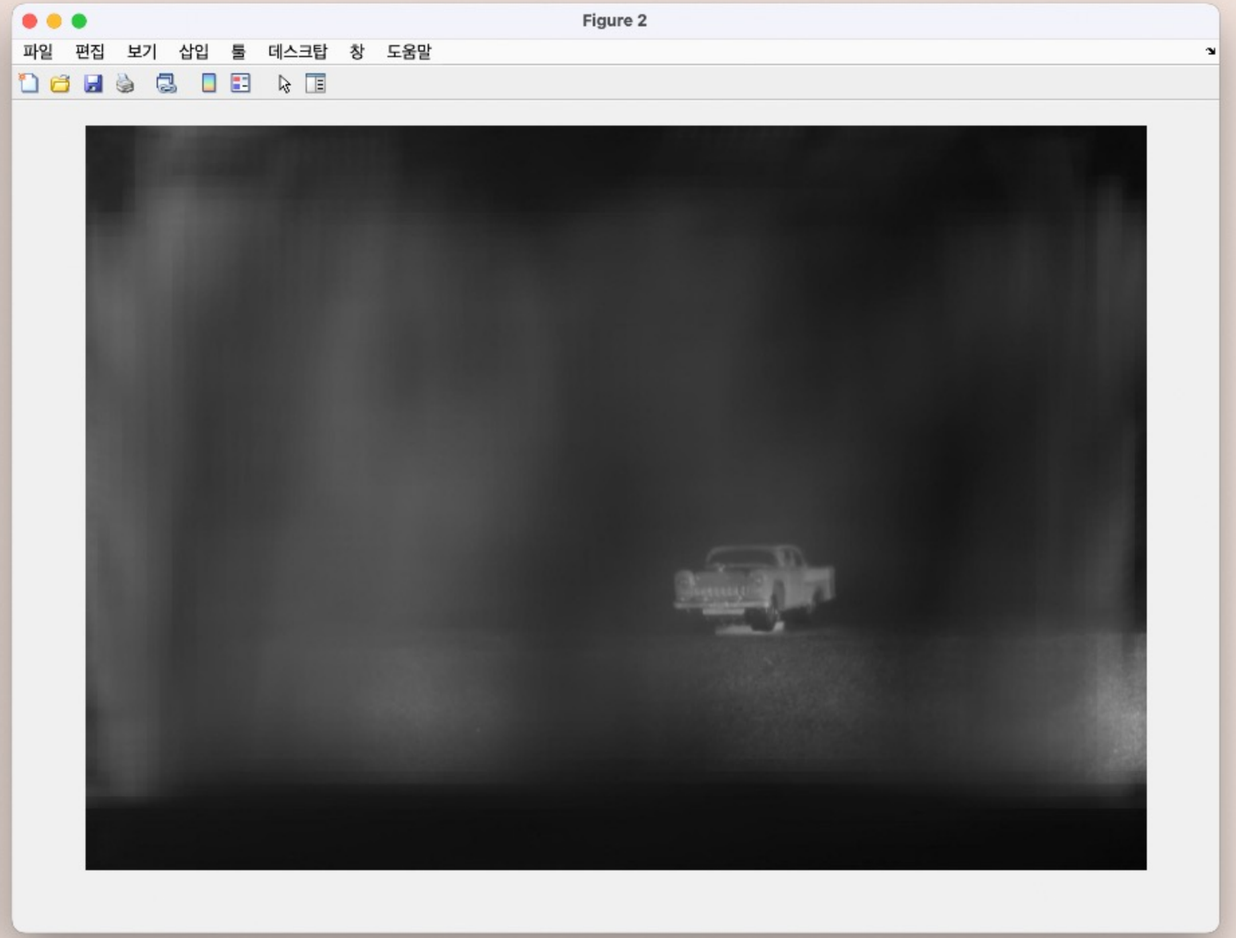
axis() 1차 수정 (z축 없앰)

```
axis([0 h1 0 v1]);
```



(high_nonuniform_455mm + high_target)

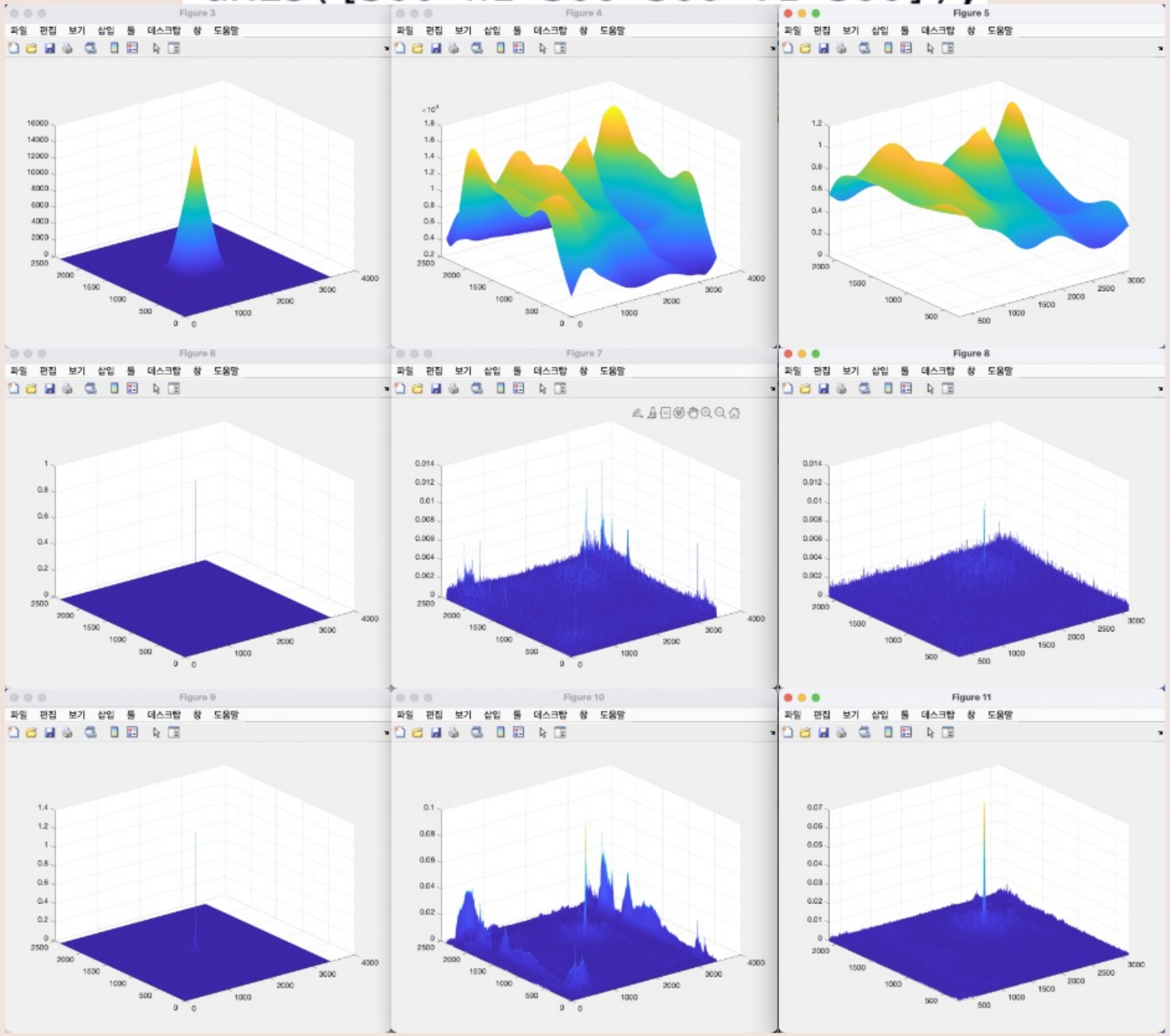




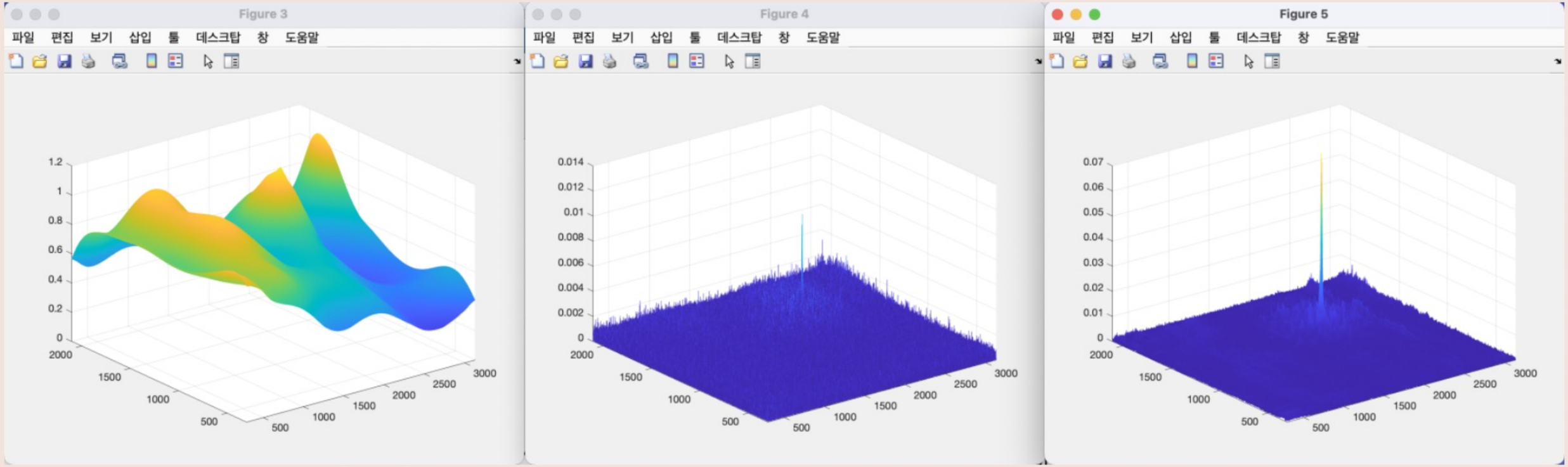
패딩된 부분이 코릴레이션 될 가능성 고려

axis() ㅅㅜ 후(x,y축 300ㅇㅣㄴ 비z축ㅛㅇ ㅓ ㅂ

```
axis([300 h1-300 300 v1-300]);
```



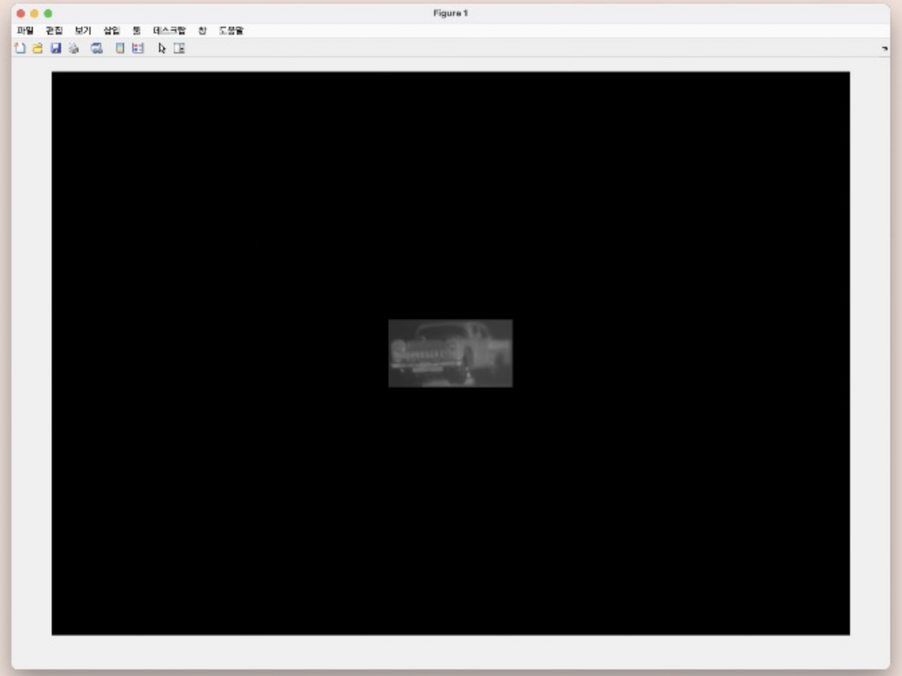
(high_nonuniform_455mm + high_target)

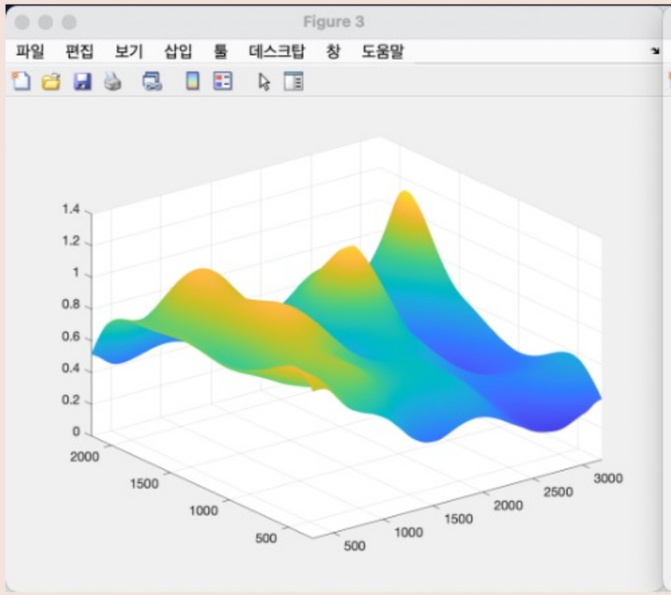


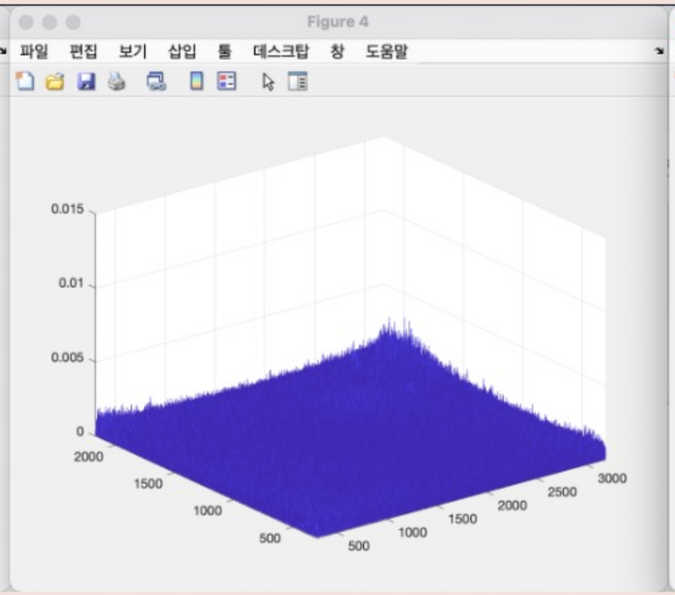
(high_nonuniform_455mm + high_target)

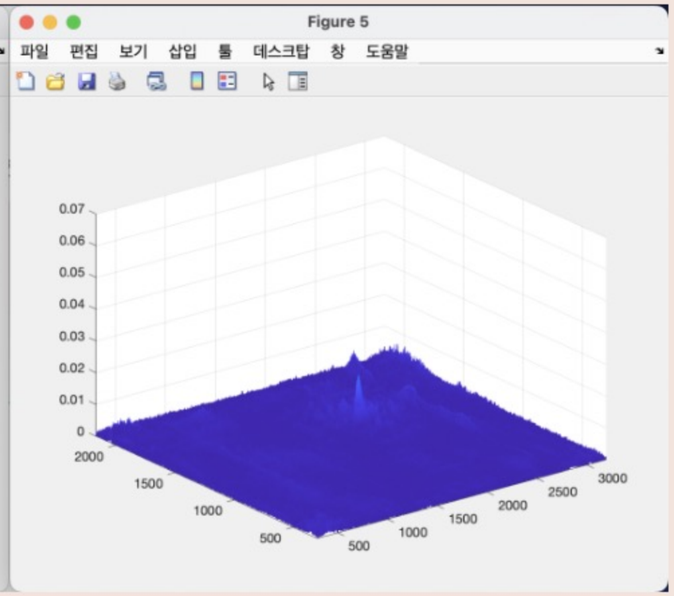
이미지 선정기준

물체의 깊이를 나타내는 변수 depth에 따라 나누어진 Input Image를
Reference Image와 상관관계를 확인하는 필터를 거쳐 그 값이
가장 높은 Input Image를 채택하여 영상의 물체의 깊이를 추정하였다.

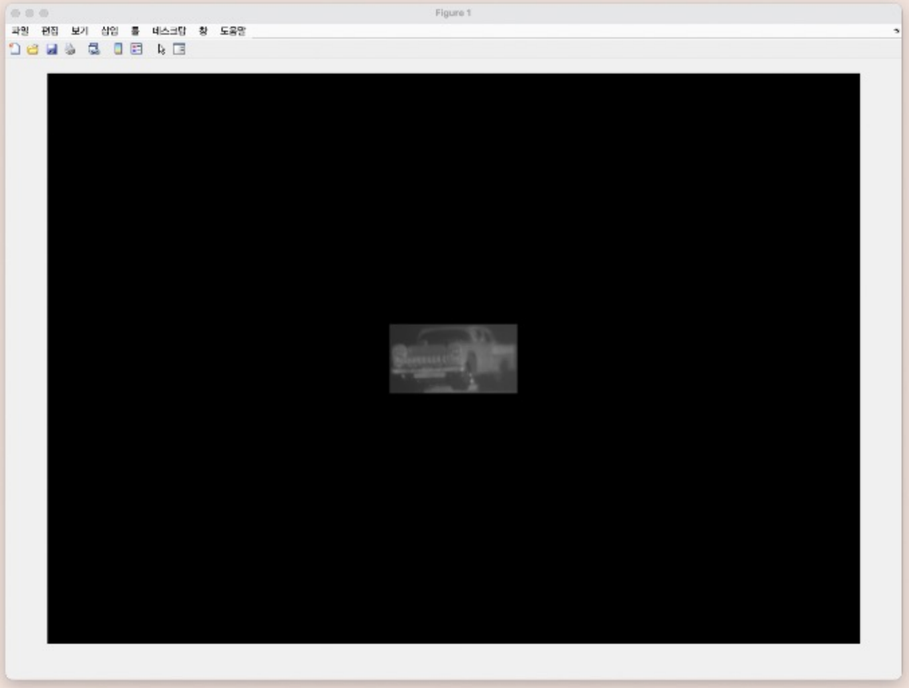


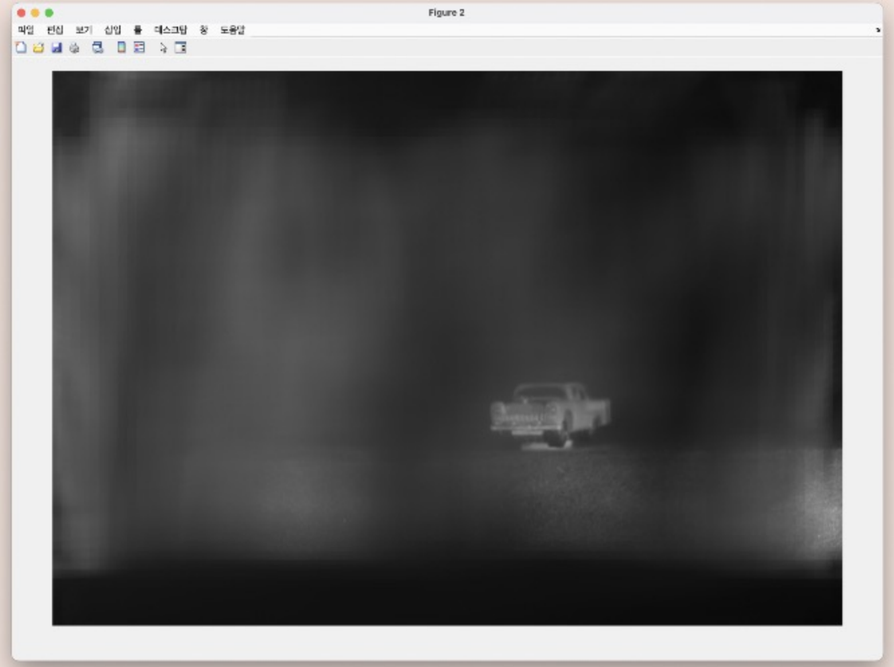


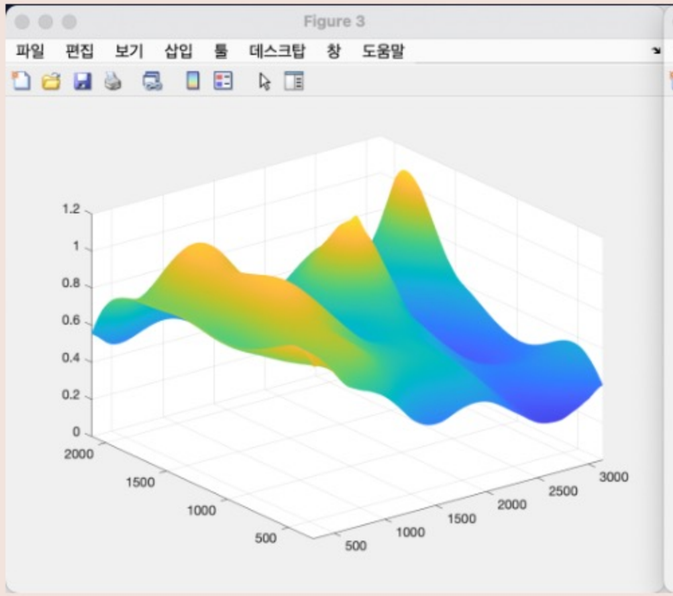


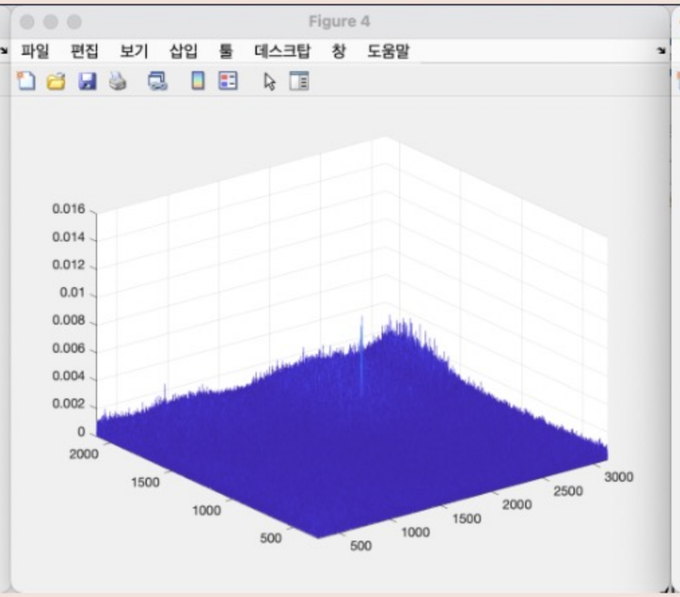


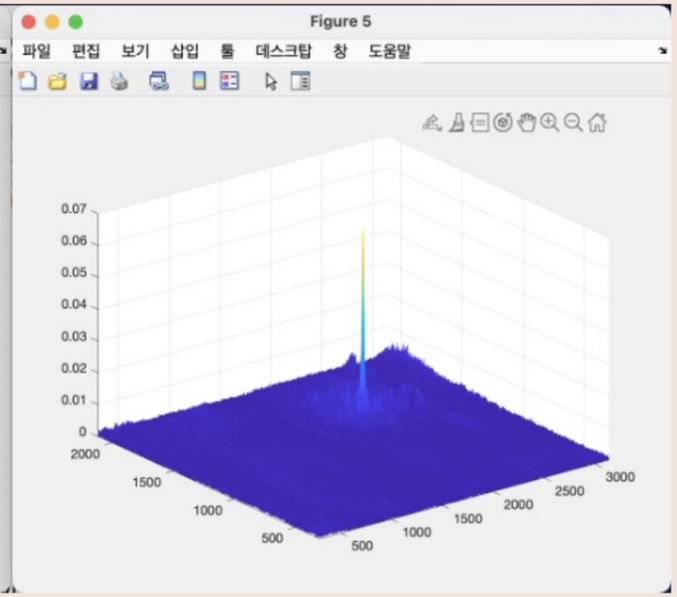
(medium_fixed_385mm + medium_target)





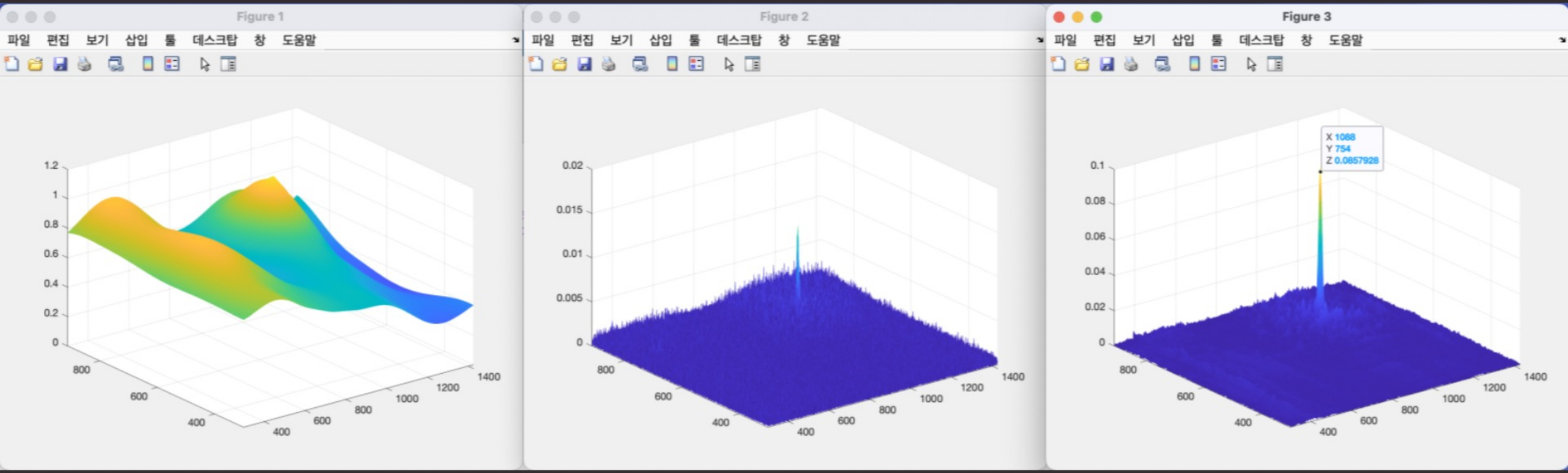




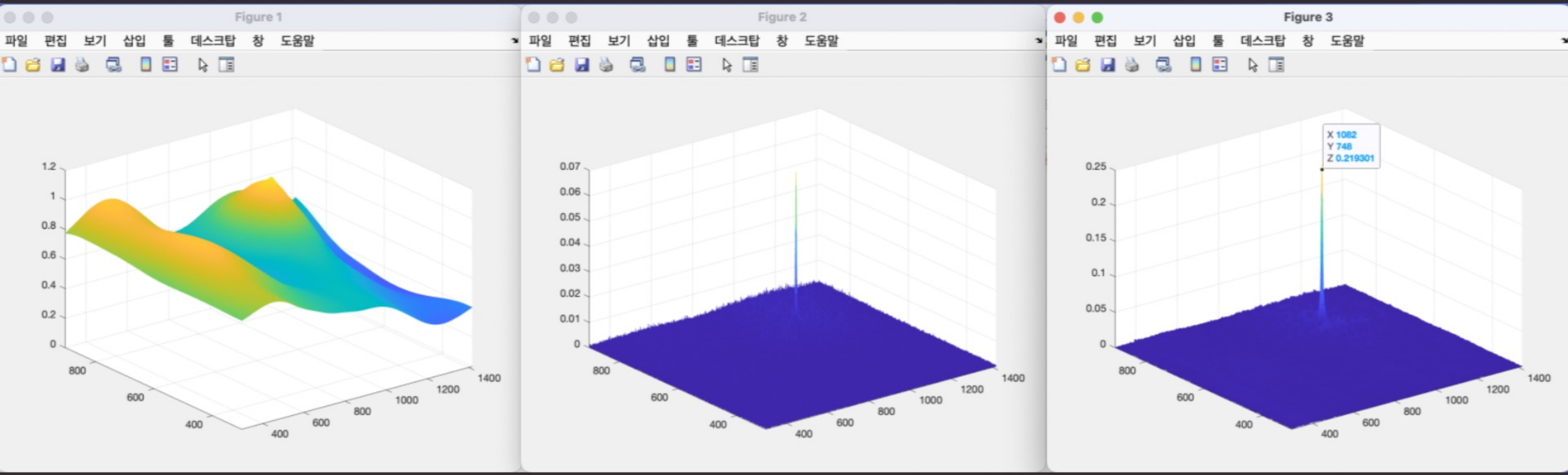


(medium_fixed_440mm + medium_target)

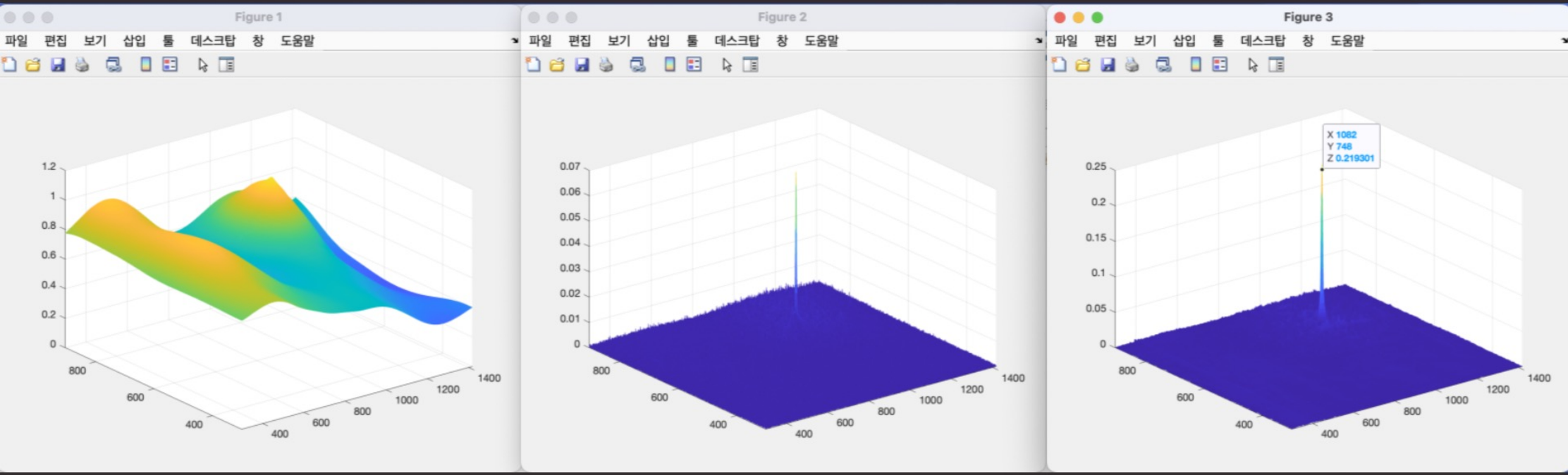
medium_fixed + medium_target



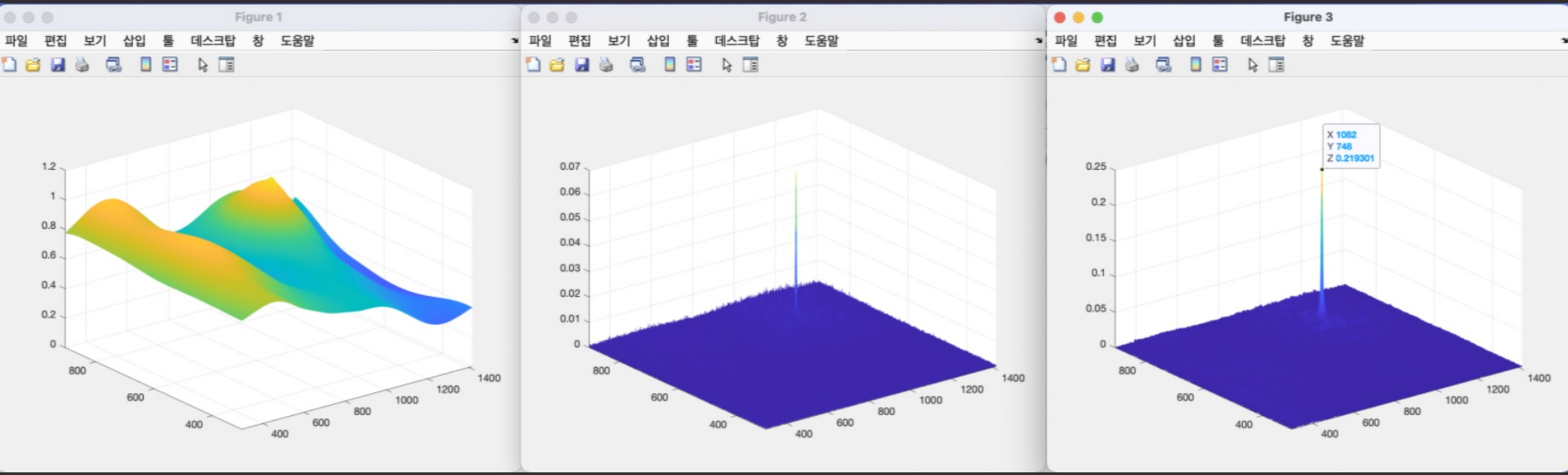
(medium_fixed_435mm + medium_target)



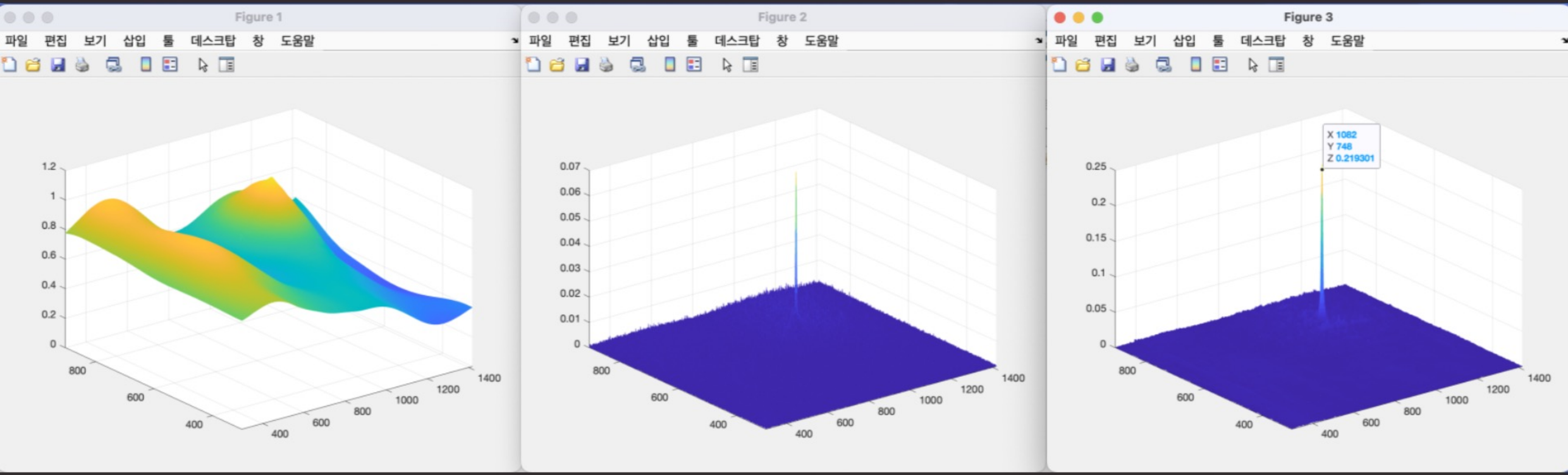
(medium_fixed_440mm + medium_target)



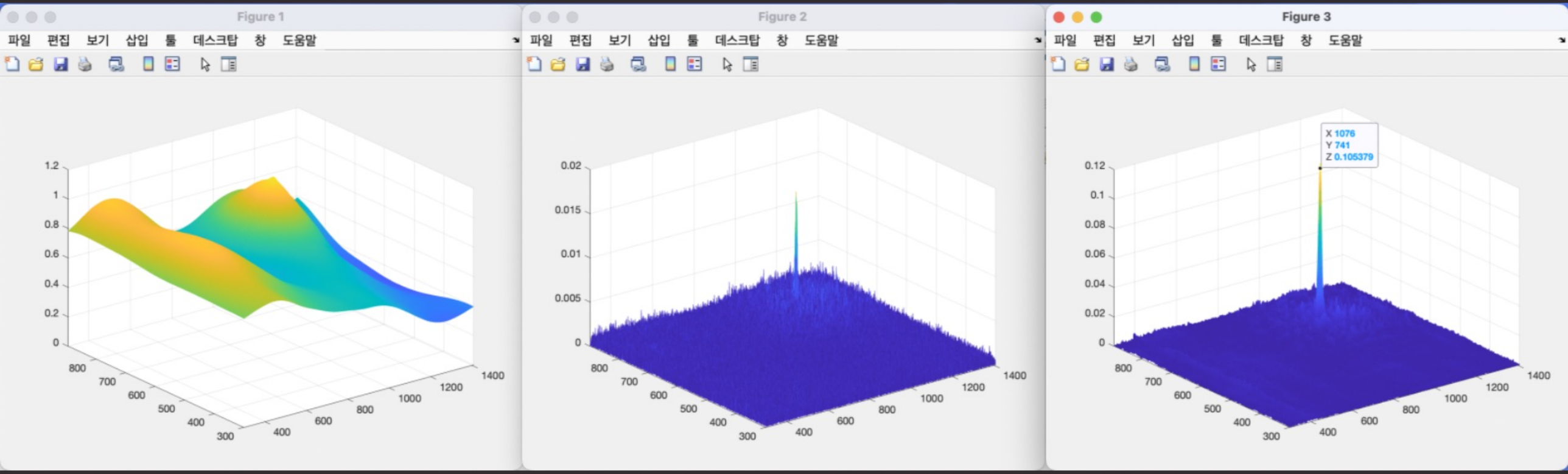
(medium_fixed_445mm + medium_target)



(medium_fixed_450mm + medium_target)



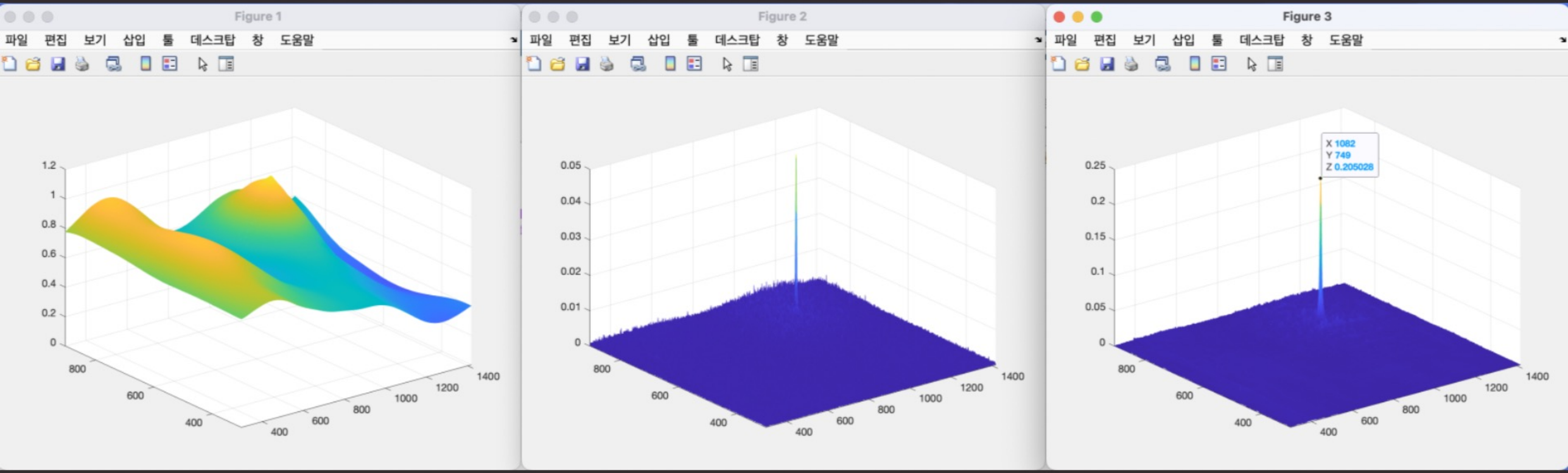
(medium_fixed_455mm + medium_target)



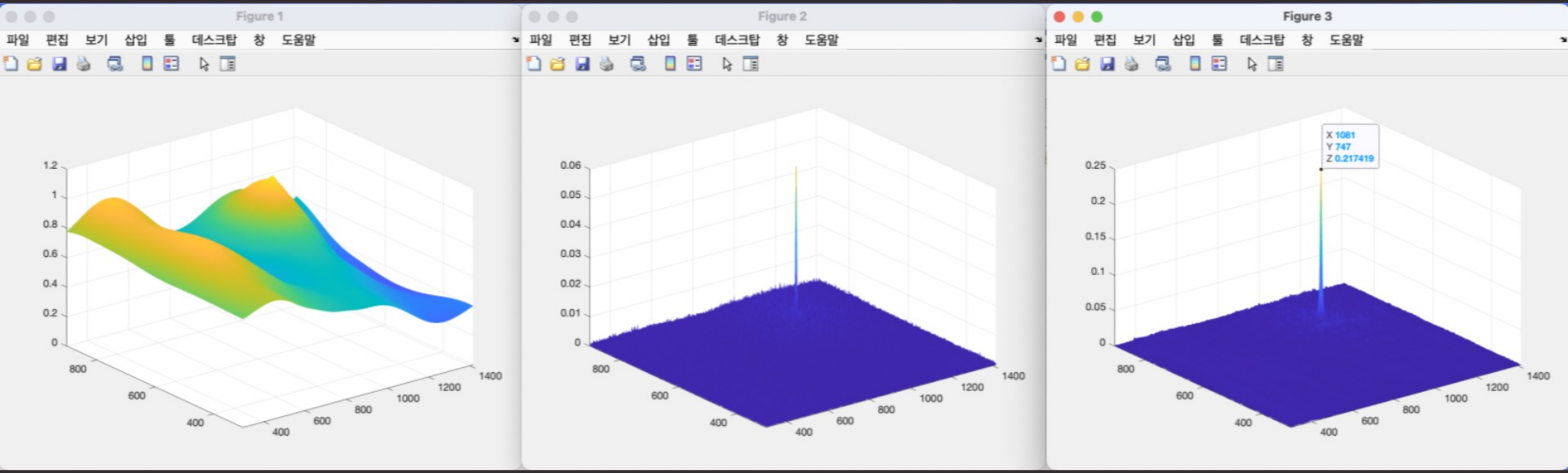
(medium_fixed_460mm + medium_target)

medium_fixed + medium_target
: depth = 440mm ~ 455mm

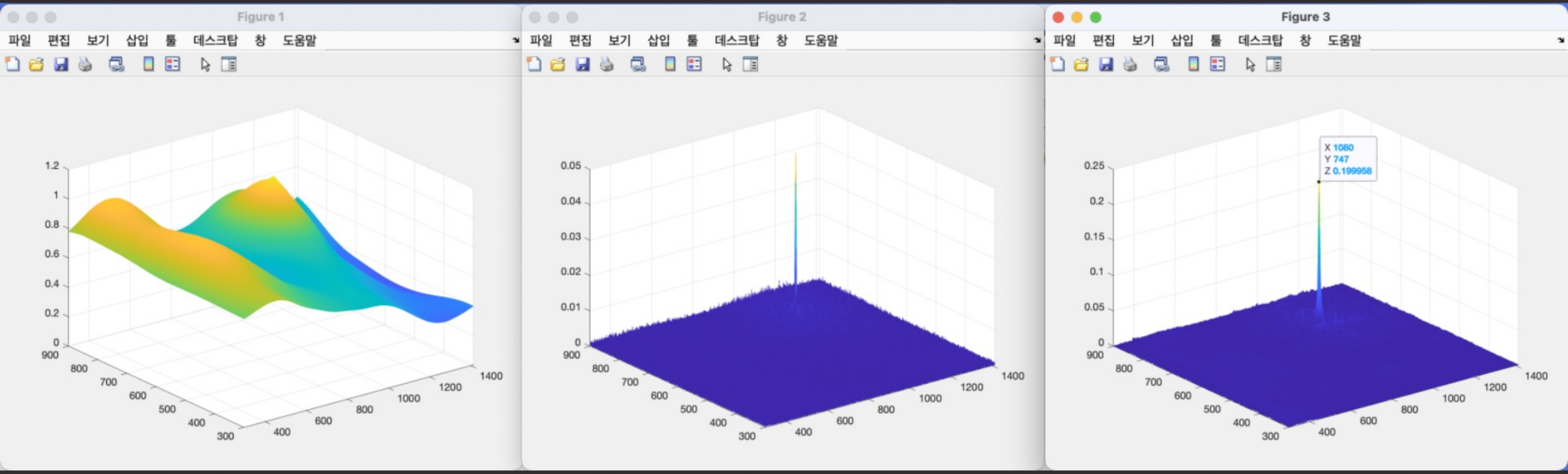
medium_nonuniform + medium_target



(medium_nonuniform_445mm + medium_target)



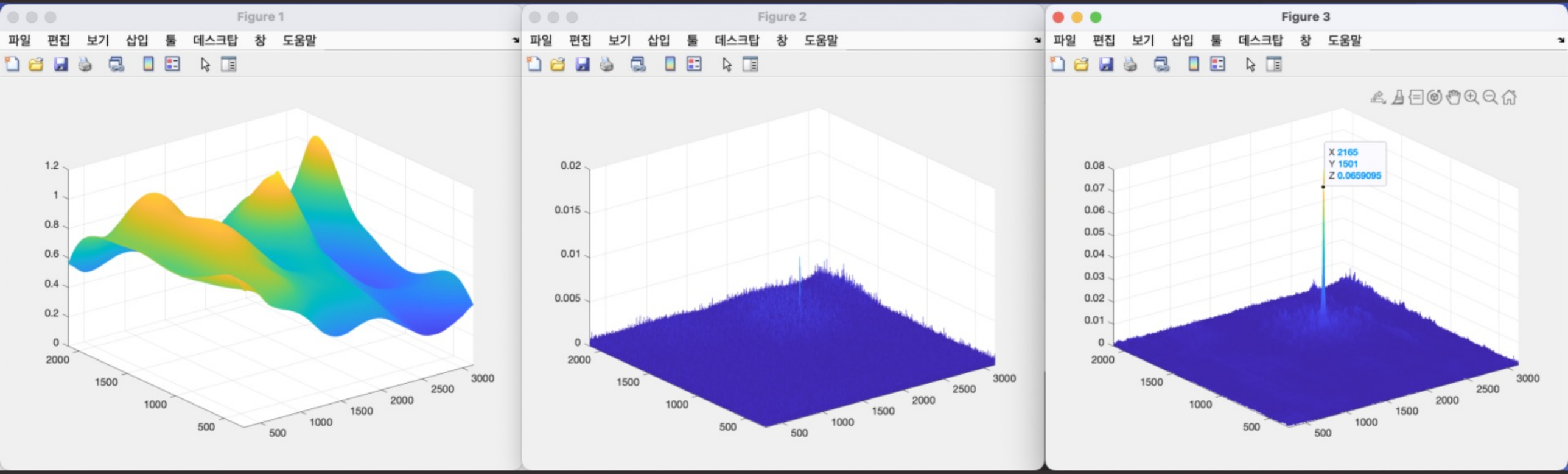
(medium_nonuniform_450mm + medium_target)



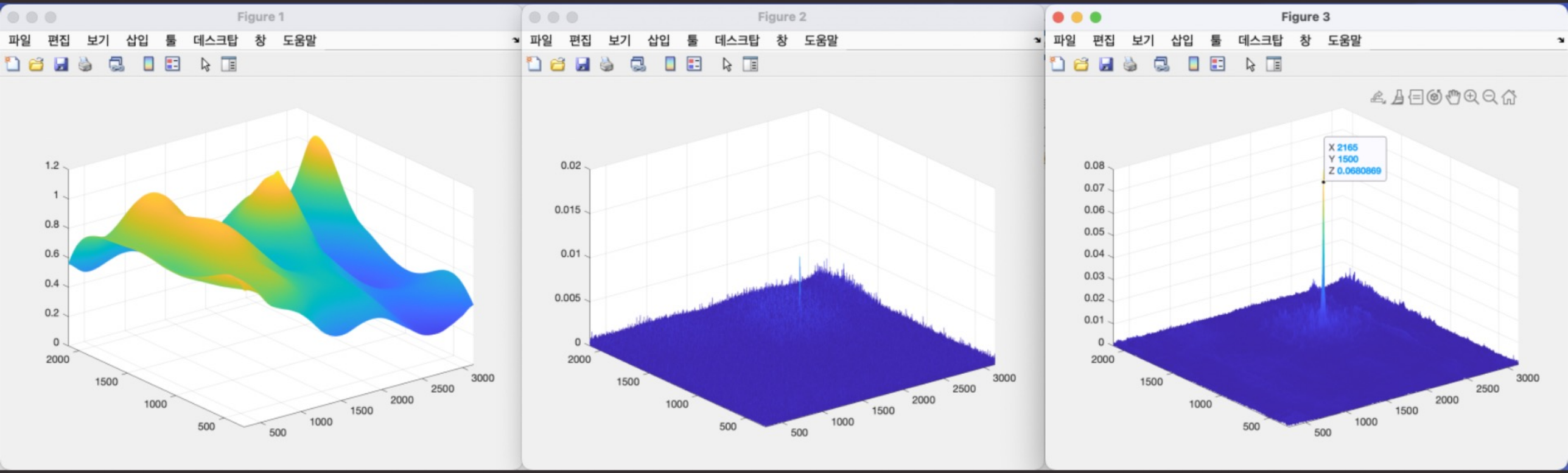
(medium_nonuniform_455mm + medium_target)

medium_nonuniform + medium_target
: depth = 450mm

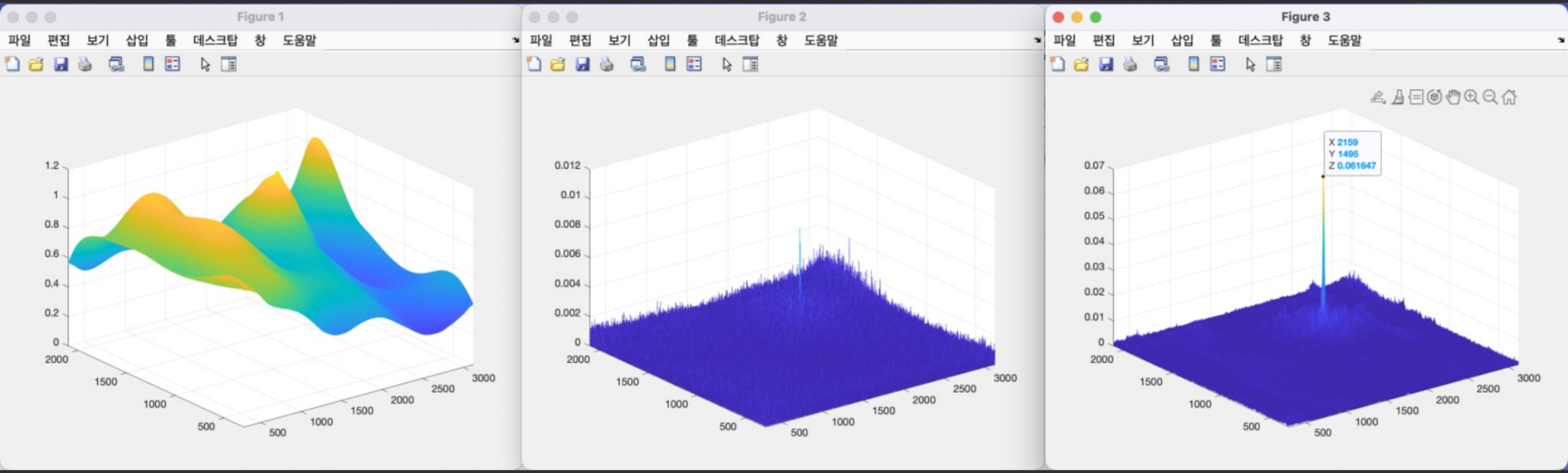
high_fixed + high_target



(high_fixed_445mm + high_target)



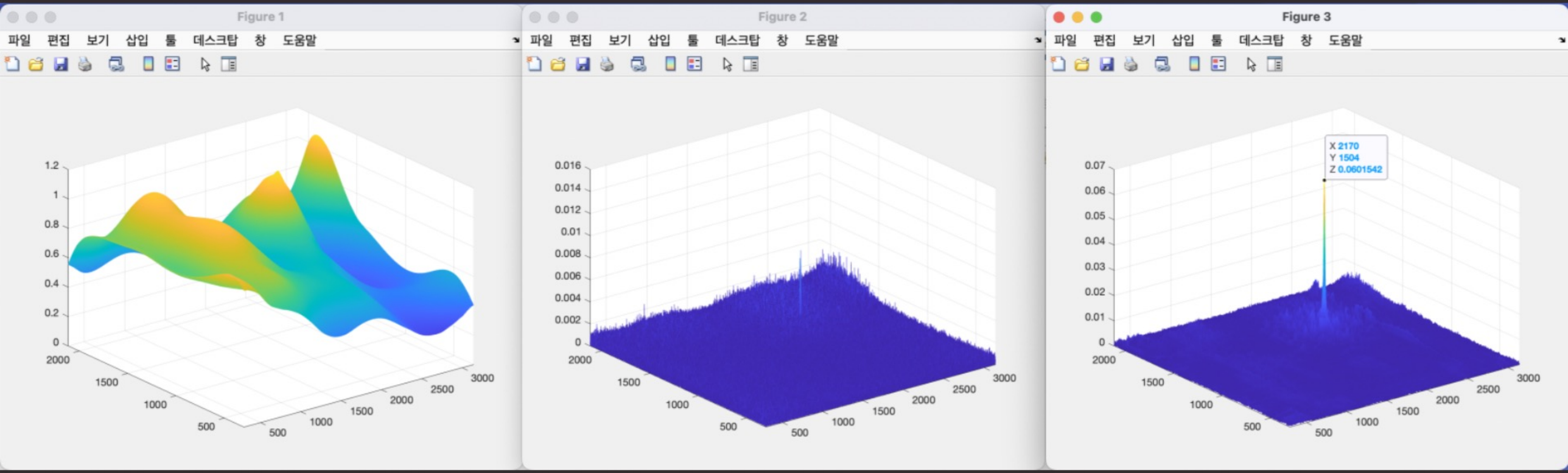
(high_fixed_450mm + high_target)



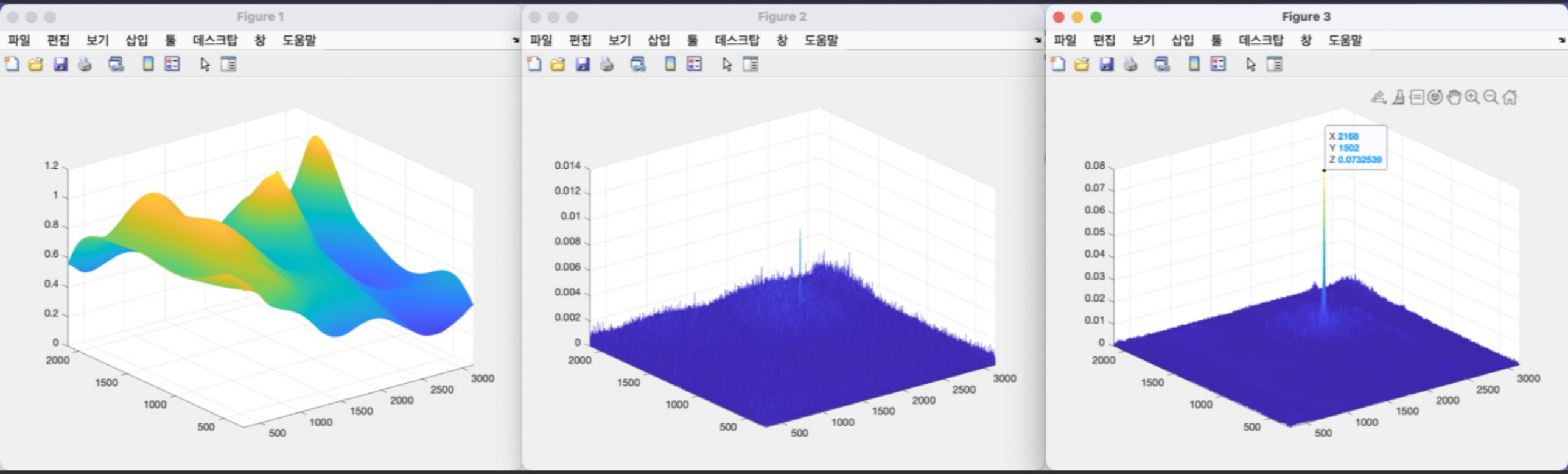
(high_fixed_455mm + high_target)

high_fixed + high_target
: depth = 450mm

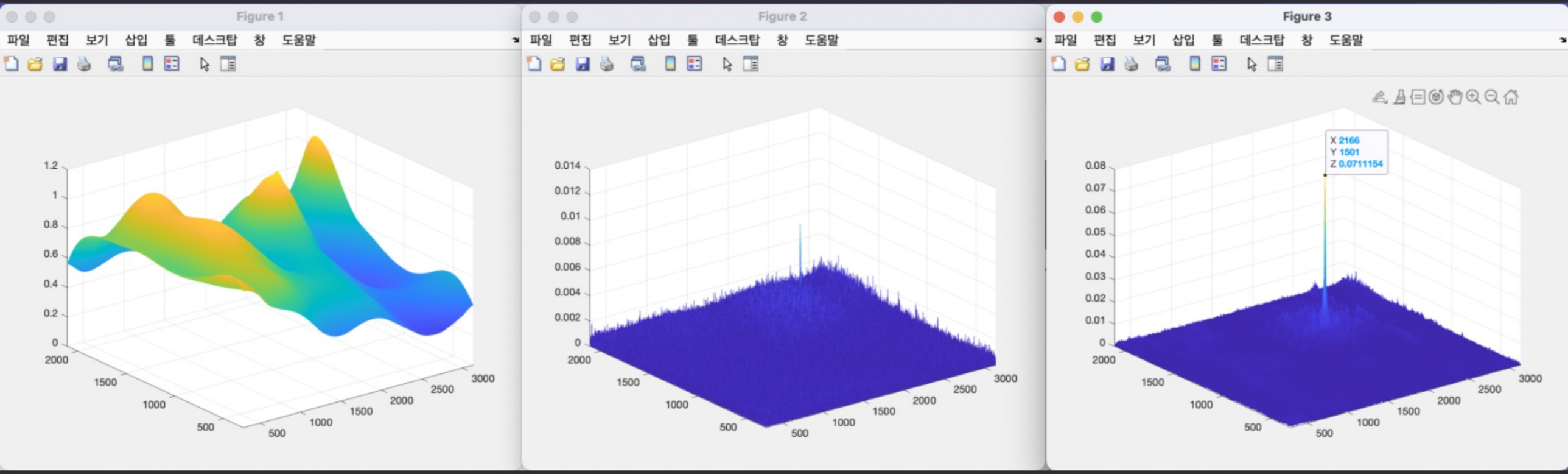
high_nonuniform + high_target



(high_nonuniform_440mm + high_target)



(high_nonuniform_445mm + high_target)



(high_nonuniform_450mm + high_target)

high_nonuniform + high_target
: depth = 445mm

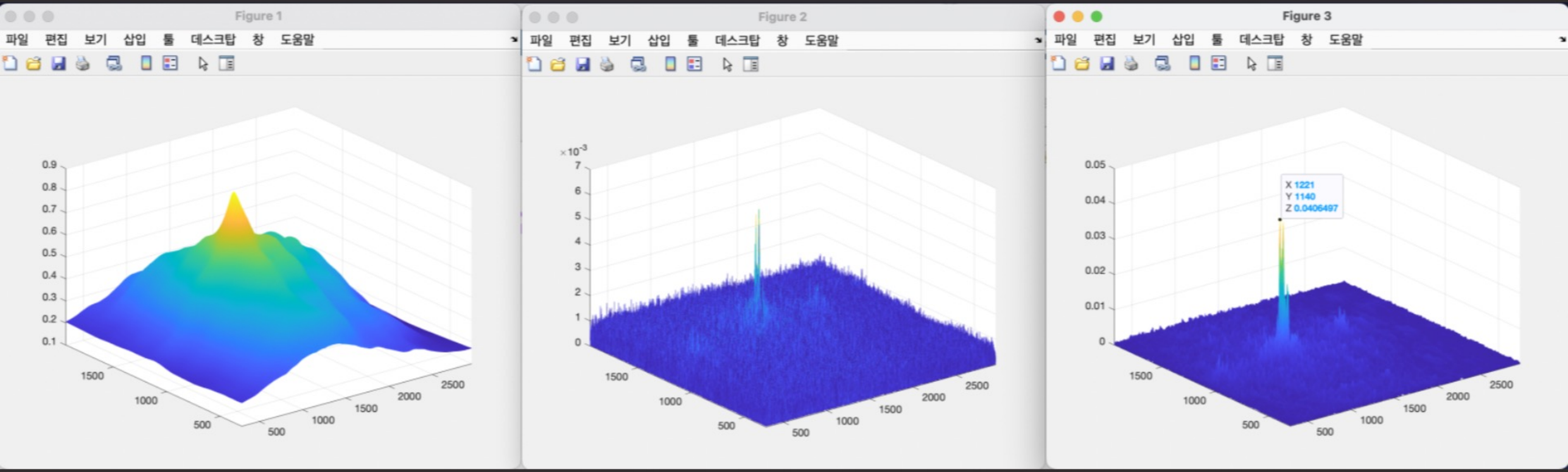
	midium	high
fixed	440~455mm	450mm
nonuniform	450mm	445mm

(first_target)

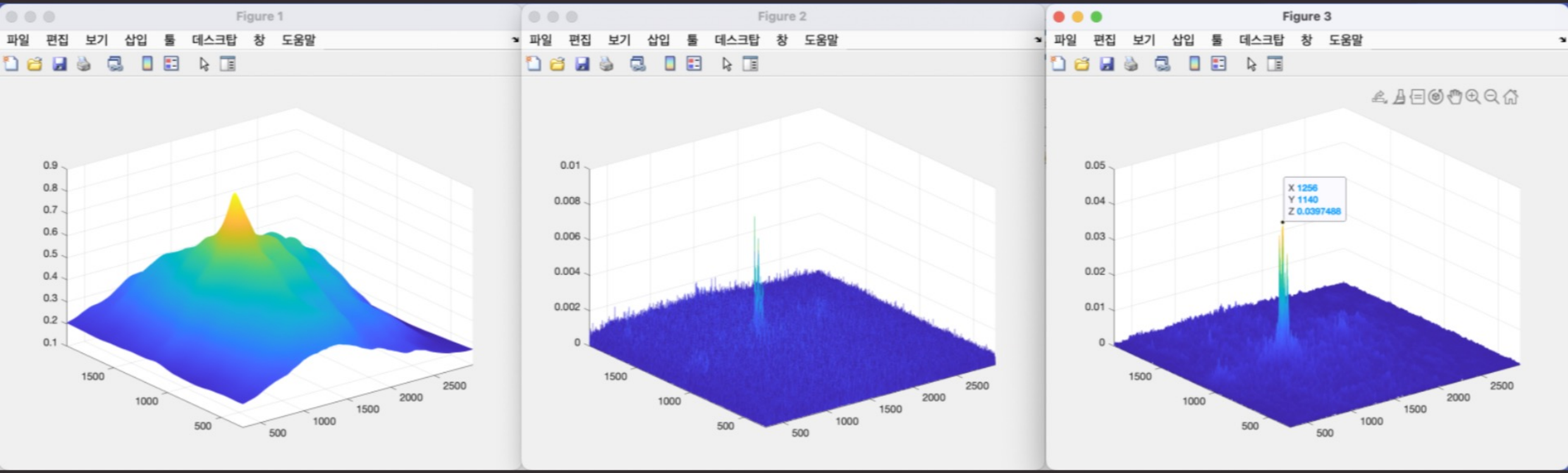
(second_target)

(third_target)

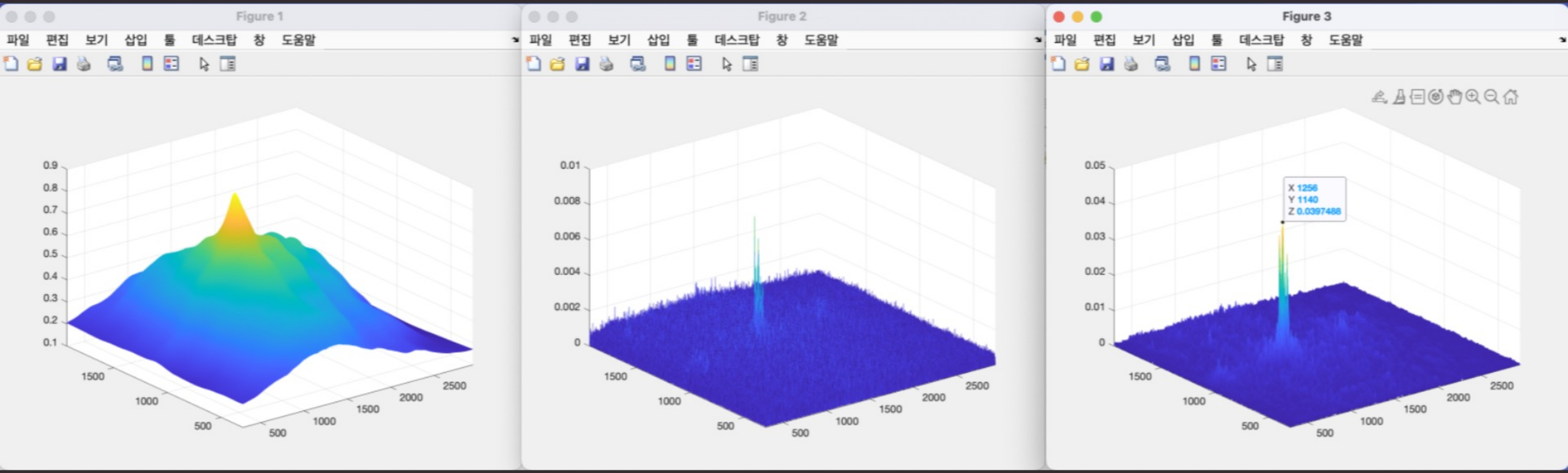
3character_fixed + firsh_target



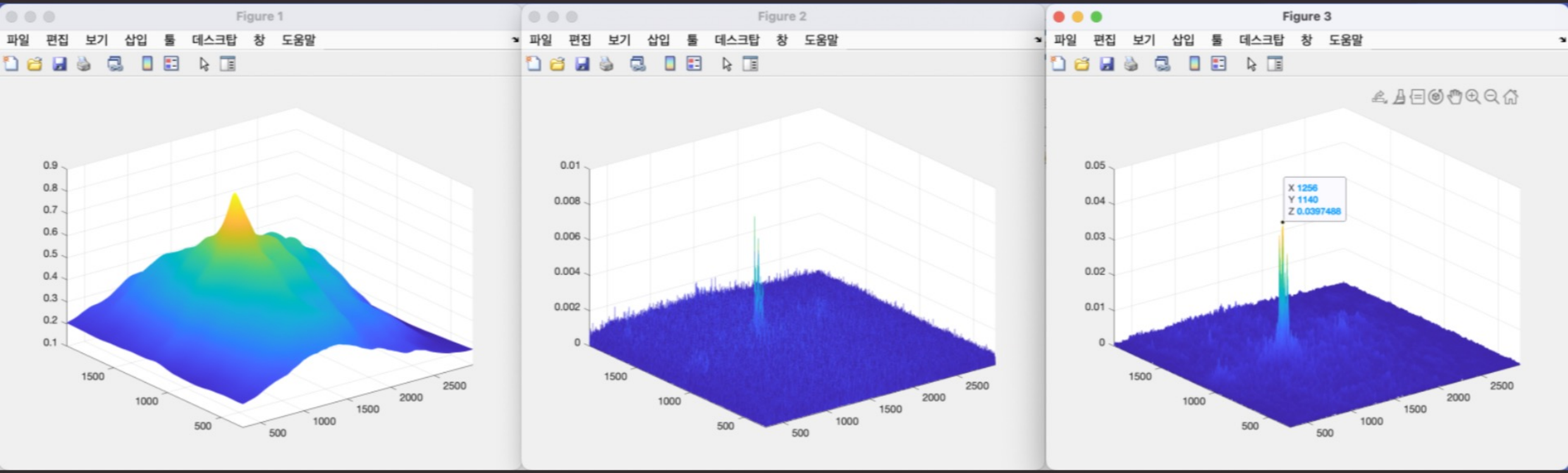
(3character_fixed_660mm + first_target)



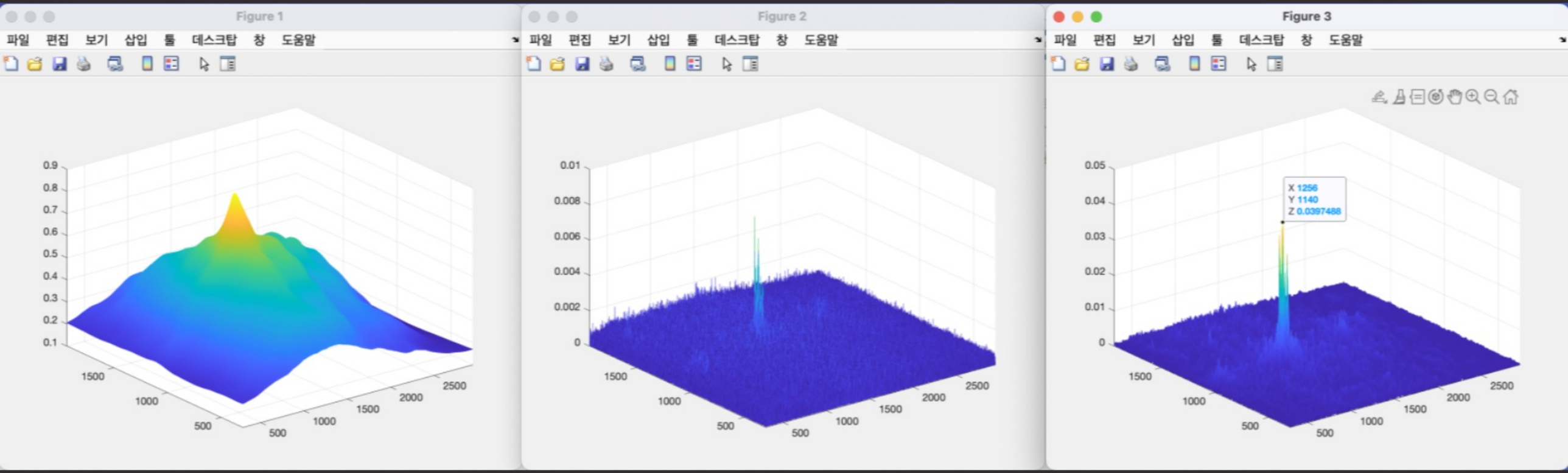
(3character_fixed_665mm + first_target)



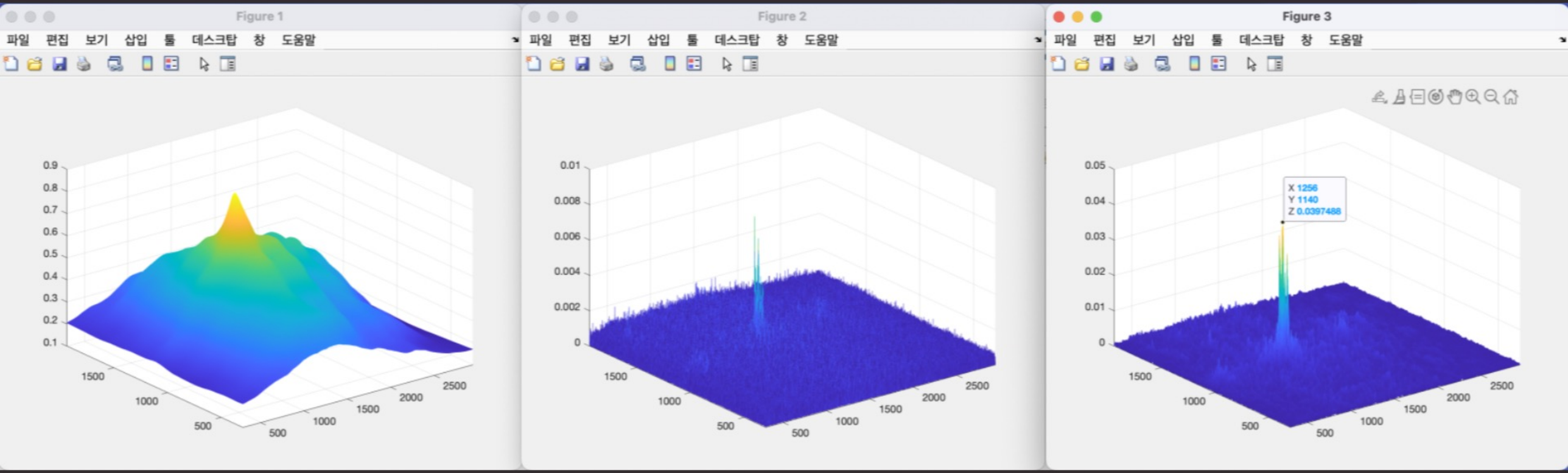
(3character_fixed_670mm + first_target)



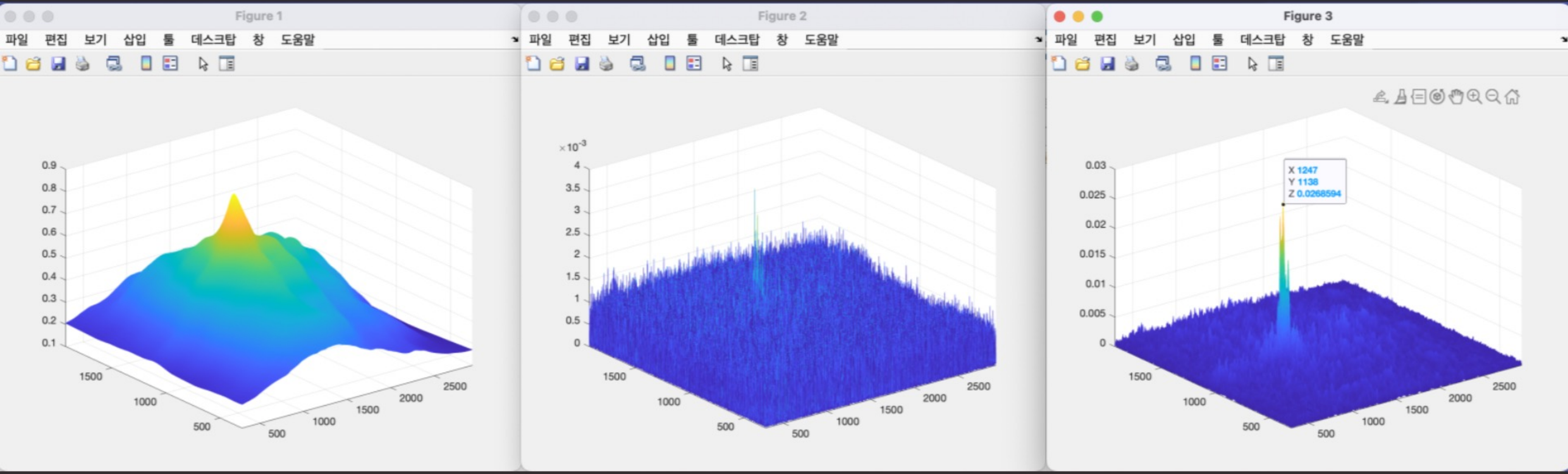
(3character_fixed_675mm + first_target)



(3character_fixed_680mm + first_target)



(3character_fixed_685mm + first_target)

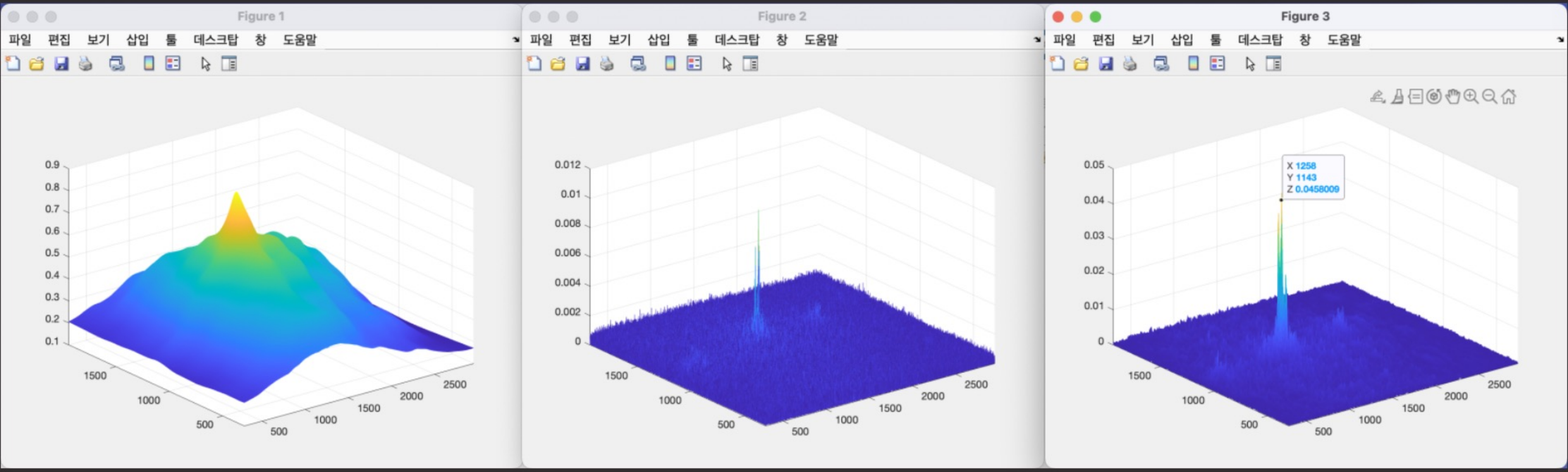


(3character_fixed_690mm + first_target)

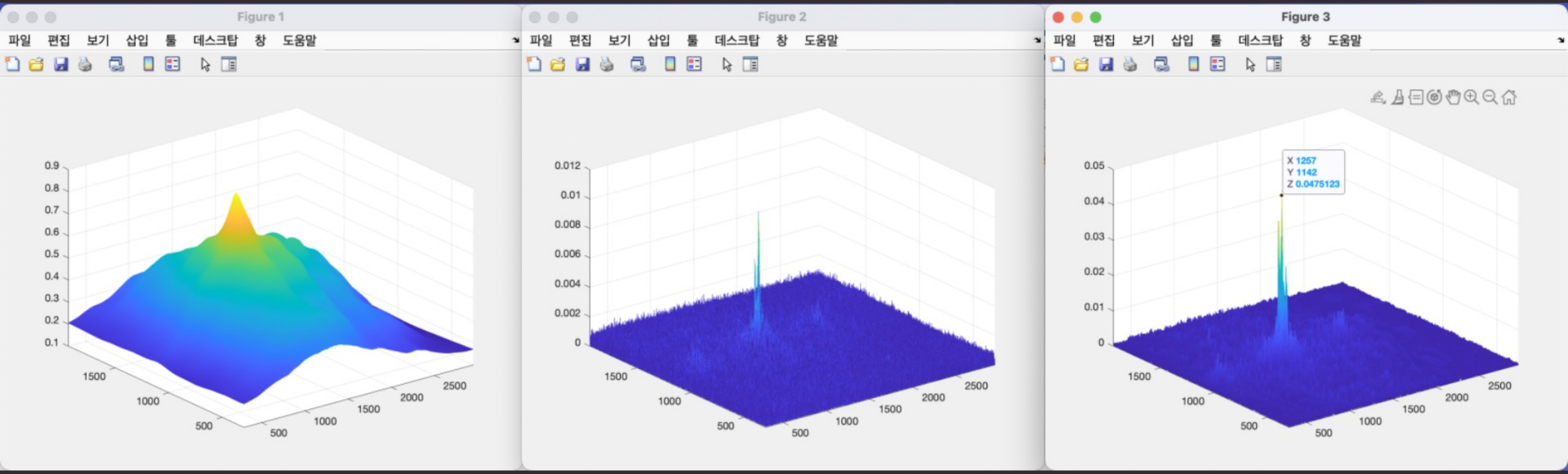
3character_fixed + firsh_target

: depth = 665mm ~ 685mm

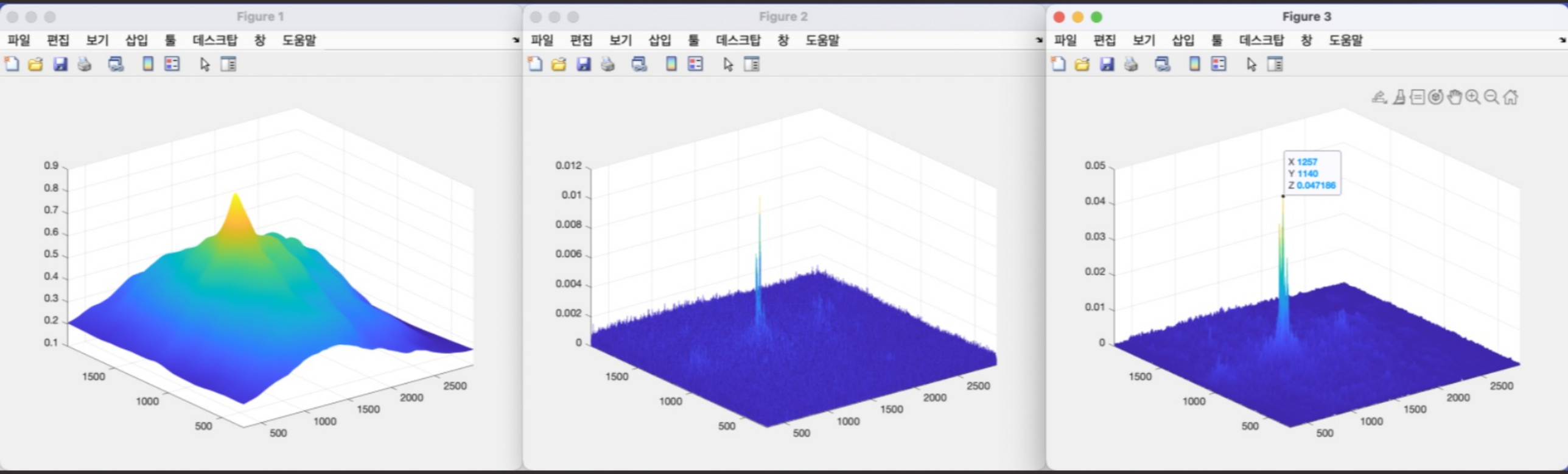
3character_nonuniform + first_target



(3character_nonuniform_655mm + first_target)



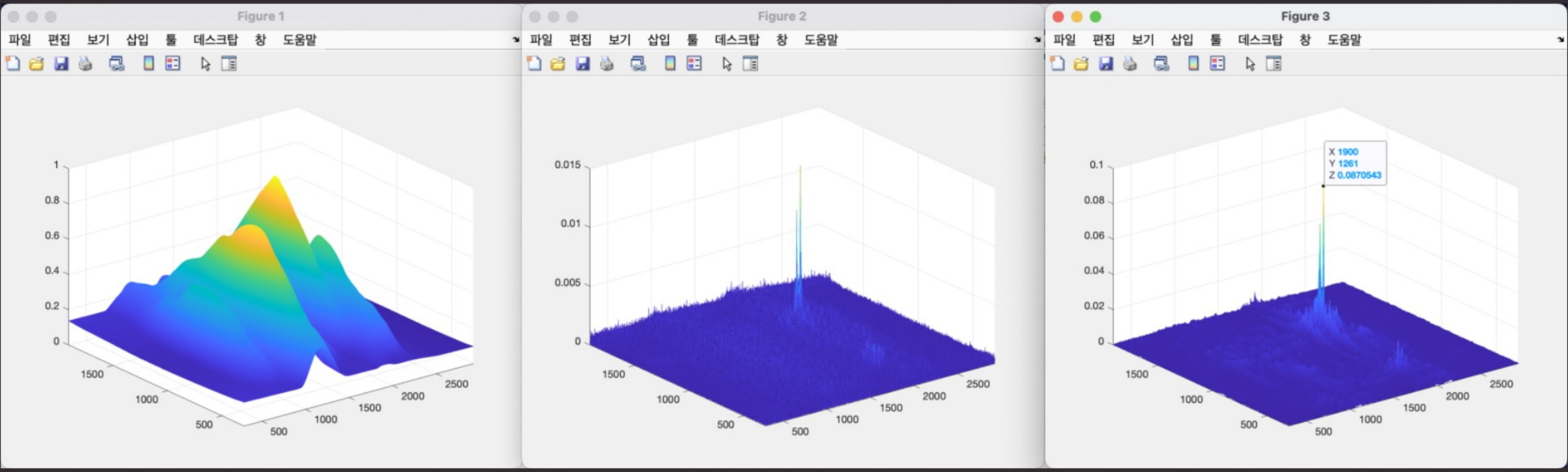
(3character_nonuniform_660mm + first_target)



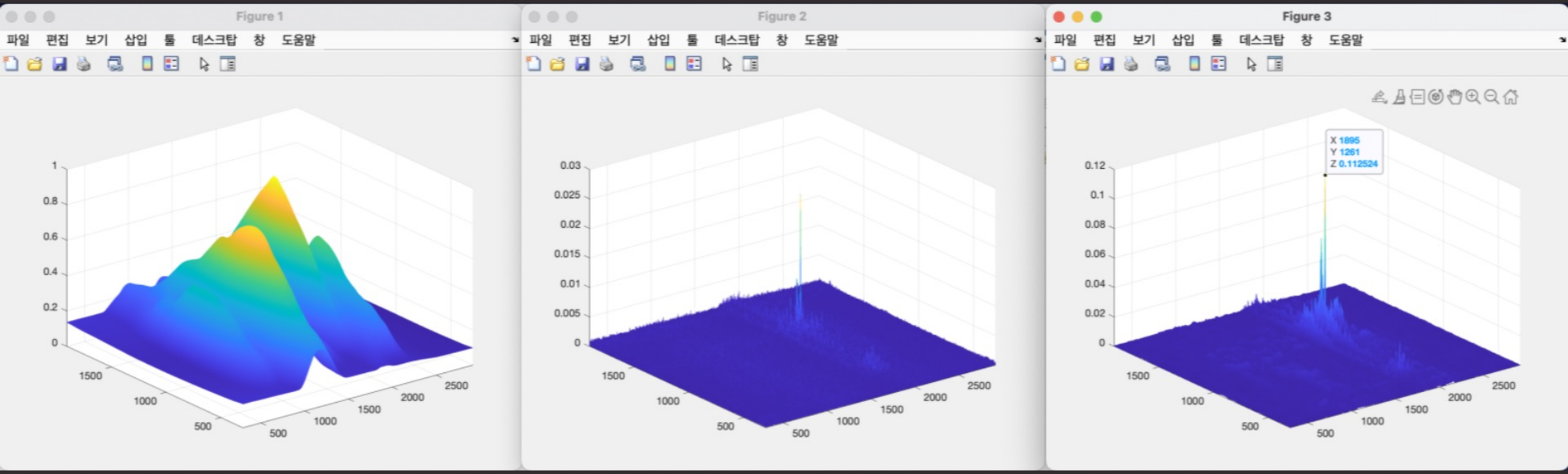
(3character_nonuniform_665mm + first_target)

3character_nonuniform + first_target
: depth = 660mm

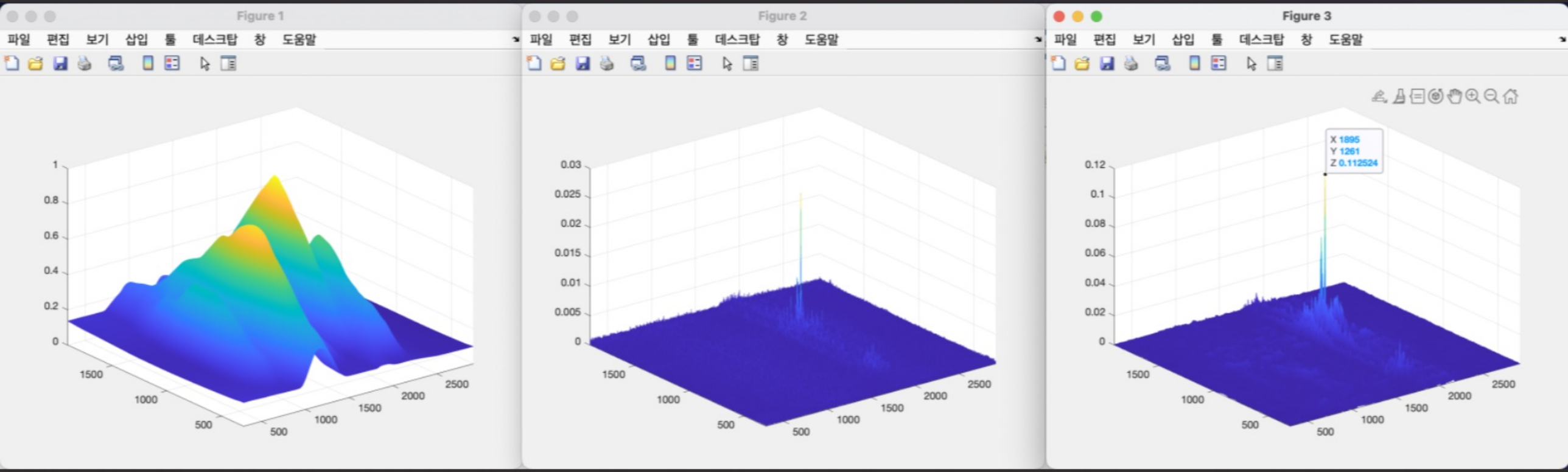
3character_fixed + second_target



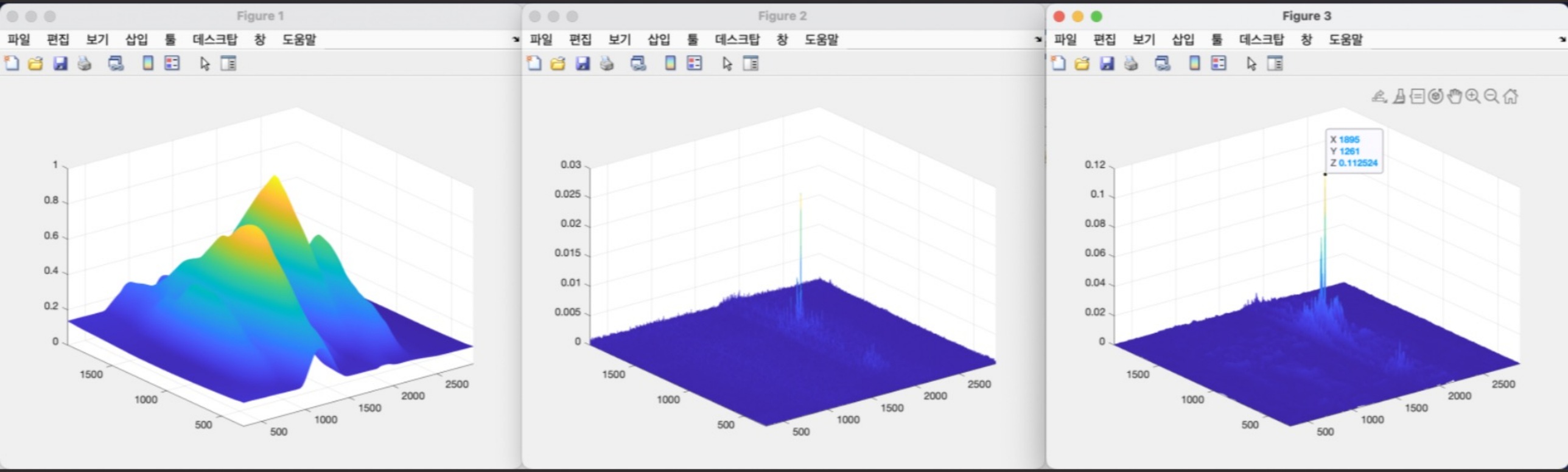
(3character_fixed_745mm + second_target)



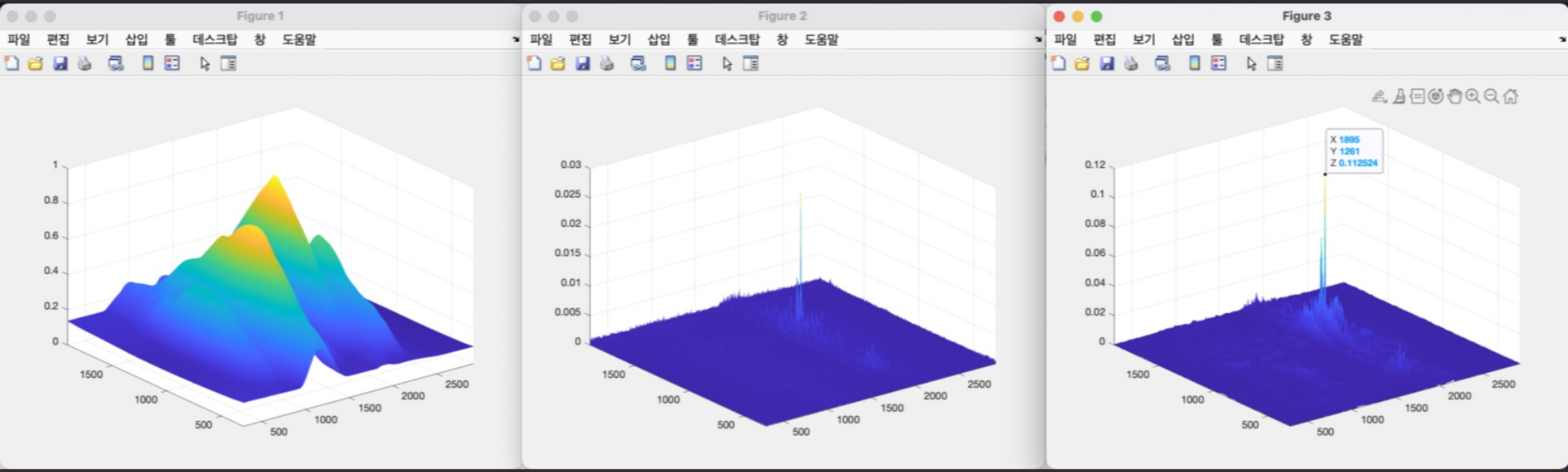
(3character_fixed_750mm + second_target)



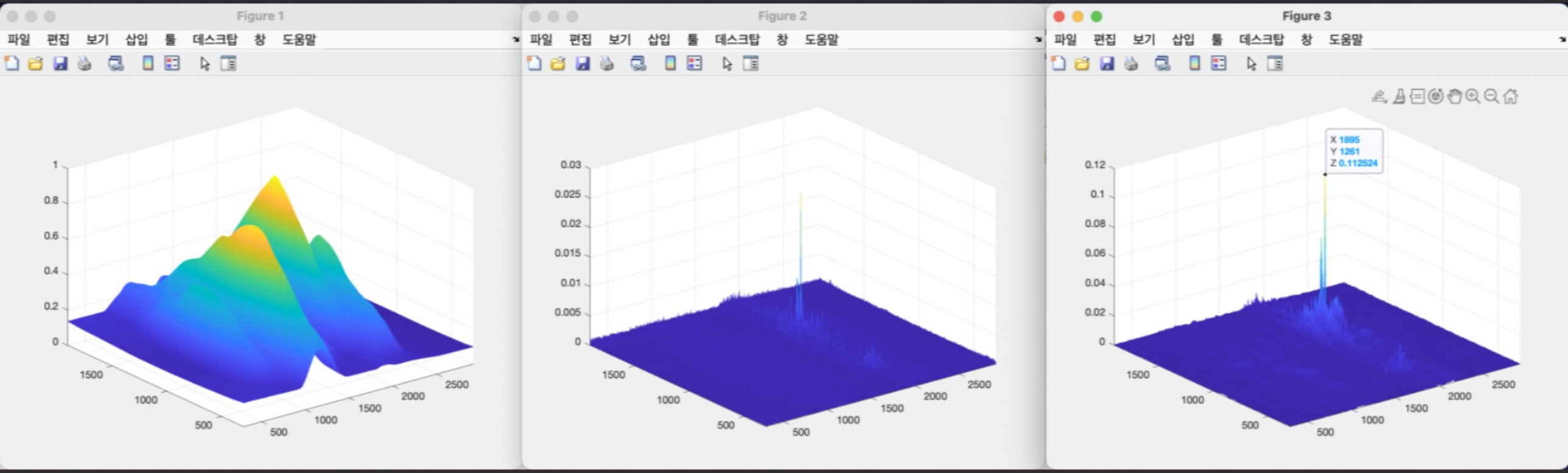
(3character_fixed_755mm + second_target)



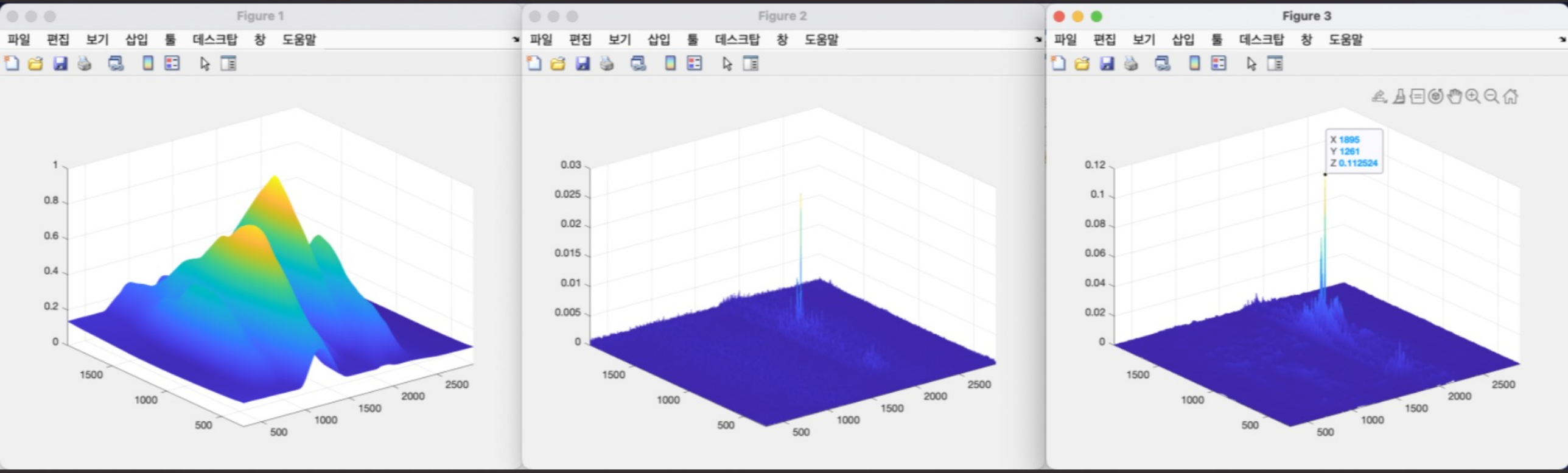
(3character_fixed_760mm + second_target)



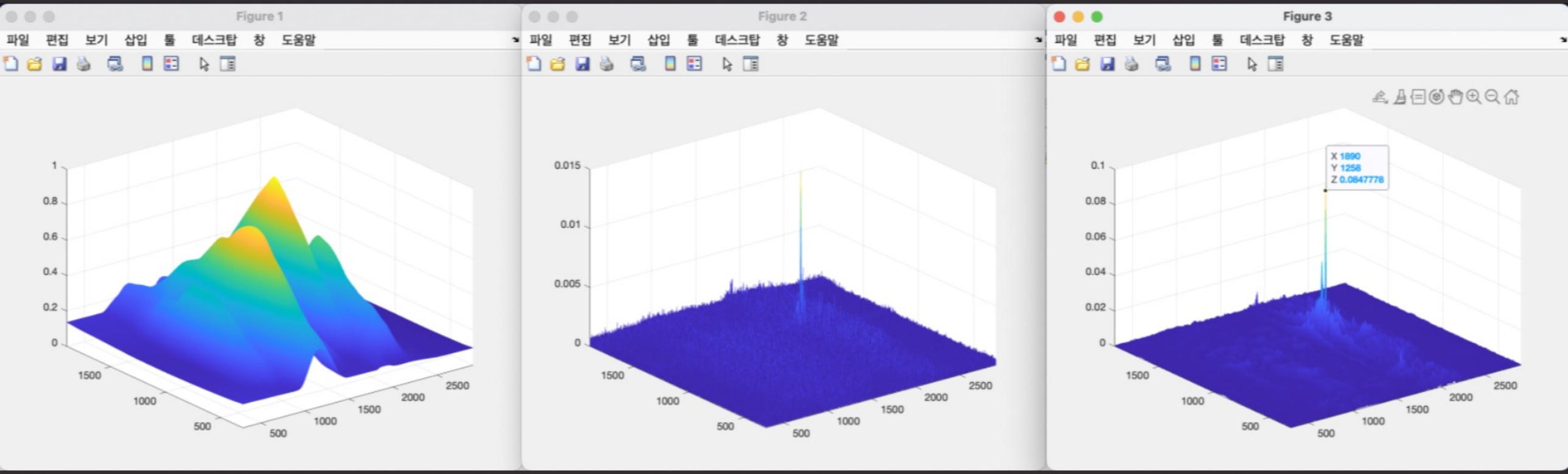
(3character_fixed_765mm + second_target)



(3character_fixed_770mm + second_target)



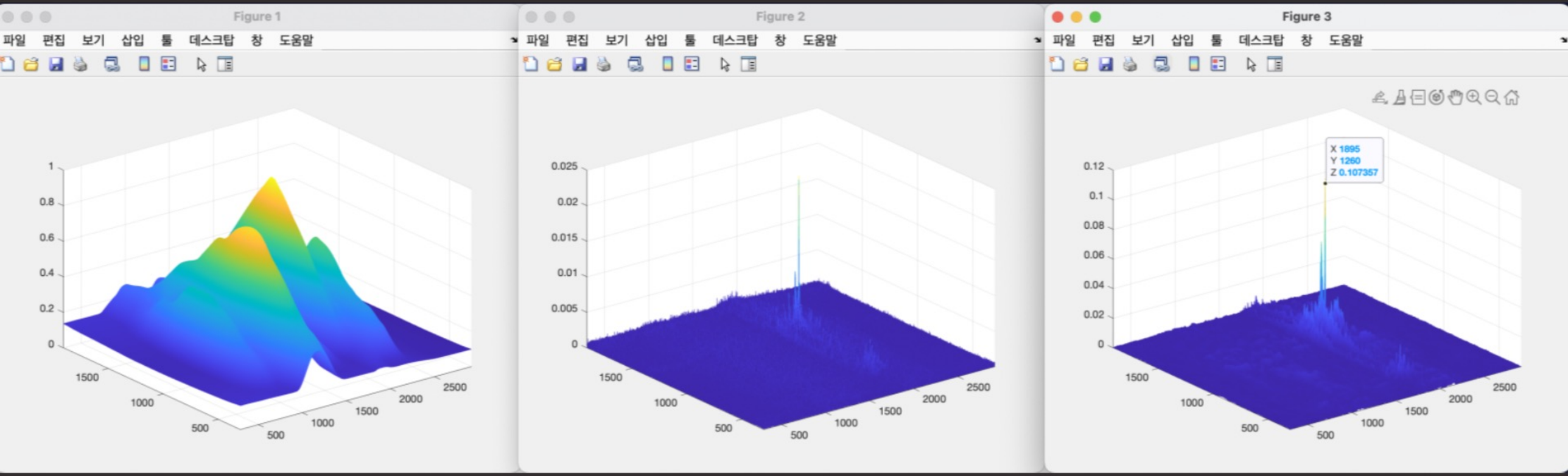
(3character_fixed_775mm + second_target)



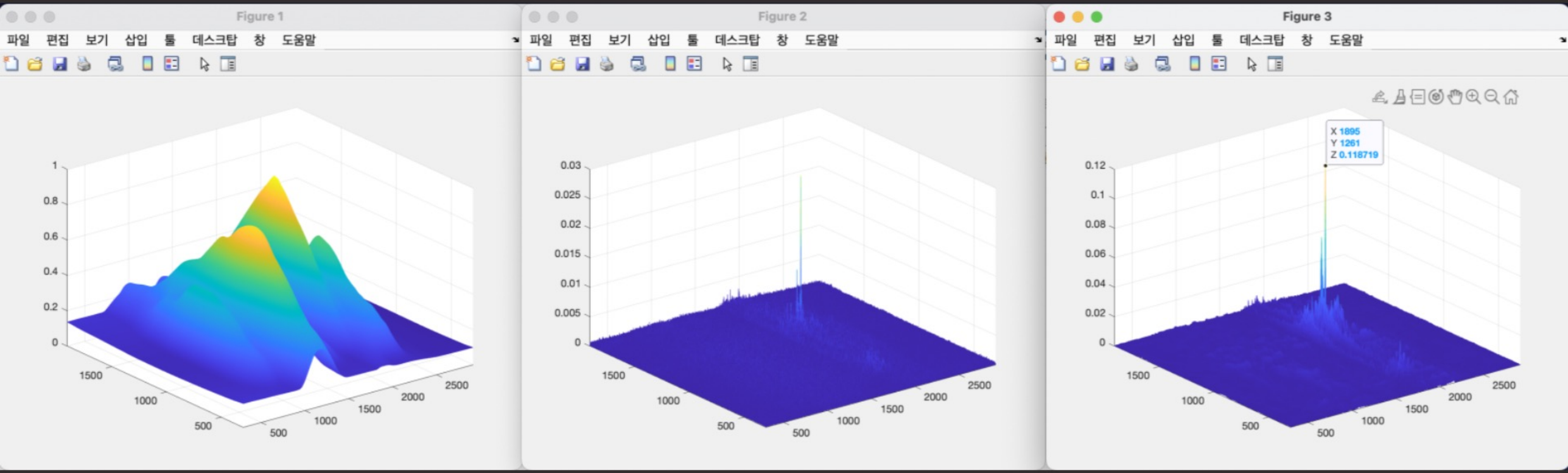
(3character_fixed_780mm + second_target)

3character_fixed + second_target
: depth = 750mm ~ 775mm

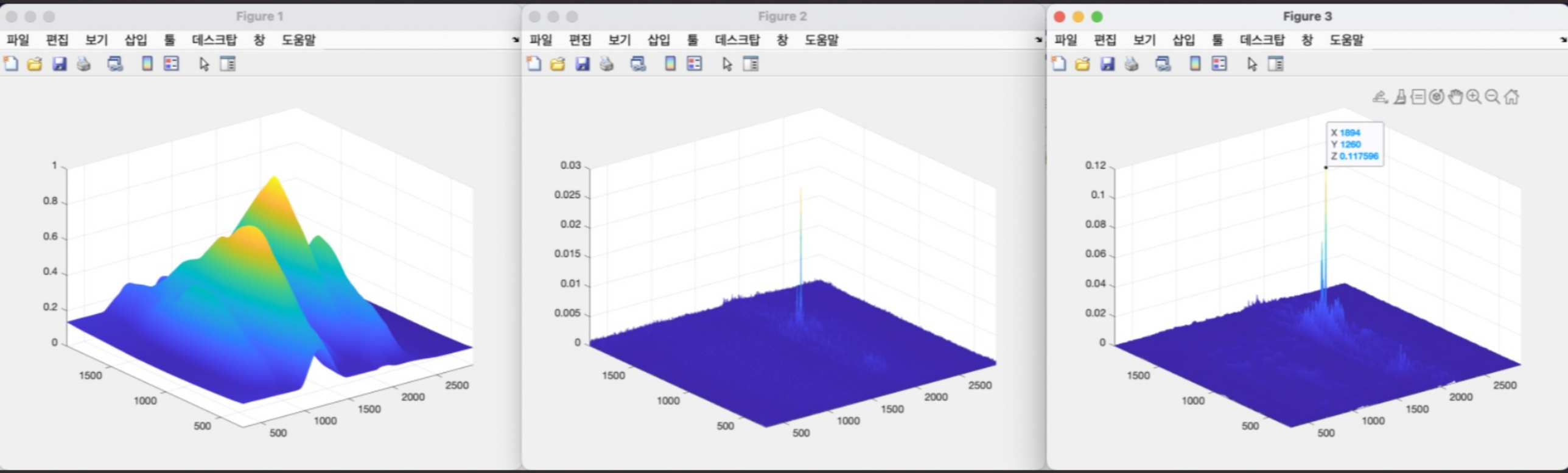
3character_nonuniform + second_target



(3character_nonuniform_760mm + second_target)



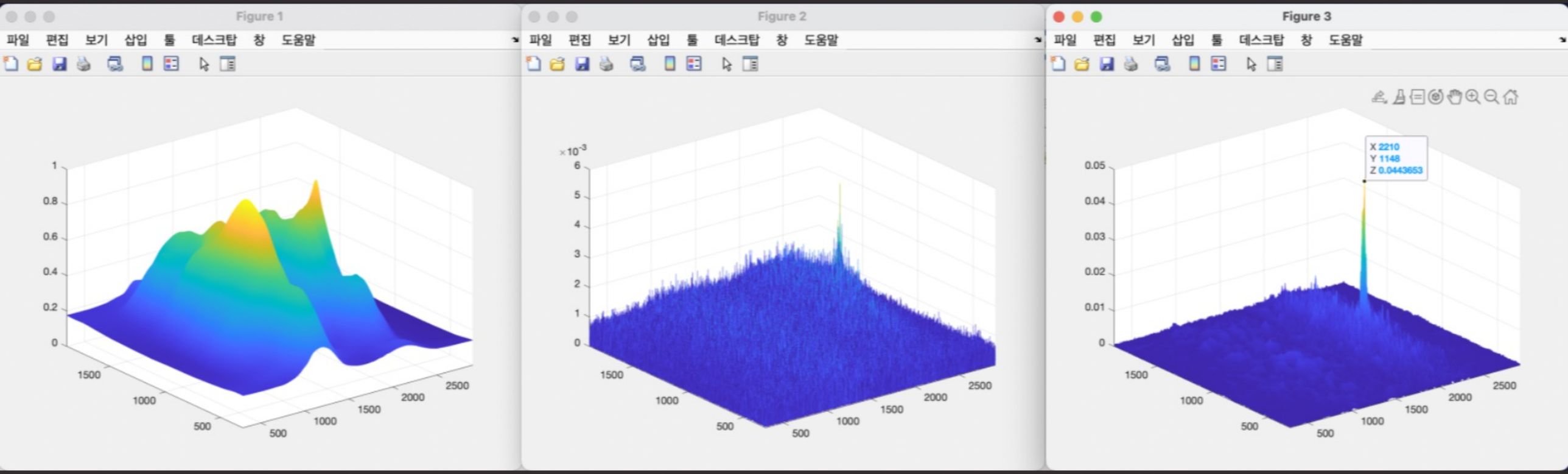
(3character_nonuniform_765mm + second_target)



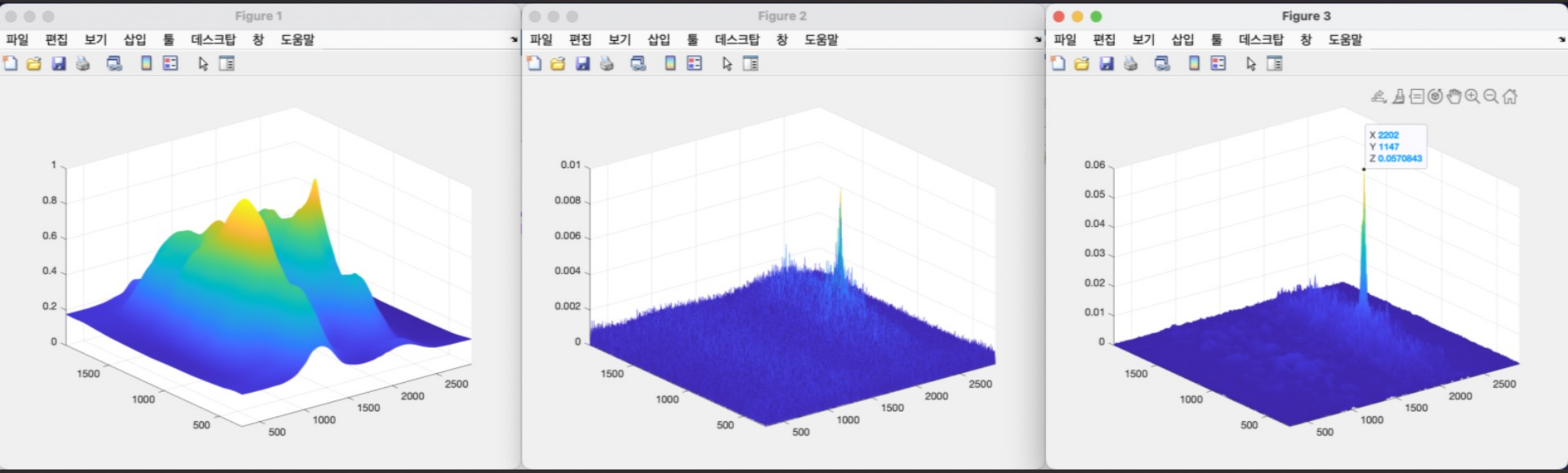
(3character_nonuniform_770mm + second_target)

3character_nonuniform + second_target
: depth = 765mm

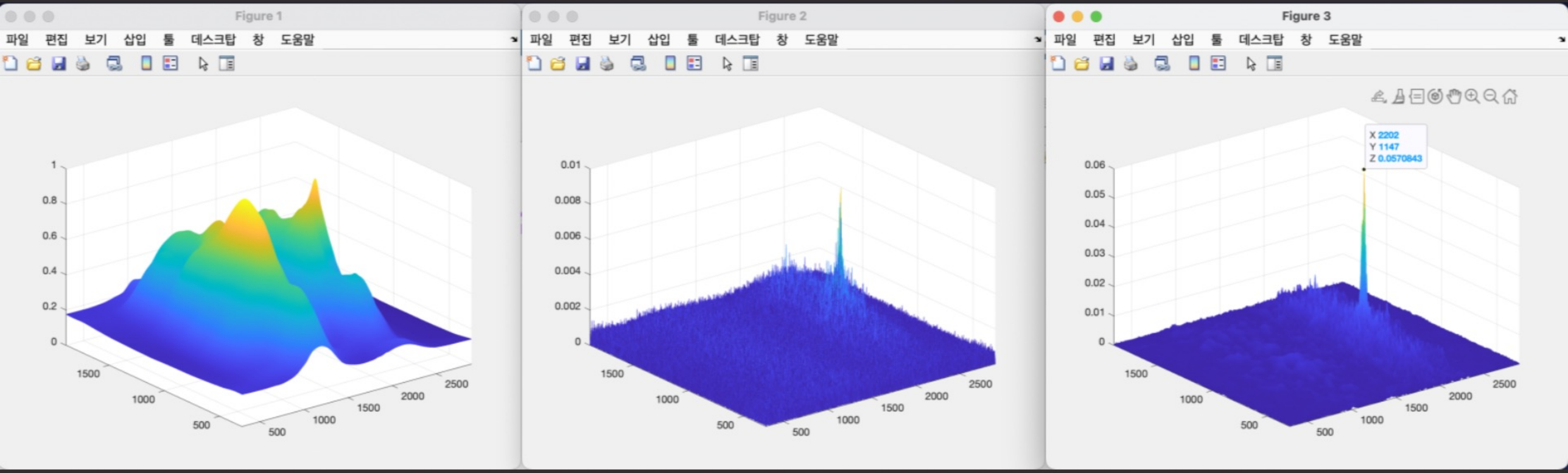
3character_fixed + third_target



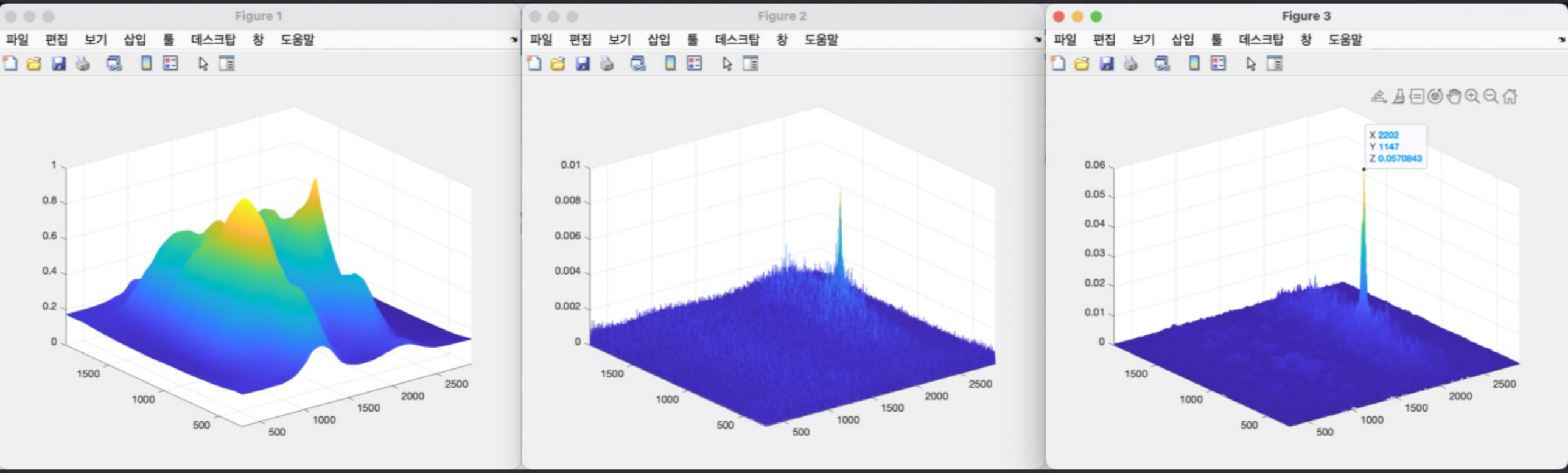
(3character_fixed_815mm + third_target)



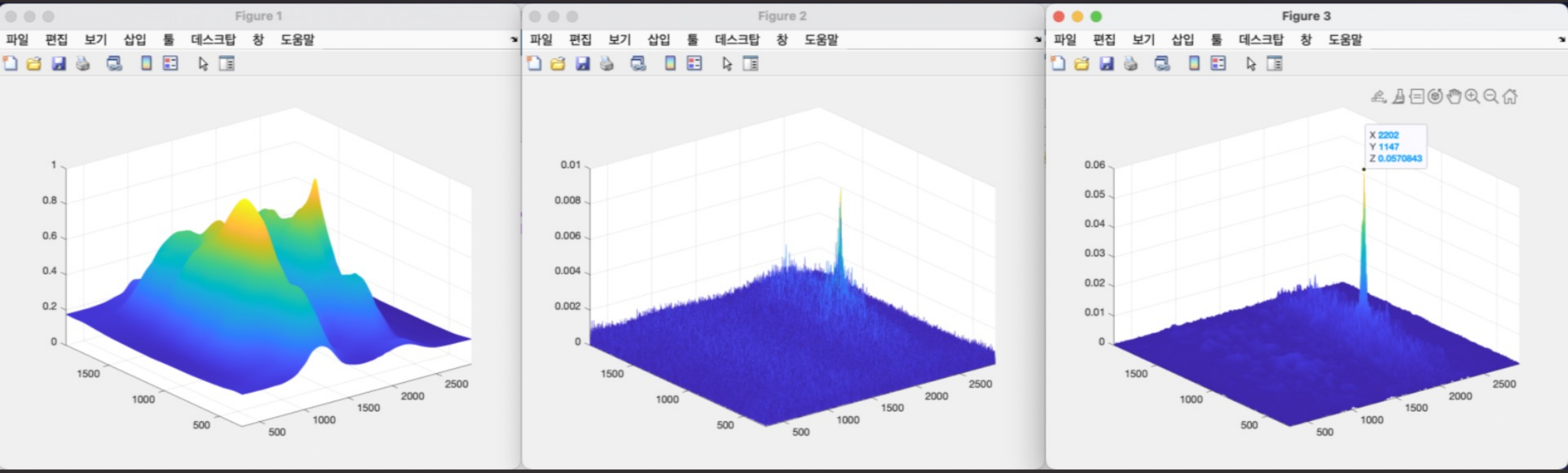
(3character_fixed_820mm + third_target)



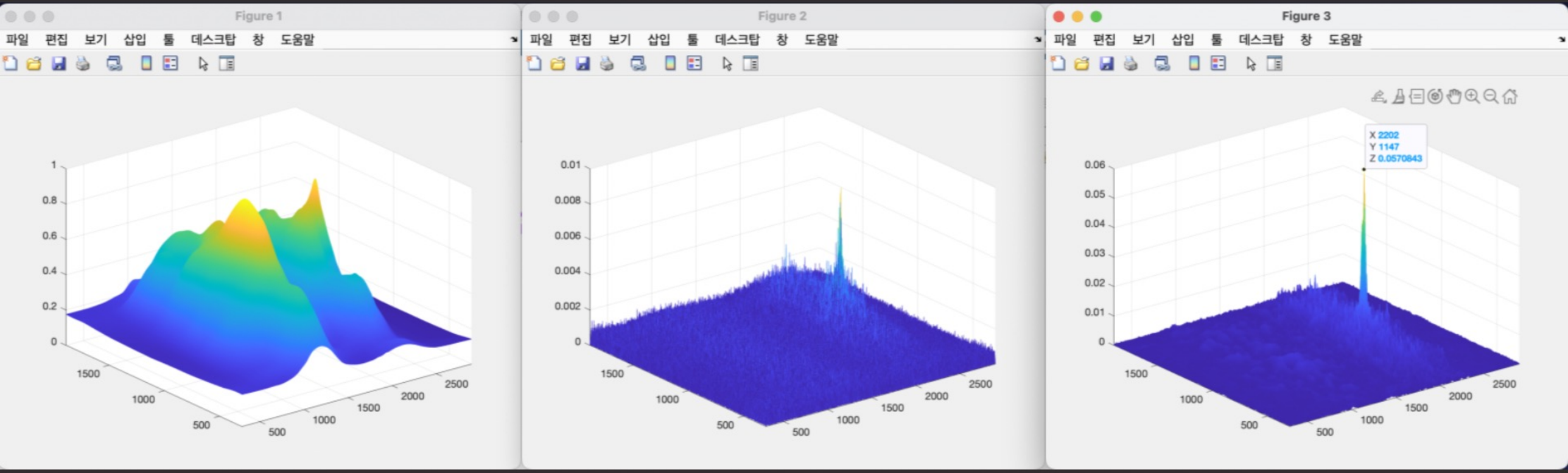
(3character_fixed_825mm + third_target)



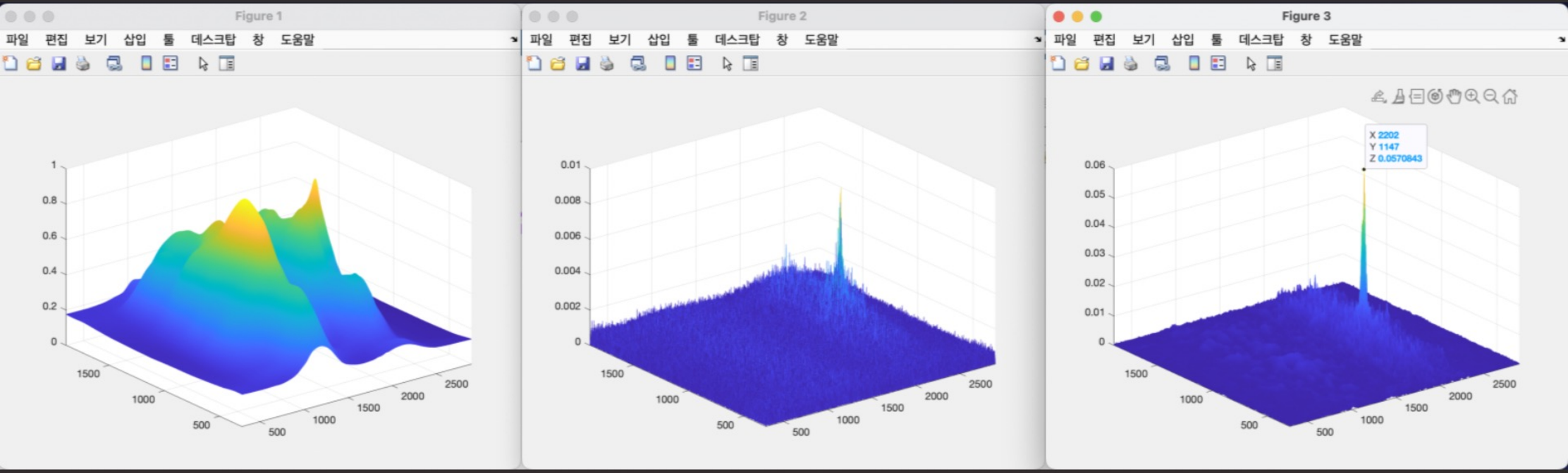
(3character_fixed_830mm + third_target)



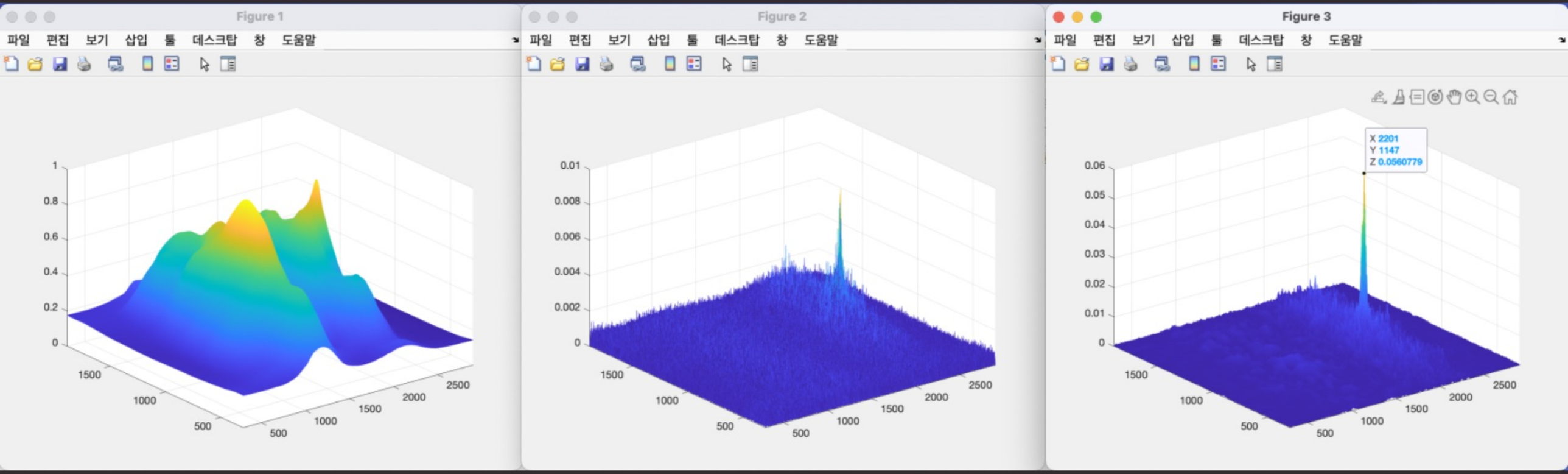
(3character_fixed_835mm + third_target)



(3character_fixed_840mm + third_target)



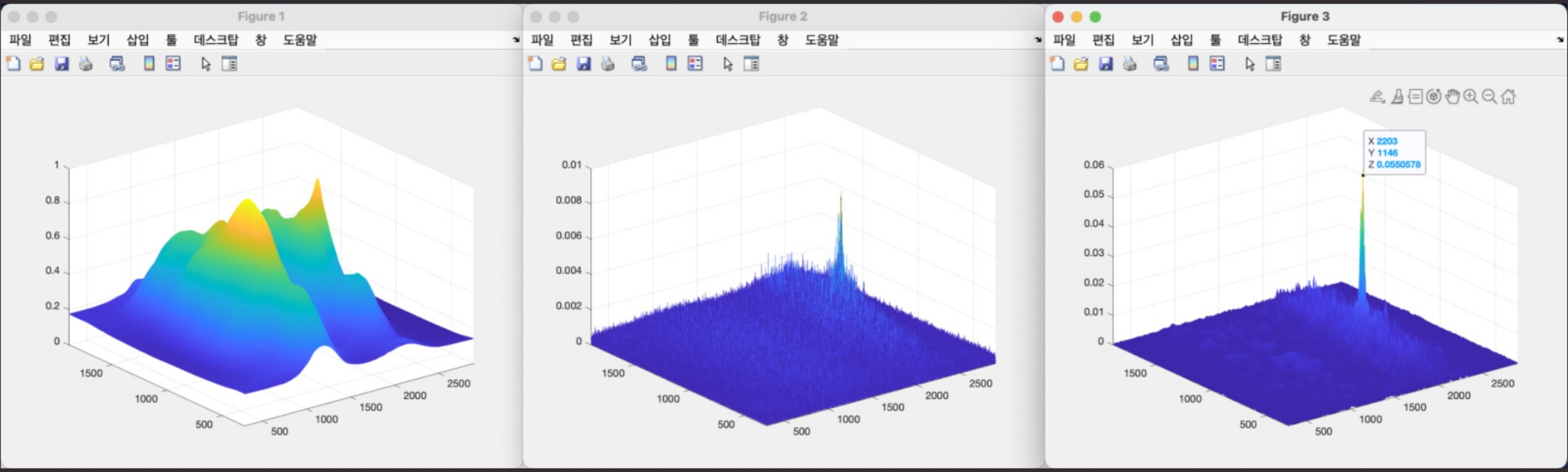
(3character_fixed_845mm + third_target)



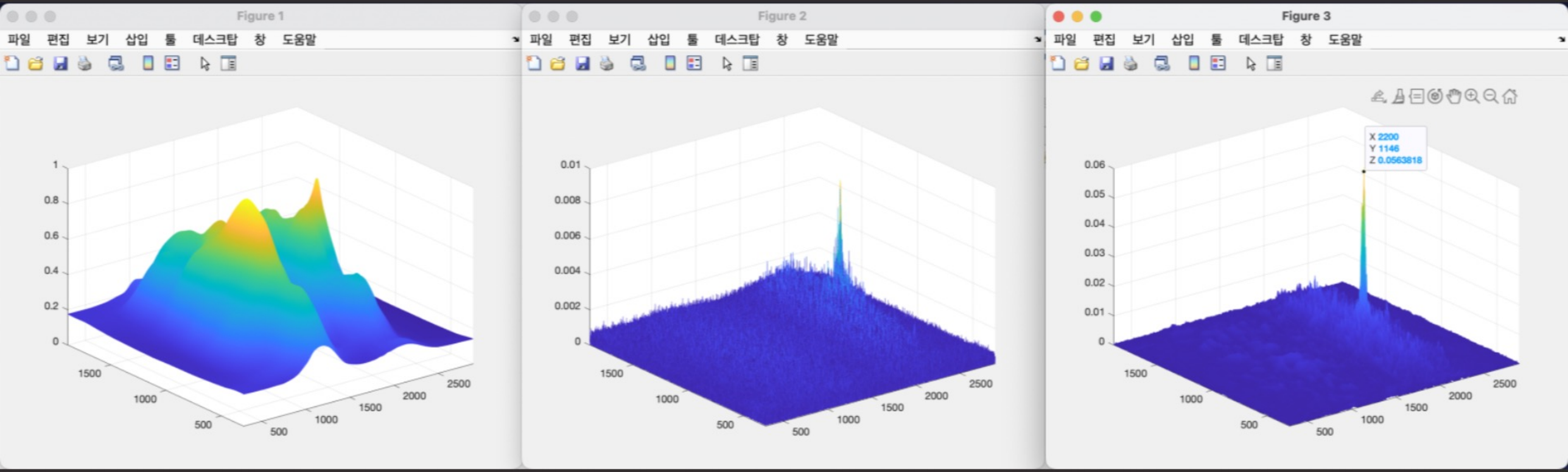
(3character_fixed_850mm + third_target)

3character_fixed + third_target
: depth = 820mm ~ 845mm

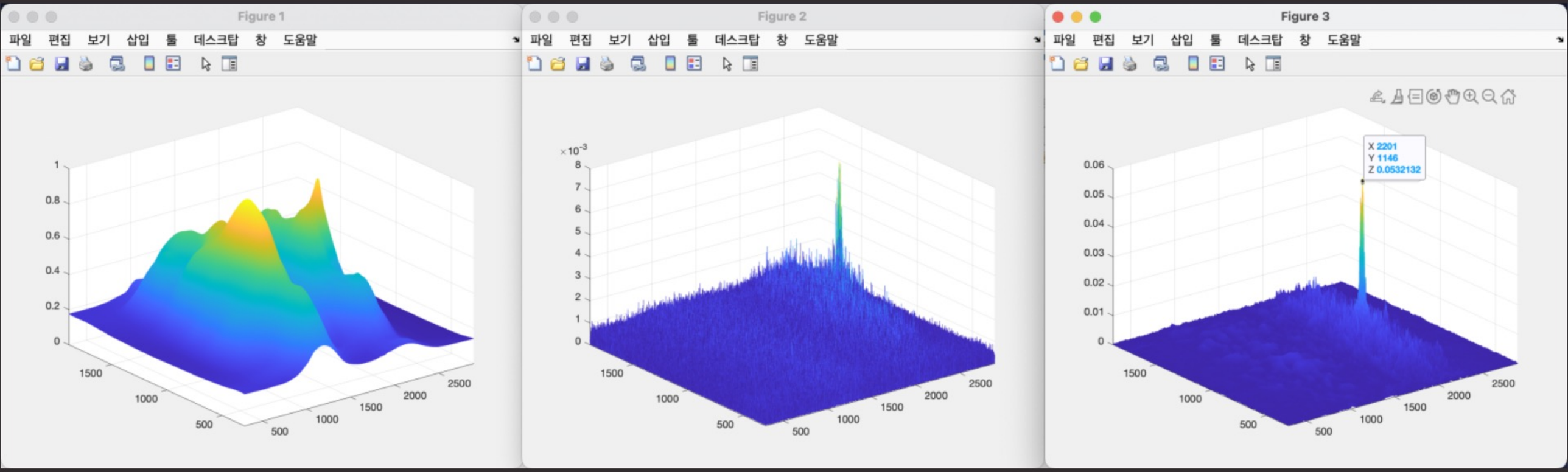
3character_nonuniform + third_target



(3character_nonuniform_840mm + third_target)



(3character_nonuniform_845mm + third_target)



(3character_nonuniform_850mm + third_target)

3character_nonuniform + third_target
: depth = 845mm

	first	second	third
fixed	665~685mm	750~775mm	820~845mm
nonuniform	660mm	765mm	845mm