Lab02 : Alpha blending, Masking 실습 보고서

정우성

4학년 201810890 융합전자공학과

**요약**

이 보고서에서는 이미지의 특정 부분을 추출하여 합성을 진행하는 실험을 진행하였다. 합성을 진행할 때 alpha blending기법과 masking기법, low-pass-filter를 사용하여 영상합성의 성능을 개선,비교,평가하였다. 실험이미지는 baboon.png,lena.png를 사용하였고 baboon 이미지의 오른쪽 눈 부분을 추출하여 lena 이미지의 오른쪽 눈 부분에 합성하였다. 실험결과 영상합성을 진행할때 alpha-blending의 alpha값을 높여 baboon이미지의 가중치를 높일 수록 원본 lena이미지와 달라지게 되어 PSNR값이 낮아지는 것을 볼 수 있었다. 또한 masking기법을 사용할 때는 크게 2가지 방법을 이용하였다. 이진 마스크를 사용하는 방법과 가우시안 마스크를 사용하는 방법이다. 이진 마스크를 사용한 결과 PSNR값이 39.20db가 나왔고 가우시안 마스크를 사용한 결과 40.17db가 나왔다. 결과적으로 가우시안 마스킹 기법이 원본과 가장 유사하게 합성을 했다는 것을 알 수 있었다. 그 후 low-pass-filter를 사용하여 고주파부분을 제거 후 합성을 진행하였다. Low-pass-filter로는 mean filter를 사용하였다. 그 결과 PSNR 값이 45.94db가 나왔다. 결과적으로 가우시안 마스킹기법과 low-pass-filter를 사용한 방법이 원본과 가장 유사하게 합성이 된 것을 알 수 있었다.

Instruction - 이 실험에서는 이미지의 특정 부분을 추출하여 합성을 진행하였다. 실험이미지는 baboon.png,lena.png를 사용하였고 baboon 이미지의 오른쪽 눈 부분을 추출하여 lena 이미지의 오른쪽 눈 부분에 합성하였다. 합성 방법으로는 alpha blending기법과 masking기법, low-pass-filter를 사용하였다. 그 후 각각의 방법에 대해 성능을 개선, 비교, 평가하였다. 성능을 확인하는 방법으로는 합성을 진행한 결과 이미지를 출력하여 시각적으로 비교해보고 PSNR값으로도 비교하여 각각의 합성방법을 비교 평가하였다.

The proposed method - 이미지의 특정 부분을 추출하여 합성을 진행할 때 크게 4가지 방법으로 진행하였다. 첫 번째 방법은 두 이미지의 특정부분을 추출하여 합성을 진행하는데 alpha blending기법을 사용하여 가중치를 조절해서 합성하는 방법이고 두 번째 방법은 두 이미지의 특정부분을 추출하고 이진 마스크를 만들어 이미지에 마스킹을 하고 합성하는 방법이고 세 번째 방법은 두 번쨰 방법과 절차는 동일하지만 마스크를 가우시안 마스크를 사용하여 합성을 진행하였고 마지막 방법으로는 세 번째 절차와 동일하게 진행하고 마스크를 씌운 이미지에 low-pass-filter를 적용하여 합성을 진행하였다.

4가지 방법의 합성 성능차이를 확인하는 방법으로는 이미지의 화질을 평가하는 객관적인 지표인 PSNR을 통해 비교하였다.

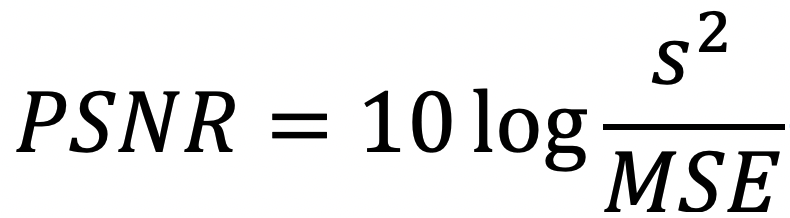


Figure 1 PSNR 계산 식

아래 이미지는 본 실험에서 사용한 이미지와 추출한 이미지이다. 각각의 이미지에서 오른쪽 눈을 추출하여 이미지 합성에 사용하였다.



Figure 2 lena.png,baboon.png

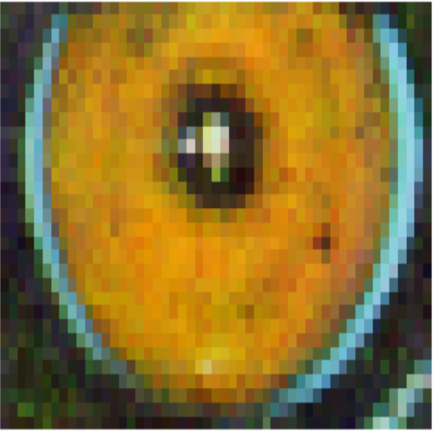
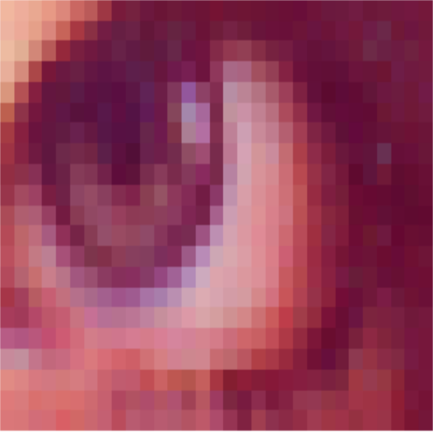


Figure 3 추출한 이미지(lena의 오른쪽 눈, baboon의 오른쪽 눈)

1. Alpha\_blending
2. 소스코드

clear;

% 이미지 불러오기

imgA = imread("baboon.png");

imgB = imread("lena.png");

% 원숭이의 오른쪽 눈 찾기

block1 = imgA(45:75, 310:360, :);

% 레나의 오른쪽 눈 찾기

block2 = imgB(260:280, 320:350, :);

% 원숭이의 오른쪽 눈 크기를 레나의 오른쪽 눈 크기와 맞추기

block1\_resized = imresize(block1, [size(block2,1) size(block2,2)]);

%이미지형 맞추기

imgA = double(imgA);

imgB = double(imgB);

imgR = imgB;

block1\_resized = double(block1\_resized);

block2 = double(block2);

%alpha

alp = 0:0.1:1;

%알파 블랜딩 알파값 설정

for i =1:length(alp)

ap = alp(i);

imgR(260:280, 320:350, :) = ap\*block1\_resized + (1-ap)\*block2;

%PSNR값

pval = psnr(imgR,imgB,255);

txt = sprintf('PSNR = %4.2fdB',pval);

disp(txt);

end

1. 결과

위 소스코드를 실행한 결과 다음과 같은 합성 이미지를 획득하였다. 가중치 값은 0부터 1까지 0.1간격으로 올리면서 설정하였으며 PSNR값을 보고 비교하였다. 아래 이미지는 가중치를 0.9를 설정하여 alpha blending합성 이미지이다.

Figure 4 Alpha blending 가중치 0.9일때 result이미지

위 이미지에서 가중치 값을 변경해가며 PSNR값을 확인한 결과는 아래 표와같다.

|  |  |
| --- | --- |
| Alpha | PSNR |
| 0.1 | 55.91dB |
| 0.2 | 49.89dB |
| 0.3 | 46.37dB |
| 0.4 | 43.87dB |
| 0.5 | 41.93dB |
| 0.6 | 40.34dB |
| 0.7 | 39.01dB |
| 0.8 | 37.85dB |
| 0.9 | 36.82dB |
| 1.0 | 35.91dB |

Figure 5 Alpha blending PSNR 결과 표

실험 결과 Alpha값을 높이면 baboon이미지에서 추출한 부분에 가중치가 높아지게 되며 원본이미지와 달라지게 되고 PSNR값이 떨어지는 것을 볼 수 있다.

1. Binary Masking
2. 소스코드

clear;

% 이미지 불러오기

imgA = imread("baboon.png");

imgB = imread("lena.png");

% 원숭이의 오른쪽 눈 찾기

block1 = imgA(45:75, 310:360, :);

% 레나의 오른쪽 눈 찾기

block2 = imgB(260:280, 320:350, :);

% 원숭이의 오른쪽 눈 크기를 레나의 오른쪽 눈 크기와 맞추기

block1\_resized = imresize(block1, [size(block2,1) size(block2,2)]);

% 원숭이 이미지에 사용할 이진마스크 생성

[row1,col1,dep1] = size(block1\_resized);

msksize1 = [row1,col1];

rad1 = min(msksize1)/1.5;

imMsk1 = CircleMsk1(msksize1, rad1);

% 레나 이미지에 사용할 이진마스크 생성

[row2,col2,dep2] = size(block2);

msksize2 = [row2,col2];

rad2 = min(msksize2)/1.5;

imMsk2 = CircleMsk2(msksize2, rad2);

%이미지형 맞추기

imgA = double(imgA);

imgB = double(imgB);

imgR = imgB;

block1\_resized = double(block1\_resized);

block2 = double(block2);

imMsk1 = double(imMsk1);

%마스킹

block1\_masked = block1\_resized.\*imMsk1;

block2\_masked = block2.\*(1-imMsk2);

%합성

imgR(260:280, 320:350, :) = block1\_masked + block2\_masked ;

%PSNR값

pval = psnr(imgR,imgB,255);

txt = sprintf('PSNR = %4.2fdB',pval);

disp(txt);

function imMsk1 = CircleMsk1(msksize1, rad1)

%중앙값,가로,세로 설정

rows1 = msksize1(2);

cols1 = msksize1(1);

center1 = msksize1/2;

%벡터생성

[x1,y1] = meshgrid(1:rows1,1:cols1);

%거리계산

dist1 = sqrt( ( x1-center1(2) ).^2 + ( y1-center1(1) ).^2 );

%이진 마스크 생성

imMsk1 = dist1 <= rad1;

end

function imMsk2 = CircleMsk2(msksize2, rad2)

%중앙값,가로,세로 설정

rows2 = msksize2(2);

cols2 = msksize2(1);

center2 = msksize2/2;

%벡터생성

[x2,y2] = meshgrid(1:rows2,1:cols2);

%거리계산

dist2 = sqrt( ( x2-center2(2) ).^2 + ( y2-center2(1) ).^2 );

%이진 마스크 생성

imMsk2 = dist2 <= rad2;

end

1. 결과

위의 소스코드를 실행한 결과 아래와 같은 합성이미지를 획득하였다. CircleMsk함수를 사용하여 이진 마스크를 만든 후 각각의 이미지에서 추출한 블럭 이미지에 마스킹을 진행한 후 합성을 하였다.

의류, 인간의 얼굴, 패션 액세서리, 머리장식이(가) 표시된 사진

자동 생성된 설명 의류, 인간의 얼굴, 패션 액세서리, 머리장식이(가) 표시된 사진

자동 생성된 설명

Figure 6 Binary masking한 합성이미지

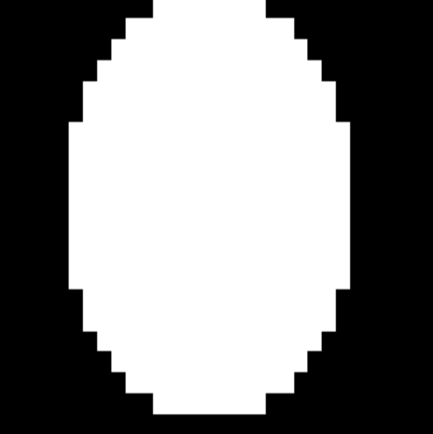


Figure 7 이진 마스크 이미지

다채로움, 스크린샷, 노랑, 픽셀이(가) 표시된 사진

자동 생성된 설명다채로움, 그래픽, 스크린샷, 픽셀이(가) 표시된 사진

자동 생성된 설명

Figure 8 이진 마스킹으로 추출한 baboon이미지와 lena이미지

위와 같은 과정으로 실험을 진행한 결과 PSNR 값은 39.20db가 나왔다.

|  |  |
| --- | --- |
| Masking | PSNR |
| Binary | 39.20db |

Figure 9 Binary Masking PSNR 결과 표

1. Gaussian Masking
2. 소스코드

clear;

% 이미지 불러오기

imgA = imread("baboon.png");

imgB = imread("lena.png");

% 원숭이의 오른쪽 눈 찾기

block1 = imgA(45:75, 310:360, :);

% 레나의 오른쪽 눈 찾기

block2 = imgB(260:280, 320:350, :);

% 원숭이의 오른쪽 눈 크기를 레나의 오른쪽 눈 크기와 맞추기

block1\_resized = imresize(block1, [size(block2,1) size(block2,2)]);

% 바분 시그마는 높게 설정하여 가우시안 마스크 생성

[row1,col1,dep1] = size(block1\_resized);

msksize1 = [row1,col1];

sigma1 = 50;

imMsk1 = GaussMsk1(msksize1, sigma1);

% 레나 시그마는 낮게 설정하여 가우시안 마스크 생성

[row2,col2,dep2] = size(block2);

msksize2 = [row2,col2];

sigma2 = 10;

imMsk2 = GaussMsk2(msksize2, sigma2);

% 이미지형 맞추기

imgA = double(imgA);

imgB = double(imgB);

imgR = imgB;

block1\_resized = double(block1\_resized);

block2 = double(block2);

imMsk1 = double(imMsk1);

imMsk2 = double(imMsk2);

% 마스킹

block1\_masked = block1\_resized.\*imMsk1;

block2\_masked = block2.\*(1-imMsk2);

% 합성

imgR(260:280, 320:350, :) = block1\_masked + block2\_masked ;

% PSNR값 계산

pval = psnr(imgR,imgB,255);

txt = sprintf('PSNR = %4.2fdB',pval);

disp(txt);

% 원숭이 이미지에 적용할 가우시안 마스크 함수

function imMsk1 = GaussMsk1(msksize1, sigma1)

% 중앙값,가로,세로 설정

rows1 = msksize1(2);

cols1 = msksize1(1);

center1 = msksize1/2;

% 벡터생성

[x1,y1] = meshgrid(1:rows1,1:cols1);

% 가우시안 계산

% exp(- (x^2 + y^2)/2\*sigma)

dist1 = exp( -(( x1-center1(2) ).^2 + ( y1-center1(1) ).^2)/(2\*sigma1) );

% 가우시안 마스크 생성

imMsk1 = dist1/max(dist1(:));

end

% 레나 이미지에 적용할 가우시안 마스크 함수

function imMsk2 = GaussMsk2(msksize2, sigma2)

% 중앙값,가로,세로 설정

rows2 = msksize2(2);

cols2 = msksize2(1);

center2 = msksize2/2;

% 벡터생성

[x2,y2] = meshgrid(1:rows2,1:cols2);

% 가우시안 계산

% exp(- (x^2 + y^2)/2\*sigma)

dist2 = exp( -(( x2-center2(2) ).^2 + ( y2-center2(1) ).^2)/(2\*sigma2) );

% 가우시안 마스크 생성

imMsk2 = dist2/max(dist2(:));

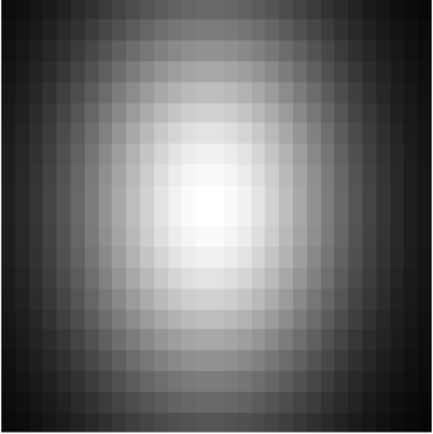
end

1. 결과

위의 소스코드를 실행한 결과 아래와 같은 합성이미지를 획득하였다. GaussMsk함수를 사용하여 가우시안 마스크를 만든 후 각각의 이미지에서 추출한 블럭 이미지에 마스킹을 진행했다. Baboon이미지에는 시그마 값을 50으로 하였고 lena이미지에는 시그마 값을 10으로 하였다. 그 후 합성을 진행했다.

Figure 10 Gaussian masking한 합성이미지

스크린샷, 다채로움, 사각형, 노랑이(가) 표시된 사진

자동 생성된 설명

Figure 11 sigma=50 가우시안 마스킹으로 추출한 baboon이미지

블랙, 스크린샷, 어둠, 흑백이(가) 표시된 사진

자동 생성된 설명다채로움, 마젠타, 보라색, 라일락이(가) 표시된 사진

자동 생성된 설명

Figure 12 sigma=10 가우시안 마스킹으로 추출한 lena이미지

위와 같은 과정으로 실험을 진행한 결과 PSNR 값은 40.17db가 나왔다.

|  |  |
| --- | --- |
| Masking | PSNR |
| Gaussian | 40.17db |

Figure 13 Gaussian Masking PSNR 결과 표

1. LOW-PASS-FILTER
2. 소스코드

clear;

% 이미지 불러오기

imgA = imread("baboon.png");

imgB = imread("lena.png");

% 원숭이의 오른쪽 눈 찾기

block1 = imgA(45:75, 310:360, :);

% 레나의 오른쪽 눈 찾기

block2 = imgB(260:280, 320:350, :);

% 원숭이의 오른쪽 눈 크기를 레나의 오른쪽 눈 크기와 맞추기

block1\_resized = imresize(block1, [size(block2,1) size(block2,2)]);

% 바분 시그마는 높게 설정하여 가우시안 마스크 생성

[row1,col1,dep1] = size(block1\_resized);

msksize1 = [row1,col1];

sigma1 = 50;

imMsk1 = GaussMsk1(msksize1, sigma1);

% 레나 시그마는 낮게 설정하여 가우시안 마스크 생성

[row2,col2,dep2] = size(block2);

msksize2 = [row2,col2];

sigma2 = 10;

imMsk2 = GaussMsk2(msksize2, sigma2);

% 이미지형 맞추기

imgA = double(imgA);

imgB = double(imgB);

imgR = imgB;

block1\_resized = double(block1\_resized);

block2 = double(block2);

imMsk1 = double(imMsk1);

imMsk2 = double(imMsk2);

% 마스킹

block1\_masked = block1\_resized.\*imMsk1;

block2\_masked = block2.\*(1-imMsk2);

% 로우패스필터적용 – meanfilter

filter = ones(3)/size(block2,1);

block1\_filtered = zeros(size(block1\_masked));

for i=1:3

block1\_filtered(:,:,i) = conv2(block1\_masked(:,:,i), filter, 'same');

end

% 합성

imgR(260:280, 320:350, :) = block1\_filtered + block2\_masked ;

% PSNR값

pval = psnr(imgR,imgB,255);

txt = sprintf('PSNR = %4.2fdB',pval);

disp(txt);

% 원숭이 이미지에 적용할 가우시안 마스크 함수

function imMsk1 = GaussMsk1(msksize1, sigma1)

% 중앙값,가로,세로 설정

rows1 = msksize1(2);

cols1 = msksize1(1);

center1 = msksize1/2;

% 벡터생성

[x1,y1] = meshgrid(1:rows1,1:cols1);

% 가우시안 계산

% exp(- (x^2 + y^2)/2\*sigma)

dist1 = exp( -(( x1-center1(2) ).^2 + ( y1-center1(1) ).^2)/(2\*sigma1) );

% 가우시안 마스크 생성

imMsk1 = dist1/max(dist1(:));

end

% 레나 이미지에 적용할 가우시안 마스크 함수

function imMsk2 = GaussMsk2(msksize2, sigma2)

% 중앙값,가로,세로 설정

rows2 = msksize2(2);

cols2 = msksize2(1);

center2 = msksize2/2;

% 벡터생성

[x2,y2] = meshgrid(1:rows2,1:cols2);

% 가우시안 계산

% exp(- (x^2 + y^2)/2\*sigma)

dist2 = exp( -(( x2-center2(2) ).^2 + ( y2-center2(1) ).^2)/(2\*sigma2) );

% 가우시안 마스크 생성

imMsk2 = dist2/max(dist2(:));

end

1. 결과

위의 소스코드를 실행한 결과 아래와 같은 합성이미지를 획득하였다. GaussMsk함수를 사용하여 가우시안 마스크를 만든 후 각각의 이미지에서 추출한 블럭 이미지에 마스킹을 진행했다. Baboon이미지에는 시그마 값을 50으로 하였고 lena이미지에는 시그마 값을 10으로 하였다. 또한 추출하여 가우시안 마스킹을 한 baboon이미지에는 low pass filter를 적용했고 그 후 합성을 진행했다.

Figure low-pass-filter 적용 마스킹한 합성 이미지

스크린샷, 블랙, 사각형, 흑백이(가) 표시된 사진

자동 생성된 설명스크린샷, 다채로움, 사각형, 노랑이(가) 표시된 사진

자동 생성된 설명

Figure sigma=50 가우시안 마스킹으로 추출한 baboon이미지

블랙, 스크린샷, 어둠, 흑백이(가) 표시된 사진

자동 생성된 설명다채로움, 마젠타, 보라색, 라일락이(가) 표시된 사진

자동 생성된 설명

Figure sigma=10 가우시안 마스킹으로 추출한 lena이미지

스크린샷, 다채로움, 사각형, 노랑이(가) 표시된 사진

자동 생성된 설명스크린샷, 사각형, 직사각형, 노랑이(가) 표시된 사진

자동 생성된 설명

Figure 마스킹하여 추출하고 low-pass-filter 적용한 baboon이미지

위와 같은 과정으로 실험을 진행한 결과 PSNR 값은 45.94db가 나왔다.

|  |  |
| --- | --- |
| Low-pass-filter | PSNR |
| Mean filter | 45.94db |

Figure low-pass-filter Masking PSNR 결과 표

Simulation result – 위와 같은 실험과정으로 실험을 진행한 결과들을 종합하고 비교해보면 다음과 같다.

1. Result Image – 시각적 평가

Figure Alpha blending 가중치 0.9일때 result이미지

의류, 인간의 얼굴, 패션 액세서리, 머리장식이(가) 표시된 사진

자동 생성된 설명 의류, 인간의 얼굴, 패션 액세서리, 머리장식이(가) 표시된 사진

자동 생성된 설명

Figure Binary masking한 합성이미지

Figure Gaussian masking한 합성이미지

Figure low-pass-filter 적용 마스킹한 합성 이미지

실험결과의 이미지를 보고 비교를 해보면 단계를 거듭할 수록 점점 자연스럽게 lena이미지에 baboon이미지의 눈이 합성되는 것을 볼 수 있다.

1. Result PSNR – 객관적 평가

|  |  |
| --- | --- |
| Alpha | PSNR |
| 0.1 | 55.91dB |
| 0.2 | 49.89dB |
| 0.3 | 46.37dB |
| 0.4 | 43.87dB |
| 0.5 | 41.93dB |
| 0.6 | 40.34dB |
| 0.7 | 39.01dB |
| 0.8 | 37.85dB |
| 0.9 | 36.82dB |
| 1.0 | 35.91dB |

Figure Alpha blending PSNR 결과

|  |  |
| --- | --- |
| Masking | PSNR |
| Binary | 39.20db |

Figure Binary Masking PSNR 결과

|  |  |
| --- | --- |
| Masking | PSNR |
| Gaussian | 40.17db |

Figure Gaussian Masking PSNR 결과

|  |  |
| --- | --- |
| Low-pass-filter | PSNR |
| Mean filter | 45.94db |

Figure 26 low-pass-filter Masking PSNR 결과

실험결과의PSNR값을 보고 비교를 해보면 일단 alpha blending의 경우 alpha값을 높여 baboon이미지의 가중치를 점차 높이는 경우PSNR값이 점차 낮아지는 것을 볼 수 있는데 이는 원본 이미지와 점차 차이가 나게 합성이 되는 것으로 볼 수 있다. 다음 Masking의 경우 이진 마스크를 사용하여 합성을 진행한 결과보다 가우시안 마스크를 사용하여 합성을 진행했을때가 더 PSNR값이 높게 나타나는것을 알 수 있었다. 마지막으로 가우시안 마스크를 사용하고 low-pass filter를 적용하여 합성을 진행한 결과 PSNR이 더욱 높게 나왔다. 이 결과 masking을 진행하고 filter까지 적용하면 영상 합성을 진행할 떄 시각적으로나. 수치상으로나 더욱 자연스럽게 합성을 할 수 있다는 것을 알 수 있었다.

Conclusions – 결과적으로 영상합성의 결과 이미지를 보고 각 합성 방법을 비교를 해보면 단계를 거듭할 수록 점점 자연스럽게 lena이미지에 baboon이미지의 눈이 합성되는 것을 볼 수 있었다. 실험결과의PSNR값을 보고 비교를 해보면 alpha blending의 경우 alpha값을 높여 baboon이미지의 가중치를 점차 높이는 경우PSNR값이 점차 낮아지는 것을 볼 수 있었다. 이는 원본 이미지와 점차 차이가 나게 합성이 되는 것으로 볼 수 있었다. 다음 Masking의 경우 이진 마스크를 사용하여 합성을 진행한 결과39.20db보다 가우시안 마스크를 사용하여 합성을 진행했을때가 40.17db로 PSNR값이 높게 나타나는 것을 알 수 있었다. 마지막으로 가우시안 마스크를 사용하고 low-pass filter를 적용하여 합성을 진행한 결과 PSNR이 45.94db로더욱 높게 나온것을 알 수 있었다. 이러한 결과를 종합해보면 영상합성을 진행할때 단순히 합성을 하기 보다는masking을 진행하여 원하는 부분만 추출하고 가우시안 마스크를 적용해서 경계값의 극단적인 면을 부드럽게 해주고 low-pass-filter까지 적용해서 고주파 영역을 제거한다면 영상 합성을 진행할 떄 시각적으로나. PSNR수치상으로나 더욱 자연스럽게 합성을 할 수 있다는 것을 알 수 있었다.