

Image Blending

17010826 김성민

01 Seam Finding

02 Multi-Band Blending

03 프로그램 순서도

04 NNDR에 따른 Matching 결과

Image Blending

목표 : Image Warping 단계에서 합친 두 이미지의 경계를 더 자연스럽게 하는 것



Image Blending

01 Seam Finding

Seam Finding



경계선에 의해 나뉘어가지가 불안정하게 표현되고
나무 뒤 쪽 건물도 부자연스러움

→ 카메라 각도에 따라 정확히 겹치지 않을 수 있음
따라서, 이를 보완하기 위한 해결책이 필요함

Seam Finding

두 겹쳐진 이미지에서, 이미지 내의 물체를 고려한 경계선을 찾기 위한 알고리즘
카메라 각도에 의해 어색해진 부분을 어느정도 해결해 줌



→
적용



Seam Finding을 이용해 찾은 경계선

Seam Finding – Dynamic Programming

Dynamic Programming을 이용한 Seam Finding

Seam Finding을 구현하는 여러 방법 중 Dynamic Programming을 사용한 방법을 알아보자.

이미지의 에너지 정의

가장 먼저 픽셀의 에너지를 정의해야 한다.

$$|\Delta x|^2 = (\Delta r_x)^2 + (\Delta g_x)^2 + (\Delta b_x)^2, |\Delta y|^2 = (\Delta r_y)^2 + (\Delta g_y)^2 + (\Delta b_y)^2, e(x, y) = |\Delta x|^2 + |\Delta y|^2$$

여기서 $\Delta r_x, \Delta g_x, \Delta b_x$ 는 픽셀의 왼쪽과 오른쪽 값의 빨강, 초록, 파랑 값의 차를 의미하고, $\Delta r_y, \Delta g_y, \Delta b_y$ 는 픽셀의 위쪽과 아래쪽 값의 빨강, 초록, 파랑 값의 차를 의미한다.

모서리 부분의 경우 픽셀의 왼쪽 혹은 다른 방향의 값이 없을 수 있는데, 그런 경우는 자기 자신의 값으로 대체한다.

에너지가 큰 부분은 픽셀 값의 변화 값이 큰 부분 → 고주파 성분이 나타남

에너지가 작은 부분은 픽셀 값의 변화 값이 큰 부분 → 저주파 성분이 나타남

에너지가 낮은 부분을 따라 경계선을 형성하면 이미지의 물체를 어느정도 피해가는 경계선을 만들 수 있을 것이다.



에너지



Seam Finding – Dynamic Programming

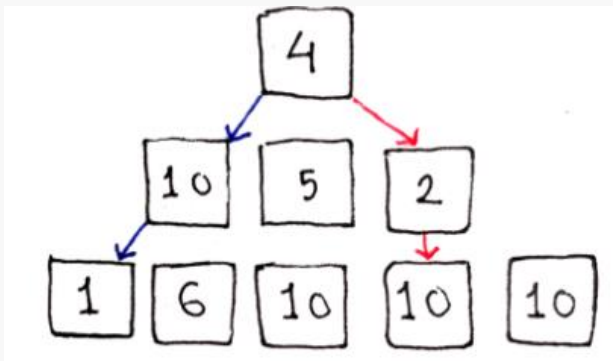
lowest-energy seam

각 픽셀의 에너지를 이용해 이미지를 가로지르는 lowest energy seam을 찾을 수 있다. (수직, 수평 모두 가능하다.)

여기서는 수직(위에서 아래로)을 기준으로 알아보자.

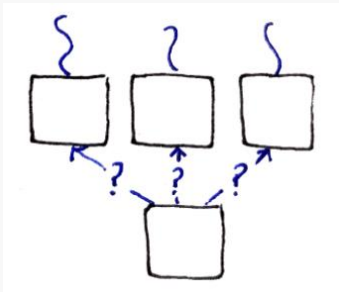
lowest energy seam을 아래와 같이 정의한다.

- seam은 한 행에 한 픽셀만 선택한다.
- lowest-energy seam은 seam의 모든 픽셀의 총 에너지가 최소화되는 seam이다.



만약 옆 그림과 같이 무작정 다음 픽셀을 에너지가 가장 작은 픽셀로 고른다면 최종적으로는 lowest energy seam을 찾을 수 없을 가능성이 크다.

이 방법은 하나의 seam을 계속하기 위해 여러 픽셀들 중 하나를 선택하는 것으로 해석할 수 있다.



반대로 하나의 픽셀에서 여러 seam 후보 중 가장 적절한 것을 선택하는 방식으로 바꿔서 해결해보자.

➡ Dynamic Programming 이용

Seam Finding – Dynamic Programming

lowest-energy seam을 찾기 위한 Dynamic Programming

모든 픽셀마다 자기 자신까지 seam이 도달했을 때, 가장 에너지가 작은 경우를 계산한다.

$M(x, y)$ 를 이미지의 top에서 (x, y) 까지의 lowest-energy seam이라고 정의한다.

먼저, top row에 있는 픽셀들의 $M(x, y)$ 은 다음과 같이 정의된다.

$$M(x, 0) = e(x, 0)$$

➡ 자신보다 위 픽셀이 없으므로 자기 자신까지의 lowest-energy seam은 자기 자신의 에너지가 된다.

다음으로 나머지 픽셀들의 $M(x, y)$ 은 다음과 같이 정의된다.

$$M(x, y) = e(x, y) + \min(M(x - 1, y - 1), M(x, y - 1), M(x + 1, y - 1))$$

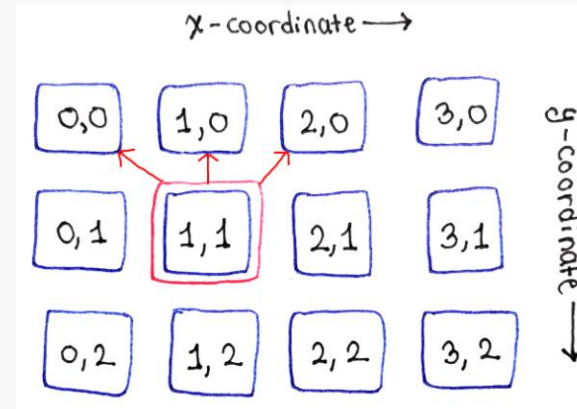
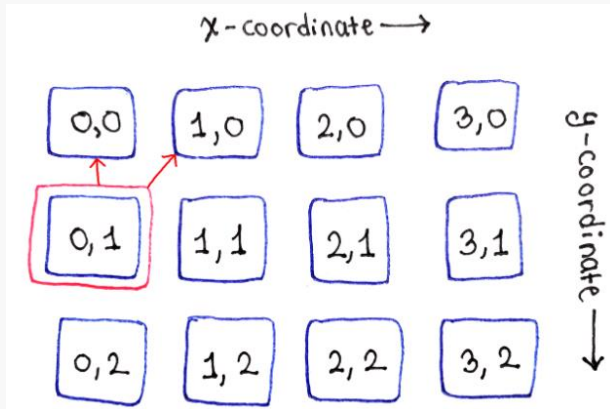
➡ 자기 자신의 에너지와 top-left, top, top-right 픽셀들 중 더 작은 seam 에너지를 더한다.

마지막으로 $M(x - 1, y - 1)$ 혹은 $M(x + 1, y - 1)$ 가 없는 경우는 해당 부분을 $\min()$ 에서 생략한다.

이렇게 마지막 행까지 계산하면 모든 픽셀에 대해 lowest-energy seam이 가지는 에너지의 크기를 알 수 있다.

Seam Finding – Dynamic Programming

반복되는 계산



맨 위에서부터 해당 픽셀까지 에너지가 가장 작은 seam을 구하기 위해 윗 픽셀들까지의 low-energy를 알아야 한다.

만약 (0, 1)에서 자신까지의 에너지가 가장 작은 seam을 구한다면 (0, 0), (1, 0)의 에너지를 참고해 이들 중 가장 작은 에너지를 자신의 에너지와 더해서 저장한다.

마찬가지로 (1, 1)에서 자신까지의 에너지가 가장 작은 seam을 구한다면 (0, 0), (1, 0), (2, 0)을 참고한다.

➡ (1, 0), (2, 0)을 여러 번 참조하게 되며, 이는 매 번 계산하기보다는 한 번만 계산해서 저장해 놓고 사용하는 것이 효율적이다. (이미지의 사이즈를 생각하면, 더 많은 픽셀들이 더 많이 참조된다.)

Seam Finding – Dynamic Programming

lowest-energy seam 찾기

모든 픽셀마다 $M(x,y)$ 를 구했다면, 맨 아래의 행의 가장 작은 $M(x,y)$ 를 선택하고, top-left, top, top-right의 값 중 가장 작은 값을 선택해가며 모든 행의 위치를 기록하며 맨 위의 행까지 올라간다.

➡ 이미 각 픽셀까지의 lowest 에너지를 구한 것이므로 이미지 전체의 lowest-energy seam이 구해진다.



seam finding을 통해 찾은 lowest-energy seam



lowest-energy seam을 이용한 image stitching

Image Blending

02 Multi-Band Blending

Image Blending

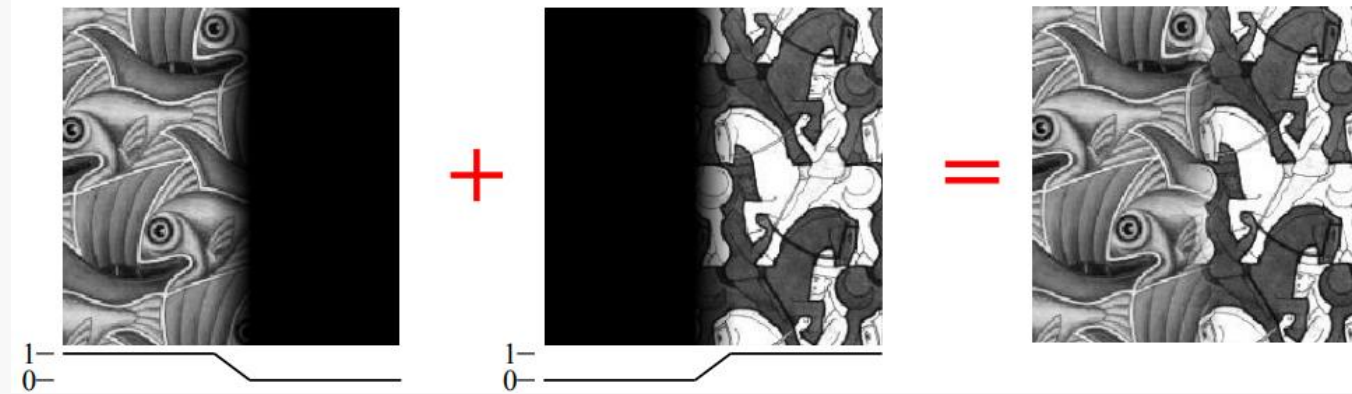
두 이미지를 합쳤을 때, 경계선 부근을 더 자연스럽게 하기 위한 과정



합쳐진 이미지들이 더 자연스럽게 보이기 위해 Blending 과정을 수행해야 한다.

Alpha Blending

Blending 구간의 투명도 a 를 이용한 Alpha Blending ($0 < a < 1$)

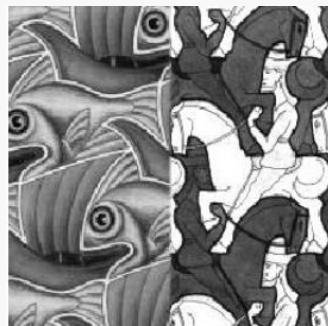


$$I_{blend} = a \times I_{left} + (1 - a) \times I_{right}$$

접합선 부근의 blending 구간에서
왼쪽 이미지는 a 만큼, 오른쪽 이미지는 $1-a$ 만큼 표현됨



두 이미지가 겹쳐서 보임



접합선이 눈에 띄

이 두 문제가 나타나지 않는 최적의 blending 구간과 a 를 찾아 적용해야 함

그런데, 이미지마다 최적의 구간과 a 값이 다를 수 있음

→ 접합선 부근의 픽셀의 강도 차이가 이미지마다 다르기 때문

Multi-Band Blending 사용

Multi-Band Blending

이미지 신호의 주파수마다 다른 α 값을 사용해 Image Blending

먼저, 두 이미지의 Gaussian Pyramid를 구한다.

1. 원본 이미지가 Gaussian Pyramid의 첫 이미지가 되고, 원본 이미지를 Gaussian Filter와 2D Convolution 연산 후 $\frac{1}{2}$ 로 크기를 줄인다.
2. 위 단계를 n 번 반복한다. (일반적으로 n 은 4-7, 그림에서는 간편하게 적은 n 으로 표시)

Gaussian Filter는 Low Pass Filter의 역할을 하며 aliasing의 문제를 방지해 줌

➡ 단계가 반복될수록 이미지의 저주파 성분만 남는다.



Gaussian Pyramid는 두 이미지 모두 만들어야 한다.

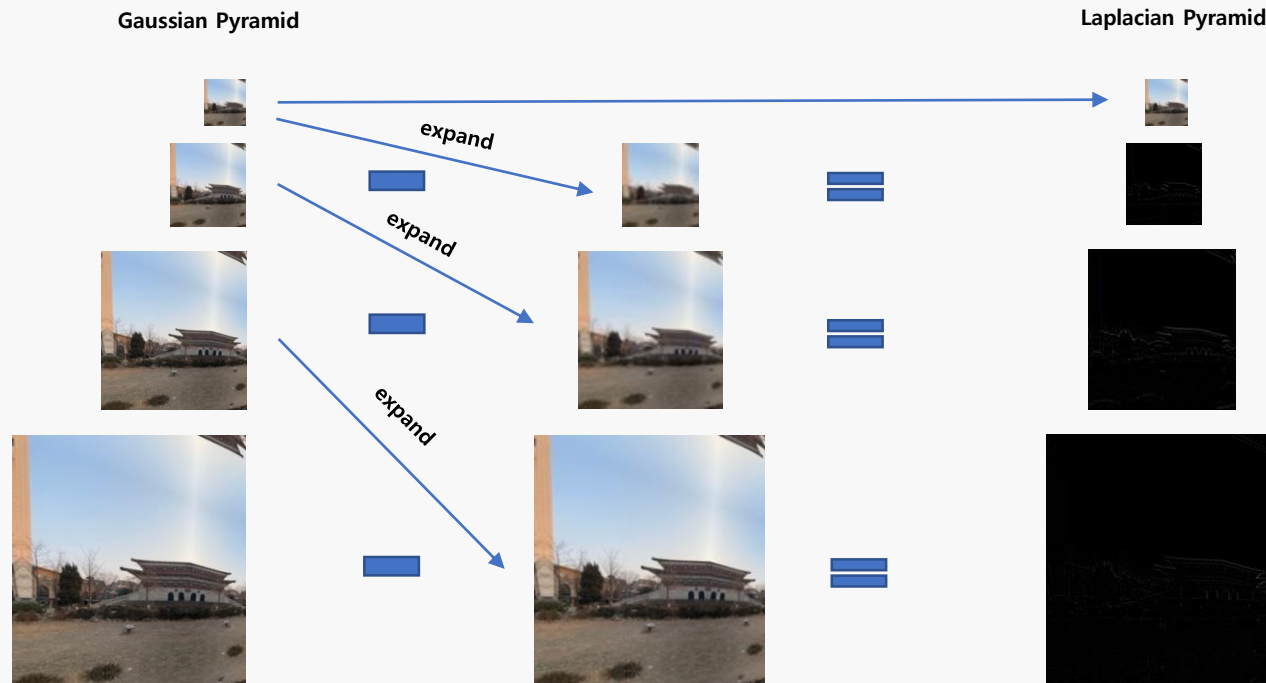
Multi-Band Blending

다음으로, 두 이미지의 **Gaussian Pyramid**를 이용해 **Laplacian Pyramid**를 구한다.

1. Gaussian Pyramid의 가장 작은 이미지 (가장 저주파의 이미지) 를 Laplacian Pyramid의 마지막 이미지가 된다.
2. Gaussian Pyramid의 두 번째로 작은 이미지와 가장 작은 이미지를 두 배로 upsampling한 이미지의 차를 구한다.
3. 위 단계를 반복해 Gaussian Pyramid의 개수와 Laplacian Pyramid의 개수가 같게 한다.

$$Laplacian Image_k = Gaussian Image_k - Upsampled Gaussian Image_{k-1}$$

➡ 단계가 반복될수록 이미지의 고주파 성분만 남는다. 덜 blur된 이미지에서 더 blur된 이미지를 뺀기 때문 (blur가 덜 될수록 고주파 성분을 더 많이 포함한다.)



Gaussian Pyramid와 마찬가지로
Laplacian Pyramid 역시 두 이미지 모두 만들어야 한다.

Multi-Band Blending

두 이미지에 대한 Mask의 Gaussian Pyramid를 구한다.

1. Mask는 0과 1로 표현되며 접합선 좌표를 기준으로 이미지가 표현되어야 하는 부분은 1, 이미지가 표현되면 안되는 부분은 0으로 나타낸다.
2. 앞의 방법을 이용해 Mask의 Gaussian Pyramid를 구한다.

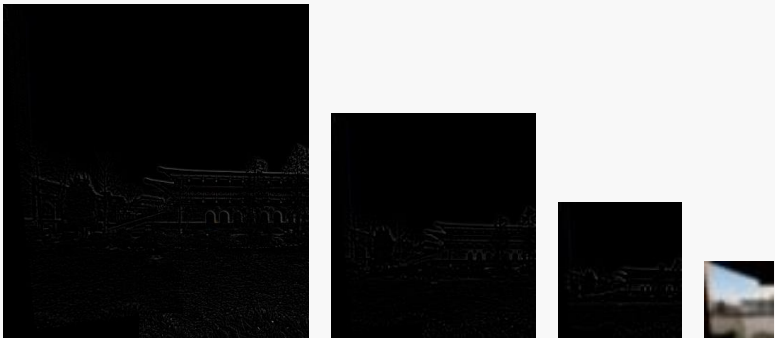
Gaussian Pyramid의 단계에 따라 마스크의 접합선 부분이 blur되는 정도가 다른데 이를 이용해 주파수 별로 다른 가중치로 blending을 수행할 수 있게 된다. 0과 1로 이루어진 Mask가 blur되면 0과 1 경계선 부근은 0과 1 사이의 값으로 표현된다.

$$\text{Blending Image}_k = \text{Laplacian Image1}_k \times \text{Mask1}_k + \text{Laplacian Image2}_k \times \text{Mask2}_k$$

→ 저주파 성분에는 blur가 더 많이 된 Mask를 이용해 blending
고주파 성분에는 blur가 덜 된 Mask를 이용해 blending

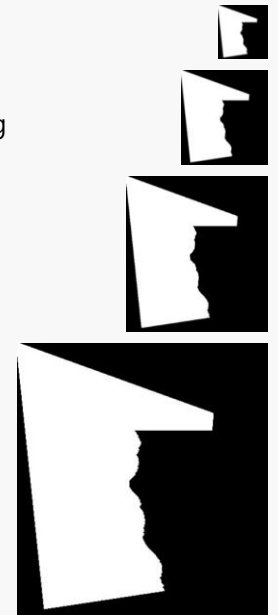
→ 물체의 윤곽선은 살리고
윤곽선이 없는 부분은 더 과감하게 blending

Mask와 Laplacian Pyramid 이미지 들로 구한 Blending Image들



각 단계별로 Laplacian Image와 Mask를 곱해 Blending 이미지를 얻게 되면 이 역시 하나의 Laplacian Pyramid와 같이 나타난다.

Mask의 Gaussian Pyramid

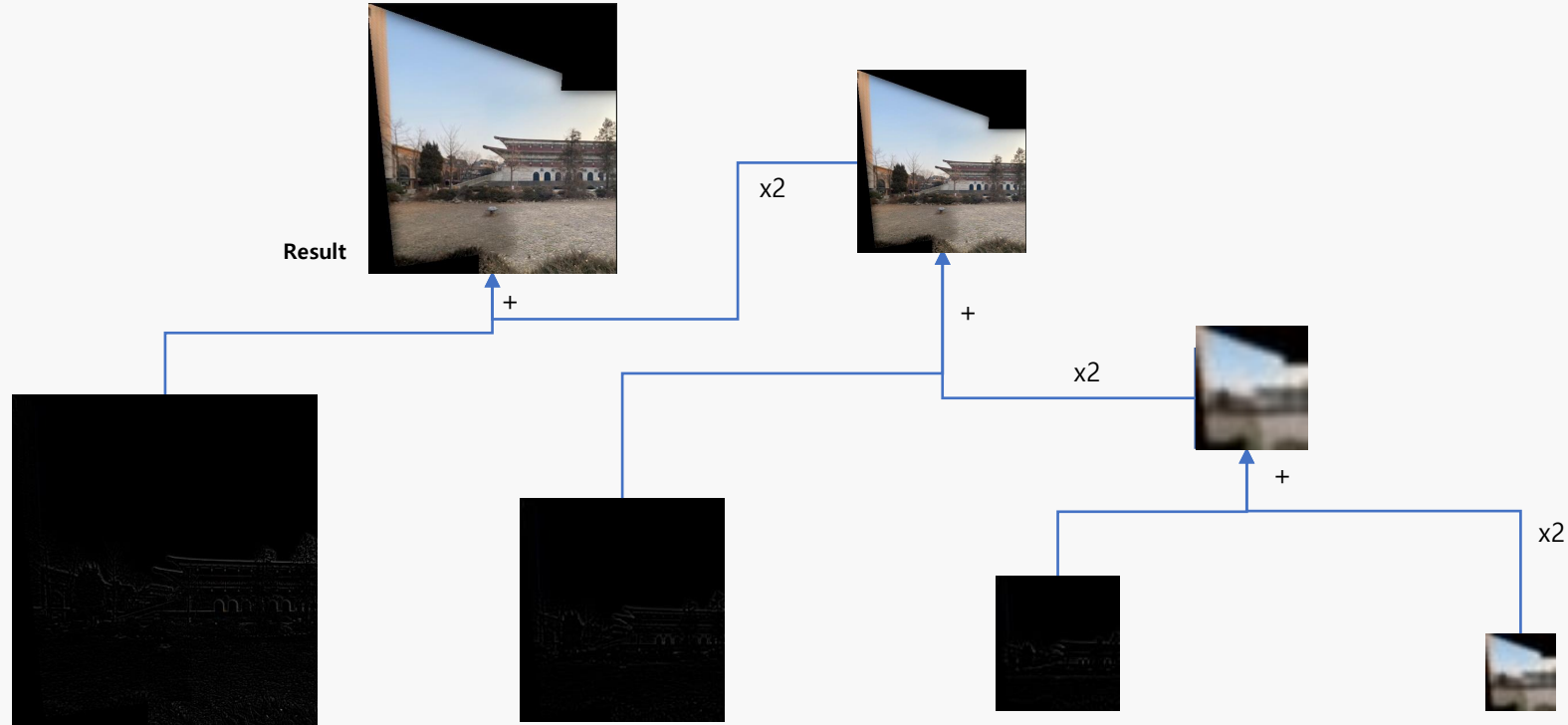


Mask 역시 두 이미지에 대한 Mask들의 Gaussian Pyramid가 필요하다.

Multi-Band Blending

Reconstruct 과정을 통한 Blending Image 구하기

1. 앞에서 구한 Blending 이미지들의 Pyramid에서 가장 작은, 마지막 이미지를 2배로 키워 두 번째로 작은 이미지와 사이즈가 같게 한다.
2. 두 이미지를 더한다.
3. 2번의 이미지를 2배로 키워 세 번째로 작은 이미지와 사이즈와 같게 한 후 두 이미지를 더한다.
4. 위 과정 반복



Multi-Band Blending

주의할 점

일반적으로 인터넷의 예제들은 앞에서 사용한 이미지와 달리 검은색 배경이 없음

이대로 blending을 하게 된다면 사진과 검은 배경 사이의 픽셀 변화 값이 크기 때문에 해당 부분이 고주파 영역으로 해석된다.

따라서 상대적으로 덜 blur된 mask로 blending 되므로 제대로 blending 되지 않음 본 ppt 16번 슬라이드 참고

만약 Mask의 접합선 부분이 이미지의 고주파 부분이면 상대적으로 문제가 덜 발생한다.

추가적으로 접합선 부분이 이미지의 저주파 부분이며 비교적 밝은 부분일수록 해당 문제가 더 많이 발생하는 것으로 예상된다.
(밝기가 영향을 미치는 이유는 픽셀 값이 클수록 mask의 가중치의 영향을 더 많이 받기 때문으로 추측함)

해결 방법

mirroring을 사용해 접합선 주변에 검정 배경이 없도록 한다.

→ 실제 이미지 테두리 주변에서 픽셀 값의 변화가 적어지기 때문에 테두리가 고주파 영역으로 해석되지 않음



이미지의 실제 테두리 주변을 미러링해서 Blending 해야 한다.

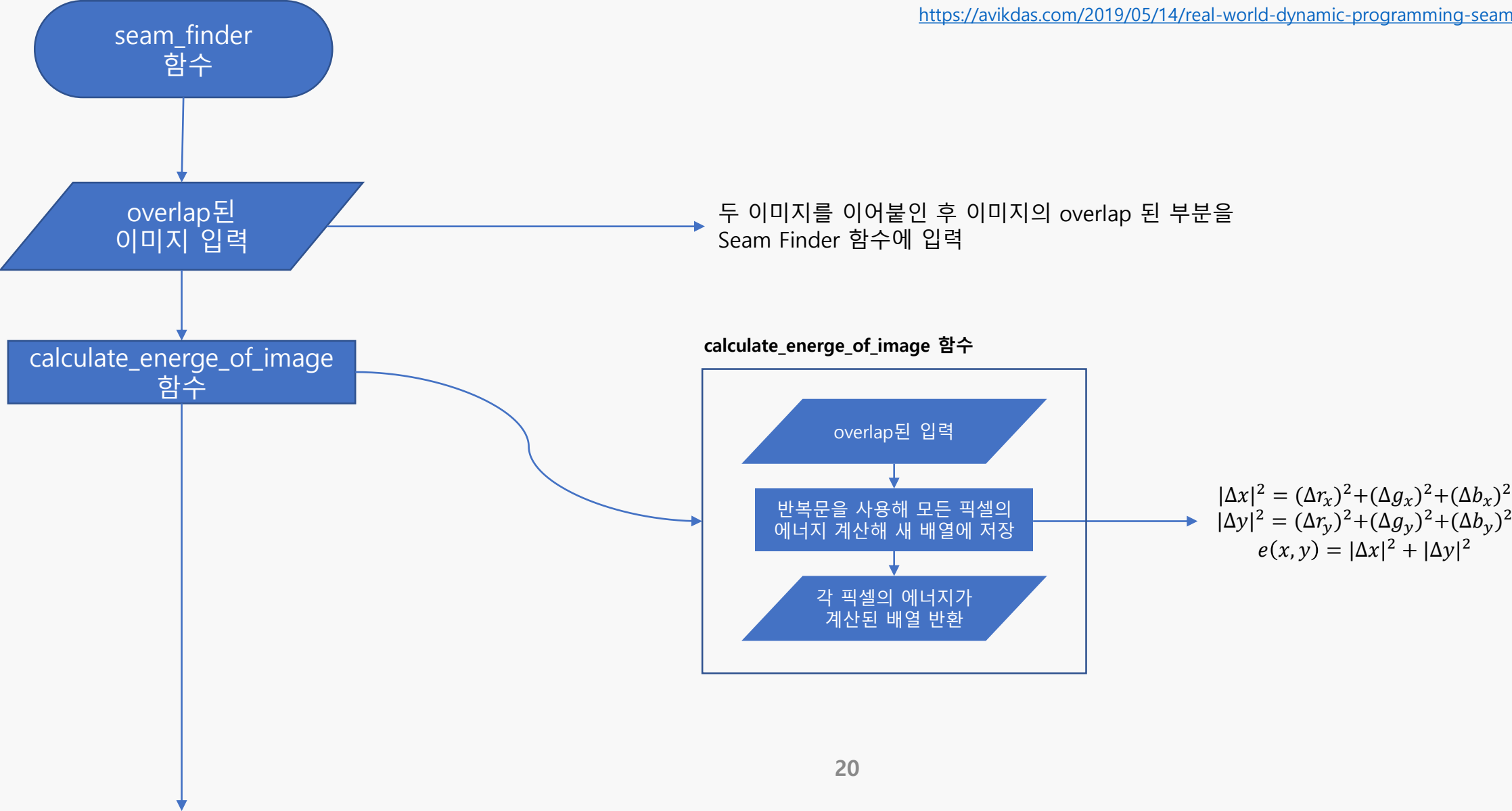


Image Blending

03 프로그램 순서도

Seam Finder 프로그램 순서도

<https://avikdas.com/2019/05/14/real-world-dynamic-programming-seam-carving.html> 참고



calculate_lowest_seam_energies_map
함수

calculate_lowest_seam_energies_map 함수

픽셀의 에너지로
이루어진 배열 입력

반복문 사용하여
top row의 픽셀들은 자기 자신의 에너지
로 lowest seam energy 저장

나머지 픽셀들은
top left, top, top right 픽셀들 중 최소값과
자기 자신의 에너지를 더해
lowest seam energy 저장

반복문 종료

lowest_seam_energy
배열 반환

calculate_lowest_seam_energies 함수

lowest_seam_energy
배열 입력

맨 아래 row부터 해당 행에서 가장 작은
값의 좌표 선택, 좌표 저장

반복문 사용하여 해당 좌표의 top left, top,
top right 픽셀들 중 최소값의 좌표를 선택,
좌표 값 저장을 반복

좌표들의 배열 반환

calculate_lowest_seam_energies 함수

↓
종료

기존의 stitching 이미지

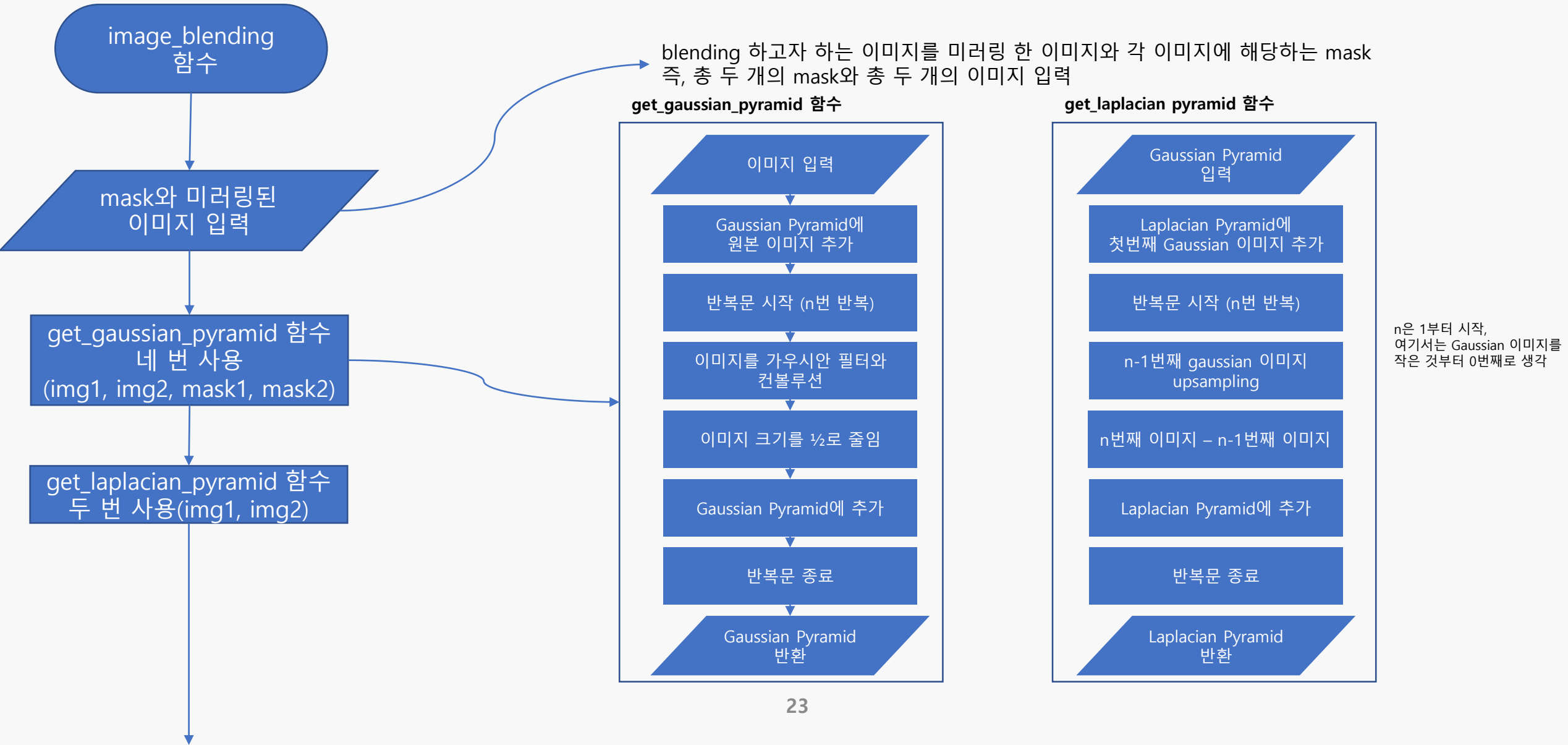


seam_finder 함수로 구한 seam 좌표로 stitching한 이미지



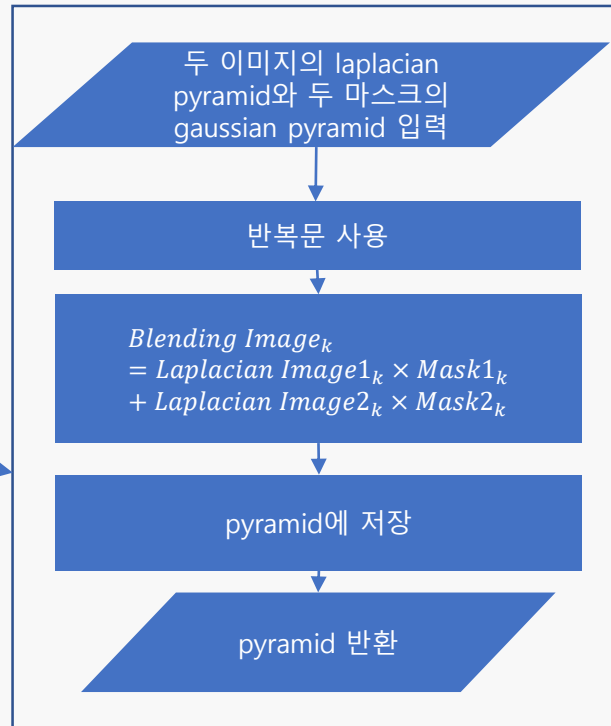
지붕이 어긋난 부분이 seam finder를 이용하자 개선되었다.

Multi-Band Blending 프로그램 순서도

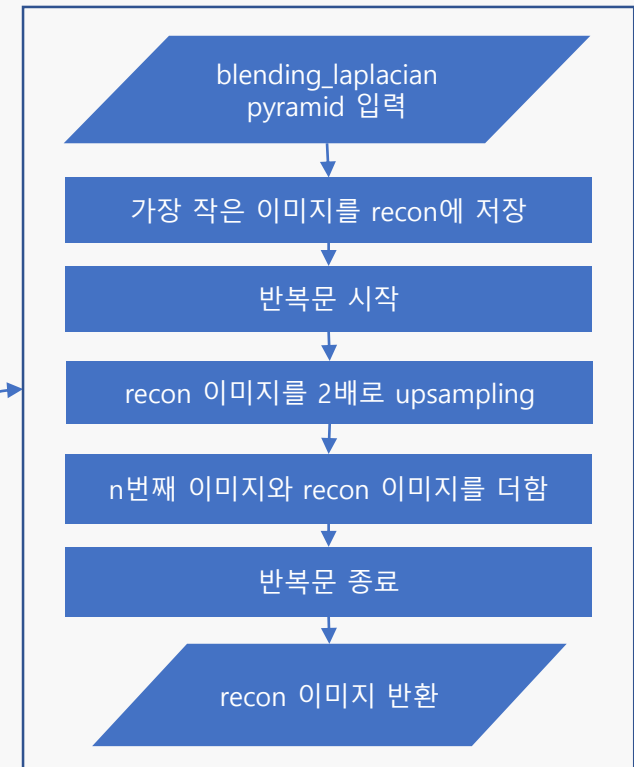


get_blending_laplacian_pyramid 함수

get_blending_laplacian_pyramid 함수



reconstruct 함수



reconstruct 함수

↓
종료

기존의 이미지



Multi-Band blending 적용



경계선 부근이 확실히 부드러워져 눈에 띄지 않는다.

THANK YOU –
감사합니다.