**OSP-Lec02-Display**

1871098 손수민

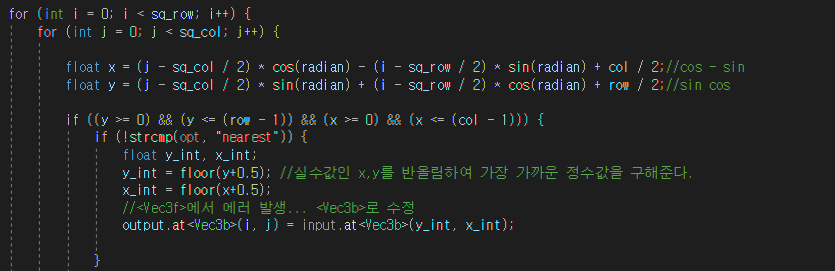
1. Rotate\_skeleton\_v2.cpp

해당 코드에서 메인이 되는 부부은 함수 myrotate이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

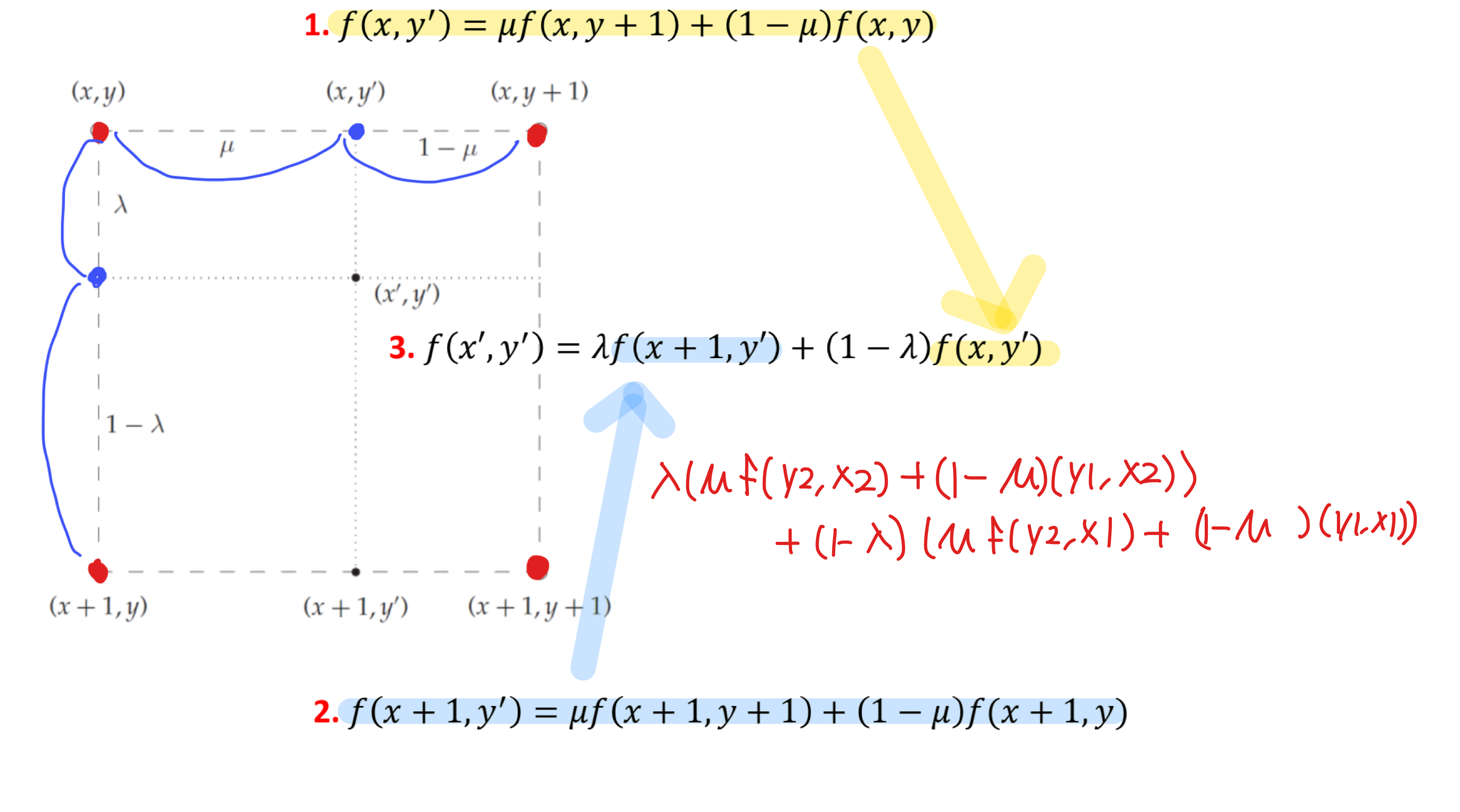
우선 각도로 입력된 angle을 라디안으로 변환하여 sinθ, cosθ 계산이 용이하게 해준다. 그 후 회전 변환될 이미지(output)의 크기를 계산하여 행과 열을 sq\_row와 sq\_col에 저장한다. 그 후 ouput을 초기화해준다.



이중 for문을 돌며 output의 (i,j)에 들어갈 값을 구하게 된다. 우선 output에서 input으로 맵핑되는 inverse warping을 진행해준다. 그렇게 되면 실수값인 x’,y’을 구할 수 있게 된다. 이는 실수값이기 때문에 정수값을 구하기 위해 interpolation 과정을 거쳐주어야 한다. 해당 코드에서 nearest interpolation과 bilinear interpolation을 구현해야 한다.

먼저, 세번째 인자값으로 nearest가 들어온다면, 가장 가까운 값을 copy-paste하면 된다. 따라서 반올림 과정을 거쳐 가장 가까운 정수값을 구해준 뒤, 이를 그대로 입력해 구현해주었다.

두번째로, 세번째 인자값으로 bilinear가 들어온다면, 가장 가까운 4개의 점을 구해준 뒤 공식에 맞추어 f(x’,y’)를 구해주면 된다.



텍스트이(가) 표시된 사진

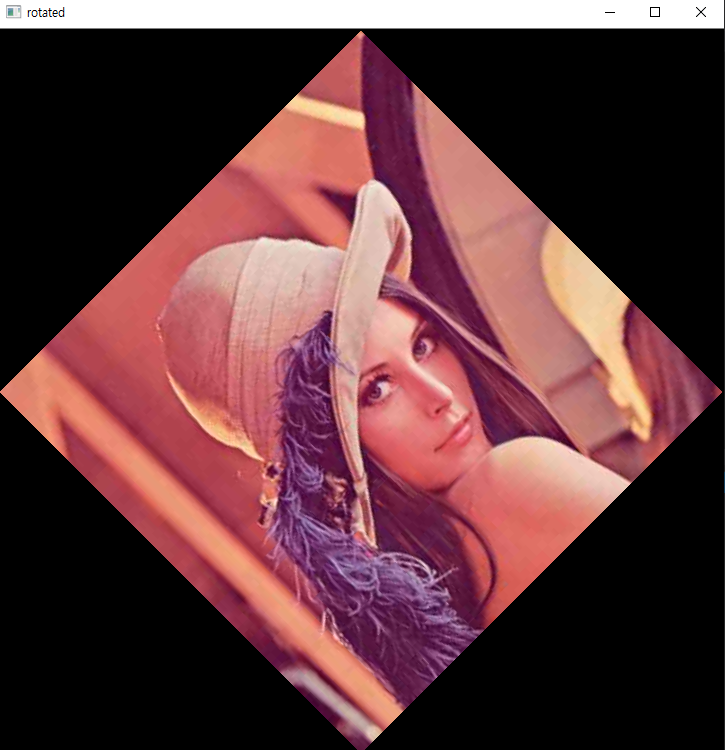
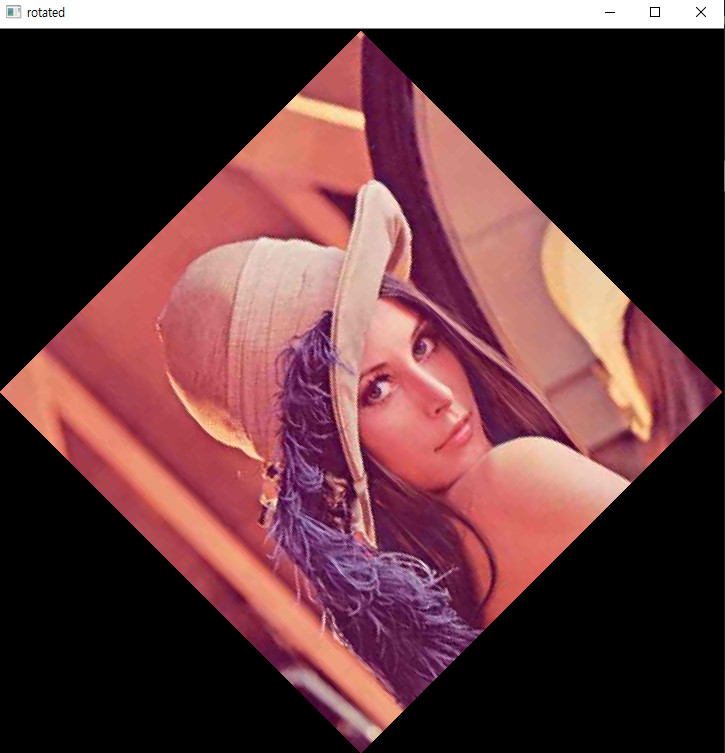
자동 생성된 설명Bilinear interpolation 하는 과정(pdf 참고)이다. 1,2의 공식을 이용해 위와 같이 3을 유도해낼 수 있다. 즉 실수값인 x,y를 올림한 값 x2,y2와 내림한 값 x1,y1을 이용해 실수값의 intensity를 구할 수 있게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명텍스트이(가) 표시된 사진

자동 생성된 설명

Main()함수에서 위에서 구현한 myrotate를 이용해 이미지를 nearset interpolation 방법을 사용해 45도 회전해보았다(좌). Bilinear interpolation 방법을 사용해서도 45도 회전해보았다(우).

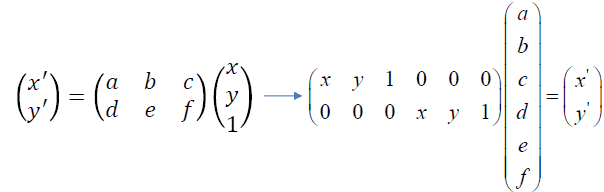


1. Stitching.cpp

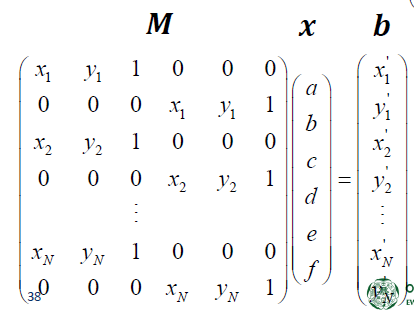
Stitiching.cpp는 affine transformation을 이용해 일치하는 픽셀들을 찾아 이미지L과 이미지R을 붙이는 code이다.

우선 두 이미지의 affine transformation matrix를 구하는 cal\_affine 함수를 알아보자.

Cal\_affine 함수는 두 이미지의 x,y 값들의 쌍과 그 개수를 입력받아 affine transformation matrix를 반환해주는 함수이다.



Affine transformation matrix에는 왼쪽과 같이 6개의 미지수가 존재한다. 오른쪽은 inverse warping을 진행하는 과정이다. 오른쪽과 같이 inverse warping에서 미지수 a,b,c,d,e,f 6개를 구하기 위해서는 등식 6개가 필요하다. 한 쌍의 맵핑되는 (x,y), (x’,y’)에서 등식 2개를 유도할 수 있으므로 적어도 3개의 쌍이 필요하다는 것을 알 수 있다.



그럼 위와 같이 Mx = b로 표현할 수 있다. 그렇다면 미지수 x를 구하기 위해서는



위와 같은 과정을 통해 구할 수 있게 된다.

텍스트, 화면, 스크린샷, 닫기이(가) 표시된 사진

자동 생성된 설명

위의 코드가 이를 구현한 것이다. Mat M은 2NX6, b는 2Nx1의 형태를 띄게 되므로 해당 형태에 맞게 Mat을 선언하고 for문을 돌며 Mat M에 x,y,1 값(ptl)을 넣어주며 초기화를 진행해준다.

그 후 M의 transpose와 invert를 구해주고 공식에 맞추어 x(affine matrix)를 반환하게 된다.

다음으로는 두 이미지를 합쳐주는 blend\_stitching이라는 함수이다.

Left image인 Mat I1, right image인 Mat I2, 회전한 이미지인 Mat &l\_f, 이미지의 bound값, blend과정에서의 alpha값을 받아 합쳐진 이미지를 생성해내는 함수이다.

텍스트이(가) 표시된 사진

자동 생성된 설명

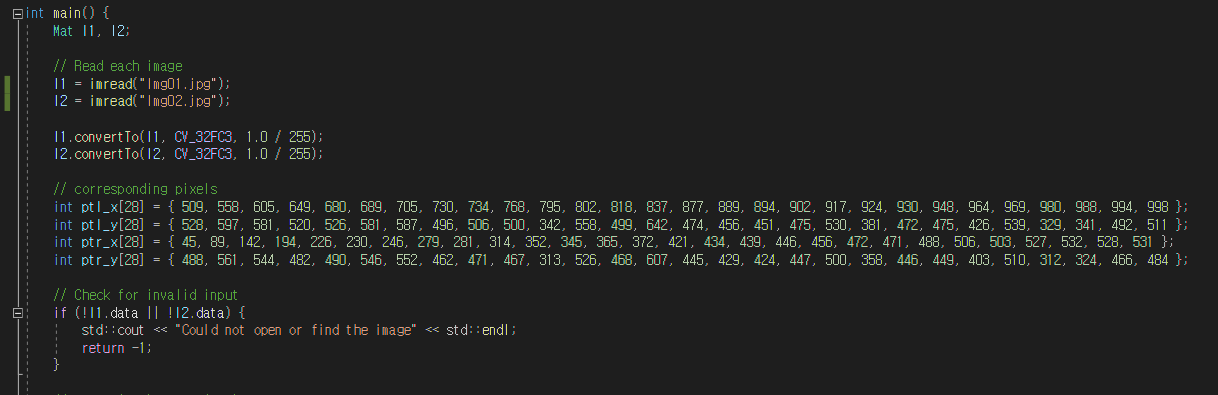
먼저 col, row 함수에 열과 행의 길이를 받아온다.

그 후 이중 for문을 돌며 I\_f.at<Vec3f>(i - bound\_u, j - bound\_l) != Vec3f(0, 0, 0) ? true : false; 해당 조건에 따라 코드가 분기 되는데, 이는 I2에서 I1으로 회전된 이미지인 l\_f가 (0,0,0)이 아니라면, 즉 회전된 이미지의 안에 존재하는 영역이라면 if문의 코드가 실행되고, 회전된 이미지의 바깥에 존재하는 영역이라면 else문이 실행된다는 의미이다.

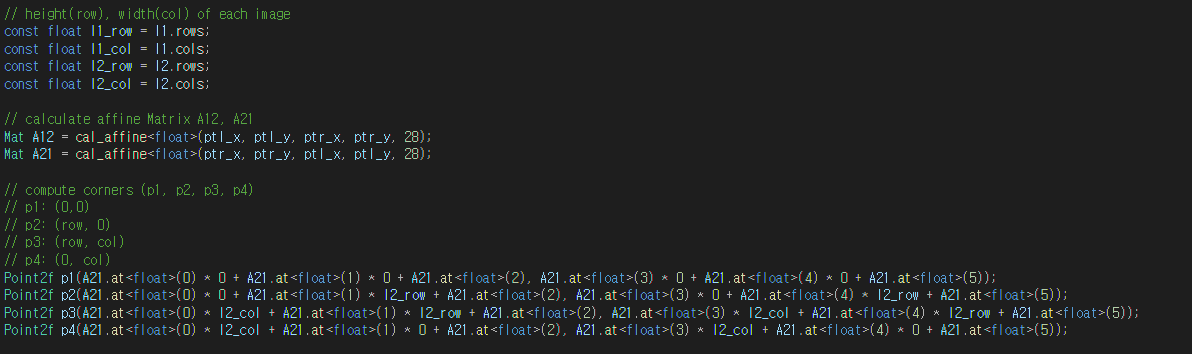
그래서 if문에서는 I\_f이미지와 I1의 이미지를 blending한 a\*I1 + (1-a)l\_f가 들어가게 되고

else문에서는 I1이미지 하나만 들어가게 되는 것이다.

마지막으로 이를 전체적으로 수행하는 main함수를 살펴보면



우선 이미지를 읽어 들여 normalization을 진행하고 제대로된 이미지인지 체크해준다. 그리고 매칭되는 (x,y),(x’,y’) pair가 28개임을 알 수 있다.



각각 이미지들의 열과 행의 길이를 저장하고, cal\_affine 함수를 통해 affine transformation matrix를 구해준다.

그 다음은 I2에서 I1에 맞춰 회전될(affine) 이미지의 각 모서리 값들을 계산하는 과정이다.

구한 값과 I1의 열과 행 길이값들을 이용해 합쳐질 이미지의 bound값을 구할 수 있다.

텍스트이(가) 표시된 사진

자동 생성된 설명

텍스트, 노트북, 스크린샷이(가) 표시된 사진

자동 생성된 설명

그 후 앞의 Rotate\_skeleton\_v2.cpp에서 구현했던 것처럼 bilinear interpolation을 이용해 inverse warping을 진행해 affine된 이미지 I\_f를 구해준다.

텍스트이(가) 표시된 사진

자동 생성된 설명

그리고 blend\_stitching을 이용해 이미지를 합성해주면 된다.

결국 다음과 같은 결과가 나오게 된다.

텍스트이(가) 표시된 사진

자동 생성된 설명