

# Image Warping

---

---

17010826 김성민

**01** 호모그래피 변환

**02** RANSAC 알고리즘

**03** 이미지 매칭 판독을 위한 확률 모델

**04** 프로그램 순서도

**목표 : Feature Matching 정보를 활용해 이미지를 자연스럽게 이어 붙이는 것**



더 나아가 사진 N장을 랜덤으로 입력했을 때  
알고리즘이 매칭해야 하는 이미지 쌍을 찾고, 자연스럽게 이어 붙이도록 하는 것

# Image Warping

## 01 호모그래피 변환

# 호모그래피란?

3차원 공간의 평면을 서로 다른 시각에서 바라봤을 때 획득되는 영상 사이의 관계로 하나의 평면을 다른 평면으로 투시 변환 하는 것과 같은 맥락

그림 가운데의 사각형 평면을 다른 위치에서 찍은 사진을  $O_L$ 과  $O_R$  이라고 하면 우리는 사각형 평면을  $O_L, O_R$ 에 투시 변환한 것이다.

→ 즉, 기존의 물체와 사진의 물체는 일종의 변환 관계를 가지는데, 이 관계가 호모그래피인 것이다.

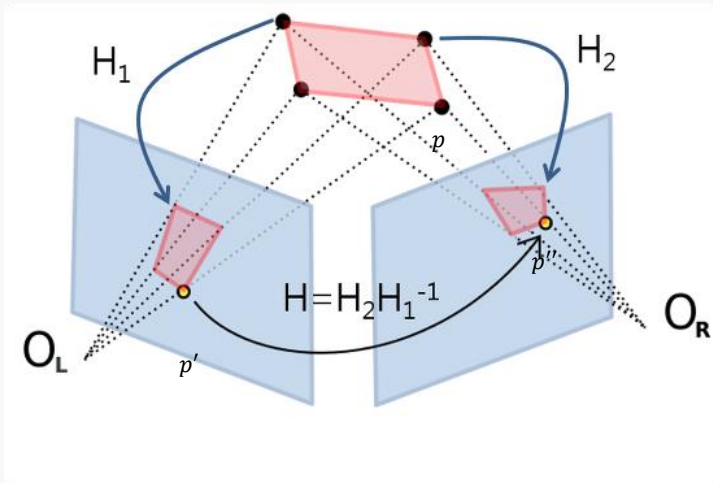
하지만 우리는 물체와 사진 사이의 변환 관계가 아닌 사진과 사진 사이의 변환 관계를 알아야 한다.

원래 물체의 좌표를  $p, O_L, O_R$ 에서 물체의 좌표를  $p', p''$ 이라 하고, 물체와  $O_L, O_R$  사이의 변환 관계를 각각  $H_1, H_2$ 라고 한다면 아래의 식이 성립한다.

$$p' = H_1 p, \quad p'' = H_2 p \quad p'' = H_2 p = H_2 (H_1^{-1} p')$$

→  $O_L, O_R$  위의 점  $p', p''$ 에 대해서도 호모그래피  $H_2 H_1^{-1}$  관계가 성립된다.

두 이미지를 합칠 때  $O_L, O_R$ 의 겹치는 부분은 서로 다른 시각에서 바라본 영상이며 이를 자연스럽게 합치기 위해서  $O_L$  상의 모든 점을  $O_R$ 에 맞게 호모그래피 변환해야 한다.



# 호모그래피 행렬 구하기

호모그래피 변환은 호모그래피 행렬을 통해 이루어지며 호모그래피 행렬은 3x3의 크기를 가진다.

$$\begin{bmatrix} h_1 & h_2 & h_3 \\ h_4 & h_5 & h_6 \\ h_7 & h_8 & h_9 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \quad \longrightarrow \quad \begin{aligned} h_1x + h_2y + h_3 &= x' : \text{Row1} \\ h_4x + h_5y + h_6 &= y' : \text{Row2} \\ h_7x + h_8y + h_9 &= 1 : \text{Row3} \end{aligned}$$

우리가 활용할 때  $x, y$ 는 변환 시킬 이미지의 matching feature이고,  $x', y'$ 은  $x, y$ 와 matching되는 feature이다.

이렇게 얻어진 방정식들로 3개의 관계식을 만들어 이를 행렬로 표현한다.

$$\begin{aligned} y' \text{Row3} - \text{Row2} &= 0 \\ -x' \text{Row3} + \text{Row1} &= 0 \\ -y' \text{Row1} + x' \text{Row2} &= 0 \end{aligned} \quad \longrightarrow \quad \begin{bmatrix} 0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y' \\ x & y & 1 & 0 & 0 & 0 & -x'x & -x'y & -x' \\ -y'x & -y'y & -y' & x'x & x'y & x' & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ h_9 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

여기서 3x9 행렬의 마지막 행은 첫째 행과 둘째 행을 조합하면 얻을 수 있는 식이므로 제외하고,  $h_9$ 는 호모그래피 행렬의 크기를 고정하기 위해 1로 정한다.

→ 우리가 구해야 할 미지수는 8개 즉, feature point 4쌍 활용하면  $h_1 \sim h_8$ 을 구할 수 있다.

matching 쌍 1개마다 방정식 2개를 얻으므로 4쌍이 모이면 8개의 방정식으로 8개의 미지수를 구할 수 있다.

# 호모그래피 행렬 구하기

점 4개를 활용해 호모그래피 행렬 구하기 : 앞에서 얻은 방정식 중 첫째 행과 둘째 행을 각 점에 대해 반복

→ 8x9 행렬이 만들어 진다.

$$\begin{bmatrix} 0 & 0 & 0 & -x_1 & -y_1 & -1 & y'_1x_1 & y'_1y_1 & y'_1 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & y'_2x_2 & y'_2y_2 & y'_2 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 & -x'_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

이 8x9 행렬을 특이값 분해(SVD) → U, S(sigma),  $V^T$  행렬을 얻음

$V$ 의 마지막 행이자  $V^T$ 의 마지막 열은  $AX=0$  꼴의 방정식의  $X$ 값이 된다.

→ 8x9 행렬을 특이값 분해해서 얻은  $V$ 행렬의 마지막 행 벡터가 호모그래피 원소들이 된다.

$h_9$ 는 1이므로  $V$ 행렬의 마지막 행 벡터의 마지막 원소가 1이 되도록 스케일 조정을 해야 함에 주의해야 한다.

$V$ 행렬의 마지막 행 벡터 =  $V^T$ 행렬의 마지막 열 벡터

# AX = 0을 풀이할 위한 특이값 분해(SVD)

앞 페이지의 행렬식을 풀기 위해 특이값 분해에 대해 간단하게 알아보자.

$m \times n$  행렬 A의 특이값 분해 :  $U \Sigma V^T$  U는  $m \times m$ ,  $\Sigma$ 는  $m \times n$ ,  $V^T$ 는  $n \times n$  행렬이다. (U와 V는 직교 행렬이고  $\Sigma$ 는 대각 행렬)

$\Sigma$ 의 원소는 0 혹은 양수이며  $\sigma_{00}$ 에서  $\sigma_{kk}$ 로 갈수록 값이 작아지며 그 원소들은 특이값이라고 한다.

한편,  $V_i$ 를  $V^T$ 의 i번째 열 벡터라고 한다면,  $V^T V_i$ 는 i번째 값만 1이고, 나머지는 0인 벡터가 된다. ( $\because V$ 는 직교행렬)

$AV_i = U \Sigma V^T V_i$ 가 되는데,  $V^T V_i$ 는 i번째 값만 0이 아니고,  $\Sigma$ 는 대각 행렬이며  $V^T V_i$ 와  $\sigma_{ii}$  원소와만 연산이 수행된다.

→  $\Sigma V^T V_i$ 는 i번째 원소가  $\sigma_{ii}$ 이고 나머지가 0인 열 벡터가 된다.

만약 i가 1이라면,  $\Sigma V^T V_1$ 은 다음과 같이 계산된다.

$$\begin{bmatrix} \sigma_{00} & 0 & 0 & \cdots \\ 0 & \sigma_{11} & 0 & \cdots \\ 0 & 0 & \sigma_{22} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \end{bmatrix} = \begin{bmatrix} 0 \\ \sigma_{11} \\ 0 \\ \vdots \end{bmatrix}$$

그런데,  $\sigma_{ii}$ 가 0이면? →  $AV_i = 0$ 이 되고, 우리는  $AX = 0$ 을 만족시키는 벡터 X를  $V_i$ 라고 할 수 있다.

특이값  $\sigma_{ii}$  중 0이 없다면? →  $AX = 0$ 을 만족시키는 벡터 X는 존재하지 않고, 이에 최대한 근사하는 벡터를 구해야 한다.  
즉, 가장 작은 특이값에 대응되는  $V_i$ 를 X의 근사값으로 사용

$\sigma_{ii}$ 가 0 혹은 가장 작은 특이값이기 위해서는 i가  $V^T$ 의 크기와 같아야 함(i가 클수록  $\sigma_{ii}$ 가 작기 때문)

→ V의 마지막 열 벡터가  $AX=0$ 의 solution이 된다.



Image Warping

## 02 RANSAC 알고리즘

# RANSAC 알고리즘

## RANSAC 알고리즘을 사용하는 이유

우리가 가진 feature matching쌍에는 잘못된 matching쌍이나 노이즈 등이 많이 포함되어 있다.

따라서 우리가 가진 feature matching 쌍 중 임의의 4쌍을 정해서 구한 호모그래피 행렬이 전체적인 feature matching에 대해 정확하지 않을 수 있음

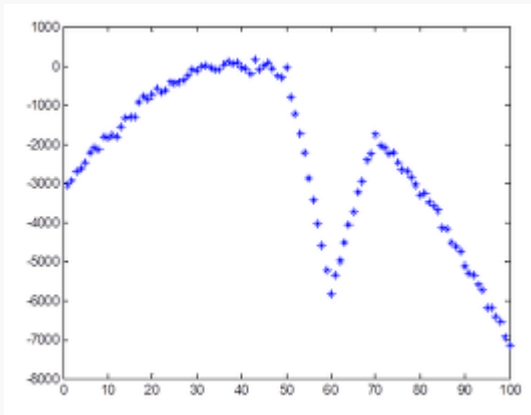
➡ RANSAC 알고리즘을 이용해 전체적인 feature matching에 적합한 호모그래피 행렬을 구해야 한다.

## RANSAC 알고리즘의 컨셉

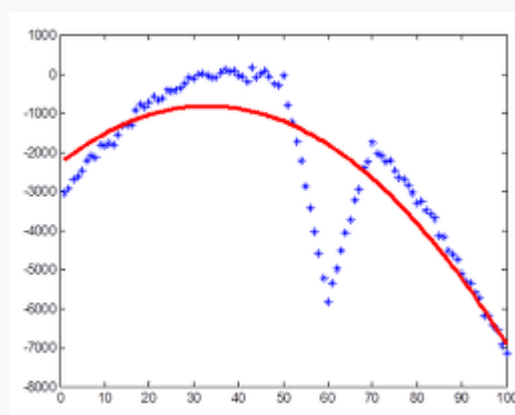
무작위로  $n$ 개의 데이터를 가져와 모델(우리의 경우, 호모그래피 행렬)을 만들고 이 모델을 전체 데이터를 이용해 평가한다.

전체 데이터 중 inlier의 비율이 가장 높은 모델을 최종 결과물로 저장한다.

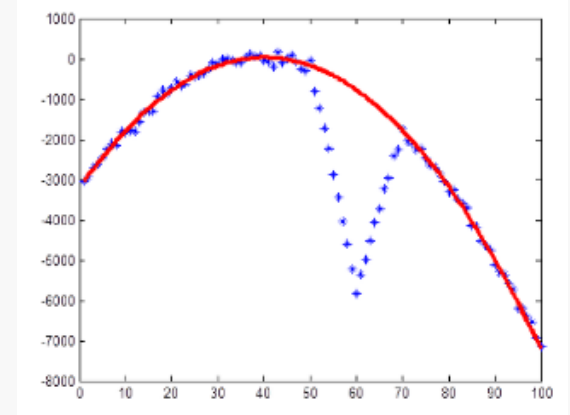
여기서 inlier란 모델과 오차가 크지 않은 데이터들을 말하고, 오차가 큰 데이터들은 outlier(이상치)라고 한다.



이상치가 많은 모델 데이터



$n$ 개의 데이터로 점들을  
가장 잘 표현하는 곡선을 찾는 과정



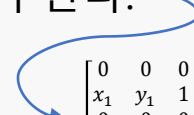
RANSAC 알고리즘을 이용해 찾은 점들을  
가장 잘 표현하는 곡선을 찾은 결과

# RANSAC을 이용해 호모그래피 행렬 구하기

1. 랜덤으로 4개의 matching 쌍을 뽑는다.

2. 4개의 matching 쌍을 이용해 이전 파트에서 설명한 방법으로 호모그래피 행렬을 구한다.

3. 모든 matching 쌍의  $x, y$ 에 대해 2에서 구한 호모그래피 행렬 변환을 한 후  $x', y'$ 과 비교한다.


$$\begin{bmatrix} 0 & 0 & 0 & -x_1 & -y_1 & -1 & y'_1x_1 & y'_1y_1 & y'_1 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & y'_2x_2 & y'_2y_2 & y'_2 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 & -x'_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ h_3 \\ h_4 \\ h_5 \\ h_6 \\ h_7 \\ h_8 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

4. 변환 결과와  $x', y'$ 의 차이가 3이상인 경우 outlier로 판단한다. (Euclidean distance)

5. 1~4 과정을  $n$ 번(사용자 지정) 반복해 전체 데이터 중 outlier의 비율이 가장 적은 호모그래피 행렬을 선택한다.

- 많이 반복할 수록 더 적합한 호모그래피 행렬을 찾을 가능성이 크다.
- Lowe의 논문에서 inlier 확률이 0.5인 경우, 500번 반복했을 때 올바른 호모그래피 행렬을 찾지 못할 확률이  $10^{-14}$ 라고 언급되어 있다.

Image Warping

# 03 이미지 매칭 판독을 위한 확률 모델

# 이미지 매칭 판독을 위한 확률 모델

랜덤으로 입력된 이미지들의 매칭 관계를 알아내기 위한 방법



이미지가 많으면 입력 이미지를 일일이 순서대로 넣어주기 힘들 + 매칭 쌍을 일일이 파악하기 힘들  
따라서, 입력 값이 무작위로 들어오더라도 프로그램이 이를 판단해야 함

**Matching되는 Feature를 이용한 확률 모델의 전체적인 컨셉**

잠재적인 matching 이미지의 각 feature matching 쌍에 대해 RANSAC inlier와 outlier가 존재한다.

이를 이용해 inlier와 outlier set를 이용해 각 set가  
올바른 이미지 match 혹은 올바르지 않은 이미지 match에 의해 생겼을 확률을 구한다.

# 이미지 매칭 판독을 위한 확률 모델

## 확률 모델을 위한 변수 정의

$m \in \{0, 1\}$  : 주어진 이미지들이 올바른/올바르지 않은 matching인 사건

$n_f$  : 주어진 이미지에 대해 오버랩된 영역의 전체 feature 수  $n_i$  : 전체 feature 중 inlier의 수

$f^{(i)}$  : i번째 feature match가 inlier일 확률, 0 또는 1  $f^{(1:n_f)} : \{f^{(i)}, i = 1, 2, \dots, n_f\}$ , 이항 분포를 따름

$P(m = 1)$  : 이미지 matching 성공 확률 (사용자 정의, 논문에서는  $10^{-6}$ )

$P1$  : 이미지 matching 성공 시 feature matching에 성공할 확률 (사용자 정의, 논문에서는 0.6)

$P0$  : 이미지 matching 실패 시 feature matching에 성공할 확률 (사용자 정의, 논문에서는 0.1)

$P_{min}$  : 이미지 matching이 성공했다고 판단할 최소 비율 (사용자 정의, 논문에서는 0.999)

Feature matching에 성공할 확률 = inlier일 확률

## 이미지 매칭 성공 시 오버랩된 영역 내에서 feature matching 성공 확률

$$P\left(f^{(1:n_f)} \middle| m = 1\right) = B(n_f; n_i, P1) = \binom{n_f}{n_i} (P1)^{n_i} (1 - P1)^{n_f - n_i}$$

## 이미지 매칭 실패 시 오버랩된 영역 내에서 feature matching 성공 확률

$$P\left(f^{(1:n_f)} \middle| m = 0\right) = B(n_f; n_i, P0) = \binom{n_f}{n_i} (P0)^{n_i} (1 - P0)^{n_f - n_i}$$

## 오버랩된 영역 내에서 feature matching 성공 확률

$$\longrightarrow P\left(f^{(1:n_f)}\right) = P\left(f^{(1:n_f)} \middle| m = 1\right) P(m = 1) + P\left(f^{(1:n_f)} \middle| m = 0\right) P(m = 0)$$

# 이미지 매칭 판독을 위한 확률 모델

오버랩된 영역 내에 특징점 matching 성공 여부에 따른 이미지 matching 성공 확률

$$P(m=1|f^{(1:n_f)}) = \frac{P(f^{(1:n_f)}|m=1)P(m=1)}{P(f^{(1:n_f)})}$$



베이즈 정리

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$= \frac{P(f^{(1:n_f)}|m=1)P(m=1)}{P(f^{(1:n_f)}|m=1)P(m=1) + P(f^{(1:n_f)}|m=0)P(m=0)}$$

$$= \frac{1}{1 + \frac{P(f^{(1:n_f)}|m=0)P(m=0)}{P(f^{(1:n_f)}|m=1)P(m=1)}}$$

이 값이 설정해 둔  $P_{min}$ 보다 크면 이미지 matching에 성공

확률 모델의 입력과 출력

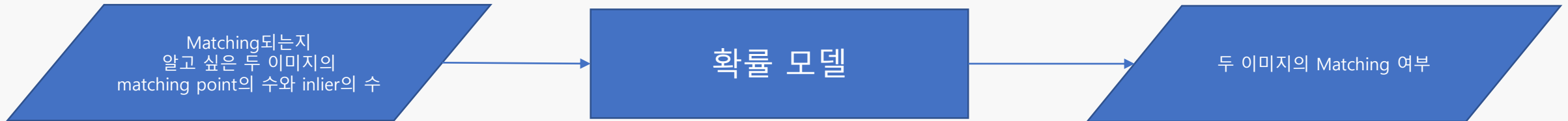
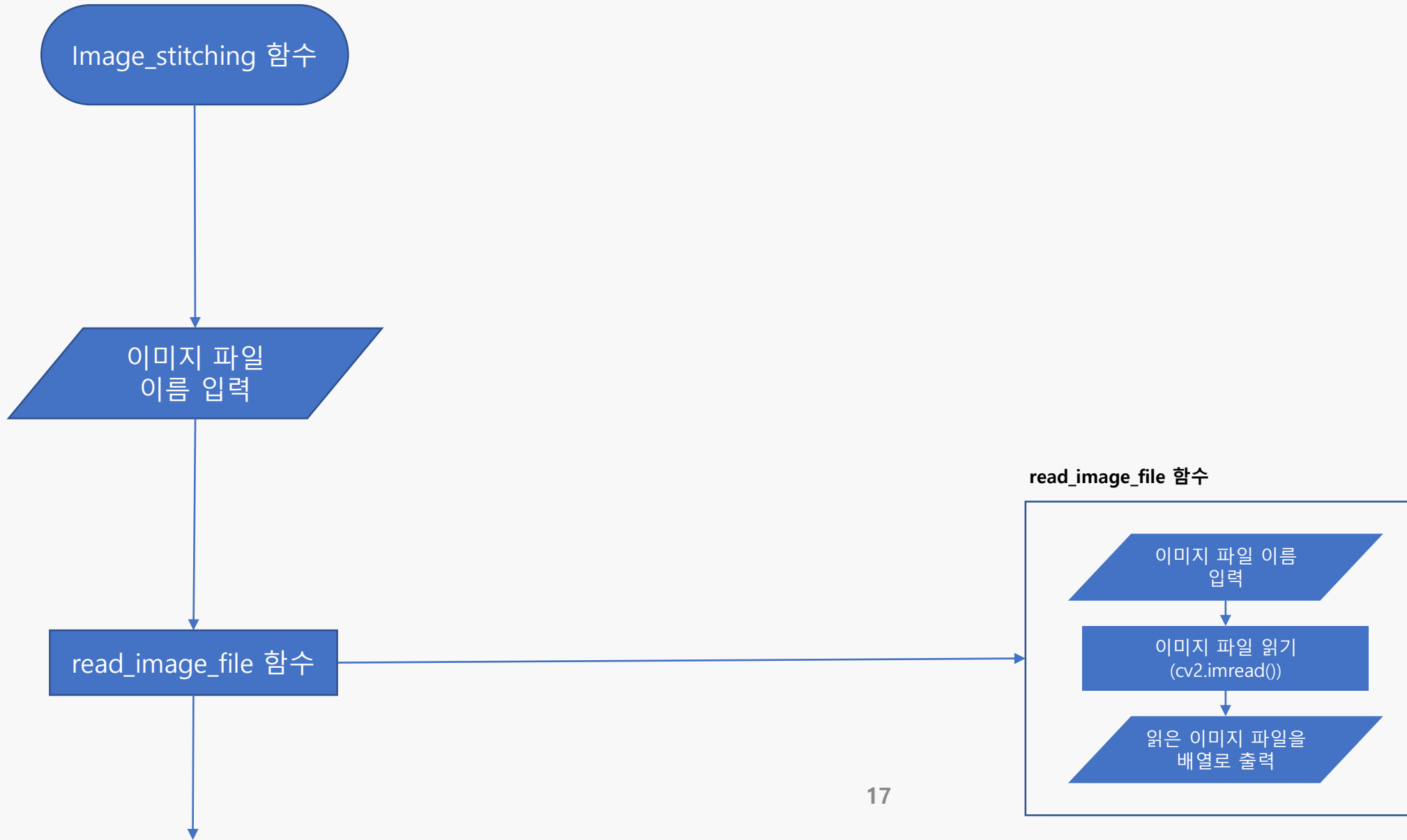


Image Warping

## 04 프로그램 순서도

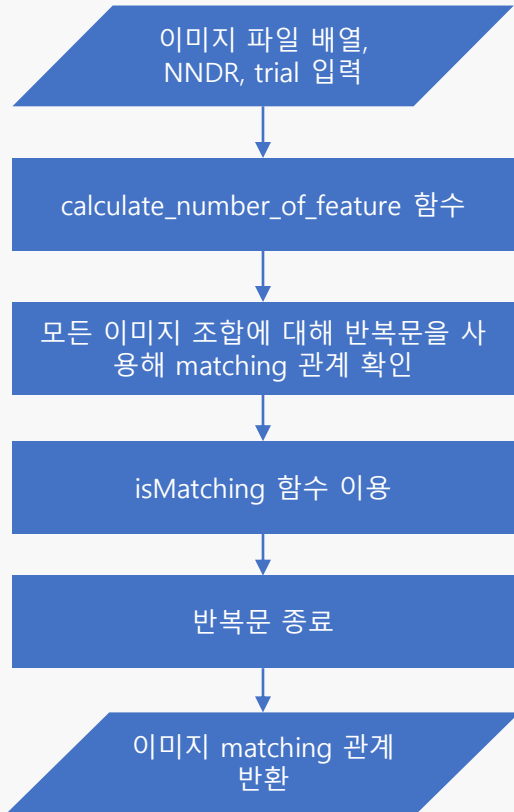


# 프로그램 순서도

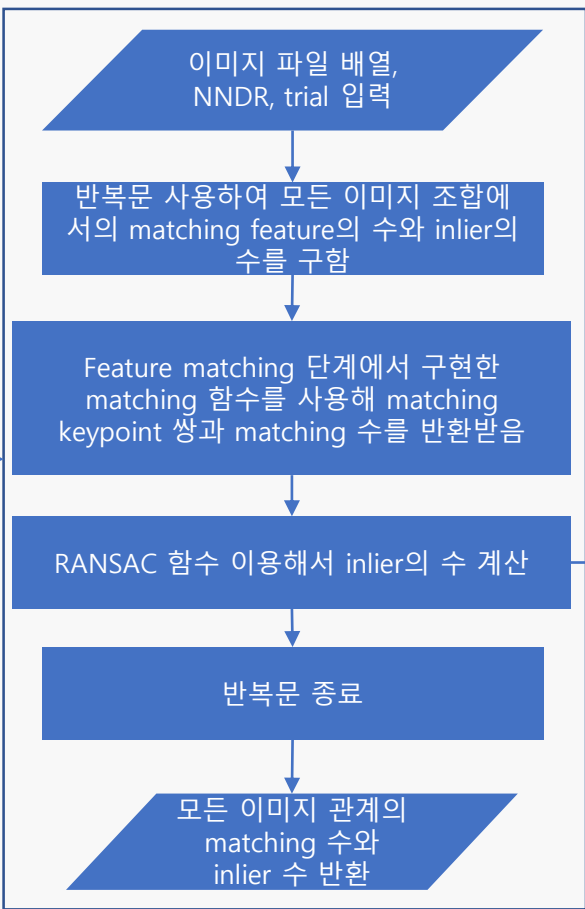


# calculate\_matching\_graph 함수

## calculate\_matching\_graph 함수

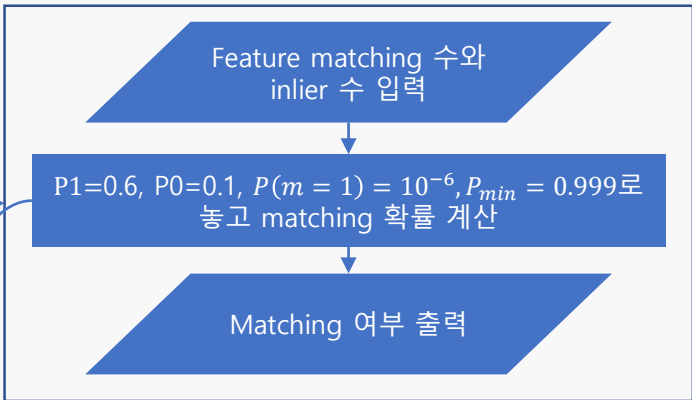


## calculate\_number\_of\_feature 함수



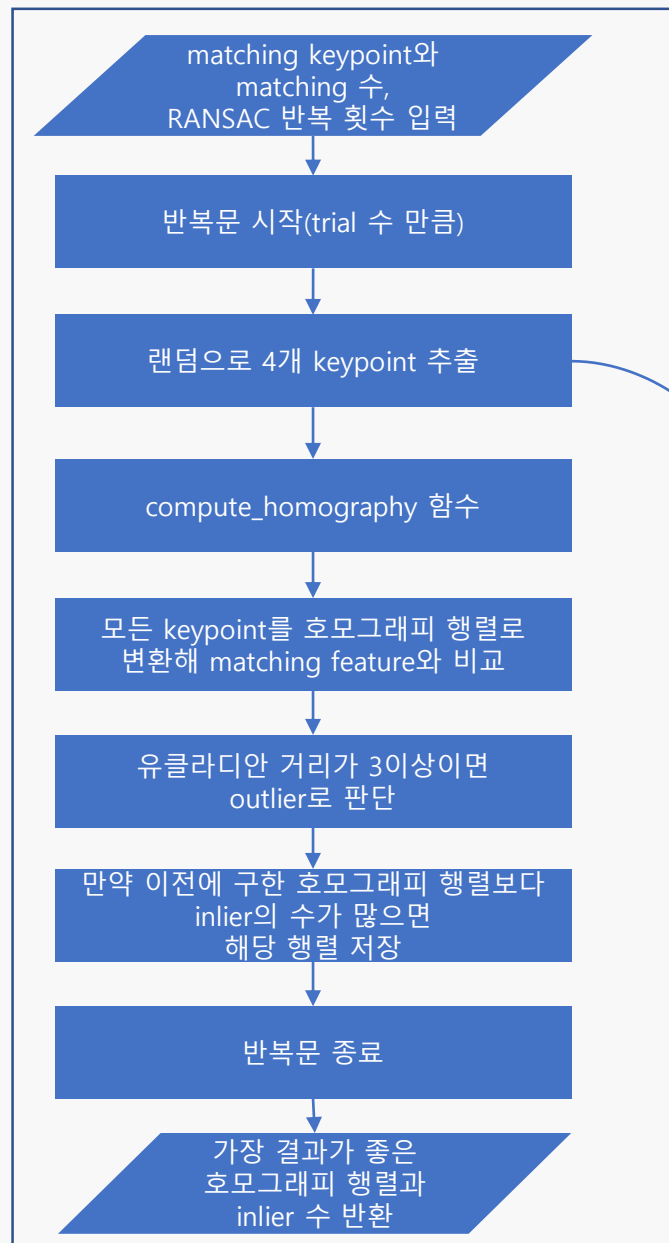
RANSAC 함수는  
뒷 페이지에 설명

## isMatching 함수

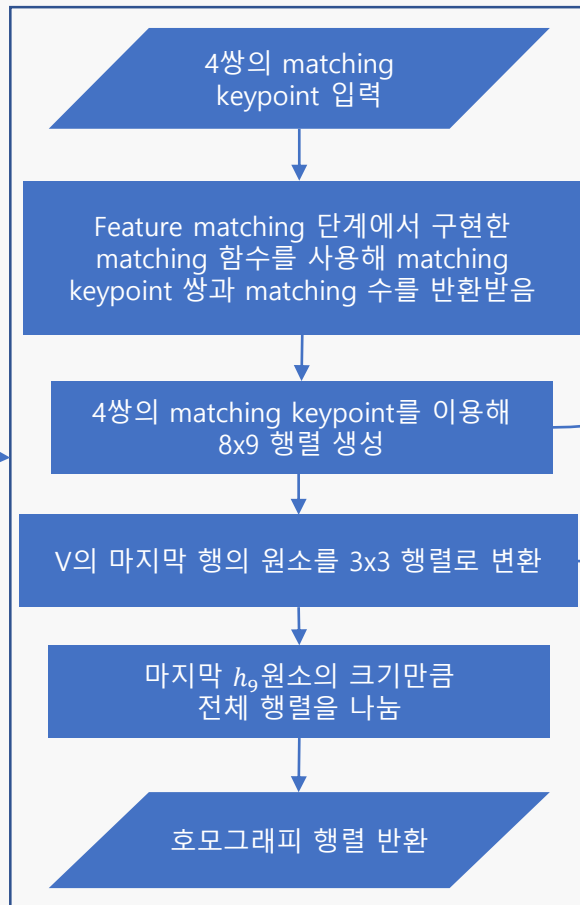


$$P(m=1|f^{(1:n_f)}) = \frac{1}{1 + \frac{P(f^{(1:n_f)}|m=0)P(m=0)}{P(f^{(1:n_f)}|m=1)P(m=1)}}$$

## RANSAC 함수

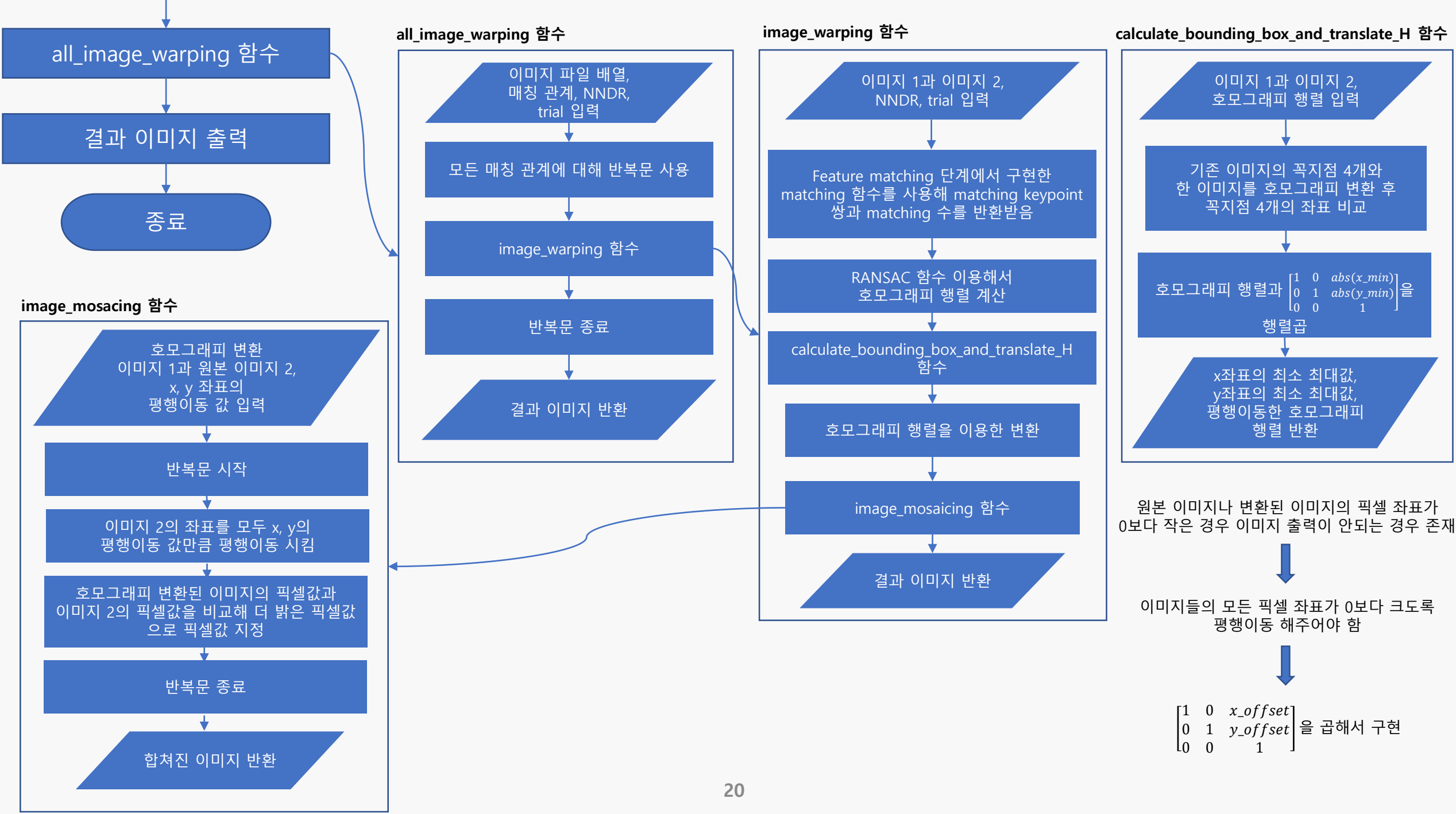


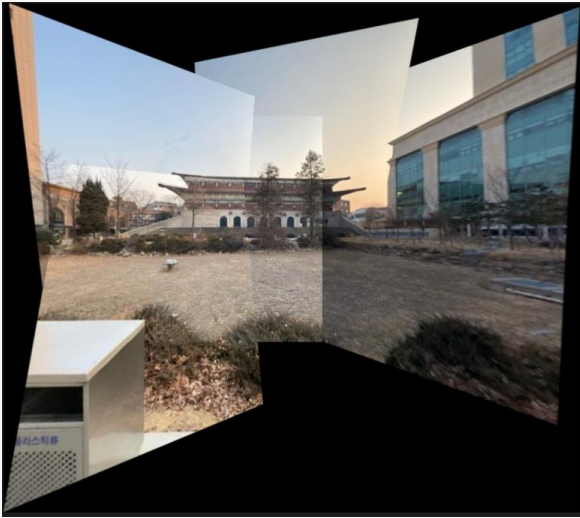
## compute\_homography 함수



$$\begin{bmatrix} 0 & 0 & 0 & -x_1 & -y_1 & -1 & y'_1x_1 & y'_1y_1 & y'_1 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1x_1 & -x'_1y_1 & -x'_1 \\ 0 & 0 & 0 & -x_2 & -y_2 & -1 & y'_2x_2 & y'_2y_2 & y'_2 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2x_2 & -x'_2y_2 & -x'_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

호모그래피 행렬

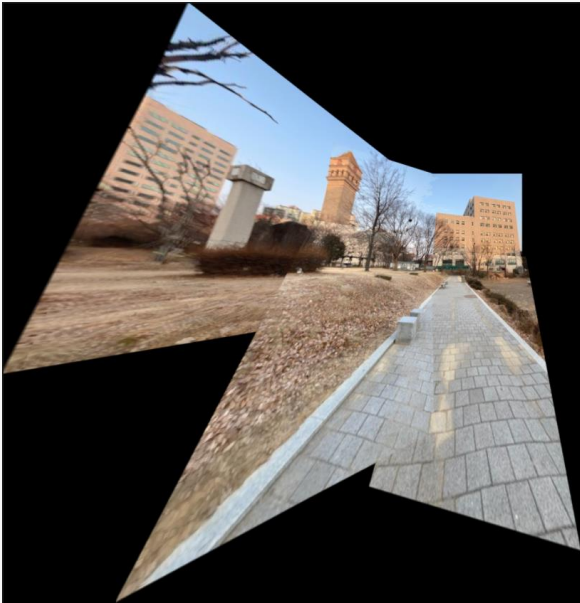




## 비교적 잘 된 stitching 결과

비교적 stitching이 잘 되었지만, AI 센터 건물의 모서리 부분이 살짝 손상되었으며 각 사진 별로 밝기 값이 달라 부자연스러운 모습을 볼 수 있다.

→ blending 등의 기법을 이용해서 해결해야 함



## 비교적 아쉬운 stitching 결과

시계탑이 수직이 아닌 약간 휘어서 나왔으며 특히, 사진 촬영자와 가까이 있던 인도가 비정상적으로 크게 나왔다.

이는 3차원 세계의 정보들을 2차원으로 담은 사진을 단순 평면으로 생각하고 호모그래피 변환을 수행한 결과 생긴 현상으로 생각한다.

THANK YOU –  
**감사합니다.**