Image Filtering (transformation)

그림 위치 옮김

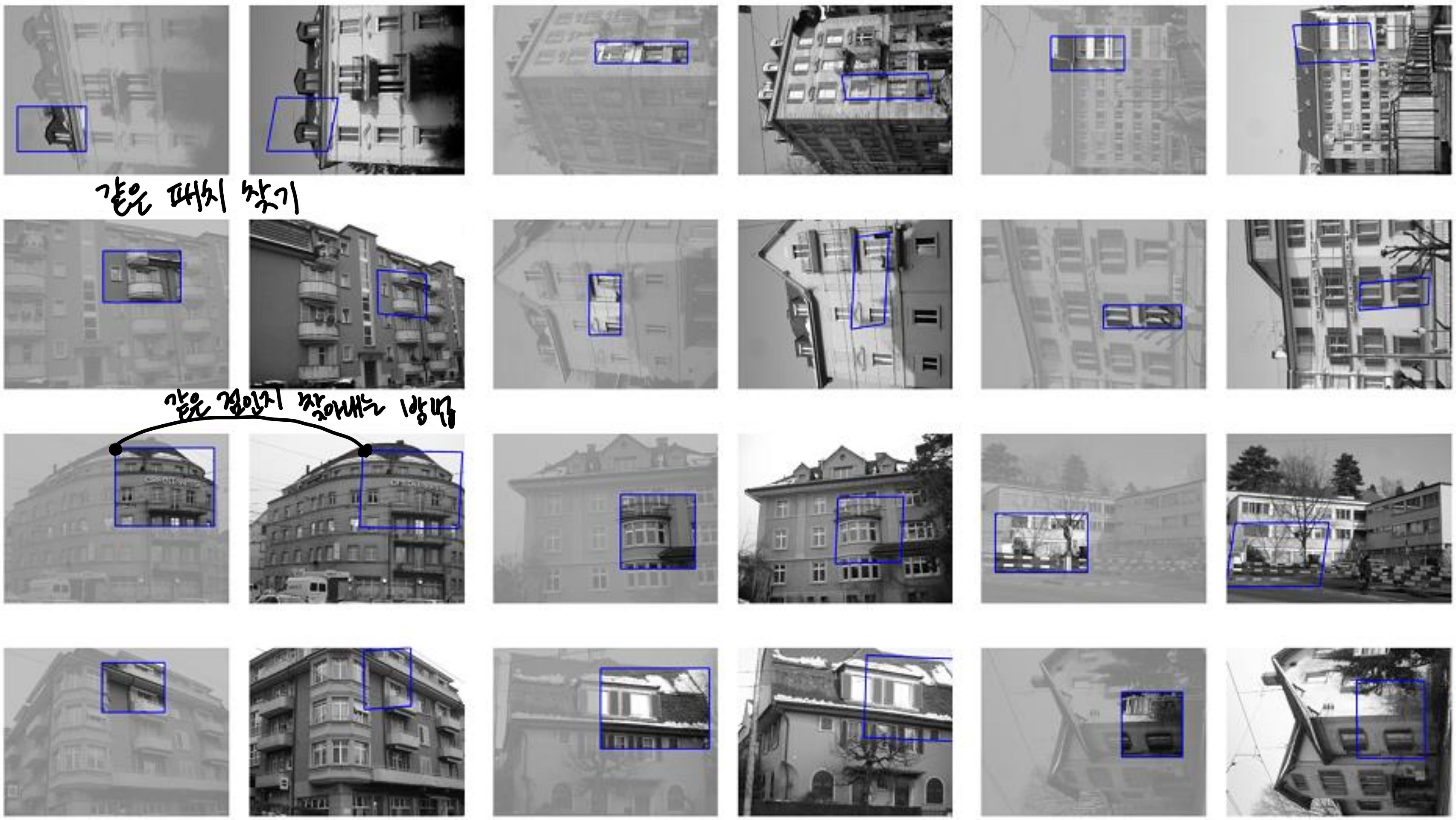
Image_filtering.py

Tae Hyun Kim

Application: Matching

- Affine template matching

같은 그림. 촬영 시점 다름

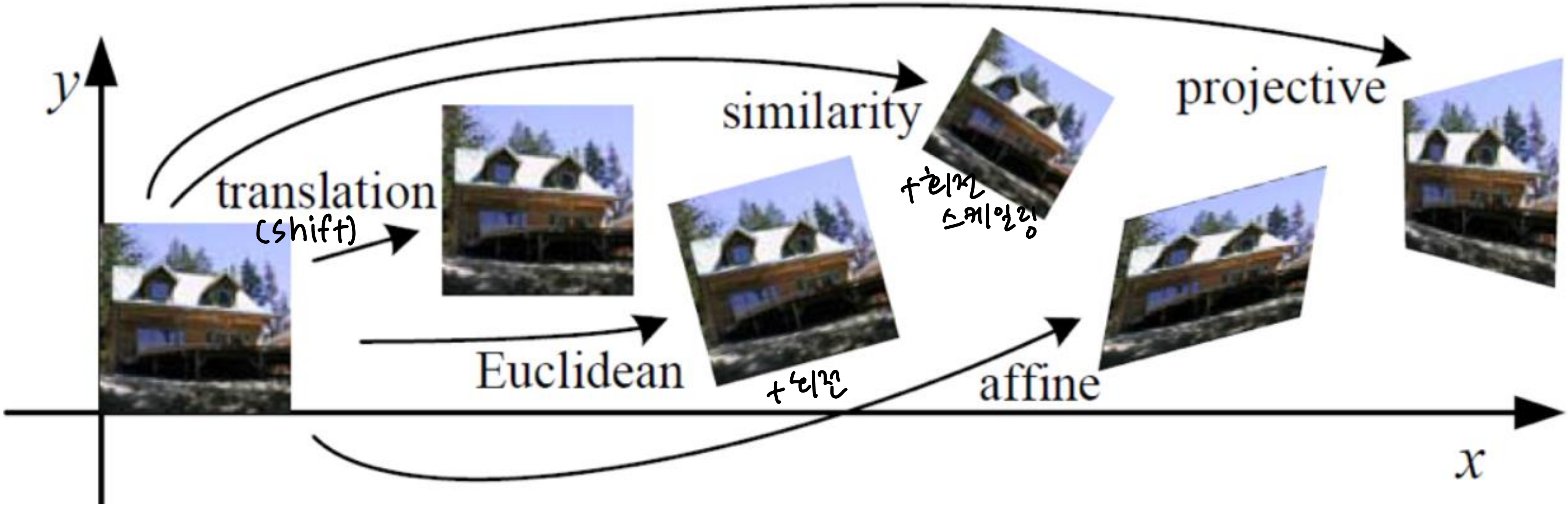


같은 패치 찾기

같은 점인지 찾아내는 방법

Affine template matching results, Koreman et al. CVPR'13

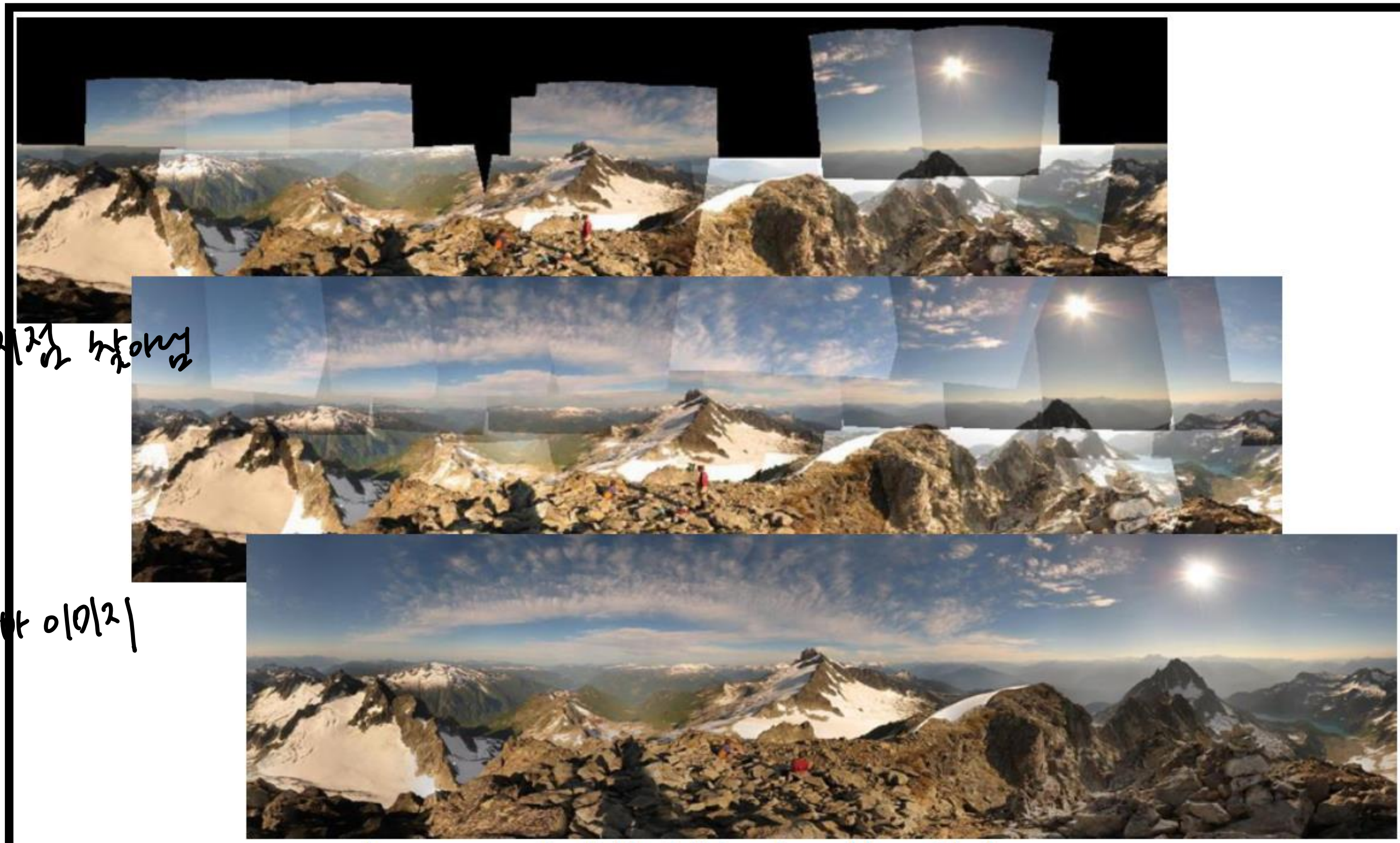
2D Transformations



Application: Stitching

파노라마 이미지를 만드는데 필요한 기술

- Basic tool for image registration and alignment



overlap 지점 찾아냄

→ align

→ 후처리

→ 파노라마 이미지

Image stitching result, Brown and Lowe. IJCV'07

2D Transformations

- Translation, Rotation, Scale, Similarity, Affine, Homography, Projective

Translation

shift

Rotation

Scale

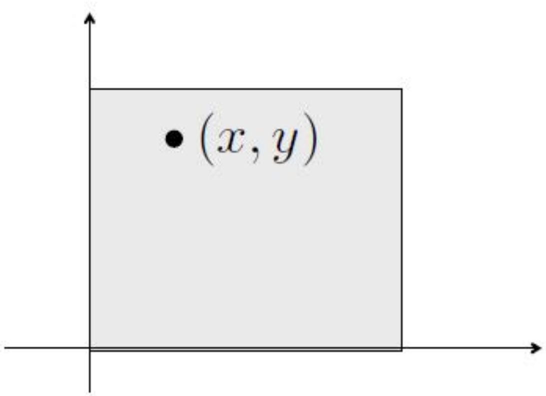
Similarity

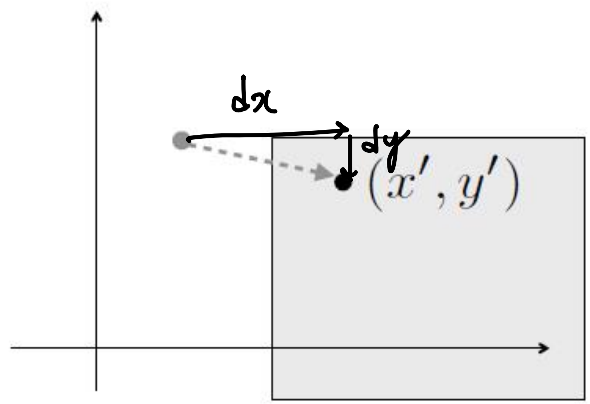
회전 + 스케일

Affine

Homography

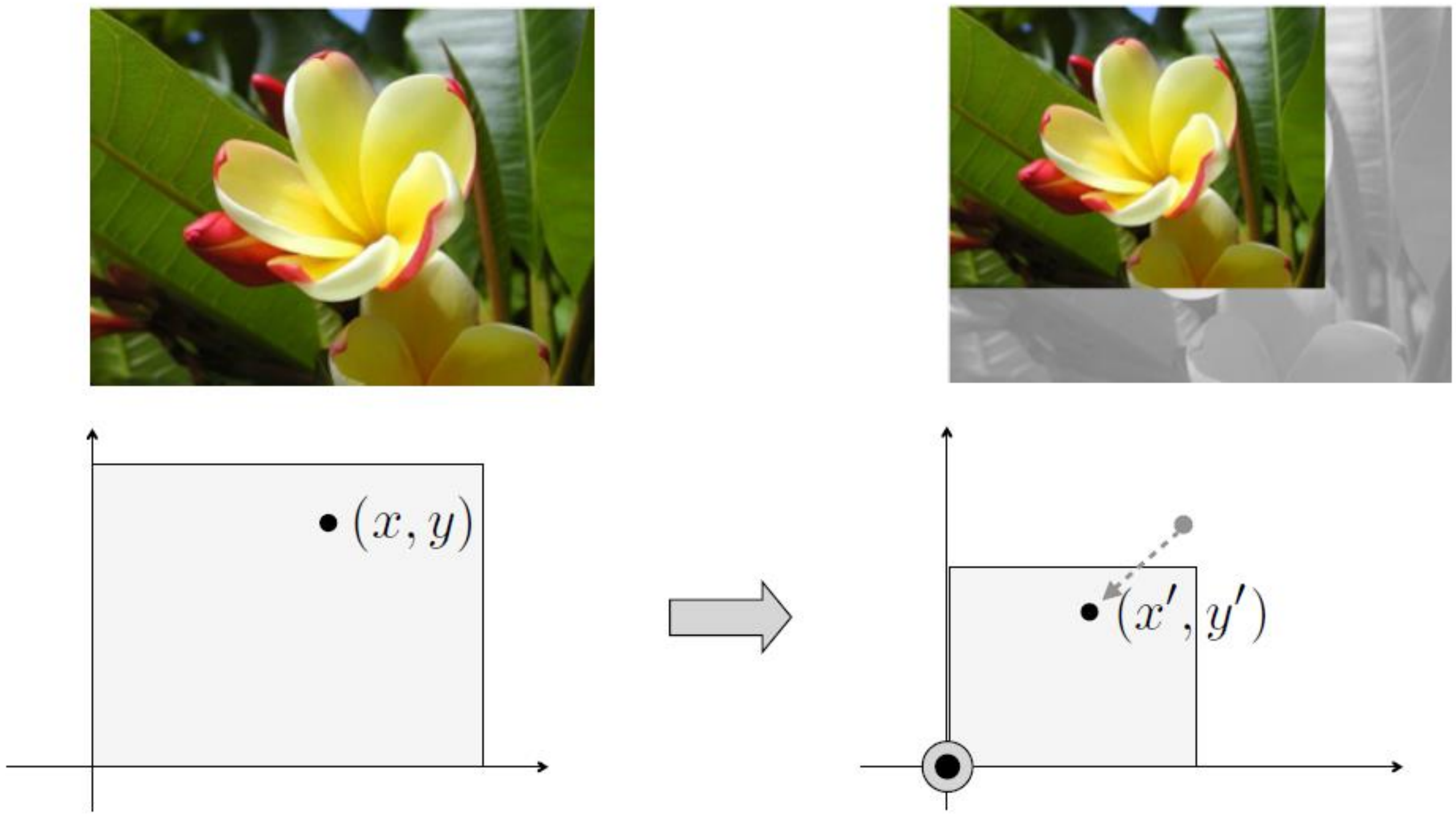
Translation





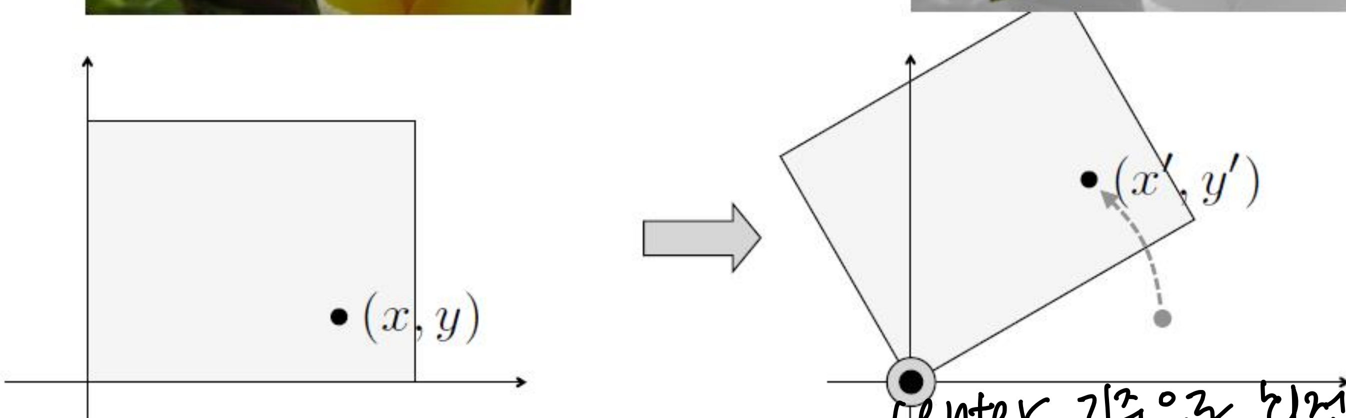
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + dx \\ y + dy \end{pmatrix}$$

Scale



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} sx \\ sy \end{pmatrix}$$

Rotation

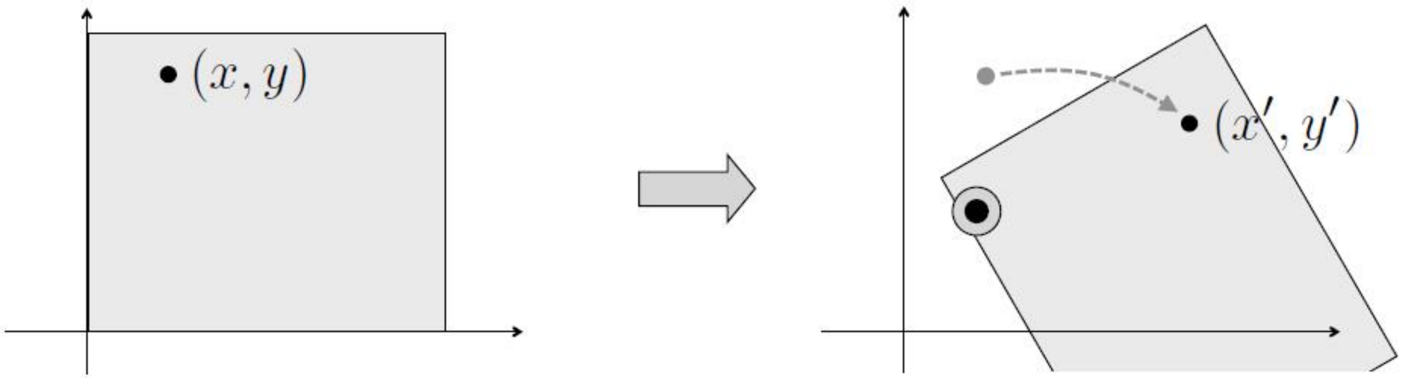


center 기준으로 회전

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{pmatrix}$$

Similarity Transform (if no scale change → $\underbrace{\text{Euclidean}}_{\text{shift, rotation}}$)

scaling, rotation, translation 다 포함됨



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} sx\cos\theta - sy\sin\theta + dx \\ sx\sin\theta + sy\cos\theta + dy \end{pmatrix}$$

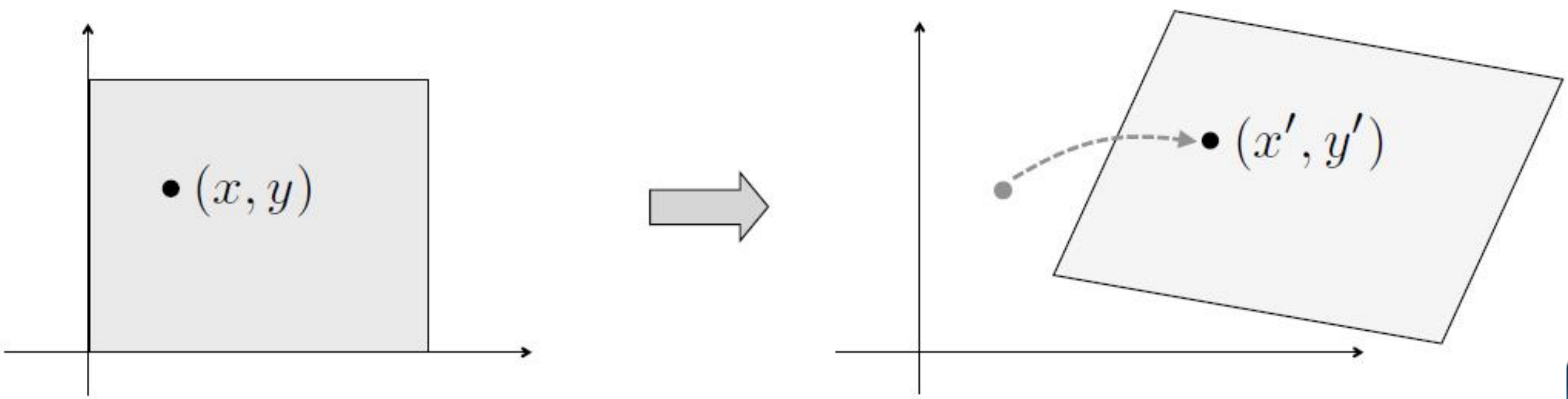
파라미터 4개 필요

Affine Transform

- Preserve points, lines, planes, & parallel lines remain parallel

각 보존X

평행한 수직선은 보존됨



$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_0 x + a_1 y + a_2 \\ a_3 x + a_4 y + a_5 \end{pmatrix}$$

파라미터 6개 필요

Summary: 2D Transformations

자유도

- Summary (DOF=?)

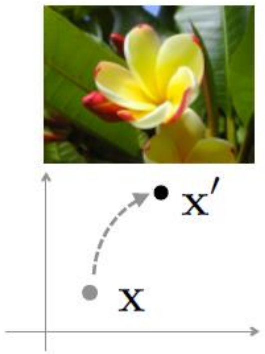
필요한 파라미터 개수

Translation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + dx \\ y + dy \end{pmatrix}$$

DOF=2

$$\mathbf{x}' = \mathbf{x} + \mathbf{d}$$



Scale

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} sx \\ sy \end{pmatrix}$$

DoF=1

$$\mathbf{x}' = s\mathbf{x}$$

Rotation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{pmatrix}$$

DoF=1

$$\mathbf{x}' = R\,\mathbf{x} = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}\mathbf{x}$$

Similarity / Euclidean

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} sx\cos\theta - sy\sin\theta + dx \\ sx\sin\theta + sy\cos\theta + dy \end{pmatrix}$$

$$\mathbf{x}' = sR\,\mathbf{x} + \mathbf{d}$$
$$\mathbf{x}' = R\,\mathbf{x} + \mathbf{d}$$

자유도 DOF가 작을수록 변환 결과가 제한적

DoF=4

Affine

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_0x + a_1y + a_2 \\ a_3x + a_4y + a_5 \end{pmatrix}$$

$$\mathbf{x}' = A\,\mathbf{x} + \mathbf{d} = \begin{pmatrix} a_0 & a_1 \\ a_3 & a_4 \end{pmatrix}\mathbf{x} + \begin{pmatrix} a_2 \\ a_5 \end{pmatrix}$$

DoF=6

Euclidean (or Cartesian) Coordinate

- Computational cost
  - What if we transform five times
    - $\begin{pmatrix} x' \\ y' \end{pmatrix} = A(A(A(A(A\begin{pmatrix} x \\ y \end{pmatrix} + B)+B)+B)+B)+B$
    - 5 multiplications, 5 additions
      - Huge computations with millions of points

좀 더 쉽게 할수있는 좌표계

Homogeneous Coordinate

- Representation
  - Change 2D into 3D by adding 1's at the end

affine transform

- $$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \mathrm{H} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- Computational cost
  - What if we transform five times

- $$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \mathrm{HHHHH} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

연산 간단해짐

- 1 multiplication only! (HHHHH is known or precomputable)

1,000,000 pixel → 1,000,000 만 연산

2D Transformations using Homogeneous Coordinates

- DOF=?

Translation: 유클리디안 호모지니어스

$$\mathbf{x}' = \mathbf{x} + \mathbf{t} \qquad \mathbf{x}' = \begin{pmatrix} I & \mathbf{t} \end{pmatrix} \bar{\mathbf{x}}$$

Euclidean, rigid body (rotation + translation):

$$\mathbf{x}' = R\,\mathbf{x} + \mathbf{t} \qquad \mathbf{x}' = \begin{pmatrix} R & \mathbf{t} \end{pmatrix} \bar{\mathbf{x}} \qquad R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

Scaled rotation (similarity):

$$\mathbf{x}' = sR\,\mathbf{x} + \mathbf{t} \qquad \mathbf{x}' = \begin{pmatrix} sR & \mathbf{t} \end{pmatrix} \bar{\mathbf{x}}$$

Affine:

$$\mathbf{x}' = A\,\bar{\mathbf{x}} \qquad A = \begin{pmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{pmatrix}$$

Projective (homography):

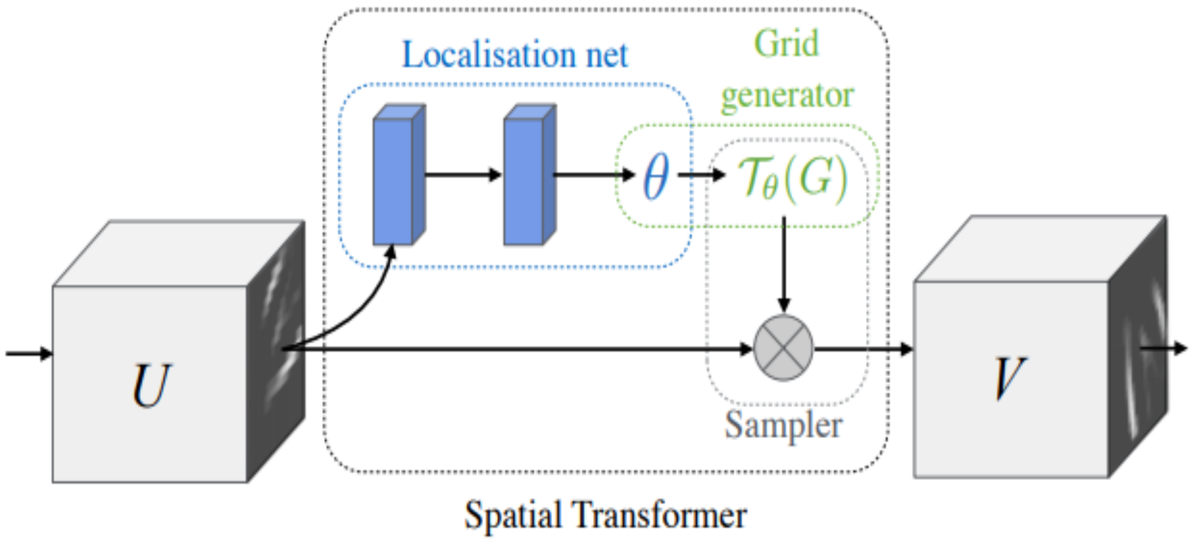
$$\bar{\mathbf{x}}' = H\,\bar{\mathbf{x}} \qquad H = \begin{pmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{pmatrix}$$

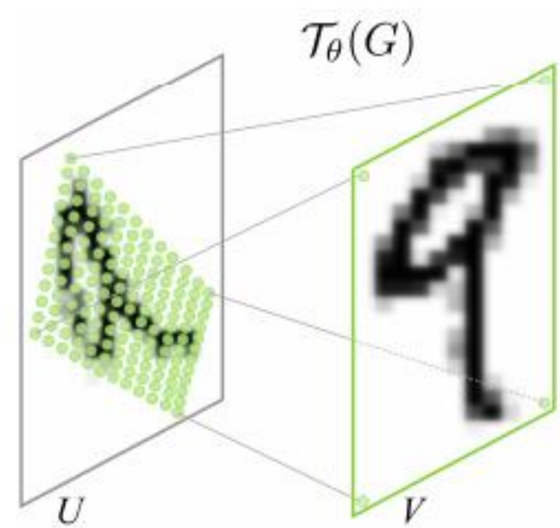
How can we compute transformation parameters?

- Spatial Transformer Networks

그림 두장 사이에 적용된
transform parameter return

NN가 파라미터 찾아줄수있음
어떻게 변환 했는지





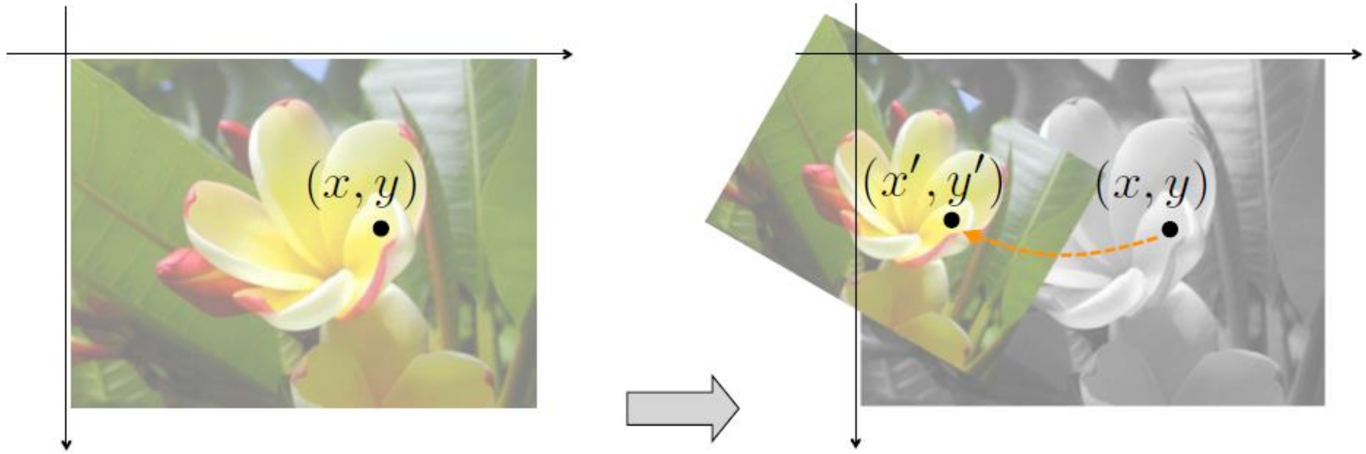
Jaderberg et al. NIPS'15

Image Warping

영상을 찌그러트리는 기술

- Easy to transfer a point with a given transformation

- How can we obtain a transformed (warped) image?

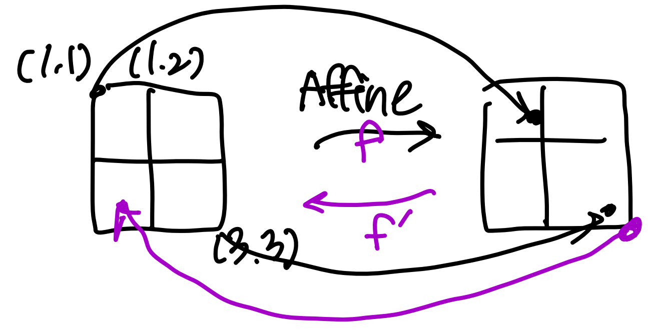


$$\mathbf{x}' = f(\mathbf{x})$$

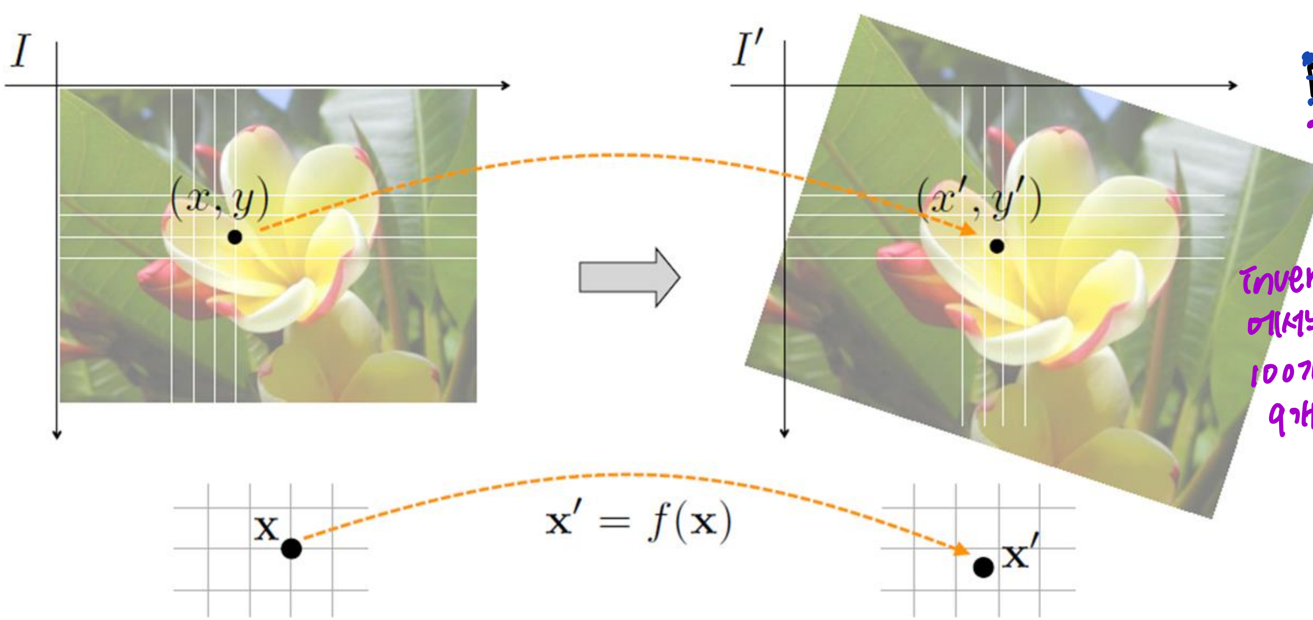
$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_0 & a_1 & a_2 \\ a_3 & a_4 & a_5 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

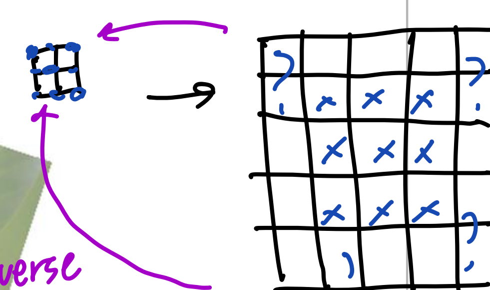
Forward Warping

- Pros
  - Easy to calculate target points
- Cons
  - Cracks & holes, floating point in the source image
    - Need interpolation



스케일링 연산에 문제 AA



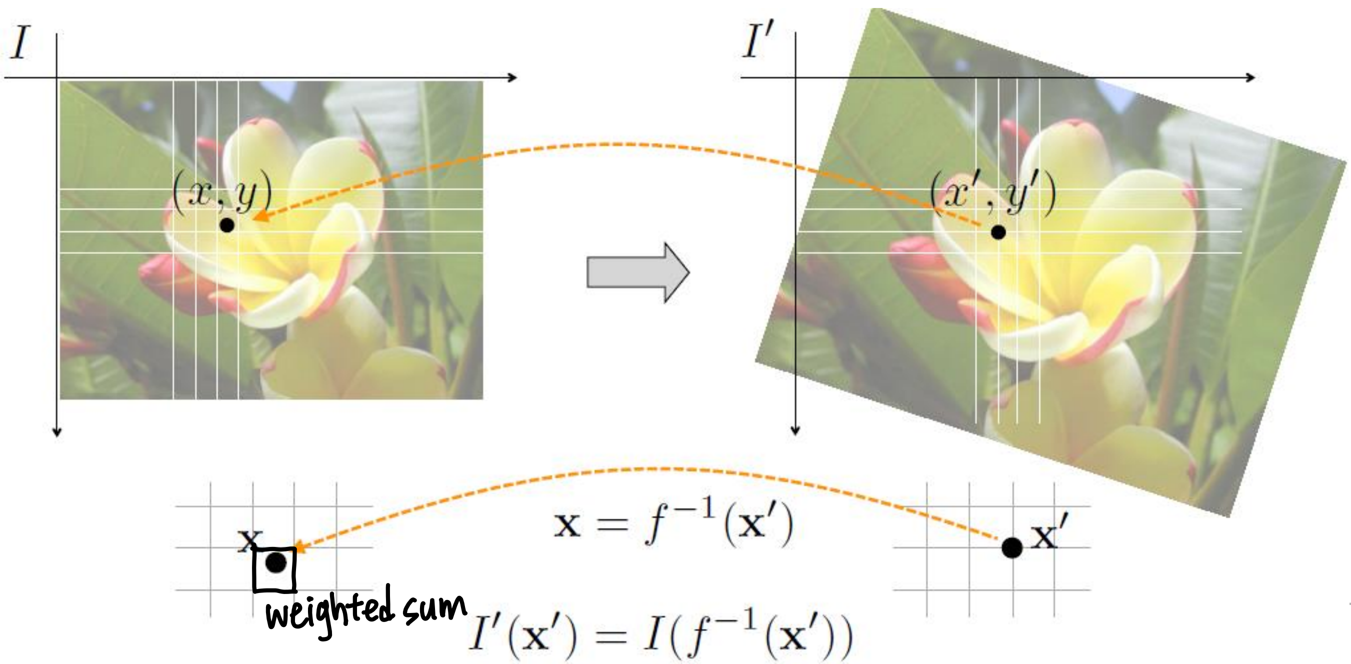


Inverse 에서는 괜찮음
100개를 9개로 다 보내기 가능

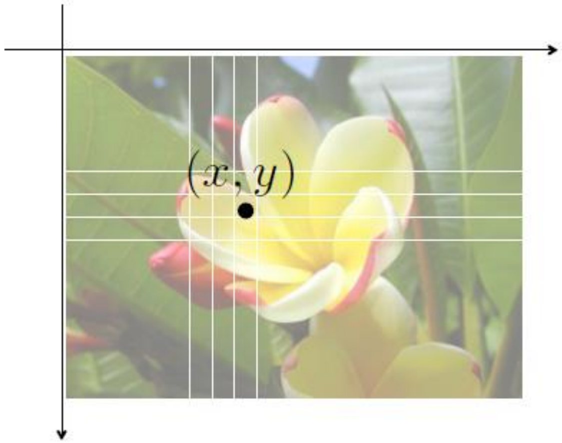
정보없는 hole 발생

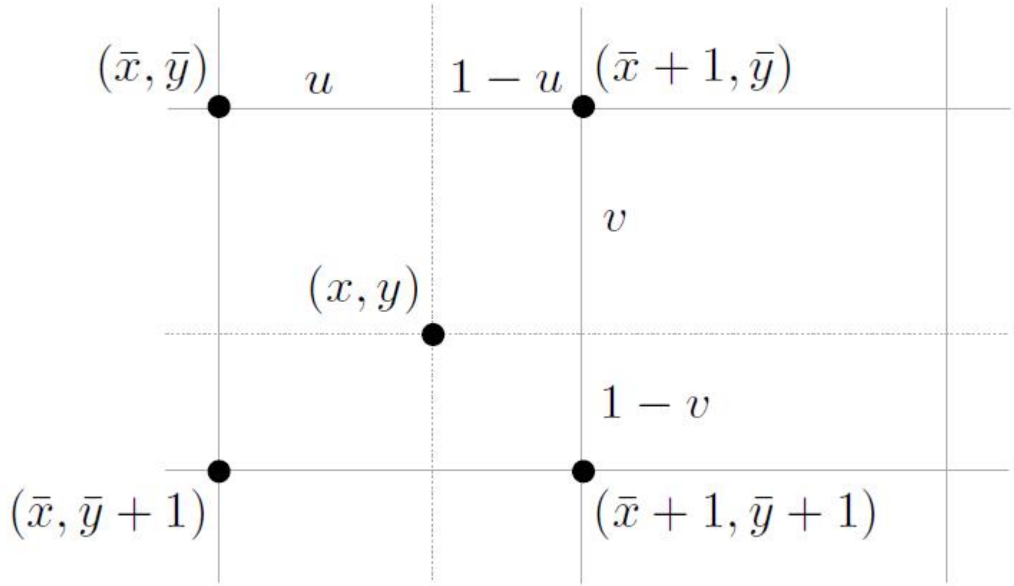
Inverse Warping

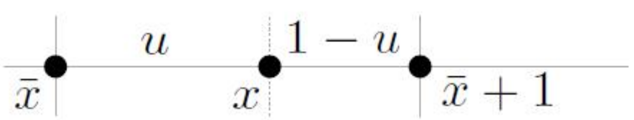
- Pros
  - Avoid cracks and holes
- Cons
  - Floating point in the source image
    - Need interpolation



Linear Interpolation







$$f(x) \quad = \quad (1-u)\, f(\bar{x}) + u\, f(\bar{x}+1)$$

$$\begin{aligned} I(x, y) \quad = \quad & (1-u)\,(1-v)\; I(\bar{x}, \bar{y}) + \\ & u\,(1-v)\; I(\bar{x}+1, \bar{y}) + \\ & (1-u)\, v\; I(\bar{x}, \bar{y}+1) + \\ & u\, v\; I(\bar{x}+1, \bar{y}+1) \end{aligned}$$

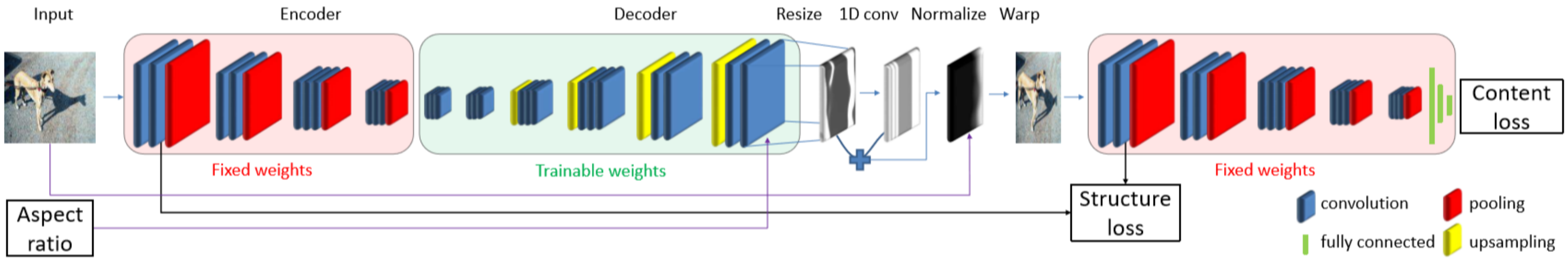
Recent Warping Technique

- Learning-based approach

Original image

Warping results
WSSDCNN, Cho et al. ICCV'17

- Network architecture



Thank you!