Feature Descriptor and Deep Image Homography

Image and Video Pattern Recognition Lab.

2018.01.19

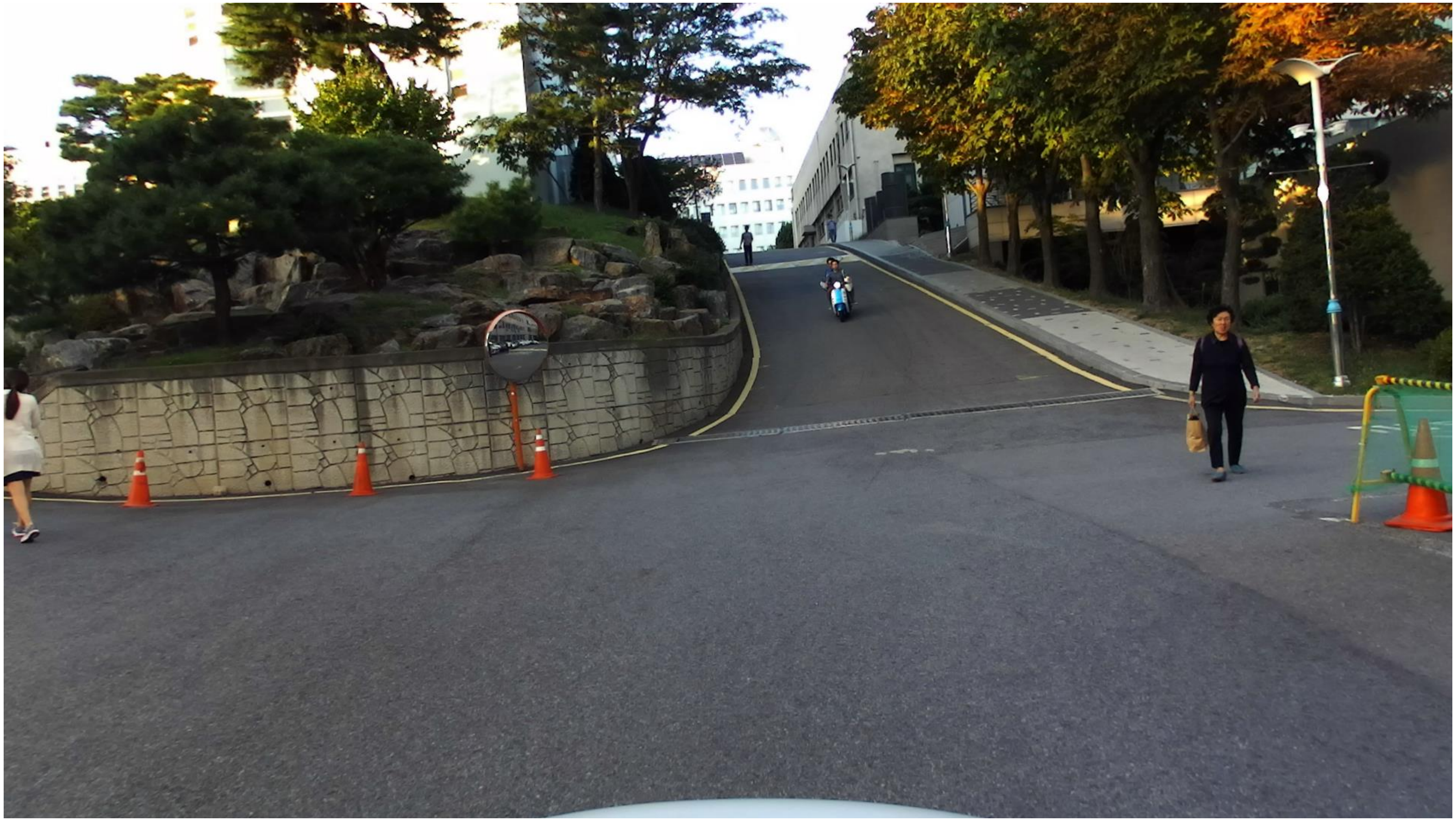
황상원/2016314162

Outline

- What is Homography?
- Conventional Homography Estimation
  - Conventional Method
  - Feature / Descriptor
- Deep Image Homography Estimation
- Conclusion

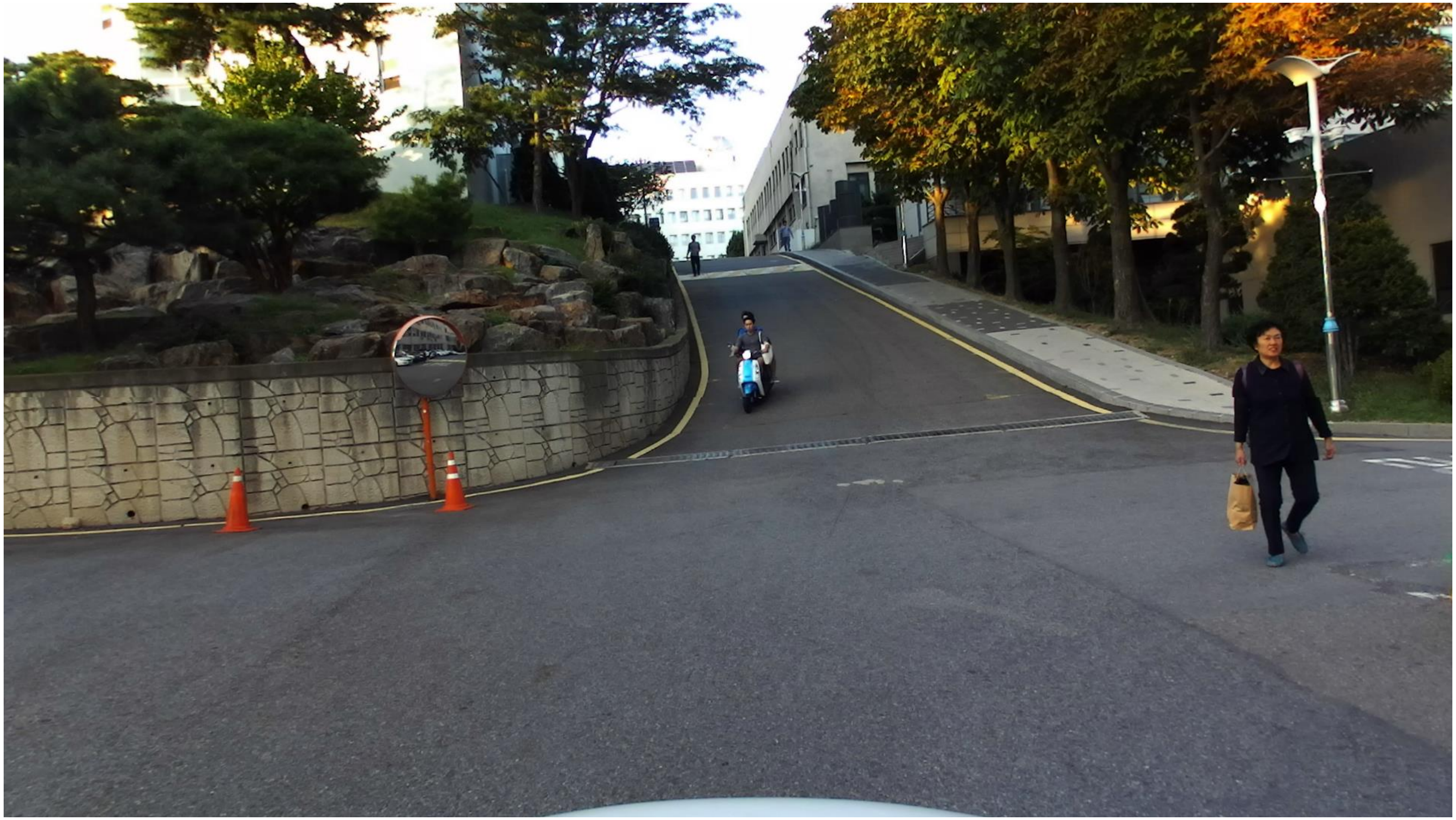
What is Homography??

What is Homography



1 Frame DB

What is Homography



2 Frame DB

Rotating / Translating Camera, Planar world

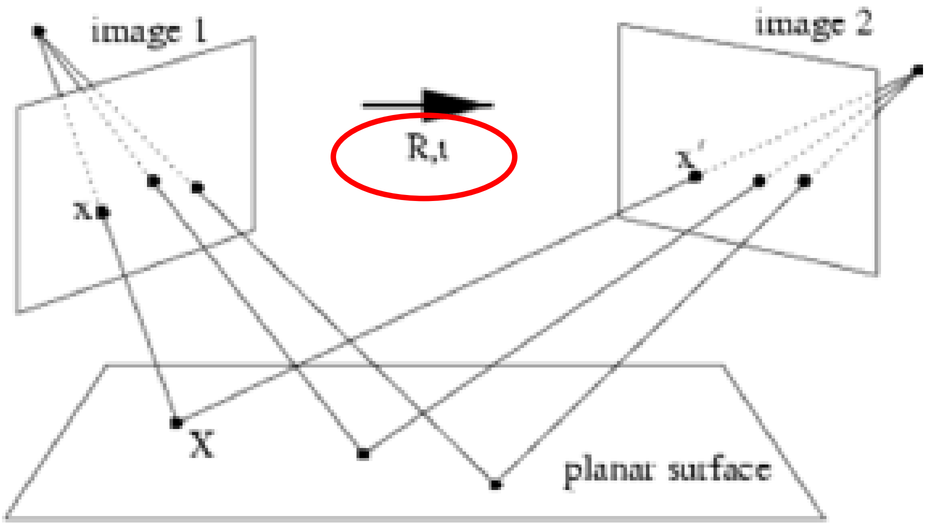



image1


image2

Relationship between two pair of images

Definition: Homography

$$\begin{pmatrix} x'_1 \\ x'_2 \\ x'_3 \end{pmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} \quad \text{or} \quad \mathrm{x}' = \mathbf{H}\,\mathrm{x}$$

8DOF

Homography=projective transformation=projectivity=collineation

❖ Homogeneous Coordinate

- Converting **to** homogeneous coordinate
  - Add one more coordinate

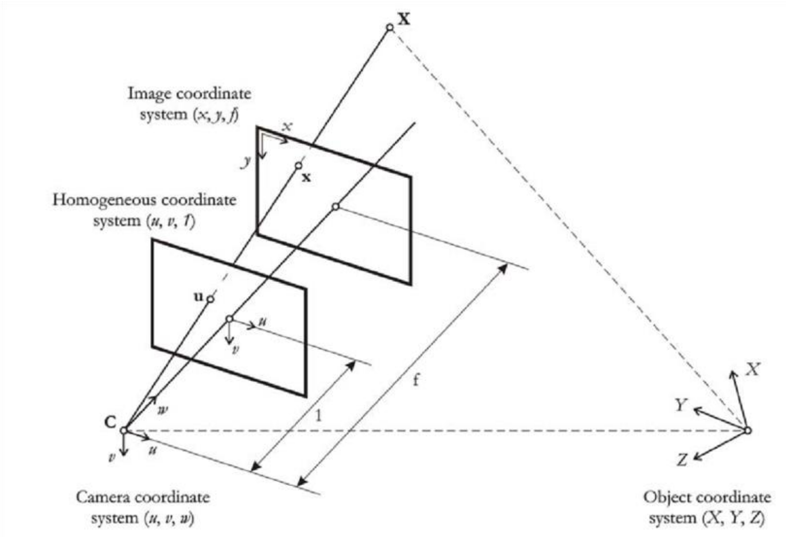
$$(x, y) \Rightarrow \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous image coordinate (2D)

- Converting **from** homogeneous coordinate

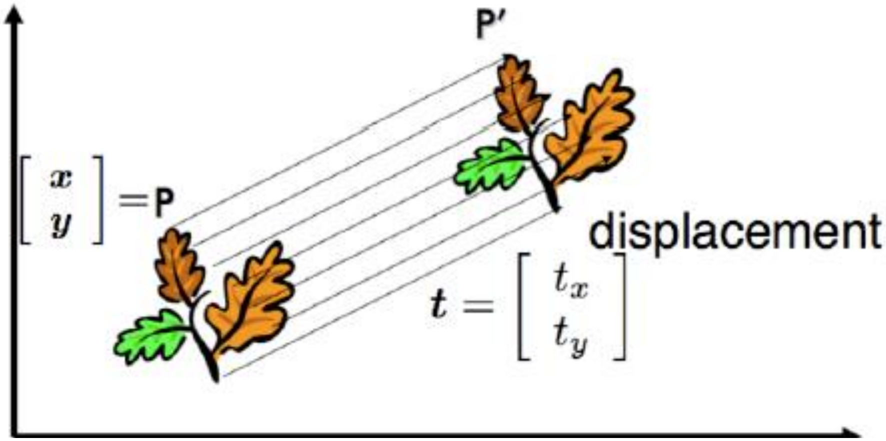
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} \Rightarrow (x/w, y/w)$$

Homogeneous image coordinate (2D)



X와 u점은 같은 곳으로 Projection된다.
(같지만 다르고, 다르지만 같은거)

❖ Translation Transformation

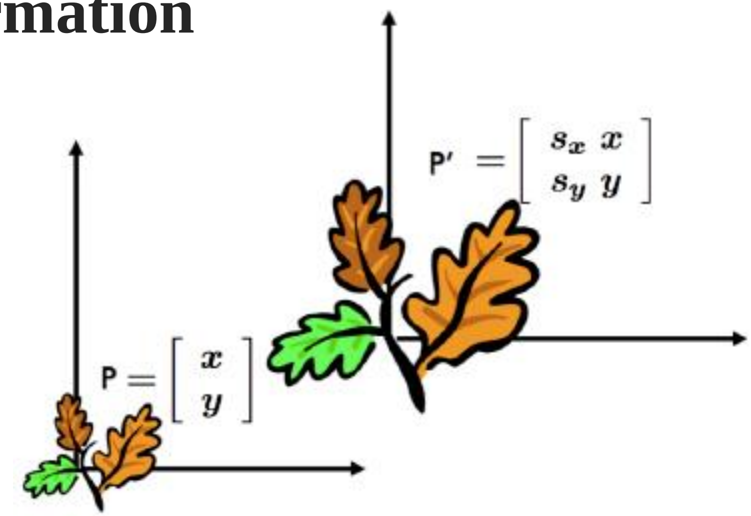


Homogeneous coordinates

$$P' = P + \boldsymbol{t} = \begin{bmatrix} x + t_x \\ y + t_y \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\Rightarrow \quad P' \rightarrow \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}}_{\text{translation matrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

❖ Scaling Transformation

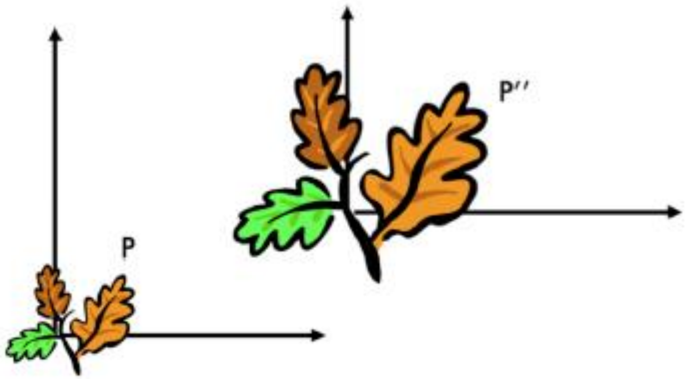


Homogeneous coordinates

$$\begin{bmatrix} s_x\ x \\ s_y\ y \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} s_x\ x \\ s_y\ y \\ 1 \end{bmatrix} = \overset{S}{\underbrace{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\text{scaling matrix}}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

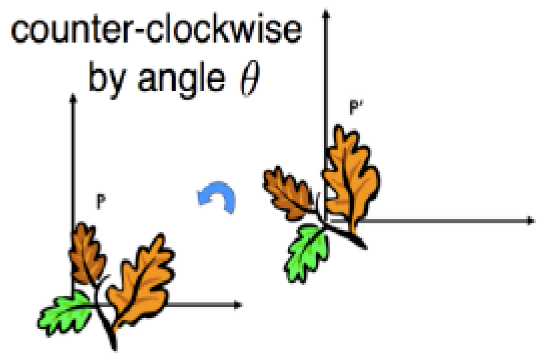
❖ Translation + Scaling Transformation

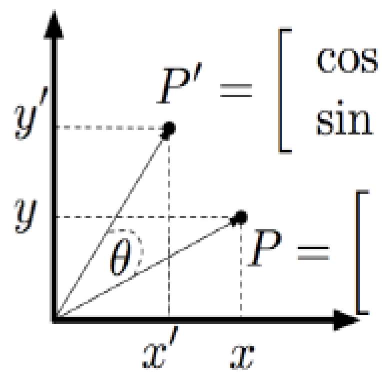


Homogeneous coordinates

$$\begin{bmatrix} x'' \\ y'' \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{\begin{bmatrix} s_x & 0 & t_x \\ 0 & s_y & t_y \\ 0 & 0 & 1 \end{bmatrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

❖ Rotation Transformation





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

$$P = \begin{bmatrix} x \\ y \end{bmatrix}$$

Homogeneous coordinates

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\overset{R}{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{rotation matrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

❖ Rotation + Translation + Scaling (Similarity Transform)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \underbrace{\overset{R}{\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{rotation matrix}} \underbrace{\overset{\boldsymbol{t}}{\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{translation matrix}} \underbrace{\overset{S}{\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}}}_{\text{scaling matrix}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} R\,S & \boldsymbol{t} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

What is Homography

❖ Aspect Ratio (가로 대 세로의 비율)

$$\begin{bmatrix} a & 0 & 0 \\ 0 & \frac{1}{a} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

❖ Shear (층 밀리기?)

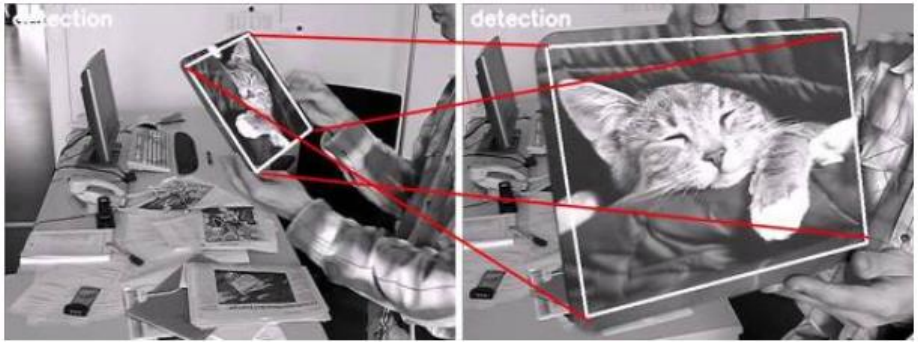
$$\begin{bmatrix} 1 & a & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

❖ **Affine Transform (Aspect ratio + Rotation + Translation + scale)**

$$\begin{bmatrix} a & b & c \\ d & e & f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

❖ **Homography (Projective Transform)**

8 DOF (∵Homogeneous)



$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$$

$$x' = u/w$$

$$y' = v/w$$

Conventional Homography Estimation

❖ Conventional Method

- Problem:

8 DOF (∵Homogeneous)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} \sim \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

8 DOF 이니까.
4개의 대응점(x,y)을 알면,
H를 구할 수 있겠군
→ 4-point homography parameterization

- Equations:

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Conventional Homography Estimation

❖ Conventional Method

- 1st Method **Set $\mathbf{h_{33} = 1}$**

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1}$$

- 2nd Method: Impose unit vector constraint (DLT algorithm)

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Constraint: $$h_{11}^2 + h_{12}^2 + h_{13}^2 + h_{21}^2 + h_{22}^2 + h_{23}^2 + h_{31}^2 + h_{32}^2 + h_{33}^2 = 1$$

❖ Conventional Method

- 1st Method **Set $\mathbf{h_{33} = 1}$**

$$x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + 1}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + 1}$$

Multiplying through by denominator

$$(h_{31}x + h_{32}y + 1)x' = h_{11}x + h_{12}y + h_{13}$$

$$(h_{31}x + h_{32}y + 1)y' = h_{21}x + h_{22}y + h_{23}$$

Rearrange

$$h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' = x'$$

$$h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' = y'$$

❖ Conventional Method

- 1^{st} Method **Set $h_{33} = 1$**

$$
\begin{array}{l}
\\
\textbf{Point 1} \\
\\
\textbf{Point 2} \\
\\
\textbf{Point 3} \\
\\
\textbf{Point 4} \\
\\
\end{array}
\overset{\textbf{2N x 8}}{\begin{bmatrix}
x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 \\
0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 \\
x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 \\
0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 \\
x_3 & y_3 & 1 & 0 & 0 & 0 & -x_3x'_3 & -y_3x'_3 \\
0 & 0 & 0 & x_3 & y_3 & 1 & -x_3y'_3 & -y_3y'_3 \\
x_4 & y_4 & 1 & 0 & 0 & 0 & -x_4x'_4 & -y_4x'_4 \\
0 & 0 & 0 & x_4 & y_4 & 1 & -x_4y'_4 & -y_4y'_4
\end{bmatrix}}
\overset{\textbf{8 x 1}}{\begin{bmatrix}
h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32}
\end{bmatrix}}
=
\overset{\textbf{2N x 1}}{\begin{bmatrix}
x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \\ x'_4 \\ y'_4
\end{bmatrix}}
$$

additional points $\vdots$ $\vdots$

❖ Conventional Method

- 1^{st} Method : **Set $h_{33} = 1$**

Linear equation problem: $\underset{2N\times 8}{\mathbf{A}}\;\underset{8\times 1}{\mathbf{h}} = \underset{2N\times 1}{\mathbf{b}}$

Solution:

$$\underset{8\times 2N}{\mathbf{A}^T}\;\underset{2N\times 8}{\mathbf{A}}\;\underset{8\times 1}{\mathbf{h}} = \underset{8\times 2N}{\mathbf{A}^T}\;\underset{2N\times 1}{\mathbf{b}}$$

$$\overbrace{(\mathbf{A}^T\mathbf{A})}^{8\times 8}\;\underset{}{\overset{8\times 1}{\mathbf{h}}} = \overbrace{(\mathbf{A}^T\mathbf{b})}^{8\times 1}$$

$$\underline{\mathbf{h} = (\mathbf{A}^T\mathbf{A})^{-1}(\mathbf{A}^T\mathbf{b})}$$

Least Square Solution

❖ Conventional Method

- 1st Method : Set $h_{33} = 1$

Linear equation probl… $\underset{}{\mathbf{A}}\ \overset{8x1}{\mathbf{h}} = \overset{2Nx1}{\mathbf{b}}$

4개의 대응되는 포인트들을 Feature Matching을 통해 완벽하게 구하는 것은 어려움.

→ General Approach가 필요.

❖ Conventional Method

- 2nd Method : Impose unit vector constraint (DLT algorithm)

$$\|\mathbf{h}\| = 1 \qquad x' = \frac{h_{11}x + h_{12}y + h_{13}}{h_{31}x + h_{32}y + h_{33}}$$

$$y' = \frac{h_{21}x + h_{22}y + h_{23}}{h_{31}x + h_{32}y + h_{33}}$$

Multiplying through by denominator

$$(h_{31}x + h_{32}y + h_{33})x' = h_{11}x + h_{12}y + h_{13}$$

$$(h_{31}x + h_{32}y + h_{33})y' = h_{21}x + h_{22}y + h_{23}$$

Rearrange $\quad h_{11}x + h_{12}y + h_{13} - h_{31}xx' - h_{32}yx' - h_{33}x' = 0$

$$h_{21}x + h_{22}y + h_{23} - h_{31}xy' - h_{32}yy' - h_{33}y' = 0$$

Conventional Homography Estimation

❖ Conventional Method

- 2nd Method : Impose unit vector constraint (DLT algorithm)

$$\begin{array}{ccccc}
 & \mathbf{2N \times 9} & \mathbf{9 \times 1} & & \mathbf{2N \times 1} \\
\begin{array}{c}\mathbf{4} \\ \mathbf{P} \\ \mathbf{O} \\ \mathbf{I} \\ \mathbf{N} \\ \mathbf{T} \\ \mathbf{S}\end{array} &
\begin{bmatrix}
x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x_1' & -y_1x_1' & -x_1' \\
0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y_1' & -y_1y_1' & -y_1' \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & & \\
 & & & & & & & &
\end{bmatrix} &
\begin{bmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \\ h_{33} \end{bmatrix} & = &
\begin{bmatrix} 0 \\ 0 \\ \\ \\ \\ \\ \\ \end{bmatrix} \\
\textbf{additional points} & \vdots & & & \vdots
\end{array}$$

❖ Conventional Method

- 2nd Method : Impose unit vector constraint (DLT algorithm)

Homogeneous equations $\underset{2N\times9}{\mathbf{A}}\ \underset{9\times1}{\mathbf{h}} = \underset{2N\times1}{\mathbf{0}}$

- Constraint가 있으므로 h = 0 안됨.
- Exact Solution이 없으므로 $\|Ah\|$를 최소화 하면 됨

Solution: SVD Decomposition

Conventional Homography Estimation

❖ Conventional Method

- 2^{nd} Method : Impose unit vector constraint (DLT algorithm)

Solution:

$$\underset{9\text{x}2N}{A^T}\ \underset{2N\text{x}9}{A}\ \underset{9\text{x}1}{h} = \underset{9\text{x}2N}{A^T}\ \underset{2N\text{x}1}{0}$$

$$\overbrace{(A^T\ A)}^{9\text{x}9}\ \underset{}{\overset{9\text{x}1}{h}} = \overset{9\text{x}1}{0}$$

λ 가 0이면 위 식을 만족하므로..
0에 가까운 작은 Eigen Value를 정한 후 그것에 대응되는 Eigen Vector는 h가 된다…!

$$(A^T A)h = \lambda h$$

❖ Conventional Method

- **한계점…**
  - 4개의 Correspondent Points를 찾는 것이 중요함!!
    - ✓ 여러 개의 Corresponding Points들을 찾아도 어차피 RANSAC을 이용해서 Optimization Problem이 존재…
  - 어떤 Feature Descriptor를 사용해야 하지…?

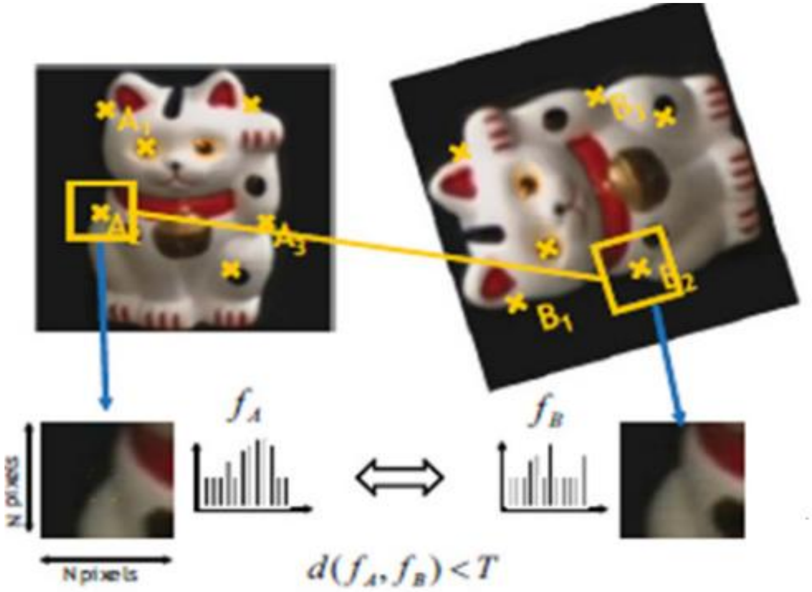
❖ Feature / Feature Descriptor

- Image Feature (ex. Harris corner, Hessian, FAST …etc)

: Edges, Corners (Interest Point), Blobs, Ridges…

- Feature Descriptor (ex. SIFT, LBP, ORB, BRIEF, FREAK …etc)

: Visual description of the patch



1. 노란색 x를 찾는다.
2. 주변 region을 정한다.
3. Region에 descriptor를 이용해서 feature를 표현한다.
4. 표현된 feature 들을 매칭한다.

Conventional Homography Estimation

❖ Feature / Feature Descriptor

- View Point, illumination 에 robust 해야 된다.
- Time efficient 해야 된다.
- Rotation, Translation, Scale 에 invariant 해야 된다… 등등..

각자의 Descriptor 들 마다 장단점이 있음.
Application에 맞게 Descriptor를 정하면 됨.

ORB Descriptor로 매칭한 사진

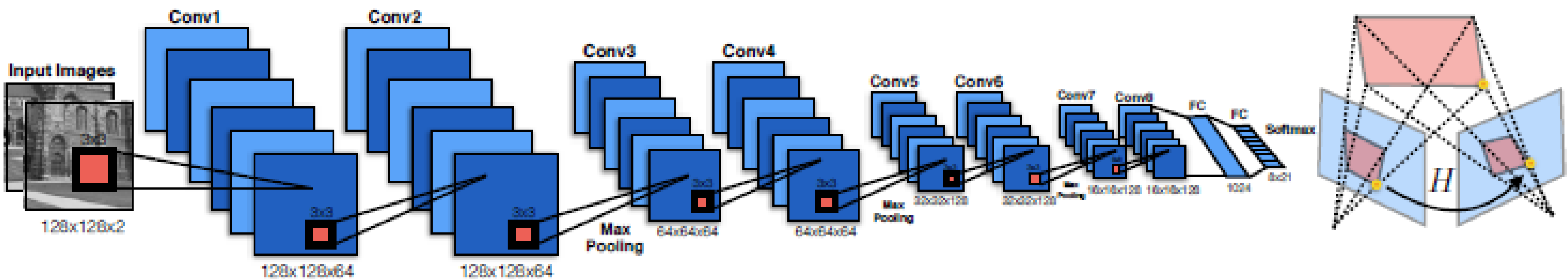
Deep Image Homography Estimation

arXiv 2016.6.13

Deep Image Homography Estimation

❖ Homography Net

- **Network Structure (VGG Style)**



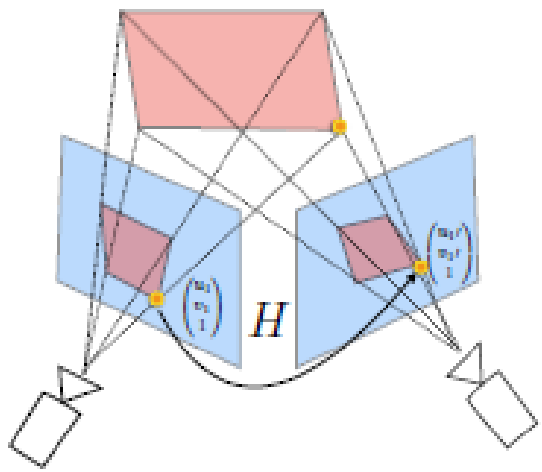
Input : 다른 view에서 찍은 이미지 2장
Output : FC 8개의 값

$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \sim \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

H = 8 DOF

❖ Homography Net

- **Trick-1 : 4-point homography parameterization**



$$H_{4point} = \begin{pmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{pmatrix}$$

1-to-1 mapping

$$H_{matrix} = \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix}$$

Fig. 2: **4-point parameterization.** We use the 4-point parameterization of the homography. There exists a 1-to-1 mapping between the 8-dof "corner offset" matrix and the representation of the homography as a 3x3 matrix.

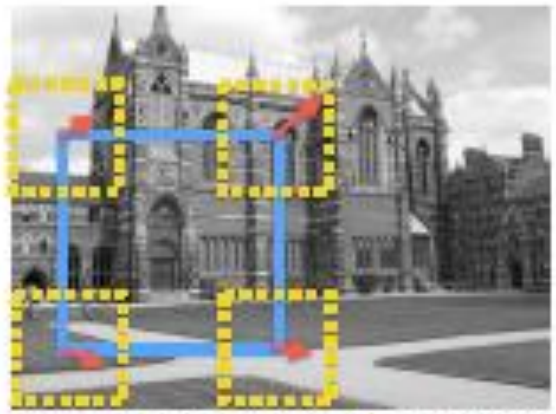
$$\begin{pmatrix} u' \\ v' \\ 1 \end{pmatrix} \sim \begin{pmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{pmatrix} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$

$$H_{4point} = \begin{pmatrix} \Delta u_1 & \Delta v_1 \\ \Delta u_2 & \Delta v_2 \\ \Delta u_3 & \Delta v_3 \\ \Delta u_4 & \Delta v_4 \end{pmatrix}$$

❖ Homography Net

- **Trick-2 : Training Data Generation**

Step 1: Randomly crop at position p. This is Patch A.



Step 2: Randomly perturb four corners of Patch A.

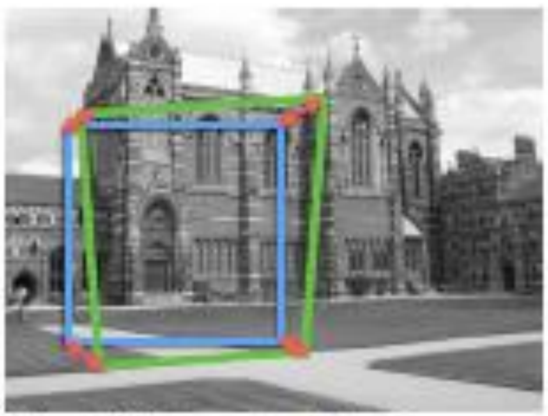
Image 에서 Randomly Crop한다.
이때 Patch의 중심점을 P라 하고
Patch는 A라 한다.

4개의 꼭지점을 perturb (혼란시키다)
→ 4개의 꼭지점에 Noise를 주어 임의로 이동시킨다.

Deep Image Homography Estimation

❖ Homography Net

- **Trick-2 : Training Data Generation**



Step 3: Compute H^{AB} given these correspondences.

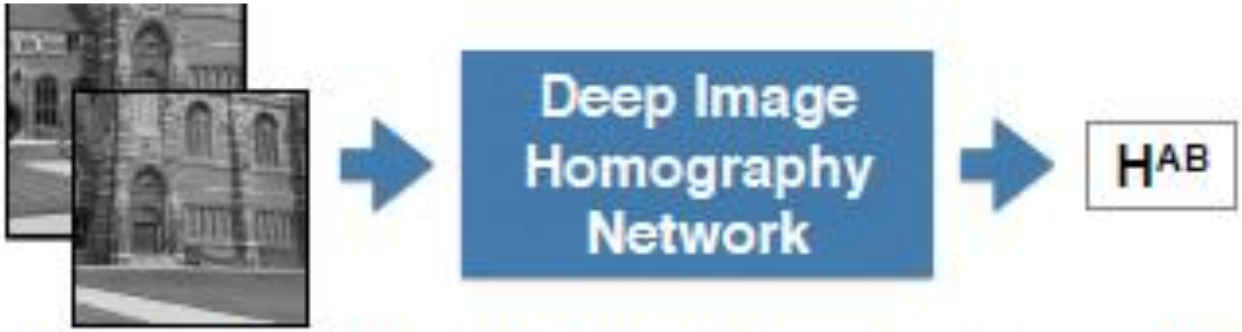
Step 4: Apply $(H^{AB})^{-1} = H^{BA}$ to the image, and crop again at position **p**, this is Patch B.

4개의 꼭지점의 위치를 알면, DLT(Direct Linear Transform)을 이용해 H^{AB} 를 구한다.

$(H^{AB})^{-1} = H^{BA}$를 이용하여 전체적인 이미지를 warping 시키고 P점에서의 Patch B를 구한다.

❖ Homography Net

- **Trick-2 : Training Data Generation**



Step 5: Stack Patch A and Patch B channel-wise and feed into the network. Set $\mathbf{H}^{AB}$ as the target vector.

Patch A,B를 input으로 넣는다.
Step 3에서 구한 H^{AB}가 나오도록 Network를 학습한다.

❖ Homography Net

- Classification vs Regression

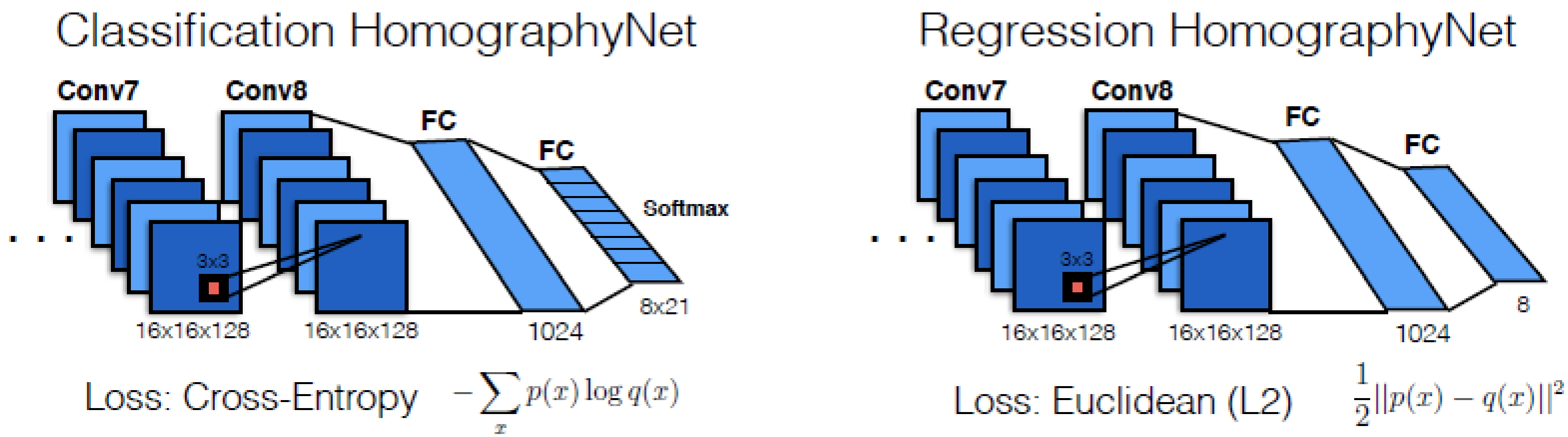
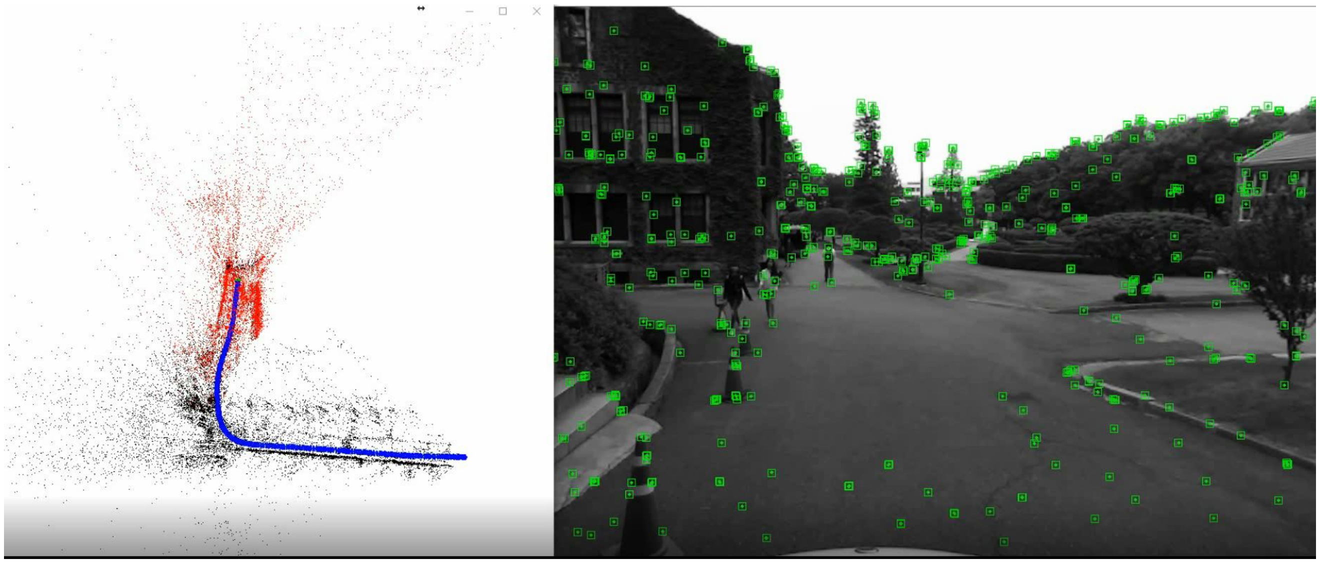


Fig. 4: **Classification HomographyNet vs Regression HomographyNet.** Our VGG-like Network has 8 convolutional layers and two fully connected layers. The final layer is 8x21 for the classification network and 8x1 for the regression network. The 8x21 output can be interpreted as four 21x21 corner distributions. See Section IV for full ConvNet details.

맨 뒤 단에 있는 FC를 8x21(Classification) 이랑 8x1(regression) 두 개다 실험을 해 봄.

❖ Homography Net

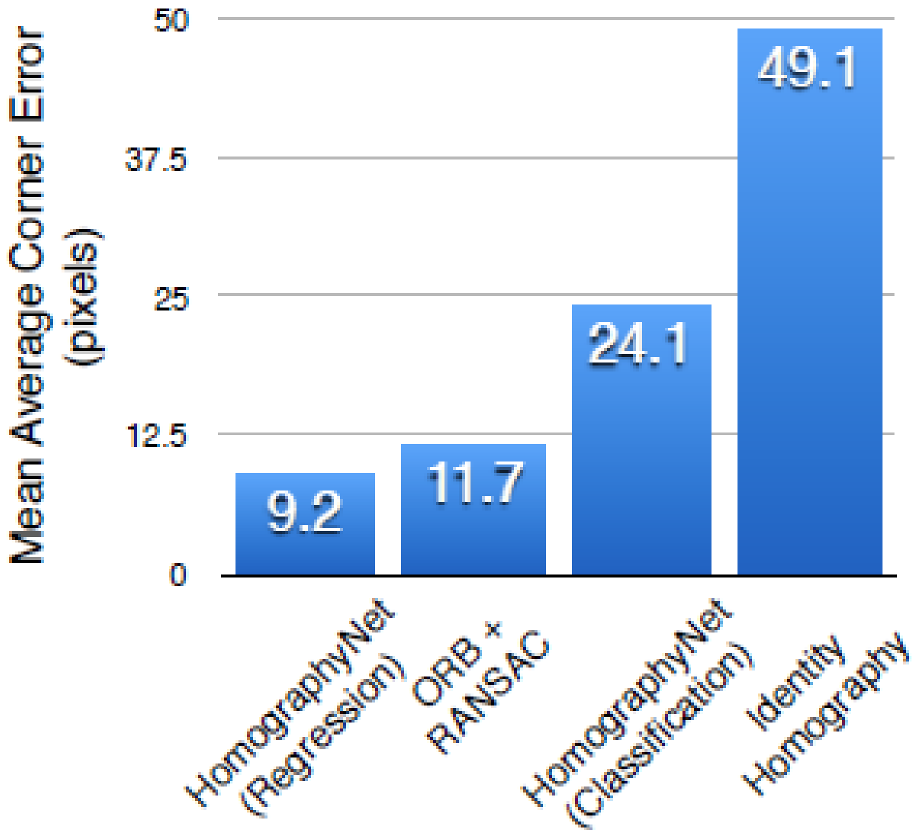
- Experiment Configuration : Tensorflow, Titan X GPU,
- Database : MS-COCO, 500,000 pairs of image patches.
- 비교 대상: ORB descriptor + RANSAC



창조씨앗 DB로 ORB descriptor를 이용한 실험

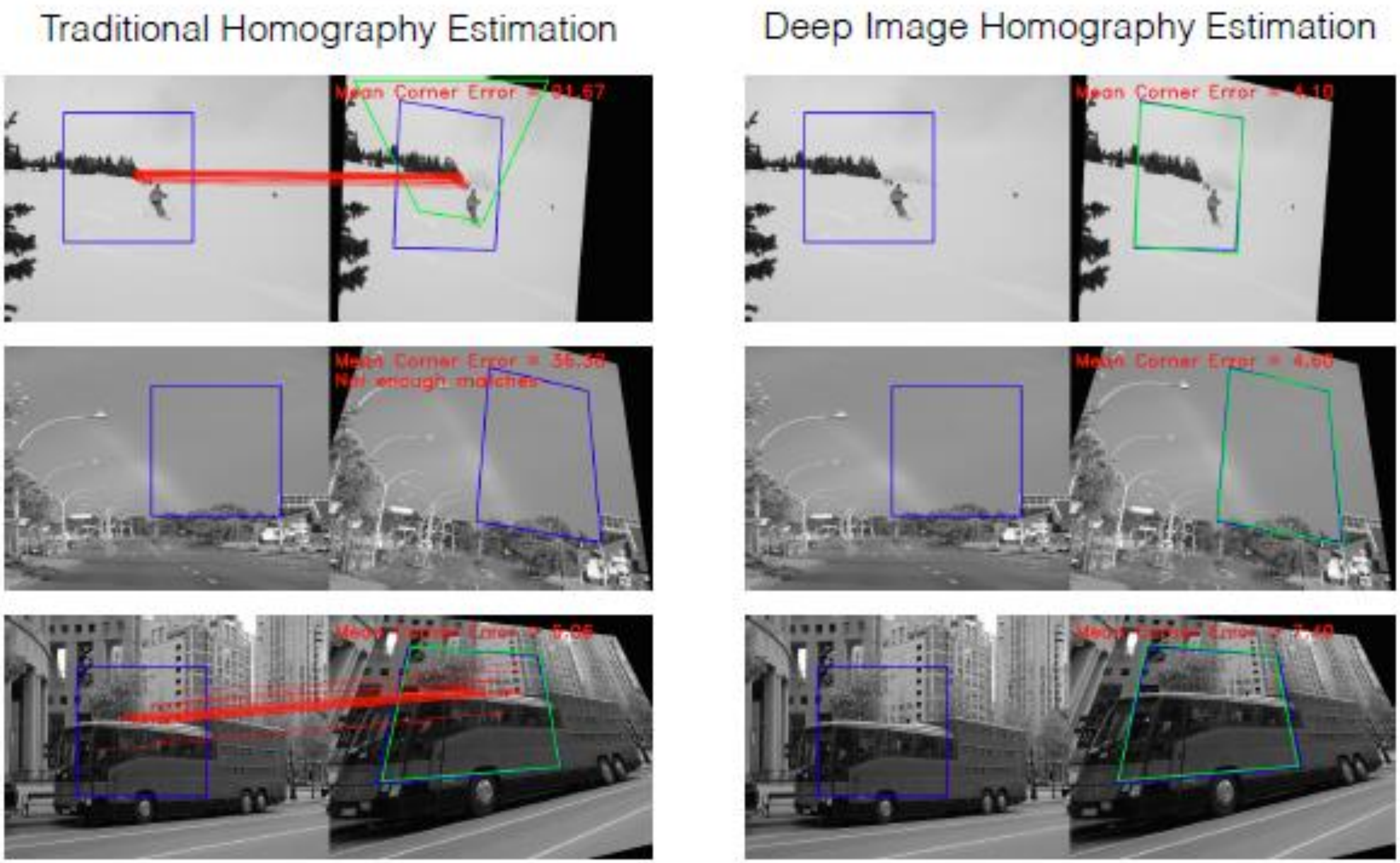
❖ Homography Net

- Result



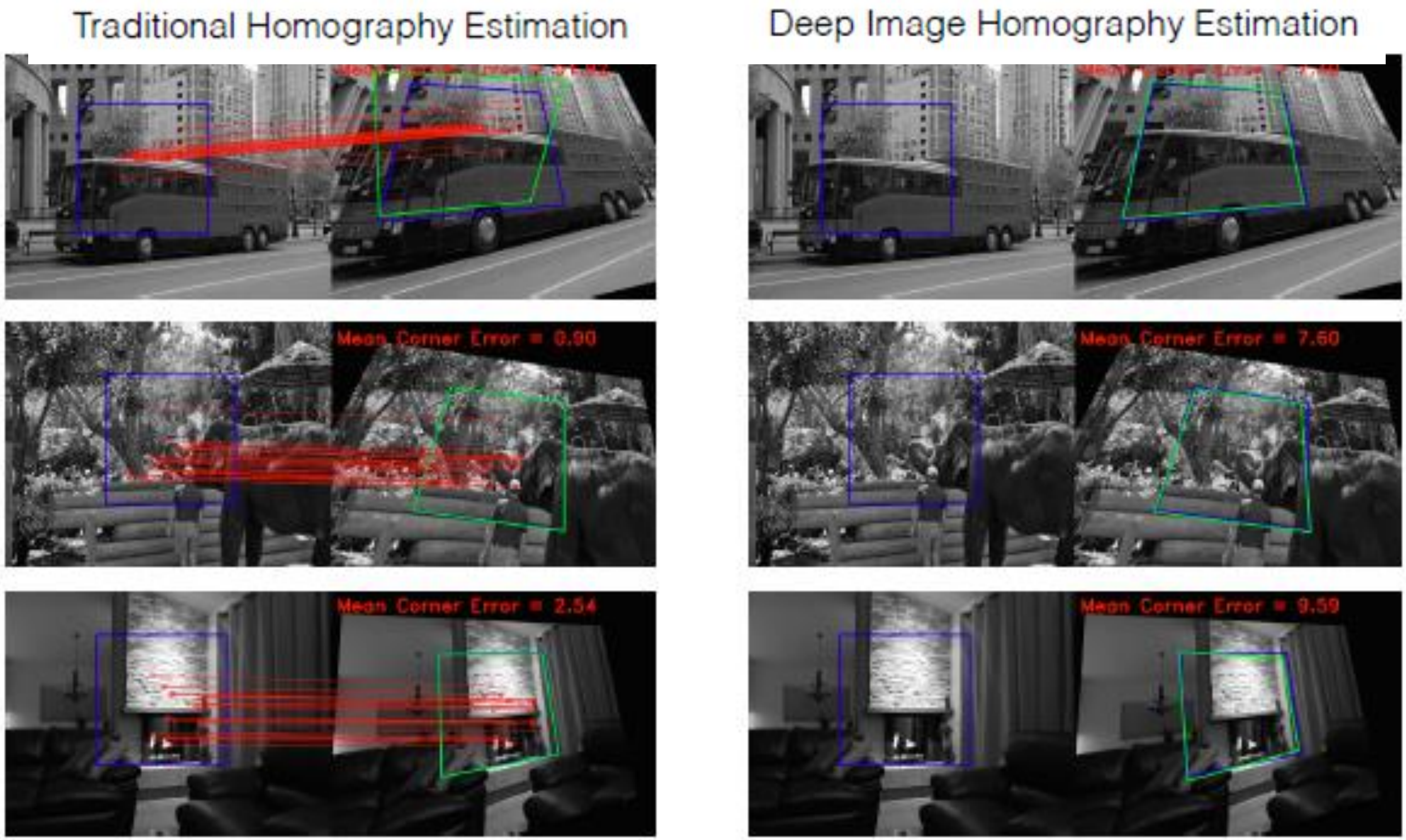
Classification은 아무래도 Quantization을 하니까 성능이 낮게 나오고, ORB 보다 성능이 좋음.

Deep Image Homography Estimation



파랑 : test할 때의 input data / 초록 : 예측한 Homography

Deep Image Homography Estimation



파랑 : test할 때의 input data / 초록 : 예측한 Homography

Conclusion

❖ Conclusion

- Ground Truth를 직접 만들어서 학습 한다는 게 신선 했음.
- 3x3 matrix 를 학습하는 것이 아닌, 4x2 matrix를 학습하는 것에 대해, 딥러닝이 뭐든 것을 다 알아서 하는 것은 아니다.
- 수학적으로 증명된 식을 이용해서 학습하면 결과가 좋아질 수 있다. (수학공부도 필요하다.)

❖ 궁금한 점.

- 일반적인 Descriptor는 spatial 정보가 존재하지만, deep 에서 뽑은 feature 들을 visualizatio하면 어떻게 보일지 궁금

❖ Future work

- Planar 환경이 아닌 non-planar환경에서 적용 하는 방법을 고민중…

Thank you
for your attention

Q&A