電腦視覺與應用
Computer Vision and Applications

Lecture06-2-Two-views geometry-case study

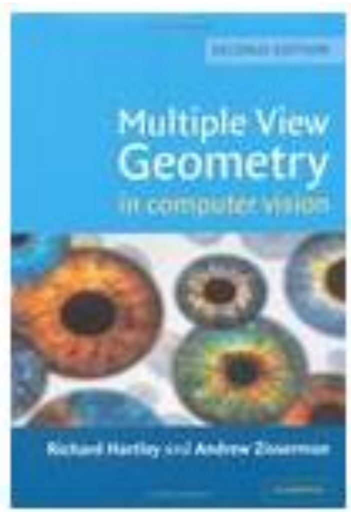
Tzung-Han Lin
National Taiwan University of Science and Technology
Graduate Institute of Color and Illumination Technology
e-mail: thl@mail.ntust.edu.tw

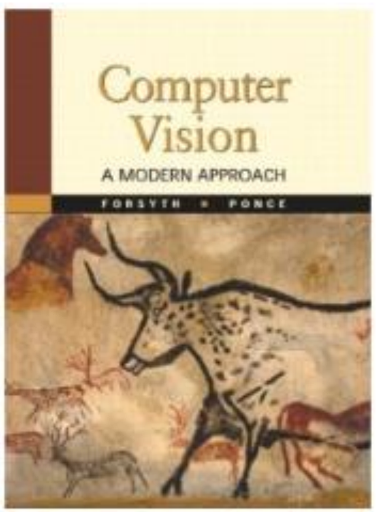
TAIWAN TECH National Taiwan University of Science and Technology

CIT 色彩與照明科技研究所 Graduate Institute of Color and Illumination Technology

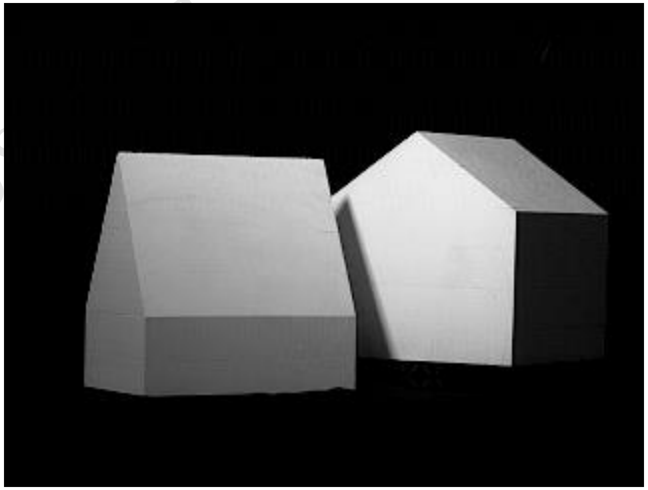
Two-views geometry

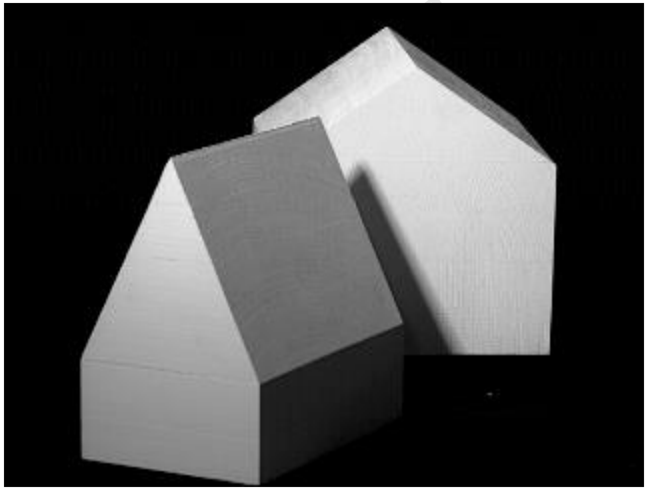
- Case study for stereo-vision & homography
- Lecture Reference at:
  - Multiple View Geometry in Computer Vision, Chapter 11
  - Computer Vision A Modern Approach, Chapter 11.

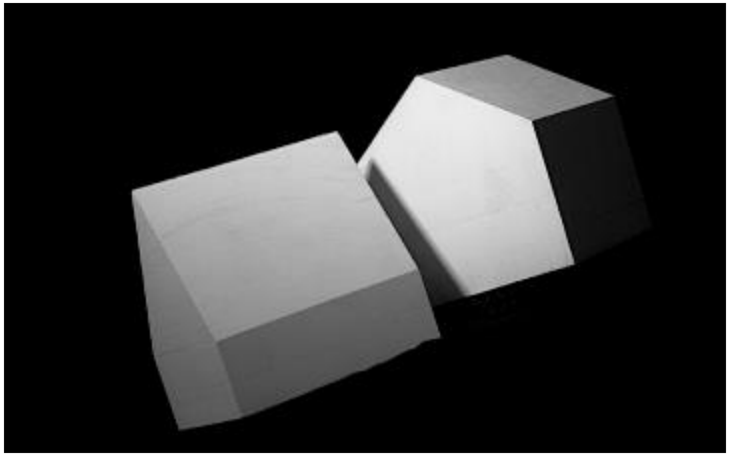


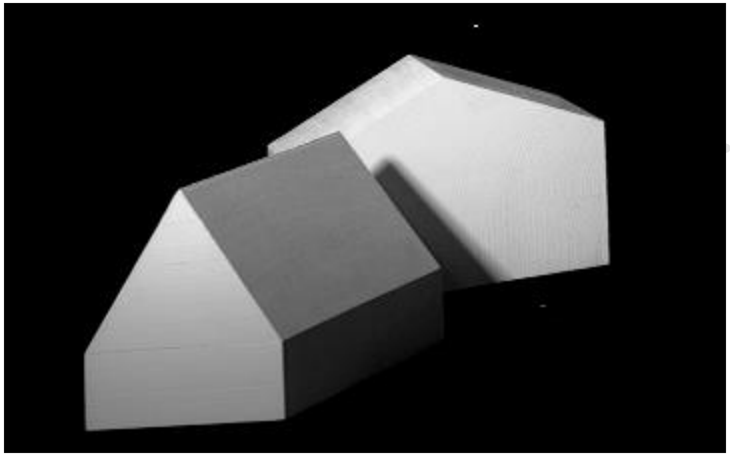


Stereo-image

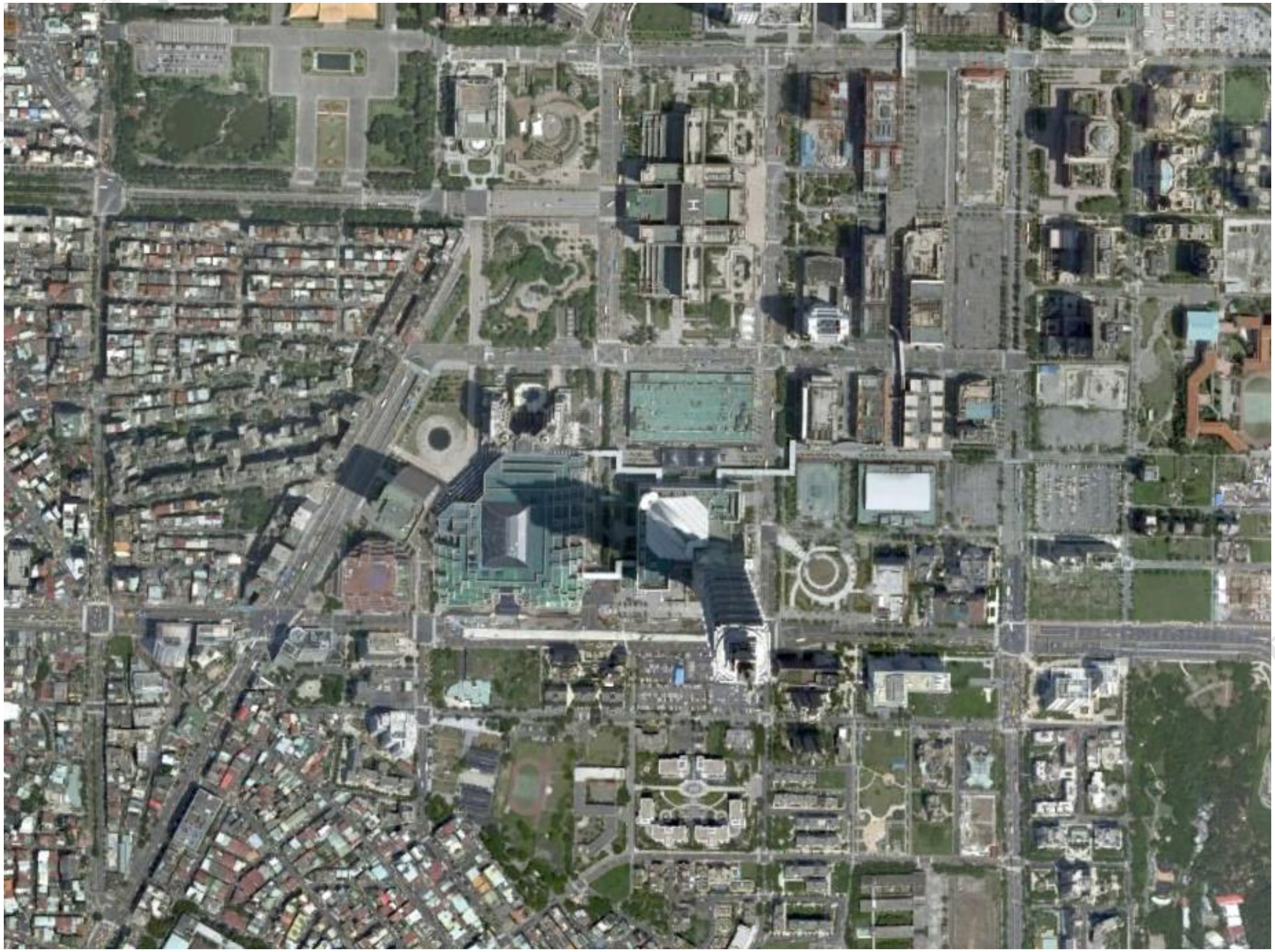






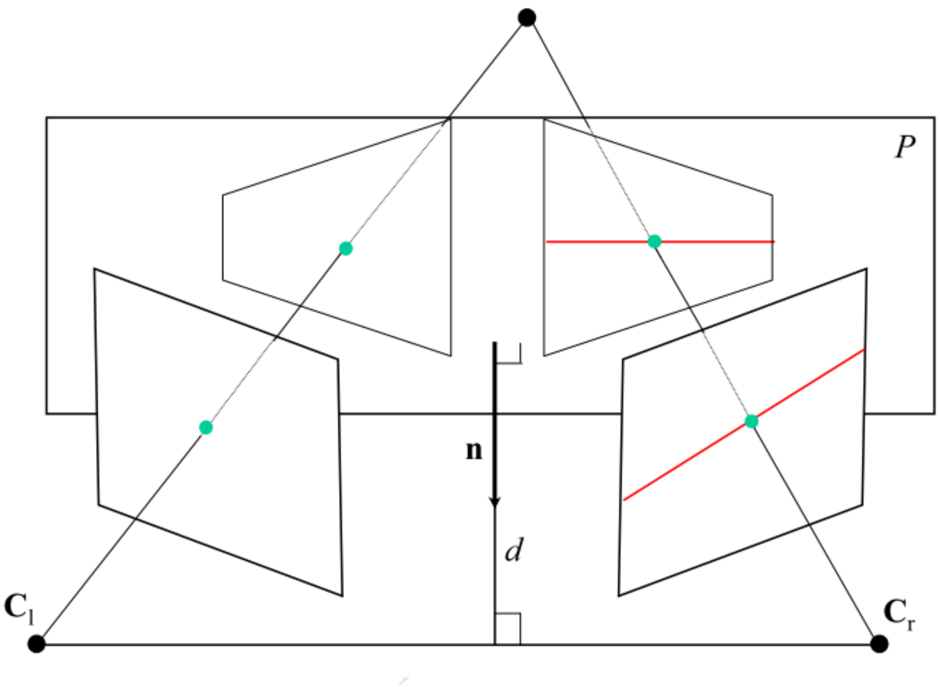


Rectified images



Rectification for stereo-image

- To projectively transfer images into a specific position



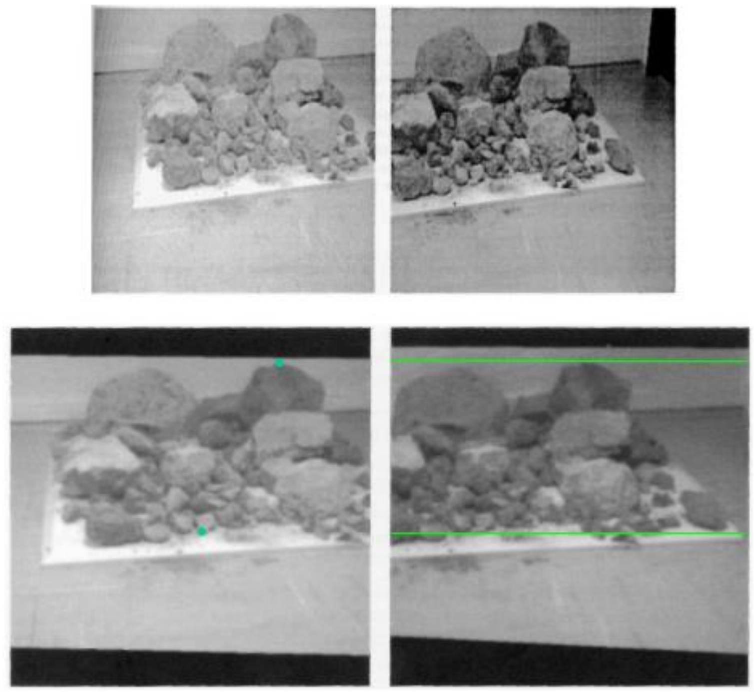
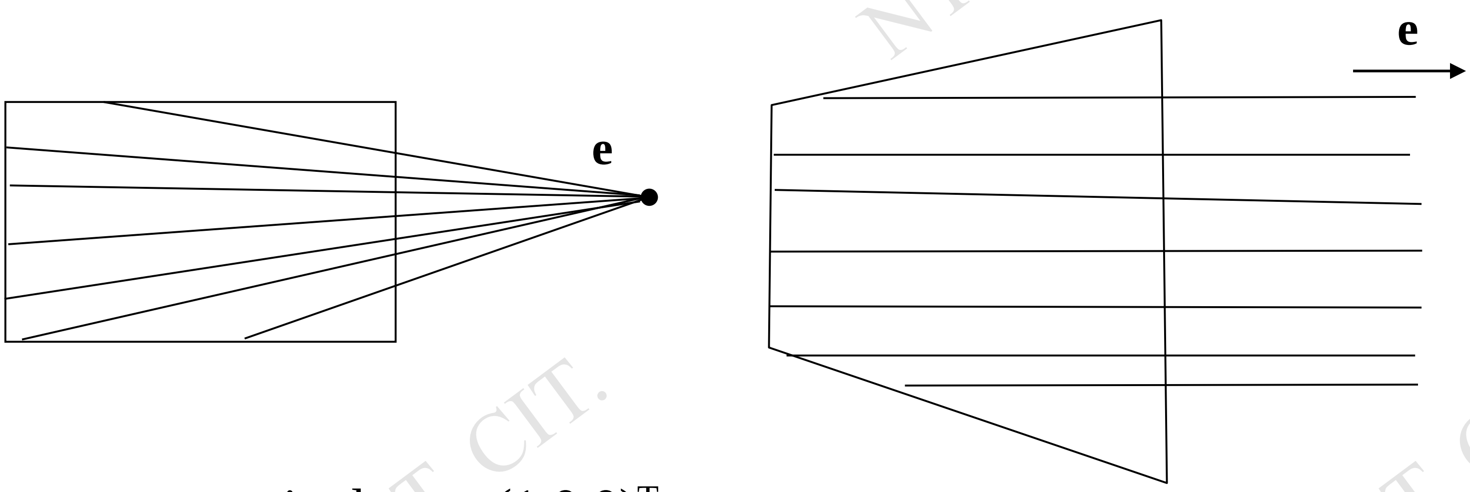


Image rectification

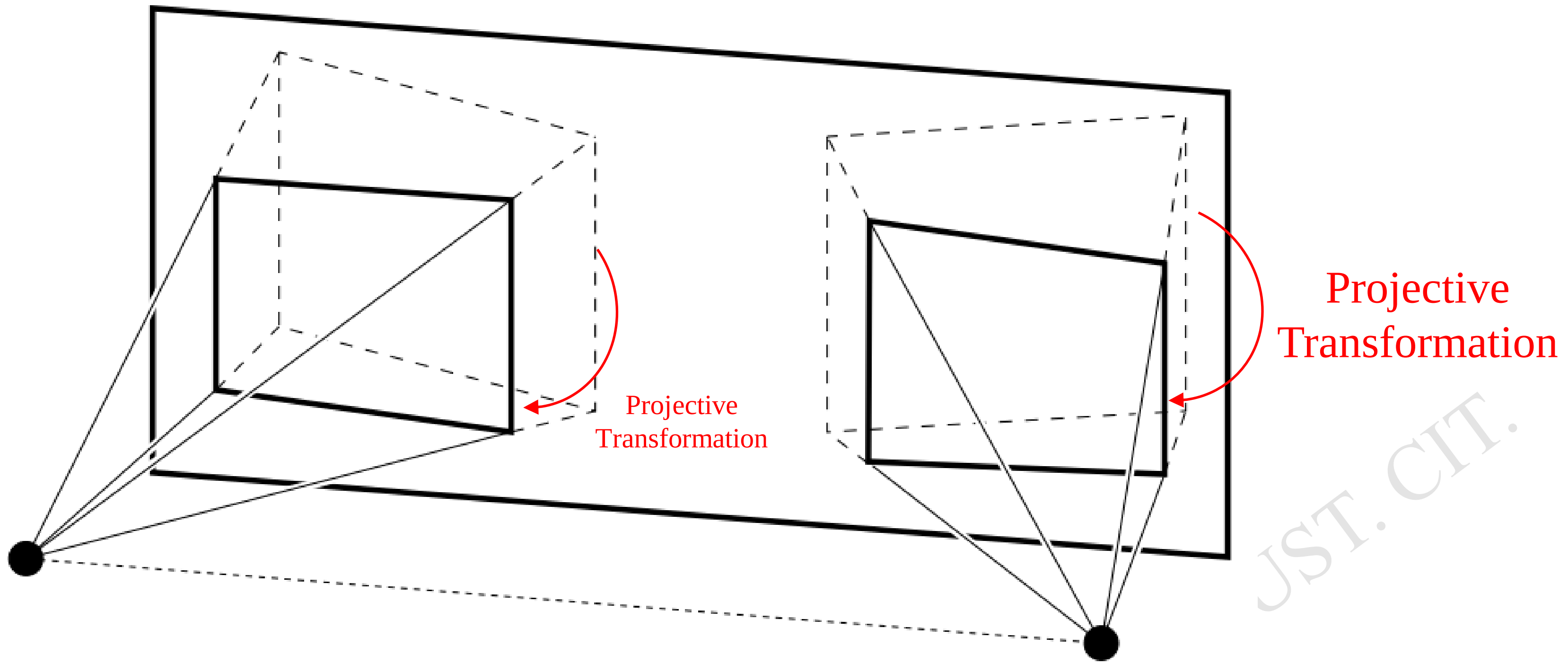
- Apply projective transformation so that epipolar lines correspond to horizontal line



map epipole **e** to $(1,0,0)^T$

try to minimize image distortion

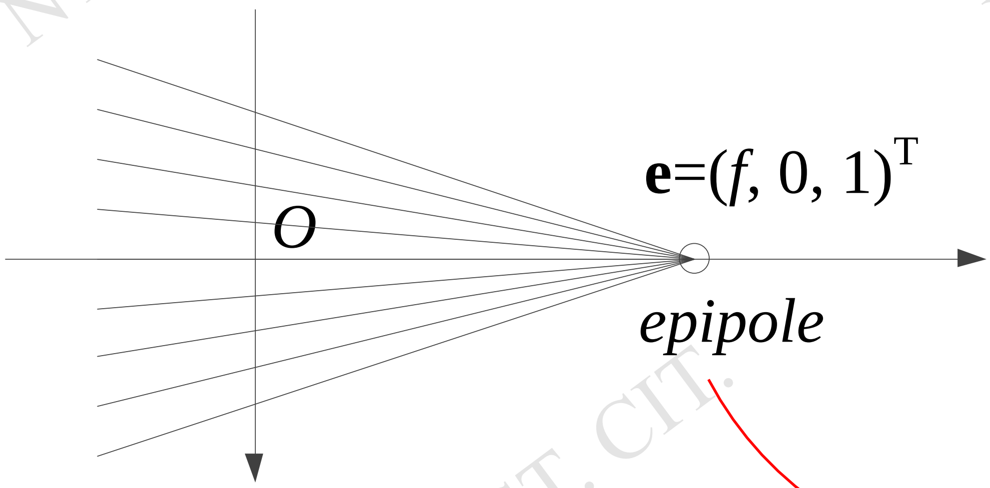
Image rectification



NOTE! This projective transformation is so called Homography!

Image rectification—solution

- To transfer epipole to the point at infinity



$$\mathbf{G} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix}$$

$\mathbf{e}_\infty = (f, 0, 0)^T$

point at infinity

$$\mathbf{e}_\infty = \mathbf{G}\mathbf{e}$$

Note! This is homography

Image rectification—solution, cont.

- However, in general, we have the configuration like the following figure.
- So, it needs a translation and a rotation to adjust the epipole on the special condition (on *x*-axis), and the homography will be derived as the format in the previous page.

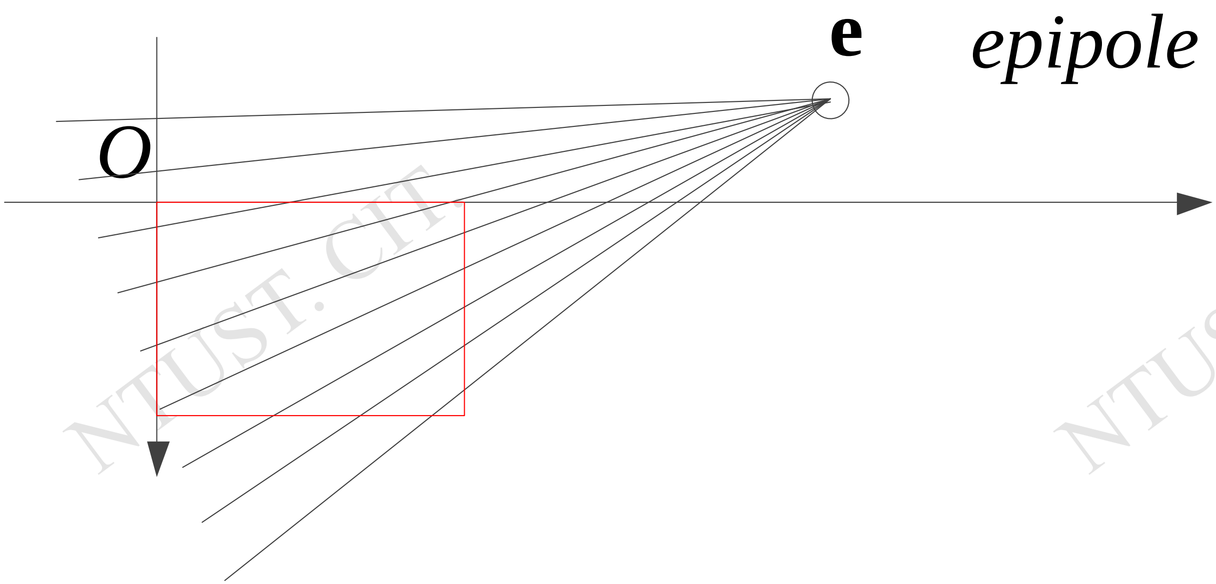


Image rectification—solution, cont.

- An appropriate choice of translation would be the center of image. For example, translate the image center to the origin.

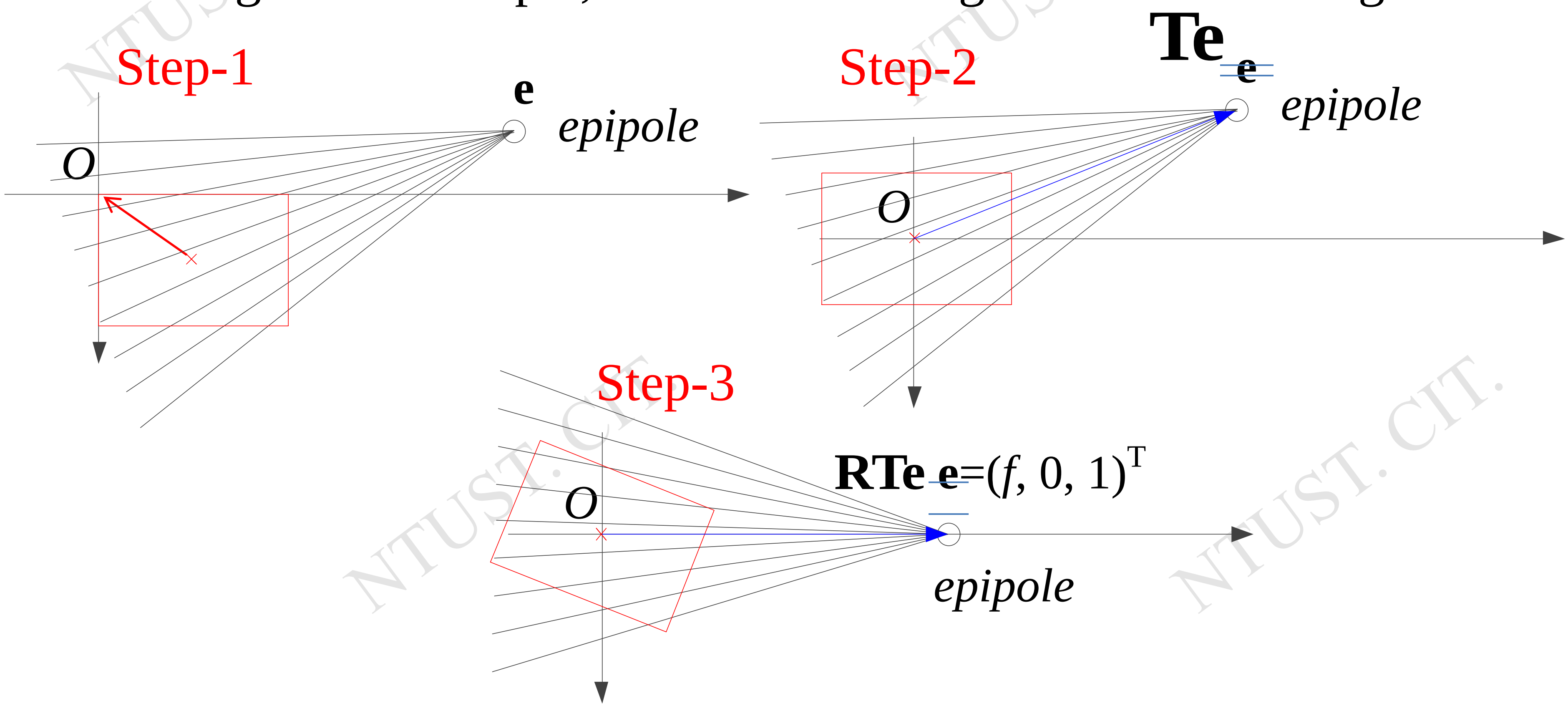
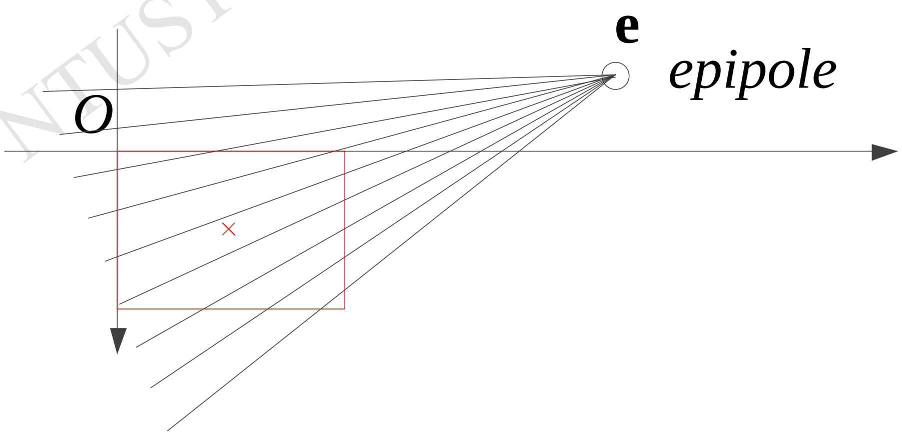


Image rectification—solution, cont.



$$\mathbf{e}_\infty = \mathbf{He}$$

$$\mathbf{e}_\infty = (f, 0, 0)^T$$

point at infinity

$$\mathbf{e}_\infty = \underbrace{\mathbf{GRT}}_{\text{homography}}\mathbf{e}$$

$$\rightarrow \quad \mathbf{e}_\infty = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix} \mathbf{RTe}$$

Image rectification—solution, cont.

$$\rightarrow \mathbf{e}_\infty = \mathbf{He} = \mathbf{GRTe} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -1/f & 0 & 1 \end{bmatrix} \mathbf{RTe}$$

- Review the procedure:

1. Determine one **F** from two image
2. Determine **e** (use cross product of two epipolar lines,line eqs: $\mathbf{l}=\mathbf{F}^T\mathbf{x}'$)
3. Determine **T** (of course, you know the image resolution. use its center)
4. Determine **R** (you already have **Te**, rotation angle should be $-\tan^{-1}\frac{y}{x}$
5. Then, get f from **RTe**.
6. Finally, you have **H**.

Note! **H** is calculated from the projective mapping (homography) of point-point. If you need line mapping according this homography, use $\mathbf{l}_{rect} = \mathbf{H}^{-T}\mathbf{l}$

Image rectification—solution, cont.

- Call the process, again.

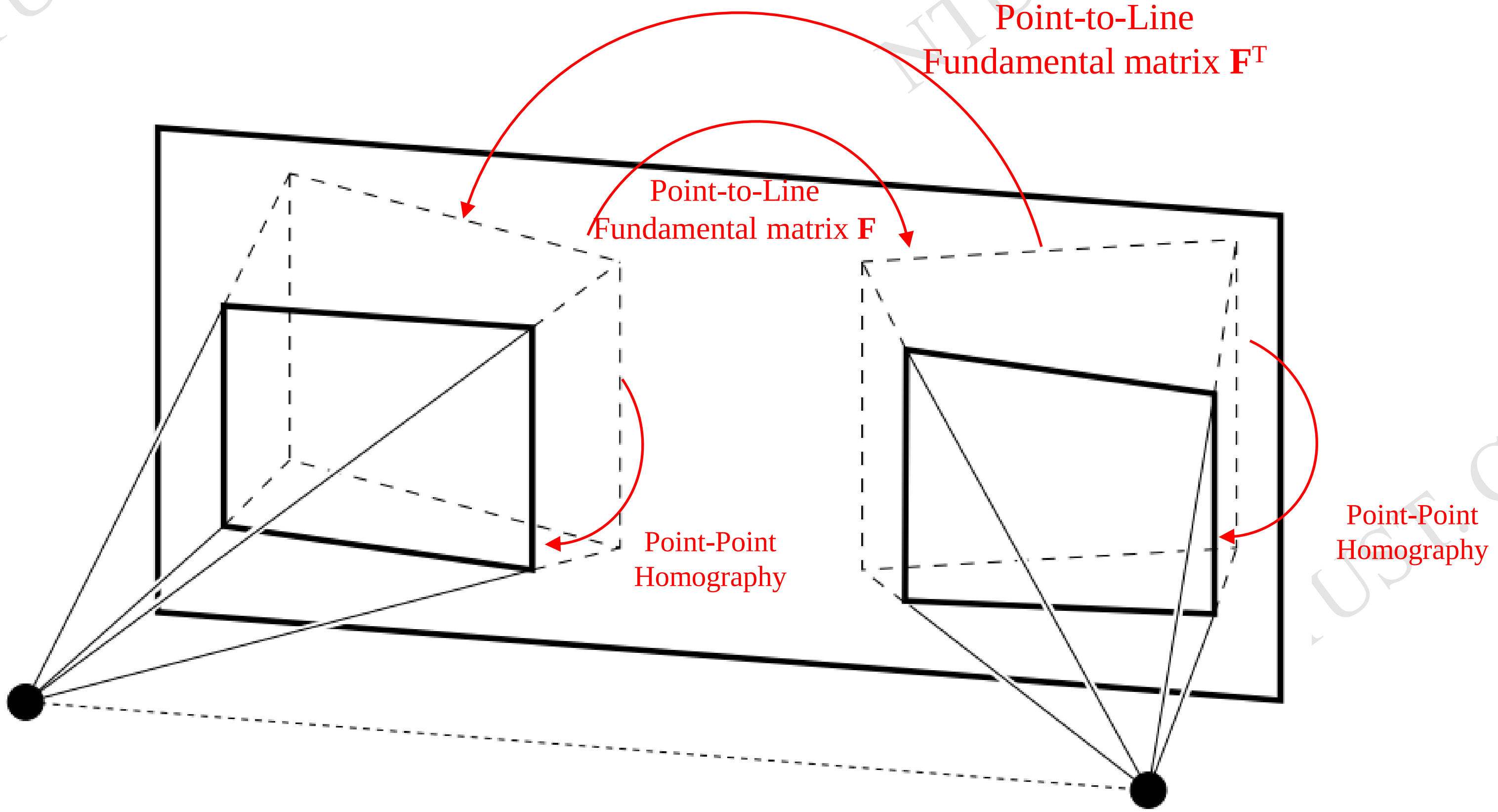
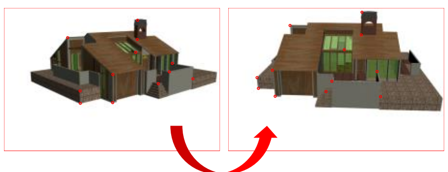


Image rectification—example

Recall the previous example.
Image resolution 720x480.



F=
-0.000219 -0.000913 0.292220
0.000103 -0.000245 0.737529
-0.142952 -0.450960 1.000000

e=
-4089.085693
1298.432373
1.000000

e'=
-552.206970
217.436905
1.000000

For 1^{st} image:

$$\mathbf{T}=\begin{bmatrix}1 & 0 & -360\\ 0 & 1 & -240\\ 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{T{*}e}=\boxed{\begin{matrix}-4449.085693\\ 1058.432373\\ 1.0\end{matrix}}$$

$$-\tan^{-1}\left(\frac{1058.432}{-4449.086}\right)$$

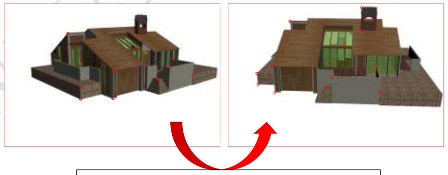
Rotation angle= 13.38º

$$\mathbf{R}=\begin{bmatrix}\cos(13.38^\circ) & -\sin(13.38^\circ) & 0\\ \sin(13.38^\circ) & \cos(13.38^\circ) & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{RTe}=\boxed{[-4573.3}\quad 0 \quad 1]^T$$

$$\therefore \mathbf{H}=\mathbf{GRT}=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -1/-4573.3 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos(13.38^\circ) & -\sin(13.38^\circ) & 0\\ \sin(13.38^\circ) & \cos(13.38^\circ) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -360\\ 0 & 1 & -240\\ 0 & 0 & 1\end{bmatrix}$$

Image rectification—example, cont.



F=
-0.000219 -0.000913 0.292220
0.000103 -0.000245 0.737529
-0.142952 -0.450960 1.000000

e=
-4089.085693
1298.432373
1.000000

e'=
-552.206970
217.436905
1.000000

For 2nd image:

$$\mathbf{T}'=\begin{bmatrix}1 & 0 & -360\\ 0 & 1 & -240\\ 0 & 0 & 1\end{bmatrix}$$

$$\mathbf{T'*e'}=\boxed{\begin{matrix}-912.2070\\ -22.5631\\ 1.0000\end{matrix}}$$

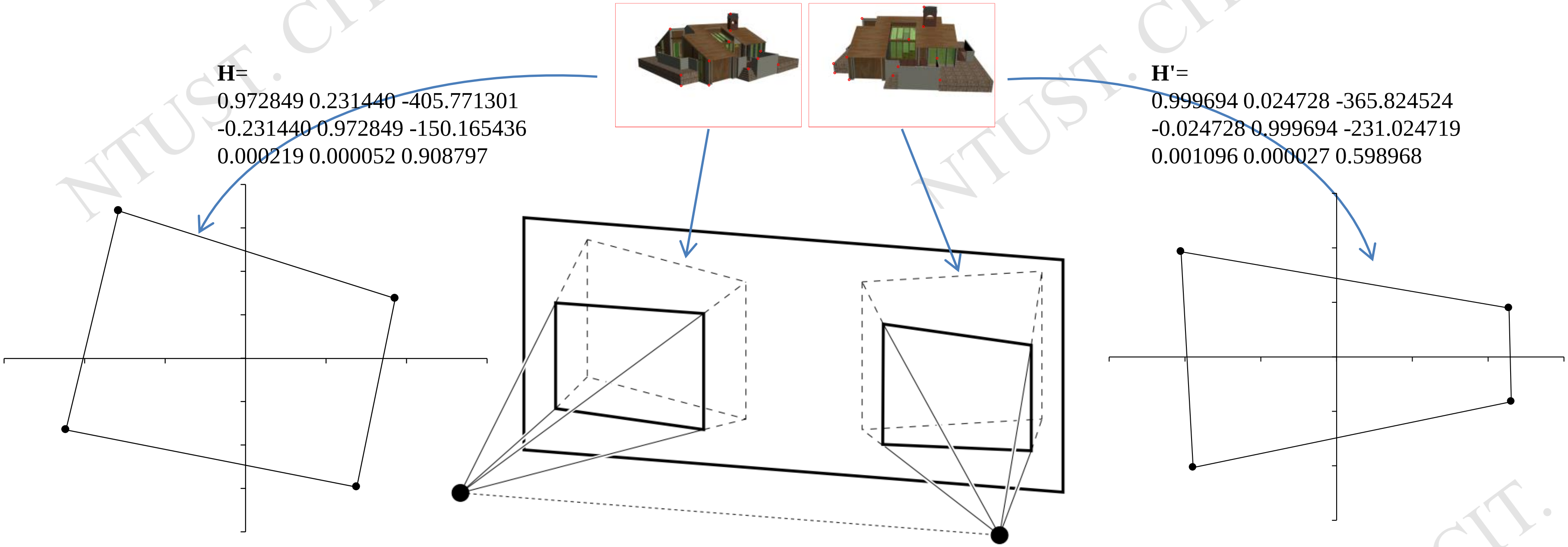
$$-\tan^{-1}\left(\frac{-22.56}{-912.2}\right)$$

Rotation angle=-1.42

$$\mathbf{R}'=\begin{bmatrix}\cos(-1.42^\circ) & -\sin(-1.42^\circ) & 0\\ \sin(-1.42^\circ) & \cos(-1.42^\circ) & 0\\ 0 & 0 & 1\end{bmatrix}$$

$$\therefore \mathbf{H}'=\mathbf{G}'\mathbf{R}'\mathbf{T}'=\begin{bmatrix}1 & 0 & 0\\ 0 & 1 & 0\\ -1/-912.486 & 0 & 1\end{bmatrix}\begin{bmatrix}\cos(-1.42^\circ) & -\sin(-1.42^\circ) & 0\\ \sin(-1.42^\circ) & \cos(-1.42^\circ) & 0\\ 0 & 0 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & -360\\ 0 & 1 & -240\\ 0 & 0 & 1\end{bmatrix}$$

Image rectification—example, cont.

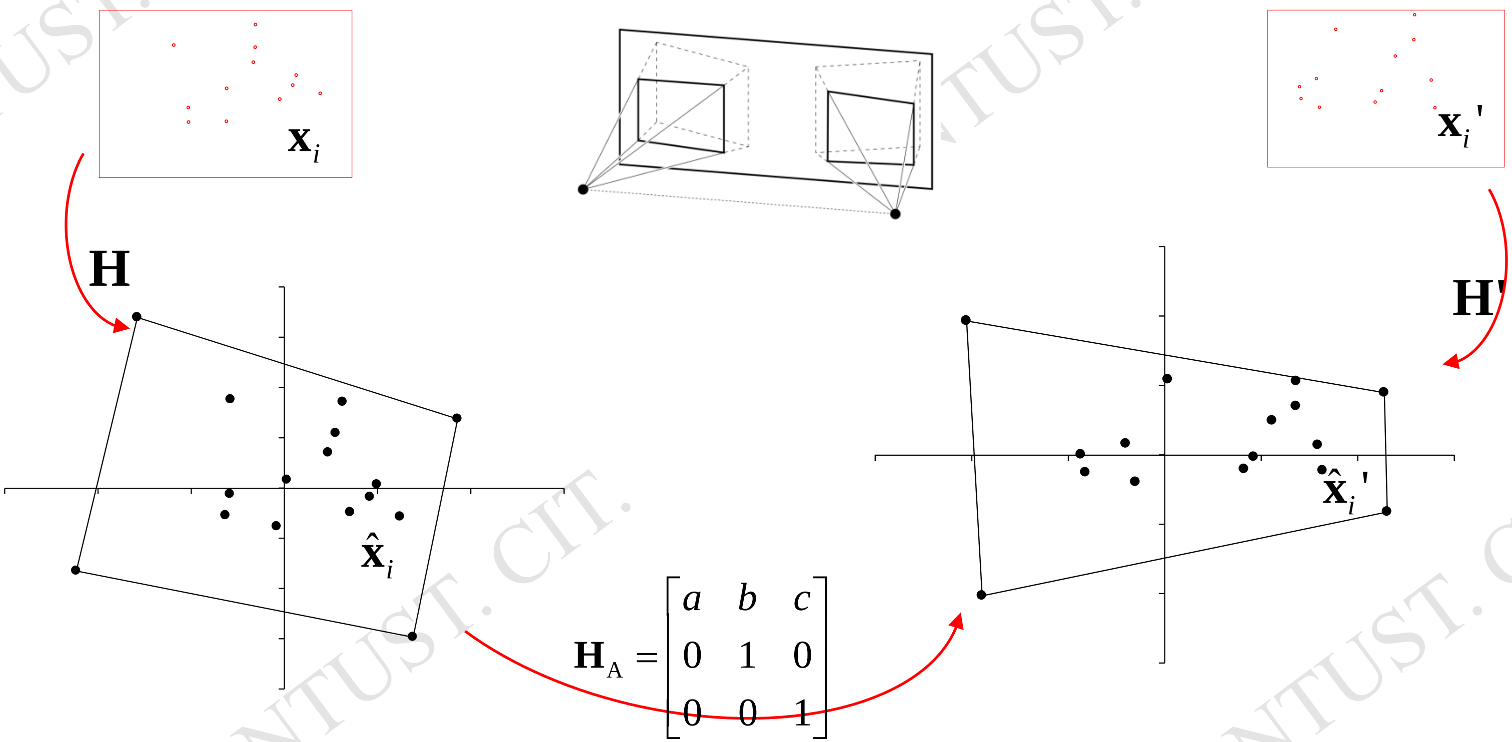


After rectification adjustment, two problems remain

1. Correspondences in two image may have disparity on *y* direction.
2. Pixel coordinates may not fall in positive region.

Image rectification—example, cont.

- Minimize the horizontal disparity (minimize dist.)



$$\mathbf{H}_A = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Minimize $\sum_i d(\mathbf{H}_A\hat{\mathbf{x}}_i, \hat{\mathbf{x}}_i')^2$

Note! The minimization is to minimize "distance" of correspondences (that are $\hat{\mathbf{x}}_i', \mathbf{H}_A\hat{\mathbf{x}}_i$)

Image rectification—example, cont.

- Minimize the horizontal disparity—cont.

Minimize $\sum_i d(\mathbf{H}_A \hat{\mathbf{x}}_i, \hat{\mathbf{x}}_i')^2$

$$\rightarrow \quad \sum_i [(a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i')^2 + (\hat{y}_i - \hat{y}_i')^2]$$

$(\hat{y}_i - \hat{y}_i')^2$ is a constant, and will not affect the chosen of solution for (a, b, c)

So, minimization terms are reduced…

Minimize $\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i')^2$

Image rectification—example, cont.

- Minimize the horizontal disparity—cont.

Minimize $\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i')^2$

For convenience

let $g = \sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i')^2$

To determine extreme value of *g*, take partial difference for *g*. (偏微分)

$$\begin{cases} \dfrac{\partial g}{\partial a} = 0 \\ \dfrac{\partial g}{\partial b} = 0 \\ \dfrac{\partial g}{\partial c} = 0 \end{cases} \Rightarrow \begin{cases} 2\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i')\hat{x}_i = 0 \\ 2\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i')\hat{y}_i = 0 \\ 2\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i') = 0 \end{cases} \Rightarrow \begin{cases} (\sum_i \hat{x}_i^2)a + (\sum_i \hat{x}_i\hat{y}_i)b + (\sum_i \hat{x}_i)c = \sum_i \hat{x}_i\hat{x}_i' \\ (\sum_i \hat{x}_i\hat{y}_i)a + (\sum_i \hat{y}_i^2)b + (\sum_i \hat{y}_i)c = \sum_i \hat{y}_i\hat{x}_i' \\ (\sum_i \hat{x}_i)a + (\sum_i \hat{y}_i)b + (\sum_i 1)c = \sum_i \hat{x}_i' \end{cases}$$

Image rectification—example, cont.

- Minimize the horizontal disparity—cont.

$$\begin{bmatrix} (\sum_i \hat{x}_i^2) & (\sum_i \hat{x}_i \hat{y}_i) & (\sum_i \hat{x}_i) \\ (\sum_i \hat{x}_i \hat{y}_i) & (\sum_i \hat{y}_i^2) & (\sum_i \hat{y}_i) \\ (\sum_i \hat{x}_i) & (\sum_i \hat{y}_i) & (\sum_i 1) \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_i \hat{x}_i \hat{x}_i' \\ \sum_i \hat{y}_i \hat{x}_i' \\ \sum_i \hat{x}_i' \end{bmatrix}$$

$$\begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} (\sum_i \hat{x}_i^2) & (\sum_i \hat{x}_i \hat{y}_i) & (\sum_i \hat{x}_i) \\ (\sum_i \hat{x}_i \hat{y}_i) & (\sum_i \hat{y}_i^2) & (\sum_i \hat{y}_i) \\ (\sum_i \hat{x}_i) & (\sum_i \hat{y}_i) & (\sum_i 1) \end{bmatrix}^{-1} \begin{bmatrix} \sum_i \hat{x}_i \hat{x}_i' \\ \sum_i \hat{y}_i \hat{x}_i' \\ \sum_i \hat{x}_i' \end{bmatrix}$$

Image rectification—example, cont.

- Minimize the horizontal disparity—cont.

- After applying transformation matrix to right image:

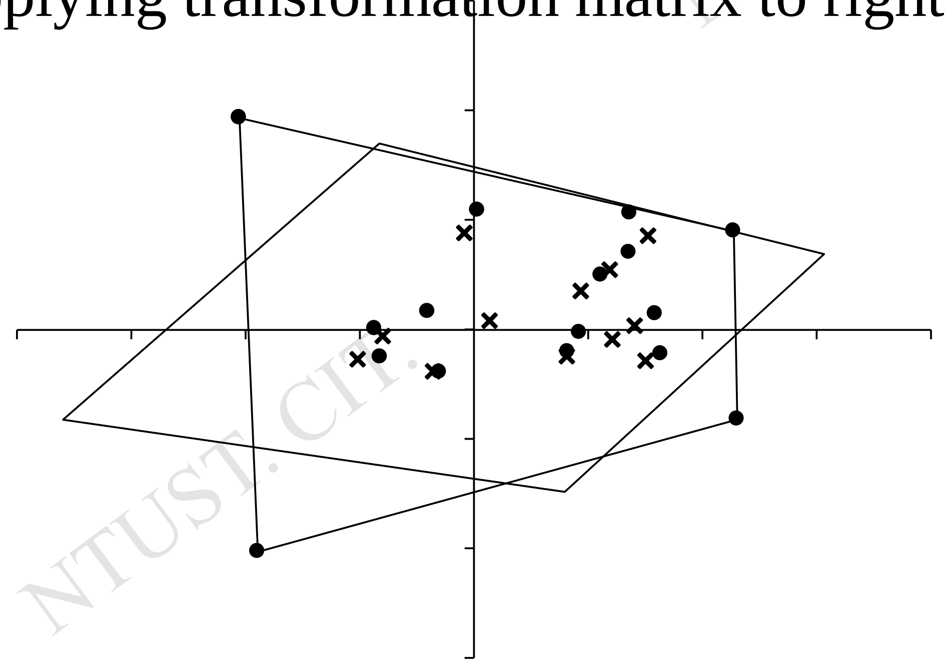


Image rectification—example, cont.

- Minimize the vertical disparity, as well as horizontal distance.

$$\text{Minimize} \quad \sum_i d(\mathbf{H}_A \hat{\mathbf{x}}_i, \hat{\mathbf{x}}_i')^2$$

$$\mathbf{H}_A = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & 1 \end{bmatrix}$$

$$\rightarrow \sum_i [(a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i')^2 + (d\hat{y}_i + e - \hat{y}_i')^2]$$

$$\begin{cases} \frac{\partial}{\partial a} = 0 \\ \frac{\partial}{\partial b} = 0 \\ \frac{\partial}{\partial c} = 0 \\ \frac{\partial}{\partial d} = 0 \\ \frac{\partial}{\partial e} = 0 \end{cases} \Rightarrow \begin{cases} 2\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i')\hat{x}_i = 0 \\ 2\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i')\hat{y}_i = 0 \\ 2\sum_i (a\hat{x}_i + b\hat{y}_i + c - \hat{x}_i') = 0 \\ 2\sum_i (d\hat{y}_i + e - \hat{y}_i')\hat{y}_i = 0 \\ 2\sum_i (d\hat{y}_i + e - \hat{y}_i') = 0 \end{cases} \Rightarrow \begin{cases} (\sum_i \hat{x}_i^2)a + (\sum_i \hat{x}_i\hat{y}_i)b + (\sum_i \hat{x}_i)c = \sum_i \hat{x}_i\hat{x}_i' \\ (\sum_i \hat{x}_i\hat{y}_i)a + (\sum_i \hat{y}_i^2)b + (\sum_i \hat{y}_i)c = \sum_i \hat{y}_i\hat{x}_i' \\ (\sum_i \hat{x}_i)a + (\sum_i \hat{y}_i)b + (\sum_i 1)c = \sum_i \hat{x}_i' \\ (\sum_i \hat{y}_i^2)d + (\sum_i \hat{y}_i)e = \sum_i \hat{y}_i\hat{y}_i' \\ (\sum_i \hat{y}_i)d + (\sum_i 1)e = \sum_i \hat{y}_i' \end{cases}$$

Image rectification—example, cont.

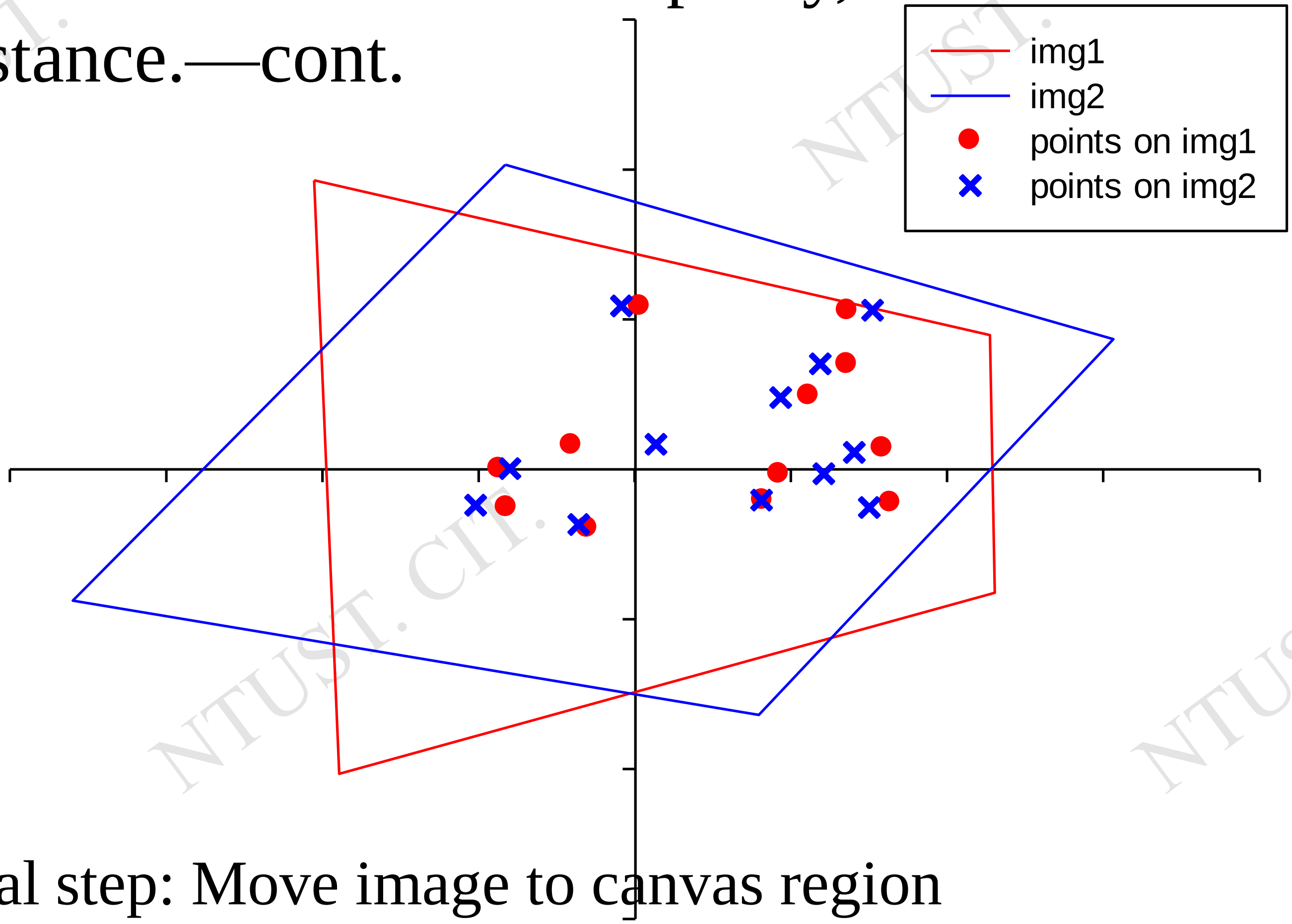
- Minimize the vertical disparity, as well as horizontal distance.—cont.

$$\begin{bmatrix} \sum_i \hat{x}_i^2 & \sum_i \hat{x}_i\hat{y}_i & \sum_i \hat{x}_i & 0 & 0 \\ \sum_i \hat{x}_i\hat{y}_i & \sum_i \hat{y}_i^2 & \sum_i \hat{y}_i & 0 & 0 \\ \sum_i \hat{x}_i & \sum_i \hat{y}_i & \sum_i 1 & 0 & 0 \\ 0 & 0 & 0 & \sum_i \hat{y}_i^2 & \sum_i \hat{y}_i \\ 0 & 0 & 0 & \sum_i \hat{y}_i & \sum_i 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = \begin{bmatrix} \sum_i \hat{x}_i\hat{x}_i' \\ \sum_i \hat{y}_i\hat{x}_i' \\ \sum_i \hat{x}_i' \\ \sum_i \hat{y}_i\hat{y}_i' \\ \sum_i \hat{y}_i' \end{bmatrix} \Rightarrow \begin{bmatrix} a \\ b \\ c \\ d \\ e \end{bmatrix} = \begin{bmatrix} \sum_i \hat{x}_i^2 & \sum_i \hat{x}_i\hat{y}_i & \sum_i \hat{x}_i & 0 & 0 \\ \sum_i \hat{x}_i\hat{y}_i & \sum_i \hat{y}_i^2 & \sum_i \hat{y}_i & 0 & 0 \\ \sum_i \hat{x}_i & \sum_i \hat{y}_i & \sum_i 1 & 0 & 0 \\ 0 & 0 & 0 & \sum_i \hat{y}_i^2 & \sum_i \hat{y}_i \\ 0 & 0 & 0 & \sum_i \hat{y}_i & \sum_i 1 \end{bmatrix}^{-1} \begin{bmatrix} \sum_i \hat{x}_i\hat{x}_i' \\ \sum_i \hat{y}_i\hat{x}_i' \\ \sum_i \hat{x}_i' \\ \sum_i \hat{y}_i\hat{y}_i' \\ \sum_i \hat{y}_i' \end{bmatrix}$$

Finally, recover $\boxed{\mathbf{H}_{\mathrm{A}} = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & 1 \end{bmatrix}}$

Image rectification—example, cont.

- Minimize the vertical disparity, as well as horizontal distance.—cont.



Final step: Move image to canvas region

Image rectification—example, cont.

- Unique solution?

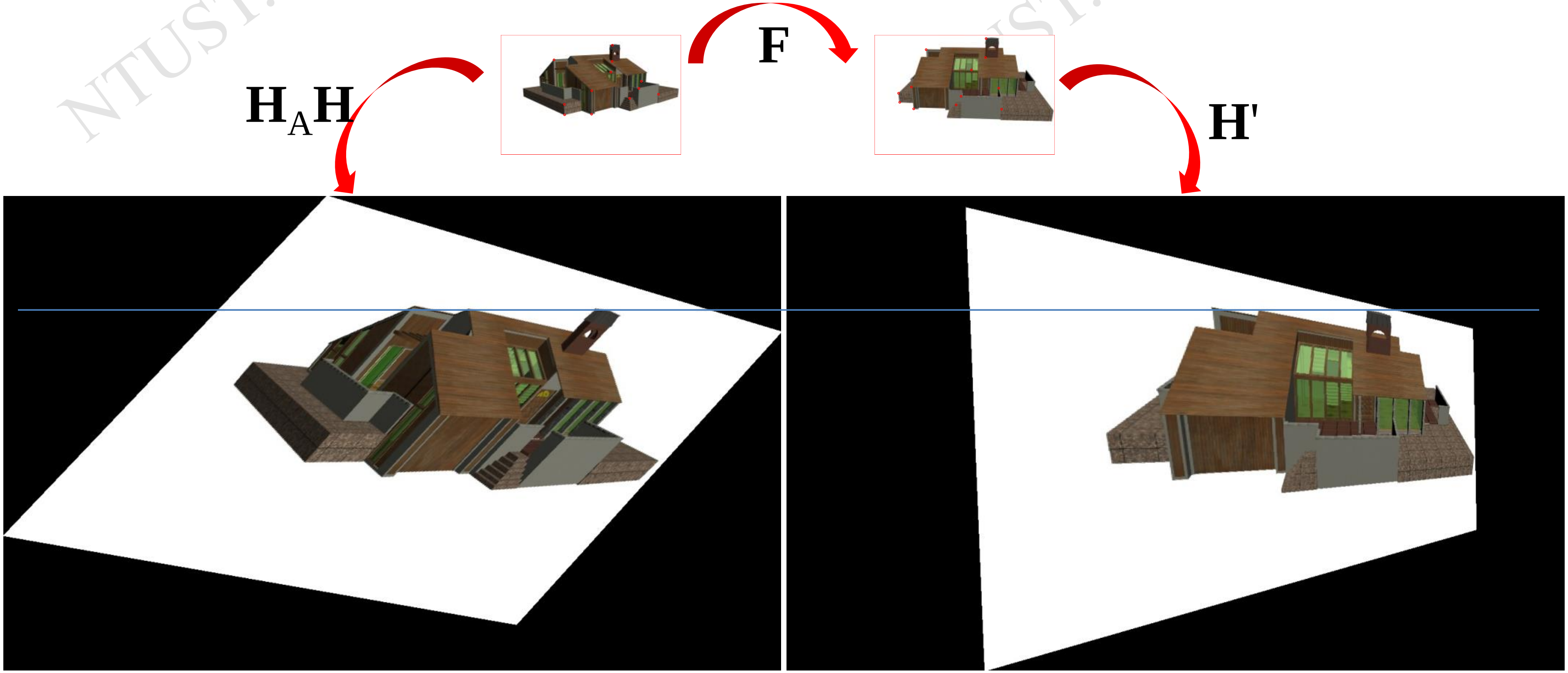


Image rectification—openCV

- StereoRectifyUncalibrated

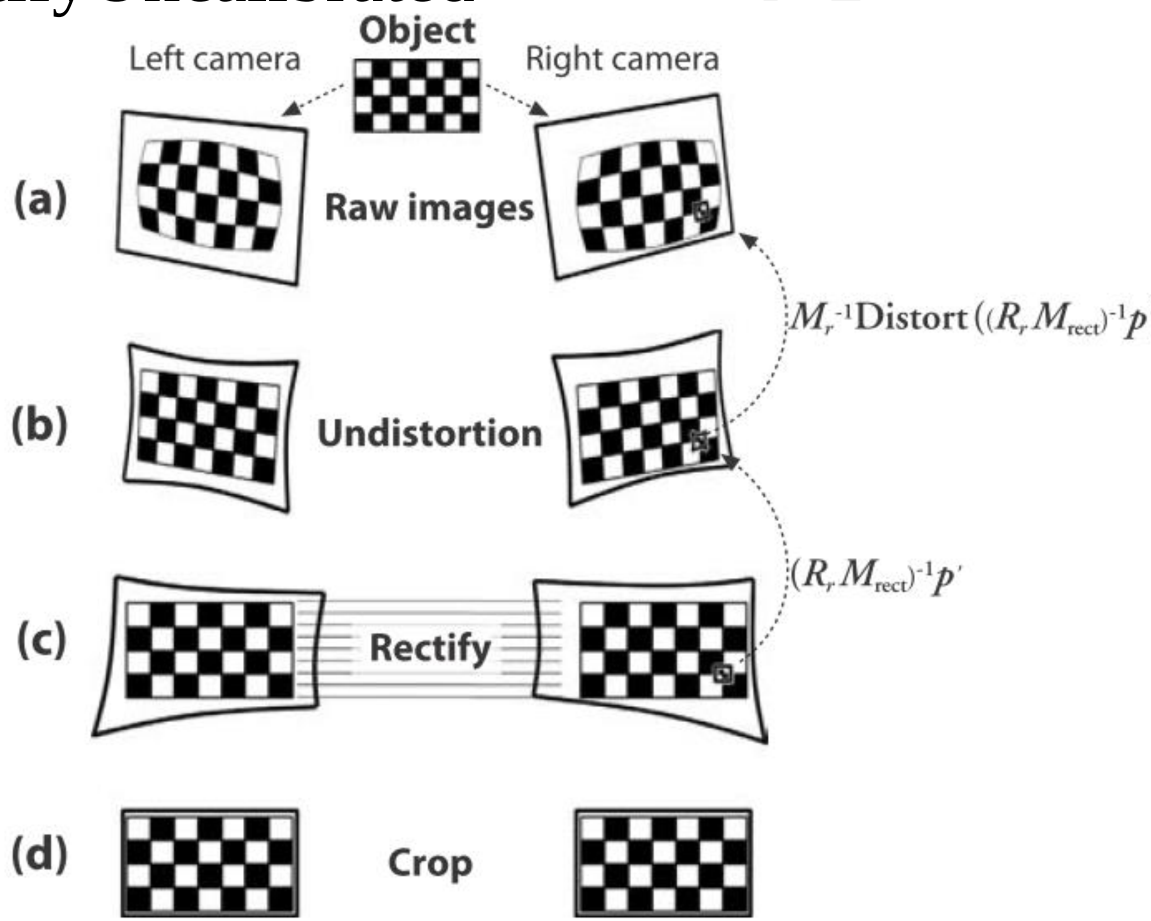


Image rectification—openCV

- The same operation in OpenCV

cvStereoRectifyUncalibrated

```
int cvStereoRectifyUncalibrated(
    const CvMat* points1,
    const CvMat* points2,
    const CvMat* F,
        CvSize imageSize,
        CvMat* Hl,
        CvMat* Hr,
        double threshold
);
```

points1, points2: The 2 arrays of corresponding 2D points. (input)

F: Fundamental Matrix (input)

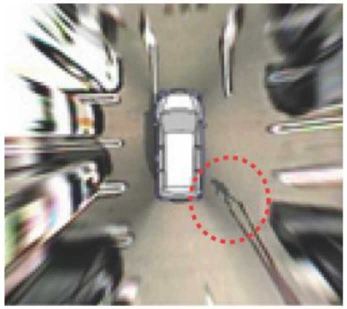
imageSize: (input)

Hl, Hr: Homography Matrix (output for Left, Right Images)

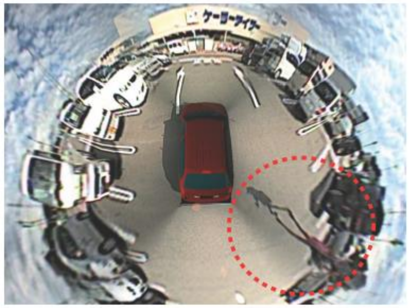
threshold: (input) for rejecting the outliers for which $|\mathbf{x}'^{T}\mathbf{F}\mathbf{x}| > \text{threshold}$

Homograhy—Applications

- Intelligent automobile

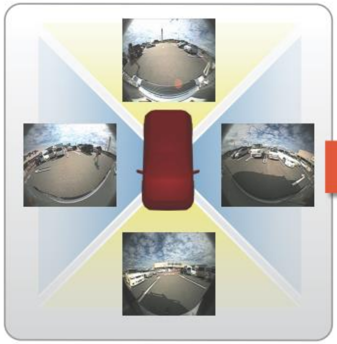


Conventional technology-based image, vehicles and people are not visible.

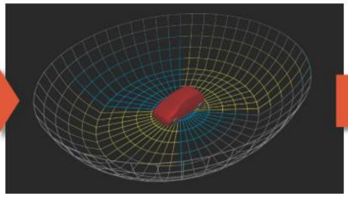


Sample view using Fujitsu Laboratories' new technology perspective from above-rear (pedestrian is visible)

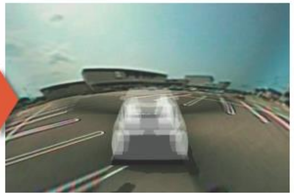
Sample view using Fujitsu's new technology, perspective from front facing vehicle (rearview pedestrian is visible)



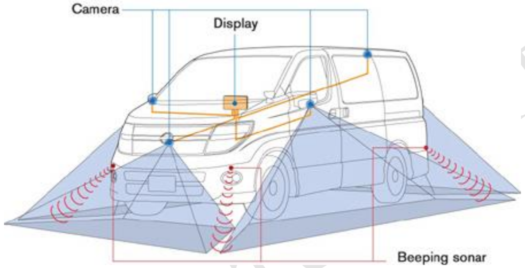
4 cameras capture video images of 4 different views (perspectives)



Virtual 3D model for projection of video images is synthesized (scene is projected virtually onto a 3D curved plane). Image is changed to the desired perspective



Desired view (perspective) is displayed



NISSAN

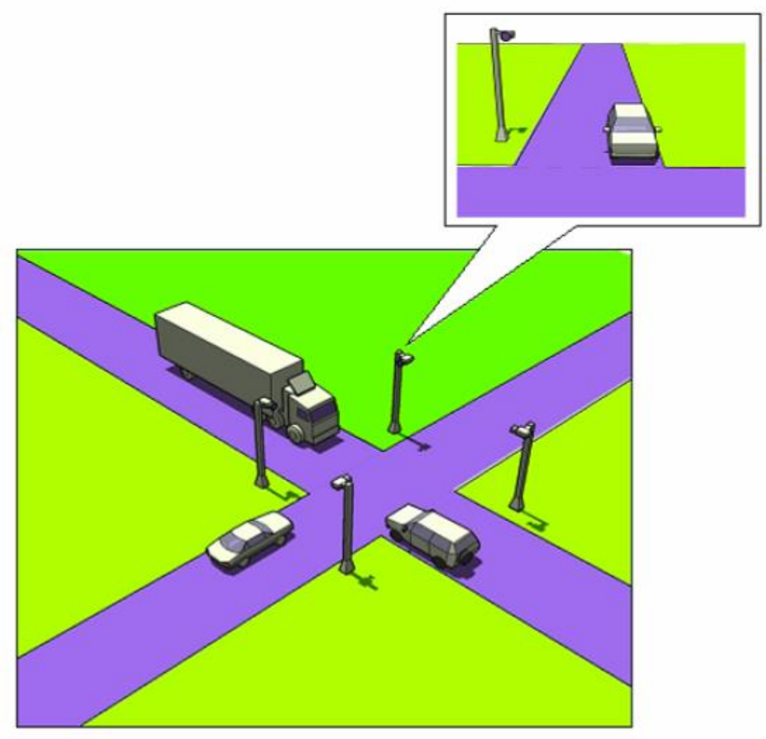
FUJITSU
http://www.fujitsu.com/downloads/MICRO/fma/pdf/360_OmniView_AppNote.pdf

Homograhy—Applications

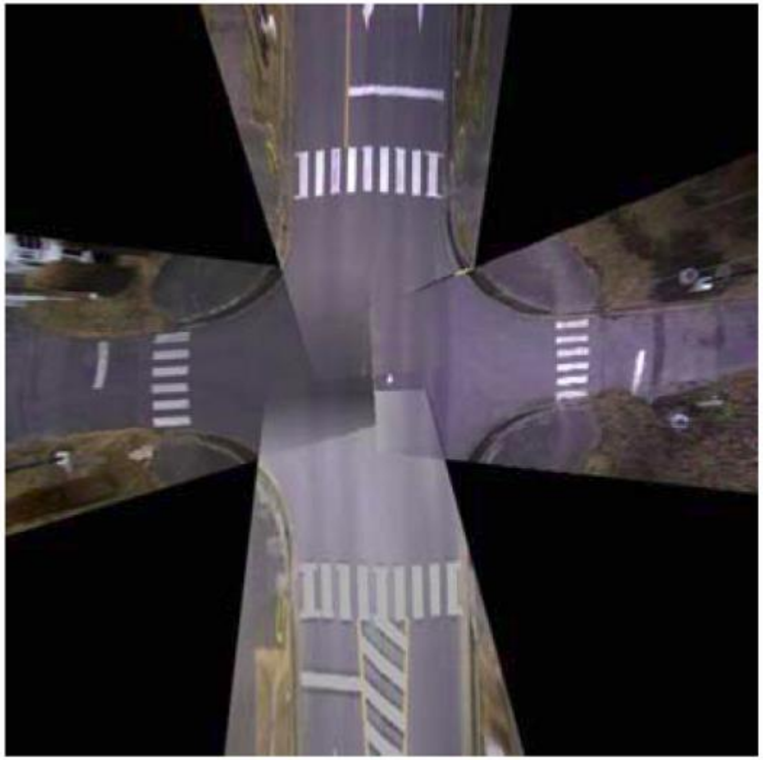
- Intelligent automobile—cont.

Homograhy—Applications

- Multi-camera surveillance
- 物聯網



筑波大學(JP)



筑波大學(JP)

Homograhy—Applications

- Application scenarios

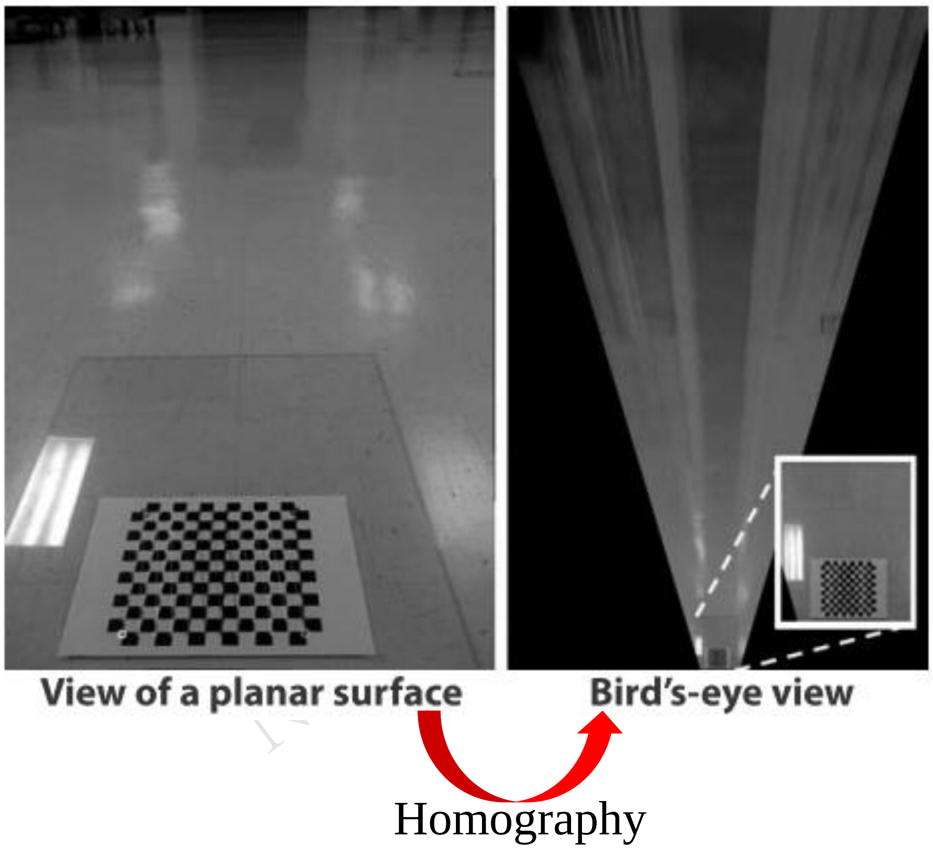
Image-Undistortion → **Homography** → **Image stitching** →

- Auto parking / Parking assistant system
- Obstacle detection (distance measurement)
- Driving / navigation (Auto or assistance)
- Safety driving
- ……

Note! In many application, wide-angle lens (fish-eye) cameras are used. In these situations, the UN-distortion processing is important.

Homograhy—Applications

- Once again, define your problem first!



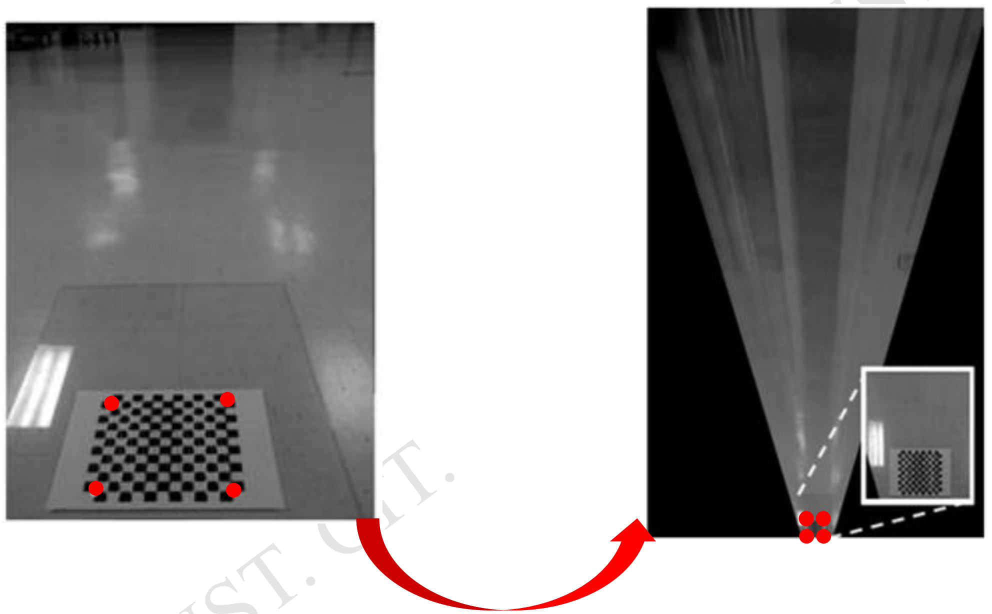
1. Pre-processing or post-processing for **H**
2. Constant **H** or various **H**

Solution: (homography)
1. Line cue ? Point cue ?
2. Correspondence
3. Scale issue

Picture from OReilly Learning OpenCV

Homograhy—Applications

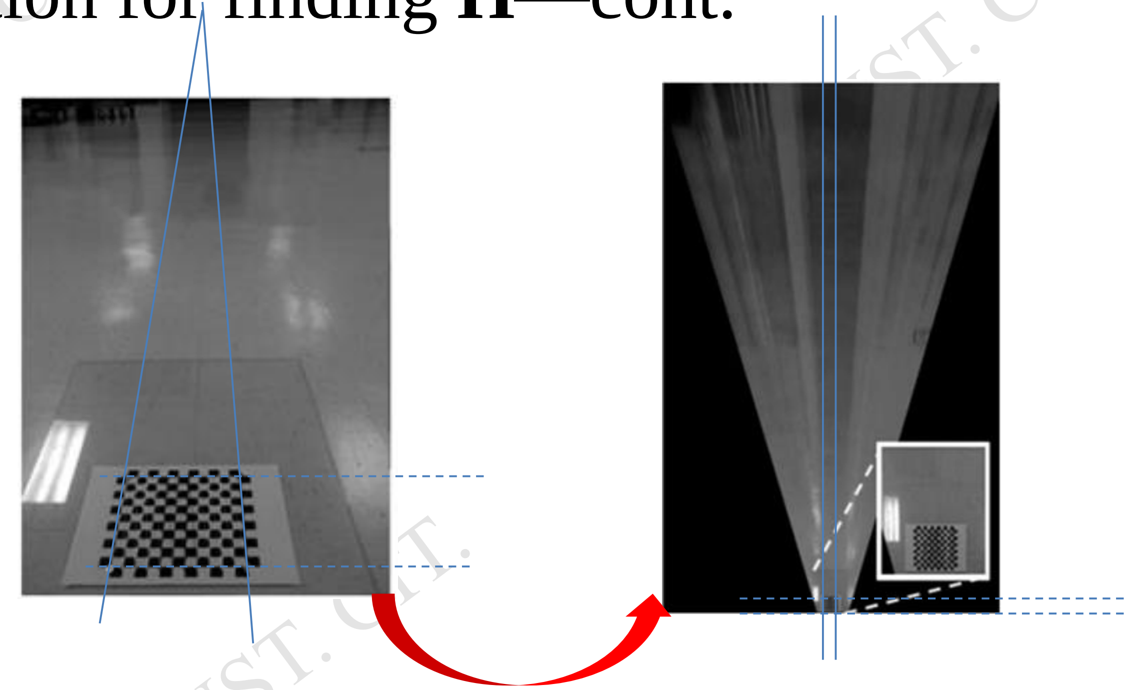
- Solution for finding **H**



Possible method
Manually assign at least 4 correspondences (define your new image-resolution well)

Homograhy—Applications

- Solution for finding **H**—cont.

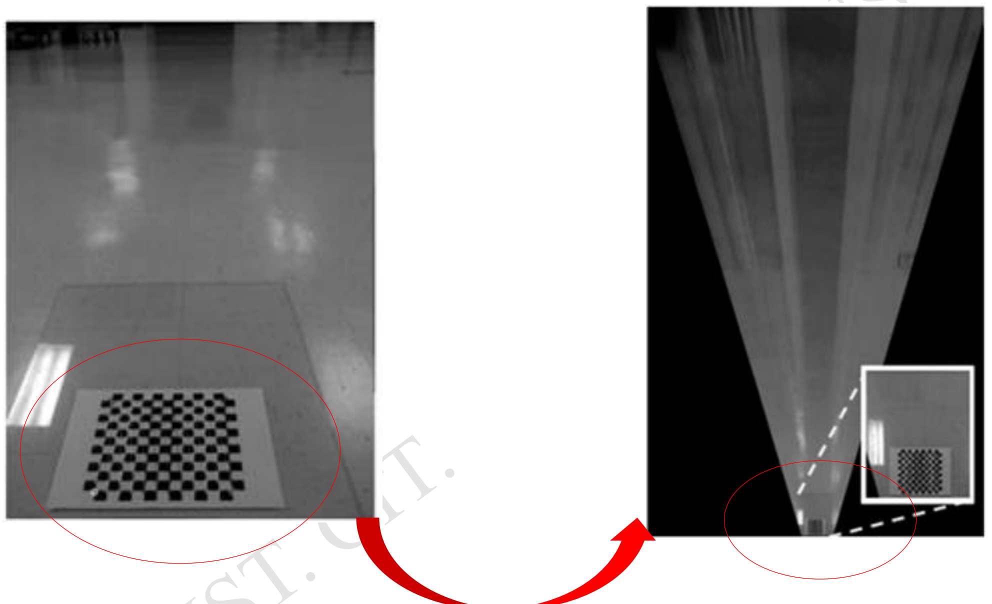


Possible method
Find vanishing points from parallel structure →can be an automatic method based on the line detection algorithm (line fitting, as well)

Picture from OReilly Learning OpenCV

Homograhy—Applications

- Solution for finding **H**—cont.



Pattern/object recognition

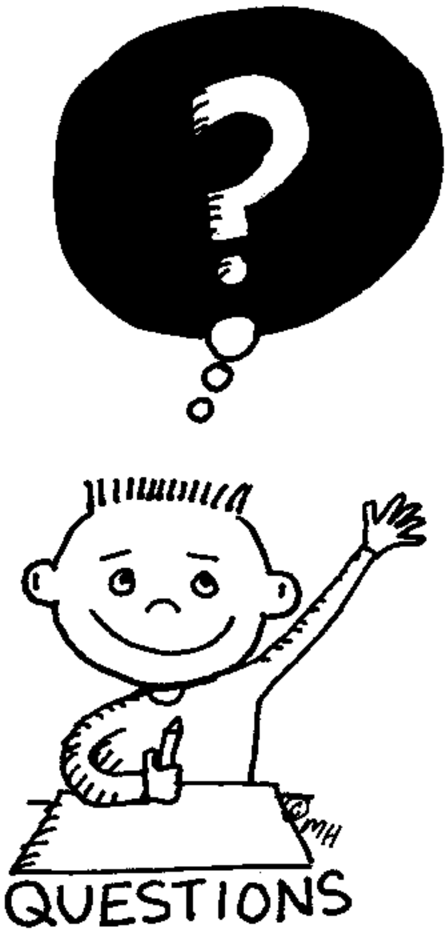
For a very simple example, the checkerboard has the fixed size and parallel grids. If you have a algorithm to determine the corners on the checkerboard, it will be easy to have correspondences for finding **H**.

Picture from OReilly Learning OpenCV

Homograhy—Applications

- Dynamic seethroughs (homography應用)

P. Barnum, Y. Sheikh, A. Datta, and T. Kanade, "Dynamic seethroughs : synthesizing hidden views of moving objects," IEEE Int. Symp. Mix. Augment. Real., pp. 111–114, Oct. 2009.



色彩與照明科技研究所
Graduate Institute of Color and Illumination Technology

Find us on: facebook

Figure from : http://www.gabankruptcylawyersnetwork.com/files/2014/08/question.gif