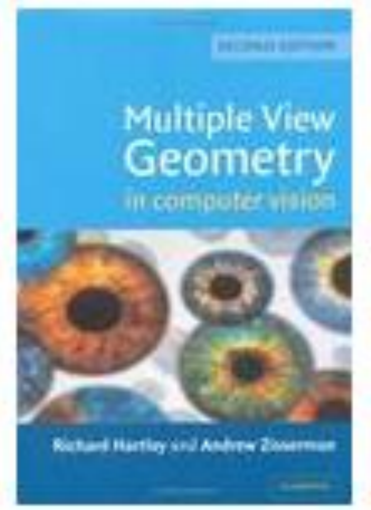
電腦視覺與應用
Computer Vision and Applications

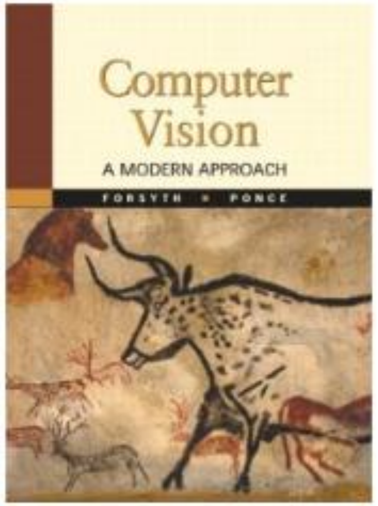
Lecture06-2-Two-views geometry-case study

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

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.

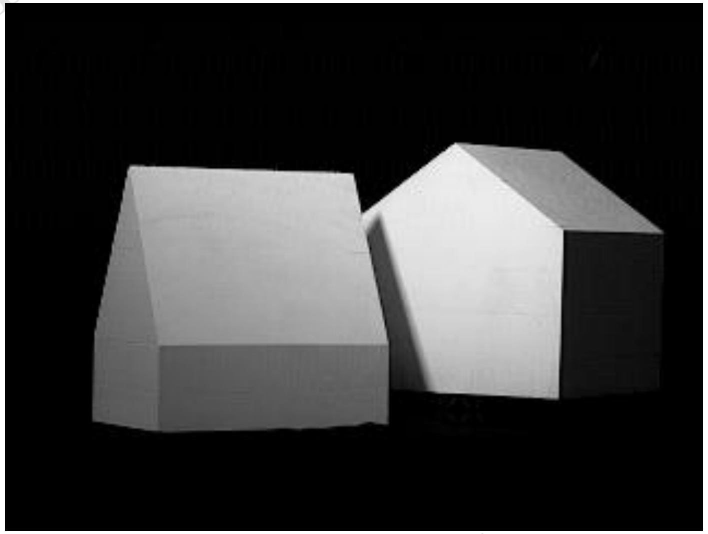


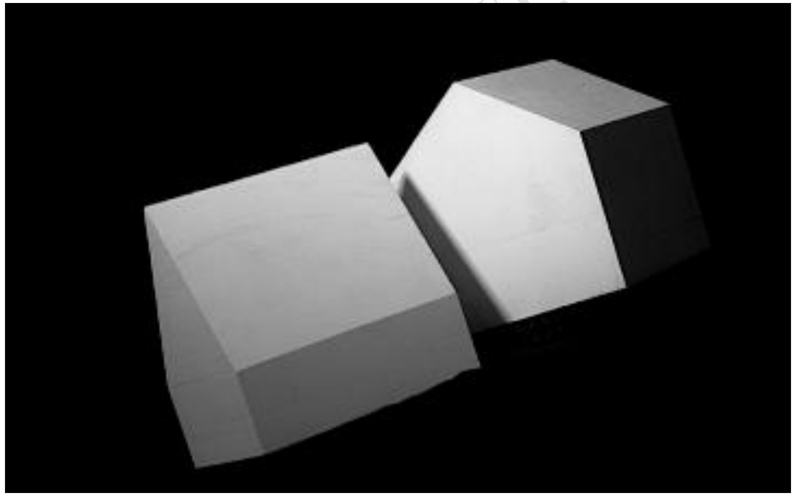
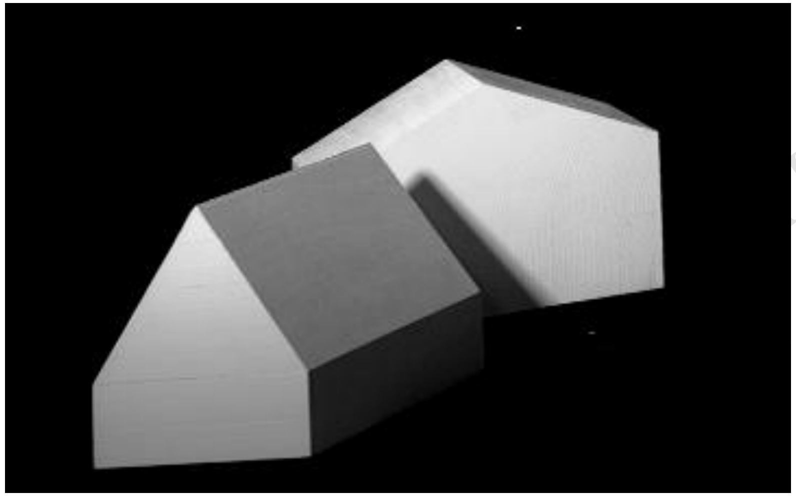


Keyword list

- Rectified, Rectification.
- Stereo image, Stereo camera
- Parallel-configured stereo camera, Converged stereo camera
- Epipolar line, Epipole
- Pyramids

Stereo-image



Condition for rectification

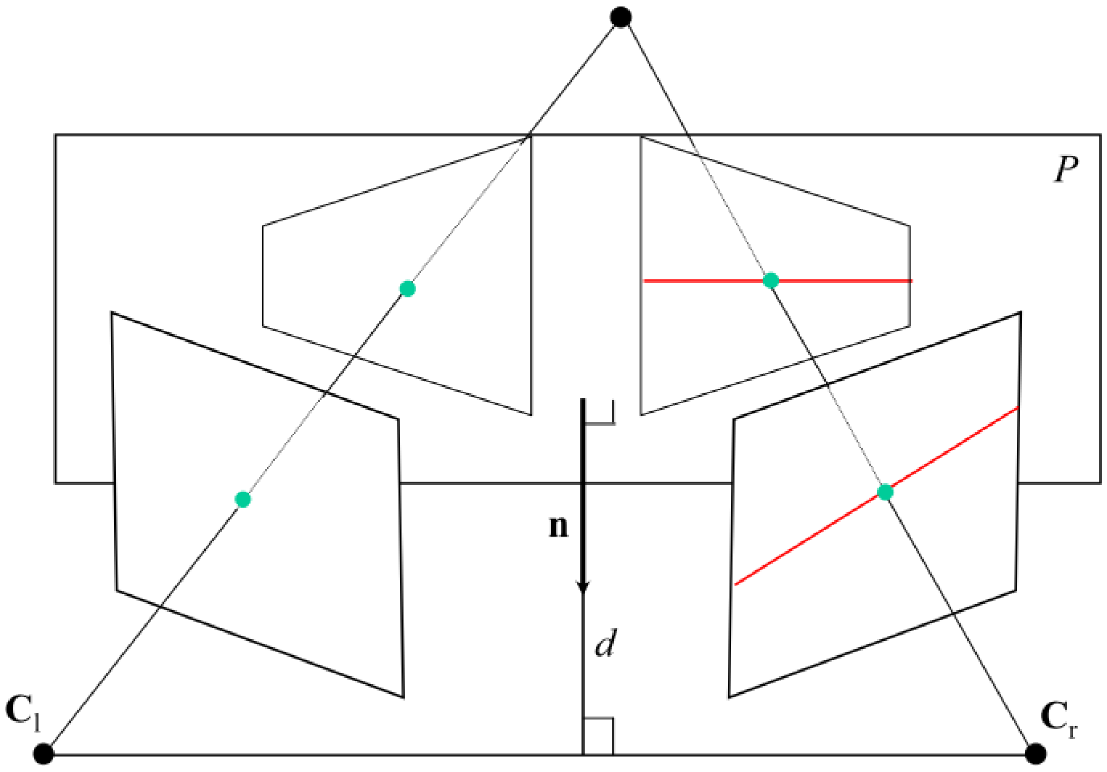
1. Calibrated stereo camera
   - Already know **K** and [**R**|**t**] for both cameras (as well as known **F**)

2. Non-calibrated stereo camera
   - Do not known any information for both cameras, the given input is only "two images"

Note: $\mathbf{K}_L$ and $\mathbf{K}_R$ can be different. (as well as different resolutions)

Rectification for stereo-image

- To projectively transfer images into a specific position



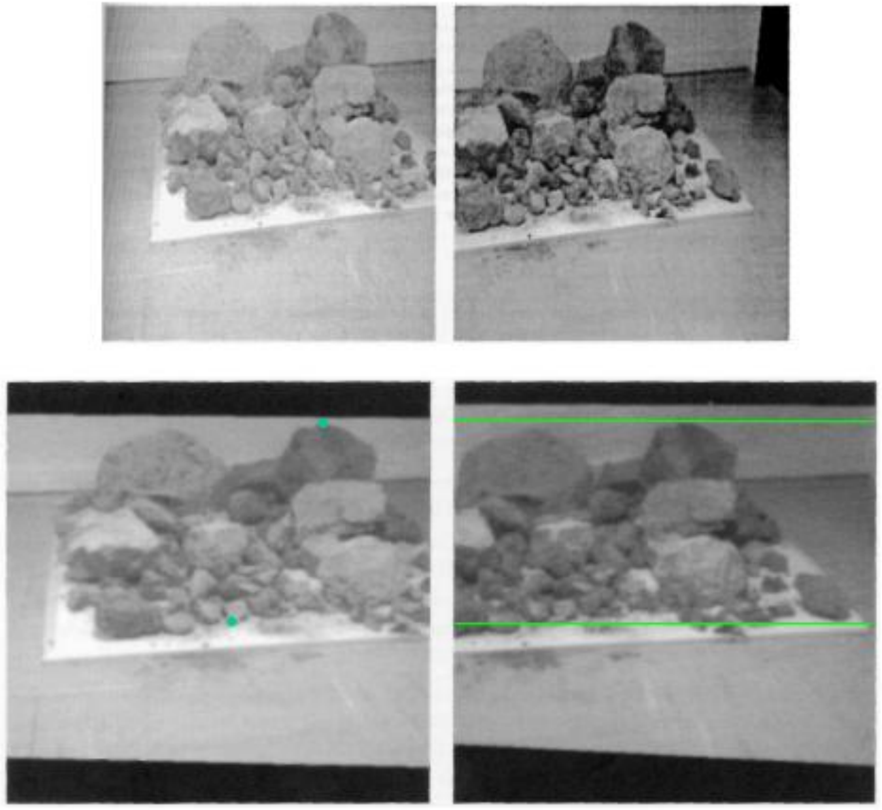
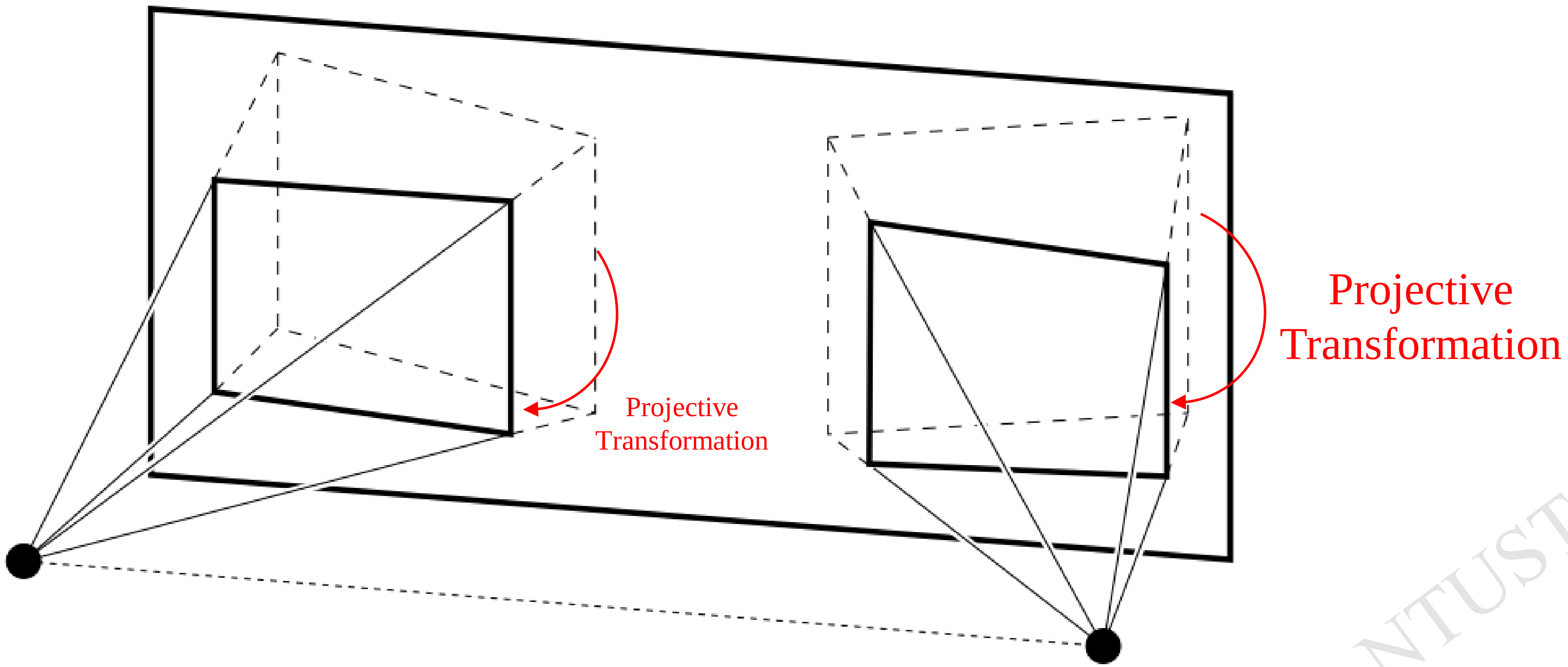


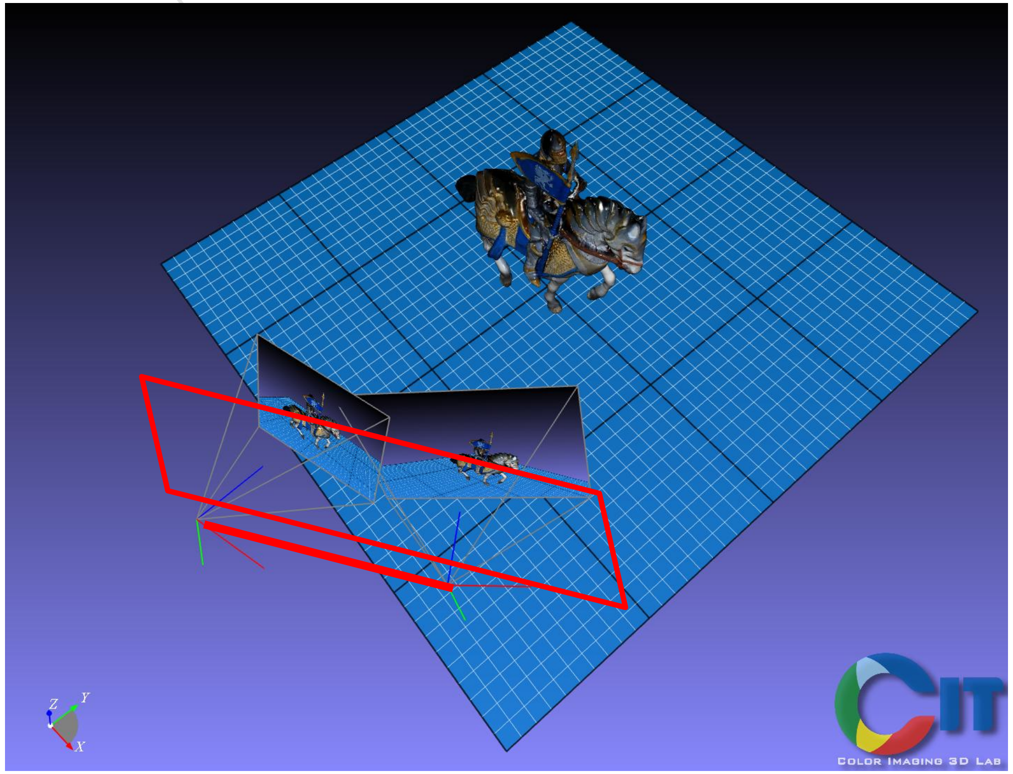
Image rectification



NOTE! This projective transformation is so called Homography!
There is NO unique solution.

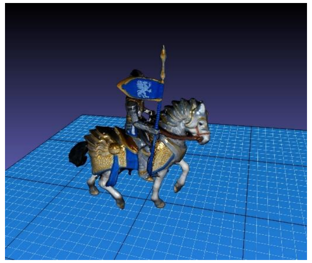
Rectification for **calibrated** stereo camera

- Note: they can have different **K** and image-width & height.



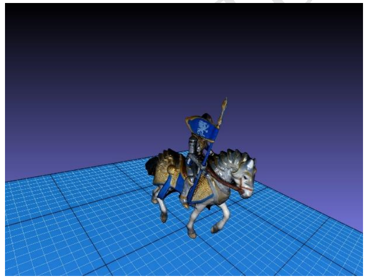
Example: Left camera
K
1290.669800 0.000000 688.000000
0.000000 1290.669922 586.000000
0.000000 0.000000 1.000000
[**R**|**t**]
0.947773 0.311687 -0.067663 -10.233355
0.079612 -0.436617 -0.896117 71.688828
-0.308851 0.843929 -0.438629 298.286102

Image size: 1376 x 1172

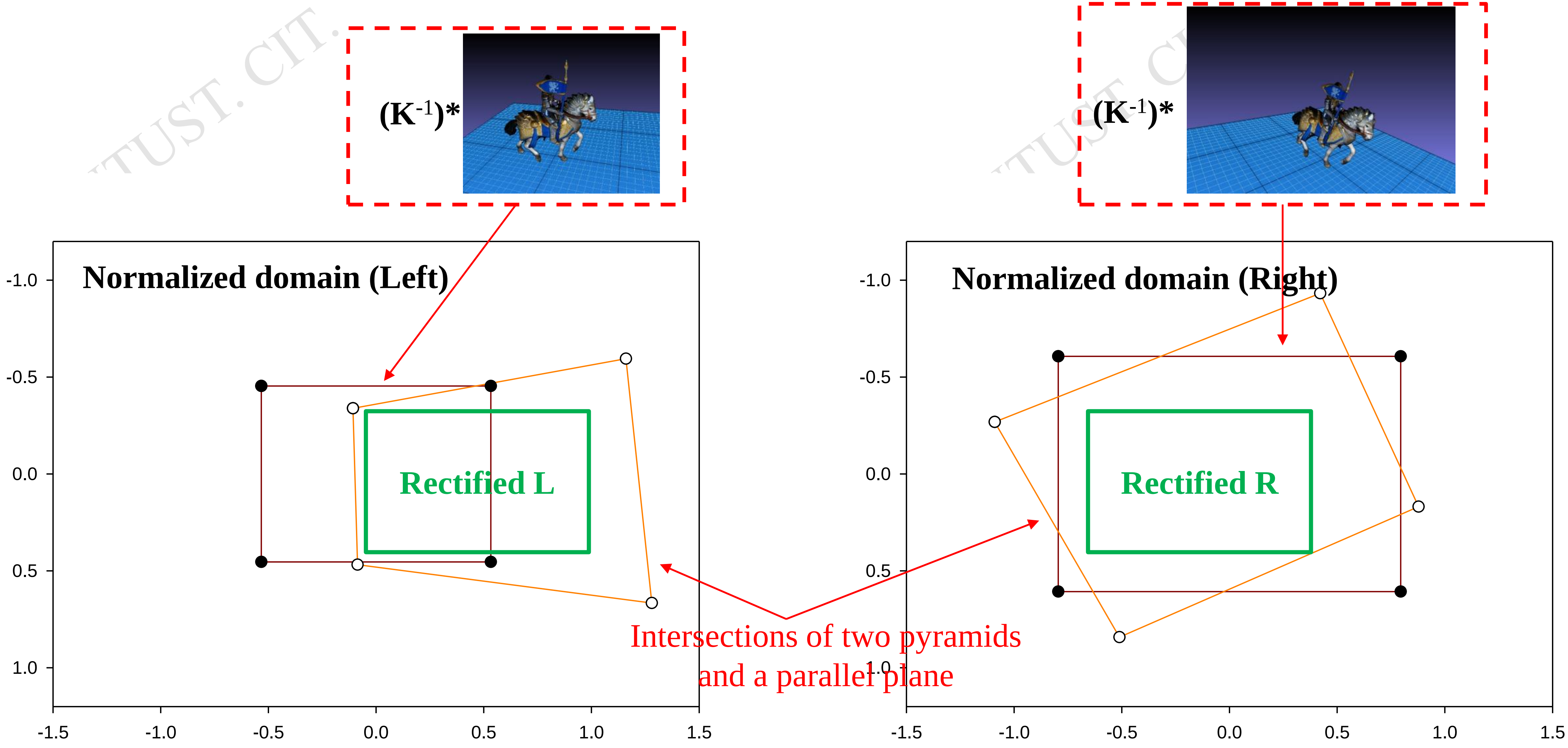


Right camera
K
916.034973 0.000000 728.000000
0.000000 916.034912 556.000000
0.000000 0.000000 1.000000
[**R**|**t**]
0.660273 0.738268 0.137843 3.202334
0.302509 -0.093444 -0.948554 104.052399
-0.687406 0.668003 -0.285034 270.232483

Image size: 1456 x 1112



Rectification for **calibrated** stereo camera

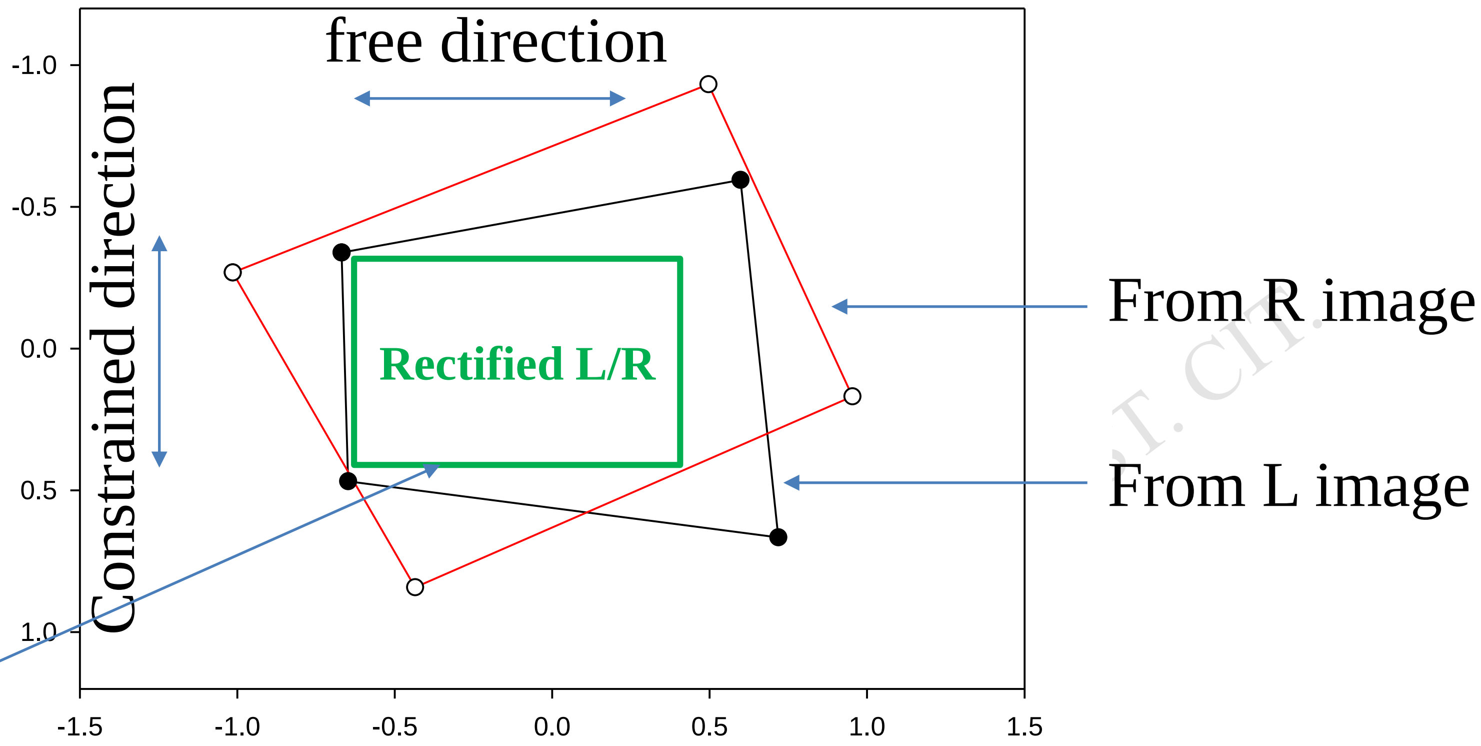


Rectification for **Non-calibrated** stereo camera

Need to define Final Iimage Size (assuem similar to input)

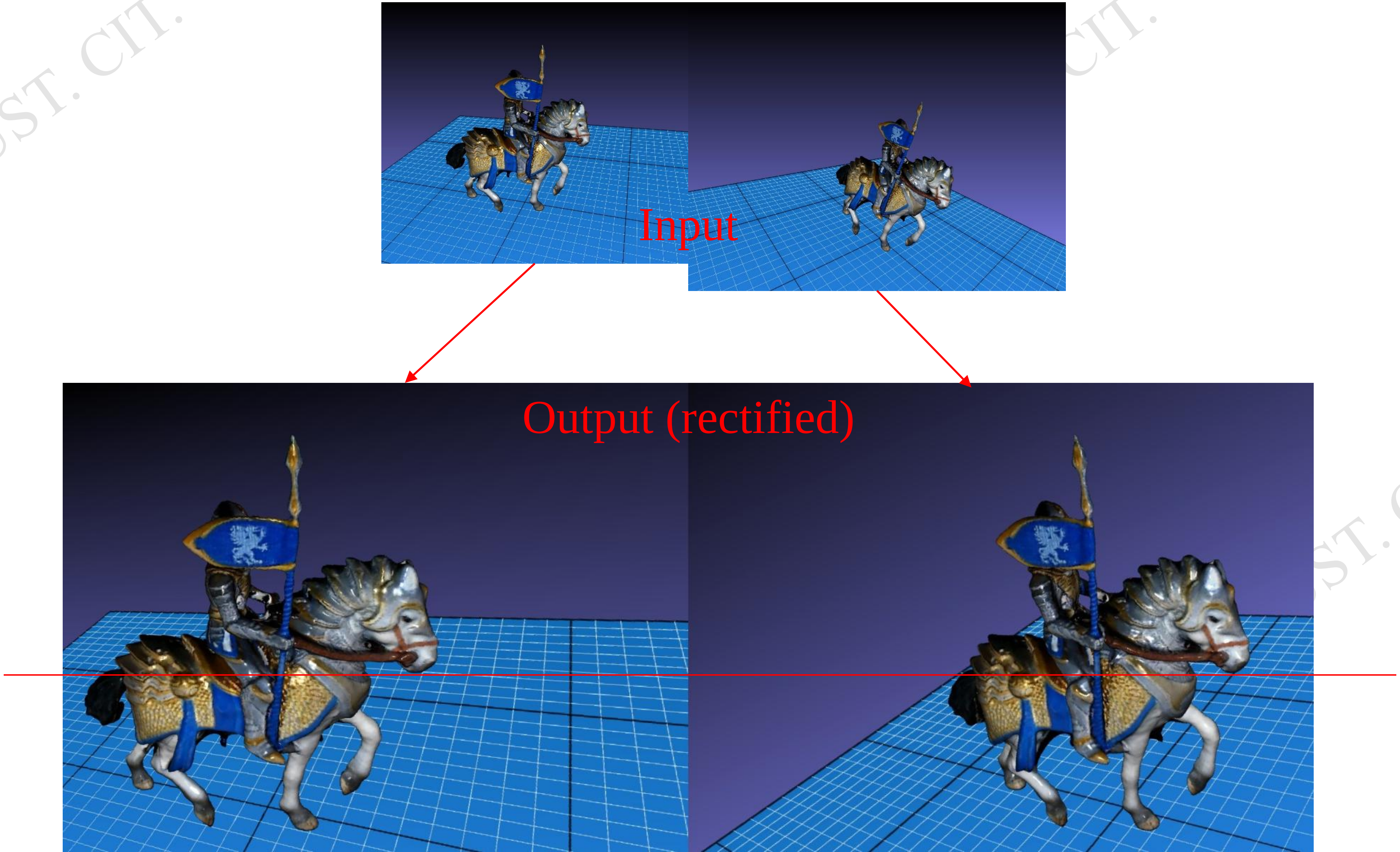
Here,we define a ratrio of 4:3

Though, the input images have different resolution, output should be the same (ex. 1280x960)



Maximum-Area Rectangle (be better)

Rectification for **Non-calibrated** stereo camera



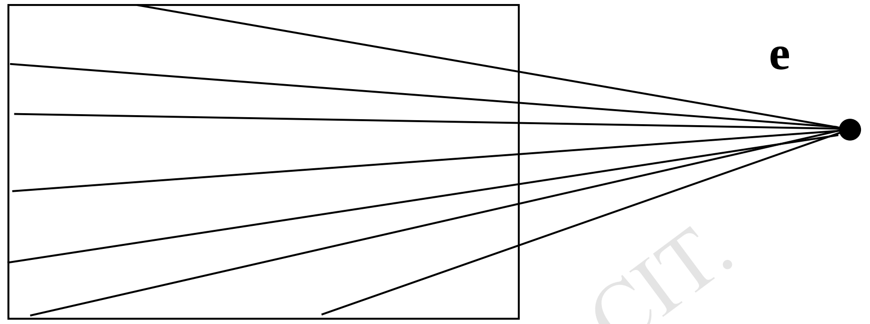
Rectification from **Non-calibrated** stereo camera

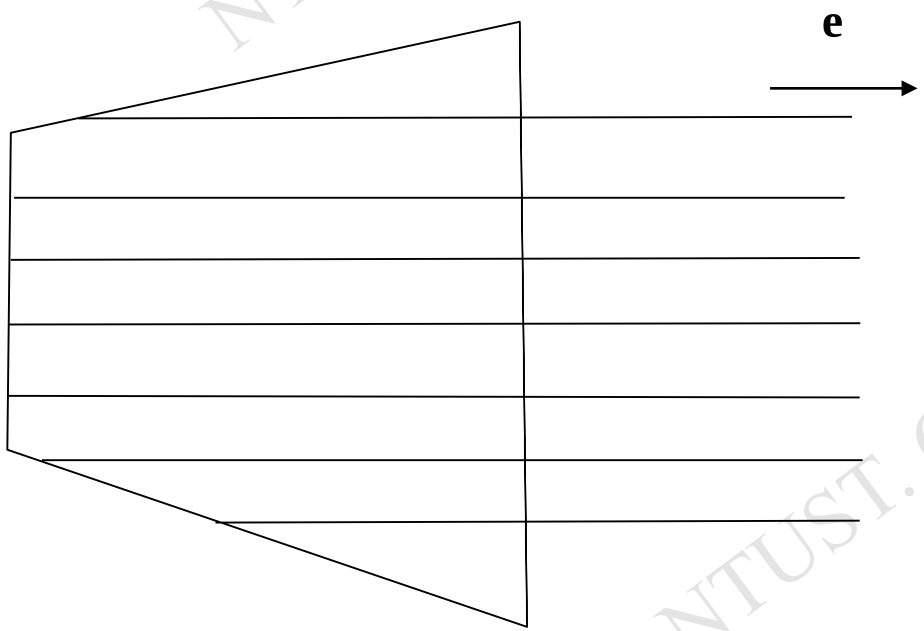
- This is the most common stituation to set-up for stereo camera without pre-calibration.
- NOTE: there is NO unique solution. But you may have a better streatage, ie. Minization for disparity, re-arrage the disparity distribution for visual perception.

- What we apply on images is always a type of "homography" matrix.

Image rectification (non-calibrated camera)

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



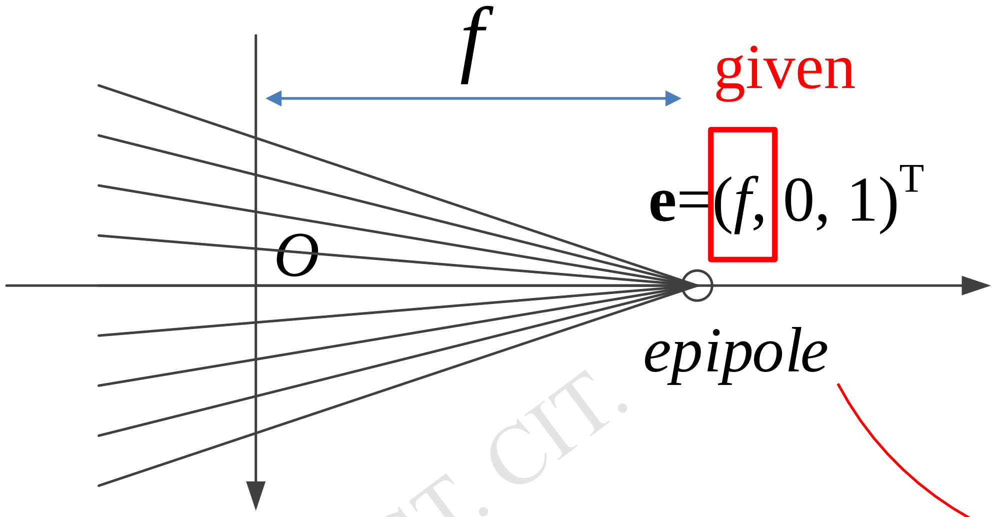


map epipole **e** to $(1,0,0)^T$

try to minimize image distortion

Image rectification—solution

- To transfer epipole to the point at infinity



given

$\mathbf{e}=(\boxed{f}, 0, 1)^T$

outcome

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

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

point at infinity

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

Note! G is a homography matrix

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.

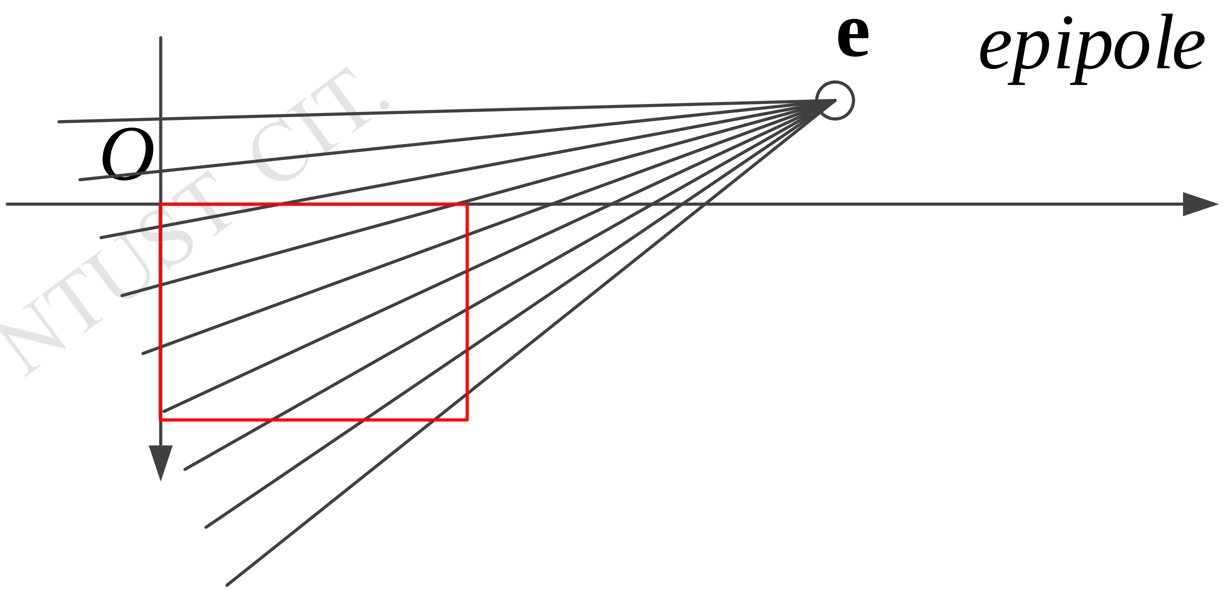


Image rectification—solution, cont.

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

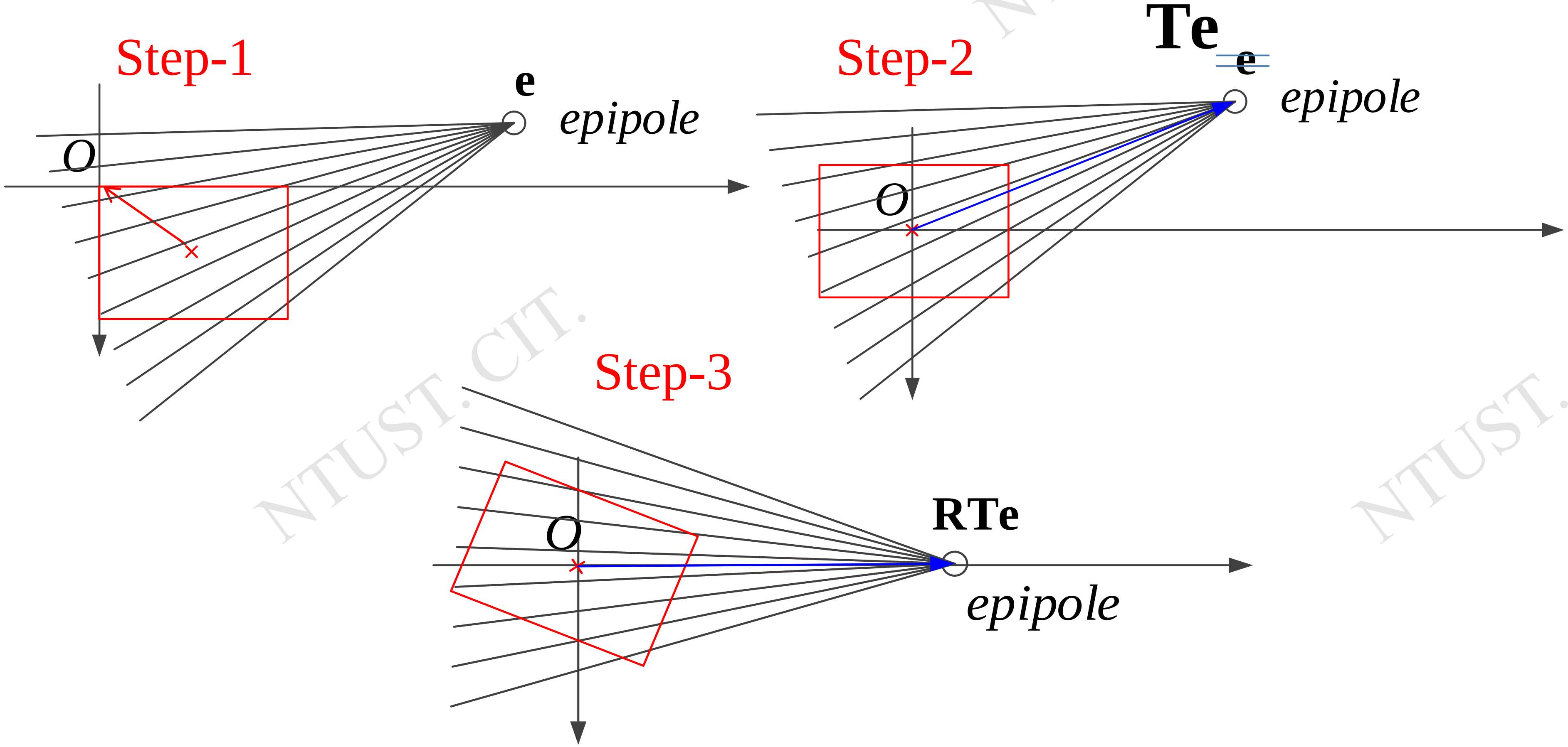
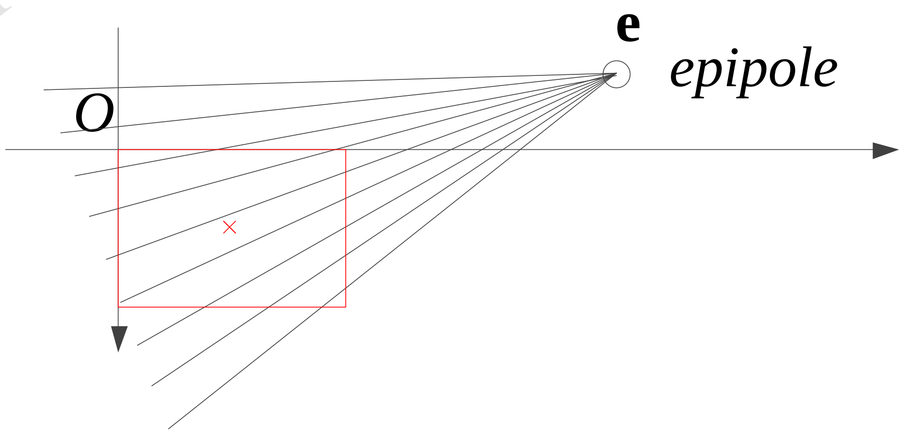


Image rectification—solution, cont.



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

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

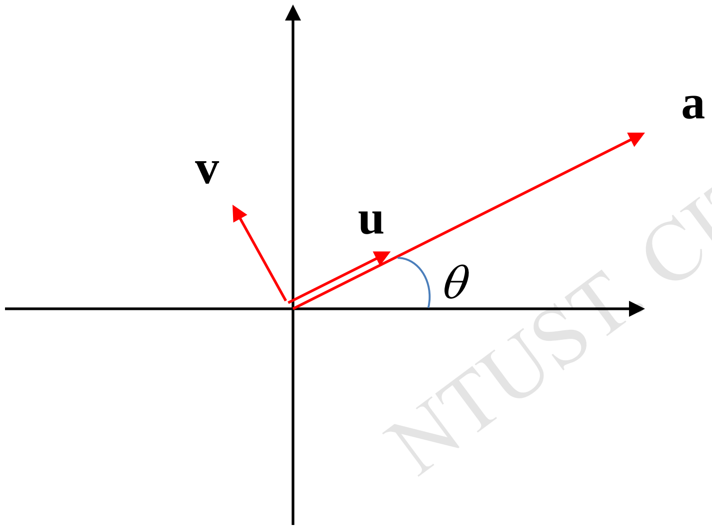
point at infinity

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

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

Image rectification—solution, cont.

- How to determine a 2D rotation matrix?
  - Method 1: find inclined angle, then build a matrix from formula
  - Method 2: build a matrix from two basises



Method 1

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

Method 2

u is an unit vector along **a**

$$\mathbf{R}=\begin{bmatrix}\boxed{\mathbf{u}} & \boxed{\mathbf{v}} & \begin{matrix}0\\0\end{matrix}\\ 0 & 0 & 1\end{bmatrix}^{-1}=\begin{bmatrix}\boxed{\quad\mathbf{u}\quad} & 0\\ \boxed{\quad\mathbf{v}\quad} & 0\\ 0\quad 0 & 1\end{bmatrix}$$

Note: it will be always positive for *x* component

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:

- Determine one **F** from two image
- Determine **e** (use cross product of two epipolar lines, line eqs: $\mathbf{l}=\mathbf{F}^T\mathbf{x}'$)
- Determine **T** (of course, you know the image resolution. use its center)
- Determine **R** (you already have $\mathbf{T}_e$, rotation angle should be $-\tan^{-1}\frac{y}{x}$

- Then, get f from **RTe**.
- 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.

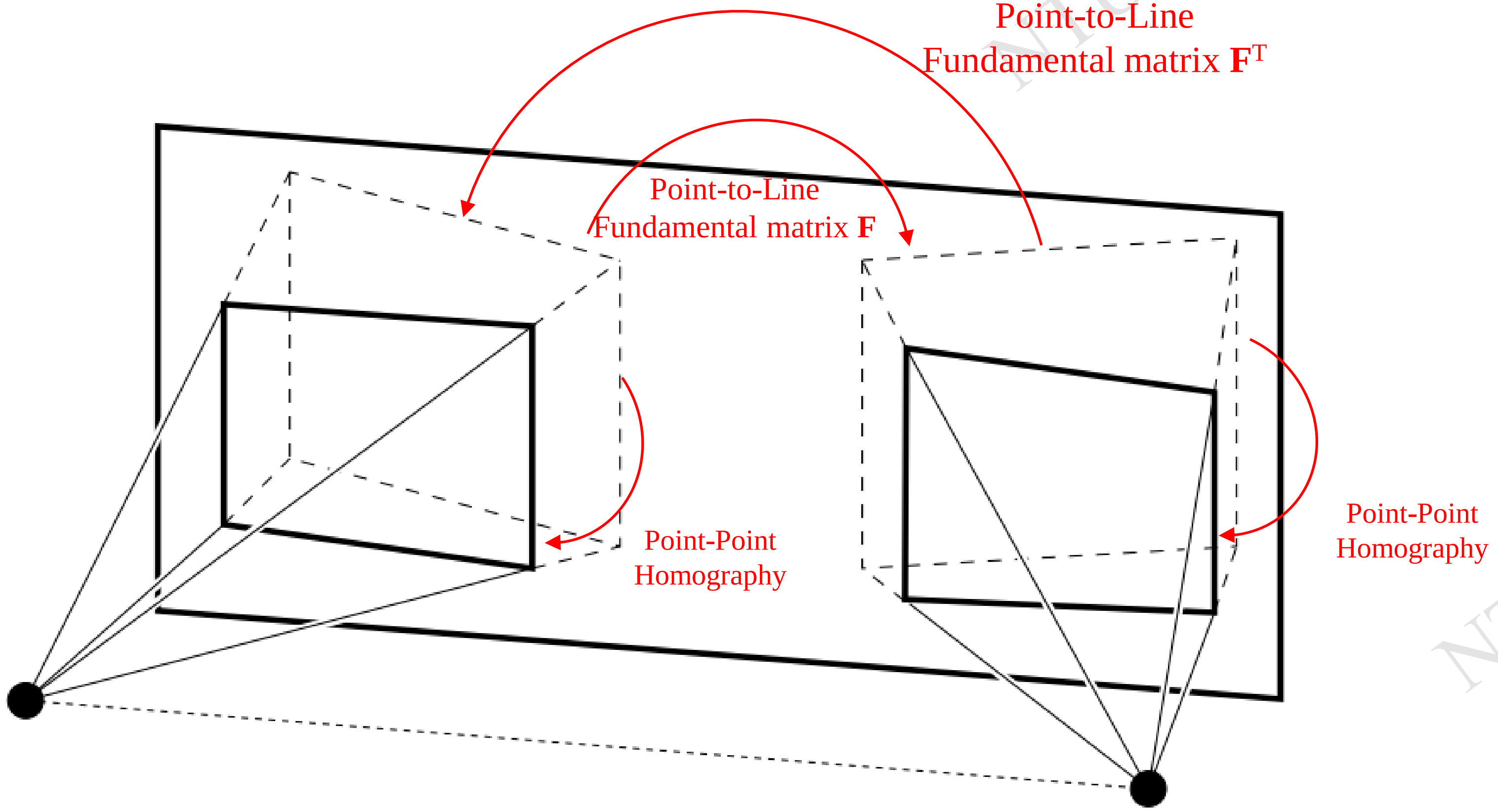
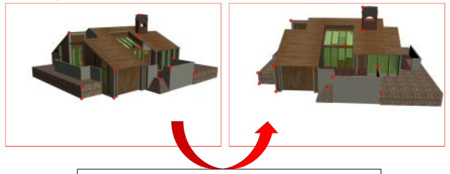


Image rectification—example (Method-1)

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

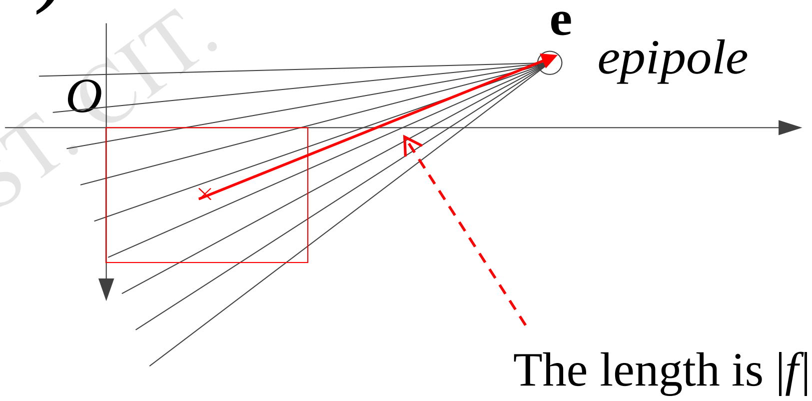
For 1st image:

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

T*e=
-4449.085693
1058.432373
1.0

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

Rotation angle= 13.38°



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

(Note the rotation direction)

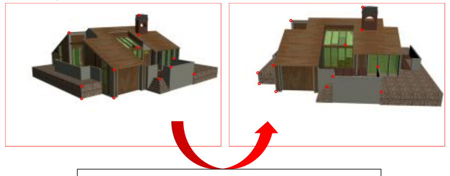
$$\mathbf{RTe}=\boxed{[-4573.3}\quad 0\quad 1]^{T}$$

For LEFT image

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

Image rectification—example (method-2)

- After you get epipole…



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

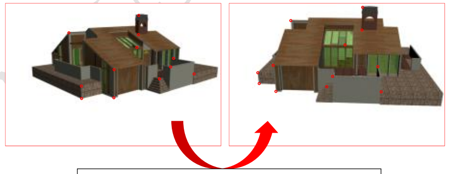
e'=
-552.206970
217.436905
1.000000

For LEFT image

```
--> a = [ epipole_l(1)-360 , epipole_l(2)-240]'
 a  =

  -4428.9946
   1053.3853

--> u = a / norm(a)
 u  =

  -0.9728625
   0.2313841

--> v = [-u(2) u(1)]'
 v  =

  -0.2313841
  -0.9728625

--> R_l =[u' 0; v' 0; 0 0 1]
 R_l  =

  -0.9728625   0.2313841   0.
  -0.2313841  -0.9728625   0.
   0.          0.          1.
```

```
> if (a(1)<0)
>            R_l = [-1 0 0; 0 -1 0; 0 0 1]*R_l
> end
 R_l  =
   0.9728625  -0.2313841   0.
   0.2313841   0.9728625   0.
   0.          0.          1.

--> f_l = R_l*T_l*epipole_l
 f_l  =

  -4552.5393
   0.
   1.

--> G_l = [ 1 0 0; 0 1 0; -1/f_l(1) 0 1]
 G_l  =
   1.          0.   0.
   0.          1.   0.
   0.0002197   0.   1.

--> H_l = G_l*R_l*T_l
 H_l  =

   0.9728625  -0.2313841  -294.6983
   0.2313841   0.9728625  -316.78528
   0.0002137  -0.0000508   0.9352673
```

Image rectification—example, cont. (Method-1)



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

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

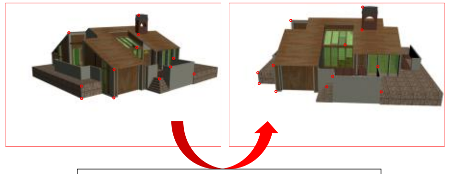
(Note the rotation direction)

For RIGHT image

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

Image rectification—example (method-2)

- After you get epipole…



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

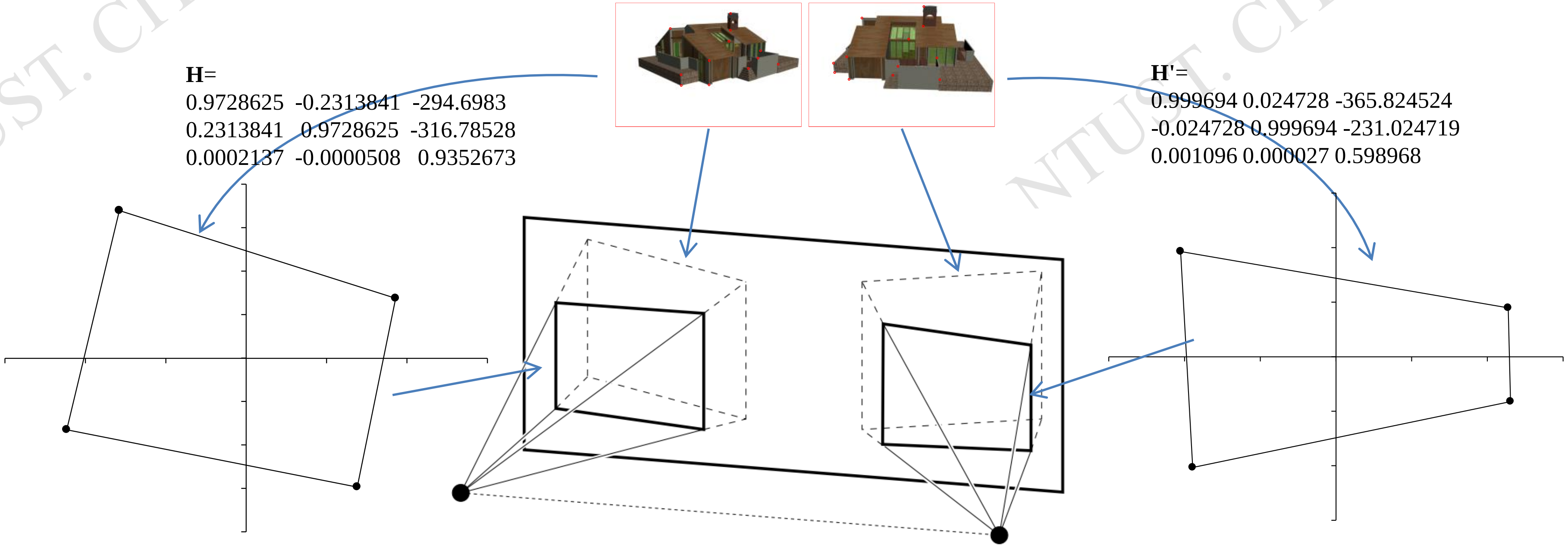
e=
-4089.085693
1298.432373
1.000000

e'=
-552.206970
217.436905
1.000000

For RIGHT image

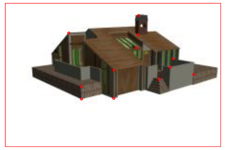
```
--> a = [ epipole_r(1)-360 , epipole_r(2)-240]'
 a  =

  -912.16183
  -22.672077

--> u = a / norm(a)
 u  =

  -0.9996912
  -0.0248476

--> v = [-u(2) u(1)]'
 v  =

   0.0248476
  -0.9996912

--> R_r =[u' 0; v' 0; 0 0 1]
 R_r  =

  -0.9996912  -0.0248476   0.
   0.0248476  -0.9996912   0.
   0.          0.          1.

> if (a(1)<0)
  >               R_r = [-1 0 0; 0 -1 0; 0 0 1]*R_r
  > end
 R_r  =

   0.9996912   0.0248476   0.
  -0.0248476   0.9996912   0.
   0.          0.          1.

--> f_r = R_r*T_r*epipole_r
 f_r  =

  -912.44355
  -2.842D-14
   1.

--> G_r = [ 1 0 0; 0 1 0; -1/f_r(1) 0 1]
 G_r  =

   1.         0.   0.
   0.         1.   0.
   0.001096   0.   1.

--> H_r = G_r*R_r*T_r
 H_r  =

   0.9996912   0.0248476  -365.85229
  -0.0248476   0.9996912  -230.98075
   0.0010956   0.0000272   0.5990412
```

Image rectification—example, cont.

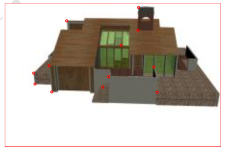


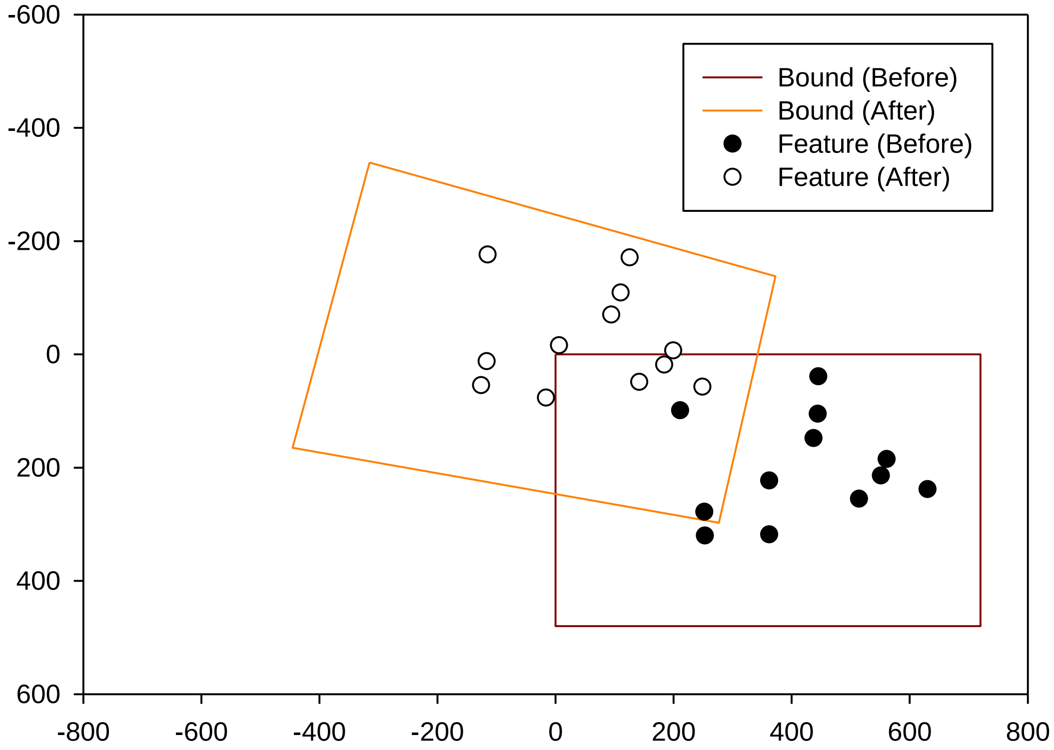
After rectification adjustment, two problems remain
1.Correspondences in two image may have disparity on *y* direction.
2.Pixel coordinates may not fall in positive region. NOTE: What you can draw now is only around quarter of an image.

Image rectification—example, cont.



NTUST. CIT.





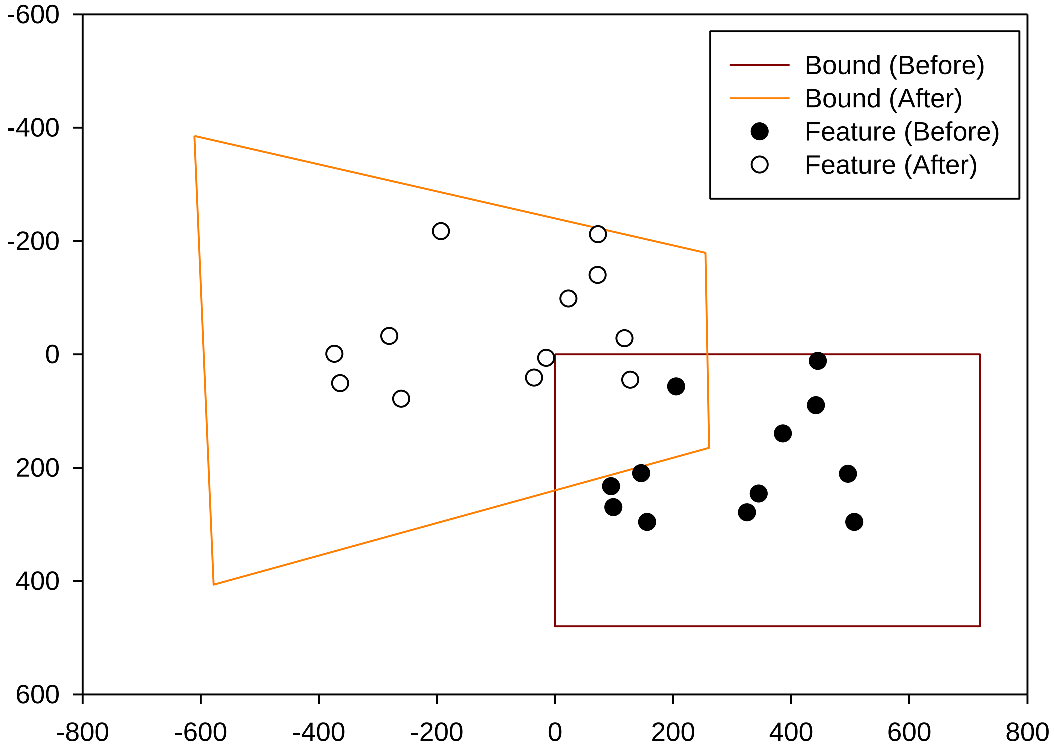
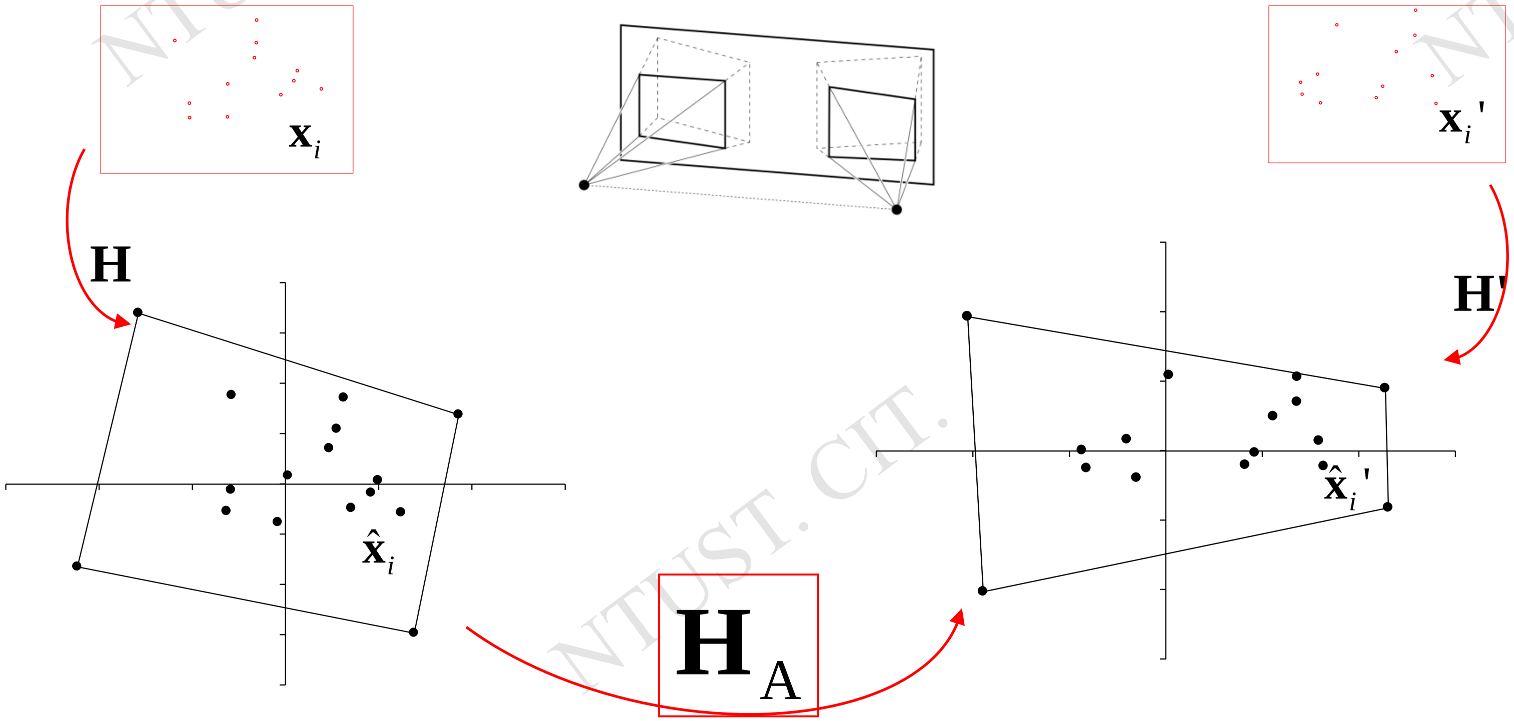


Image rectification—example, cont.

- Minimize the horizontal & vertical disparity (minimize dist.)



$$\mathbf{H}_{\mathrm{A}} = \begin{bmatrix} a & b & c \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$ Horizontal disparity

$$\mathbf{H}_{\mathrm{A}} = \begin{bmatrix} 1 & b & 0 \\ 0 & d & e \\ 0 & 0 & 1 \end{bmatrix}$$ Vertical disparity

$$\mathbf{H}_{\mathrm{A}} = \begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & 1 \end{bmatrix}$$ Both direction

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

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

Image rectification—example, cont.

- Minimize the horizontal & vertical disparity (minimize dist.)

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

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

the unknown to be optimized is (a, b, c, d, e)

So, minimization terms are reduced…

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

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.

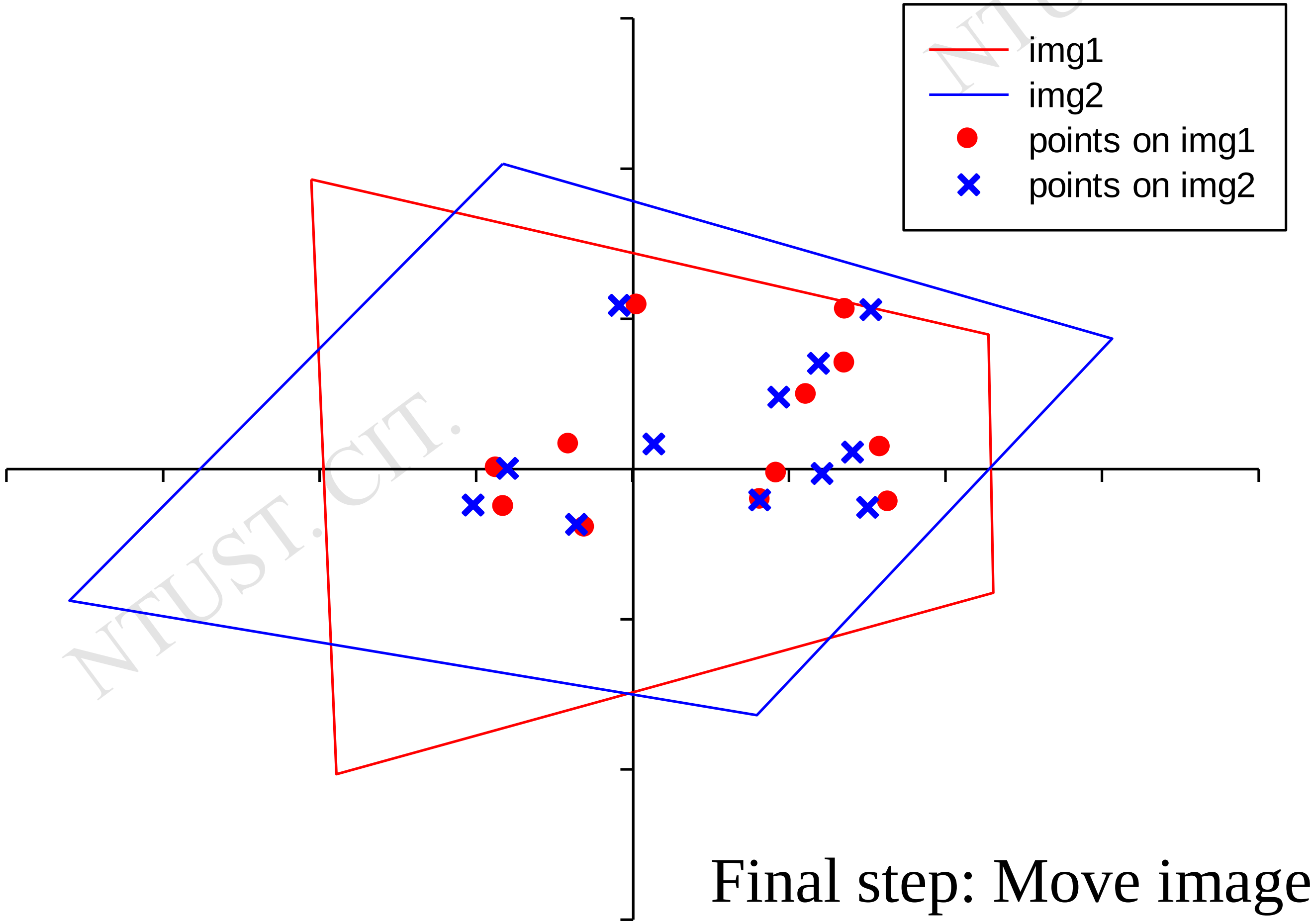
- Minimize the vertical disparity, as well as horizontal distance.—cont.

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

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? NO
- You also can optimize “the distribution” of disparity for more purposes.

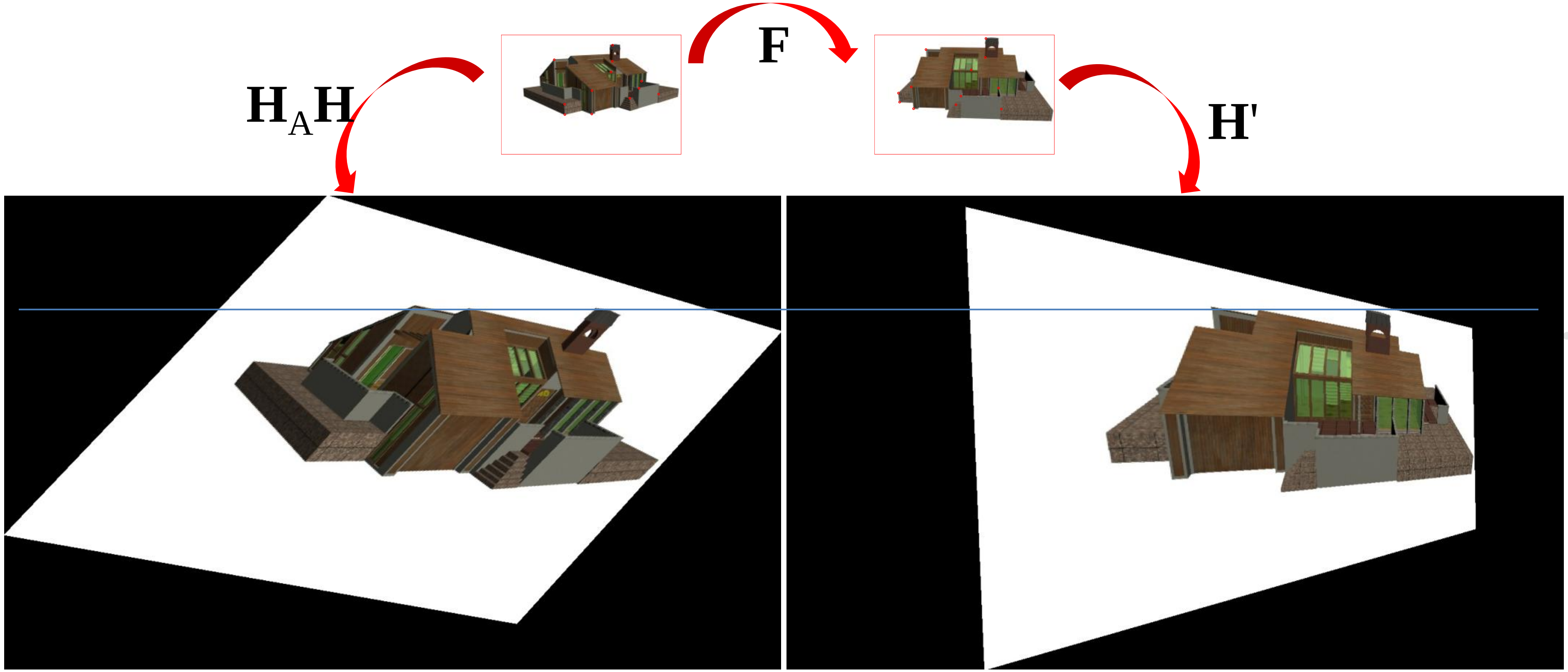


Image rectification—example, cont.

- Need to define **new K** for both cameras (they share the same **K**)
- That means you already define resolution for new images. →Normally, we keep as similar to the original (says 720x480 in this example) as possible.
- Next question is: Crop black region or not? And how to determine the maximum-area rectangle in an overlap region?

Image rectification—openCV

- StereoRectifyUncalibrated

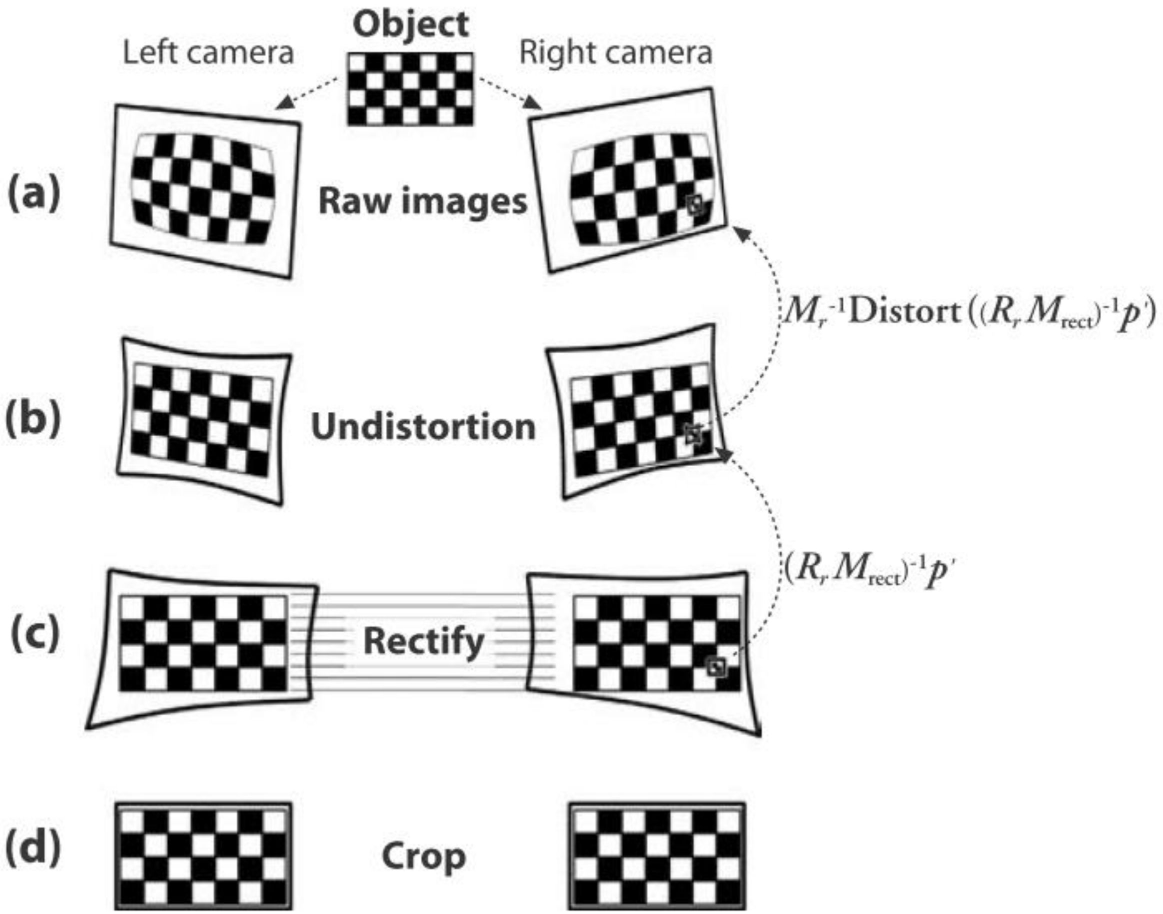


Image rectification—openCV

- 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)

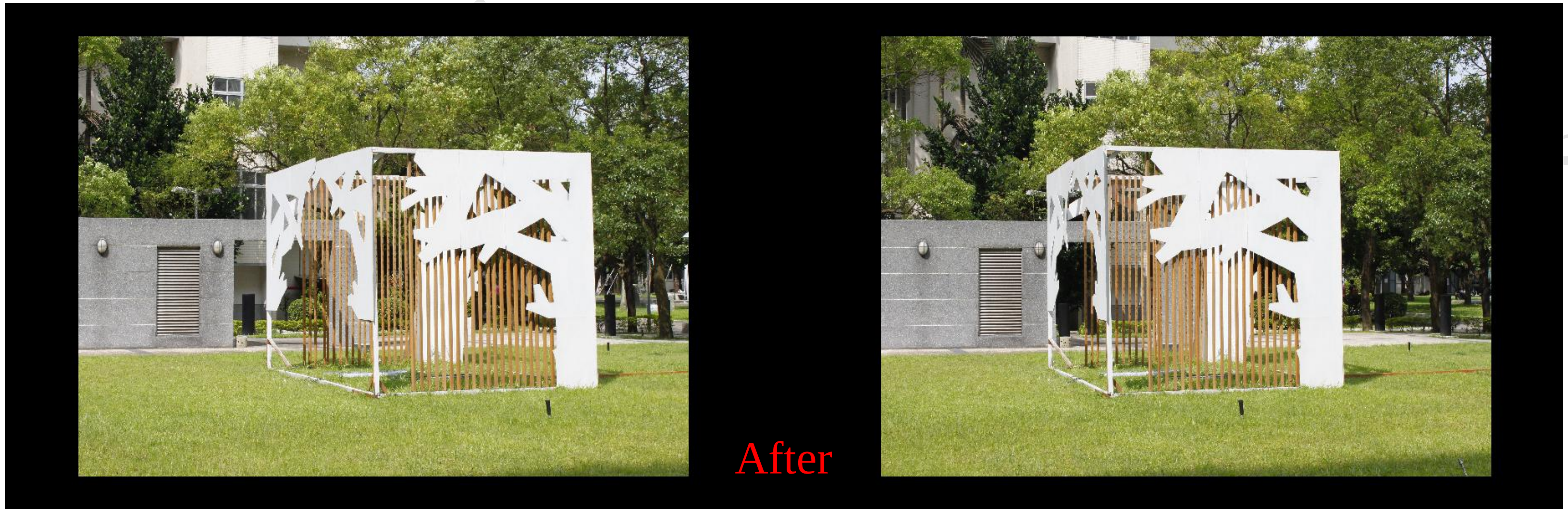
F → Fundamental Matrix (input)

imageSize → (input)

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

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

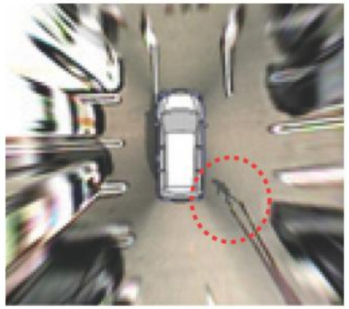
Reference software (stereophoto maker)



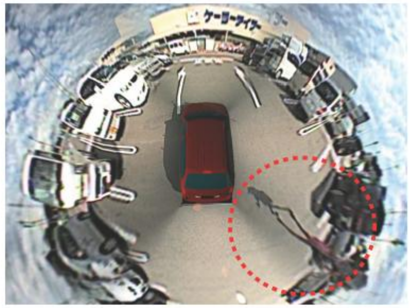
http://stereo.jpn.org/eng/stphmkr/

Homography—Applications

- Intelligent automobile

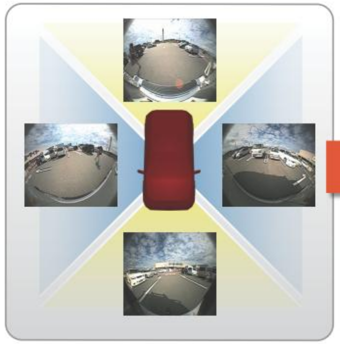


Conventional technology-based image, vehicles and people are not visible.

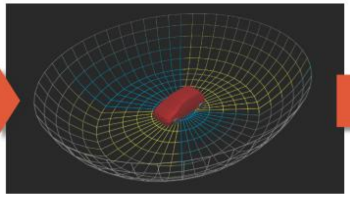


Sample view using Fujitsu Laboratories' new technology perspective from above-rear (pedestrian is visible)

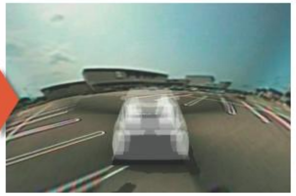
Sample view using Fujitsu's new technology, perspective from front facing vehicle (rearview pedestrian is visible)



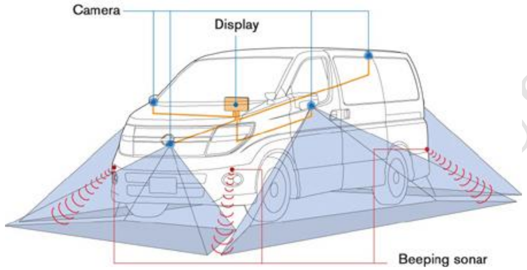
4 cameras capture video images of 4 different views (perspectives)



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

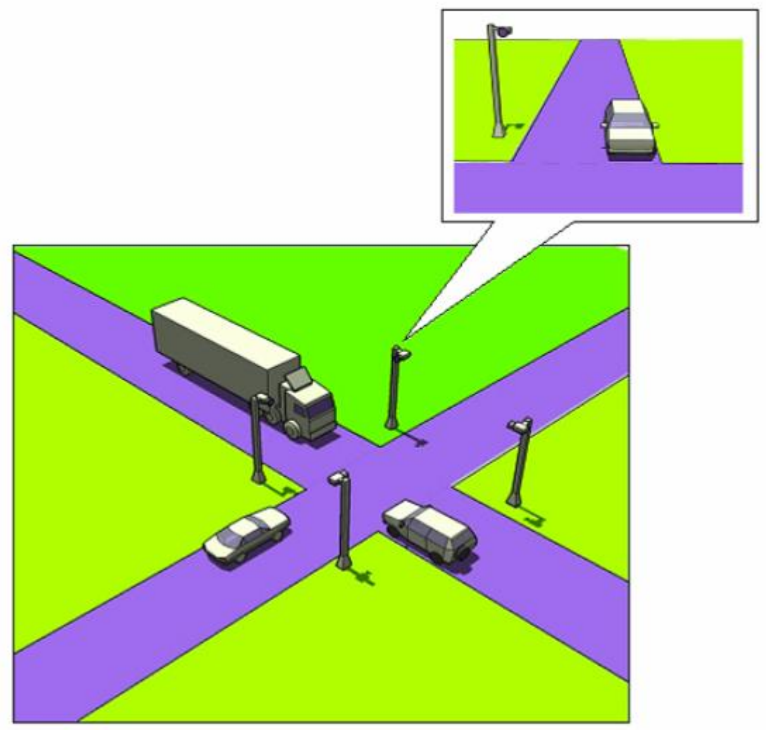
FUJITSU
http://www.fujitsu.com/downloads/MICRO/fma/pdf/360_OmniView_AppNote.pdf

Homography—Applications

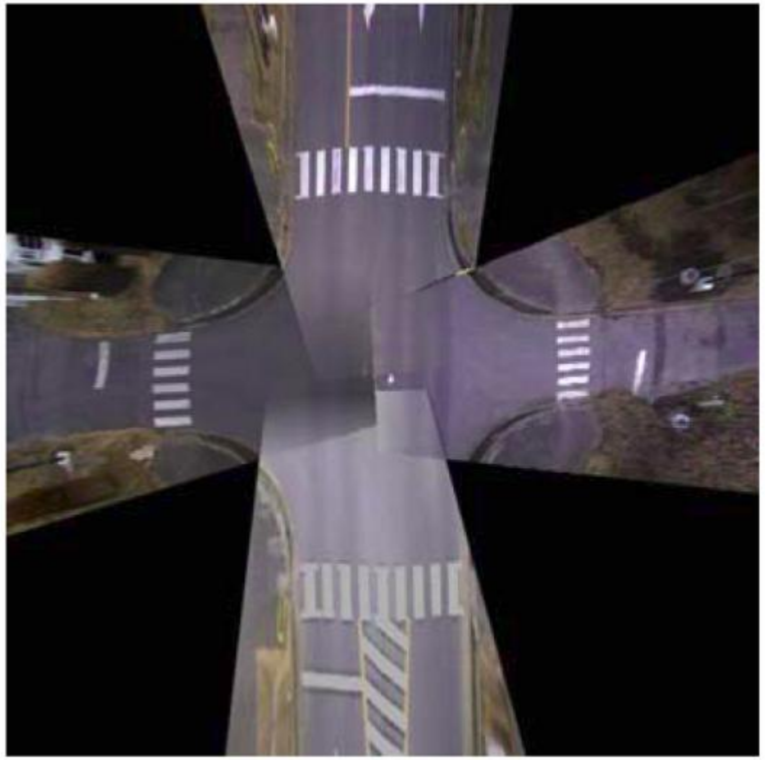
- Intelligent automobile—cont.

Homography—Applications

- Multi-camera surveillance
- Internet of things (IoT)



筑波大學(JP)



筑波大學(JP)

Homography—Applications

- Application scenarios

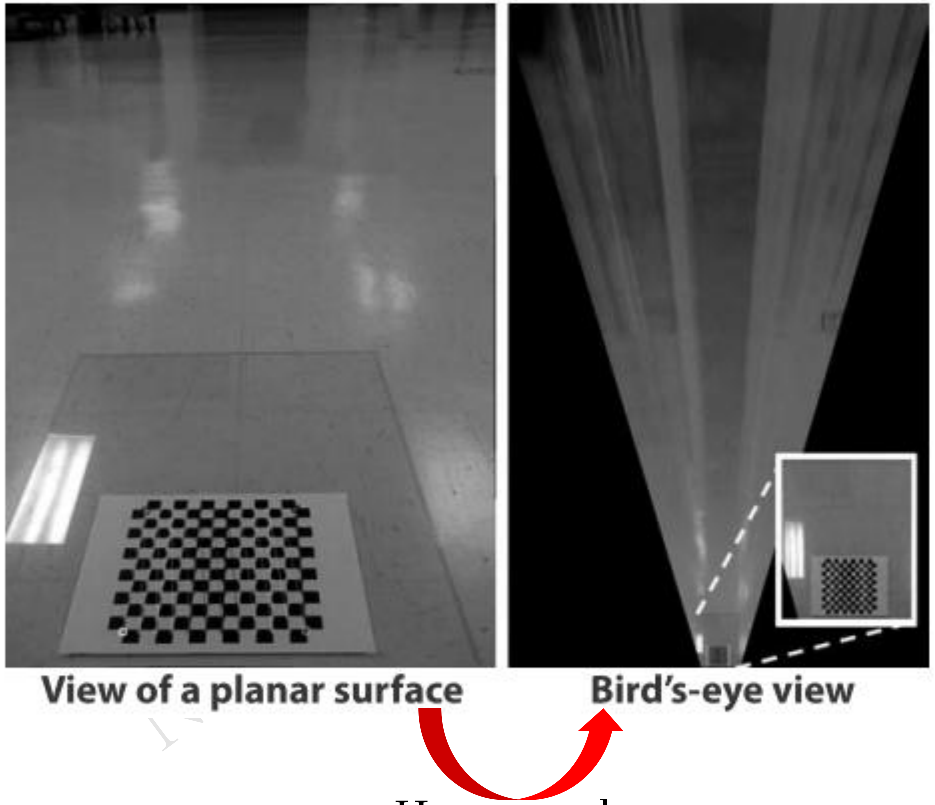
Image-Undistortion → **Homography** → **Image stitching** →

- 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.

Homography—Applications

- Once again, define your problem first!



Homography

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

Homography—Applications

- Image stitching (auto-stitching)

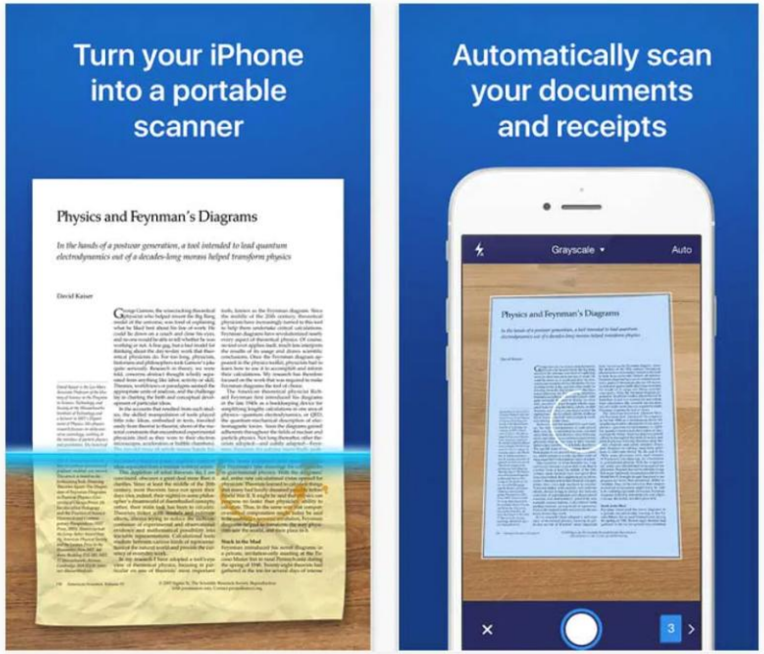
25 of 57 images aligned

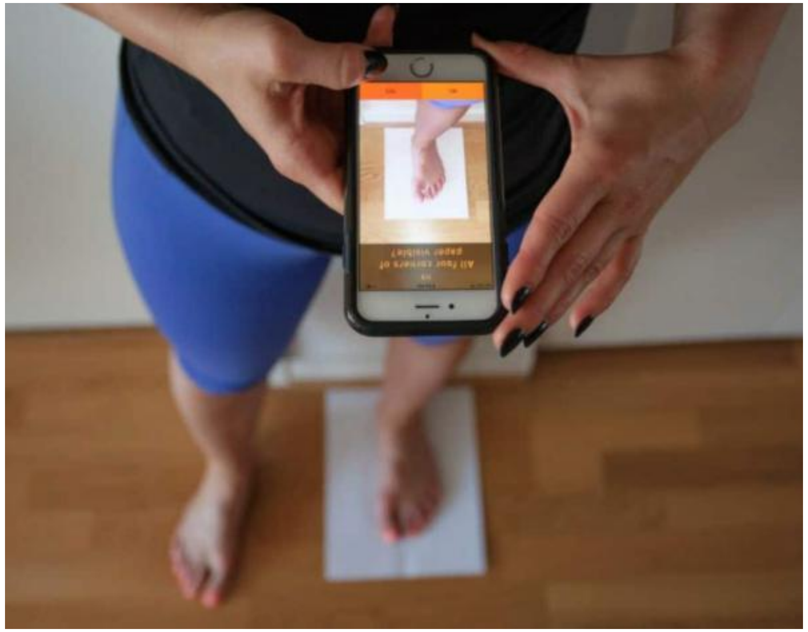
All 57 images aligned

http://matthewalunbrown.com/autostitch/autostitch.html

Homography—Applications

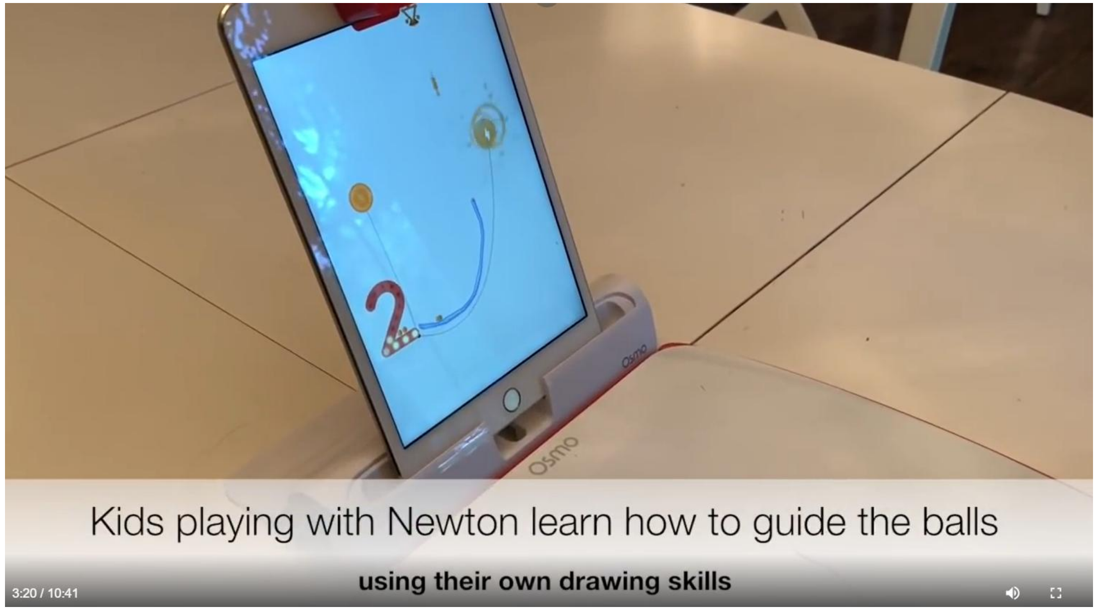
- Document scan App (Scan documents to PDFs)
- Foot measuremet App

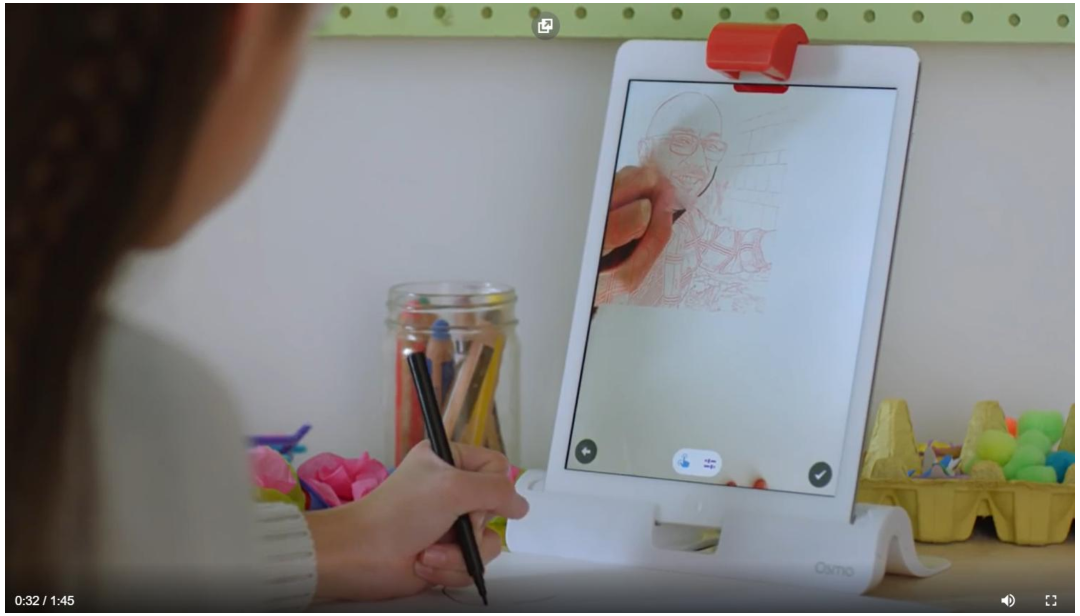




Homography—Applications

- OSMO Kit: Interactive game





https://www.youtube.com/watch?v=0upQlA6K5YI

Homograhy—Applications

- Dynamic see through (homography)

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.

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

Find us on: facebook