Thị giác máy tính

Bài 7: 3D Vision

1

Nội dung

- Giới thiệu
- Ước lượng 3D
- Hình học lưỡng cực (Epipolar geometry)
- Stereo calibration
- Stereo rectification
- Stereo matching

2

Tại sao cần nhiều views?

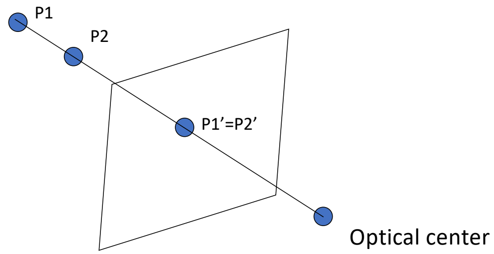
- 1 view: thông tin về độ sâu và cấu trúc bị mất 1 phần
  ➔ gây nhập nhằng, ko rõ ràng

Images from Lana Lazebnik

3

Tại sao cần nhiều views?

- 1 view: thông tin về độ sâu và cấu trúc bị mất 1 phần
  ➔ gây nhập nhằng, ko rõ ràng

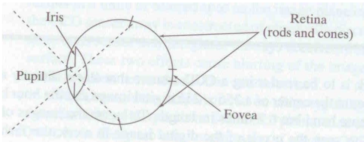


Kristen Grauman

4

Stereo

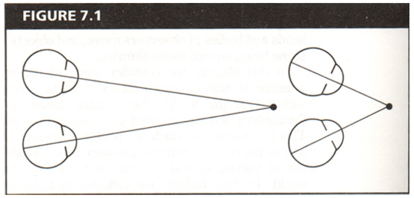
- Thị giác người:
  - 2 camera



Pupil/Iris – điều tiết ánh sáng

Retina – chứa bào thị giác, nơi ảnh được tái tạo

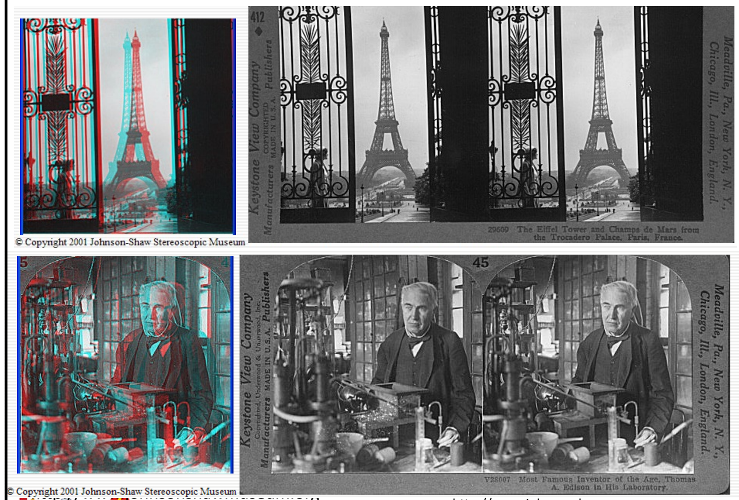
Fovea – nơi tập trung cao nhất các tế bào hình nón



Mắt người tập trung vào 1 điểm trong không gian để các hình ảnh tương ứng tạo ở trung tâm của fovea

From Bruce and Green, Visual Perception, Physiology, Psychology and Ecology

6



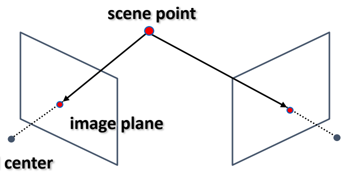
http://www.johnsonshawmuseum.org

11

Ước lượng hình dạng 3D

- Từ 1 view:
  - "Shape from X": shading, texture, focus, motion…
- **Sử dụng nhiều hơn 1 view (Stereo)**:
  - Tái tạo hình dạng từ 2 camera
  - Xây dựng hình ảnh 3D từ 2 hoặc nhiều ảnh từ các góc nhìn khác nhau

Ý tưởng chính:



Kristen Grauman

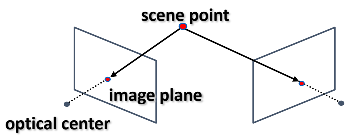
12

Ước lượng độ sâu với 2 views

Two cameras, simultaneous views

Single moving camera and static scene

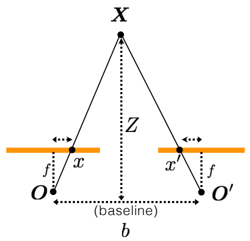
Thông tin cần có:
- Thông tin về vị trí camera ("calibration")
- Các điểm tương đồng giữa 2 ảnh



Kristen Grauman

13

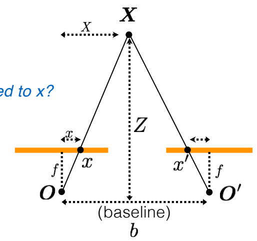
Simple stereo system



- 2 camera có trục quang học //, tham số camera đã biết (calibrated cameras):

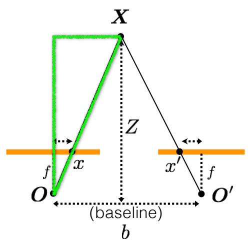
14

How is X related to x?



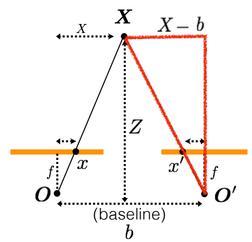
15

$$\frac{X}{Z} = \frac{x}{f}$$



16

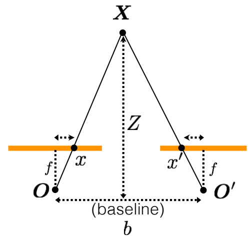
$$\frac{X}{Z} = \frac{x}{f}$$



$$\frac{X-b}{Z} = \frac{x'}{f}$$

18

$$\frac{X}{Z} = \frac{x}{f} \qquad \frac{X-b}{Z} = \frac{x'}{f}$$



Disparity

$d = x - x'$ (wrt to camera origin of image plane)

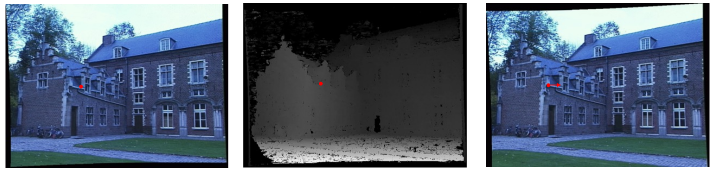
$= \frac{bf}{Z}$

inversely proportional to **depth**

19

Depth from disparity

image I(x,y) | Disparity map D(x,y) | image I´(x´,y´)



(x´,y´)=(x+D(x,y), y)

Nếu xác định được **các điểm tương ứng** trong 2 ảnh ➔ có thể **ước lượng độ sâu tương đối của các điểm ảnh…**

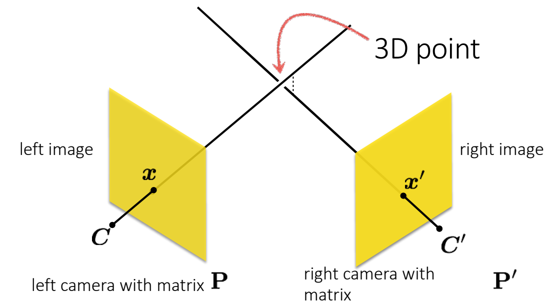
20

Stereo system – tổng quát

- 2 camera có trục quang học không song song

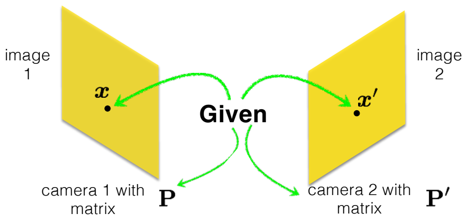
21

Tái tạo điểm 3D

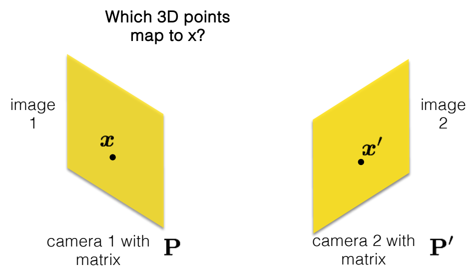


22

Phương pháp tam giác (triangulation)

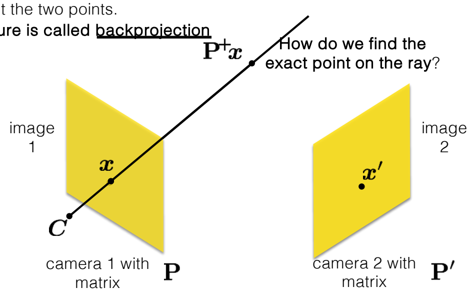


Triangulation

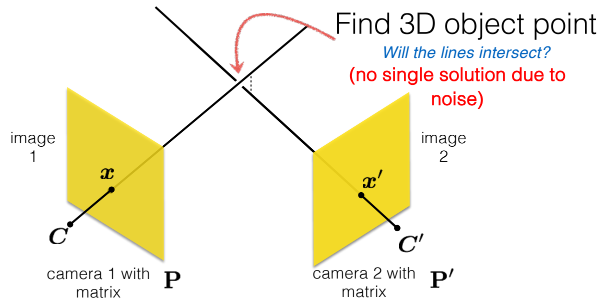


Triangulation

Create two points on the ray:
1) find the camera center; and
2) apply the pseudo-inverse of P on x.
Then connect the two points.
This procedure is called backprojection



Triangulation



Triangulation

Given a set of (noisy) matched points

$$\{\boldsymbol{x}_i, \boldsymbol{x}'_i\}$$

and camera matrices

$$\mathbf{P}, \mathbf{P}'$$

Estimate the 3D point

$$\mathbf{X}$$

27

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} p_1 & p_2 & p_3 & p_4 \\ p_5 & p_6 & p_7 & p_8 \\ p_9 & p_{10} & p_{11} & p_{12} \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

Do the same after first expanding out the camera matrix and points

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} - & \boldsymbol{p}_1^\top & - \\ - & \boldsymbol{p}_2^\top & - \\ - & \boldsymbol{p}_3^\top & - \end{bmatrix} \begin{bmatrix} | \\ \boldsymbol{X} \\ | \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \alpha \begin{bmatrix} \boldsymbol{p}_1^\top \boldsymbol{X} \\ \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_3^\top \boldsymbol{X} \end{bmatrix}$$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \times \begin{bmatrix} \boldsymbol{p}_1^\top \boldsymbol{X} \\ \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_3^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} y\boldsymbol{p}_3^\top \boldsymbol{X} - \boldsymbol{p}_2^\top \boldsymbol{X} \\ \boldsymbol{p}_1^\top \boldsymbol{X} - x\boldsymbol{p}_3^\top \boldsymbol{X} \\ x\boldsymbol{p}_2^\top \boldsymbol{X} - y\boldsymbol{p}_1^\top \boldsymbol{X} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

30

Concatenate the 2D points from both images

Two rows from camera one

Two rows from camera two

$$\begin{bmatrix} y\boldsymbol{p}_3^\top - \boldsymbol{p}_2^\top \\ \boldsymbol{p}_1^\top - x\boldsymbol{p}_3^\top \\ y'\boldsymbol{p}_3'^\top - \boldsymbol{p}_2'^\top \\ \boldsymbol{p}_1'^\top - x'\boldsymbol{p}_3'^\top \end{bmatrix} \boldsymbol{X} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\mathbf{A}\boldsymbol{X} = \mathbf{0}$$

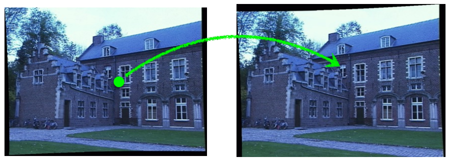
How do we solve homogeneous linear system?

S V D !

32

Ước lượng 3D?

Task: Match point in left image to point in right image



Left image　　Right image

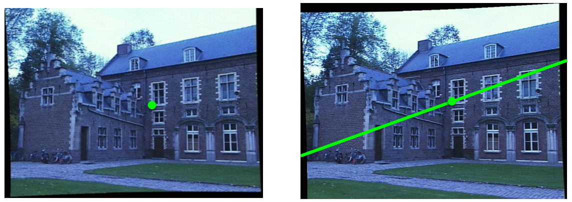
How would you do it?

- *Dense matching (for each point)*
- *Avoid search over entire image*

Epipolar geometry!

33

Ước lượng 3D



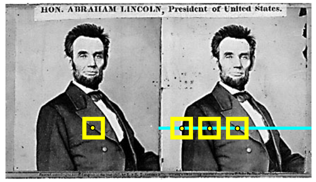
Left image　　　　Right image

1. Select point in one image (how?)
2. Form epipolar line for that point in second image (how?)
3. Find matching point along line (how?)
4. Perform triangulation (how?)

What are the disadvantages of this procedure?

34

Ước lượng 3D




```
1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity
```

$$Z = \frac{bf}{d}$$

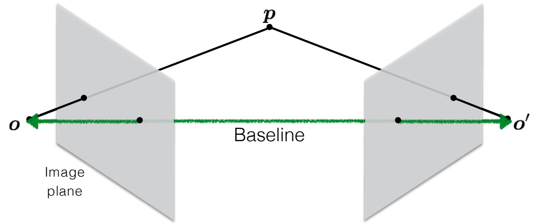
35

Next …

- Epipolar geometry (hình học lưỡng cực)
- Stereo calibration:
  - tìm các tham số của hệ thống 2 camera
  - Ma trận F (Fundamental matrix), E (Essential matrix)
- Stereo rectification: hiệu chỉnh 2 ảnh sao cho
  - Cặp epipolar line tương ứng giữa 2 ảnh nằm trên cùng 1 hàng
  - Epipolar lines trong mỗi ảnh: song song với nhau

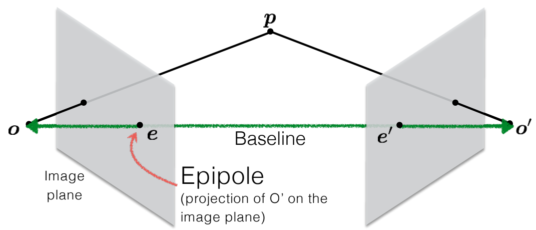
36

Epipolar geometry



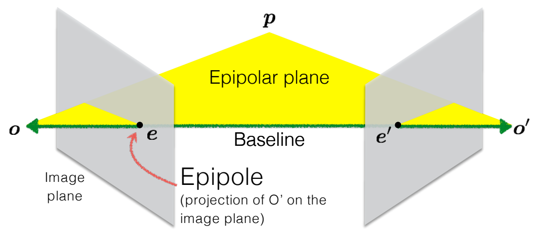
37

Epipolar geometry



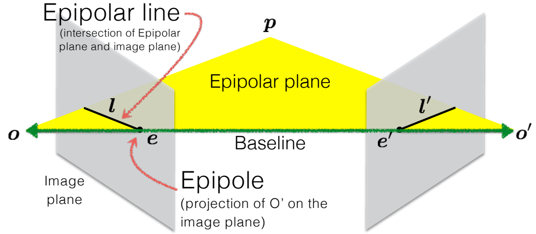
38

Epipolar geometry



39

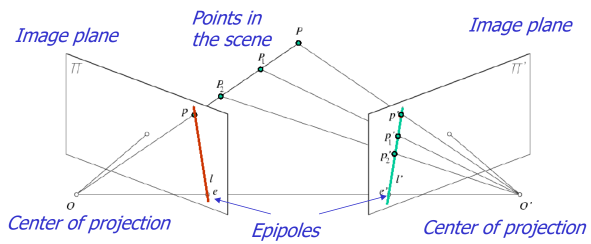
Epipolar geometry



40

Epipolar constraint

- Một điểm ảnh trong ảnh bên trái được quan sát thấy trên epipolar line tương ứng ở ảnh bên phải
- Tất cả epipolar lines đều đi qua epipole của trên ảnh đó



42

Epipolar constraint

- Một điểm ảnh trong ảnh bên trái được quan sát thấy trên epipolar line tương ứng ở ảnh bên phải

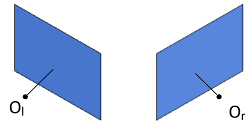
➔ Giảm thời gian tìm kiếm điểm tương ứng: chỉ cần tìm các điểm trên epipolar line

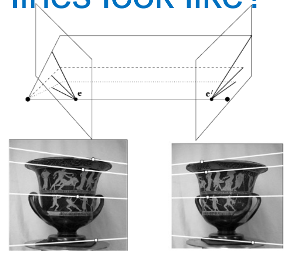
Image from Andrew Zisserman

43

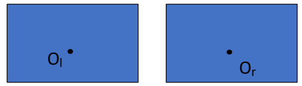
What do the epipolar lines look like?

Converging cameras





Parallel cameras



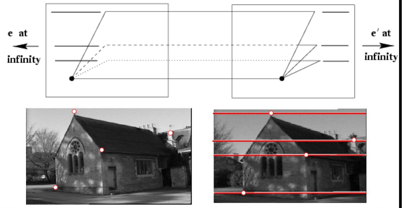


Figure from Hartley & Zisserman

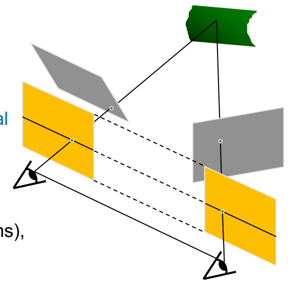
44

Stereo image rectification

- Về ứng dụng, rất thuận lợi nếu epipolar lines tương ứng là các hàng (rows) của từng ảnh

➔ rectification:

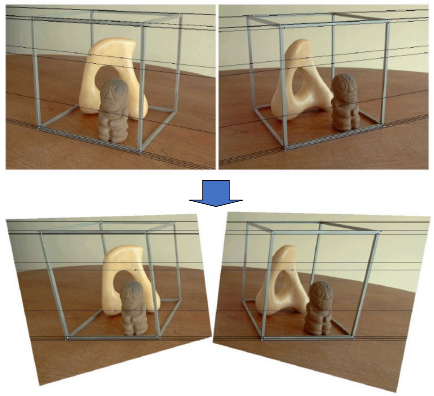
  – reproject image planes onto a common plane parallel to the line between optical centers
  – pixel motion is horizontal after this transformation
  – Need two homographies (3x3 transforms), one for each input image reprojection



Slide credit: Li Zhang

45

Stereo image rectification: example



Source: Alyosha Efros

46

Stereo Calibration

Essential and Fundamental matrix : E, F

- Mối tương quan giữa 2 camera:
  - Essential matrix **E**
    - thể hiện vị trí tương đối của camera thứ 2 so với camera thứ 1
    - Gồm 1 phép dịch T và 1 phép quay R
  - Fundamental matrix **F**
    - Chứa thông tin về ma trận E và tham số trong của 2 camera

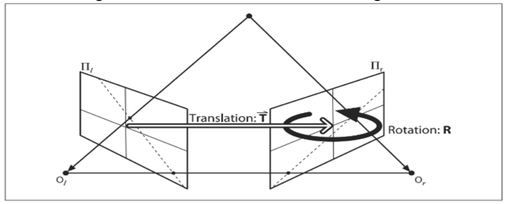
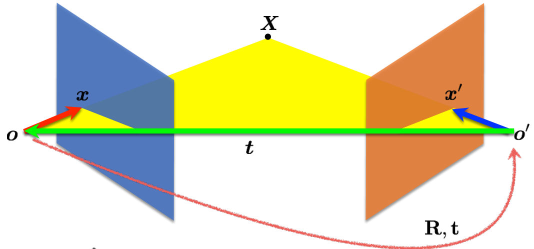


Figure 12-9. The essential geometry of stereo imaging is captured by the essential matrix E, which contains all of the information about the translation T and the rotation R, which describe the location of the second camera relative to the first in global coordinates

Ma trận E (Essential Matrix)



$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$

Giả sử:

- x và x': được biểu diễn 2D trong hệ tọa độ camera
- Nếu ma trận tham số trong K = I ➔ biểu diễn trong hệ tọa độ camera = biểu diễn trong hệ tọa độ ảnh

Ma trận E

rigid motion: $\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$

coplanarity: $(\boldsymbol{x} - \boldsymbol{t})^\top (\boldsymbol{t} \times \boldsymbol{x}) = 0$

$$(\boldsymbol{x}'^\top \mathbf{R})(\boldsymbol{t} \times \boldsymbol{x}) = 0$$
$$(\boldsymbol{x}'^\top \mathbf{R})([\mathbf{t}_\times]\boldsymbol{x}) = 0$$
$$\boldsymbol{x}'^\top (\mathbf{R}[\mathbf{t}_\times])\boldsymbol{x} = 0$$
$$\boxed{\boldsymbol{x}'^\top \mathbf{E} \boldsymbol{x} = 0}$$

Essential Matrix
[Longuet-Higgins 1981]

$[t_x]$: skew symmetrix matrix for vector t

Tính chất của ma trận E

Longuet-Higgins equation	$\boldsymbol{x}'^{\top}\mathbf{E}\boldsymbol{x} = 0$	
Epipolar lines	$\boldsymbol{x}^{\top}\boldsymbol{l} = 0$ $\boldsymbol{l}' = \mathbf{E}\boldsymbol{x}$	$\boldsymbol{x}'^{\top}\boldsymbol{l}' = 0$ $\boldsymbol{l} = \mathbf{E}^{T}\boldsymbol{x}'$
Epipoles	$\boldsymbol{e}'^{\top}\mathbf{E} = \mathbf{0}$	$\mathbf{E}\boldsymbol{e} = \mathbf{0}$

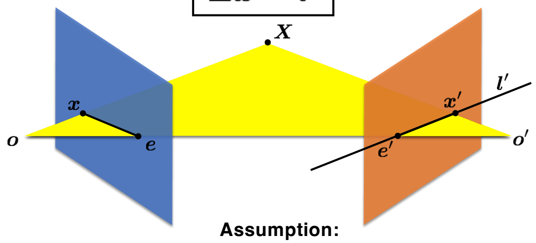
(2D points expressed in camera coordinate system)

51

E và epipolar line

Nếu biết tọa độ 1 điểm trong hệ quy chiếu camera 1, và biết E
➔ xác định được epipolar line tương ứng ở camera 2

$$\boxed{\mathbf{E}\boldsymbol{x} = \boldsymbol{l}'}$$



Assumption:
2D points expressed in camera coordinate system
(i.e., intrinsic matrices are identities)

52

Trường hợp tổng quát: E

- Giả sử 2 camera đều xác định được tham số trong (calibrated camera) K, K'

$$\hat{\boldsymbol{x}}'^{\top}\mathbf{E}\hat{\boldsymbol{x}} = 0$$

The essential matrix operates on image points expressed in **2D coordinates** expressed in the camera coordinate system

$$\hat{\boldsymbol{x}}' = \mathbf{K}'^{-1}\boldsymbol{x}' \qquad \underset{\text{camera point}}{\hat{\boldsymbol{x}}} = \mathbf{K}^{-1}\underset{\text{image point}}{\boldsymbol{x}}$$

53

Trường hợp tổng quát: E

$\hat{\boldsymbol{x}}$ tọa độ trong hệ quy chiếu camera

$$\hat{\boldsymbol{x}}'^{\top}\mathbf{E}\hat{\boldsymbol{x}} = 0$$

$$\hat{\boldsymbol{x}}' = \mathbf{K}'^{-1}\boldsymbol{x}' \qquad \underset{\text{camera point}}{\hat{\boldsymbol{x}}} = \mathbf{K}^{-1}\underset{\text{image point}}{\boldsymbol{x}}$$

$\boldsymbol{x}$ Tọa độ trong hệ quy chiếu ảnh

$$\mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}\boldsymbol{x} = 0$$
$$\boldsymbol{x}'^{\top}(\mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1})\boldsymbol{x} = 0$$
$$\boldsymbol{x}'^{\top}\mathbf{F}\boldsymbol{x} = 0$$

➔ F: Fundamental matrix

54

Tính chất của ma trận F

Longuet-Higgins equation	$\boldsymbol{x}'^{\top}\mathbf{F}\boldsymbol{x} = 0$	
Epipolar lines	$\boldsymbol{x}^{\top}\boldsymbol{l} = 0$ $\boldsymbol{l}' = \mathbf{F}\boldsymbol{x}$	$\boldsymbol{x}'^{\top}\boldsymbol{l}' = 0$ $\boldsymbol{l} = \mathbf{F}^{T}\boldsymbol{x}'$
Epipoles	$\boldsymbol{e}'^{\top}\mathbf{F} = \mathbf{0}$	$\mathbf{F}\boldsymbol{e} = \mathbf{0}$

(2D points expressed in image coordinates)

55

Essential and Fundamental matrix : E, F

- Ma trận **E** :
  - E: chỉ phụ thuộc vào tham số ngoài
  - Nếu đã biết tham số trong K, K' ➔ **E** cho phép xác định tham số ngoài

$$\mathbf{E} = \mathbf{R}[\mathbf{t}_{\times}]$$

- Ma trận **F:**
  - **F** phụ thuộc vào tham số trong và tham số ngoài
  - Nếu K, K', R, t không biết ➔ ước lượng F? (weak calibration)

$$\mathbf{F} = \mathbf{K}'^{-\top}\mathbf{E}\mathbf{K}^{-1}$$

56

Weak calibration – Ước lượng F

- Weak calibration
  - Intrinsics and extrinsics parameters: unknown
  - i.e. estimate F without calibration matrix **K** ou **P** of 2 camera
- F : 3x3 matrix,
  - rank(F)= 2
  - 8 unknown values
- Need solve many (>8) pair of corresponding pixels

57

Thuật toán 8 điểm

$$\boldsymbol{x}_m'^{\top}\mathbf{F}\boldsymbol{x}_m = 0$$

$$\begin{bmatrix} x'_m & y'_m & 1 \end{bmatrix} \begin{bmatrix} f_1 & f_2 & f_3 \\ f_4 & f_5 & f_6 \\ f_7 & f_8 & f_9 \end{bmatrix} \begin{bmatrix} x_m \\ y_m \\ 1 \end{bmatrix} = 0$$

1 cặp điểm ➔ 1 phương trình

$$\begin{aligned} x_m x'_m f_1 + x_m y'_m f_2 + x_m f_3 + \\ y_m x'_m f_4 + y_m y'_m f_5 + y_m f_6 + \\ x'_m f_7 + y'_m f_8 + f_9 = 0 \end{aligned}$$

58

Thuật toán 8 điểm

$$\boldsymbol{x}_m'^{\top}\mathbf{F}\boldsymbol{x}_m = 0$$

$$\begin{bmatrix} x_1x_1' & x_1y_1' & x_1 & y_1x_1' & y_1y_1' & y_1 & x_1' & y_1' & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_Mx_M' & x_My_M' & x_M & y_Mx_M' & y_My_M' & y_M & x_M' & y_M' & 1 \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix} = \mathbf{0}$$

$$\mathbf{A}\boldsymbol{X} = \mathbf{0}$$

Total Least Squares

minimize $\|\mathbf{A}\boldsymbol{x}\|^2$

subject to $\|\boldsymbol{x}\|^2 = 1$

S V D !

59

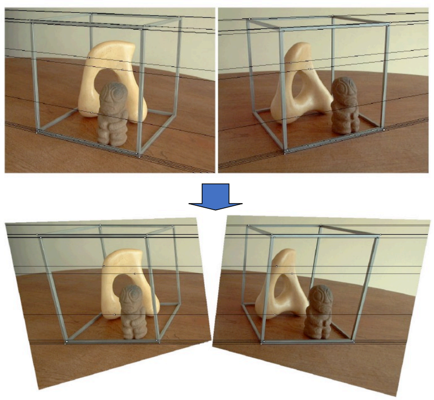
Ước lượng F (opencv)

- Phát hiện các điểm quan trọng trên 2 ảnh
  - Vd: sử dụng DoG, SIFT
- So khớp để tìm ra các cặp điểm tương đồng
  - VD: nearest neighbor + ratio of distance
- Giải bài toán tối ưu ➔ F

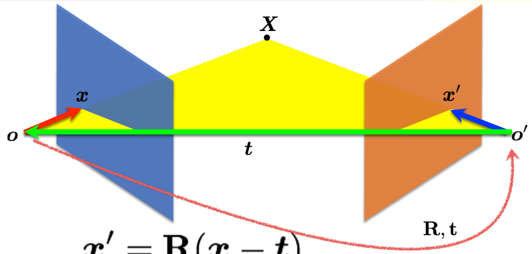
(thư viện có hàm hỗ trợ: findFundamentalMatrix)

61

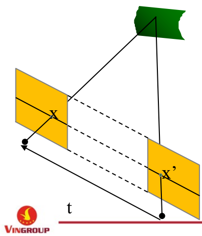
Stereo rectification



64



$$\boldsymbol{x}' = \mathbf{R}(\boldsymbol{x} - \boldsymbol{t})$$



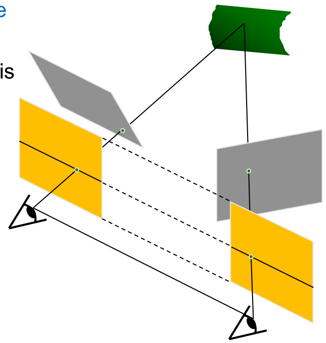
Epipolar lines nằm // theo trục hoành thì:

$R = I \qquad t = (T, 0, 0)$

65

Stereo image rectification

- Reproject image planes onto a common plane parallel to the line between optical centers
- pixel motion is horizontal after this transformation
- Need two homographies (3x3 transforms), one for each input image reprojection



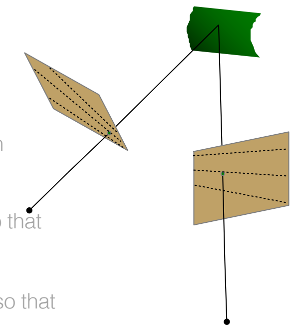
Slide credit: Li Zhang

Stereo Rectification

1. **Rotate** the right camera by **R**
   (aligns camera coordinate system orientation only)
2. Rotate (**rectify**) the left camera so that the epipole is at infinity
3. Rotate (**rectify**) the right camera so that the epipole is at infinity
4. Adjust the **scale**

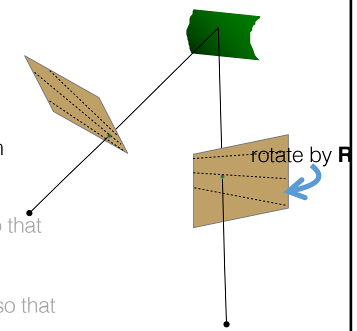
Stereo Rectification

1. Compute **E** to get **R**
2. **Rotate** the right camera by **R**
   (aligns camera coordinate system orientation only)
3. Rotate (**rectify**) the left camera so that the epipole is at infinity
4. Rotate (**rectify**) the right camera so that the epipole is at infinity
5. Adjust the **scale**



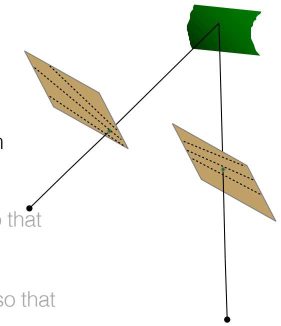
Stereo Rectification

1. Compute **E** to get **R**
2. **Rotate** the right camera by **R**
   (aligns camera coordinate system orientation only)
3. Rotate (**rectify**) the left camera so that the epipole is at infinity
4. Rotate (**rectify**) the right camera so that the epipole is at infinity
5. Adjust the **scale**



Stereo Rectification

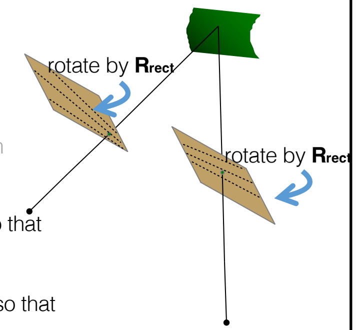
1. Compute **E** to get **R**
2. **Rotate** the right camera by **R** (aligns camera coordinate system orientation only)
3. Rotate (**rectify**) the left camera so that the epipole is at infinity
4. Rotate (**rectify**) the right camera so that the epipole is at infinity
5. Adjust the **scale**



70

Stereo Rectification

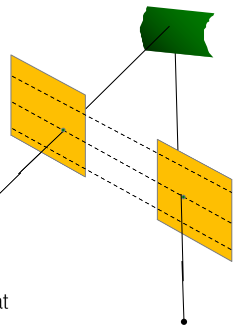
1. Compute **E** to get **R**
2. **Rotate** the right camera by **R** (aligns camera coordinate system orientation only)
3. Rotate (**rectify**) the left camera so that the epipole is at infinity - $\mathbf{R}_{rect}$
4. Rotate (**rectify**) the right camera so that the epipole is at infinity - $\mathbf{R}_{rect}$
5. Adjust the **scale**



71

Stereo Rectification

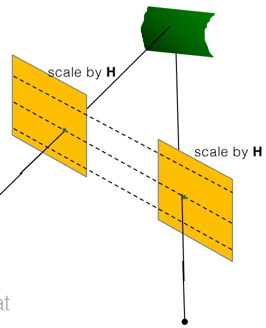
1. Compute **E** to get **R**
2. **Rotate** the right camera by **R** (aligns camera coordinate system orientation only)
3. Rotate (**rectify**) the left camera so that the epipole is at infinity - $\mathbf{R}_{rect}$
4. Rotate (**rectify**) the right camera so that the epipole is at infinity - $\mathbf{R}_{rect}$
5. Adjust the **scale**



72

Stereo Rectification

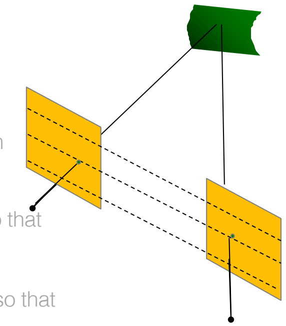
1. Compute **E** to get **R**
2. **Rotate** the right camera by **R** (aligns camera coordinate system orientation only)
3. Rotate (**rectify**) the left camera so that the epipole is at infinity
4. Rotate (**rectify**) the right camera so that the epipole is at infinity
5. Adjust the **scale**



73

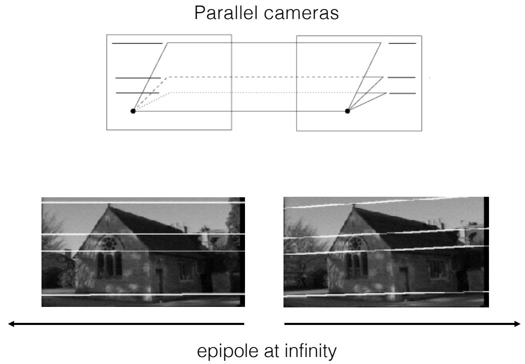
Stereo Rectification

1. Compute **E** to get **R**
2. **Rotate** the right camera by **R** (aligns camera coordinate system orientation only)
3. Rotate (**rectify**) the left camera so that the epipole is at infinity
4. Rotate (**rectify**) the right camera so that the epipole is at infinity
5. Adjust the **scale**



74

Epipolar lines // khi Epipoles ở vô cực



82

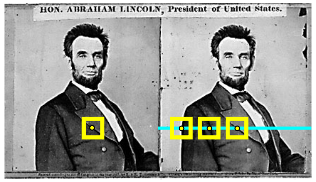
Stereo Rectification Algorithm

1. Estimate **E** using the 8 point algorithm (SVD)
2. Estimate the epipole **e** (SVD of **E**)
3. Build $\mathbf{R}_{rect}$ from **e**
4. Decompose **E** into **R** and **T**
5. Set $\mathbf{R}_1 = \mathbf{R}_{rect}$ and $\mathbf{R}_2 = \mathbf{R}\mathbf{R}_{rect}$
6. Rotate each left camera point (warp image) $[x'\ y'\ z'] = \mathbf{R}_1\ [x\ y\ z]$
7. Rectified points as $\mathbf{p} = f/z'[x'\ y'\ z']$
8. Repeat 6 and 7 for right camera points using $\mathbf{R}_2$

85

Stereo matching

86




```
1. Rectify images
   (make epipolar lines horizontal)
2. For each pixel
   a. Find epipolar line
   b. Scan line for best match
   c. Compute depth from disparity
```

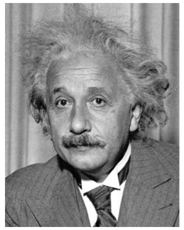
How would you do this?

$$Z = \frac{bf}{d}$$

87

Template matching

How do we detect the template in the following image?

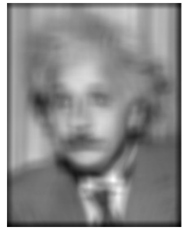


filter

output

$$h[m,n] = \sum_{k,j} g[k,l]\, f[m+k, n+l]$$

image



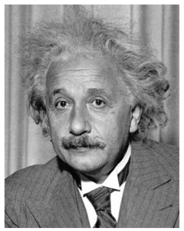
Solution 1: Filter the image using the template **as filter kernel**.

Increases for higher local intensities.

88

Template matching

How do we detect the template in the following image?



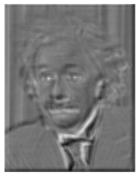
output

filter

template mean

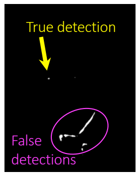
$$h[m,n] = \sum_{k,l} (g[k,l] - \bar{g}) f[m+k, n+l]$$

image



output

thresholding



Solution 2: Filter the image using a ***zero-mean*** **template.**

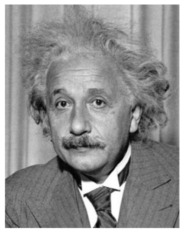
What went wrong?

Not robust to high-contrast areas

89

Template matching

How do we detect the template in the following image?



filter

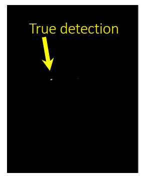
output

$$h[m,n] = \sum_{k,l} (g[k,l] - f[m+k, n+l])^2$$

image

1-output

thresholding



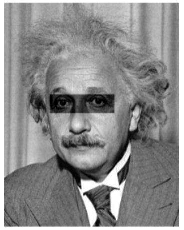
Solution 3: **Use sum of squared differences (SSD)**.

What could go wrong?

90

Template matching

How do we detect the template in the following image?



filter

output

$$h[m,n] = \sum_{k,l} (g[k,l] - f[m+k, n+l])^2$$

image

1-output

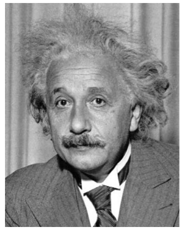
Not robust to local intensity changes

Solution 3: **Use sum of squared differences (SSD)**.

91

Template matching

How do we detect the template in the following image?



Observations so far:

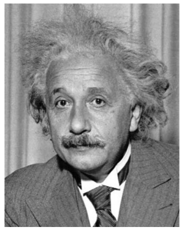
- subtracting mean deals with brightness bias
- dividing by standard deviation removes contrast bias

Can we combine the two effects?

92

Template matching

How do we detect the template in the following image?



filter

template mean

What will the output look like?

output

$$h[m,n] = \frac{\sum_{k,l}(g[k,l] - \bar{g})(f[m+k,n+l] - \bar{f}_{m,n})}{\sqrt{\left(\sum_{k,l}(g[k,l] - \bar{g})^2 \sum_{k,l}(f[m+k,n+l] - \bar{f}_{m,n})^2\right)}}$$

image

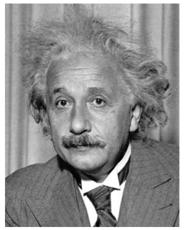
local patch mean

Solution 4: **Normalized cross-correlation (NCC)**.

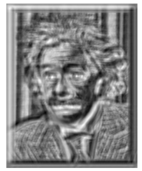
93

Template matching

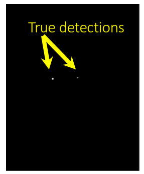
How do we detect the template in the following image?



1-output



thresholding

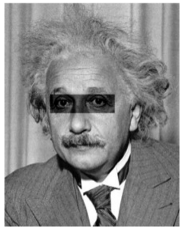


Solution 4: **Normalized cross-correlation (NCC)**.

94

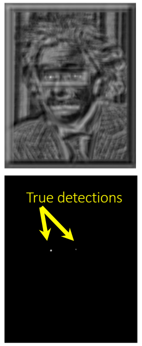
Template matching

How do we detect the template in the following image?



1-output

thresholding



Solution 4: **Normalized cross-correlation (NCC)**.

95

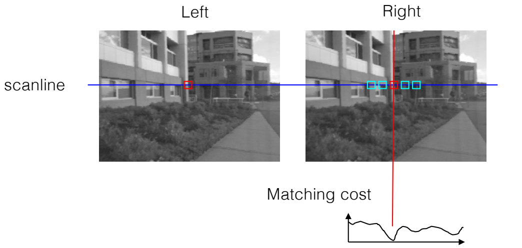
What is the best method?

It depends on whether you care about speed or invariance.

- Zero-mean: Fastest, very sensitive to local intensity.
- Sum of squared differences: Medium speed, sensitive to intensity offsets.
- Normalized cross-correlation: Slowest, invariant to contrast and brightness.

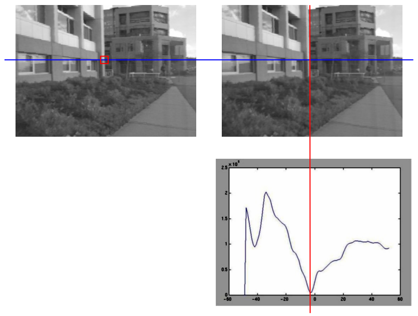
96

Stereo Block Matching



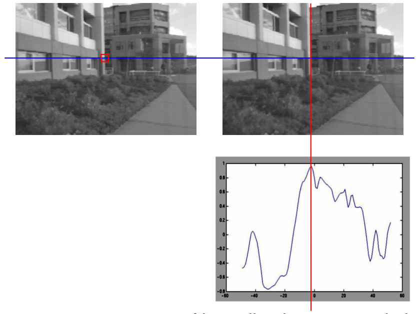
- Slide a window along the epipolar line and compare contents of that window with the reference window in the left image
- Matching cost: **SSD** or **normalized correlation**

97



SSD

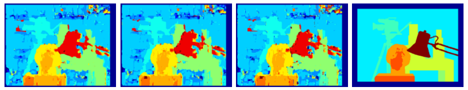
98



Normalized cross-correlation

99

Similarity Measure	Formula
Sum of Absolute Differences (SAD)	$\sum_{(i,j)\in W} \lvert I_1(i,j) - I_2(x+i,y+j) \rvert$
Sum of Squared Differences (SSD)	$\sum_{(i,j)\in W} \left(I_1(i,j) - I_2(x+i,y+j)\right)^2$
Zero-mean SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \bar{I}_1(i,j) - I_2(x+i,y+j) + \bar{I}_2(x+i,y+j) \rvert$
Locally scaled SAD	$\sum_{(i,j)\in W} \lvert I_1(i,j) - \frac{\bar{I}_1(i,j)}{\bar{I}_2(x+i,y+j)} I_2(x+i,y+j) \rvert$
Normalized Cross Correlation (NCC)	$\frac{\sum_{(i,j)\in W} I_1(i,j).I_2(x+i,y+j)}{\sqrt[2]{\sum_{(i,j)\in W} I_1^2(i,j).\sum_{(i,j)\in W} I_2^2(x+i,y+j)}}$



SAD SSD NCC Ground truth

100

Effect of window size

W = 3 W = 20

Smaller window
- \+ More detail
- \- More noise

Larger window
- \+ Smoother disparity maps
- \- Less detail
- \- Fails near boundaries

101

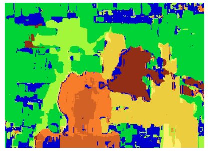
When will stereo block matching fail?

102

Improving stereo matching

103

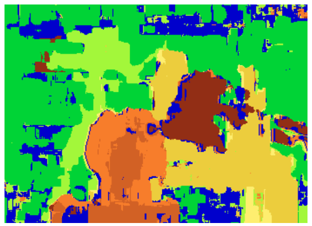
Block matching



Ground truth

What are some problems with the result?

104



How can we improve depth estimation?

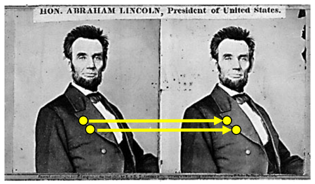
Too many discontinuities.
We expect disparity values to change slowly.

➔ Let's make an assumption:
depth should change smoothly

105

Stereo matching as …

Energy Minimization



What defines a good stereo correspondence?

1. **Match quality**
   – Want each pixel to find a good match in the other image
2. **Smoothness**
   – If two pixels are adjacent, they should (usually) move about the same amount

106

energy function
(for one pixel)

$$E(d) = \underbrace{E_d(d)}_{\text{data term}} + \lambda \underbrace{E_s(d)}_{\text{smoothness term}}$$

Want each pixel to find a good match in the other image

(block matching result)

Adjacent pixels should (usually) move about the same amount

(smoothness function)

107

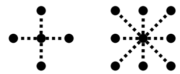
$$E(d) = \boxed{E_d(d)} + \lambda \boxed{E_s(d)}$$

$$\underset{\text{data term}}{E_d(d)} = \sum_{(x,y)\in I} C(x, y, d(x, y))$$

SSD distance between windows centered at I(x, y) and J(x+ d(x,y), y)

$$\underset{\text{smoothness term}}{E_s(d)} = \sum_{(p,q)\in \mathcal{E}} V(d_p, d_q)$$

$\mathcal{E}$: set of neighboring pixels

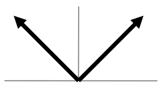


108

$$\underset{\text{smoothness term}}{E_s(d)} = \sum_{(p,q)\in \mathcal{E}} V(d_p, d_q)$$

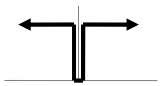
$$V(d_p, d_q) = |d_p - d_q|$$

L1 distance

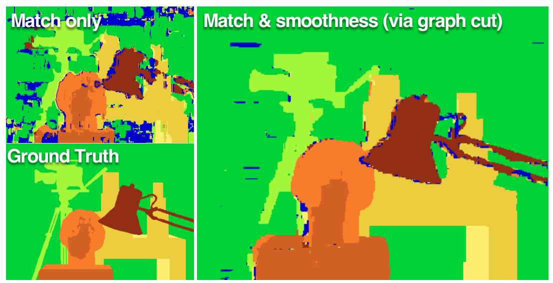


$$V(d_p, d_q) = \begin{cases} 0 & \text{if } d_p = d_q \\ 1 & \text{if } d_p \neq d_q \end{cases}$$

"Potts model"



109



Y. Boykov, O. Veksler, and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, PAMI 2001

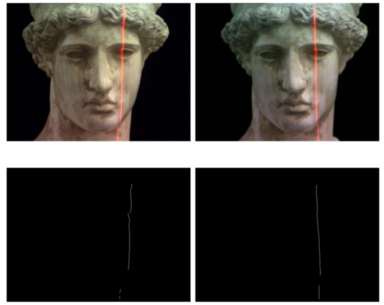
111

All of these cases remain difficult, what can we do?

112

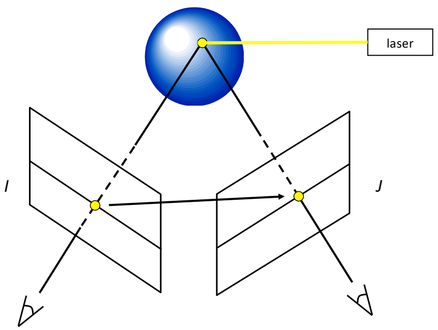
➔ Use controlled (structured) light, e.g.: laser

Disparity between laser points on the same scanline in the images determines the 3-D coordinates of the laser point on object



113

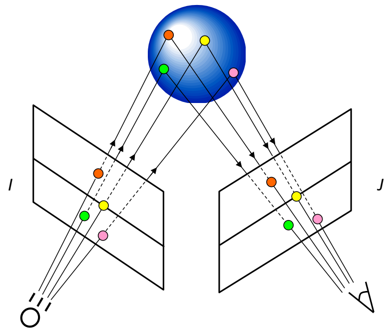
Structured light and two cameras



114

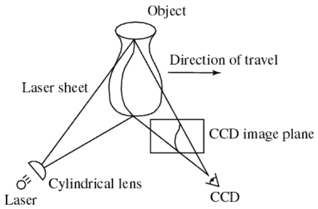
Structured light and one camera

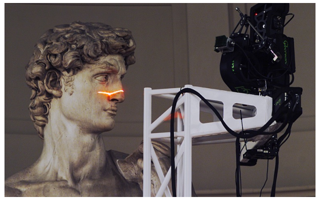
Projector acts like “reverse” camera



115

Example: Laser scanner





Digital Michelangelo Project
http://graphics.stanford.edu/projects/mich/

116

116

References

- Lecture 16, 17: CS 376: Computer Vision, Kristen Grauman, University of Texas at Austin
- Lectures: 16-385, Computer Vision - Spring 2017, 2020
- Z. Zhang, "A flexible new technique for camera calibration," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330-1334, Nov. 2000, doi: 10.1109/34.888718.
- Computer Vision II: Multiple View Geometry
  https://vision.in.tum.de/teaching/online/mvg

117

117

25 SOICT
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attention!

soict.hust.edu.vn/ fb.com/groups/soict

118

118