ATV depth estimation

Mechanical Engineering

주기영

ENGINEERING
DEPT. OF MECHANICAL ENG.
SEOUL NATIONAL UNIVERSITY
서 울 대 학 교 기 계 공 학 부

Project overview

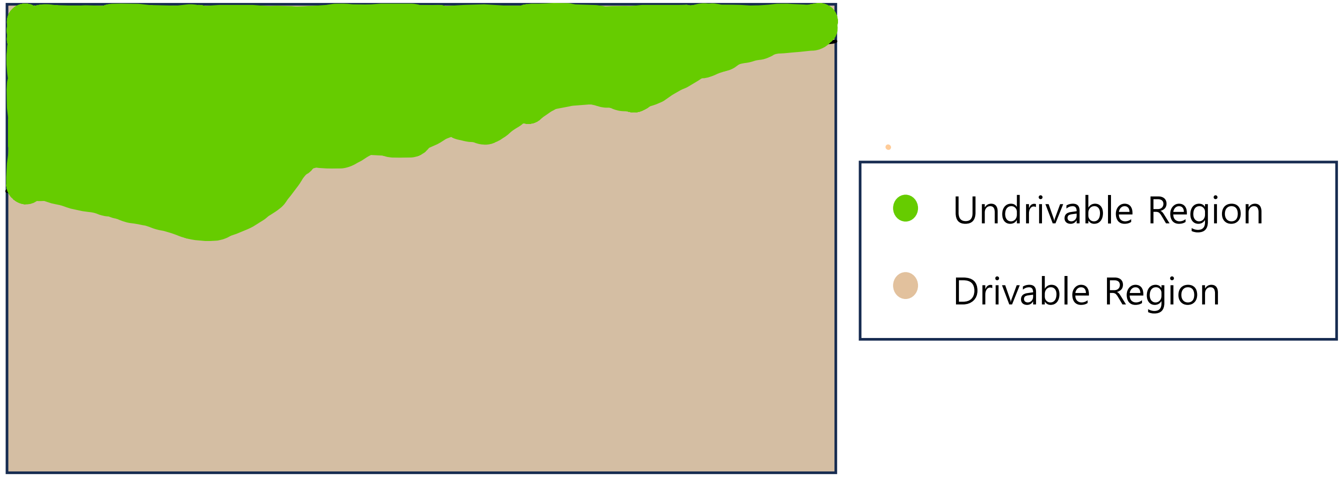
Detection of drivable regions in off-road conditions

Project overview

Depth estimation

- Segment image into regions or objects

- Segment image into drivable/undrivable region

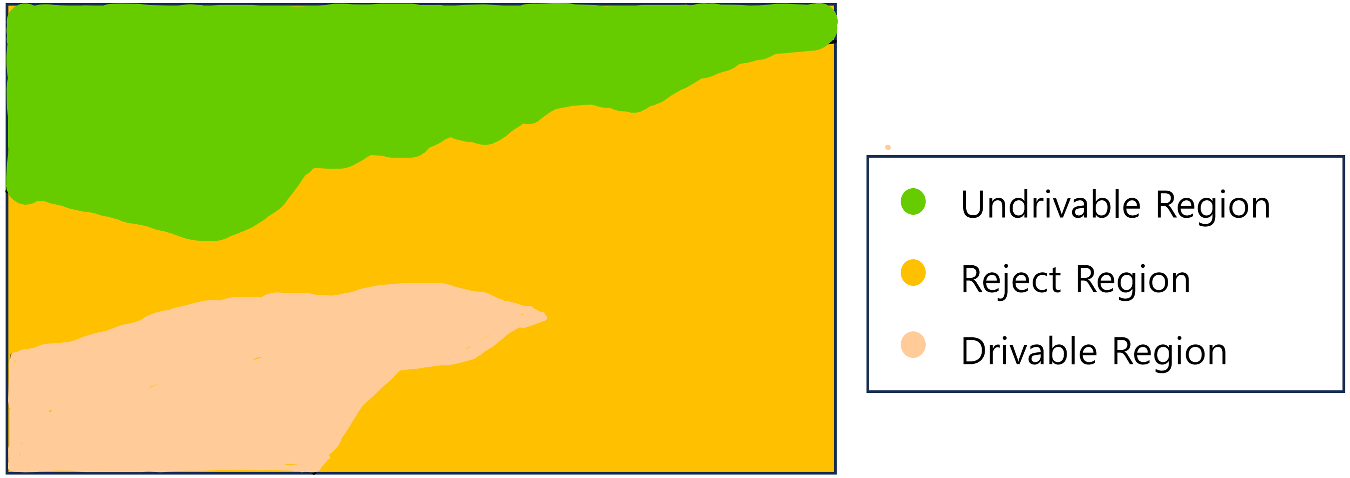


Project overview

Depth estimation

- Calculate distance to the target region

- Designate steep incline regions or wall regions as reject region



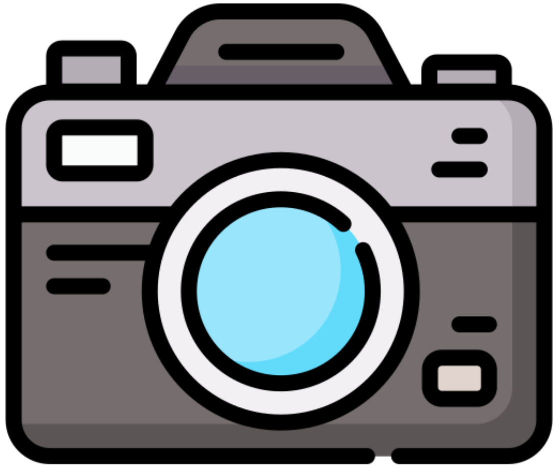
Project overview

For atnomous driving

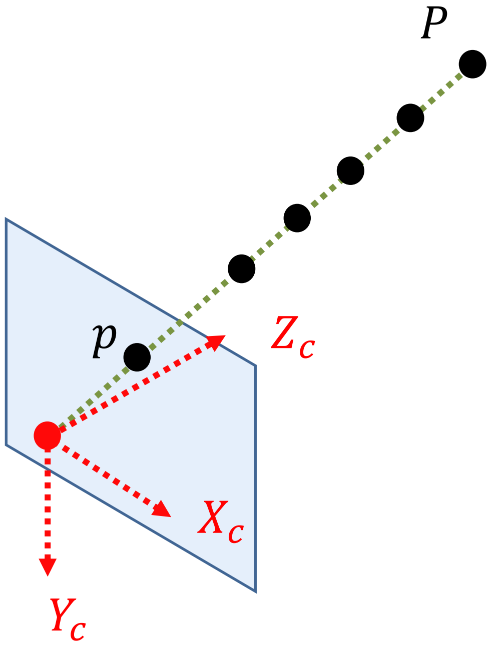
- Image real-time processing is needed

Semantic Segmentation	- Precision is more important - Using deep learning
Depth estimation	- Not using deep learning for real time processing - Stereo depth estimation

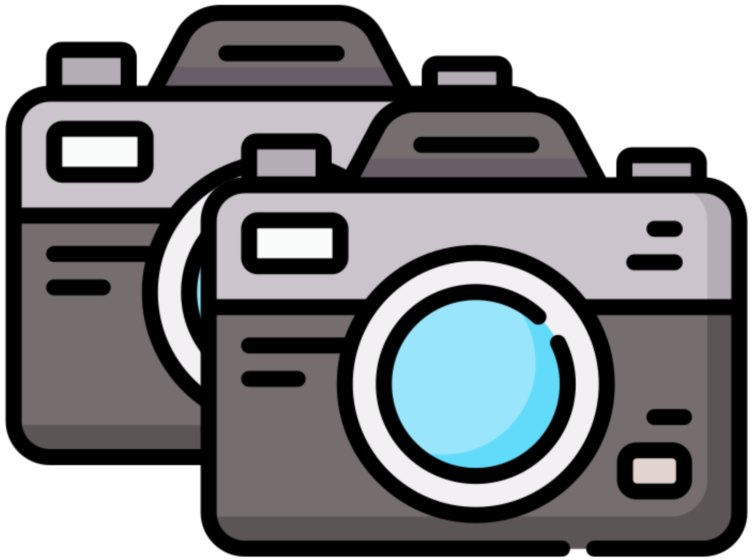
Why do we use stereo camera?



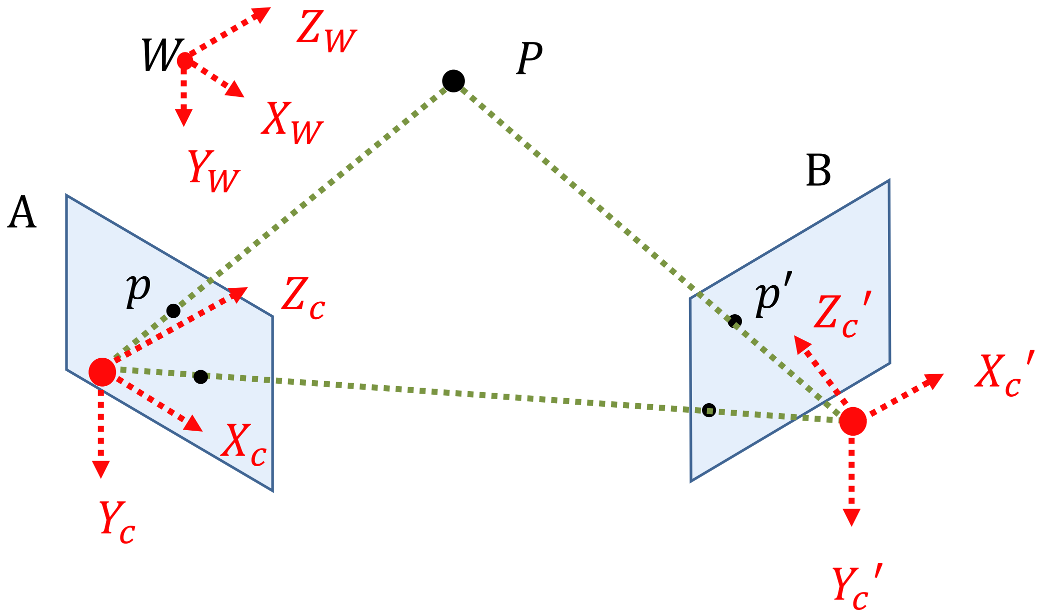
Ray of possible position



Why do we use stereo camera?



specific 3D cordinate

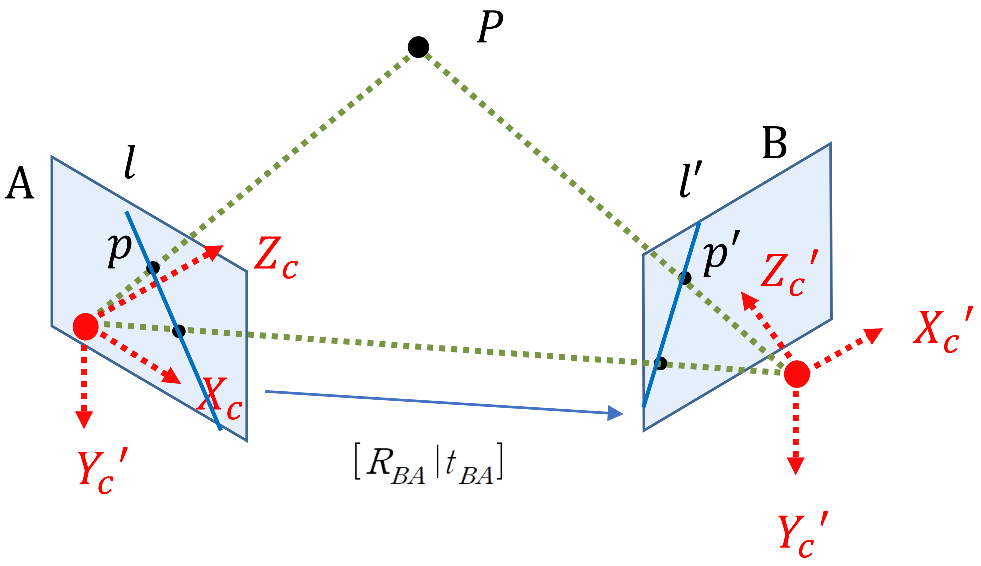


P : World coordinate

p : A image coordinate of P

p' : A image coordinate of P

For calculating depth



$[R_BA|t_BA]$: Transformation matrix

⟶ **Camera calibration**

p : A image coordinate of P

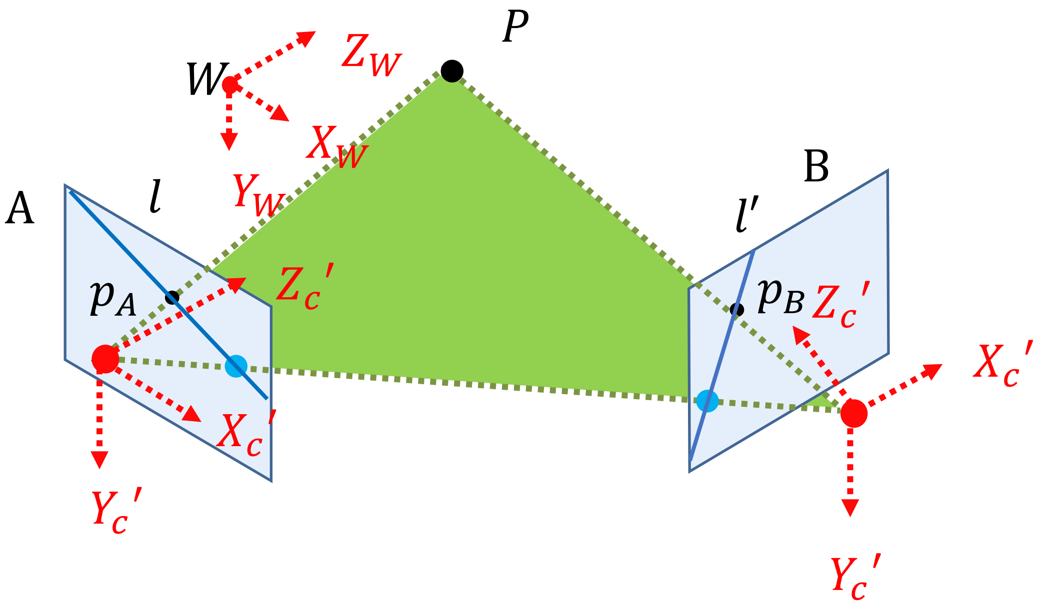
p' : A image coordinate of P

⟶ **Epipolar geometry**

⟶ **We can find world cordinate of P**

Epipolar Geometry

The geometric relationship between two camera views of the same 3D Point

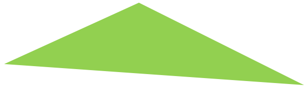


P : World cordinate

p_A : Point projected onto camera A

p_B : Point projected onto camera B

Epipolar plane



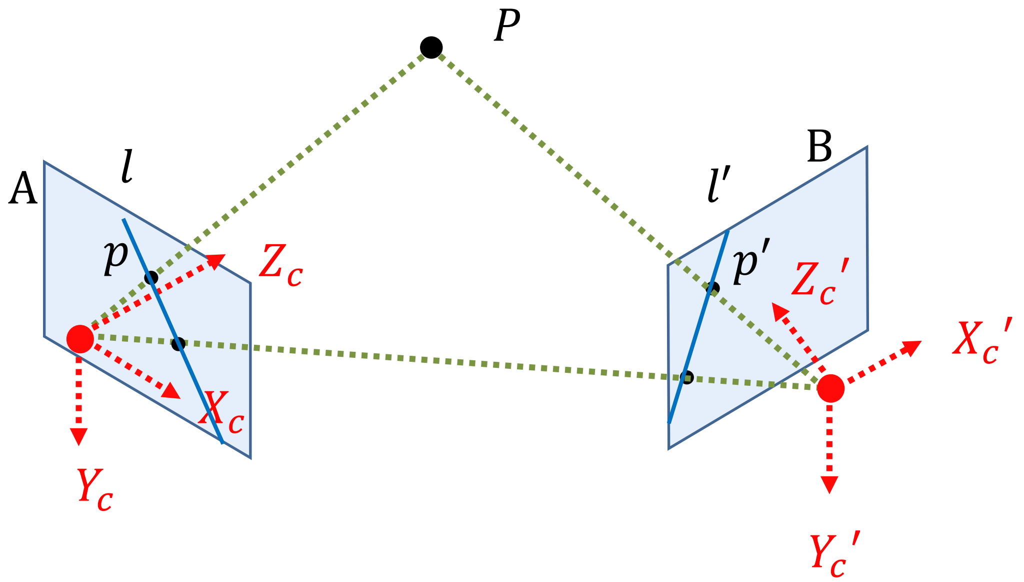
Epiline (Epipolar line)

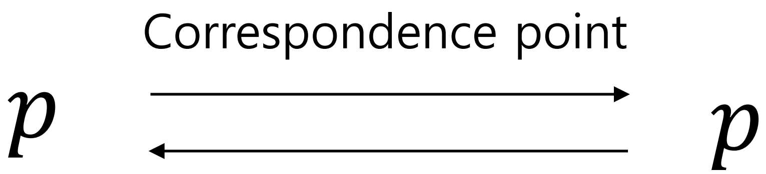
Epipole

Epipolar Geometry

Correspondence point

Pairs of image points representing the same world point





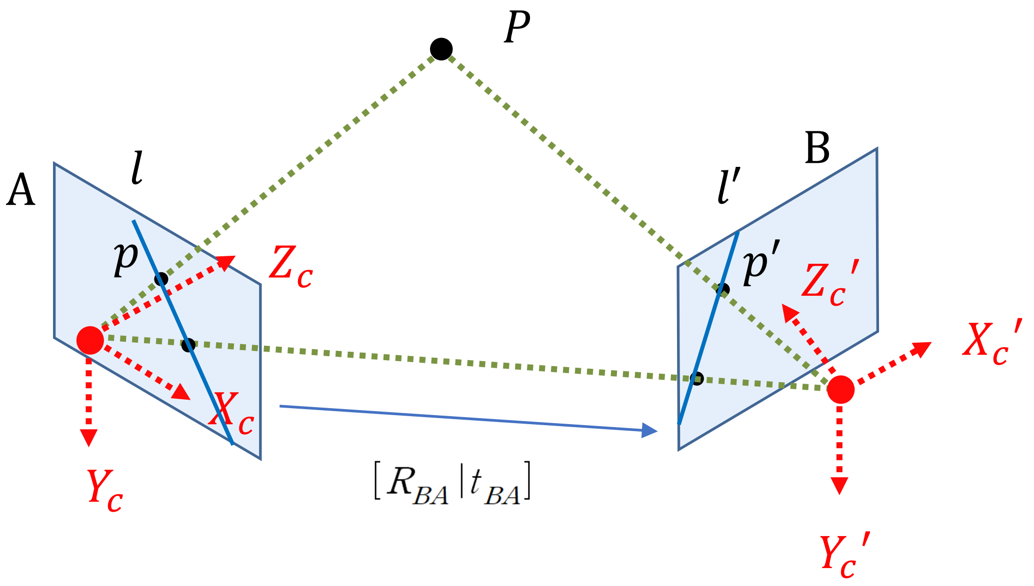
p lies on the epiline l

p' lies on the epiline l'

Epipolar Geometry

Essential Matrix

matrix that relates corresponding points between two images



$$[t]_x = \begin{pmatrix} 0 & -t_1 & t_2 \\ t_1 & 0 & -t_3 \\ -t_2 & t_3 & 0 \end{pmatrix}$$

p : A image coordinate of P

p' : A image coordinate of P

E : Essential matrix

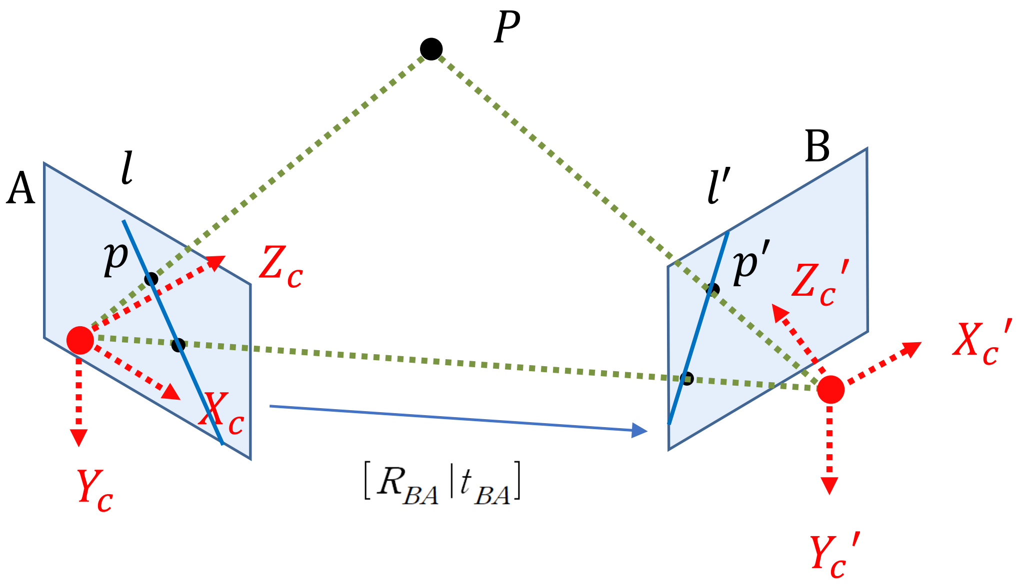
$[R_B A | t_B A]$: Transformation matrix

$$E = [t_{BA}]_x R_{BA} \longrightarrow p'^T E p = 0$$

Epipolar Geometry

Projective geometry

u : A line	$\longrightarrow \quad x^T u = 0$
x : A point on a line u	



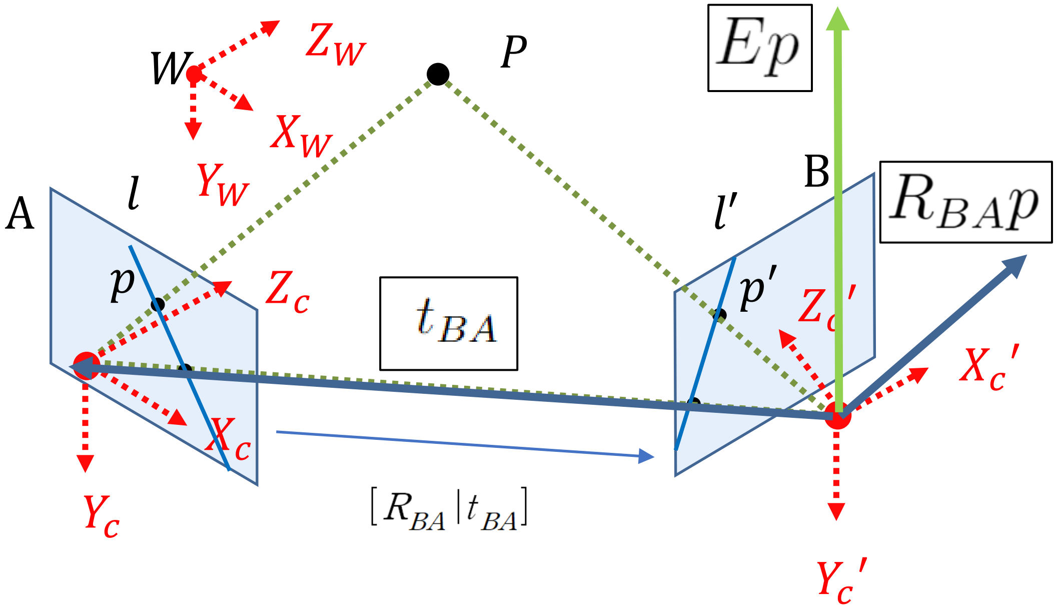
By the definition of epiline $\longrightarrow \quad p'^T l' = 0$

$$p'^T E p = 0 \quad \begin{cases} p' & \text{: Correspondence point} \\ Ep & \text{: Epiline } l' \end{cases}$$

Epipolar Geometry

Meaning of *Ep* vector

$$E = [t_{BA}]_x R_{BA} \longrightarrow Ep = [t_{BA}]_x R_{BA} p$$



3D World Coordinate system

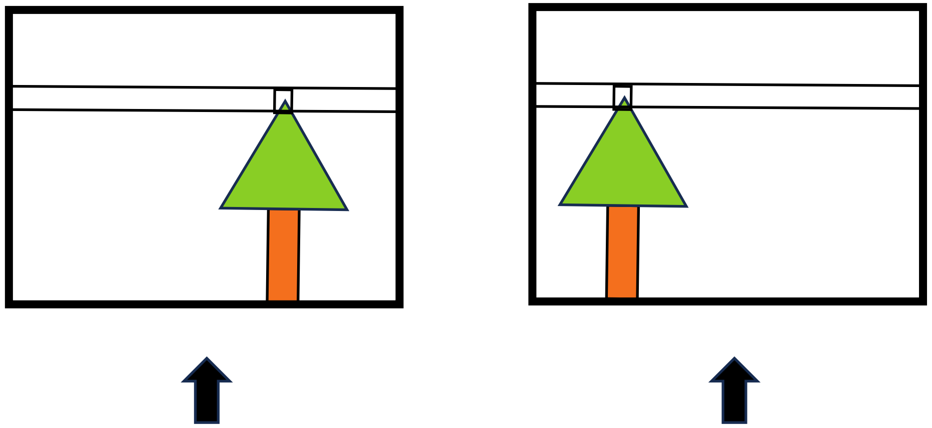
Normal vector of the Epipolar plane

B Image coordinate system

Homogeneous expression of Epiline

Depth Estimation

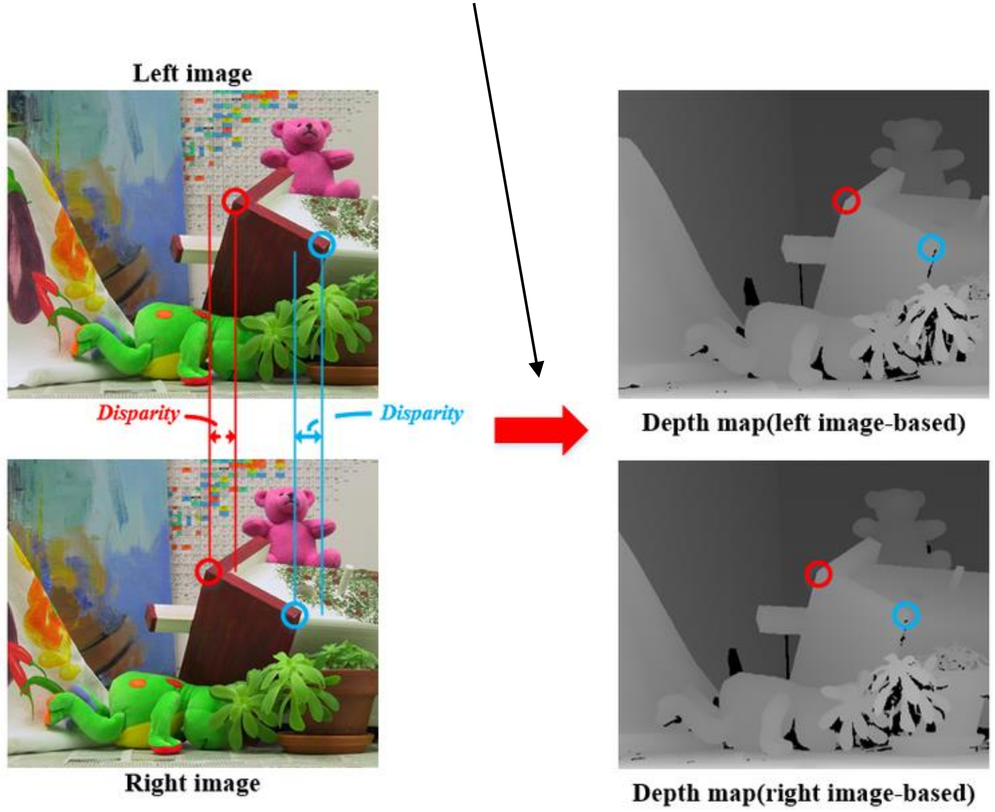
Finding correspondence



Reference Image

Target Image

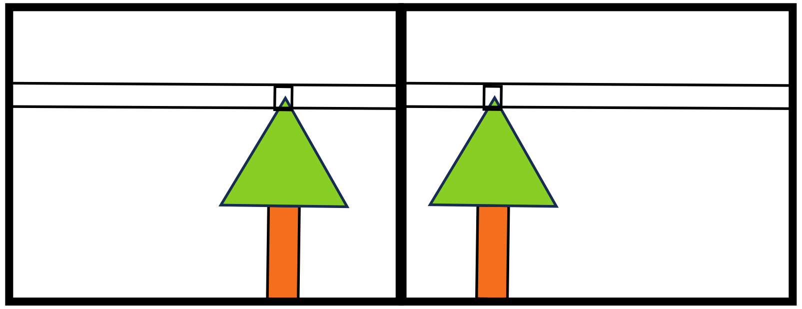
Triangulization



Depth Estimation

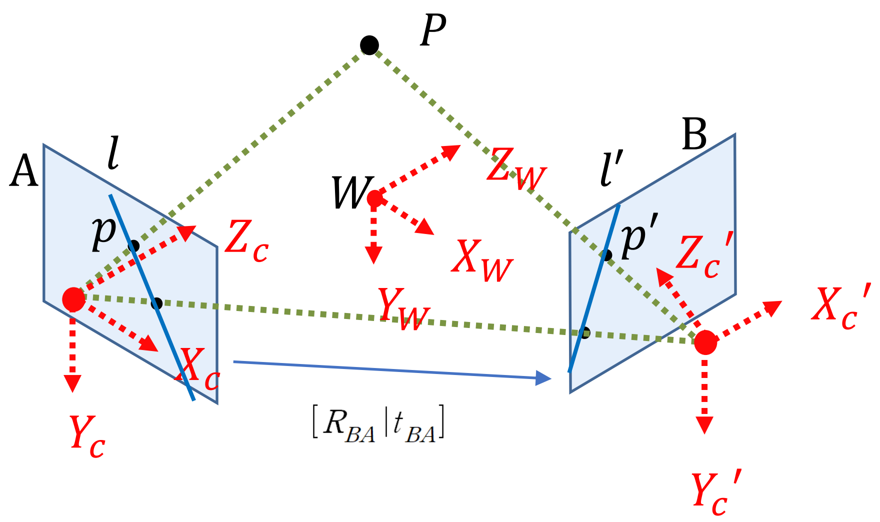
Step 1 : Correspondence Matching

- The process of calculating p'



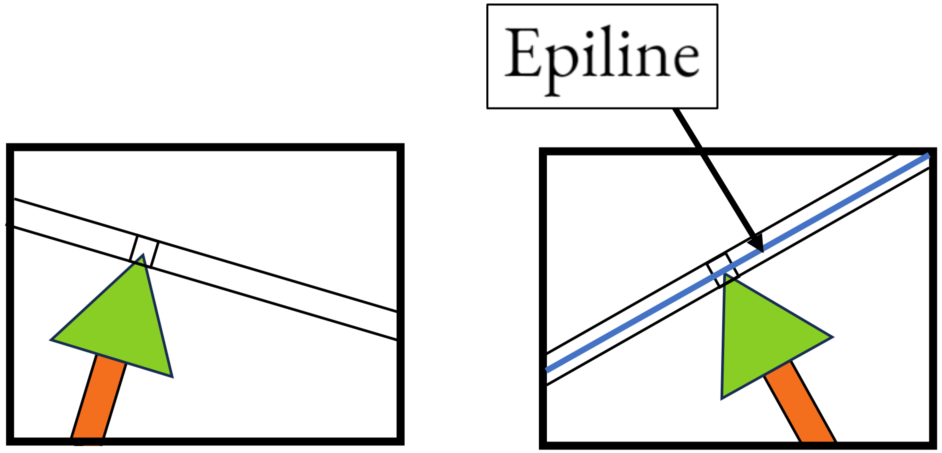
Step 2 : Triangulization

- The process of calculating world coordinate of P



Correspondence Matching

Using Epipolar geometry



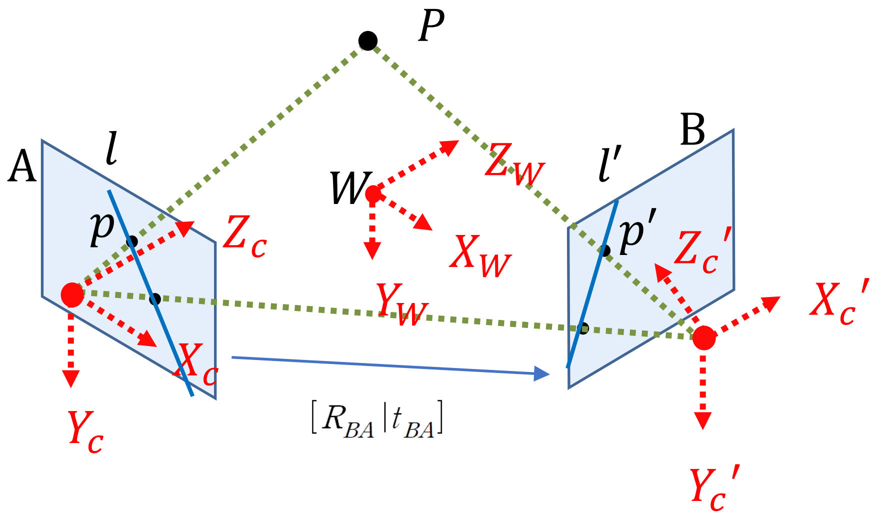
Finding Epiline

Using Essential matrix

Finding correspondence point

Using Block matching

Triangulization



Known

Camera calibration

$[R_W A|t_W A]$: Extrinsic parmeter of camera A

$[R_W B|t_W B]$: Extrinsic parmeter of camera B

Correspondence matching

p : A image coordinate of P

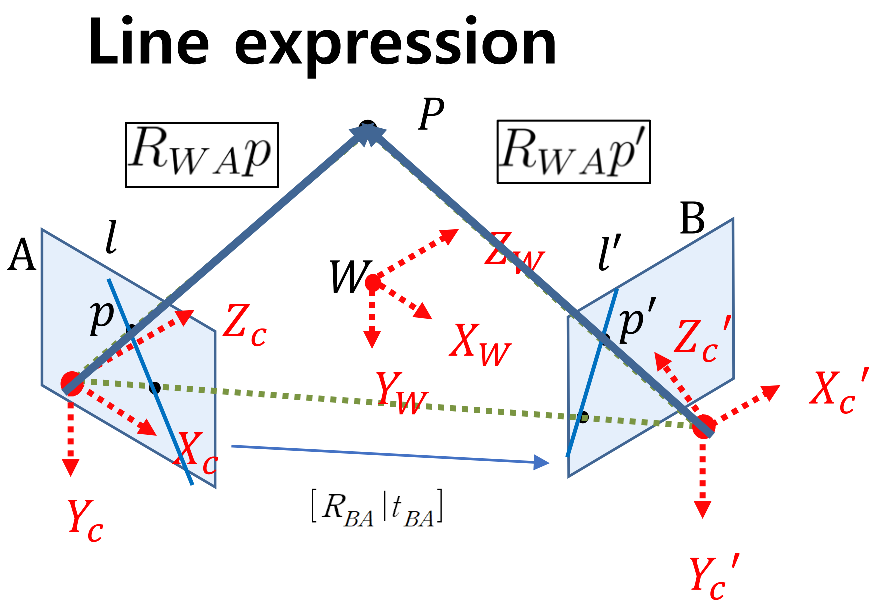
p' : A image coordinate of P

Unknown

P : World cordinate

The process of finding world coordinate

Triangulization



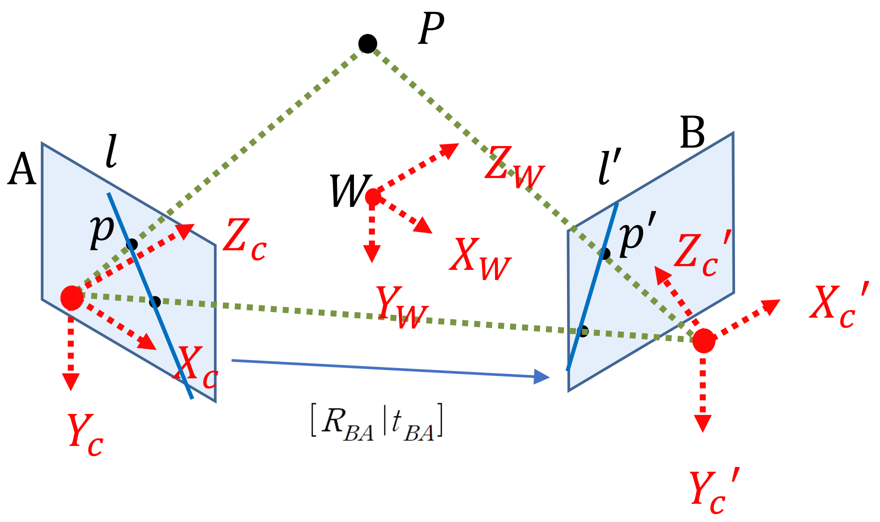
$$m : su + p$$

- m : Line
- u : Direction vector
- p : point

$$m_A : s_A R_{WA} p + t_{WA}$$

$$m_B : s_B R_{WA} p' + t_{WA}$$

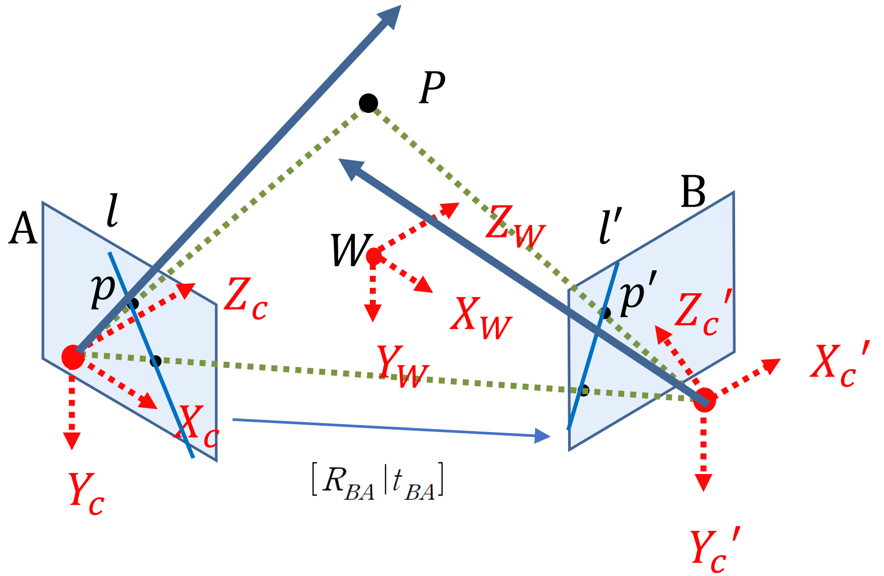
Intersection point of two lines, m_1 and m_2 $\longrightarrow$ P



Directly using two image with arbitrary camera position

→ Some considerations

Considering



1. Existence of intersection point

- The loss of spatial information continuity

- Camera calibration Error, Pixel Noise

⟶ The intersection of the two lines does not exist

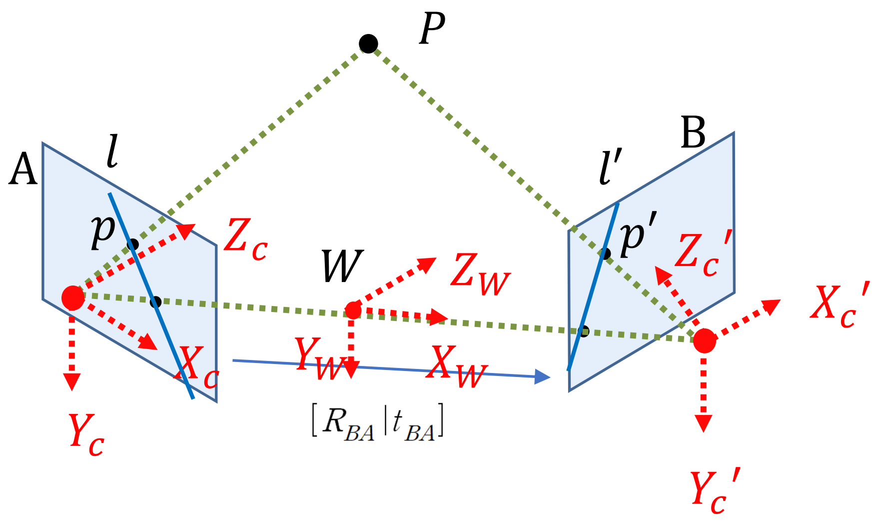
Considering

2. We can set 1D coordinate

- we don't know all world coordinate

- We can set proper coordinate for simple problem

⟶ For calculating depth, we only know one coordinate of z

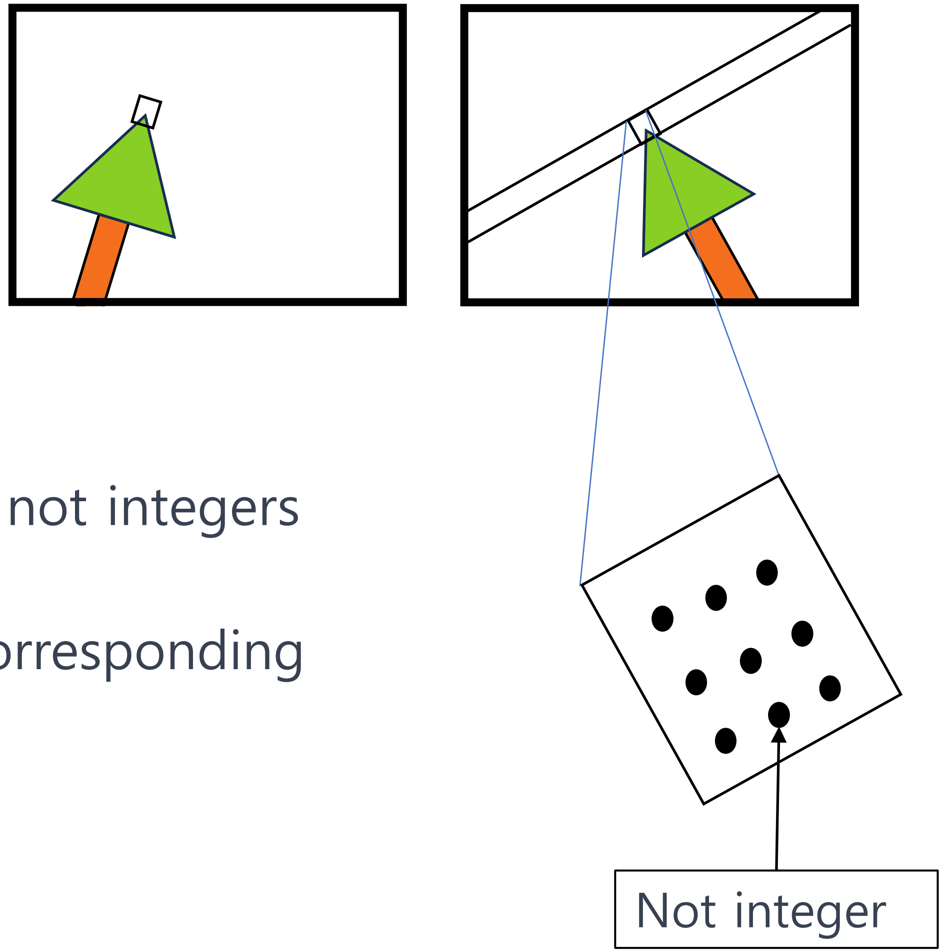


X_W : Baseline direction

Z_W : Desired Depth direction

Considering

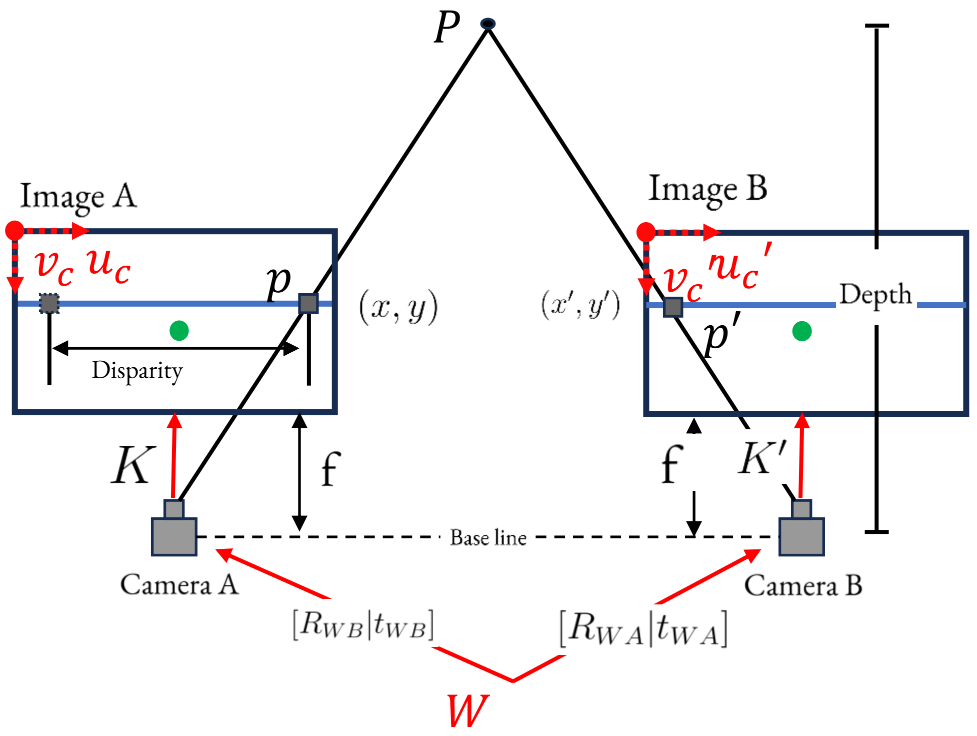
3. Problem with Block matching Algorithm



- The coordinates of the points inside the block are not integers

- Epiline is formed diagonally, it is difficult to find corresponding points along the Epiline

Triangulization(Ideal Modeling)



Camera calibration

$$\lambda \tilde{m} = K[R_{CW}|t_{CW}]\tilde{w}$$

- $\tilde{m}$: Image coordinate
- K : Intrinsic parameter
- $[R_{CW}|t_{CW}]$: Extrinsic parameter
- $\tilde{w}$: World cordinate

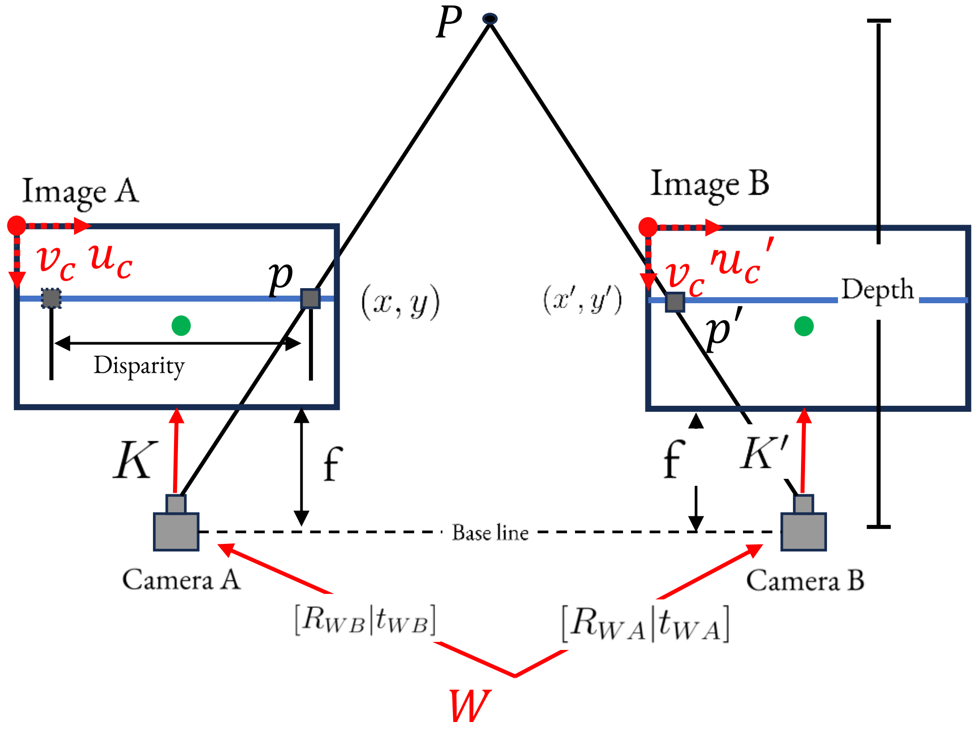
Same Intrinsic parameter

$K = K'$ $\longrightarrow$
- Same principal point coordinate
- Same Image ratio

Same Rotation matrix

$R_{WA} = R_{WB}$ $\longrightarrow$ Same camera posture

Triangulization(Ideal Modeling)



Important Effect of Ideal Modeling

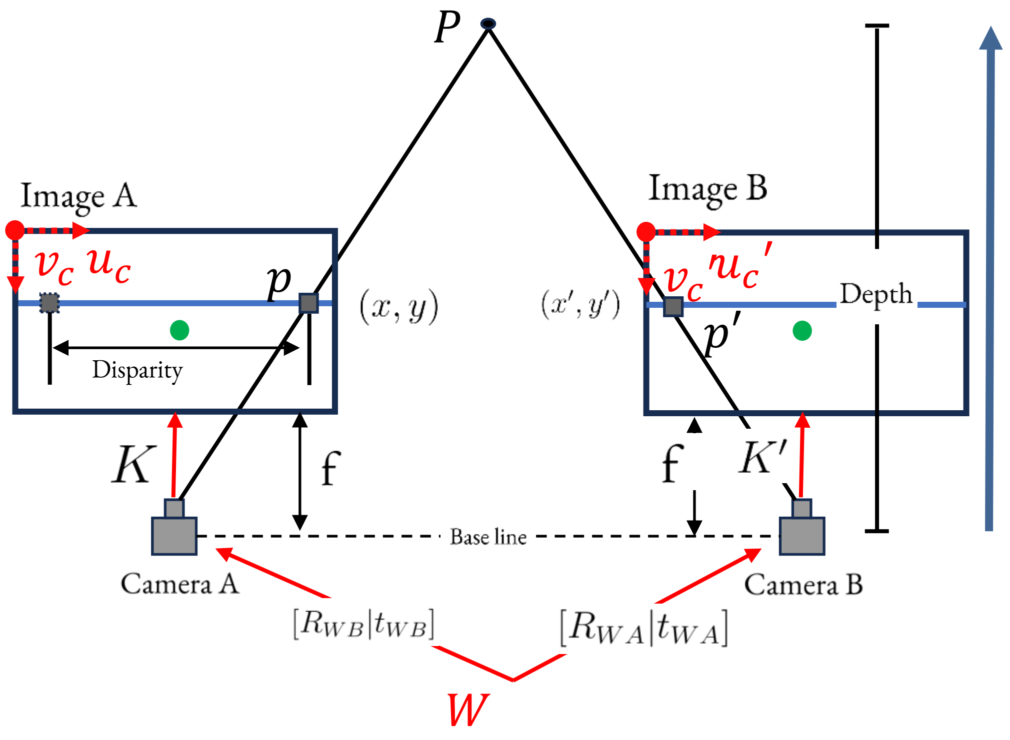
- The two image planes exist within the same plane
- All Epiline is always parallel to the horizontal axis

Because

Intersection line of Epipolar plane and image plane are always parallel to Baseline

Triangulization(Ideal Modeling)

Recall three considerations



Existence of intersection point

We don't consider intersection point

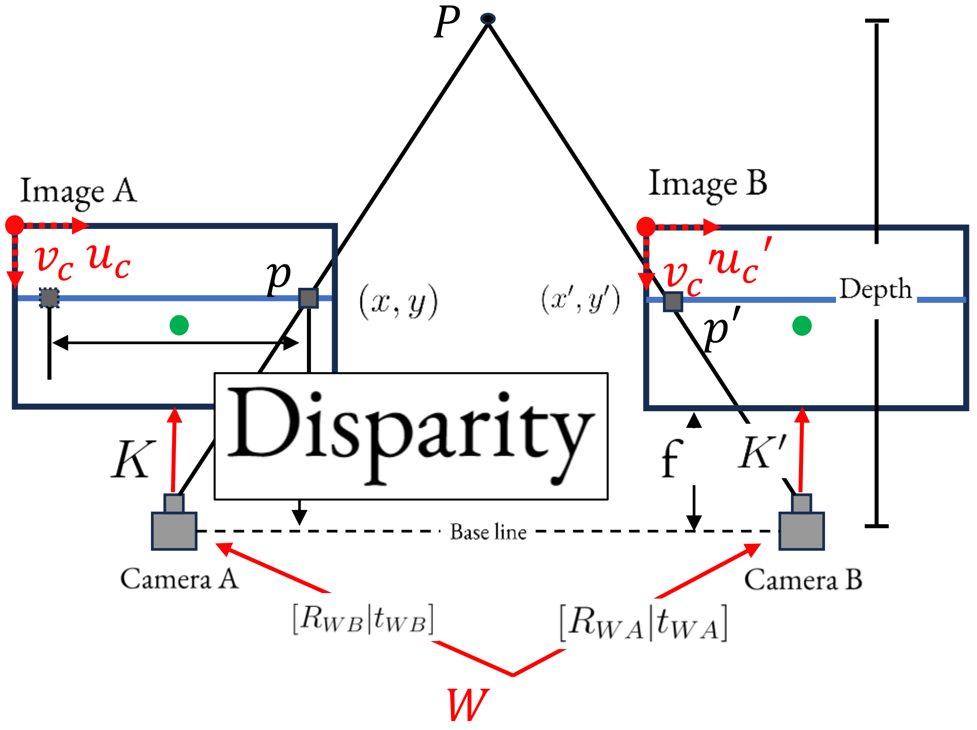
We can set 1D coordinate

Depth calculation in the direction the camera is facing is very easy

Problem with Block matching Algorithm

Epiline is parallel to the horizontal axis

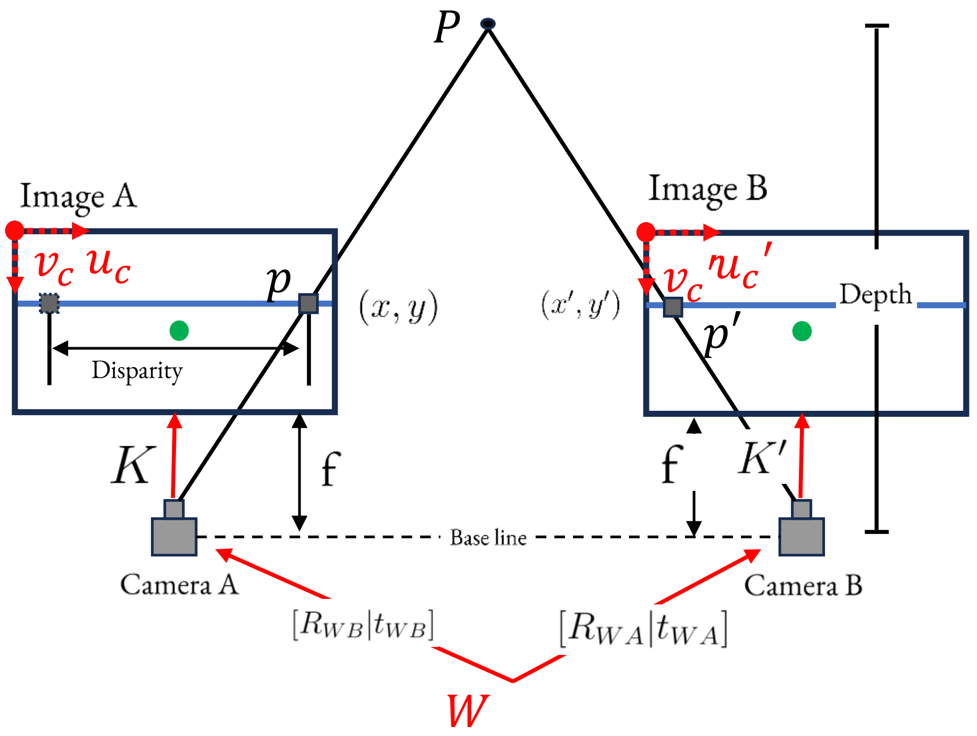
Triangulization(Ideal Modeling)



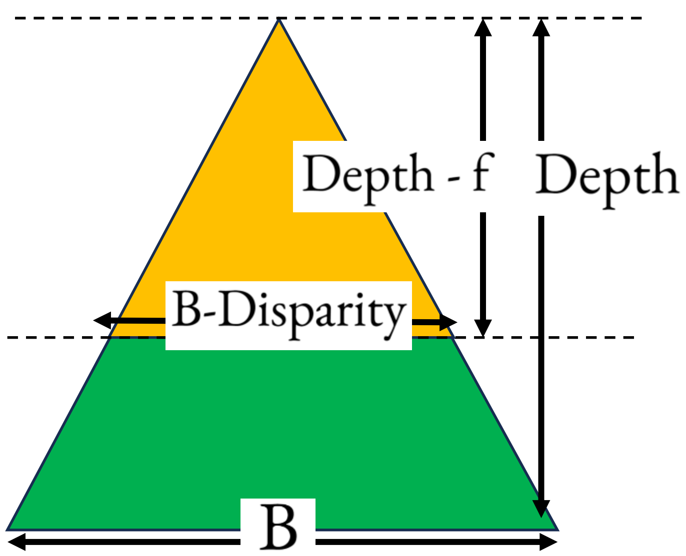
Disparity

horizontal pixel shift between correspondence point in a pair of stereo images

Triangulization(Ideal Modeling)



Simple Triangulization

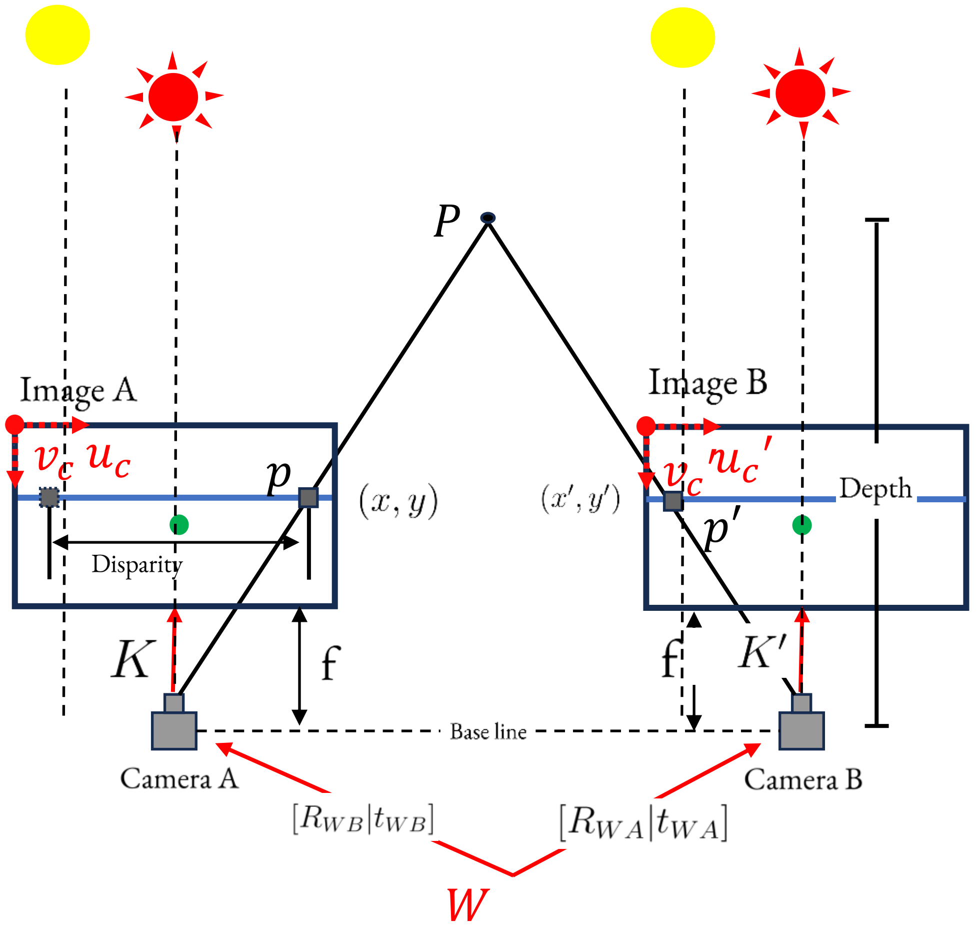


$$Depth - f : Depth = B - Disparity : B \longrightarrow Depth = \frac{f \times B}{Disparity}$$

In ideal modeling

Why should the Intrinsic parameters be the same?

Triangulization(Ideal Modeling)

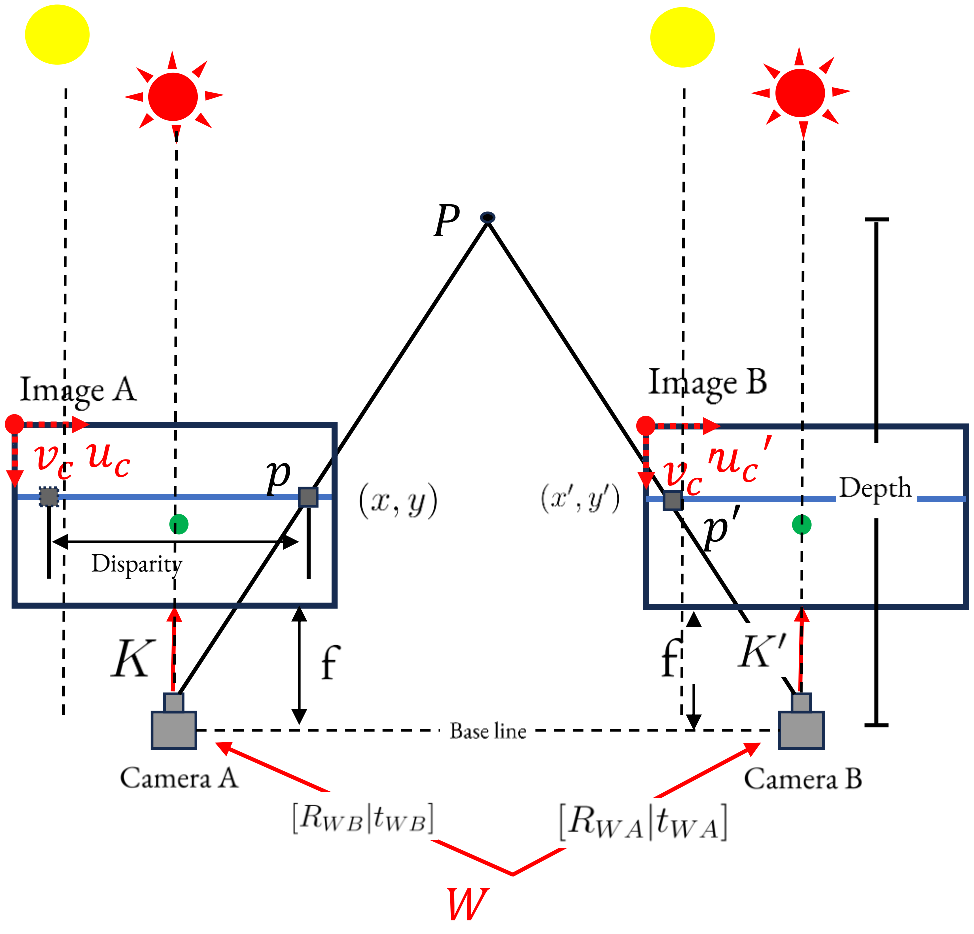


$$Depth = \frac{f \times B}{Disparity}$$

$$\lim_{Depth \to \infty} Disparity = 0$$

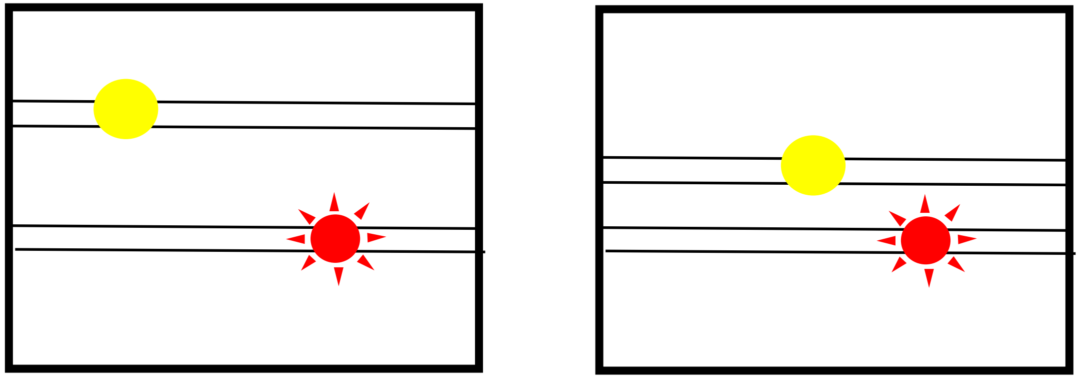
→ Disparity of objects that have very high depth is 0

Triangulization(Ideal Modeling)



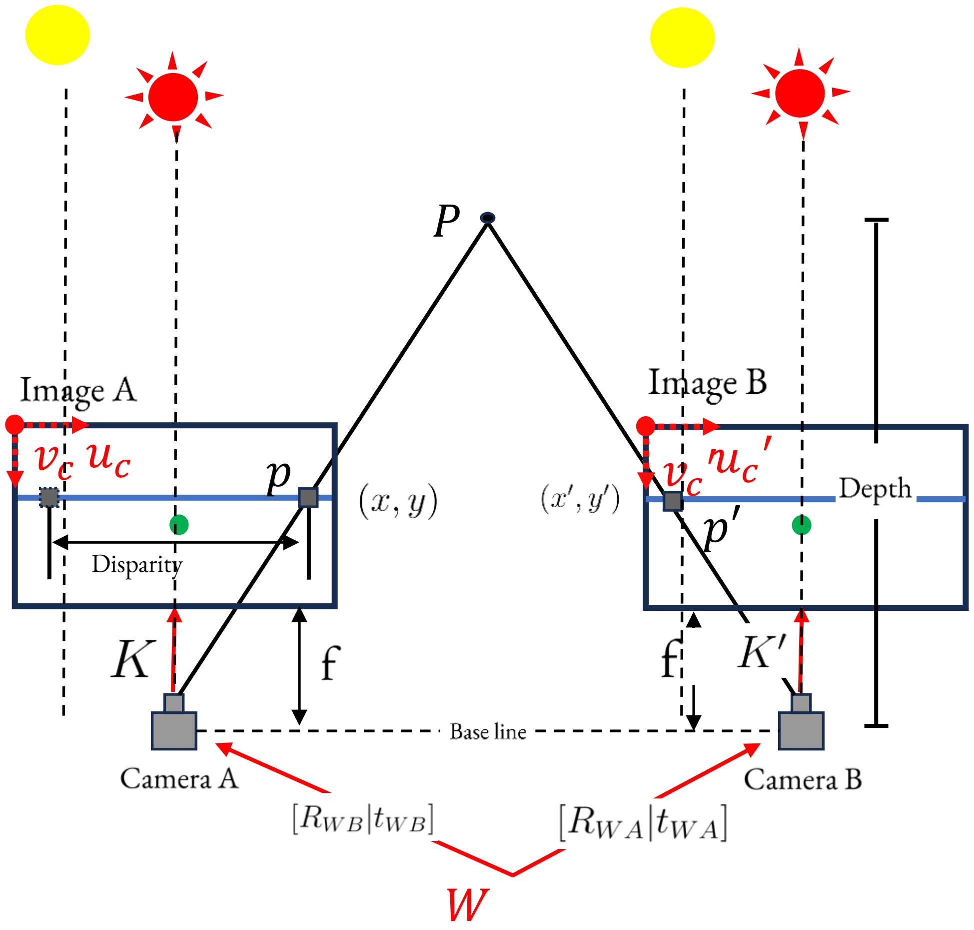
Case 1 (Different f_x, f_y)

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$



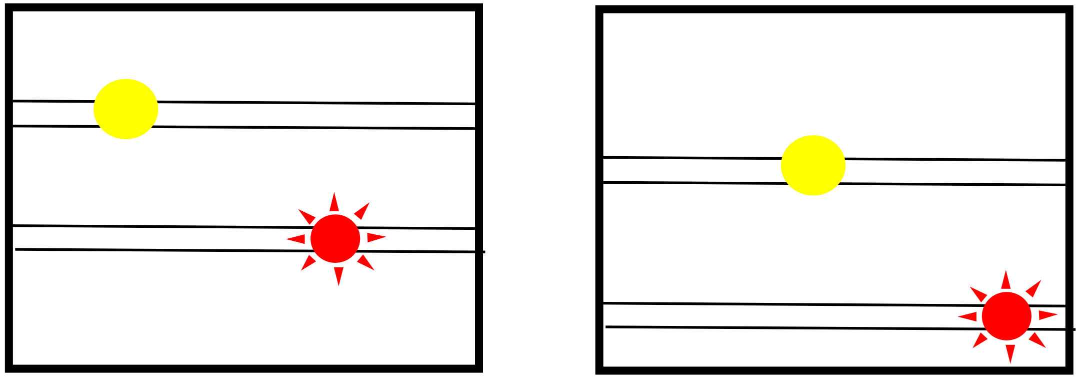
The Epipolar lines of the moon do not have the same vertical axis coordinate

Triangulization(Ideal Modeling)



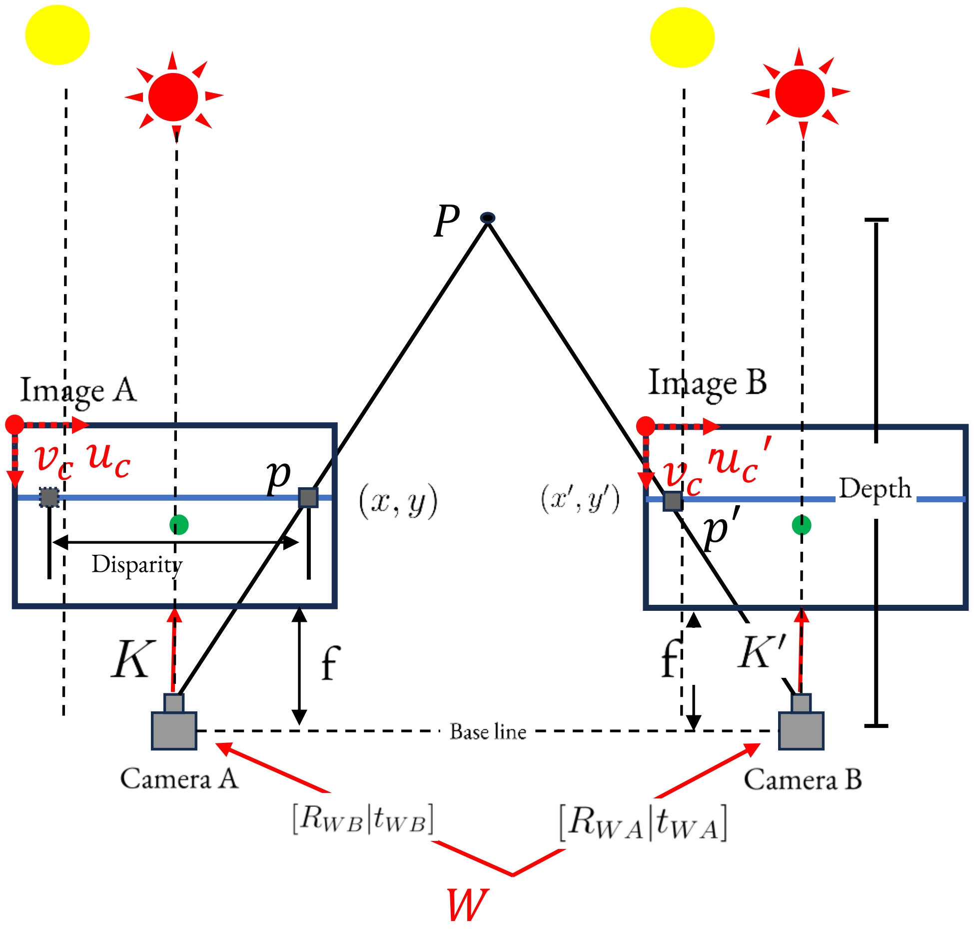
Case 2 (Different c_x, c_y)

$$K = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix}$$

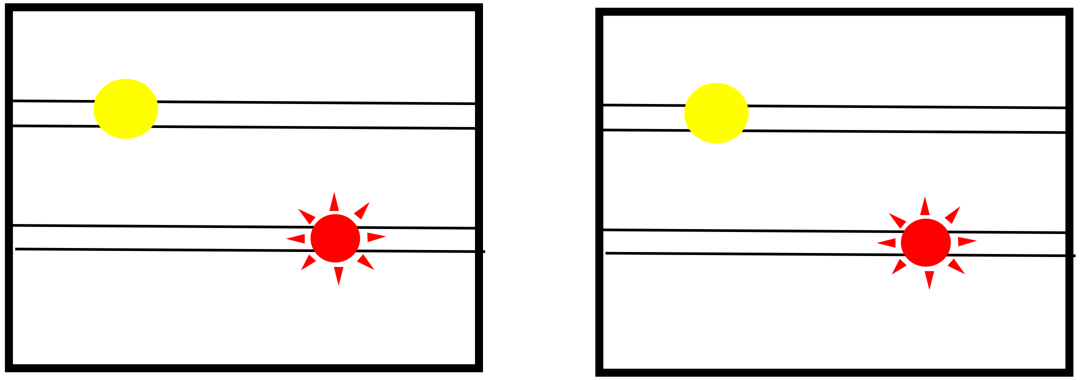


The Epipolar lines of both sun and moon do not have the same vertical axis coordinate

Triangulization(Ideal Modeling)



Case 3 (Same Intrinsic parameter)



Triangulization(Ideal Modeling)

Conclusion

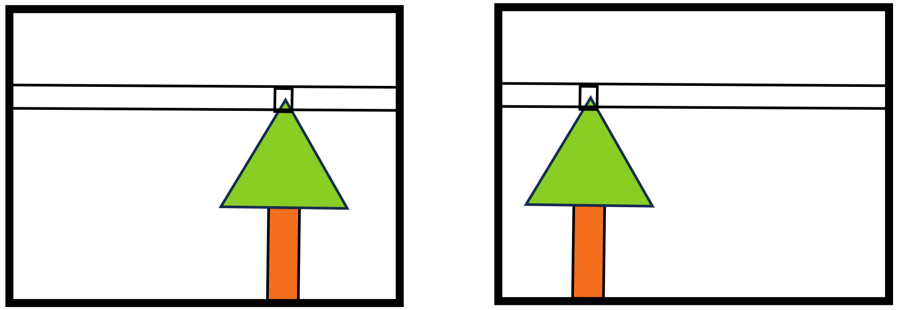
Same Intrinsic parameter

- All Epiline is always parallel to the horizontal axis

Same Camera

- All pair of correspondence points share the same vertical axis coordinates

Simple correspondence matching



Simple triangulization

$$Depth = \frac{f \times B}{Disparity}$$

We have stereo images from various camera position

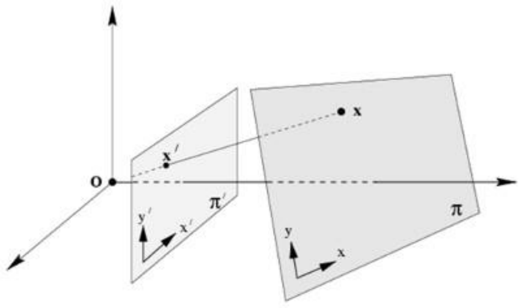
→ Rectification

Rectification

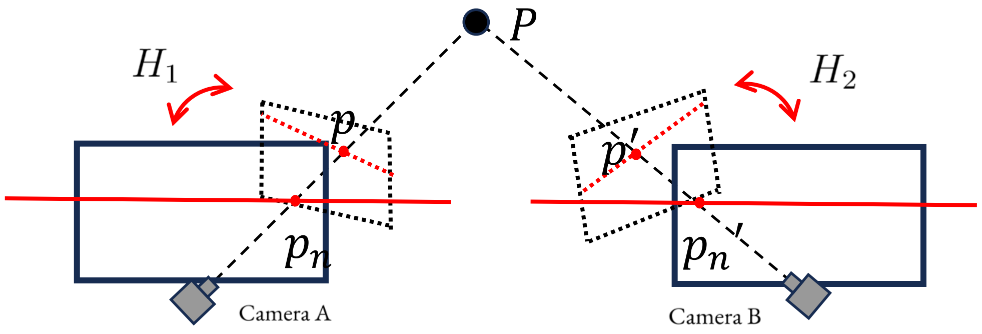
Homography

- 3x3 matrix representing a projective transformation between two planes

- Two images taken at the same location can be overlapped using homography

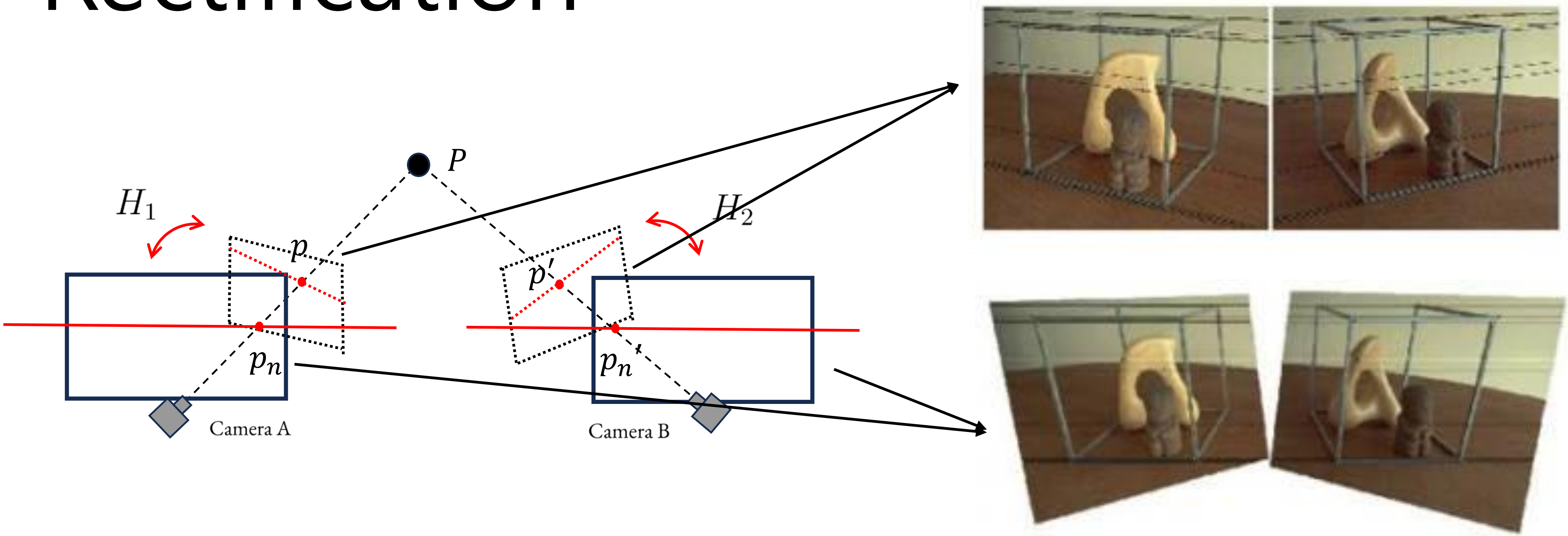


$$w\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$



Using two Homography matrix

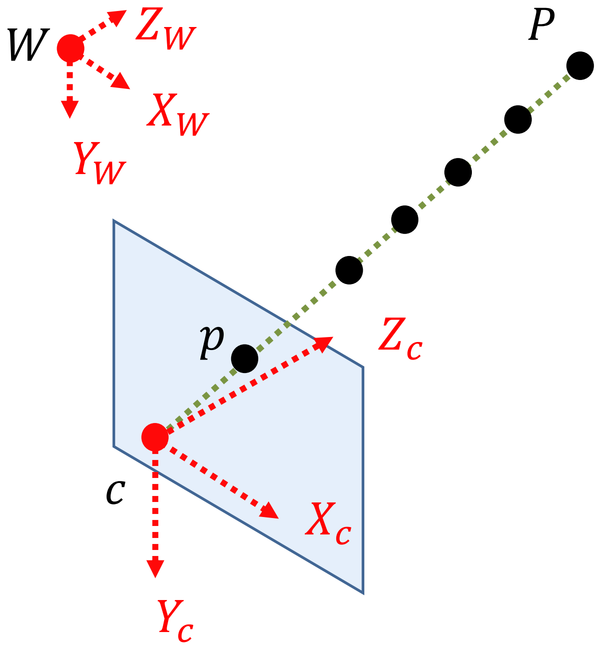
Rectification



Object

- All apilines are parallel to the horizontal axis of the image plane
- Two image coordinates for the same point P have the same vertical coordinates

Rectification



Camera calibration

$$\lambda \tilde{m} = K[R_{CW}|t_{CW}]\tilde{w}$$

- $\tilde{m}$: Image coordinate
- K : Intrinsic parameter
- $[R_{CW}|t_{CW}]$: Extrinsic parameter
- $\tilde{w}$: World cordinate

Camera center

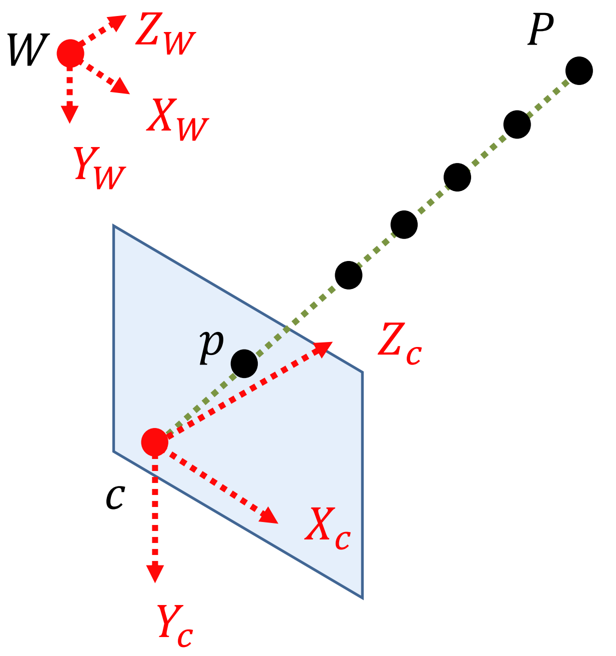
$$c = -R_{CW}^{-1} t_{CW}$$
$$t_{CW} = -R_{CW} c$$

c : World coordinate of camera center

Projection matrix

$$\tilde{P} = K[R_{CW}|t_{CW}] = K[R_{CW}| - R_{CW}c] \quad (\text{where } Q = KR_{CW})$$
$$\tilde{P} = [Q| - Qc]$$

Rectification



Camera calibration

$$\lambda\tilde{m} = K[R_{CW}| - R_{CW}c]\tilde{w}$$

- $\tilde{m}$: Image coordinate
- K : Intrinsic parameter
- $[R_{CW}|t_{CW}]$: Extrinsic parameter
- $\tilde{w}$: World cordinate

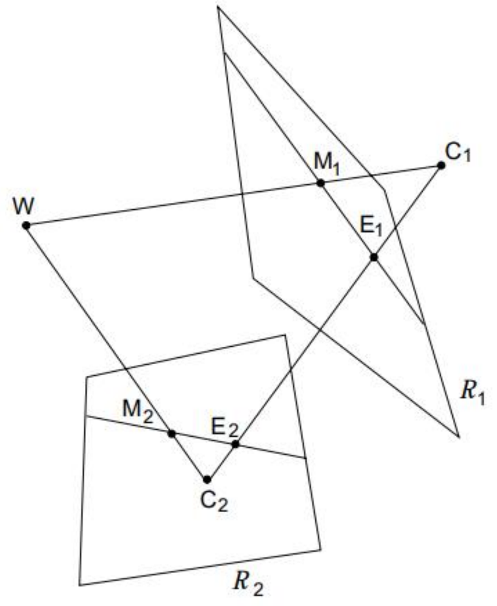
World cordinate

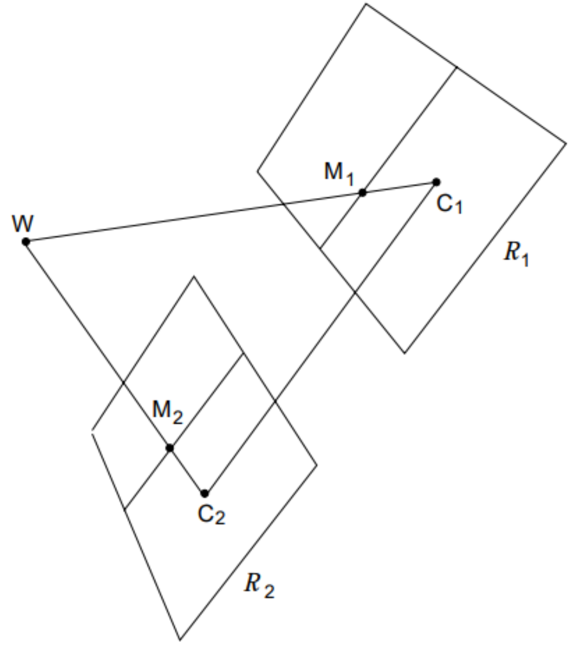
$$\tilde{w} = \begin{pmatrix} w \\ 1 \end{pmatrix} \longrightarrow \lambda\tilde{m} = K[R_{CW}|-R_{CW}\,c]\begin{pmatrix} w \\ 1 \end{pmatrix}$$

$$\lambda\tilde{m} = KR_{CW}\,(w - c)$$

$$w = c + (KR_{CW})^{-1}\lambda\tilde{m}$$

Rectification



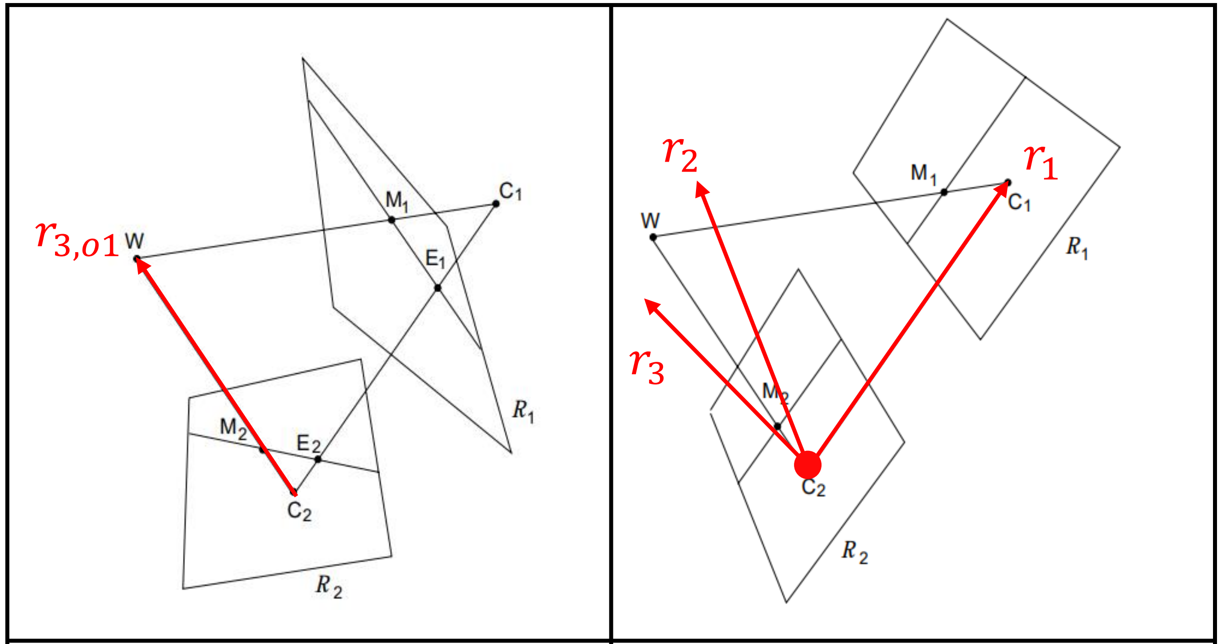


$$\tilde{P}_o1 = K_{o1}[R_{o1}| - R_{o1}c_1]$$
$$\tilde{P}_o2 = K_{o2}[R_{o2}| - R_{o2}c_2]$$

$$\tilde{P}_{n1} = K_n[R_n| - R_nc_1]$$
$$\tilde{P}_{n2} = K_n[R_n| - R_nc_2]$$

Rectification：The process of making a virtual camera

Rectification



$$\tilde{P}_o1 = K_{o1}[R_{o1}| - R_{o1}c_1]$$
$$\tilde{P}_o2 = K_{o2}[R_{o2}| - R_{o2}c_2]$$

$$\tilde{P}_{n1} = K_n[R_n| - R_n c_1]$$
$$\tilde{P}_{n2} = K_n[R_n| - R_n c_2]$$

$\boldsymbol{R_n}$

$$R = \begin{bmatrix} r_1^T \\ r_2^T \\ r_3^T \end{bmatrix}$$

r_1 : camera x-dir (world expression)
r_2 : camera y-dir (world expression)
r_3 : camera z-dir (world expression)

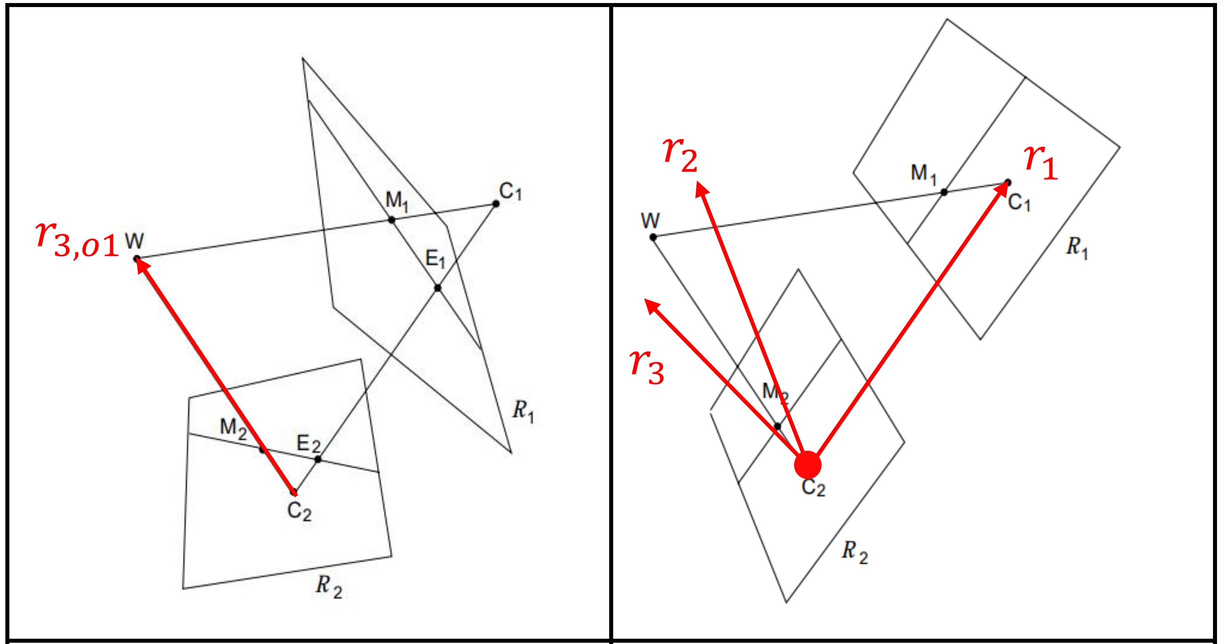
$$\tilde{P}_o1 = [Q_{o1}| - Q_{o1}c_1]$$
$$\tilde{P}_o2 = [Q_{o2}| - Q_{o2}c_2]$$

$\longrightarrow$ c_1, c_2

$$r_1 = \frac{c_1 - c_2}{\|c_1 - c_2\|} \qquad r_2 = r_{3,o1} \times r_1 \qquad r_3 = r_1 \times r_2$$

$r_3, o1$: The principal axis dir (old camera)

Rectification



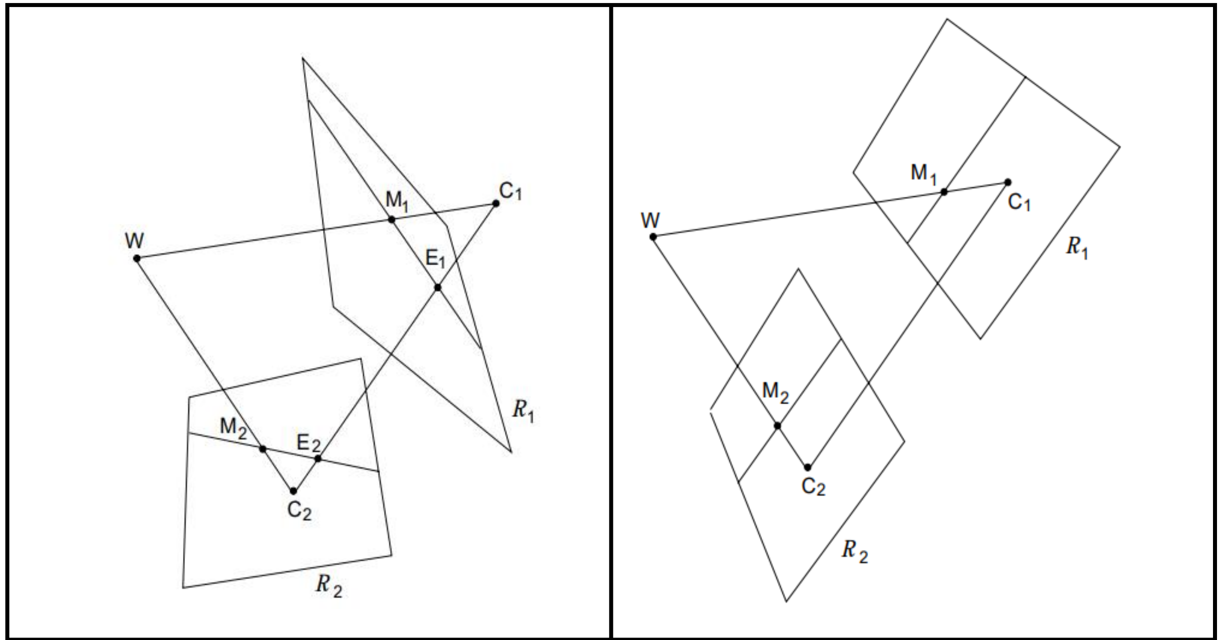
$\tilde{P}_o1 = K_{o1}[R_{o1} \| -R_{o1}c_1]$ $\tilde{P}_o2 = K_{o2}[R_{o2} \| -R_{o2}c_2]$	$\tilde{P}_{n1} = K_n[R_n \| -R_n c_1]$ $\tilde{P}_{n2} = K_n[R_n \| -R_n c_2]$

$\boldsymbol{A_n}$

$$A_n = (A_1 + A_2)/2$$

By averaging the two intrinsic parameter

Rectification



$$\tilde{P}_o1 = K_{o1}[R_{o1}| - R_{o1}c_1]$$
$$\tilde{P}_o2 = K_{o2}[R_{o2}| - R_{o2}c_2]$$

$$\tilde{P}_{n1} = K_n[R_n| - R_n c_1]$$
$$\tilde{P}_{n2} = K_n[R_n| - R_n c_2]$$

$$w = c_1 + \lambda_{o1}(K_{o1}R_{o1})^{-1}\tilde{m}_{o1}$$
$$w = c_1 + \lambda_{n1}(K_nR_n)^{-1}\tilde{m}_{n1}$$

The world coordinate is same

$$\tilde{m}_{n1} = \lambda_1'\underline{(K_nR_n)(K_{o1}R_{o1})^{-1}}\tilde{m}_{o1}$$

$$\downarrow$$

$$H_1 = (K_nR_n)(K_{o1}R_{o1})^{-1}$$

H_2 is also calculated in the same way

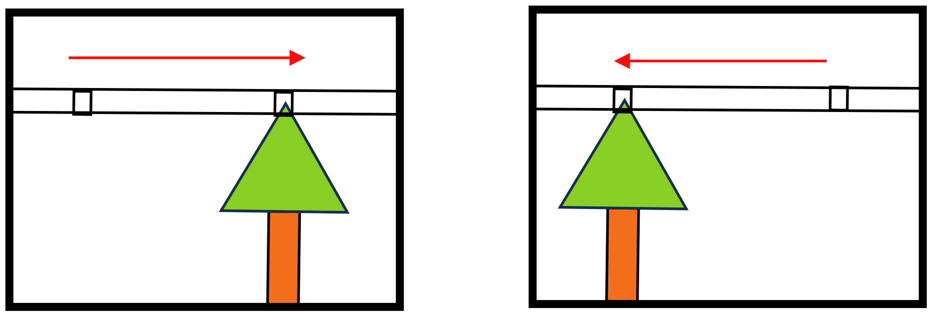
Algorithm design

Rectification → We can assume ideal modeling in designing algorithm

- All Epiline is always parallel to the horizontal axis

- All pair of correspondence points share the same vertical axis coordinates

---



- Starting from own coordinates and moving along the horizontal axis (Direction is different)

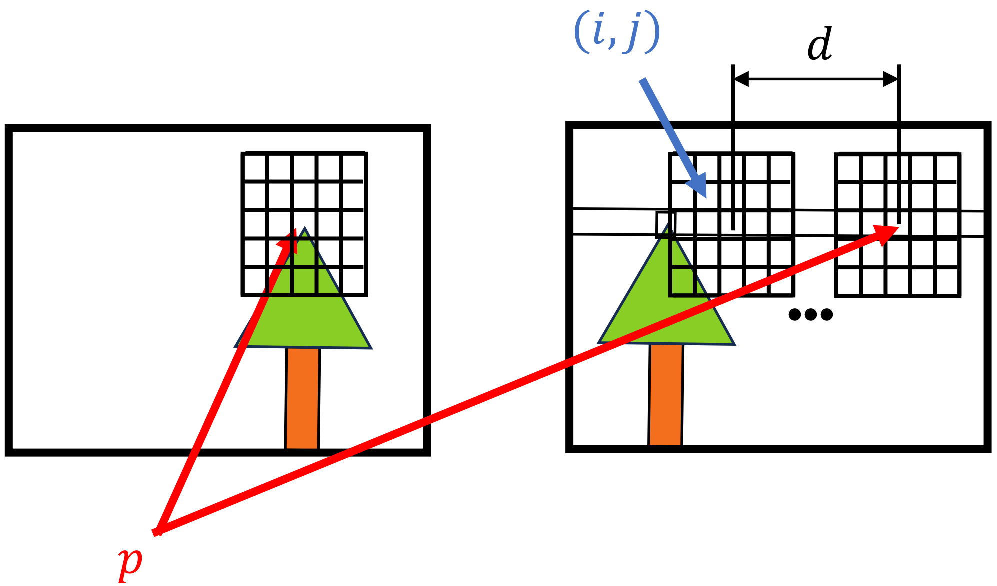
- Calculating the disparity with the corresponding point

$$Depth = \frac{f \times B}{Disparity}$$

→ **Disparity Map** → **Depth Map**

Block Matching

Comparing pixels on a block-by-block basis to find matching points



- Comparing pixels one by one has very low accuracy
- It measures how well the features within the block match each other

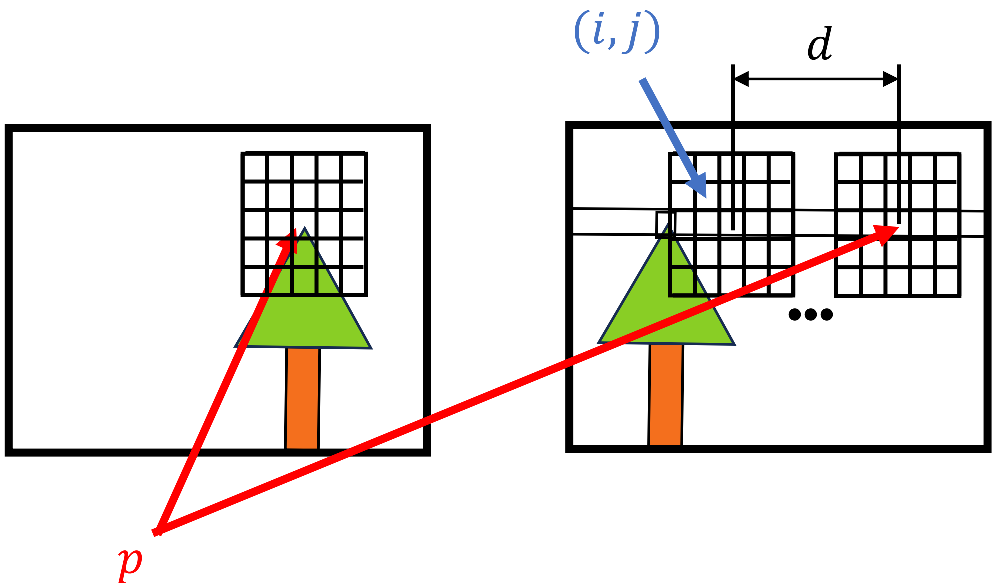
→ **Cost function**

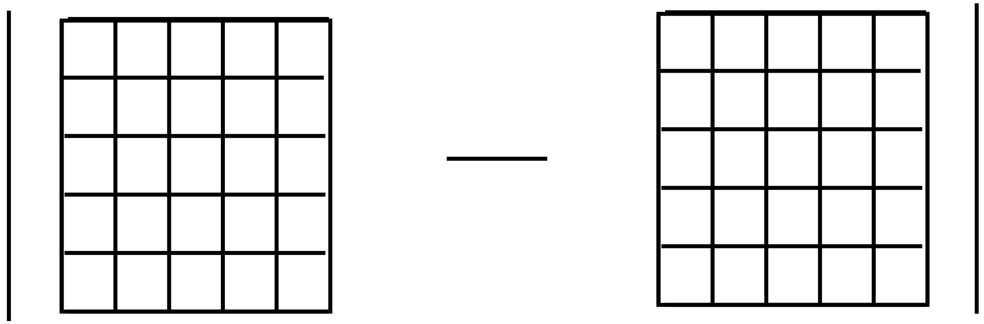
$$C(p,d) = \sum_{(i,j)\in W} f(i,j,p,d)$$

Matching Cost

Sum of Absolute Differences

$$C(p,d) = \sum_{(i,j)\in W} |I_l(i,j) - I_r(i-d,j)|$$



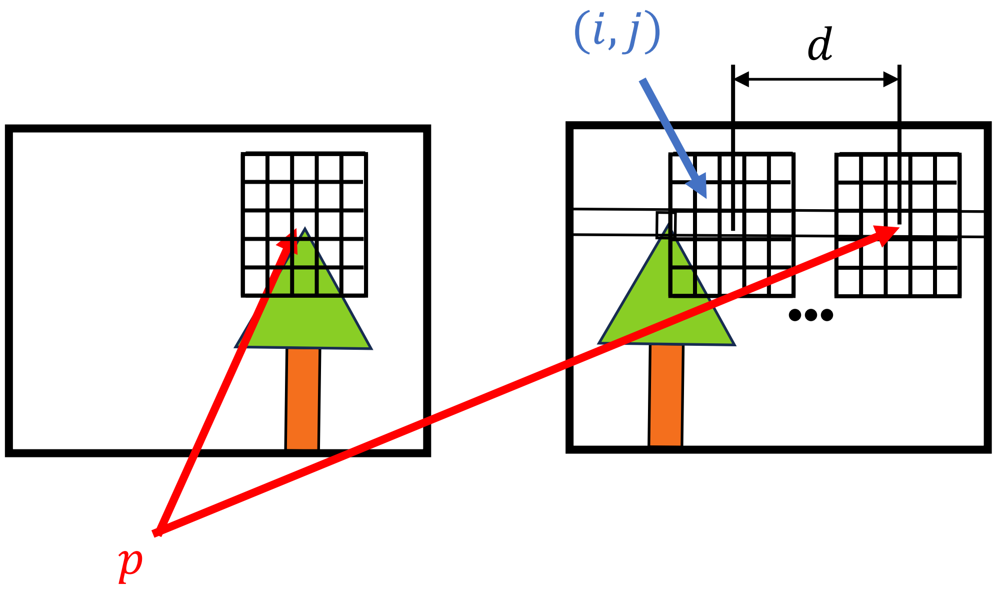


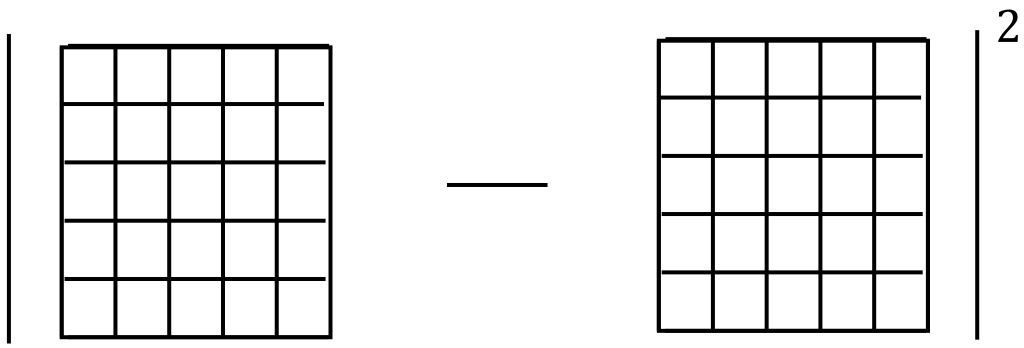
- SAD is sensitive to noise and brightness change
- Feature matching is good when cost is low

Matching Cost

Sum of Squared Differences

$$C(p,d) = \sum_{(i,j)\in W} |I_l(i,j) - I_r(i-d,j)|^2$$



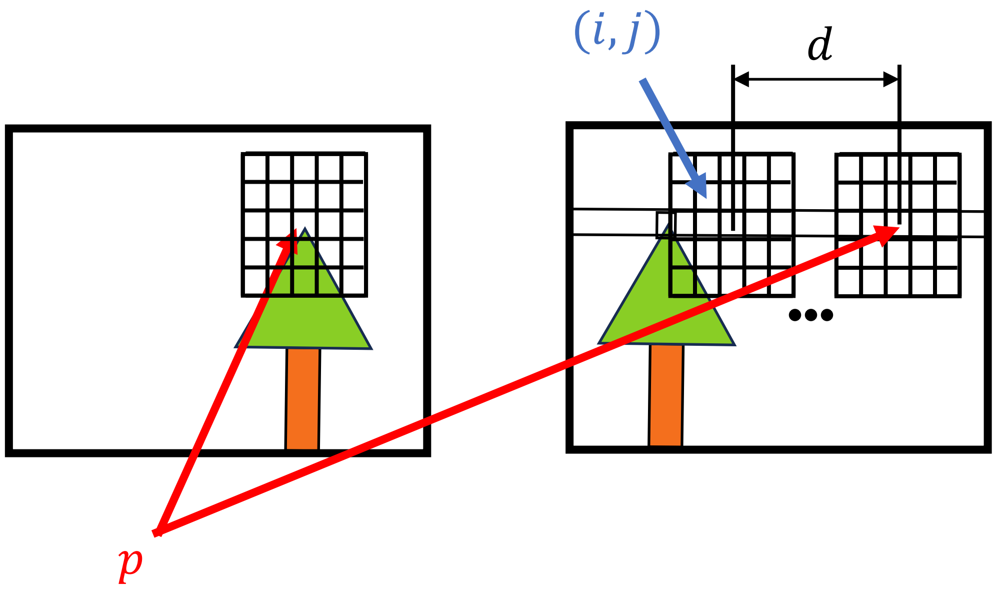


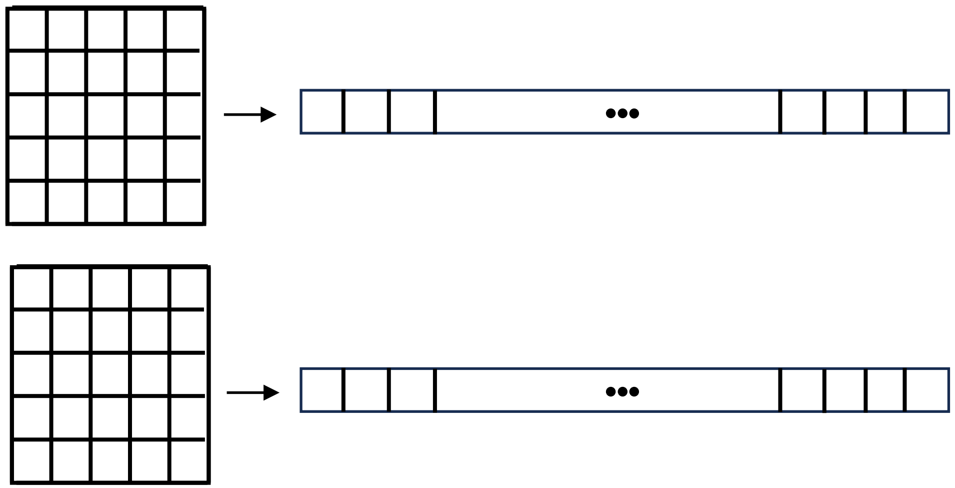
- SAS is sensitive to noise and brightness change
- Feature matching is good when cost is low

Matching Cost

Normalized Cross Correlation

$$C(p,d)=\frac{\sum_{(i,j)\in W} I_l(i,j)I_r(i-d,j)}{\sqrt{\sum_{(i,j)\in W} I_l^2(i,j)\sum_{(i,j)\in W} I_r^2(i-d,j)}}$$

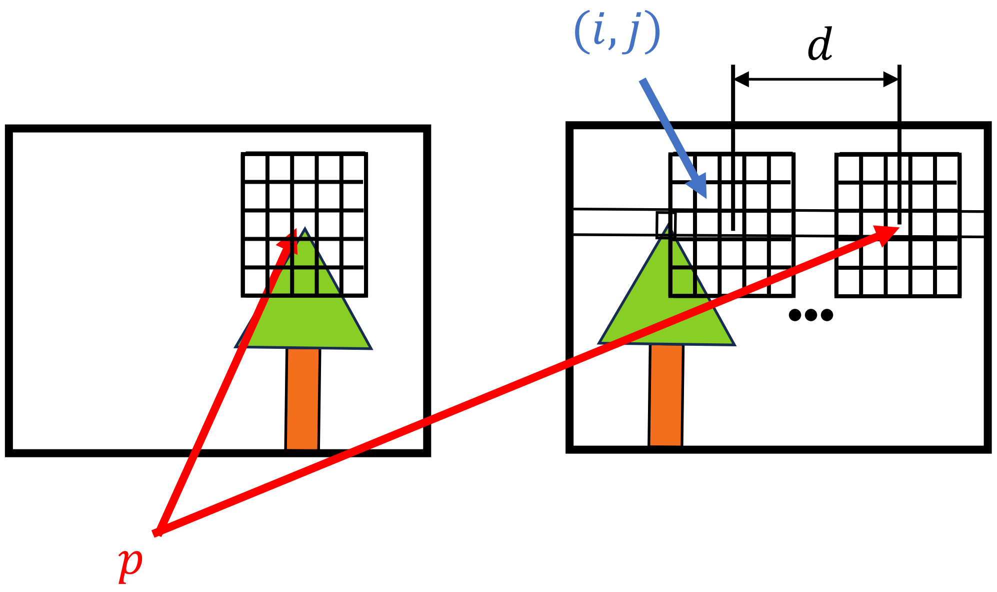


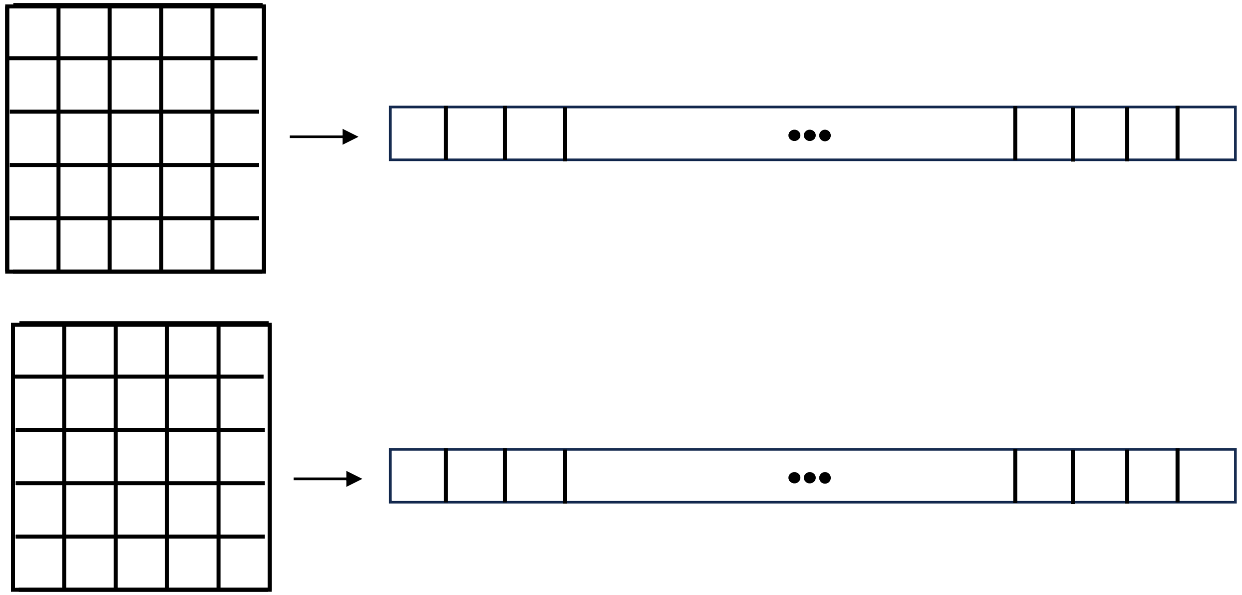


- NCC is less sensitive to noise and brightness change
- Feature matching is good when cost is closer to 1

Matching Cost

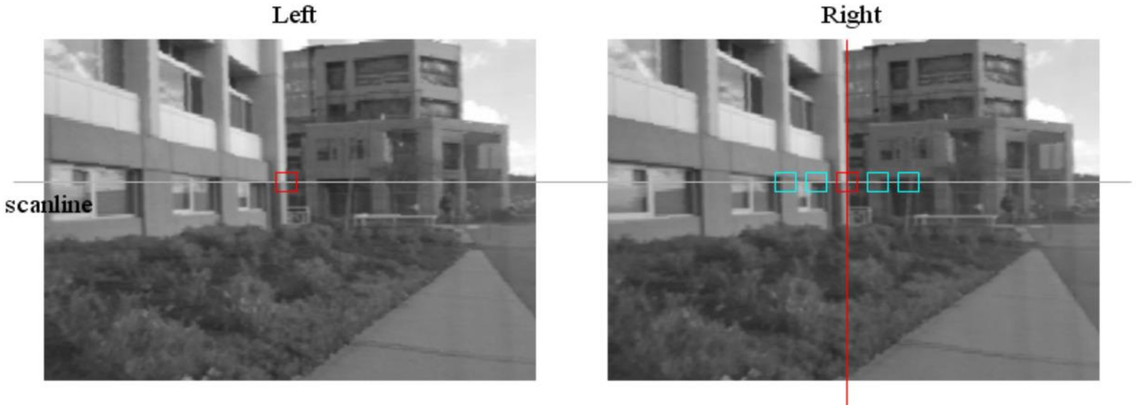
Census transform





- If pixel values are higher than the central pixel value, assign 1
- If two arrays are different, assign 1
- Cost : How many different value in two array

Local Matching



Cost function

$$C(p, d) = \sum_{(i,j)\in W} f(i, j, p, d)$$

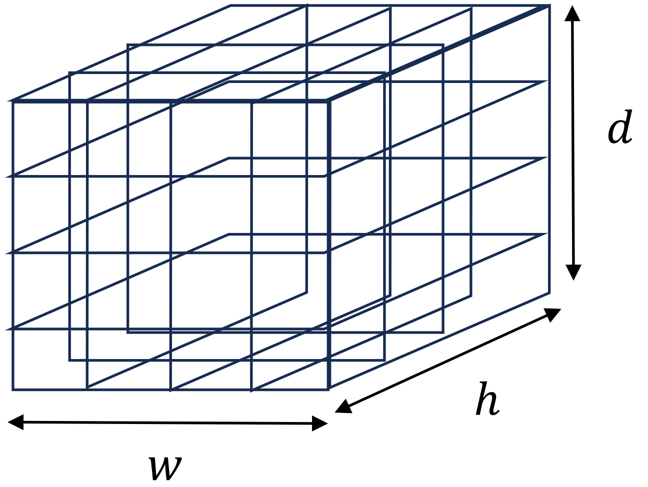
⟶ 3 variable function

Local Matching

1. Cost calculation

2. Making cost volume

2. Select d value at each pixel

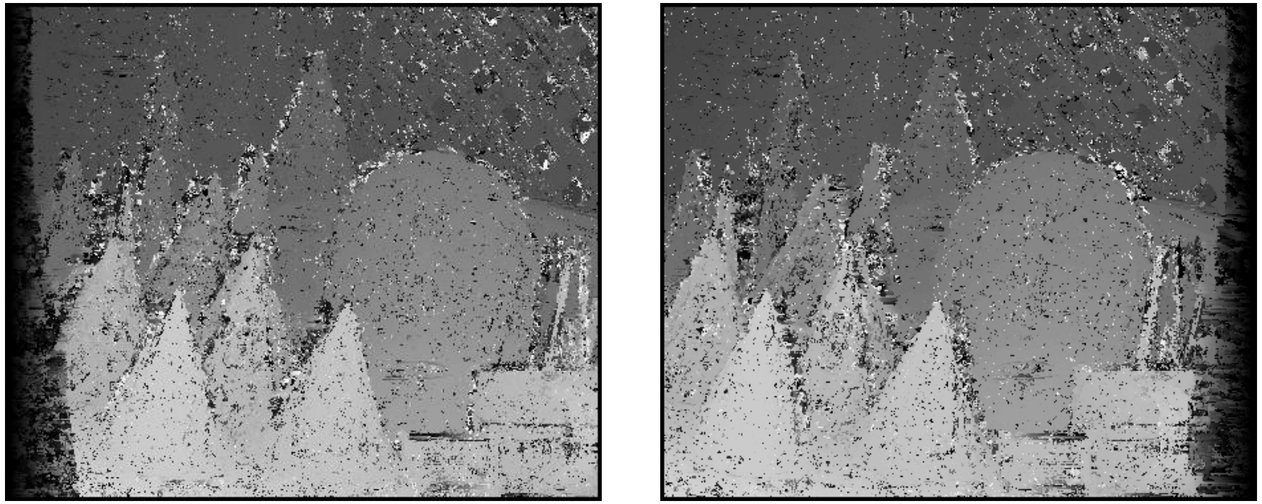


⟶ The process of finding the corresponding point with the highest similarity at each pixel

Local matching

Problem

Continuity of cost volume is not considered



Local matching

pros

- Time complexity of algorithm is low

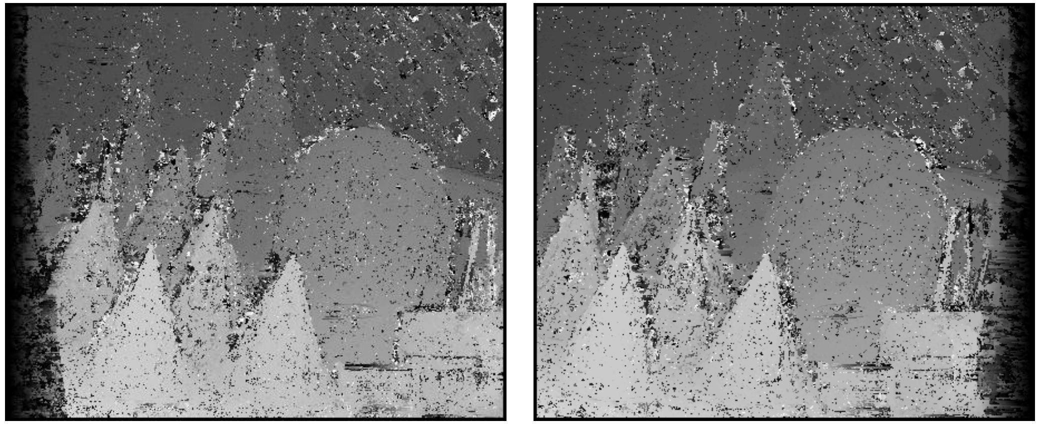
cons

- It is heavily influenced by noise

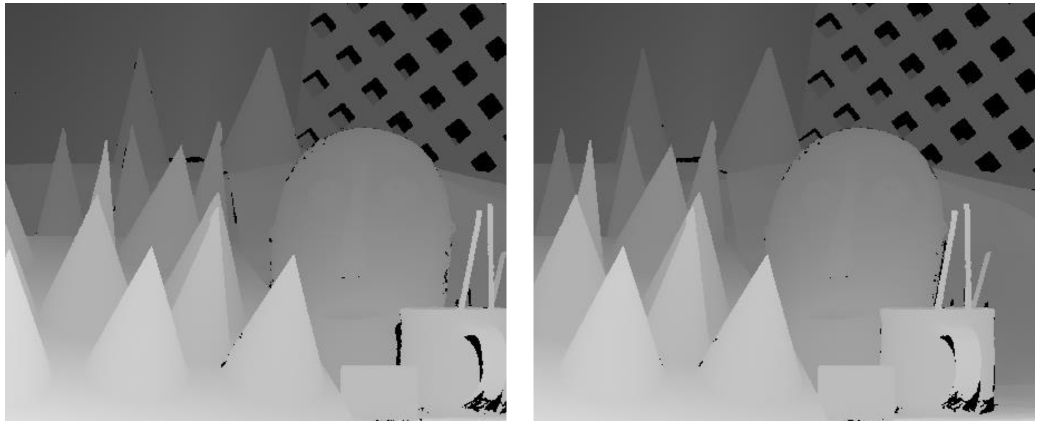
- Impossible to make accurate depth map

- Depth map is not continuos

Local matching

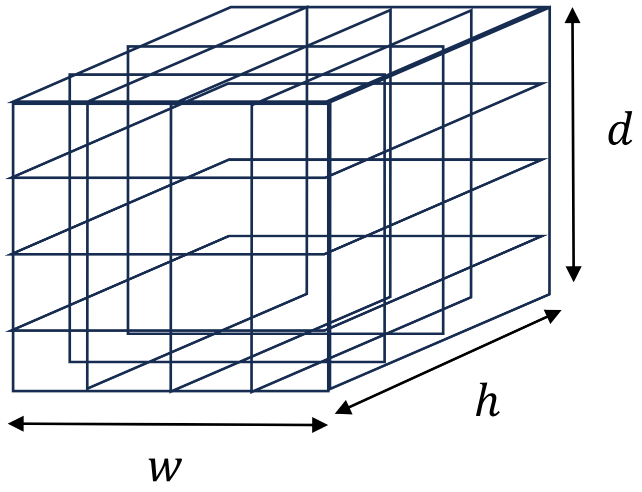


Ground truth



Energy Function

Global Matching



1. Cost calculation
2. Making cost volume
3. Defining Energy function
4. Optimize Energy function

← Same with Local matching

→ That is, Global matching considers the continuity of cost volume

Semi Global Matching

$$\mathcal{C} = \mathcal{C}_{data} + \lambda \mathcal{C}_{discon}$$

$$C_{data}(p,d) = \sum_{p \in W} f(p,d)$$

similarity

continuity

$$\mathcal{C}(d) = \sum_{p} \left(\mathcal{C}_{data}(\mathbf{p}, d_{\mathbf{p}}) + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_1 \cdot T[|d_{\mathbf{p}} - d_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_2 \cdot T[|d_{\mathbf{p}} - d_{\mathbf{q}}| > 1] \right)$$

Semi Global Matching

$$\mathcal{C}(d) = \sum_{p} \left(\mathcal{C}_{data}(\mathbf{p}, d_{\mathbf{p}}) + \sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{p}}} P_1 \cdot T[|d_{\mathbf{p}} - d_{\mathbf{q}}| = 1] + \sum_{\mathbf{q}\in\mathcal{N}_{\mathbf{p}}} P_2 \cdot T[|d_{\mathbf{p}} - d_{\mathbf{q}}| > 1] \right)$$

$\mathcal{N}_{\mathbf{p}}$: local neighborhood around pixel $\mathbf{p}$ in the reference image I

$$T(\arg) = \begin{cases} 1 & (\arg = true) \\ 0 & (\arg = false) \end{cases}$$

P_1 Penalty for small disparity change

P_2 Penalty for large disparity change

Semi Global Matching

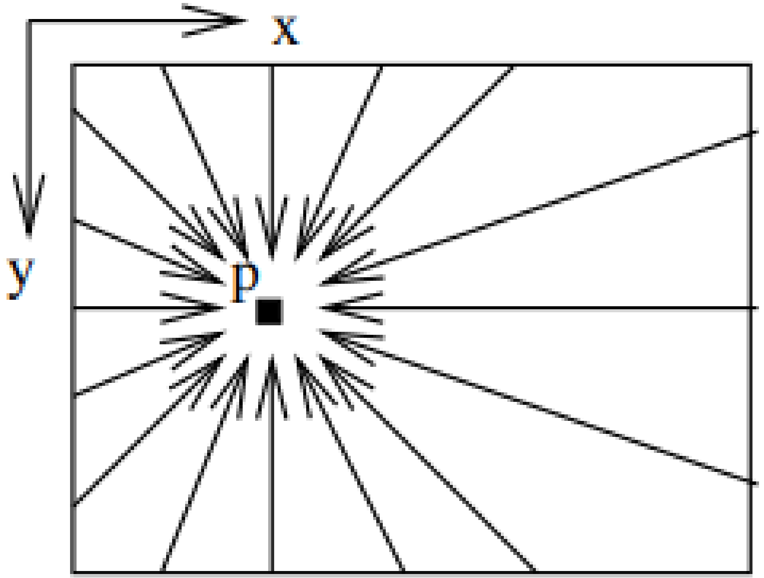
$$\mathcal{C}(d) = \sum_{p} \left(\mathcal{C}_{data}(\mathbf{p}, d_{\mathbf{p}}) + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_1 \cdot T[|d_{\mathbf{p}} - d_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_2 \cdot T[|d_{\mathbf{p}} - d_{\mathbf{q}}| > 1] \right)$$

To determine the disparity of the current pixel, it is necessary to simultaneously determine the disparities of adjacent pixels.

2D global optimization(Simultaneously minimizing the disparity value for all image pixels)

➡ N-P complete problem(The time complexity increases exponentially)

Semi Global Matching



Assumption

Adjacent pixels coming from different directions do not influence each other

Result

2D optimization -> Several 1D optimization

Semi Global Matching

Energy function

$$\mathcal{C}(d) = \sum_{p} \left(\mathcal{C}_{data}(\mathbf{p}, d_{\mathbf{p}}) + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_1 \cdot T[|d_{\mathbf{p}} - d_{\mathbf{q}}| = 1] + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_2 \cdot T[|d_{\mathbf{p}} - d_{\mathbf{q}}| > 1] \right)$$

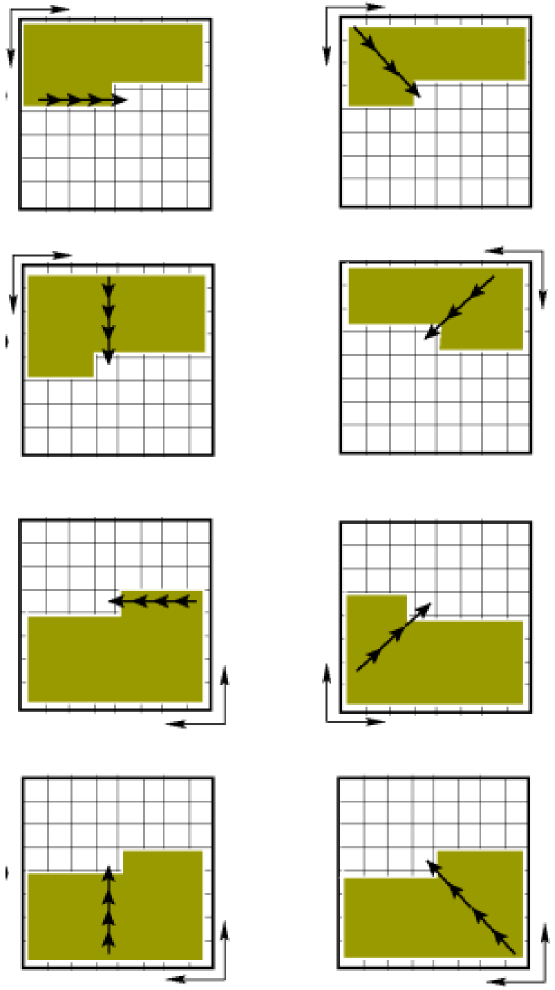
Path cost

$$L_r(p,d) = C(p,d) + \min \begin{Bmatrix} L_r(p-r,d), \\ L_r(p-r,d \pm 1) + P_1, \\ \min L_r(p-r,k) + P_2 \end{Bmatrix} - \min L_r(p-r,k)$$

Total cost

$$C(p,d) = \sum_{r} L_r(p,d)$$

Semi Global Matching



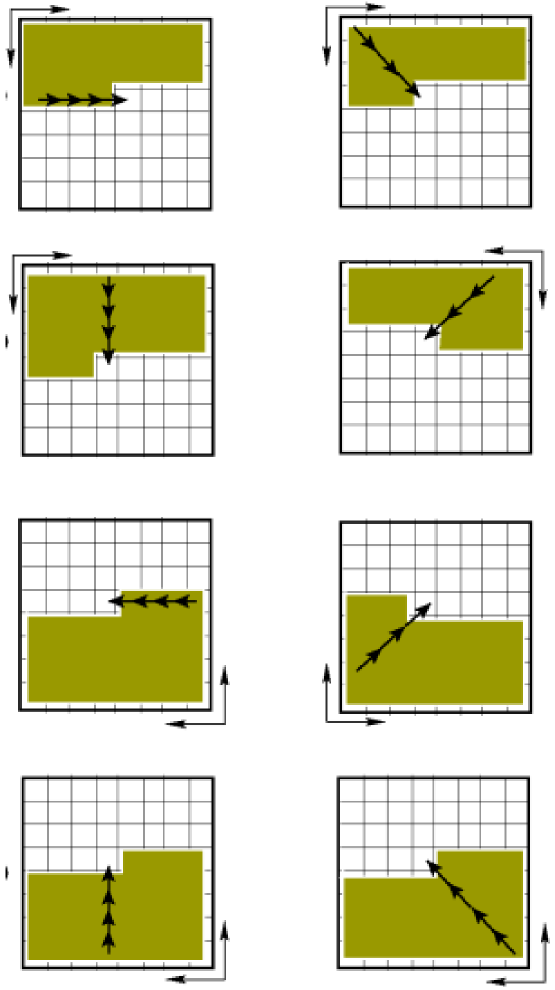
Dynamic Programming

$$L_r(p,d) = C(p,d) + \min \begin{bmatrix} L_r(p-r,d), \\ L_r(p-r,d\pm 1) + P_1, \\ \min L_r(p-r,k) + P_2 \end{bmatrix} - \min L_r(p-r,k)$$

$L_r(p,d)$ ➡ **4 variable function**

store path cost in $(width, height, depth, path)$ array

Semi Global Matching

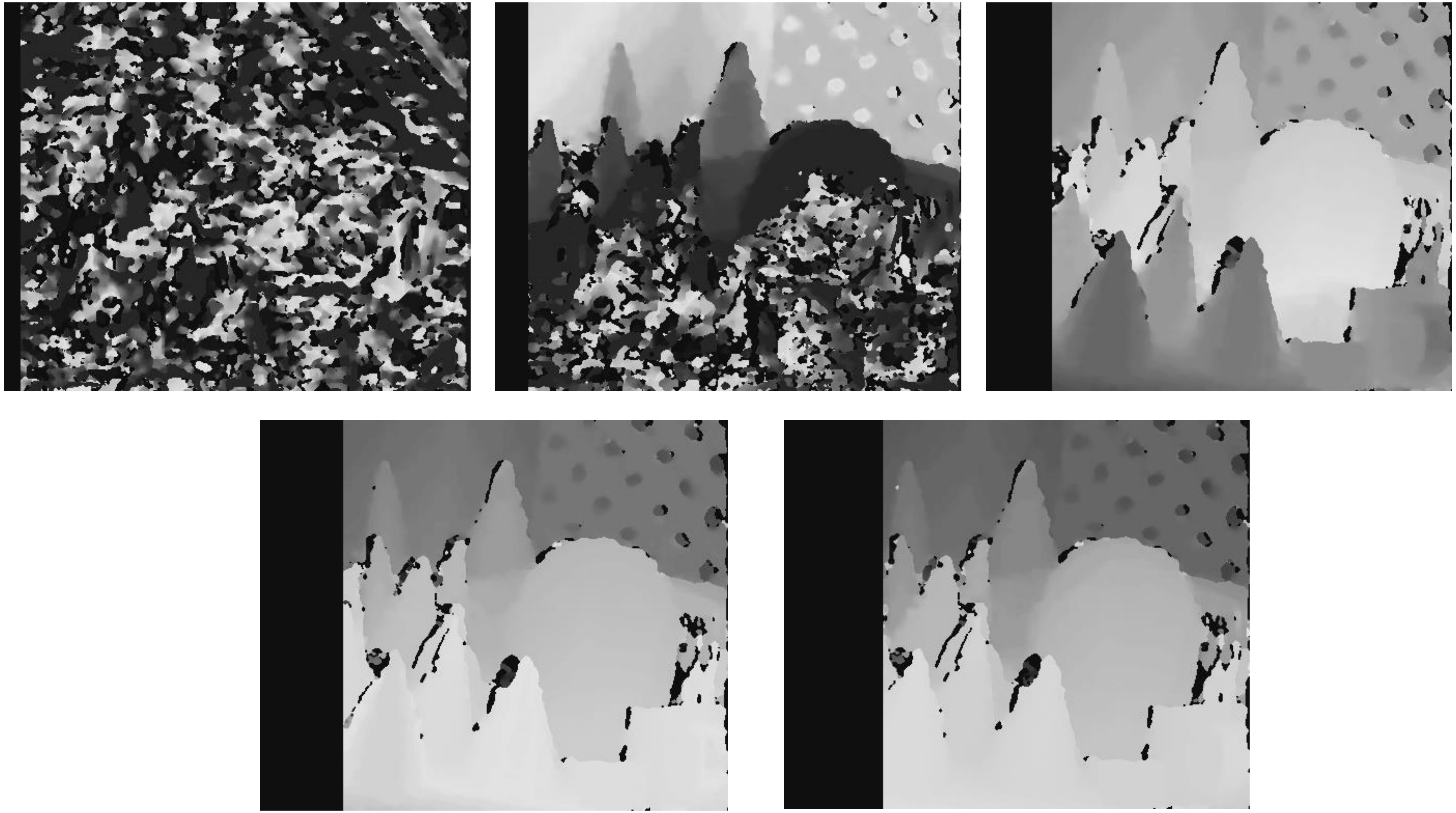


Parallel processing

SGBM algorithm can perform parallel processing when calculating path costs for each direction

➡ Advantage for real time processing

Semi Global Matching



Thank You