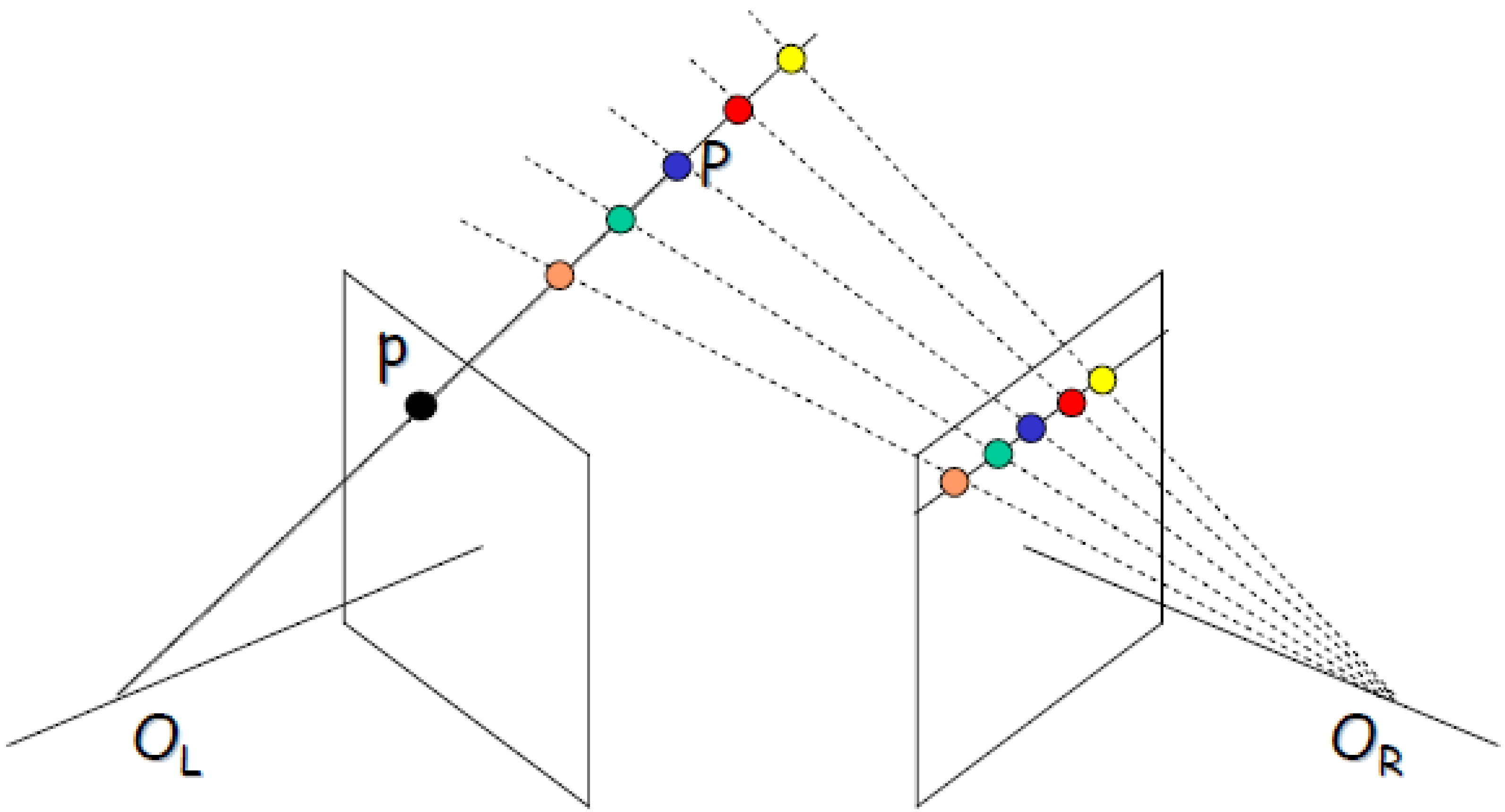
AI6121 Computer Vision

Stereo Vision

Overview



Contents and Learning Objectives

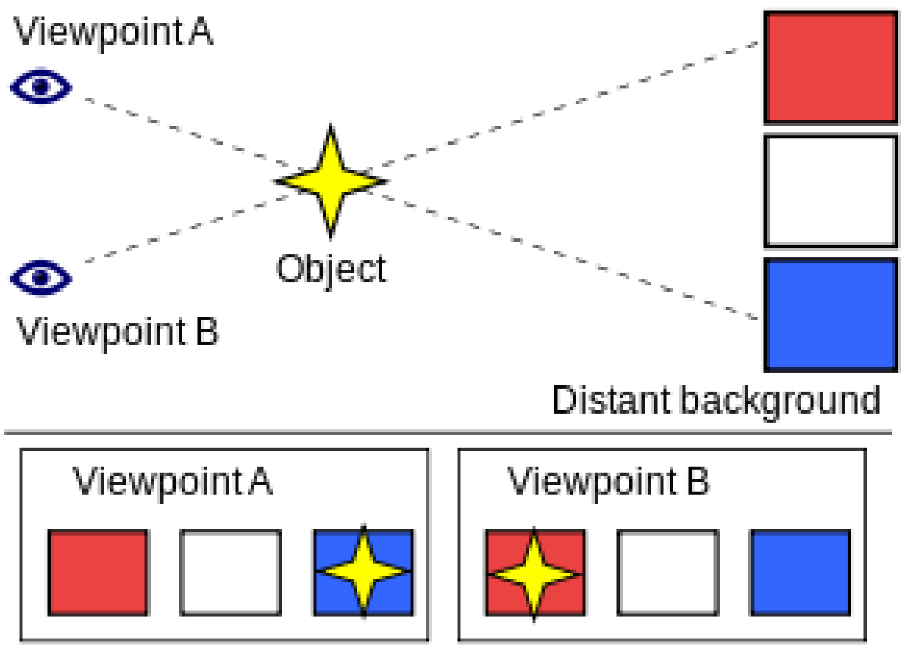
1. Introduction
2. Simple 2D Triangulation
3. Point Matching
4. 3D Reconstruction

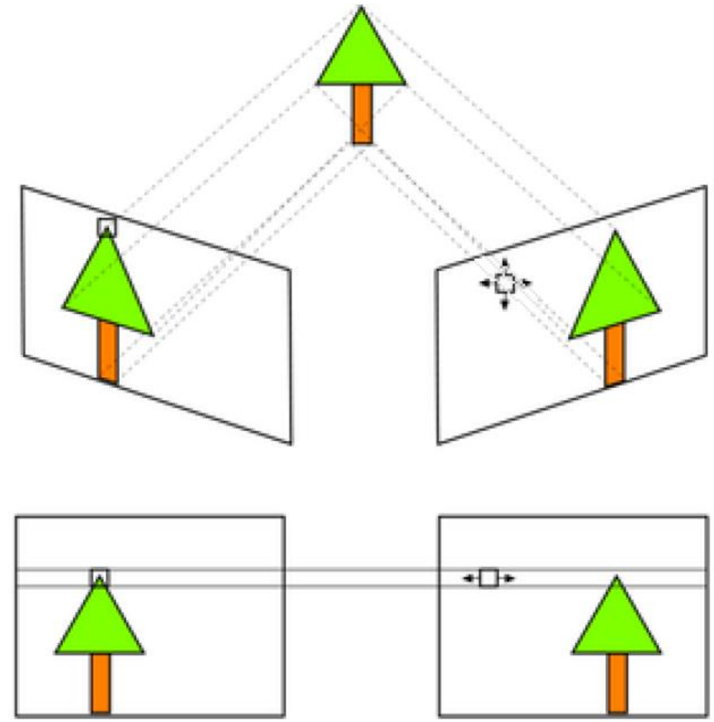
1. Introduction

https://www.format.com/magazine/resources/photography/3d-gif-nishika-n8000

1. Introduction

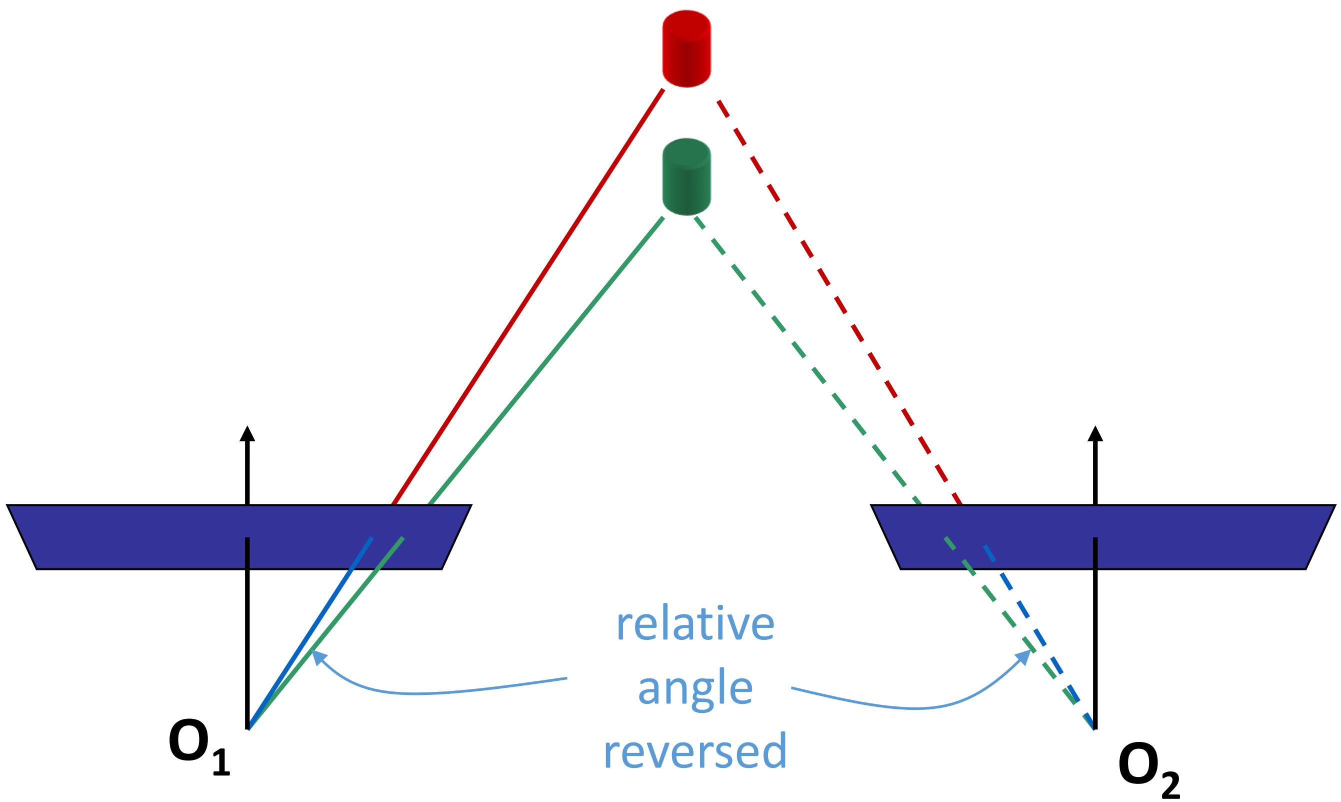
Parallax is a displacement or difference in the apparent position of an object viewed along two different lines of sight, and is measured by the angle or semi-angle of inclination between those two lines.





1. Introduction

Change in relative angular displacement of image points across different camera views when seeing 3D points.



1. Introduction

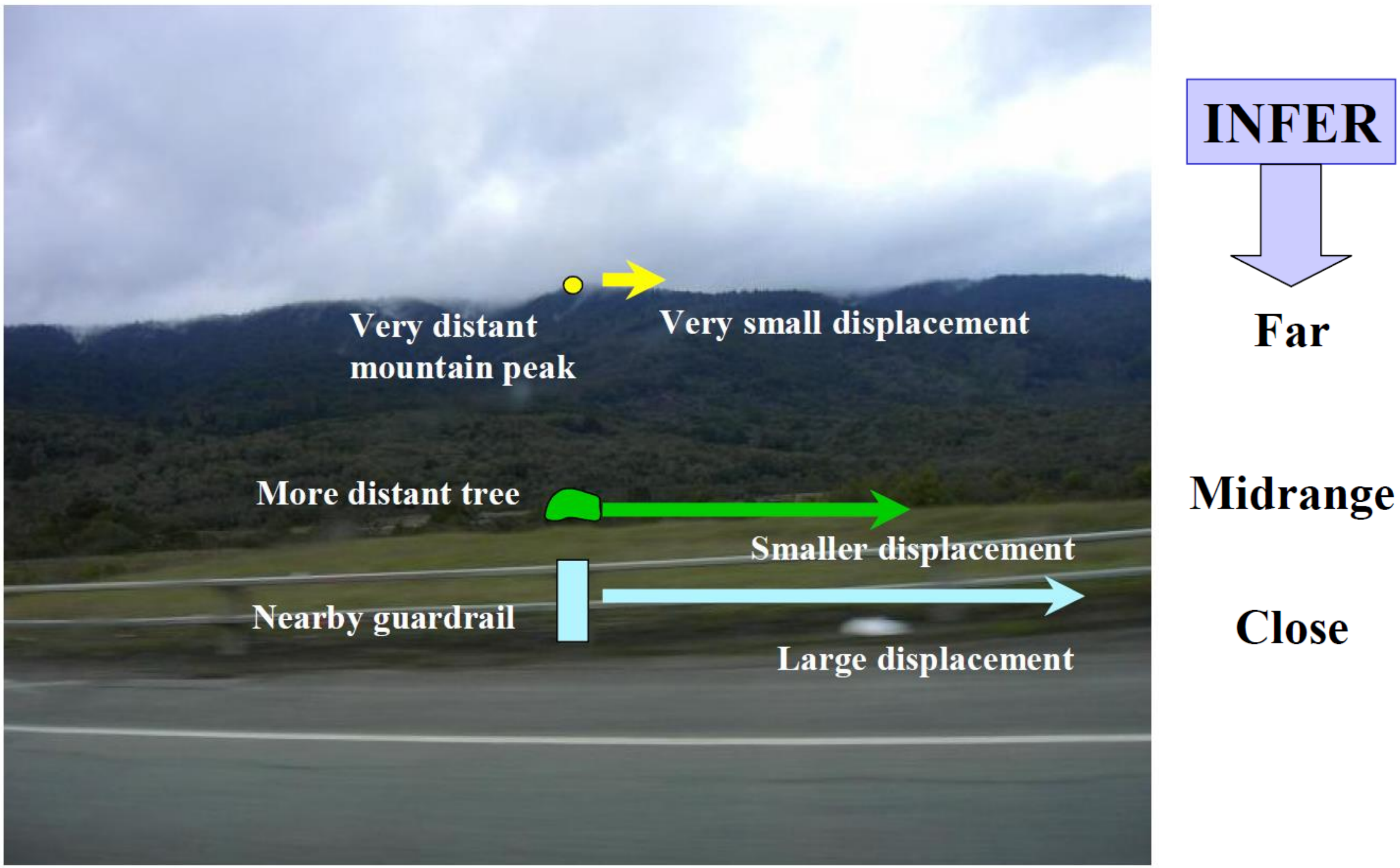
Experiment I:
Open left/right eye (the other closed) and fix it on one finger tip that points to a faraway object. Close the opened eye and open the other eye to see what happens.

Experiment II:
Open one eye (either left or right) and fix it on two finger tips that both point to the same faraway object. Close the opened eye and open the other eye to see what happens.

Can just shift head position without switching eyes.

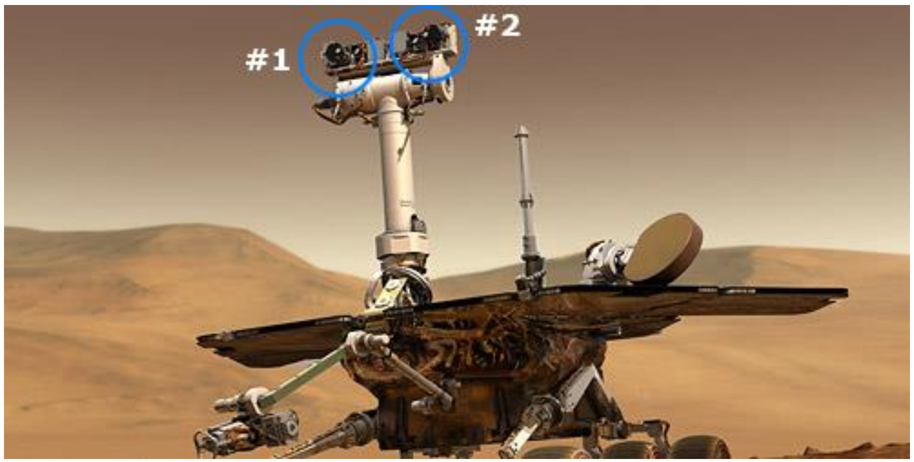
1. Introduction

How do we perceive depth information?



1. Introduction

Mimicking human eye system: the most intuitive way

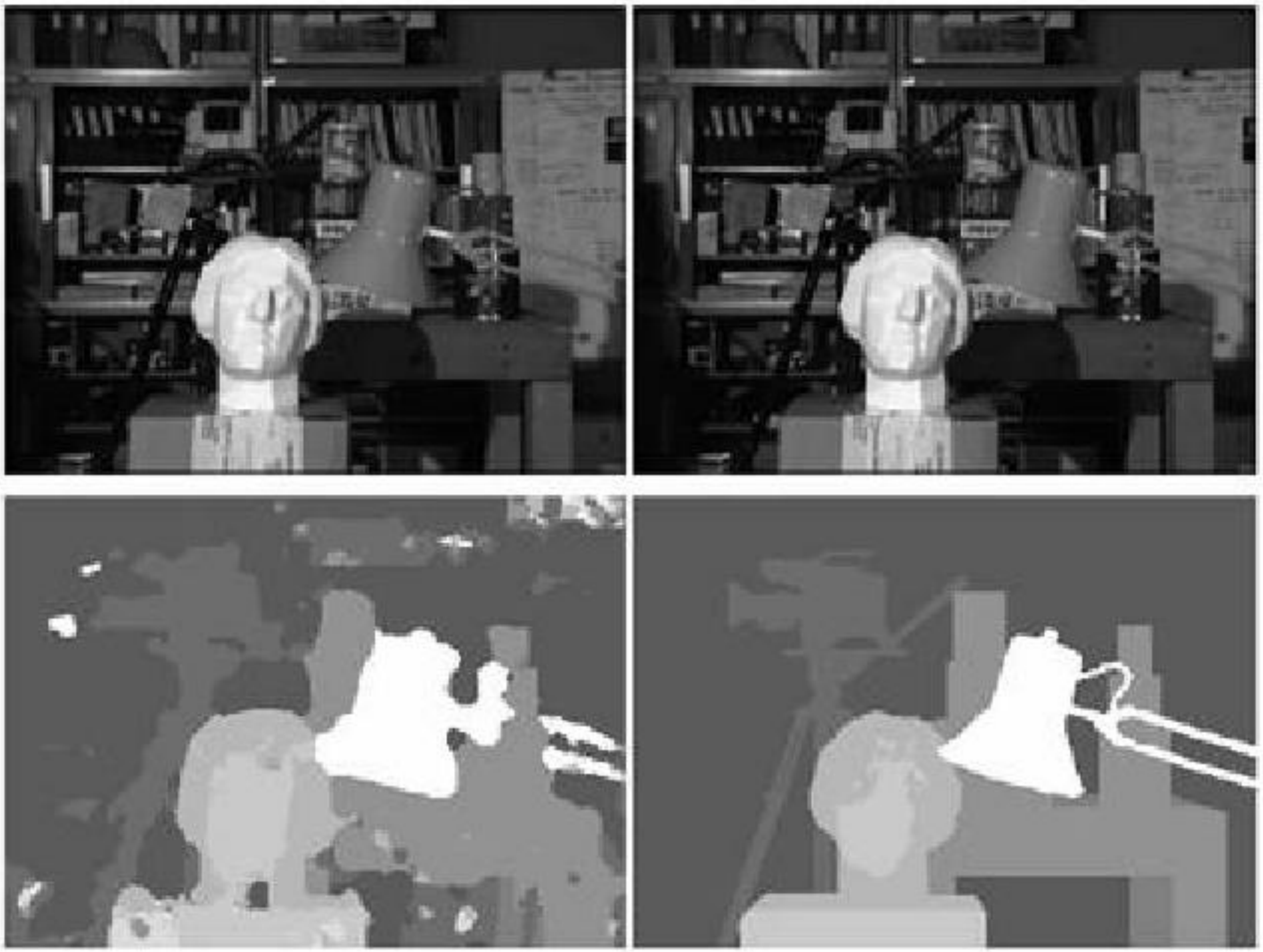


3D Stereo Camera on the Mars Rover - courtesy of NASA

The imaging source

1. Introduction

Inverted Depth Maps: Each pixel is associated with a depth value



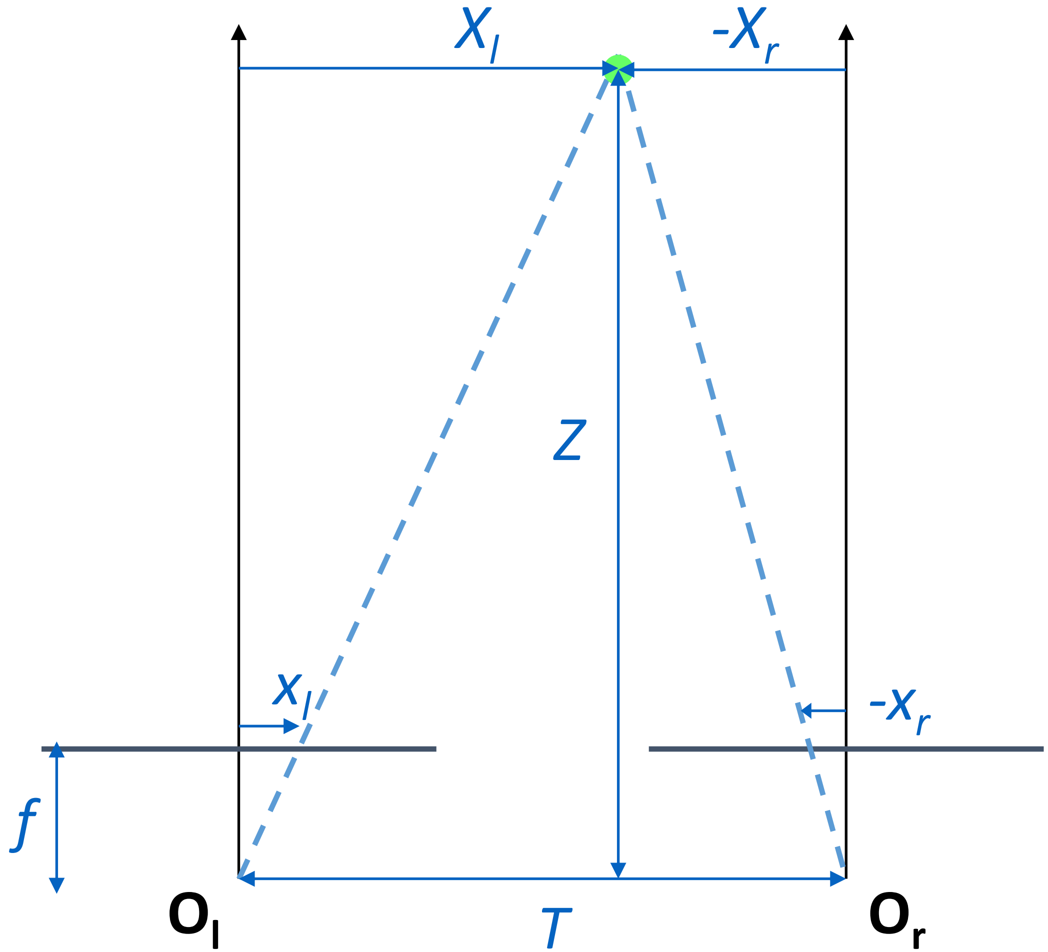
2. Simple 2D Triangulation

Assume a camera translates by T along x-direction in camera frame. With camera perspectives

- $x_l = f\frac{X_l}{Z}, \quad x_r = f\frac{X_r}{Z}$
- $x_l - x_r = f\frac{X_l - X_r}{Z} = \frac{fT}{Z}$

Depth becomes:

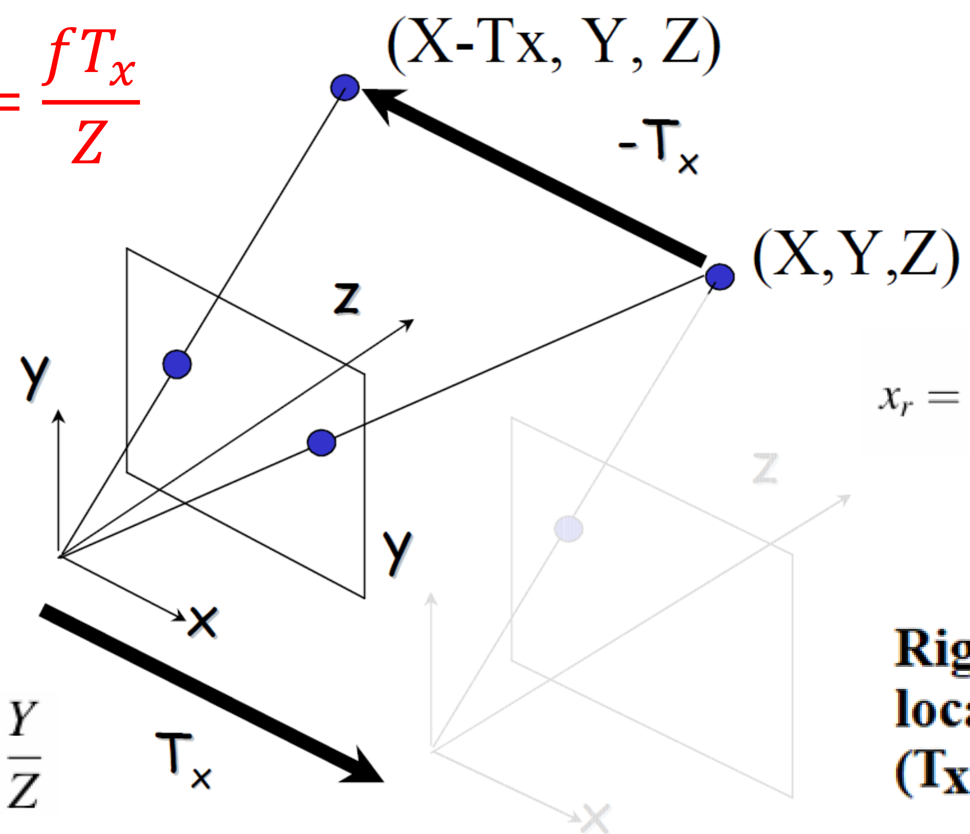
- $Z = \frac{fT}{x_l - x_r} = \frac{fT}{d}$
- Disparity: $d = x_l - x_r = \frac{fT}{Z}$



2. Simple 2D Triangulation

Another way of interpretation - translating camera to the right by T_x is equivalent to leaving the camera stationary and translating the world to the left by T_x.

$$x_l - x_r = d = \frac{fT_x}{Z}$$



$$x_r = f\frac{X - T_x}{Z} \qquad y_r = f\frac{Y}{Z}$$

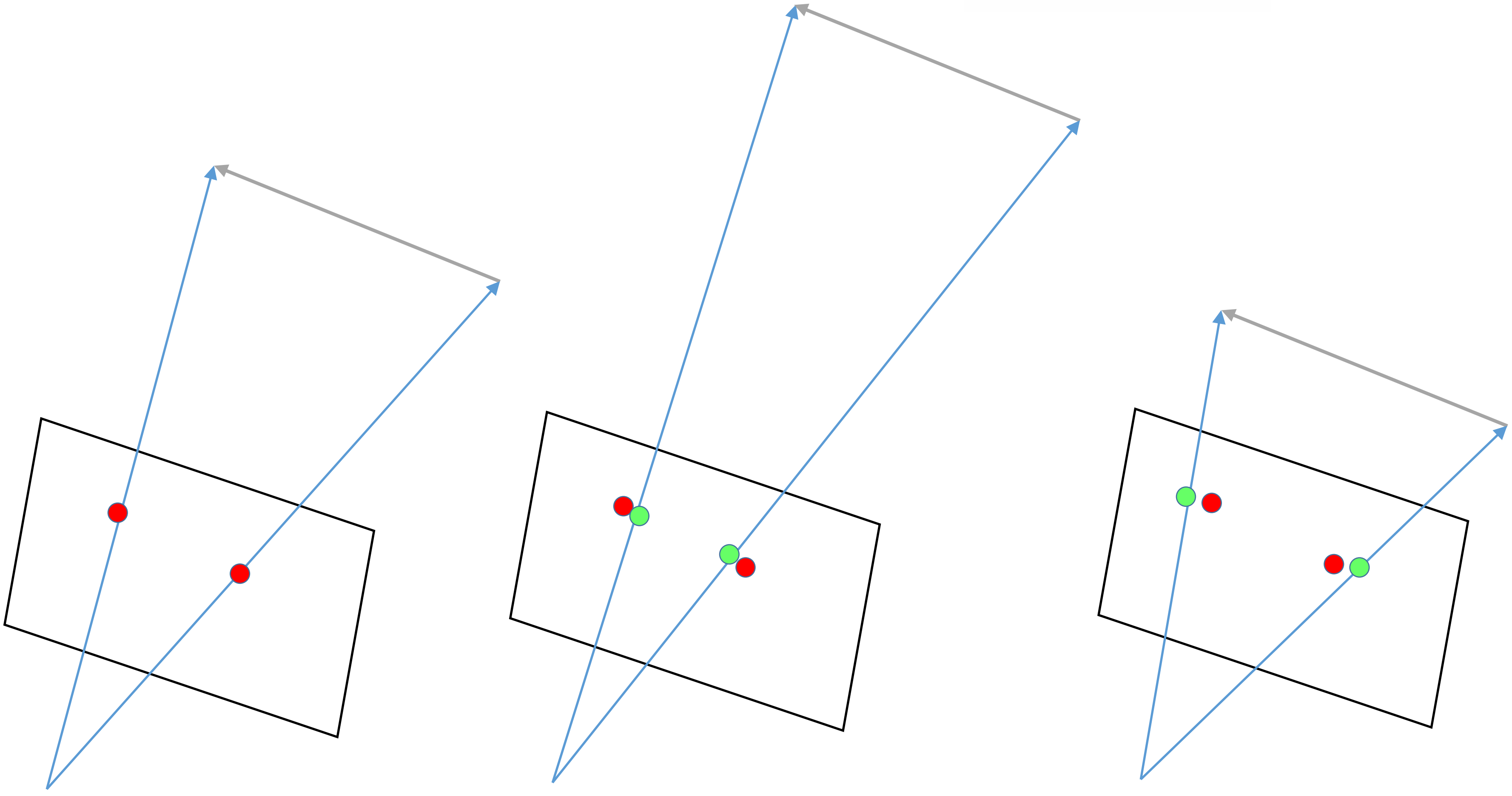
Left camera located at (0, 0, 0)

$$x_l = f\frac{X}{Z} \qquad y_l = f\frac{Y}{Z}$$

Right camera located at (Tx, 0, 0)

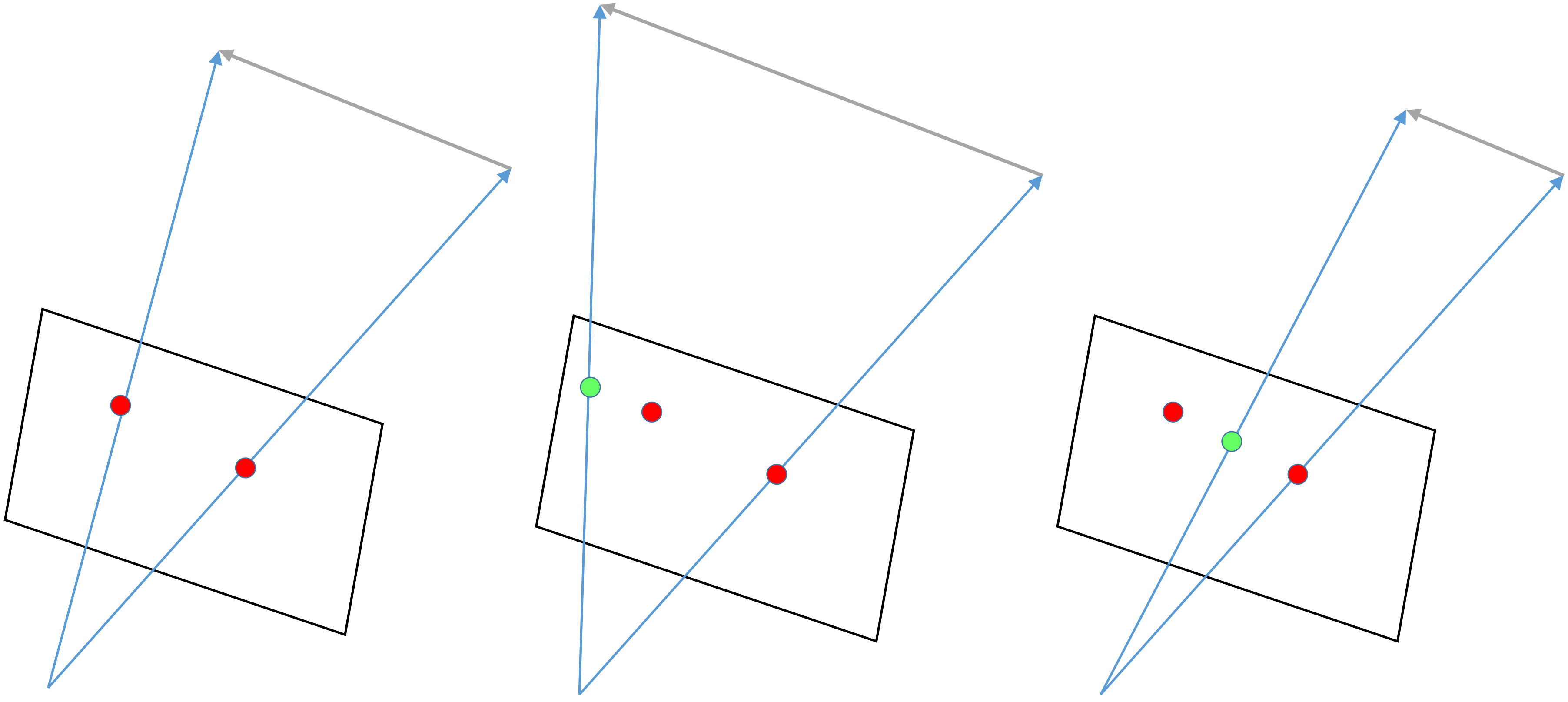
2. Simple 2D Triangulation

What will affect the disparity estimation in $d = \frac{f\,T_x}{Z}$



2. Simple 2D Triangulation

What will affect the disparity estimation in $d = \frac{f\,T_x}{Z}$



2. Simple 2D Triangulation

What will affect the depth estimation?

Let's use w for disparity (to make things clearer as we'll do differentiation)

So $Z = f\dfrac{T}{w}$

Differentiate Z w.r.t. w: $\dfrac{dZ}{dw} = -f\dfrac{T}{w^2}$

Absolute magnitude of error: $|\delta Z| = \left|\dfrac{dZ}{dw}\delta w\right| = \left|f\dfrac{T}{w^2}\delta w\right|$

Substituting for w: $|\delta Z| = \left|f\dfrac{T}{w^2}\delta w\right| = \left|f\dfrac{T}{(fT/Z)^2}\delta w\right| = \left|\dfrac{Z^2}{fT}\delta w\right|$

Go back to original symbol: $|\delta Z| = \left|\dfrac{Z^2}{fT}\delta d\right|$

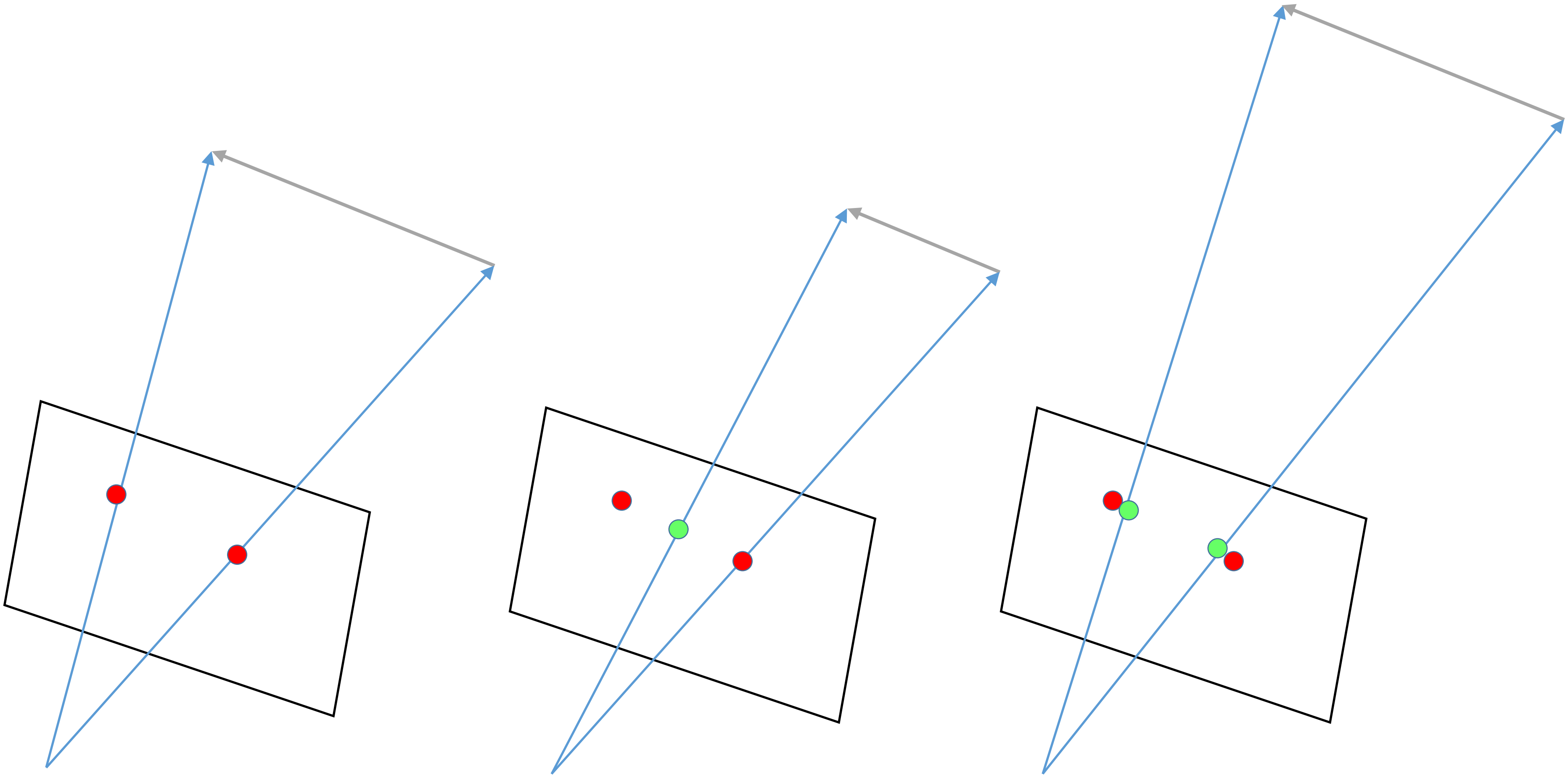
2. Simple 2D Triangulation

$$d = \frac{f\,T_x}{Z}$$

$$|\delta Z| = \left|\frac{Z^2}{fT}\delta d\right|$$

If $T \rightarrow 0$, then $\delta Z \rightarrow \infty$: Two cameras should not be too close

If $Z \rightarrow \infty$, $\delta Z \rightarrow \infty$: Lower accuracy for faraway 3D points



3. Point Matching

- From two points in the left and right images, respectively, we can reconstruct the 3D point
- This means that these two image points are captured from the same 3D point.
- How do we know it?
- This is the correspondence problem: for each important image point in the first image, we need to find the corresponding image point in the second image

3 Point Matching

- Appearance-based Matching

  Match points with similar appearances in two images

- Feature-based Matching

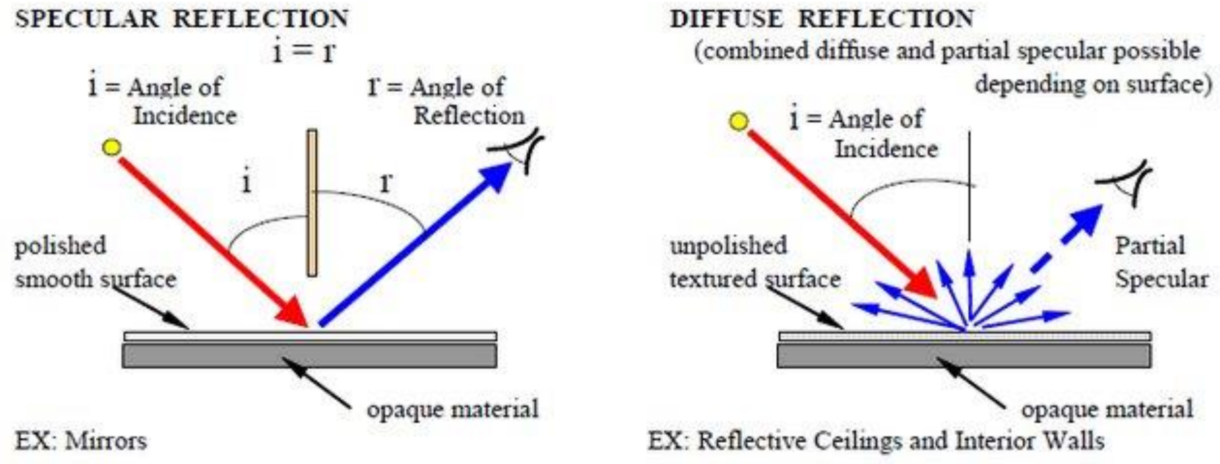
  Match similar features (edges, corners, …) in two images

3 Point Matching: Appearance based

Assumptions: Corresponding points in 2 images have image patches that look almost identical.

- Minimal geometric distortion
- Lambertian reflectance
- No occlusion

These assumptions are reasonable for stereo cameras with small baseline distance.

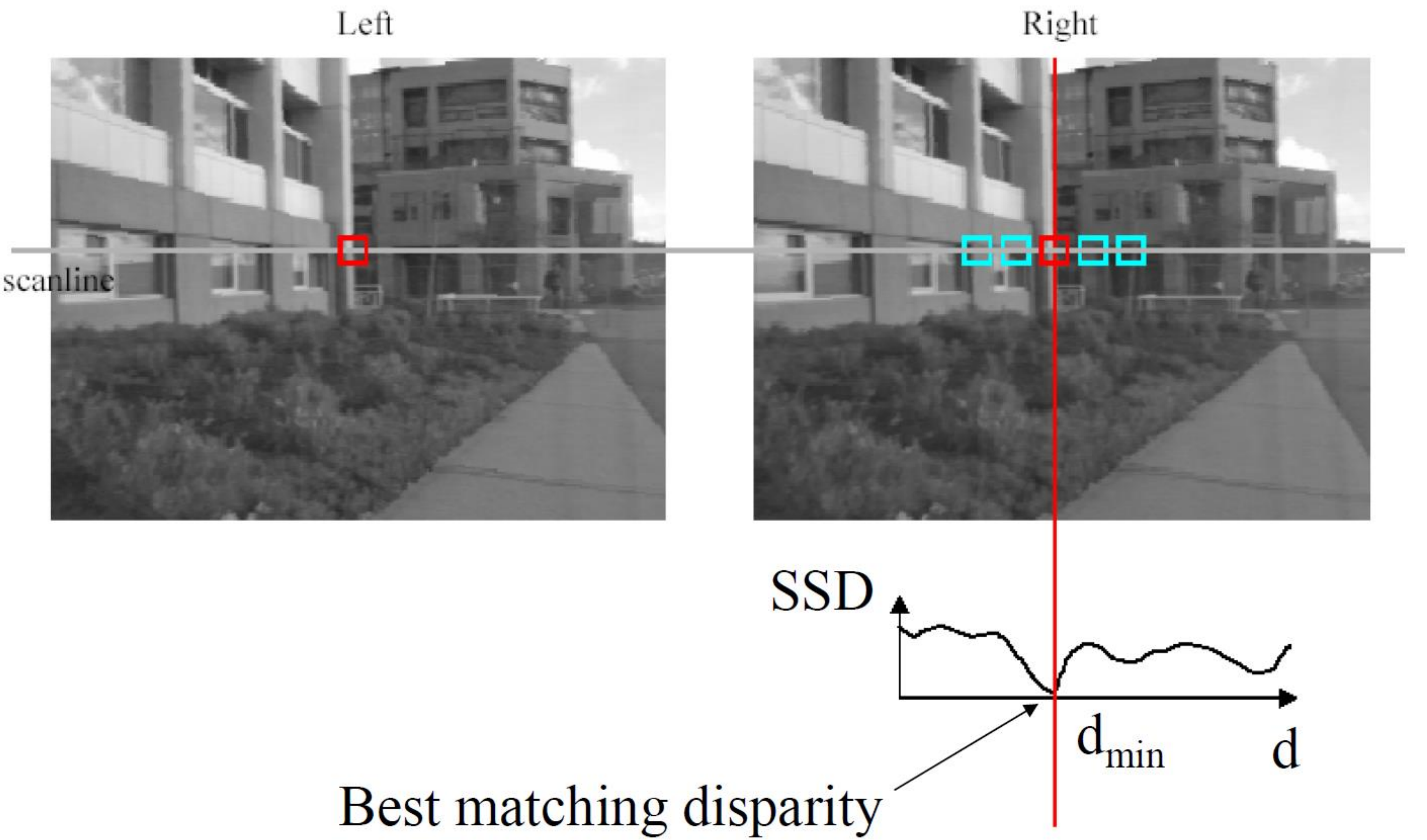


3 Point Matching: Appearance based

For a small image patch ***g*** [(2*N*+1)×(2*N*+1)] in the first image, find the corresponding image patch center location (*x*,*y*) with the smallest Sum-of-Squared Difference (SSD) in the second image ***I*** that

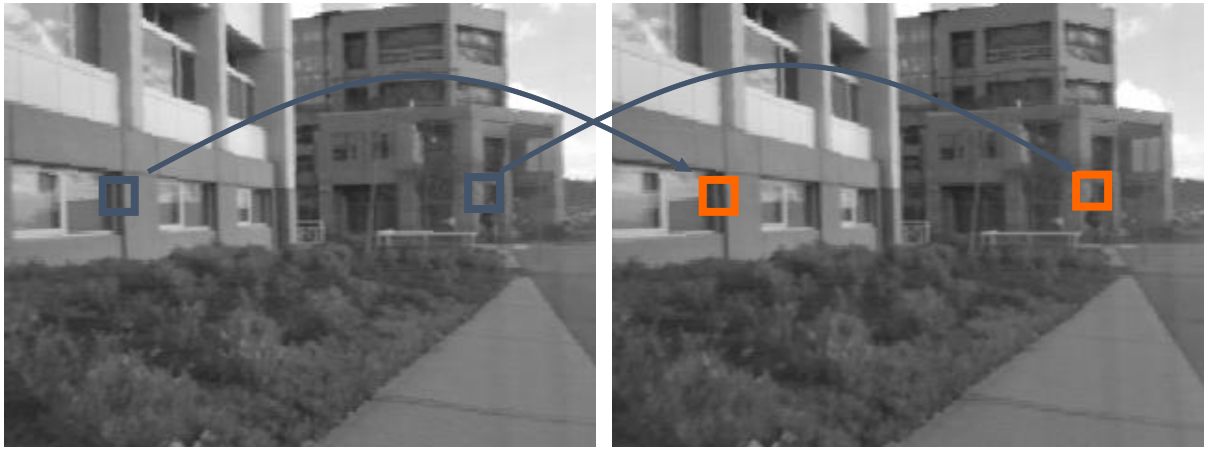
$$\underset{(x,y)}{\operatorname{argmax}} \sum_{u=-N}^{N} \sum_{v=-N}^{N} [I(x+u, y+v) - g(u,v)]^2$$

3 Point Matching: Appearance based



3 Point Matching: Appearance based

Stereo images



Disparity map

3 Point Matching: Appearance based

$$|\delta Z| = \left|\frac{Z^2}{fT}\delta d\right|$$

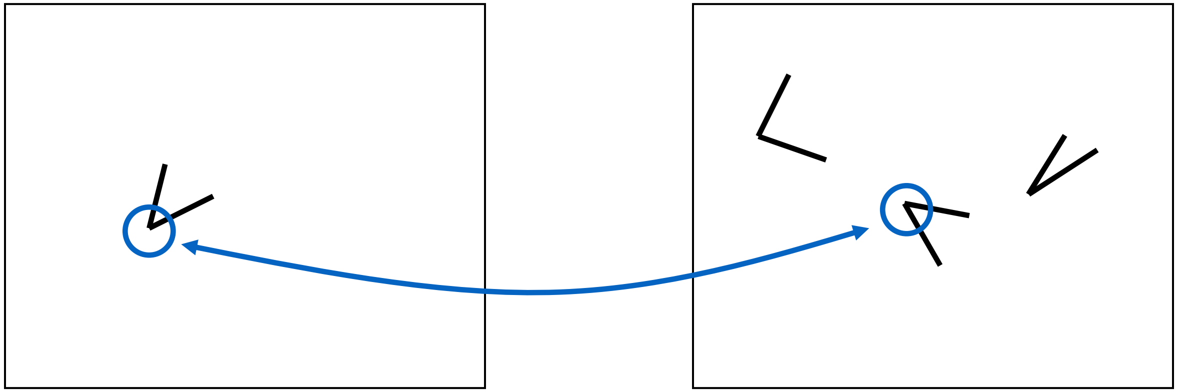
- Correspondence by appearance matching is most accurate when cameras are close – small baseline
- 3D estimation is more accurate when cameras are far apart – large baseline
- In real 3D stereo applications, need reasonable tradeoff.
- SSD based correspondence search is not robust

3 Point Matching: Feature based

- It's for matching image points that have large differences in viewpoints
- To extract image points with local properties that are approximately invariant to larger changes of viewpoint
- Need to select feature points as 3D reconstruction requires sparse point correspondence only.
- Example features and properties:
  - Corners – angle of corner
  - Curves – maximum radius of curvature

3 Point Matching: Feature based

- Match feature points across images directly by finding similar properties
- Example:
  - —Match corner points in two images with the same angle
  - —Corner points can have very different orientation

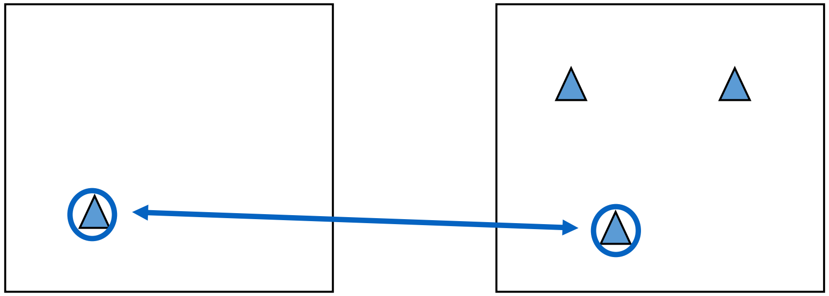


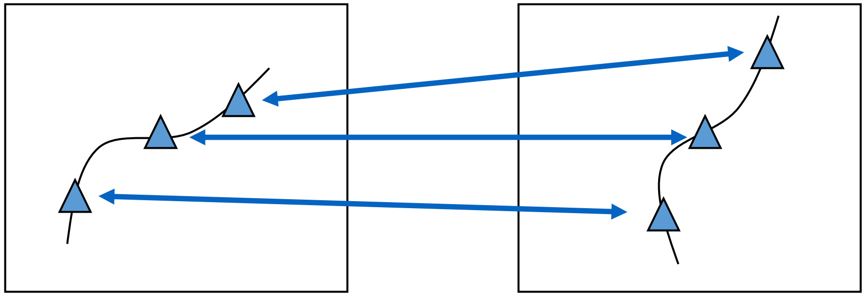
3 Point Matching: Feature based

Even with feature properties, there may be multiple candidates. Heuristics can be used to select the correct correspondence, though they are empirical and could be unreliable.

Proximity – match feature point to candidate with nearest (*x*, *y*) position in second image

Ordering – feature points on a contour in 1st image must match those on a contour in the 2nd image in the same order



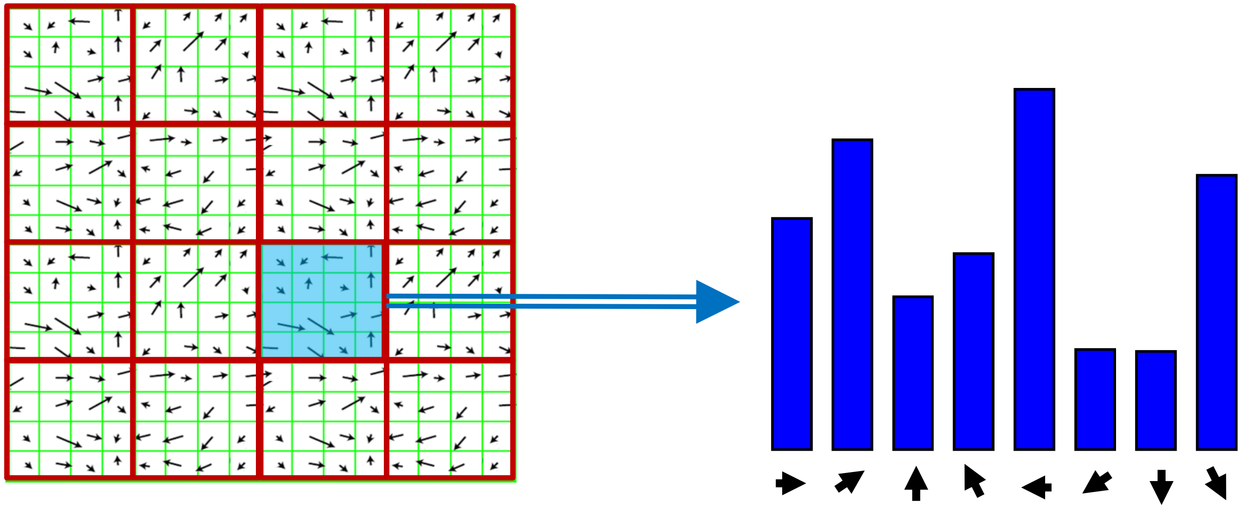


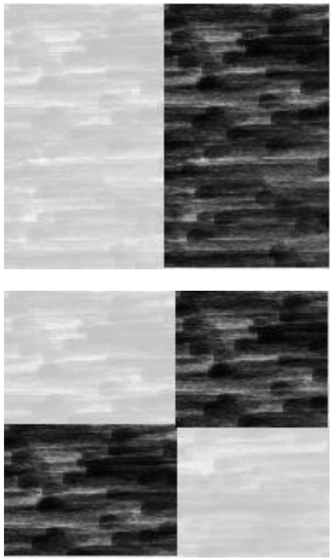
3 Point Matching: Feature based

- Allows greater baseline / larger variation in viewpoints
- Method
  - Extract *feature* points with properties which are invariant to viewpoint changes
  - Match feature points with same properties across images
- May require heuristics to help find correct matching
  - E.g. proximity, ordering

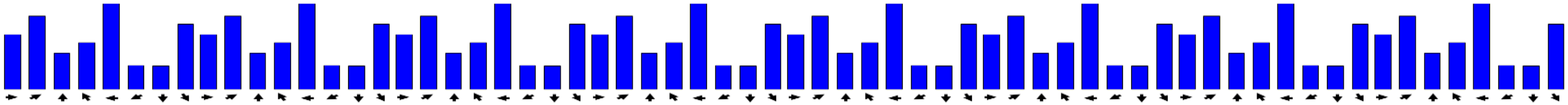
3 Point Matching: Feature based*

Scale invariant feature transform (**SIFT**) has been widely used for robust feature point detection and description.





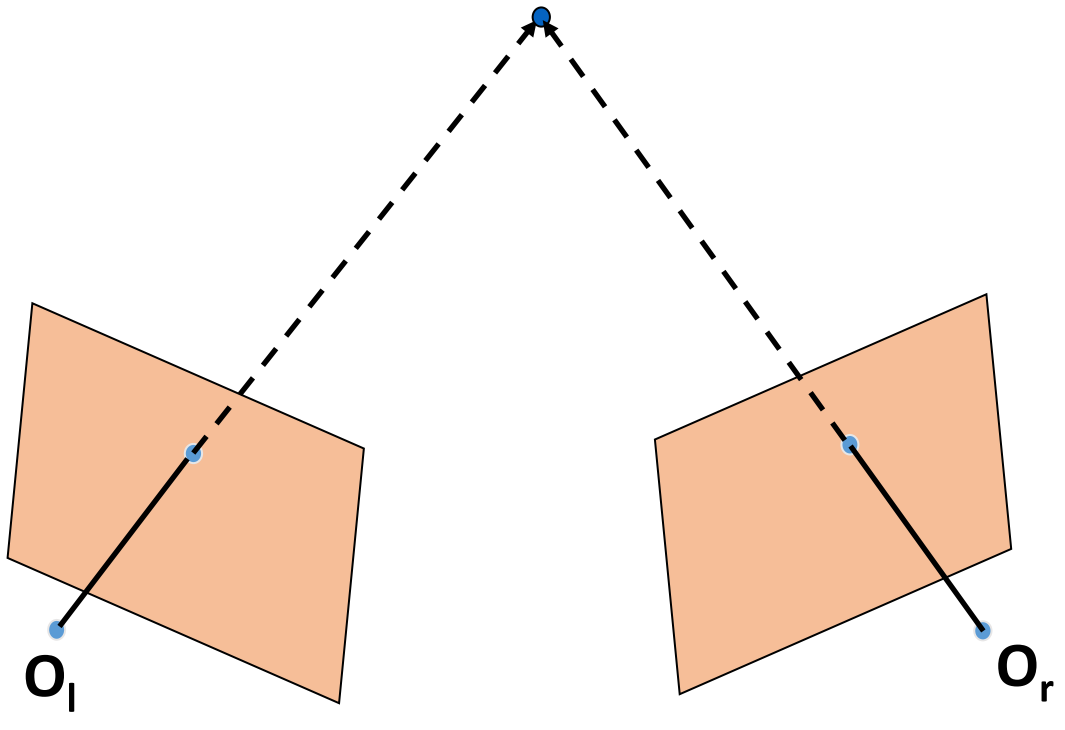
The result: 128 dimensions feature vector.



4. 3D Reconstruction

By triangulation with calibrated cameras

- for each image point pair in the two cameras, we know the associated 3D rays
- Find intersection of rays from corresponding image points



4. 3D Reconstruction

Assume we have projection matrices from camera calibration

L camera:

$$\begin{bmatrix} kx_l \\ ky_l \\ k \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_9 & a_{10} & a_{11} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

R camera:

$$\begin{bmatrix} mx_r \\ my_r \\ m \end{bmatrix} = \begin{bmatrix} b_1 & b_2 & b_3 & b_4 \\ b_5 & b_6 & b_7 & b_8 \\ b_9 & b_{10} & b_{11} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

The unknowns: *X*, *Y*, *Z*

4. 3D Reconstruction

Rewrite for left camera

$$x_l = \frac{a_1 X + a_2 Y + a_3 Z + a_4}{a_9 X + a_{10} Y + a_{11} Z + 1} \qquad y_l = \frac{a_5 X + a_6 Y + a_7 Z + a_8}{a_9 X + a_{10} Y + a_{11} Z + 1}$$

$$\begin{bmatrix} kx_l \\ ky_l \\ k \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & a_3 & a_4 \\ a_5 & a_6 & a_7 & a_8 \\ a_9 & a_{10} & a_{11} & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$

$$kx_l = a_1 X + a_2 Y + a_3 Z + a_4$$

$$kx_r = a_5 X + a_6 Y + a_7 Z + a_8$$

$$k = a_9 X + a_{10} Y + a_{11} Z + 1$$

$$\rightarrow \quad (a_9 x_l - a_1) X + (a_{10} x_l - a_2) Y + (a_{11} x_l - a_3) Z = (a_4 - x_l)$$

$$(a_9 y_l - a_5) X + (a_{10} y_l - a_6) Y + (a_{11} y_l - a_7) Z = (a_8 - y_l)$$

Similarly for right camera

$$(b_9 x_r - b_1) X + (b_{10} x_r - b_2) Y + (b_{11} x_r - b_3) Z = b_4 - x_r$$

$$(b_9 y_r - b_5) X + (b_{10} y_r - b_6) Y + (b_{11} y_r - b_7) Z = b_8 - y_r$$

4. 3D Reconstruction

Recall what we learned in camera calibration

$$\begin{bmatrix} kx_{im} \\ ky_{im} \\ k \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & 1 \end{bmatrix} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$\begin{cases} kx_{im} = m_{11}X_w + m_{12}Y_w + m_{13}Z_w + m_{14} \\ ky_{im} = m_{21}X_w + m_{22}Y_w + m_{23}Z_w + m_{24} \\ k = m_{31}X_w + m_{32}Y_w + m_{33}Z_w + 1 \end{cases}$$

The above equations are rearranged as follow

$$\begin{cases} X_w m_{11} + Y_w m_{12} + Z_w m_{13} + m_{14} - x_{im} X_w m_{31} - x_{im} Y_w m_{32} - x_{im} Z_w m_{33} = x_{im} \\ X_w m_{21} + Y_w m_{22} + Z_w m_{23} + m_{24} - y_{im} X_w m_{31} - y_{im} Y_w m_{32} - y_{im} Z_w m_{33} = y_{im} \end{cases}$$

These are two equations but 11 unknowns m_{ij}

4. 3D Reconstruction

- Write in the matrix form

$$\underbrace{\begin{bmatrix} a_9 x_l - a_1 & a_{10} x_l - a_2 & a_{11} x_l - a_3 \\ a_9 y_l - a_5 & a_{10} y_l - a_6 & a_{11} y_l - a_7 \\ b_9 x_r - b_1 & b_{10} x_r - b_2 & b_{11} x_r - b_3 \\ b_9 y_r - b_5 & b_{10} y_r - b_6 & b_{11} y_r - b_7 \end{bmatrix}}_{\mathbf{W}} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \underbrace{\begin{bmatrix} a_4 - x_l \\ a_8 - y_l \\ b_4 - x_r \\ b_8 - y_r \end{bmatrix}}_{\mathbf{q}}$$

- Compute 3D world coordinates *X*, *Y* and *Z* via pseudo-inverse

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = \mathbf{W}^{+}\mathbf{q} = (\mathbf{W}^{\mathrm{T}}\mathbf{W})^{-1}\mathbf{W}^{\mathrm{T}}\mathbf{q}$$

- Computationally expensive – compute new pseudo-inverse for each pair of image correspondences

4. 3D Reconstruction

1. Fix the pose of your cameras, e.g. by placing them on tripods
2. Using a calibration chart,
    - Calibrate your cameras using known correspondences of 3D points to 2D image points
3. For an arbitrary scene in front of your cameras, your system should automatically:
    - Find corresponding image points between camera images
    - Compute 3D coordinates for each correspondence

5. Summary

- 3D stereo vision and parallax
- Triangulation
- Point Matching
  - Appearance-based matching
  - Feature-based matching
- 3D reconstruction