

Rectification and Disparity

Nassir Navab

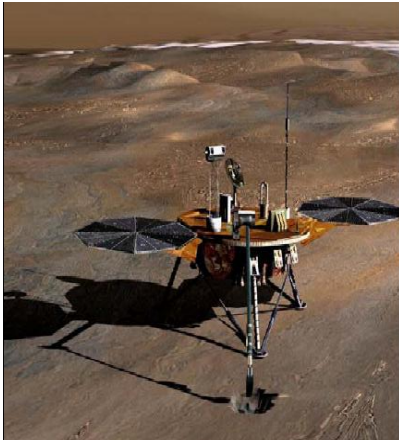
Slides prepared by Christian Unger

What is Stereo Vision?

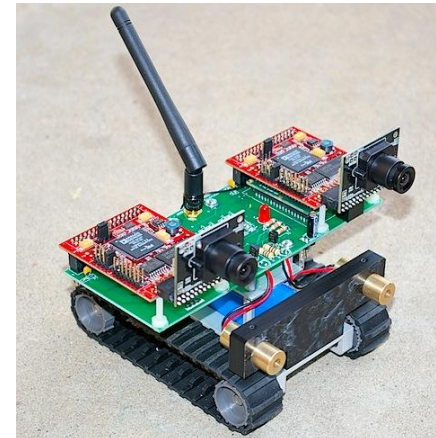
Introduction

- A technique aimed at inferring **dense** depth measurements efficiently using two cameras.
- Wide and old research area in computer vision.
- Contents of the talks:
 - Radial Undistortion: Compensate effects of radial lens distortion.
 - Rectification: Transforming the epipolar geometry into a canonical form.
 - Disparity: Definition and relation to depth.
 - Binocular Stereo Matching: The process of establishing dense correspondences between two images.
 - Triangulation: Computing a 3D reconstruction.

Stereo-Rigs and Applications.



NASA
Mars
Exploration

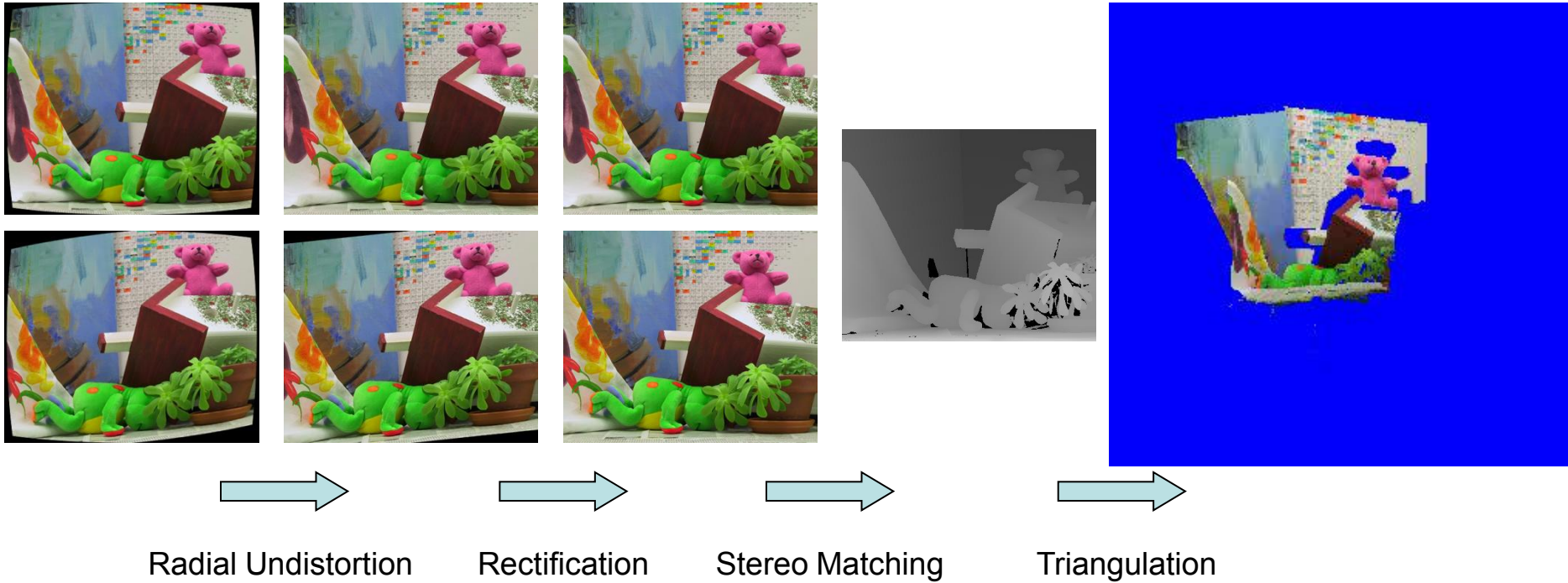


Stereo Vision Systems: Outline.

A Stereo Vision System is usually composed of these processing steps:

1. Radial Undistortion
2. Rectification
3. Stereo Matching
4. Triangulation

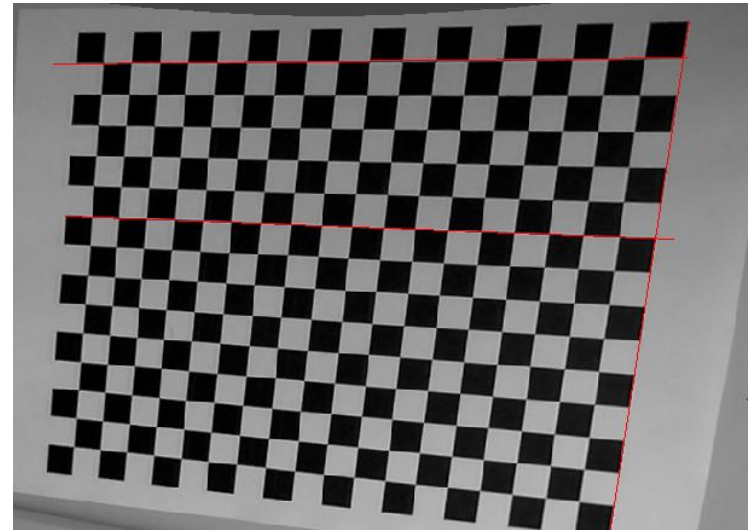
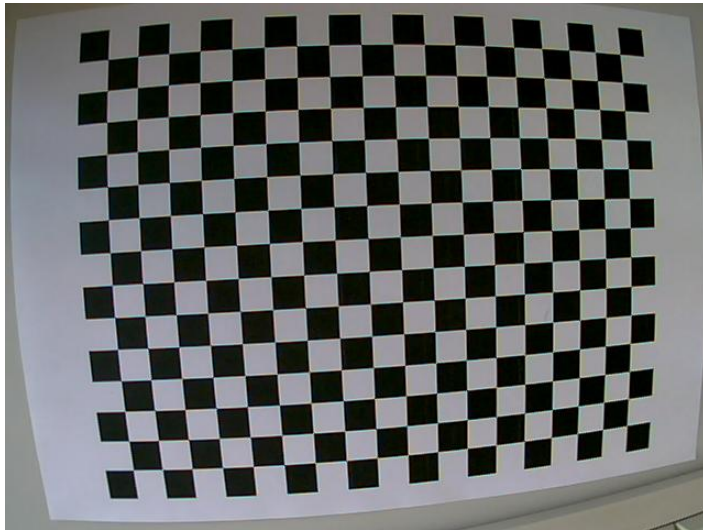
Stereo Vision Systems: Outline.



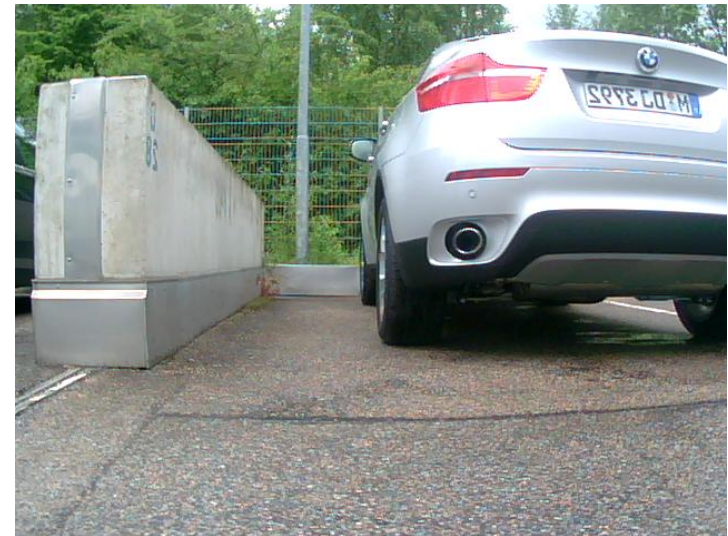
Stereo Vision Systems: Radial Undistortion.

Compensate for the effects of radial lens distortion.

Use calibration patterns to make straight lines straight.



Radial Lens Distortion: Examples.



Radial Lens Distortion: Mathematical Background.

We use a simplified version of Brown's model (barrel distortion) to formulate the undistortion:

- The raw-camera-image contains distorted image points:

$$\mathbf{x}_D = (x_D, y_D)$$

- We seek the undistorted image points

$$\mathbf{x}_U = (x_U, y_U)$$

- Brown's model is given as:

$$\mathbf{x}_U = \mathbf{x}_D + L(r) \cdot (\mathbf{x}_D - \mathbf{x}_C)$$

- With

$$L(r) = K_1 r^2 + K_2 r^4 + \dots$$

$$r = \sqrt{(x_D - x_C)^2 + (y_D - y_C)^2}$$

Radial Lens Undistortion: Approach.

Offline phase:

- Estimate the distortion parameters using 2D-3D correspondences (e.g. using a checker-board).

Online phase:

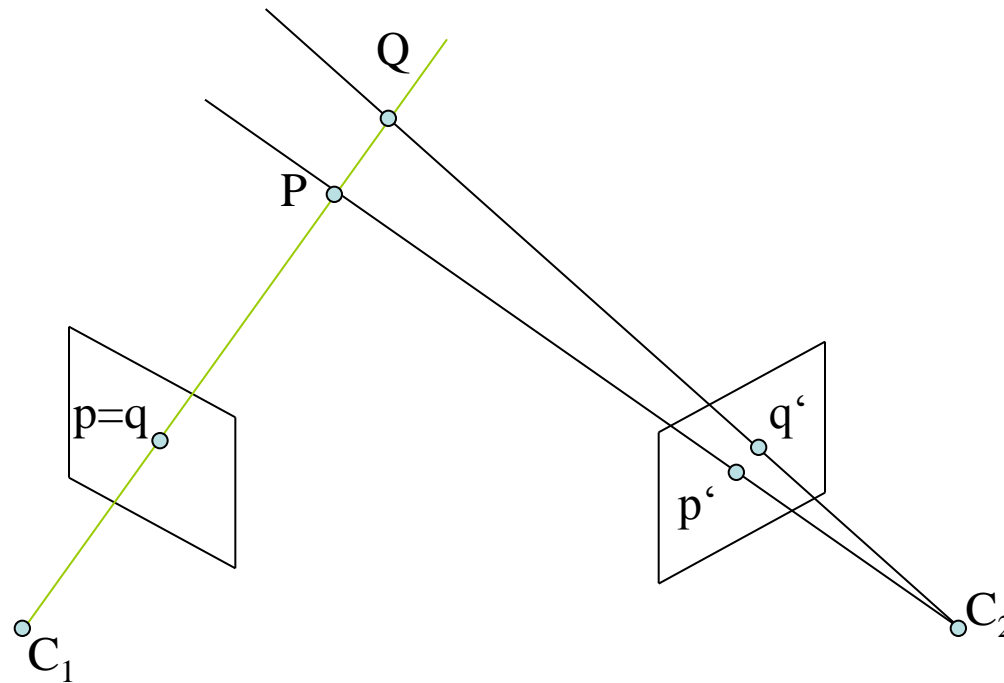
- Warp images using a reverse distortion (“backward warping”: for every undistorted pixel we have to compute the distorted location).
- Due to the non-linearity of the distortion-function, this must be done numerically.
- In practice, usually a lookup-table is computed offline.

Stereo Vision Systems: Rectification.

Align the image planes (move the epipoles to infinity and match up epipolar lines).

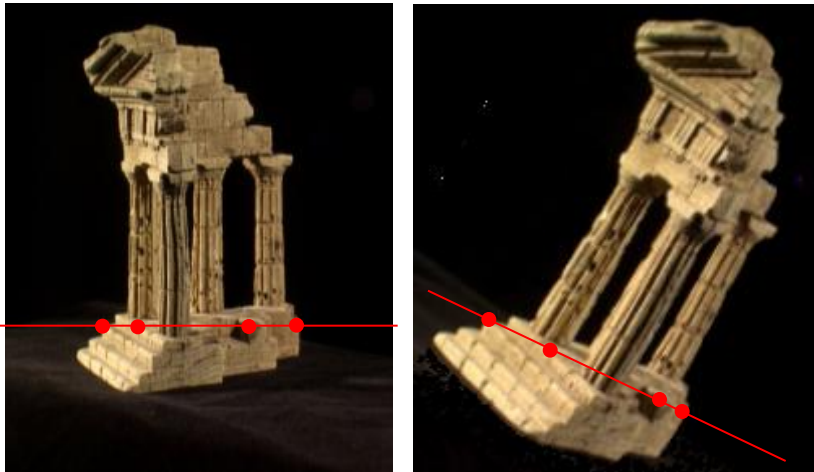


Reminder: Depth from Corresponding Points.



With two (or more) cameras we can infer depth from correspondences using **triangulation**.

Reminder: Epipolar Geometry.



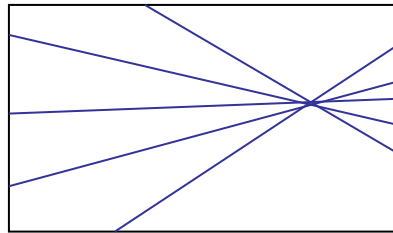
Since correspondences are constrained by epipolar geometry, only a one-dimensional search domain has to be considered:

$$\mathbf{l}' = \mathbf{F}\mathbf{x}$$

Rectification: Stereo in Canonical Form.

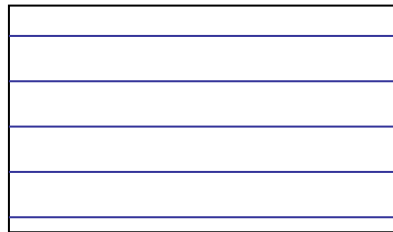
In general: epipolar lines may be slanted.

=> Computation of the epipolar line for every image point is costly!

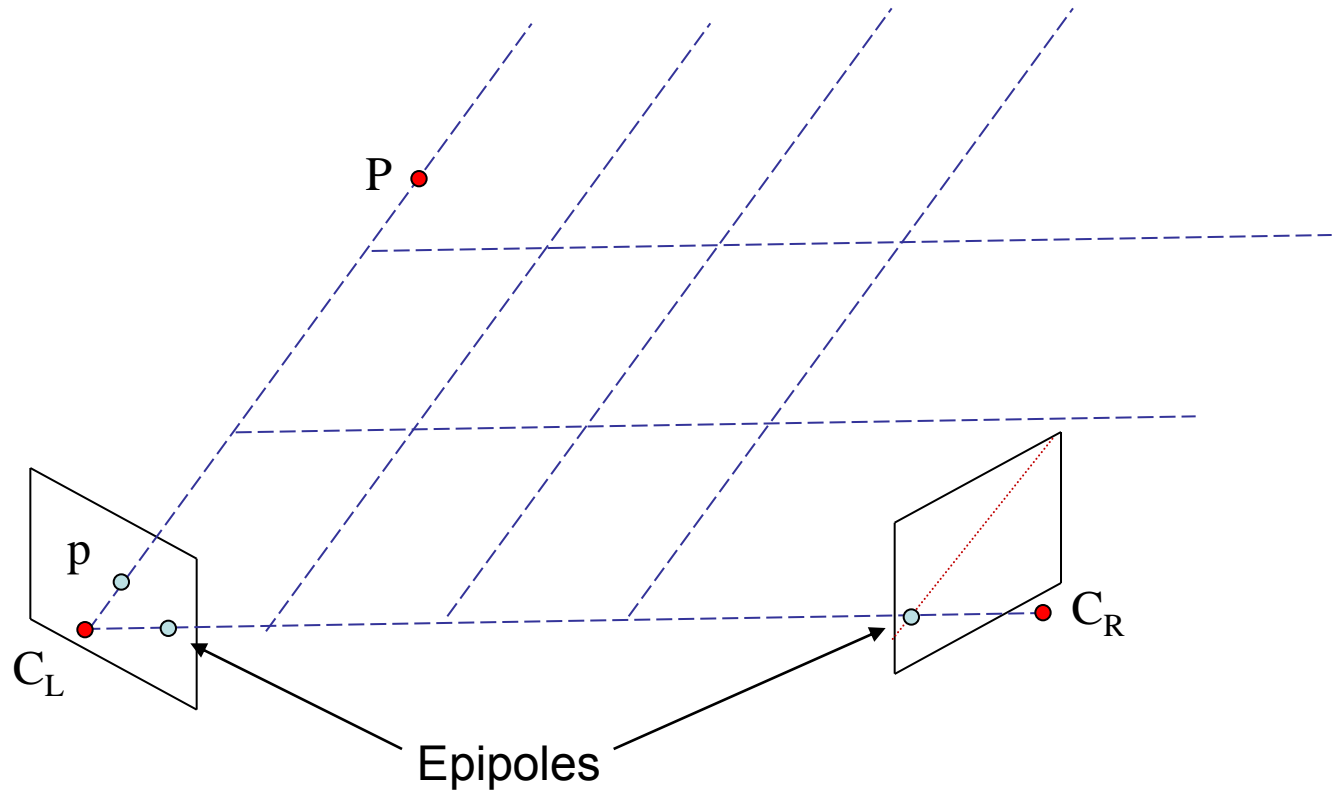


It is more desirable to have parallel, horizontal epipolar lines.

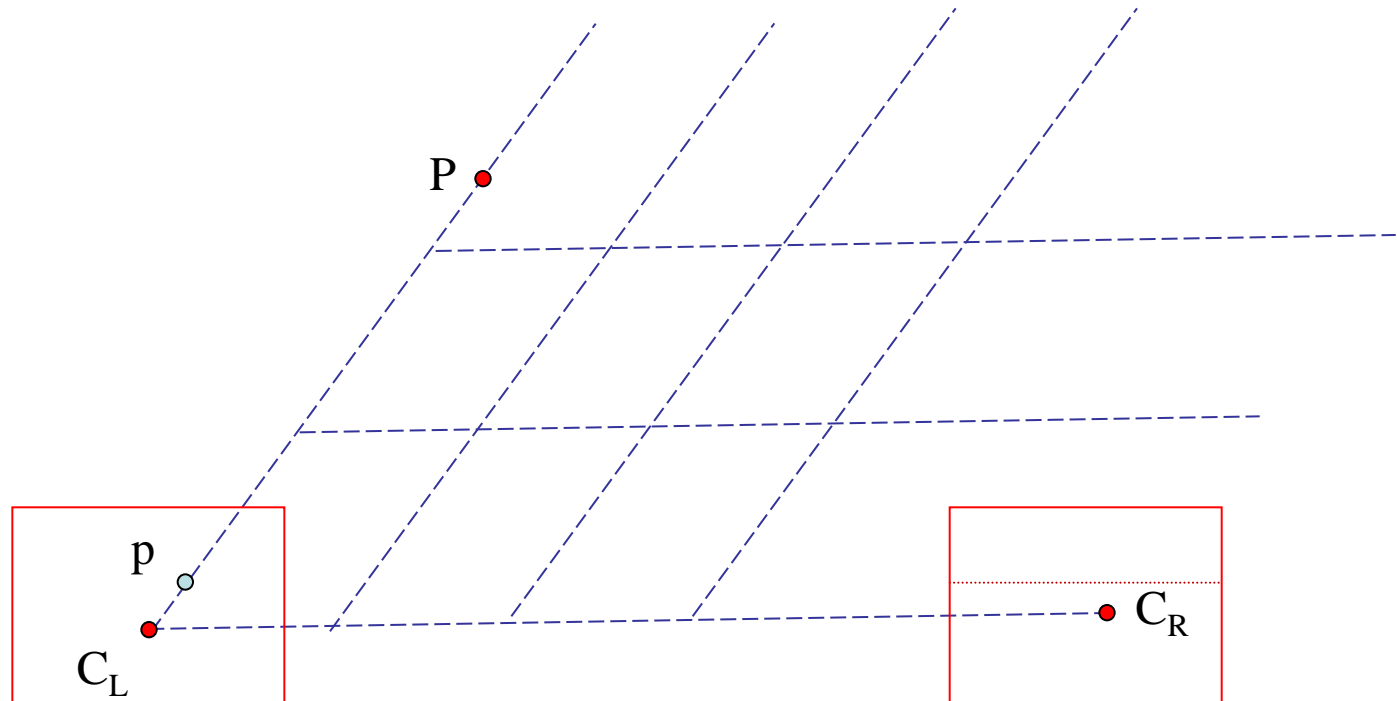
=> Simple search along scanlines!



Rectification: Stereo in Canonical Form.

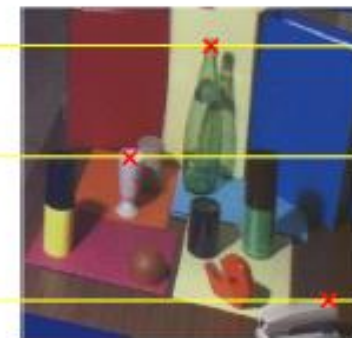
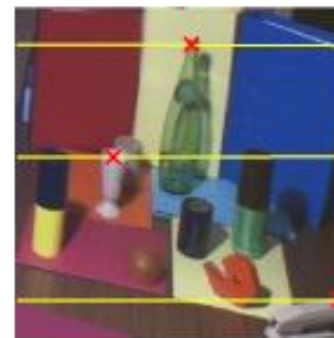
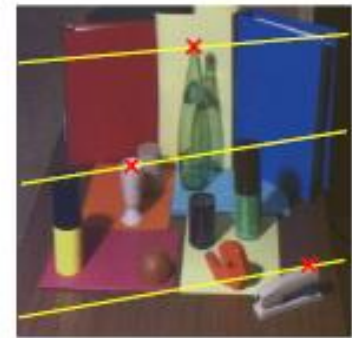
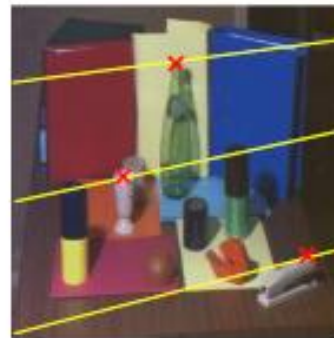


Rectification: Stereo in Canonical Form.



Rotate the cameras virtually, so that the two image-planes become co-planar.

Rectification: Stereo in Canonical Form.



Rectification: An Interpretation.

In General:

Rectification = Put the epipoles to a predefined position
= Align the image planes (rotation, focal length)

Standard approach:

- Make epipolar lines run parallel to the x-axis. For that,
- move the epipole to infinity and
- match up epipolar lines between views.

By this convention, points will only move in the x-direction – there will be no movement in y-direction!

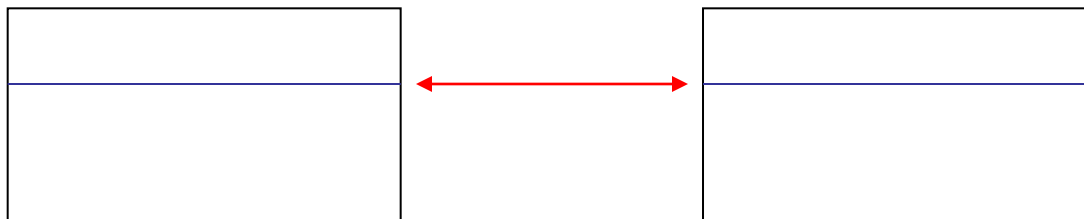
Rectification: Minimize Image Distortion.

Let H' be a projective transformation that sends the epipole e' to infinity.

Then, the transformation H for the second camera might be chosen so as to minimize the sum-of-squared distances:

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

Furthermore, H might be constrained to **match up** corresponding epipolar lines:



Rectification: Algorithm Outline.

1. Find initial correspondences
2. Compute the fundamental matrix
3. Compute a projective transformation H' that maps the epipole e' to infinity $(1, 0, 0)^T$
4. Find the matching projective transformation H that minimizes

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

5. Warp the first image according to H and the second image according to H'

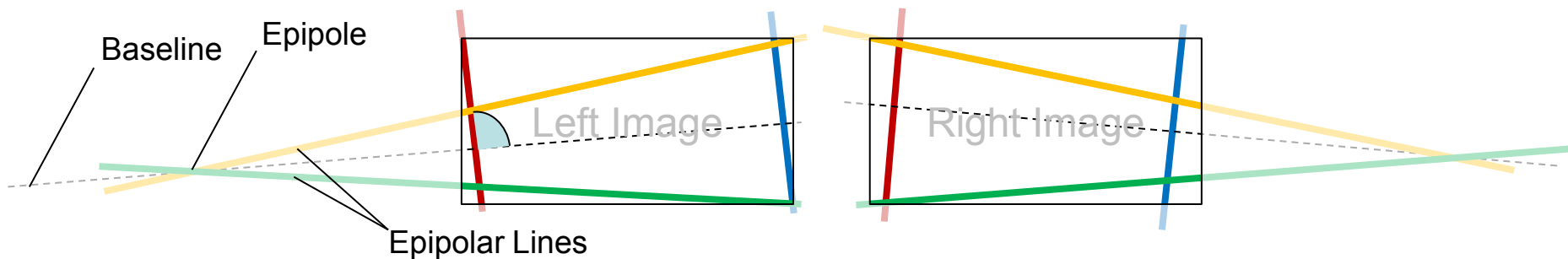
Rectification: Another Interpretation.

In practice the presented algorithm (Hartley-Zisserman) works, but may lead to undesirable results (e.g. „black borders“).

Argumentation via the DOF of a homography:

- Moving the epipole to infinity constrains only 2 DOF.
- Some clarifications and repetitions:
 - Epipolar lines intersect at the epipole.
 - The image of the baseline in rectified images is parallel to the x-axis.
- Now, we use four line-to-line correspondences:
 - We use 2 epipolar lines to constrain top and bottom image borders
 - We use another 2 lines being orthogonal to the baseline to constrain left and right image borders
 - These four lines define a rectangle => ensure that the rectangle is completely within the FOV.

Rectification: Another Interpretation.



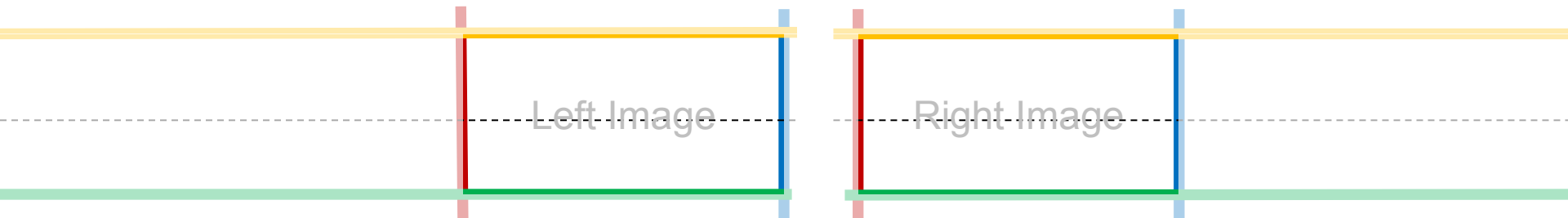
=>

- Make **top** and **bottom** lines horizontal, to move the epipole to infinity.
- Map **left** and **right** lines to the left and right image border to avoid “black regions”).

Given lines in the left image, how to obtain the lines in the right image:

- Use corresponding epipolar lines (for **top** and **bottom**).
- Use points at infinity to ensure a proper disparity offset and scale for (**left** and **right**).

Rectification: Another Interpretation.



=>

- Make **top** and **bottom** lines horizontal, to move the epipole to infinity.
- Map **left** and **right** lines to the left and right image border to avoid “black regions”).

Given lines in the left image, how to obtain the lines in the right image:

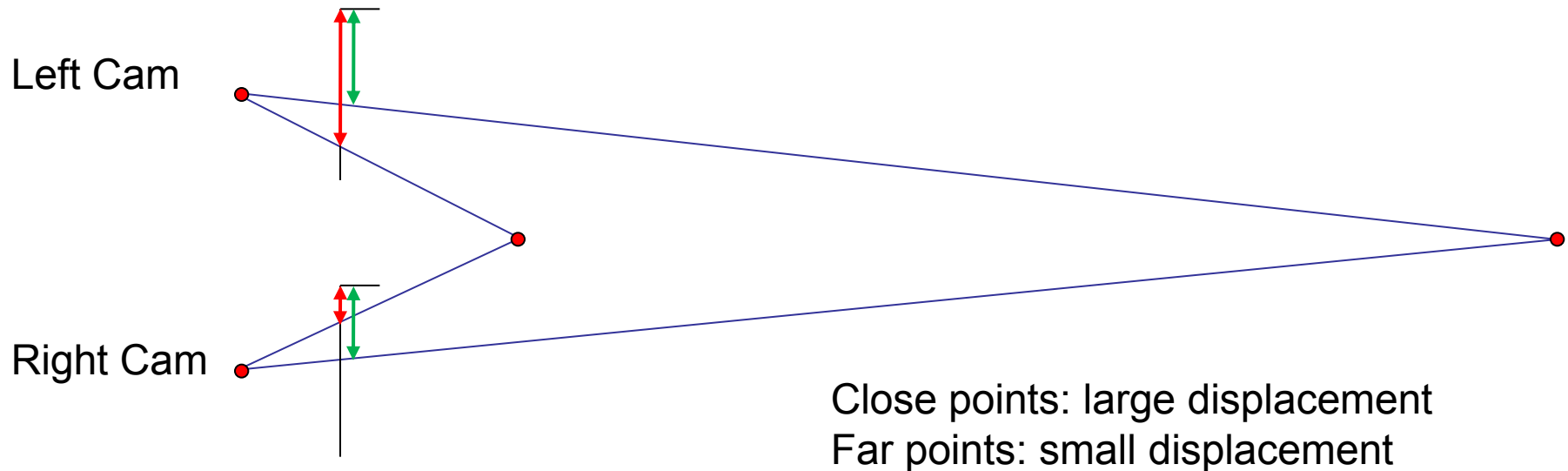
- Use corresponding epipolar lines (for **top** and **bottom**).
- Use points at infinity to ensure a proper disparity offset and scale for (**left** and **right**).

Disparity: Definition.

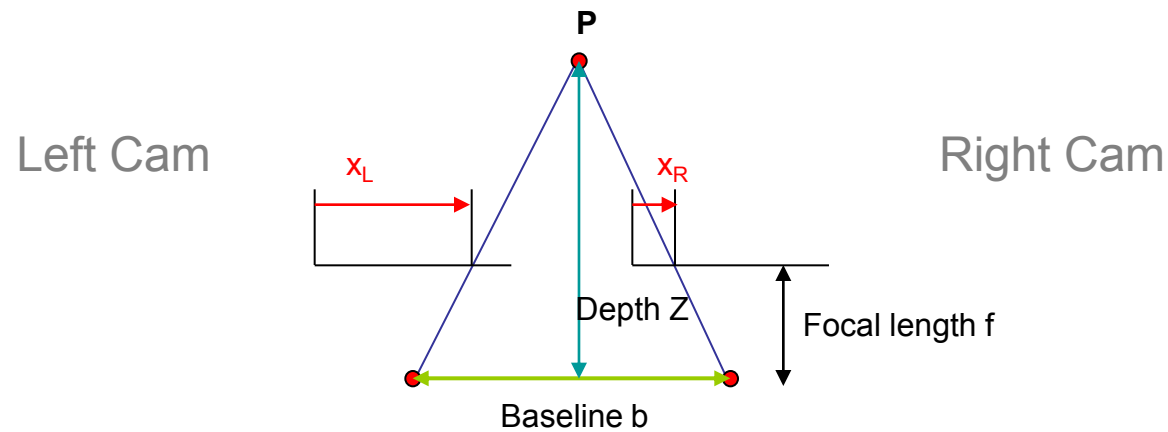
Assume a calibrated, rectified stereo setup.

Definition (Disparity)

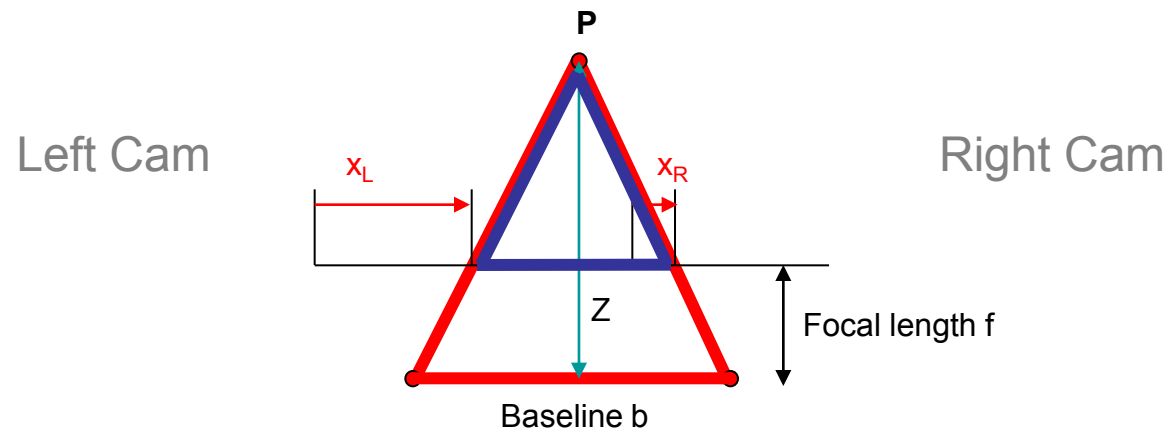
The disparity measures the displacement of a point between the two images.



Disparity: Relationship to Depth.

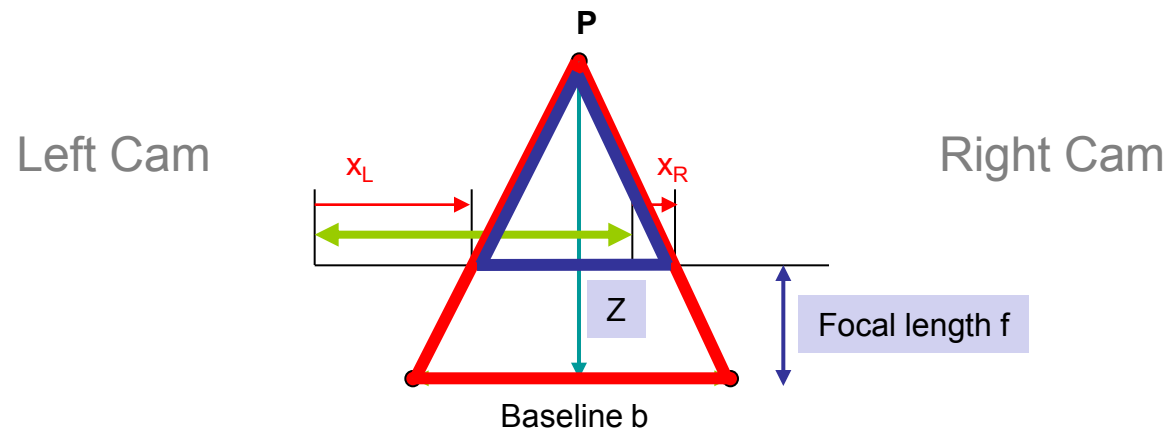


Disparity: Relationship to Depth.



Similar Triangles!

Disparity: Relationship to Depth.



We get:

$$\frac{b}{Z} = \frac{(b + x_R) - x_L}{Z - f} \Rightarrow d = x_L - x_R = \frac{f \cdot b}{Z}$$

Disparity and the Baseline.

$$d = \frac{b \cdot f}{Z}$$

Using this relationship we can draw important conclusions:

- Disparity values are inverse proportional to the depth of a point Z :
 - Far points have low disparity (for example the horizon has disparity of zero).
 - Close points have a high disparity.
- The disparity is proportional to the baseline b
 - The larger the baseline, the higher the disparity.
- The disparity resolution scales linearly with imager resolution:
 - High resolutions allow accurate disparity measurements.

Disparity Maps: Dense Correspondences.

Disparity for every pixel in the image => **Disparity Map**

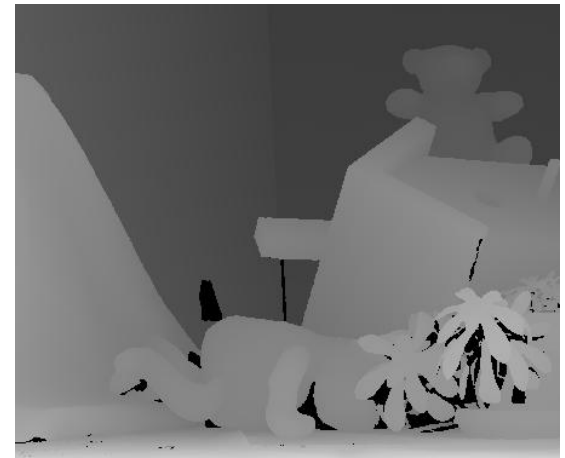
Typically encoded using intensities: „close points are bright – far ones are dark“



Left Image



Right Image



Disparity Map

The Horopter.

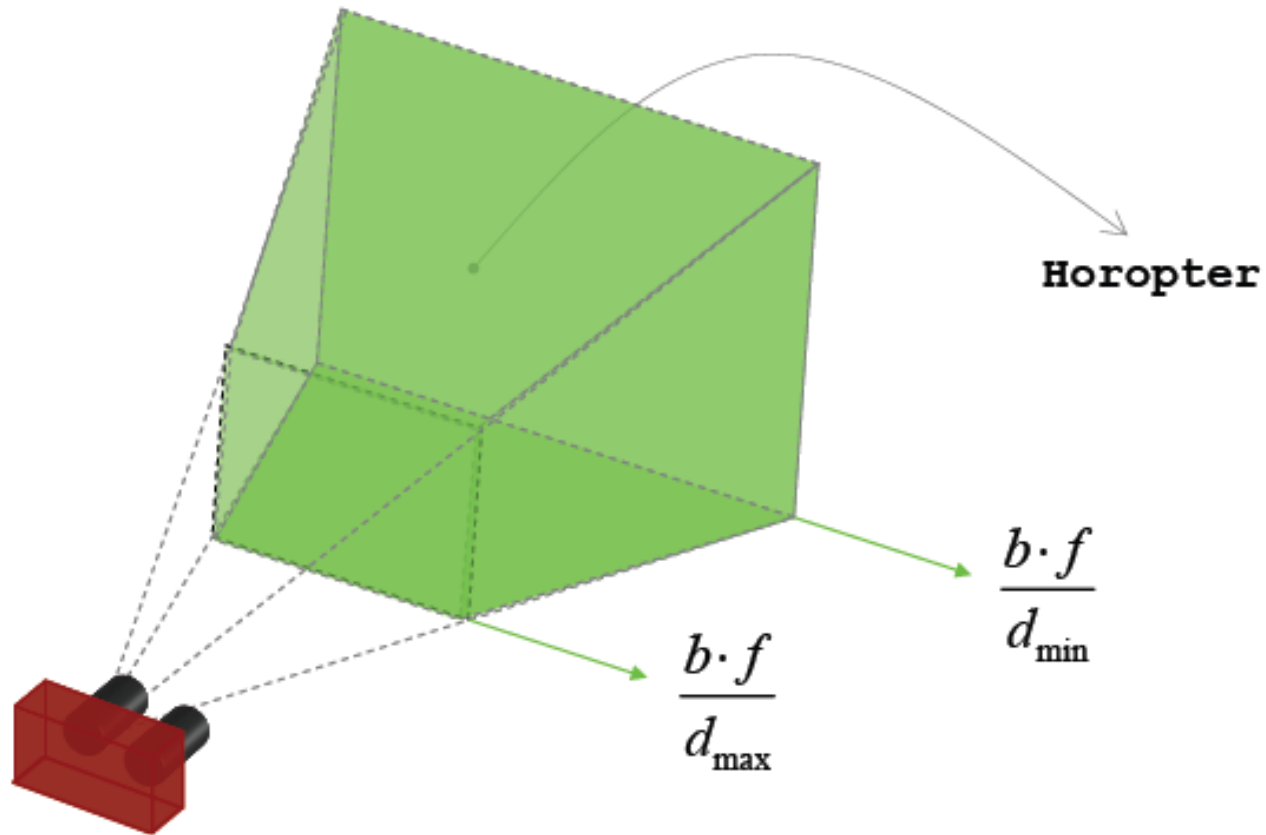


Image of S. Mattocchia.

Disparity-Planes.

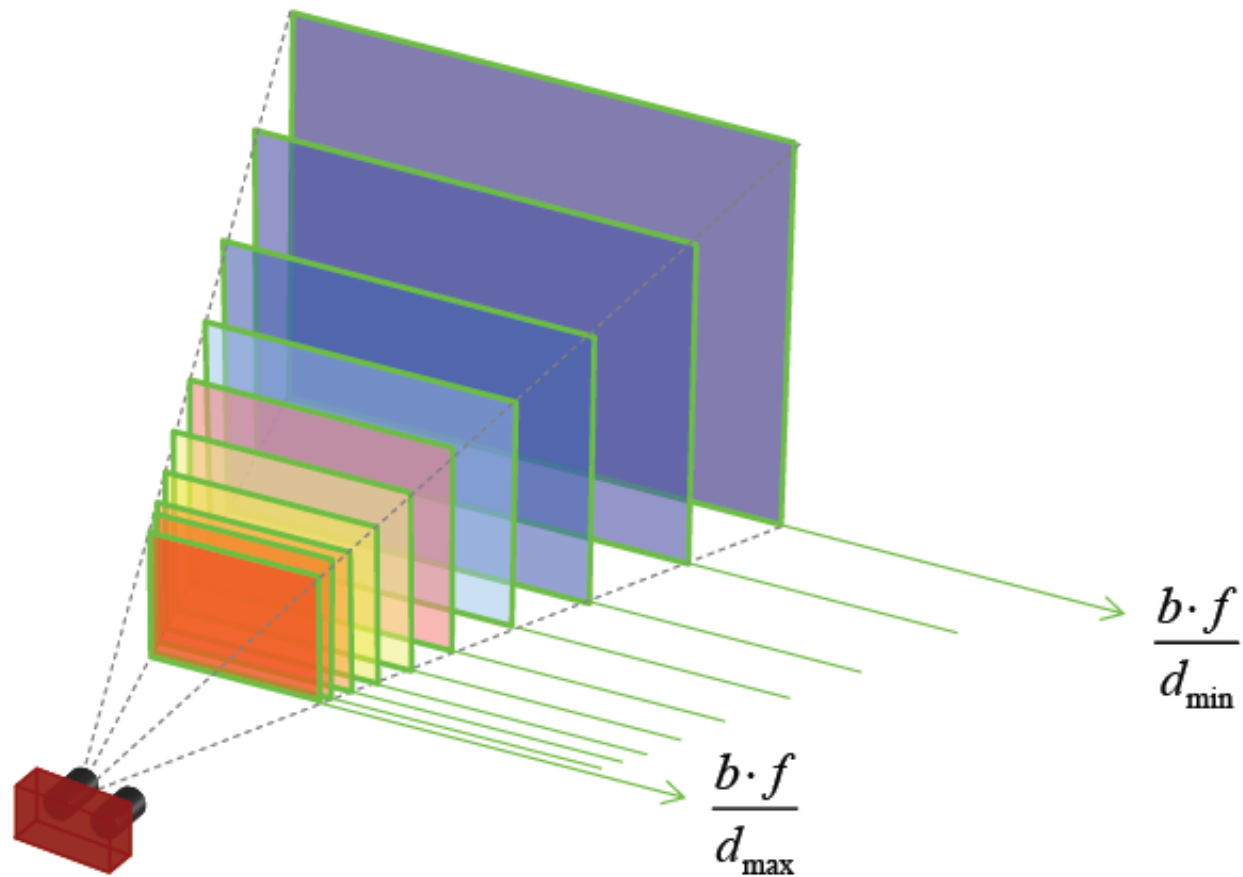


Image of S. Mattocchia.

Stereo Vision Systems: Stereo Matching and Triangulation.

Stereo Matching

Compute a disparity for every pixel of the image.

Next Topic!

Triangulation

Compute the 3D position for every disparity.

Next but one Topic!