Computer Vision
(Summer Semester 2023)

Lecture 9

Stereo Vision

- Slides based on
  - Szeliski’s book “Computer Vision: Algorithms and Applications”: http://szeliski.org/Book/
  - Slides from Brown University: https://cs.brown.edu/courses/csci1430/
  - Slides from Elli Angelopoulou from summer term 2014: https://www5.cs.fau.de/lectures/ss-14/computer-vision-cv/

··· Follow

International Conference on 3D Vision
@3DVconf

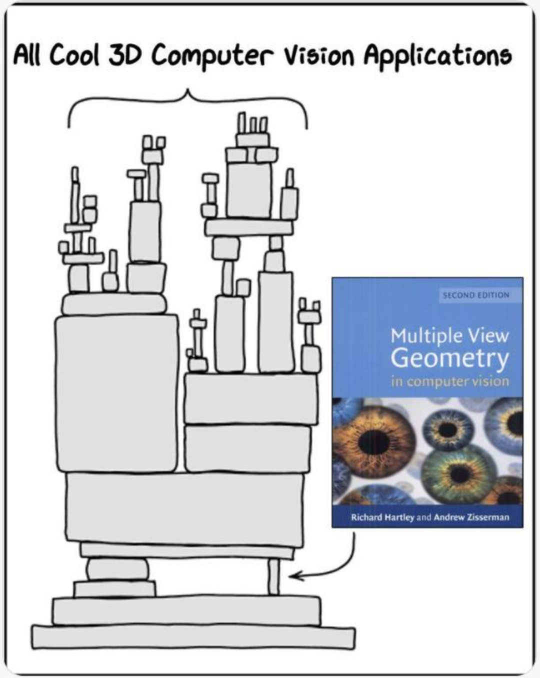
Since 2013, 3DV has provided a premier platform for disseminating research results covering a broad variety of topics in computer vision and graphics.

3dv.org Joined February 2021

45 Following **2,139** Followers

International Conference on 3D Vision @3DVconf · 10h
Do you have the blue bible of MVG? 📘

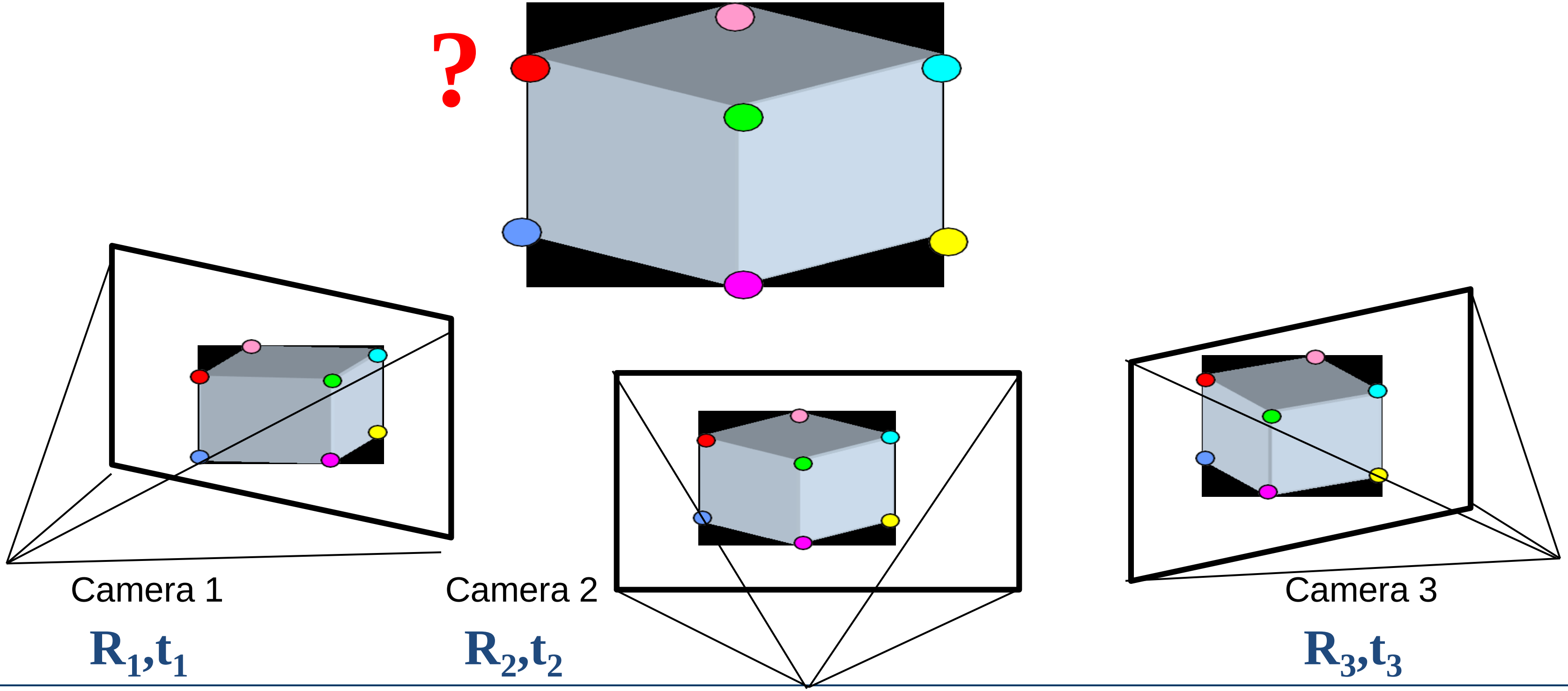
#3DV2024



5 44 189 13.5K

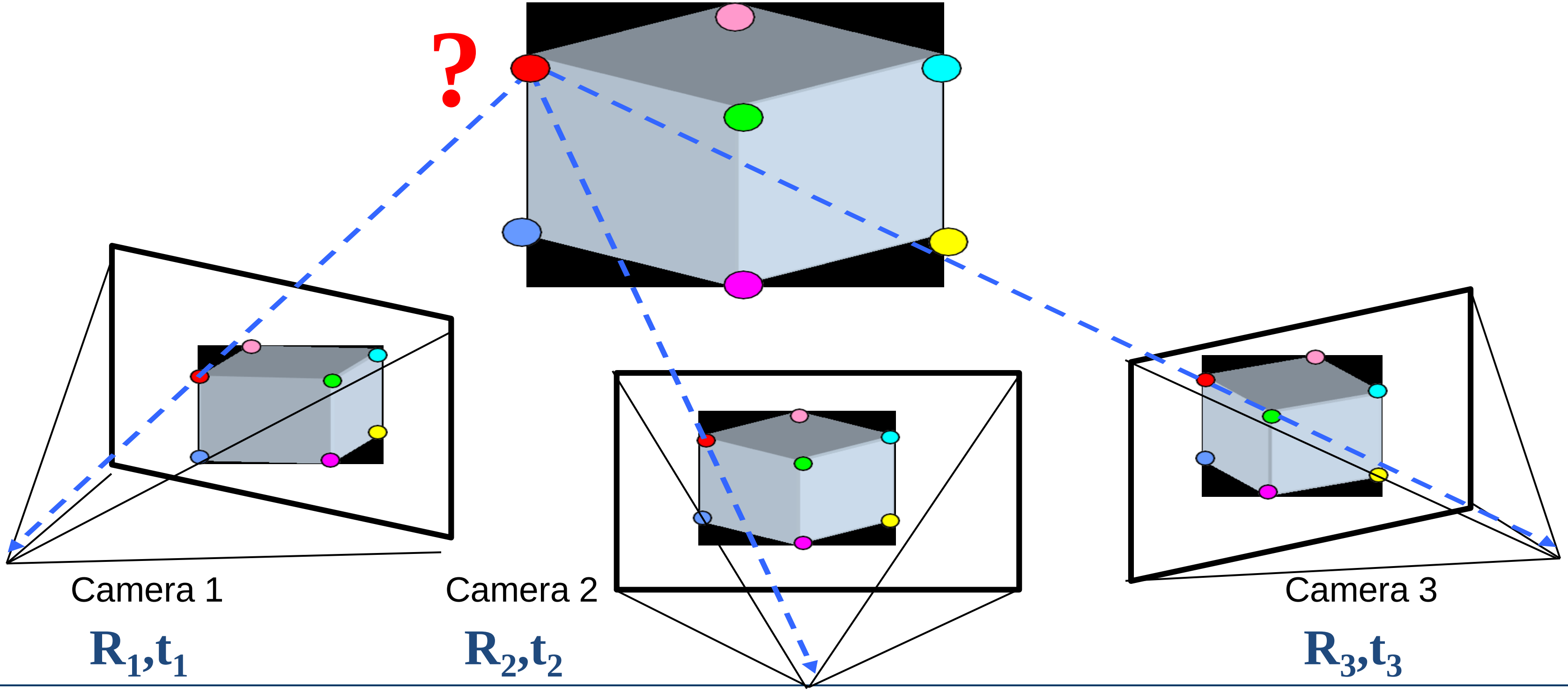
Multi-view geometry problems

- **Structure:** Given projections of the same 3D point in two or more images, compute the 3D coordinates of that point



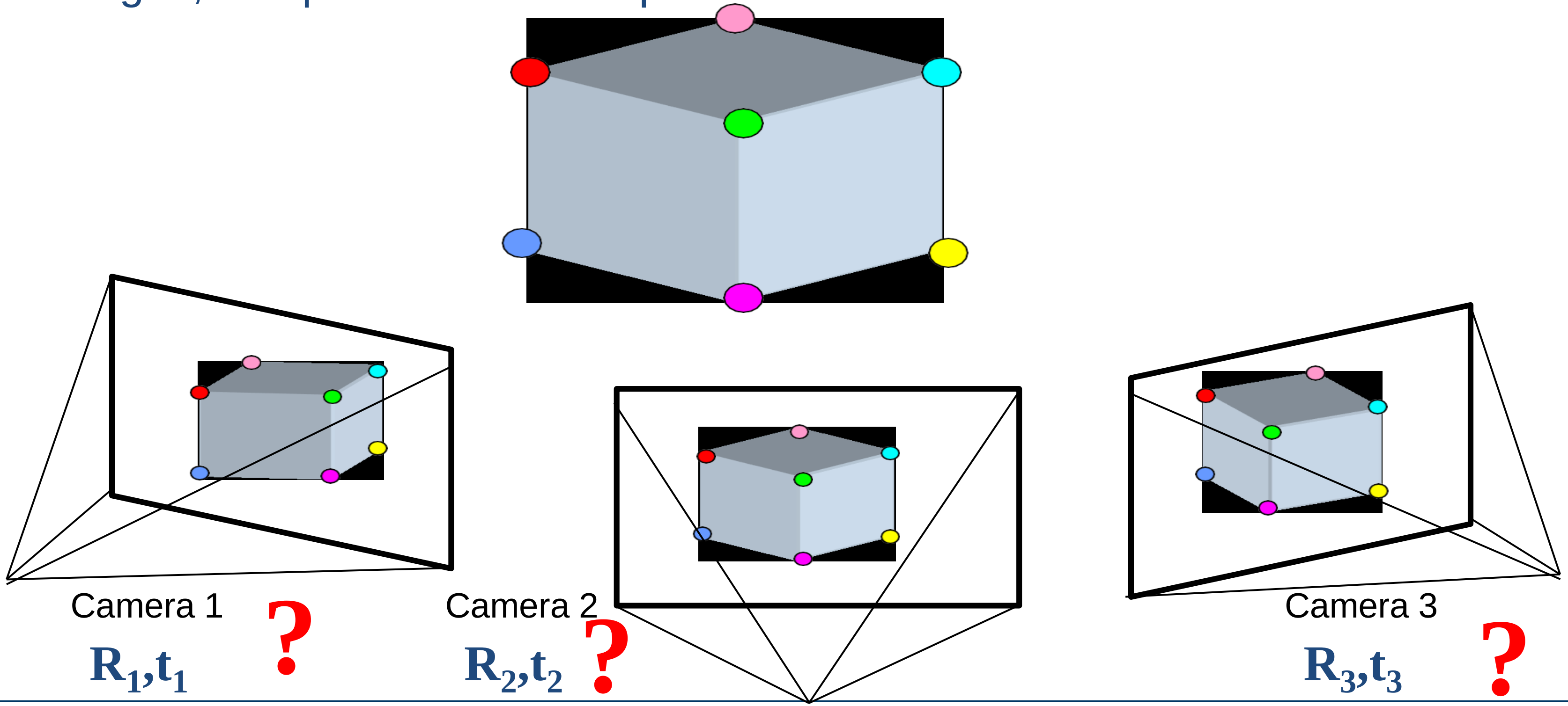
Multi-view geometry problems

- **Structure:** Given projections of the same 3D point in two or more images, compute the 3D coordinates of that point



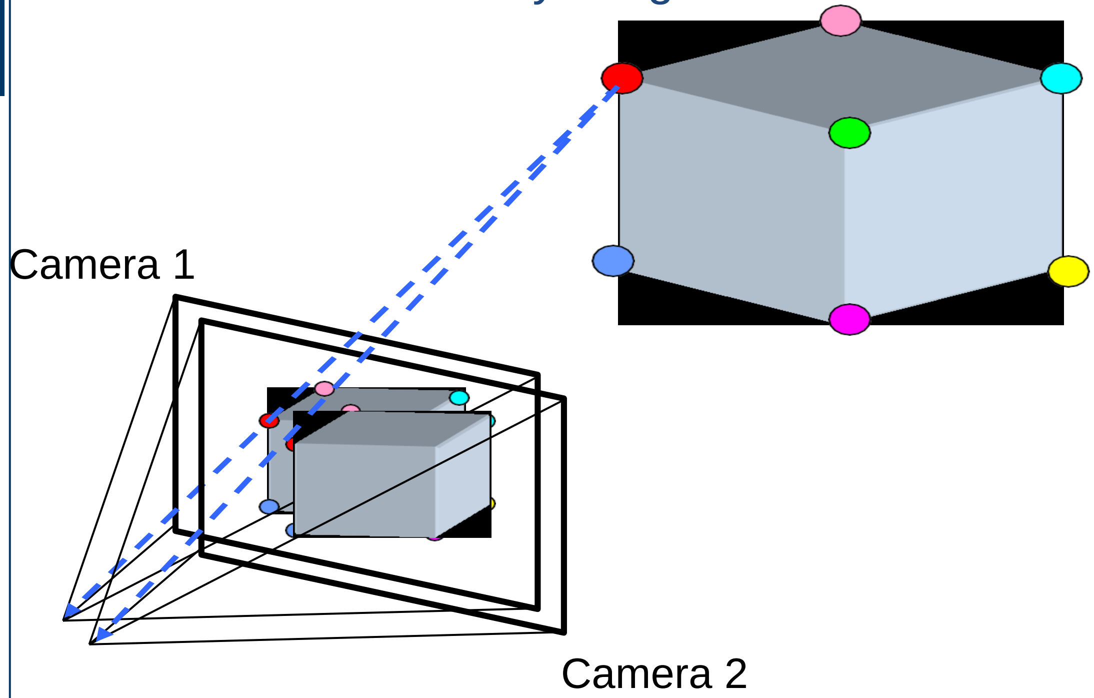
Multi-view geometry problems

- **Motion:** Given a set of corresponding points in two or more images, compute the camera parameters



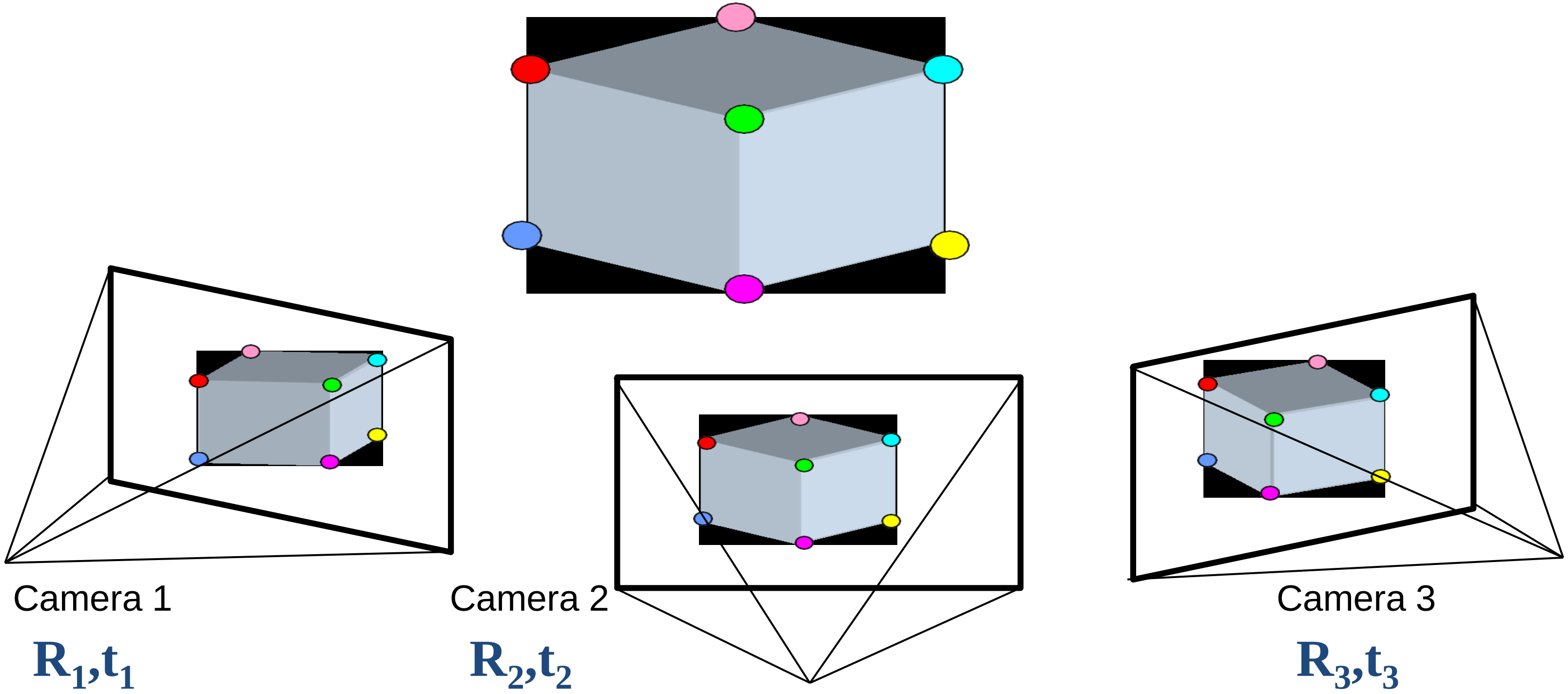
Multi-view geometry problems

- **Optical flow:** Given two images, find the location of a world point in a second close-by image with no camera info.



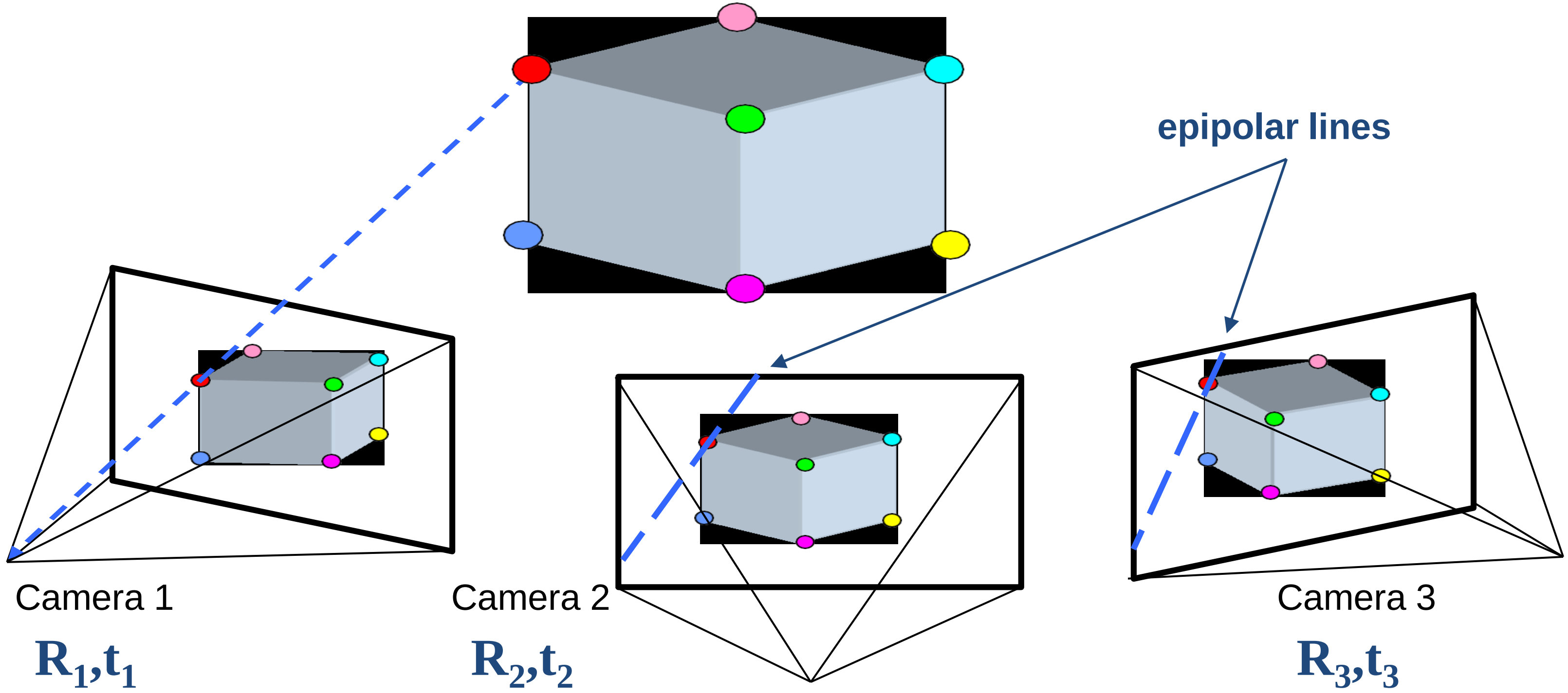
Multi-view geometry problems

- **Stereo correspondence:** Given a point in one of the images, where could its corresponding points be in the other images?



Multi-view geometry problems

- **Stereo correspondence:** Given a point in one of the images, where could its corresponding points be in the other images?

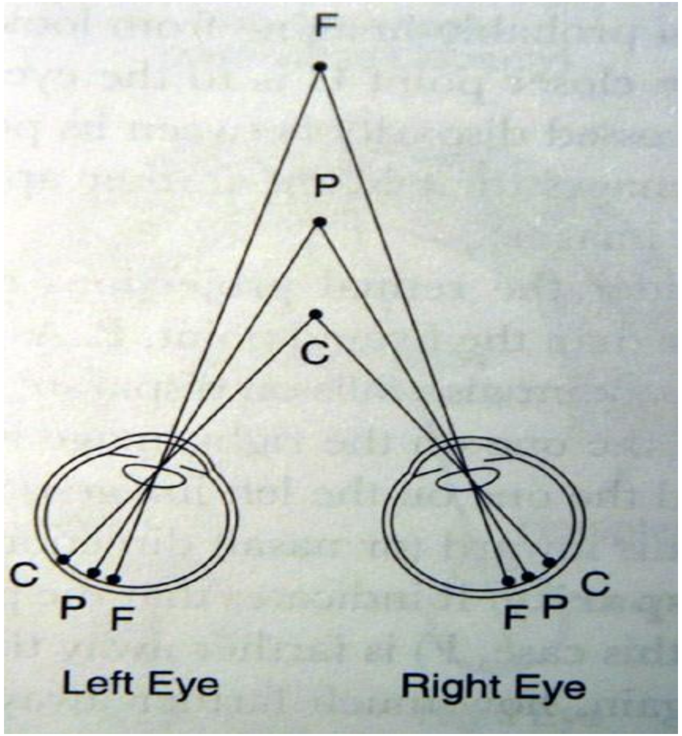


Binocular Stereo Example

Stereo Vision

- **Goal:** Infer information about the 3-D structure and distances of a scene from two or more images taken from different viewpoints.
- A stereo system must solve two subproblems:
  - Correspondence problem
  - Reconstruction
- *Correspondence Problem*: which pixel (point) on the left image and on the right image are projections of the same scene point.
- Once the point correspondence is established, we can compute the relative shift, the *disparity*, between the two projections.
- *Reconstruction*: The disparity data is then converted to a 3D map. In order to transform the disparity data to 3D measurements, we need some form of knowledge about the geometry of the stereo system.

Depth from Binocular Disparity



Sign and magnitude of disparity

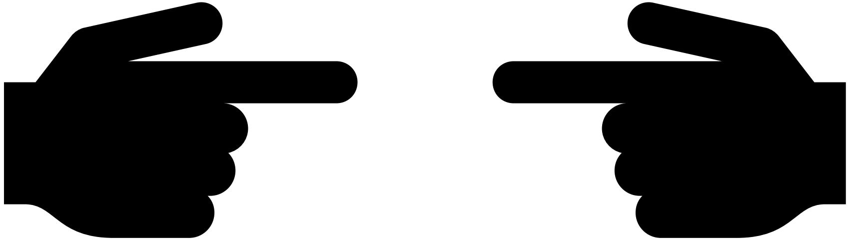
P converging point

C object nearer projects to the outside of the **P**, disparity = +

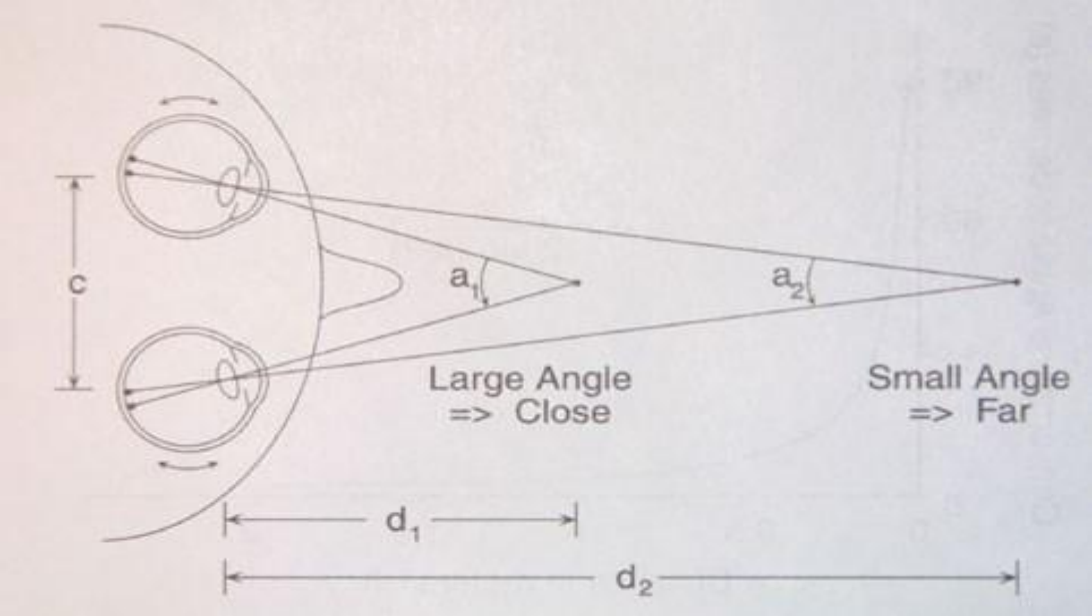
F object farther projects to the inside of the **P**, disparity = -

Disparity

Put your hands in front of your face and look at a far object with one eye closed then the other



Depth from Convergence

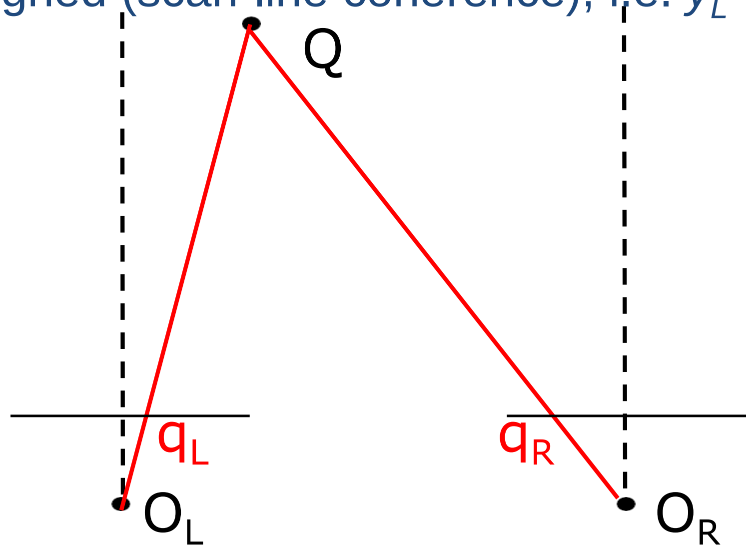


$$d = \frac{c}{2tan(a/2)}$$

Human performance: up to 2-2.5 meters

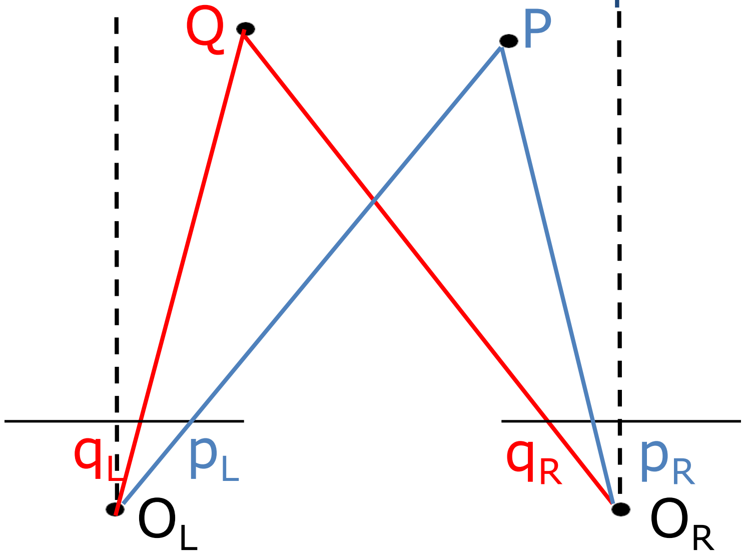
Simple Binocular Stereo Setup

- Parallel optical axes, i.e. the fixation point (the point where the 2 optic axes intersect) is at infinity.
- Both image planes lie on the same plane.
- Their scan lines are aligned (scan-line coherence), i.e. $y_L = y_R$.



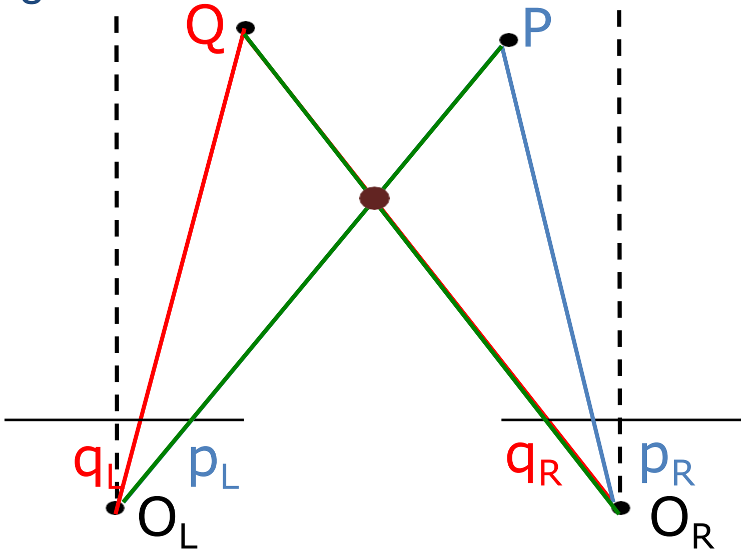
Correspondence and Triangulation

- When we correspond correctly (i.e. q_L with q_R and p_L with p_R), the intersection of the corresponding rays gives the 3D location of scene point that generated the projections (i.e. Q and P accordingly).
- Only points lying on the same scanline can correspond



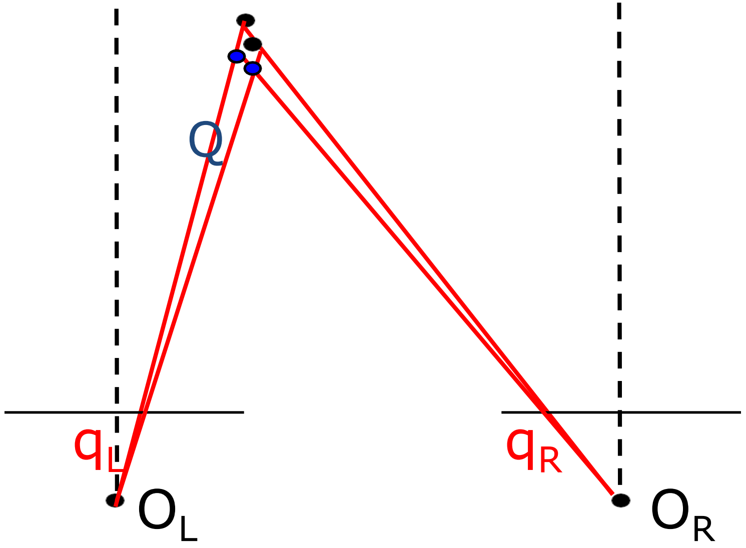
Impact of Correspondence

- A mistake in correspondence, e.g. q_R is matched with p_L, will result in the intersection of rays that correspond to projections of distinct points (Q and P) . As a result the wrong 3D location is recovered.

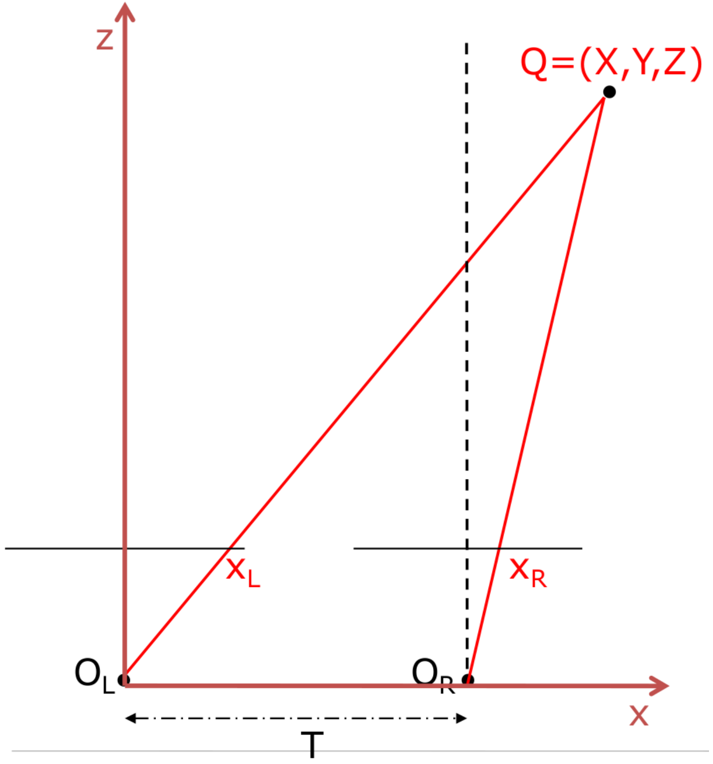


Noise and Correspondence

- The noise in the image capture process (sensor noise, quantization, discretization) introduces inaccuracies in the projection rays that directly affect the triangulation process.

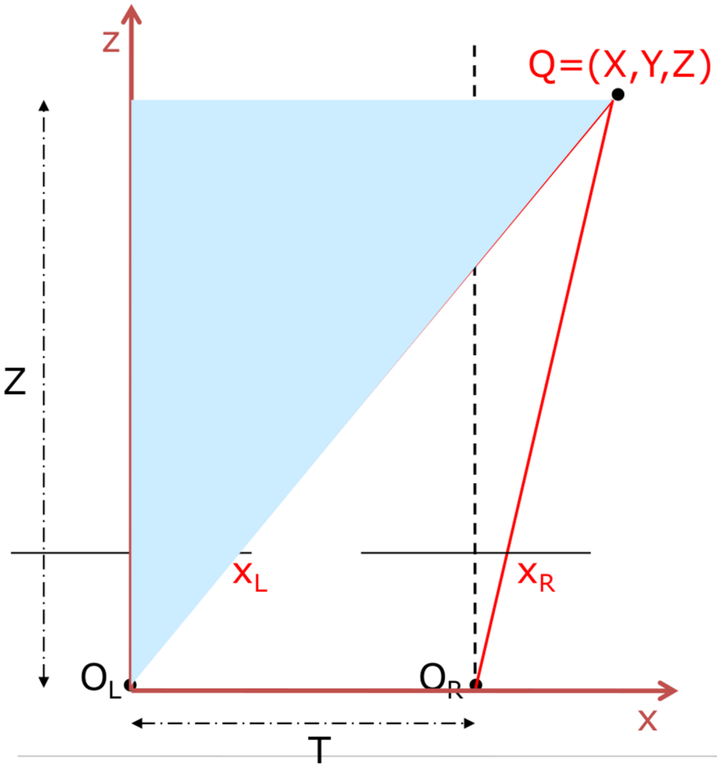


Triangulation

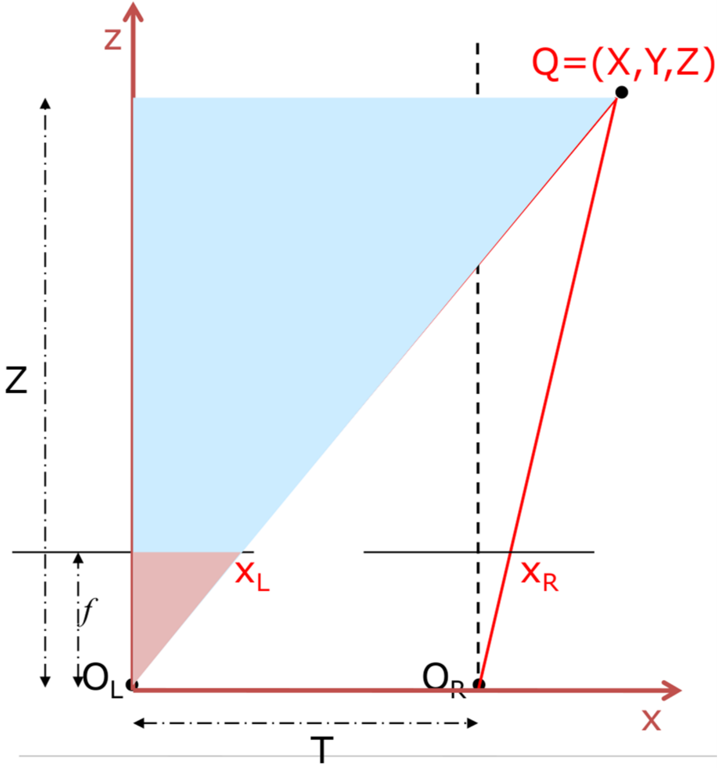


- Assume that the correspondence has been correctly established.
- Under the simple binocular setup (parallel optic axes and scan-line coherence), the only difference between the two projections q_L and q_R is in the x- component, i.e. x_L versus x_R.
- Let T be the **baseline**, i.e. the distance between the two COPs.

Triangulation



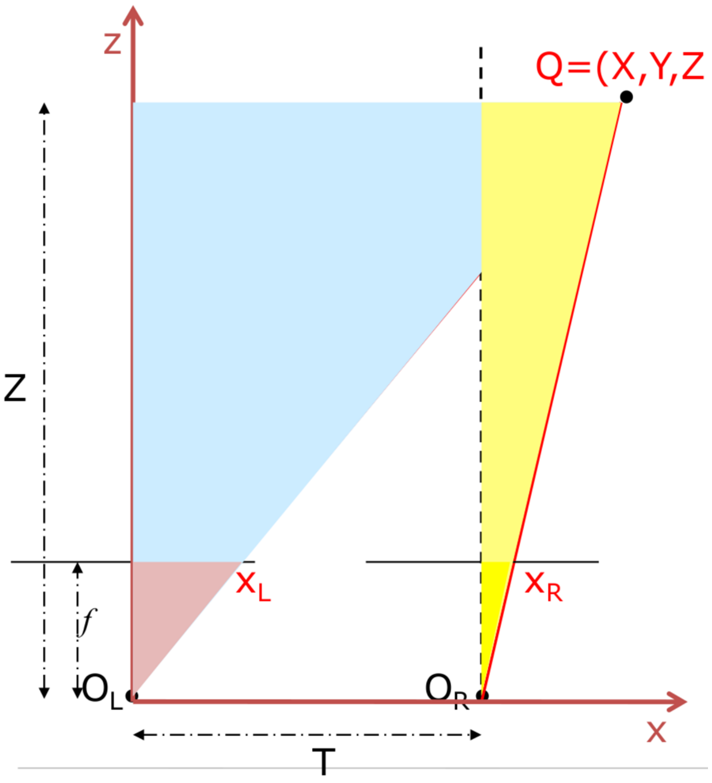
Triangulation



From the similar triangles:

$$\frac{x_L}{f} = \frac{X}{Z} \Rightarrow X = x_L \frac{Z}{f}$$

Triangulation



From the similar triangles:

$$\frac{x_L}{f} = \frac{X}{Z} \Rightarrow X = x_L \frac{Z}{f}$$

From the 2nd set of similar triangles:

$$\frac{x_R}{f} = \frac{X - T}{Z}$$

By replacing X in the 2nd eq.:

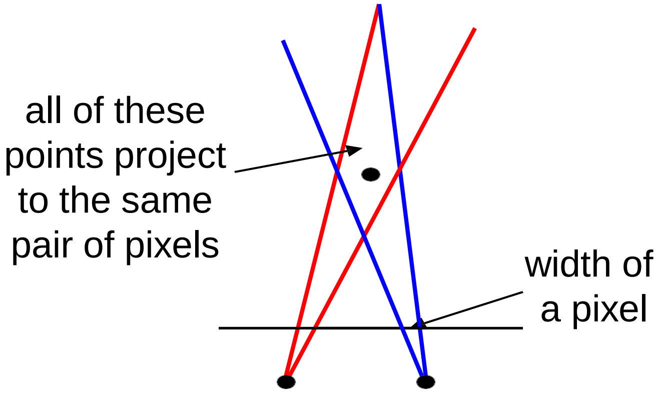
$$\frac{x_R}{f} = \frac{x_L \frac{Z}{f} - T}{Z} \Rightarrow x_R Z = x_L Z - fT$$

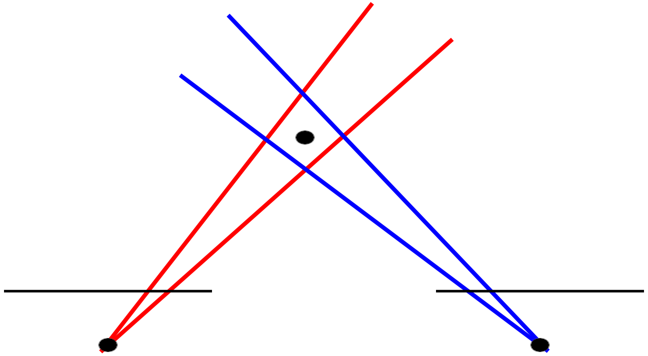
$$\Rightarrow \boxed{Z = f \frac{T}{x_L - x_R} = f \frac{T}{d}}$$

where *d* is the *disparity*:

$$d = x_L - x_R$$

Impact of Baseline





Small Baseline

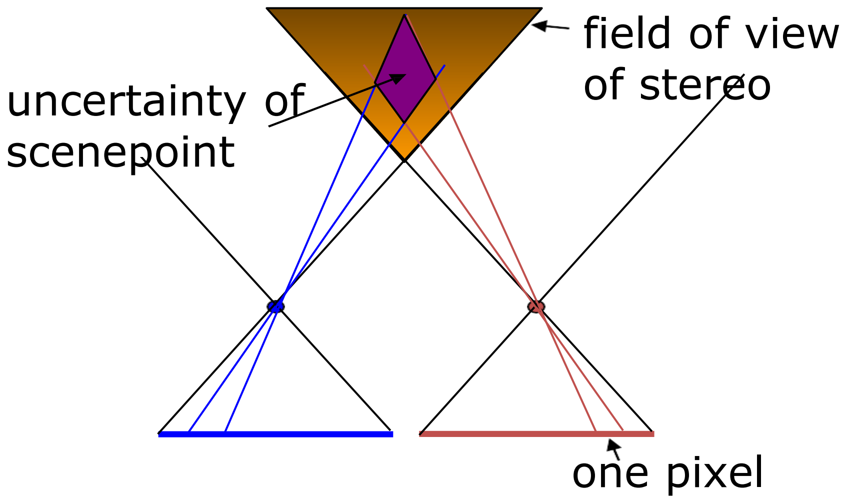
Large Baseline

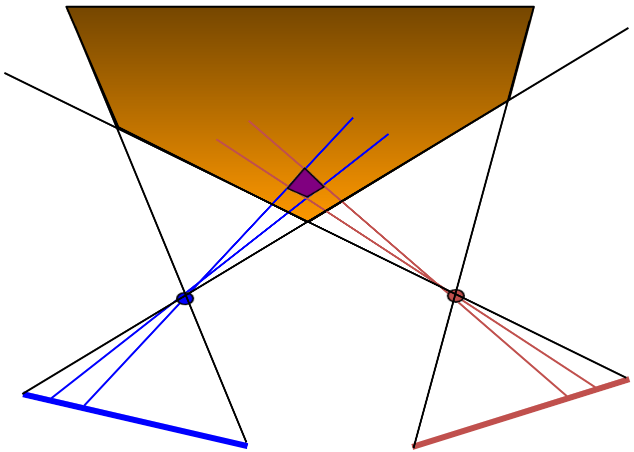
What's the optimal baseline?

- Too small: large depth error
- Too large: difficult search problem

- Appearance may change between the 2 viewpoints
- Decrease in the region of the scene that is mutually visible.

Vergence



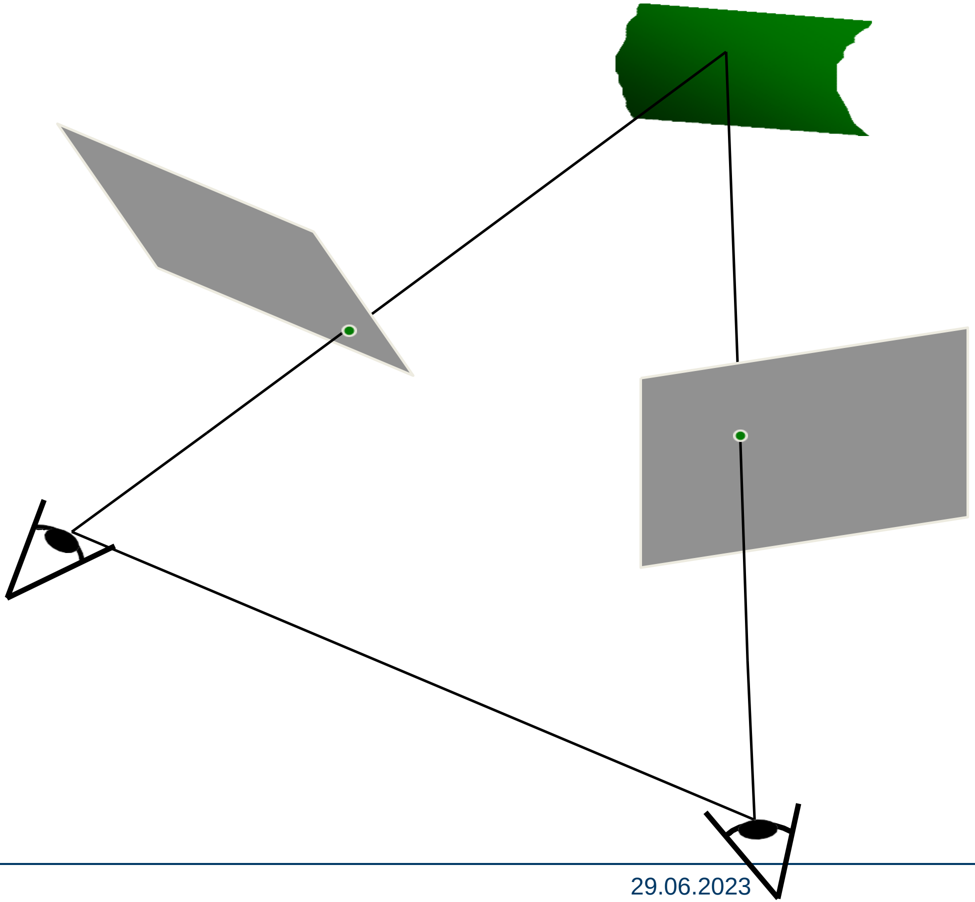


Optical axes of the two cameras need not be parallel

- Solution: Vergence (turn cameras towards each other)
  - Increases the field of view
  - Increases accuracy in the correspondence

Stereo Image Rectification

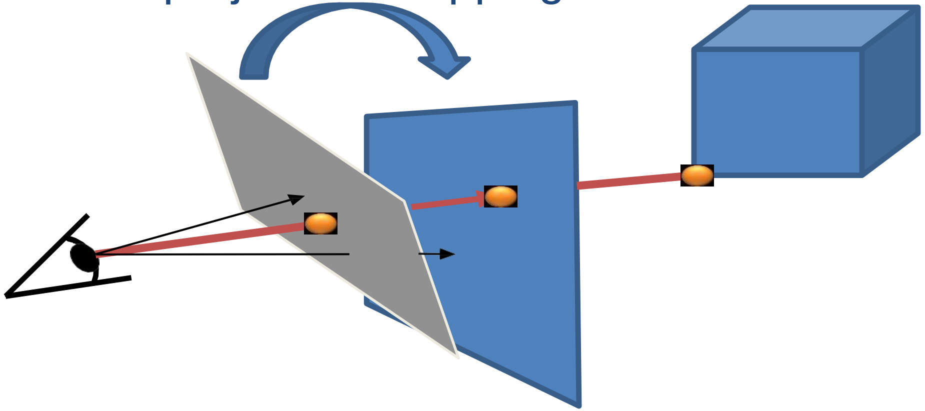
- So far we have assumed:
  - parallel optic axes
  - scan-line coherence
- Such a setup can lead to inaccuracies.
- More commonly cameras are *verged*, i.e. **the 2 optic axes intersect each other.**
- Can we use the same math?



Stereo Image Rectification

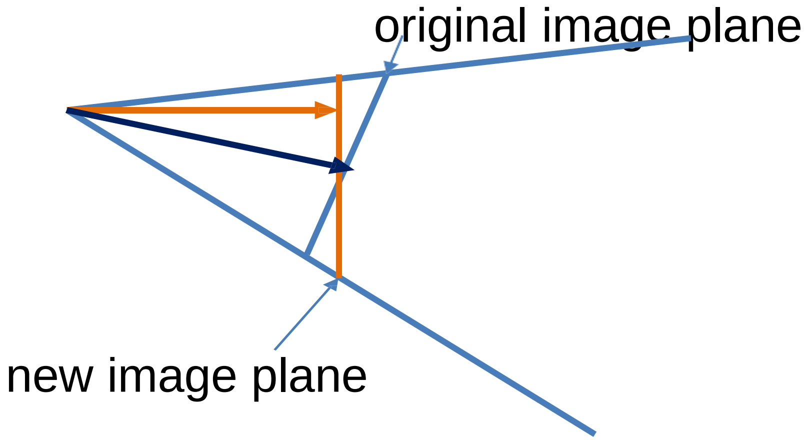
- Yes, with a simple trick: we can warp an image to a new image plane
- This warping is a projective mapping
- It assumes that the camera position does not change

projective mapping



Stereo Image Rectification

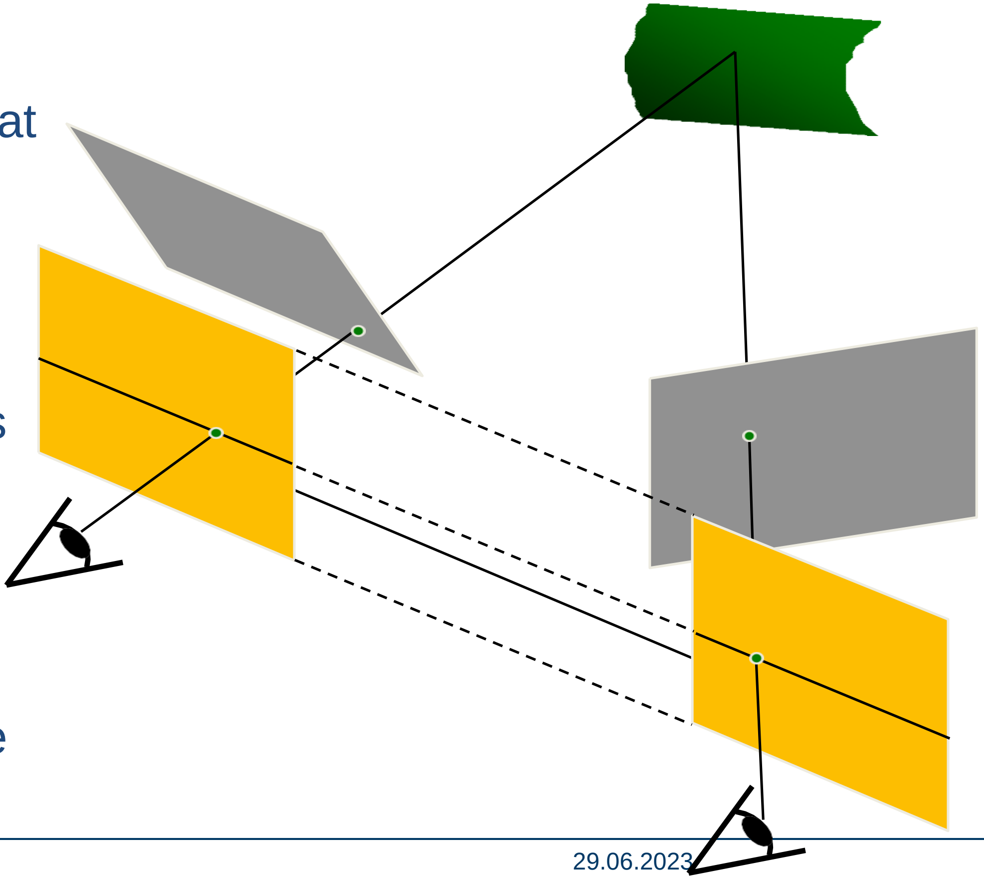
- 2D example



Stereo Image Rectification

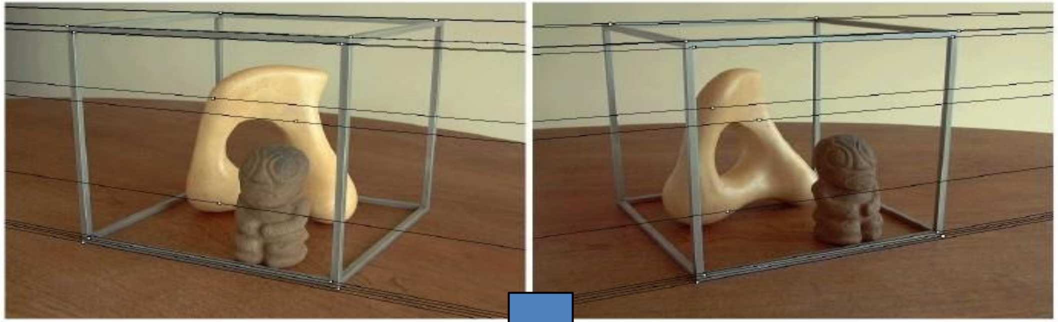
Rectification

- Re-project the two images so that the previous simple setup is achieved
- Define a new image plane which is perpendicular on the plane spawn by the two cameras positions and the view directions
- Warp images to this plane, create two new images, which are now scan-line coherent.
- Do all the computations on these **rectified** images.



Stereo Rectification Example

original images



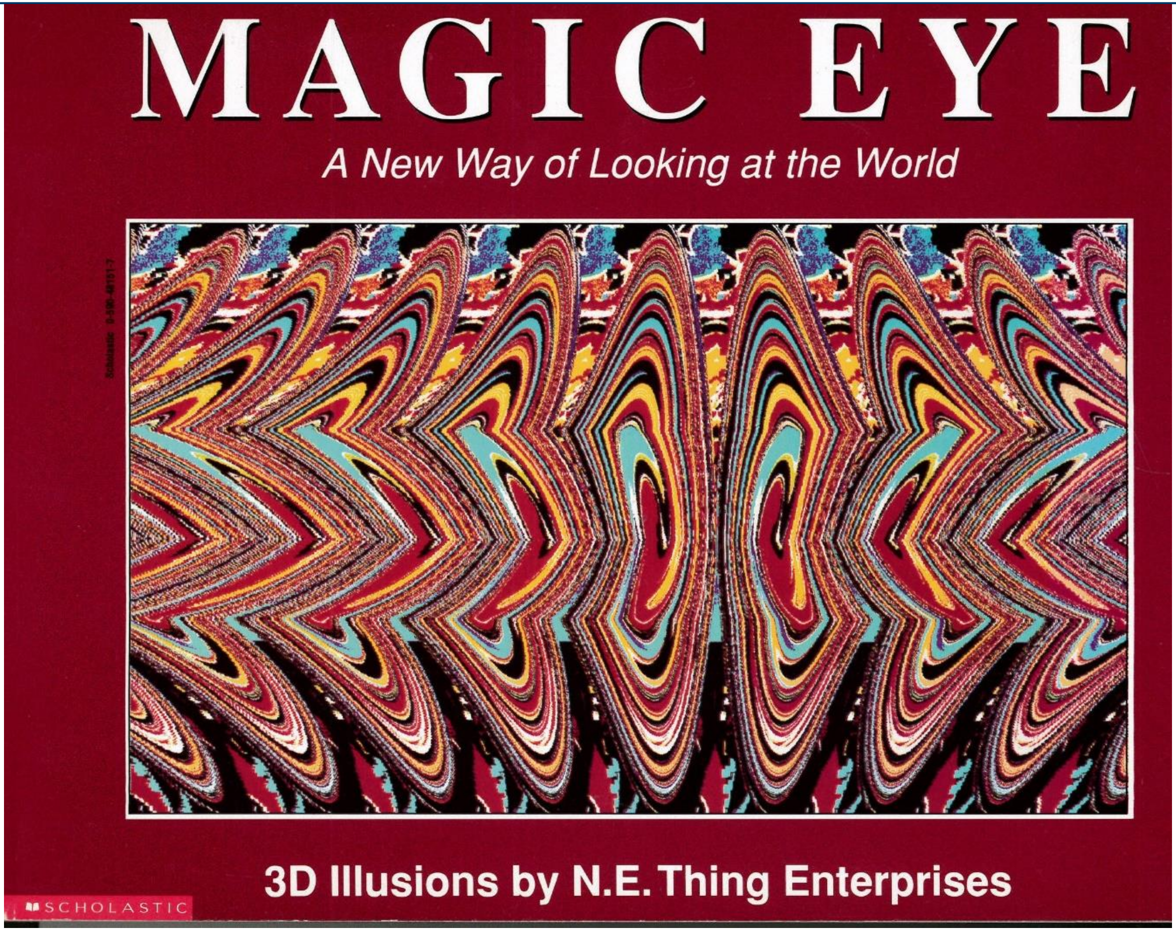
after rectification:
scan-line
coherent

all point correspondences on the same scanline

Correspondence Problem

- Assumptions:
  - Most scene points are visible from both viewpoints
  - Corresponding image regions look similar
- It is a search problem: Given an element in the left image, search the right image to find the corresponding element.
- Three underlying questions:
  - What do we match between the two images? (objects, edges, pixels, sets of pixels?)
  - What measure of similarity do we use?
  - Can we search in a systematic way?

→ **we discussed the same problem already for optical flow !**


MAGIC EYE
A New Way of Looking at the World
3D Illusions by N.E. Thing Enterprises
SCHOLASTIC

Point Correspondence

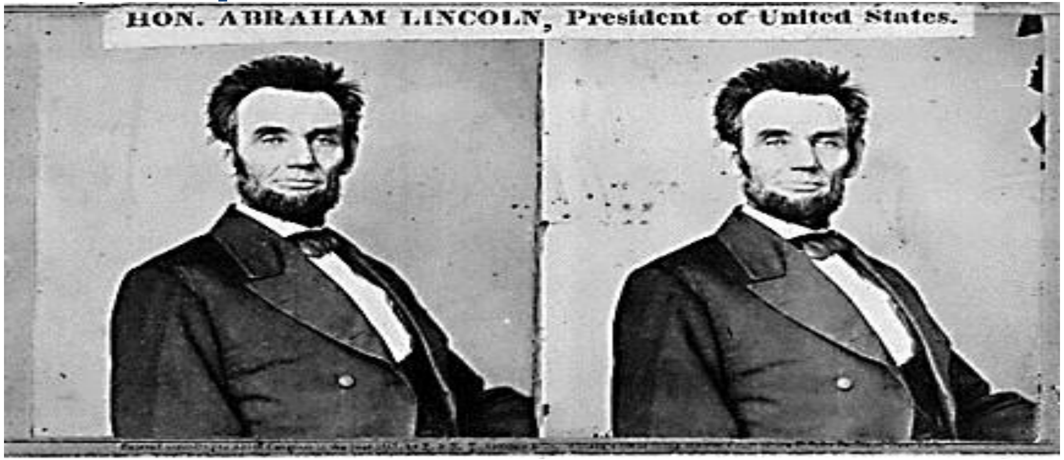
1. Create an image of suitable size. Fill it with random dots. Duplicate the image.
2. Select a region in one image.
3. Shift this region horizontally by a small amount. The stereogram is complete.

To view the stereogram, focus on a point behind the image by a small amount until the two images "snap" together

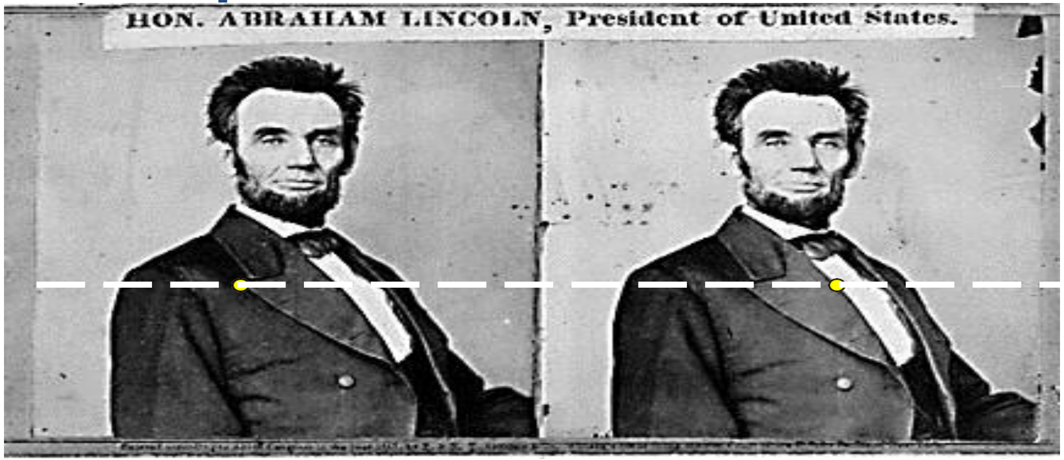
https://en.wikipedia.org/wiki/Random_dot_stereogram

Julesz: had huge impact because it showed that recognition not needed for stereo.

Point Correspondence in Practice



Point Correspondence in Practice

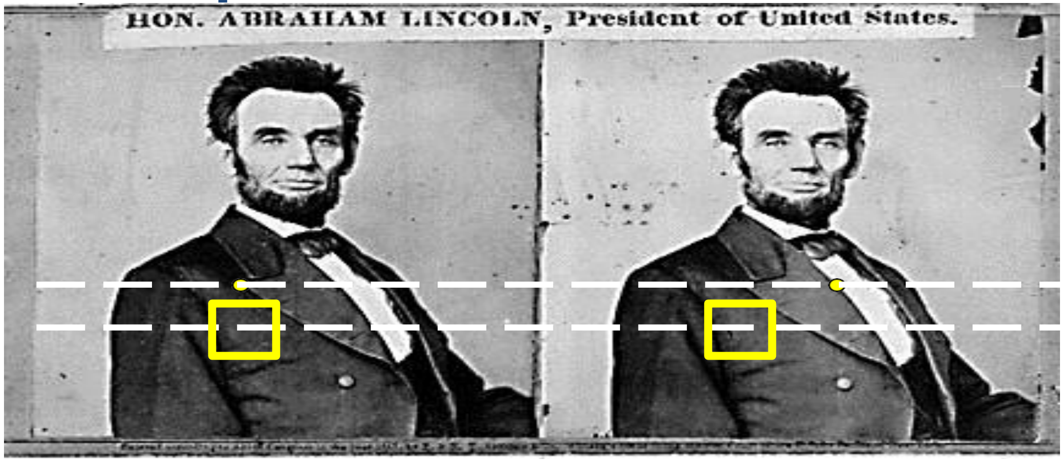


For each scan-line (special case of epipolar line)

For each pixel in the left image

- compare with every pixel on same epipolar line in right image

Point Correspondence in Practice



For each scan-line (more properly epipolar line)

For each pixel in the left image

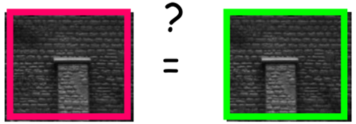
- compare with every pixel on same epipolar line in right image
- pick pixel with closest intensity value (or more general minimum match cost).
- This will never work, so:

Improvement: match ***windows***

Compare Regions around Points

- **Idea:** Compare intensity profiles around neighborhoods of potential points.
- Elements to be matched are now image windows of fixed size.
- The similarity measure is the correlation between windows in the two images.

Similarity Metrics



f g

$$SSD = \sum_{[i,j] \in R} (f(i,j) - g(i,j))^2$$

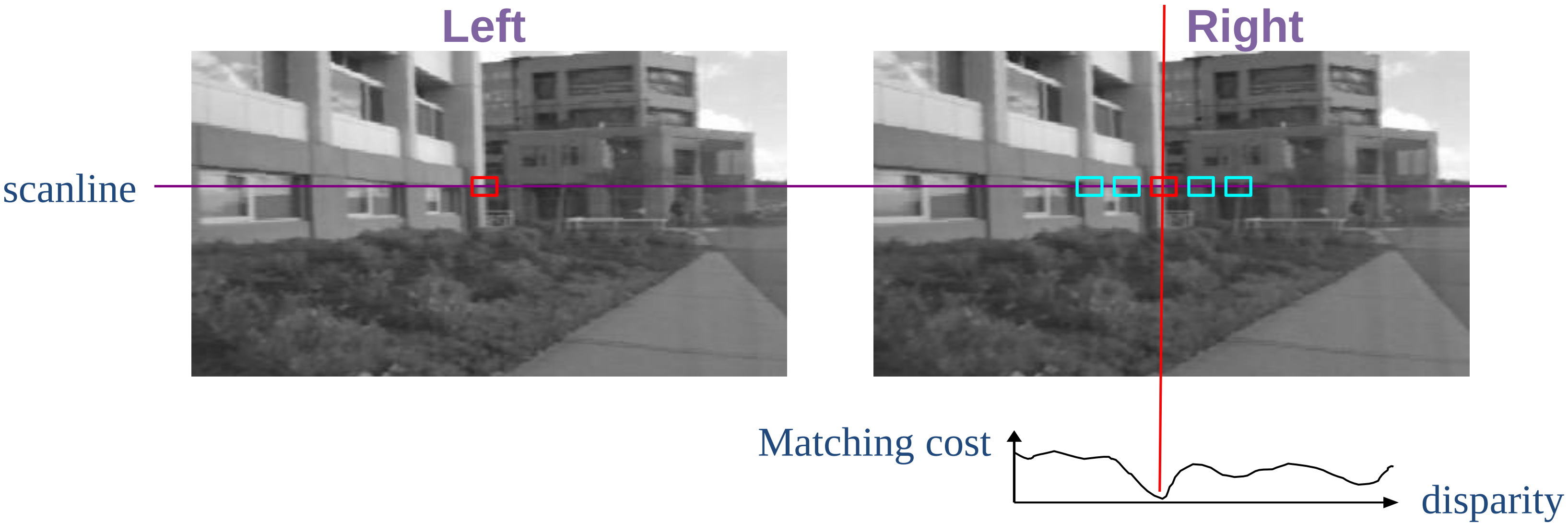
$$C_{fg} = \sum_{[i,j] \in R} f(i,j) g(i,j)$$

$$NC_{fg} = \frac{1}{n-1} \sum_{[i,j] \in R} \frac{(f(i,j) - \bar{f})(g(i,j) - \bar{g})}{\sigma_f \sigma_g}$$

Most popular

For each window, match to the closest window on the horizontal (epipolar) line in the other image.

Correspondence search with similarity constraint



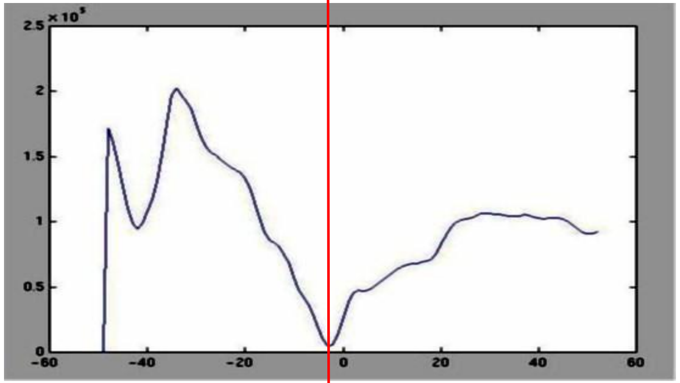
- Slide a window along the right scanline and compare contents of that window with the reference window in the left image
- Matching cost: **SSD** or **normalized correlation**

Correspondence search with similarity constraint

Left

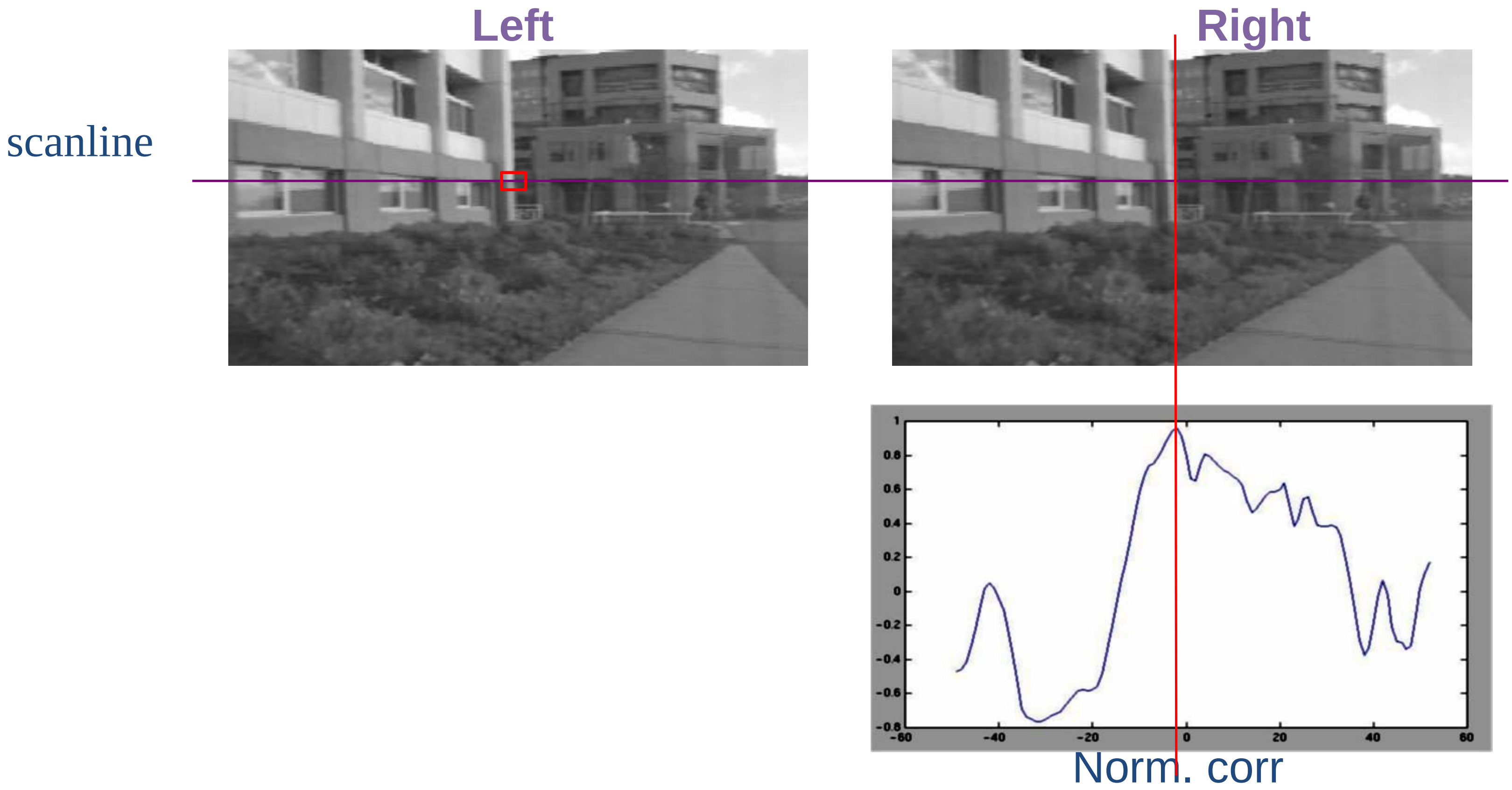
Right

scanline



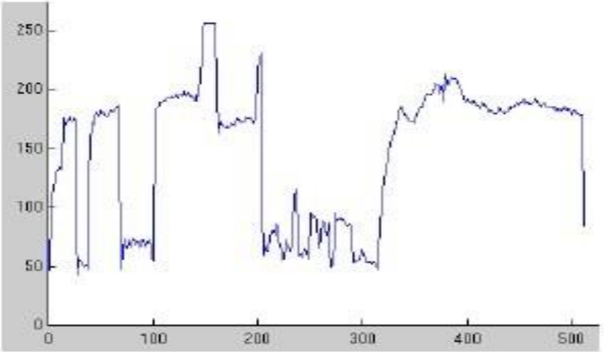
SSD

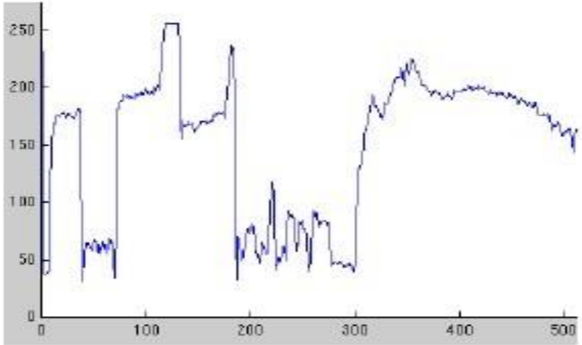
Correspondence search with similarity constraint



Correspondence problem

Intensity profiles





- Clear correspondence between intensities, but also noise and ambiguity

Correlation-based window matching

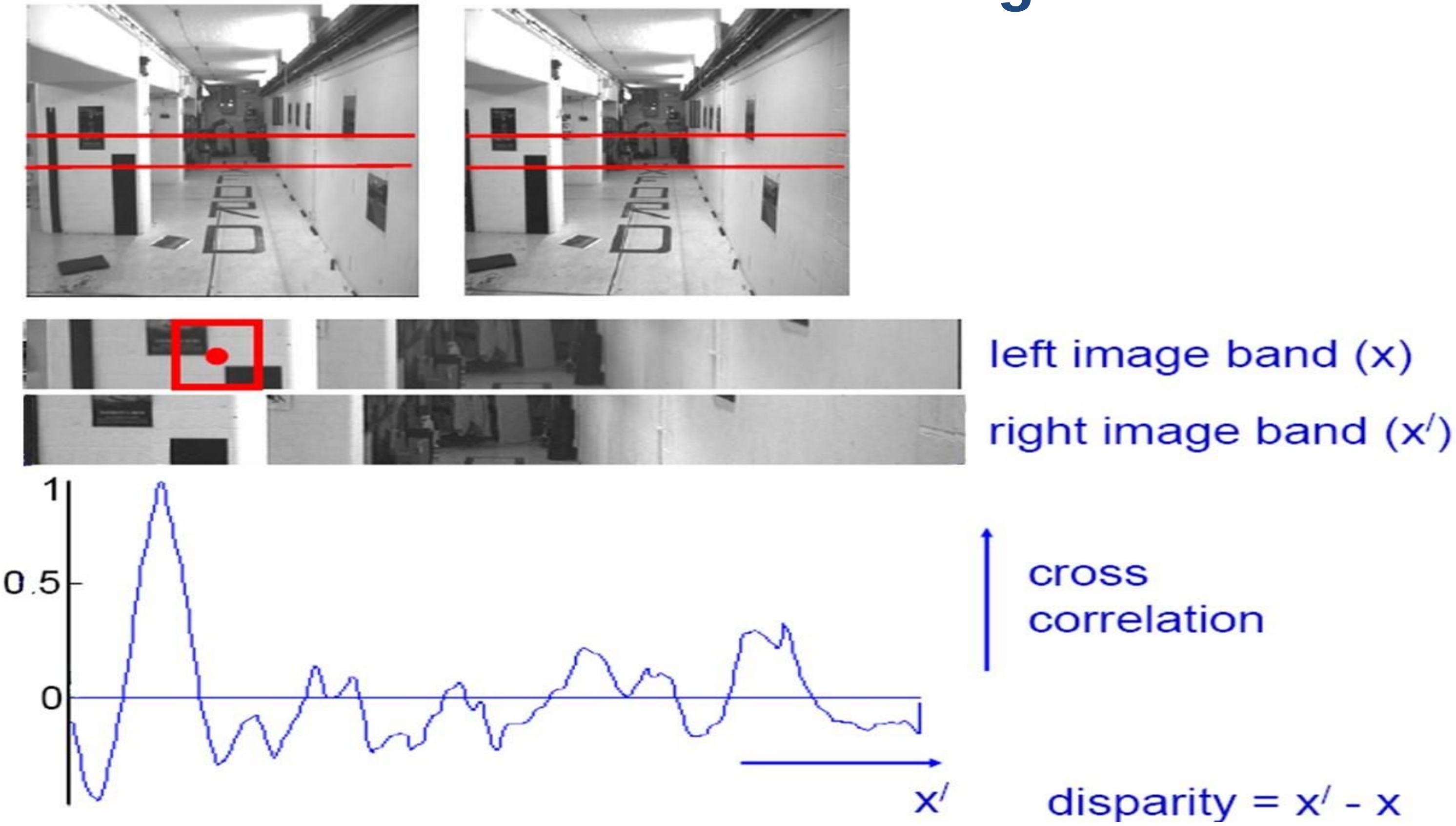
left image band (x)

Correlation-based window matching

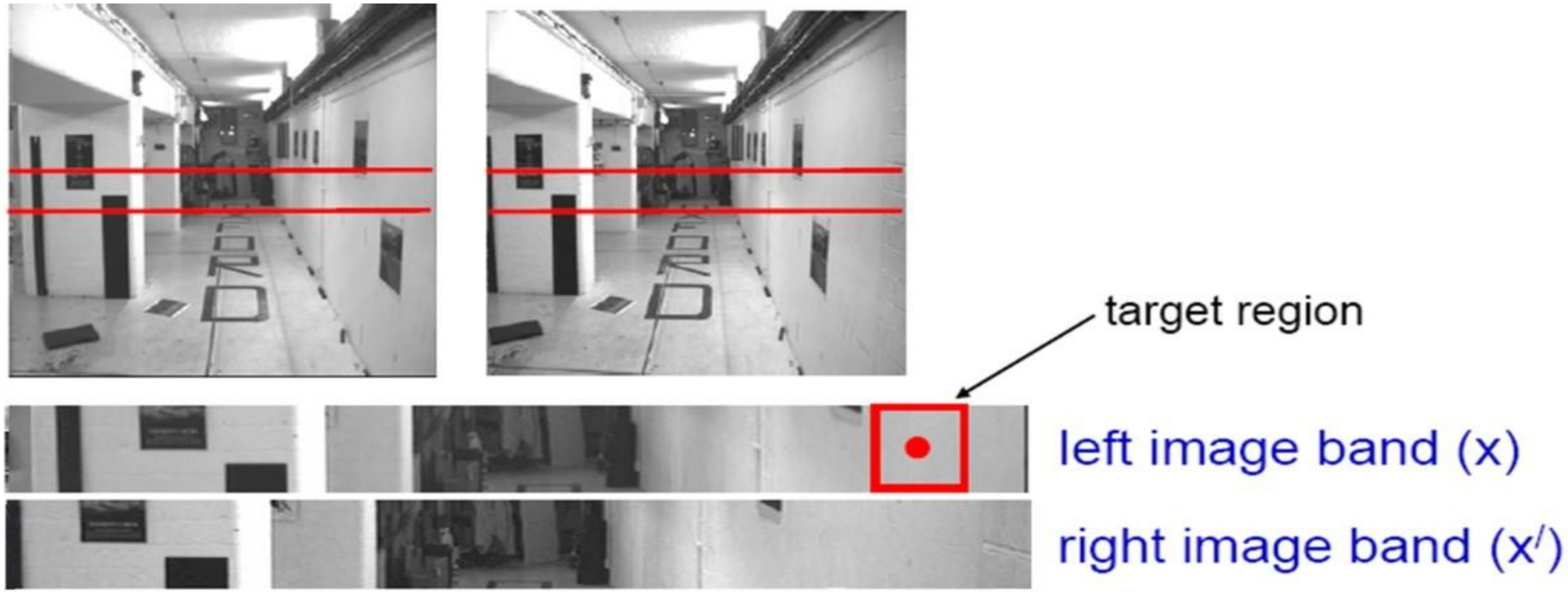
left image band (x)

right image band (x')

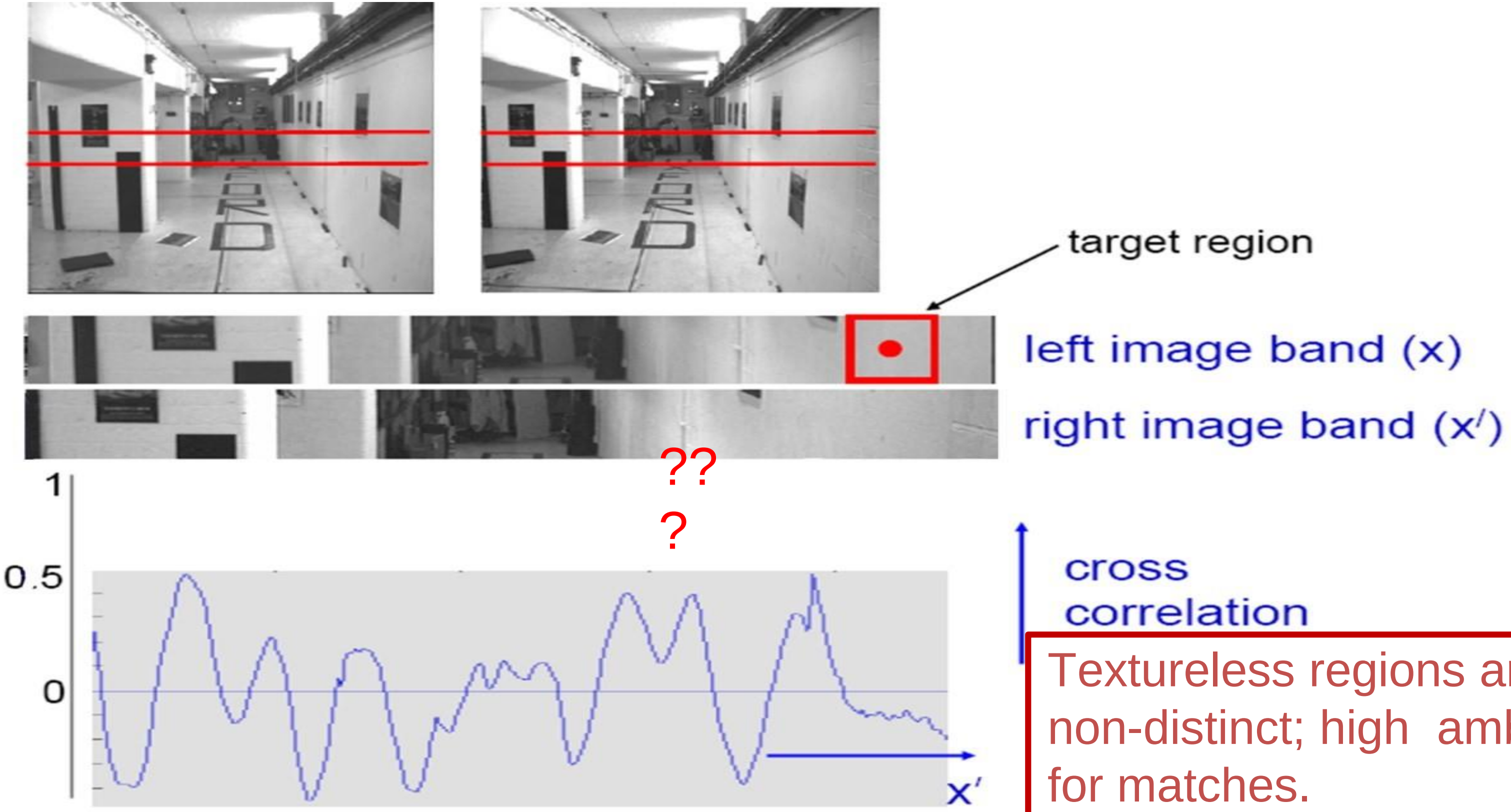
Correlation-based window matching



Correlation-based window matching



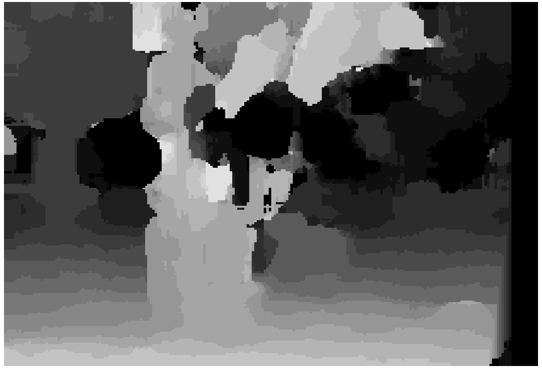
Correlation-based window matching



Textureless regions are non-distinct; high ambiguity for matches.

Window Size

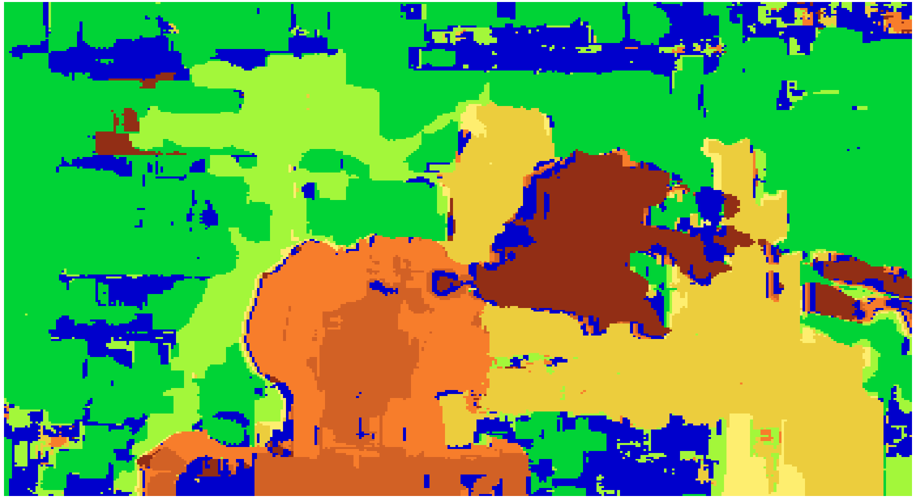
W = 3



W = 20

- Smaller window: more detail, more noise.
- Larger window: less noise, less detail
- Better results with adaptive window size

Results with window search



Window-based matching
(best window size)

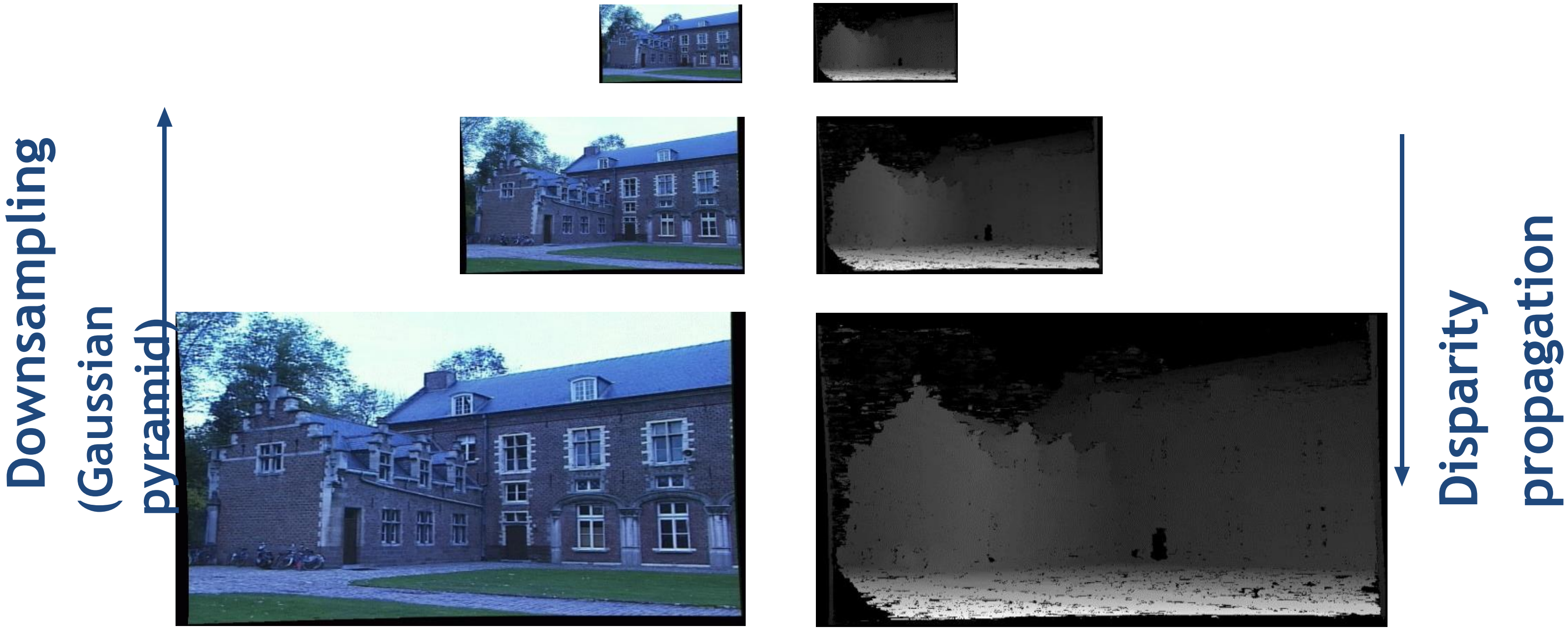
'Ground truth'

Better solutions

- Beyond individual correspondences to estimate disparities:
- Optimize correspondence assignments jointly
  - Scanline at a time (DP)
  - Full 2D grid (graph cuts)

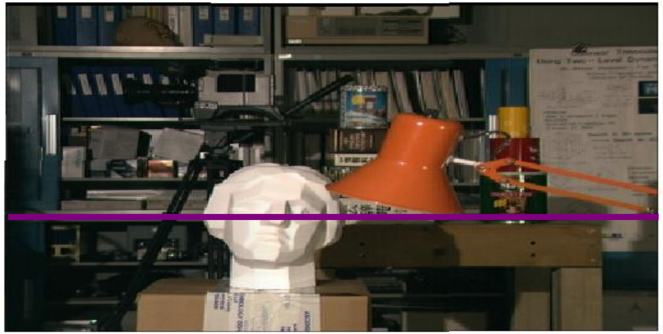
Hierarchical Correspondence

- Allows faster computation
- Can handle large disparity ranges

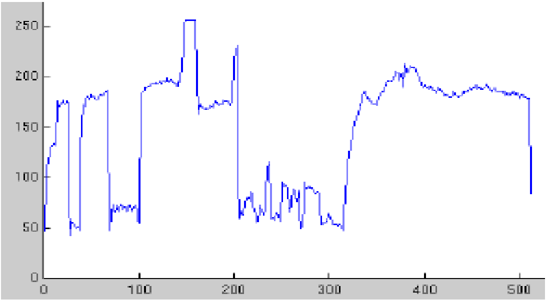


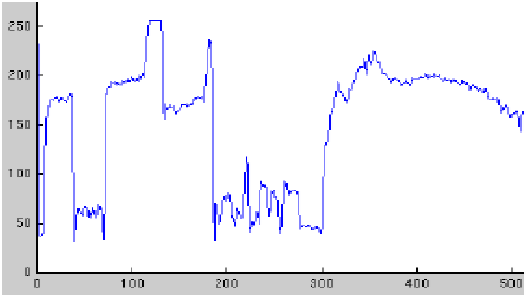
Scanline stereo

- Try to coherently match pixels on the entire scanline
- Different scanlines are still optimized independently

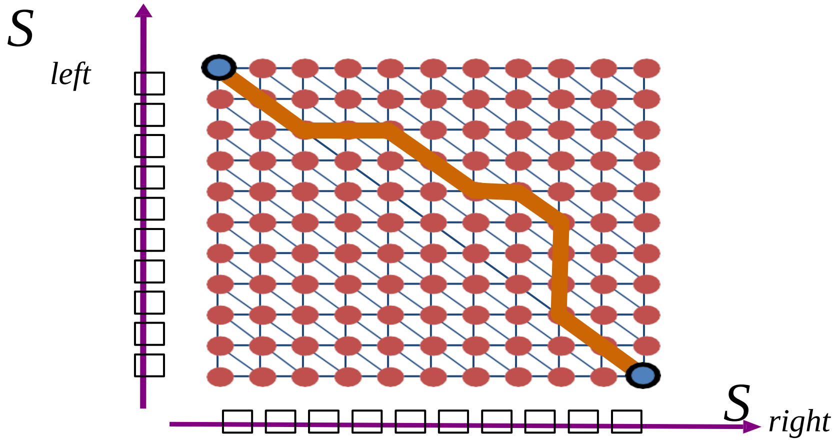


intensity

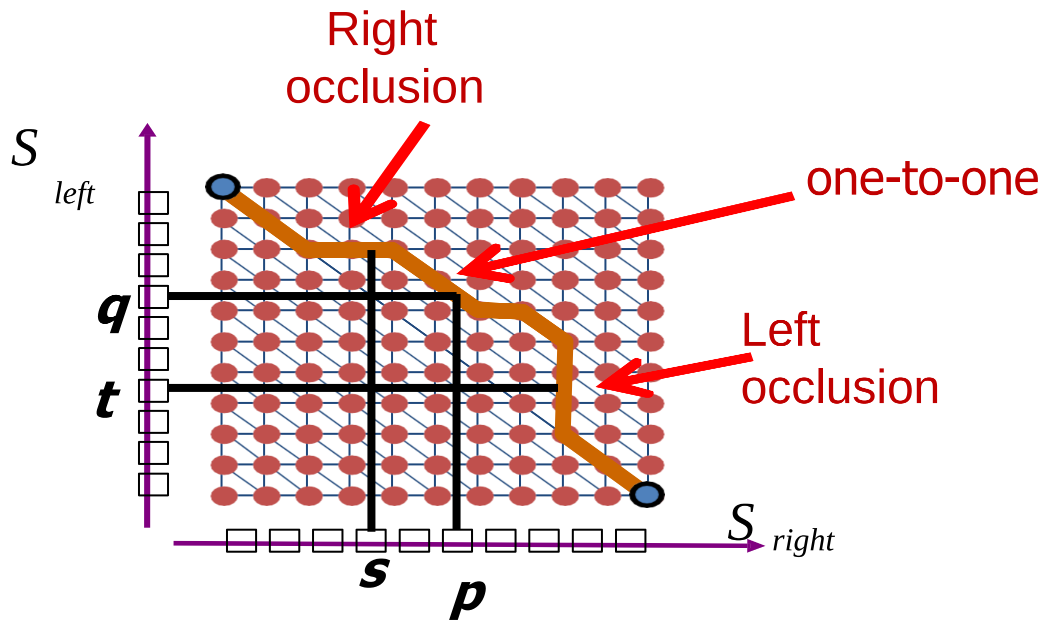


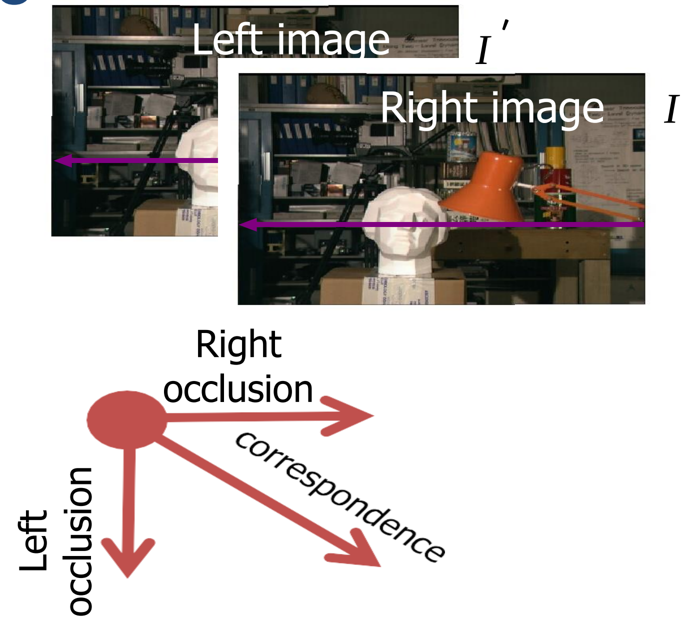


“Shortest paths” for scan-line stereo



“Shortest paths” for scan-line stereo



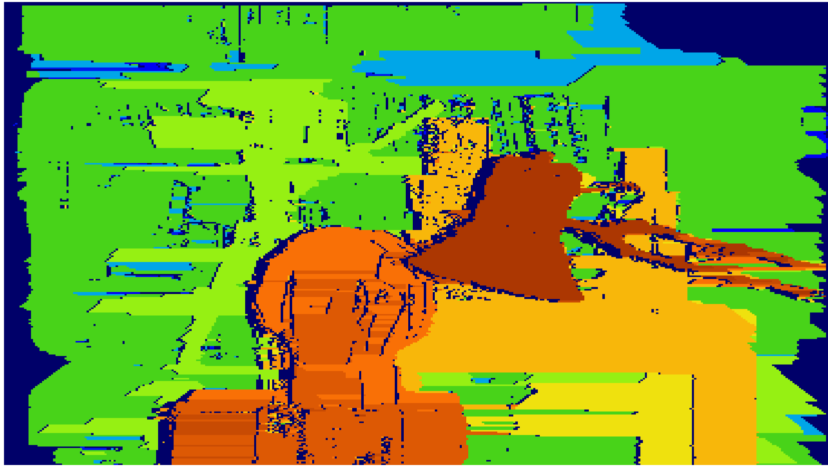


Can be implemented with dynamic programming

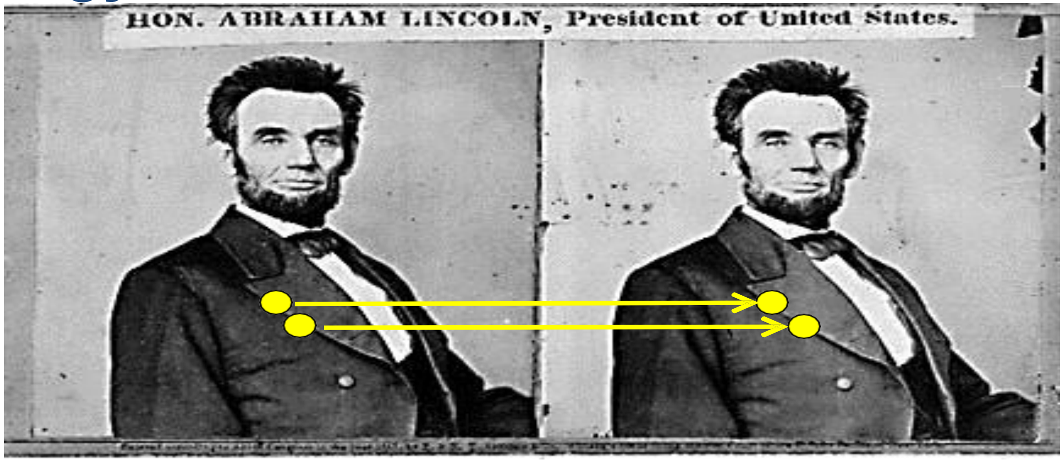
Ohta & Kanade ’85, Cox et al. ’96, Intille & Bobick, ‘01

Coherent stereo on 2D grid

- Scanline stereo generates streaking artifacts
- Can’t use dynamic programming to find spatially coherent disparities/ correspondences on a 2D grid
- E.g. look at Hirschmüller et al.: “Accurate and efficient stereo processing by semi-global matching and **mutual information”**

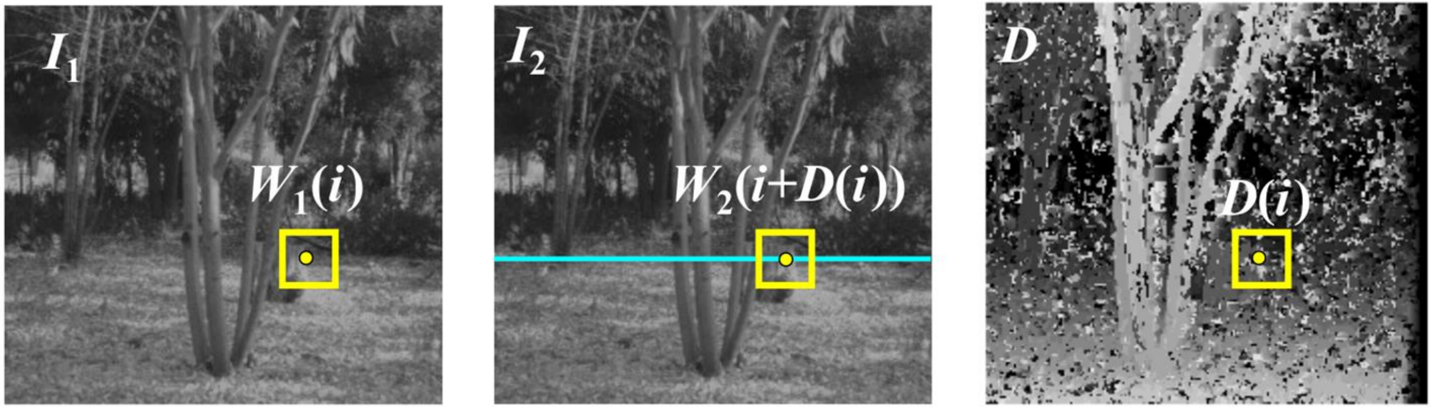


Stereo as energy minimization



- 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

Stereo matching as energy minimization



$$E = \alpha E_{data}(I_1, I_2, D) + \beta E_{smooth}(D)$$

$$E_{data} = \sum_i \left(W_1(i) - W_2(i + D(i))\right)^2$$

$$E_{smooth} = \sum_{\text{neighbors } i,j} \rho\left(D(i) - D(j)\right)$$

Energy functions of this form can be minimized using *graph cuts.*

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

Better results…

Graph cut method

Ground truth

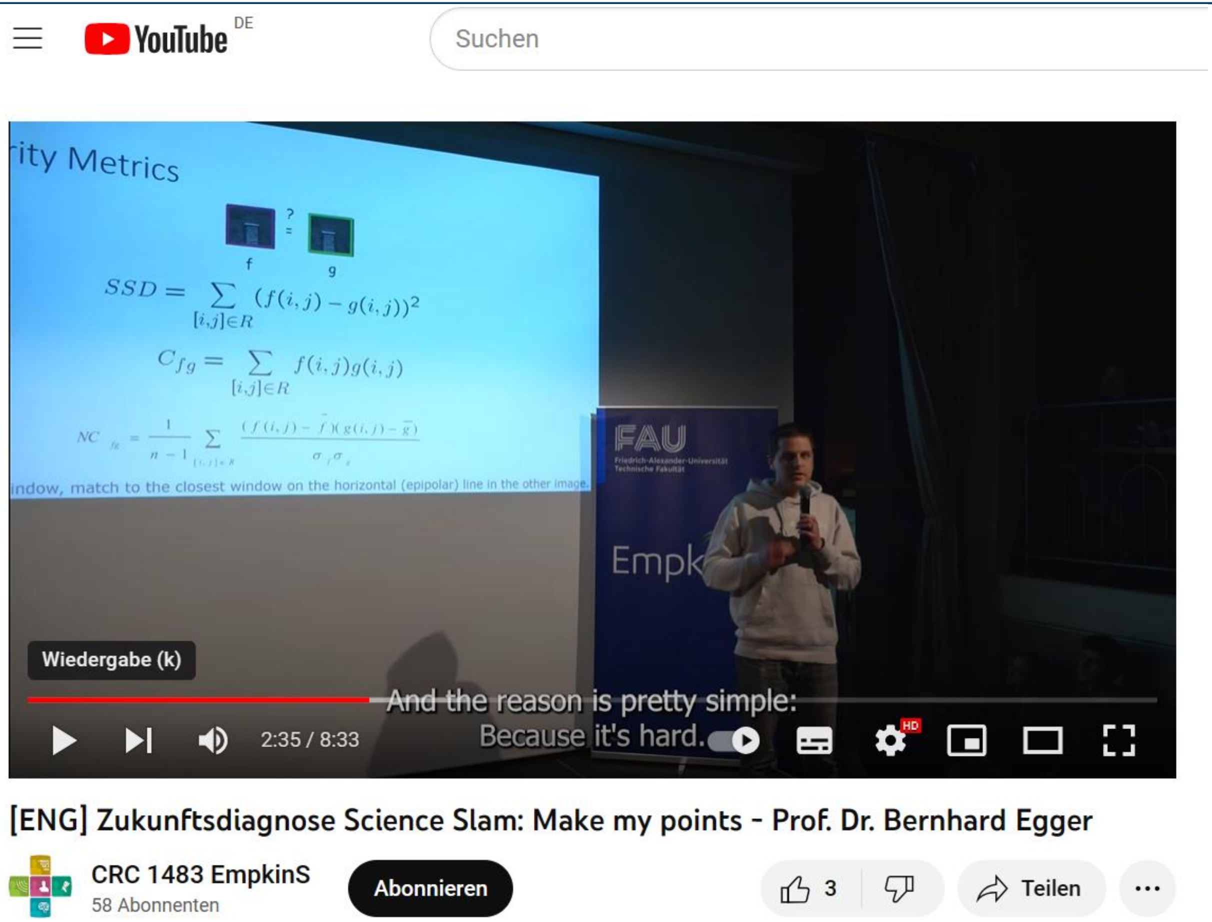
Boykov et al., [Fast Approximate Energy Minimization via Graph Cuts](), International Conference on Computer Vision, September 1999.

Challenges

- Low-contrast ‘textureless’ image regions
- Occlusions
- Violations of brightness constancy
  - Specular reflections
- Really large baselines
  - Foreshortening and appearance change
- Camera calibration errors

Image Sources

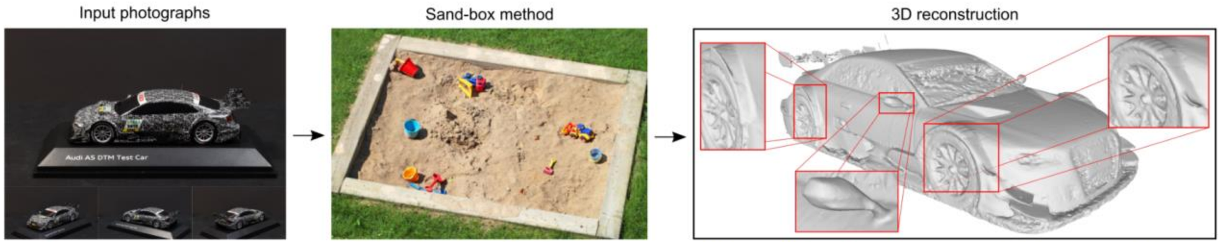
1. The slides on image rectification are courtesy of J. Chai, http://faculty.cs.tamu.edu/jchai/cpsc641_spring10/lectures/lecture9.ppt
2. A number of slides in this presentation have been adapted by the presentation of S. Narasimhan, http://ww.cs.cmu.edu/afs/cs/academic/class/15385-s06/lectures/ppts/lec-14.ppt
3. The Lincoln image is courtesy of S. Seitz.
4. The window-matching slide is courtesy of O. Camps.
5. The example slide on hierarchical correspondence algorithms is courtesy of ETH, http://www.inf.ethz.ch/personal/pomarc/courses/gcv/class07.ppt


YouTube DE
Suchen
ity Metrics
f
g
SSD = ∑[i,j]∈R (f(i,j) − g(i,j))²
C_fg = ∑[i,j]∈R f(i,j)g(i,j)
indow, match to the closest window on the horizontal (epipolar) line in the other image.
FAU
Empk
Wiedergabe (k)
And the reason is pretty simple:
Because it's hard.
2:35 / 8:33
[ENG] Zukunftsdiagnose Science Slam: Make my points - Prof. Dr. Bernhard Egger
CRC 1483 EmpkinS
58 Abonnenten
Abonnieren
3
Teilen

A Free Computer Vision Lesson for Car Manufacturers
or
It is Time to Retire the Erlkönig

Maximilian Weiherer Bernhard Egger
Fantastic-Amazing-University Erlangen-Nürnberg (FAU)

maximilian.weiherer@fau.de
bernhard.egger@fau.de



Oh no – we can recover very detailed 3D shape from Erlkönig photographs. How could this have happened? And even worse, the reconstruction is best in the parts where there is a pattern. Well, we guess someone didn't know about the basics of computer vision.

Abstract

In award winning prior work [1], we identified the inability of autonomous cars to honk as the key reason that they are not broadly deployed on our streets. In this work [2], however, we suggest that the core reason is the lack of most basic computer vision knowledge of car manufacturers. To hide their most fancy new cars they put a special camouflage pattern on their so called *Erlkönig* prototypes.