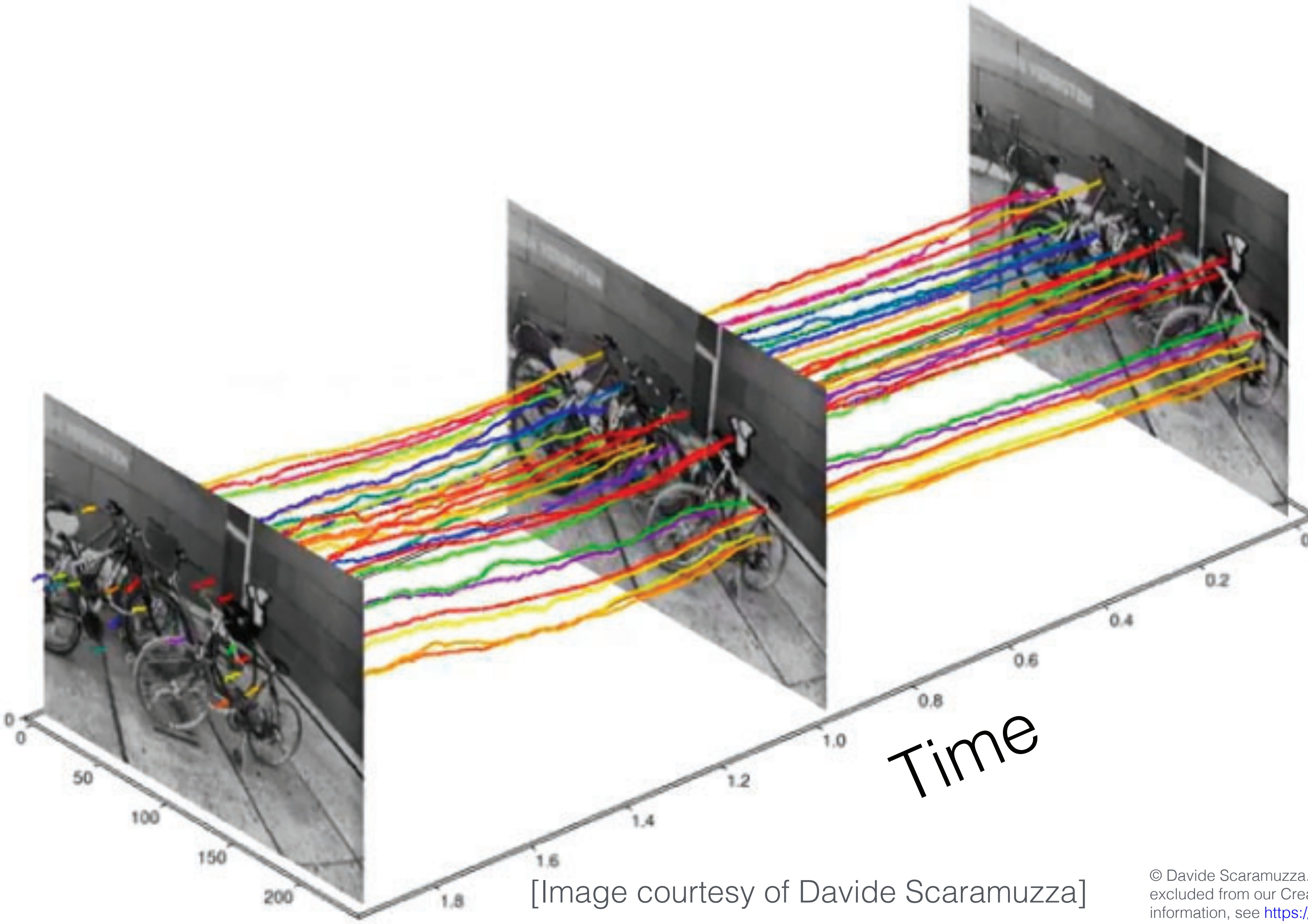
16.485: VNAV - Visual Navigation for Autonomous Vehicles



[Image courtesy of Davide Scaramuzza]

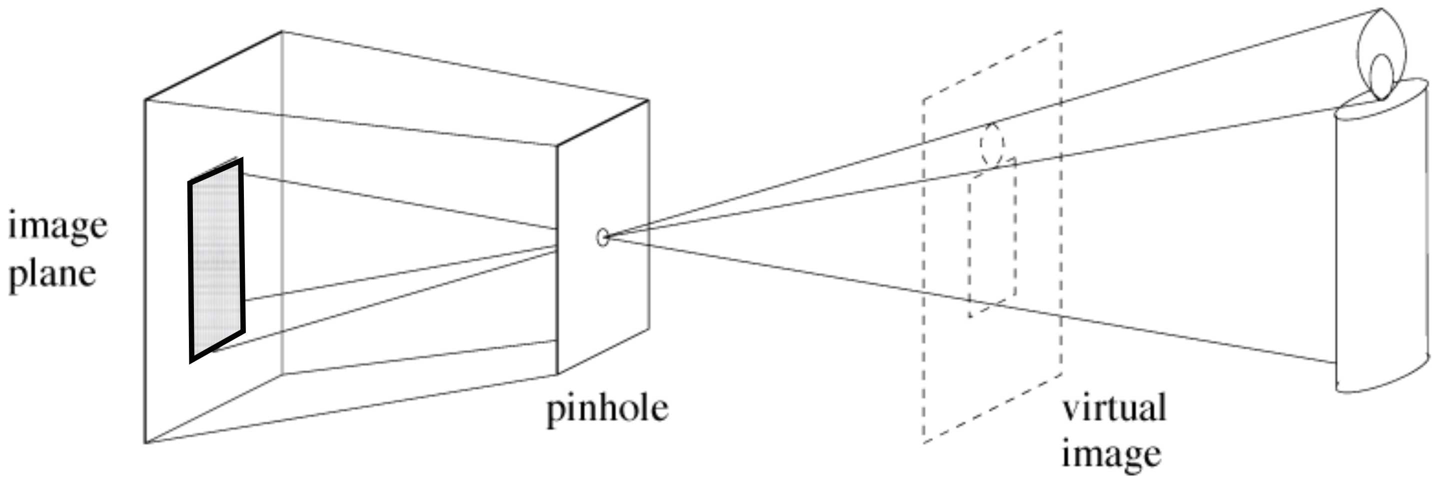
© Davide Scaramuzza. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

Luca Carlone

Lecture 12-13: Feature Detection and Tracking

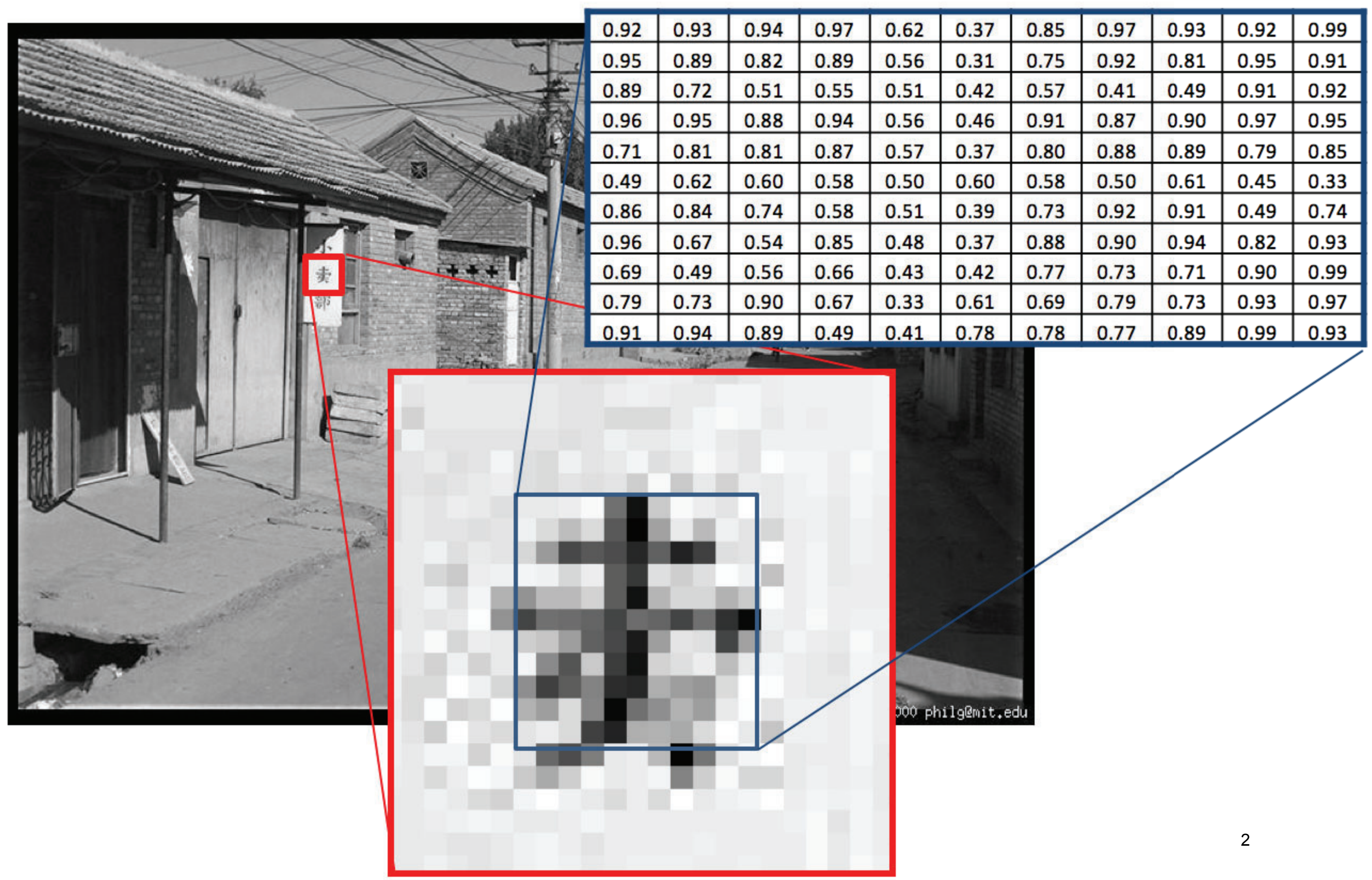
Part of the following slides are inspired and built on the lecture slides of Professor Frank Dellaert's course.

Digital Photography



2D array of
"light sensors"

- CCD (charge-coupled device, 1960)
- CMOS (complementary metal-oxide semiconductor, 1963)



0.92	0.93	0.94	0.97	0.62	0.37	0.85	0.97	0.93	0.92	0.99
0.95	0.89	0.82	0.89	0.56	0.31	0.75	0.92	0.81	0.95	0.91
0.89	0.72	0.51	0.55	0.51	0.42	0.57	0.41	0.49	0.91	0.92
0.96	0.95	0.88	0.94	0.56	0.46	0.91	0.87	0.90	0.97	0.95
0.71	0.81	0.81	0.87	0.57	0.37	0.80	0.88	0.89	0.79	0.85
0.49	0.62	0.60	0.58	0.50	0.60	0.58	0.50	0.61	0.45	0.33
0.86	0.84	0.74	0.58	0.51	0.39	0.73	0.92	0.91	0.49	0.74
0.96	0.67	0.54	0.85	0.48	0.37	0.88	0.90	0.94	0.82	0.93
0.69	0.49	0.56	0.66	0.43	0.42	0.77	0.73	0.71	0.90	0.99
0.79	0.73	0.90	0.67	0.33	0.61	0.69	0.79	0.73	0.93	0.97
0.91	0.94	0.89	0.49	0.41	0.78	0.78	0.77	0.89	0.99	0.93

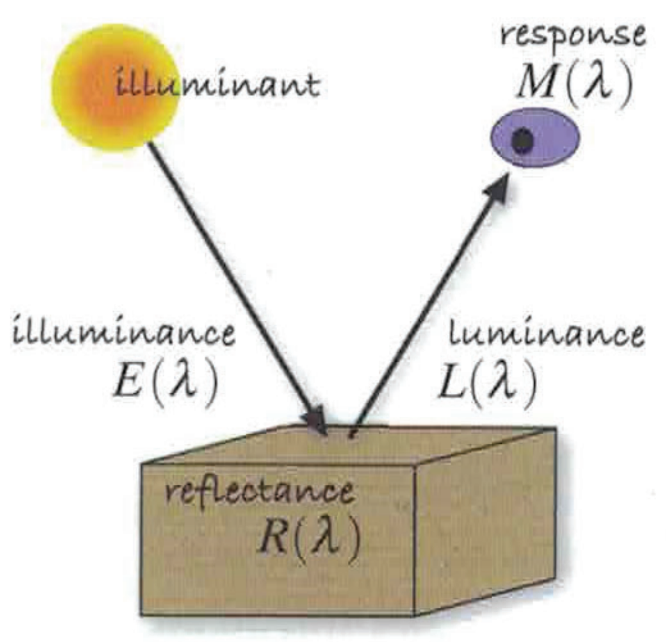
Appearance: Light and Colors

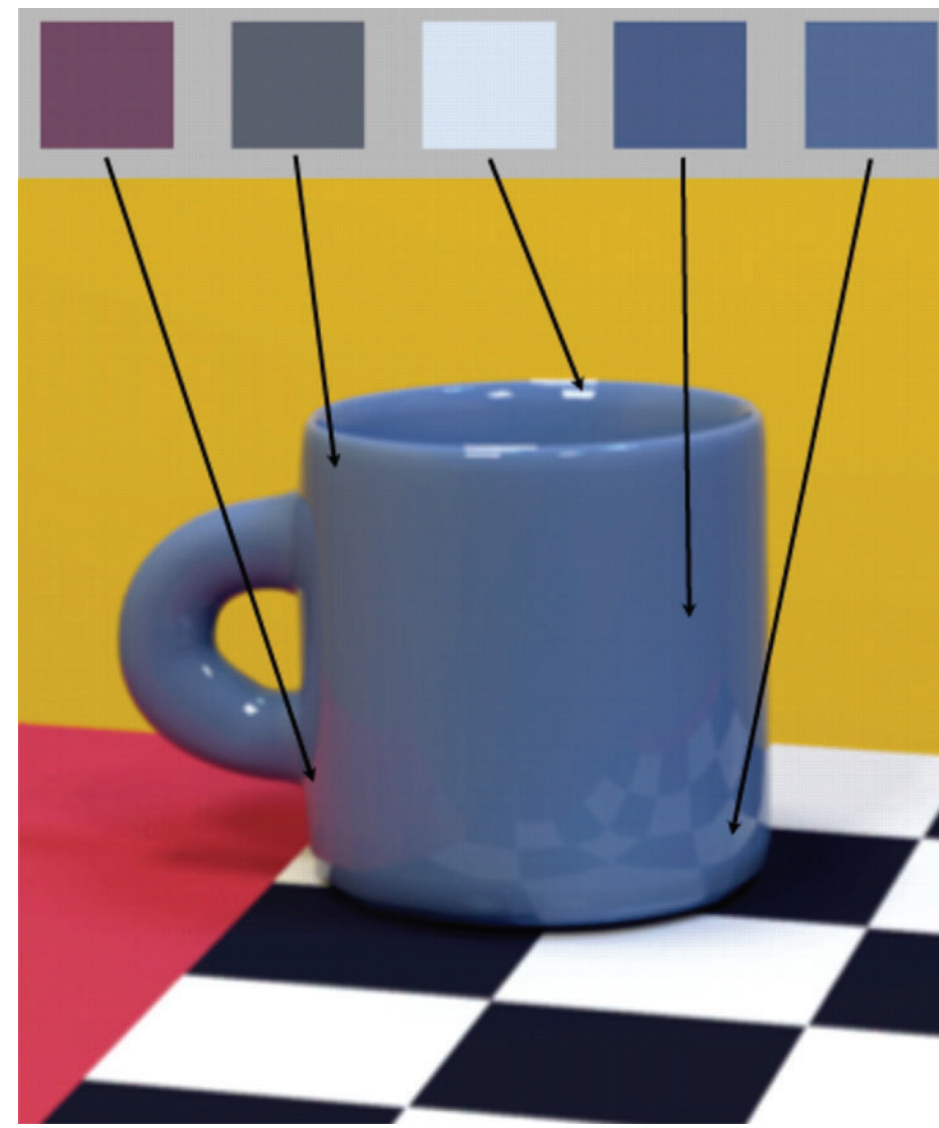
R
(G=0,B=0)

G
(R=0,B=0)

B
(R=0,G=0)

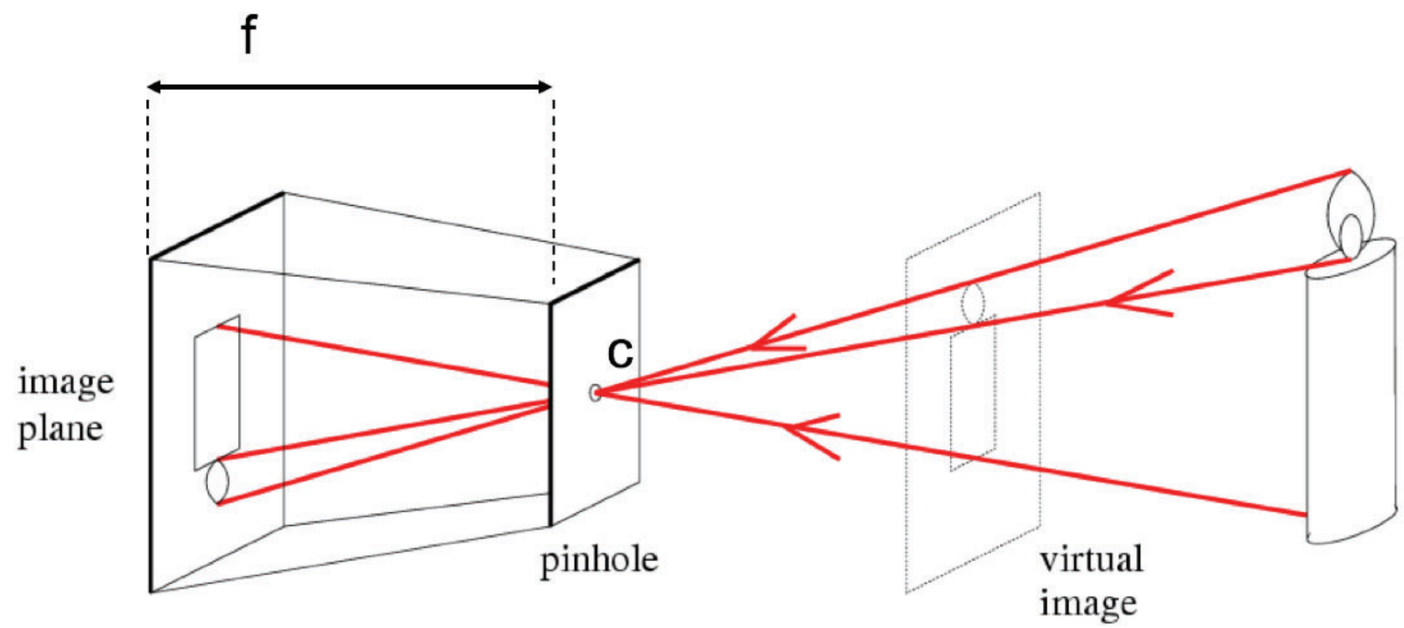
Perceived appearance is the result of (i) geometry, (ii) illumination, (iii) material properties



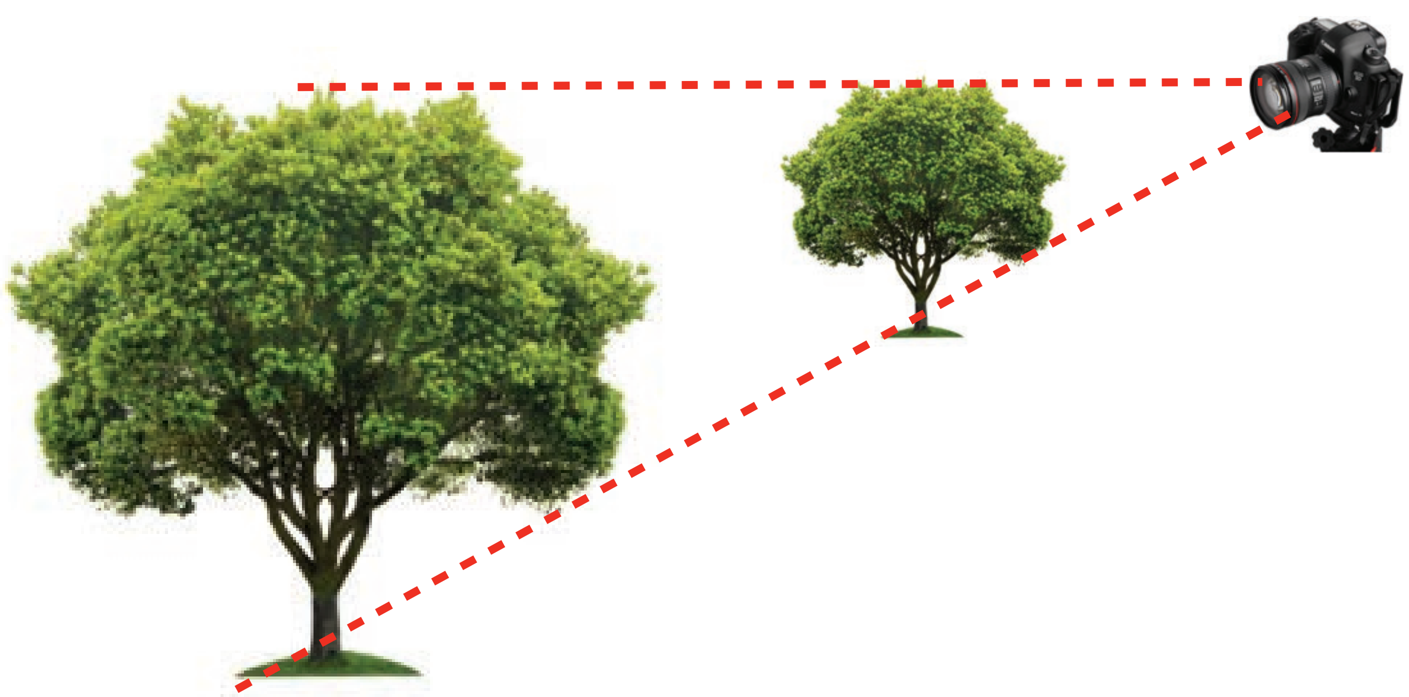


Perspective Projection Recap

- what is lost?
    - depth?

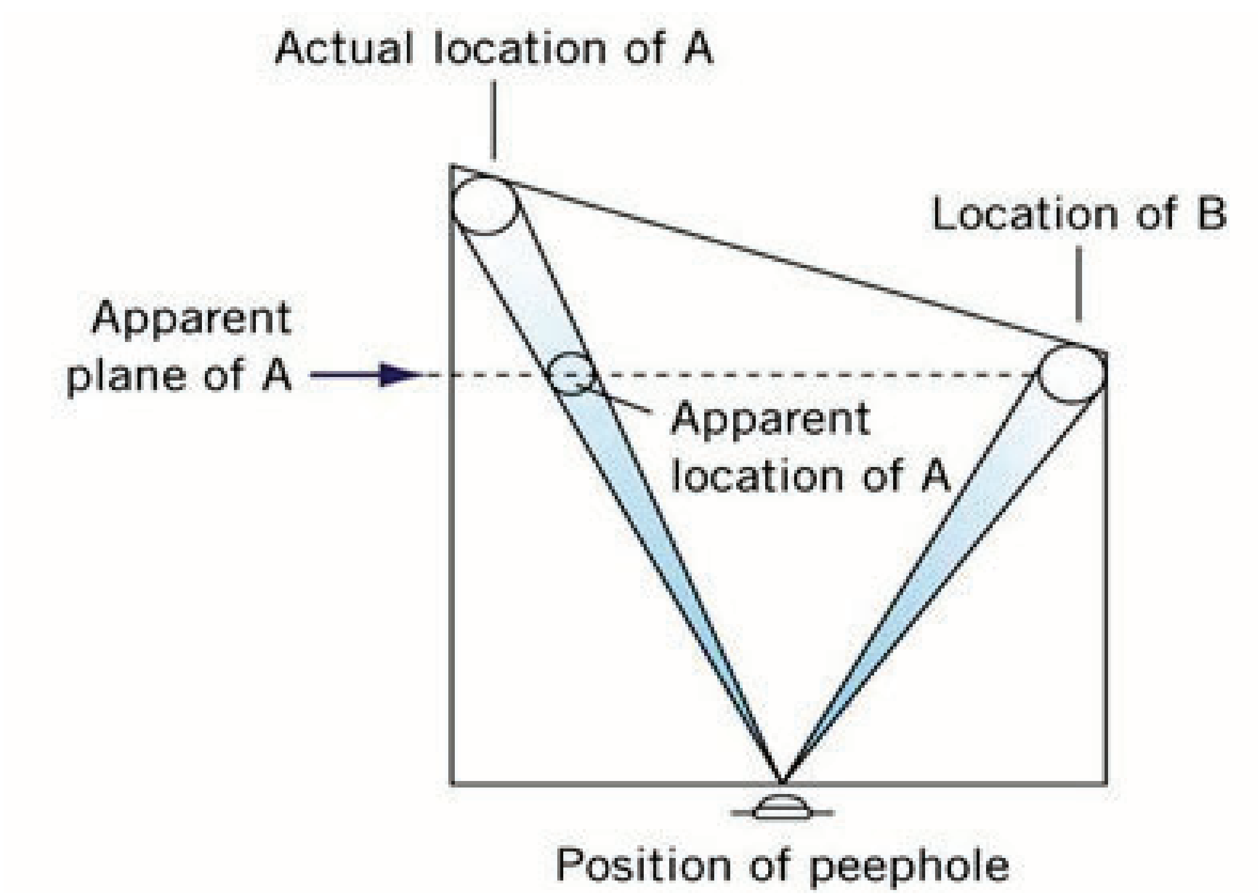


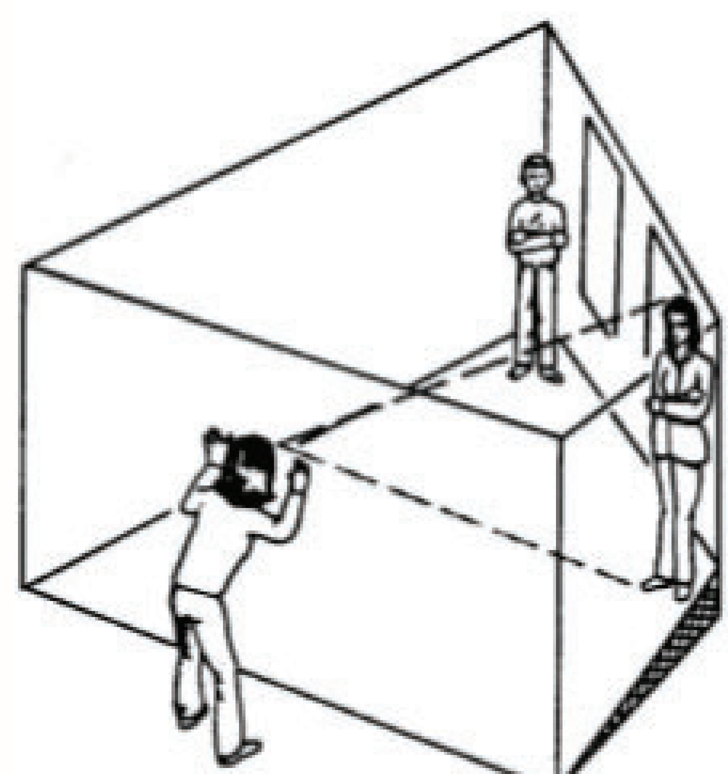
f = focal length
c = center of the camera



Ames Room

© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

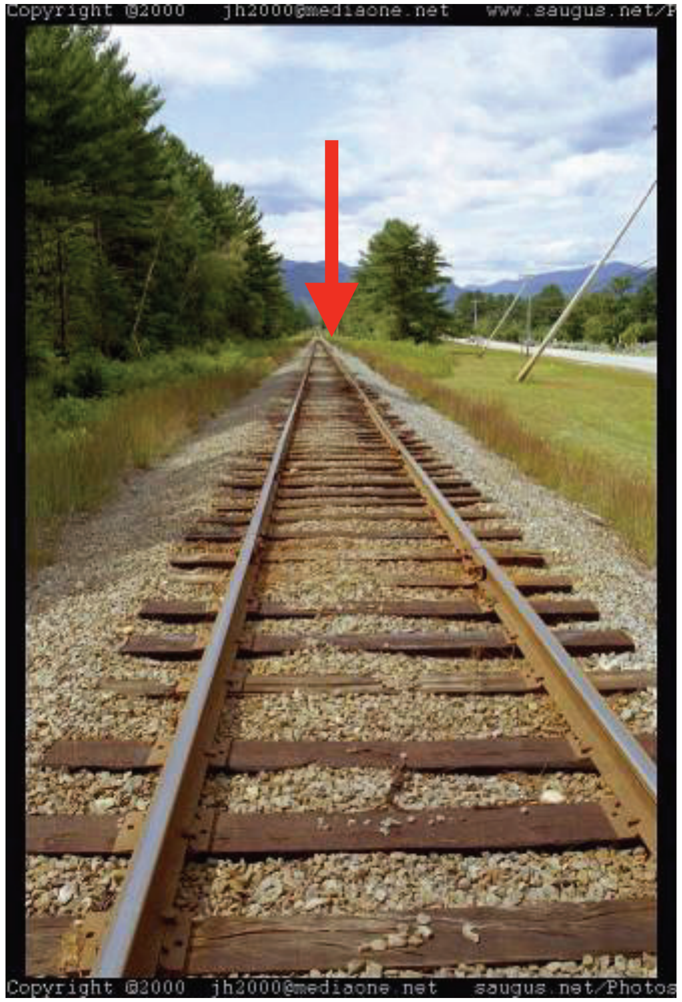


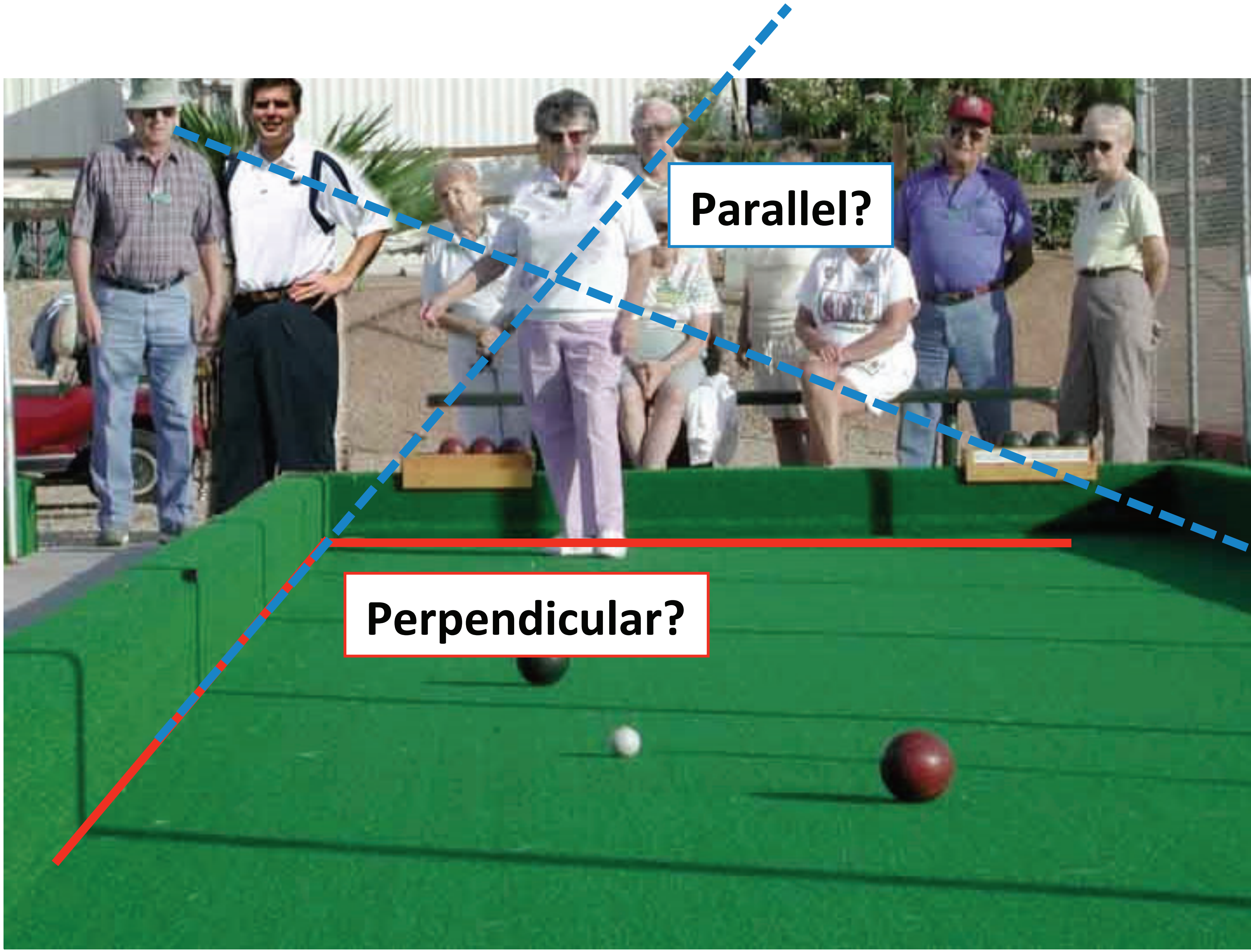


Ames, 1946

Perspective Projection Recap

- what is lost?
    - depth?
    - length?
    - angles?





© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

Parallel lines which intersect …

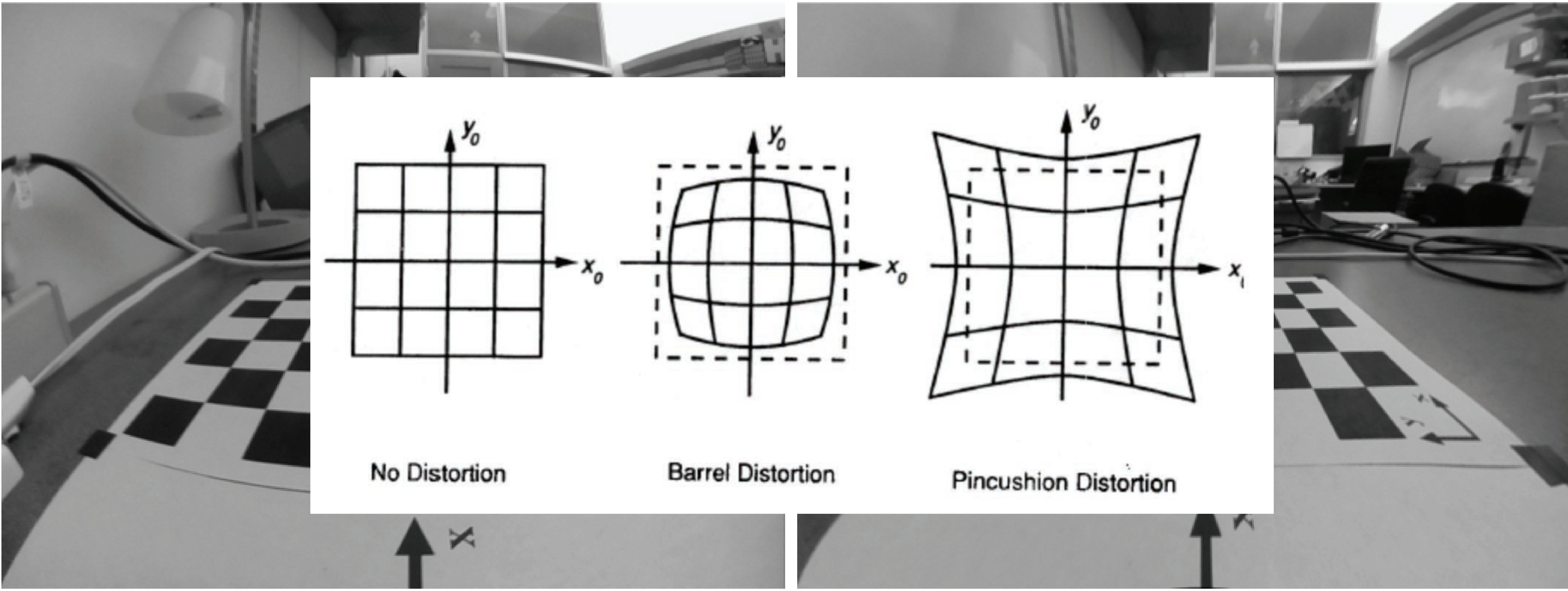
Perspective Projection Recap

- what is preserved?
  - straight lines remain straight

© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

The final Touch: Adding a Lens

- Pinhole model is based on the geometry of the **camera obscura**
- In practice: add a **lens** in front of the aperture to capture more light
- Pinhole model holds, but **distortion** may appear due lens imperfections

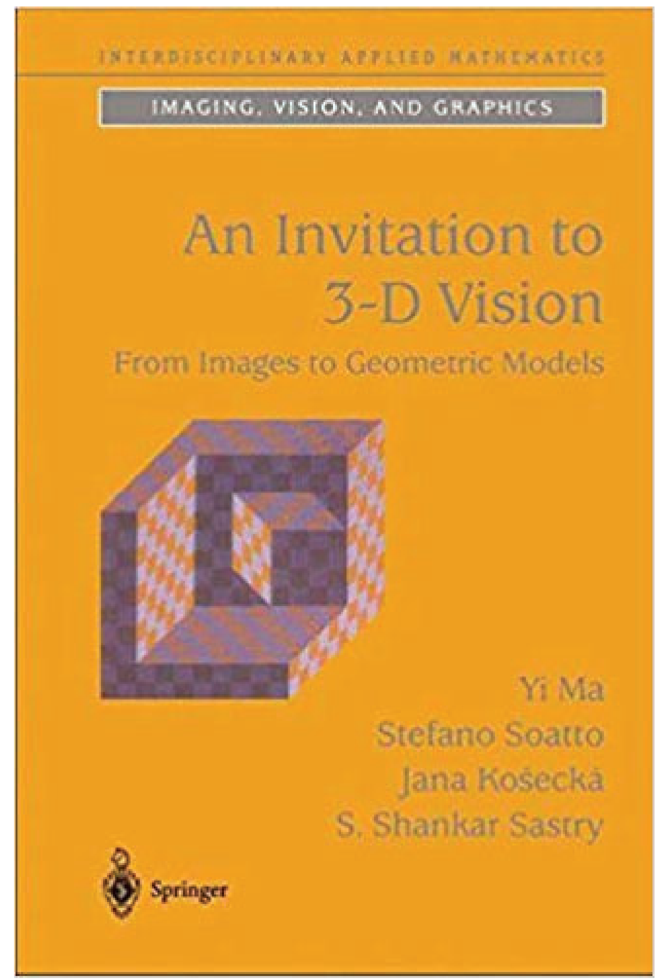


© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

- distortion can be described mathematically using **distortion parameters**
  - can be estimated during calibration and compensated for (**undistortion**)

Today

- Feature Detection
- Feature Tracking
- Feature Matching



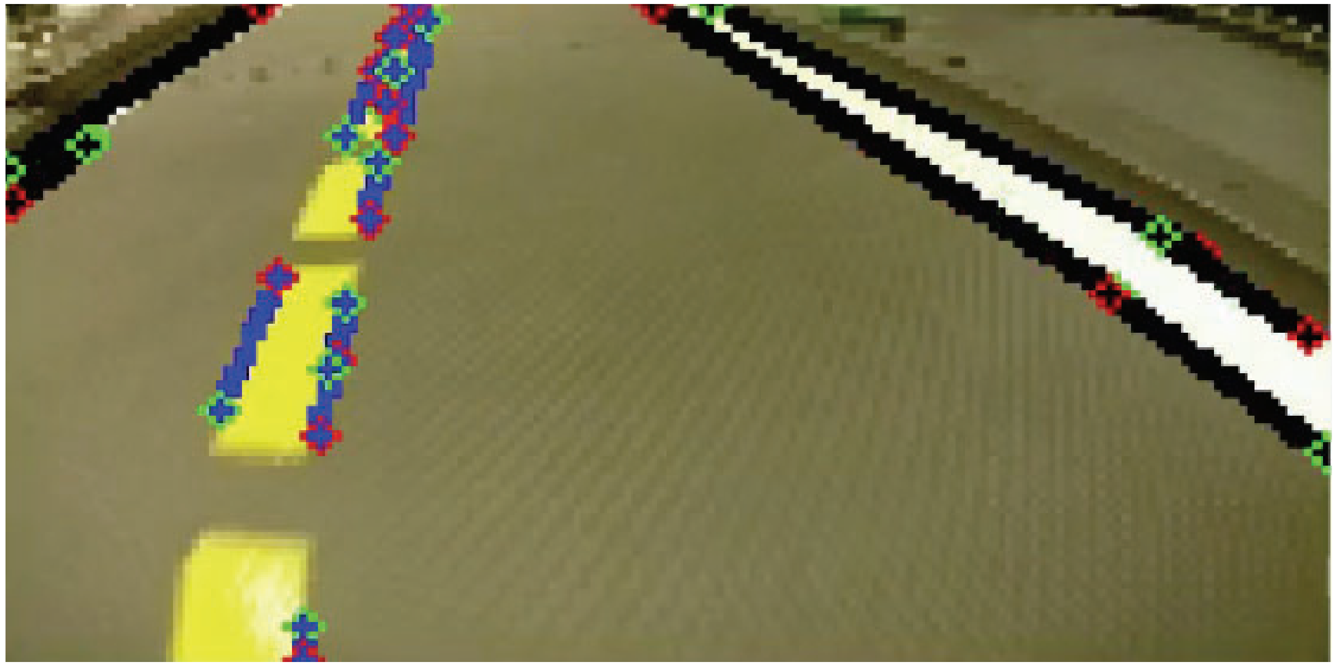
Chapter 4

Image Primitives and Correspondence

Feature detection

What is a feature?

- a *recognizable* structure in the environment
  - lines, corners
  - geometric primitives (e.g., circles)
  - objects (high-level features)
  - …

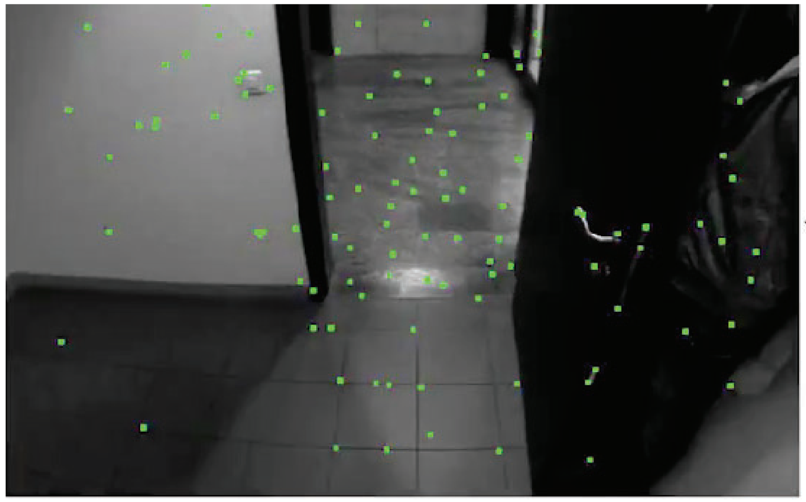


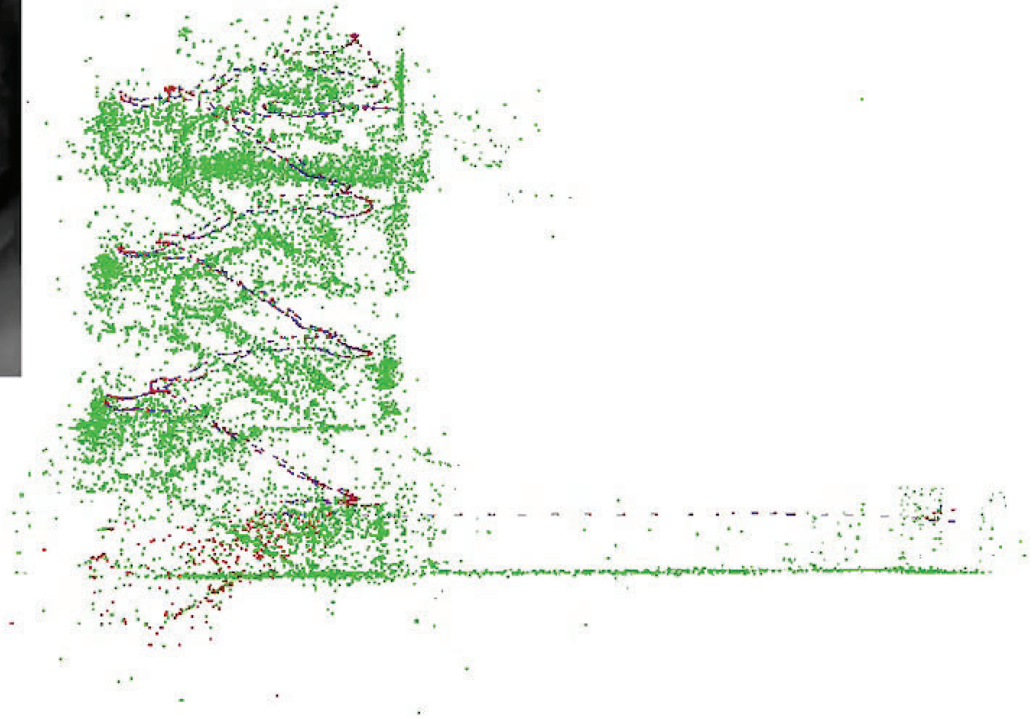
Why extracting features?

- data compression
  - # of pixels in a modern camera: 4416 x 1242 ~ 5M
  - # of parameters to describe a line: 2 (4 for a segment)
- easier to describe mathematically: points, lines, …

Corner Detection

- **Why do we care**?
  - Motion tracking
  - 3D reconstruction
  - Object recognition
  - …





Corner Detection

- **corners:** also known as interest points, keypoints, or point features
  - easily identifiable points in the image
  - or: if given a corner in image I_1, we can easily find corresponding pixel in I_2 (both images are picturing the same scene from different viewpoints)

© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

Corner Detection

- **corners:** also known as interest points, keypoints, or point features
  - easily identifiable points in the image
  - or: if given a corner in image $\boldsymbol{I_1}$, we can easily find corresponding pixel in $\boldsymbol{I_2}$ (both images are picturing the same scene from different viewpoints)

© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

Corner Detection

- **corners:** also known as interest points, keypoints, or point features
  - easily identifiable points in the image
  - or: if given a corner in image I_1, we can easily find corresponding pixel in I_2 (both images are picturing the same scene from different viewpoints)

© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

Corner Detection

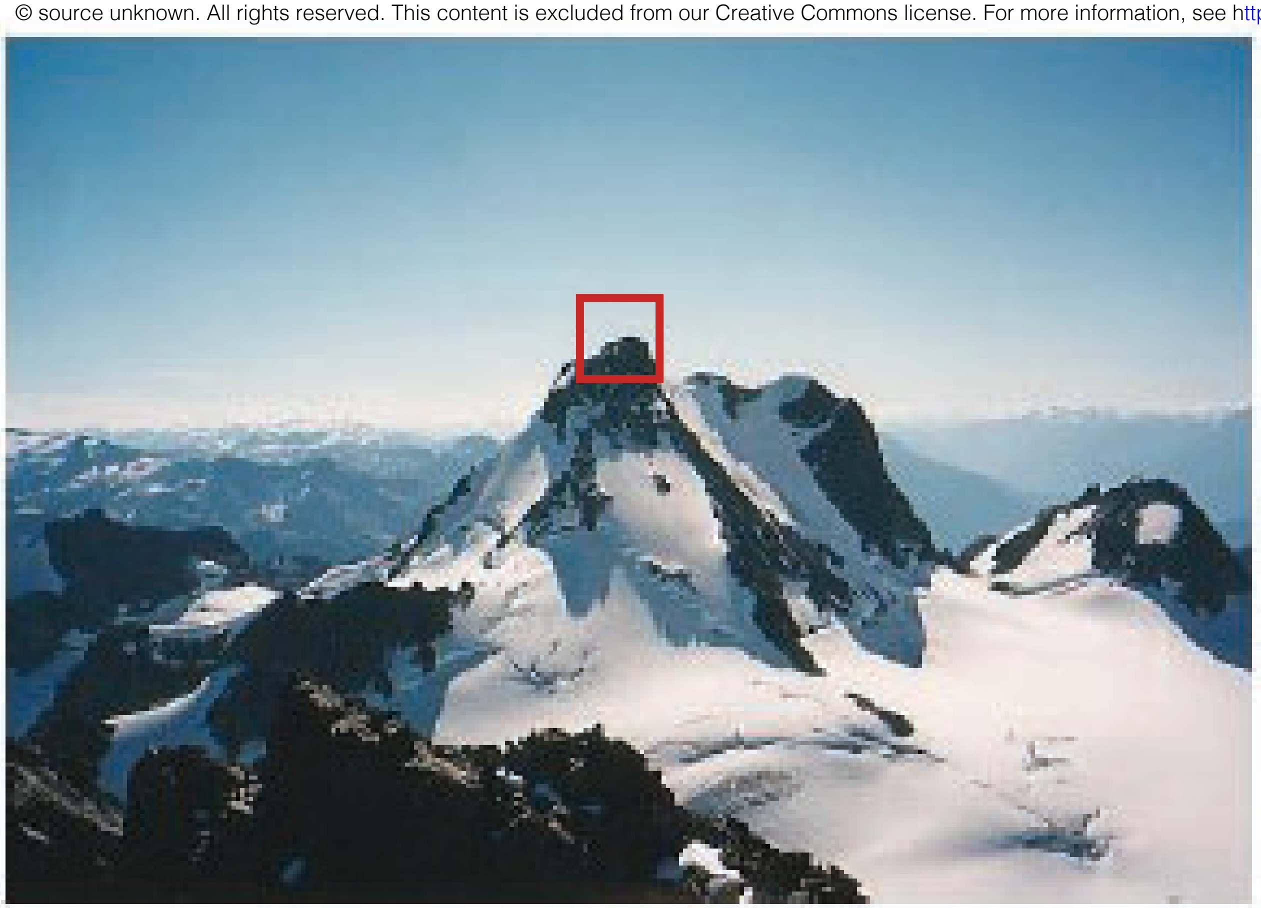
- **corners:** also known as interest points, keypoints, or point features
  - easily identifiable points in the image
  - or: if given a corner in image I_1, we can easily find corresponding pixel in I_2 (both images are picturing the same scene from different viewpoints)

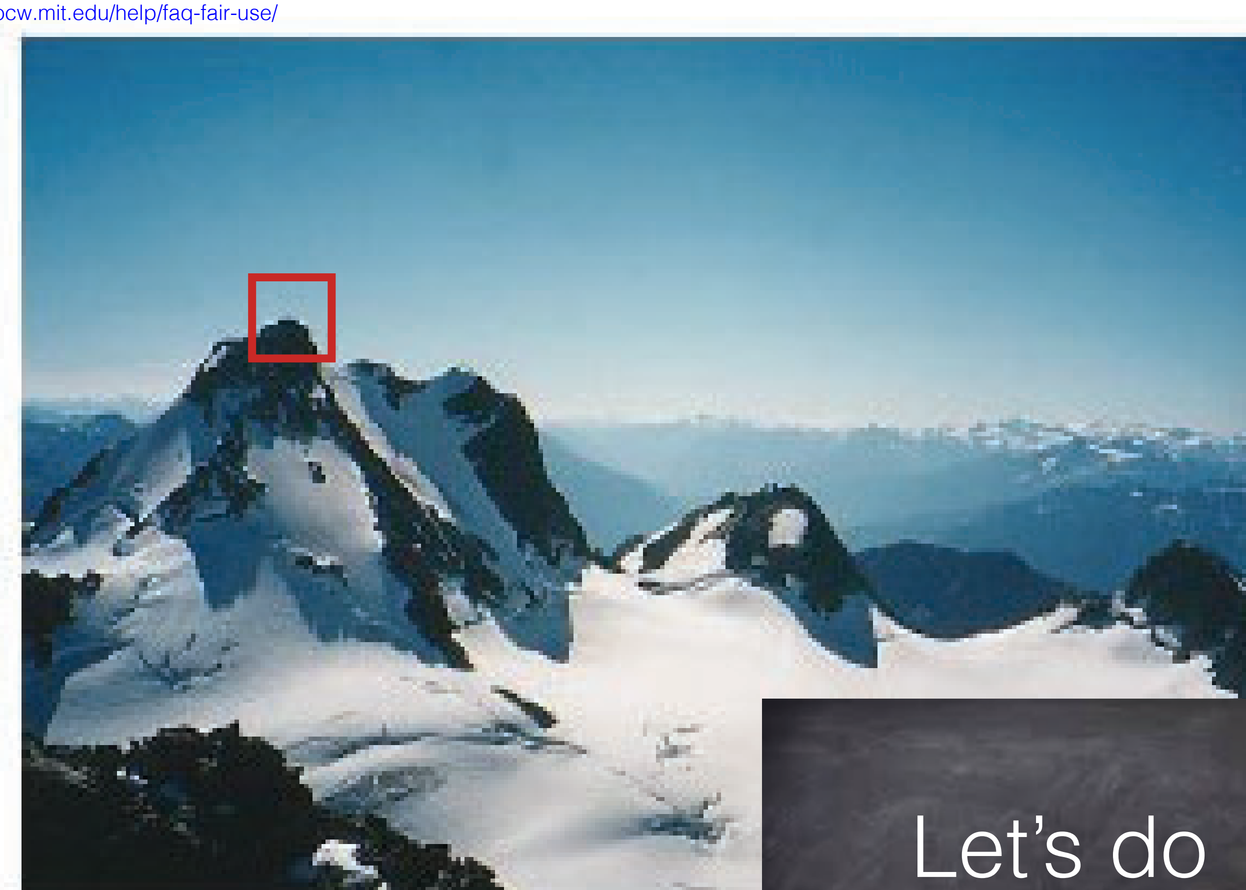
© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

Corner Detection

- **corners:** also known as interest points, keypoints, or point features
  - easily identifiable points in the image
  - or: if given a corner in image $\boldsymbol{I_1}$, we can easily find corresponding pixel in $\boldsymbol{I_2}$ (both images are picturing the same scene from different viewpoints)

© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/



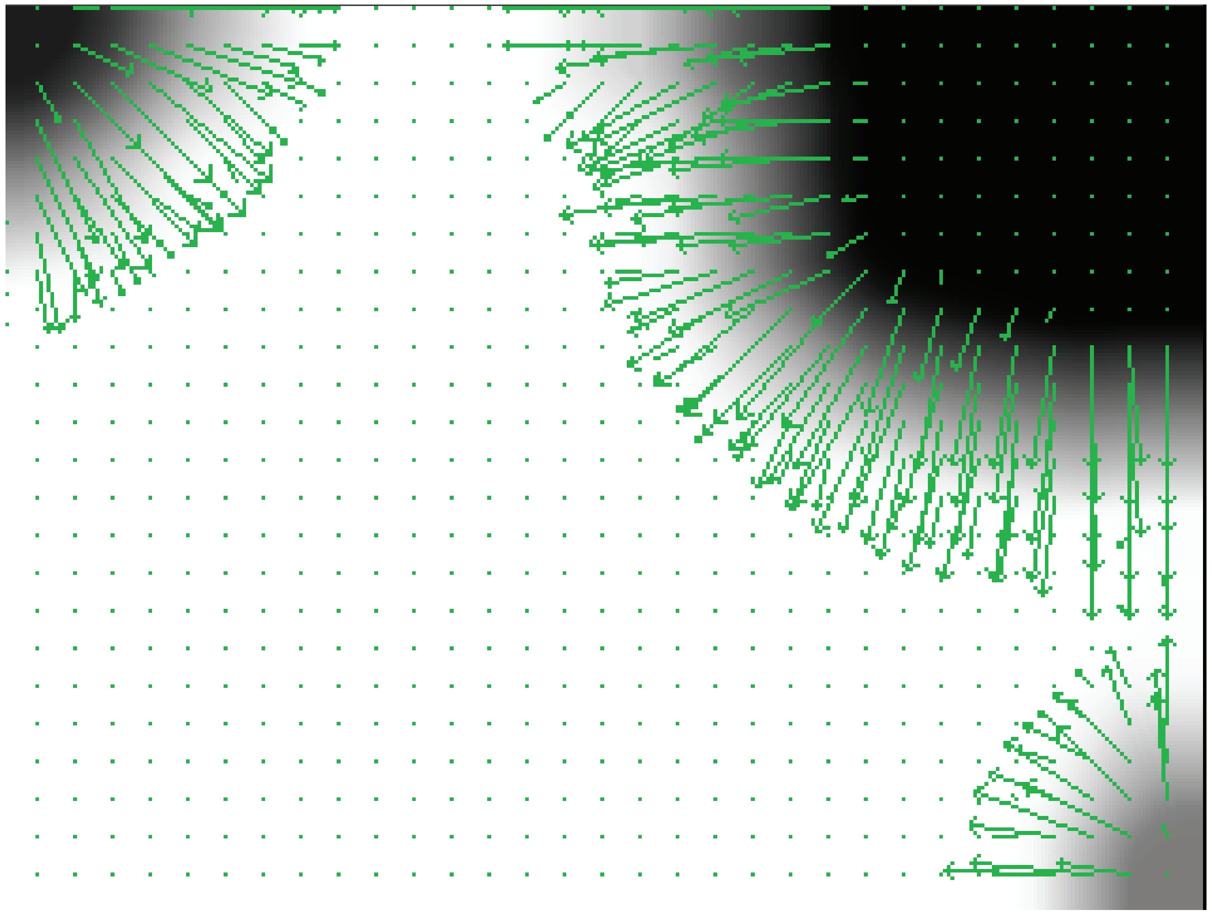


Let's do some math

Image Gradients

$$\nabla \mathcal{I}(\boldsymbol{x}) = \nabla \mathcal{I}(u, v) = \begin{bmatrix} \frac{\partial \mathcal{I}(u,v)}{\partial u} \\ \frac{\partial \mathcal{I}(u,v)}{\partial v} \end{bmatrix}$$

$$\nabla \mathcal{I}(\boldsymbol{x}) = \nabla \mathcal{I}(u, v) \approx \begin{bmatrix} \frac{\mathcal{I}(u+h,v)-\mathcal{I}(u,v)}{h} \\ \frac{\mathcal{I}(u,v+h)-\mathcal{I}(u,v)}{h} \end{bmatrix}$$

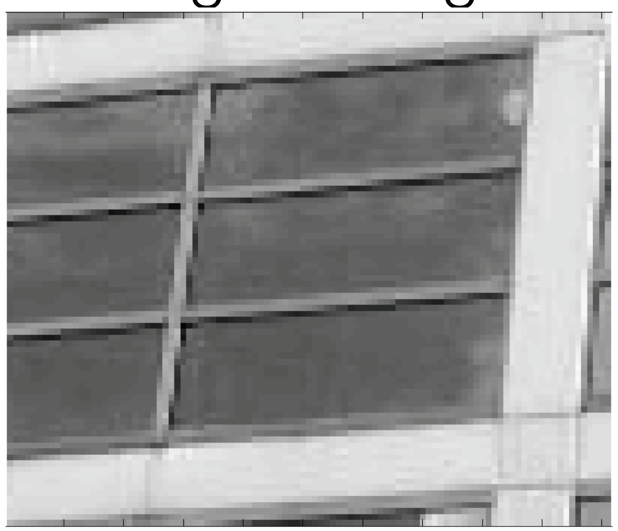


From gradients to finite differences

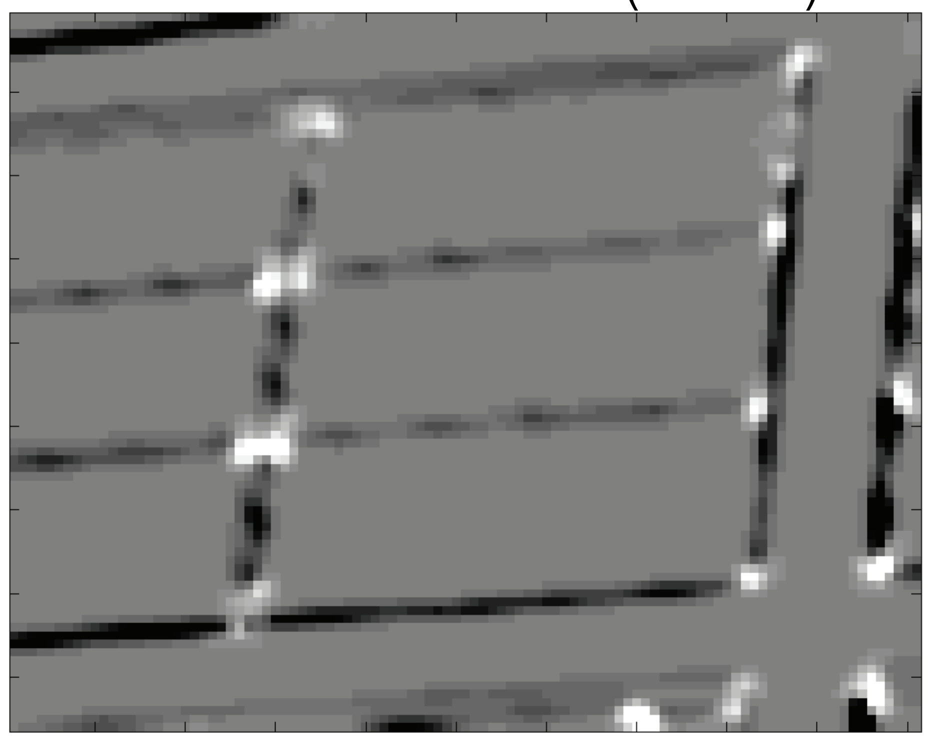
Corner Detection

- **we can compute a "cornerness score" at each pixel in the image**
- peaks are the most distinguishable corners

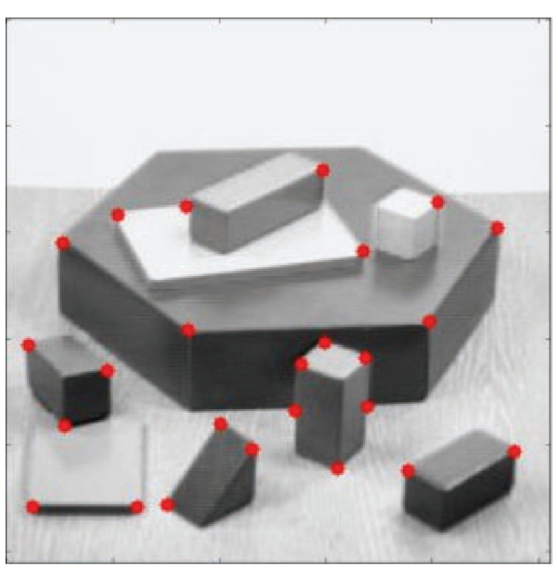
original image

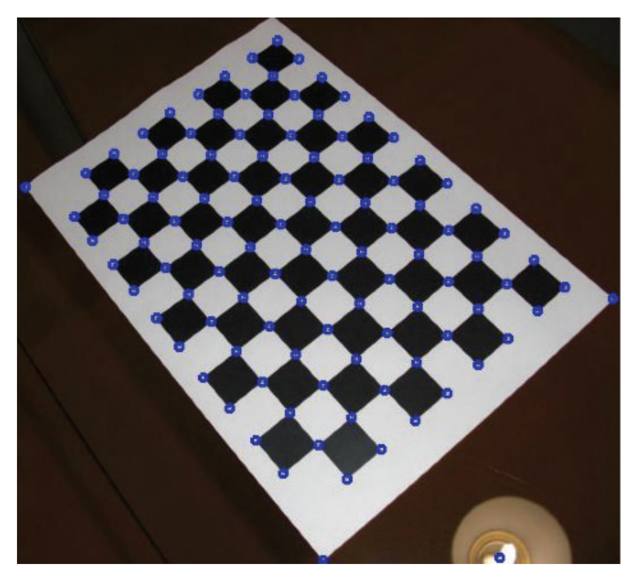


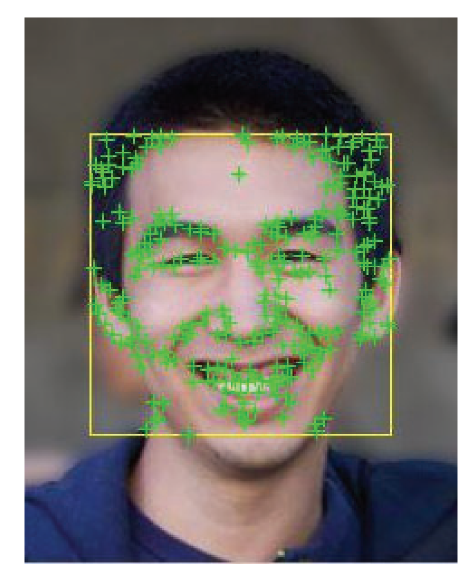
cornerness score (Harris)



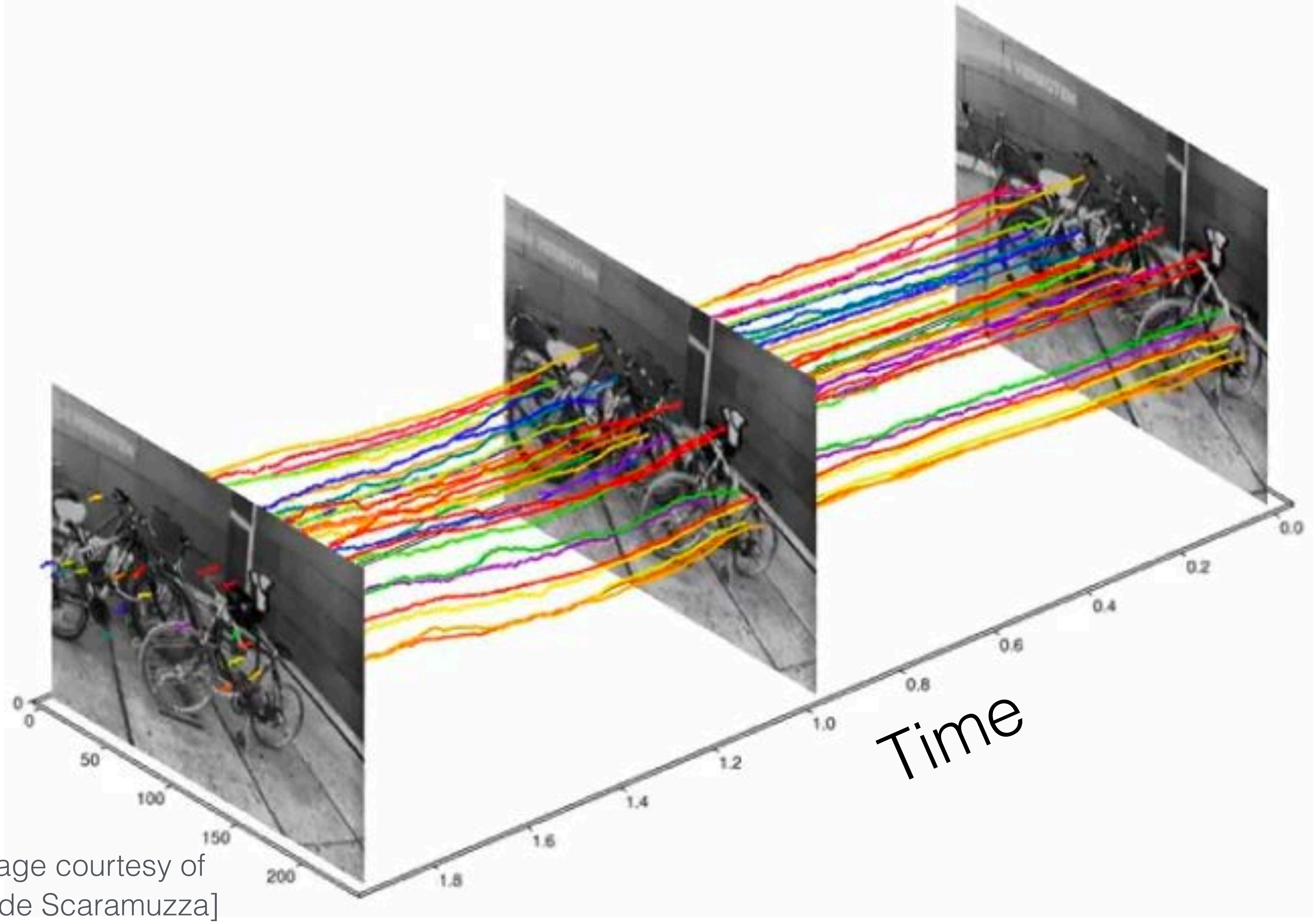
peaks







16.485: VNAV - Visual Navigation for Autonomous Vehicles



© Davide Scaramuzza. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

[Image courtesy of Davide Scaramuzza]

Luca Carlone

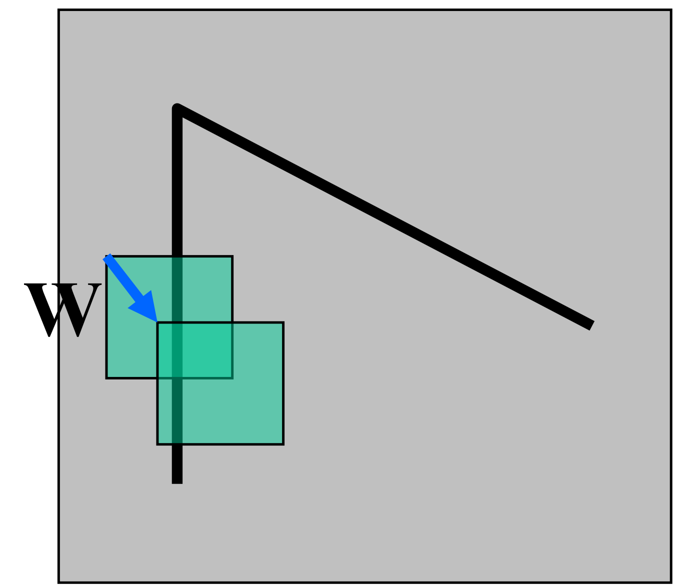
Corner Detection

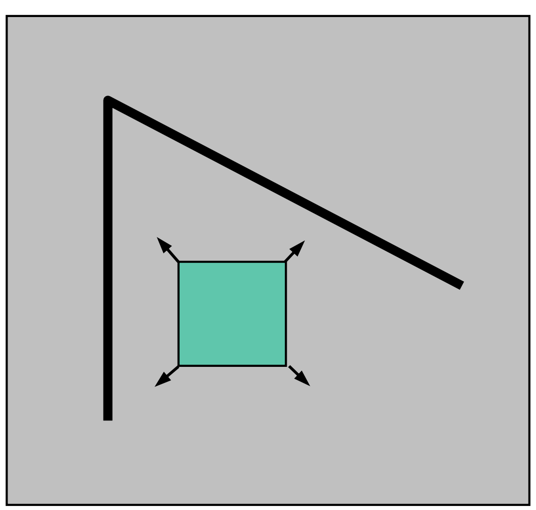
$$\bar{\boldsymbol{x}} = \begin{bmatrix} u \\ v \end{bmatrix}$$

$$\boldsymbol{G} = \sum_{\boldsymbol{x} \in W(\bar{\boldsymbol{x}})} \nabla \mathcal{I}(\boldsymbol{x}) \nabla \mathcal{I}(\boldsymbol{x})^{\mathsf{T}}$$

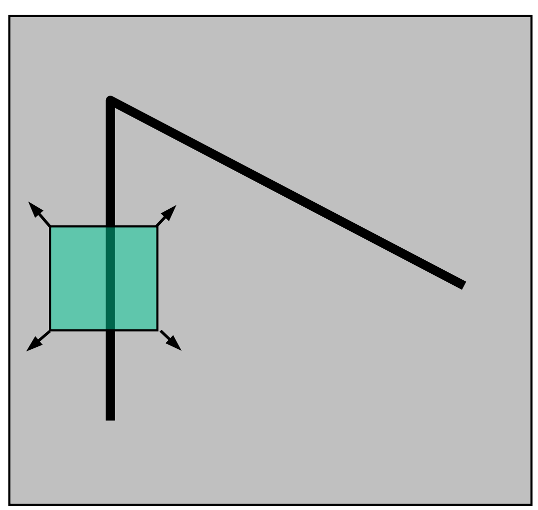
- **finding corners in images:**
  - Consider shifting window **W** by $\boldsymbol{\delta}$
    - How do the pixels in **W** change?
    - compare the windows using **sum of squared differences** (SSD) error:

$$\sum_{\boldsymbol{x} \in W(\bar{\boldsymbol{x}})} \|\mathcal{I}(\boldsymbol{x} + \boldsymbol{\delta}) - \mathcal{I}(\boldsymbol{x})\|^2$$

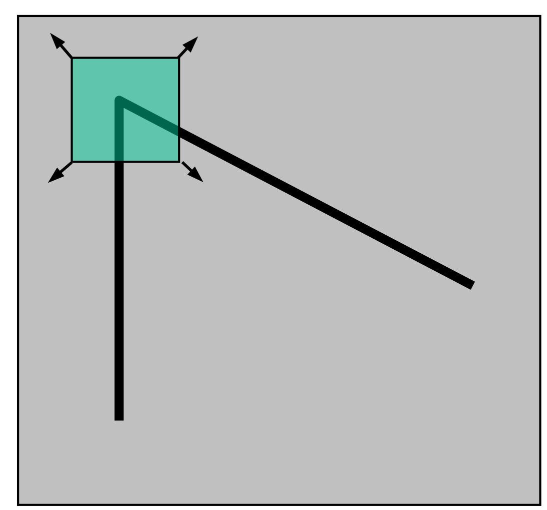




"flat" region:
no change in all directions



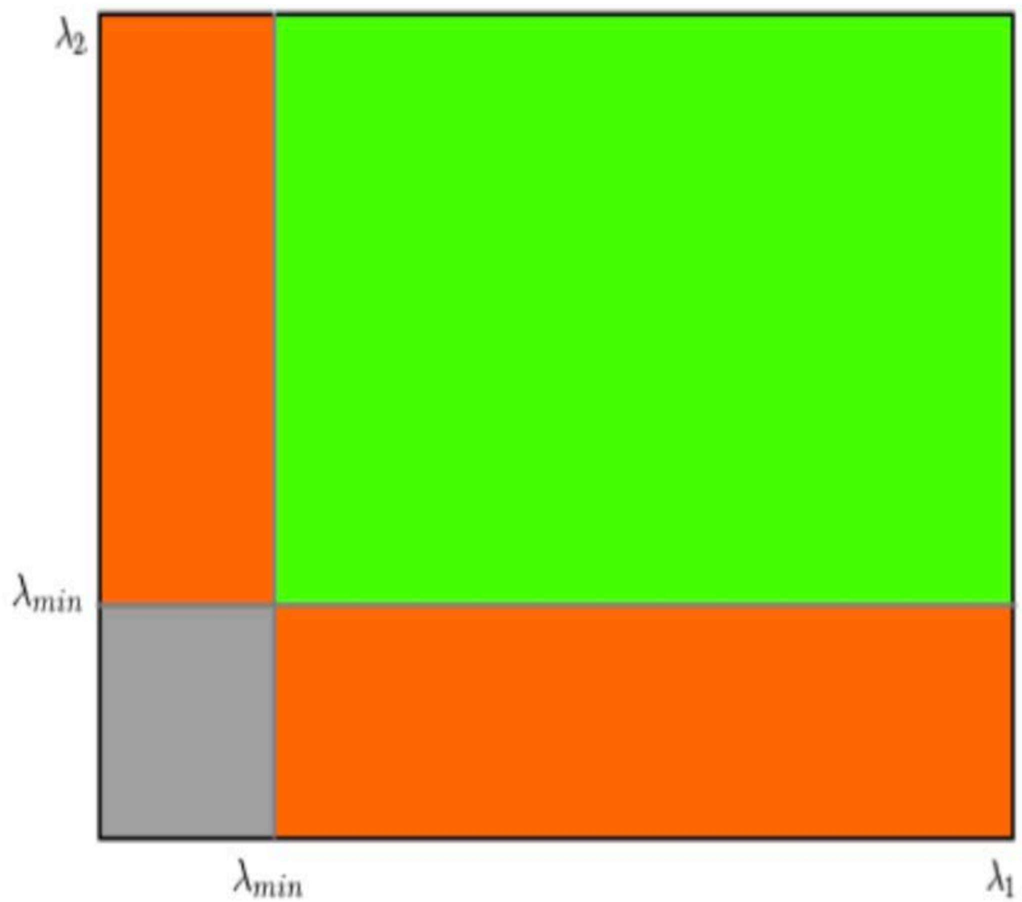
"edge":
no change along the edge direction

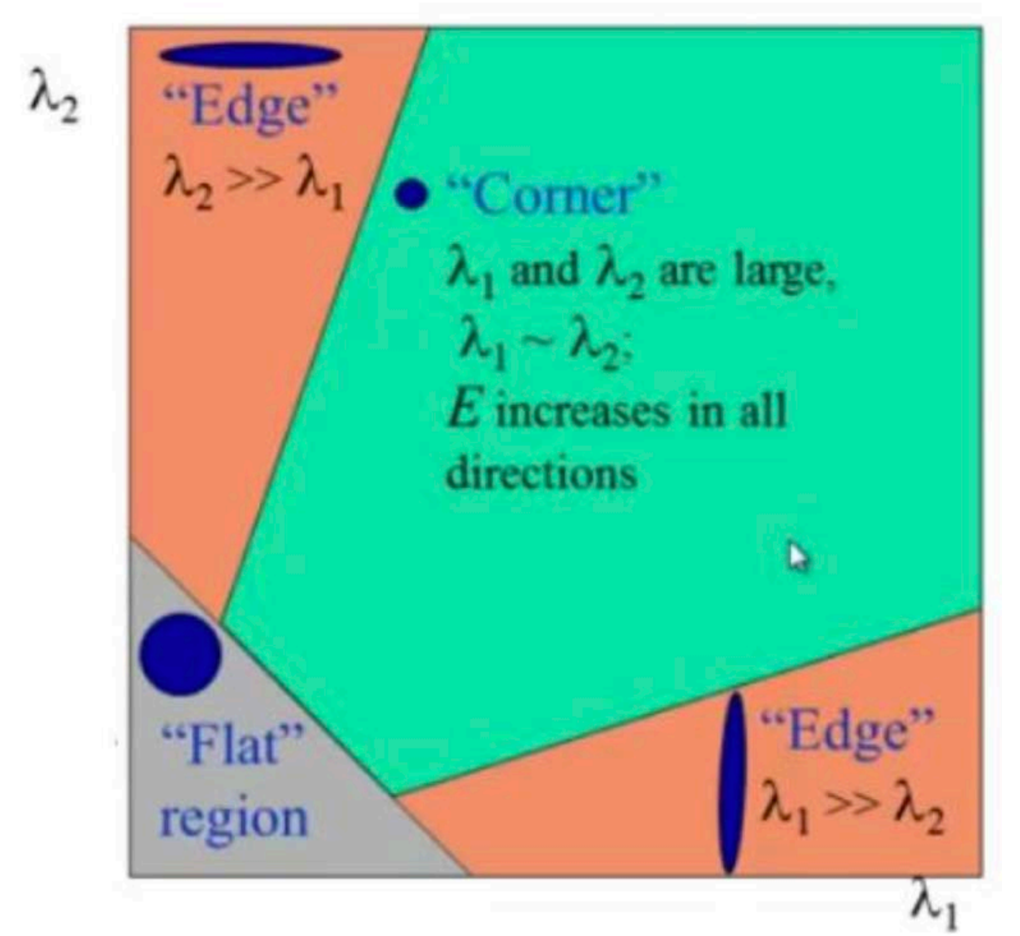


"corner":
significant change in all directions, i.e., even the minimum change is large

"Cornerness" Scores

Calling λ_1 and λ_2 the eigenvalues of the matrix **G**





$$S(\boldsymbol{G}) = \lambda_{\min}(\boldsymbol{G})$$

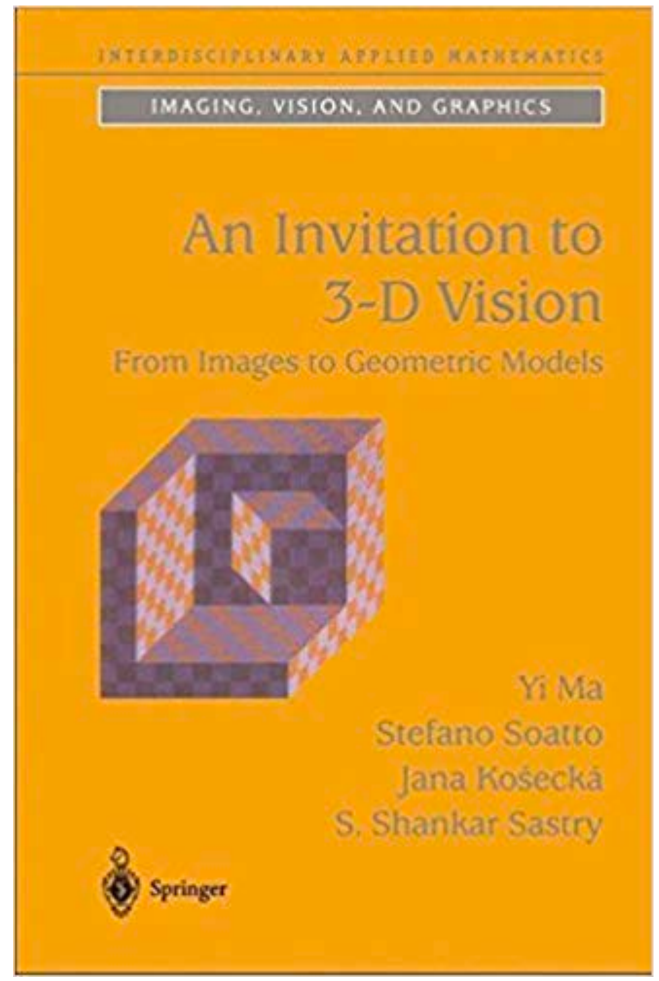
Shi-Tomasi corner detector

$$C(\boldsymbol{G}) = \det(\boldsymbol{G}) - k\ \mathrm{tr}\,(\boldsymbol{G})^2$$

Harris corner detector

Today

- Feature Detection
- Feature Tracking
- Feature Matching

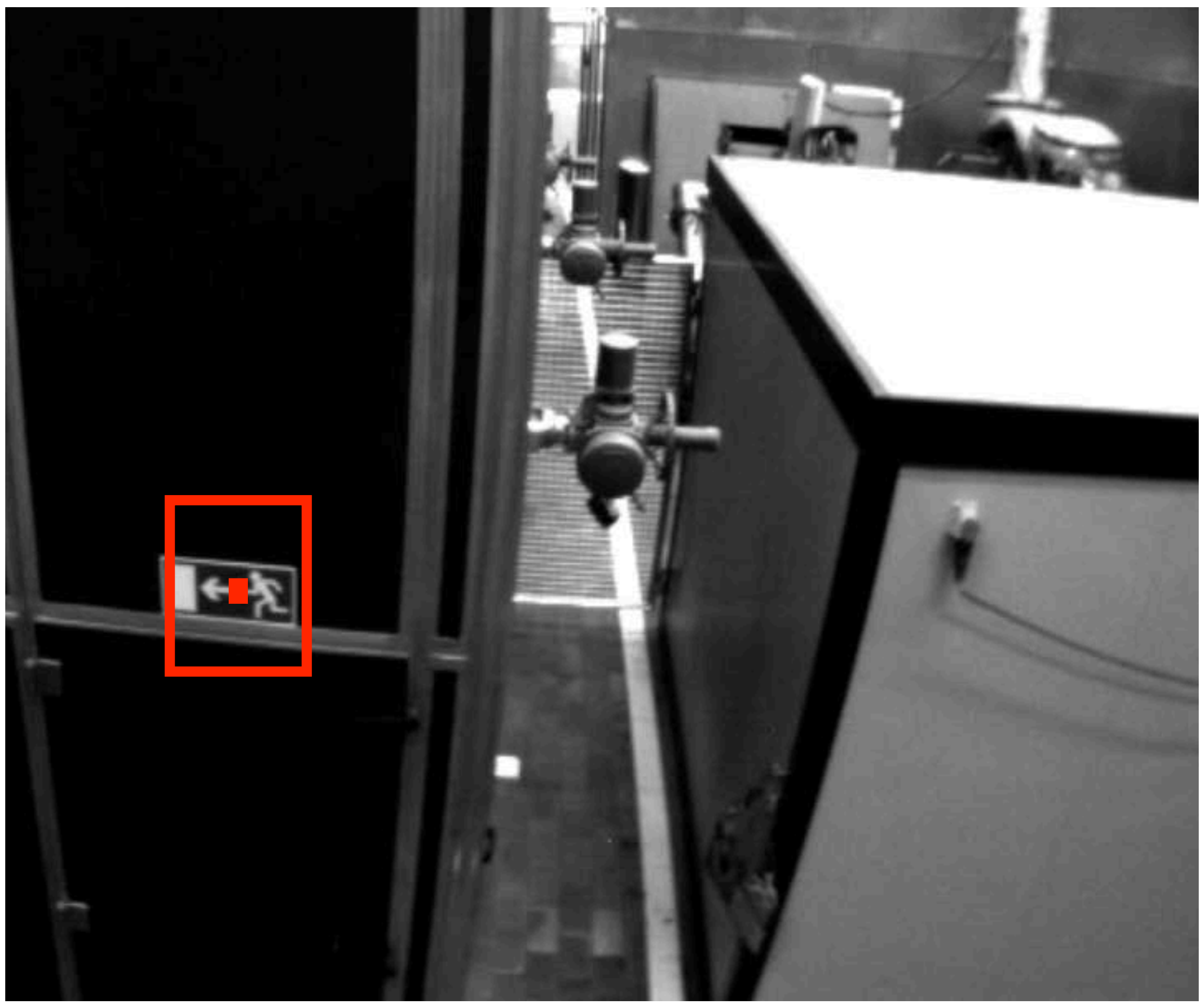


Chapter 4

Image Primitives and Correspondence

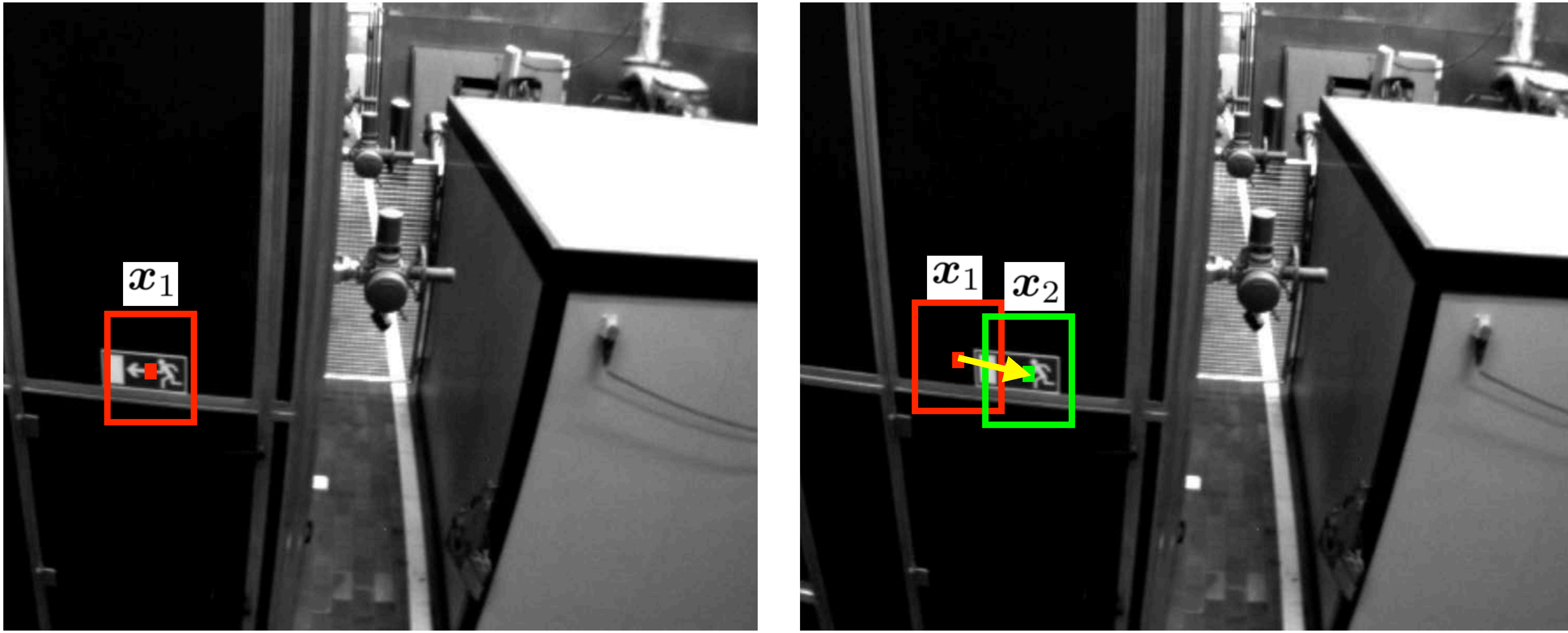
Correspondences

given a corner in image I_1 (and its neighborhood),
how can we find corresponding pixel in I_2 ?




- **Feature tracking** (~ optical flow)
- **Feature matching** (descriptor-based)

Feature Tracking

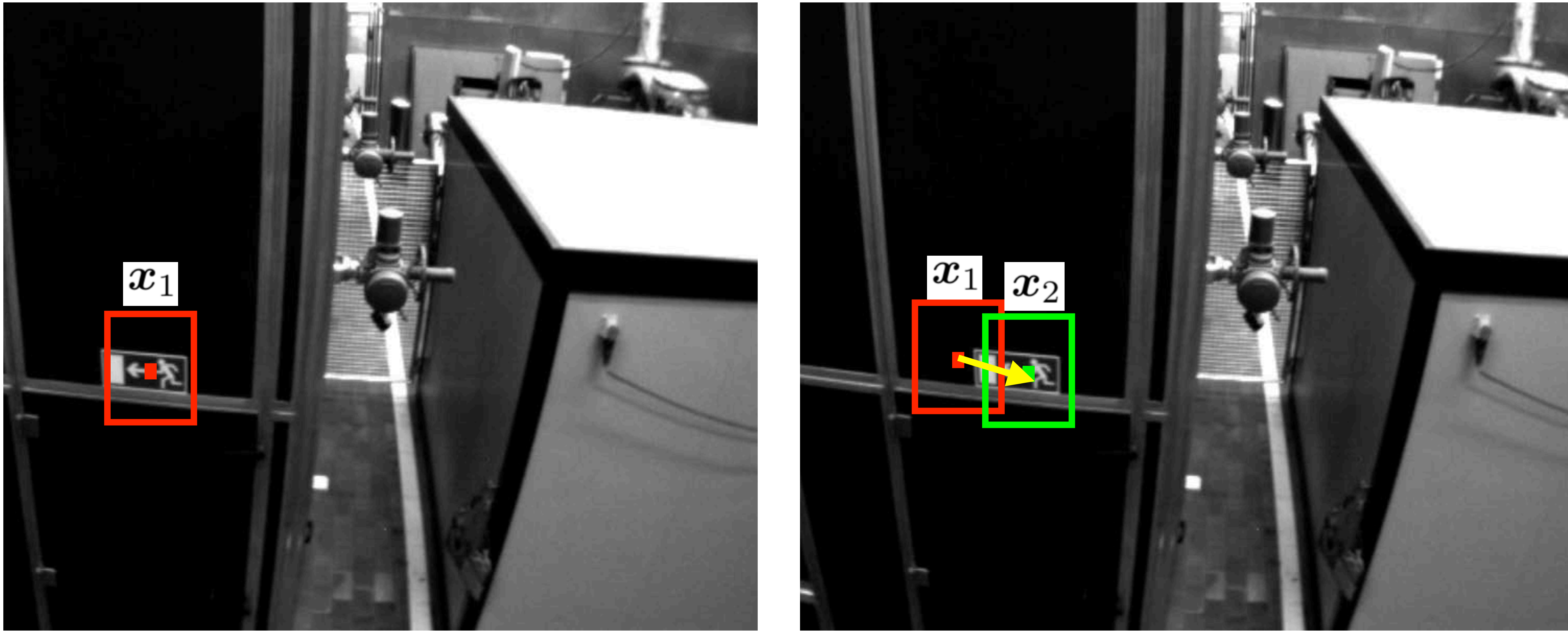


Computing the corresponding pixel (x_2) is the same as computing the displacement $\boldsymbol{\delta}$

$$\boldsymbol{x}_2 = \boldsymbol{x}_1 + \boldsymbol{\delta}$$

(translational motion model)

Feature Tracking

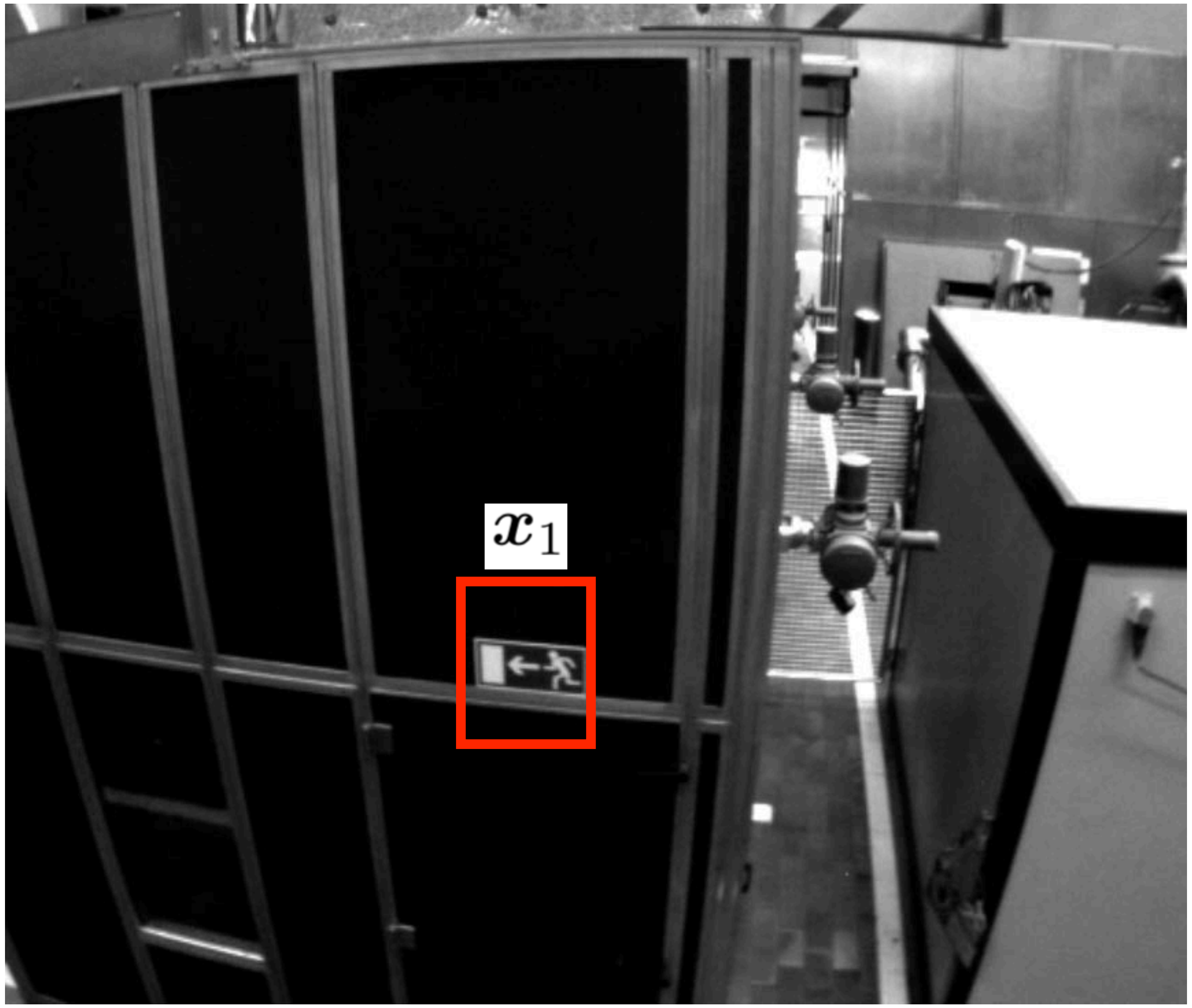


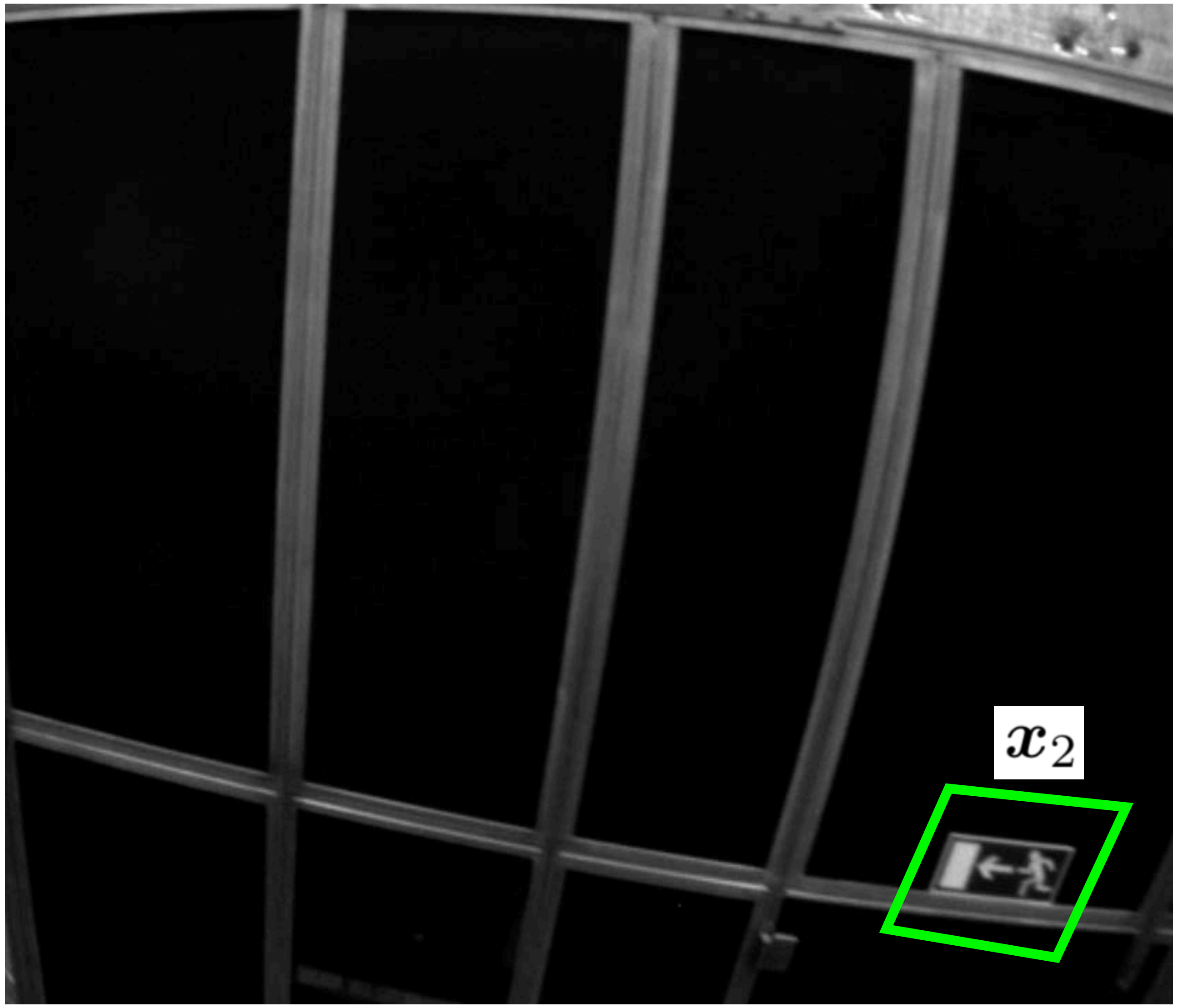
Computing the corresponding pixel ($\boldsymbol{x}_2$) is the same as computing the displacement $\boldsymbol{\delta}$

$$\min_{\boldsymbol{\delta}} \sum_{\boldsymbol{y} \in W(\boldsymbol{x}_1)} \|\mathcal{I}_1(\boldsymbol{y}) - \mathcal{I}_2(\boldsymbol{y} + \boldsymbol{\delta})\|^2$$

(translational motion model)

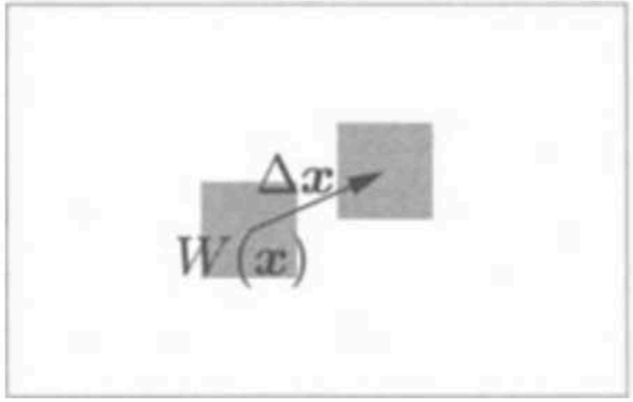
Feature Tracking

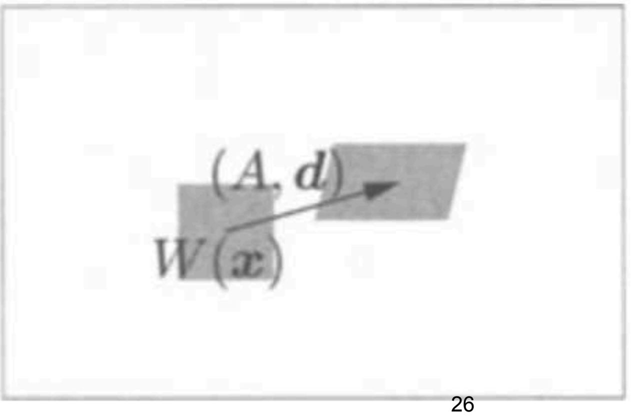




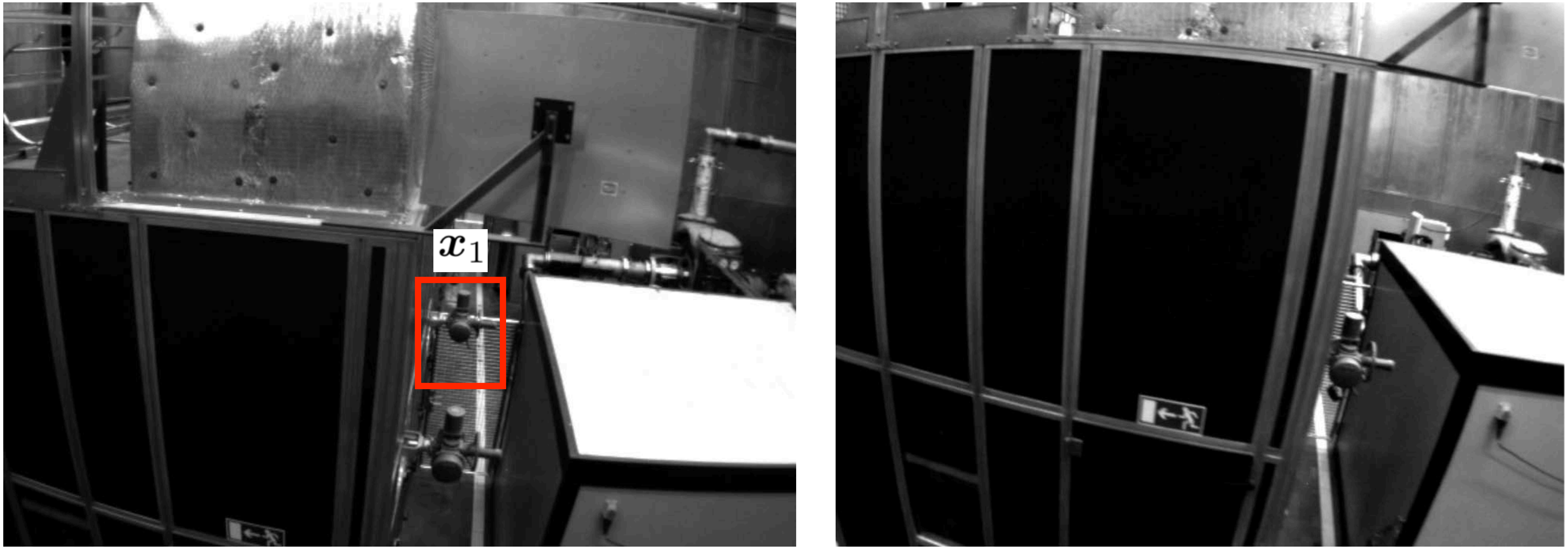
$$\min_{\boldsymbol{A},\boldsymbol{\delta}} \sum_{\boldsymbol{y}\in W(\boldsymbol{x}_1)} \|\mathcal{I}_1(\boldsymbol{y}) - \mathcal{I}_2(\boldsymbol{A}\boldsymbol{y} + \boldsymbol{\delta})\|^2$$

(affine motion model)



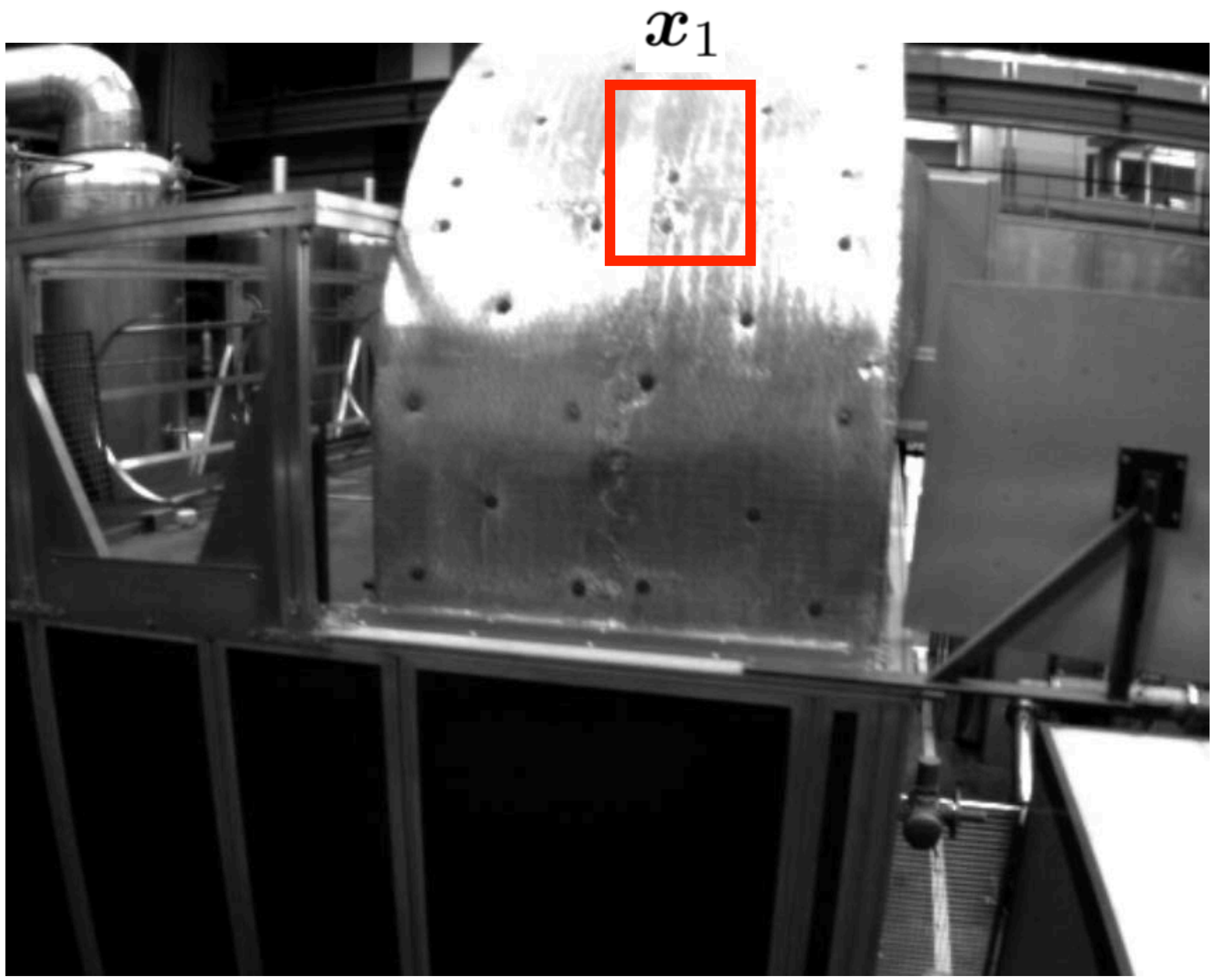


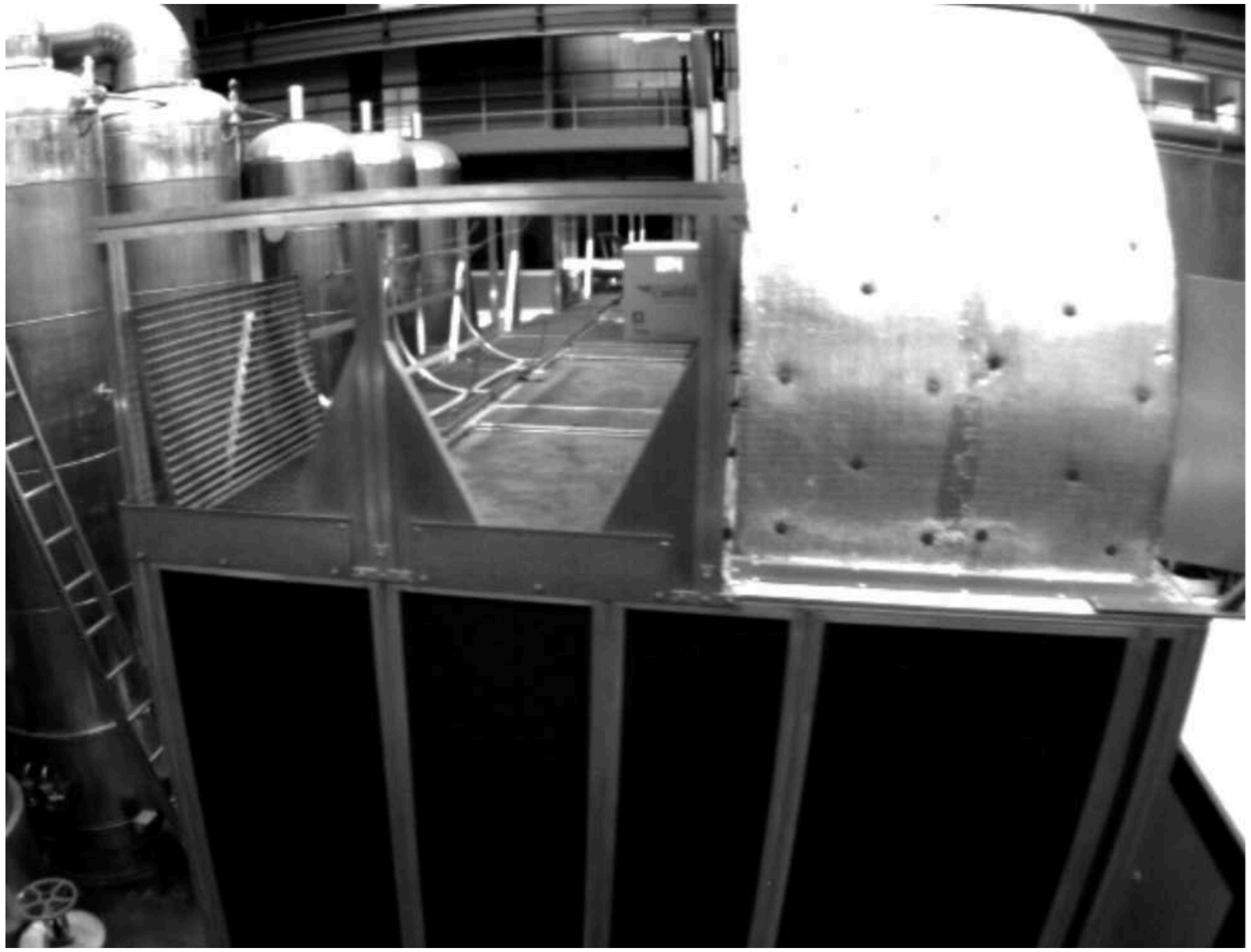
Hidden Assumptions



Pixel motion models not valid in presence of occlusions

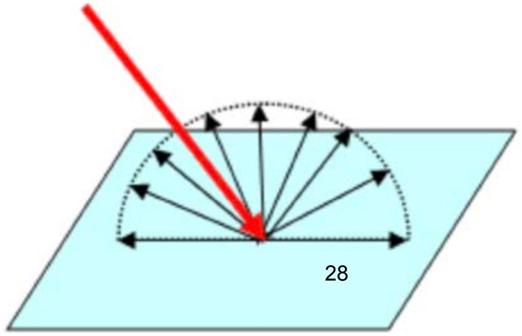
Hidden Assumptions





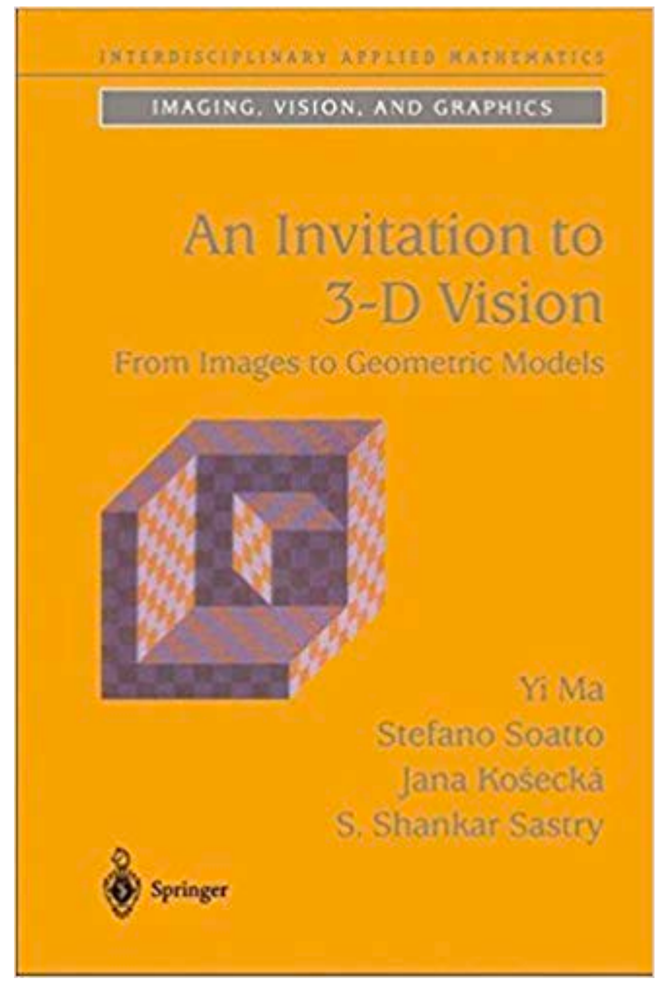
Matching image patches assume that the brightness does not change due to viewpoint changes (**brightness constancy constraints**)

True for Lambertian surfaces



Today

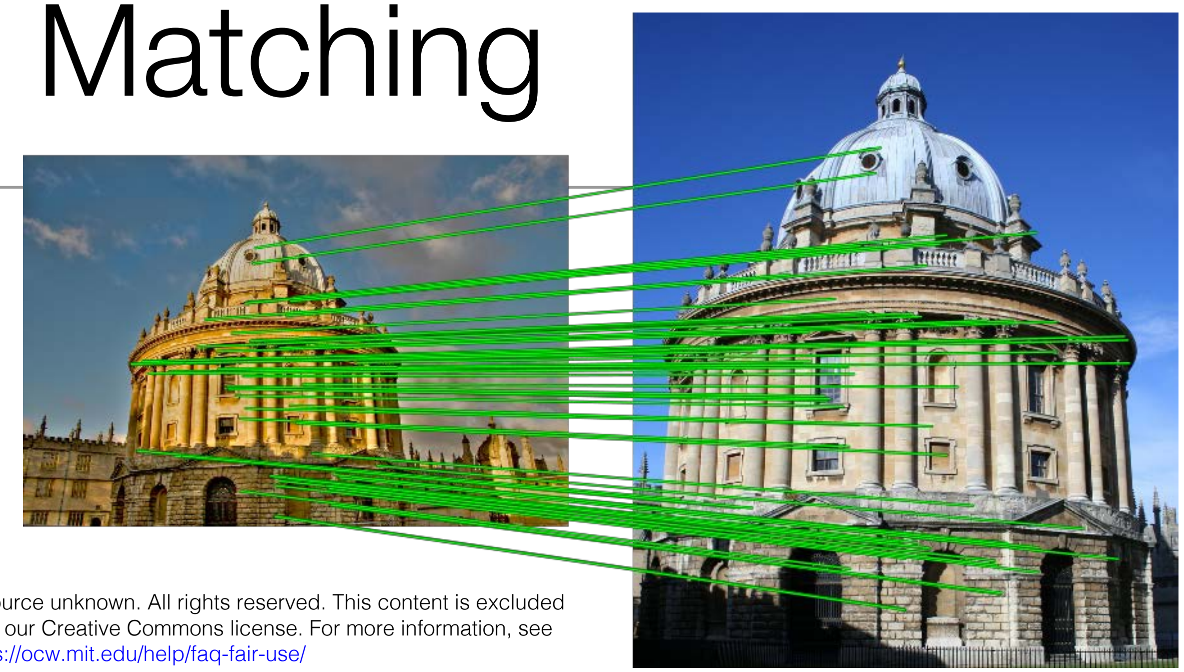
- Feature Detection
- Feature Tracking
- Feature Matching



Chapter 4

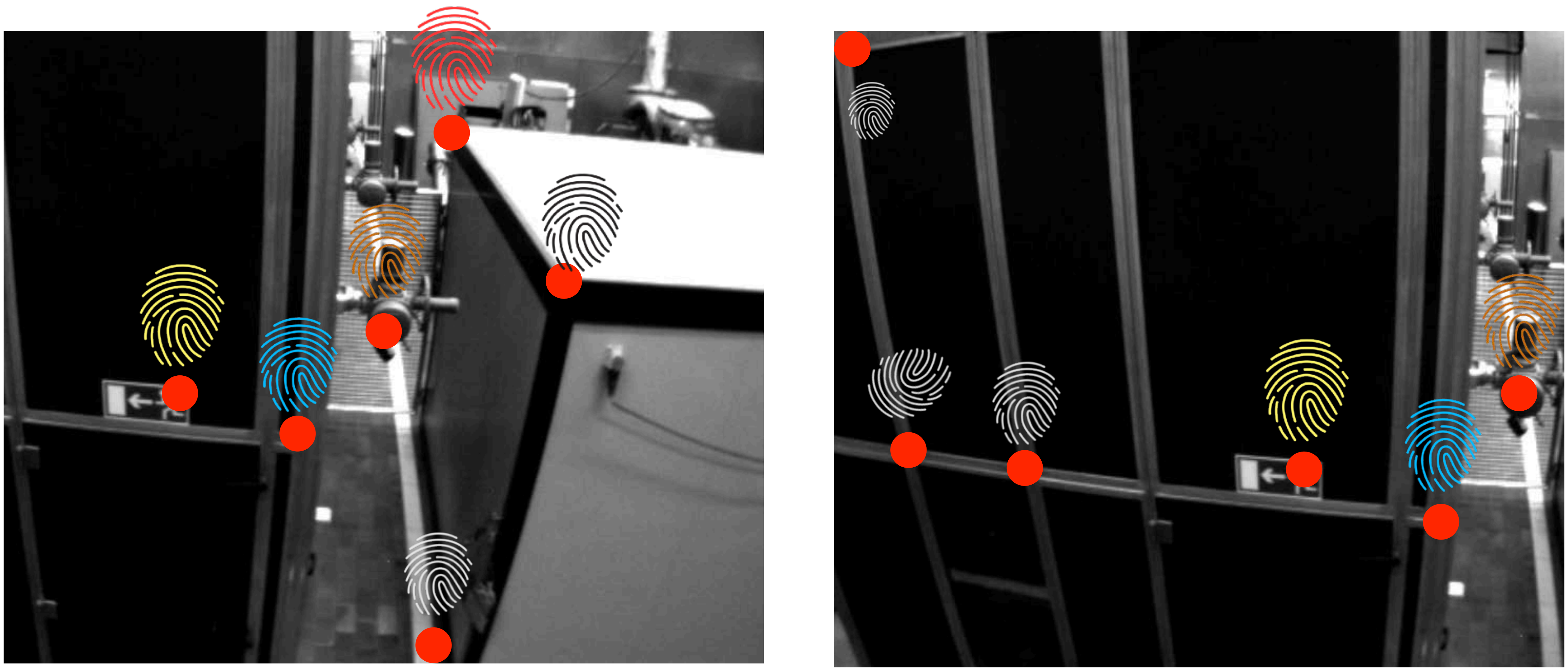
Image Primitives and Correspondence

Descriptor-based Feature Matching



© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

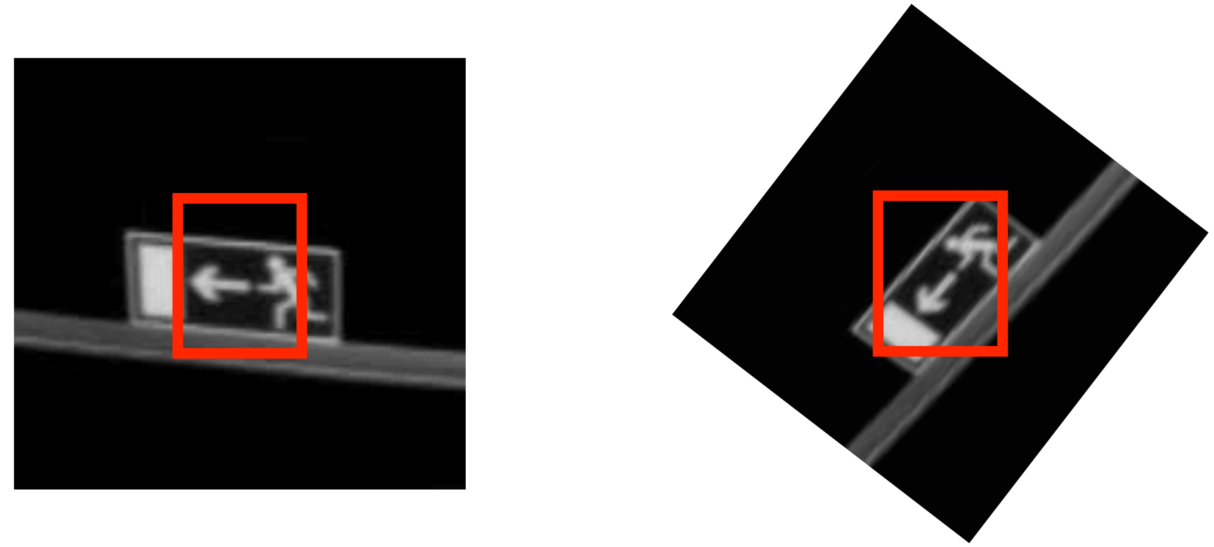
Feature tracking does not typically work for large changes of viewpoint (**large baseline**)



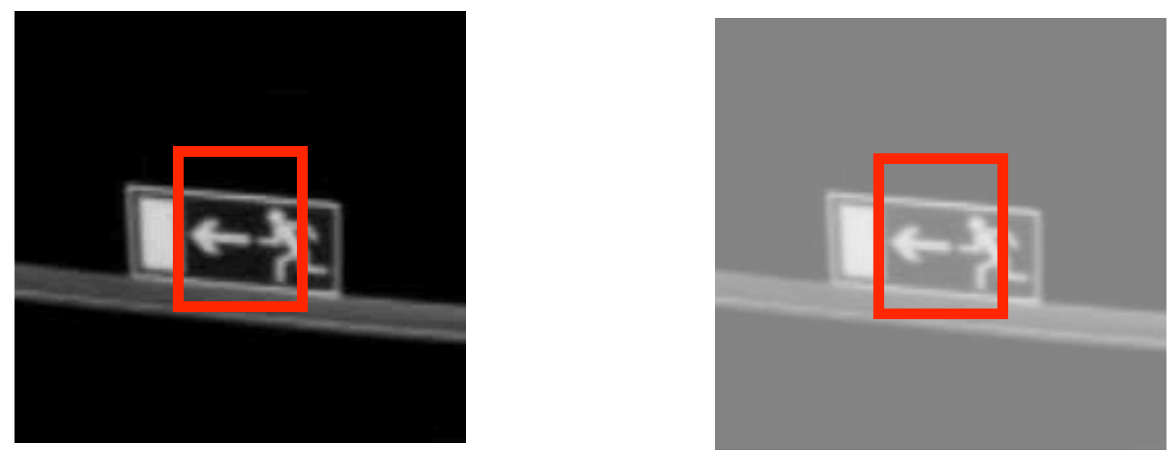
Descriptor is a signature we attach to a (point) feature, that describes local appearance

Ideal Properties of a Detector/Descriptor

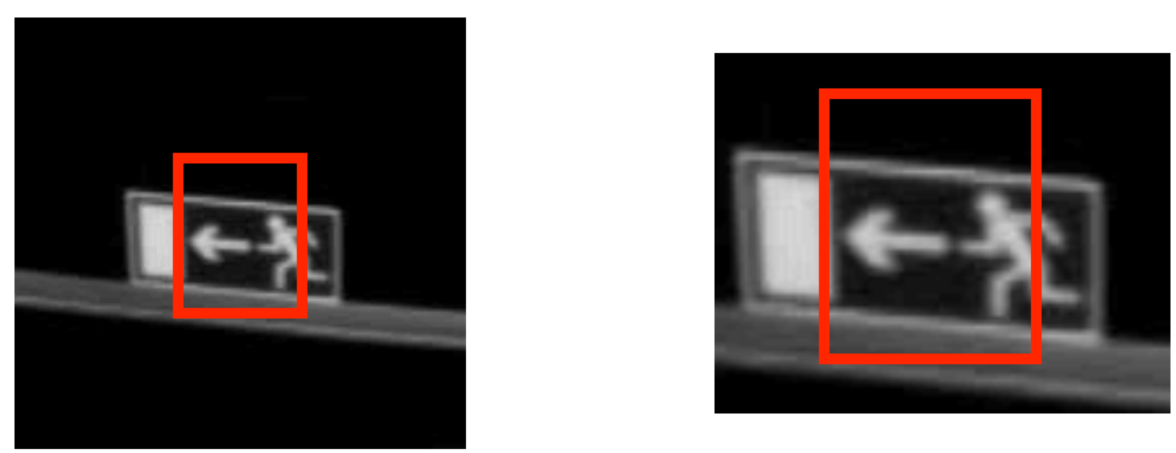
Rotation invariance
(more generally:
Viewpoint invariance)



Illumination invariance



Scale invariance

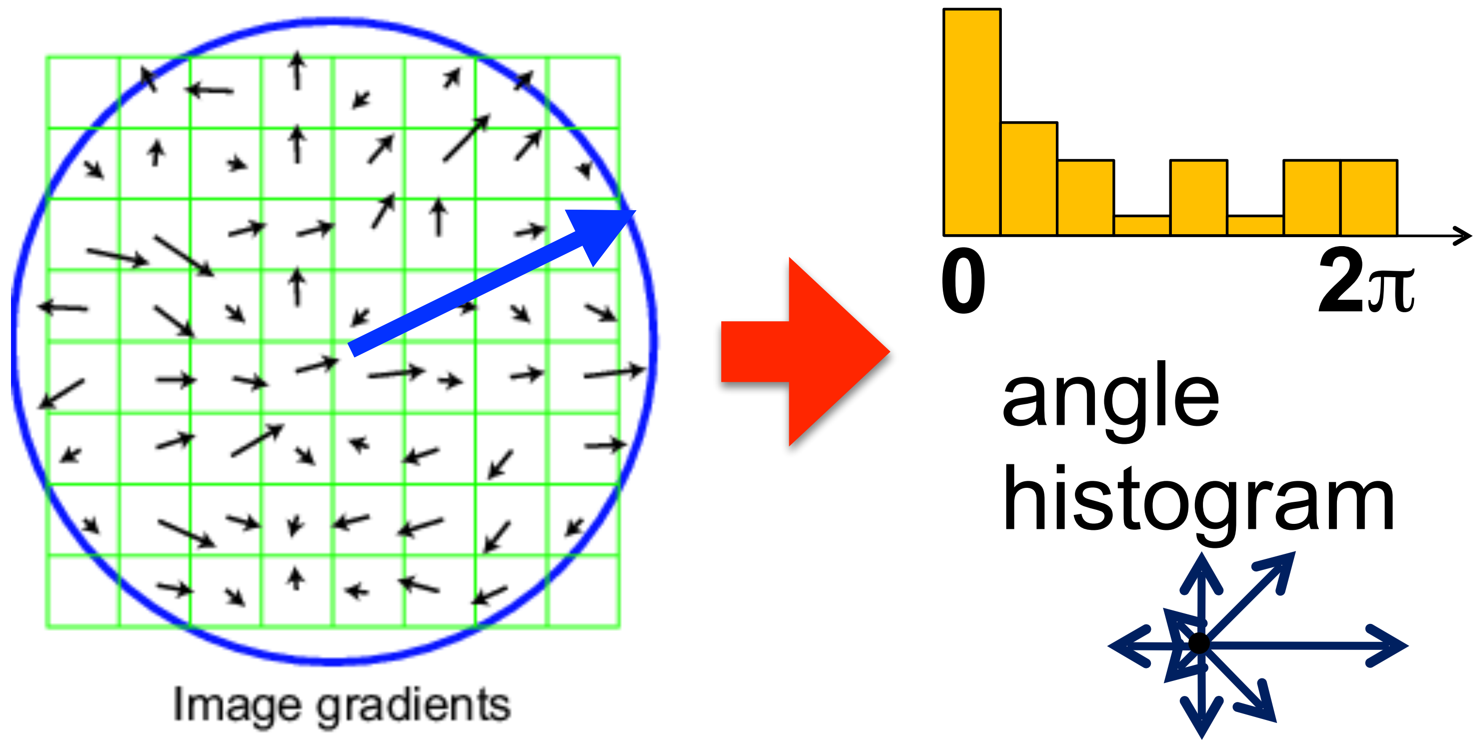


(more in the Lab 5 handouts: repeatability, efficiency ..)

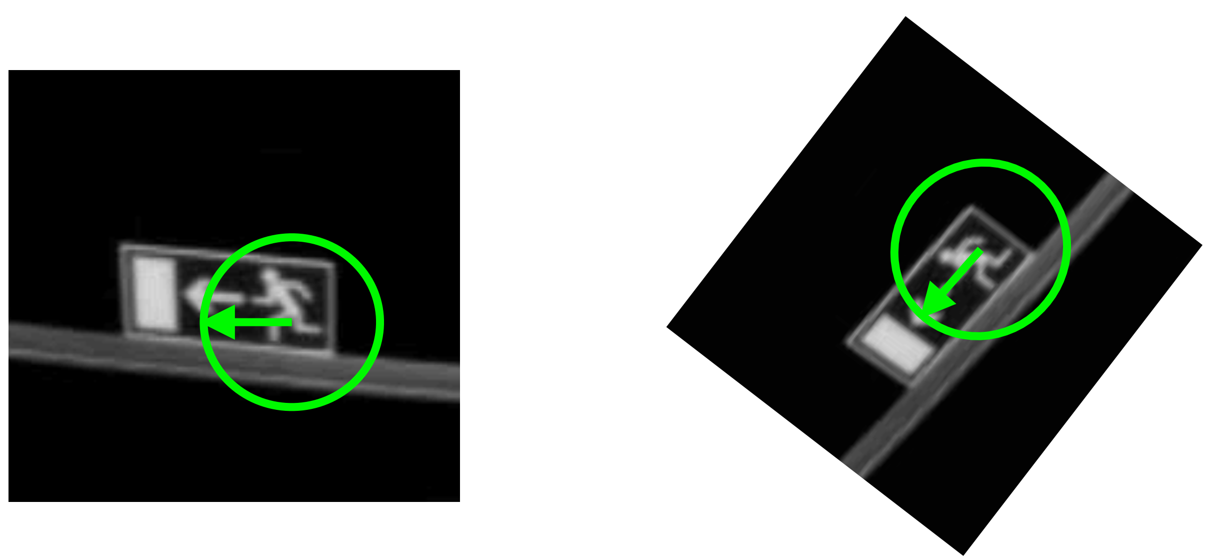
Example: SIFT Descriptor (1/2)

SIFT: Scale-Invariant Feature Transform

- Take 16x16 square window around detected feature
- Compute gradient orientation and magnitude for each pixel
- Create histogram of gradients weighted by magnitude
- Peak is **orientation** of feature



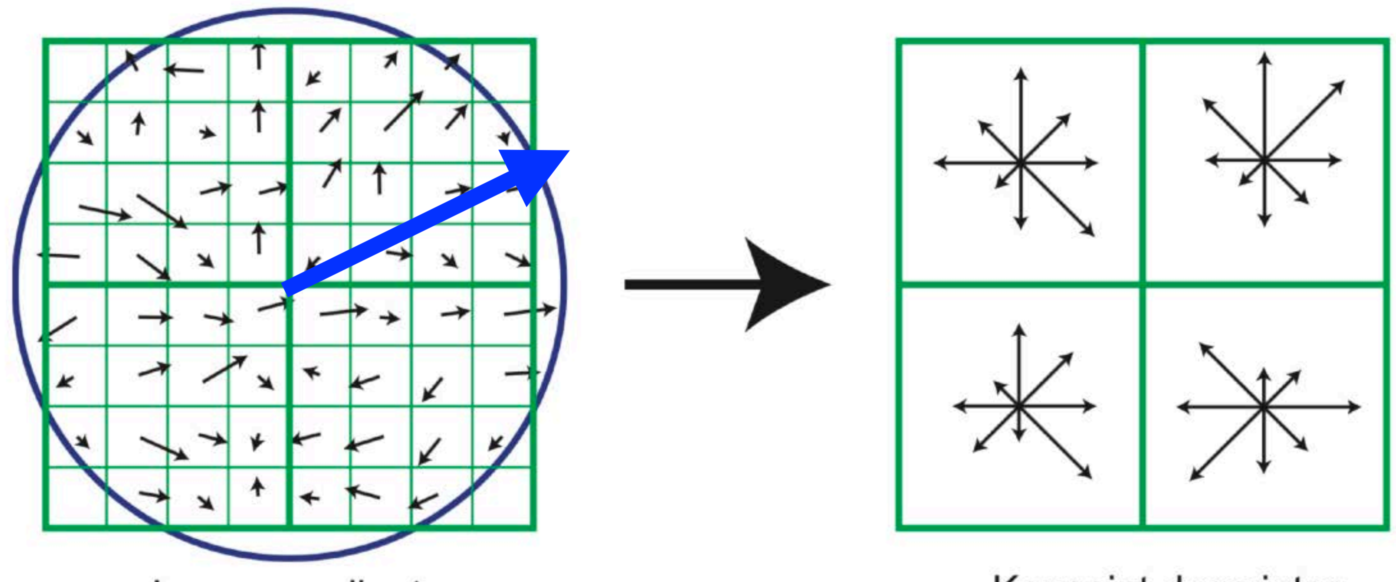
© Dr. Mubarak Shah. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/



D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 1150-1157 vol.2, doi: 10.1109/ICCV.1999.790410 © IEEE. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

Lowe, David G. (1999). "Object recognition from local scale-invariant features", CVPR'99

Example: SIFT Descriptor (2/2)



© Dr. Mubarak Shah. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

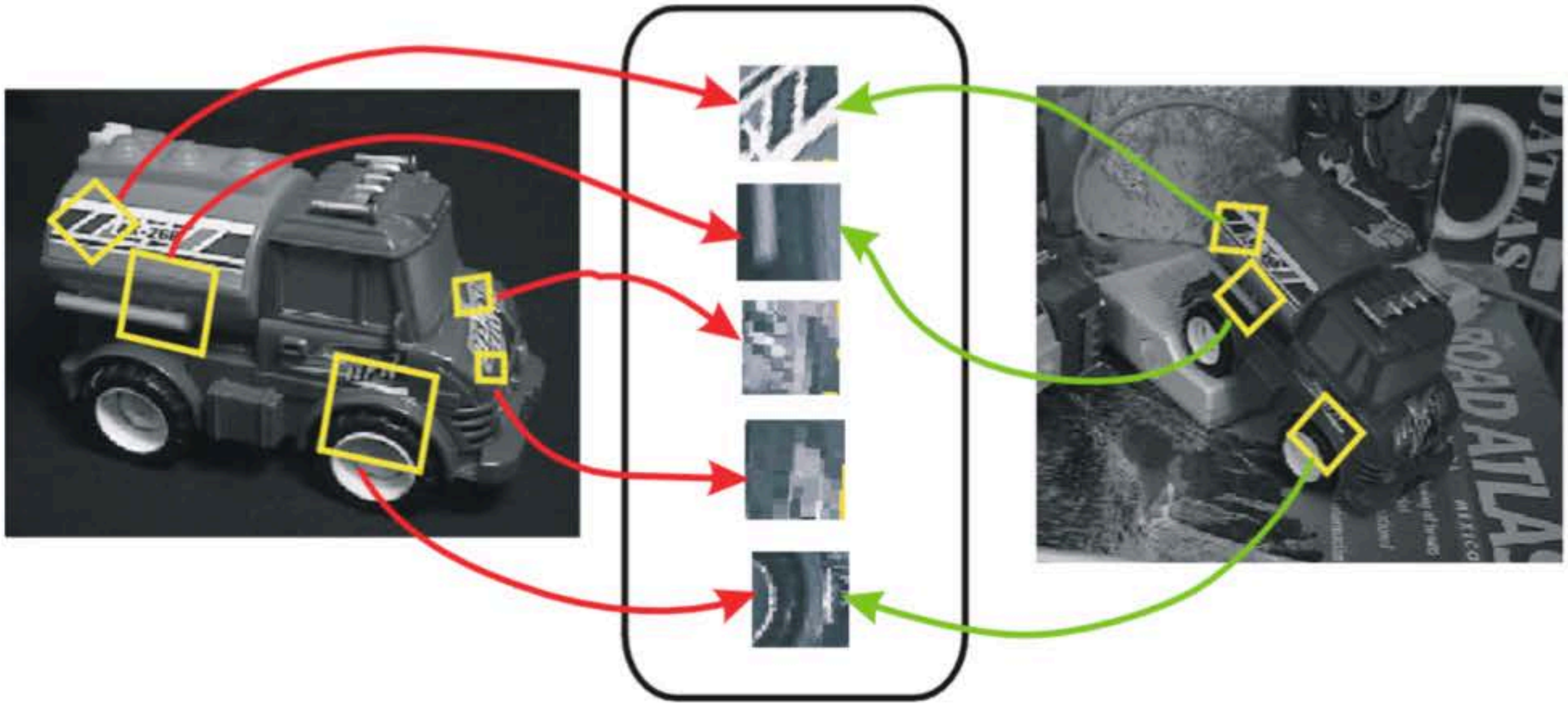
How to get SIFT descriptor?

- Transform all gradients with respect to (main) orientation
- Split window in 16 squares and for each compute a histogram with 8 sectors
- Stack histogram into a **descriptor** vector of 16 x 8 = 128 scalars
- Normalize to have norm = 1

D. G. Lowe, "Object recognition from local scale-invariant features," Proceedings of the Seventh IEEE International Conference on Computer Vision, Kerkyra, Greece, 1999, pp. 1150-1157 vol.2, doi: 10.1109/ICCV.1999.790410 © IEEE. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

Lowe, David G. (1999). "Object recognition from local scale-invariant features", CVPR'99

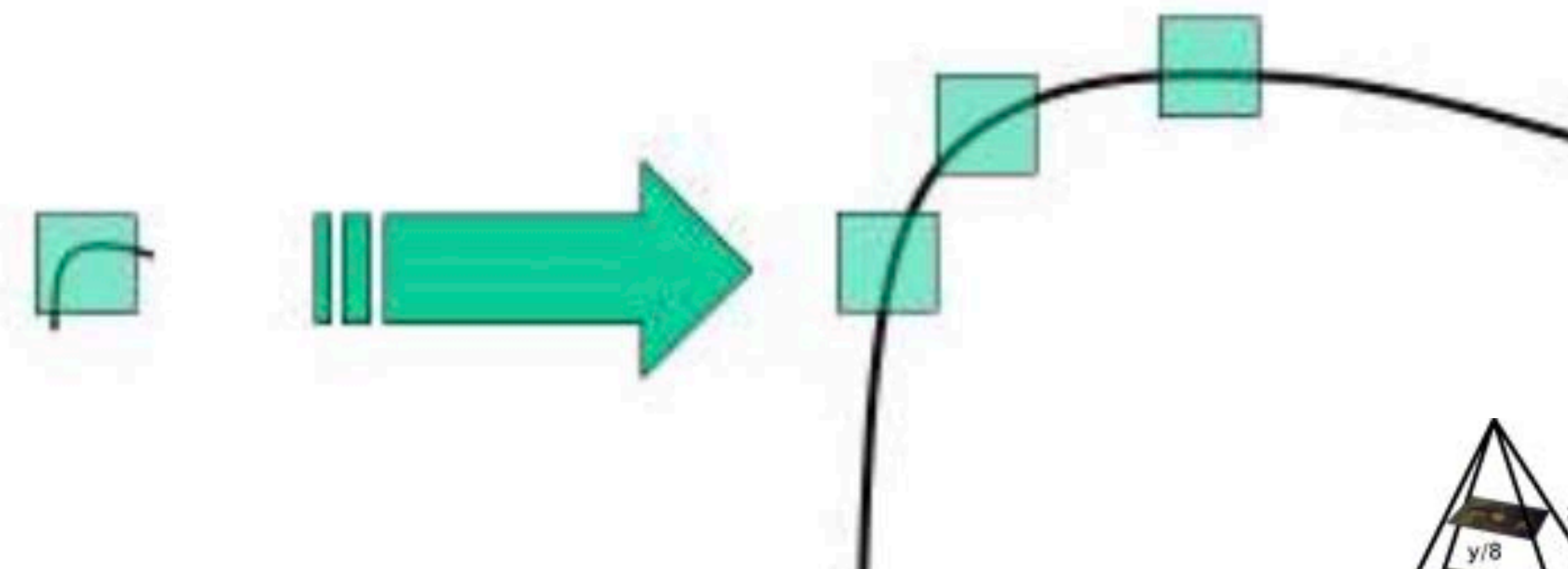
Feature Matching

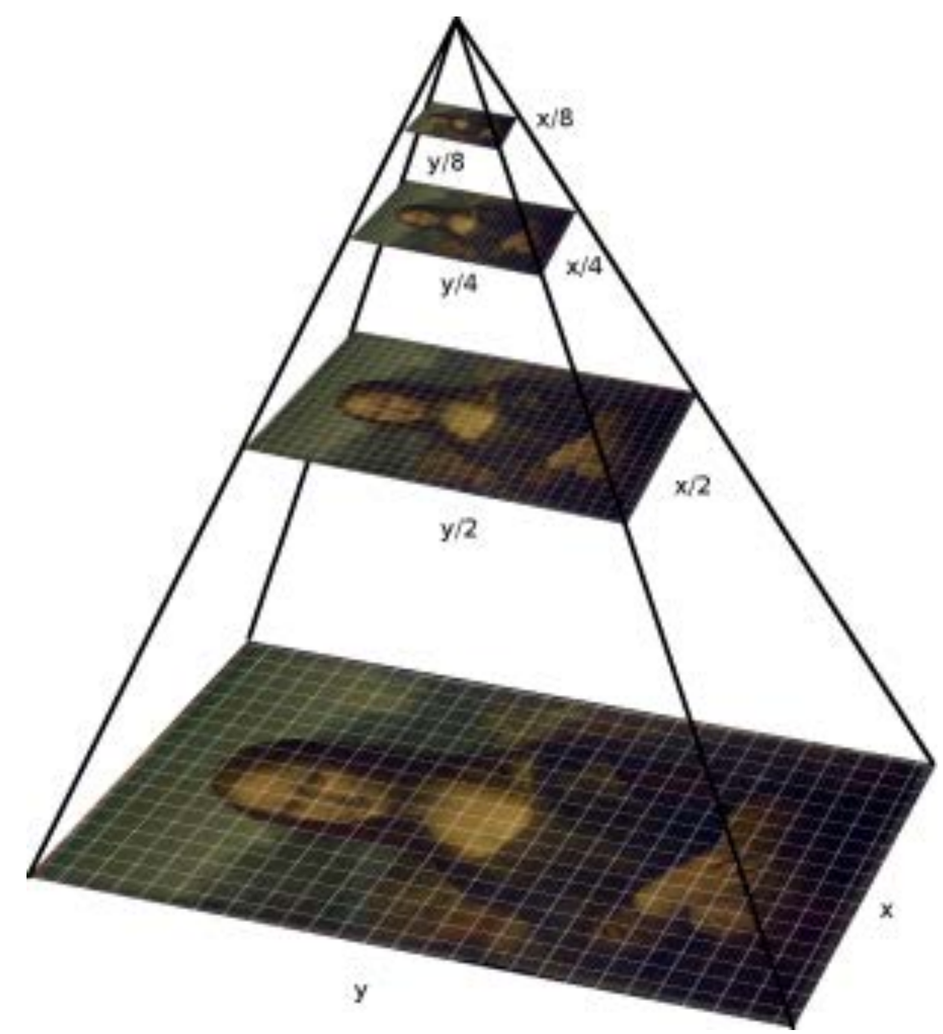


© Frank Dellaert. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

- For each descriptor in $\boldsymbol{I_1}$ find closest descriptor in $\boldsymbol{I_2}$ (nearest neighbor)
- Speed up with **approximate** nearest neighbor algorithms (FLANN library)

Are Harris Corners **Scale** invariant?





- **Other detectors have been proposed**:
  huge literature:
  SIFT, SURF, ORB, BRIEF, MSER, …
- **blob detectors**:
  process the image at different scales

https://www.youtube.com/watch?time_continue=3964&v=NPcMS49V5hg

Zebras, Horsefly, and Optical Flow

© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

https://www.theatlantic.com/science/archive/2019/02/why-do-zebras-have-stripes-flies/583114/

But still controversial: https://www.cnn.com/2020/08/18/world/zebra-stripes-fly-bites-study-trnd-scn/index.html

MIT OpenCourseWare
https://ocw.mit.edu/

16.485 Visual Navigation for Autonomous Vehicles (VNAV)
Fall 2020

For information about citing these materials or our Terms of Use, visit: https://ocw.mit.edu/terms.