CIS 5800

Machine Perception

Instructor: Lingjie Liu

Lec 18: April 9, 2025

Administrivia

Final Exam:

FINAL EXAM

Wednesday, May 7, 2025: 3:00pm to 5:00pm in DRLB A1

[The info is on courses.upenn.edu](https://courses.upenn.edu)

David Rittenhouse Laboratory

David Rittenhouse Laboratory

209 South 33rd Street, Philadelphia, PA 19104

View on Campus Map | Get Directions

Area Manager
Smith, Edward
edsmith@upenn.edu
215-898-8650

Building Manager
Trumbo, James
jtrumbo@sas.upenn.edu
215-651-1516

Building Code: 510

Phase: Completed

Year Built: 1954

Floors: 6

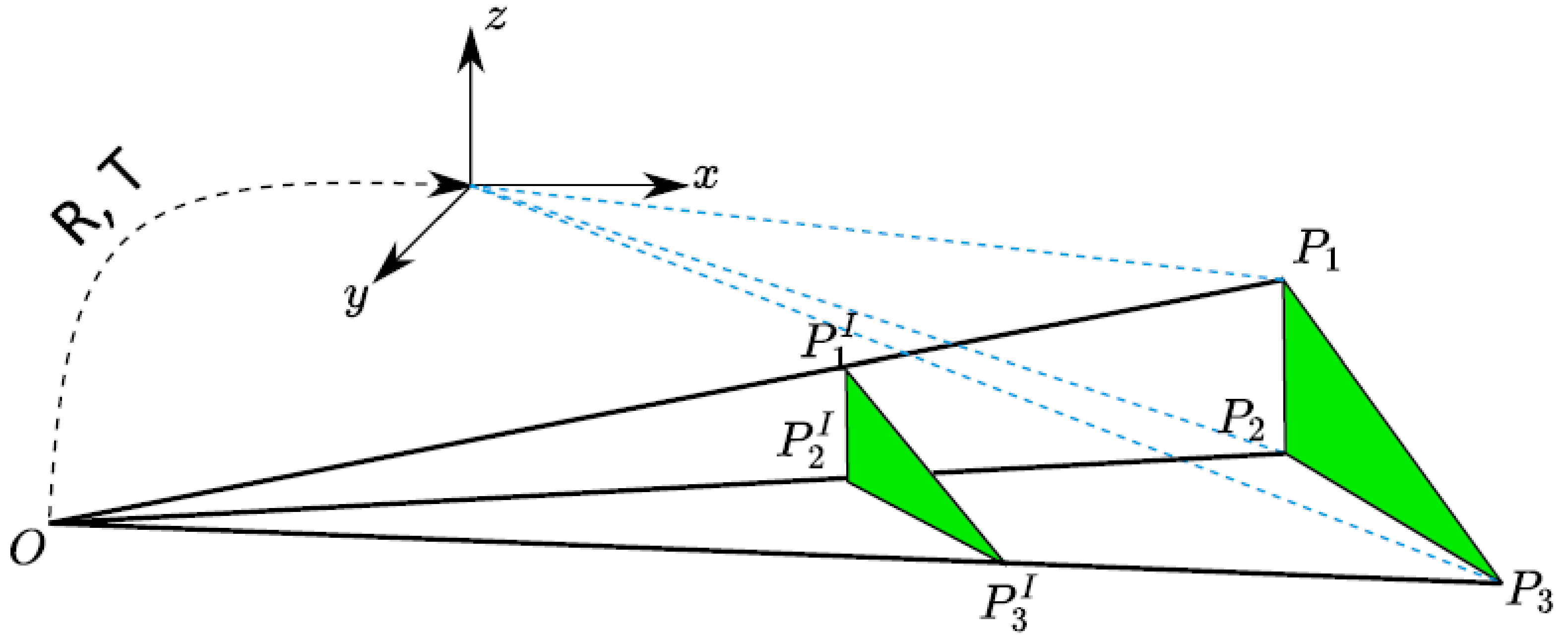
Architect: 1967

Other Name: Rittenhouse Labs, DRL, Rittenhouse, David Rittenhouse Laboratory, Benjamin Franklin Lab

Scope and format:
haven’t fully decided it. Will let you know the details next week.

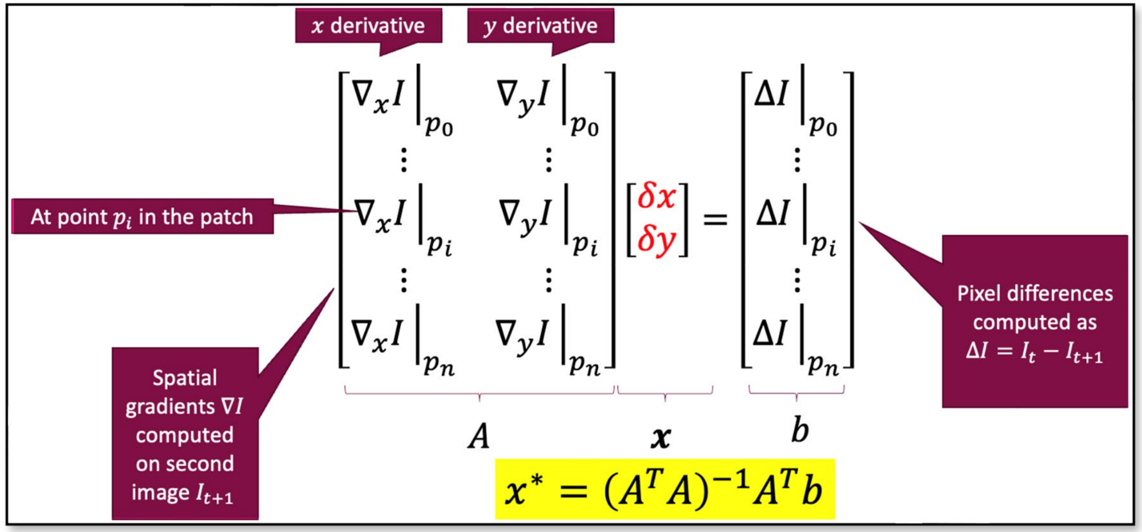
Correction: DOF in P3P is 6

- DOF in P3P should be 6, instead of 5



Given the point correspondences, find camera pose R, T

Recap: we discussed the invertibility of $A^T A$ for optical flow



We assumed we could invert, i.e. compute $(A^T A)^{-1}$

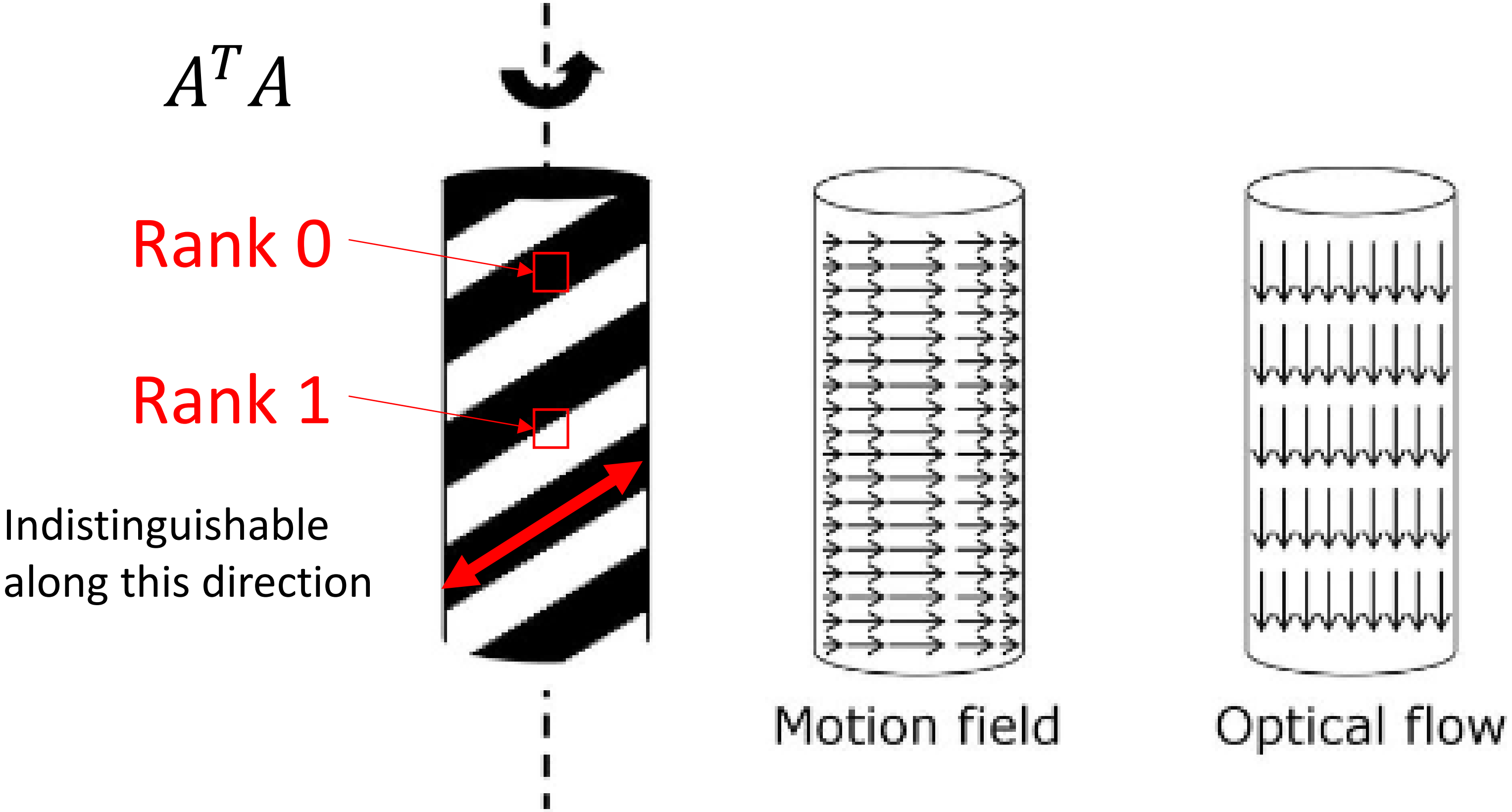
When would this fail? $(A^T A)_{2\times 2}$ is low-rank!

When is it rank 0? “Spatial gradients are all zero.”

When is it rank 1? “Spatial gradients are all aligned.”

Recap: Rank 0 and Rank 1 Regions in Barber’s Pole

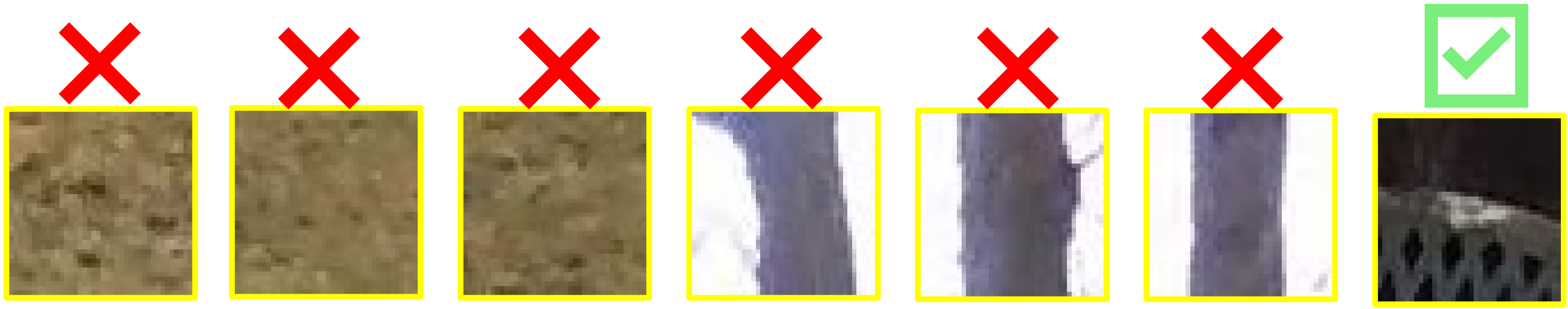
$A^T A$



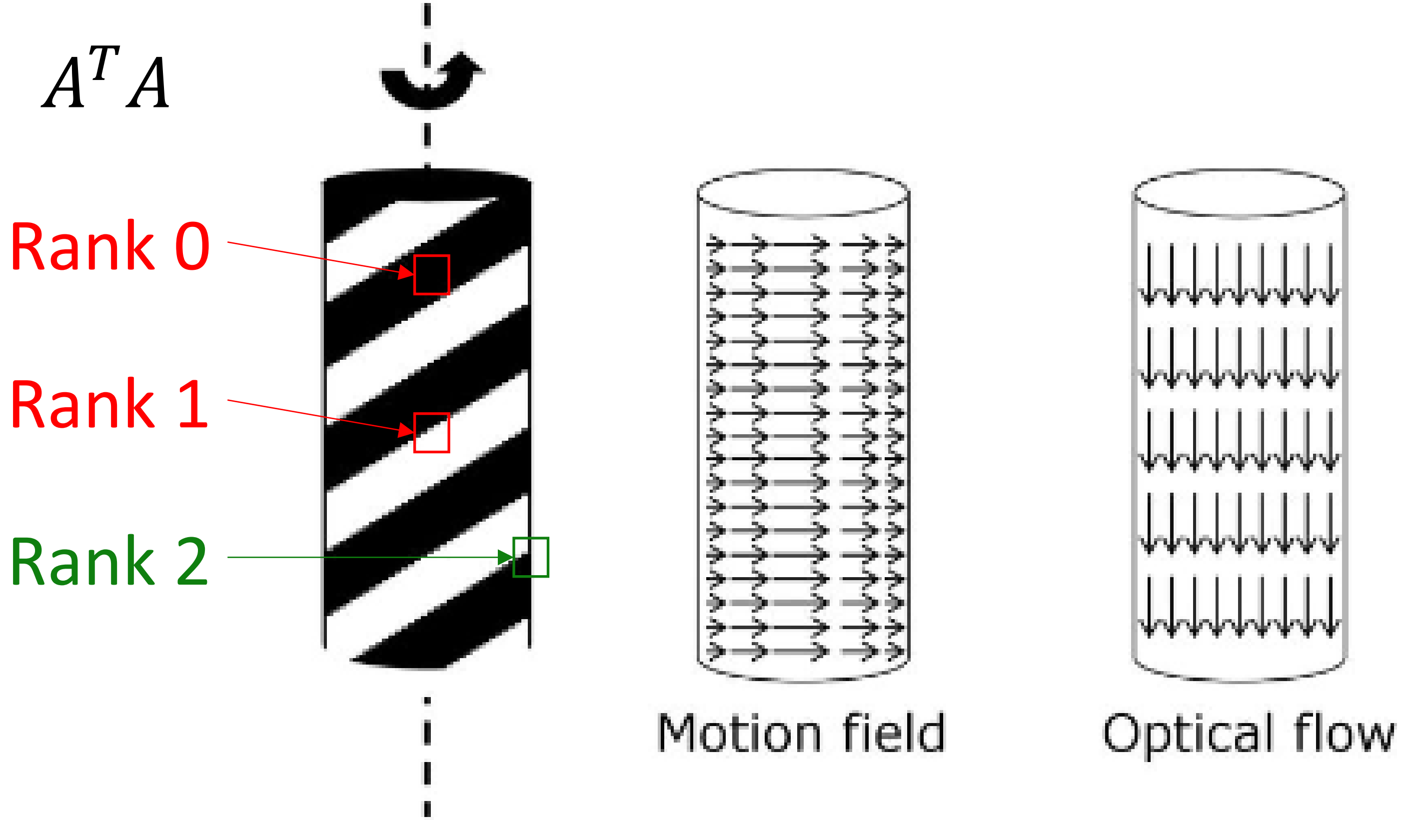
Recap: Consequence of Uninvertibility of $A^T A$

- Most places in the image are ‘uninteresting’ – we can’t track them – the interesting places are ‘sparse’.
- Flat regions are bad, edges are bad.
- “Corners” and high-texture regions are good.

Need to find such “features” that are easily trackable.

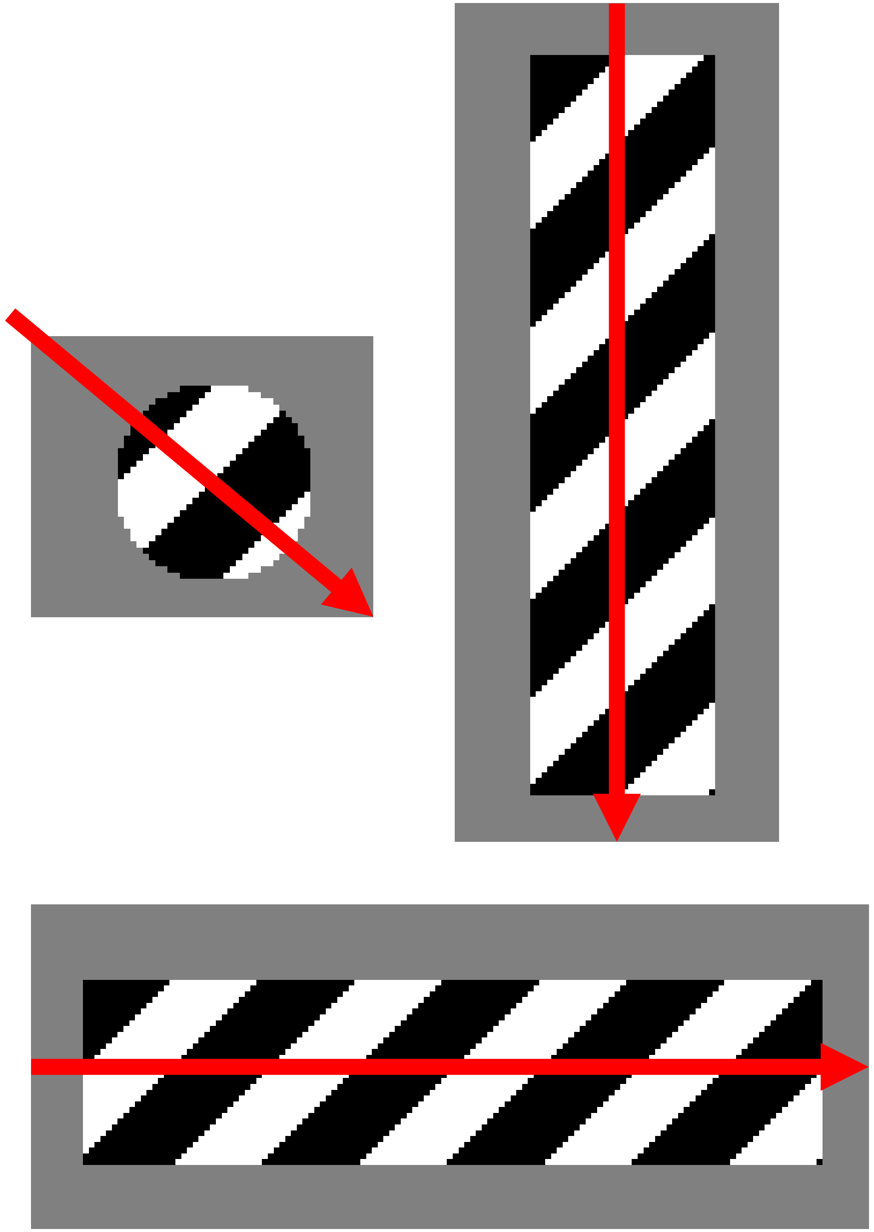


Recap: Rank 2 Region



https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/OWENS/LECT12/node4.html

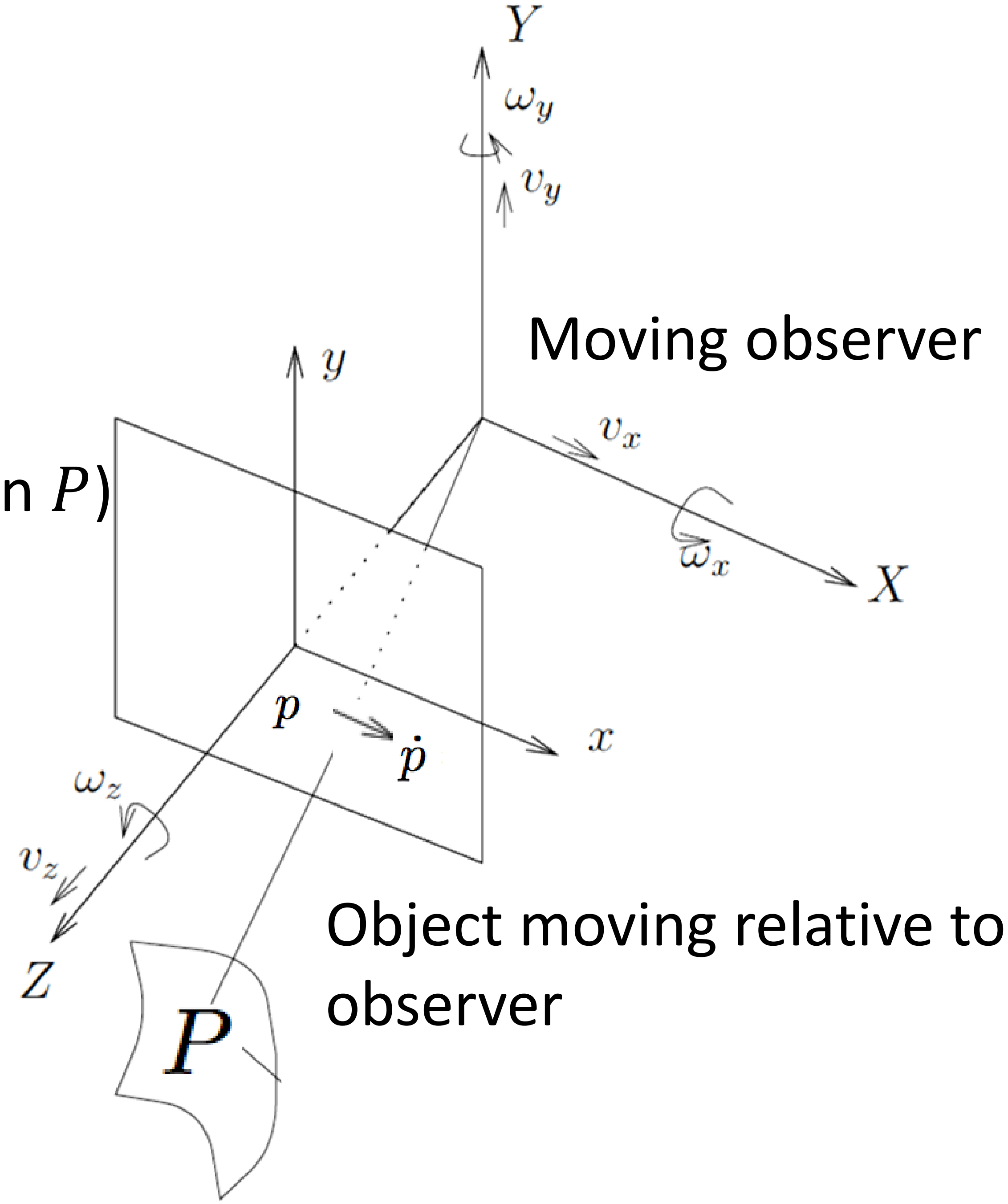
Why does the same motion appear to be different when looking at it through different apertures?



Predicting motion from flow

Projection equations for calibrated camera
$x = \frac{X}{Z}$, $y = \frac{Y}{Z}$
or in vector notation $p = \frac{1}{Z}P$

(homogeneous $p = (x, y, [1])$, Euclidean P)



Equation 1: pixel motion in terms of 3D object motion

Projection equations for calibrated camera

$x = \frac{X}{Z}, \; y = \frac{Y}{Z}$

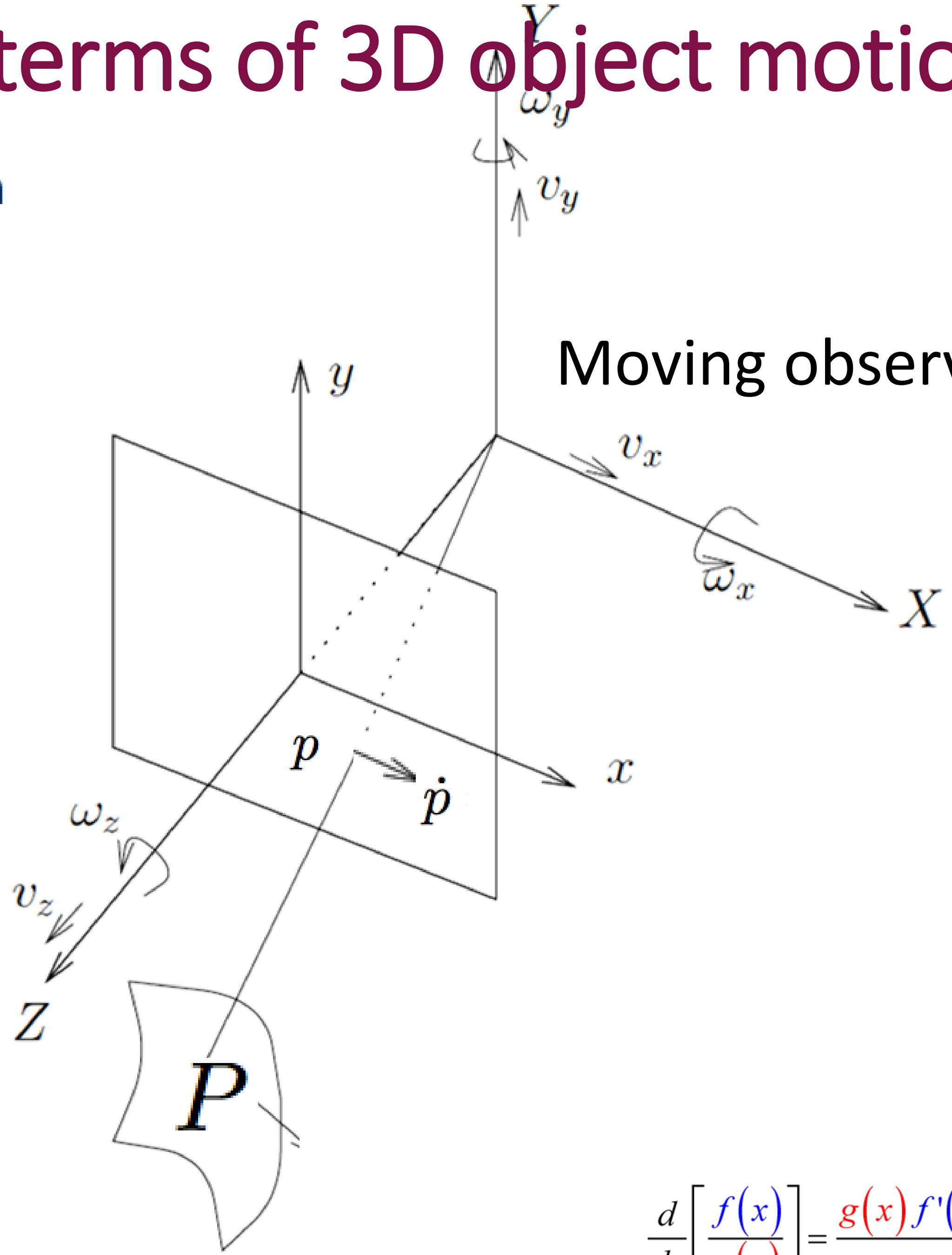
or in vector notation $p = \frac{1}{Z} P$

Differentiating w.r.t. time yields:

$$\boxed{\dot{p} = \frac{\dot{P}}{Z} - \frac{\dot{Z}}{Z} p}$$

Velocity of projection

Projection of 3D velocity



Moving observer

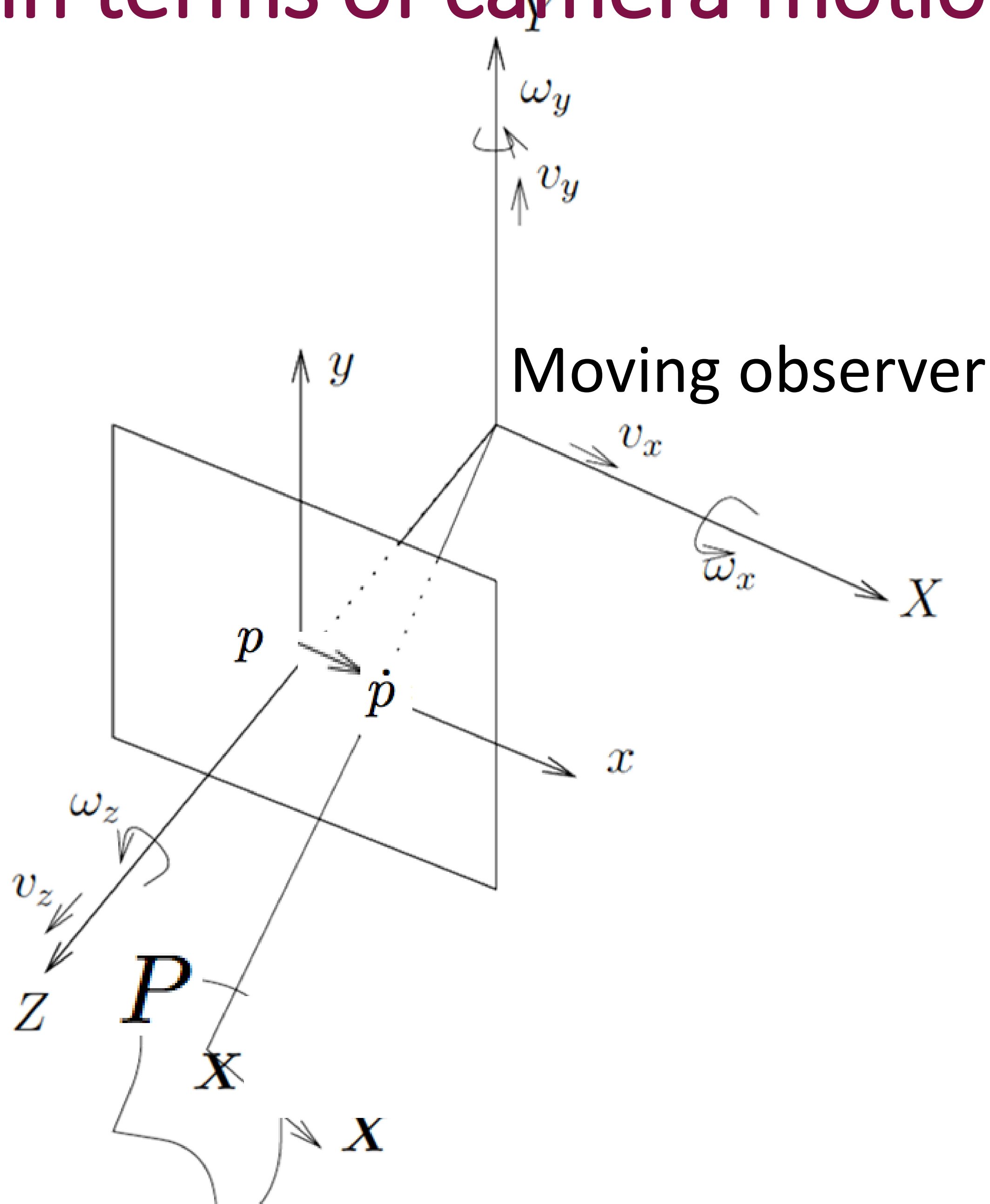
$$\frac{d}{dx}\left[\frac{f(x)}{g(x)}\right] = \frac{g(x)f'(x) - f(x)g'(x)}{\left(g(x)\right)^2}$$

Equation 2: 3D object motion in terms of camera motion

For a camera moving at velocity $\boldsymbol{V}$ spinning at angular velocity $\boldsymbol{\Omega}$

All in camera coords

$$\dot{P} = -\Omega \times P - V$$



Combining the two key equations

$$\dot{p} = \frac{\boxed{\dot{P}}}{Z} - \frac{\dot{Z}}{Z}p \qquad\qquad \dot{P} = -\Omega \times P - V$$

Notation abuse warning: $p = (x, y, 1)$, but we will sometimes write $\dot{p} = (\dot{x}, \dot{y})^T$

And $V = \left[V_x, V_y, V_z\right]^T$

Prove it now on the blackboard

$$\dot{p} = \underbrace{\frac{1}{Z}\begin{bmatrix} xV_z - V_x \\ yV_z - V_y \end{bmatrix}}_{\text{translational flow}} + \underbrace{\begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\Omega}_{\text{rotational flow independent of depth}}$$

Optical flow has two additive components: translational and rotational.

Assume that optical flow is computable and is equal to the motion field.

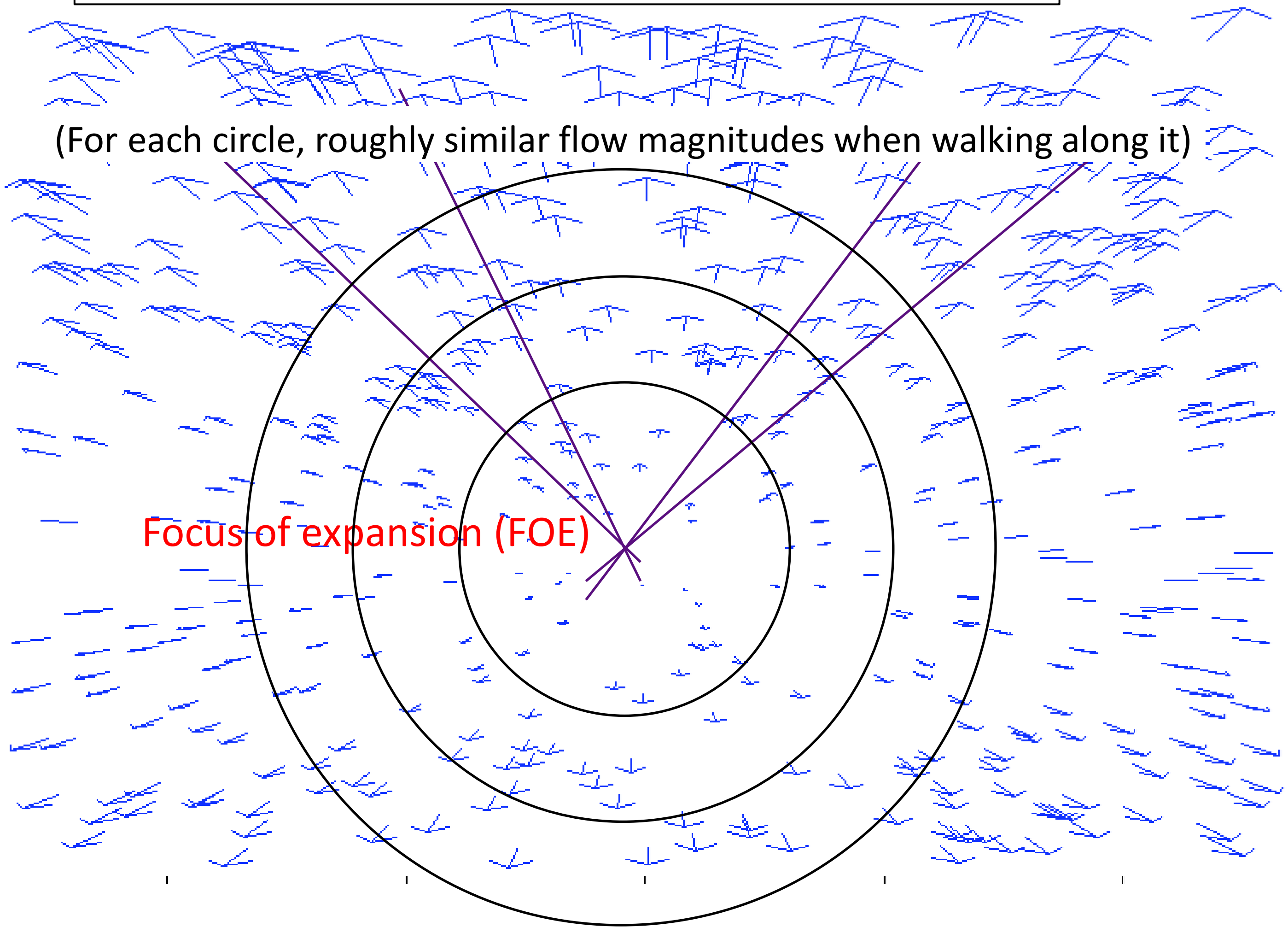
Given the optical flow field, can we work out the camera motion?

But first, let's go back to understanding translational and rotational flow terms from the decomposition

$$\dot{p} = \underbrace{\frac{1}{Z}\begin{bmatrix} xV_z - V_x \\ yV_z - V_y \end{bmatrix}}_{\text{translational flow}} + \underbrace{\begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\Omega}_{\text{rotational flow independent of depth}}$$

Translational Flow Part 1: Distance from FOE

Q: What must the world look like for this image to be a translational flow map for forward motion?
A: Z must be approx. constant



$$\dot{\boldsymbol{p}} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \frac{V_z}{Z} \left[\begin{array}{c} x - \boxed{\frac{V_x}{V_z}} \\ y - \boxed{\frac{V_y}{V_z}} \end{array} \right]$$

Focus of expansion (FOE)

Change in x (or y) coordinate is:

- proportional to how far away that coordinate is from V_x/V_z (or V_y/V_z)

$$\dot{p} = \underbrace{\boxed{\frac{1}{Z} \begin{bmatrix} xV_z - V_x \\ yV_z - V_y \end{bmatrix}}}_{\text{translational flow}} + \underbrace{\begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix} \Omega}_{\text{rotational flow independent of depth}}$$

Translational Flow Part 2: Inverse Time-To-Collision

Inverse "time to collision" of object Z plane with camera

$$\dot{\boldsymbol{p}} = \begin{bmatrix} \dot{x} \\ \dot{y} \end{bmatrix} = \boxed{\frac{V_z}{Z}} \begin{bmatrix} x - \\ y - \end{bmatrix} \boxed{\begin{matrix} \frac{V_x}{V_z} \\ \frac{V_y}{V_z} \end{matrix}}$$

Focus of expansion (FOE)

Change in pixel x (or y) coordinate is:

- proportional to how far away that pixel coordinate is from the focus of expansion V_x/V_z (or V_y/V_z)
- inversely proportional to the time to collision of the camera with the Z plane of the object in the camera coordinate system.

Given a fixed camera motion, and fixed pixel distance from the FOE, flow ∝ inverse "depth" i.e. point moves less if farther away.*

*Here "depth" means Z coordinate i.e. distance from camera $Z = 0$ plane, not distance from camera center.

Flashback: We have seen FOE before. Epipole!

Recall that in 2-view geometry, the epipole in one image plane is the image of the *other* camera center.

- Suppose camera center moves from $(0,0,0)$ at time 0 to (V_x, V_y, V_z) at time 1.
- The image of the time-1 camera at time 0 is $\frac{V_x}{V_z}, \frac{V_y}{V_z}$, i.e. the FOE!

Note that FOE does not depend on the scene, just the motion!

Can also arrive at the same conclusion by thinking about FOE as the intersection of flow vectors. How is this related to the epipole?

$$\dot{\boldsymbol{p}} = \begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \frac{V_z}{Z}\begin{bmatrix} x - \frac{V_x}{V_z} \\ y - \frac{V_y}{V_z}\end{bmatrix}$$

Focus of expansion (FOE)

ZH Ch 9

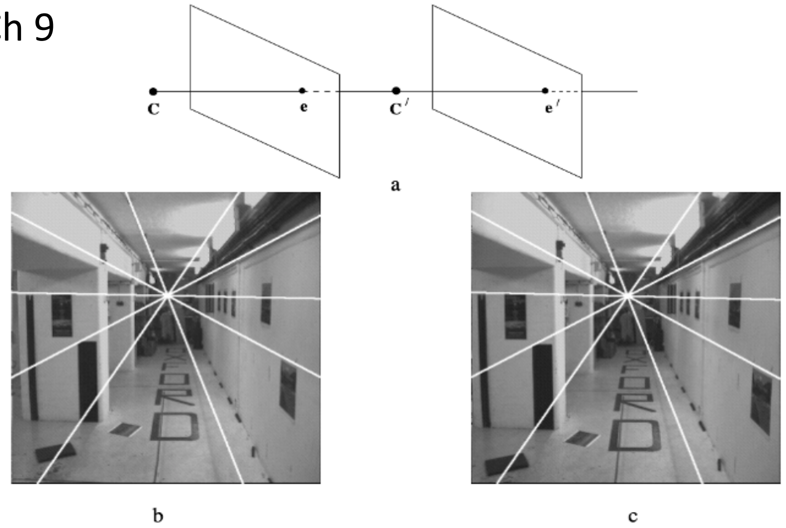


Fig. 9.8. **Pure translational motion.** *(a) under the motion the epipole is a fixed point, i.e. has the same coordinates in both images, and points appear to move along lines radiating from the epipole. The epipole in this case is termed the* Focus of Expansion *(FOE). (b) and (c) the same epipolar lines are overlaid in both cases. Note the motion of the posters on the wall which slide along the epipolar line.*

Rotational Flow

$$\dot{p} = \underbrace{\frac{1}{Z}\begin{bmatrix} xV_z - V_x \\ yV_z - V_y \end{bmatrix}}_{\text{translational flow}} + \underbrace{\begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\Omega}_{\text{rotational flow independent of depth}}$$

$$\dot{p} = \underbrace{\begin{bmatrix} xy & -(1+x^2) & y \\ (1+y^2) & -xy & -x \end{bmatrix}\Omega}_{\text{rotational flow independent of depth}}$$

Not just robots, animals know $\boldsymbol{\Omega}$ through the vestibular system (inner ear)!

If we know angular velocity $\boldsymbol{\Omega}$ (e.g. from IMU gyroscope) we can:

(1) compute optical flow $\dot{\boldsymbol{p}}$ from the images (e.g. with LK)

(2) then from $\boldsymbol{\Omega}$, estimate rotational flow $\dot{\boldsymbol{p}}_{\mathrm{rot}}$ at each pixel independent of the scene.

(3) then get $\dot{\boldsymbol{p}}_{\mathrm{trans}} = \dot{\boldsymbol{p}} - \dot{\boldsymbol{p}}_{\mathrm{rot}}$

What can we do knowing the rotational and translational flows separately in this way?

Turns out, we can efficiently find the camera velocity V (up to scale)
and also time to collision!

Finding FOE $\sim \boldsymbol{V}$ upto scale ambiguity

Remember, in SfM too, we only computed translation *to scale*!

- We said earlier, FOE = $\left[\frac{V_x}{V_z}, \frac{V_y}{V_z}\right] \in \mathbb{R}^2$
- In homogeneous $\mathbb{P}^2$ coordinates, we can write FOE as $\boldsymbol{V} \sim \left[V_x, V_y, V_z\right]$
- For point with known translational flow (we temporarily use the notation $\dot{\boldsymbol{p}}$ instead of $\dot{\boldsymbol{p}}_{\text{trans}}$), its "flow line" is: $p_1 \times (p_1 + \dot{p_1}) = p_1 \times \dot{p_1}$
- FOE is the intersection of all flow lines. So, $(p_1 \times \dot{p_1})^T V = 0$
- Given $n \geq 2$ points and flows, $\boldsymbol{V}$ lies on each flow line:

$$\underbrace{\begin{pmatrix} (p_1 \times \dot{p}_1)^T \\ p_2 \times \dot{p}_2)^T \\ \cdots \\ p_n \times \dot{p}_n)^T \end{pmatrix}}_{A} V = 0$$

We know how to find null vectors!

$V \leftarrow$ the smallest right singular vector of A!

So, given camera angular velocity $\boldsymbol{\Omega}$, we can compute camera velocity $\boldsymbol{V}$ (to scale)

Next, Finding Time-To-Collision (TTC)

Inverse "time to collision"

$$\dot{\boldsymbol{p}}_{\text{trans}} = \begin{bmatrix}\dot{x}\\ \dot{y}\end{bmatrix} = \boxed{\frac{V_z}{Z}} \begin{bmatrix} x - \boxed{\frac{V_x}{V_z}} \\ y - \boxed{\frac{V_y}{V_z}} \end{bmatrix}$$

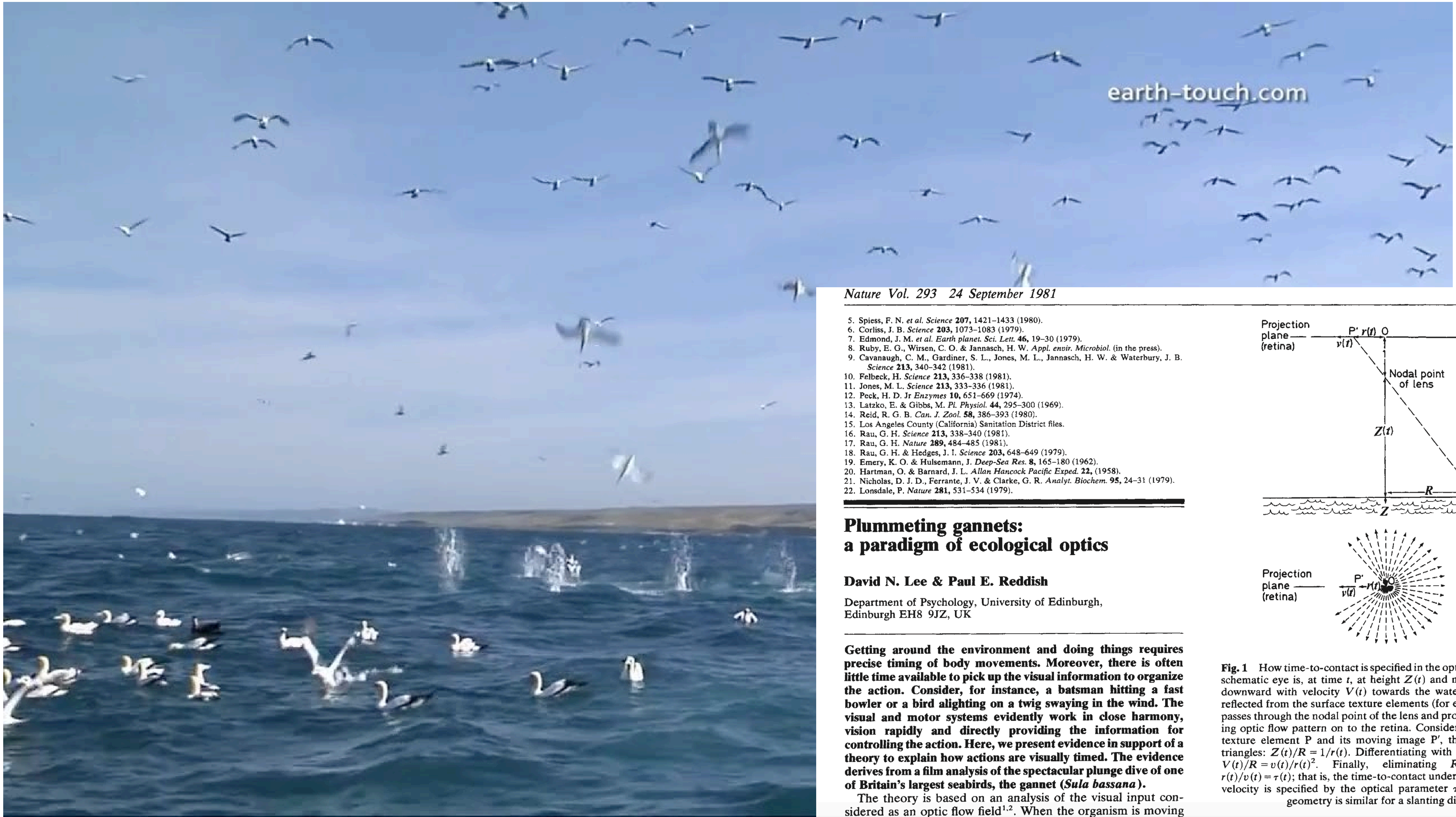
Focus of expansion (FOE)

Having computed the FOE, we can compute:

Get inverse TTC: $\frac{V_z}{Z} = \frac{||\dot{p}_{\text{trans}}||}{||p - FOE||}$

Animals do this!

Time to Collision (TTC)



5. Spiess, F. N. *et al. Science* **207,** 1421–1433 (1980).
6. Corliss, J. B. *Science* **203,** 1073–1083 (1979).
7. Edmond, J. M. *et al. Earth planet. Sci. Lett.* **46,** 19–30 (1979).
8. Ruby, E. G., Wirsen, C. O. & Jannasch, H. W. *Appl. envir. Microbiol.* (in the press).
9. Cavanaugh, C. M., Gardiner, S. L., Jones, M. L., Jannasch, H. W. & Waterbury, J. B. *Science* **213,** 340–342 (1981).
10. Felbeck, H. *Science* **213,** 336–338 (1981).
11. Jones, M. L. *Science* **213,** 333–336 (1981).
12. Peck, H. D. Jr *Enzymes* **10,** 651–669 (1974).
13. Latzko, E. & Gibbs, M. *Pl. Physiol.* **44,** 295–300 (1969).
14. Reid, R. G. B. *Can. J. Zool.* **58,** 386–393 (1980).
15. Los Angeles County (California) Sanitation District files.
16. Rau, G. H. *Science* **213,** 338–340 (1981).
17. Rau, G. H. *Nature* **289,** 484–485 (1981).
18. Rau, G. H. & Hedges, J. I. *Science* **203,** 648–649 (1979).
19. Emery, K. O. & Hulsemann, J. *Deep-Sea Res.* **8,** 165–180 (1962).
20. Hartman, O. & Barnard, J. L. *Allan Hancock Pacific Exped.* **22,** (1958).
21. Nicholas, D. J. D., Ferrante, J. V. & Clarke, G. R. *Analyt. Biochem.* **95,** 24–31 (1979).
22. Lonsdale, P. *Nature* **281,** 531–534 (1979).

Plummeting gannets: a paradigm of ecological optics

David N. Lee & Paul E. Reddish

Department of Psychology, University of Edinburgh, Edinburgh EH8 9JZ, UK

Getting around the environment and doing things requires precise timing of body movements. Moreover, there is often little time available to pick up the visual information to organize the action. Consider, for instance, a batsman hitting a fast bowler or a bird alighting on a twig swaying in the wind. The visual and motor systems evidently work in close harmony, vision rapidly and directly providing the information for controlling the action. Here, we present evidence in support of a theory to explain how actions are visually timed. The evidence derives from a film analysis of the spectacular plunge dive of one of Britain's largest seabirds, the gannet (*Sula bassana*).

The theory is based on an analysis of the visual input considered as an optic flow field[1,2]. When the organism is moving

Fig. 1 How time-to-contact is specified in the optic flow field. The schematic eye is, at time t, at height $Z(t)$ and moving vertically downward with velocity $V(t)$ towards the water surface. Light reflected from the surface texture elements (for example, ripples) passes through the nodal point of the lens and projects an expanding optic flow pattern on to the retina. Considering an arbitrary texture element P and its moving image P′, then from similar triangles: $Z(t)/R = 1/r(t)$. Differentiating with respect to time: $V(t)/R = v(t)/r(t)^2$. Finally, eliminating R, $Z(t)/V(t) = r(t)/v(t) = \tau(t)$; that is, the time-to-contact under constant closing velocity is specified by the optical parameter $\tau(t)$. The optical geometry is similar for a slanting dive.

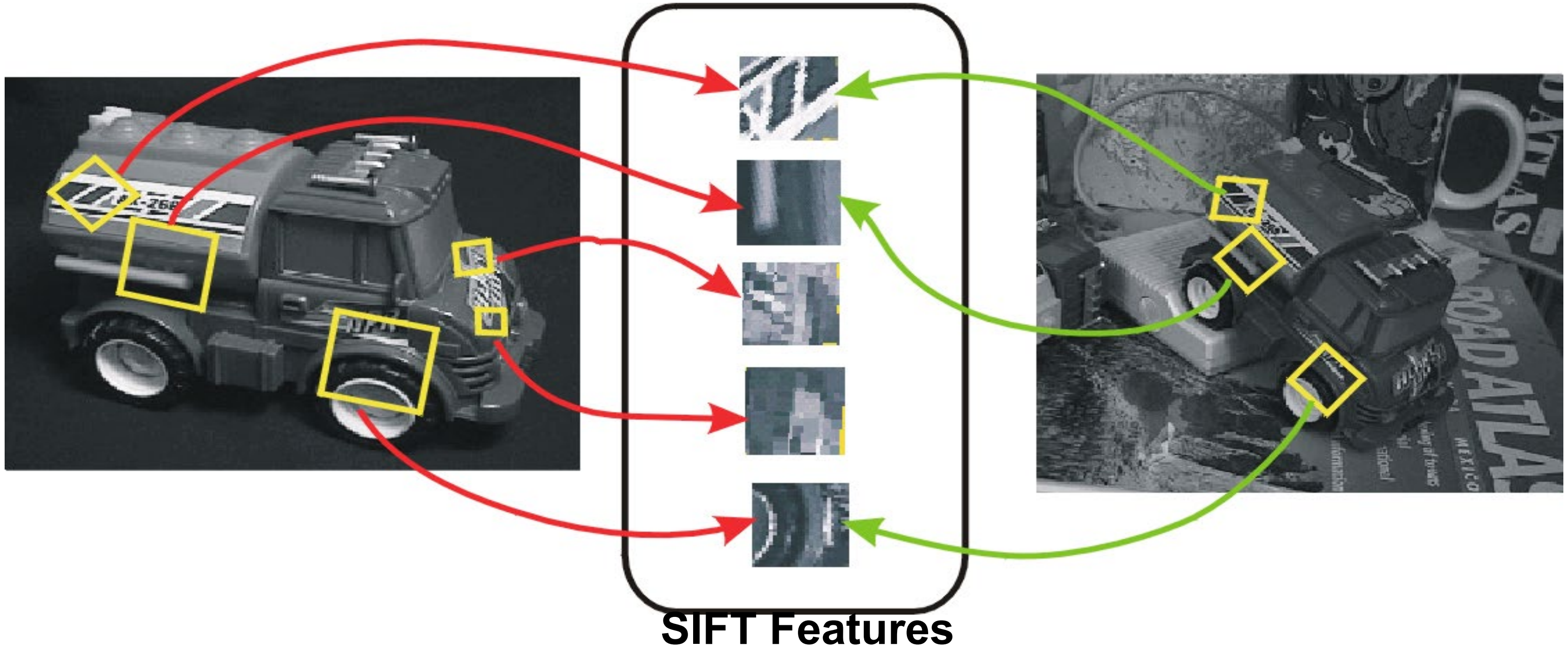
We know 2-view SfM can compute motion and depths from optical flow given only (5) point correspondences

We’ve seen how to efficiently compute motion from optical flow, if Ω known, plus (2) point correspondences.

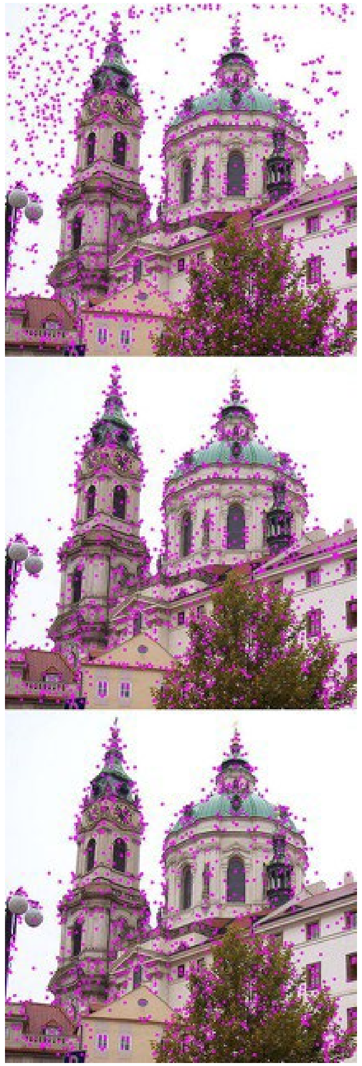
SIFT (Scale Invariant Feature Transform)

Motivation of SIFT

- Image content is transformed into local feature coordinates that are invariant to translation, rotation, scale, and other imaging parameters



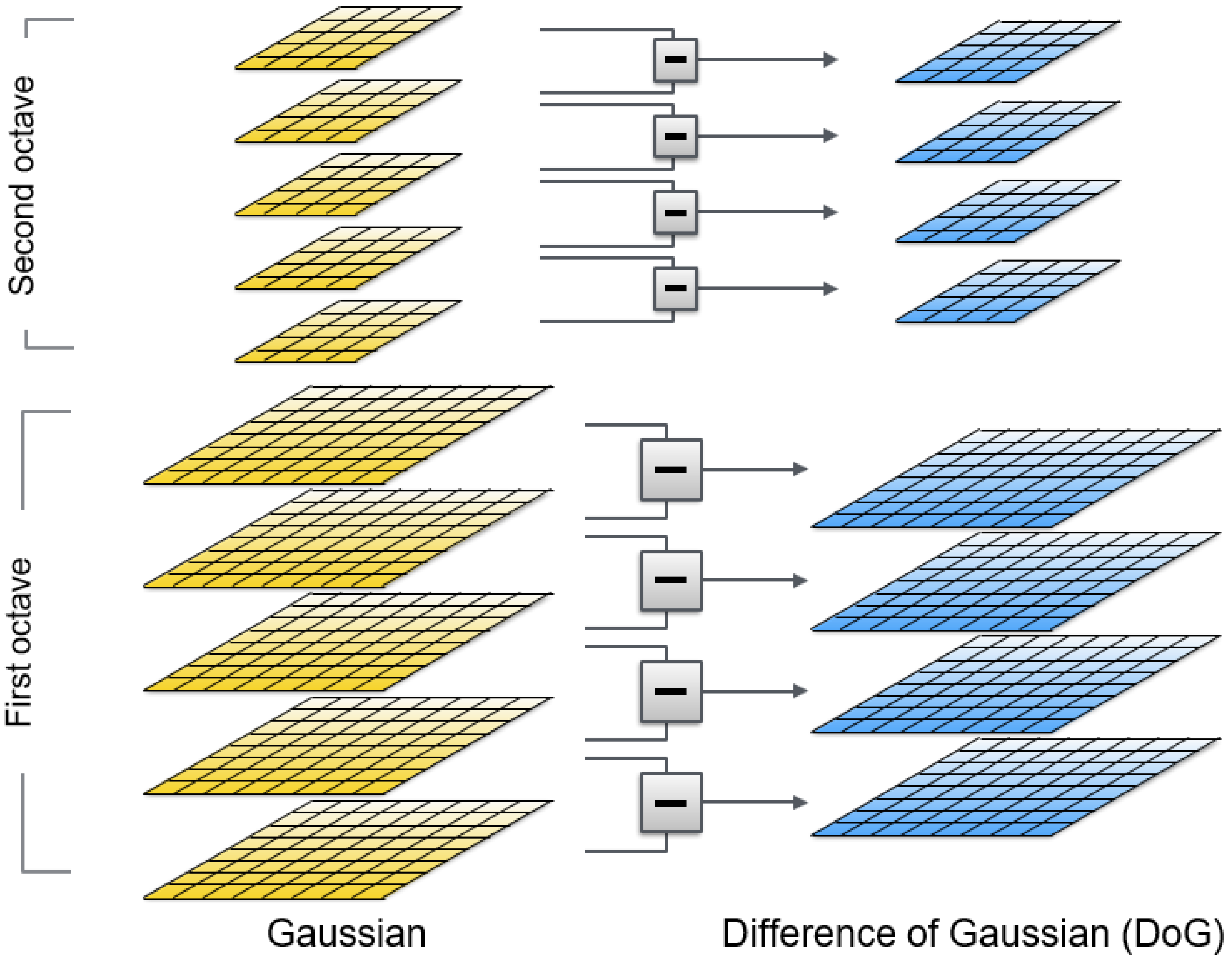
What is SIFT (Scale Invariant Feature Transform)



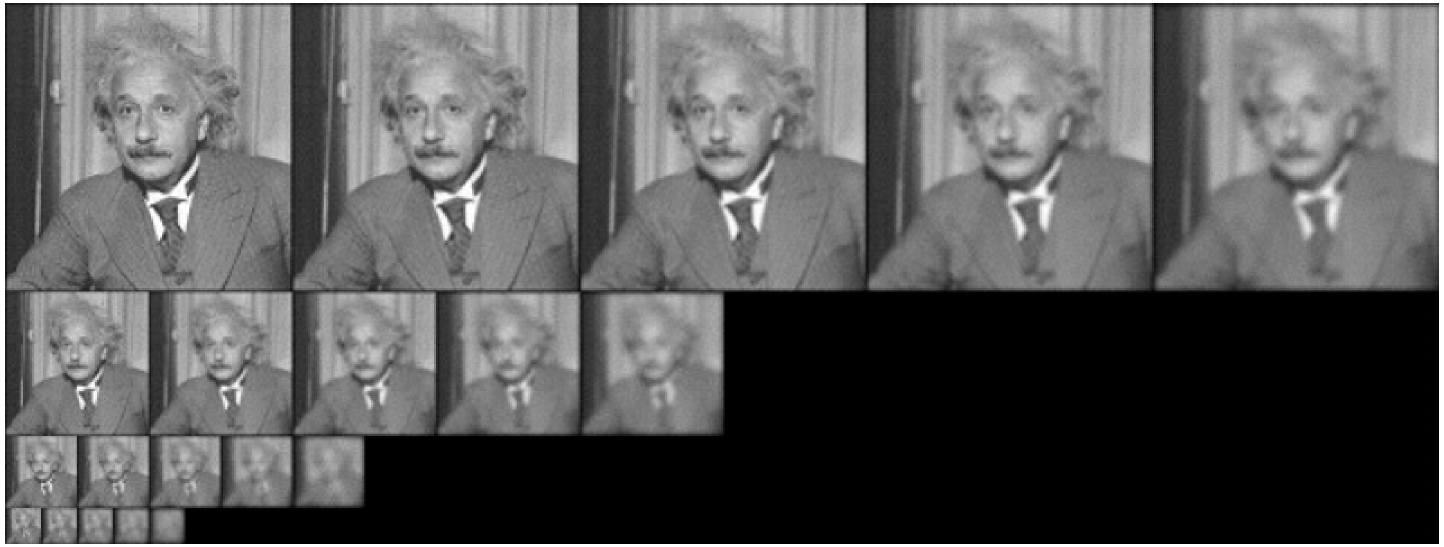
- SIFT describes both a **detector** and **descriptor**

1. Multi-scale extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

1. Multi-scale extrema detection



1. Multi-scale extrema detection

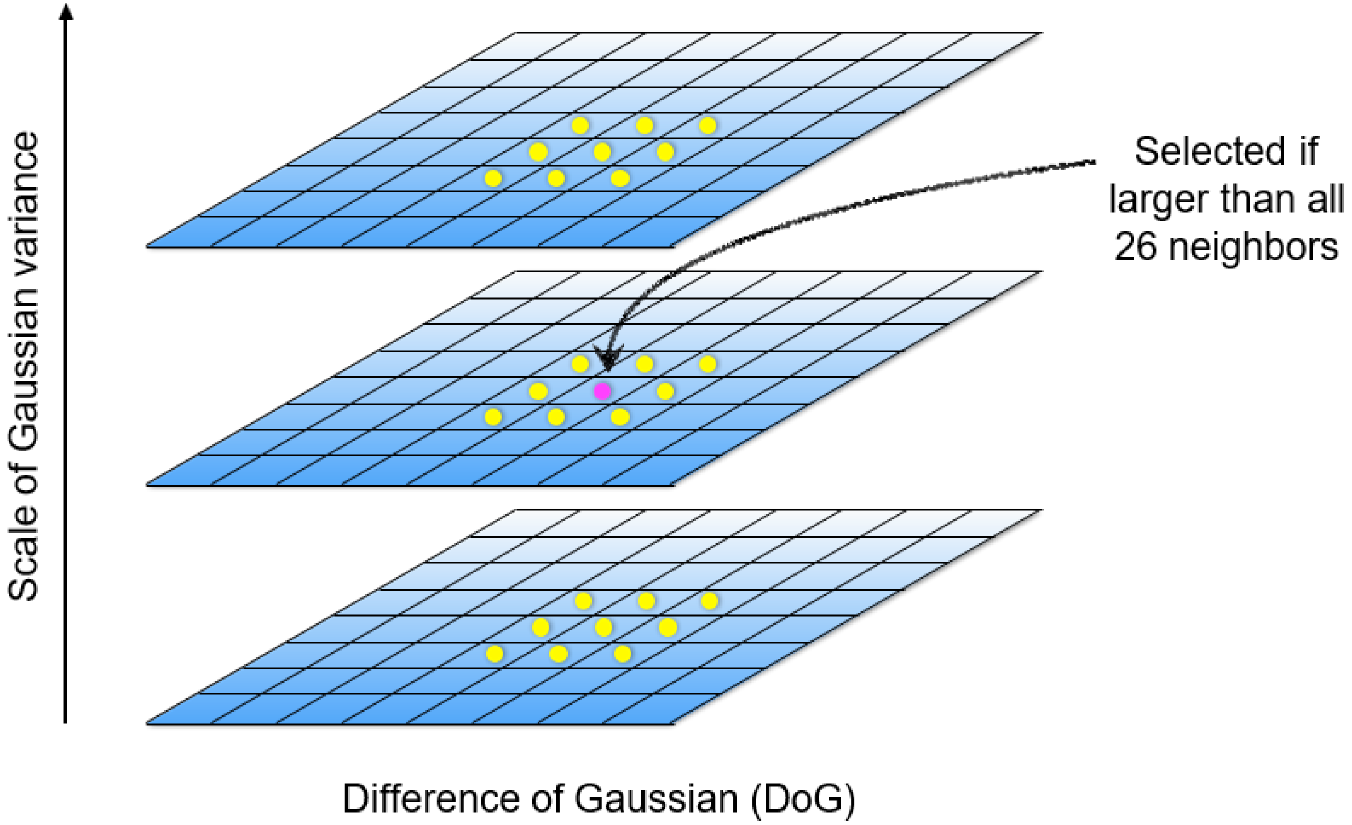


Gaussian

Laplacian

1. Multi-scale extrema detection

Scale-space extrema



2. Keypoint Localization

2nd order Taylor series approximation of DoG scale-space

$$f(\mathbf{x}) = f + \frac{\partial f}{\partial \mathbf{x}}^T \mathbf{x} + \frac{1}{2}\mathbf{x}^T \frac{\partial^2 f}{\partial \mathbf{x}^2}\mathbf{x}$$

$$\mathbf{x} = \{x, y, \sigma\}$$

Take the derivative and solve for extrema

$$\mathbf{x}_m = -\frac{\partial^2 f}{\partial \mathbf{x}^2}^{-1} \frac{\partial f}{\partial \mathbf{x}}$$

Additional tests to retain only strong features

3. Orientation assignment

For a keypoint, **L** is the **Gaussian-smoothed** image with the closest scale,

$$m(x,y) = \sqrt{\underbrace{(L(x+1,y) - L(x-1,y))^2}_{\text{x-derivative}} + \underbrace{(L(x,y+1) - L(x,y-1))^2}_{\text{y-derivative}}}$$

$$\theta(x,y) = \tan^{-1}((L(x,y+1) - L(x,y-1))/(L(x+1,y) - L(x-1,y)))$$

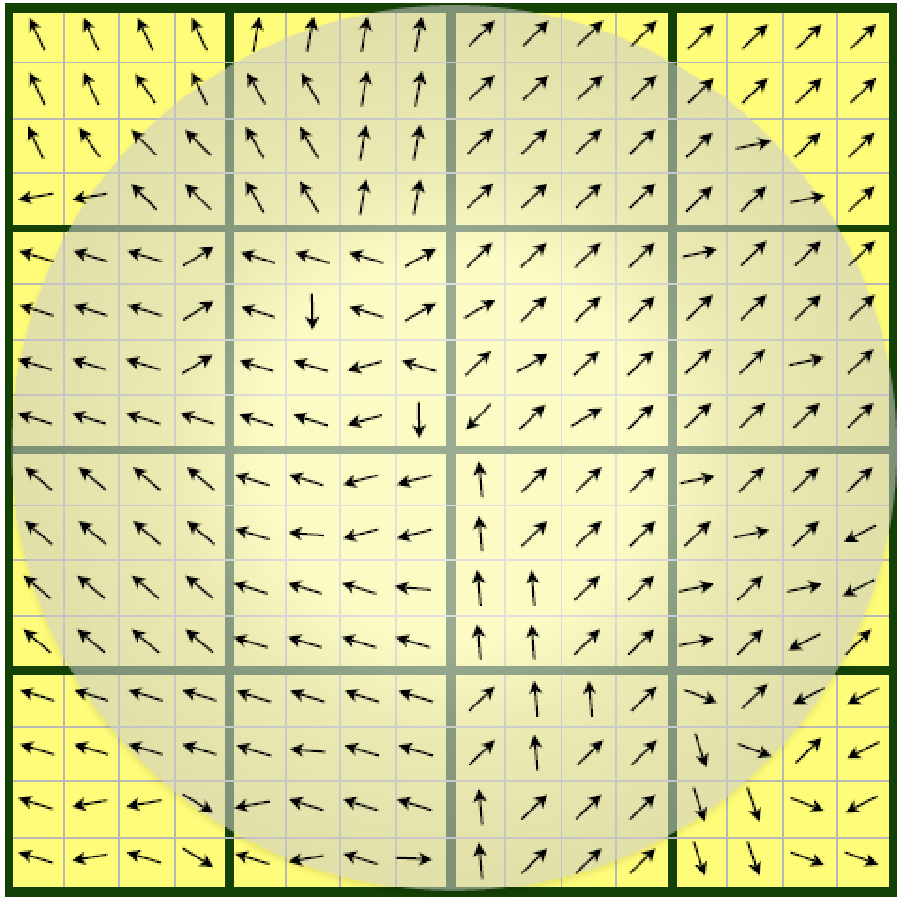
Detection process returns

$$\{\underbrace{x, y}_{\text{location}}, \underbrace{\sigma}_{\text{scale}}, \underbrace{\theta}_{\text{orientation}}\}$$

4. Keypoint Descriptor

Image Gradients

(4 x 4 pixel per cell, 4 x 4 cells)

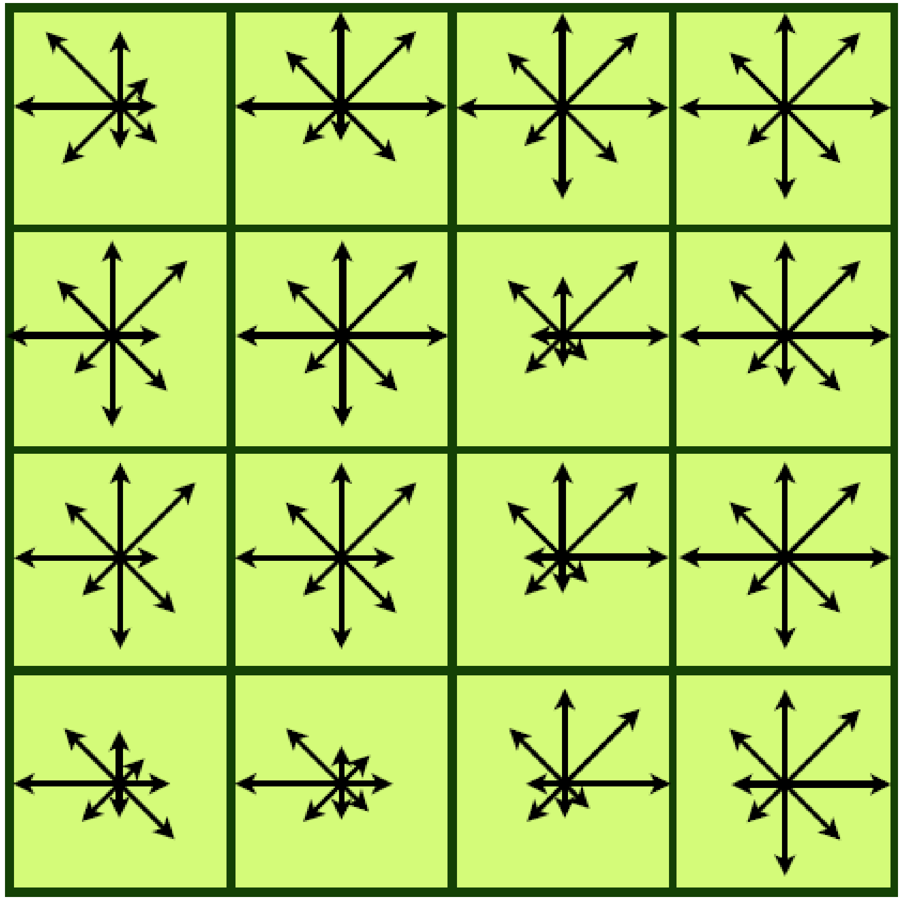


Gaussian weighting

(sigma = half width)

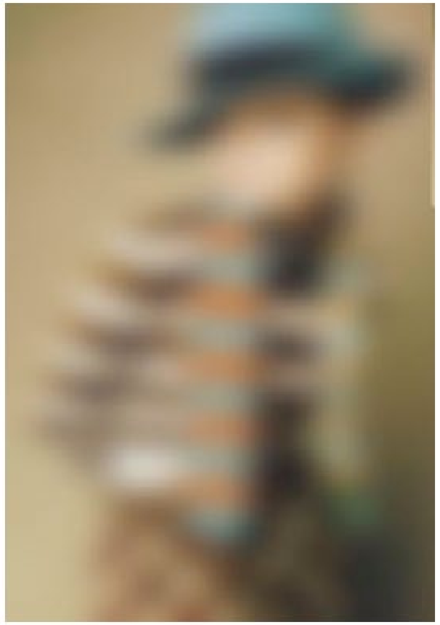
SIFT descriptor

(16 cells x 8 directions = 128 dims)

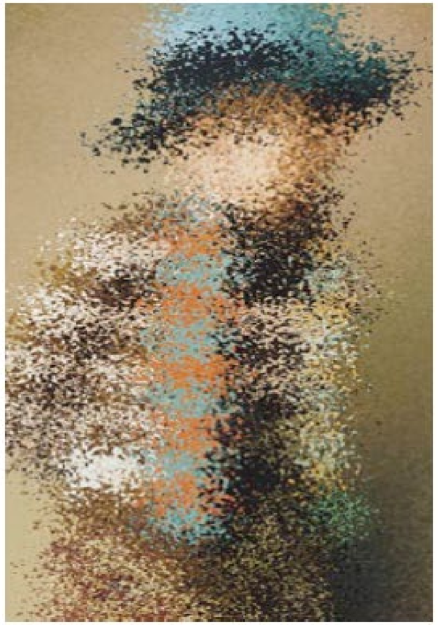


Discriminative power

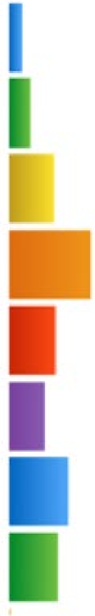
Raw pixels



Sampled



Locally orderless



Global histogram

Generalization power

Summary: Methods for Computing Motion

- Given flow (or other) pointwise correspondences between nearby images,
  - Plus Ω, can solve for $FOE \sim V$ with just 2 2D-2D correspondences
  - Plus depths, we can solve V and Ω with just 3 2D-2D correspondences
  - Alone, we can solve for V and Ω with 5 correspondences (SfM)
  - Known 3D scene, we can solve for single-frame camera pose with 3 2D-3D correspondences (PnP)

Taking Stock Of 2-View Geometry: What We've Learned

- Given 2D point correspondences between 2 views:
  - SfM: Finding 3D structure and camera motion (8-point algorithm)
  - Finding homographies between views and building panoramas
- Given 2D point correspondences + camera rotation, find translation
- Given 2D point correspondences + depth, find rotation + translation
- Given camera pose and 2D point correspondences:
  - Triangulation to find structure (used, e.g., in motion capture systems)
- Robustness to noisy correspondences:
  - Hough Transforms
  - RANSAC
- Coming up next, extending SfM to > 2 views:
  - The Incremental Approach, through SLAM / odometry. ORB-SLAM
  - The Global Approach i.e. Bundle Adjustment