Course Review

Week 12

Announcements

- **Drop-in Sessions:** Please bring your assignment/exam questions to the following locations
  - 14:00-16:00 Wednesday **22 May**, 1.23 Hanna Neumann (Hoang)
  - 15:00-17:00 Friday **24 May**, 1.23 Hanna Neumann (Yiran)
  - 09:00-10:30 Friday **24 May**, Melville Hall (Dylan)

Semester 1 SELT is live!

20 May – Survey opens
Check your email or Wattle page for available surveys

21 – 24 May
Food and prizes just for completing your SELT!

Survey runs for 4 weeks
Please provide constructive and respectful feedback *(your teacher can't identify you)*

16 June - Survey closes
IR team perform screening of comments for welfare concerns

8 July
SELT feedback is made available to teachers and course convenors to improve future course delivery

Join us for

free food and prizes

21 – 24 May

11.00am – 2.00pm

CSIT Lab entrance or lawn

(weather dependent)

Australian National University

Semester 1 SELT - survey journey

The Student Experience of Learning & Teaching survey allows students to give feedback on their courses and teachers. It is **voluntary** and **confidential**, and run by the Institutional Research (IR) team.

20 May – Survey opens
Check your email or Wattle page for available surveys

Survey runs for 4 weeks
Please provide constructive and respectful feedback *(your teacher can't identify you)*

16 June - Survey closes
IR team perform screening of comments for welfare concerns

27 June
Grades are released to students

8 July
SELT feedback is made available to teachers and course convenors to improve future course delivery

Australian National University

Find out more on the *Info for Students* webpage:

https://services.anu.edu.au/learning-teaching/education-data/student-experience-of-learning-teaching-selt/information-for

SELT - Frequently asked questions

What kind of feedback is helpful?

Think about your experience of the course and teaching, and what worked or didn't work for you.

When writing feedback, focus on respectful and constructive language – if you were a teacher, what type of feedback would help you improve the class?

Can teachers see who left specific feedback?

SELT is confidential, and teachers cannot see, or ask to see, the identity of a respondent. Unless you self-identify, for example by using names or describing specific events, teachers cannot identify you.

Australian National University

Find out more on the *Info for Students* webpage:

https://services.anu.edu.au/learning-teaching/education-data/student-experience-of-learning-teaching-selt/information-for

Weekly Study Plan: Overview

Wk	Starting	Lecture	Lab	Assessment
1	19 Feb	Introduction	✗	
2	26 Feb	Low-level Vision 1	1	
3	4 Mar	Low-level Vision 2	1	
		Mid-level Vision 1		
4	11 Mar	Mid-level Vision 2	1	CLab1 report due Friday
		High-level Vision 1		
5	18 Mar	High-level Vision 2	2	
6	25 Mar	High-level Vision 3[1]	2	
	1 Apr	Teaching break	✗	
	8 Apr	Teaching break	✗	
7	15 Apr	3D Vision 1	2	CLab2 report due Friday
8	22 Apr	3D Vision 2	3	
9	29 Apr	3D Vision 3	3	
10	6 May	3D Vision 4	3	
		Mid-level Vision 3		
11	13 May	High-level Vision 4	✗	CLab3 report due Friday
12	20 May	Course Review	✗	

Weekly Study Plan: Part B

Wk	Starting	Lecture	By
7	15 Apr	3D vision: introduction, camera model, single-view geometry	Dylan
8	22 Apr	3D vision: camera calibration, two-view geometry (homography)	Dylan
9	29 Apr	3D vision: two-view geometry (epipolar geometry, triangulation, stereo)	Dylan
10	6 May	3D vision: multiple-view geometry	Weijian
		Mid-level vision: optical flow, shape-from-X	Dylan
11	13 May	High-level vision: self-supervised learning, detection, segmentation	Dylan
12	20 May	Course review	Dylan

Outline

1. Final Exam Details
2. Course Review

Final Exam

Final Exam: Logistics

- **Time:** 9:00am–12:15pm
- **Date:** Saturday 1 June 2024
- **Location:** 7-11 Barry Drive, First Floor Left Side
  - Check! Some of you are in different locations
- **Duration:**
  - Reading Time: 15 minutes
  - Writing Time: 3 hours
- **Worth 55%**
- **Not** a hurdle
- **Permitted materials:**
  - Calculator (non-programmable)
  - One A4 page with notes on both sides
- **Supplied materials:**
  - Exam paper
  - 20-page booklet
  - Scribble paper

Final Exam: Instructions

- There are 8 questions.
- Specify which question you are answering by putting the **question number at the top of the page**.
- **Each question** must begin on a **new page**.
  - Multi-part questions (e.g., question 1, parts a and b) may be answered on the same page but should be clearly labelled (e.g., 1a, 1b).
- Ensure your answers (text, equations, and diagrams) are legible.
- Some questions are open-ended; if you think that some aspect is ambiguous, **state your assumptions** clearly and answer with respect to those assumptions. Better responses will consider the implications of different assumptions.

Final Exam: Scope and Breakdown

- **Scope:** Material from all lectures (Weeks 1–11)
- **Content Proportions:**
  - Multiple choice (mixture of topics): ~20%
  - Low- and mid-level vision (image formation, representation, processing, filtering, image features, optical flow, shape-from-X): ~15%
  - High-level vision (neural networks, classification, detection, segmentation): ~15%
  - 3D vision (single-view, two-view, multiple-view geometry, shape-from-X): ~50%

Final Exam: Types of Questions

- **Question Types:**
  - Multiple choice:
    - There is at least one correct answer for each question.
    - Any correct choices will incur partial positive marks; any incorrect choices will incur partial negative marks of the same value.
    - For example, if A, B and C are correct and the response is A, B and D, then 33% of the total marks will be awarded.
  - Short response questions
  - Calculation questions
  - Algorithm analysis questions
  - Algorithm design questions

Final Exam: Difficulty

- Difficulty level:
  - Similar to previous years (modulo subjectivity)
- Difference between UG (4528) and PG (6528):
  - An extended, advanced question for Masters students

Final Exam: How do I Prepare for the Exam?

- Review:
  - Lectures (verbal and written)
  - Tutorial sheets
  - Assignments
  - Textbooks chapters for more details if confused (Szeliski, Hartley & Zisserman)
- Practice:
  - Past papers, practice questions, tutorial sheets
- Some exam questions will be familiar to you if you have done the past papers

Final Exam: General Advice

- Be strategic:
  - Focus on the questions with the most marks if others are taking too long
  - Focus on the questions that you are sure about before tackling others
  - Not everyone will answer every question: this is expected
- Show your working:
  - Before doing a calculation, write out the equation
  - Then substitute in the numeric values
  - Then perform the computation
  - You can get marks even if you slip up with a matrix-vector multiplication
- Similarly for short answer questions: explain your response

Final Exam: General Advice

- Be concise: think, then write – few questions require more than a couple of sentences
- Make connections: not everything is directly from the lecture notes, some questions require you to synthesize two different ideas, or extrapolate (slightly) beyond what you have seen already
- Keep calm and carry on

Course Review

Outline

1. Convolution and Filters
2. Image Features
3. Neural Networks
4. Model Fitting
5. Single-view Geometry
6. Two-View Geometry

Weeks 1–6

- Understand the process of histogram modification & equalisation
- Geometric image formation, how to represent geometric transformations in matrix multiplication form
- Image filtering, understand different types of filters, the effect after applying different filters
  - Understand the difference between linear and non-linear filters
  - Basic calculations for the filtering process
- Understand filters for edge detection and bilateral filtering
- Understand the corner response function & Harris corner detector
  - Understand why the Harris corner detector is rotation invariant, why the SIFT detector and descriptor is scale and rotation invariant and how SIFT is used in different applications

Weeks 1–6

- Understand neural networks:
  - The forward pass (calculation)
  - The backward pass (building the computation graph, backpropagation)
- Understand how to calculate the receptive field of the convolution operation at different layers, with pooling, with different strides
- Understand techniques of Dropout and Batch Normalisation in the neural network
- Understand basic techniques for domain adaptation

Convolution

- Flip filter (bottom-to-top and right-to-left), then apply correlation

Correlation:

$$G[x,y] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[x+u, y+v]$$

$$G = H \otimes F$$

Convolution:

$$G[x,y] = \sum_{u=-k}^{k} \sum_{v=-k}^{k} H[u,v]F[x-u, y-v]$$

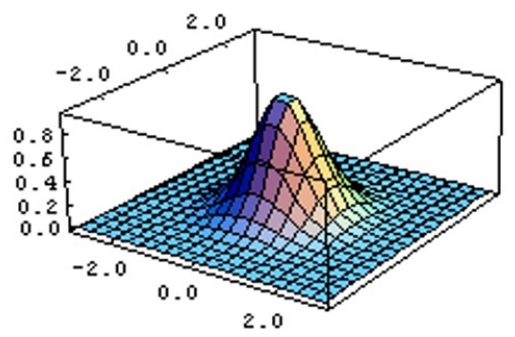
$$\boxed{G = H \star F}$$

Simple Filters

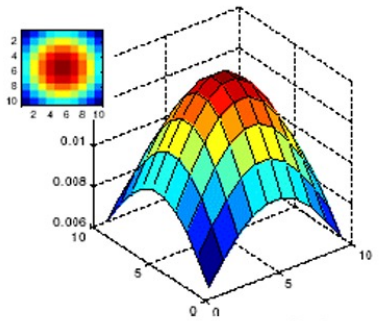
- Box filter: replace pixel by the average of its neighbours (window)
- Gaussian filter: blurs the image, nearest pixels have most influence

• Kernel: approximation of a 2d Gaussian function:

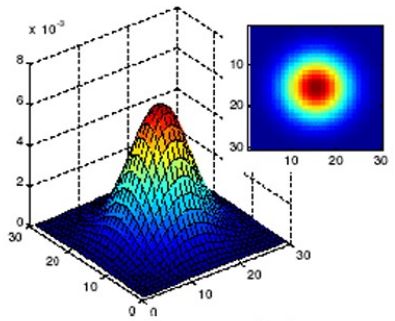
$$h(u,v) = \frac{1}{2\pi\sigma^2} \exp^{-\frac{u^2+v^2}{2\sigma^2}}$$



• Two factors for Kernel: size and variance



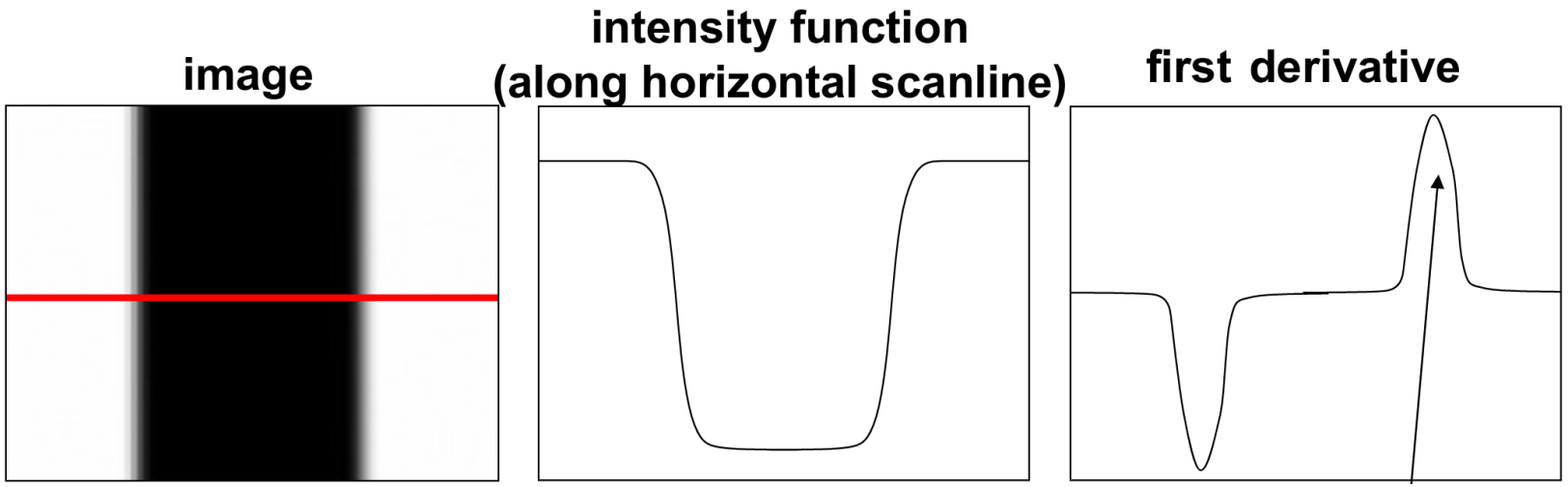
σ = 5 with 10x 10 kernel



σ = 5 with 30 x 30 kernel

Edge Detection

- **Edge:** a place of rapid change in the image intensity function
- **Methods:**
  - First-order derivative: search for extrema, e.g., Sobel operator, Derivative-of-Gaussian



Edge Detection

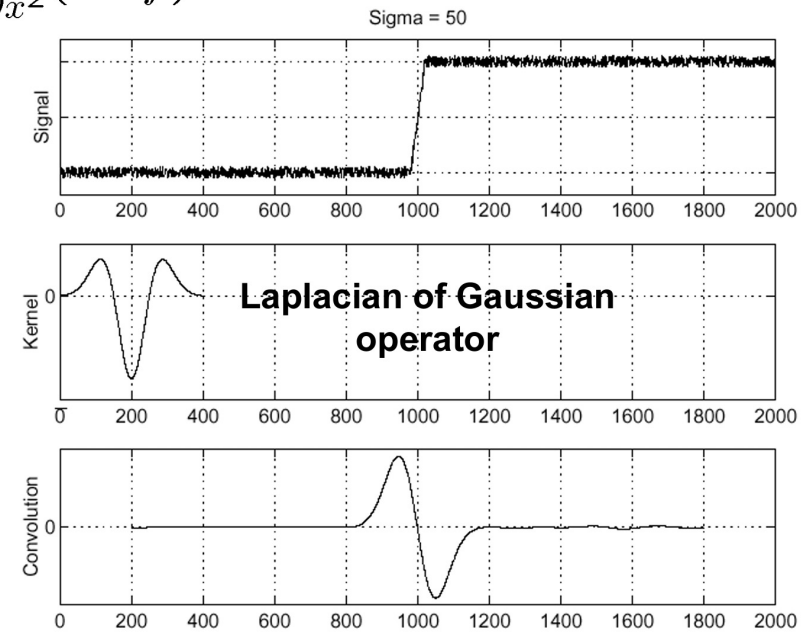
- **Edge:** a place of rapid change in the image intensity function
- **Methods:**
  - Second-order derivative: search for zero-crossings, e.g., Laplacian, Laplacian-of-Gaussian

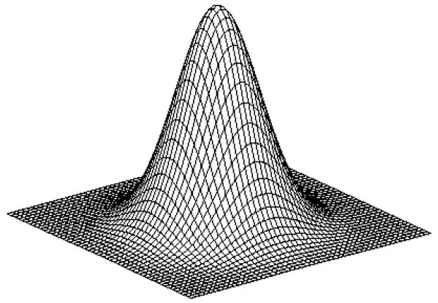
Consider $\frac{\partial^2}{\partial x^2}(h \star f)$

f

$\frac{\partial^2}{\partial x^2}h$

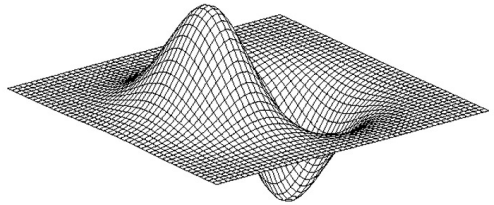
$(\frac{\partial^2}{\partial x^2}h) \star f$





Gaussian

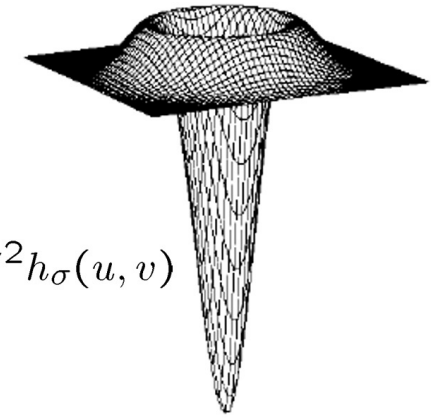
$$h_\sigma(u,v) = \frac{1}{2\pi\sigma^2}e^{-\frac{u^2+v^2}{2\sigma^2}}$$



derivative of Gaussian

$$\frac{\partial}{\partial x}h_\sigma(u,v)$$

Laplacian of Gaussian



$$\nabla^2 h_\sigma(u,v)$$

Nonlinear Filters

- Median filter: replaces centre pixel by median within window
  - Good for removing salt-and-pepper (impulse) noise
- Bilateral filter: smooth while preserving edges
  - Domain filter & range filter

$$\mathbf{g}(i,j) = \frac{\sum_{k,l} \mathbf{f}(k,l) w(i,j,k,l)}{\sum_{k,l} w(i,j,k,l)}$$

$$d(i,j,k,l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2}\right)$$

$$r(i,j,k,l) = \exp\left(-\frac{\|\mathbf{f}(i,j) - \mathbf{f}(k,l)\|^2}{2\sigma_r^2}\right)$$

$$w(i,j,k,l) = \exp\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{\|\mathbf{f}(i,j) - \mathbf{f}(k,l)\|^2}{2\sigma_r^2}\right)$$

$$BF[I]_{\mathbf{p}} = \frac{1}{W_{\mathbf{p}}} \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)\, G_{\sigma_r}(|I_{\mathbf{p}} - I_{\mathbf{q}}|)\, I_{\mathbf{q}}$$

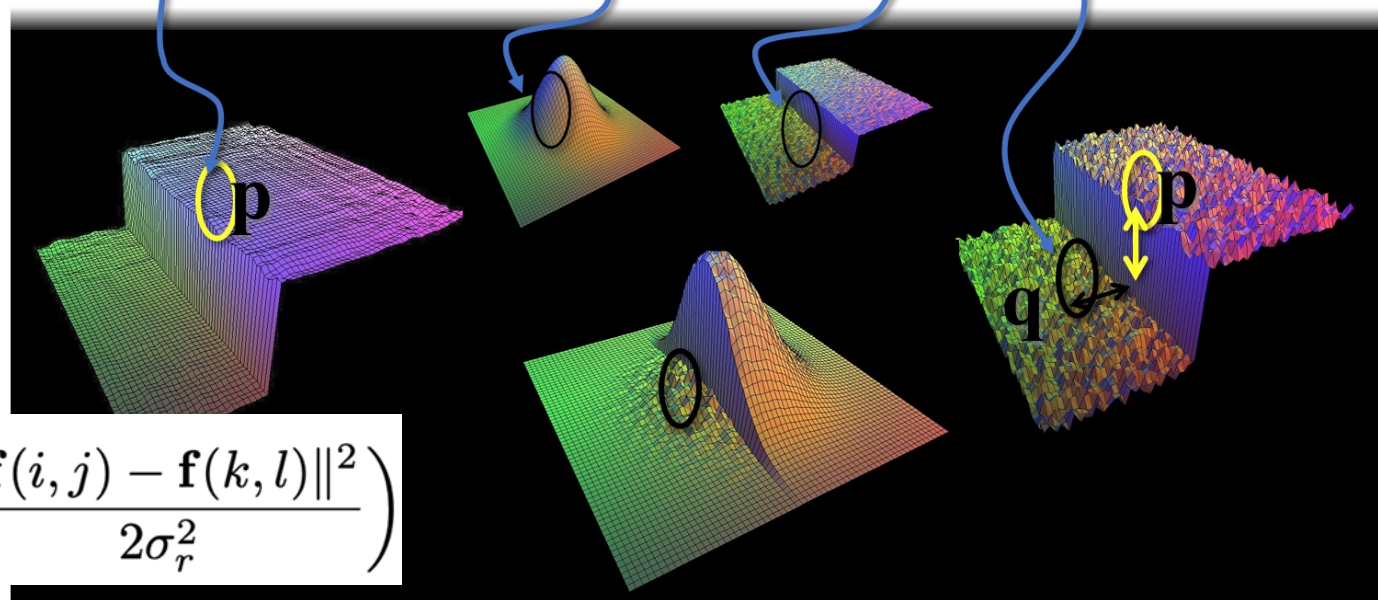
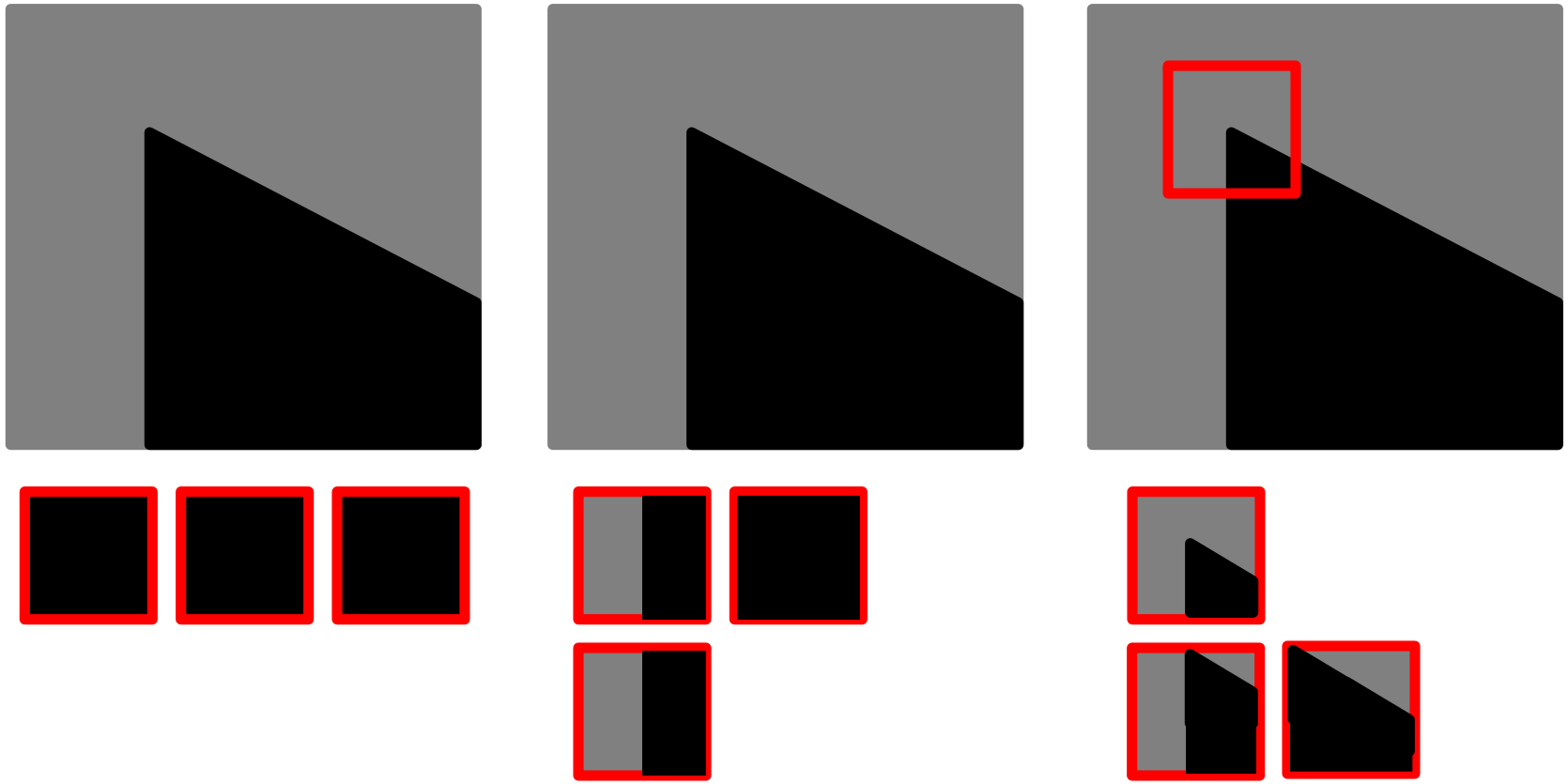


Image Features

- Harris corners
- SIFT features

Corner Detector



flat

edge

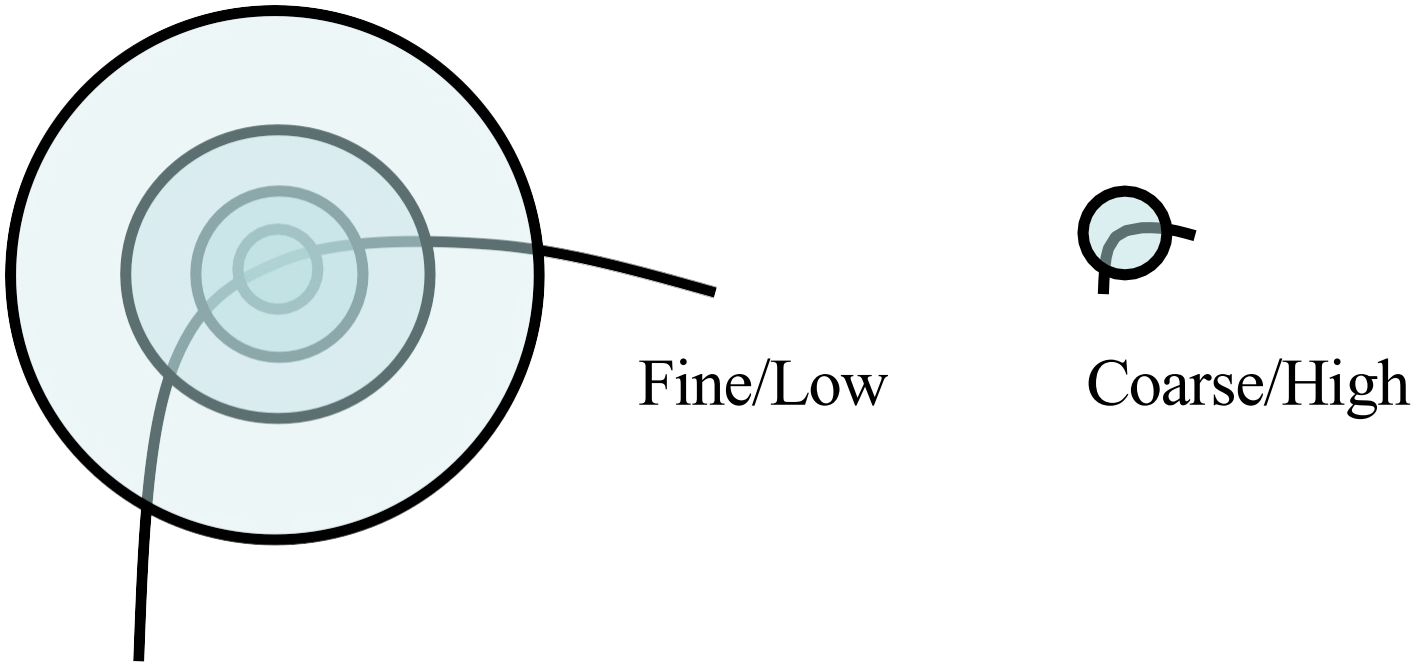
corner
isolated point

Harris Corner

- Compute the moment (autocorrelation) matrix M
  - Captures the structure of the local neighbourhood
  - Measure based on eigenvalues of M
    - 2 strong eigenvalues => interest point
    - 1 strong eigenvalue => edge or contour
    - 0 or very weak eigenvalues => flat region
- Corner strength: $R = \det M - k\mathrm{Tr}(M)^2$
- Interest point detection
  - threshold on the eigenvalues
  - local maximum for localisation

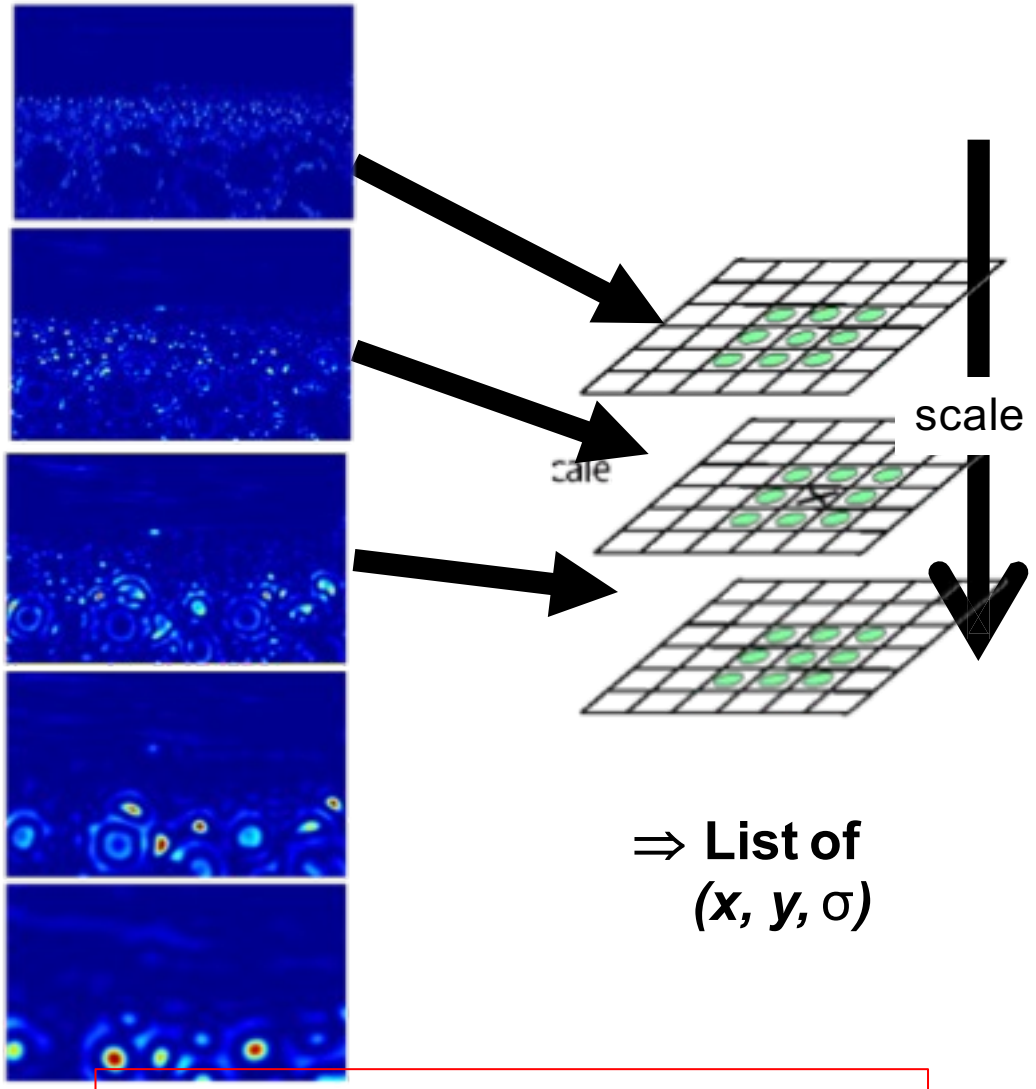
SIFT for Scale Invariant Detection

- Consider regions (e.g., circles) of different scales around a point
- Regions of corresponding sizes (at different scales) will look the same in both images



SIFT

$$L_{xx}(\sigma) + L_{yy}(\sigma)$$



- Interest points are local maxima in both position and scale

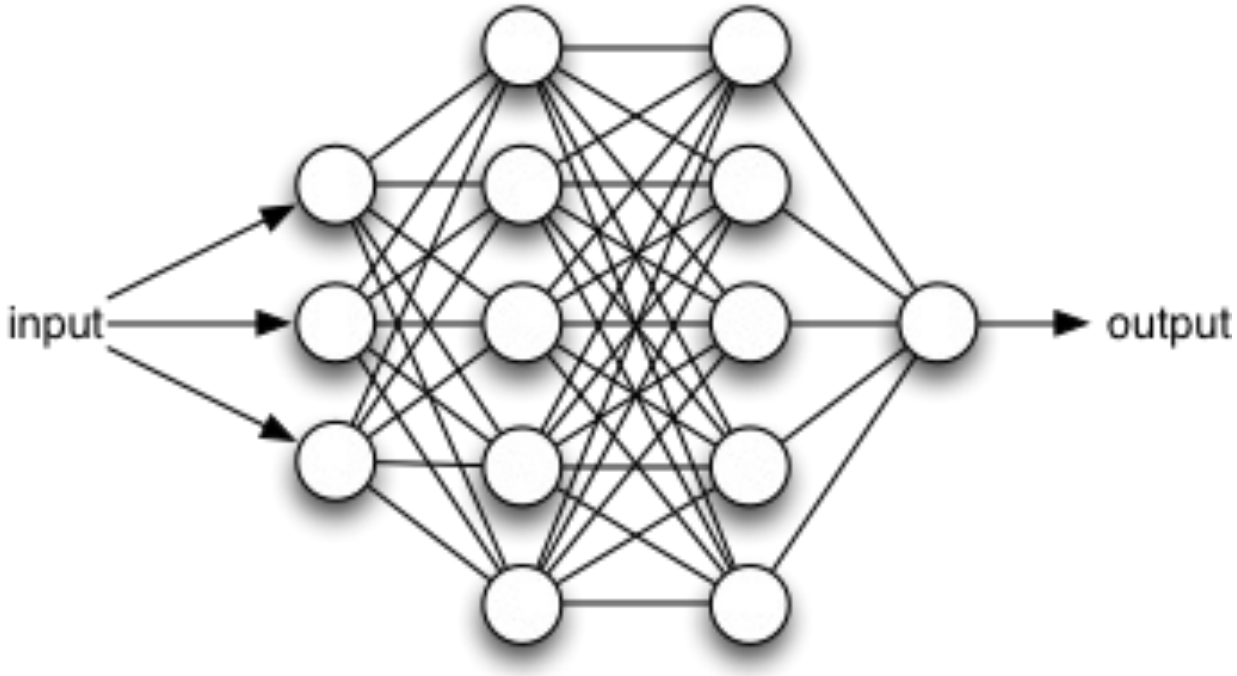
Squared filter response maps

⇒ **List of** ***(x, y,*** σ***)***

Difference of Gaussians in space and scale

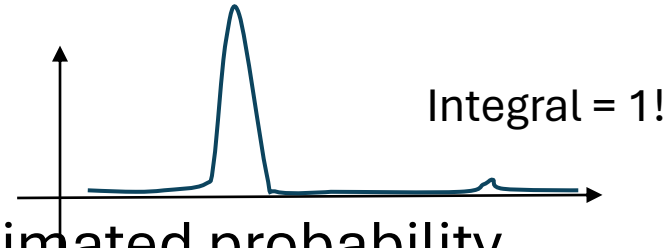
Multi-Layer Perceptron (MLP)

- A fully-connected neural network with non-linear activation functions



Multi-class Classification

- We need multiple outputs (1 output per class)
- We would like to estimate the conditional probabilities $p(\mathrm{y}=\mathrm{c}|\mathrm{x})$
  - → Softmax activation function at the output
  - $\mathbf{o}(\mathbf{a}) = \mathrm{softmax}(\mathbf{a}) = \left[\frac{\exp(a_1)}{\sum_c \exp(a_c)}, \cdots, \frac{\exp(a_C)}{\sum_c \exp(a_c)}\right]^\top$
  - Strictly positive
  - Sums up to one
  - Predicted class is the one with highest estimated probability



Multilayer Neural Network

- Could have *L* hidden layers:
  - Layer pre-activation for k>0:

$$\mathbf{a}^{(k)}(\mathbf{x}) = \mathbf{b}^{(k)} + \mathbf{W}^{(k)}\mathbf{h}^{(k-1)}(\mathbf{x})$$

$$\left(\mathbf{h}^{(0)}(\mathrm{x}) = \mathbf{x}\right)$$

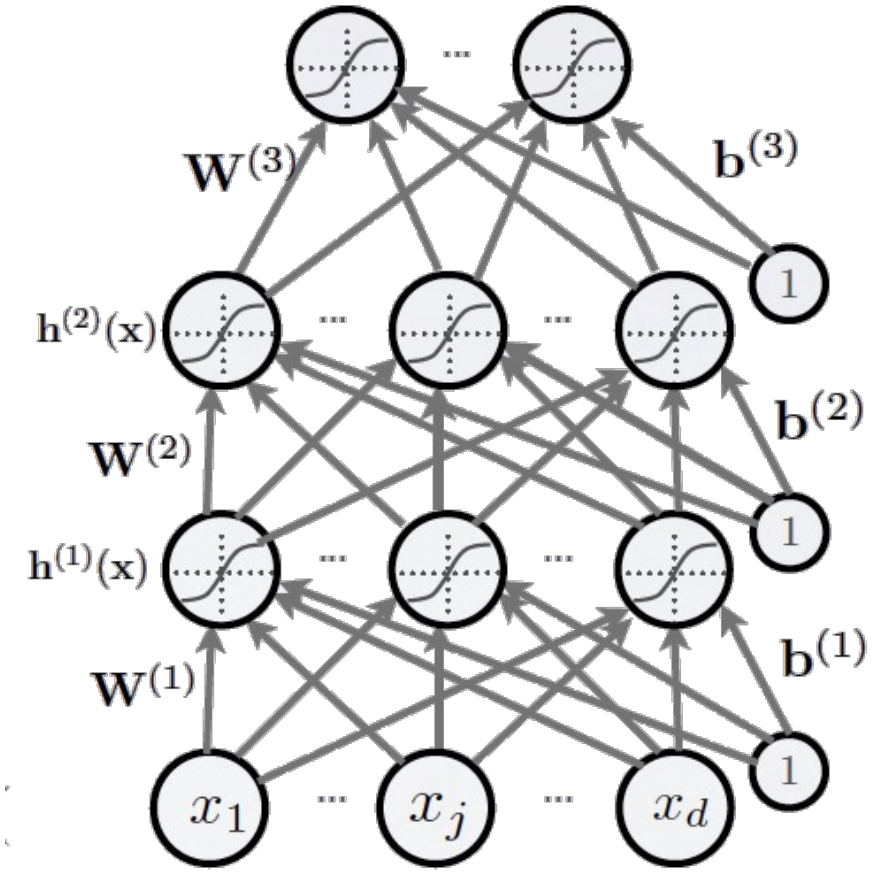
  - Hidden layer activation (*k* from 1 to *L*):

$$\mathbf{h}^{(k)}(\mathbf{x}) = \mathbf{g}\left(\mathbf{a}^{(k)}(\mathbf{x})\right)$$

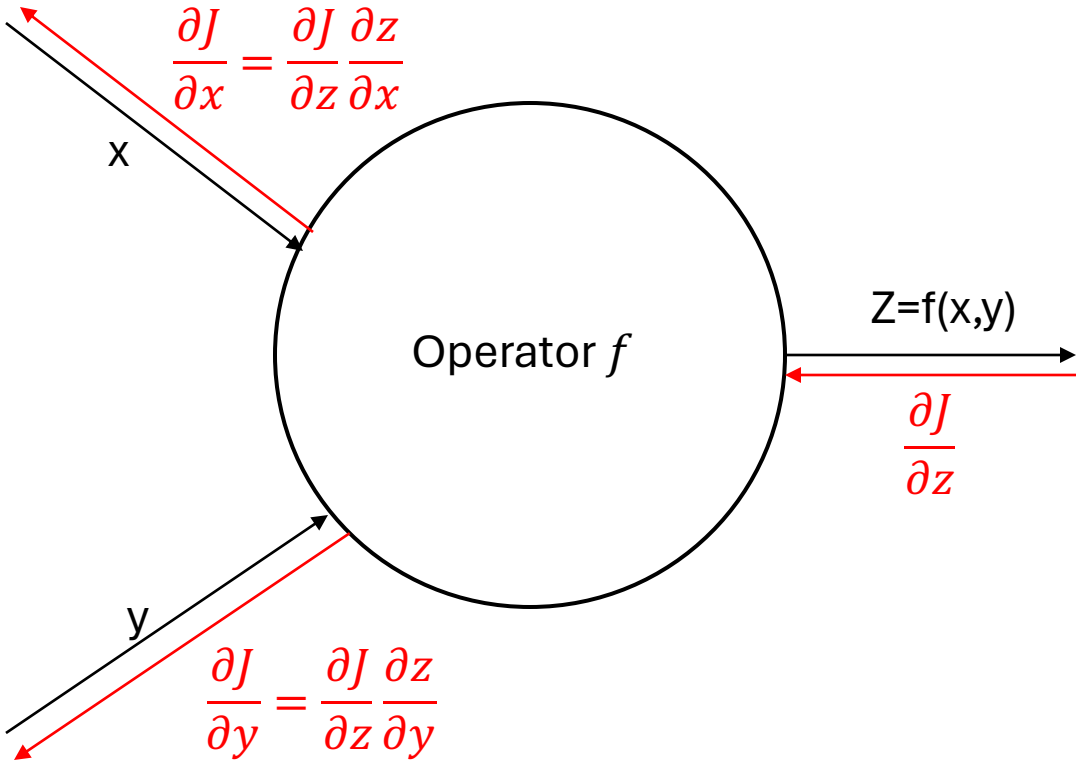
  - Output layer activation (*k*=*L*+1):

$$\mathbf{h}^{(L+1)}(\mathbf{x}) = \mathbf{o}\left(\mathbf{a}^{(L+1)}(\mathbf{x})\right) = \mathbf{f}(\mathbf{x})$$

  - $\mathbf{o}(\cdot)$ can be the softmax function



Back-Propagation Algorithm



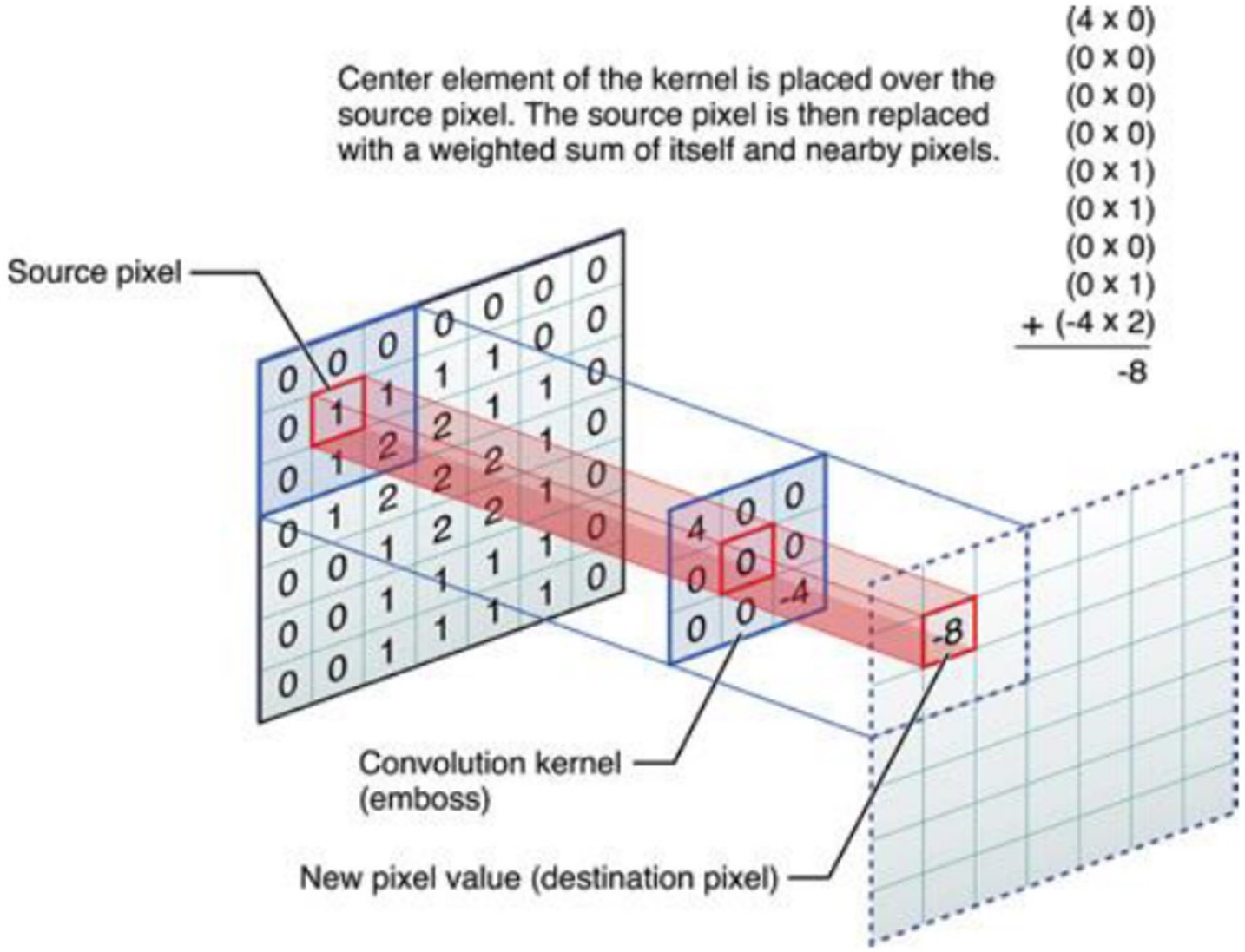
$$y_k = \frac{\exp(b_k)}{\sum_{l=1}^{K} \exp(b_l)}$$

Loss Functions for NNs

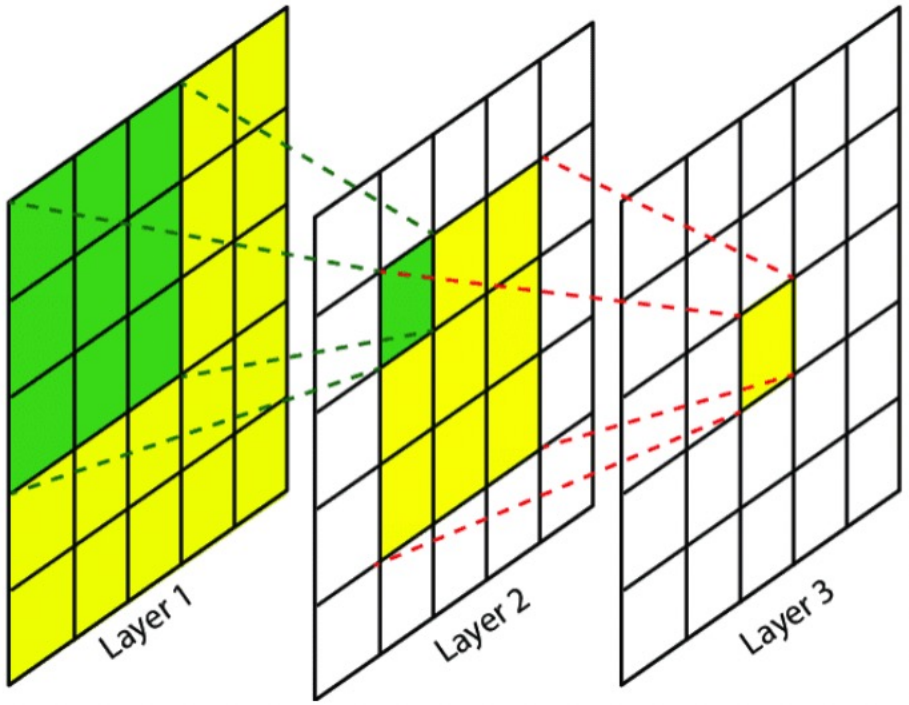
- Regression:
  - Quadratic loss (i.e., mean squared error)
- Classification:
  - Cross-entropy (i.e., negative log likelihood)
  - This requires probabilities, so we add an additional "softmax" layer at the end of our network

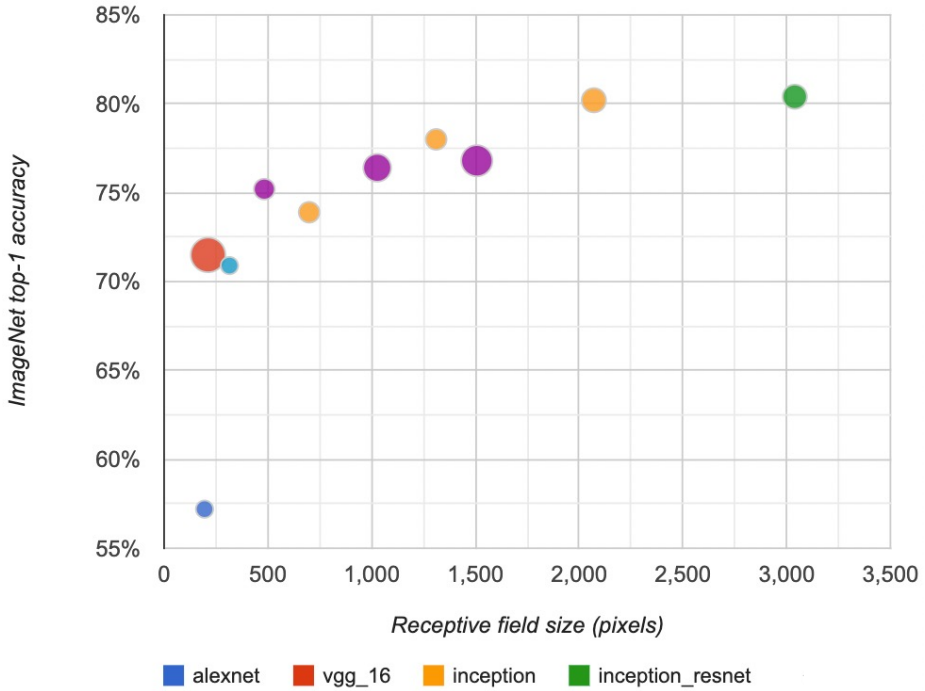
	Forward	Backward
Quadratic	$J = \frac{1}{2}(y - y^*)^2$	$\frac{dJ}{dy} = y - y^*$
Cross Entropy	$J = y^* \log(y) + (1 - y^*)\log(1 - y)$	$\frac{dJ}{dy} = y^* \frac{1}{y} + (1 - y^*) \frac{1}{y - 1}$

Convolutional Networks



Pooling Layer: Receptive Field Size

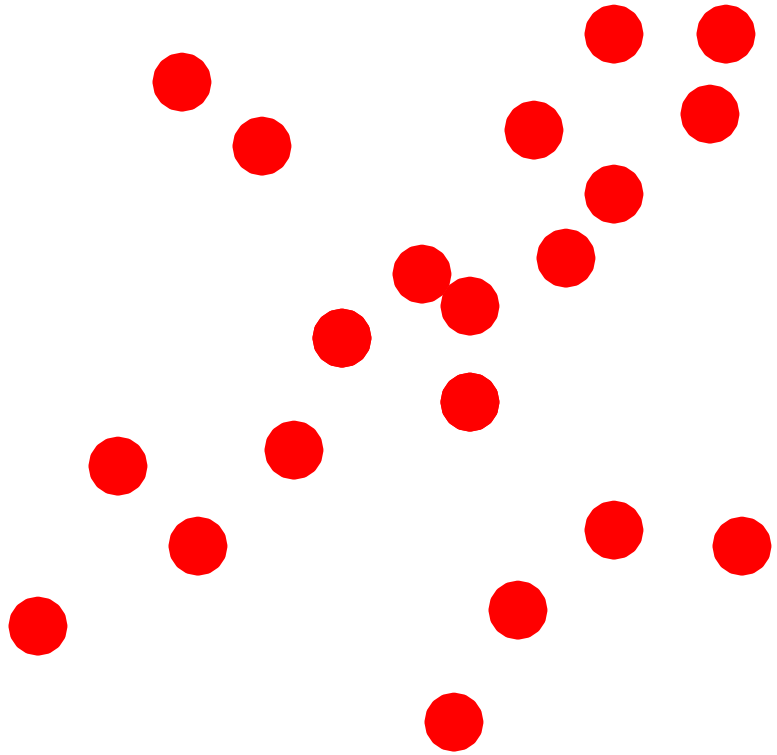




https://distill.pub/2019/computing-receptive-fields/

RANSAC: RANdom SAmple Consensus

- Fischler & Bolles 1981
- Basic idea (for line fitting):
  1. Randomly select two samples (minimal number for a line)
  2. Check how many other samples fall close to the line (within a distance threshold)
  3. Choose the line which has the largest number (“consensus”)



RANSAC: How Many Samples to Choose?

- e: probability that a point is an outlier
  s: number of points in a sample
  N: number of samples (we want to compute this)
  p: desired probability that we get a good sample

$$1 - (1 - (1 - e)^s)^N = p$$
$$\therefore (1 - (1 - e)^s)^N = 1 - p$$
$$\therefore N \log(1 - (1 - e)^s) = \log(1 - p)$$
$$\therefore N = \frac{\log(1 - p)}{\log(1 - (1 - e)^s)} = 16.008 \rightarrow 17$$

- For a 99% probability of obtaining a good model, with a 50% outlier probability and s=2 samples per model

RANSAC: Pros and Cons

- Pros:
  - Robust to outliers
  - Applicable for large number of parameters
- Cons:
  - Computational time grows quickly with the fraction of outliers and the number of parameters
  - Not good for getting multiple line fits
  - Can rapidly find a good model, not necessarily a great model
- Common applications:
  - Computing a rotation and translation between two images (will be used later for fundamental matrix in 2- view geometry)

41

Homogeneous Coordinates

- To homogeneous: $(x, y) \rightarrow (x, y, 1)$ $(x, y, z) \rightarrow (x, y, z, 1)$
- From homogeneous: $(x, y, w) \rightarrow (x/w, y/w)$ $(x, y, z, w) \rightarrow (x/w, y/w, z/w)$

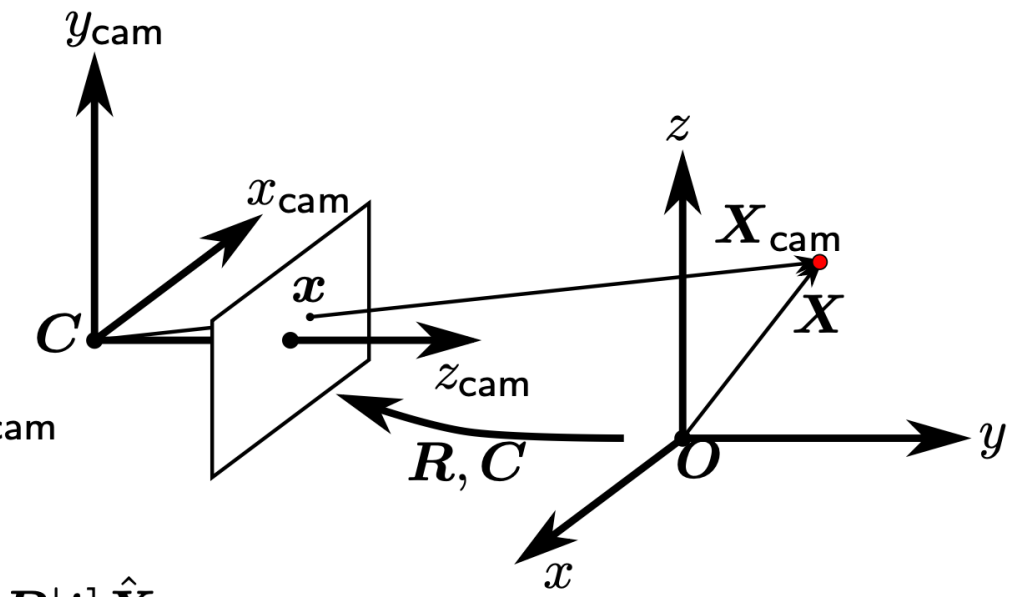
- Invariant to scaling: $k \begin{bmatrix} x \\ y \\ w \end{bmatrix} = \begin{bmatrix} kx \\ ky \\ kw \end{bmatrix} \Rightarrow \begin{bmatrix} \frac{kx}{kw} \\ \frac{ky}{kw} \end{bmatrix} = \begin{bmatrix} \frac{x}{w} \\ \frac{y}{w} \end{bmatrix}$

Homogeneous Cartesian

- A **point** in Cartesian coordinates is a **ray** in homogeneous cords
- What does it mean when the final coordinate is zero?

Summary: Camera Projection Matrix

- image plane
- camera centre $\boldsymbol{C}$
- principal axis z_{cam}
- image coordinates $\boldsymbol{x}$
- world coordinates $\boldsymbol{X}$
- camera coordinates $\boldsymbol{X}_{\text{cam}}$



$$\hat{\boldsymbol{x}} = \boldsymbol{P}\hat{\boldsymbol{X}}$$

$$= \boldsymbol{K}\boldsymbol{R}[\boldsymbol{I}|-\boldsymbol{C}]\hat{\boldsymbol{X}} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]\hat{\boldsymbol{X}}$$

$$= \begin{bmatrix} \alpha & \gamma & u_0 \\ 0 & \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \hat{\boldsymbol{X}}$$

$$= \begin{bmatrix} m_x f_x & \gamma & m_x p_x \\ 0 & m_y f_y & m_y p_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \end{bmatrix} \hat{\boldsymbol{X}}$$

- Focal lengths
- Skew
- Principal point offsets
- Rotation
- Translation

Summary: Camera Calibration

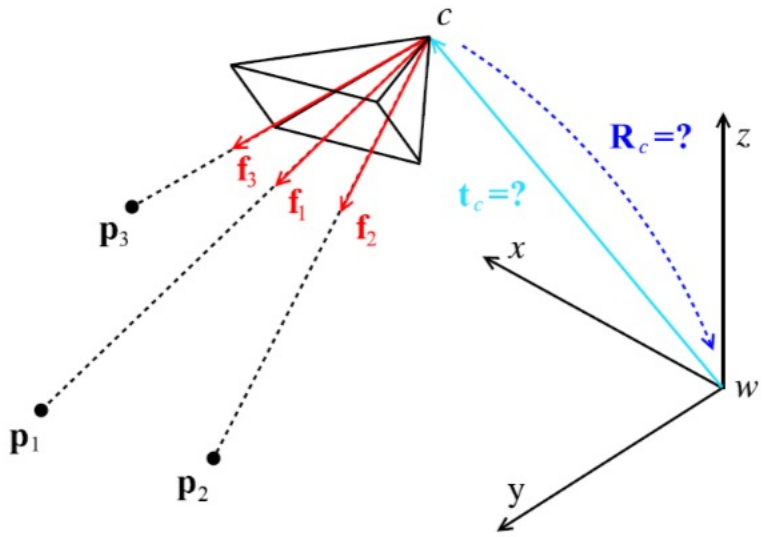
- **Estimate:** $\mathbf{P}, \mathbf{K}, \mathbf{R}, \boldsymbol{C}$
- **Given:** 2D–3D point correspondences $\{\boldsymbol{x}_i, \boldsymbol{X}_i\}$
- **Use:** Direct Linear Transformation (DLT) algorithm to get $\mathbf{P}$
  - Assemble matrix $\mathbf{A}$, where

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{0}^\top & -w_i\mathbf{X}_i^\top & y_i\mathbf{X}_i^\top \\ w_i\mathbf{X}_i^\top & \mathbf{0}^\top & -x_i\mathbf{X}_i^\top \end{bmatrix}$$

  - Take SVD of $\mathbf{A}$, where $\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^\top$; $\boldsymbol{p}$ is the last column of $\mathbf{V}$
  - Reshape vector $\boldsymbol{p}$ into matrix $\mathbf{P}$
- **Decompose:** $\mathbf{P} \rightarrow \boldsymbol{C}$
  - $\mathbf{PC} = \mathbf{0} \Rightarrow \boldsymbol{C}$ is the right null space vector of $\mathbf{P}$
  - Take SVD, where $\mathbf{P} = \mathbf{U}\Sigma\mathbf{V}^\top$; $\boldsymbol{C}$ is the last column of $\mathbf{V}$
- **Decompose:** $\mathbf{P} \rightarrow \mathbf{K}, \mathbf{R}$
  - $\mathbf{K}$ is upper triangular, $\mathbf{R}$ is orthogonal
  - Perform an RQ-decomposition of $\mathbf{KR}$ where $\mathbf{P} = [\mathbf{KR} | -\mathbf{KR}\boldsymbol{C}]$

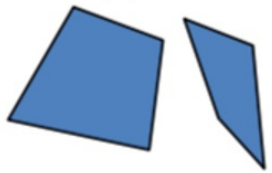
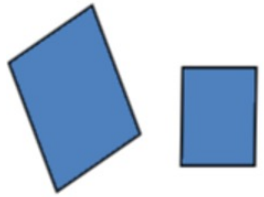
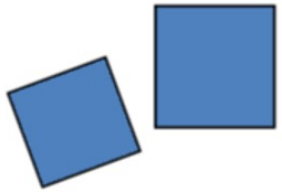
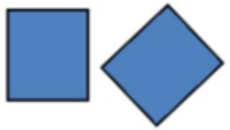
Camera Resectioning / Perspective-n-Point

- Degrees-of-freedom:
  - 6: 3 (translation) + 3 (rotation)
- Minimal solution:
  - 3 point correspondences
  - Perspective-3-point (P3P) problem
  - Variable elimination leads to a 4^{th} order polynomial
    - 4 solutions
    - Use a fourth point correspondence to disambiguate



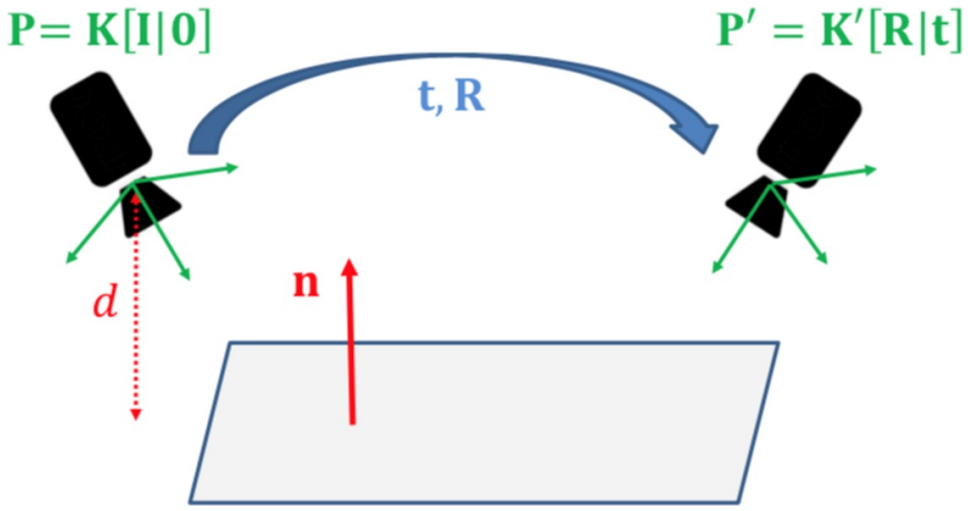
A Hierarchy of 2D Transformations

Homography

		Transformed Square	Invariants
Projective 8 DoF	$\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$		• Concurrency, collinearity, order of contact (intersection, tangency, inflection, etc.), cross ratio
Affine 6 DoF	$\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		• Parallelism, ratio of areas, ratio of lengths on parallel lines (e.g., midpoints), linear combinations of vectors (centroids), line at infinity I_∞
Similarity 4 DoF	$\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		• Ratios of lengths, angles, circular points I,J
Euclidean 3 DoF	$\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$		• Lengths, areas

Planar Homography

- Transforms points to points
- Depends on:
  - Camera intrinsic parameters
  - Relative motion parameters
  - 3D plane parameters (normal vector and depth)



$$\mathbf{H} = \mathbf{K}' \left(\mathbf{R} - \frac{\boldsymbol{t}\boldsymbol{n}^{\mathsf{T}}}{d} \right) \mathbf{K}^{-1}$$

Direct Linear Transformation (DLT) Algorithm for Homography Estimation

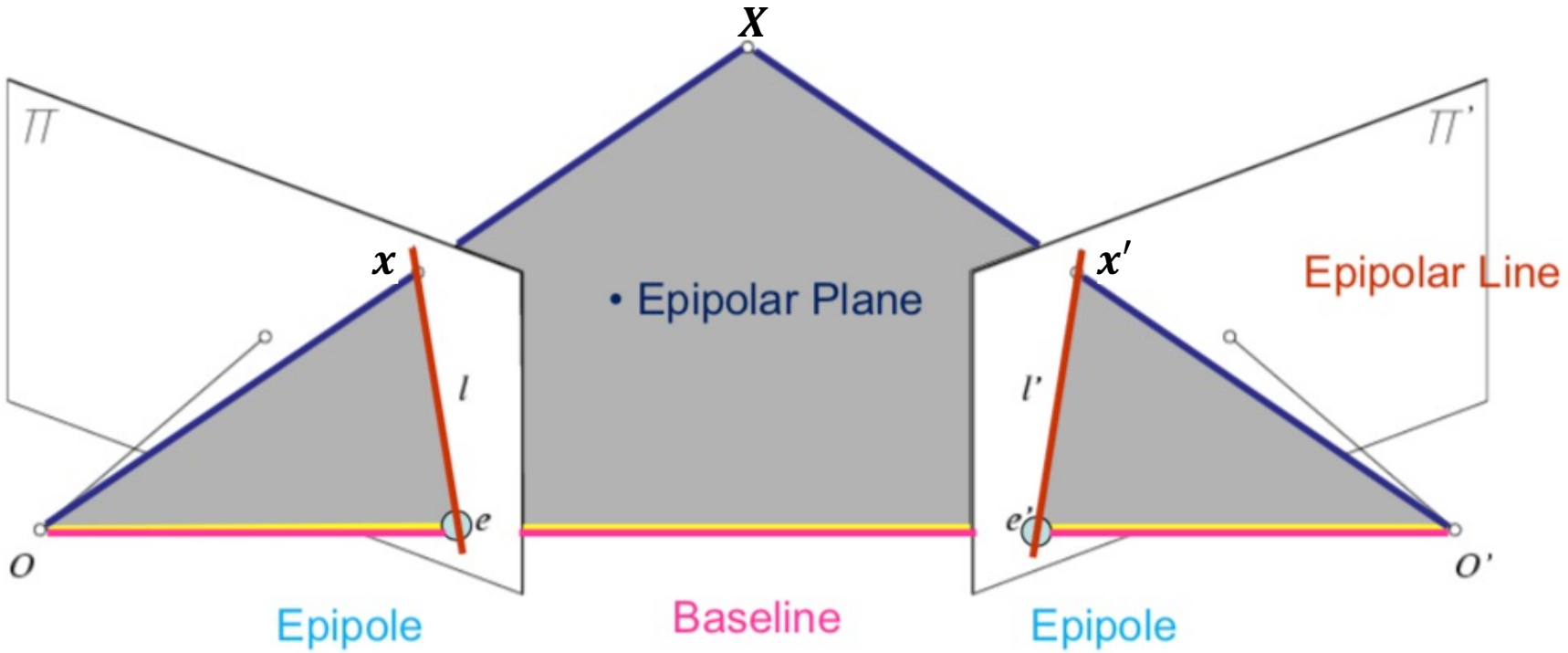
- Equations are linear in $\mathbf{h}$
- Only 2 out of 3 equations are linearly independent, so pick two

$$\underbrace{\begin{bmatrix} \mathbf{0}^\top & -w_i'\mathbf{x}_i^\top & y_i'\mathbf{x}_i^\top \\ w_i'\mathbf{x}_i^\top & \mathbf{0}^\top & -x_i'\mathbf{x}_i^\top \end{bmatrix}}_{\mathbb{R}^{2\times 9}} \underbrace{\begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \end{pmatrix}}_{\mathbb{R}^9} = \mathbf{A}_i\mathbf{h} = \mathbf{0}$$

- Holds for any homogeneous representation, e.g., $(x_i', y_i', 1)$
- Homography matrix has 8 DoF:
  - 9 parameters defined up to scale
  - Linear solution requires at least 4 points (two DoF per point)

Epipolar Geometry: Terminology

- **Baseline:** line joining the camera centres
- **Epipole:** point of intersection of baseline with image plane
- **Epipolar plane:** plane containing baseline and world point
- **Epipolar line:** intersection of epipolar plane with the image plane



Essential & Fundamental Matrices: Summary

- Algebraic representations of epipolar geometry:
  - Projection matrices (given intrinsics + extrinsics)
  - **Essential matrix** (given intrinsics): $\mathbf{E} = [\boldsymbol{t}]_{\times}\mathbf{R}$
  - **Fundamental matrix**: $\mathbf{F} = \mathbf{K}'^{-\top}[\boldsymbol{t}]_{\times}\mathbf{R}\mathbf{K}^{-1} = [\boldsymbol{e}']_{\times}\mathbf{K}'\mathbf{R}\mathbf{K}^{-1}$ [HZ p.244]
- Epipolar constraint for corresponding points $\{\boldsymbol{x}, \boldsymbol{x}'\}$:

$$\boldsymbol{x}'^{\top}\mathbf{F}\boldsymbol{x} = \mathbf{0}$$

  - $\mathbf{F}$ is rank 2 and is known only up to scale → 7 DoF
  - What is $\mathbf{F}\boldsymbol{x}$?
- **Estimation:** DLT (again!); assemble a matrix $\mathbf{A}$, compute the SVD, enforce rank 2 (with another SVD); or use a nonlinear solver

The Essential Matrix E

$$x_c'^{\top}[t]_{\times}Rx_c = x_c'^{\top}Ex_c = 0 \Rightarrow \boxed{E \triangleq [t]_{\times}R}$$

- $[t]_{\times}$: translation cross-product matrix in $\mathbb{R}^{3\times3}$ of rank 2

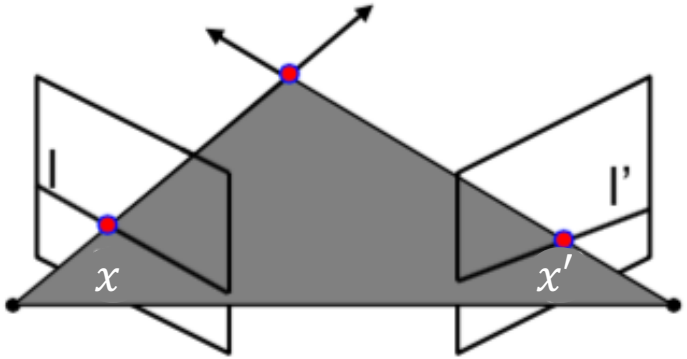
$$[t]_{\times} = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

- R: rotation matrix in $\mathbb{R}^{3\times3}$
- x_c and x_c': homogeneous point vectors in $\mathbb{R}^3$
- $E \triangleq [t]_{\times}R$ in $\mathbb{R}^{3\times3}$ is defined as the Essential matrix
  - Relates the *camera* coordinate frames of camera 1 to camera 2

The Essential Matrix E: Properties

- $E \triangleq [t]_\times R$
- Properties:
  - 5 DoF
  - Rank 2
  - Singular values: $\sigma_1 = \sigma_2$ and $\sigma_3 = 0$
- Constraints:
  - $\det(E) = 0$
  - $2EE^\top E - \mathrm{tr}(EE^\top)E = 0$
- Estimation:
  - “The Five-Point Algorithm” [Nister 2004; Li & Hartley 2006]
  - “The Eight-Point Algorithm” (i.e., DLT) – see later slides

The Fundamental Matrix F



- We have: $x'^{\mathsf{T}} F x = 0$
- And $x'^{\mathsf{T}} l' = 0$
  - Since x' is on the epipolar line l'
- Therefore: $l' = Fx$
- Similarly: $l = F^{\mathsf{T}} x'$
- That is, for a given point x in the first image, we can find the **epipolar line** $l' = Fx$ in the second image
  - On which we can search for the corresponding point

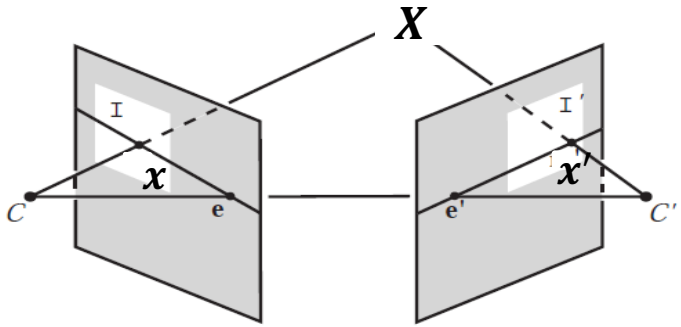
2D line representation:

- In 2D, a line is represented as
$$ax + by + c = 0$$
or equivalently,
$$\begin{bmatrix} a & b & c \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = 0$$
- We use $l = [a, b, c]^{\mathsf{T}}$
  - Orthogonal to vector (a, b)
  - Dot product of the line vector and any point on the line is 0

The Fundamental Matrix F: Properties

- F is the unique 3x3 rank 2 matrix that satisfies $x'^{\top} F x = 0$ for all $x' \leftrightarrow x$
  1. **Transpose:** if F is fundamental matrix for (x, x'), then $F^{\top}$ is the fundamental matrix for (x', x)
  2. **Epipolar lines:** F maps from a point x to a line $l' = Fx$
     - Similarly, $l = F^{\top} x'$
  3. **Epipoles:** lie on all epipolar lines, thus $x'^{\top} F x = 0 \; \forall x \Rightarrow Fe = 0$
     - Similarly, $F^{\top} e' = 0$

Triangulation: DLT Linear Solution



Given $\mathbf{P}, \mathbf{P}', \boldsymbol{x}, \boldsymbol{x}'$

1. Precondition points and projection matrices
2. Create matrix $\mathbf{A}$
3. SVD: $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^\top$
4. $\boldsymbol{X} = \mathbf{V}_{:,-1}$ (last column of $\mathbf{V}$)
5. Then refine with respect to a geometric error

$$\boldsymbol{x} = \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \qquad \boldsymbol{x}' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix}$$

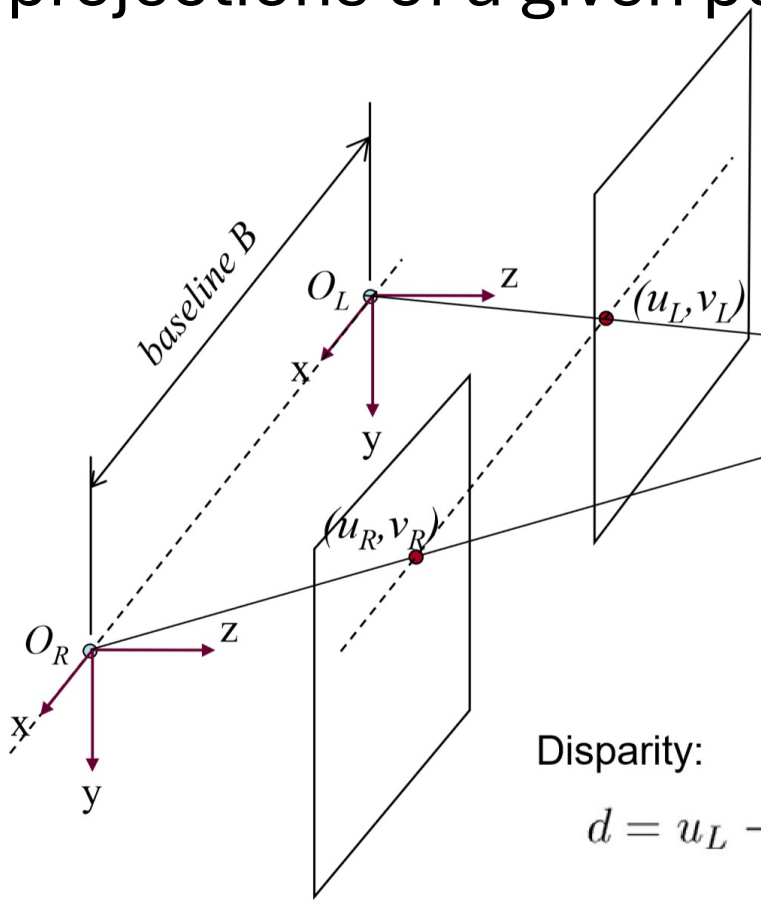
$$\mathbf{P} = \begin{bmatrix} \mathbf{p}_1^\top \\ \mathbf{p}_2^\top \\ \mathbf{p}_3^\top \end{bmatrix} \qquad \mathbf{P}' = \begin{bmatrix} \mathbf{p}_1'^\top \\ \mathbf{p}_2'^\top \\ \mathbf{p}_3'^\top \end{bmatrix}$$

$$\mathbf{A} = \begin{bmatrix} u\mathbf{p}_3^\top - \mathbf{p}_1^\top \\ v\mathbf{p}_3^\top - \mathbf{p}_2^\top \\ u'\mathbf{p}_3'^\top - \mathbf{p}_1'^\top \\ v'\mathbf{p}_3'^\top - \mathbf{p}_2'^\top \end{bmatrix}$$

Code: http://www.robots.ox.ac.uk/~vgg/hzbook/code/vgg_multiview/vgg_X_from_xP_lin.m

Stereo Vision: Disparity-to-Depth

- Disparity (human vision): the positional difference between the two retinal projections of a given point in space [Qian, Neuron '97]



$P_L = (X, Y, Z)$ (in O_L coordinate system)

$$(u_L, v_L) = \left(f\frac{X}{Z}, f\frac{Y}{Z}\right)$$

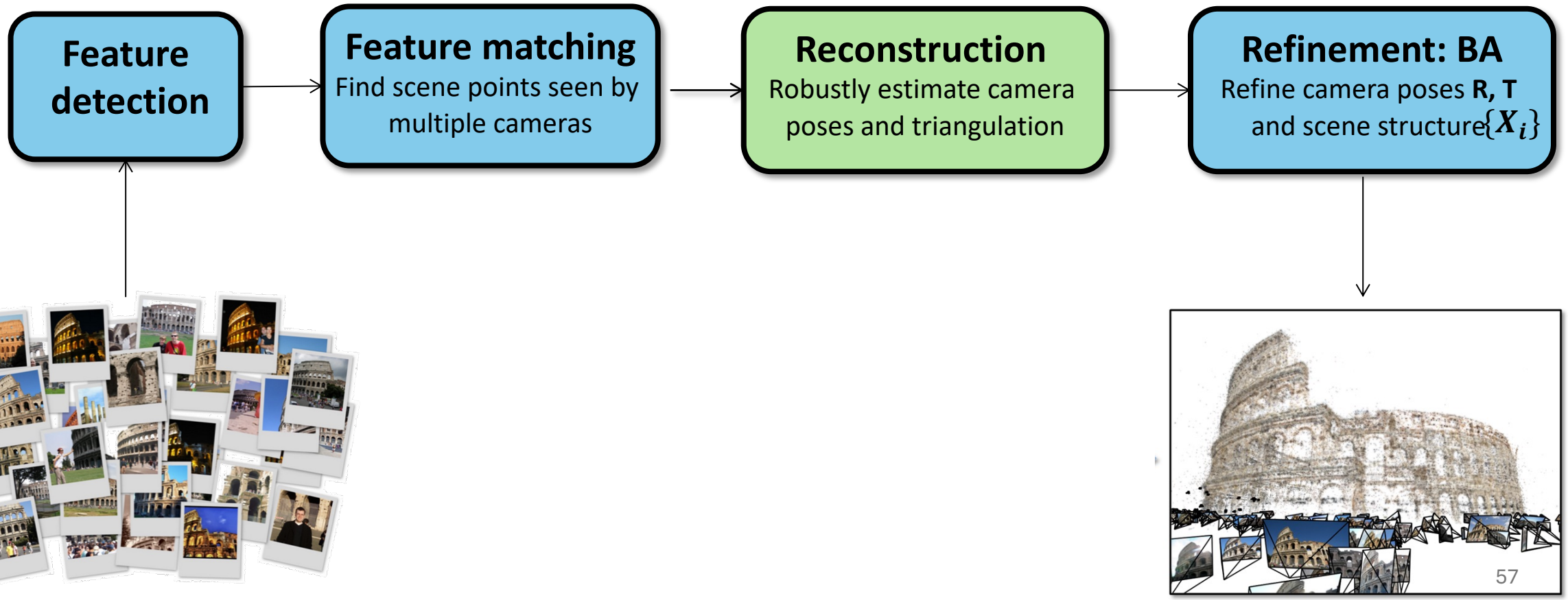
$$(u_R, v_R) = \left(f\frac{X-B}{Z}, f\frac{Y}{Z}\right)$$

Disparity:

$$d = u_L - u_R = f\frac{B}{Z} \Rightarrow \boxed{Z = f\frac{B}{d}}$$

Structure From Motion

- Core Steps of SfM



Optical Flow

- **Problem:** Compute a flow field that takes pixels in the first image to their location in the second image.

$$(dx, dy) = f\big(I(t), I(t+dt)\big)_{(x,y)}$$

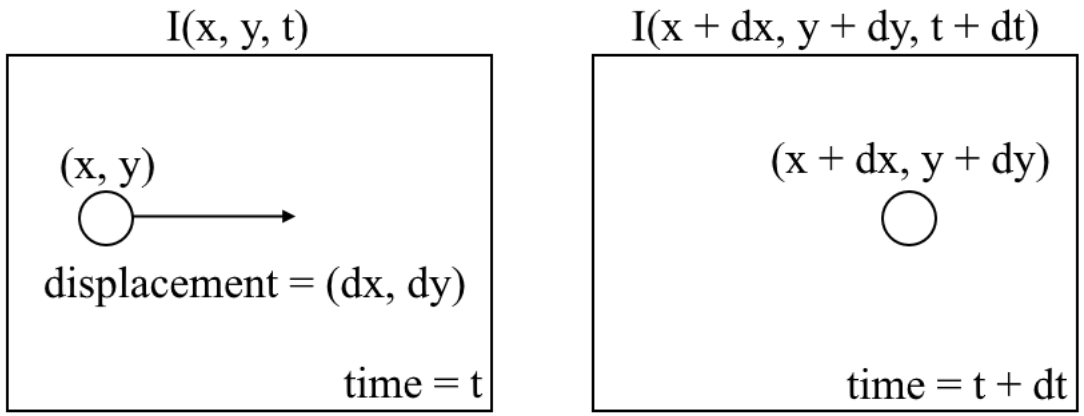


Image Credit: Chuan-en Lin

Brightness/Colour Constancy Assumption

$$I_x u + I_y v = -I_t \leftrightarrow \frac{\partial I}{\partial x}\frac{\mathrm{d}x}{\mathrm{d}t} + \frac{\partial I}{\partial y}\frac{\mathrm{d}y}{\mathrm{d}t} = -\frac{\partial I}{\partial t}\frac{\mathrm{d}t}{\mathrm{d}t}$$

- From first-order Taylor expansion of $I(x, y, t) = I(x + \mathrm{d}x, y + \mathrm{d}y, t + \mathrm{d}t)$
  - Note that all partial derivatives are evaluated at (x, y, t), e.g., $I_x(x, y, t)$
- We have one equation and two unknowns (u, v): the optical flow
- Assumes a smooth visual gradient over the area of motion

Lucas–Kanade Optical Flow Algorithm

$$[I_x \quad I_y] \begin{bmatrix} u \\ v \end{bmatrix} = -I_t$$

- Least squares solution: $\boldsymbol{u}^\star = \operatorname{argmin}_{\mathbf{u}} \|A\boldsymbol{u} - \boldsymbol{b}\|^2 = (A^\top A)^{-1} A^\top \boldsymbol{b}$
- $A^\top A = \begin{bmatrix} \sum_{\boldsymbol{p}\in\mathcal{P}} I_x I_x & \sum_{\boldsymbol{p}\in\mathcal{P}} I_x I_y \\ \sum_{\boldsymbol{p}\in\mathcal{P}} I_y I_x & \sum_{\boldsymbol{p}\in\mathcal{P}} I_y I_y \end{bmatrix} \in \mathbb{R}^{2\times 2}$ (autocorrelation matrix)
- $A^\top \boldsymbol{b} = -\begin{bmatrix} \sum_{\boldsymbol{p}\in\mathcal{P}} I_x I_t \\ \sum_{\boldsymbol{p}\in\mathcal{P}} I_y I_t \end{bmatrix} \in \mathbb{R}^{2\times 1}$
  where the sum is over pixels $\boldsymbol{p}$ in patch $\mathcal{P}$
- You saw this in Harris corner detector!

Lucas & Kanade, 1981

Lucas–Kanade Optical Flow Algorithm

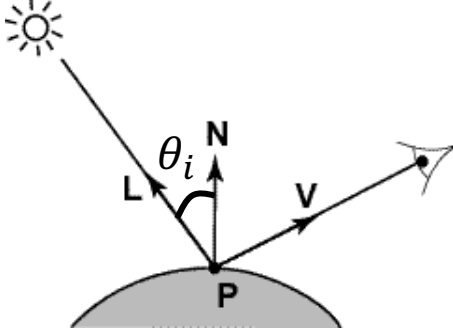
- When is $A^\top A\boldsymbol{u} = A^\top \boldsymbol{b}$ solvable?
  - $A^\top A$ invertible:
    - Determinant non-zero
  - $A^\top A$ not too small, otherwise estimate is sensitive to noise:
    - Eigenvalues λ_1 and λ_2 of $A^\top A$ should not be too small
  - $A^\top A$ well-conditioned:
    - λ_1/λ_2 should not be too large (λ_1: larger eigenvalue)
- Implications:
  - Harris corners are where λ_1 and λ_2 are both big; this is also when Lucas–Kanade optical flow estimation works best
    - Corners are good places to compute optical flow

Lucas & Kanade, 1981

Visual Cues: Shape-from-

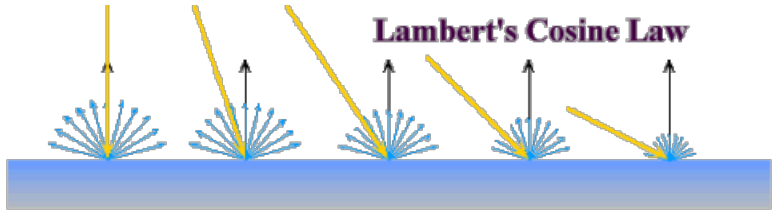
- Shading
- Texture
- Focus
- Motion
- Perspective distortion
- Colour
- Size
- Occlusion
- Stereo
- Specular highlights
- Inter-reflections
- Symmetry
- Light Polarisation
- Structured light (active)
- Time-of-flight (active)
- Shadow
- Silhouette

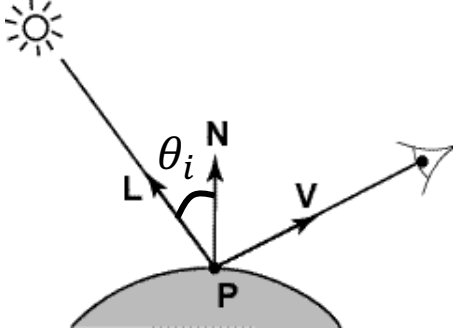
Shape-from-Shading



- Lambertian reflectance assumption gives us $I = N^\top L = \cos\theta$
- Not enough information to compute normal: 1 equation, 2 DoF
- Add additional information: e.g., smoothness
  - Variational shape-from-shading
- Then, convert normal maps to depth maps via integration

Lambertian Reflection





$$R_r = \rho N \cdot L R_i$$
$$I = \rho N \cdot L$$
$$\boxed{I = N \cdot L = \cos\theta_i}$$

- R_i: incident light intensity
- R_r: observed light intensity
- L: unit illuminant direction
- N: unit normal direction
- I: image intensity at point **P**
- ρ: material albedo

- Simplifying assumptions
  - $I = R_r$: camera response function f is the identity function
    - Can also assume linearly proportional
    - (If required, can perform radiometric calibration)
  - $R_i = 1$: light source intensity is 1
    - Can achieve this by dividing each pixel in the image by R_i
  - $\rho = 1$: material albedo is 1

Photometric Stereo

$$\underbrace{[I_1 \;\; I_2 \;\; I_3]}_{I} = \underbrace{\rho N^{\top}}_{G} \underbrace{[L_1 \;\; L_2 \;\; L_3]}_{L}$$

Image intensity matrix I is known

I 1x3

G 1x3

L 3x3

Light source matrix L is known

$$G = IL^{-1}$$

- ρ and N are unknowns (ρ may differ across surface)
- Surface normal: $N = \frac{G}{\|G\|}$
- Albedo: $\rho = \|G\|$
- When is L invertible? >= 3 light directions are linearly independent
- More than 3 lights? Solve using least squares

Next Lecture

- Practice exam questions
- Q&A
- Drop-in session