- Class logistics
- Introduction to computer vision
- Instance-level recognition

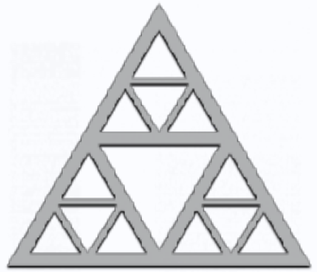
Gül Varol

IMAGINE team, École des Ponts ParisTech

gul.varol@enpc.fr

https://gulvarol.github.io/

@RecVis, 08.10.2024



With many slides from: J. Sivic, I. Laptev, O. Chum, K. Grauman, J. Hays, D. Hoiem, S. Lazebnik, B. Leibe, D. Lowe, J. Philbin, J. Ponce, D. Nister, C. Schmid, N. Snavely, A. Zisserman, H. Sawhney, K. Derpanis

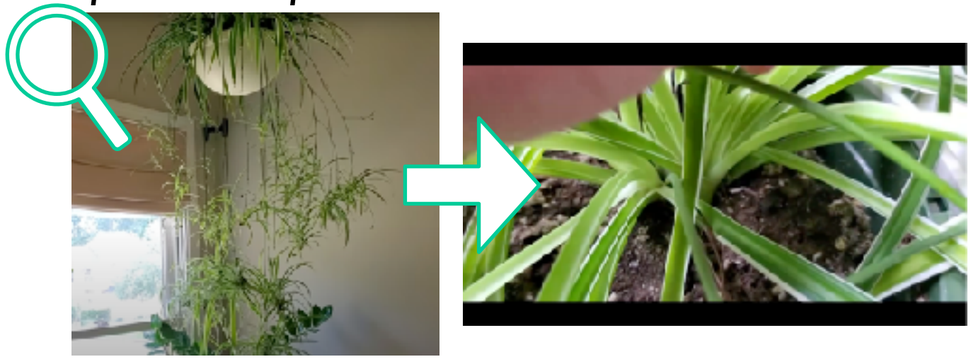
About me

Language & Dynamic Visual Data

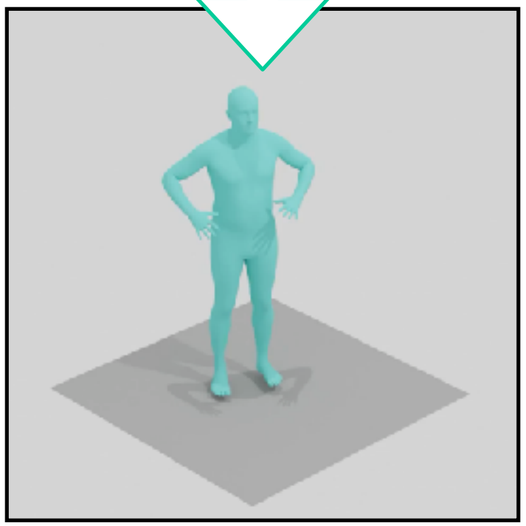
- Text-to-Video retrieval
- Sign language videos
- Movie description
- 3D Human motion generation
- …

Predicted Audio Description: Snape points at Harry. Harry's eyes close in horror.

"prune this plant"



{ put hands on the waist, move torso left }



About me

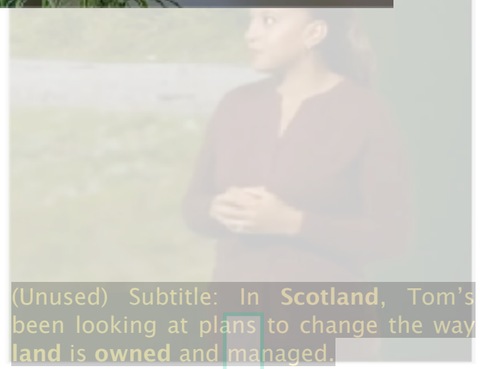
Language & Dynamic Visual Data

- Text-to-Video retrieval
- Sign language videos
- Movie description
- 3D Human motion generation
- …

Predicted Audio Description: Snape points at Harry. Harry's eyes close in horror.

"prune this plant"

{ put hands on the waist, move torso left }



(Unused) Subtitle: In **Scotland**, Tom's been looking at plans to change the way **land** is **owned** and managed.

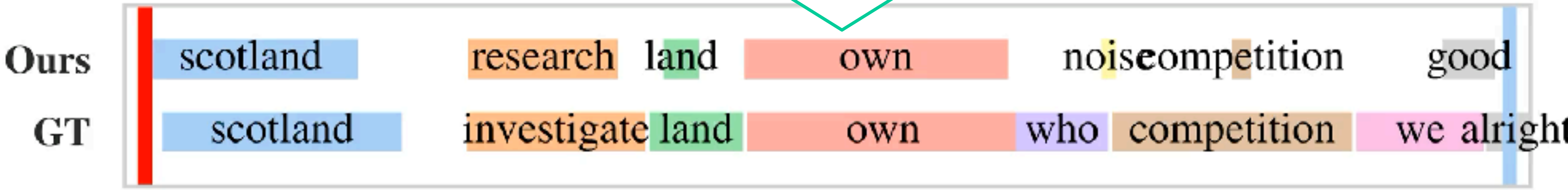
Ours	scotland	research	land	own	noisecompetition	good	
GT	scotland	investigate	land	own	who	competition	we alright

About me

Language & Dynamic Visual Data

- Text-to-Video retrieval
- Sign language videos
- Movie description
- 3D Human motion generation
- ...

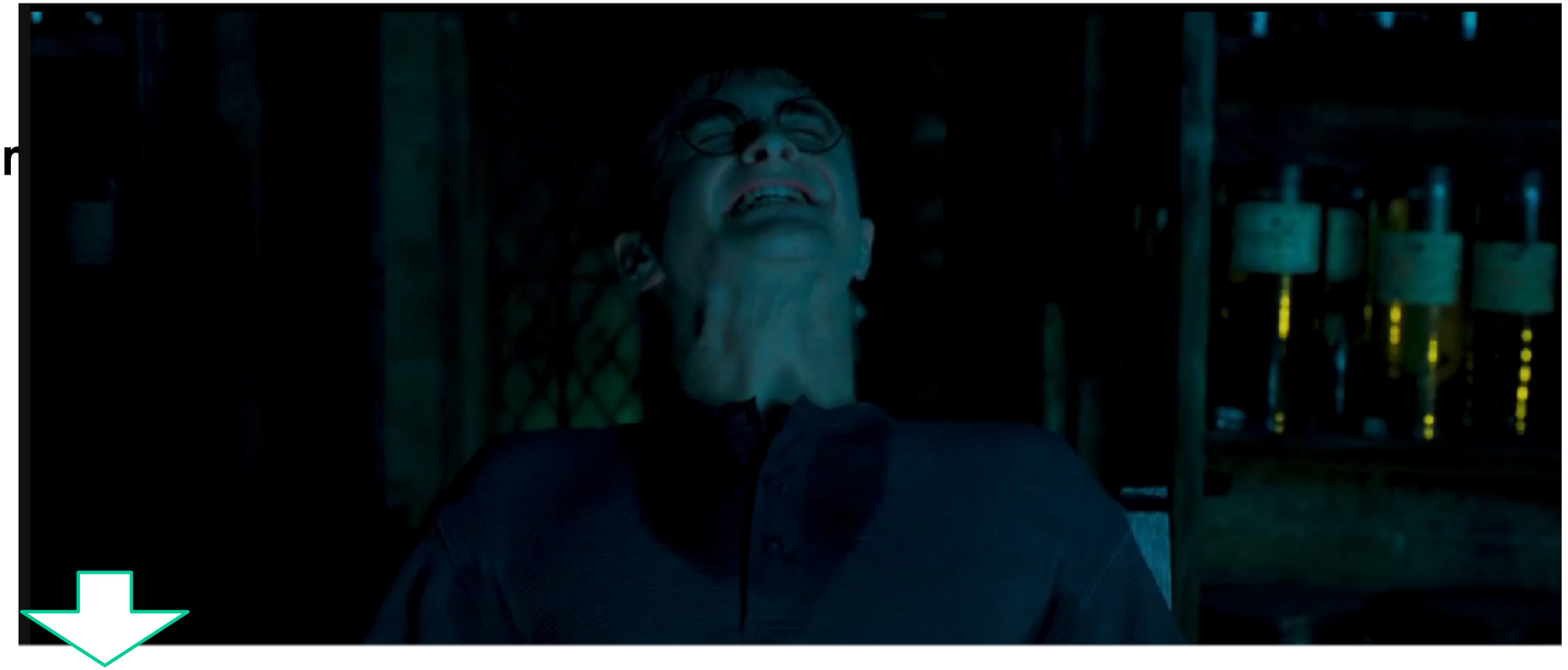
Predicted Audio Description: Sn
Harry. Harry's eyes close in horr



About me

Language & Dynamic Visual Data

- Text-to-Video retrieval
- Sign language videos
- Movie description
- 3D Human motion gener
- …



Predicted Audio Description: Snape points at Harry. Harry's eyes close in horror.

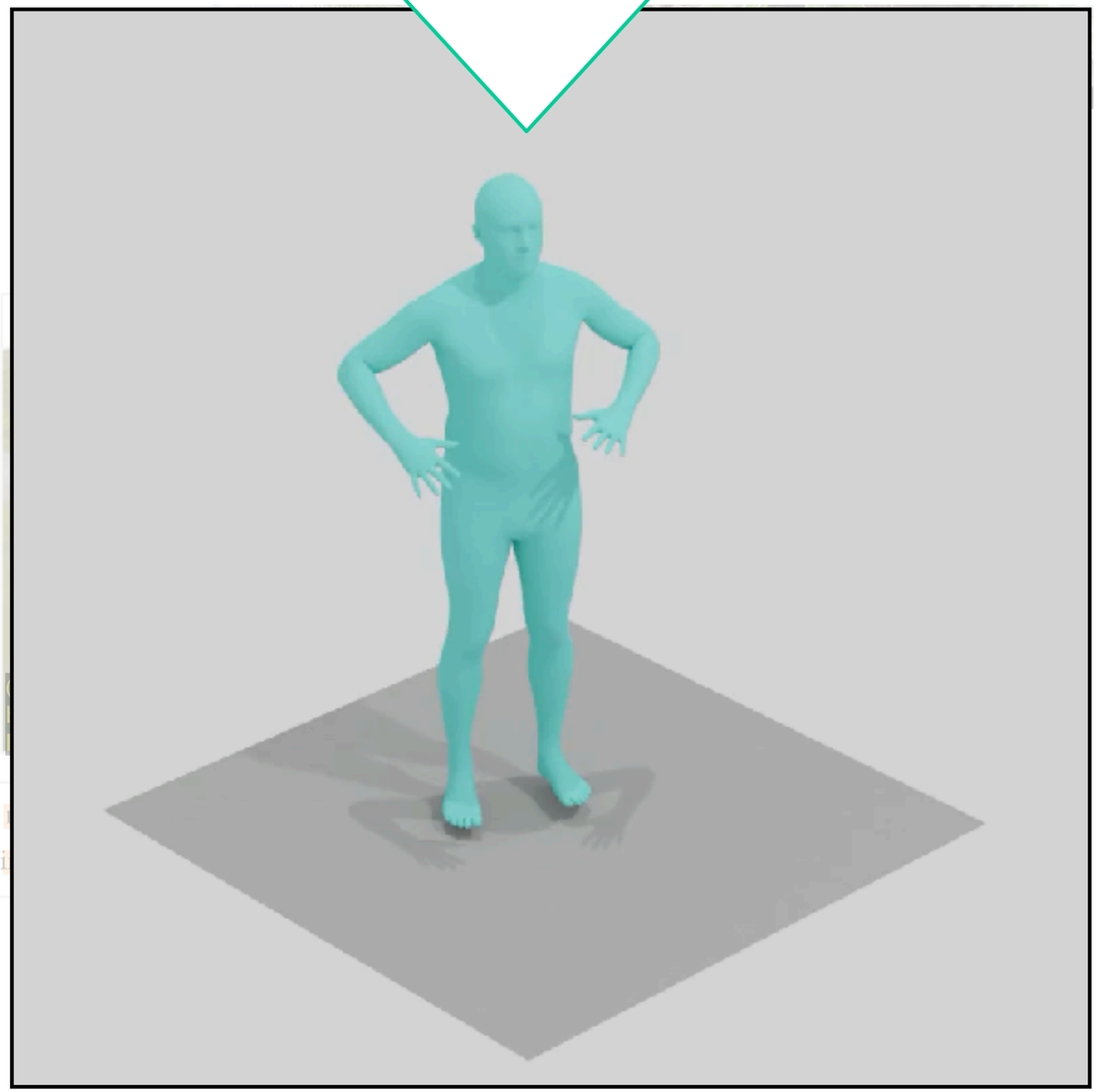
About me

Language & Dynamic Visual Data

- Text-to-Video retrieval
- Sign language videos
- Movie description
- 3D Human motion generation
- …

Predicted Audio Description: Snape points at Harry. Harry's eyes close in horror.

{ put hands on the waist, move torso left }



About you

Join at

slido.com

#3021 898

First words that come to your mind when hearing “computer vision”?

IMAGINE computer vision team, ENPC

École des Ponts
ParisTech

imagine.enpc.fr/

Keep an eye on internships

Research

IMAGINE and WILLOW teams are active in computer vision.

http://imagine.enpc.fr

http://www.di.ens.fr/willow/

There will be master internships available. Talk to us if you are interested!

Class Logistics

http://imagine.enpc.fr/~varolg/teaching/recvis24/

Object recognition and computer vision 2024

Reconnaissance d'objects et vision artificielle (RecVis)

Lecturers

Gül Varol

(✉ Main lecturer)

Contact for other Qs

Jean Ponce

Oct 15

Cordelia Schmid

Nov 26

Ivan Laptev

Dec 3

Mathieu Aubry

Dec 10

Teaching Assistants (TAs)

Ricardo Garcia

Lucas Ventura

Contact TAs for questions about assignments

ricardo-jose.garcia-pinel@inria.fr

lucas.ventura@enpc.fr

Schedule

Tuesdays 16h - 19h
Location: Salle Dussane

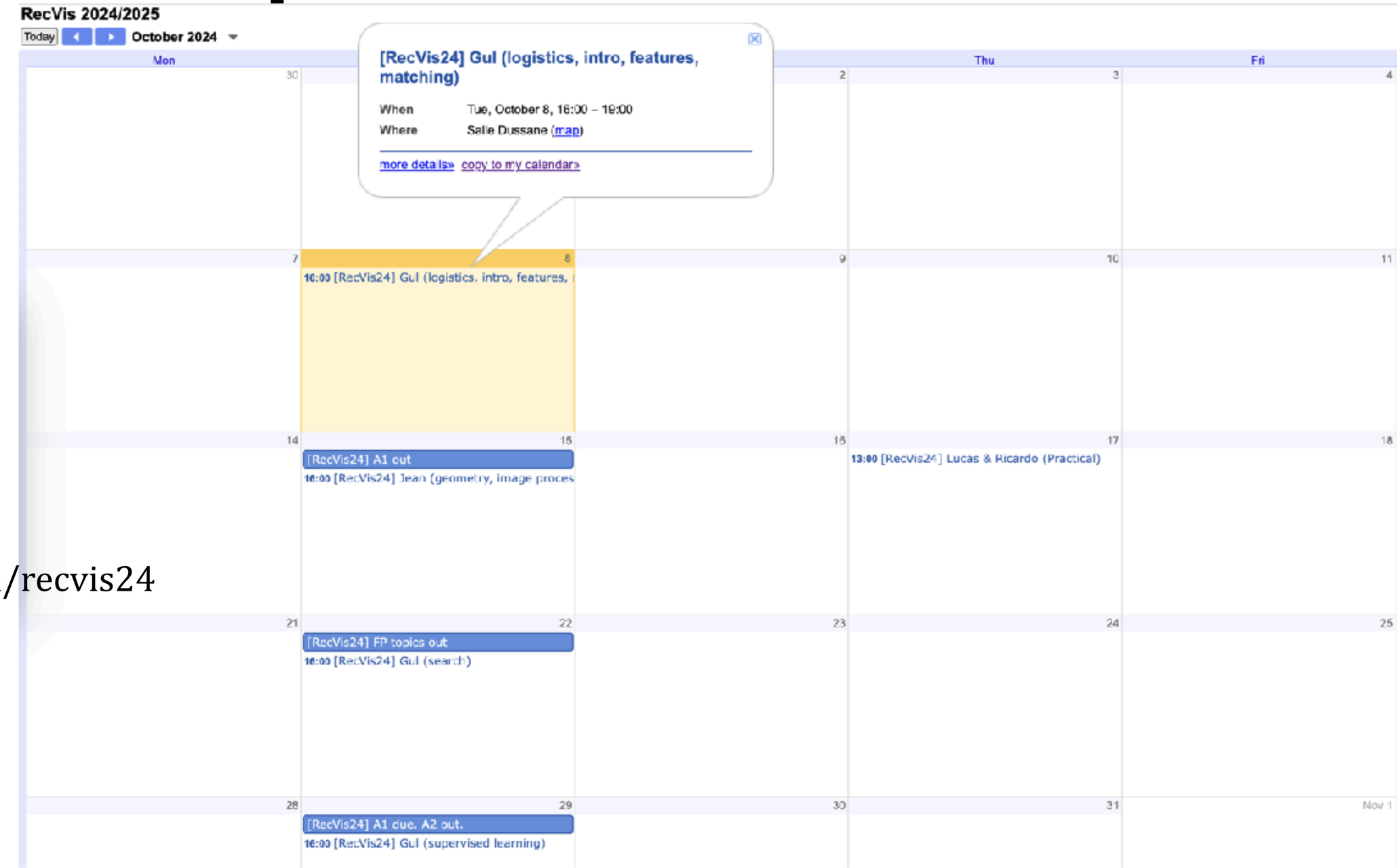
!!!

Follow updates & exceptions on class webpage

#	Date	Lecturer	Topic and reading materials	Slides
			Instance-level recognition	
1	Oct 8	Gül Varol	Class logistics: assignments, final projects, grading; Introduction to visual recognition; Instance-level recognition: local features, correspondence, image matching materials	
2	Oct 15	Jean Ponce	Camera geometry; Image processing materials Assignment 1 (A1) out.	
Practical	Oct 17 *(1-3pm) Inria, 48 Rue Barrault, 75013*	TAs	Pytorch/Kaggle/Google Cloud tutorial. Presentations by TAs about their PhD topics.	
3	Oct 22	Gül Varol	Large-scale image and video search Final project (FP) topics are out at the end of the lecture.	
			Category-level recognition	
4	Oct 29	Gül Varol	Supervised learning and deep learning; Optimization and regularization for neural networks A1 due. A2 out.	
5	Nov 5	Gül Varol	Neural networks for visual recognition: CNNs and image classification materials A3 out.	
6	Nov 12 *Salle Evariste Galois*	Gül Varol	Beyond CNNs: Transformers; Beyond classification: Object detection; Segmentation; Human pose estimation materials A2 due.	
			Advanced topics	
7	Nov 19 *Salle Evariste Galois*	Gül Varol	Generative models; Vision & language materials FP proposal due.	
8	Nov 26	Cordelia Schmid	Human action recognition in videos A3 due.	
9	Dec 3	Ivan Laptev	Vision for robotics	
10	Dec 10	Mathieu Aubry	3D computer vision	
FP	Jan 6	Gül Varol	FP presentations Presentations may be virtual, the schedule will be announced later. FP report due **Jan 13**.	

Schedule: Add public calendar

https://tinyurl.com/recvis24



Practical information: Participation

Class webpage : http://imagine.enpc.fr/~varolg/teaching/recvis24/
Google Classroom : Register with the code **lwdpcwo** to receive announcements.
Time : 16h00-19h00, Tuesdays, starting Oct 8
Location : Salle Dussane, ENS Ulm, 45 rue d'Ulm 75005, Paris
Format : In-person lectures. Slides provided after each lecture.

For externals : You are welcome to attend the course (either for auditing or validation) provided there are enough free places in the lecture hall. If your school requires a proof of attendance, you need to get signatures from teachers after every lecture.

Practical information: Grading

- **3 programming assignments (50%)**
  - A1: Instance-level recognition
  - A2: Neural networks
  - A3: Image classification Kaggle competition

Policy

Assignments are strictly individual

Copy-paste of the code, results, parts of the report ⇨ 0p.

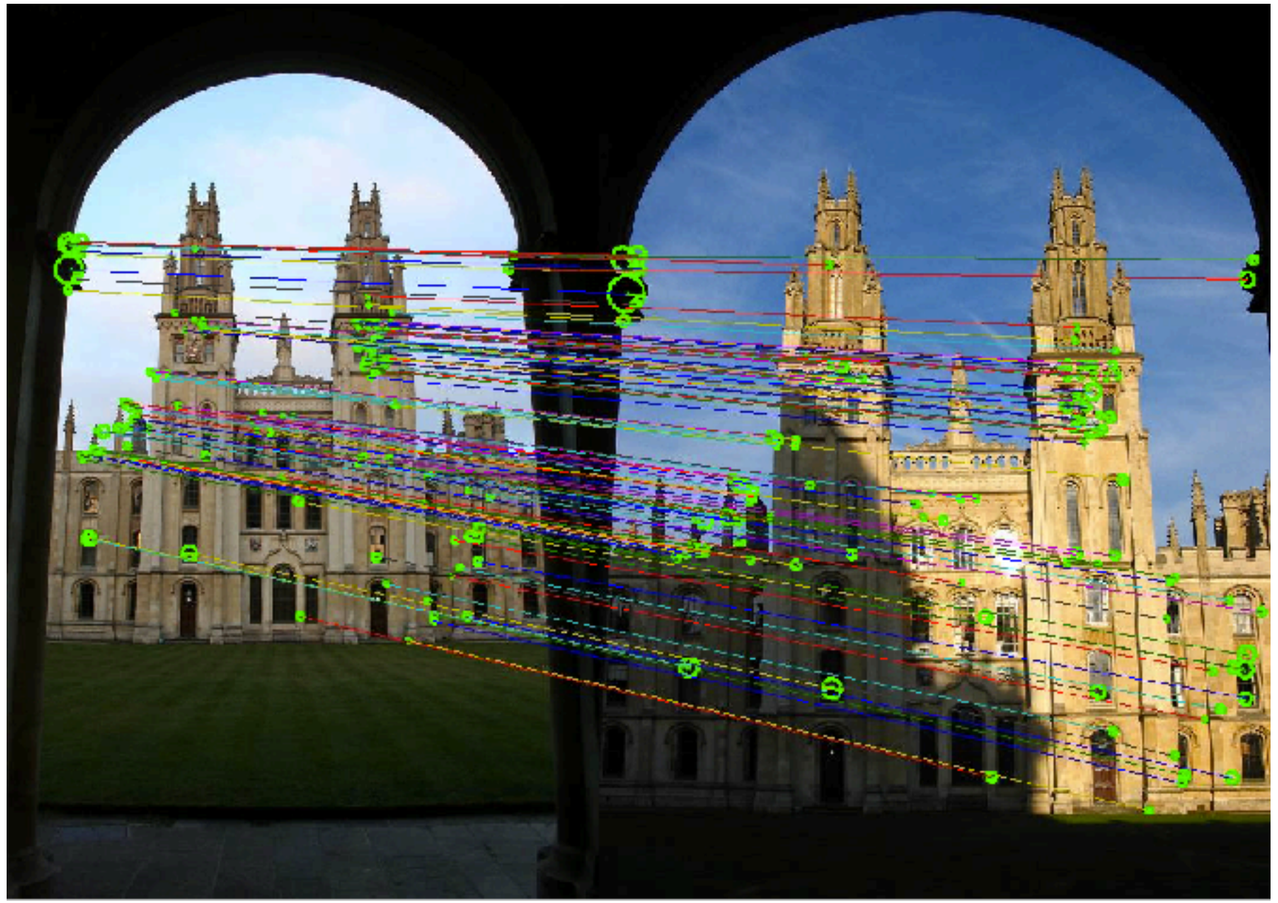
- **Final project (50%)**
  - More independent work, resulting in a report and a class presentation.
  - We will provide Google Cloud credits for each student.

FPs can be done in groups of max 2 people

Experience with Python (numpy, pytorch) will be useful, but TAs will provide an optional crash-lecture on **Pytorch** for computer vision, as well as **Kaggle** and **Google Cloud** usage.

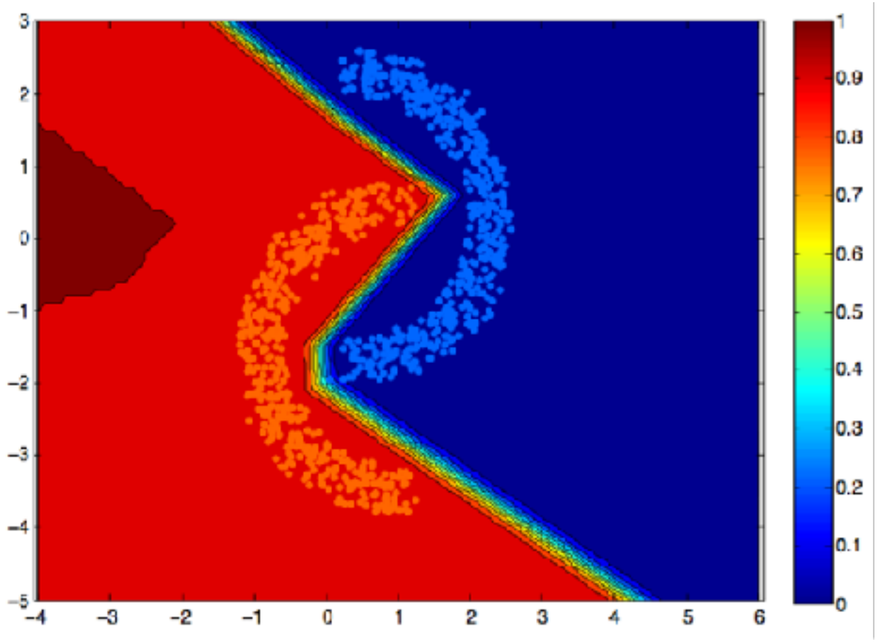
Assignment I: Instance-level recognition

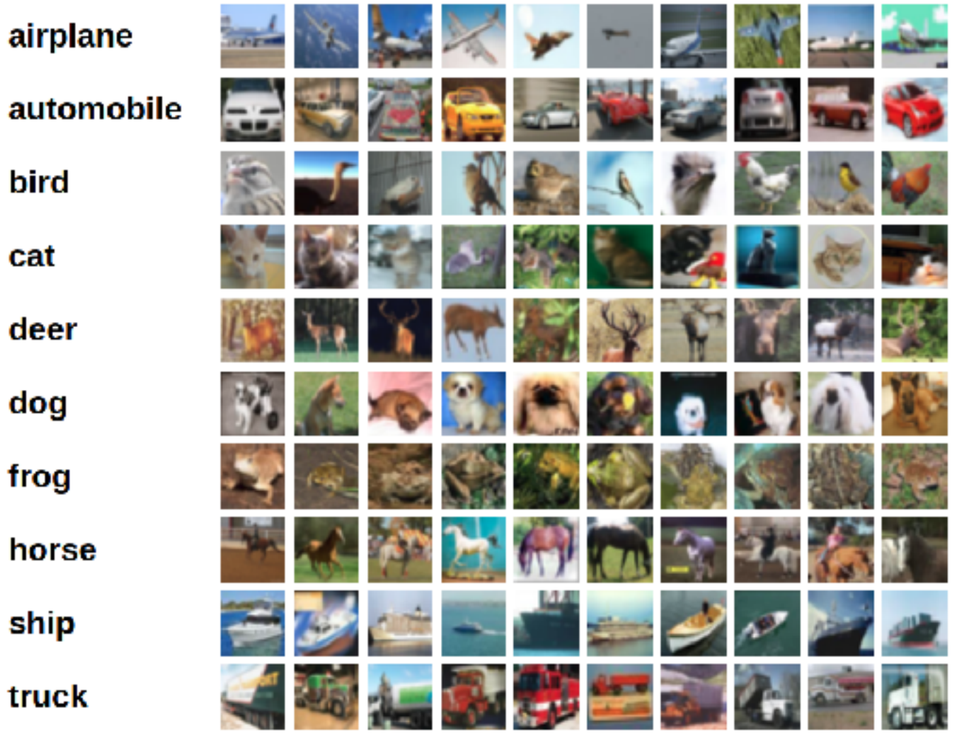
- Part I: Sparse features for matching specific objects in images
  - Feature detector and descriptor
  - Robust match filtering techniques
  - Augmented reality
- Part II: Compact descriptors for image retrieval



Assignment II: Neural networks

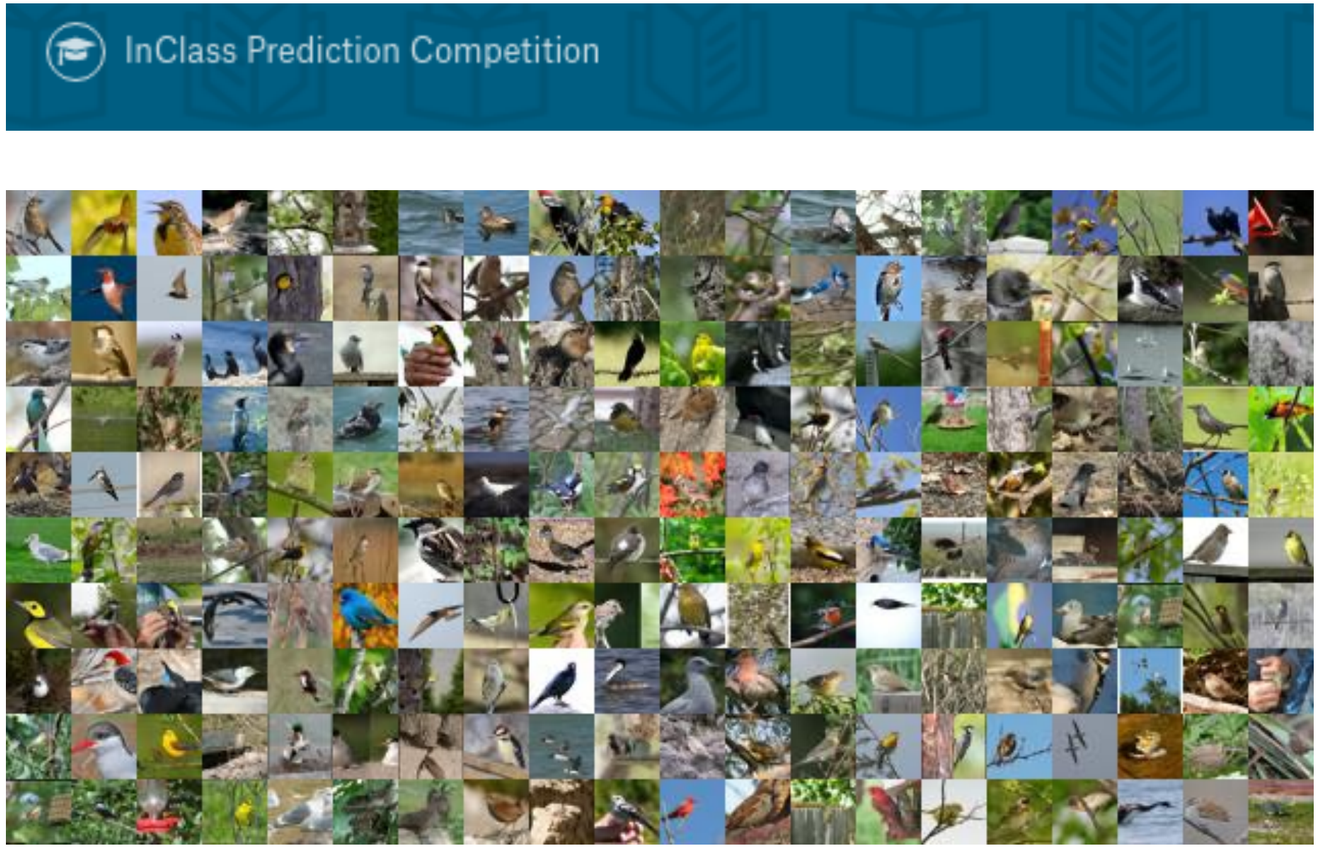
- Part 1: Neural Network's theory:
  - Forward pass, Backward pass
  - Parameter update
- Part 2: Building blocks of convolutional neural networks
- Part 3: Training a CNN on CIFAR-10 dataset with PyTorch





Assignment III: Image classification competition

- Class Kaggle competition
- Example task: Bird image classification - the assignment will cover a similar task



Final Project (FP)

- Can be done individually or as a **group of max 2 people**
- The proposed project topics are from the recent top-conference publications in computer vision, see example topics from 2023 here: http://imagine.enpc.fr/~varolg/teaching/recvis23/
- Student-defined projects are welcome.
- Final project can be joint with another MVA course.
- We arranged $100 Google Cloud credits for the project.
  - This will be announced through Google Classroom before projects start

- Select the topic + write project proposal
- Present the work in the class
- Write project report

Practical by TAs

Fill-in the TAs session participation form linked from the class webpage by **Tue Oct 15.**

The tutorial will be on **Thu Oct 17**, between **13h00-15h00** at:
INRIA/Willow, 48 **rue Barrault, 75013 Paris.**

Who should participate?

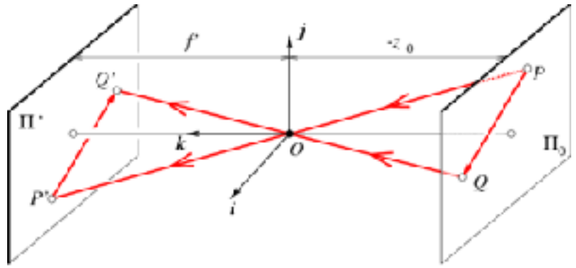
- Students with no or limited experience with PyTorch/Kaggle/Google Cloud
- Students curious to know about life of a PhD student in computer vision / robotics
- Attendance is optional.

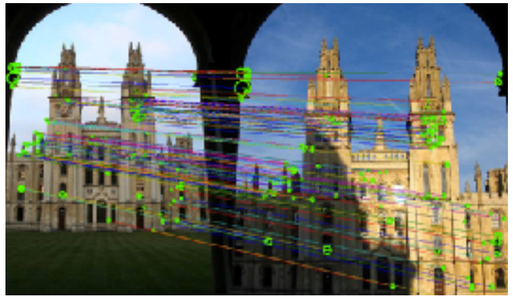
Agenda:

- Presentations by TAs on their PhD topics, followed by Q&A
- Introduction to PyTorch for computer vision
- Kaggle competition (used for Assignment 3)
- Using Google Cloud (used for Final Project)

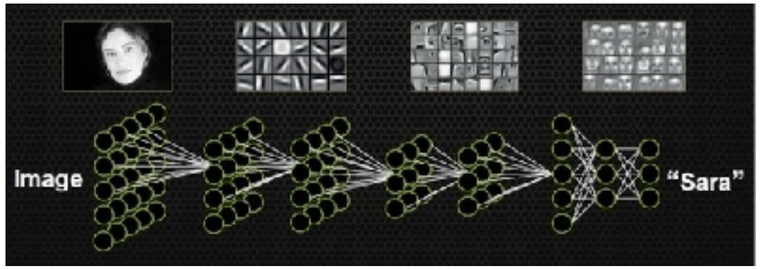
Course outline

1. Instance-level recognition
   - Camera geometry
   - Image processing
   - Image correspondence
   - Large-scale image and video search





2. Category-level recognition
   - Supervised learning
   - Neural networks for visual recognition
   - Object recognition, detection, and segmentation

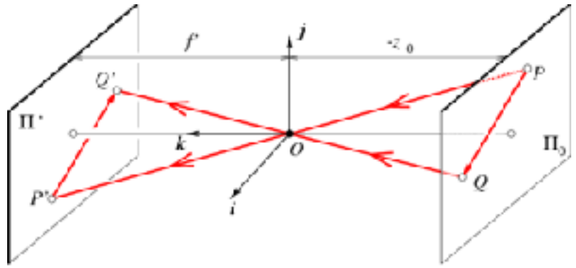


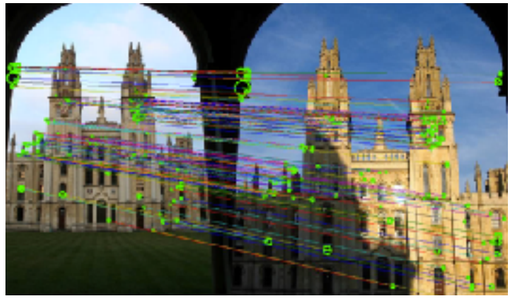
3. Advanced topics
   - Generative models; Vision & language
   - Human action recognition in videos
   - Vision for robotics
   - 3D computer vision

Vibrant portrait painting of Salvador Dali with a robotic half face

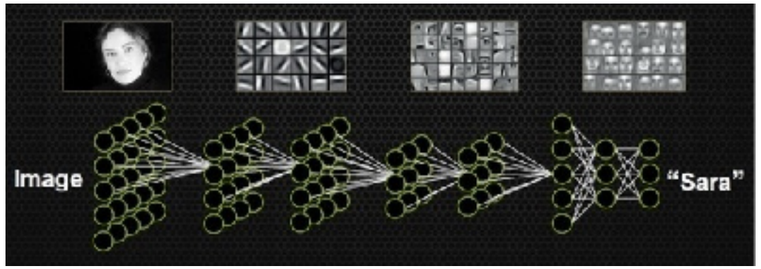
Course outline

1. Instance-level recognition
   - Camera geometry
   - Image processing
   - Image correspondence
   - Large-scale image and video search





2. Category-level recognition
   - Supervised learning
   - Neural networks for visual recognition
   - Object recognition, detection, and segmentation

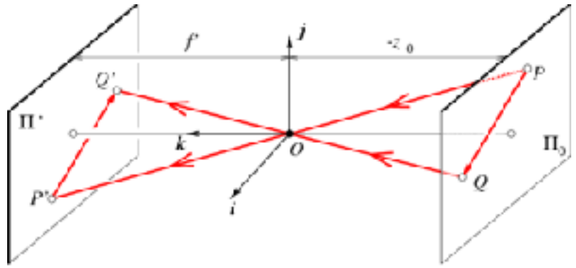


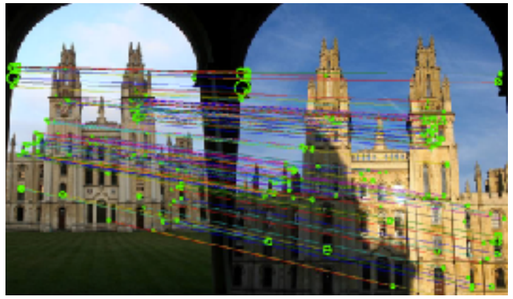
3. Advanced topics
   - Generative models; Vision & language
   - Human action recognition in videos
   - Vision for robotics
   - 3D computer vision

Vibrant portrait painting of Salvador Dali with a robotic half face

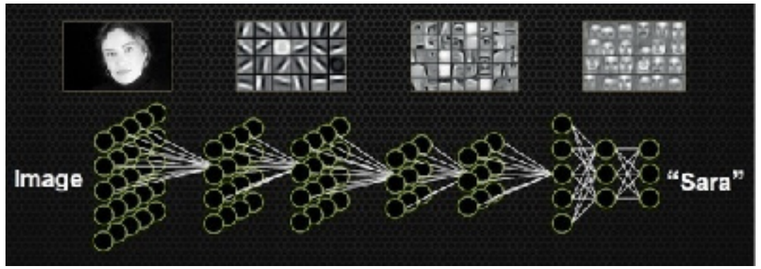
Course outline

1. Instance-level recognition
   - Camera geometry
   - Image processing
   - Image correspondence
   - Large-scale image and video search





2. Category-level recognition
   - Supervised learning
   - Neural networks for visual recognition
   - Object recognition, detection, and segmentation



3. Advanced topics
   - Generative models; Vision & language
   - Human action recognition in videos
   - Vision for robotics
   - 3D computer vision

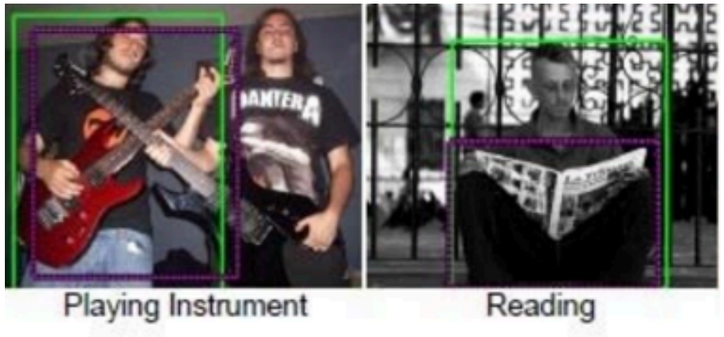
Vibrant portrait painting of Salvador Dali with a robotic half face

3. Advanced topics

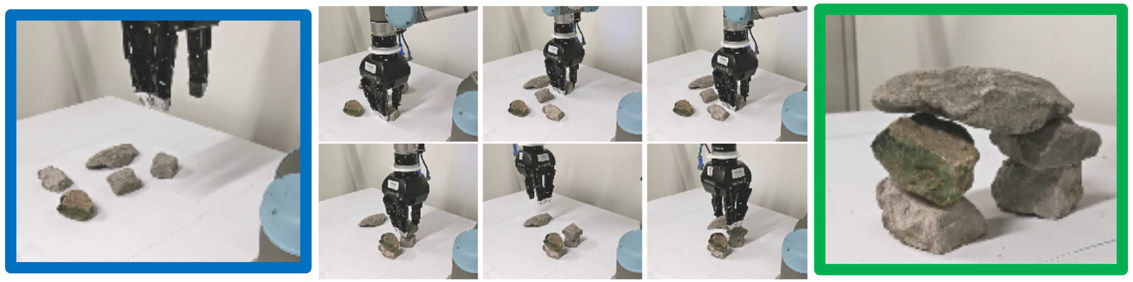
- Generative models; Vision & language

Vibrant portrait painting of Salvador Dali with a robotic half face

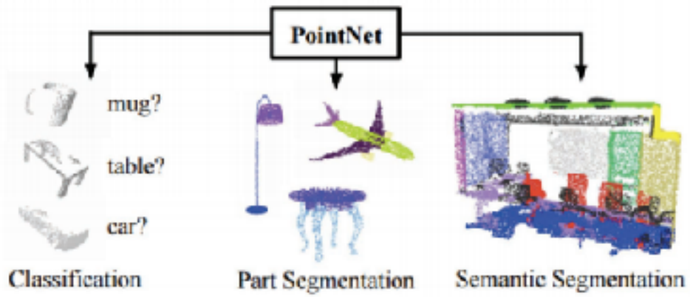
- Human action recognition in videos



- Vision for robotics



- 3D computer vision



Resources

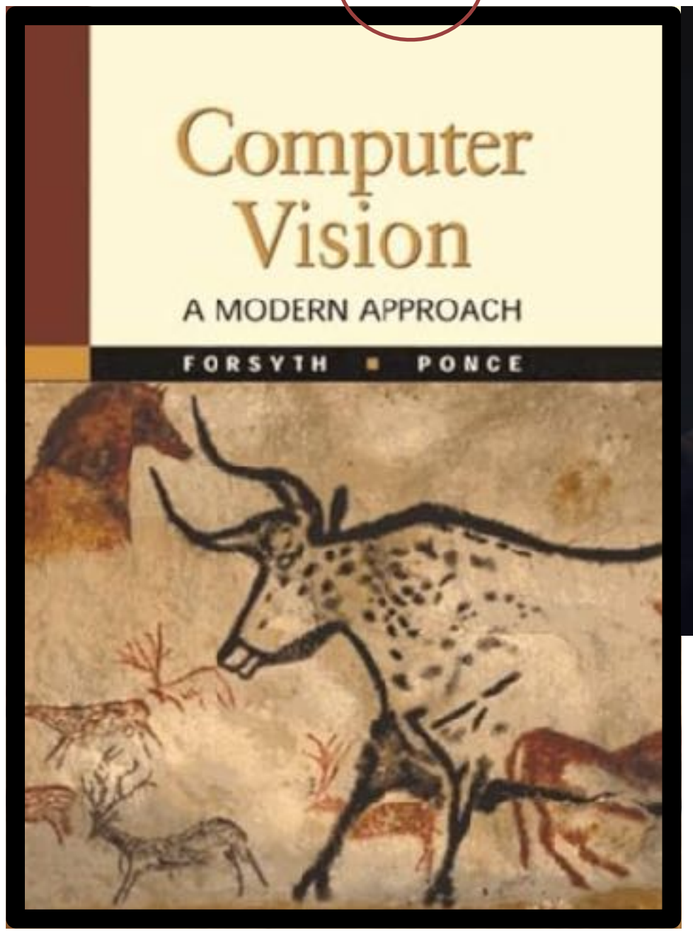
- D.A. Forsyth and J. Ponce, "Computer Vision: A Modern Approach", Prentice-Hall, 2nd edition, 2011
- J. Ponce, M. Hebert, C. Schmid and A. Zisserman "Toward Category-Level Object Recognition", Lecture Notes in Computer Science 4170, Springer-Verlag, 2007
- O. Faugeras, Q.T. Luong, and T. Papadopoulo, "Geometry of Multiple Images", MIT Press, 2001.
- R. Hartley and A. Zisserman, "Multiple View Geometry in Computer Vision", Cambridge University Press, 2004.
- J. Koenderink, "Solid Shape", MIT Press, 1990
- R. Szeliski, "Computer Vision: Algorithms and Applications, 2nd ed.", 2022. Online book.
- Computer Vision: Models, Learning, and Inference by Simon J.D. Prince (2012)
- Understanding Deep Learning by Simon J.D. Prince (2023)
- Deep Learning by I. Goodfellow, Y. Bengio and A. Courville (2016)
- Michael Nielsen's online book on Neural Networks and Deep Learning (2019)
- David Forsyth's Applied Machine Learning textbook draft (2019)
- Andrej Karpathy blog

...

Let me know if you find a good resource (book, videos etc) to link from the webpage.

Resources

- D.A. Forsyth and J. Ponce, "Computer Vision: A Modern Approach", Prentice-Hall, 2nd edition, 2011



l Object Recognition", Lecture Notes in Computer Science 4170,

ges", MIT Press, 2001.

n", Cambridge University Press, 2004.

2. Online book.

012)

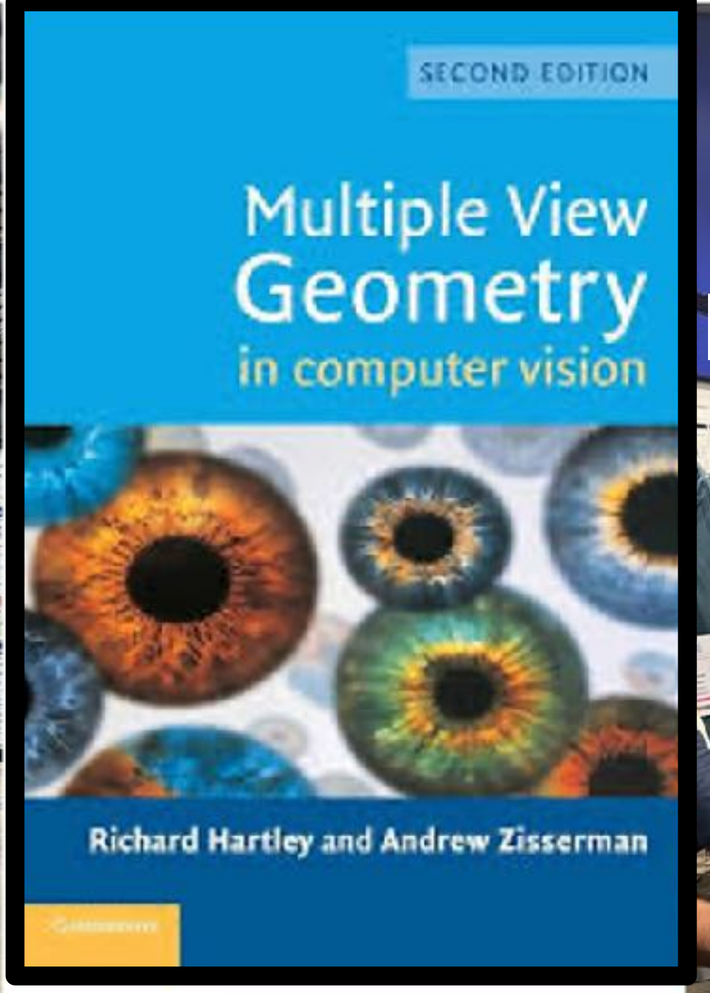
019)

Jean Ponce

Next Tue (Oct 15)

Last week at European Conference on Computer Vision (ECCV) 2024

Richard Hartley



Andrew Zisserman

Photo credit: Kostas Daniilidis

Publication Venues

Read “good” papers

1) Conference on **C**omputer **V**ision and **P**attern **R**ecognition (**CVPR**)
2) **I**nternational **C**onference on **C**omputer **V**ision (**ICCV**)
3) **E**uropean **C**onference on **C**omputer **V**ision (**ECCV**)
4) Transactions on **P**attern **A**nalysis and **M**achine **I**ntelligence (**TPAMI**)
5) **I**nternational **J**ournal of **C**omputer **V**ision (**IJCV**)

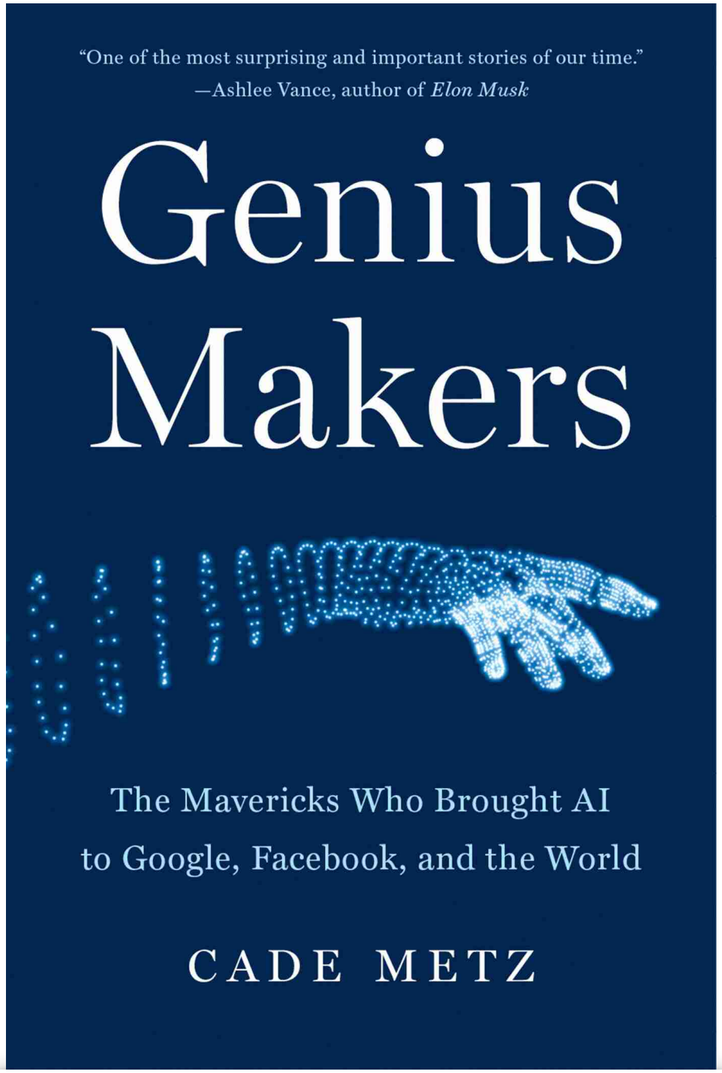
Conference proceedings

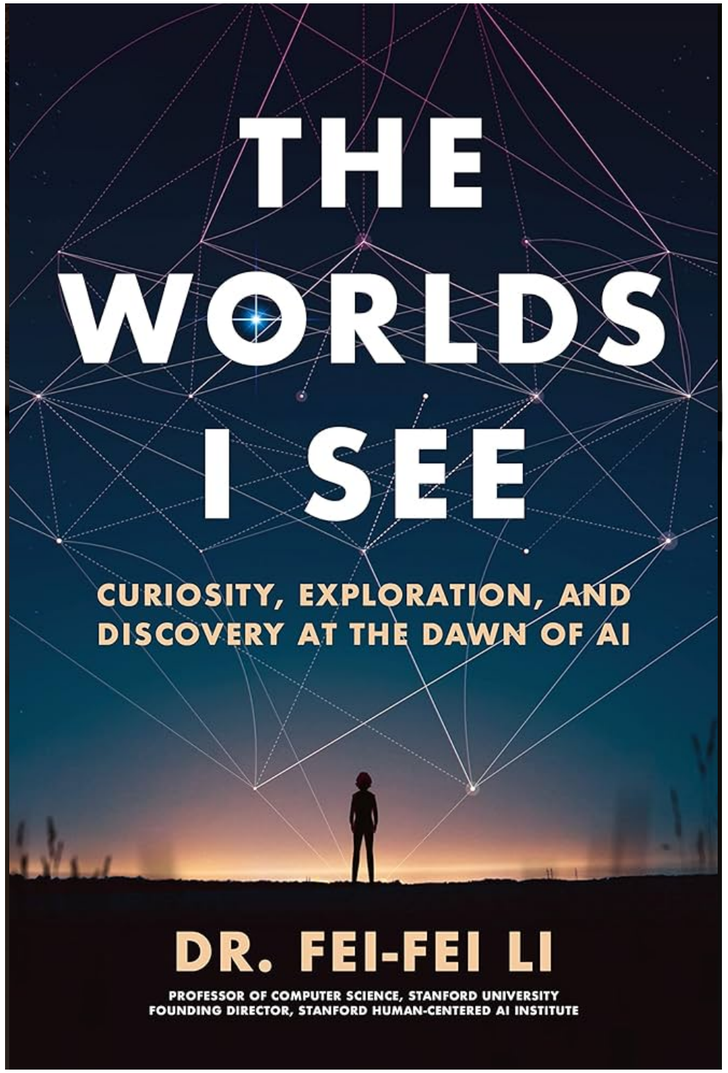
Journals

…
ACCV, WACV, BMVC, CVIU…

…
NeurIPS, ICLR, ICML…
EMNLP…
IROS…

Reading recommendations



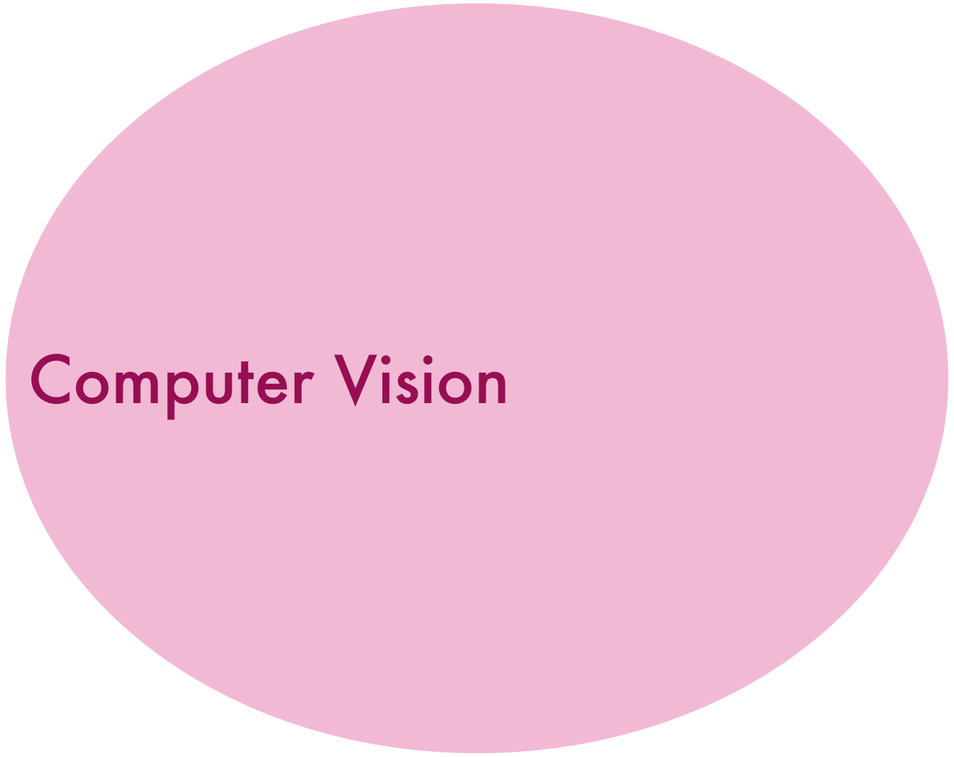


Recap

1. Register on the Google Classroom
    - Assignment submissions, discussions and announcements will be done on Google Classroom.
    - Assignment 1 – Instance-level recognition (due Oct 29 2024)
2. Fill-in TAs session participation form (by Oct 15)

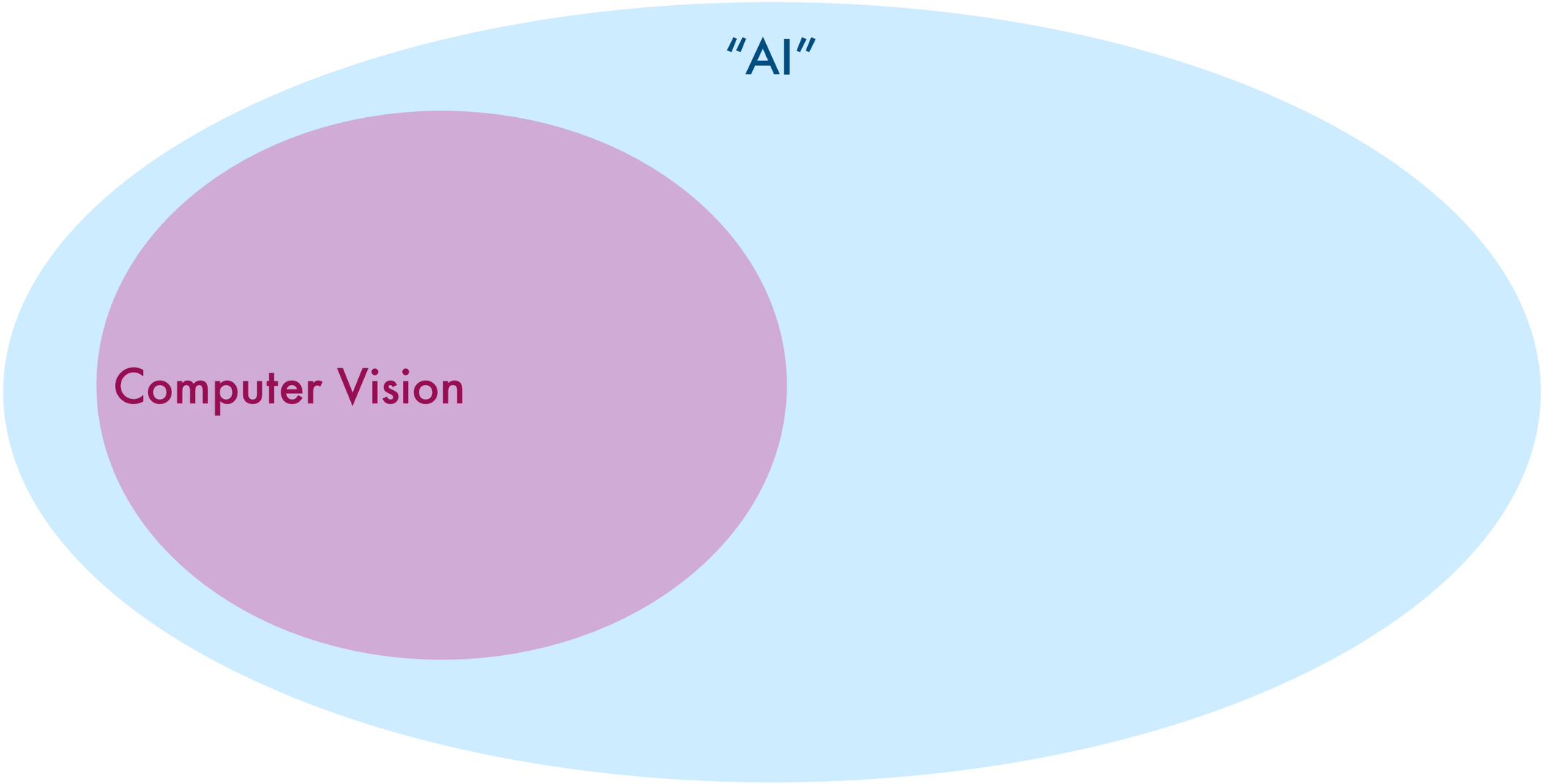
Computer Vision 101

Definitions



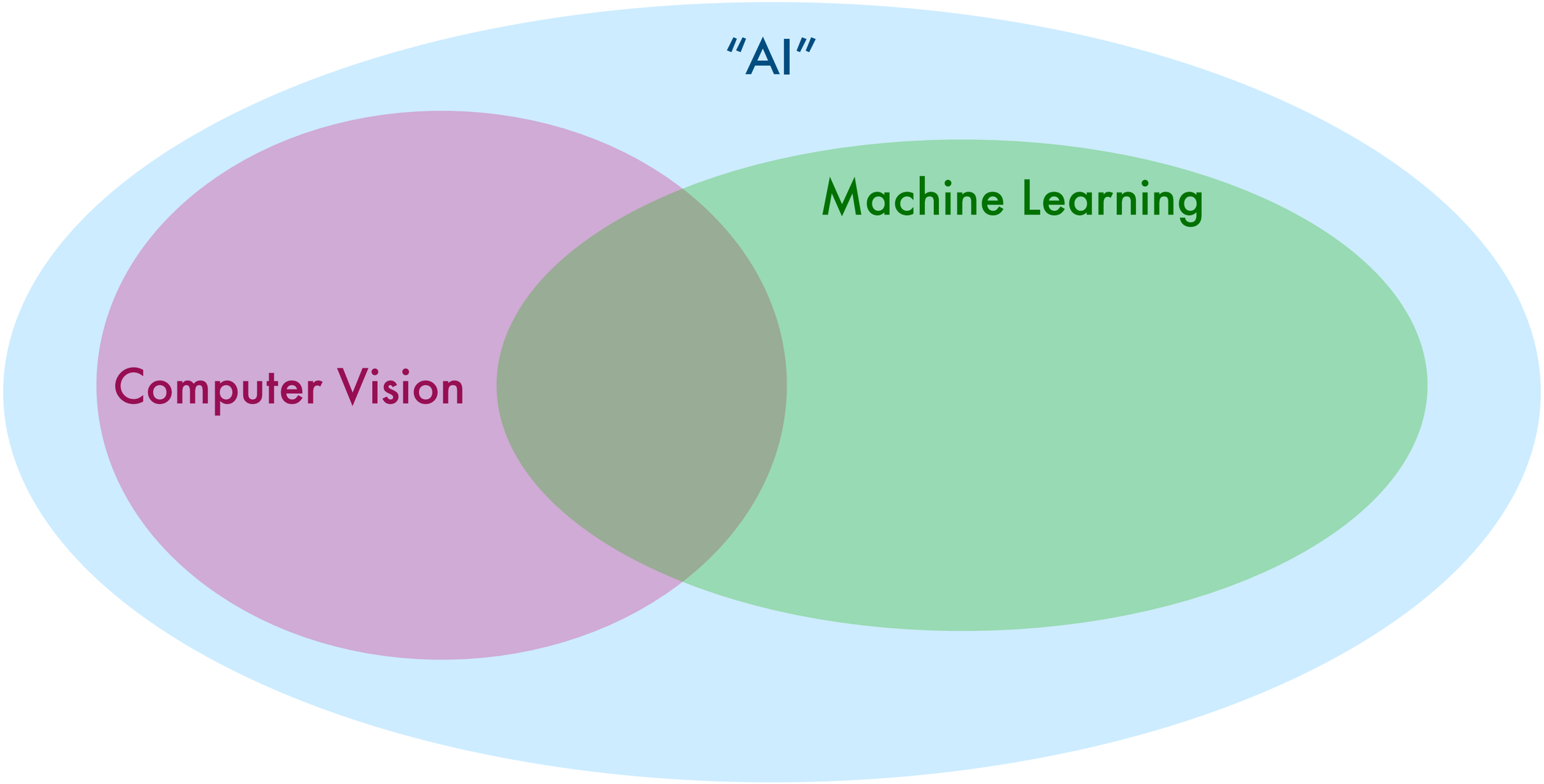
Slide credit: Justin Johnson

Definitions



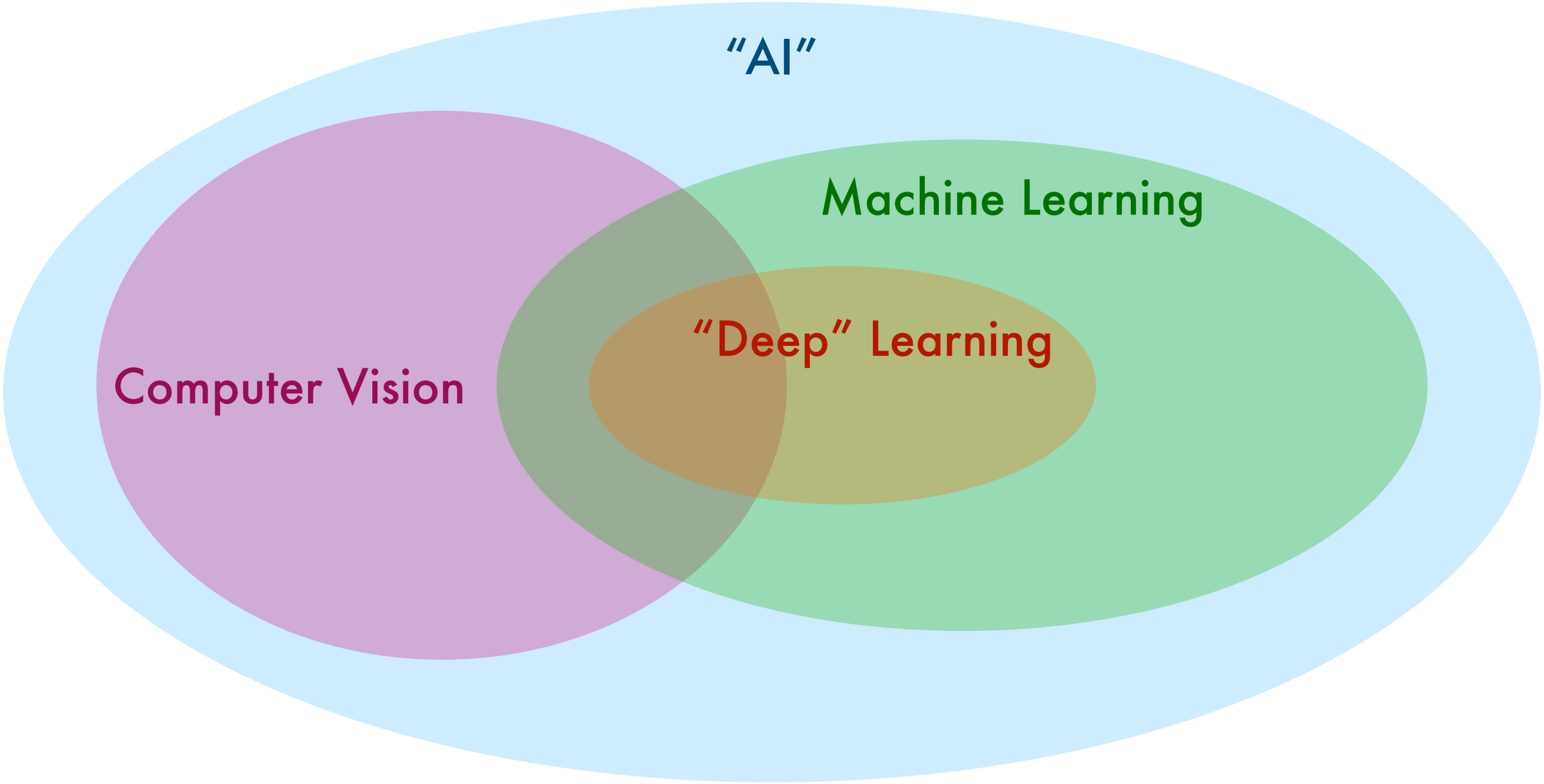
Slide credit: Justin Johnson

Definitions



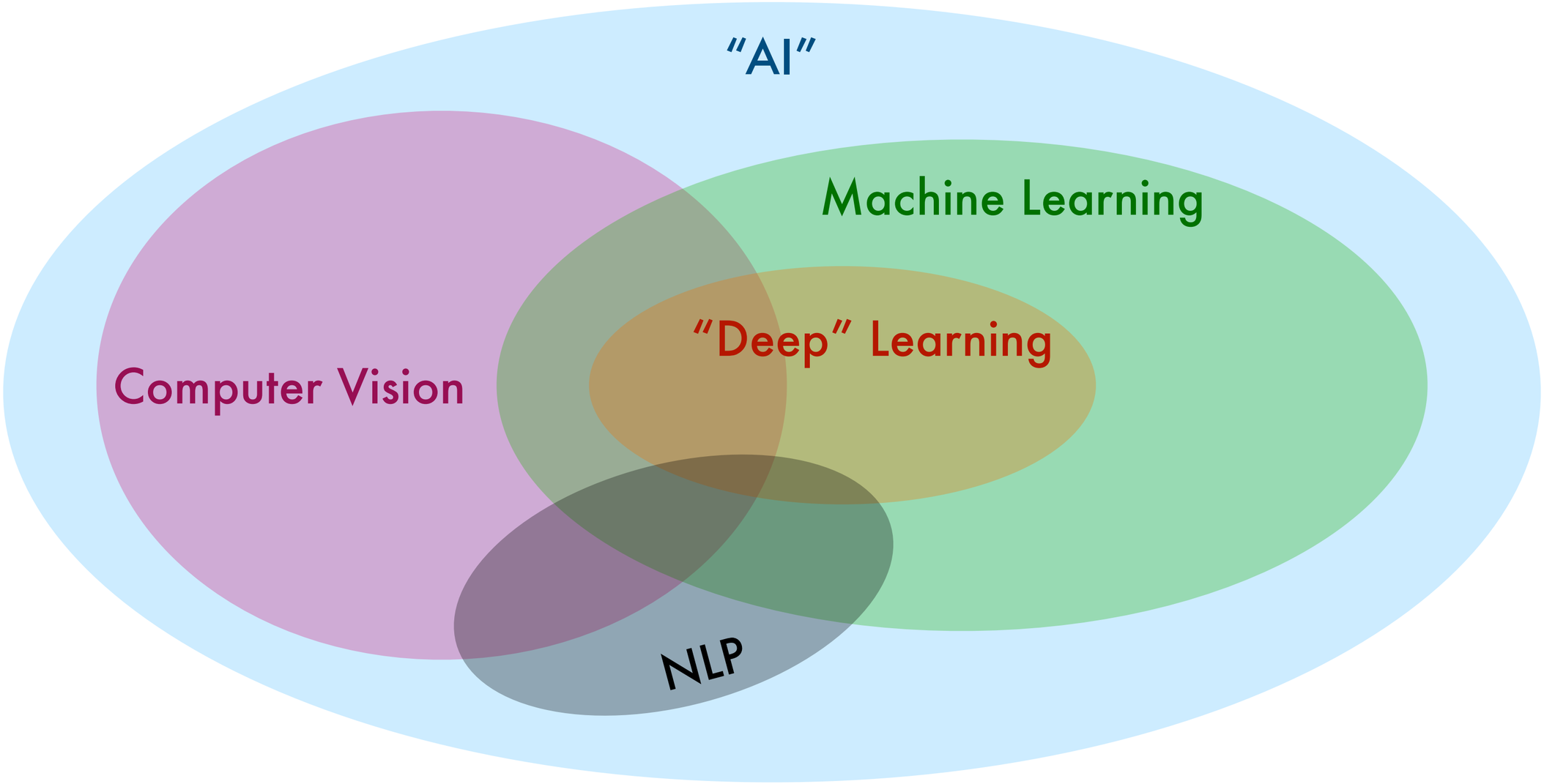
Slide credit: Justin Johnson

Definitions



Slide credit: Justin Johnson

Definitions



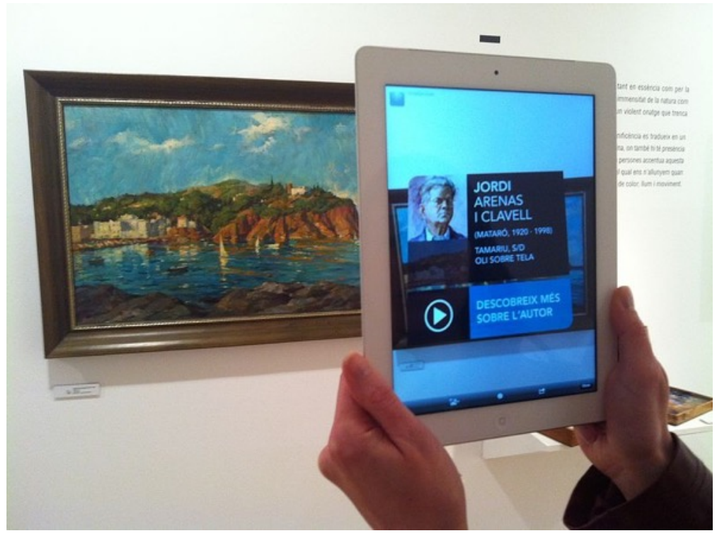
Slide credit: Justin Johnson

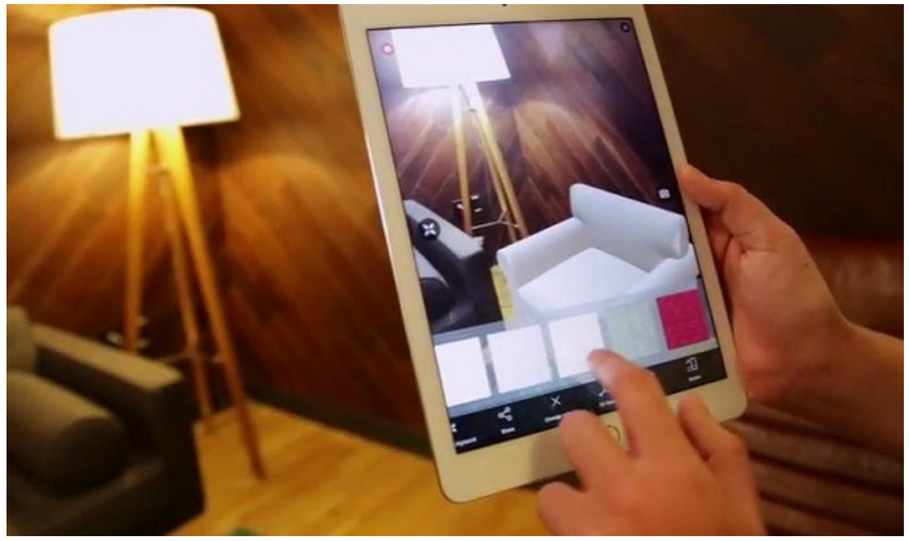
Definitions

"AI"	Any technique that enables **computers** to **mimic human** behavior
Machine Learning	Ability to learn **without** explicitly being **programmed**
"Deep" Learning	Extract patterns from data using **neural networks**
Computer Vision	Extracting meaning from **visual** signals
NLP	Extracting meaning from **textual** signals

Slide credit: Alexander Amini

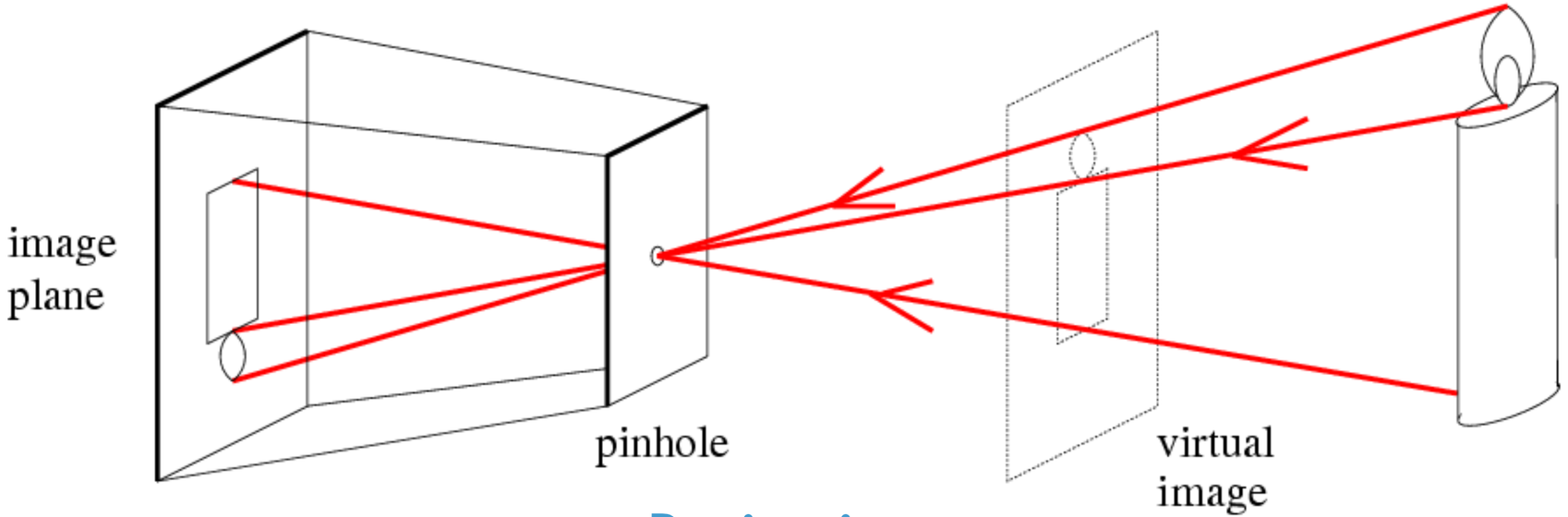
What is computer vision?





Slide credit: Fei-Fei Li & Justin Johnson & Serena Yeung

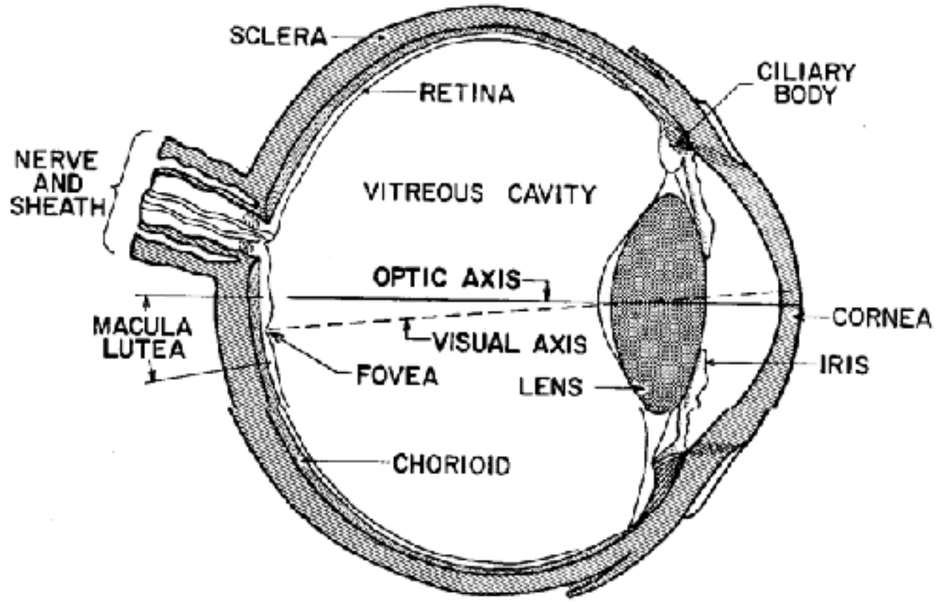
Photograph: Projection of the 3D world into 2D images



2D image

Projection

3D world



More about camera geometry next week

Slide credit: Jean Ponce

Slide credit:
Kosta Derpanis

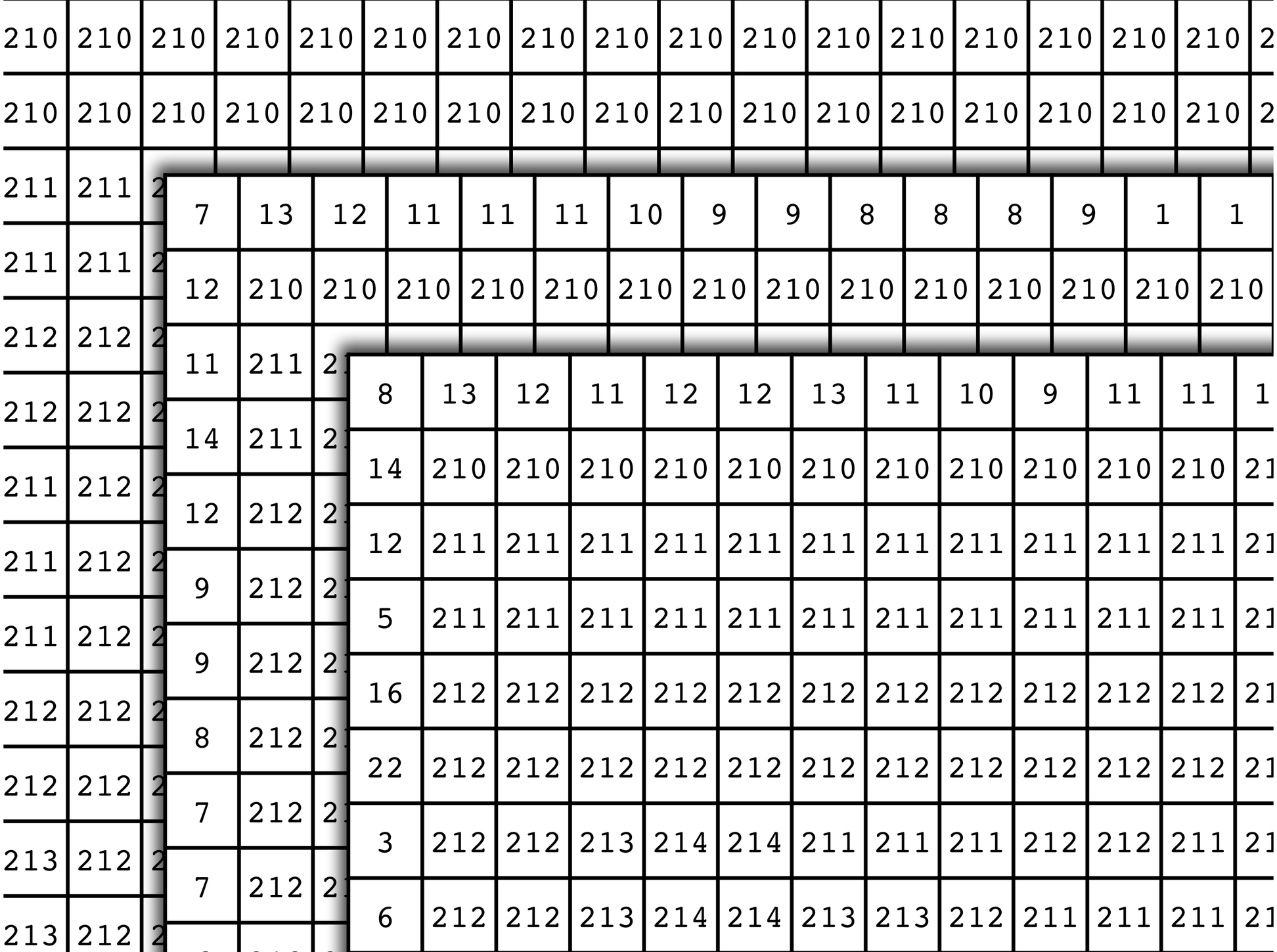
Slide credit:
Kosta Derpanis

Slide credit:
Kosta Derpanis

Slide credit:
Kosta Derpanis

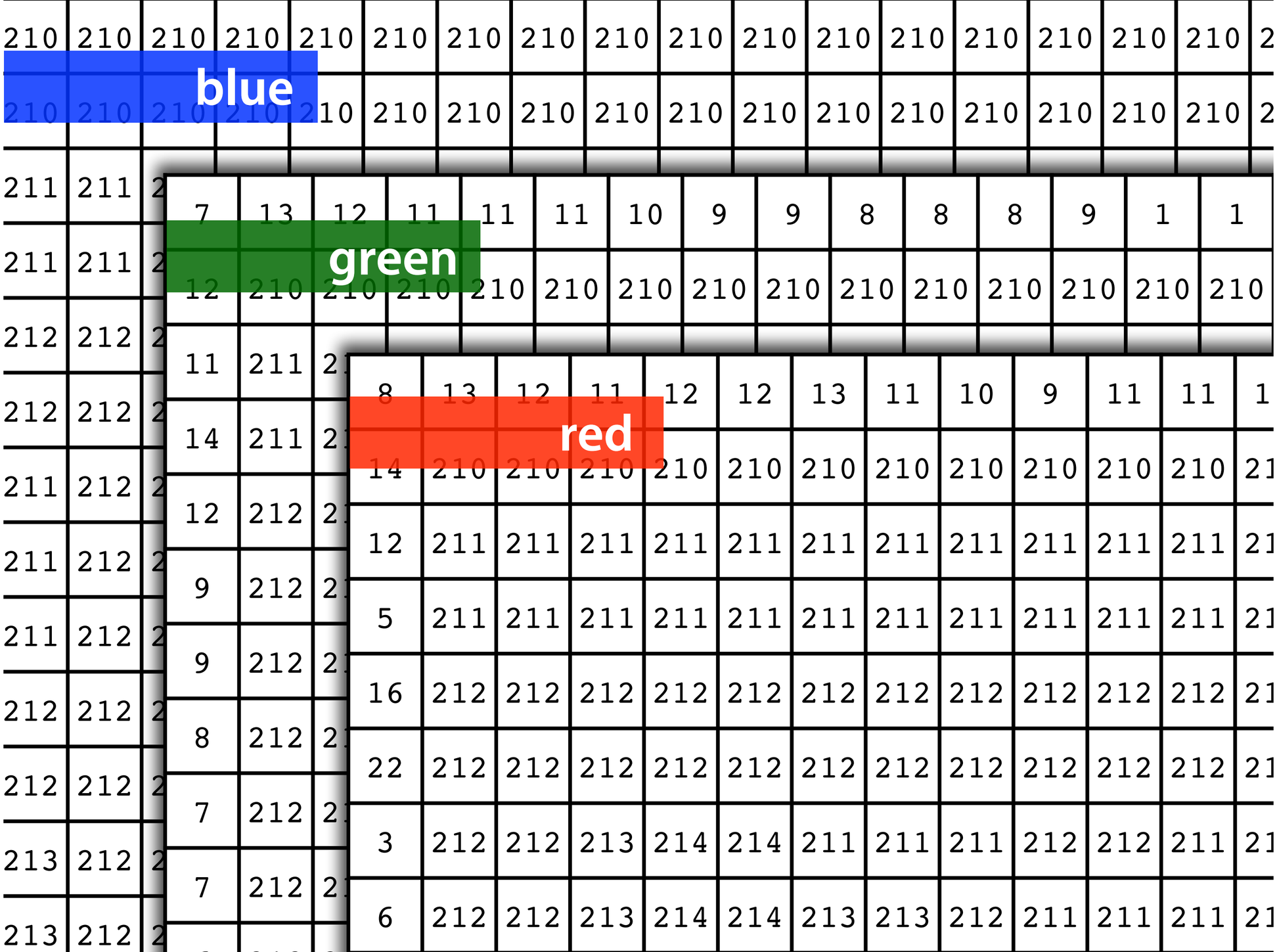
8	13	12	11	12	12	13	11	10	9	11	11	13	12	10	8	8	2
14	210	210	210	210	210	210	210	210	210	210	210	210	210	210	210	210	2
12	211	211	211	211	211	211	211	211	211	211	211	211	211	211	211	211	2
5	211	211	211	211	211	211	211	211	211	211	211	211	211	211	211	211	2
16	212	212	212	212	212	212	212	212	212	212	212	212	212	212	212	212	2
22	212	212	212	212	212	212	212	212	212	212	212	212	212	212	212	212	2
3	212	212	213	214	214	211	211	211	212	212	211	212	212	213	214	214	2
6	212	212	213	214	214	213	213	212	211	211	211	212	212	213	214	214	2
5	212	212	213	213	213	215	214	212	210	209	211	212	212	213	213	213	2
1	212	212	212	212	212	214	212	210	208	208	212	212	212	212	212	212	2
13	212	212	212	212	212	210	209	209	208	209	212	212	212	212	212	212	2
13	212	212	211	211	211	209	209	209	210	210	213	212	212	211	211	211	2
213	212	212	211	210	210	210	211	212	213	211	213	212	212	211	210	210	2

Slide credit:
Kosta Derpanis

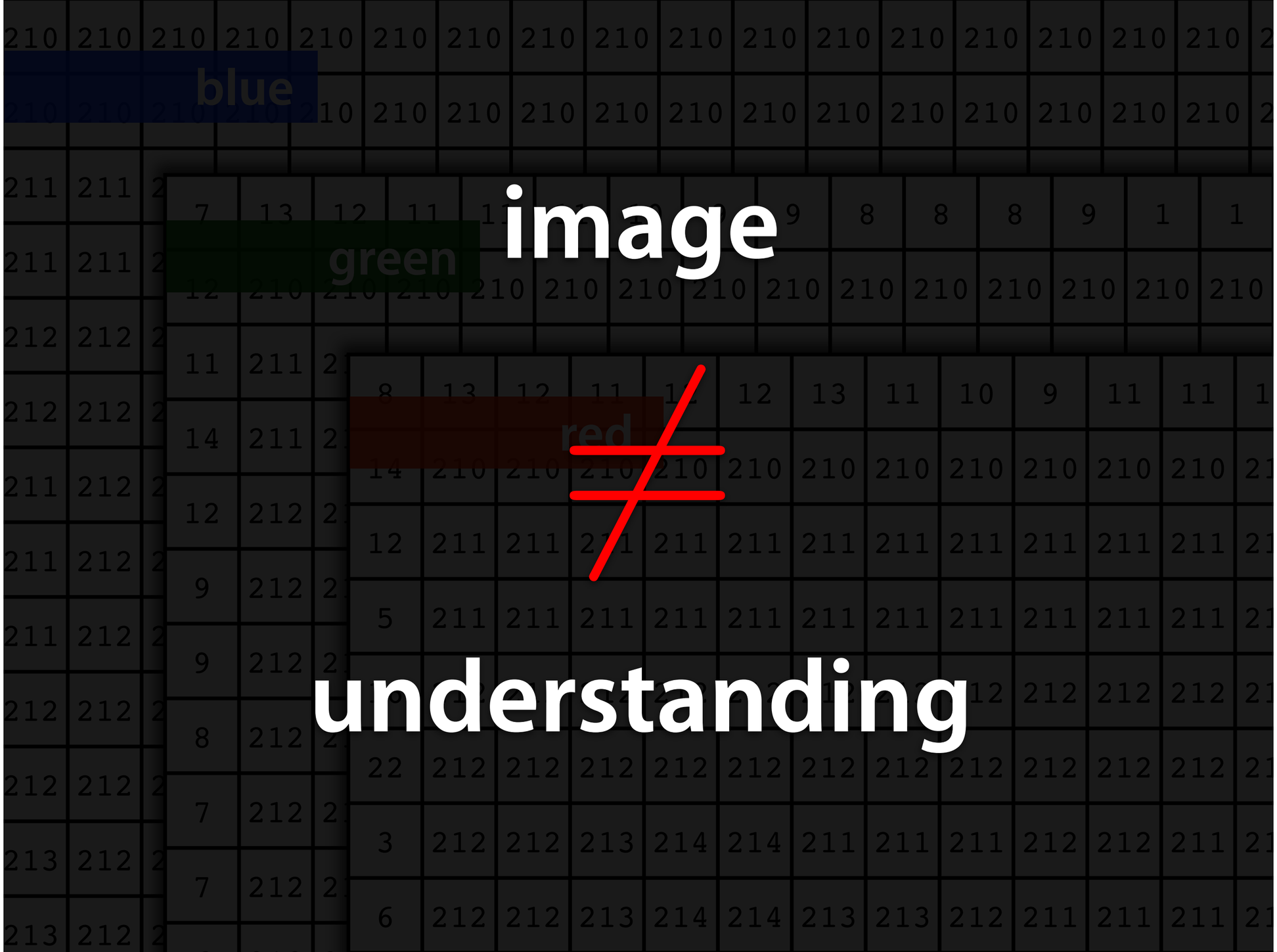


Slide credit:
Kosta Derpanis

RGB Pixels



Slide credit:
Kosta Derpanis



Slide credit:
Kosta Derpanis

What is computer vision?

Extracting meaning from visual signals *

Object recognition,
Object detection,
Pixel-level segmentation,
3D localization,
etc.

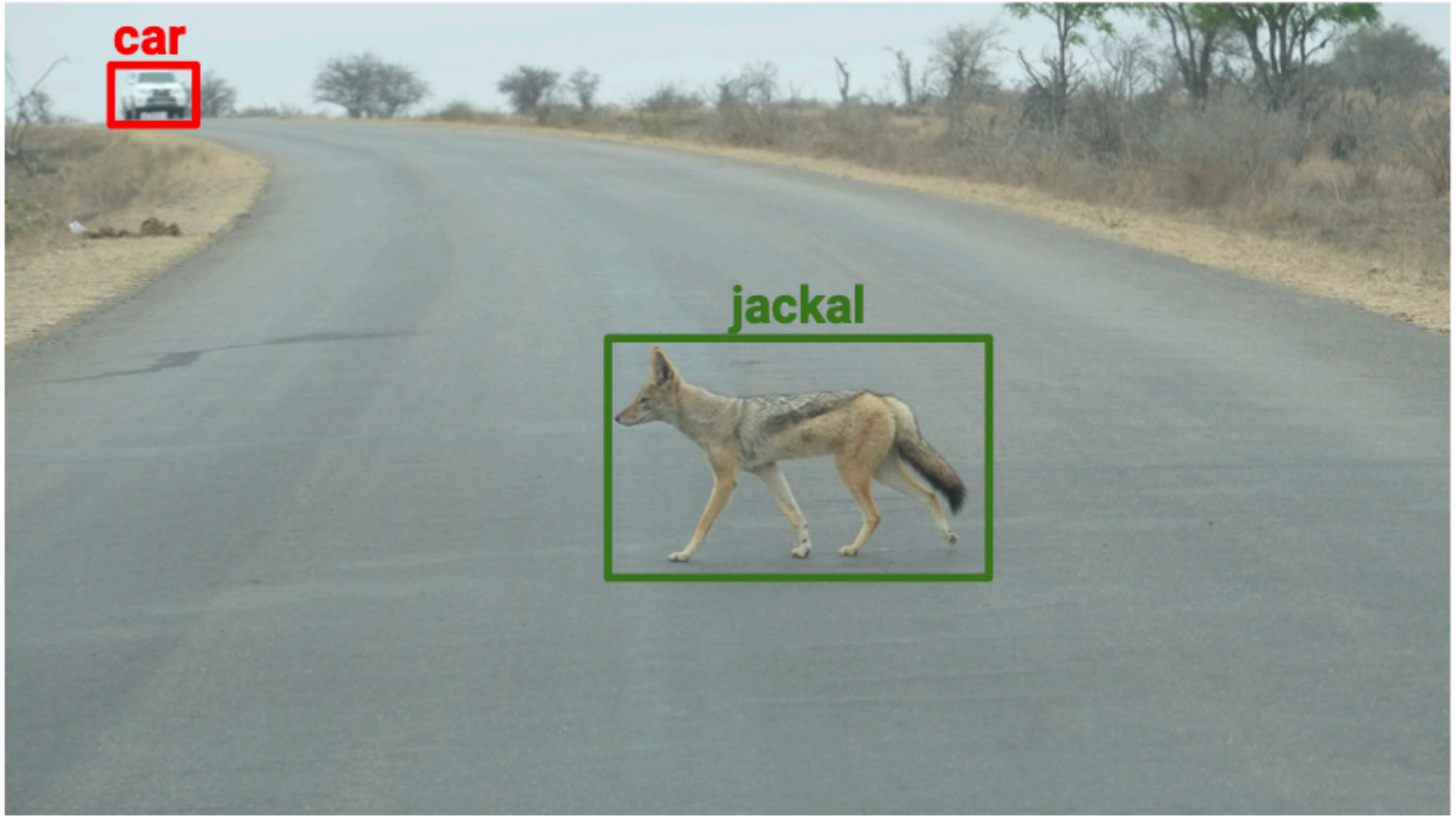
*Visual signal: Image, video, depth, 3D point cloud, MRI, scans, …

Slide credit: Naila Murray

Example tasks

Slide credit: Naila Murray

Object recognition and localization (detection)



Slide credit: Naila Murray

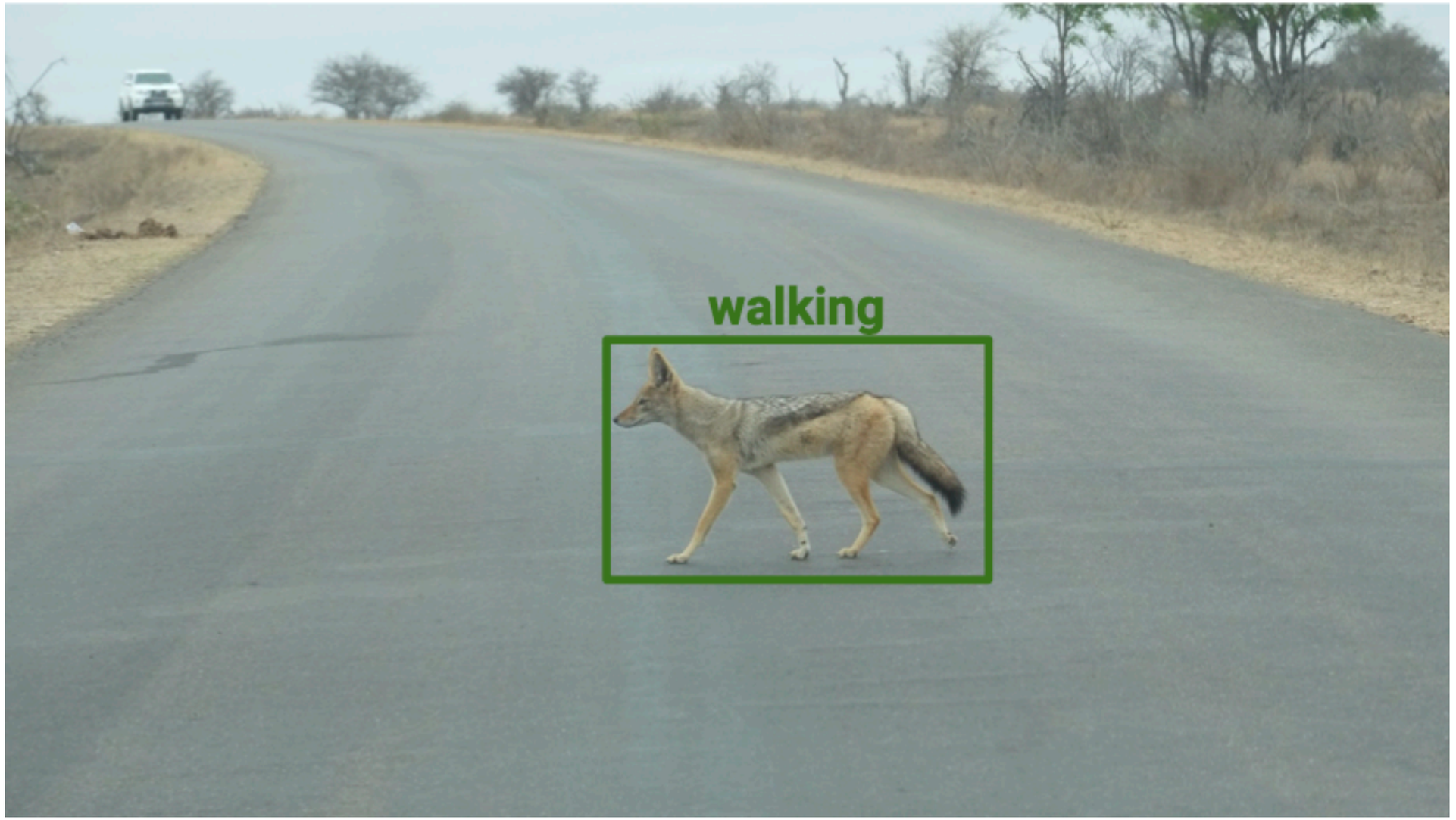
Visual question answering

Q: Is this an outdoor scene?
A: Yes

Q: What is the weather like?
A: Cloudy but dry

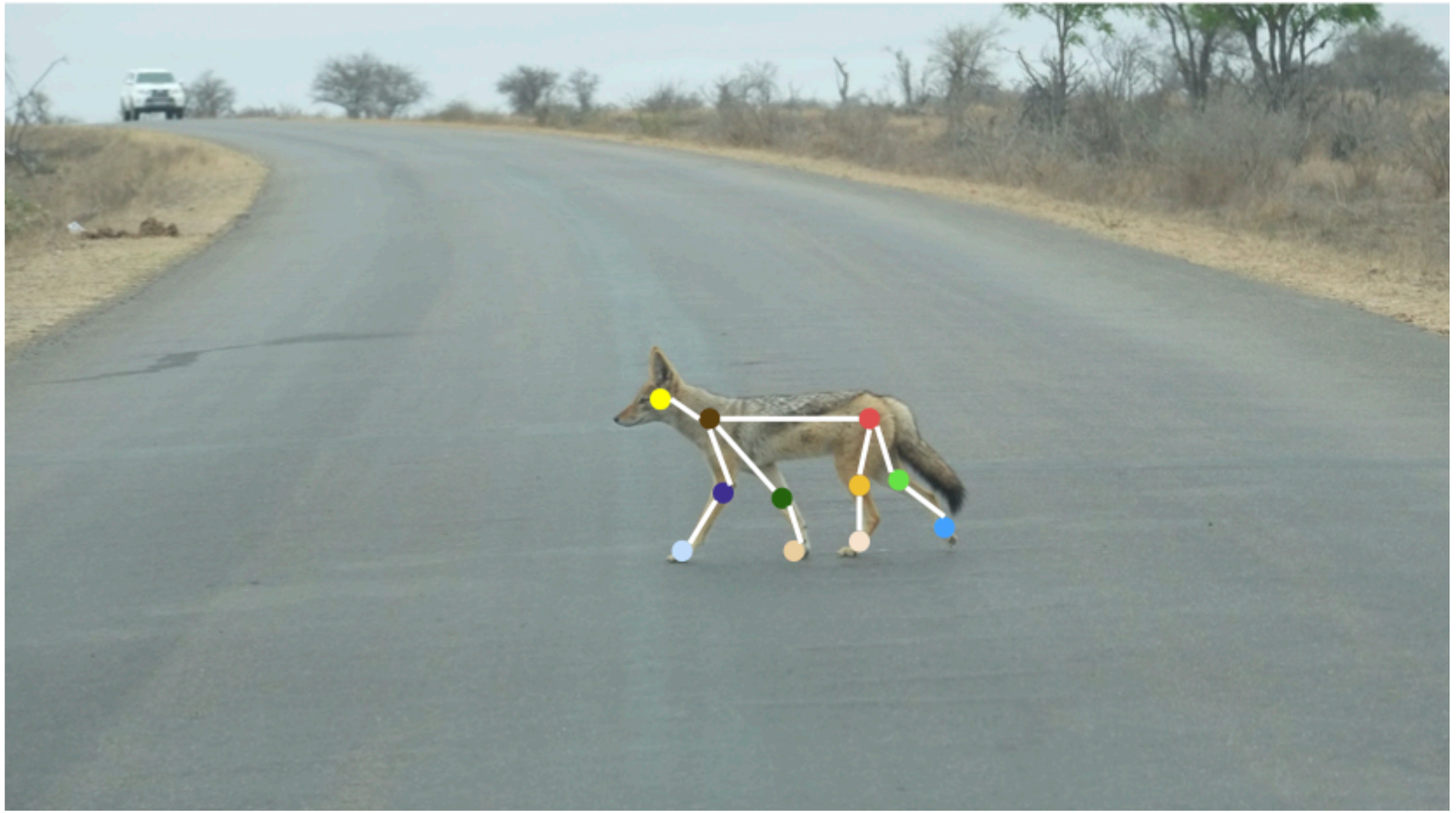
Slide credit: Naila Murray

Activity recognition



Slide credit: Naila Murray

Pose estimation



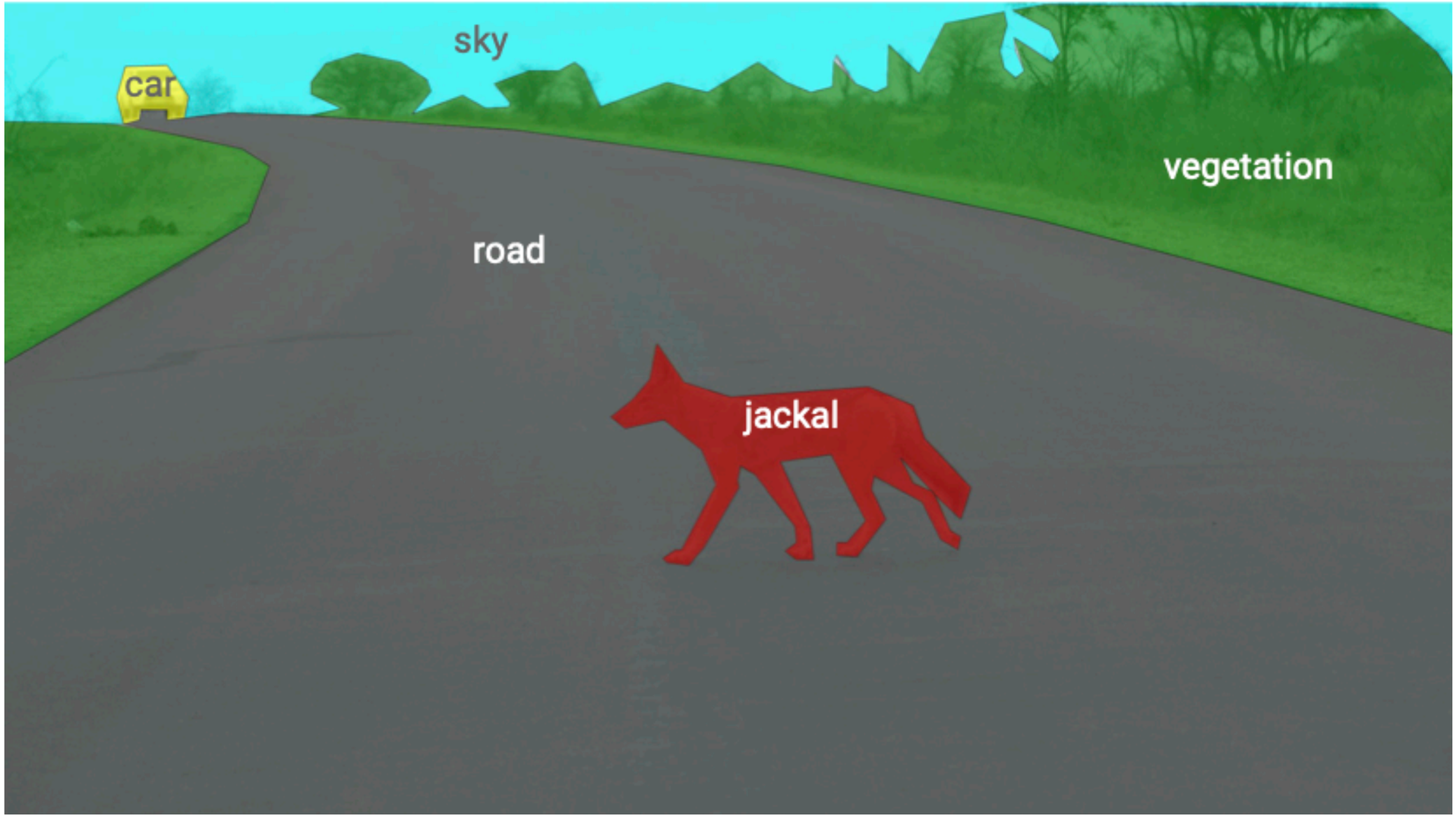
Slide credit: Naila Murray

Captioning

A jackal walking across a rural asphalt road

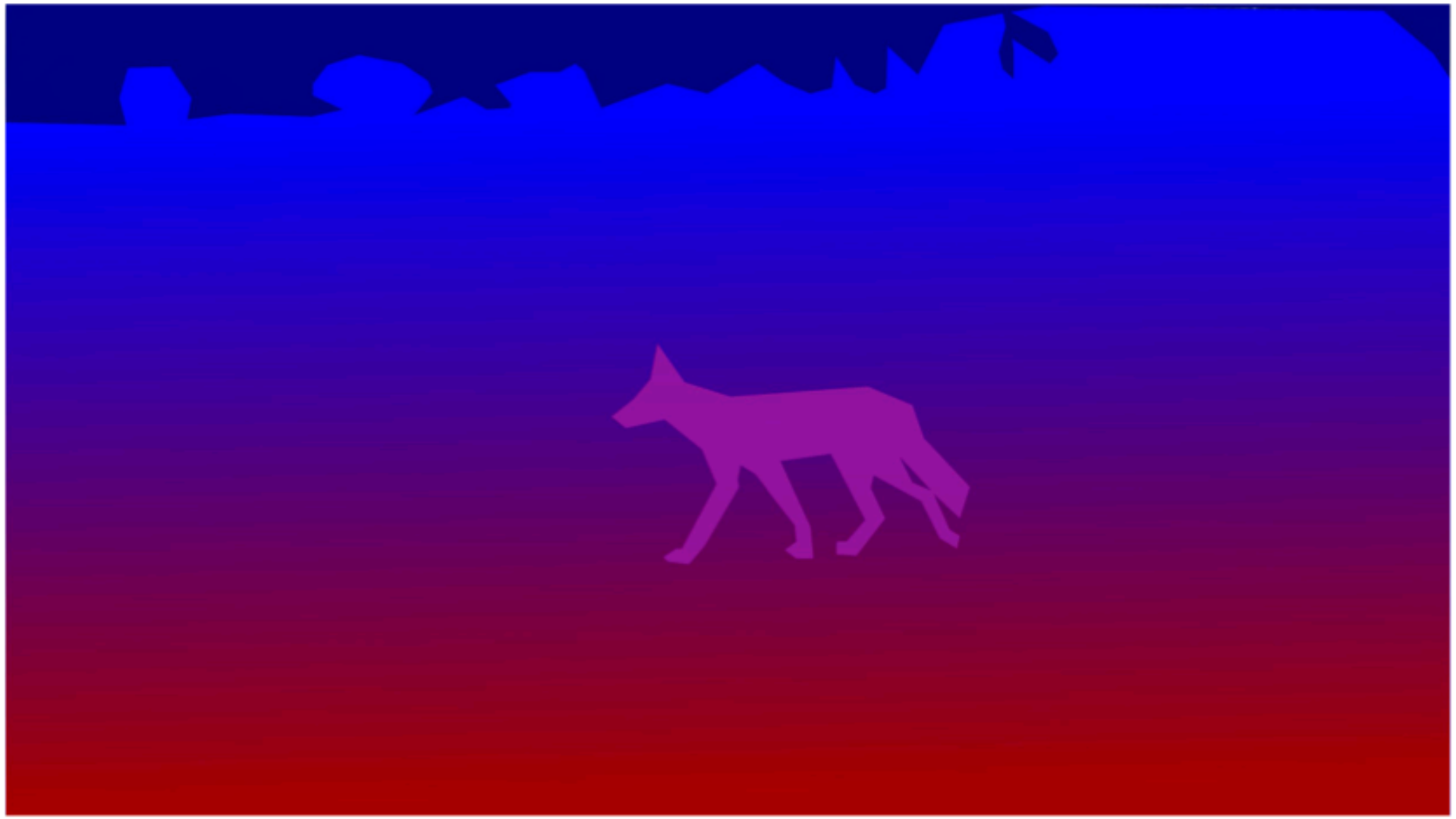
Slide credit: Naila Murray

Semantic segmentation



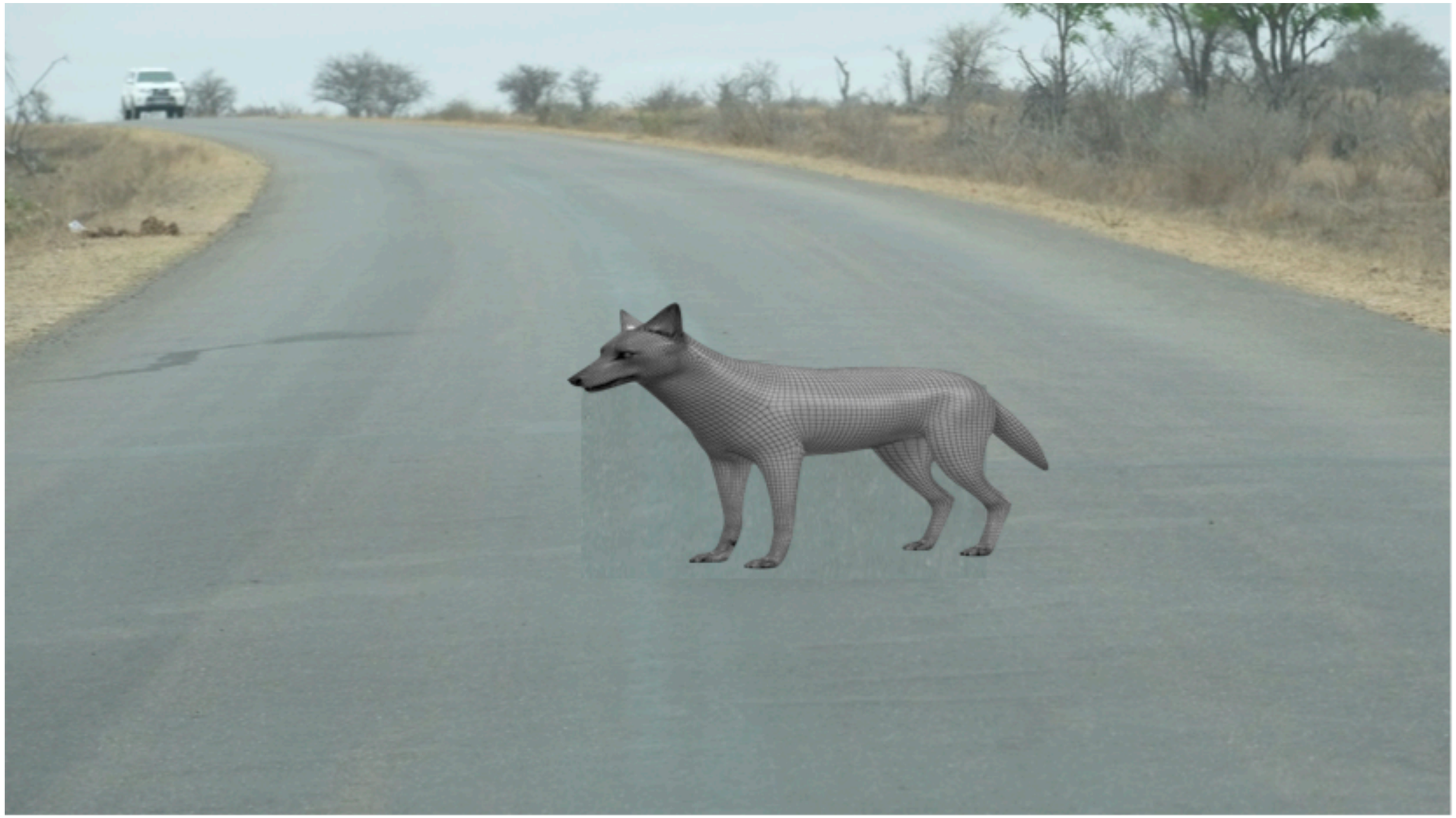
Slide credit: Naila Murray

Depth estimation



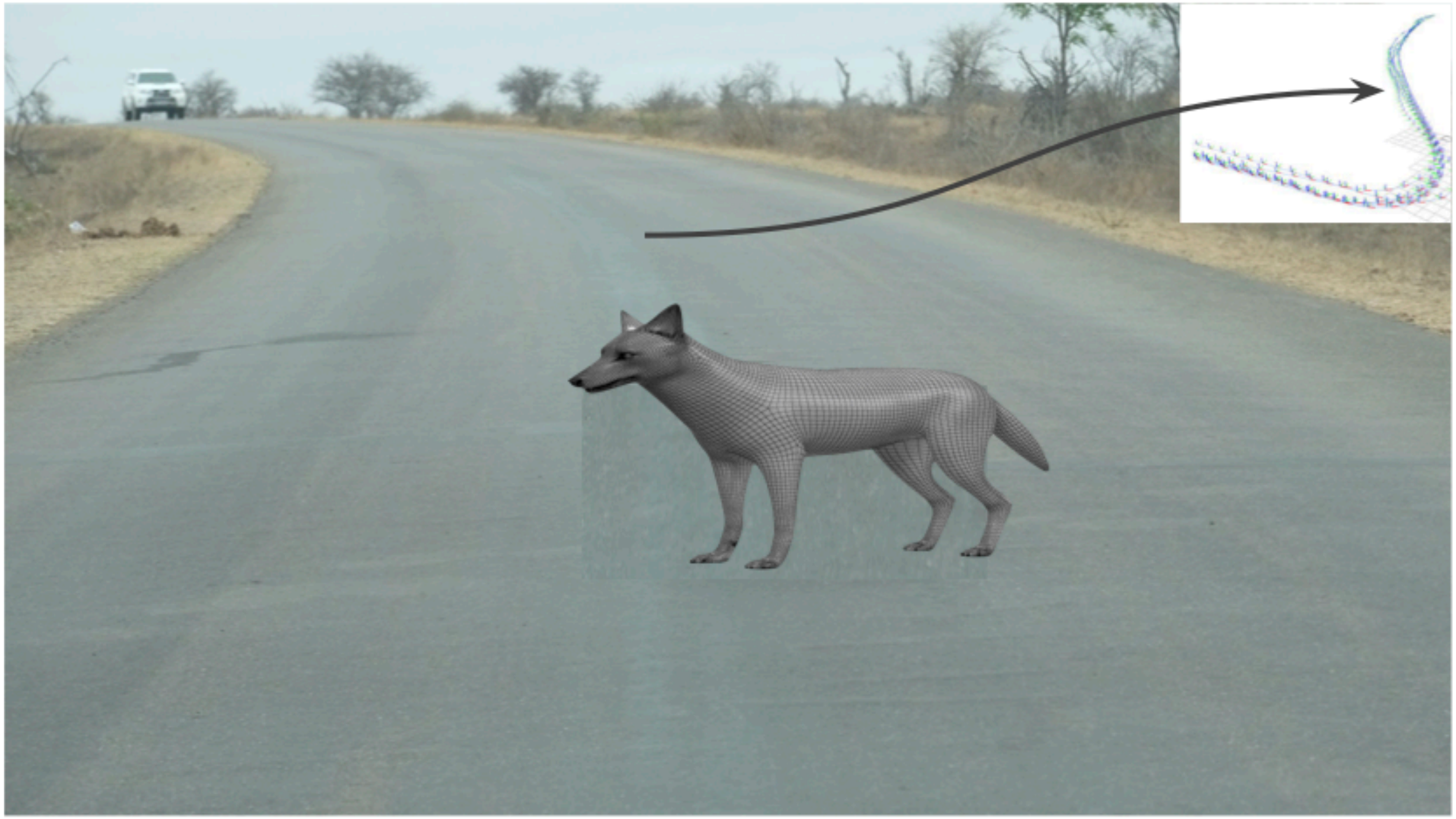
Slide credit: Naila Murray

3D shape estimation



Slide credit: Naila Murray

Visual localization



Slide credit: Naila Murray

Applications

Face detection

Slide credit: Kosta Derpanis

Self-driving cars

Slide credit: Kosta Derpanis

Shopping

Slide credit: Kosta Derpanis

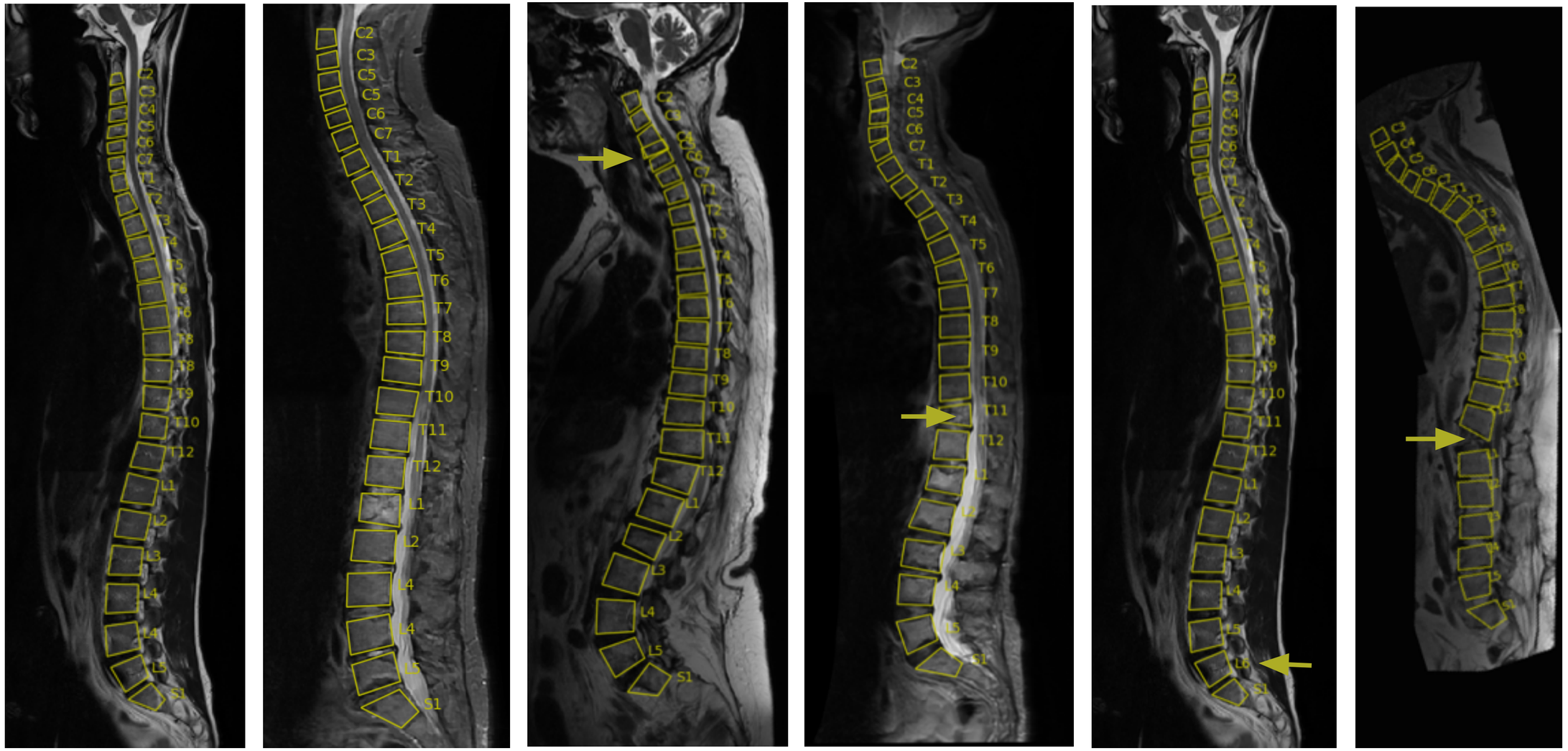
Google street view

Slide credit: Kosta Derpanis

Robotics

By: Igor Kalevatykh

Medicine

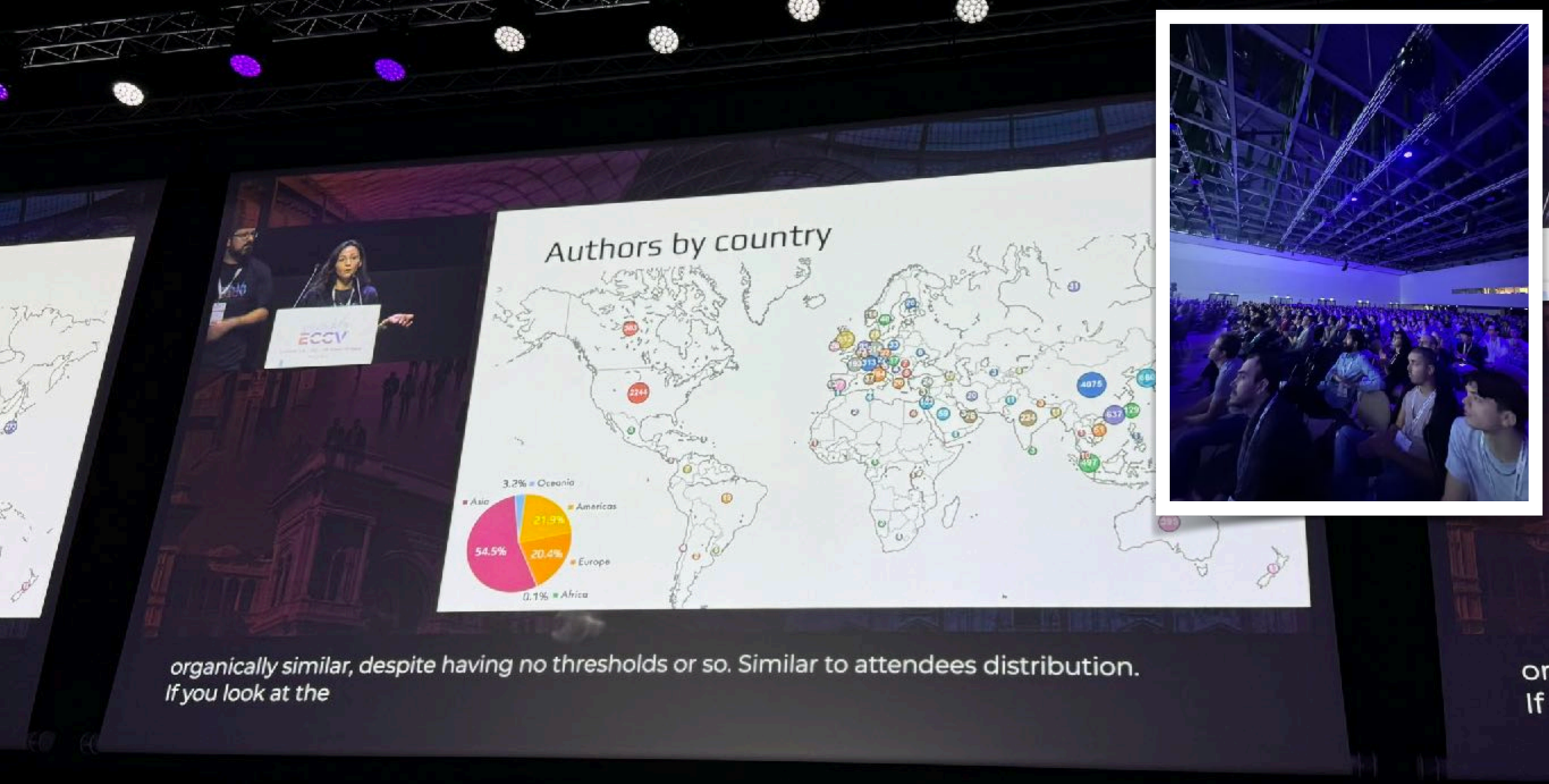


[Windsor 2020]

Accessibility

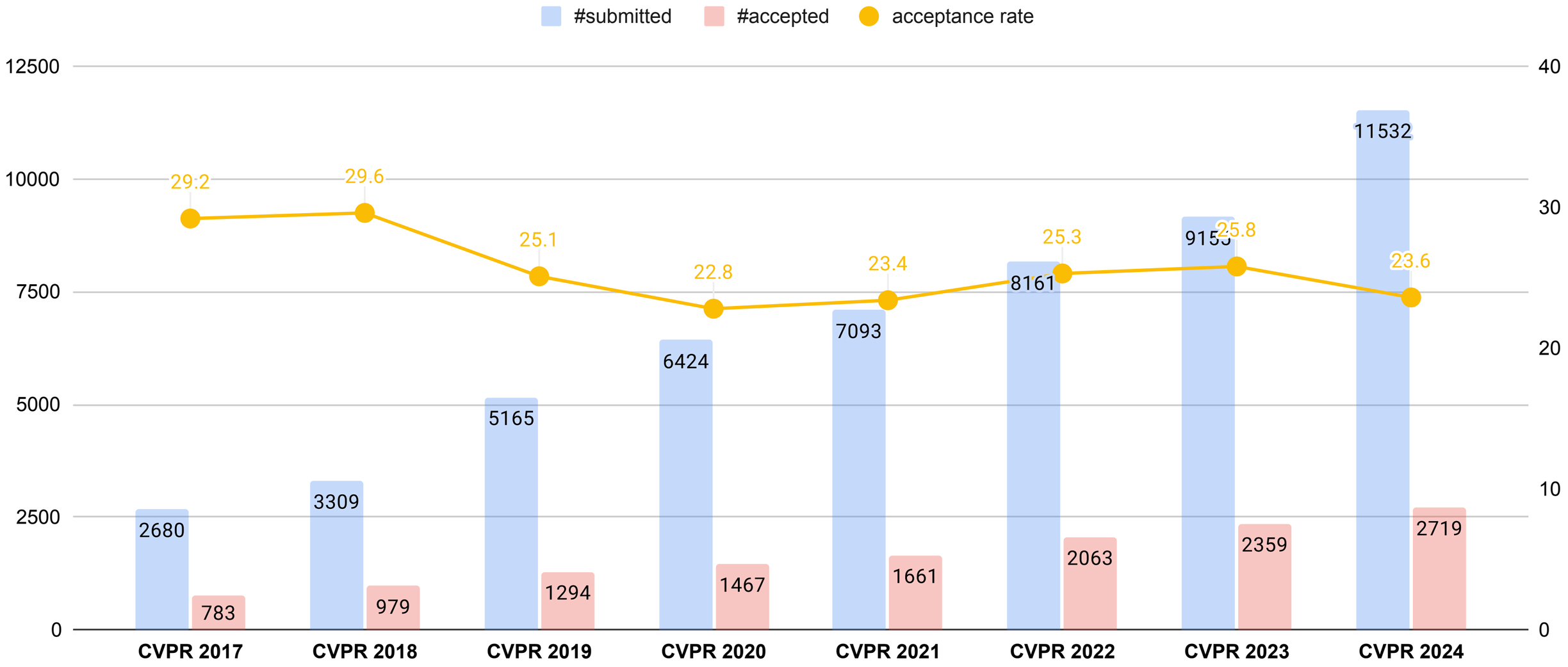
https://medium.com/sap-machine-learning-research/try-9e1ed9ae09ed

Some thoughts



ECCV 2024 Opening Ceremony @Milan
Oct 1, 2024

Over the years



Subjective View of the Computer Vision Field

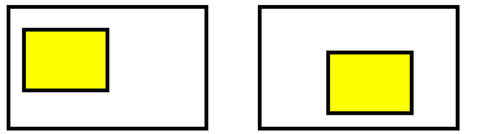
- Things work => Money => Big community => competition => unhealthy…
  - "Naughtiest of the community is very naughty" (D. Forsyth)
- CV today is very different than CV in 2023… (fast moving field)
- The community largely moved beyond cat/dog object classification.
  - Recent trends: open vocabulary recognition, vision & language, robotics, multimodal video data, continual learning…
  - "Low-level" tasks are sometimes building blocks for more complex pipelines: object detection, segmentation, flow…

Typical Challenges

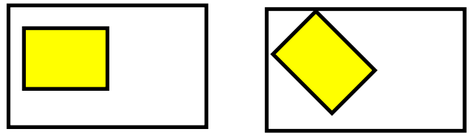
- Increasing computational costs (especially data sizes)
- Data licensing (privacy [faces], copyright [movies])
- Annotation cost (time cost for watching videos, need for experts for sign language…)
- Difficulty of evaluation (metrics for generative models), bias in test sets (does not evaluate generalization in the wild)

Instance-level recognition

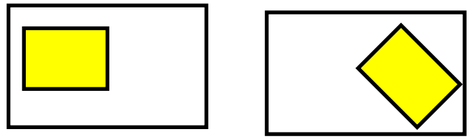
Geometry recap: Hierarchy of 2D Geometric Transformations



- Translation (T)



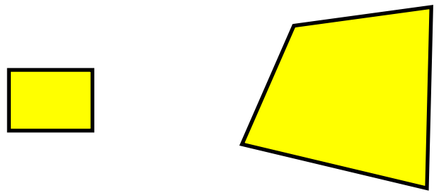
- Rotation (R)



- Euclidean / Rigid (R+T) $\begin{bmatrix} r_{11} & r_{12} & t_x \\ r_{21} & r_{22} & y_y \\ 0 & 0 & 1 \end{bmatrix}$

- Similarity (+ scaling) $\begin{bmatrix} sr_{11} & sr_{12} & t_x \\ sr_{21} & sr_{22} & y_y \\ 0 & 0 & 1 \end{bmatrix}$

- Affine (+ shear) $\begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$



- Projective / Homography $\begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}$

Preserves:

Lengths, angles

Angles, ratios of lengths

Parallelism

Collinearity

Agenda: Instance-level recognition

1) Introduction to local features

2) Interest point detectors (e.g., Harris, scale invariance)

3) Comparison of patches (SSD, ZNCC on pixel values)

4) Feature descriptors (e.g., SIFT)

5) Matching and recognition with local features

6) Local feature aggregation for a single image-level description

Agenda: Instance-level recognition

1) Introduction to local features

2) Interest point detectors (e.g., Harris, scale invariance)

3) Comparison of patches (SSD, ZNCC on pixel values)

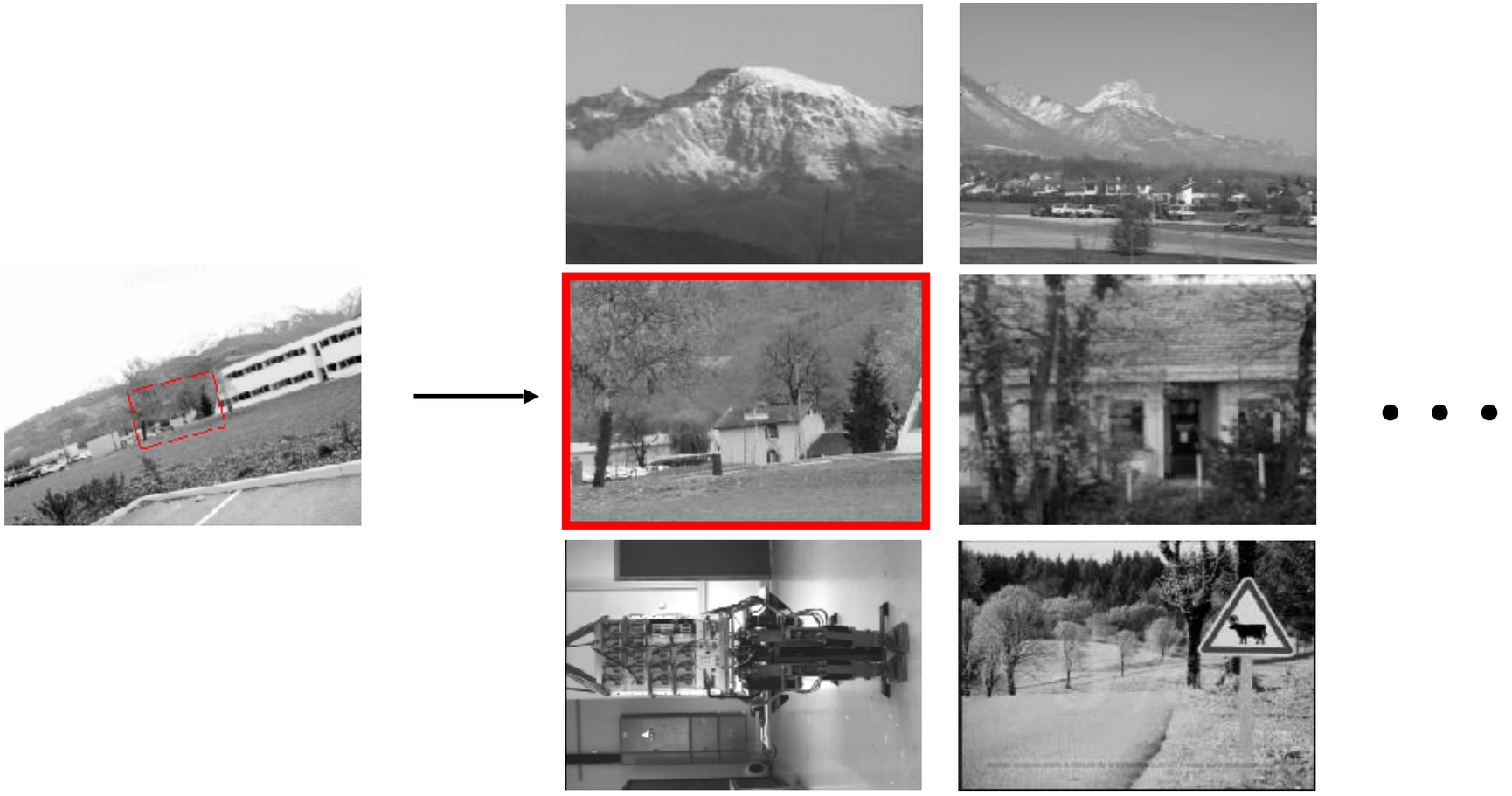
4) Feature descriptors (e.g., SIFT)

5) Matching and recognition with local features

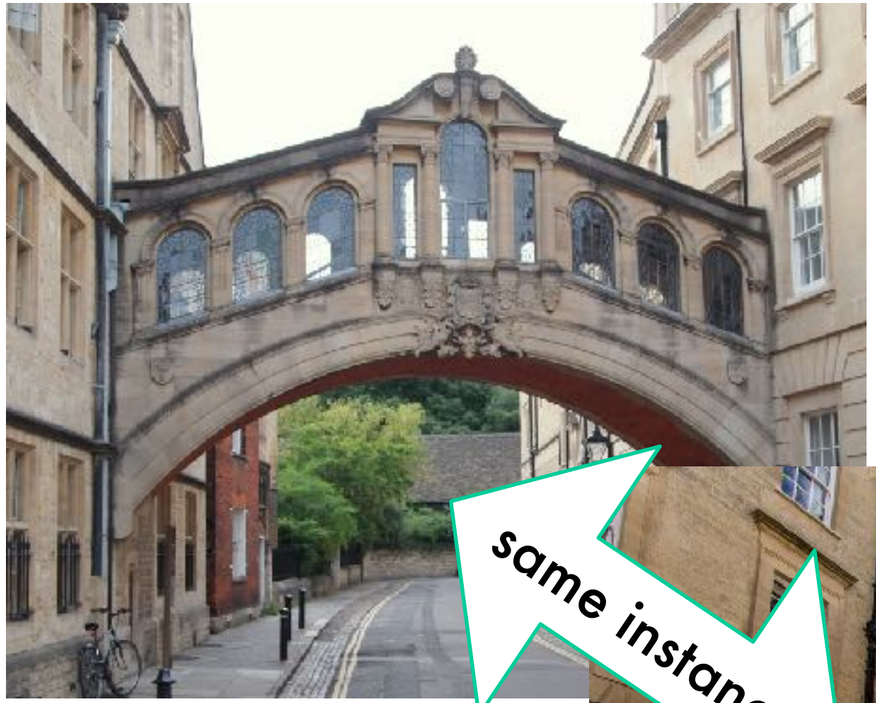
6) Local feature aggregation for a single image-level description

Instance-level recognition

Search for particular objects and scenes in large databases



Instance-level vs Category-level



Bridge of Sighs, Oxford

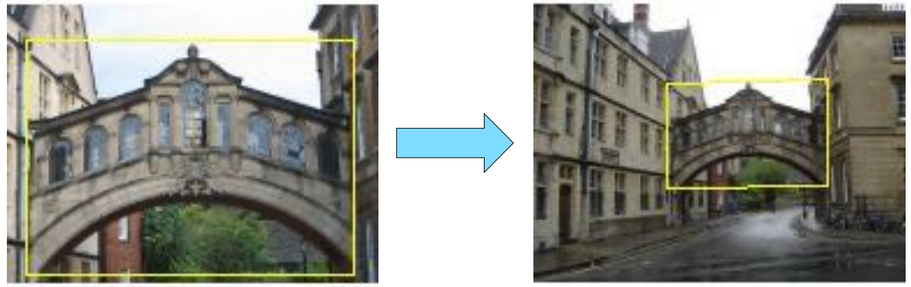
same instance

same category

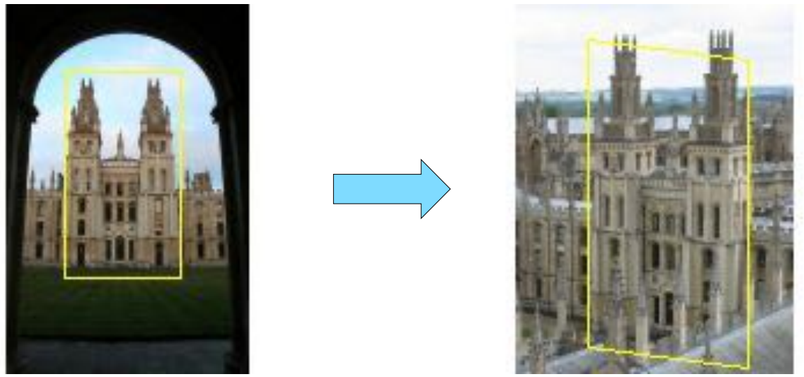
Pont Neuf, Paris

Difficulties

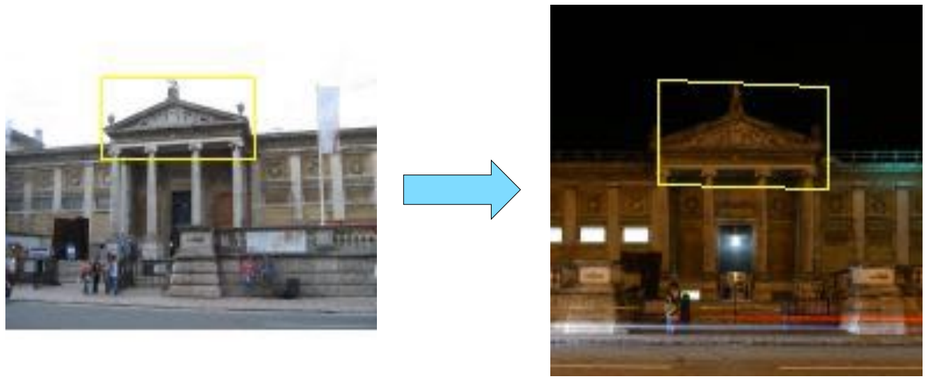
Finding the object despite possibly large changes in
scale, viewpoint, lighting and partial occlusion → **requires invariant description**



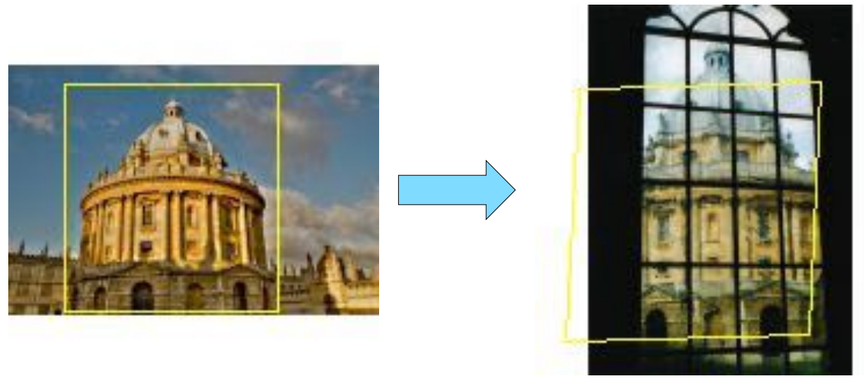
Scale



Viewpoint



Lighting



Occlusion

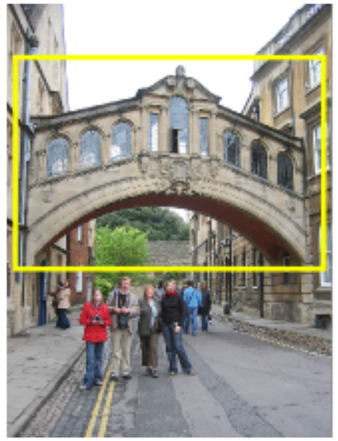
Difficulties

- Very large image collections → **need for efficient indexing**
  - ➡ Flickr has 2 billion photographs, more than 1 million added daily*
  - ➡ Facebook has 15 billion images (~27 million added daily)*
  - ➡ Large personal collections

*Potentially outdated numbers

Applications

Search photos on the web for particular places



Find these landmarks

...in these images and 1M more

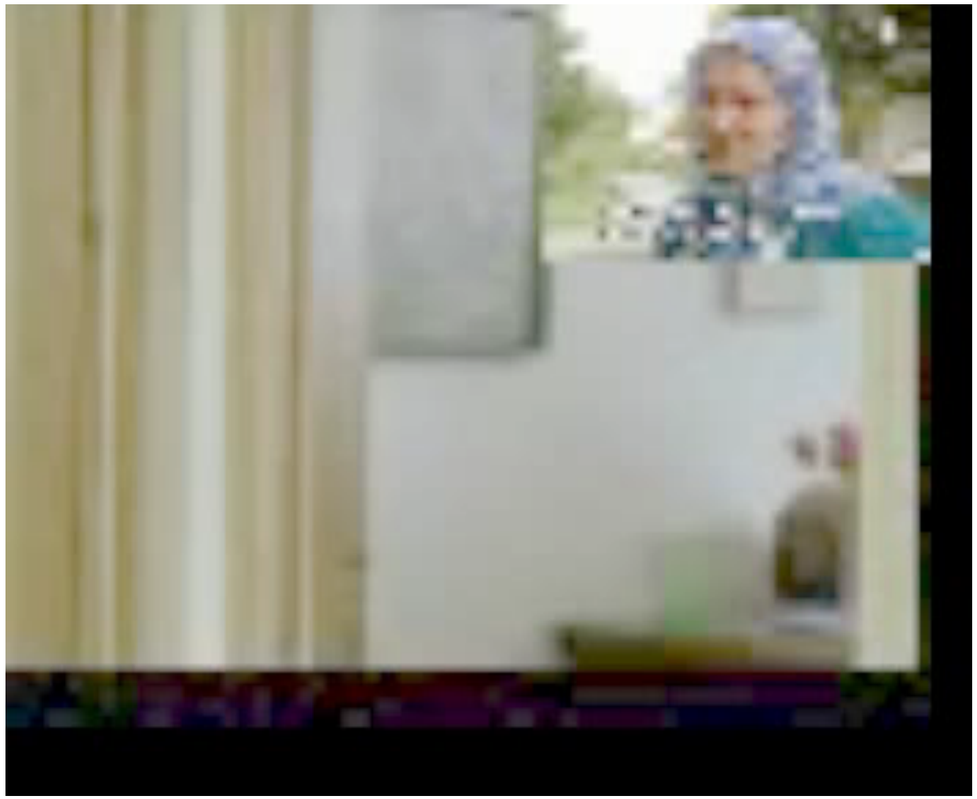
Applications

- Finding stolen/missing objects in a large collection

Applications

- Copy detection for images and videos

Query video



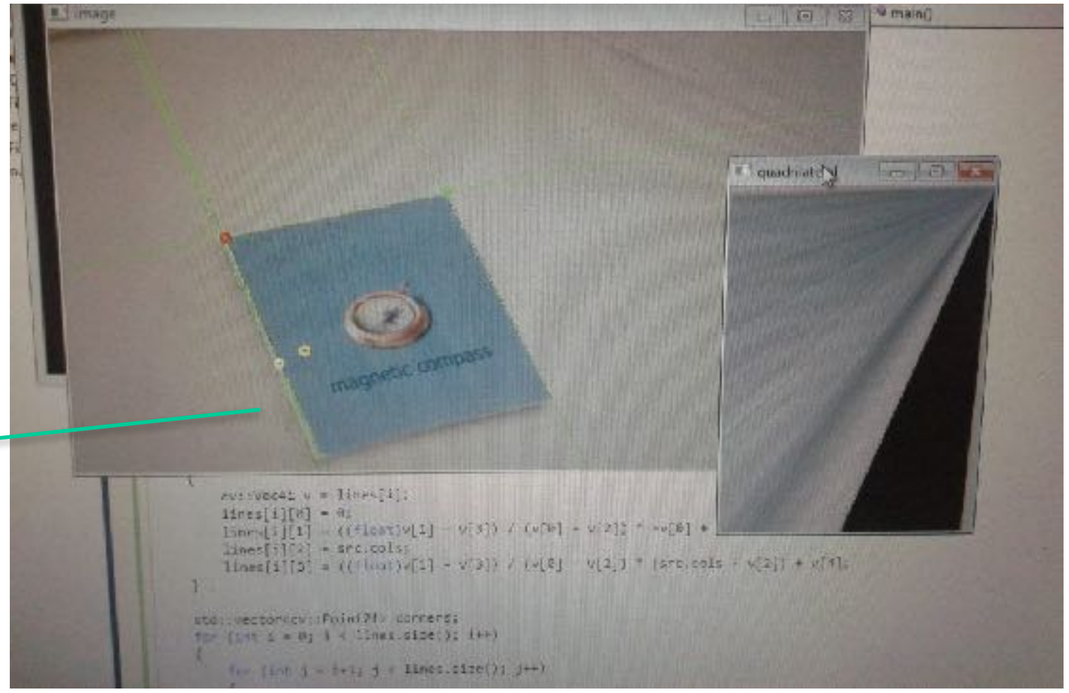
Search in 200h of video

Applications

- Sony Aibo – Robotics
  - Recognize docking station
  - Communicate with visual cards
  - Place recognition
  - Loop closure in SLAM

Applications

- Template matching



Agenda: Instance-level recognition

1) Introduction to local features

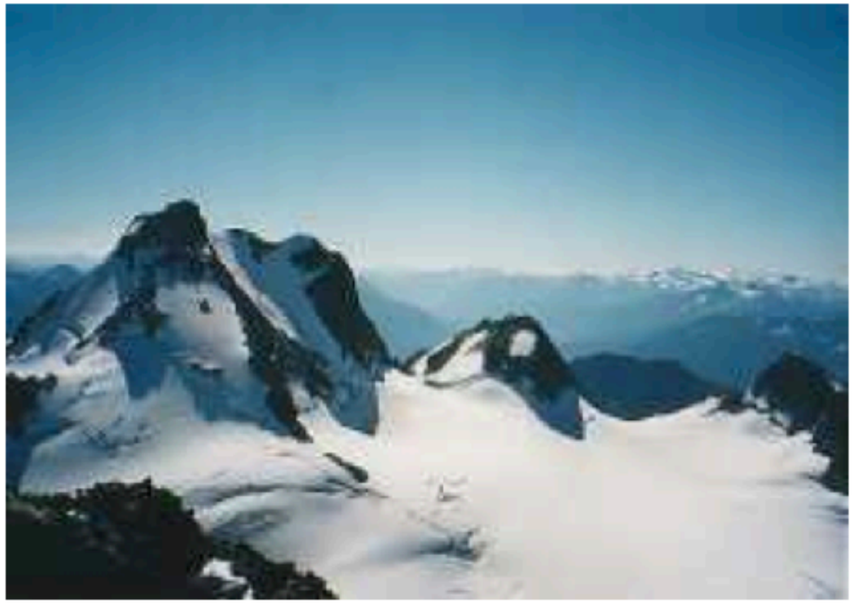
2) Interest point detectors (e.g., Harris, scale invariance)

3) Comparison of patches (SSD, ZNCC on pixel values)

4) Feature descriptors (e.g., SIFT)

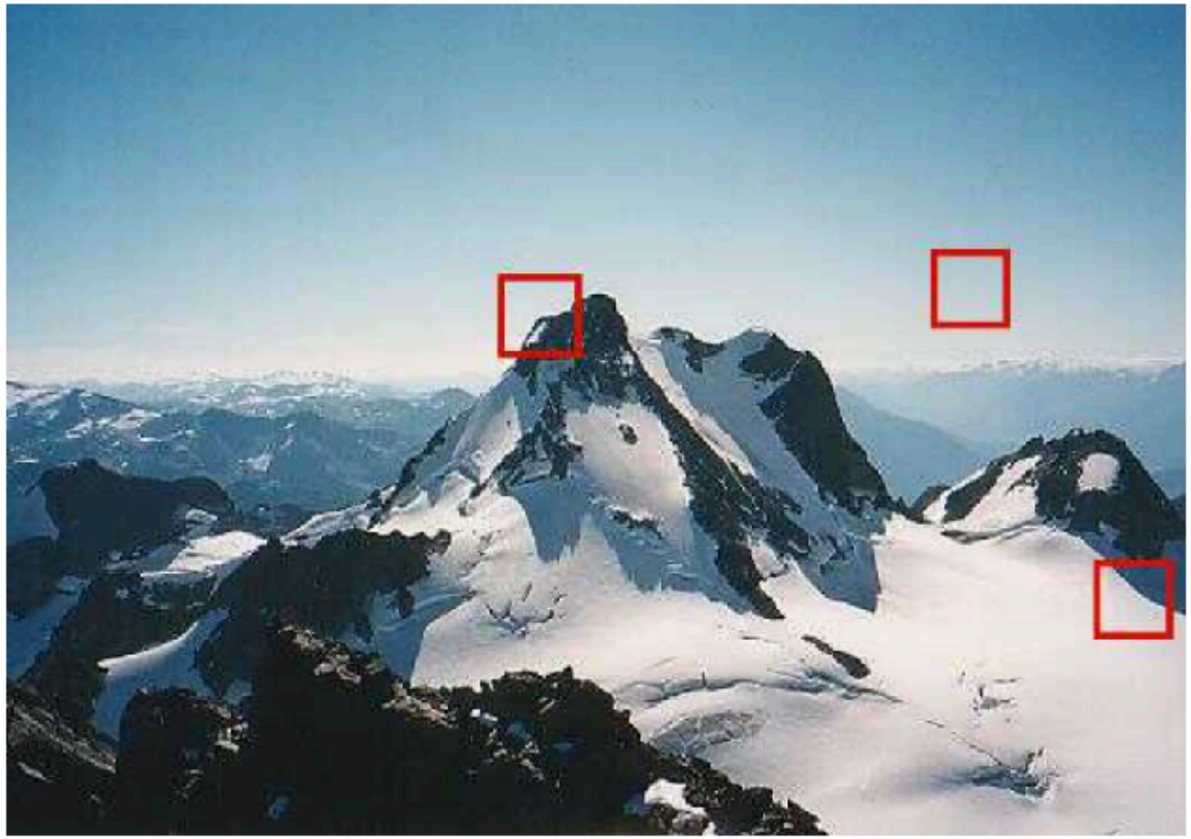
5) Matching and recognition with local features

6) Local feature aggregation for a single image-level description

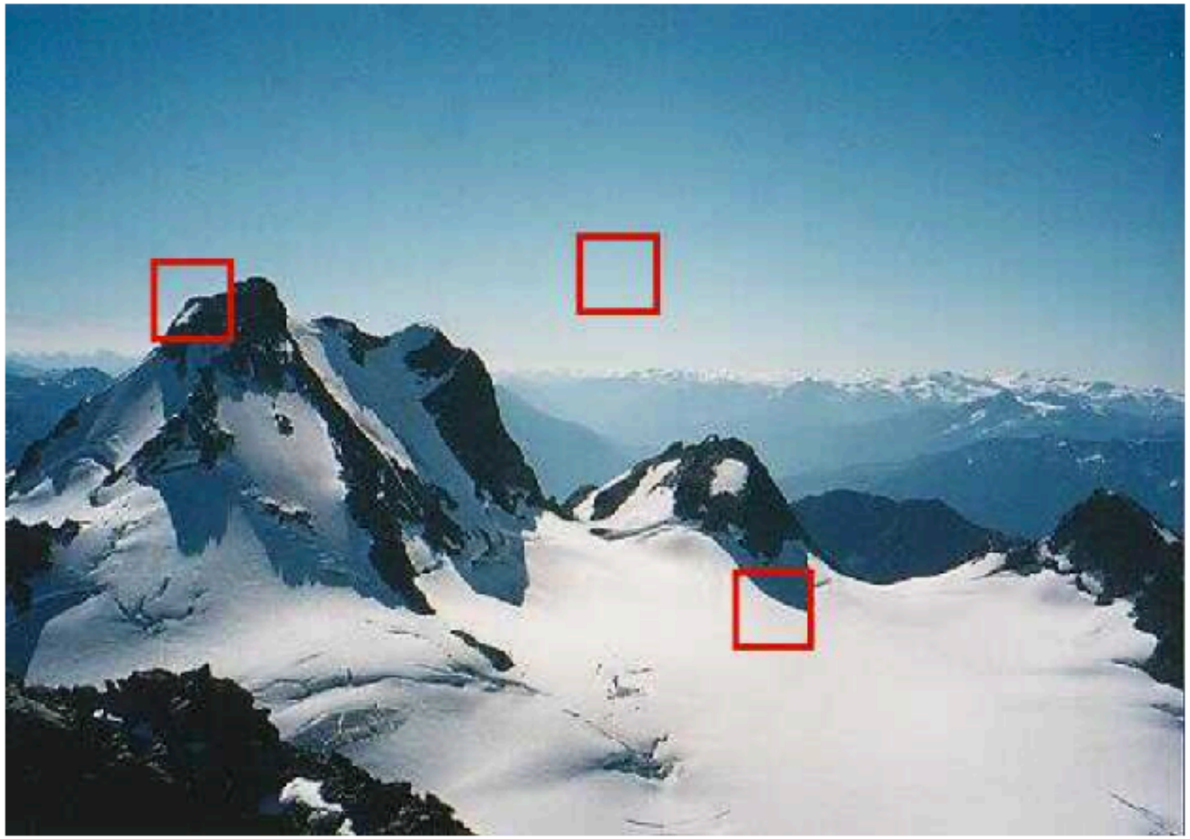



Two pairs of images to be matched. What kinds of features might one use to establish a set of correspondences between these images?


Figure 7.2 Szeliski



Textureless patches are nearly impossible to localize.



Patches with large contrast changes (gradients) are easier to localize.

Figure 7.3 Szeliski

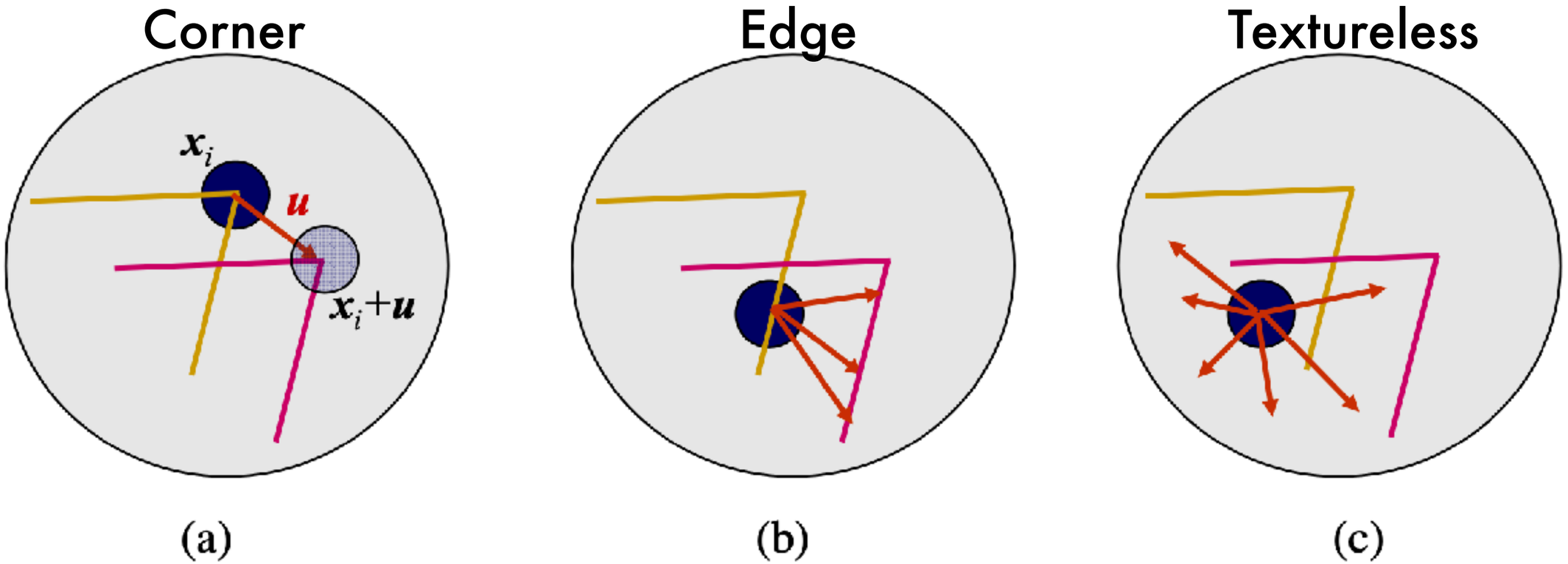
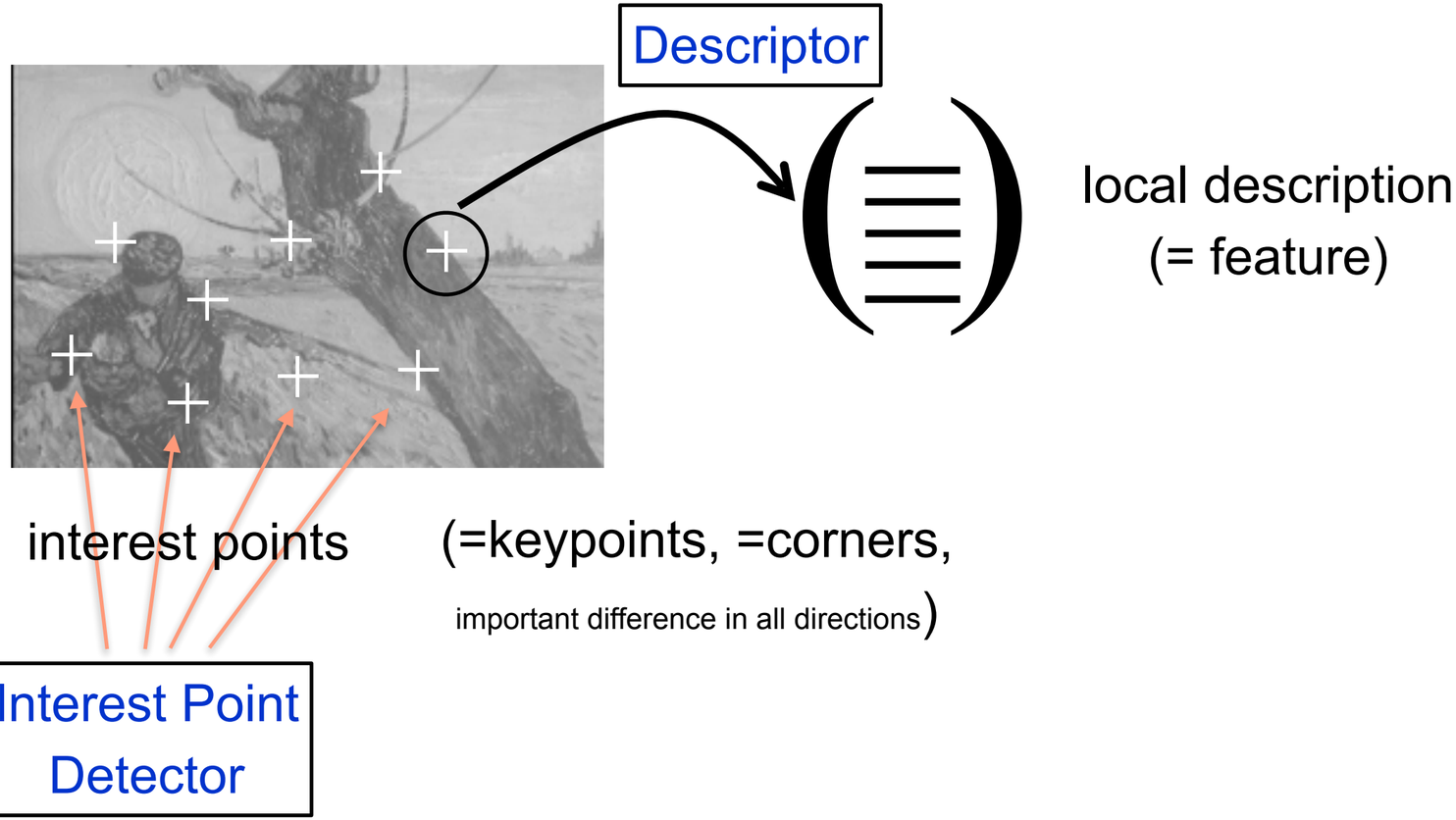


Figure 7.4 *Aperture problems for different image patches: (a) stable ("corner-like") flow; (b) classic aperture problem (barber-pole illusion); (c) textureless region. The two images I_0 (yellow) and I_1 (red) are overlaid. The red vector* $\mathbf{u}$ *indicates the displacement between the patch centers and the $w(\mathbf{x}_i)$ weighting function (patch window) is shown as a dark circle.*

Figure 7.4 Szeliski

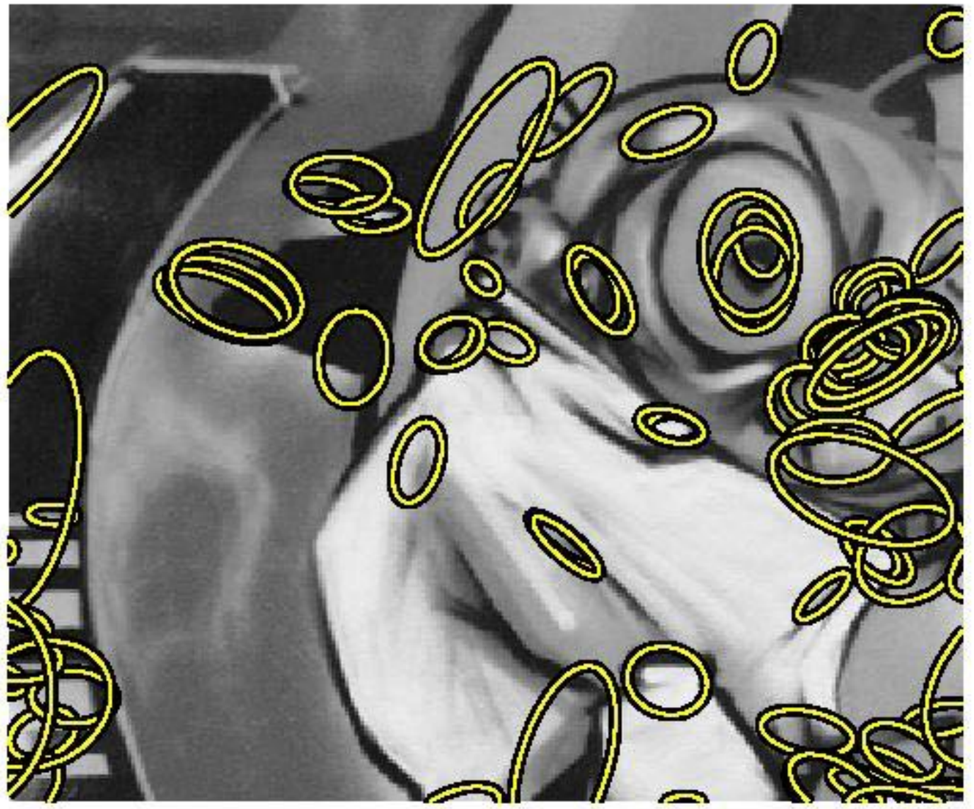
Local features



A **corner** is a point whose **local neighborhood** stands in two dominant and different edge directions. In other words, a corner can be interpreted as the junction of two edges, where an edge is a **sudden change in image brightness**. Corners are the important features in the image, and they are generally termed as **interest points** which are **invariant** to **translation**, **rotation** and **illumination**. Although corners are only a small percentage of the image, they contain the **most important features** in restoring image information... [Harris corner detection, Wikipedia]

Interest points / invariant regions

Harris detector



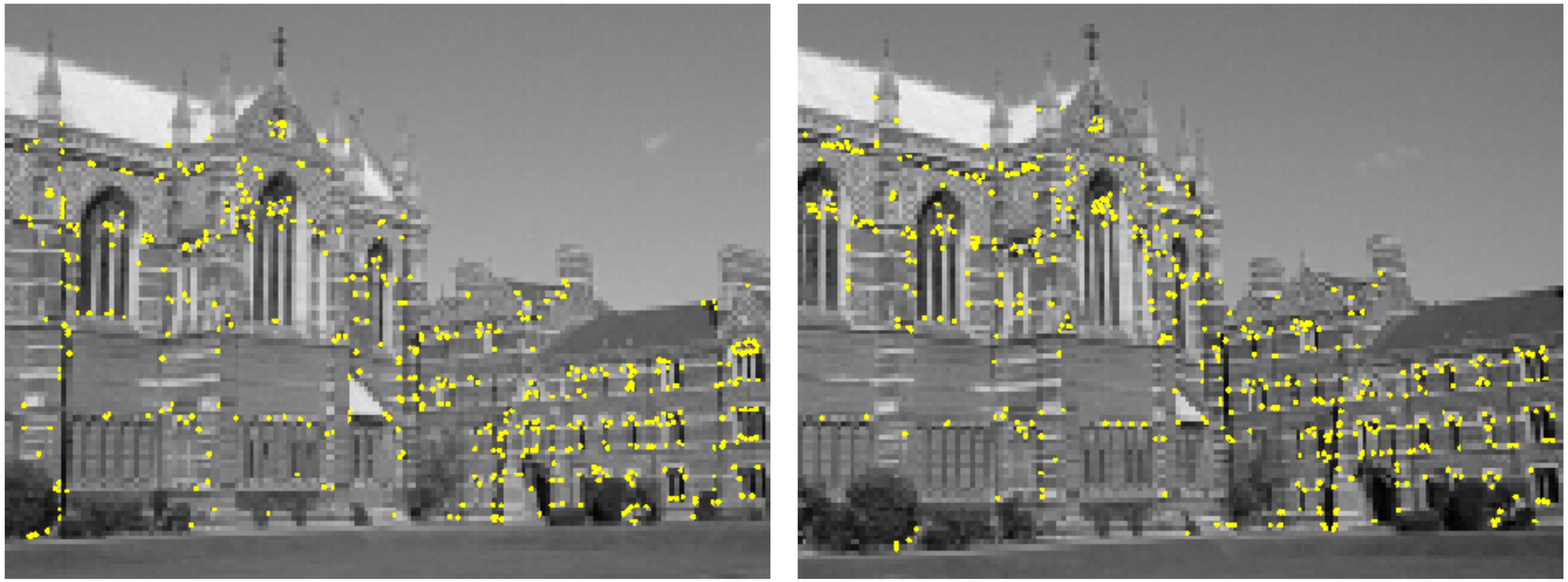
Scale invariant detector

Matching of local descriptors

What can go wrong in matching this image pair?

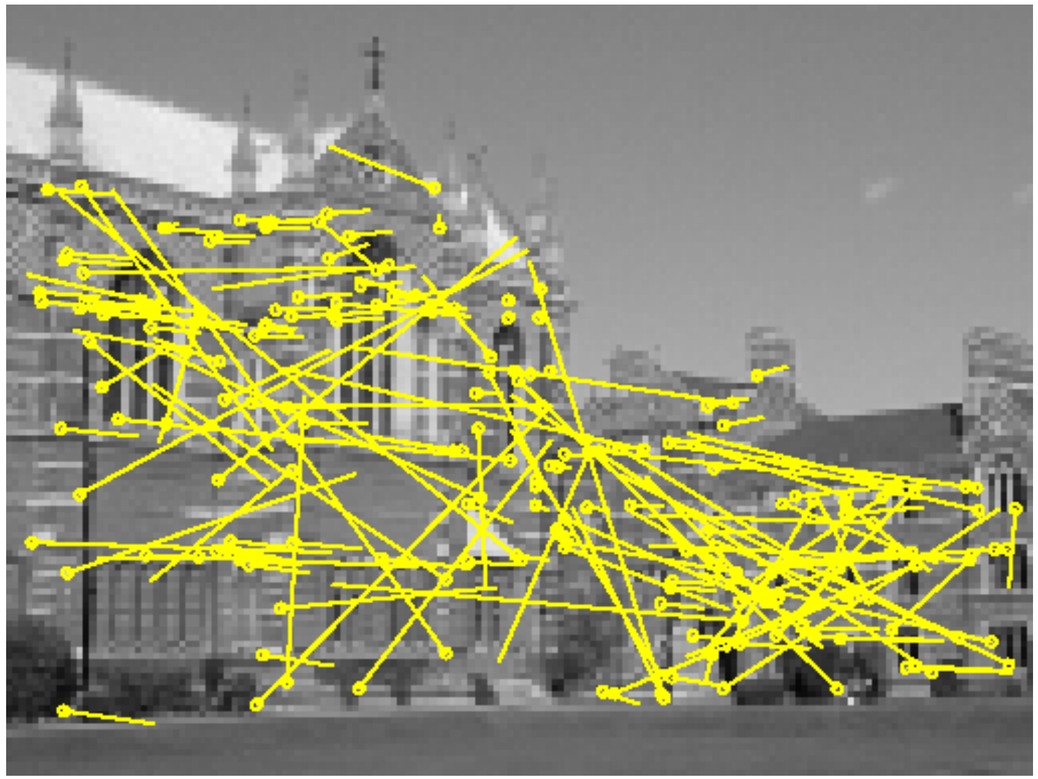
Find corresponding locations in the image

Illustration – Matching



Interest points extracted with Harris detector (~ 500 points)

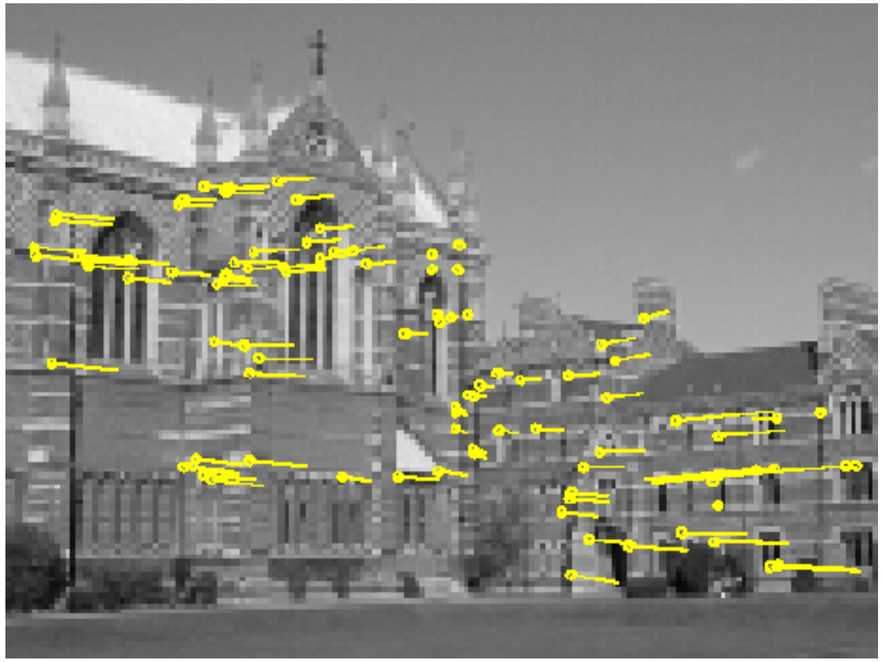
Illustration – Matching



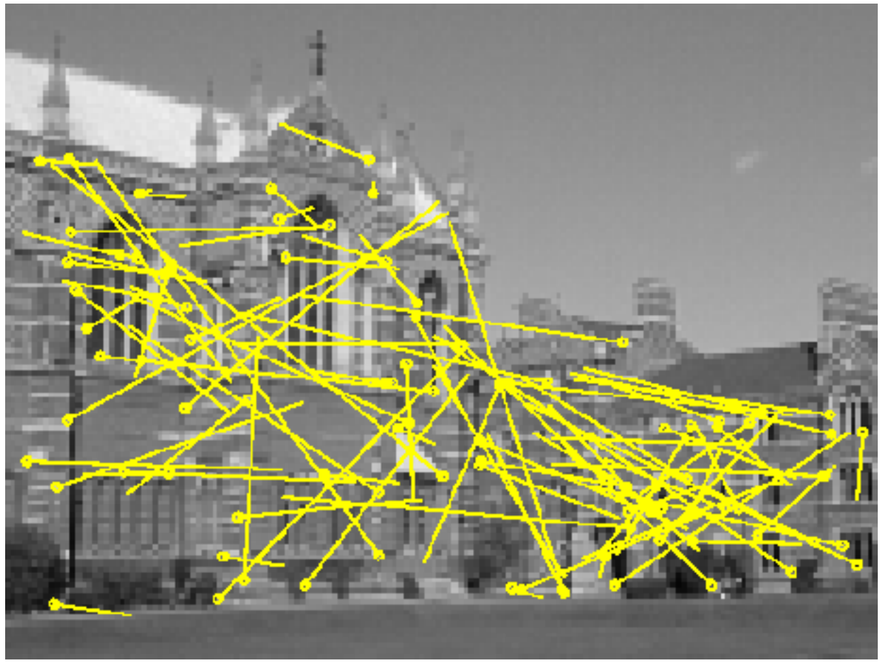
Interest points matched based on cross-correlation (188 pairs)

Illustration – Matching

Global constraint - Robust estimation of the fundamental matrix



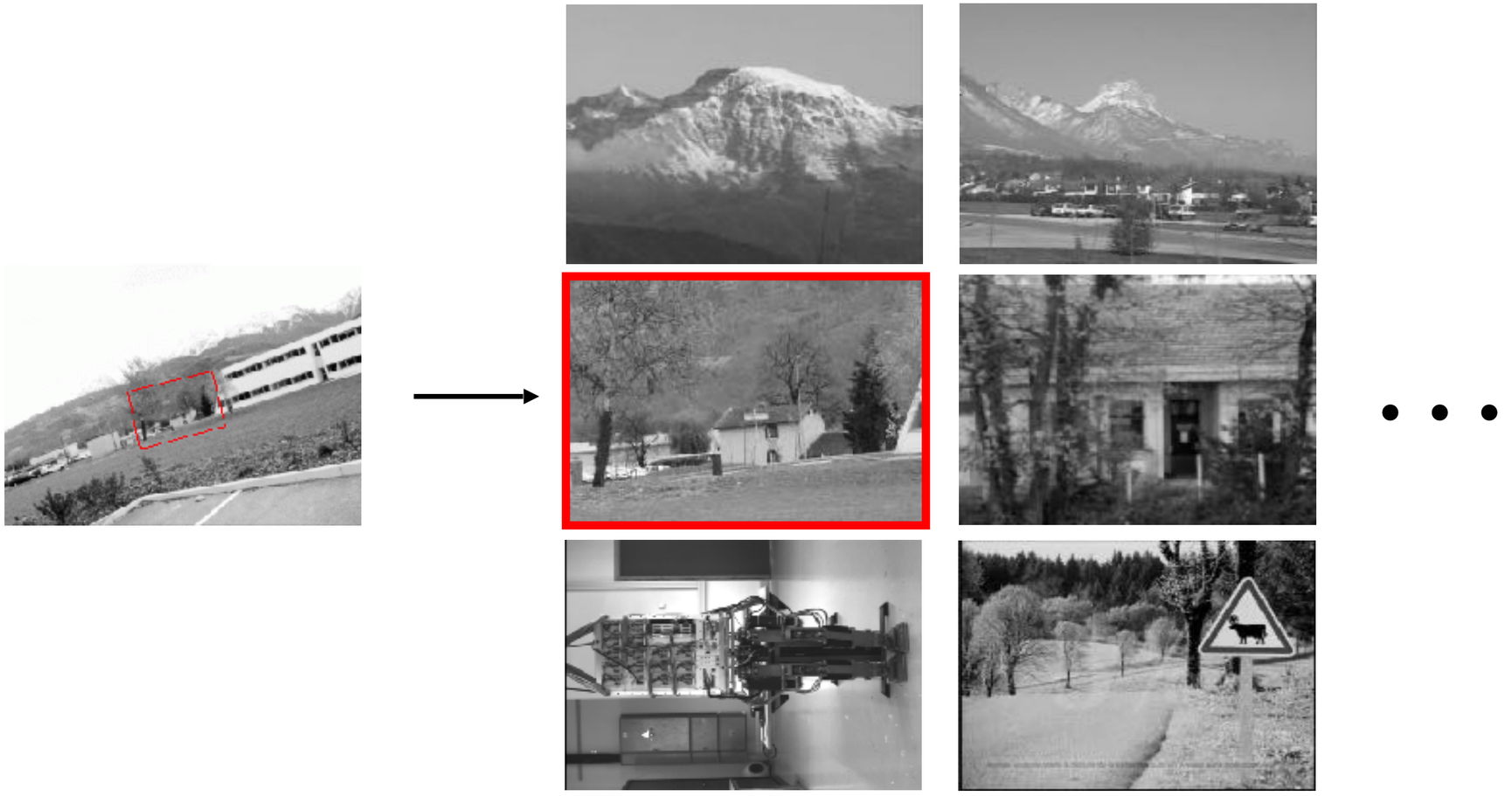
99 inliers



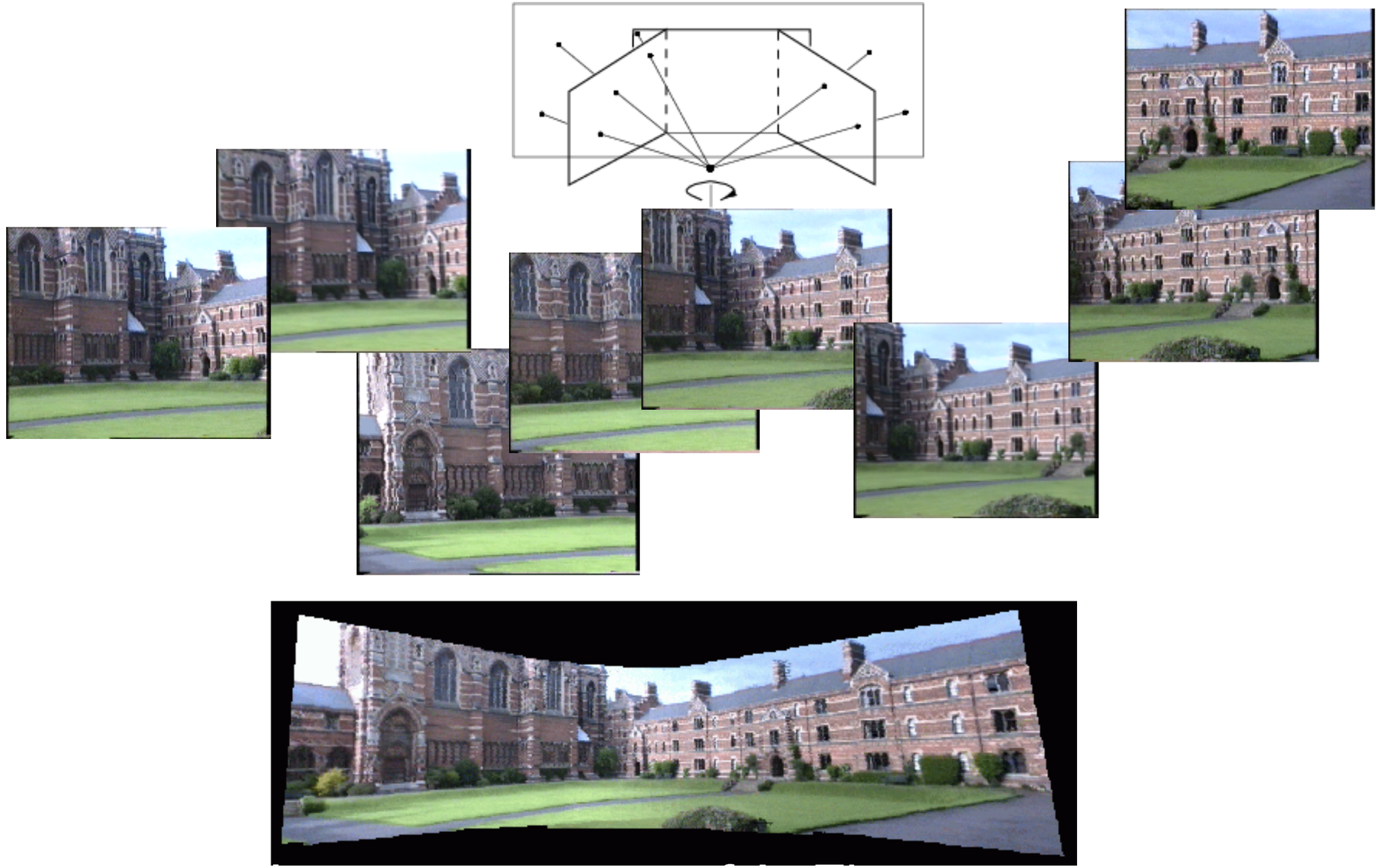
89 outliers

Application: Instance-level recognition

Search for particular objects and scenes in large databases



Application: Panorama stitching



Agenda: Instance-level recognition

1) Introduction to local features

2) Interest point detectors (e.g., Harris, scale invariance)

3) Comparison of patches (SSD, ZNCC on pixel values)

4) Feature descriptors (e.g., SIFT)

5) Matching and recognition with local features

6) Local feature aggregation for a single image-level description

Harris detector [Harris & Stephens'88]

Based on the idea of auto-correlation

Important difference in all directions => interest point

Harris detector

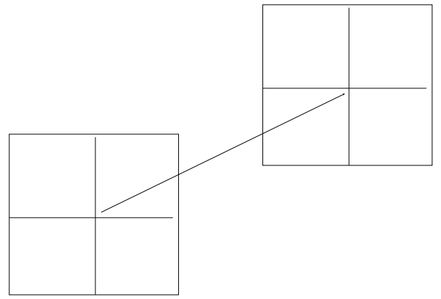
Auto-correlation function for a point $\mathbf{x} = (x, y)$ and a shift $\Delta\mathbf{u} = (\Delta x, \Delta y)$

(displacement vector)

$\Delta\mathbf{u} = (\Delta x, \Delta y)$

$$E_{AC}(\Delta\mathbf{u}) = \sum_{i \in W} w(\mathbf{x}_i)(I(\mathbf{x}_i + \Delta\mathbf{u}) - I(\mathbf{x}_i))^2$$

(spatially varying weighting function)



W

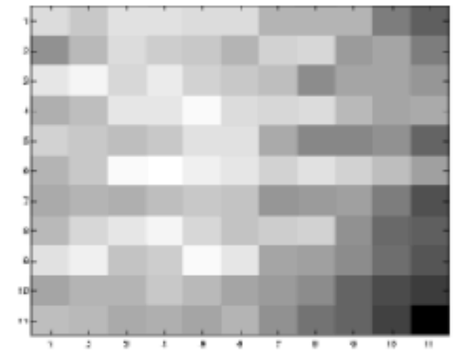
(window)

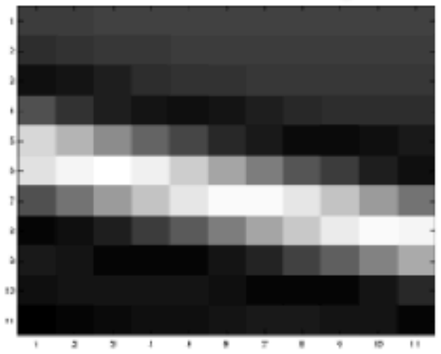
$E_{AC}(\Delta\mathbf{u})$ {

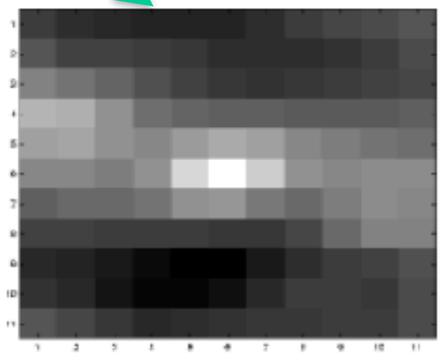
- small in all directions → uniform region
- large in one direction → contour
- large in all directions → interest point

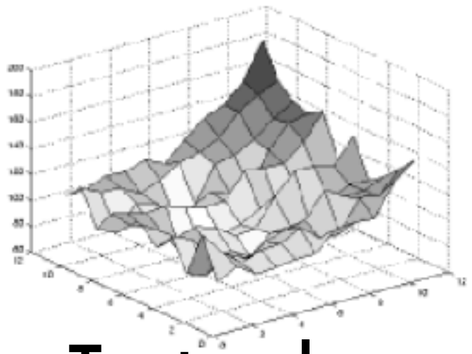
"Strictly speaking, a correlation is the product of two patches [...] using the term here in a more qualitative sense. The weighted **sum of squared differences** is often called an SSD surface."

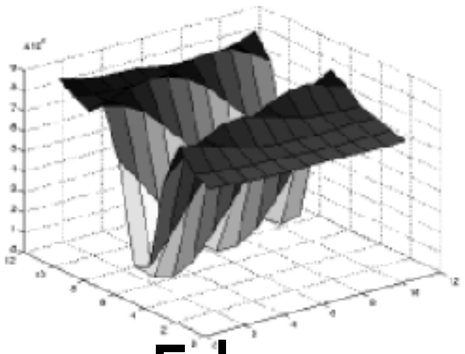
auto-correlation surfaces

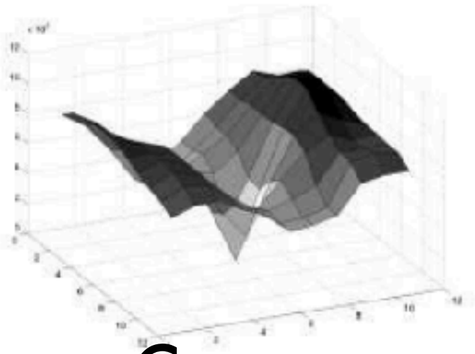












Textureless

Edge

Corner

Figure 7.5 Szeliski

Harris detector

Taylor Series expansion:

$$E_{AC}(\Delta\mathbf{u}) = \sum_{i \in W} w(\mathbf{x}_i)(I(\mathbf{x}_i + \Delta\mathbf{u}) - I(\mathbf{x}_i))^2$$

$$\approx \sum_{i \in W} w(\mathbf{x}_i)(I(\mathbf{x}_i) + \nabla I(\mathbf{x}_i) \,.\, \Delta\mathbf{u} - I(\mathbf{x}_i))^2$$

$$= \sum_{i \in W} w(\mathbf{x}_i)(\nabla I(\mathbf{x}_i) \,.\, \Delta\mathbf{u})^2$$

$$= \Delta\mathbf{u}^T \mathbf{A} \Delta\mathbf{u}$$

replaced the weighted summations with discrete convolutions with the weighting kernel w

e.g., Harris detector uses a [-2 -1 0 1 2] filter.

Other variants convolving with horizontal/vertical derivatives of a Gaussian.

(image gradient)

$$\nabla I(\mathbf{x}_i) = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})(\mathbf{x_i})$$

(auto-correlation matrix)

$$\mathbf{A} = w * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

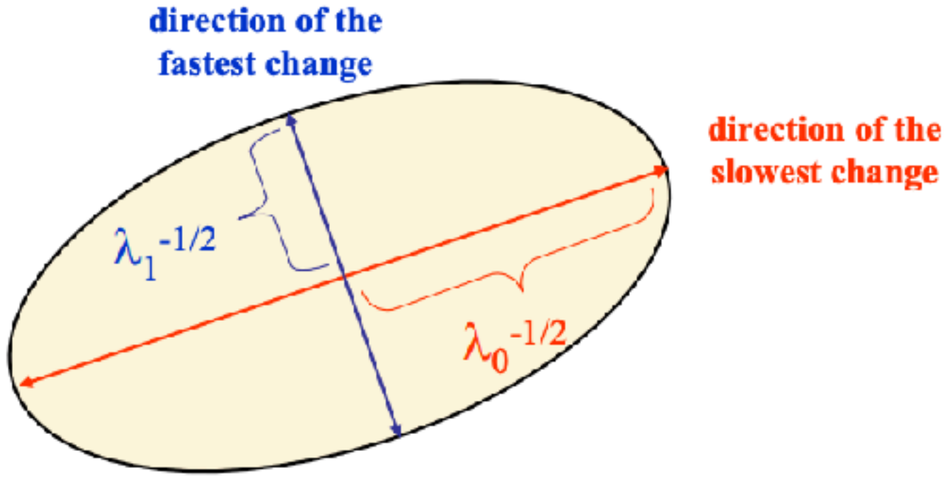
I_x *(partial derivative in horizontal axis)*

Harris detector

- The sum can be smoothed with a Gaussian
- Gaussian window instead of square window

$$\mathbf{A}(\mathbf{x},\mathbf{y}) = G \otimes \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

- captures the structure of the local neighborhood
- measure based on eigenvalues of this matrix
  - 2 strong eigenvalues => interest point
  - 1 strong eigenvalue => contour
  - 0 eigenvalue => uniform region

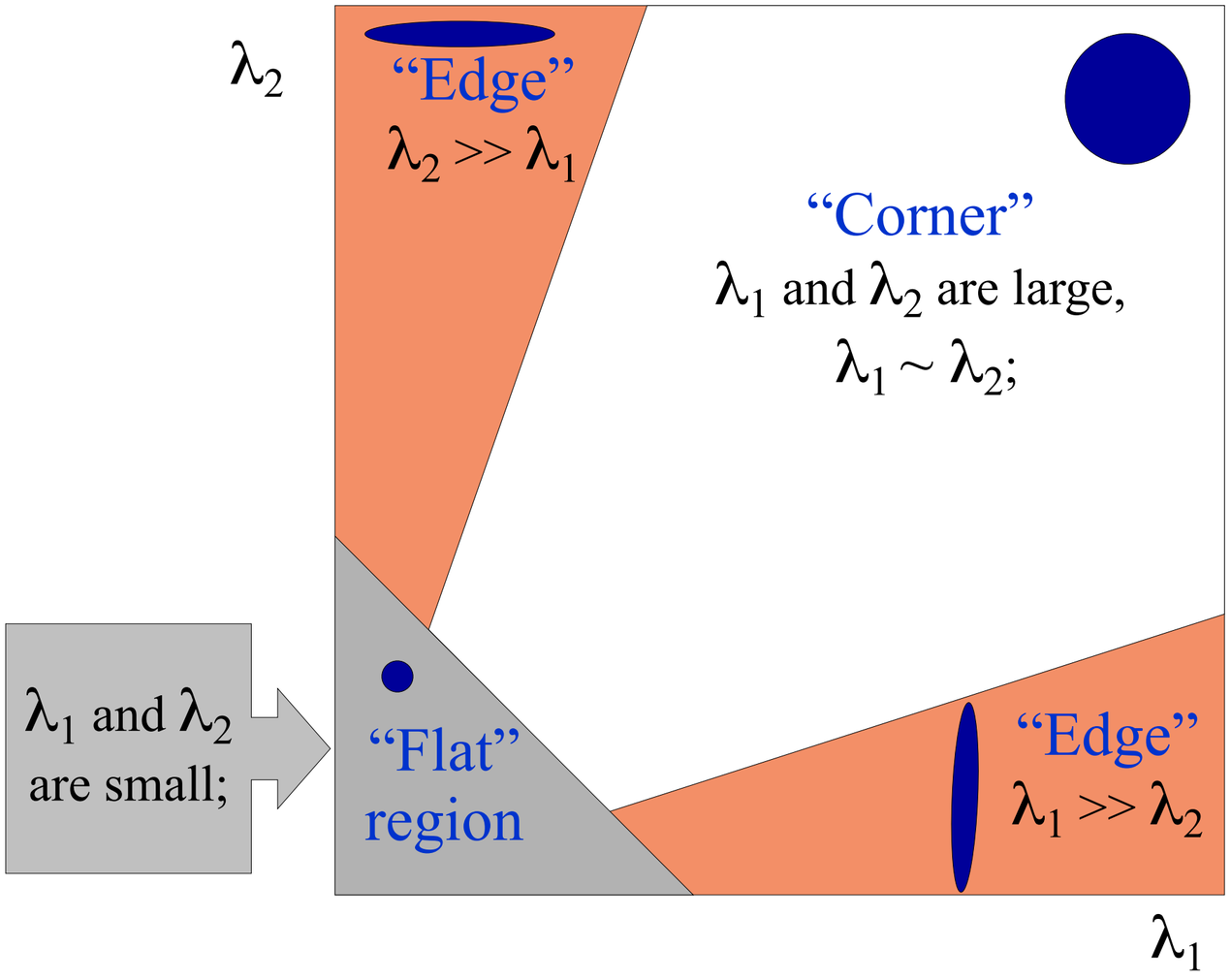


Uncertainty ellipse corresponding to an eigenvalue analysis of the autocorrelation matrix A.

Figure 7.6 Szeliski

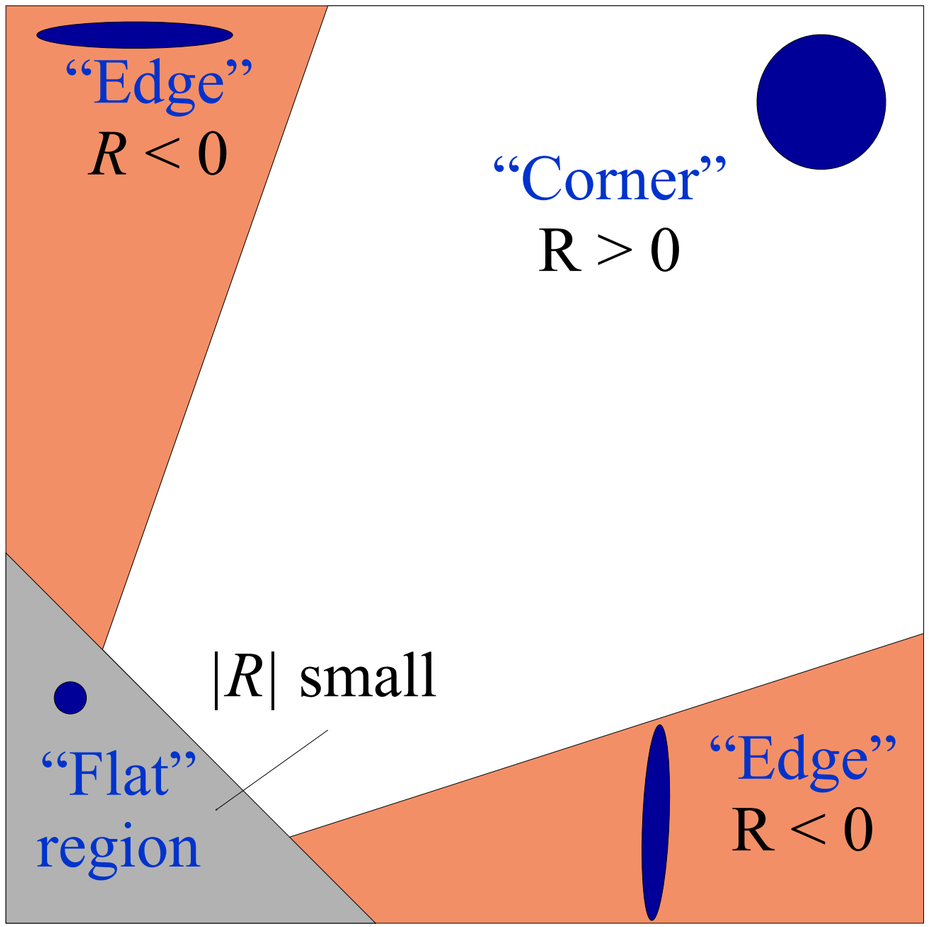
Interpreting the eigenvalues

Classification of image points using eigenvalues of autocorrelation matrix



Corner response function

A simpler quantity, proposed by Harris and Stephens (1988)

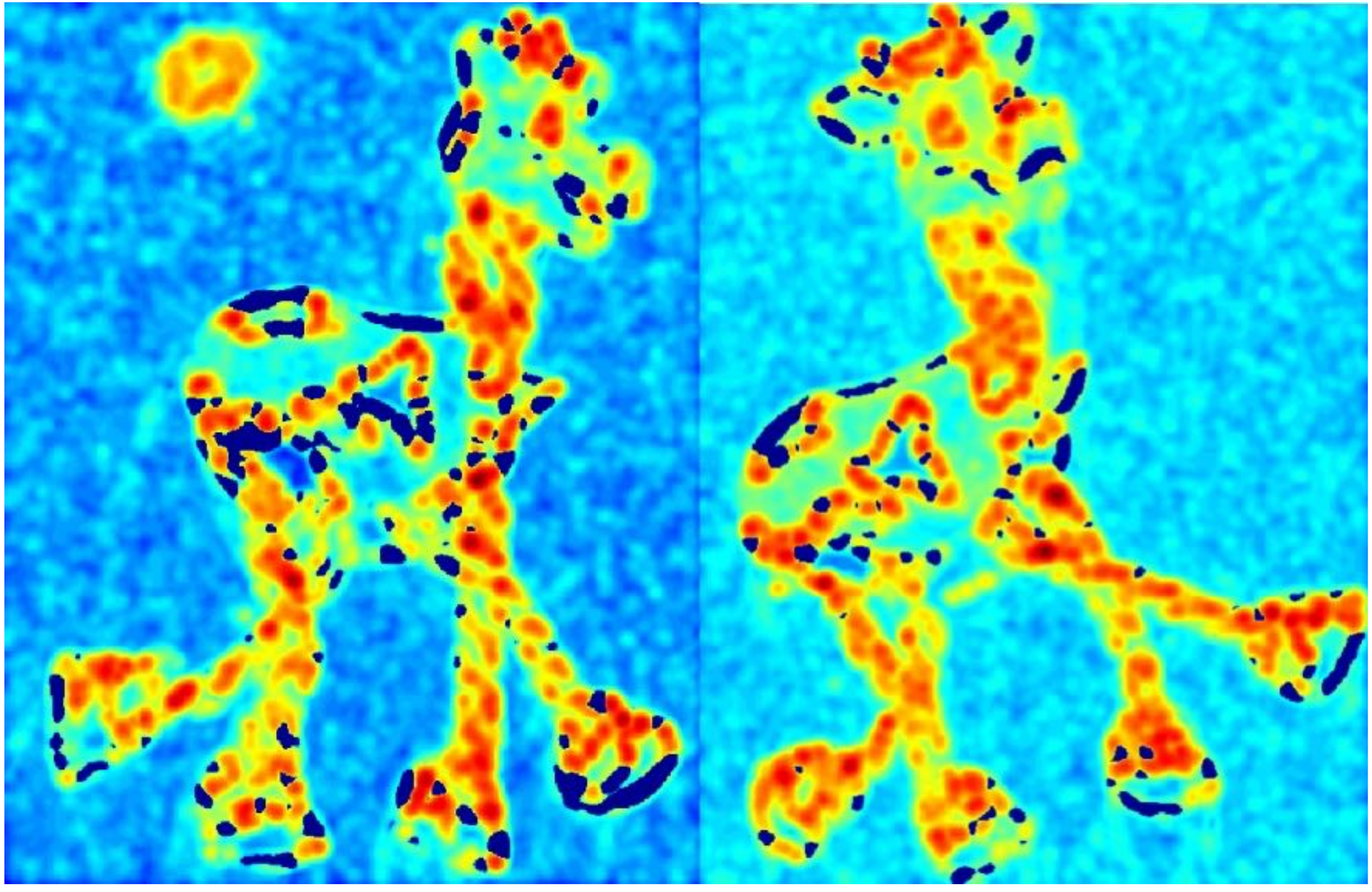


$$R = \det(\mathbf{A}) - \alpha \operatorname{trace}(\mathbf{A})^2$$
$$= \lambda_1\lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

Harris Detector: Steps

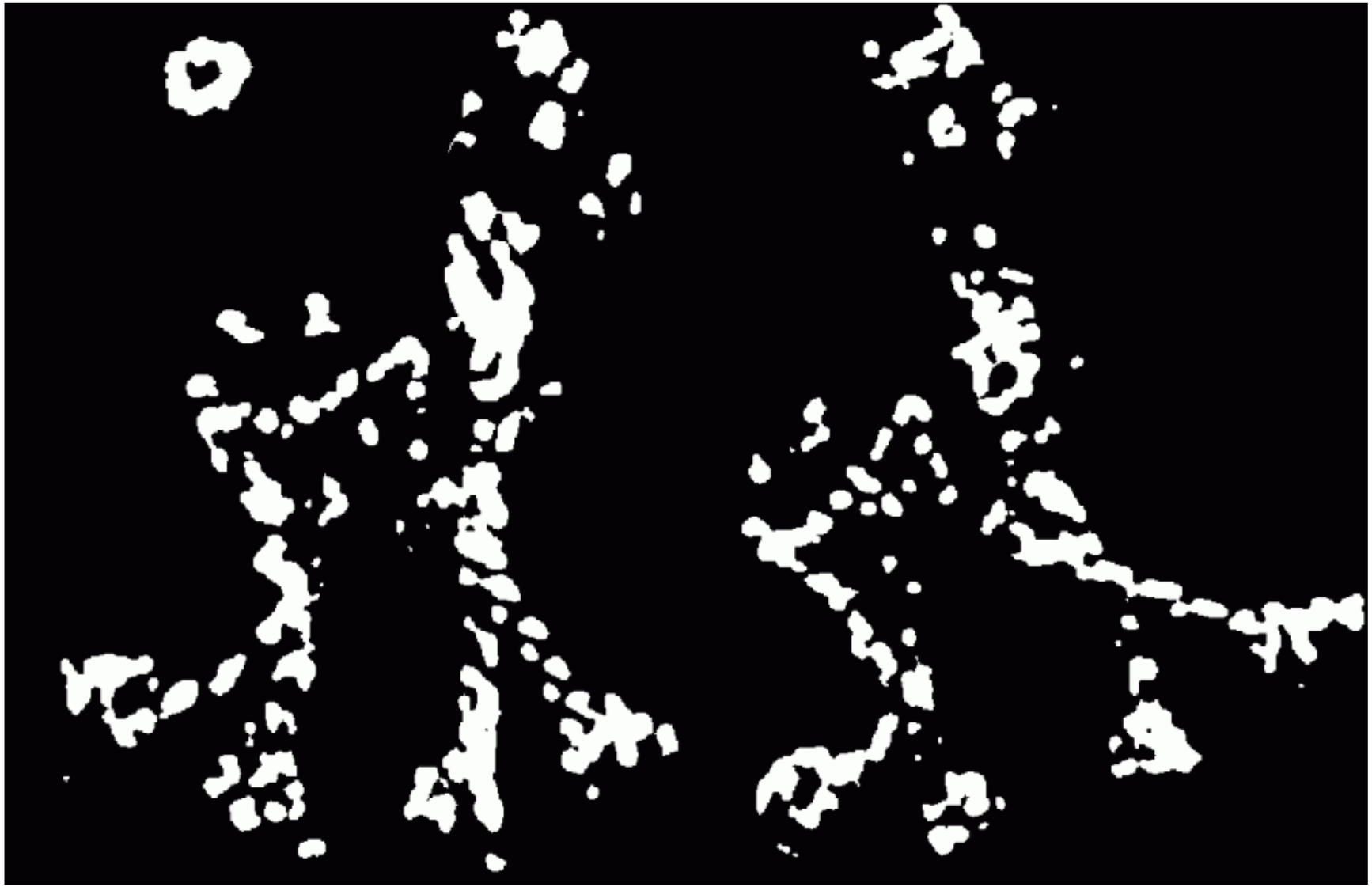
Harris Detector: Steps

Compute corner response R



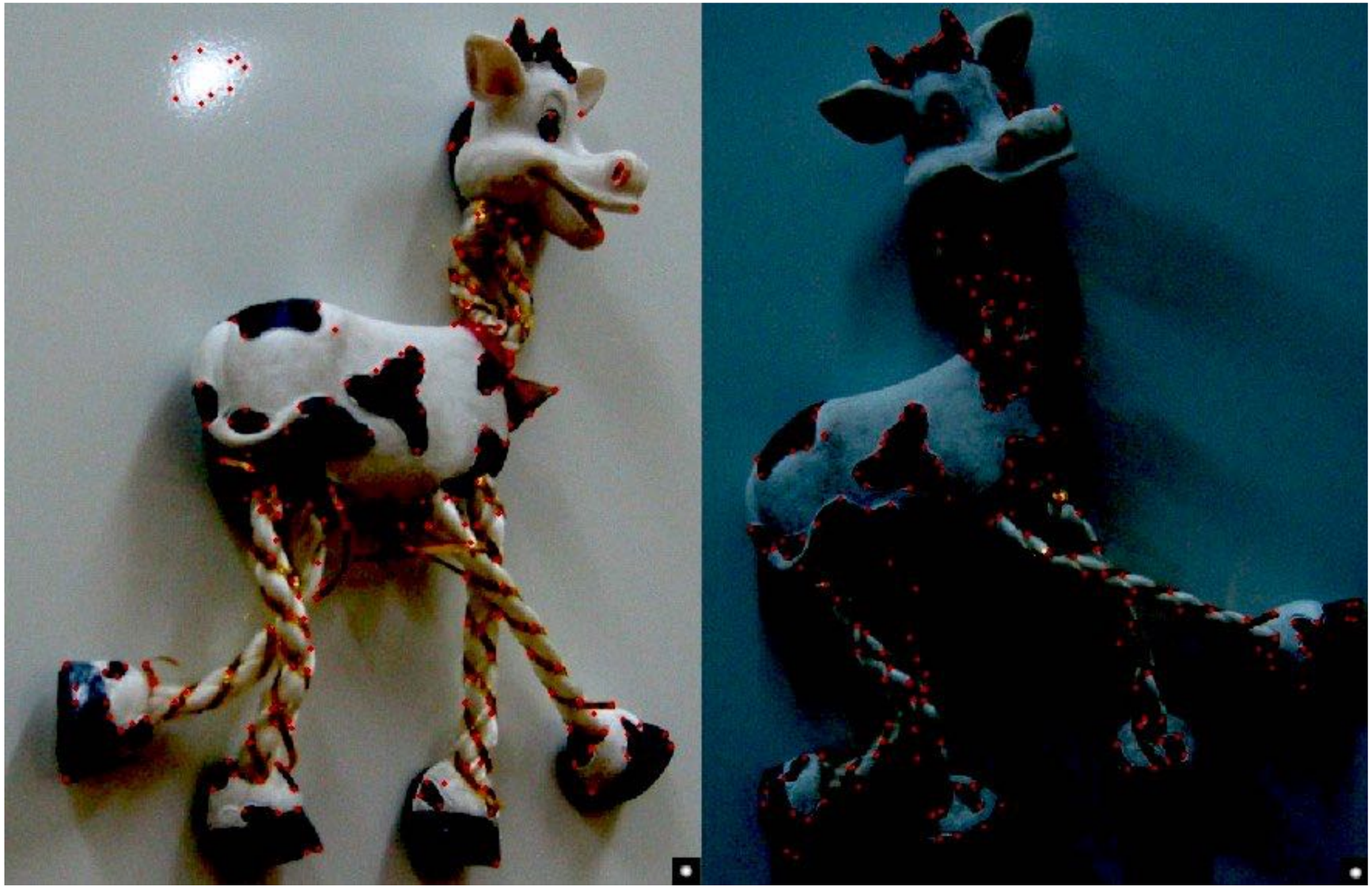
Harris Detector: Steps

Find points with large corner response: R>threshold



Harris Detector: Steps

Take only the points of local maxima of *R (non-maximum suppression)*

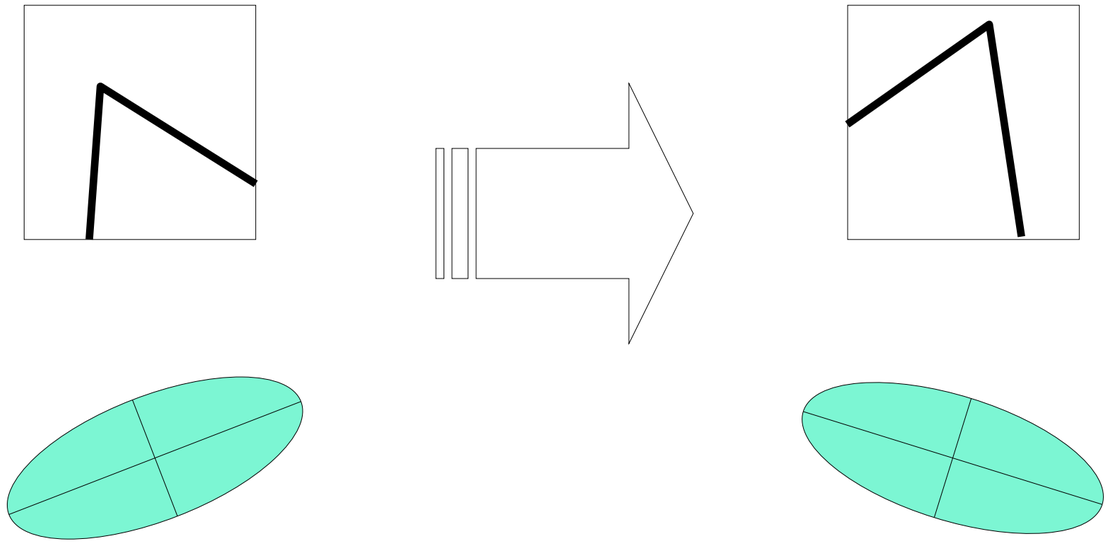


Harris detector: Summary of steps

1. Compute Gaussian derivatives at each pixel
2. Compute second moment matrix **A** in a Gaussian window around each pixel
3. Compute corner response function R
4. Threshold R
5. Find local maxima of response function (non-maximum suppression)

Harris Detector: Invariance Properties

- Rotation



Ellipse rotates but its shape (i.e. eigenvalues) remains the same

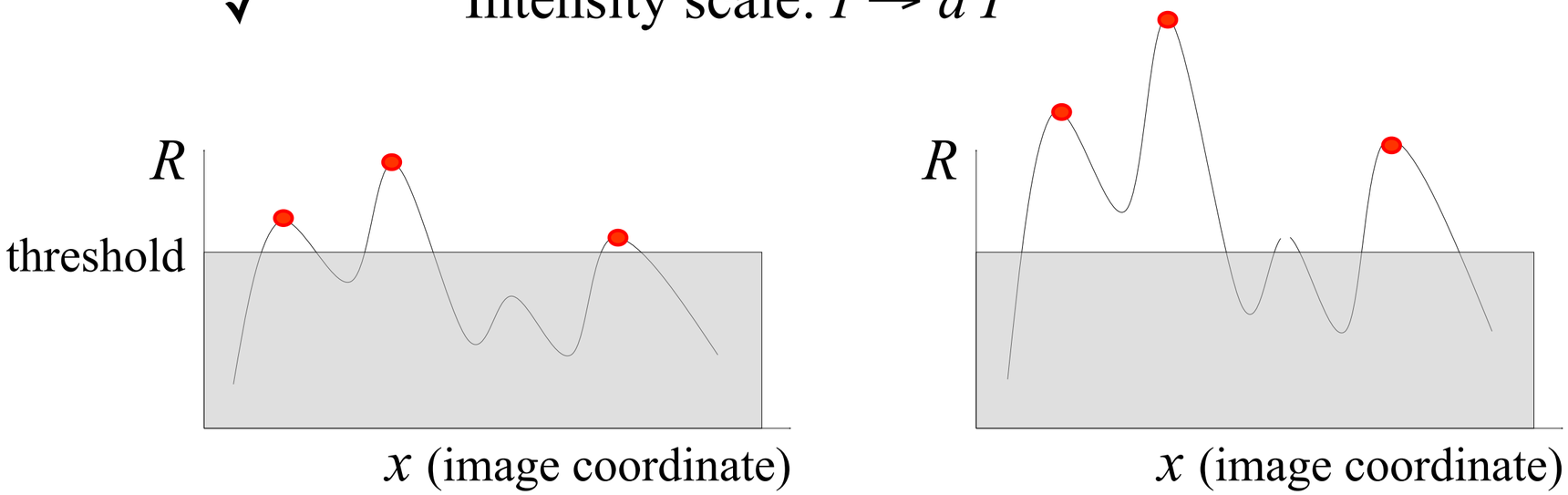
Corner response R is invariant to image rotation

Harris Detector: Invariance Properties

- Affine intensity change

✓ Only derivatives are used => invariance to intensity shift $I \rightarrow I + b$

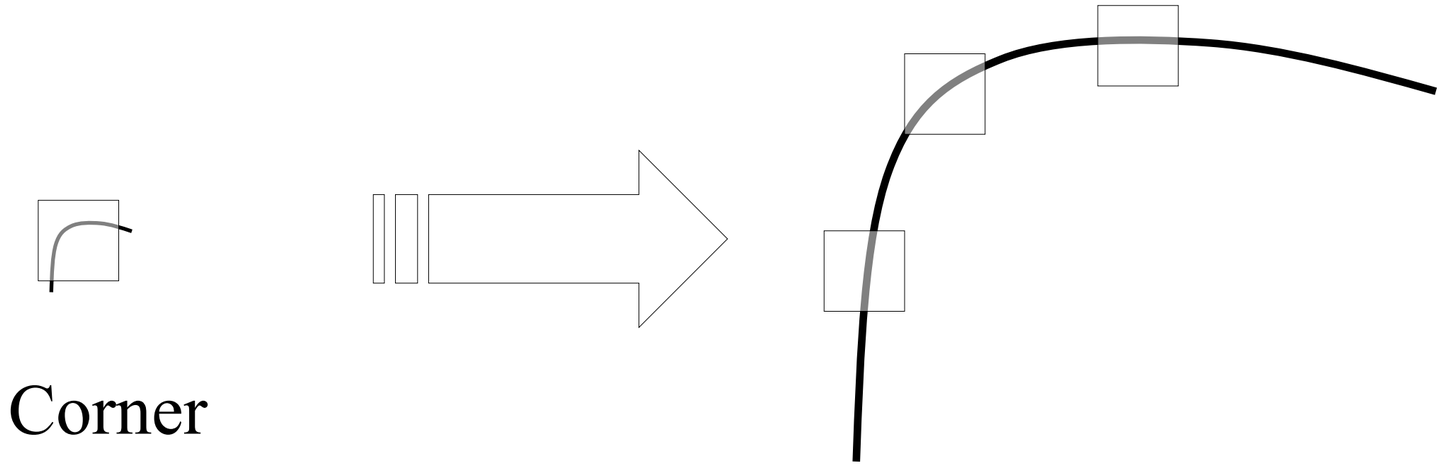
✓ Intensity scale: $I \rightarrow a\, I$



Partially invariant to affine intensity change, dependent on type of threshold

Harris Detector: Invariance Properties

- Scaling



Corner

All points will be classified as edges

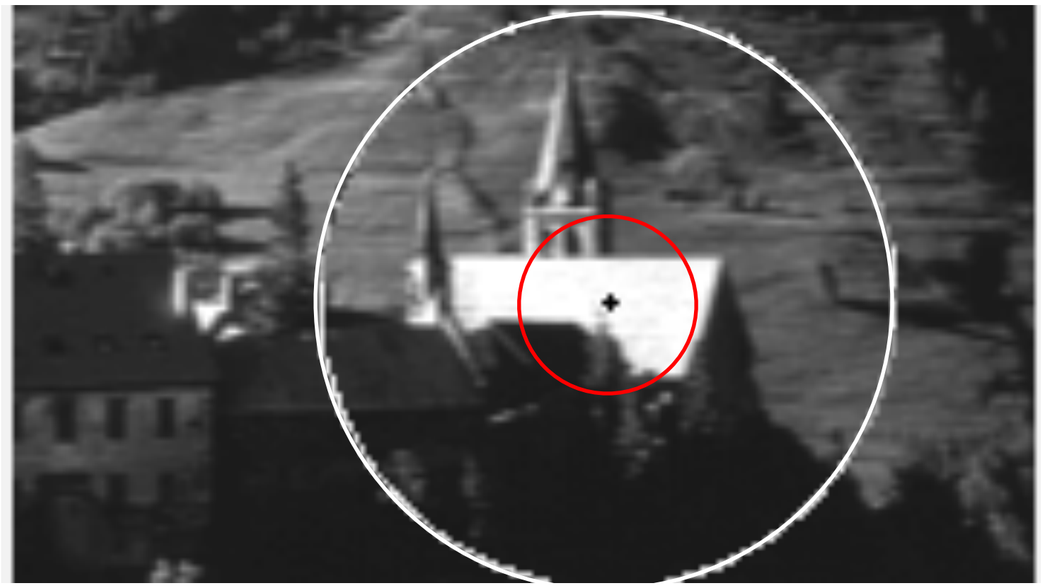
Not invariant to scaling

Agenda: Instance-level recognition

1) Introduction to local features
2) Interest point detectors (e.g., Harris, scale invariance)
3) Comparison of patches (SSD, ZNCC on pixel values)
4) Feature descriptors (e.g., SIFT)
5) Matching and recognition with local features
6) Local feature aggregation for a single image-level description

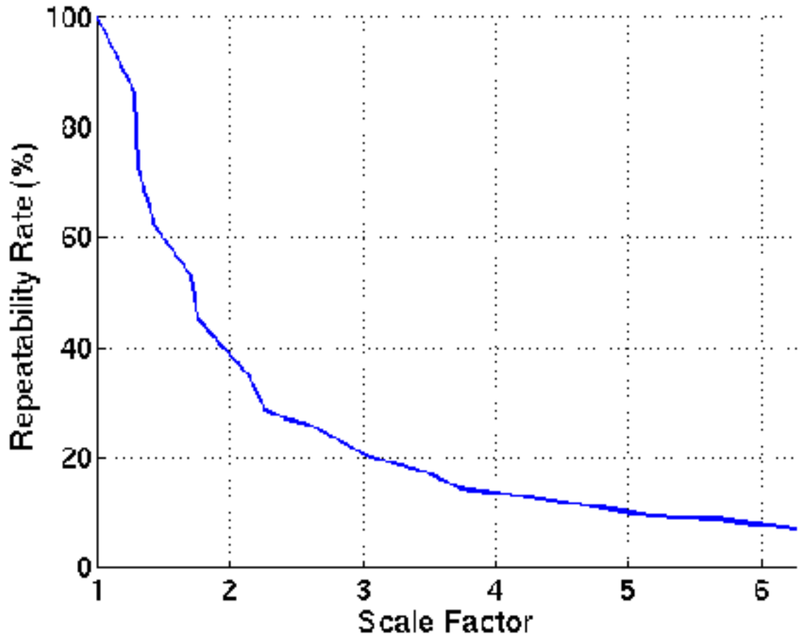
Scale invariance - motivation

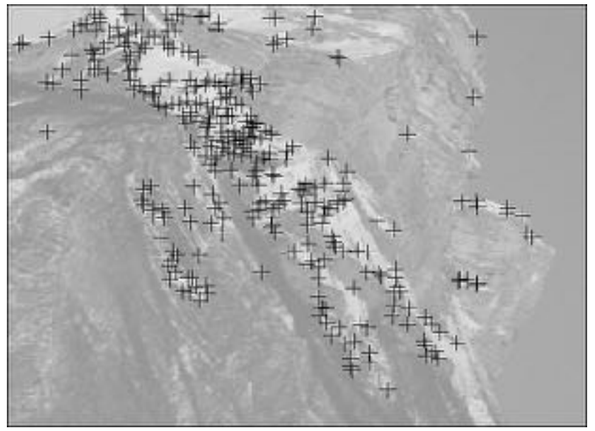
- Description regions have to be adapted to scale changes

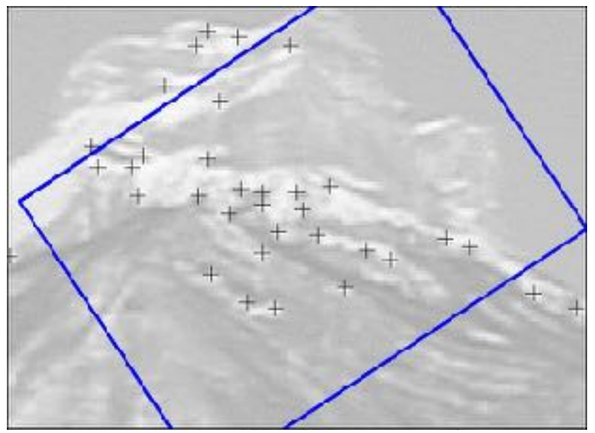


- Interest points have to be repeatable for scale changes

Harris detector + scale changes





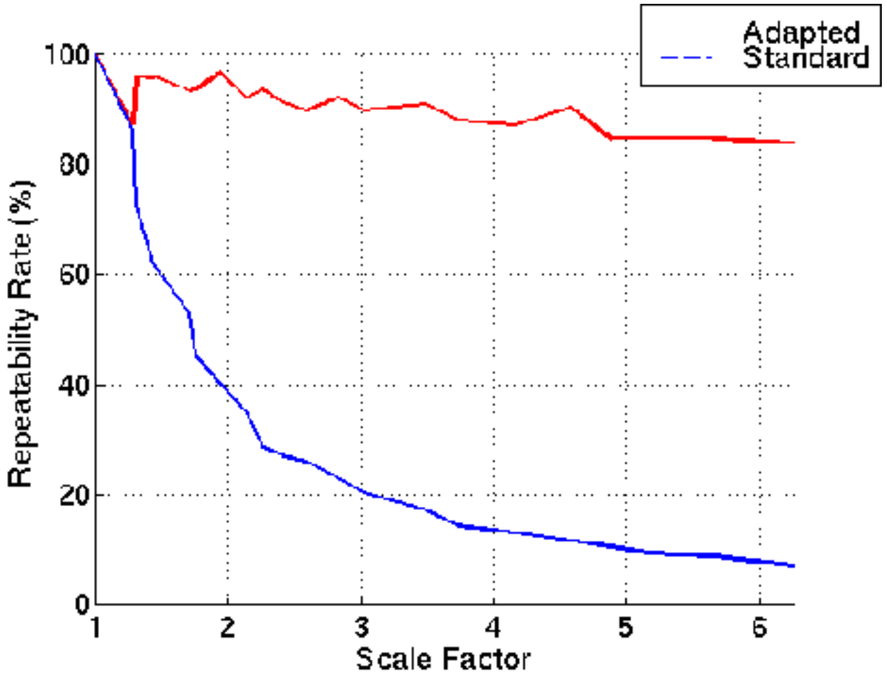


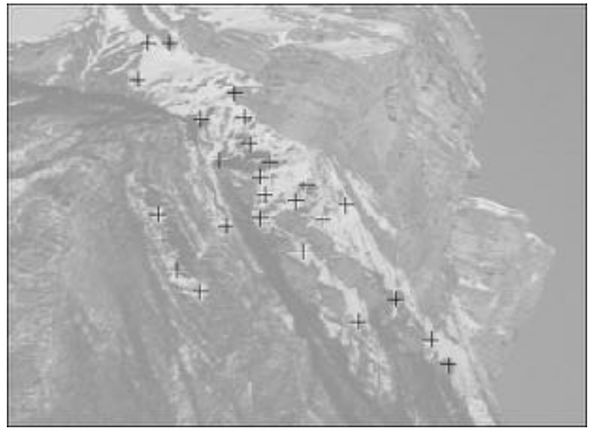
Repeatability rate

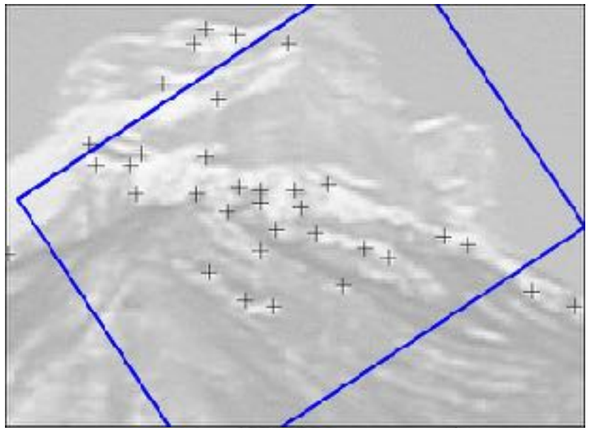
$$R(\varepsilon) = \frac{|\{(\mathbf{a}_i, \mathbf{b}_i) \,|\, dist(H(\mathbf{a}_i), \mathbf{b}_i) < \varepsilon\}|}{\max(|\mathbf{a}_i|, |\mathbf{b}_i|)}$$

Harris detector with adaptation to scale

Scale-adapted derivative calculation



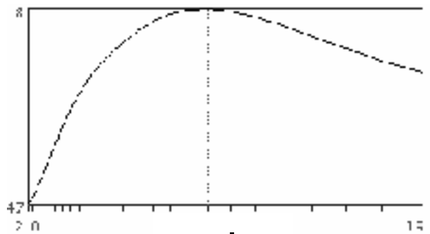




Scale selection

- For a point, compute a value (gradient, Laplacian etc.) at several scales
- Normalization of the values with the scale factor, e.g., Laplacian $|s^2(L_{xx} + L_{yy})|$
- Select scale s^* at the maximum $\rightarrow$ characteristic scale

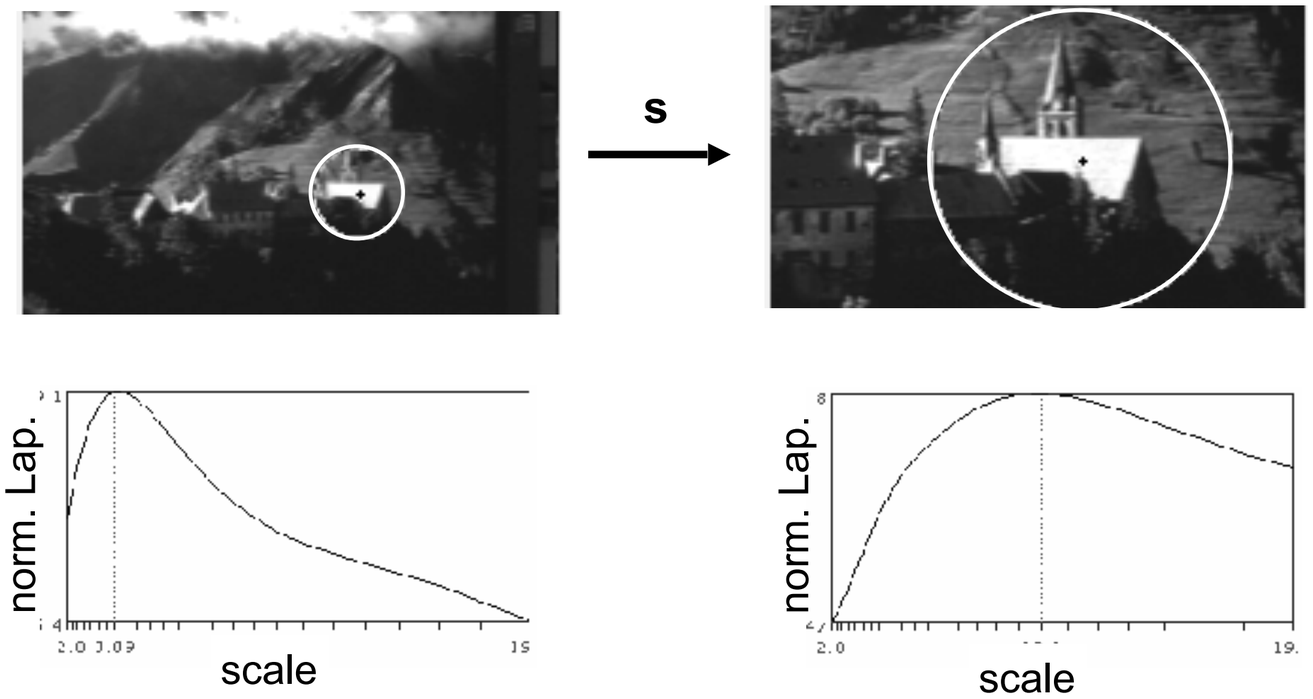
$|s^2(L_{xx} + L_{yy})|$



scale

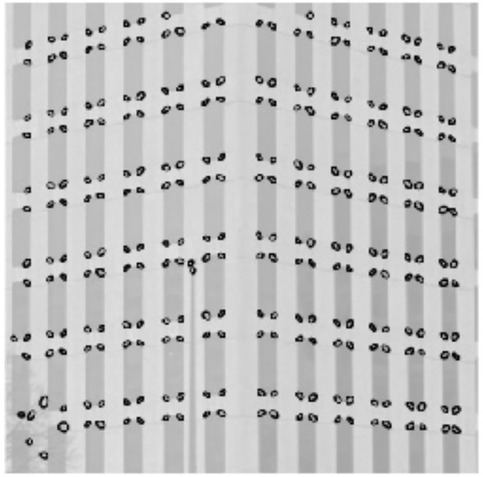
Scale selection

- Scale invariance of the characteristic scale

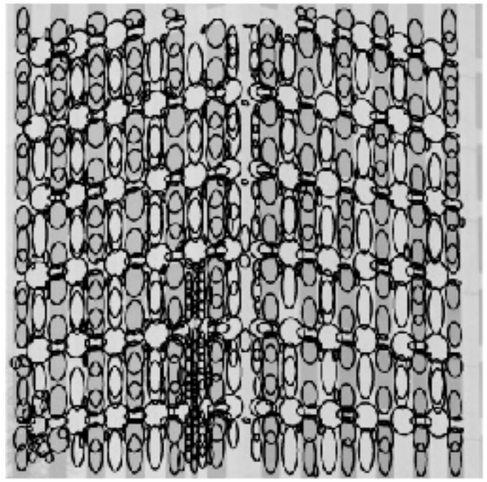


Scale-invariant detectors

- Harris-Laplace (Mikolajczyk & Schmid’01)
- Laplacian detector (Lindeberg’98)
- Difference of Gaussian (SIFT detector, Lowe’99)



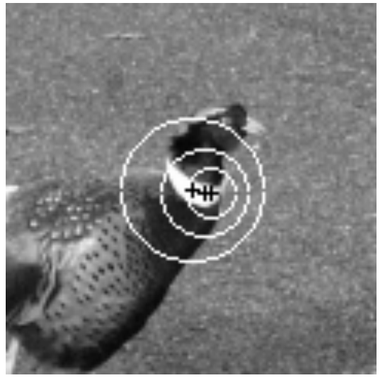
Harris-Laplace

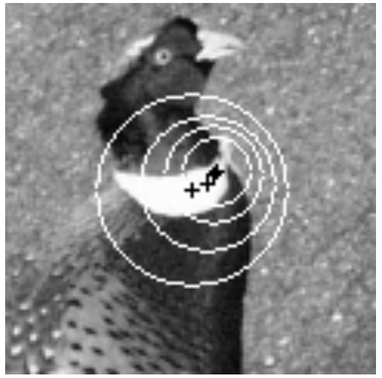


Laplacian

Harris-Laplace

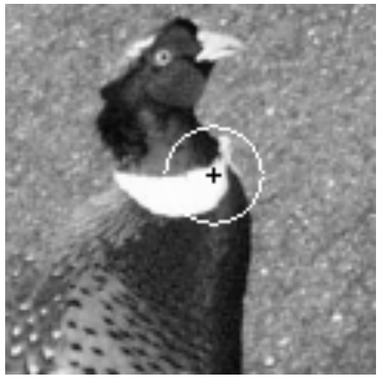
multi-scale Harris points





selection of points at

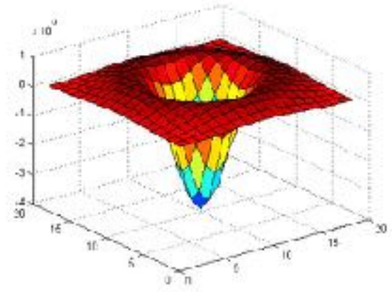
maximum of Laplacian



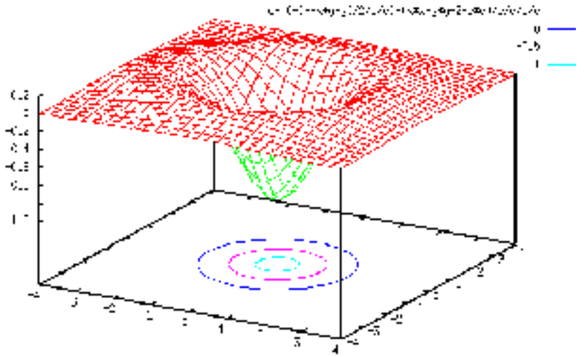
➡ invariant points + associated regions [Mikolajczyk & Schmid’01]

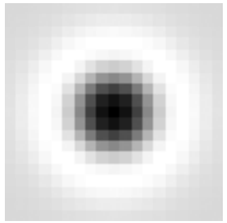
LOG detector

$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$



Laplacian of Gaussian (LOG): Circularly symmetric operator for **blob detection in 2D**

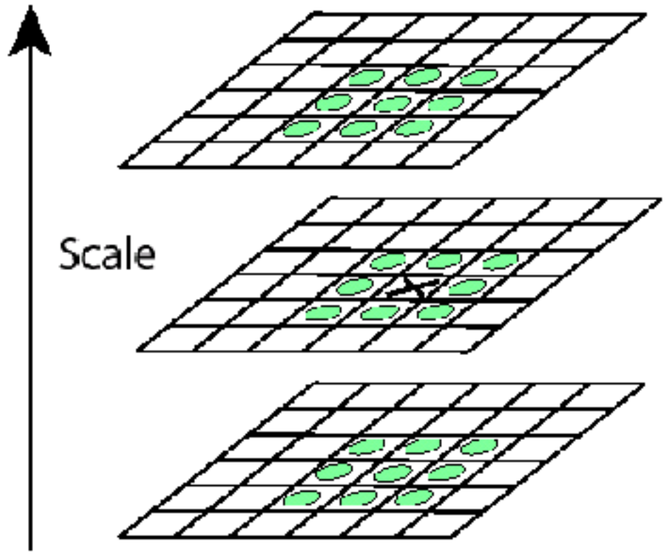




$$LOG = s^2(G_{xx}(\sigma) + G_{yy}(\sigma))$$

Convolve image with scale-normalized Laplacian at several scales

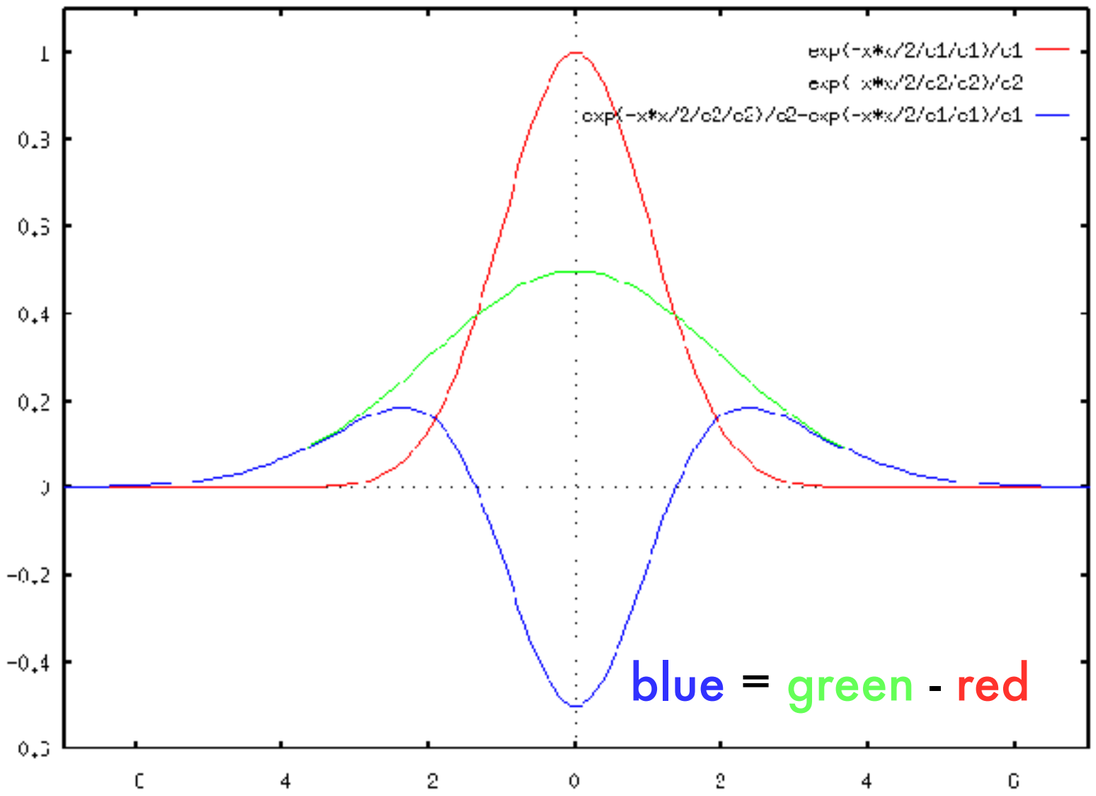
Detection of maxima and minima of Laplacian in scale space



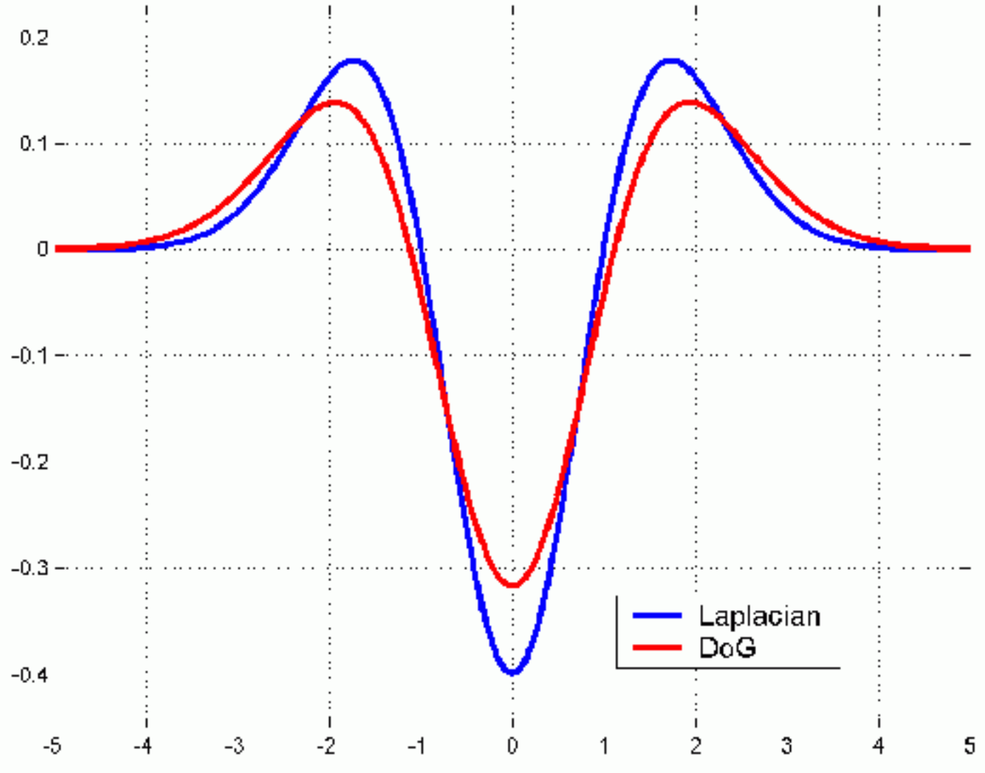
Efficient implementation: DOG (SIFT) detector

- Difference of Gaussian (DOG) approximates the Laplacian

$$DOG = G(k\sigma) - G(\sigma)$$

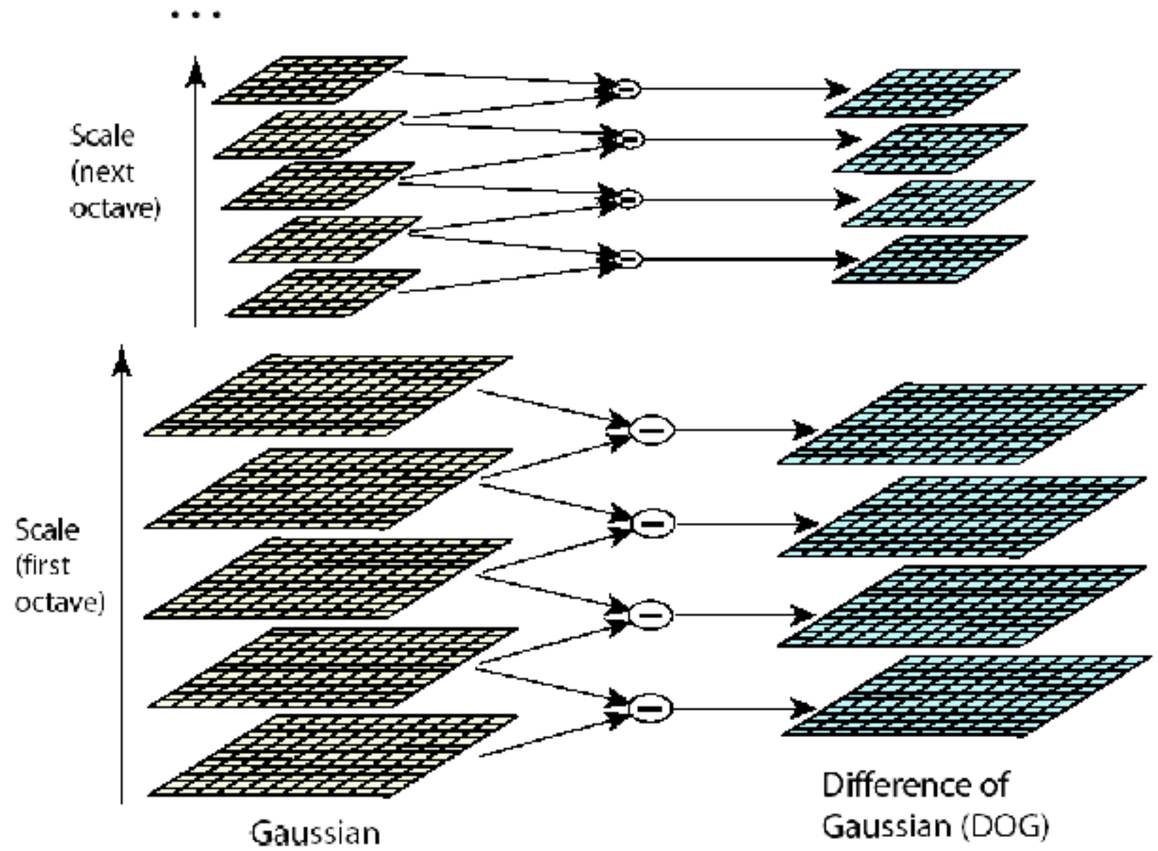


- Error due to the approximation



Efficient implementation: DOG (SIFT) detector

- Fast computation, scale space processed one octave at a time



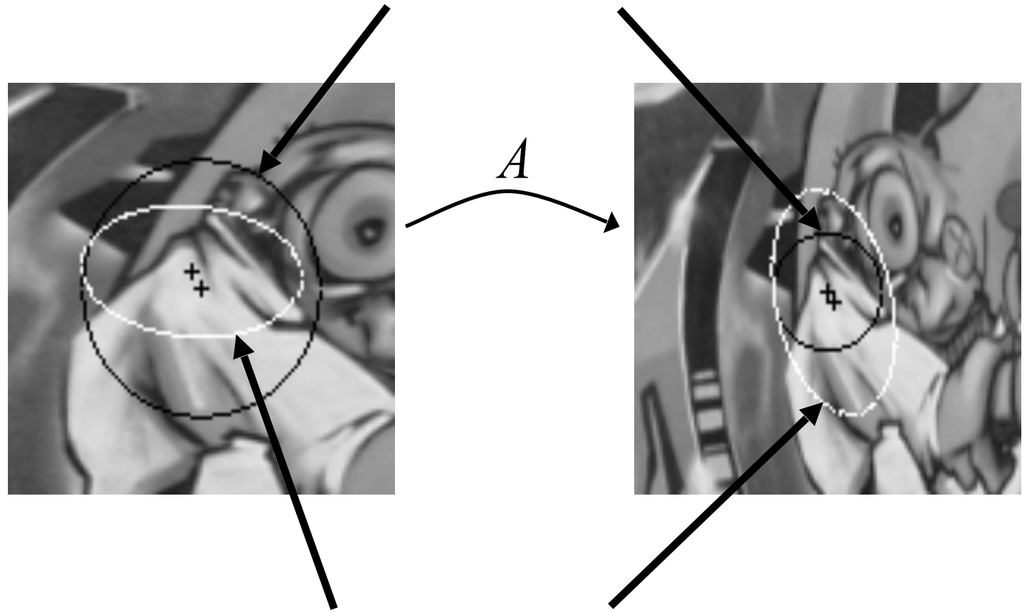
David G. **Lowe**. "Distinctive image features from scale-invariant keypoints."I*JCV* 60 (2), **2004**.

Efficient implementation: DOG (SIFT) detector

Not covered: Affine invariant regions

- Scale invariance is not sufficient for large baseline changes

detected scale invariant region



projected regions, viewpoint changes can locally be approximated by an affine transformation A

We have detected interest points, let’s now **compare patches around those points**.

Agenda: Instance-level recognition

1) Introduction to local features

2) Interest point detectors (e.g., Harris, scale invariance)

3) Comparison of patches (SSD, ZNCC on pixel values)

4) Feature descriptors (e.g., SIFT)

5) Matching and recognition with local features

6) Local feature aggregation for a single image-level description

Comparison of patches - SSD (sum of squared differences)

Comparison of the intensities in the neighborhood of two interest points

$$\frac{1}{(2N+1)^2}\sum_{i=-N}^{N}\sum_{j=-N}^{N}\left(I_1(x_1+i,y_1+j)-I_2(x_2+i,y_2+j)\right)^2$$

Small difference values → similar patches

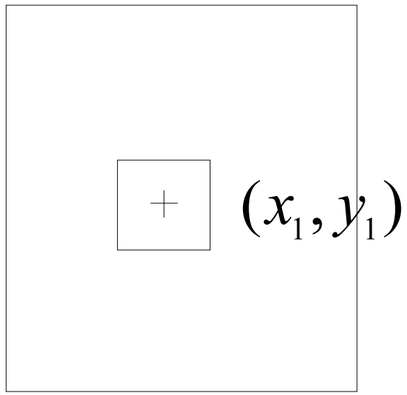


image 1

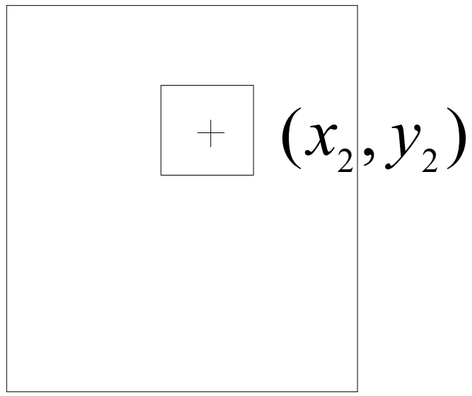


image 2

Comparison of patches - Zero-normalized SSD

SSD : $\frac{1}{(2N+1)^2}\sum_{i=-N}^{N}\sum_{j=-N}^{N}(I_1(x_1+i,y_1+j)-I_2(x_2+i,y_2+j))^2$

Invariance to photometric transformations?

Intensity changes (I → I + b)

=> Normalizing with the mean of each patch

$$\frac{1}{(2N+1)^2}\sum_{i=-N}^{N}\sum_{j=-N}^{N}((I_1(x_1+i,y_1+j)-m_1)-(I_2(x_2+i,y_2+j)-m_2))^2$$

Intensity changes (I → aI + b)

=> Normalizing with the mean and standard deviation of each patch

$$\frac{1}{(2N+1)^2}\sum_{i=-N}^{N}\sum_{j=-N}^{N}\left(\frac{I_1(x_1+i,y_1+j)-m_1}{\sigma_1}-\frac{I_2(x_2+i,y_2+j)-m_2}{\sigma_2}\right)^2$$

Zero-normalized cross correlation (ZNCC)

Zero-normalized SSD (sum of squared differences)

$$\frac{1}{(2N+1)^2}\sum_{i=-N}^{N}\sum_{j=-N}^{N}\left(\frac{I_1(x_1+i,y_1+j)-m_1}{\sigma_1}-\frac{I_2(x_2+i,y_2+j)-m_2}{\sigma_2}\right)^2$$

Zero-normalized cross correlation (ZNCC)

$$\frac{1}{(2N+1)^2}\sum_{i=-N}^{N}\sum_{j=-N}^{N}\left(\frac{I_1(x_1+i,y_1+j)-m_1}{\sigma_1}\right)\cdot\left(\frac{I_2(x_2+i,y_2+j)-m_2}{\sigma_2}\right)$$

ZNCC values between -1 and 1, 1 when identical patches
in practice threshold around 0.5

Invariance to rotation?

Agenda: Instance-level recognition

1) Introduction to local features

2) Interest point detectors (e.g., Harris, scale invariance)

3) Comparison of patches (SSD, ZNCC on pixel values)

4) Feature descriptors (e.g., SIFT)

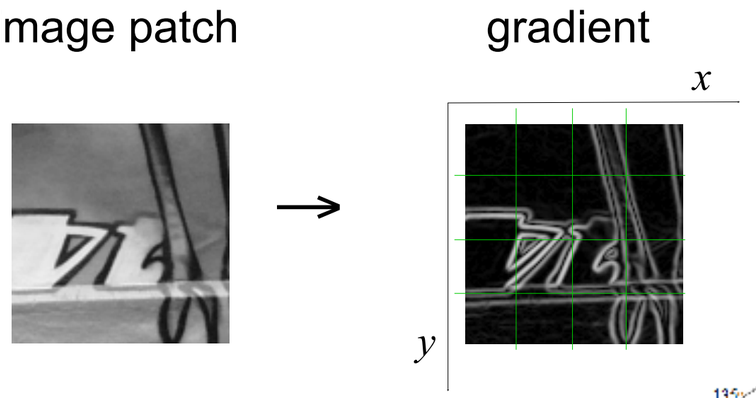
5) Matching and recognition with local features

6) Local feature aggregation for a single image-level description

Local descriptors (patch representation)

- Pixel values
- Greyvalue derivatives, differential invariants [Koenderink'87]
- SIFT descriptor [Lowe'99]
- SURF descriptor [Bay et al.'08]
- DAISY descriptor [Tola et al.'08, Windler et al'09]
- LIOP descriptor [Wang et al.'11]
- Patch descriptors based on CNN features [Brox et al.'15, Paulin et al.'15, Zagoruyko'15…]
- …

SIFT descriptor [Lowe'99]



- Descriptor computation:
  - Divide patch into 4x4 sub-patches
  - Compute histogram of gradient orientations (8 reference angles) inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions

- Soft-assignment to spatial bins
- Normalization of the descriptor to norm one
  - Robustness to illumination changes
- Comparison with Euclidean distance

- Advantage over raw vectors of pixel values
  - Gradients less sensitive to illumination change
  - Pooling of gradients over the sub-patches achieves robustness to small shifts, but still preserves some spatial information

SIFT descriptor - rotation invariance

(Rotational normalization)

- Estimation of the dominant orientation
  - Extract gradient orientations
  - Create histogram over gradient orientations in the patch
  - Assign canonical orientation at peak of this histogram

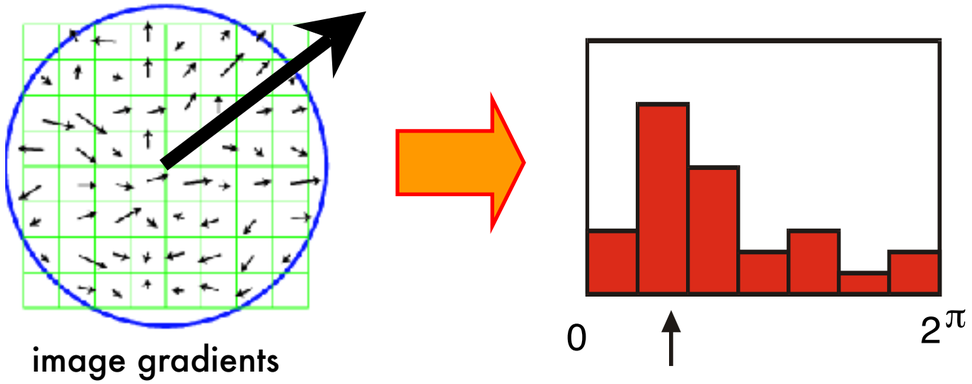
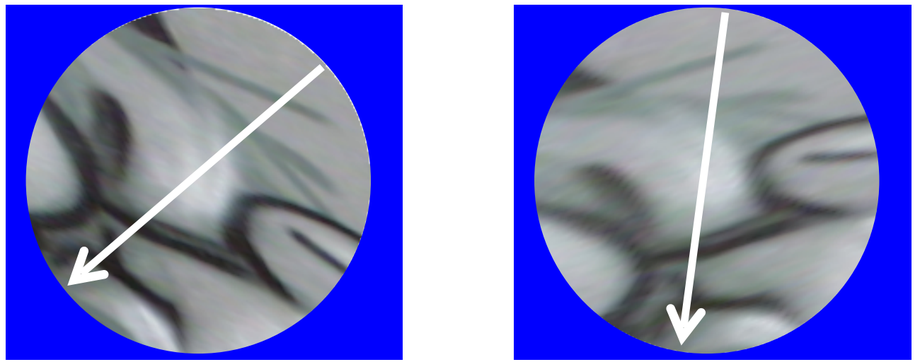


image gradients
(e.g., 8x8 pixel patch)

- Rotate patch in dominant direction



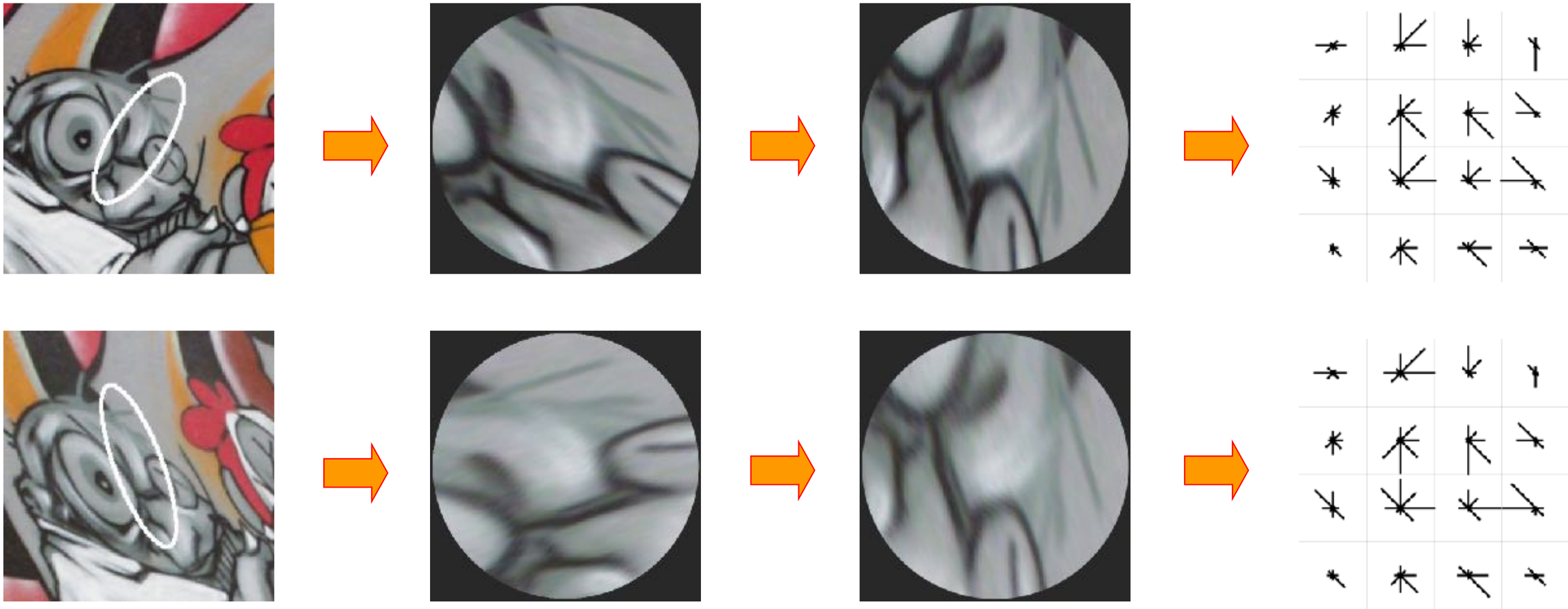
SIFT descriptor - rotation invariance

Extract affine regions

Normalize regions

Eliminate rotational ambiguity

Compute appearance descriptors

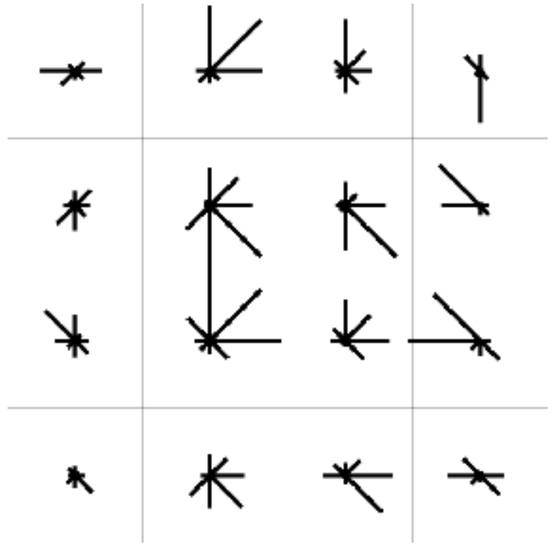


SIFT (Lowe '04)

SIFT detector and SIFT descriptor

SIFT detector
Interest points

SIFT descriptor
128-d representation of the patch



SIFT (Lowe '04)

(Parenthesis: CNN based descriptors)

"Learned" features in upcoming lectures

- Based on global / full image features
  - Does not find patch-level matches
  - More compact
  - Example: Deep Image Retrieval: Learning global representations for image search (DIR) [ECCV 2016]

- Based on local features
  - Patch-level matches possible
  - Indexing scheme necessary
  - Example: Large-Scale Image Retrieval with Attentive Deep Local Features (DELF) [ICCV 2017]

Agenda: Instance-level recognition

1) Introduction to local features

2) Interest point detectors (e.g., Harris, scale invariance)

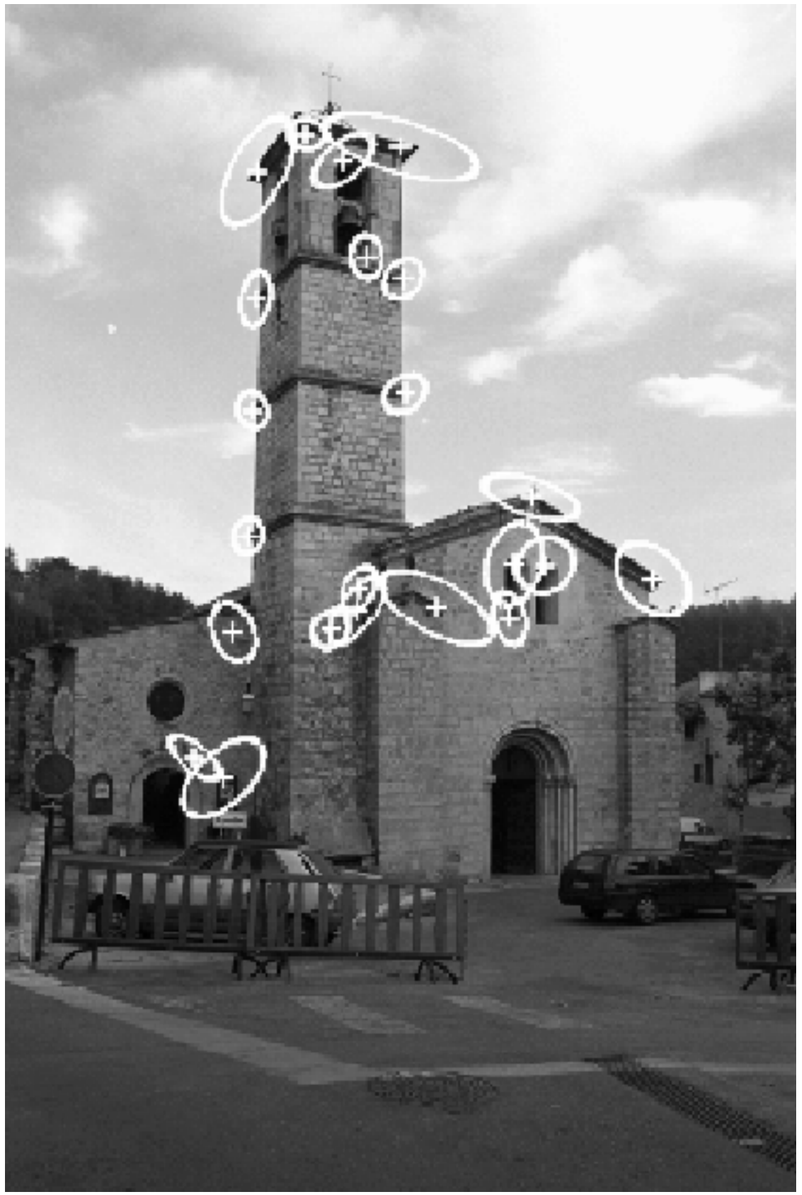
3) Comparison of patches (SSD, ZNCC on pixel values)

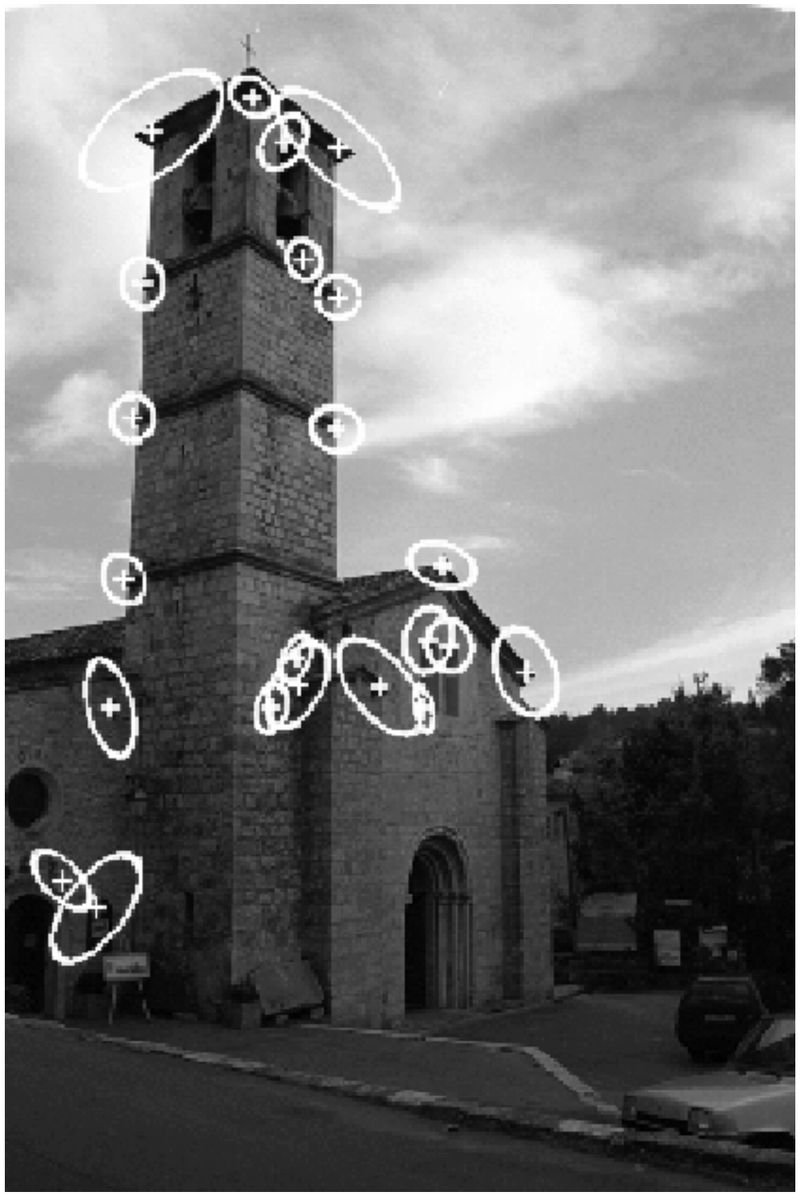
4) Feature descriptors (e.g., SIFT)

5) Matching and recognition with local features

6) Local feature aggregation for a single image-level description

Matching of descriptors





Matching of descriptors

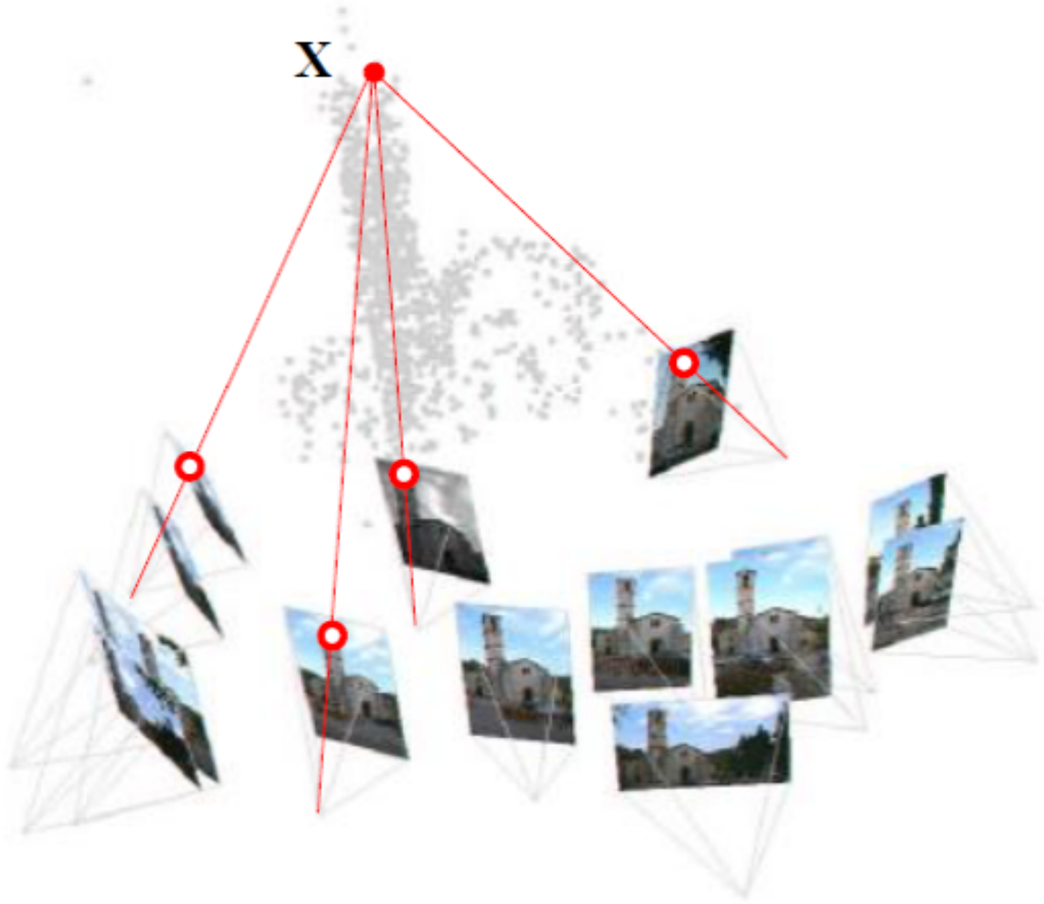
Matching and 3D reconstruction

- Establish correspondence between two (or more) images

[Schaffalitzky and Zisserman ECCV 2002]

Matching and 3D reconstruction

- Establish correspondence between two (or more) images



[Schaffalitzky and Zisserman ECCV 2002]

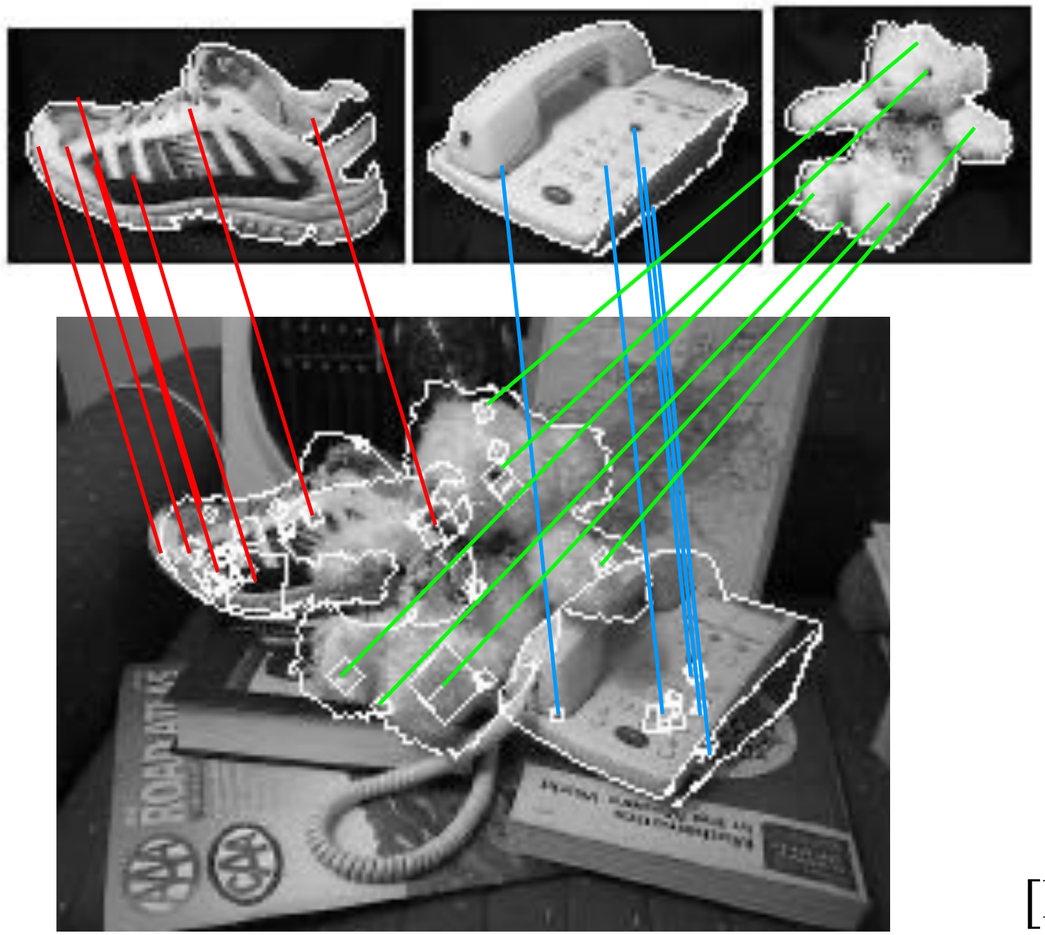
Building Rome in a Day

57,845 downloaded images, 11,868 registered images

[Agarwal, Snavely, Simon, Seitz, Szeliski, ICCV'09]

Object recognition

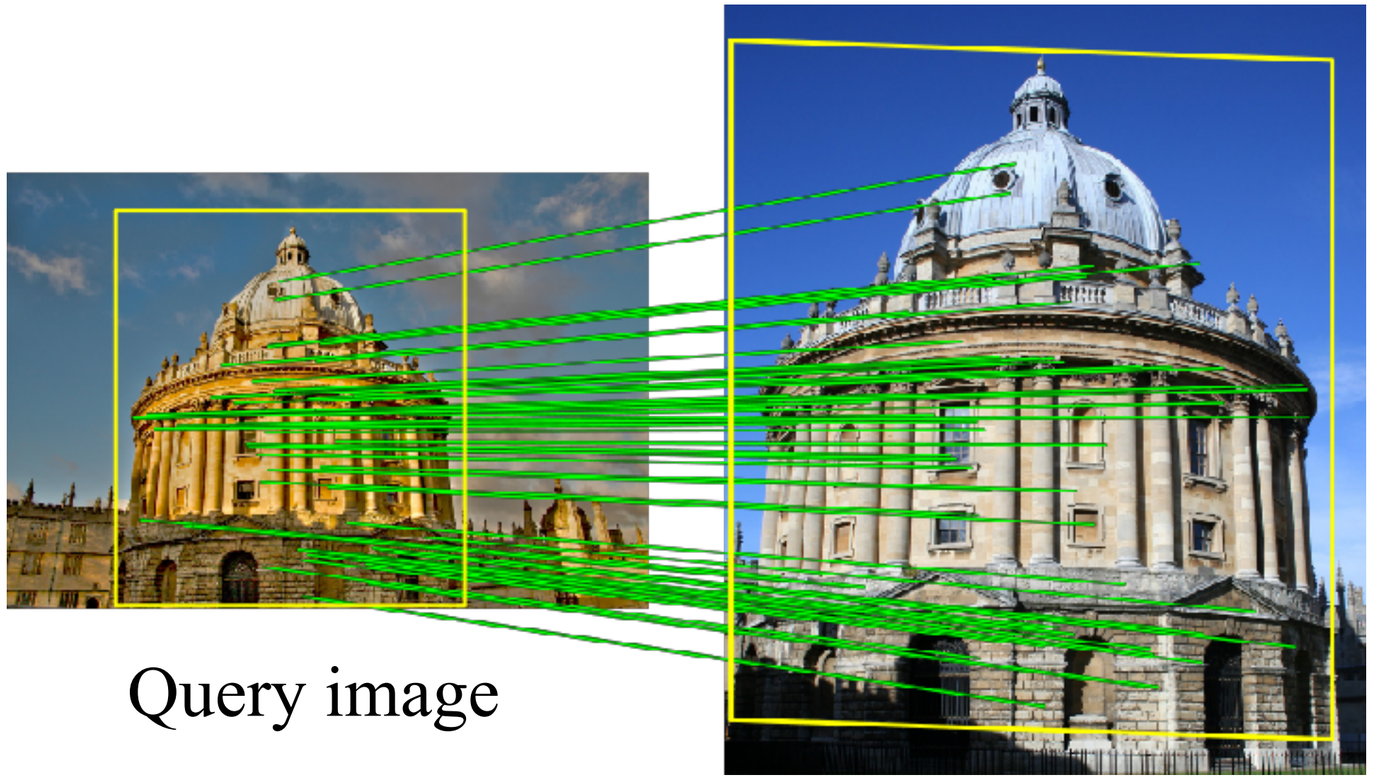
- Establish correspondence between the target image and (multiple) images in the model database



[D. Lowe, 1999]

Visual search

- Establish correspondence between the query image and all images from the database depicting the same object or scene

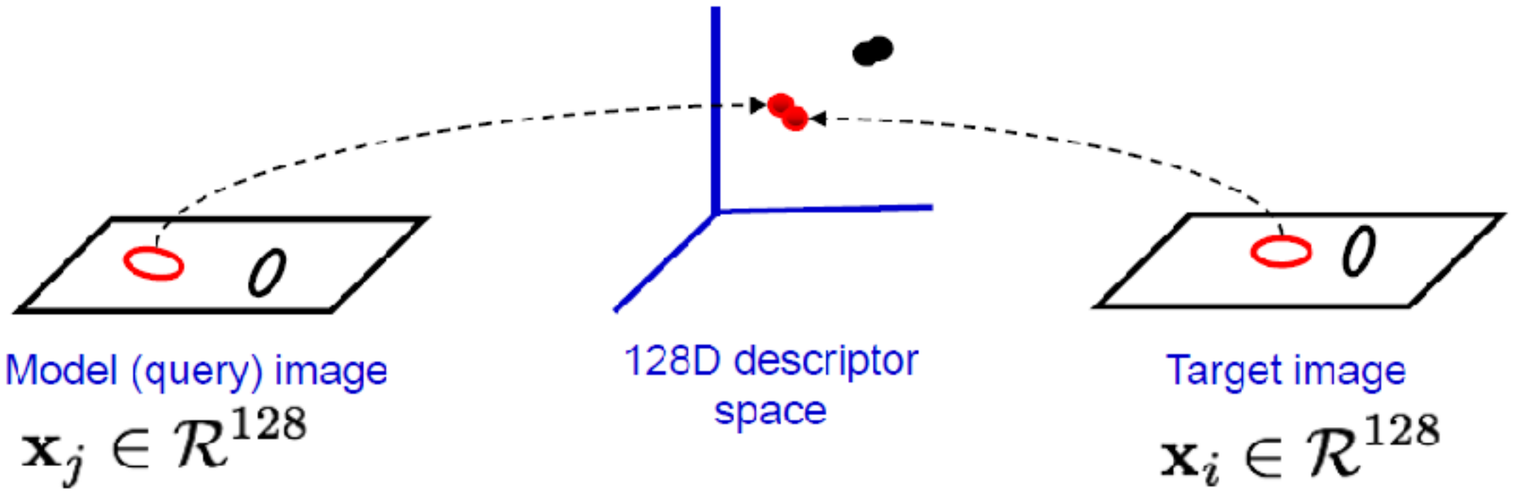


Query image

Database image(s)

Matching of descriptors

- Find the nearest neighbor in the second image for each descriptor, for example SIFT



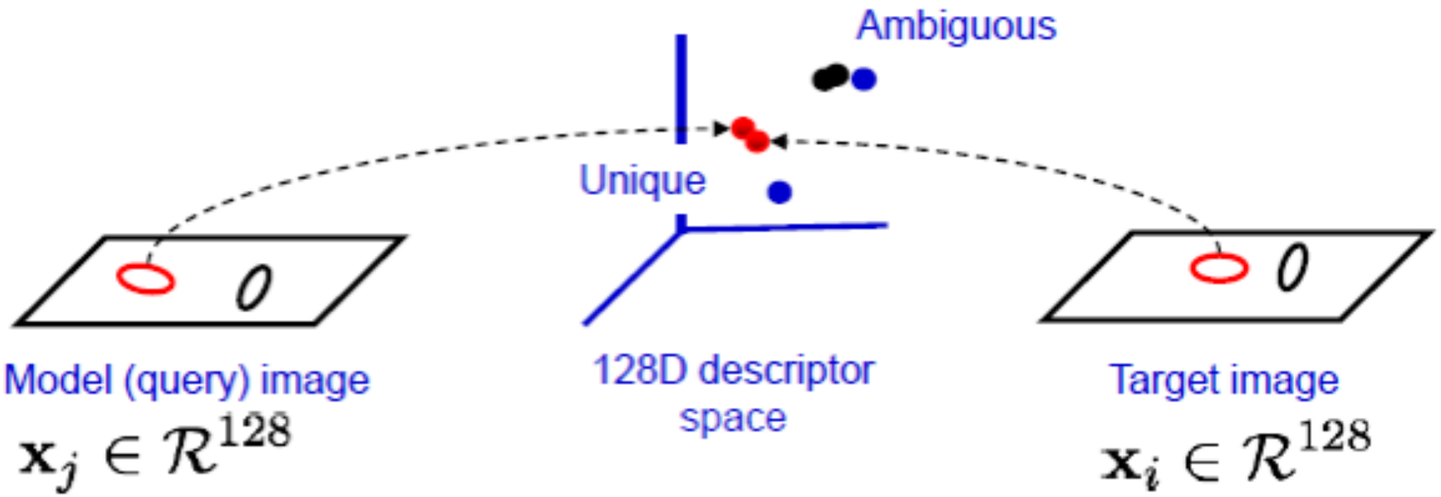
Need to solve some variant of the "nearest neighbor problem" for all feature vectors, $\mathbf{x}_j \in \mathcal{R}^{128}$, in the query image:

$$\forall j \; NN(j) = \arg\min_i ||\mathbf{x}_i - \mathbf{x}_j||,$$

where, $\mathbf{x}_i \in \mathcal{R}^{128}$, are features in the target image.

Matching of descriptors

- Pruning strategies
  - Ratio with respect to the second best match (d1/d2 << 1) [Lowe, ’04]



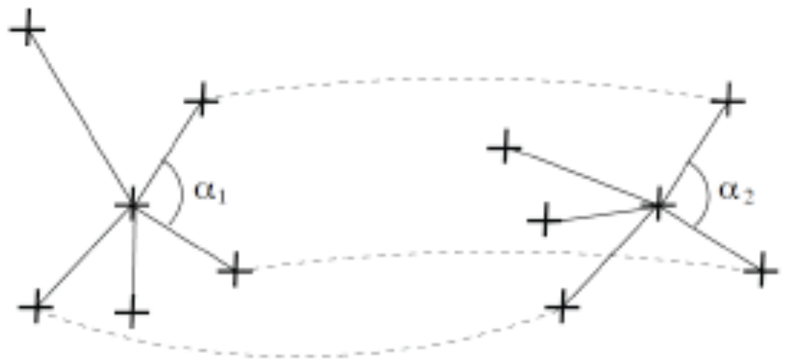
If the 2nd nearest neighbour is much further than the 1st nearest neighbour, the match is more “unique” or discriminative.

Measure this by the ratio: $r = d_{1NN} / d_{2NN}$

r is between 0 and 1
r is small the match is more unique.

Matching of descriptors

- Pruning strategies
  - Ratio with respect to the second best match (d1/d2 << 1)
  - Local neighborhood constraints (semi-local constraints)



Neighbors of the point have to match and angles have to correspond.

Note that in practice not all neighbors have to be matched correctly.

Matching of descriptors

- Pruning strategies
  - Ratio with respect to the second best match (d1/d2 << 1)
  - Local neighborhood constraints (semi-local constraints)
  - Backwards matching (matches are NN in both directions)

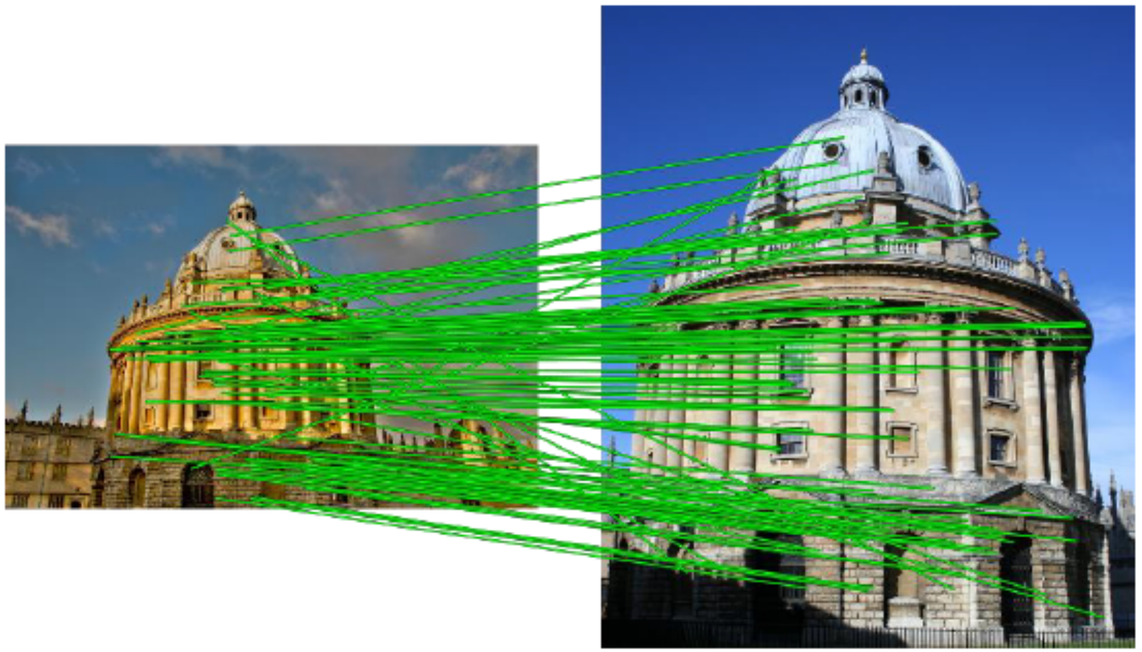
Matching of descriptors

- Pruning strategies
  - Ratio with respect to the second best match (d1/d2 << 1)
  - Local neighborhood constraints (semi-local constraints)
  - Backwards matching (matches are NN in both directions)

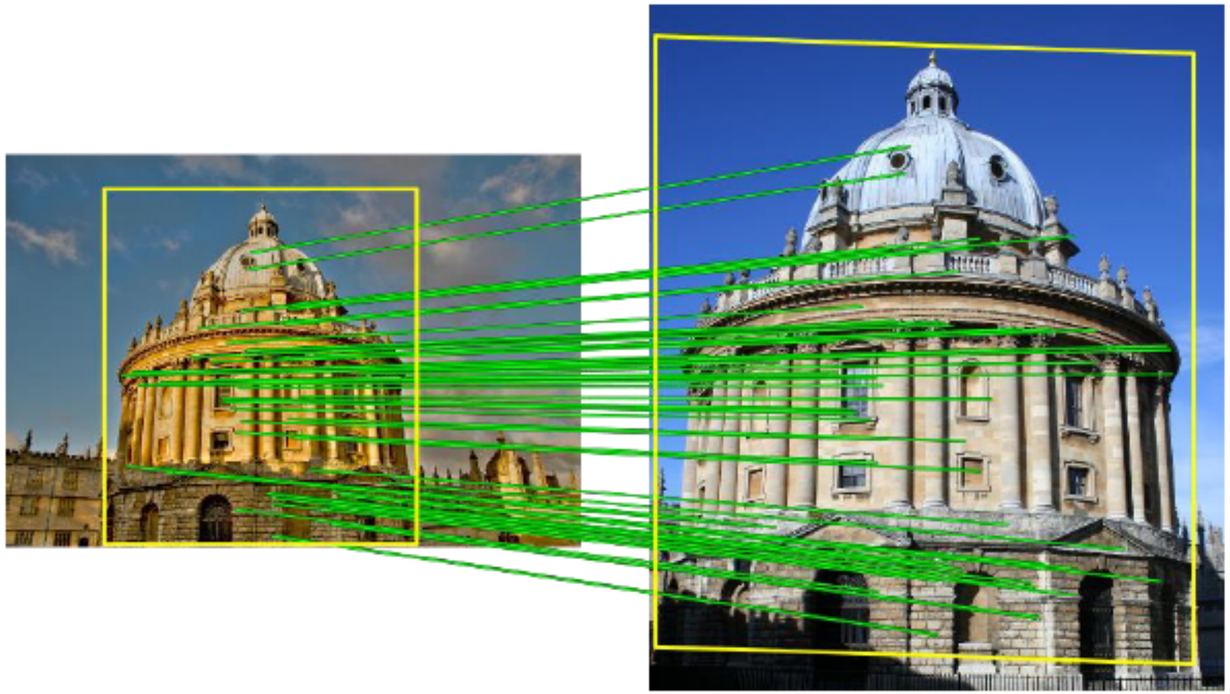
- Geometric verification with global constraint
  - All matches must be consistent with a global geometric transformation
  - However, there are many incorrect matches
  - **Need to estimate simultaneously the geometric transformation and the set of consistent matches**

Geometric verification with global constraint

- Example of a geometric verification



Tentative matches



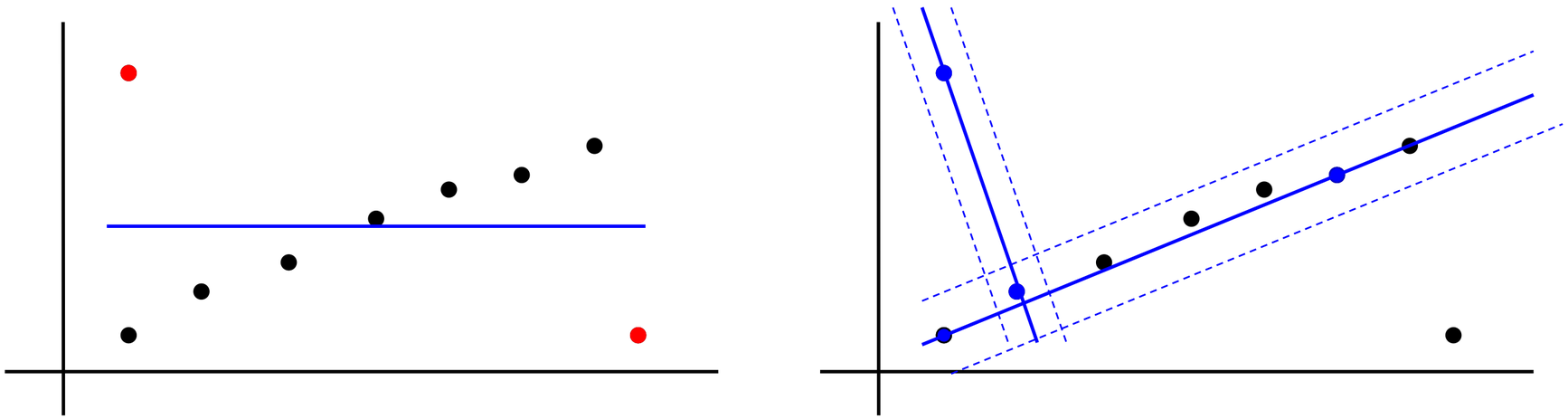
Matches consistent with an affine transformation

Robust estimation of global constraints

- **RANSAC** (RANdom Sampling Consensus) [Fishler&Bolles'81]
- **Hough** transform [Lowe'04]

RANSAC: Example of robust line estimation

Fit a line to 2D data containing outliers



There are two problems

1. a line fit which minimizes perpendicular distance
2. a classification into inliers (valid points) and outliers

Solution: use robust statistical estimation algorithm RANSAC

(RANdom Sample Consensus) [Fishler & Bolles, 1981]

Slide credit: A. Zisserman

RANSAC robust line estimation

Repeat

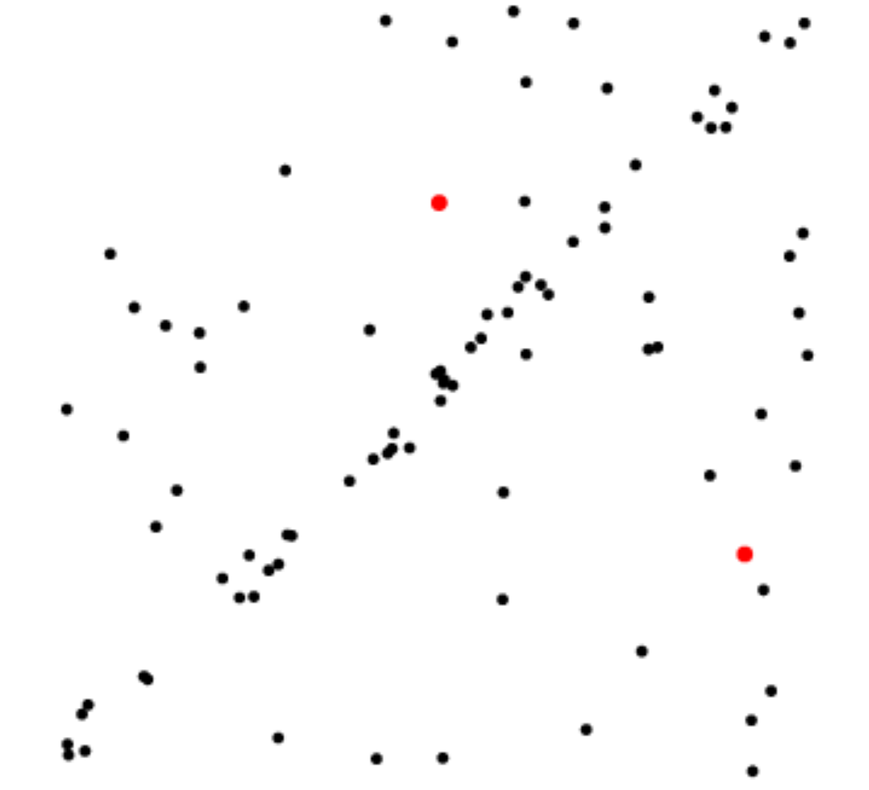
1. Select random sample of 2 points
2. Compute the line through these points
3. Measure support (number of points within threshold distance of the line)

Choose the line with the largest number of inliers

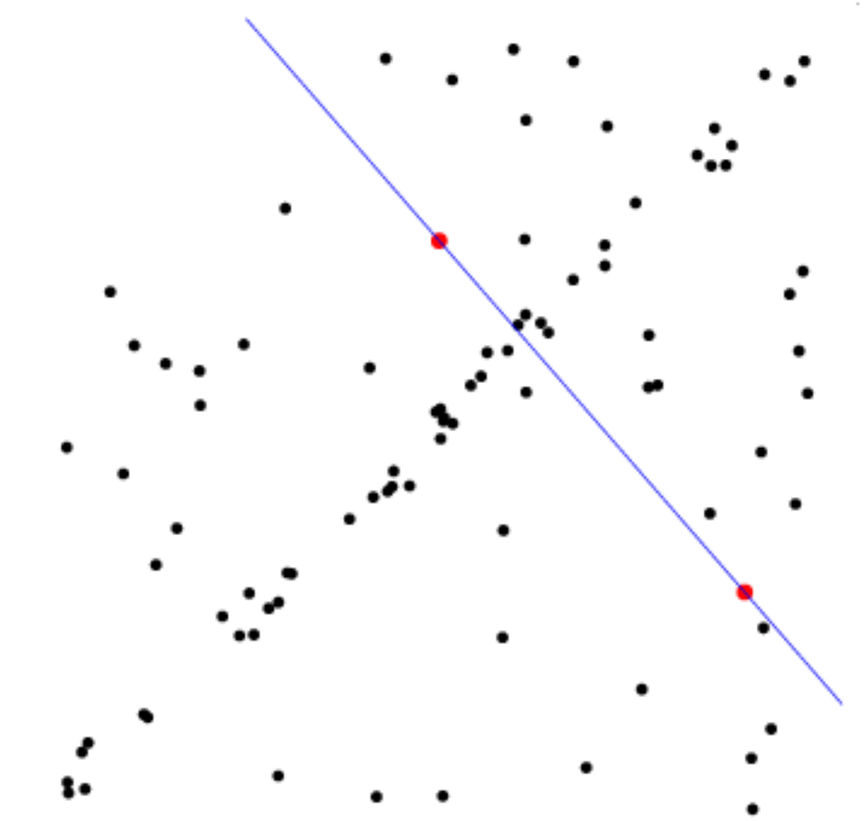
- Compute least squares fit of line to inliers (regression)

Slide credit: A. Zisserman

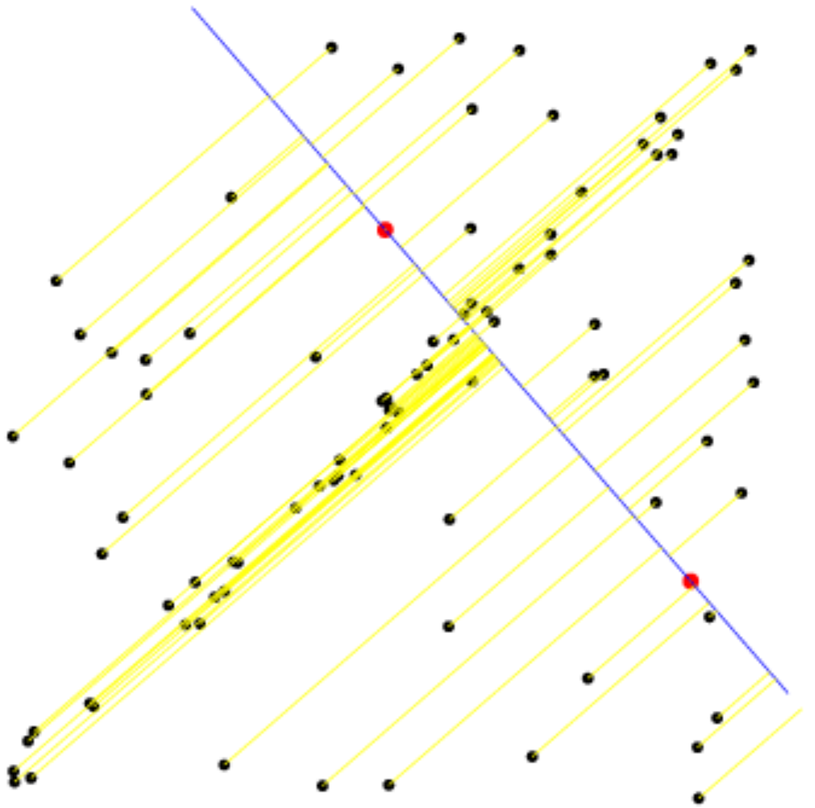
Slide credit: O. Chum



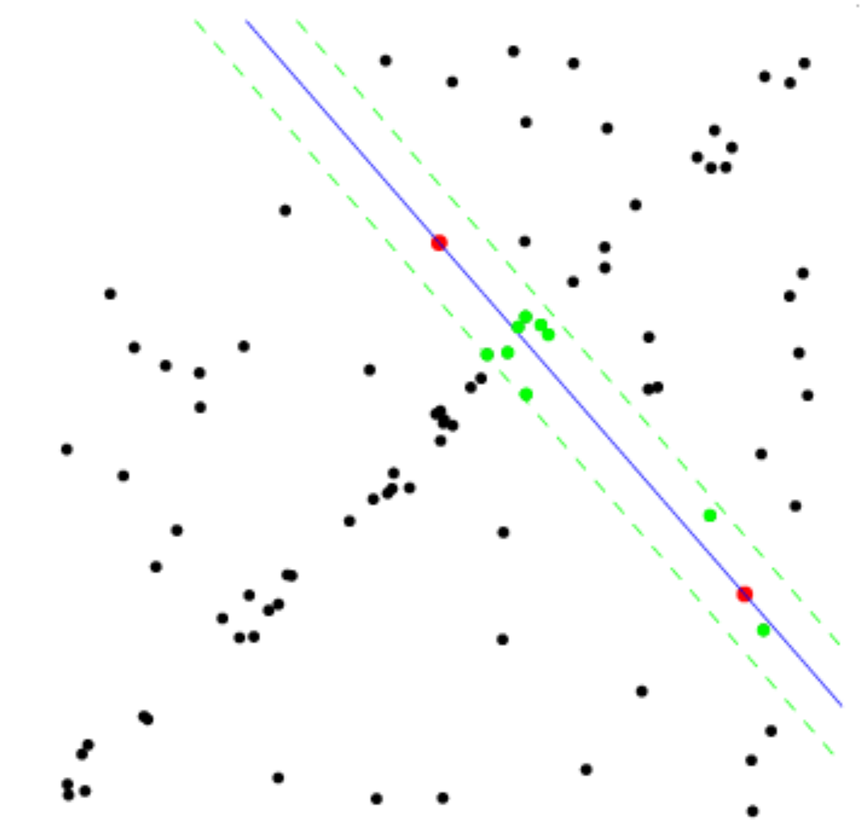
Slide credit: O. Chum



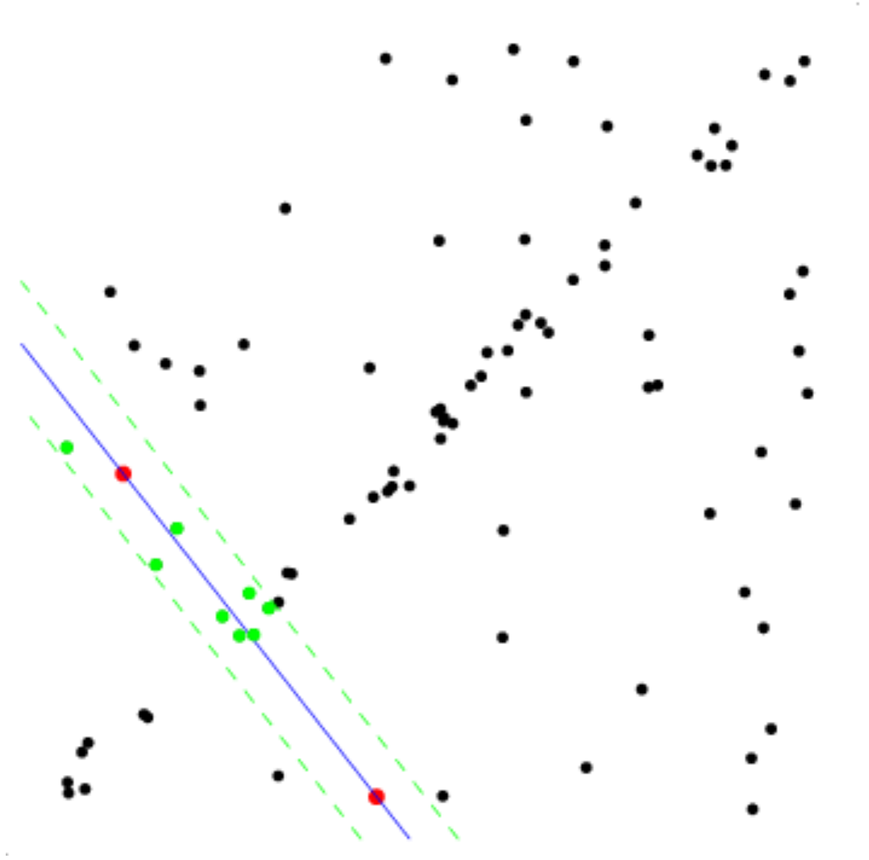
Slide credit: O. Chum



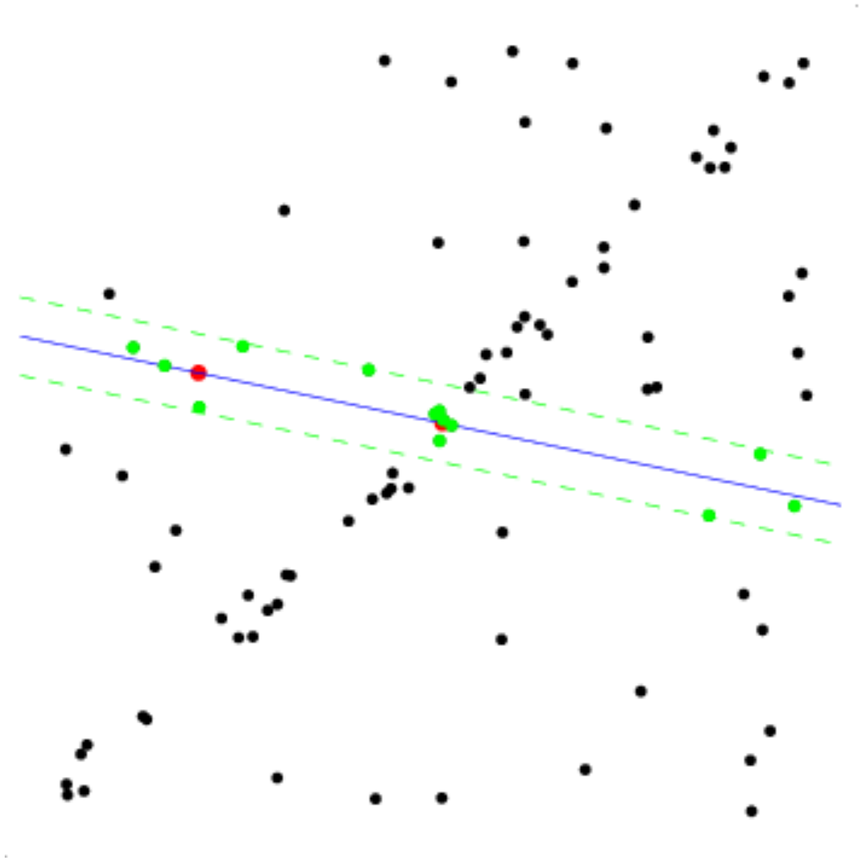
Slide credit: O. Chum



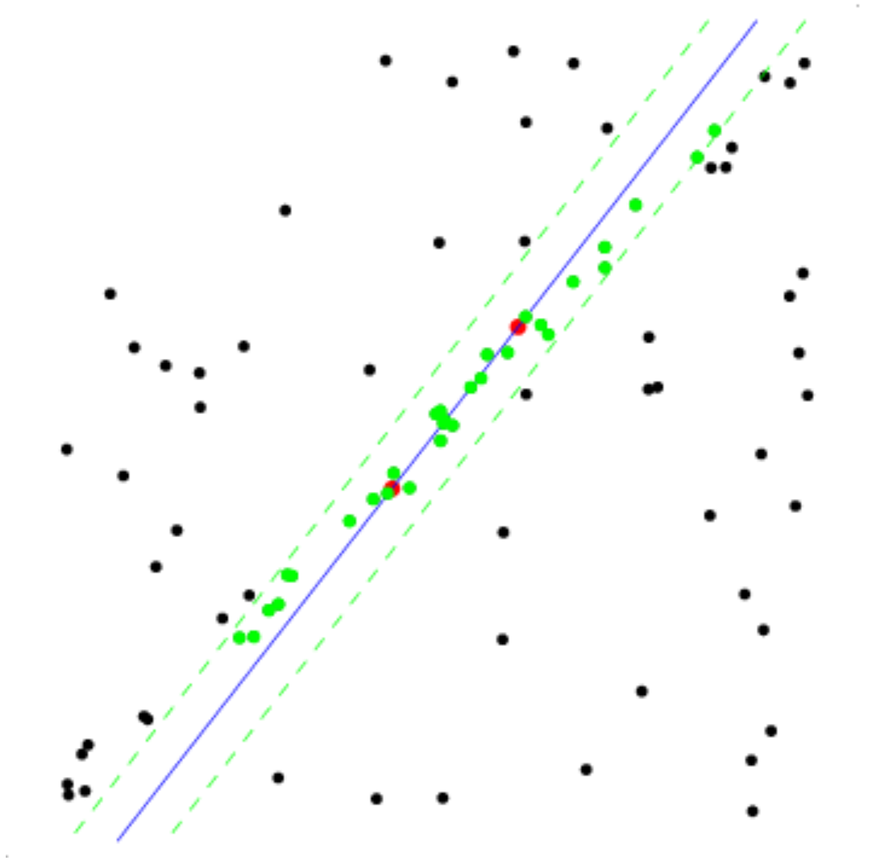
Slide credit: O. Chum



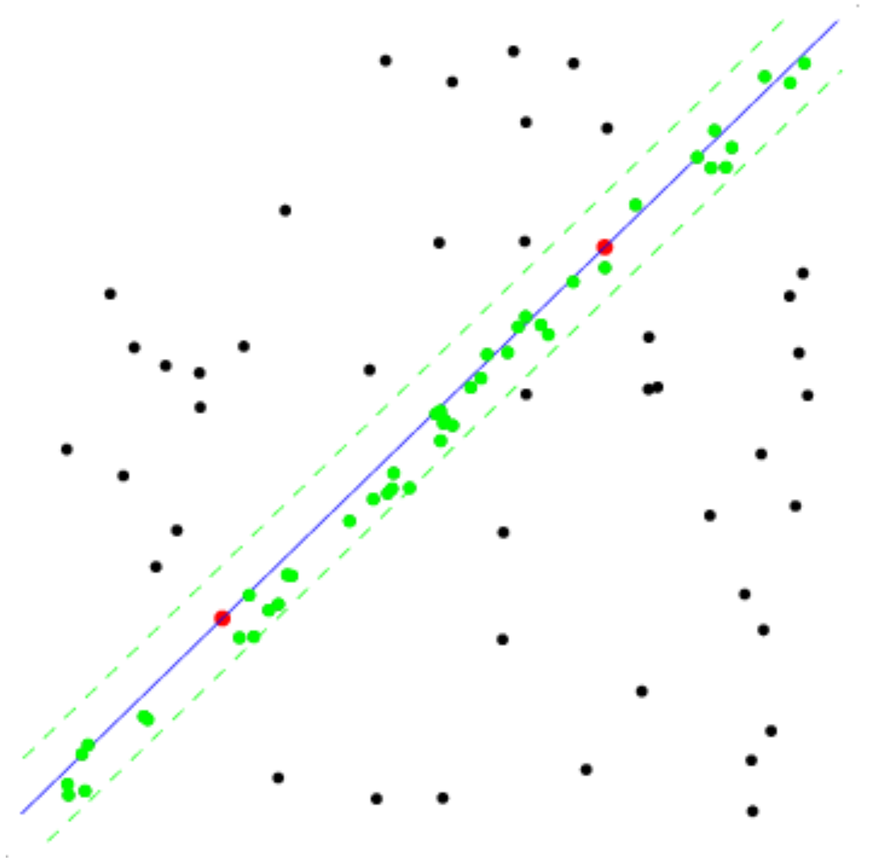
Slide credit: O. Chum



Slide credit: O. Chum



Slide credit: O. Chum

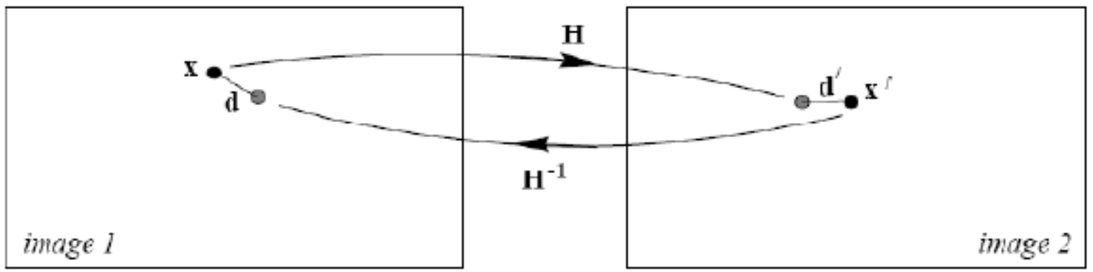


Slide credit: O. Chum

RANSAC Algorithm

- Robust estimation of a homography with RANSAC
  - Repeat
    - Select 4 point matches
    - Compute 3x3 homography
    - Measure support (number of inliers within threshold, i.e. $d^2_{\text{transfer}} < t$)

$$d^2_{\text{transfer}} = d(\mathbf{x}, \mathbf{H}^{-1}\mathbf{x}')^2 + d(\mathbf{x}', \mathbf{H}\mathbf{x})^2$$



  - Choose (H with the largest number of inliers)
  - Re-estimate H with all inliers

Robust estimation of global constraints

- **RANSAC** (RANdom Sampling Consensus) [Fishler&Bolles'81]
- **Hough** transform [Lowe'04]

Strategy 2: Hough transform

- General outline:
  - Discretize parameter space into bins
  - For each feature point in the image, put a **vote** in every bin in the parameter space that could have generated this point
  - Find bins that have the most votes

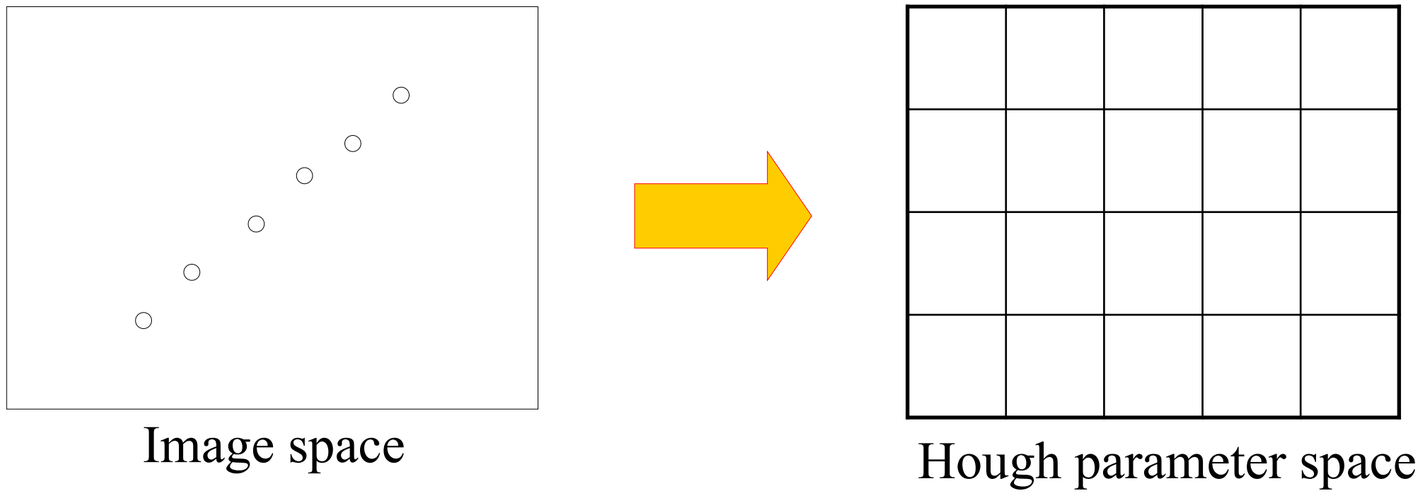


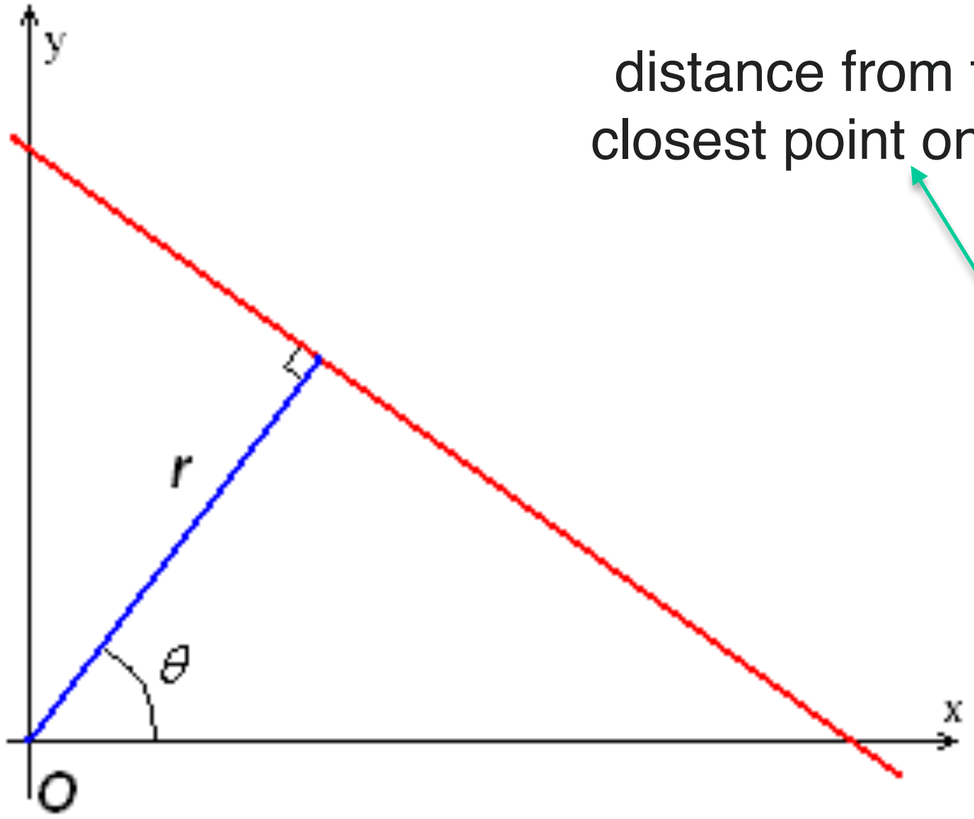
Image space

Hough parameter space

P.V.C. Hough, *Machine Analysis of Bubble Chamber Pictures,* Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959

Hough transform for lines

A straight line $y = mx + b$ can be represented as a point (r, θ) in the parameter space.



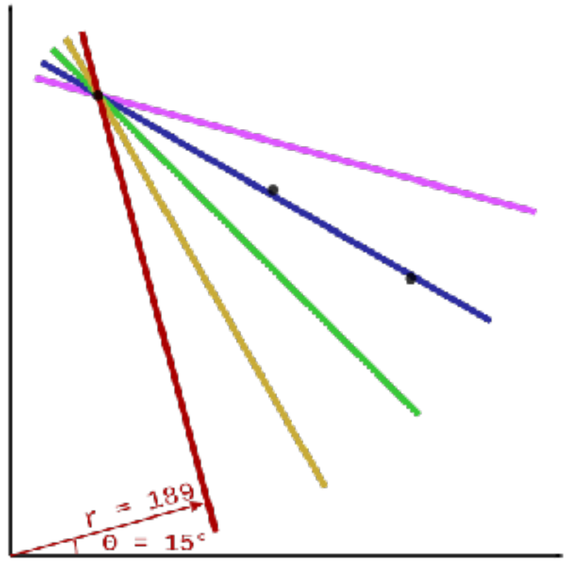
distance from the origin to the closest point on the straight line

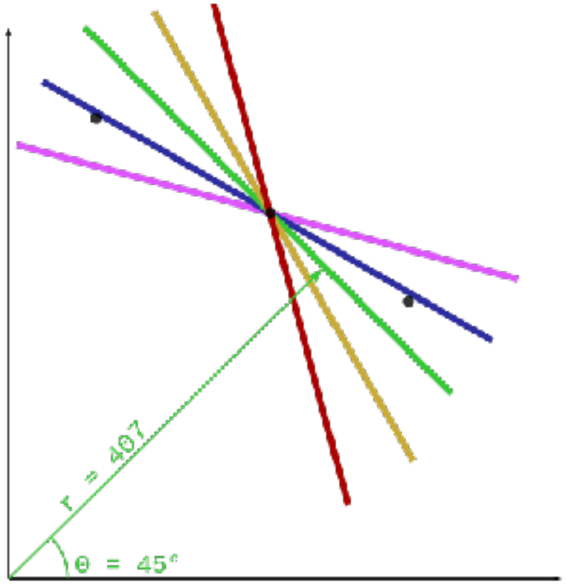
$$r = x\cos\theta + y\sin\theta,$$

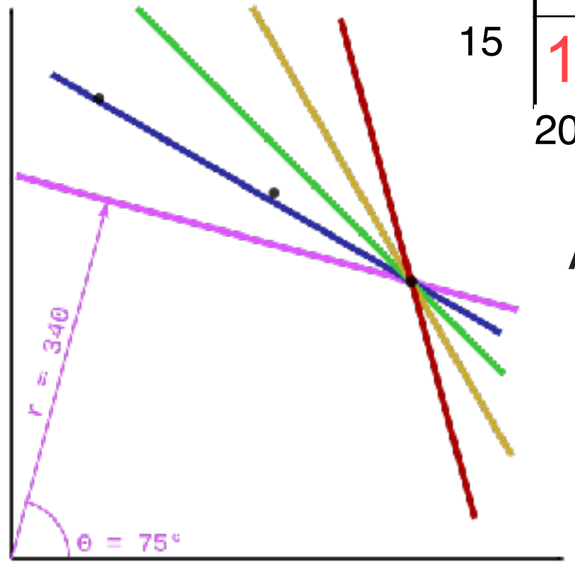
angle between the x-axis and the line connecting the origin with that closest point

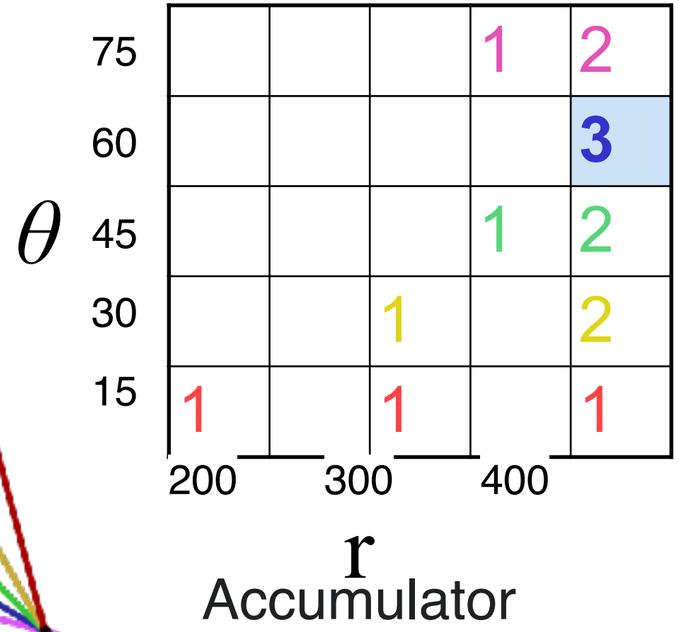
https://en.wikipedia.org/wiki/Hough_transform

Hough transform for lines









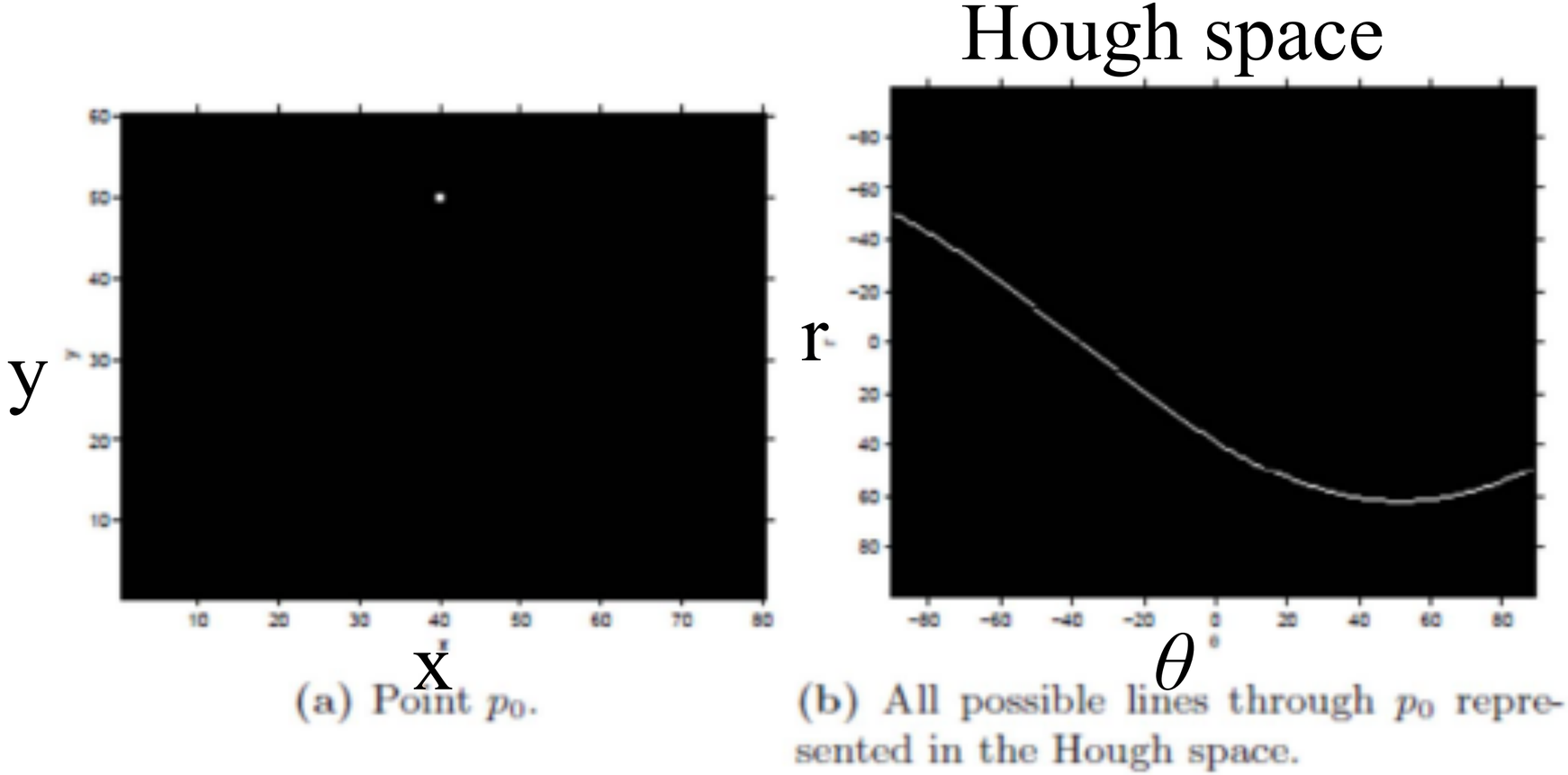
θ	r
15	189.0
30	282.0
45	355.7
60	407.3
75	429.4

θ	r
15	318.5
30	376.8
45	407.3
60	409.8
75	385.3

θ	r
15	419.0
30	443.6
45	438.4
60	402.9
75	340.1

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

Hough transform for lines

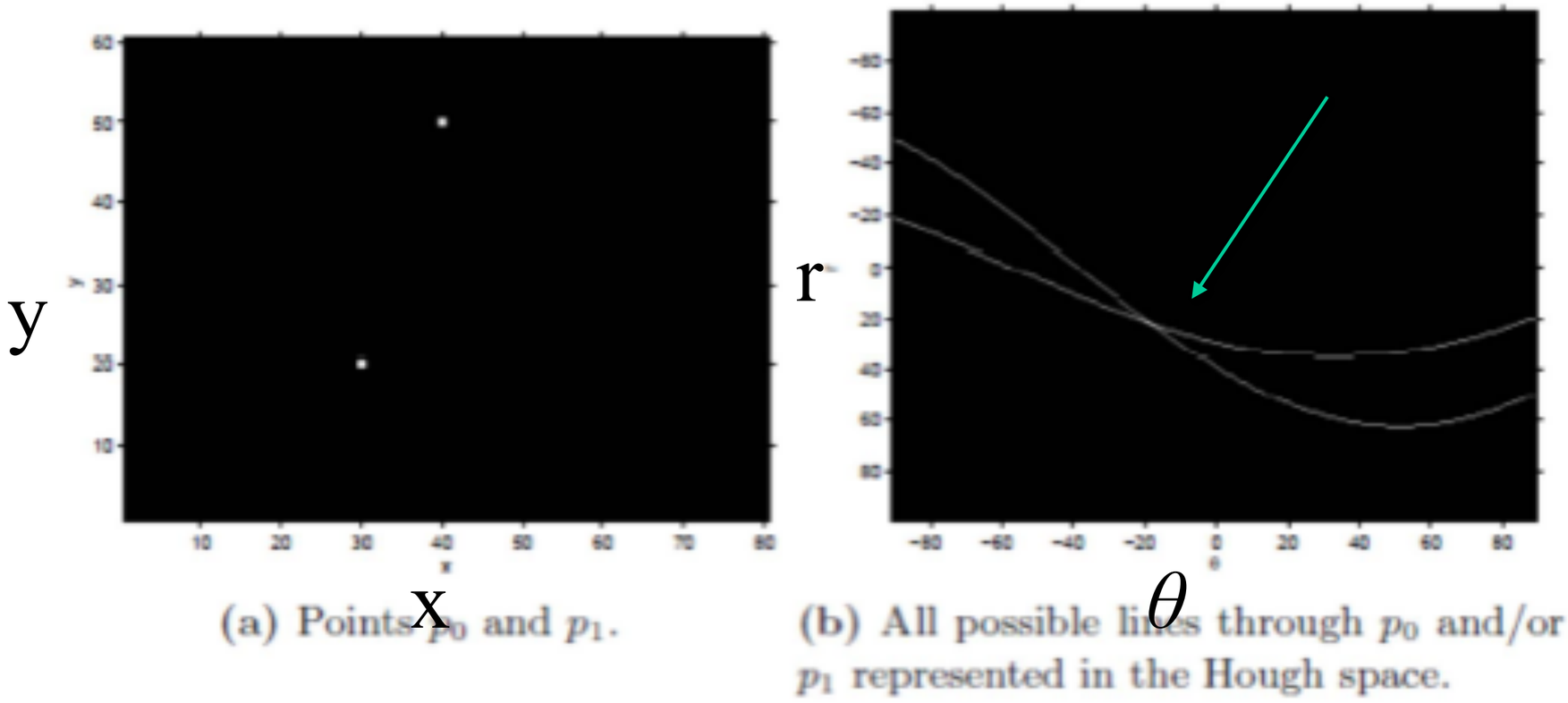


(a) Point p_0.

(b) All possible lines through p_0 represented in the Hough space.

Given a single point in the plane, the set of all straight lines going through that point corresponds to a sinusoidal curve in the (r, θ) plane, which is unique to that point.

Hough transform for lines



(a) Points p_0 and p_1.

(b) All possible lines through p_0 and/or p_1 represented in the Hough space.

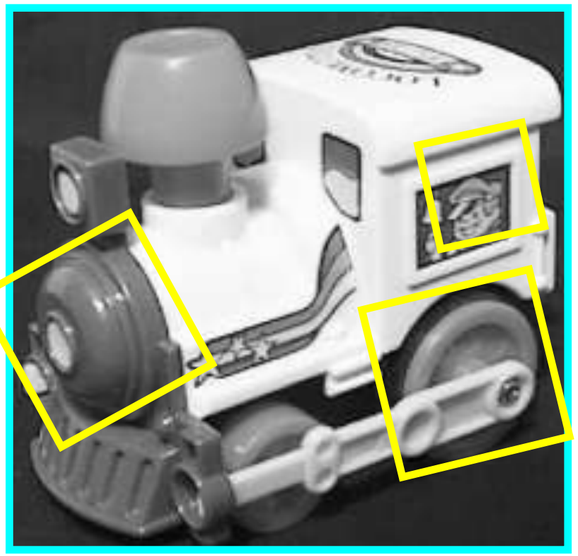
A set of two or more points that form a straight line will produce sinusoids crossing at the (r, θ) for that line.

Hough transform for feature matching (object recognition)

Suppose our features are scale- and rotation-covariant

- Then a single feature match provides an alignment hypothesis: translation (tx, ty), scale (s), orientation (θ)
- Of course, a hypothesis obtained from a single match is unreliable
- Solution: Coarsely **quantize the transformation space**. Let each match vote for its hypothesis in the quantized space.

model



David G. Lowe. **"Distinctive image features from scale-invariant keypoints",** *IJCV* 60 (2), pp. 91-110, 2004.

Hough transform for feature matching

Compute similarity transformation from a single correspondence:

$$(x_A, y_A, s_A, \theta_A) \leftrightarrow (x'_A, y'_A, s'_A, \theta'_A)$$

- Translation (tx, ty)
- Scale (s)
- Orientation (θ)

$$\theta = \theta'_A - \theta_A$$

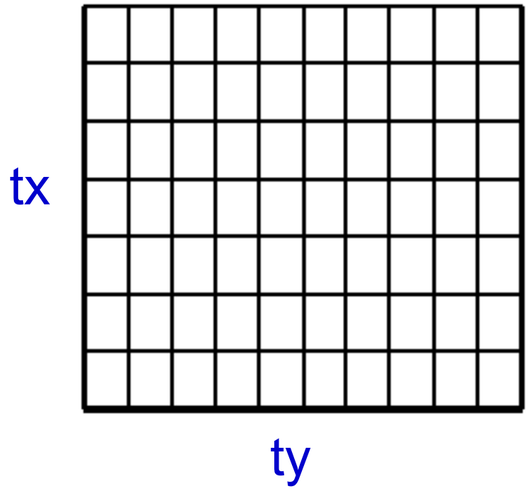
$$s = s'_A / s_A$$

$$t_x = x'_A - sR(\theta)x_A$$

$$t_y = y'_A - sR(\theta)y_A$$

Basic algorithm outline

H: 4D-accumulator array
(only 2-d shown here)




```
1.  Initialize accumulator H to all zeros.
2.  For each tentative match:
        Compute transformation hypothesis: tx, ty, s, θ
        Increase vote H(tx,ty,s,θ) += 1
    end
3.  Find all bins (tx,ty,s,θ) where H(tx,ty,s,θ) has at least 3 votes.
```

- Correct matches will consistently vote for the same transformation,
  - while mismatches will spread votes.
- Cost:
  - Linear scan through the matches (step 2),
  - Followed by a linear scan through the accumulator (step 3).

Comparison

Hough Transform

•Advantages

- Can handle high percentage of outliers (>95%)
- Extracts groupings from clutter in linear time

•Disadvantages

- **Quantization issues**
- Only practical for **small number of dimensions** (up to 4)

•Improvements available

- Probabilistic Extensions
- Continuous Voting Space
- Can be generalized to arbitrary shapes and objects

RANSAC

•Advantages

- General method suited to large range of problems
- Easy to implement
- “Independent” of number of dimensions
- No accumulator needed, space-efficient, less prone to the choice of bin size

•Disadvantages

- Basic version only handles moderate number of outliers (<50%)
- **More hypotheses may need to be generated and tested** than those obtained by finding peaks in the accumulator array.

•Many variants available, e.g.

- PROSAC: Progressive RANSAC [Chum05]
- Preemptive RANSAC [Nister05]

Summary

- Finding correspondences in images is useful for
  - Image matching, panorama stitching
  - Object recognition
  - Image search

- Beyond local point matching
  - Semi-local relations
  - Global geometric relations:
    - Epipolar constraint
    - 3D constraint (when 3D model is available)
    - 2D tnfs: Similarity / Affine / Homography
  - Algorithms:
    - RANSAC
    - Hough transform

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

$$\mathbf{x} = \mathbf{P}\mathbf{X}$$

$$\mathbf{x}' = \mathbf{H}\mathbf{x}$$

Agenda: Instance-level recognition

1) Introduction to local features

2) Interest point detectors (e.g., Harris, scale invariance)

3) Comparison of patches (SSD, ZNCC on pixel values)

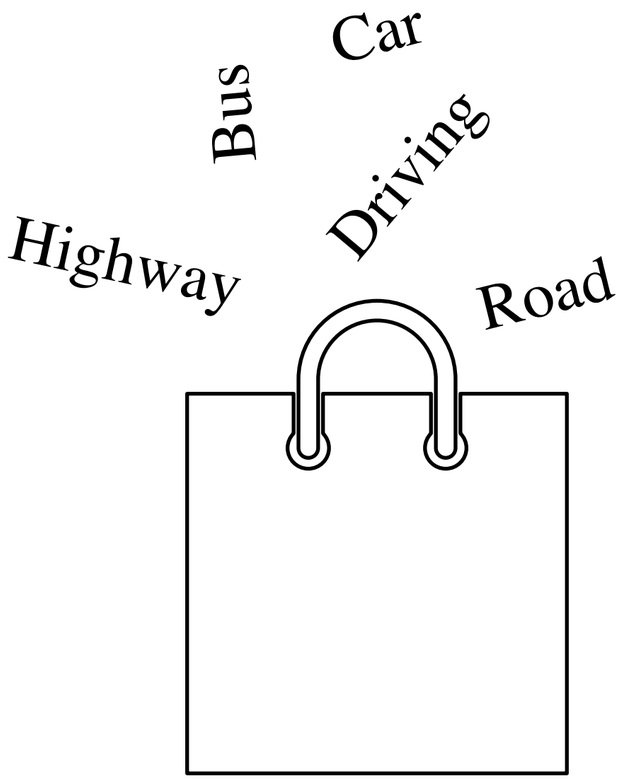
4) Feature descriptors (e.g., SIFT)

5) Matching and recognition with local features

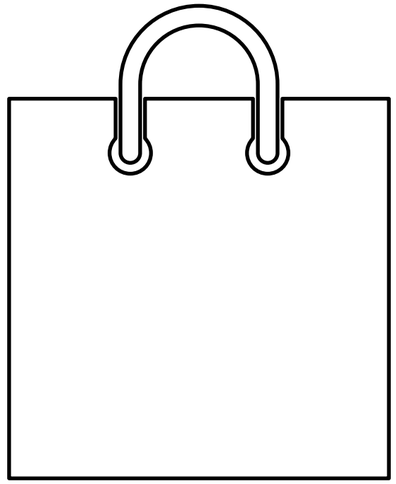
6) Local feature aggregation for a single image-level description

Need for aggregation

- Memory footprint of local features can be very high for one image.
- Example:
  - An image with 256 x 256 resolution (65536 pixels)
  - Densely extracted SIFT features from a grid of 32 x 32
  - 32 x 32 = 1024 features, each with 128-dimensions.
  - 1024 x 128 = 131072-dimensional image feature
  - Bigger than the original pixel dimensionality.



Bag of Words

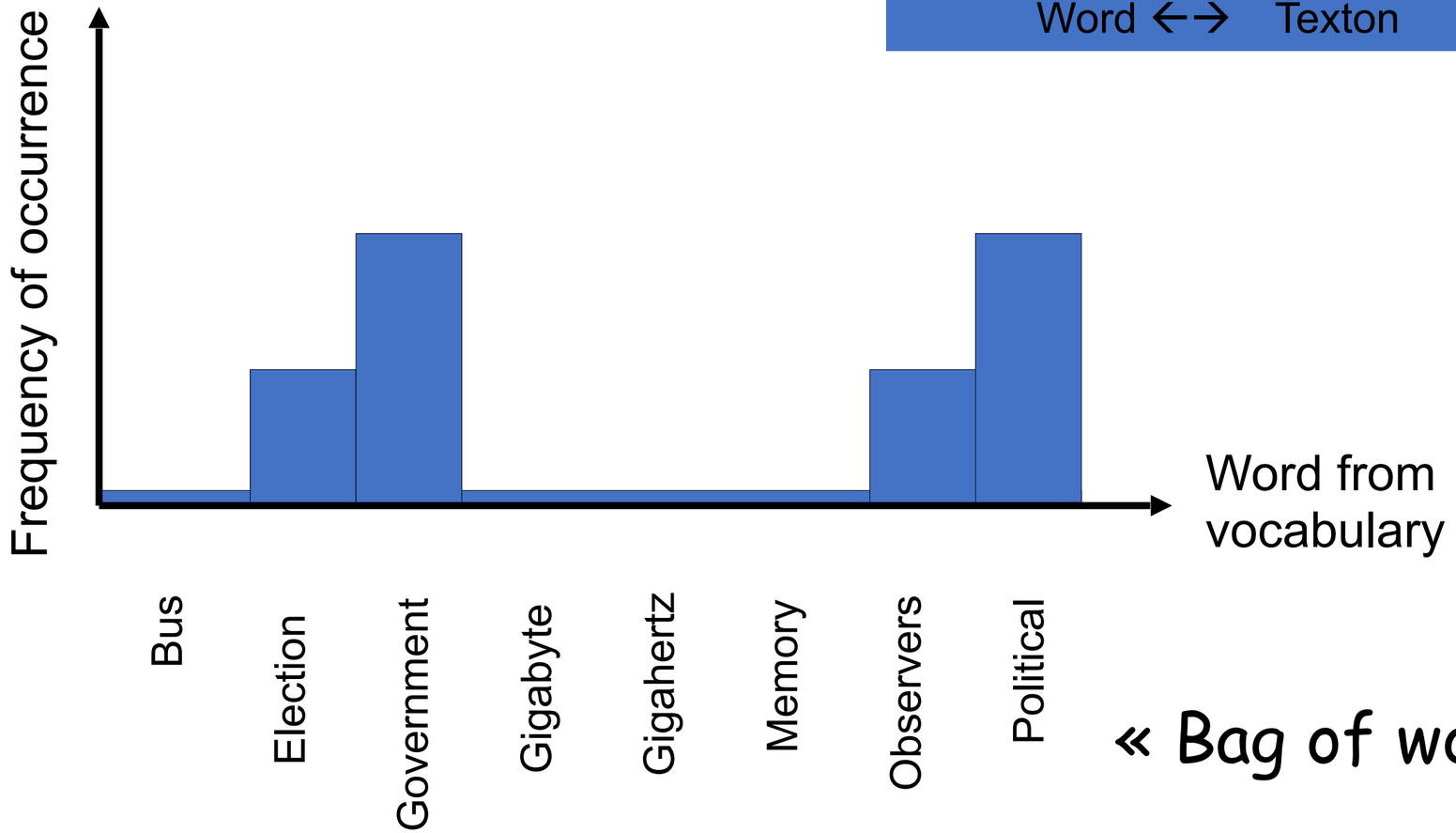


Bag of **Visual** Words

Analogy with Text Analysis

Political observers say that the government of Zorgia does not control the political situation. The government will not hold elections …

Analogy:
Text fragment ←→ Image region
Word ←→ Texton

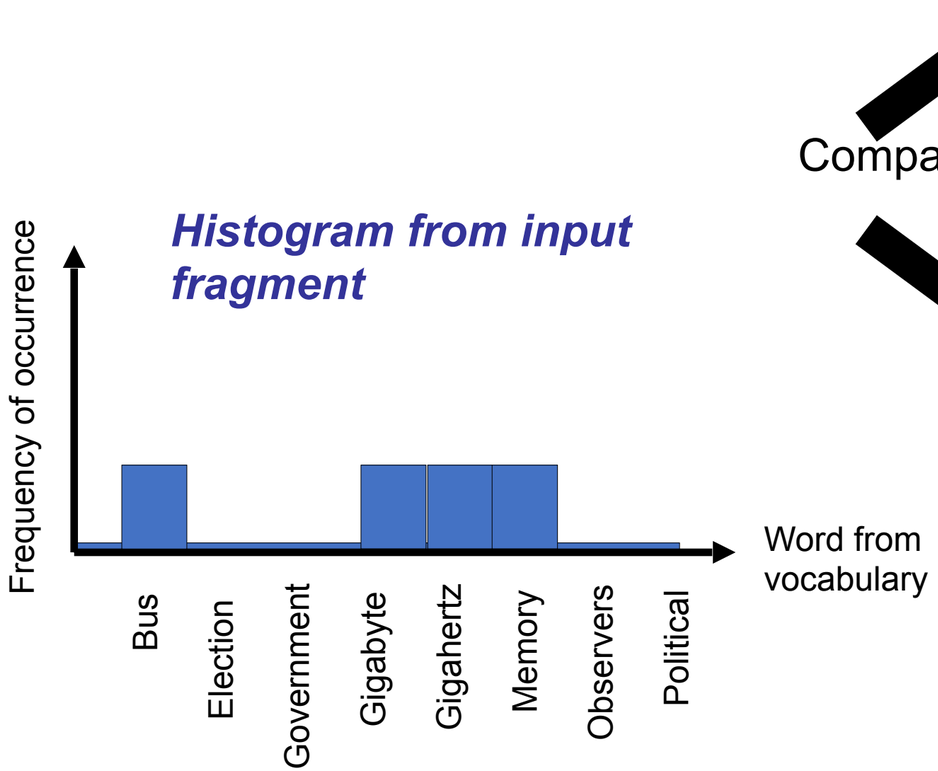


Why "bag"?

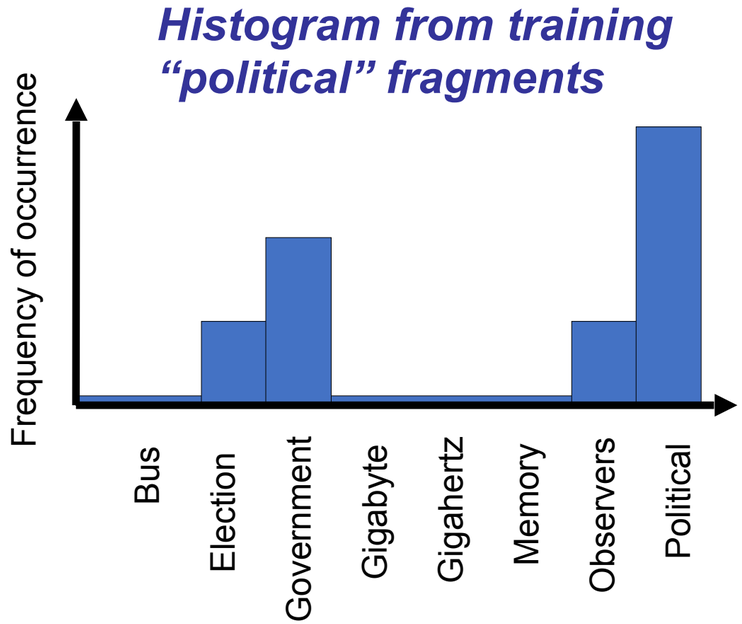
« Bag of words »

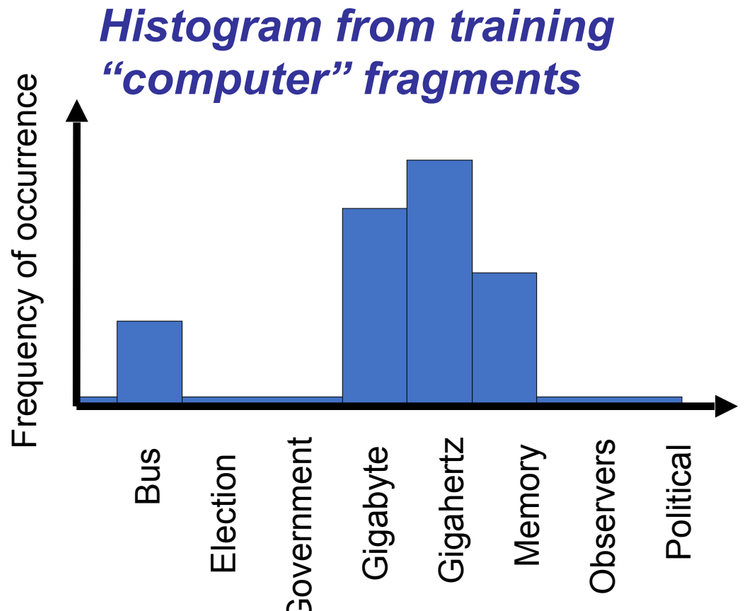
Analogy with Text Analysis

The ZH-20 unit is a 200Gigahertz processor with 2Gigabyte memory. Its strength is its bus and high-speed memory……

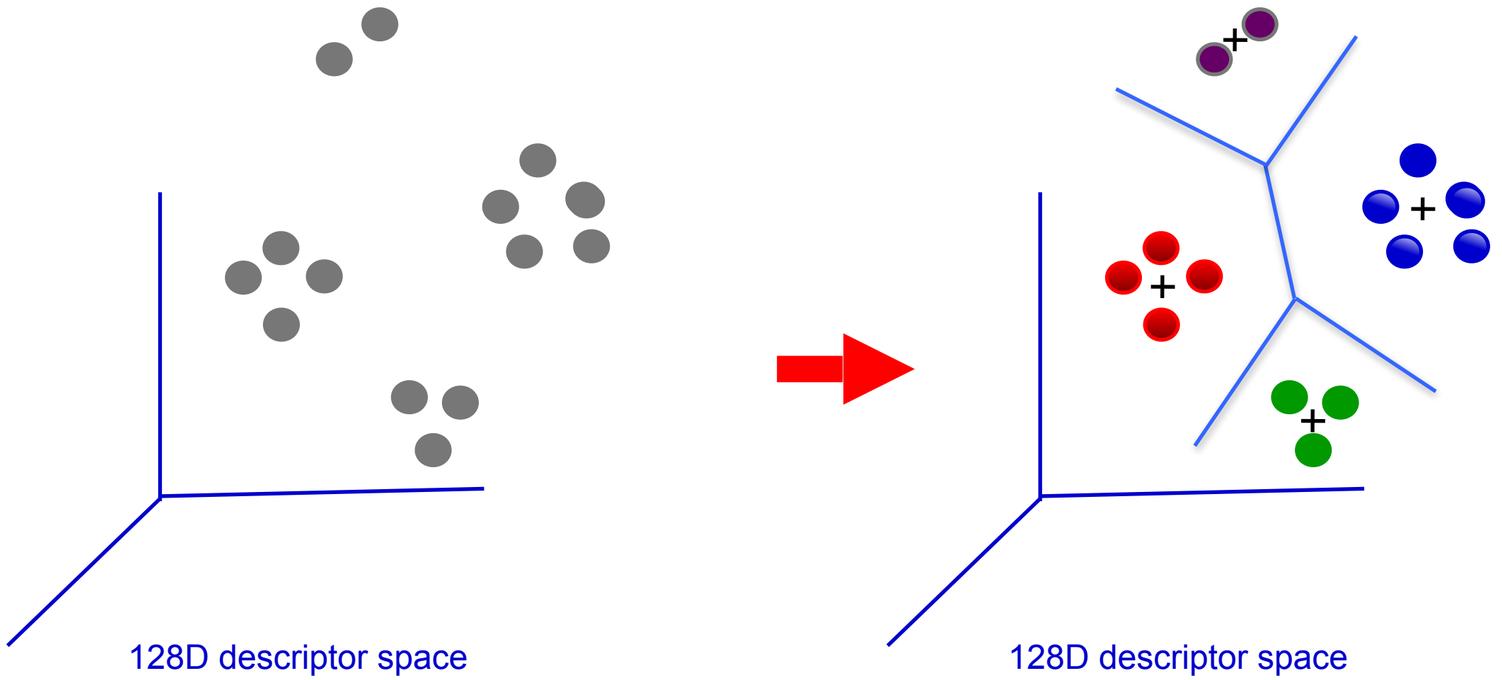


Compare





Build a visual vocabulary



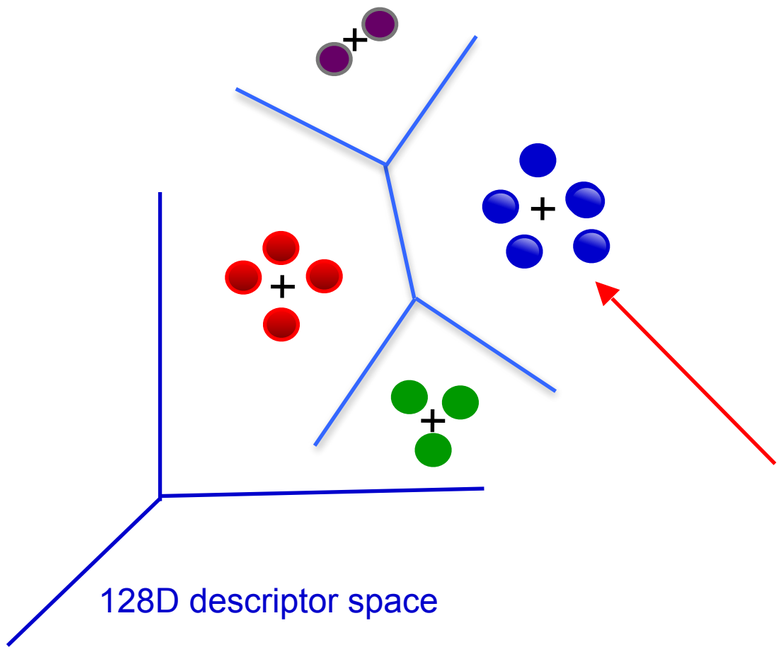
Vector quantize descriptors

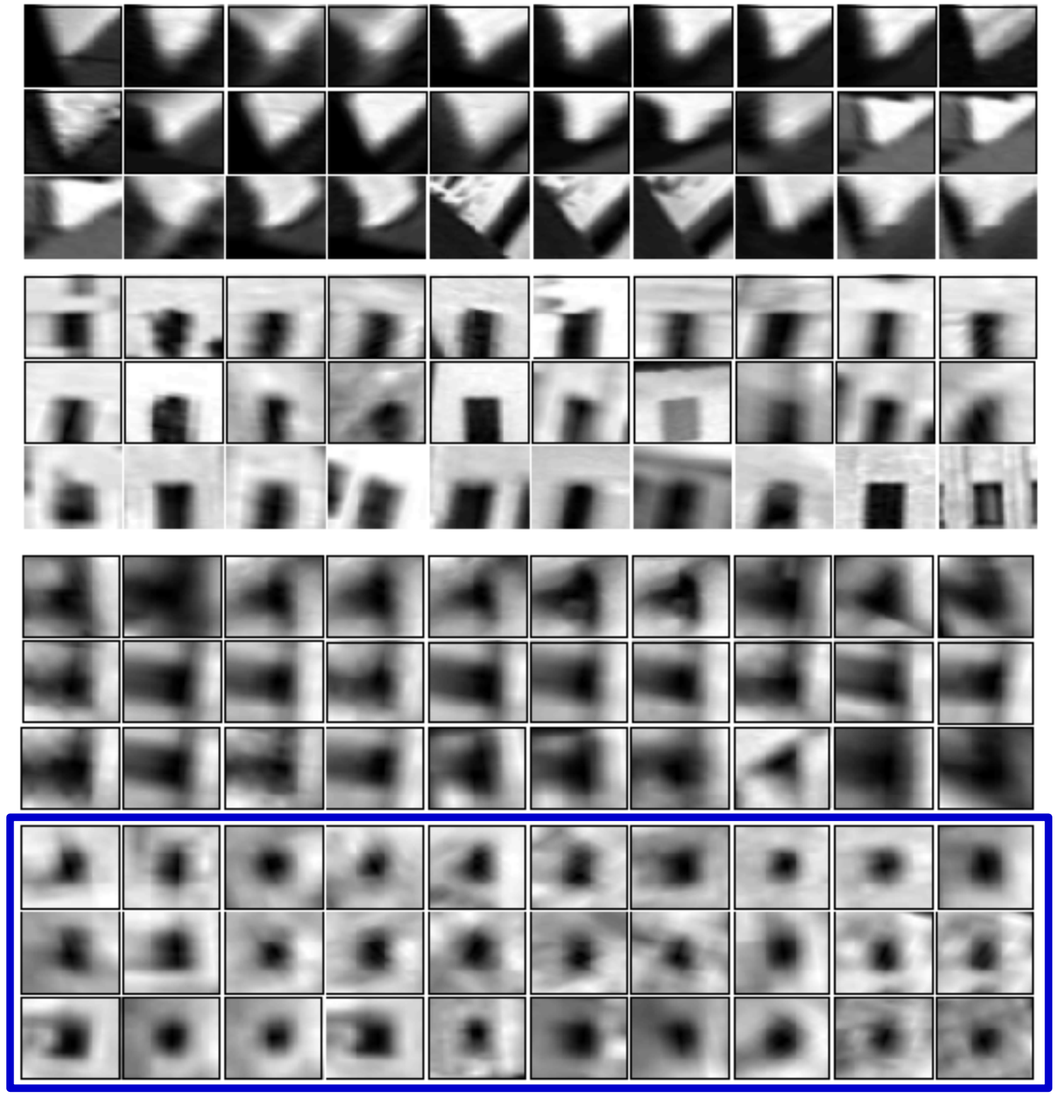
- Compute SIFT features from a subset of images
- K-means clustering (need to choose K)

[Sivic and Zisserman, ICCV 2003]

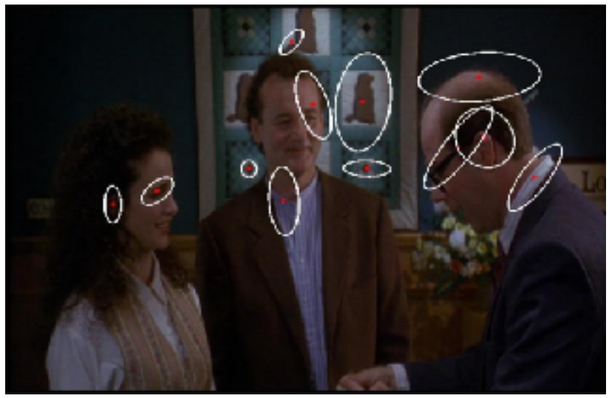
Visual words

Example: each group of patches belongs to the same visual word





Step 1: feature extraction



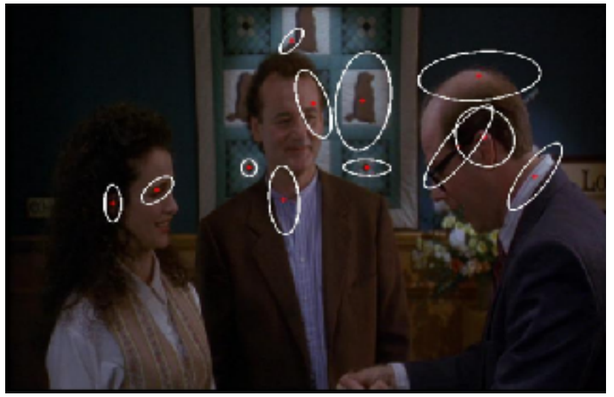
Sparse sampling

- SIFT as interest point detector

Dense sampling

- Interest points do not necessarily capture “all” features

Step 1: feature extraction

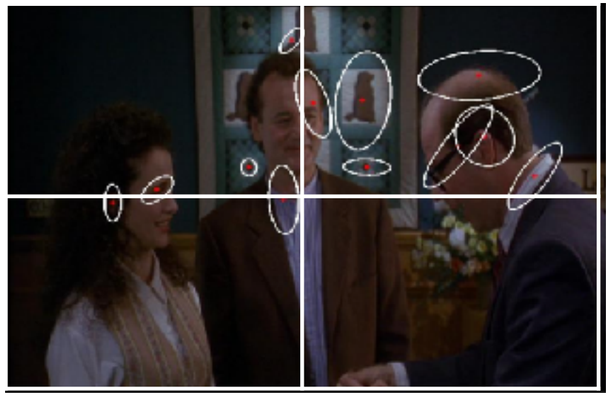


Sparse sampling

- SIFT as interest point detector

Dense sampling

- Interest points do not necessarily capture “all” features
- Spatial pyramid (Lazebnik, Schmid & Ponce, CVPR 2006)



Step 2: Quantization

Cluster descriptors

- K-means
- Gaussian mixture model

Assign each visual word to a cluster

- Hard or soft assignment

Build frequency histogram

Examples for visual words

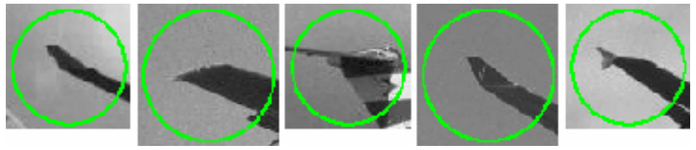
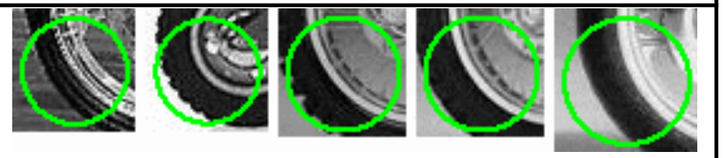
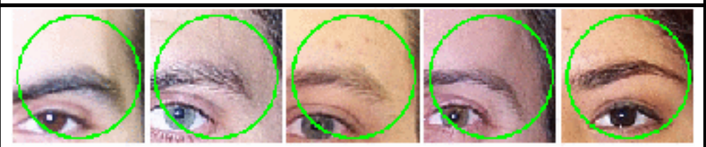
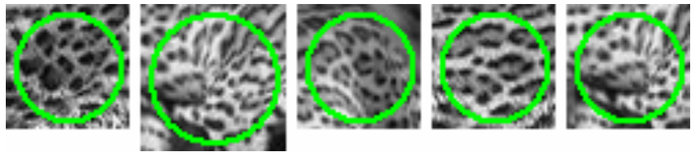
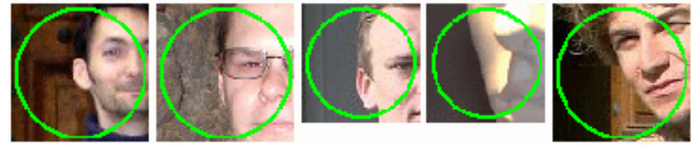
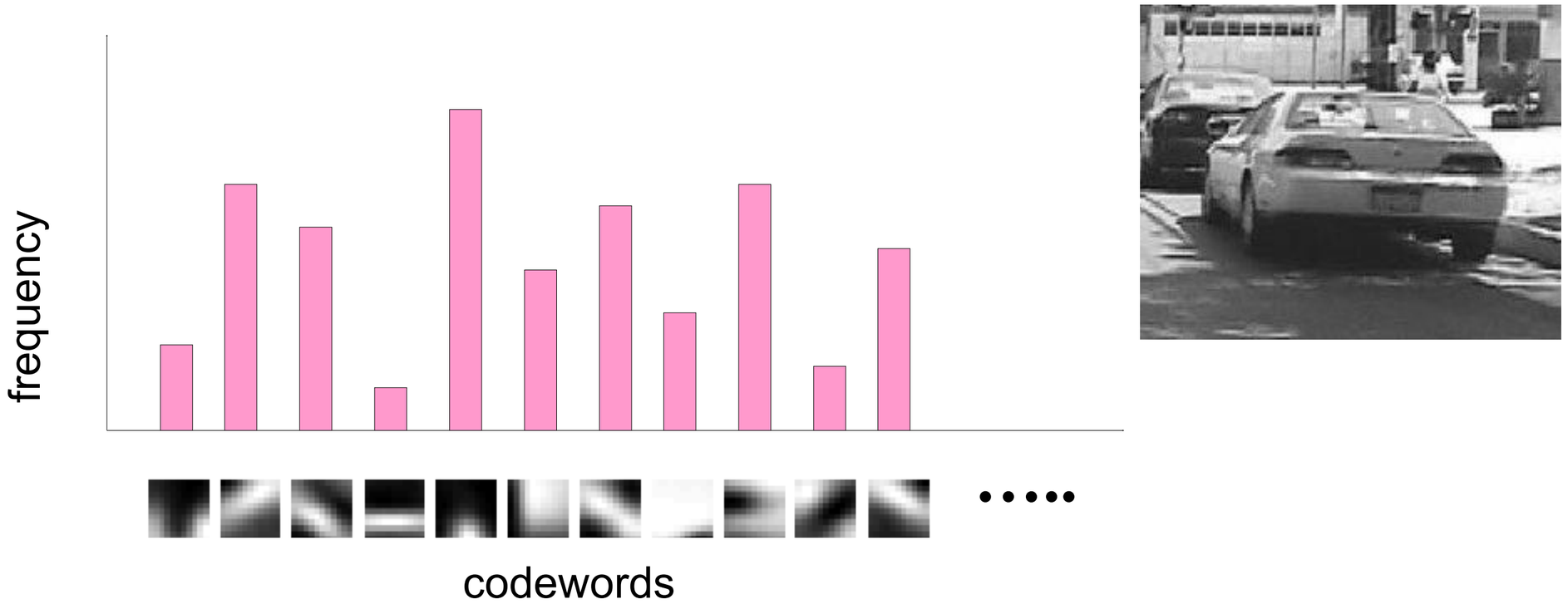
Airplanes		
Motorbikes		
Faces		
Wild Cats		
Leaves		
People		
Bikes		

Image representation



- Each image is represented by an aggregated histogram vector, typically 1000-4000 dimensional
- Normalized with L2 norm
- Fisher Vectors [Perronnin et al. ECCV’10]: improvements over Bag of Features

Agenda: Instance-level recognition

1) Introduction to local features

2) Interest point detectors (e.g., Harris, scale invariance)

3) Comparison of patches (SSD, ZNCC on pixel values)

4) Feature descriptors (e.g., SIFT)

5) Matching and recognition with local features

6) Local feature aggregation for a single image-level description