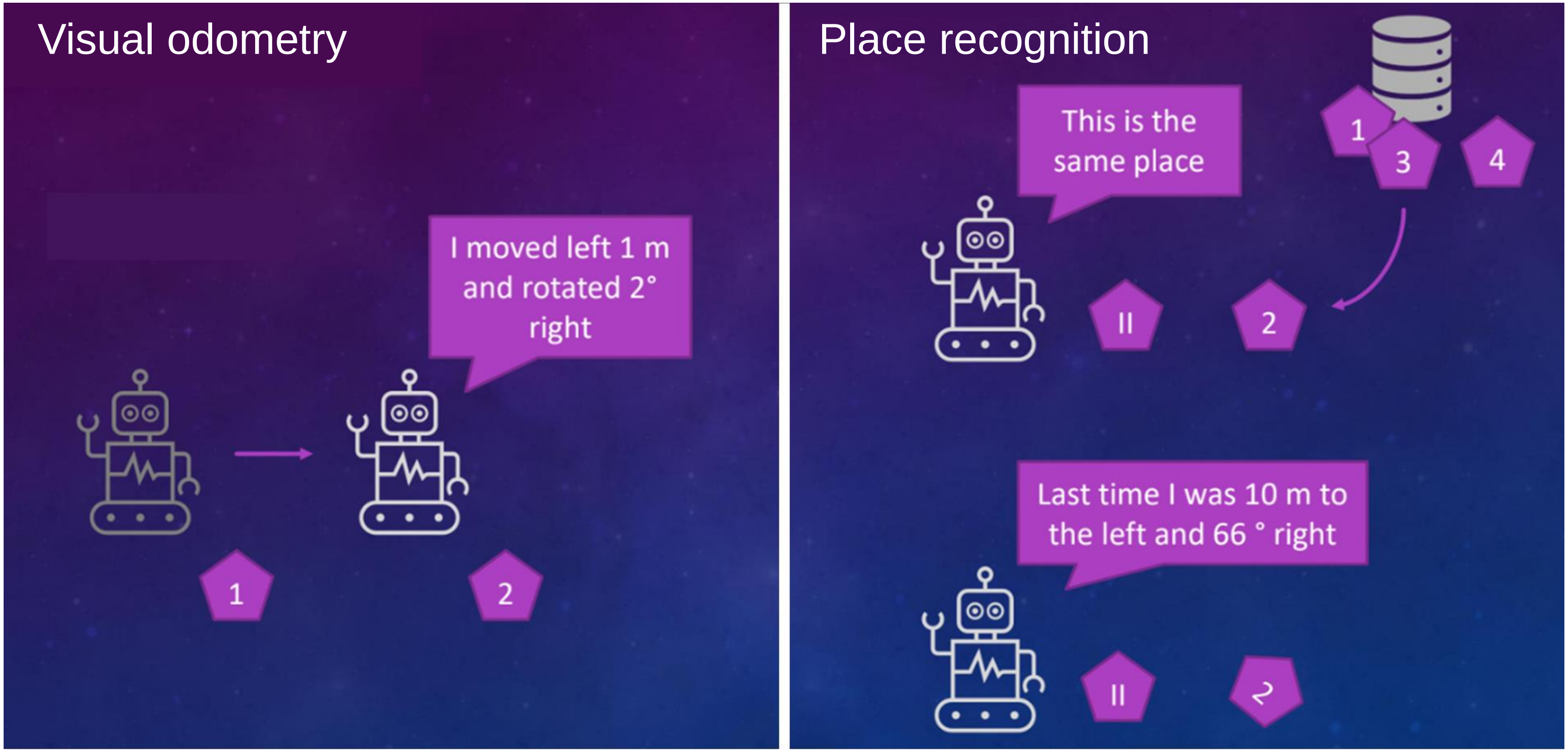
SPATIAL REASONING

VISION-BASED LOCALIZATION

Dr TIAN Jing
tianjing@nus.edu.sg

Vision-based localization



Reference: ECCV 2022 Tutorial, Self-Supervision on Wheels: Advances in Self-Supervised Learning from Autonomous Driving Data, https://gidariss.github.io/ssl-on-wheels-eccv2022/

Topics

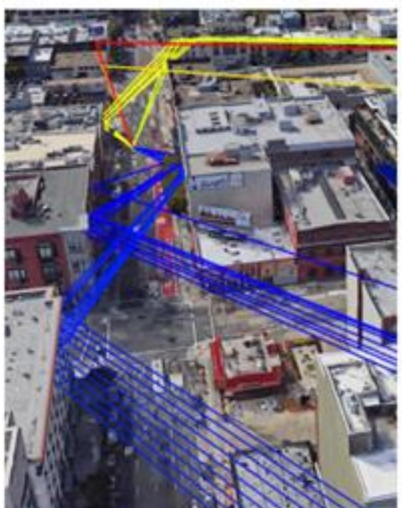
- Visual odometry
- Visual place recognition pipeline
  - Feature extraction
  - Feature encoding
  - Feature indexing
- Workshop on place recognition

Positioning solution

Objective of localization

- Refer to environment (where am I?), useful for navigation.
- Refer to machine itself (what is camera’s posture?), useful for display (e.g., *Augmented Reality* (AR)).

GPS
globally absolute
inaccurate



Wheel odometry
for robots
prone to drift

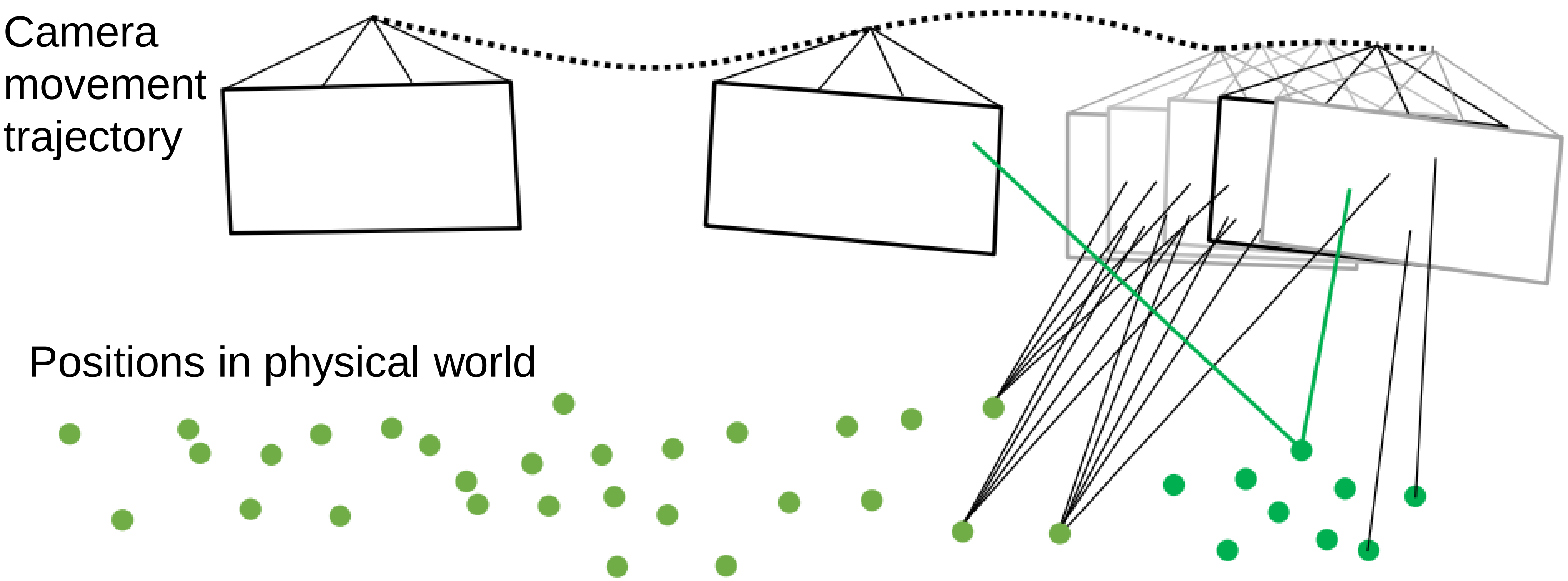
Vision
accurate
cameras are cheap

Positioning solution

- Vision data can be used as a complement to
  - Wheel odometry (might be affected by wheel slippage, such as on sand or wet floor)
  - GPS (might be degraded, expensive, or GPS-denied environments, such as underwater, aerial, or Mars)
- Vision-based solution is accurate. According to the leaderboard on KITTI dataset, the state-of-the-art methods can achieve a translational error of 0.5%-1% of the travelled distance.

Visual odometry

- Use case: A mobile robot (carrying cameras) moves around in the campus.
- Objective: Infer the camera movement parameters (also called *camera pose*, can be used to infer the robot movement) using the images captured by the cameras.
- Idea: The same point (in the physical world) can be seen in consecutive frames. However, it appears on the different positions in the images due to the camera movement when it captures consecutive frames.
- This technique is called visual odometry. It is the process of determining the position and orientation of a robot by analyzing the camera images.



Visual odometry

Assumptions/requirement of visual odometry

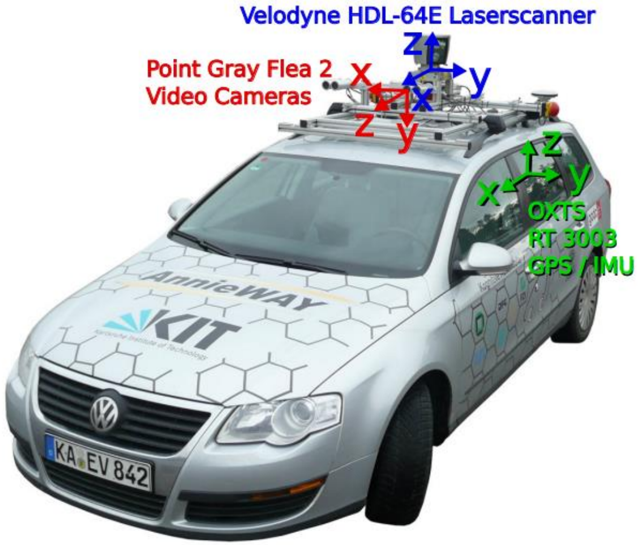
- Sufficient illumination in the environment.
- Sufficient texture/edge to allow distinct features extraction from the images.
- Distinct scene content (non repetitive).
- Sufficient scene overlap between consecutive frames.
- Dominance of static scene over moving objects in the environment.

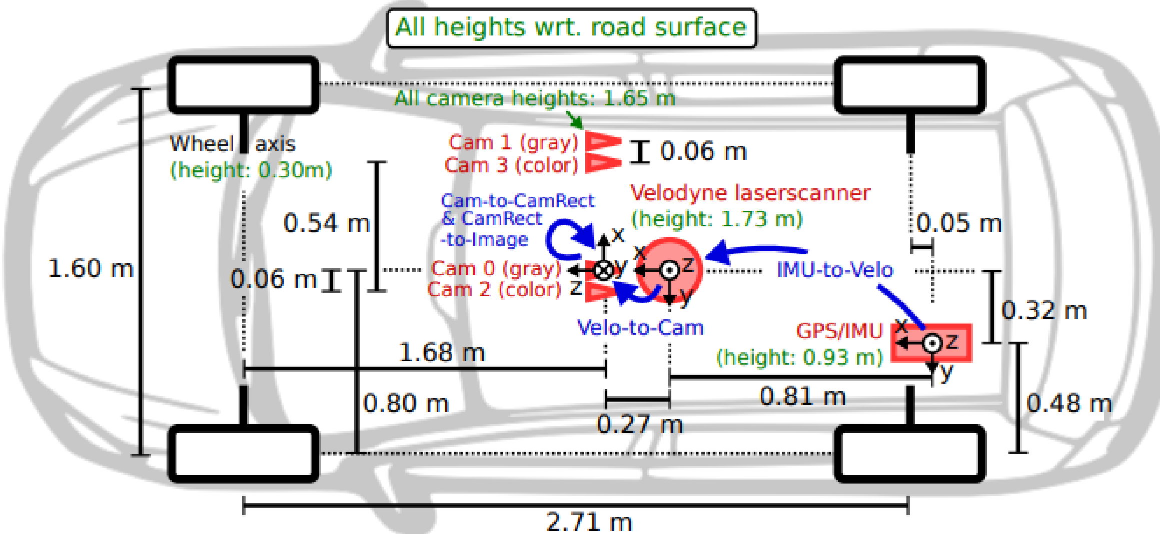


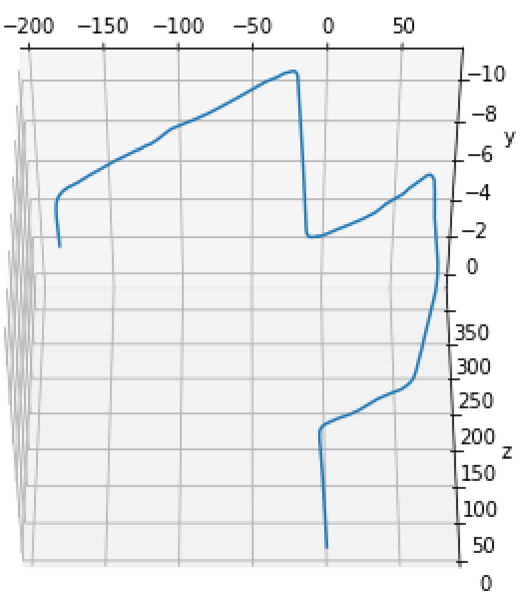


Photo: https://www.flickr.com/photos/_pavan_/24618687070;
https://www.huiacoustics.com/product/soundproof-room-divider;
https://lenniechua.com/2018/12/15/2018-singapore-thailand-drive-driving-in-heavy-rain/

KITTI Dataset







X direction	Right
Y direction	Up
Z direction	Forward

- sequences
  - 00
    - image_0
    - image_1
    - calib.txt
    - times.txt
- poses
  - 00.txt

Data acquisition
- Two stereo cameras + LiDAR
- 6 hours recording at 10 frames per second

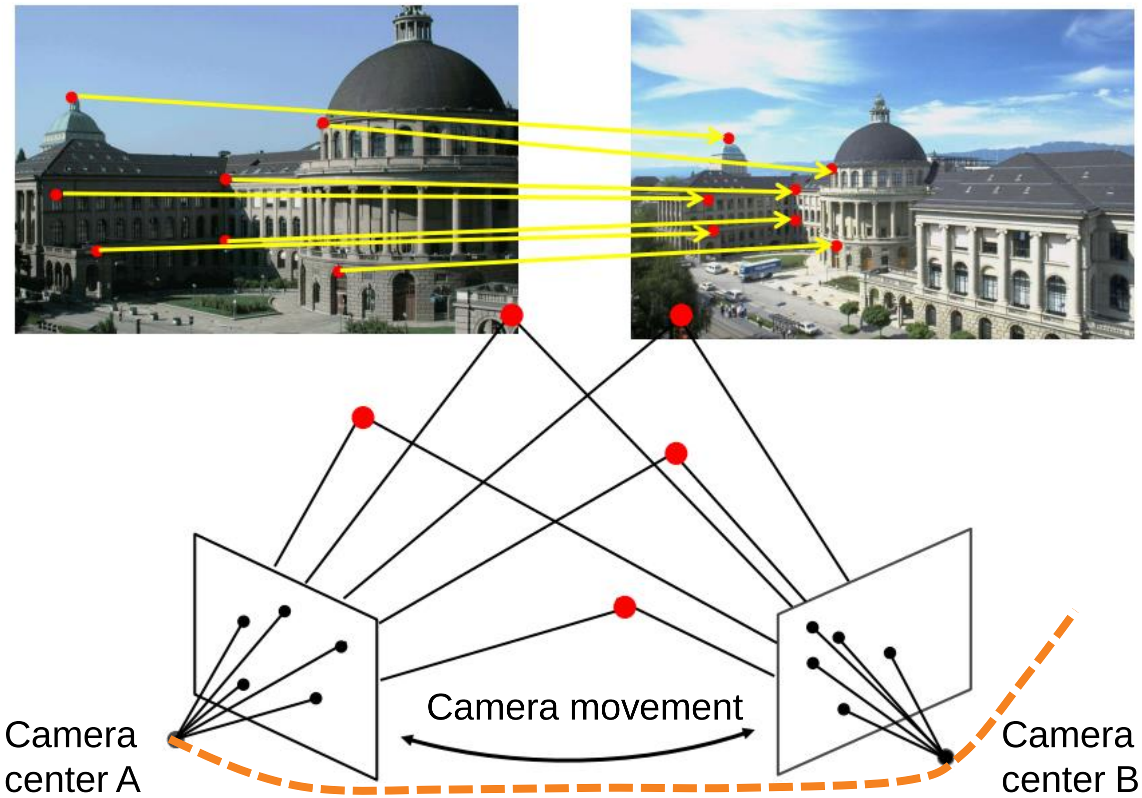
image_0, image_1: Stereo image sequence

calib.txt: Stereo camera calibration

times.txt: Timestamp

00.txt: Ground truth camera poses (used in performance evaluation)

Key idea



Q1. The relationship between two coordinates of the same point due to the coordinate reference is changed (camera movement).
→ Extrinsic matrix
Q2. The relationship between camera reference system to image reference system
→ Pinhole camera model
Q3. The relationship between image reference system to pixel reference system
→ intrinsic matrix

Coordinate system	Origin	Dimension	Unit
Camera reference system	Camera center point	3D	Physical (meter)
Image reference system	Center point of image plane (*charge-coupled device* (CCD))	2D	Physical (meter)
Pixel reference system	Top left point of the image	2D	Digital (pixel)

Extrinsic matrix

From the same point in the physical world

Homogeneous coordinate
(studied in last day's class)

Translation only

$$\begin{bmatrix} X_{c1} \\ Y_{c1} \\ Z_{c1} \\ 1 \end{bmatrix} = \begin{bmatrix} \underbrace{\begin{matrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{matrix}}_{\mathbf{T}} \end{bmatrix} \times \begin{bmatrix} X_{c2} \\ Y_{c2} \\ Z_{c2} \\ 1 \end{bmatrix}$$

Coordinate
(with respect to
camera center A)

Coordinate
(with respect to
camera center B)

Rotation only: Around z axis

$$\begin{bmatrix} X_{c1} \\ Y_{c1} \\ Z_{c1} \\ 1 \end{bmatrix} = \begin{bmatrix} \underbrace{\begin{matrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}}_{\mathbf{R}_z} \end{bmatrix} \times \begin{bmatrix} X_{c2} \\ Y_{c2} \\ Z_{c2} \\ 1 \end{bmatrix}$$

Translation + rotation

$$\begin{bmatrix} X_{c1} \\ Y_{c1} \\ Z_{c1} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_{c2} \\ Y_{c2} \\ Z_{c2} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & t_x \\ \sin\theta & \cos\theta & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} X_{c2} \\ Y_{c2} \\ Z_{c2} \\ 1 \end{bmatrix}$$

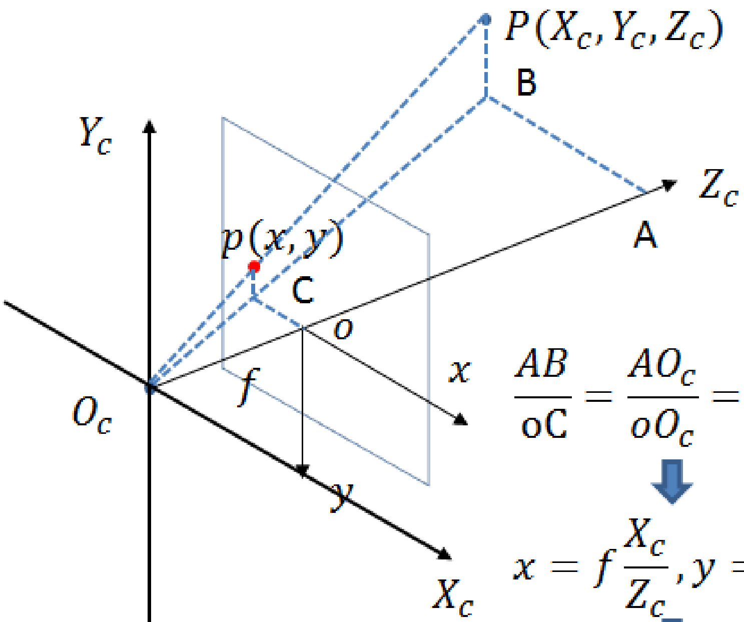
Extrinsic matrix

Pinhole camera model

Pinhole camera model: Convert from camera reference system $P(X_c, Y_c, Z_c)$ to the image reference system $P(x, y)$. It depends on camera model focal length f.

Convert from camera reference system $\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$ To be image reference system $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f/z_c & 0 & 0 & 0 \\ 0 & f/z_c & 0 & 0 \\ 0 & 0 & 1/z_c & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$



$$\frac{AB}{oC} = \frac{AO_c}{oO_c} = \frac{PB}{pC} = \frac{X_c}{x} = \frac{Z_c}{f} = \frac{Y_c}{y}$$

Triangle similarity theorem

$$x = f\frac{X_c}{Z_c}, y = f\frac{Y_c}{Z_c}$$

Re-arrange as matrix format

$$Z_c \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix}$$

$\Delta ABO_c \sim \Delta oCO_c$

$\Delta PBO_c \sim \Delta pCO_c$

Reference: Module 2, Vision Algorithms for Mobile Robotics, http://rpg.ifi.uzh.ch/teaching.html

Intrinsic matrix

Suppose for the CMOS/CCD sensor, each pixel has a physical size d_x, d_y, the image plane origin is located at the position $(u_0, v_0, 1)$, then $u = \frac{x}{d_x} + u_0, v = \frac{y}{d_y} + v_0$

Convert from image reference system $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ To be pixel reference system $\begin{bmatrix} u \\ v \\ 1 \end{bmatrix}$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/d_x & 0 & u_0 \\ 0 & 1/d_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} 1/d_x & 0 & u_0 \\ 0 & 1/d_y & v_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f/z_c & 0 & 0 & 0 \\ 0 & f/z_c & 0 & 0 \\ 0 & 0 & 1/z_c & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

$$z_c \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f/d_x & 0 & u_0 & 0 \\ 0 & f/d_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix}$$

$$K = \begin{bmatrix} \alpha_x & \gamma & u_0 & 0 \\ 0 & \alpha_y & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

- α_x, α_y focal length in pixels
- γ skew between x and y axes (often zero)
- u_0, v_0 principal point (typically center of image)

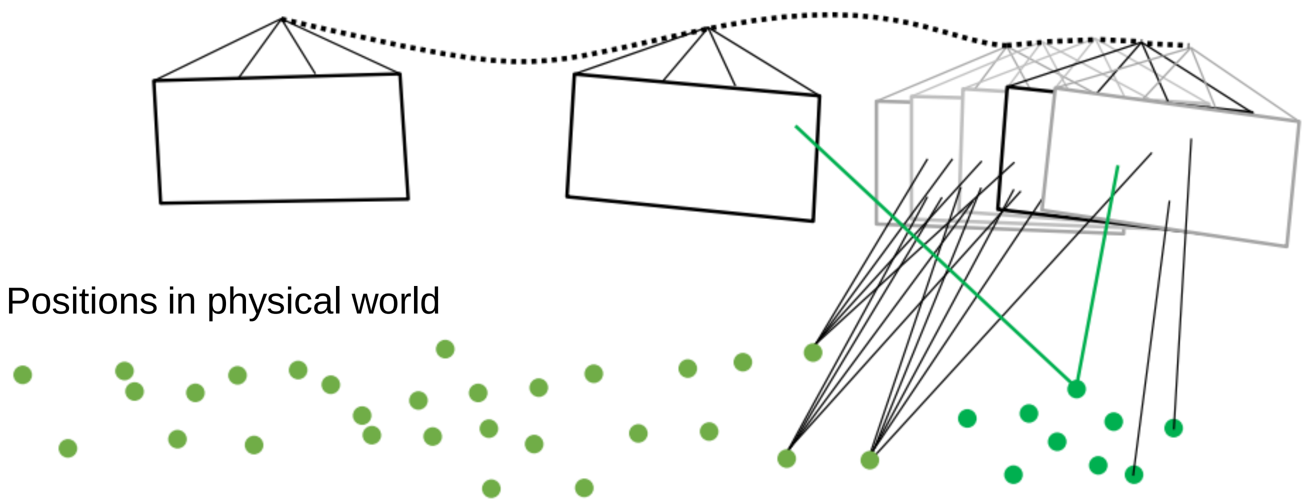
Example: Given an image resolution of 640×480 pixels and a focal length of 210 pixels, the intrinsic matrix could be

$$K = \begin{bmatrix} 210 & 0 & 320 & 0 \\ 0 & 210 & 240 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

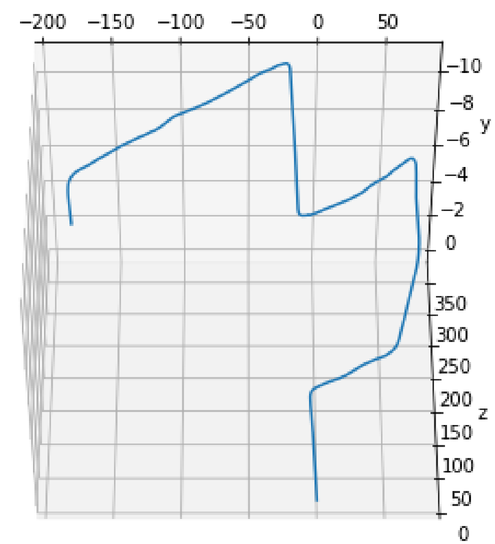
Reference: https://www.mathworks.com/help/vision/ug/camera-calibration.html

Summary of visual odometry

- Step 1: Given two consecutive frames, find the pair of matched points.
- Step 2: For each point, apply its coordinate (in terms of <u>pixels</u>), pinhole camera model (the depth is estimated using two images captured by the stereo cameras), and the intrinsic matrix (given by manufacture or calibrated offline) to obtain its coordinate (in terms of <u>physical world</u>).
- Step 3: Given the pair of matched points (and their coordinates in terms of <u>physical world</u>), estimate the extrinsic matrix to obtain the camera motion (movement of the car/human carrying the camera).



Estimated camera movement trajectory



Topics

- Visual odometry
- Visual place recognition pipeline
  - Feature extraction
  - Feature encoding
  - Feature indexing
- Workshop on place recognition

Visual place recognition: Motivation

Global localization problem

- Perform localization via the pre-collected gallery (pre-collected images about the places with location annotation, such as GPS tags), instead of using the starting frame as the coordinate reference frame only.

Loop closing problem (in visual odometry)

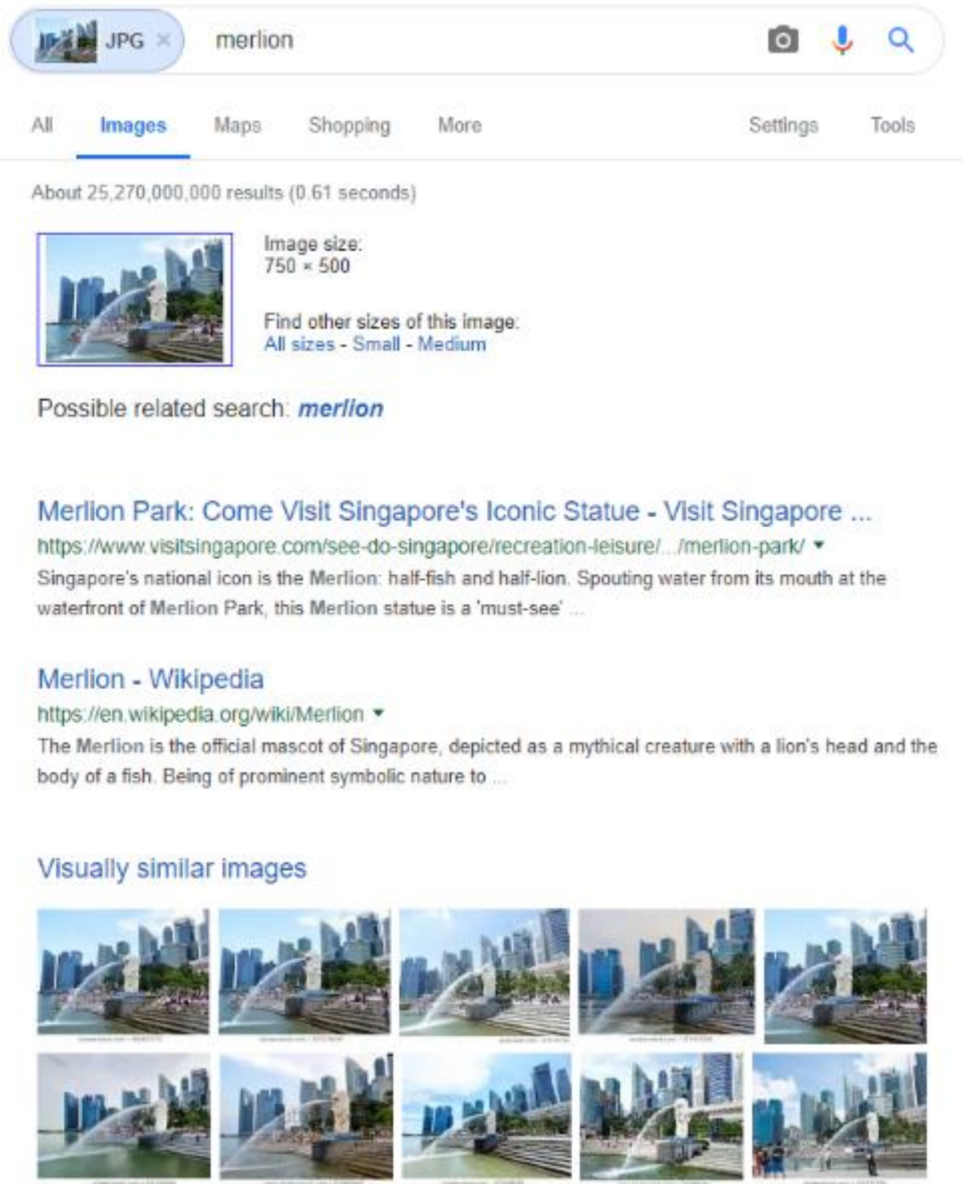
- When you go back to a previously visited area.
- Loop closure detection to avoid duplication (e.g., a cleaning robot).
- Loop correction to compensate the accumulated camera pose error drift (re-localization).

Our idea is to use Place Recognition, which matches the input photo to a set of photos in the gallery to recognize the place/location of the input photo.

Visual place recognition: Motivation

Image-based location and place recognition

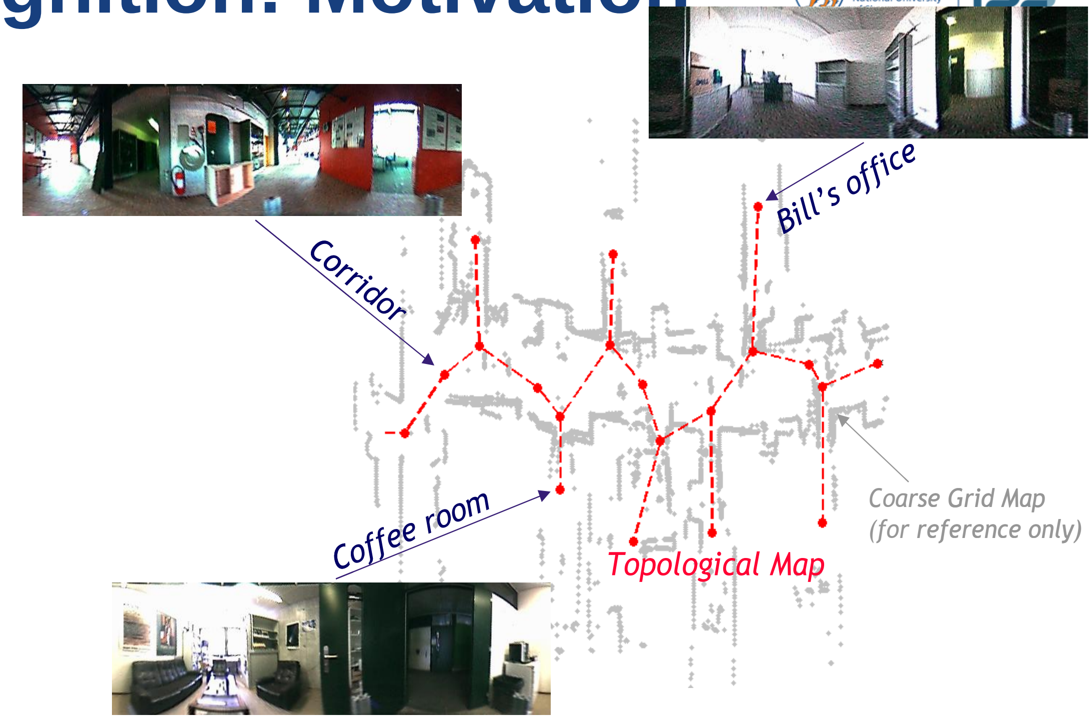
- **Image retrieval**: Have I seen this image before? Which images in my database look similar to it?
- Example: Google Reverse Image Search



Visual place recognition: Motivation

Image-based location and place recognition

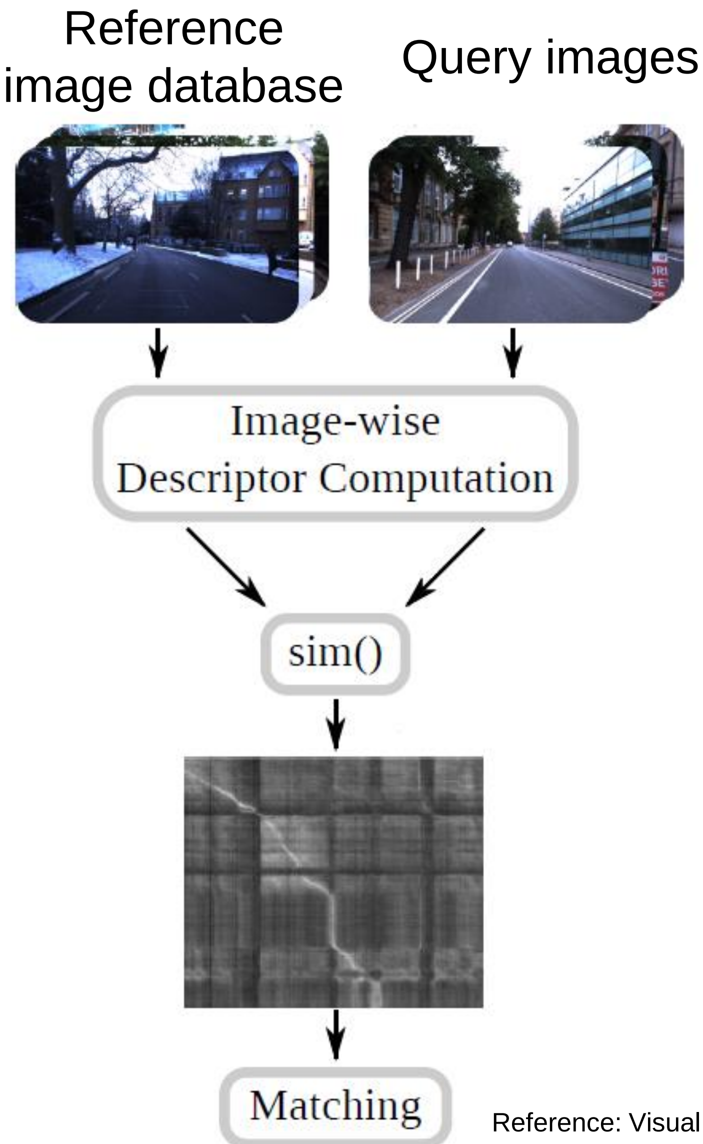
- **Robotics**: Has the robot been to this place before? Which images were taken around the same location?
- Example: SLAM (simultaneous localization and mapping), which is the backbone of spatial awareness of a robot.



- A map is necessary for localizing the robot
  - Pure localization with a known map
  - SLAM: no a priori knowledge of the robot's workspace
- An accurate camera pose estimate is necessary for building a map of the environment
  - Mapping with known robot poses
  - SLAM: the robot poses have to be estimated along the way

Source: Cornelia Fermüller, Path planning, CMSC498F, CMSC828K (Spring 2016), Robotics and Perception, http://users.umiacs.umd.edu/~fer/cmsc498F-828K/cmsc-498F-828K.htm

Visual place recognition: Intuition



- Compute a descriptor for each image.
- Measure the similarity between database and query descriptors.
- Result is a pairwise descriptor similarity matrix, which is the basis for matching decisions between the reference image database and query image set.

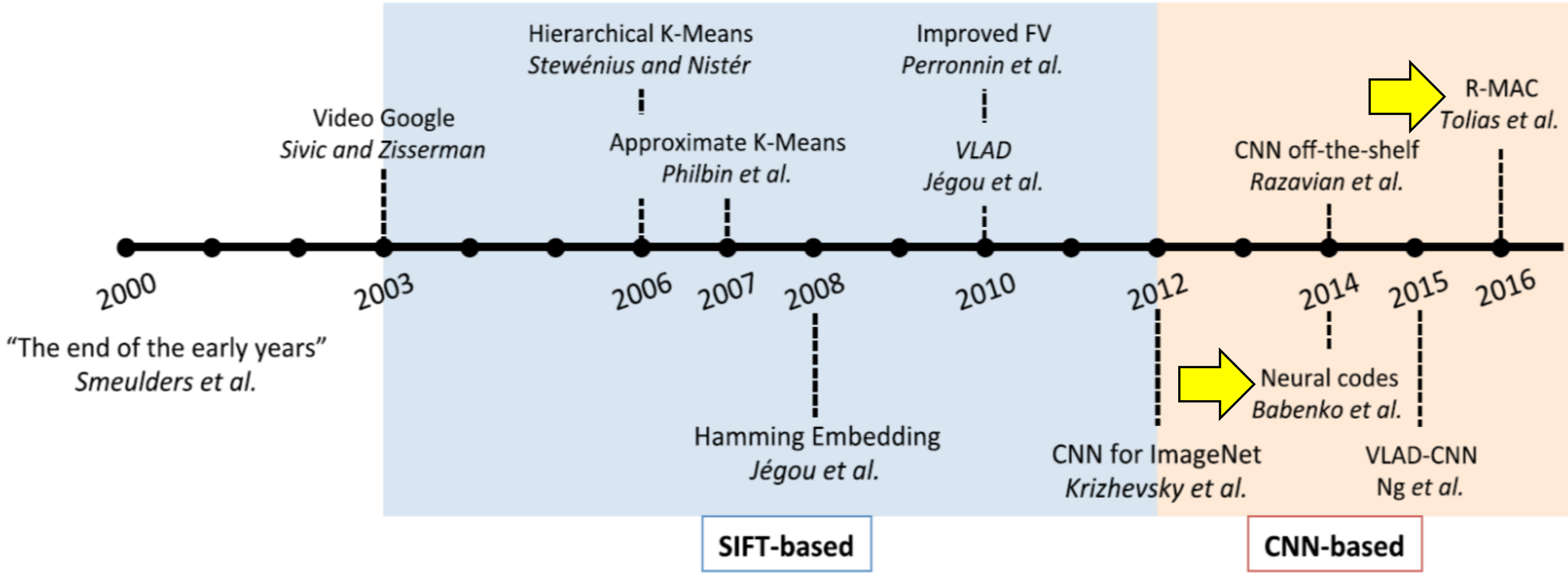
Reference: Visual Place Recognition: A Tutorial. IEEE Robotics & Automation Magazine 2023, pp. 2-16.

Visual place recognition: Challenge

- Lighting changes: Different time of day
- Changes in camera viewpoint
- Occlusions and ambiguous objects: People, cars, trees.

Reference: N. Piasco, et al., A survey on Visual-Based Localization: On the benefit of heterogeneous data, Pattern Recognition, 2018, pp. 90-109.

Visual place recognition: Literature



Milestones: After a survey of methods before the year 2000 [1], Video Google was proposed in 2003 [2], marking the beginning of the BoW model [3]. Although SIFT-based methods were still moving forward, CNN-based methods began to gradually take over, such as the fine-tuned CNN model for generic instance retrieval [4, 5].

Reference: L. Zheng, Y. Yang, Q. Tian, "SIFT Meets CNN: A Decade Survey of Instance Retrieval," IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 40, No. 5, May 2018, pp. 1224-1244.

[1] A. W. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," IEEE Trans. Pattern Anal. Mach. Intell., vol. 22, no. 12, pp. 1349-1380, Dec. 2000.

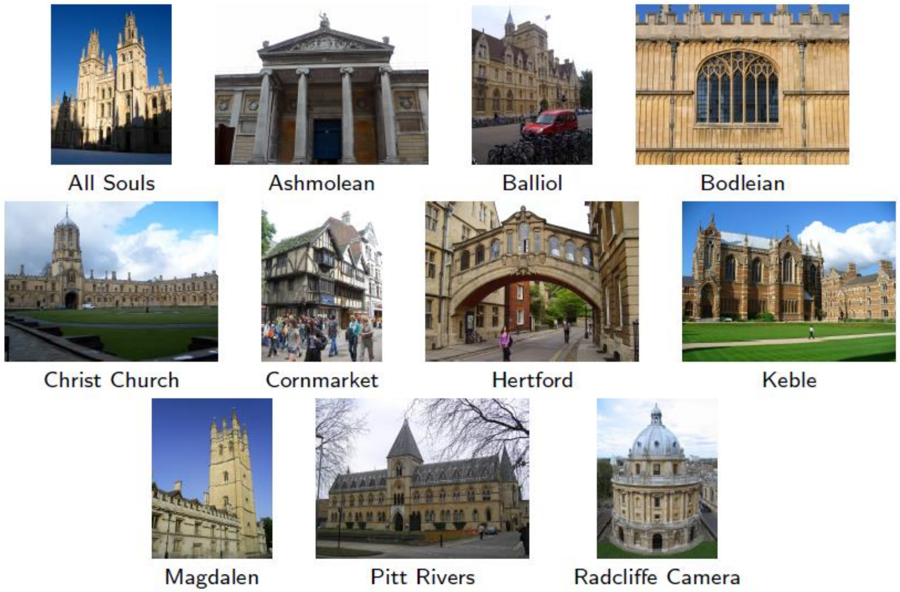
[2] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," ICCV 2003.

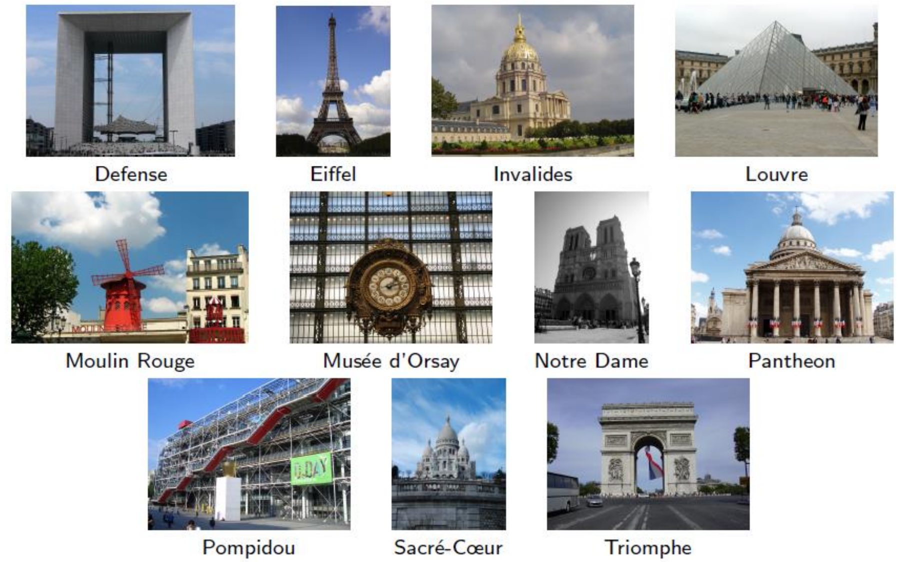
[3] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," CVPR 2010.

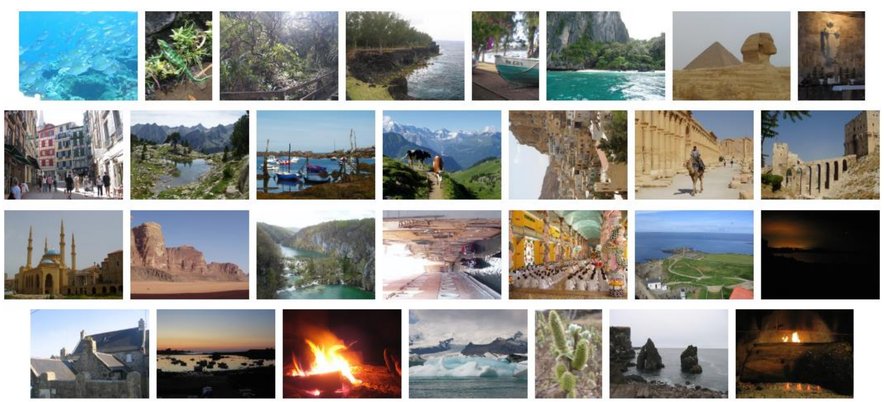
[4] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," ECCV 2014.

[5] G. Tolias, R. Sicre, and H. Jegou, "Particular object retrieval with integral max-pooling of CNN activations," ICLR 2016.

Visual place recognition: Major dataset







Dataset	# image	# query	Content
Oxford5k	5,062	55	Buildings
Paris6k	6,412	55	Buildings
Holidays	1,491	500	Scene

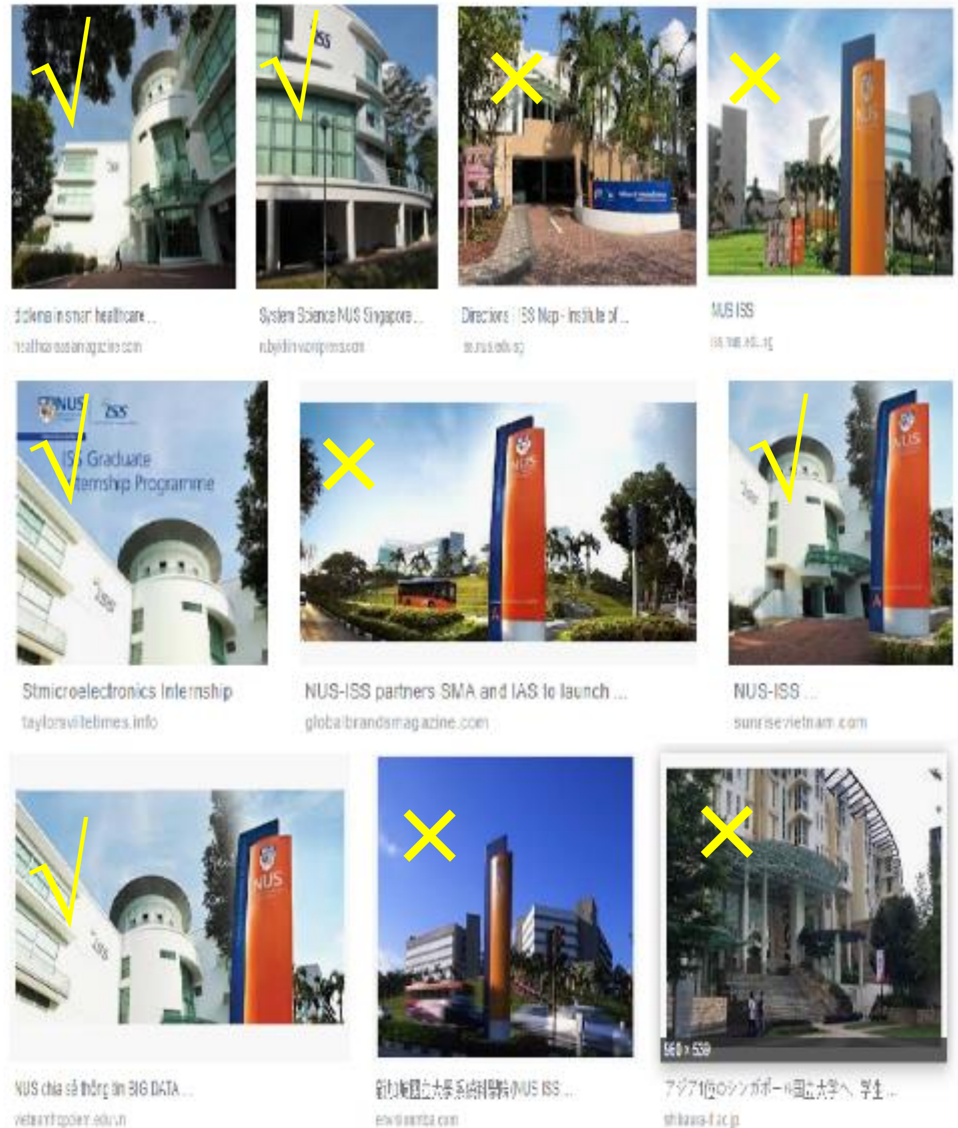
Reference:
- J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, “Object retrieval with large vocabularies and fast spatial matching,” CVPR 2017.
- H. Jegou, M. Douze, C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” ECCV 2008.
- J. Philbin, O. Chum, M. Isard, J. Sivic, A. Zisserman, “Lost in quantization: Improving particular object retrieval in large scale image databases,” CVPR 2008.

Performance metric

Query image (single input)

- How to evaluate the system performance based on <u>this single query</u>?
- How to evaluate the system performance based on <u>multiple queries</u>?

Returned results (ranked) from the gallery

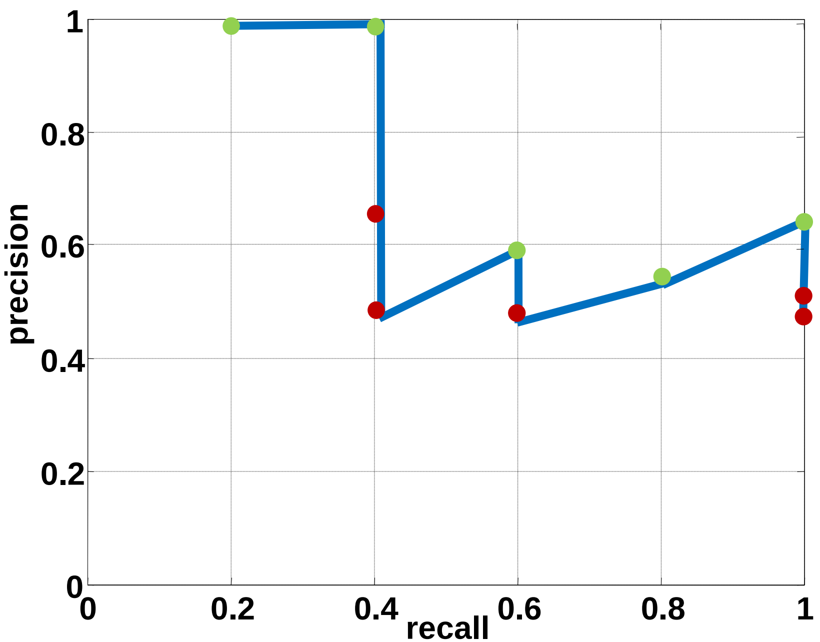


Performance metric

Ranked list of returned results with True/False labels (in previous slide example).

K	1	2	3	4	5	6	7	8	9	10
Label	T	T	F	F	T	F	T	T	F	F
TP	1	2	2	2	3	3	4	5	5	5
P	1	1	2/3	2/4	3/5	3/6	4/7	5/8	5/9	5/10
GTP	Supposed to be 5 for this query image. It depends on dataset.									

Precision = #relevant / #returned
Recall = #relevant / #total relevant



- K: current rank
- TP: true positives
- P: precision $= {}^{TP}/_{K}$
- GTP: total number of ground truth positives in the dataset

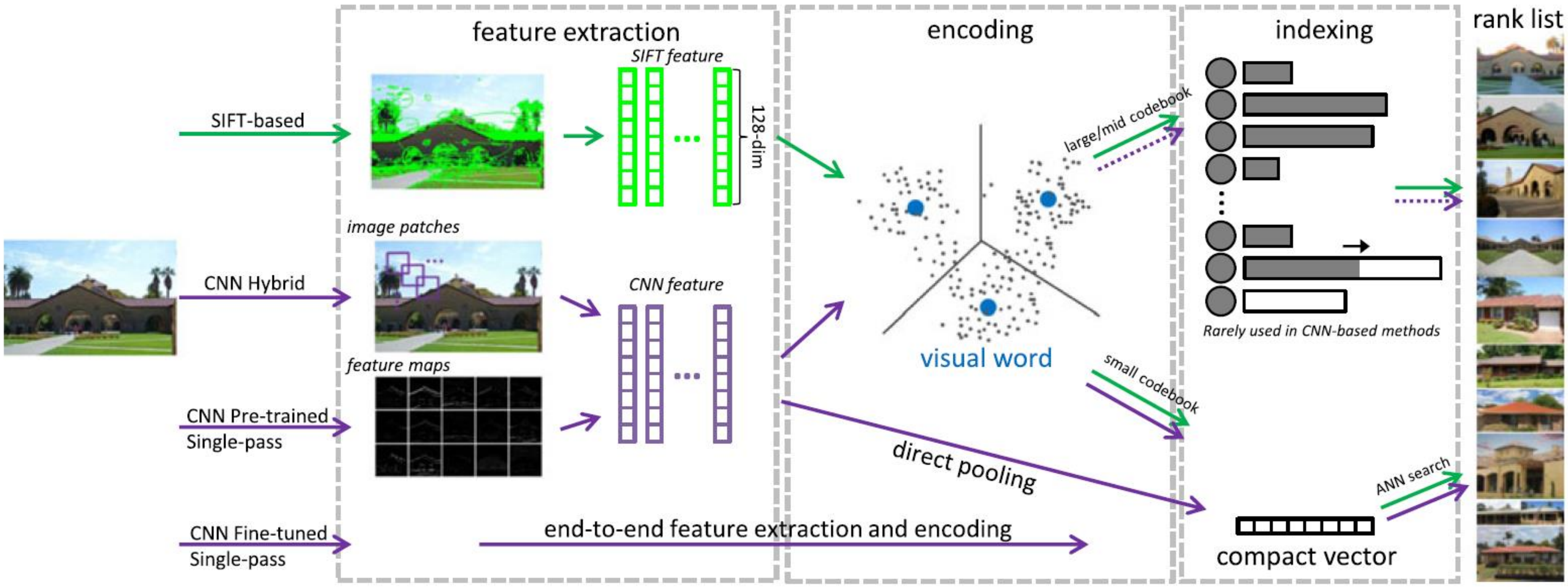
Summation of precision values of correct results / GTP

$$\frac{\left(1 + 1 + \frac{3}{5} + \frac{4}{7} + \frac{5}{8}\right)}{5}$$

- Average precision = average precision (for a single query)
- Mean average precision (mAP) = mean of average precision over all queries

Visual place recognition pipeline (1)

Feature extraction → Feature encoding (dictionary) → Feature indexing (search)



Reference: L. Zheng, Y. Yang, Q. Tian, SIFT Meets CNN: A Decade Survey of Instance Retrieval, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 40, No. 5, May 2018, pp. 1224-1244.

Feature extraction

<table>
<tr><th colspan="4">Features</th><th>Remark</th></tr>
<tr><td rowspan="4">Hand-crafted</td><td>Global</td><td>Image feature</td><td>Color histogram</td><td>Vision Systems course</td></tr>
<tr><td rowspan="3">Local</td><td>Patch feature</td><td>LBP, HoG</td><td>Vision Systems course</td></tr>
<tr><td rowspan="3">Point-based patch feature</td><td>SIFT</td><td>Previous day course</td></tr>
<tr><td>ORB</td><td>Following slides</td></tr>
<tr><td rowspan="3">Learned (for the purpose of place recognition)</td><td>Local</td><td>LIFT, SuperPoint</td><td>Following slides</td></tr>
<tr><td rowspan="2">Global</td><td colspan="2">Pre-trained (off the shelf) CNN</td><td>Following slides</td></tr>
<tr><td colspan="2">Tuned/re-trained CNN</td><td>Following slides</td></tr>
</table>

- Global features can help coarse place recognition (e.g., ISS building entrance).
- Local features can help fine place recognition (e.g.,. Facing entrance door).

ORB: Oriented FAST and rotated BRIEF

FAST (Features from accelerated segment test)

- Objective: Determine a pixel p (intensity value I_p) in the image as an interest point or not based on its neighboring pixels (say a circle of 16 pixels).
- Determine the pixel p is a keypoint, if there exists a set of n continuous pixels in the circle (of 16 pixels) which are all brighter than $I_p + t$, or all darker than $I_p - t$, with an appropriate threshold value t.
- Faster version: First compare the intensity of pixels 1, 5, 9 and 13 of the circle with I_p. At least three of these four pixels should satisfy the threshold criterion so that the interest point will exist.
  - If at least three of the four-pixel values I_1, I_5, I_9, I_{13} are not above or below $I_p + t$, then p is not an interest point (corner). In this case reject the pixel p as a possible interest point.
  - Else: check all 16 pixels and check if 12 contiguous pixels fall in the criterion.

Rotation calibration: It computes the intensity weighted centroid of the patch with located corner at center. The direction of the vector from this key point to centroid gives the orientation.

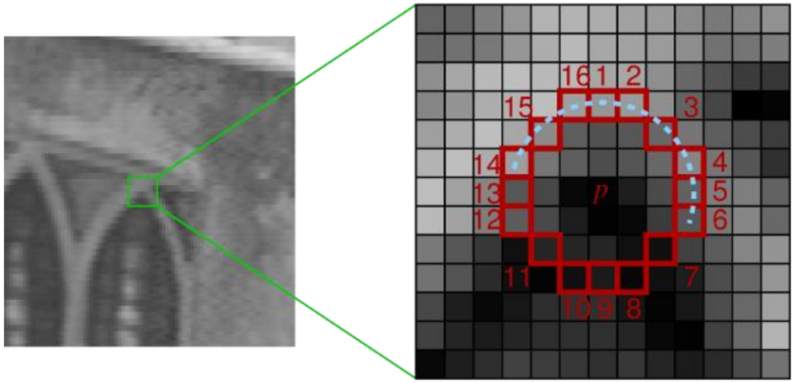
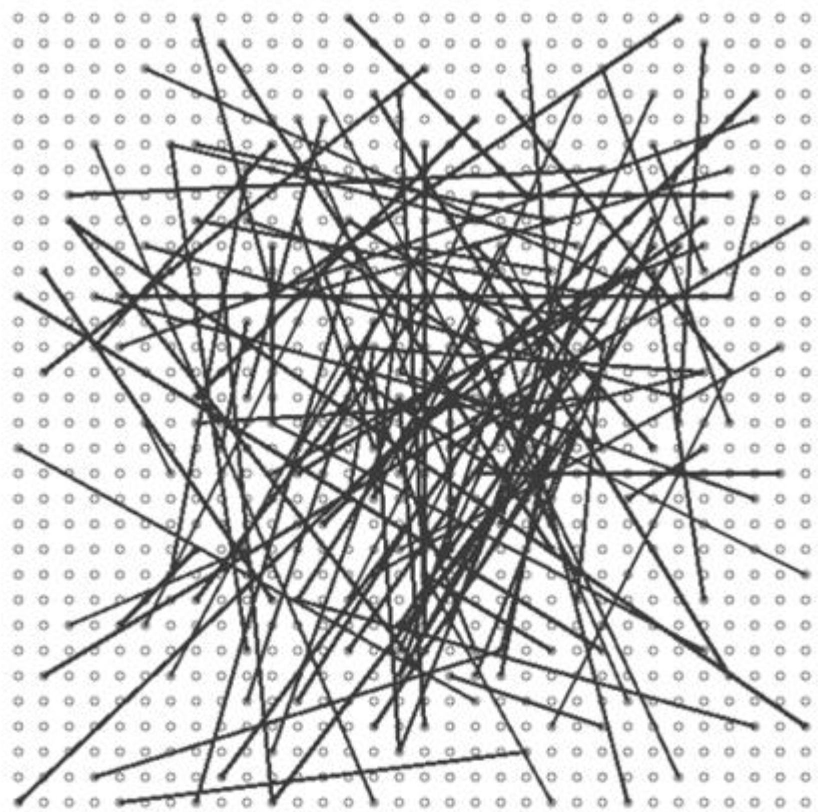


Photo: https://medium.com/software-incubator/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf

ORB: Oriented FAST and rotated BRIEF

Brief (Binary robust independent elementary feature)

- For each detected keypoint (previous slide), sample a set (e.g., 128) of intensity pairs (e.g., pixels a and b) within a squared patch centered at the keypoint
- Create a descriptor using a vector (e.g., 128) of binary code: 1, if $a > b$, else 0.
- Dimension of this feature: Number of pairs (e.g., 128)
- It is a binary descriptor, suitable for very fast Hamming distance matching (just count of the number of bits that are different in the descriptors).
- The pattern is generated randomly only once; then the same pattern is used for all patches. Some works replace this random pattern as a fixed structure pattern.

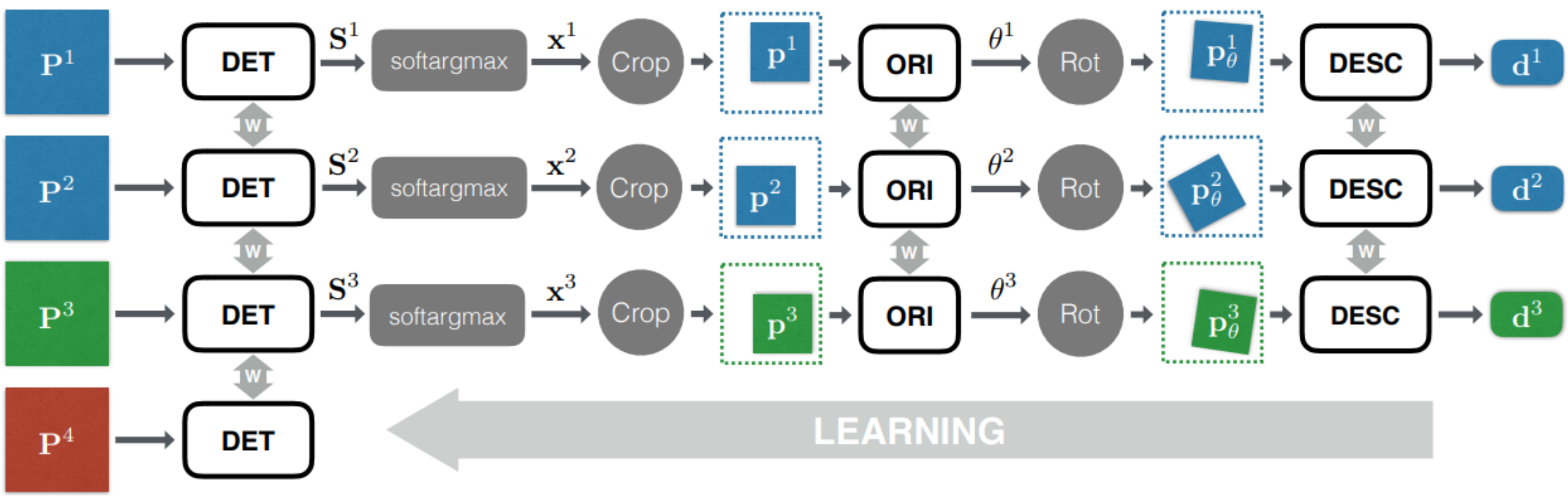


Reference: https://medium.com/@deepanshut041/introduction-to-orb-oriented-fast-and-rotated-brief-4220e8ec40cf

LIFT: Learned invariant feature transform

LIFT: Learned invariant feature transform

- Learning-based descriptor.
- A network to detect keypoint (via the score map).
- A network predicts the patch orientation that is used to derotate the patch.
- A network is used to generate a patch descriptor (128 dimensional).



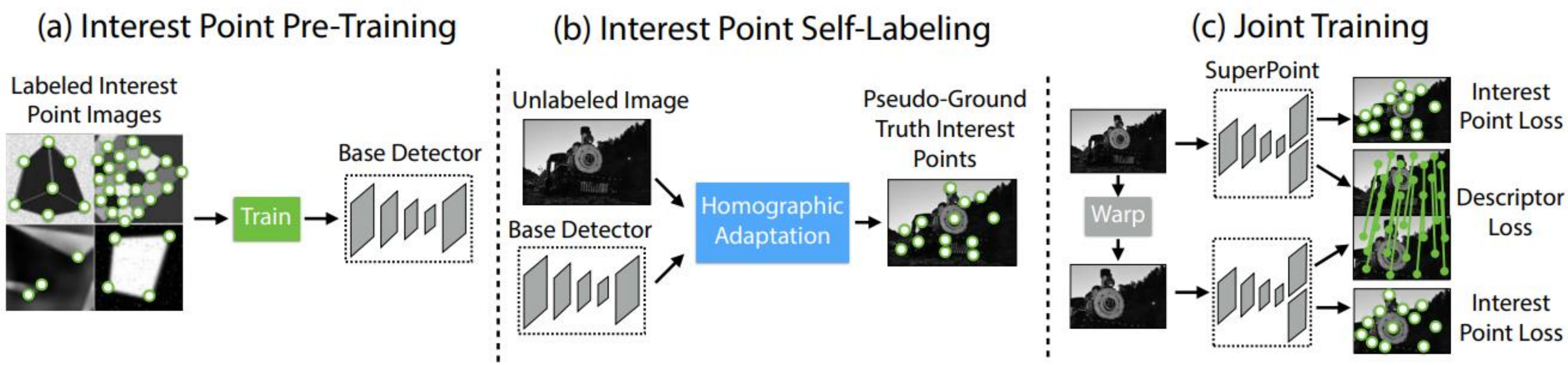
Model training

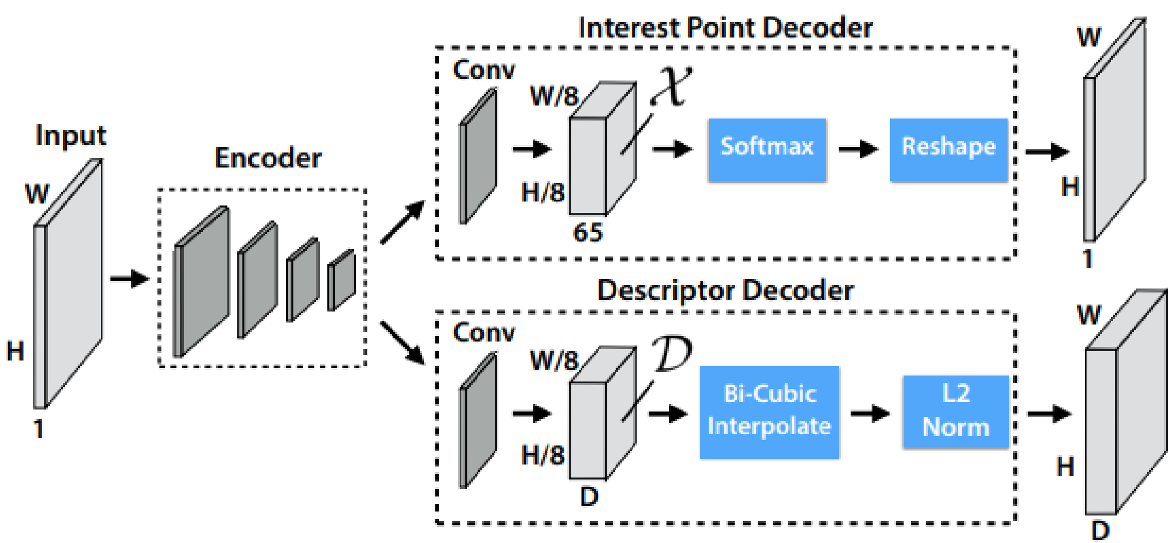
- A Siamese training architecture with four branches, Patches P1 and P2 (blue) are different views of the same physical point, and used as positive examples to train the Descriptor; P3 (green) is a different 3D point as a negative example for the Descriptor; P4 (red) contains no distinctive feature points and is only used as a negative example to train the Detector.
- Loss: Includes detector loss (via the score map), orientation loss (via the rotation), descriptor loss (via the descriptors)

Reference: LIFT: Learned invariant feature transform, ECCV 2016, https://arxiv.org/abs/1603.09114

SuperPoint: Self-supervised interest point detection and description

SuperPoint: Self-Supervised Interest Point Detection and Description





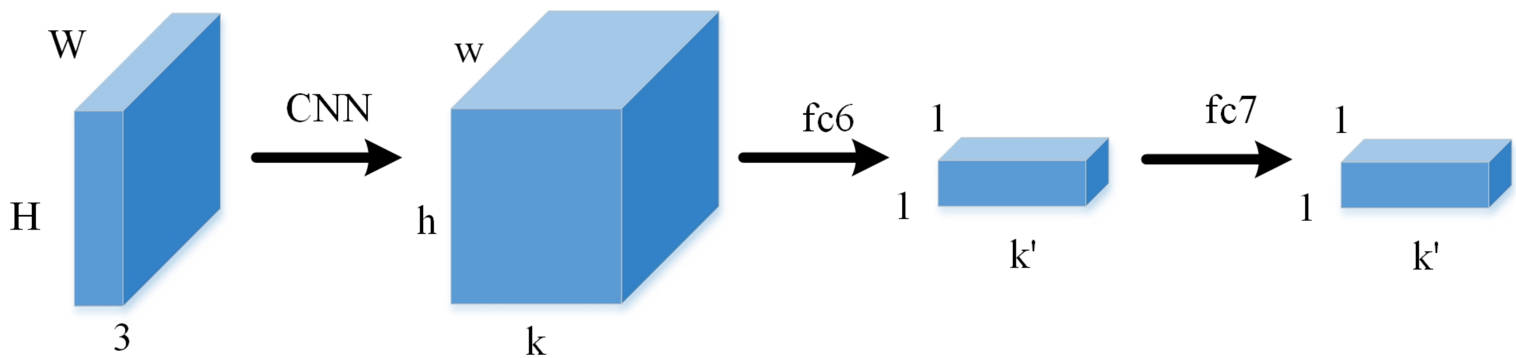
Some tricks to reduce manual annotation of keypoints in images. The final loss is the sum of two intermediate losses: one for the interest point detector, and one for the descriptor. We use pairs of synthetically warped images including (a) pseudo-ground truth interest point locations and (b) the ground truth correspondence from a randomly generated homography that relates the two images.

Reference: SuperPoint: Self-Supervised Interest Point Detection and Description, CVPR 2018, https://openaccess.thecvf.com/content_cvpr_2018_workshops/papers/w9/DeTone_SuperPoint_Self-Supervised_Interest_CVPR_2018_paper.pdf

Pre-trained CNN: Neural code

- Use of feature activation from the top layers of CNN network as high level descriptor
- 3-channel RGB input, 227×227
- AlexNet last pooling layer, global descriptor of dimension $w \times h \times k = 6 \times 6 \times 256 = 9216$
- Alternatively, fully connected layers fc_6, fc_7, global descriptors of dimension $k' = 4096$

Appendix, full (simplified) AlexNet architecture:
[227x227x3] INPUT
[55x55x96] CONV1: 96 11x11 filters at stride 4, pad 0
[27x27x96] MAX POOL1: 3x3 filters at stride 2
[27x27x96] NORM1: Normalization layer
[27x27x256] CONV2: 256 5x5 filters at stride 1, pad 2
[13x13x256] MAX POOL2: 3x3 filters at stride 2
[13x13x256] NORM2: Normalization layer
[13x13x384] CONV3: 384 3x3 filters at stride 1, pad 1
[13x13x384] CONV4: 384 3x3 filters at stride 1, pad 1
[13x13x256] CONV5: 256 3x3 filters at stride 1, pad 1
[6x6x256] MAX POOL3: 3x3 filters at stride 2
[4096] FC6: 4096 neurons
[4096] FC7: 4096 neurons
[1000] FC8: 1000 neurons (class scores)



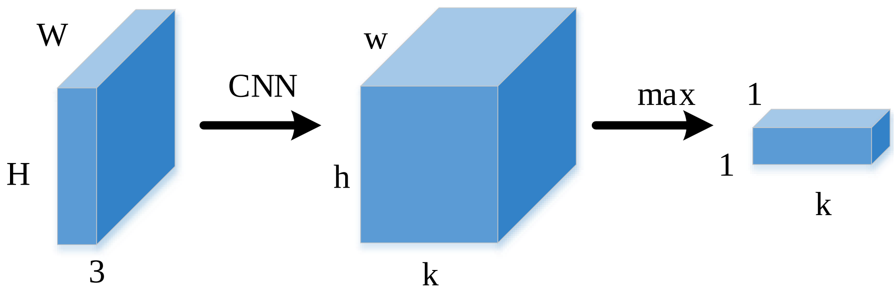
Reference: A. Babenko, et al., Neural Codes for Image Retrieval, ECCV 2014, https://arxiv.org/abs/1404.1777

Pre-trained CNN: Maximum activations

Maximum activations of convolutions (MAC)

- Given a set of 2D convolutional feature channel responses $X = \{X_i\}, i = 1, 2, \cdots k$, spatial max-pooling over all location is given as $f = [f_{\Omega,1}, \cdots, f_{\Omega,k}]$, where $f_{\Omega,i} = \max_{p \in \Omega} X_i(p)$, Ω is the set of valid spatial locations, $X_i(p)$ is the response at the particular position p, k is the number of feature channels

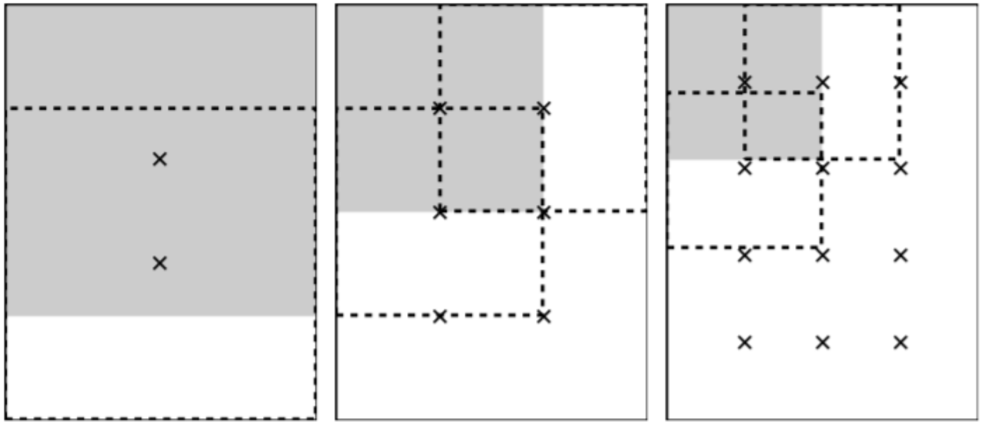
Global feature vector
(max-pooling per activation map)

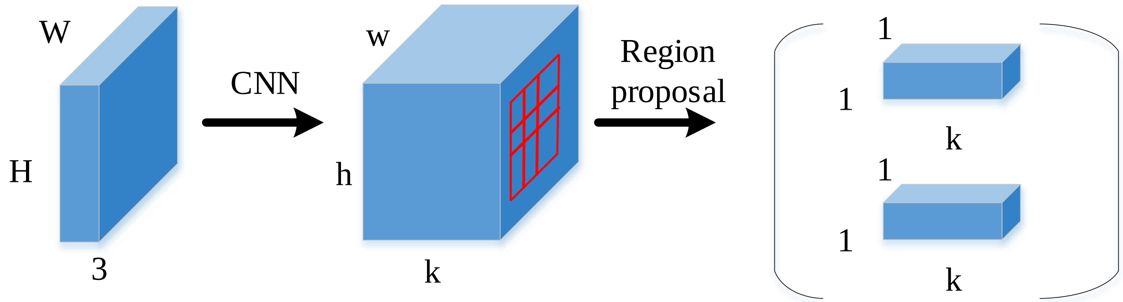


Reference: G. Tolias, et al., Particular object retrieval with integral max-pooling of CNN activations, ICLR 2016, https://arxiv.org/abs/1511.05879

Pre-trained CNN: Maximum activations

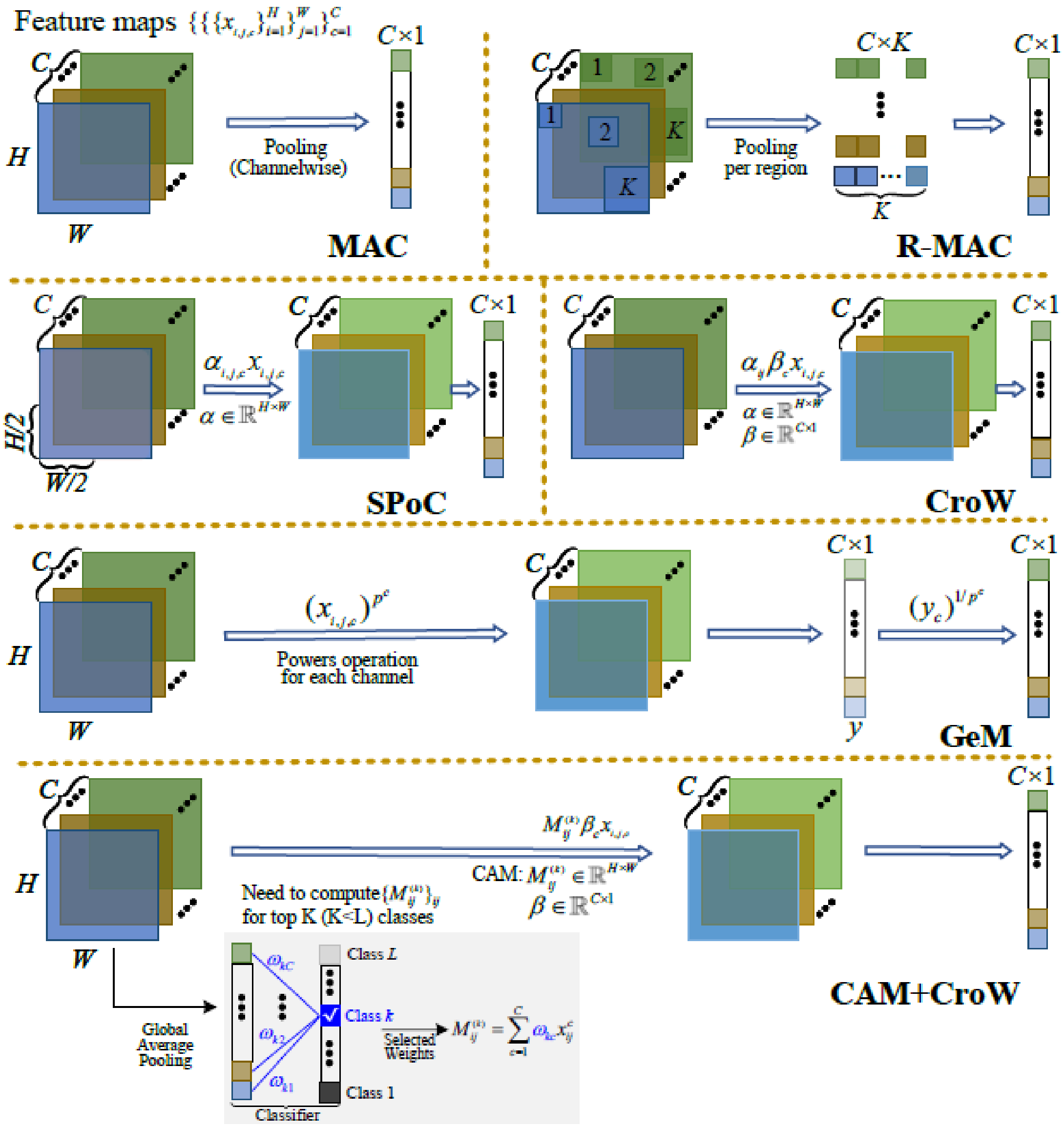
- Sampling region: Sample regions extracted at different scales. We show the top-left region of each scale (gray colored region) and its neighbouring regions towards each direction (dashed borders). The cross indicates the region centre.
- Regional feature vector: Fixed multi-scale overlapping spatial region pooling.





Reference: G. Tolias, et al., Particular object retrieval with integral max-pooling of CNN activations, ICLR 2016, https://arxiv.org/abs/1511.05879

Pre-trained CNN: Others

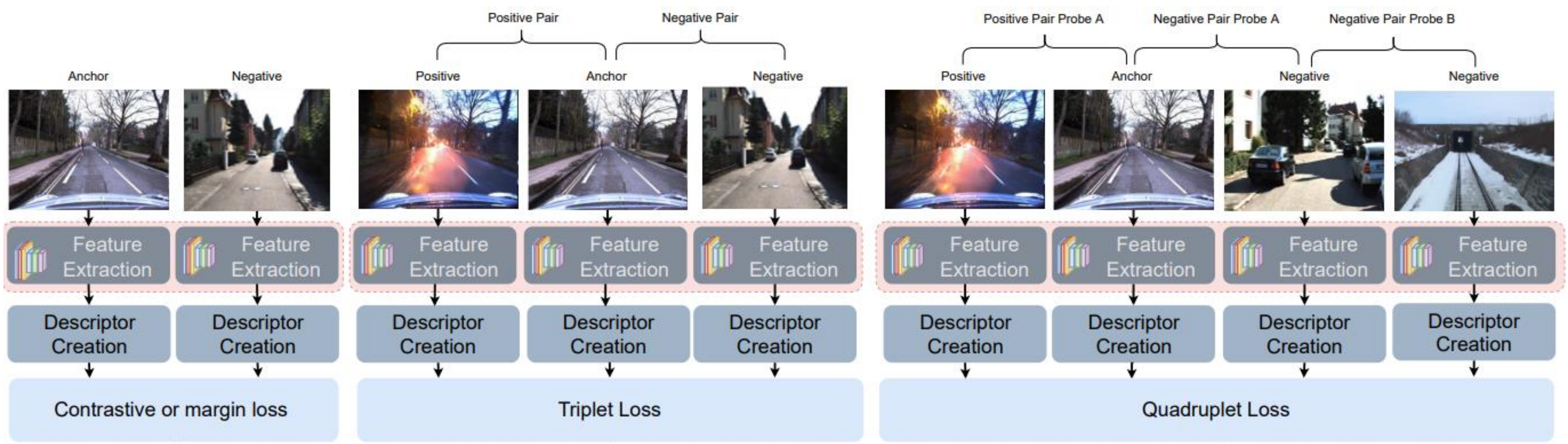


Many other methods that use activation maps obtained from the pre-trained CNN models …

Deep Learning for Instance Retrieval: A Survey, https://arxiv.org/abs/2101.11282

Re-trained CNN: Contrastive CNN

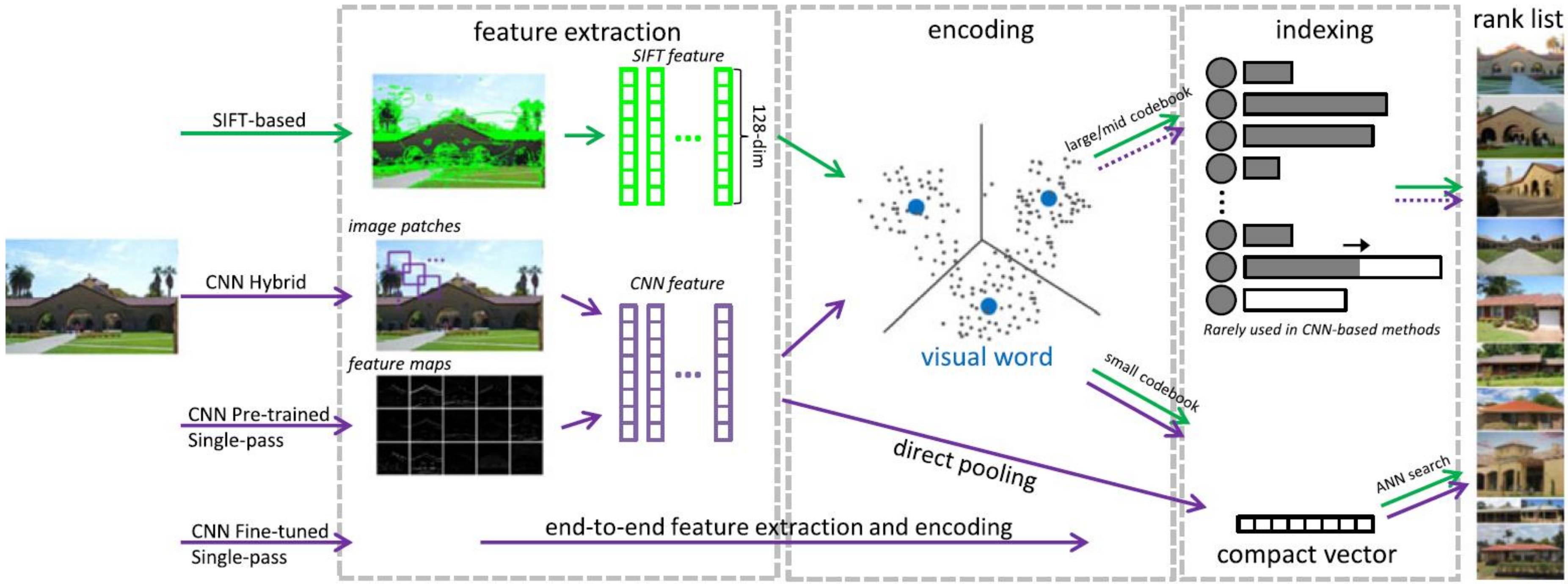
- Contrastive loss has two branches with shared parameters. It computes the similarity distance between the output descriptors of the branches, forcing the networks to decrease the distance between positive pairs (input data from the same place) and increase the distance between negative pairs.
- Triplet loss function computes the distance between a positive and a negative pair at the same iteration, relying, thus, on three branches.
- Quadruplet loss pushes the negative pairs from the positives pairs w.r.t different probe samples, while triplet loss only pushes the negatives from the positives w.r.t from the same probe. The additional constraint of the quadruplet loss reduces the intra-class variations and enlarges the inter-class variations.



Reference: Place recognition survey: An update on deep learning approaches, https://arxiv.org/pdf/2106.10458.pdf

Visual place recognition pipeline (2)

Feature extraction → Feature encoding (dictionary) → Feature indexing (search)

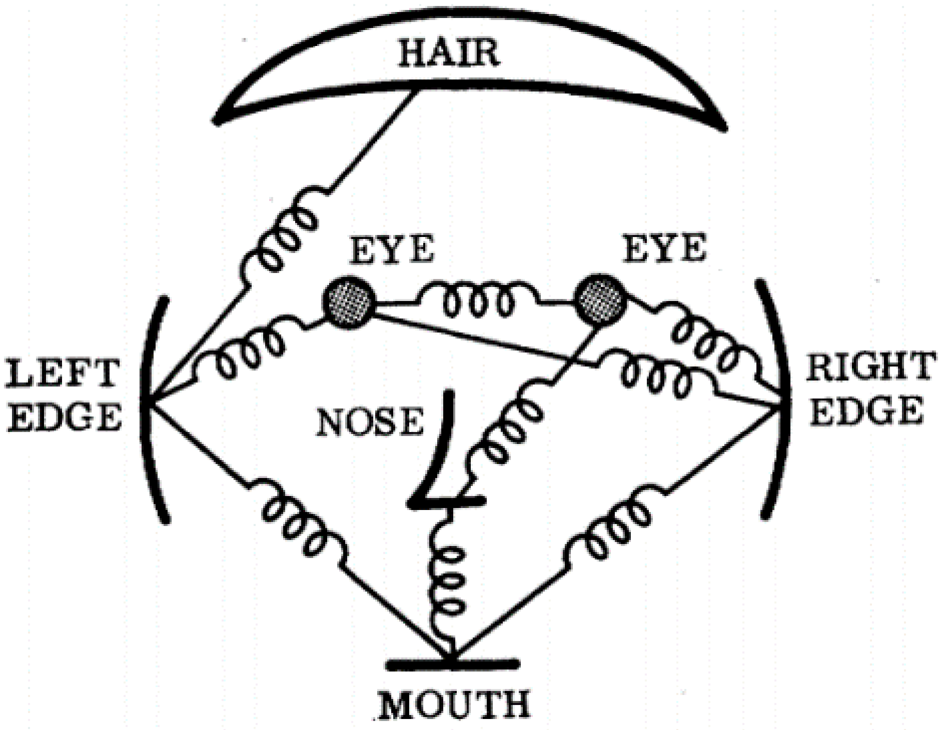


Reference: L. Zheng, Y. Yang, Q. Tian, SIFT Meets CNN: A Decade Survey of Instance Retrieval, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 40, No. 5, May 2018, pp. 1224-1244.

Intuition: Part model

Model

- Object as a set of parts
- Relative locations between parts
- Appearance of part



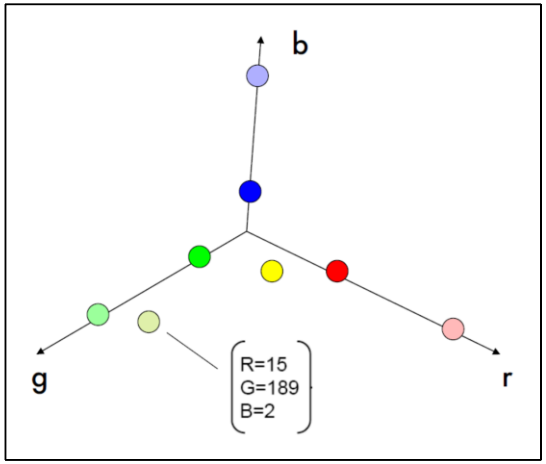
Reference: M. A. Fischler, and R. A. Elschlager, The representation and matching of pictorial structures, IEEE Trans. on Computer, Vol. 22, No. 1, 1973, pp. 67-92, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.118.7951&rep=rep1&type=pdf

Intuition: Histogram

- Consider a histogram h over integers $C = \{0,1,2,3,4\}$, computed from the following samples.
- Each sample is encoded (hard assigned into one vector, all such vectors are pooled (averaged) into one vector.
- C is a codebook or vocabulary.

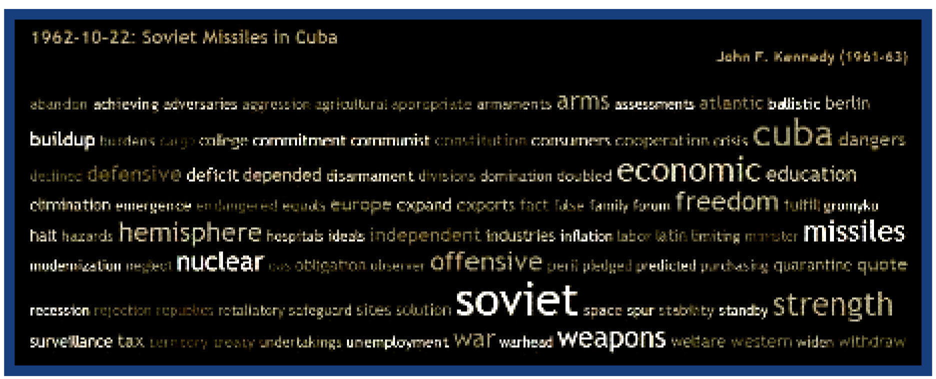
$$
\begin{array}{cccccccccc}
C & = & \{ & 0 & 1 & 2 & 3 & 4 & \} & \\
\hline
3 & \rightarrow & (& 0 & 0 & 0 & 1 & 0 &) & \\
2 & \rightarrow & (& 0 & 0 & 1 & 0 & 0 &) & \\
0 & \rightarrow & (& 1 & 0 & 0 & 0 & 0 &) & \\
3 & \rightarrow & (& 0 & 0 & 0 & 1 & 0 &) & \\
2 & \rightarrow & (& 0 & 0 & 1 & 0 & 0 &) & \\
2 & \rightarrow & (& 0 & 0 & 1 & 0 & 0 &) & + \\
\hline
h & = & (& 1 & 0 & 3 & 2 & 0 &) & / \quad 6
\end{array}
$$

An example on color space



Intuition: Keywords in document

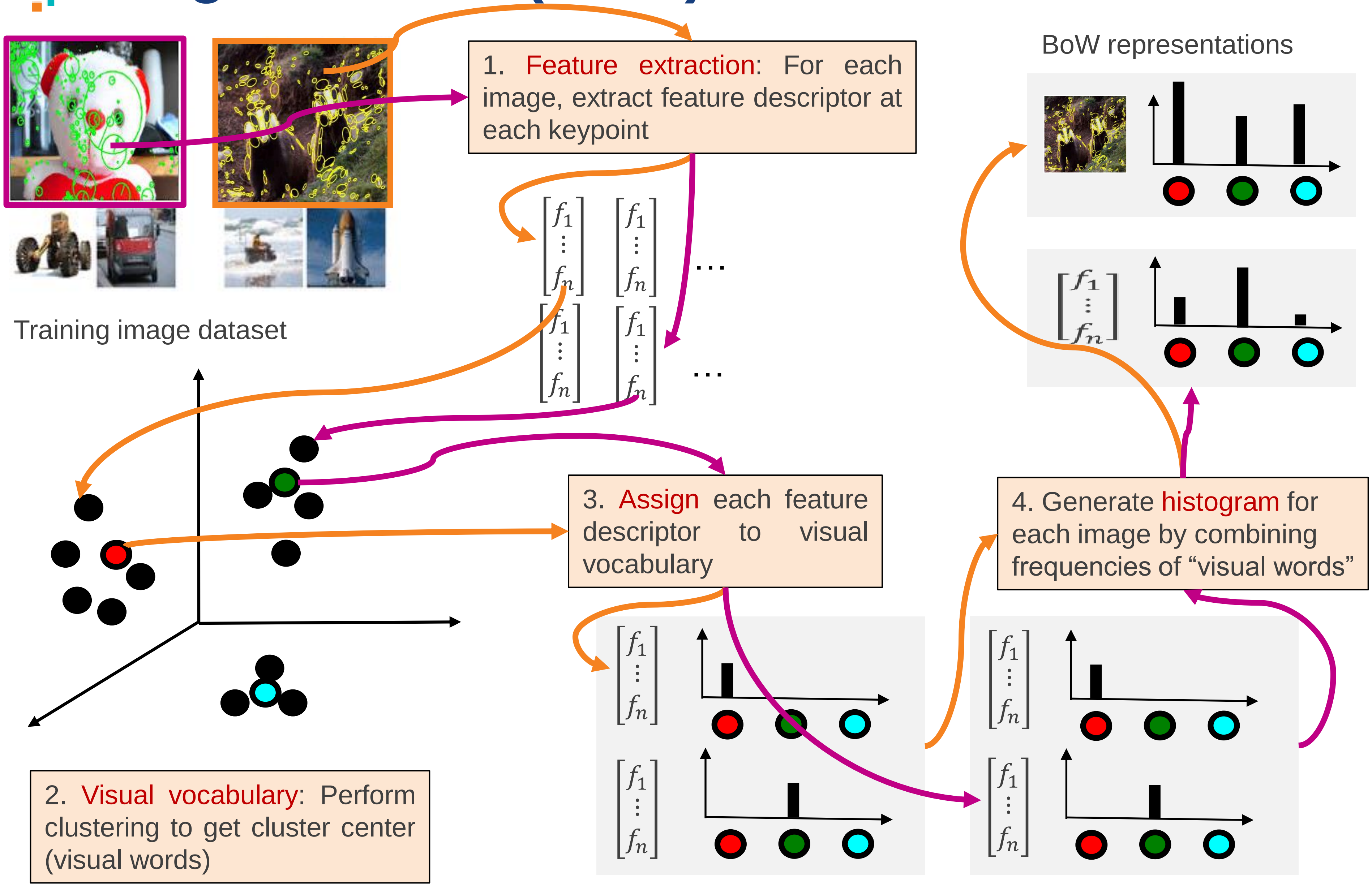
Document representation: Frequencies of keywords from a dictionary.



Reference:

1. G. Salton and M. J. McGill, Introduction to Modern Information Retrieval, 1986
2. US Presidential Speeches Tag Cloud, http://chir.ag/phernalia/preztags/

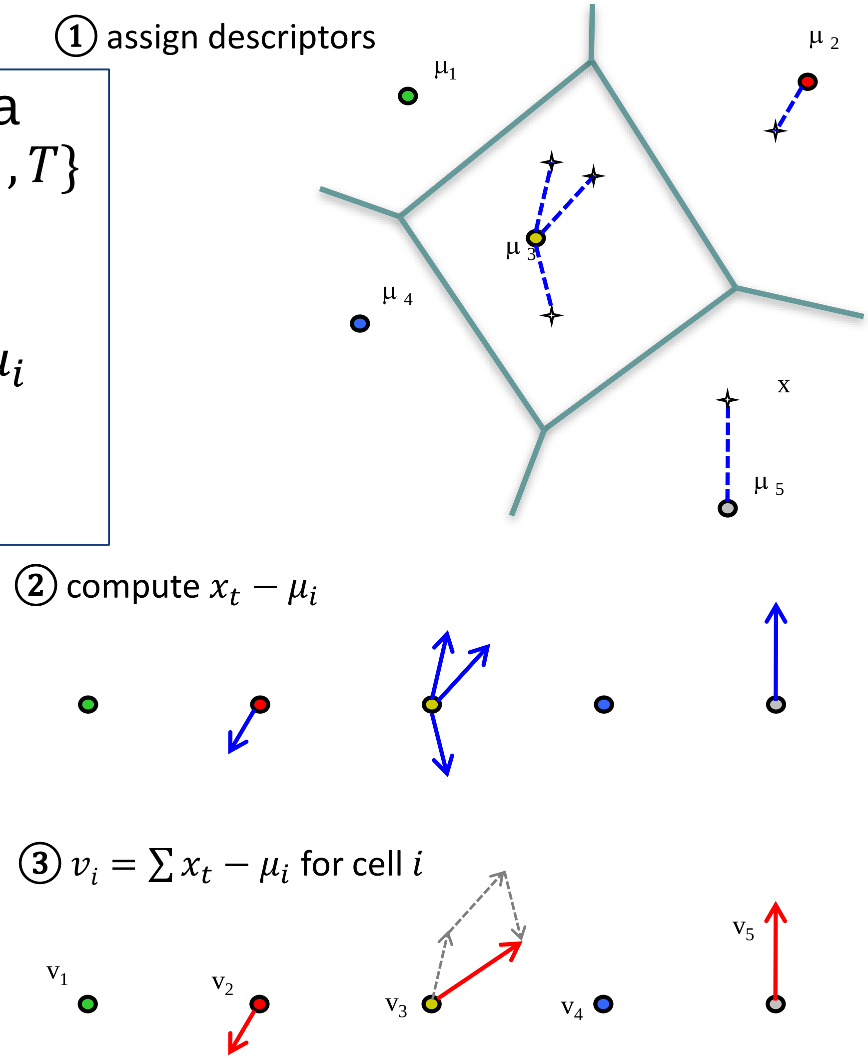
Bag-of-words (BoW): Overview



VLAD: Vector of locally aggregated descriptors

Given a codebook $\{\mu_i, i = 1, \cdots, N\}$ and a set of input descriptors $X = \{x_t, t = 1, \cdots, T\}$

- ① assign: $NN(x_t) = \arg\min_{\mu_i} \|x_t - \mu_i\|$
- ②③ compute: $v_i = \sum_{x_t : NN(x_t) = \mu_i} x_t - \mu_i$
- concatenate v_i



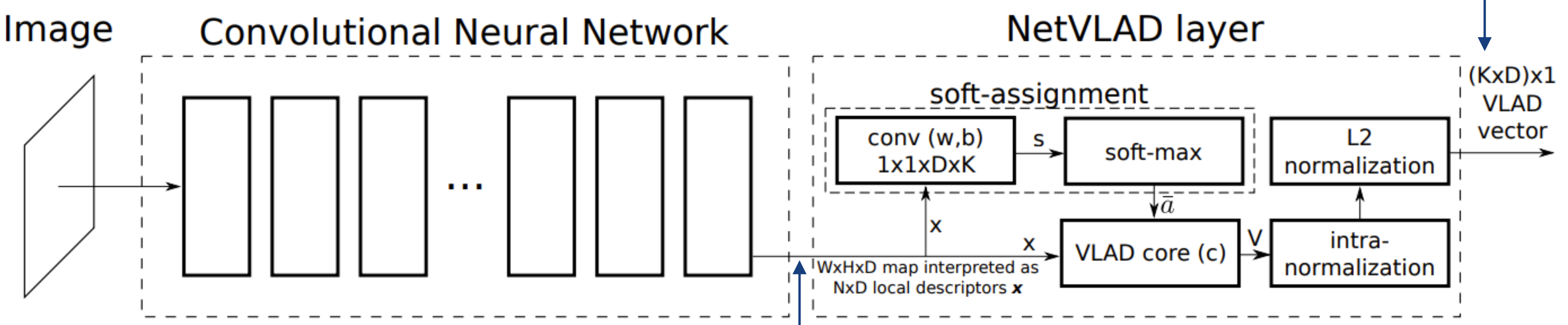
$$v_i = \underbrace{\sum_t}_{\substack{\text{Sum over all (blue) descriptors in}\\ \text{each cell. Then, all (red) residual}\\ \text{vectors are normalized.}}} \overbrace{a_i(x_t)}^{\text{0/1 assignment of } x_t \text{ to cluster } i} \underbrace{(x_t - c_i)}_{\text{Residual vector}}$$

Reference: H. Jegou, et al.,, Aggregating local image descriptors into compact codes, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 34, No. 9, 2012, pp.1704-1716. https://hal.inria.fr/inria-00633013/document/

NetVLAD

- An NetVLAD layer is integrated into the existing CNN framework to extract image-level descriptors. It includes clustering (soft assignment according to learned cluster centers), residual calculation (VLAD core) and normalization.
- Model training: Contrastive learning using labelled images depicting the same places from the Internet.

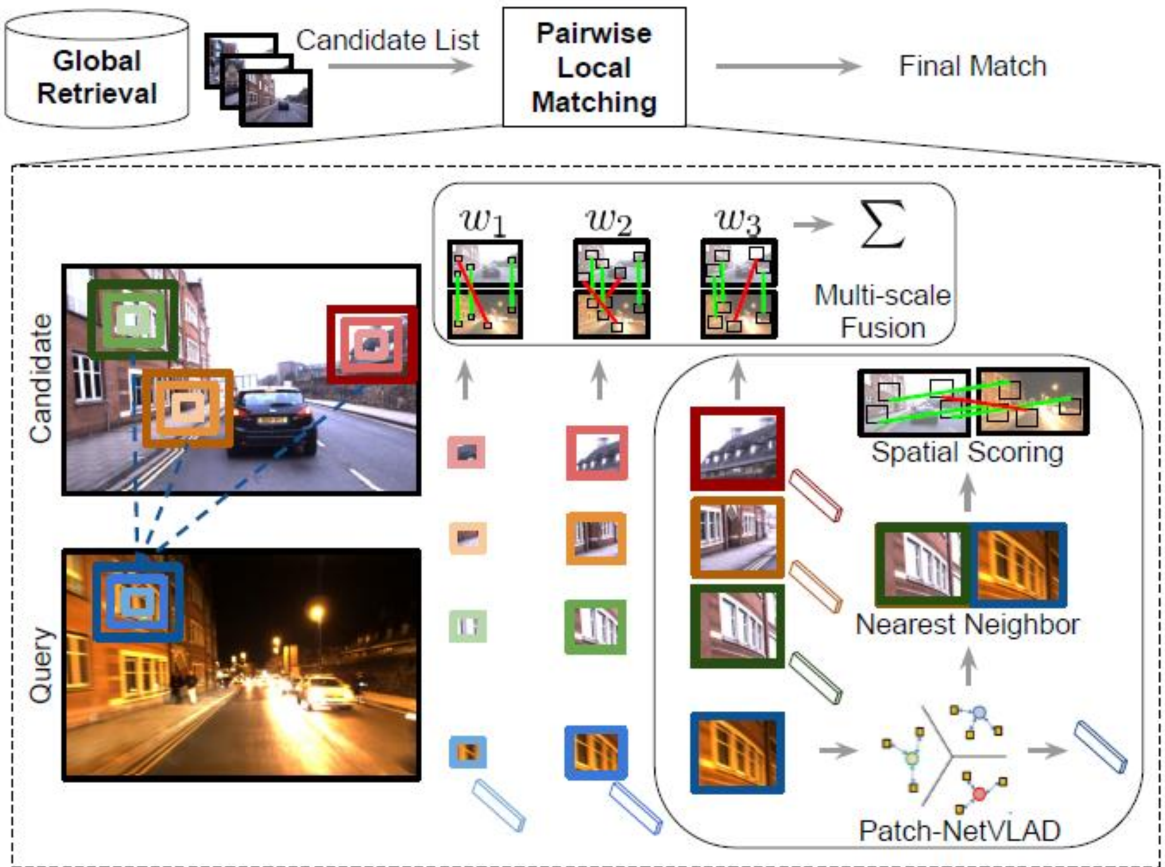
K: # clusters



Input: $(W \times H) \times D$ feature vectors
(dense descriptors obtained from CNN)

Reference: NetVLAD: CNN architecture for weakly supervised place recognition, CVPR 2016, https://arxiv.org/abs/1511.07247

Patch NetVLAD



Patch-NetVLAD takes as input an initial list of most likely reference matches to a query image, ranked using NetVLAD descriptor comparisons. For top ranked candidate images, it computes new patch-level descriptors at multiple scales to perform local cross-matching of these descriptors across query and candidate images with geometric verification, and uses these match scores to re-order the initial list, producing the final image retrievals.

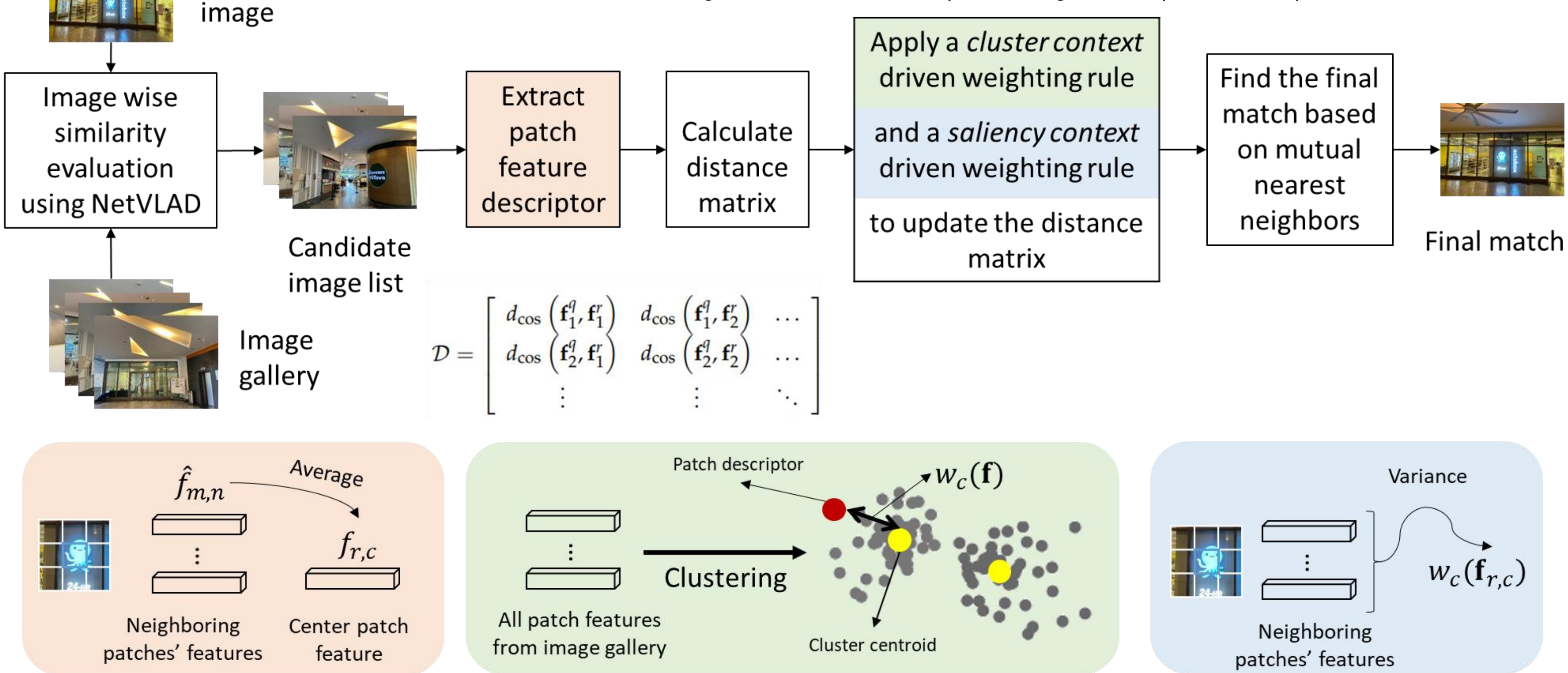
Reference: Patch-NetVLAD: Multi-Scale Fusion of Locally-Global Descriptors for Place Recognition, CVPR 2021, https://arxiv.org/abs/2103.01486

Contextual patch-NetVLAD

Contextual patch-NetVLAD:

- A context-driven patch feature descriptor aggregates features from each patch's surrounding neighborhood.
- A context-driven feature matching mechanism utilizes cluster and saliency context-driven weighting rules to assign higher weights to patches that are less similar to densely populated or locally similar regions.

Reference: Contextual Patch-NetVLAD: Context-aware patch feature descriptor and patch matching mechanism for visual place recognition, https://www.mdpi.com/1424-8220/24/3/855

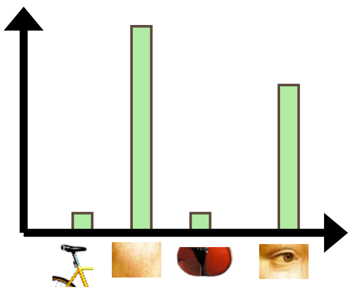


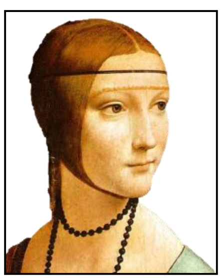
Context driven patch feature descriptor | A *cluster context* driven weighting rule | A *saliency context* driven weighting rule

BoW: Similarity evaluation

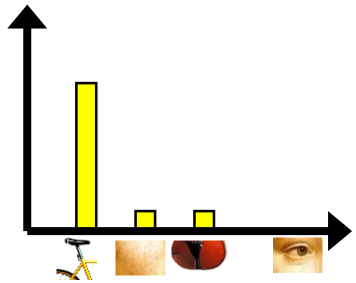
- Evaluate similarity of two images based on their BoW representations

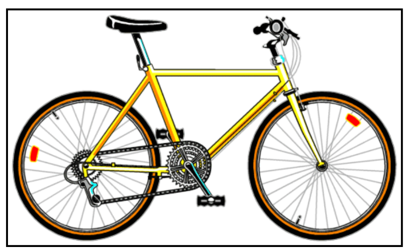
$$\mathbf{p} = [1, 8, 1, 4]$$





$$\mathbf{q} = [5, 1, 1, 0]$$





Histogram Intersection

$$\mathbf{H}_1 = (10, 0, 0, 0, 100, 10, 30, 0, 0)$$
$$\mathbf{H}_2 = (0, 40, 0, 0, 0, 6, 0, 110, 0)$$
$$S = \sum_{i=1}^{N} \min(H_1(i), H_2(i)) = 6$$

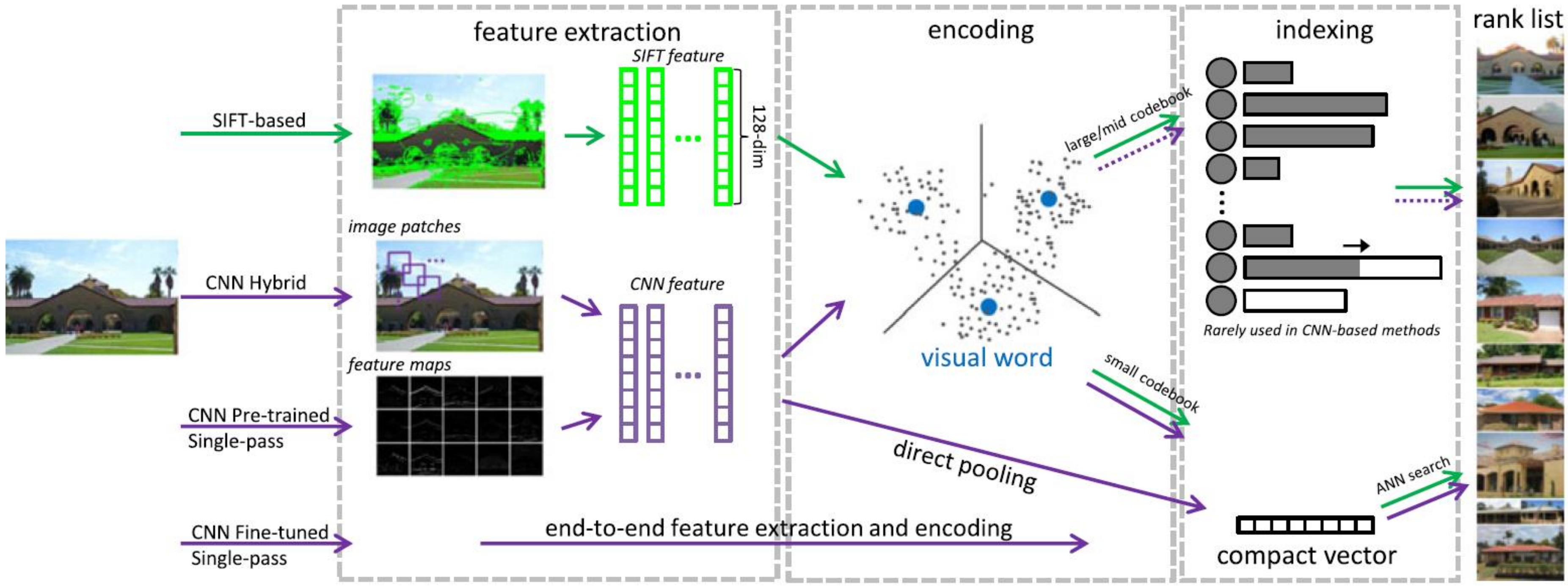
Euclidean distance

$$\mathbf{H}_1 = (10, 0, 0)$$
$$\mathbf{H}_2 = (0, 40, 0)$$
$$S = \sqrt{\sum_{i=1}^{N} (H_1(i) - H_2(i))^2} = 41.23$$

Reference: http://vision.cs.utexas.edu/376-spring2018/slides/lecture18-spring2018.pdf

Visual place recognition pipeline (3)

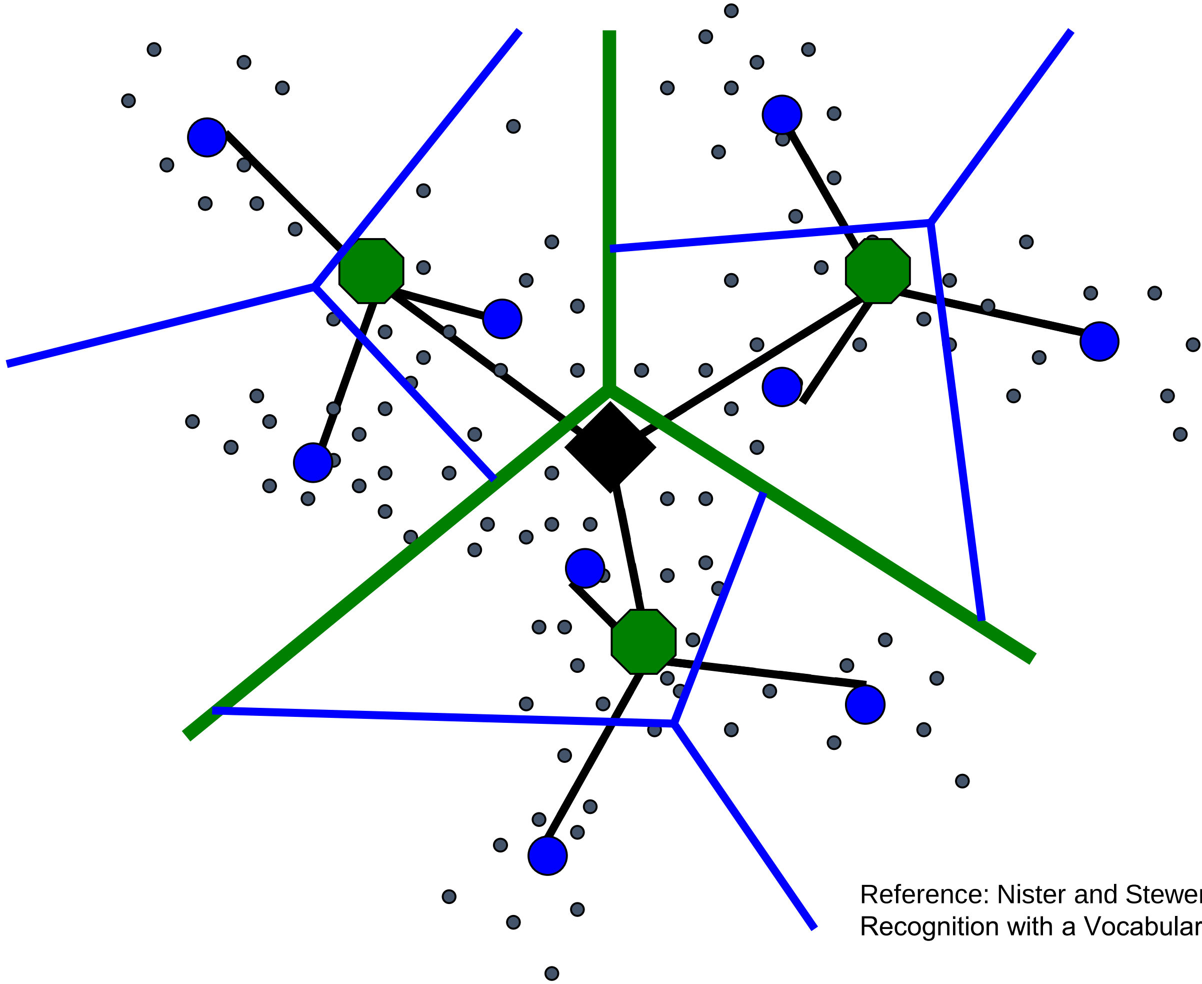
Feature extraction → Feature encoding (dictionary) → **Feature indexing (search)**



Reference: L. Zheng, Y. Yang, Q. Tian, SIFT Meets CNN: A Decade Survey of Instance Retrieval, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 40, No. 5, May 2018, pp. 1224-1244.

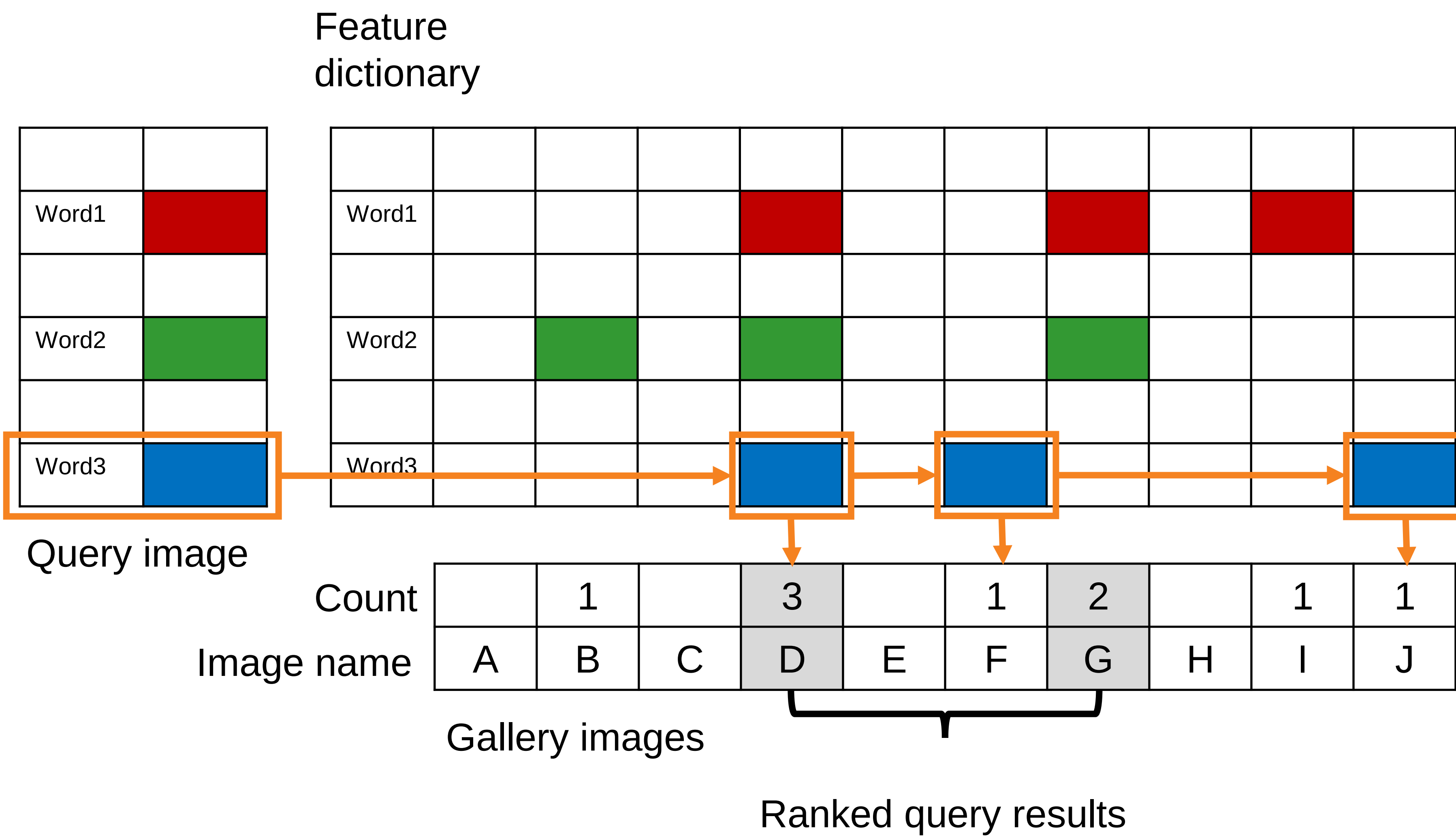
Vocabulary trees (1): Hierarchical clustering for large vocabularies

- Tree construction:



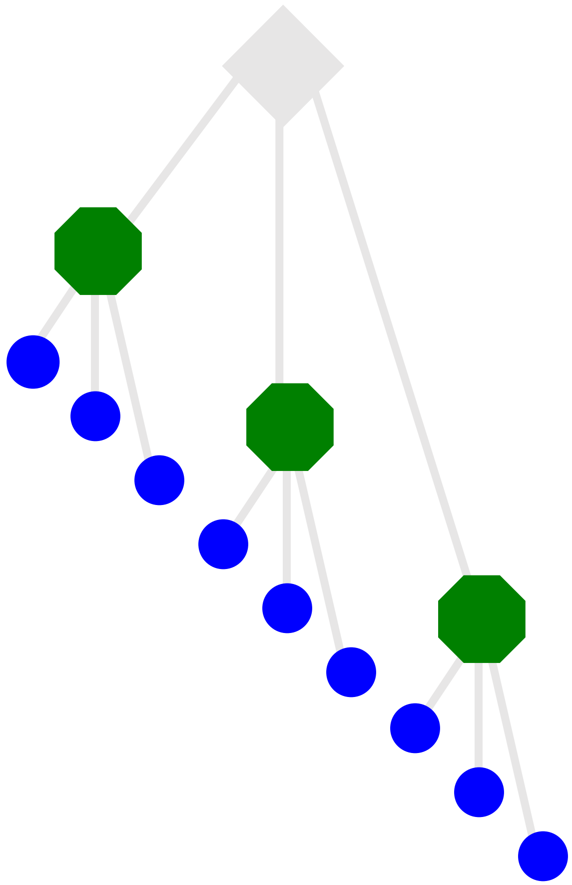
Reference: Nister and Stewenius, "Scalable Recognition with a Vocabulary Tree", CVPR 2006

Vocabulary trees (2): Inverted file index



Vocabulary tree

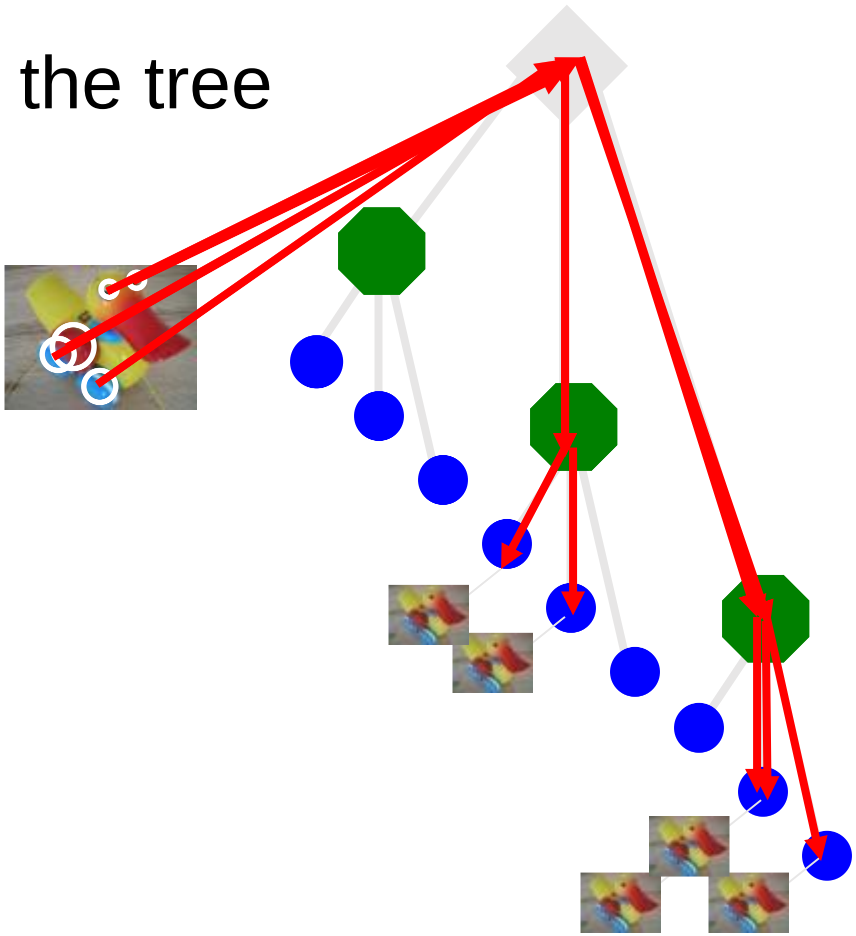
- Indexing: Filling the tree



Reference: Nister and Stewenius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

Vocabulary tree

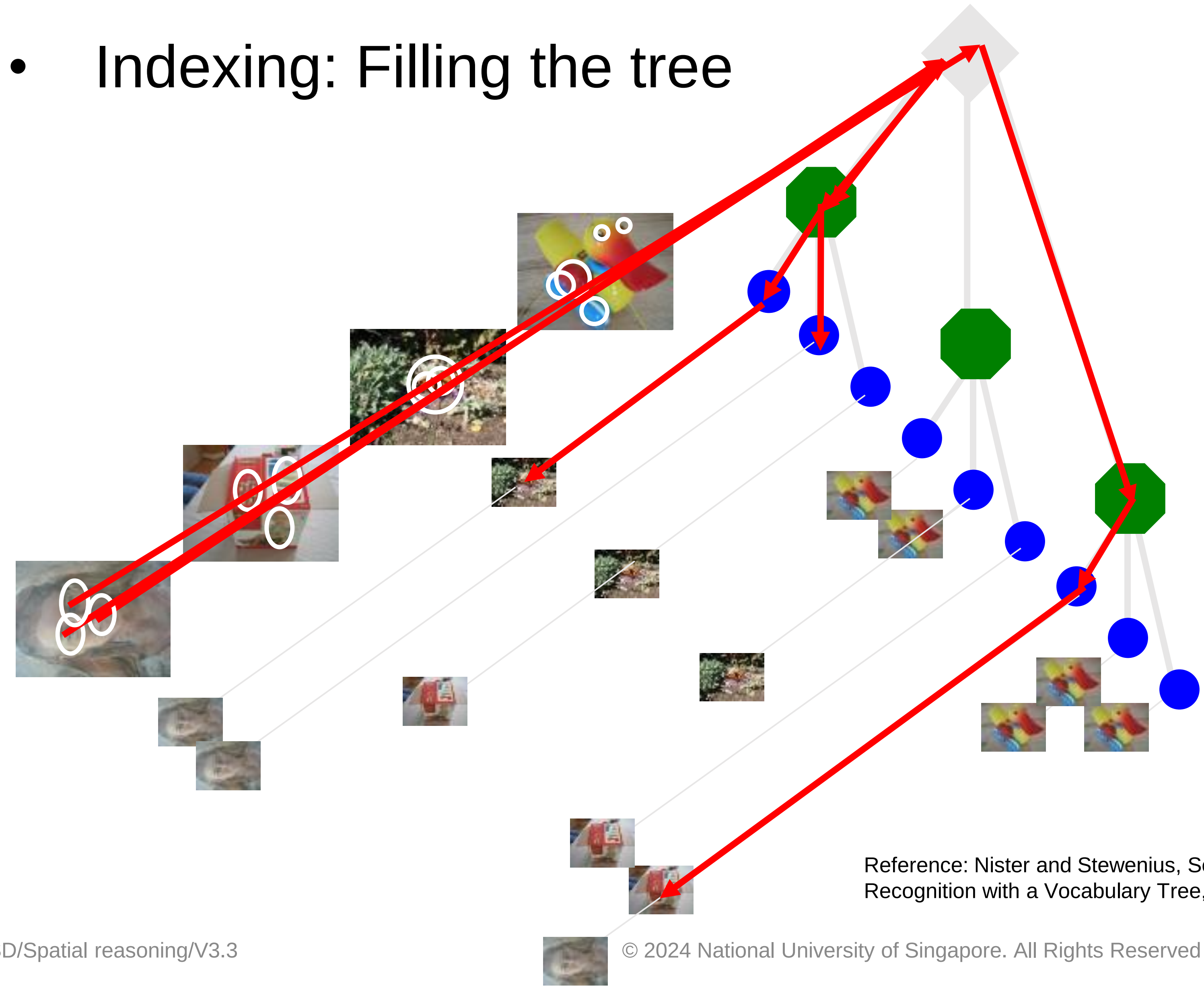
- Indexing: Filling the tree



Reference: Nister and Stewenius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

Vocabulary tree

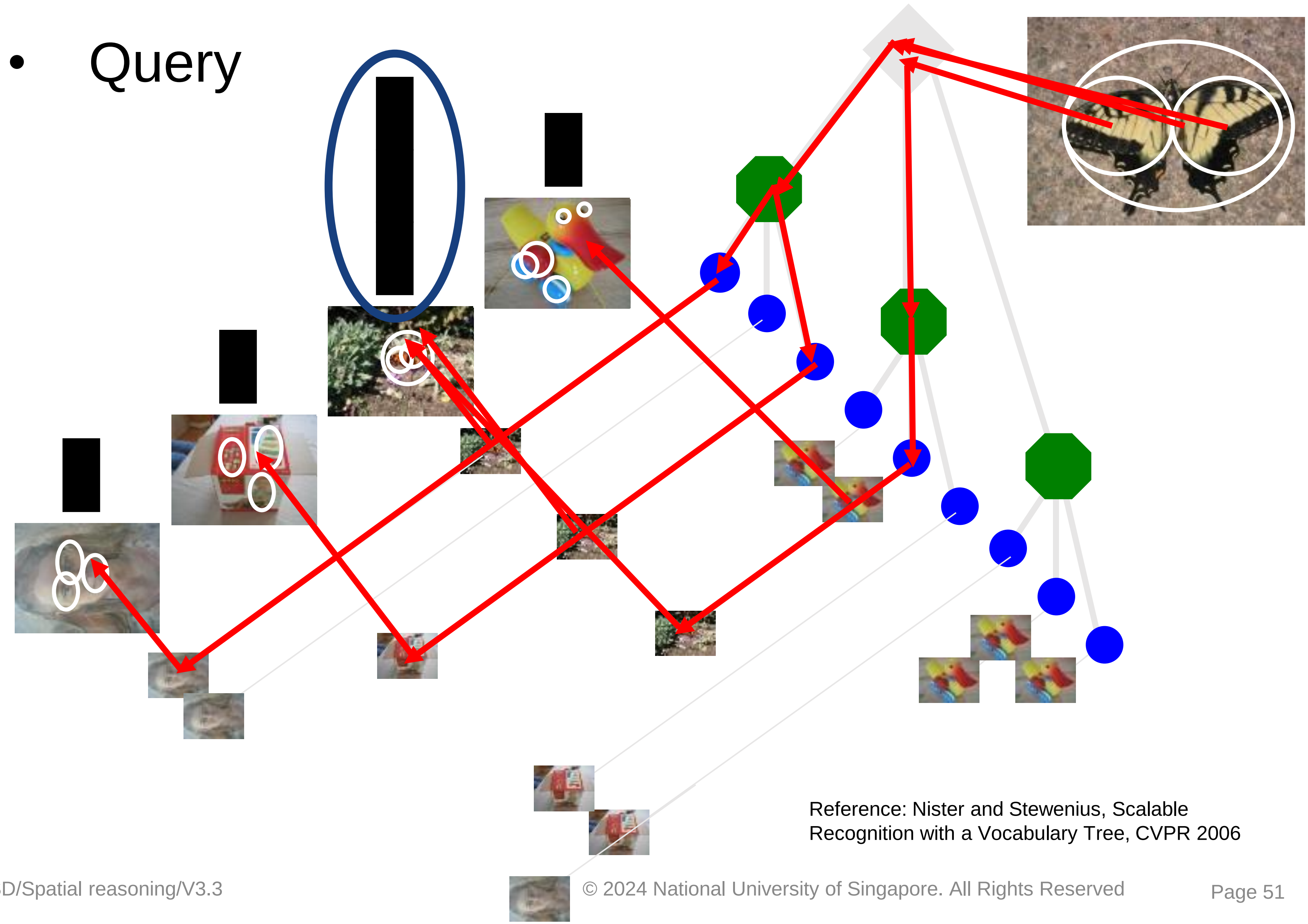
- Indexing: Filling the tree



Reference: Nister and Stewenius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

Vocabulary tree

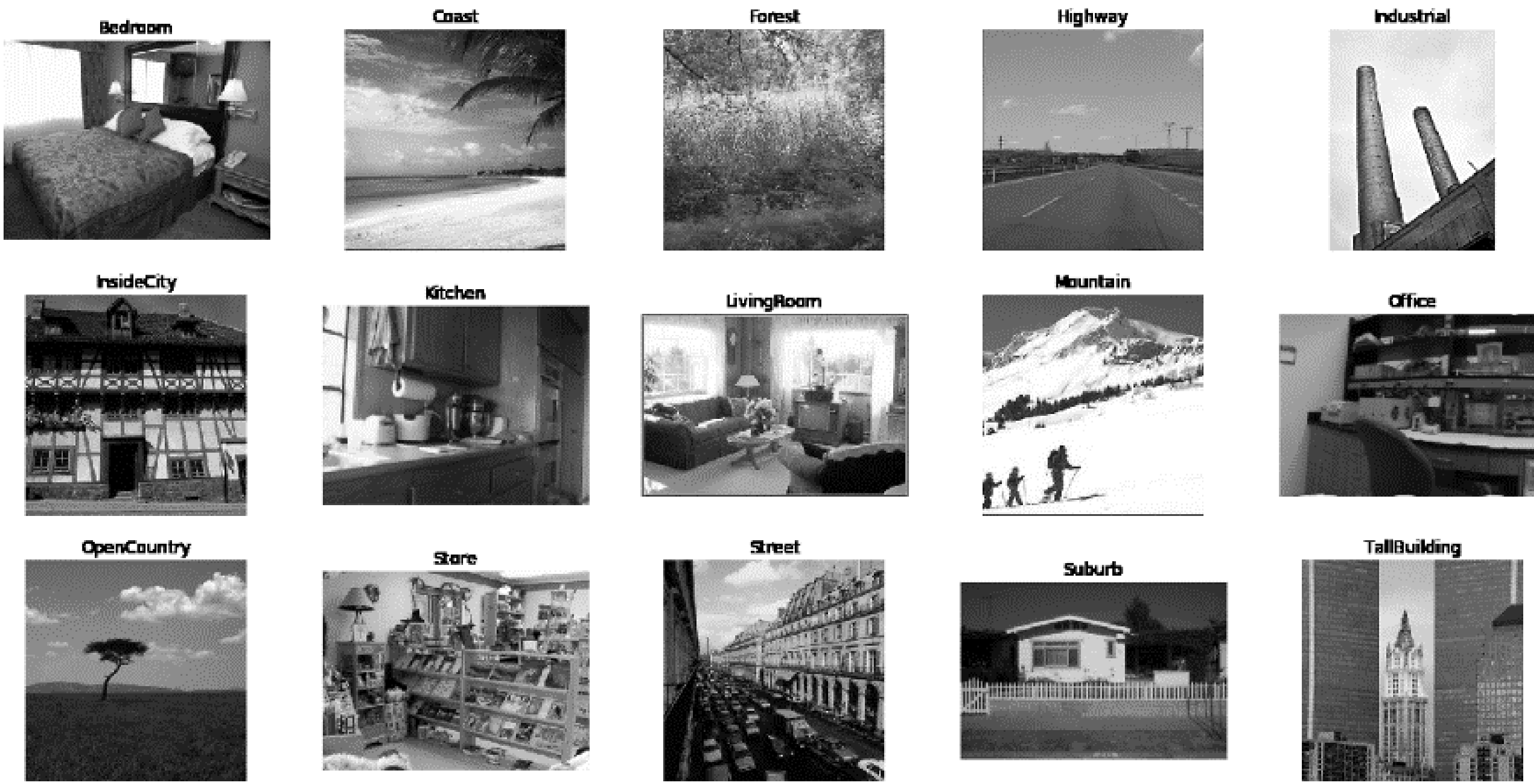
- Query



Reference: Nister and Stewenius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

Workshop

- Objective: Perform image-based place recognition.
- Dataset: Scene recognition, https://www.cc.gatech.edu/~hays/compvision/proj4/



Workshop

- Neural code: A. Babenko, et al., Neural Codes for Image Retrieval, ECCV 2014, https://arxiv.org/abs/1404.1777
- Global sum-pooling: A. Babenko and V. Lempitsky, Aggregating Deep Convolutional Features for Image Retrieval, ICCV 2015, https://arxiv.org/abs/1510.07493

Thank you!

Dr TIAN Jing
Email: tianjing@nus.edu.sg