Long Range Image Matching and Its Applications

Harpreet S Sawhney

Microsoft / Vision & Mixed Reality

May 12th , 2020

Jumble of Disparate Photos







How do you align and build a panorama?

- Humans are not generally good at alignment!
- This is a task where machines are superior : Fast, Accurate, Precise

- Ingredients for a solution?

How do you align and build a panorama?

- Humans are not generally good at alignment!
- This is a task where machines are superior : Fast, Accurate, Precise

- Ingredients for a solution?

1. Features that are invariant or quasi-invariant to larger changes in viewpoint and illumination.
    1. Patches used in the last lecture will be hopelessly inadequate. Why?

2. Aligning with limited overlap

   Local Alignment

3. Aligning and constraining the placement of multiple pictures simultaneously

   Global Alignment

4. Bundle Adjustment

Automatic Panoramic Image Stitching using Invariant Features

Matthew Brown and David G. Lowe
{mbrown|lowe}@cs.ubc.ca
Department of Computer Science,
University of British Columbia,
Vancouver, Canada.

International Journal of Computer Vision. 74(1), pages 59-73, 2007

http://matthewalunbrown.com/papers/ijcv2007.pdf

AutoStitch: a new dimension in automatic image stitching

http://matthewalunbrown.com/autostitch/autostitch.html#publications

Location Recognition with Pre-built Image and SfM Database

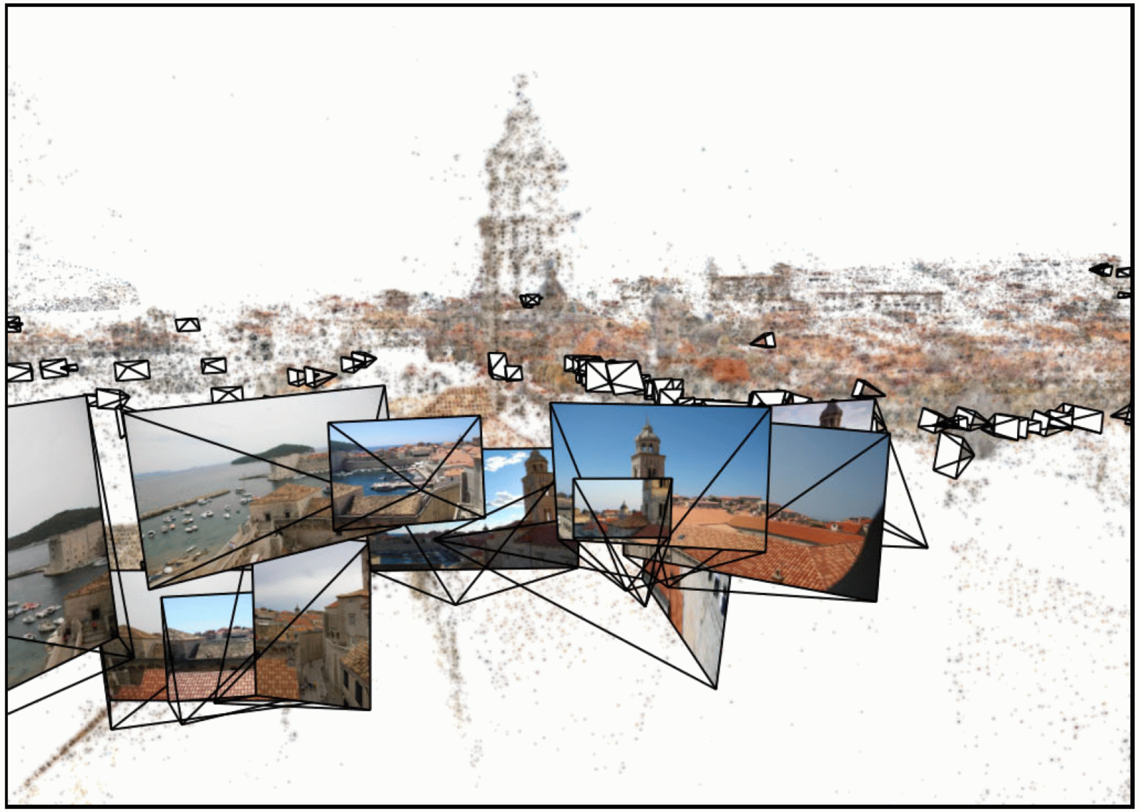


Fig. 5. Visualization of registration and localization on the Dubrovnik data set, showing the camera locations and their corresponding views (i.e. registered test images), as well as the 3D point cloud of the (full) model. Two more examples are shown in Figure 6.

Location Recognition using Prioritized Feature Matching

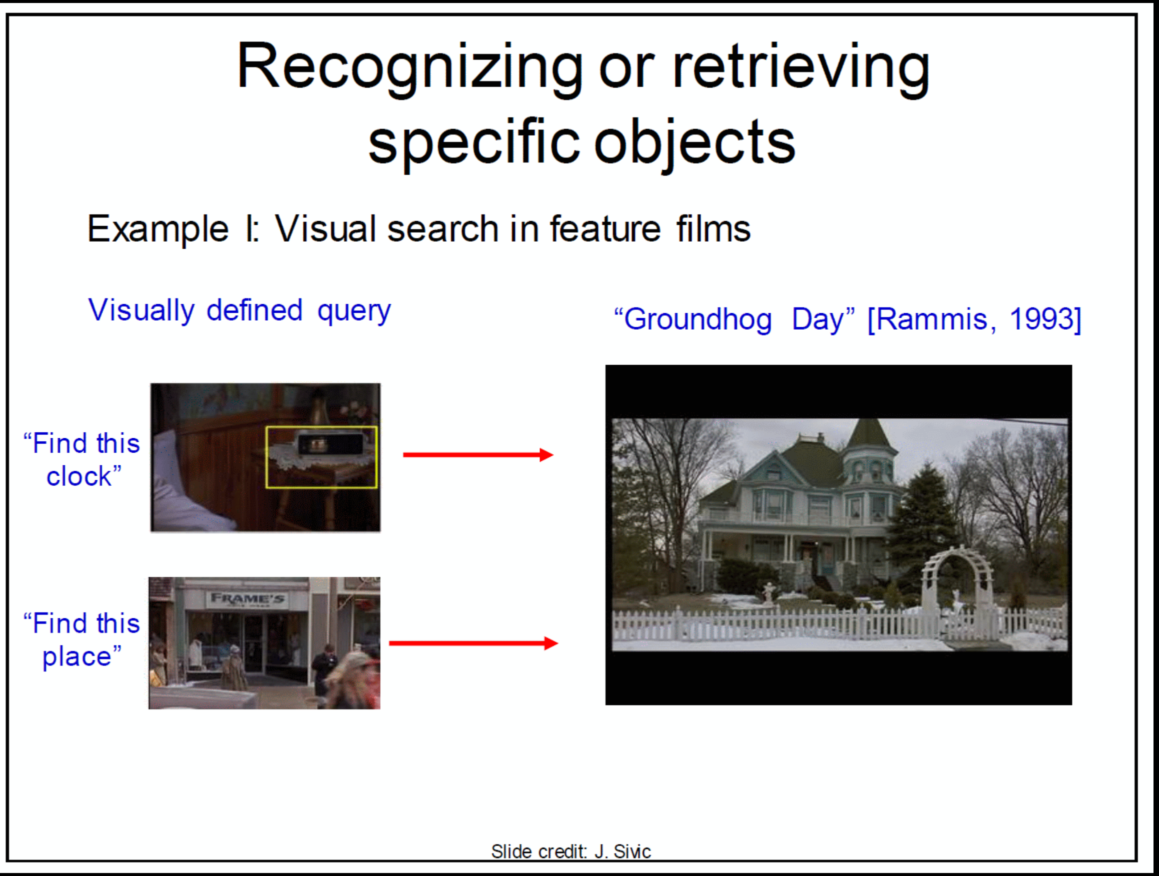
Yunpeng Li   Noah Snavely   Dan Huttenlocher

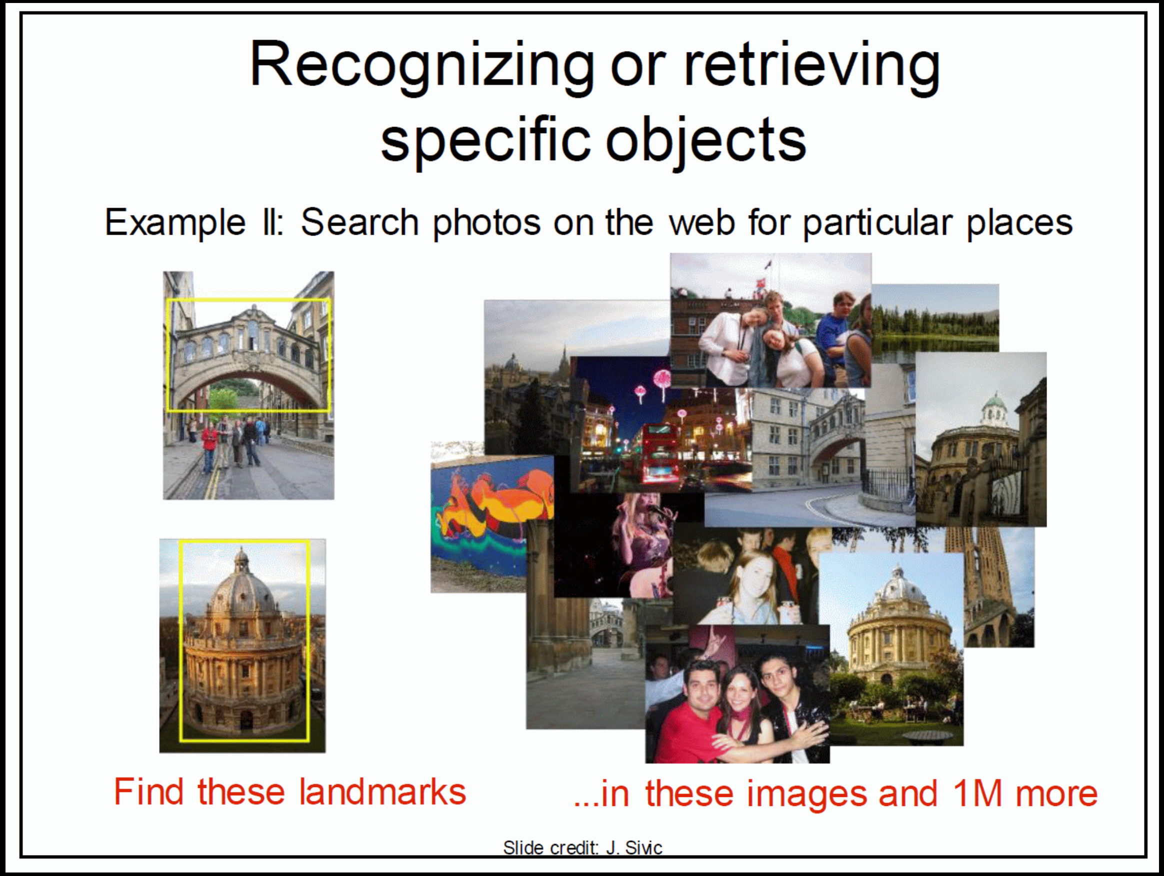
https://research.cs.cornell.edu/p2f/

ECCV 2010

Again need is for Long-range feature Matching under large time gaps between database images and query images.

Instance Recognition & Retrieval: Specific Entity in a Large Database





- Again long range feature matching against a large database of objects

Intuition behind Scale / Rotation / Affine Invariant Feature

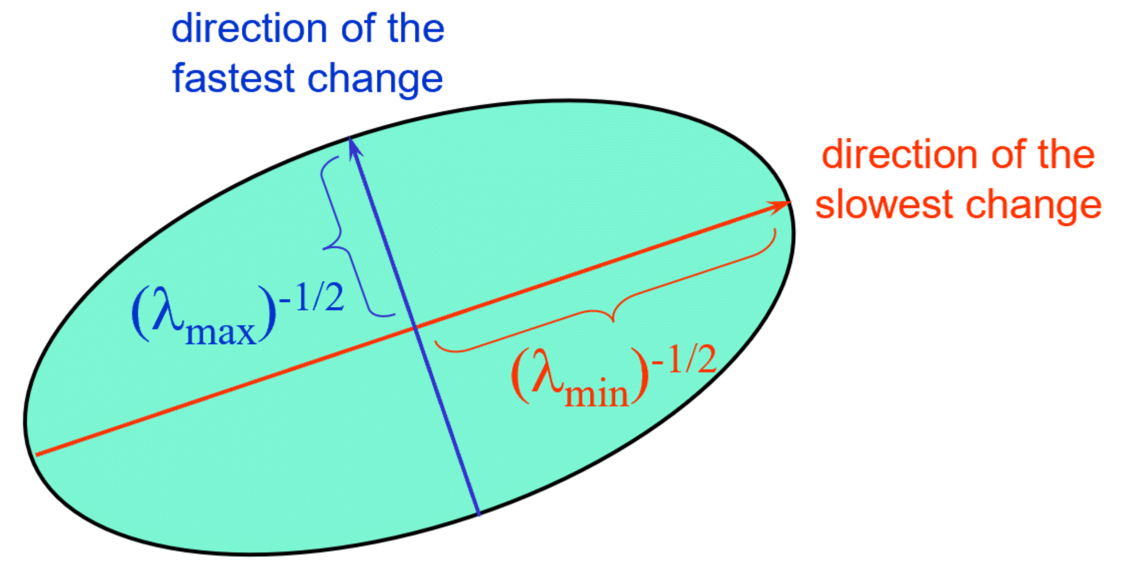
- How can we detect all the flowers?
- A Blob detector computed over scale-space is a stable localizer of a feature's position
- And if we can make its descriptor invariant to Rotations and/or Affine transformations we can match it under large scale transformations

Are Harris corners good features for Long Range Matching?

- Recall the Second Moment Matrix and its use in Corner Detection:

$$M = R^T \begin{bmatrix} \lambda_{max} & 0 \\ 0 & \lambda_{min} \end{bmatrix} \mathrm{R}$$

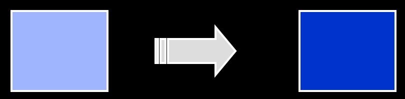
- Geometrically this is an ellipse with orientation R and axes determined by the eigenvalues.



- Cornerness determined by both eigenvalues non-zero and almost equal.

Corneress : Variations with Affine intensity change

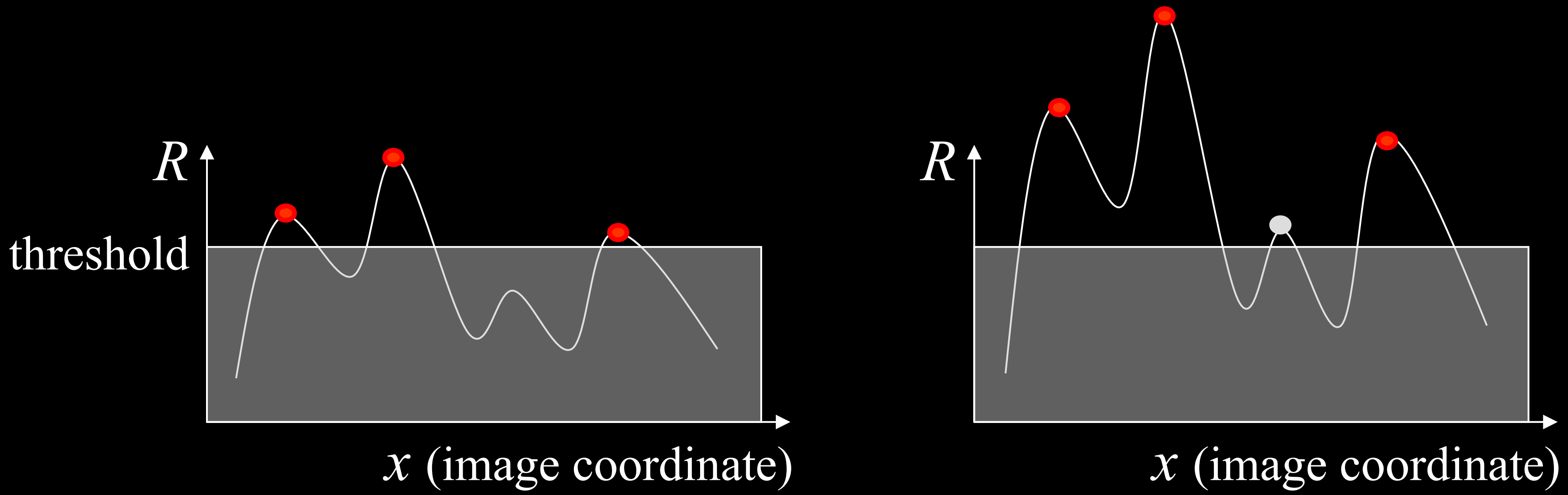
via Svetlana



$I \rightarrow a\,I + b$

Only derivatives are used, so invariant to intensity shift $I \rightarrow I + b$

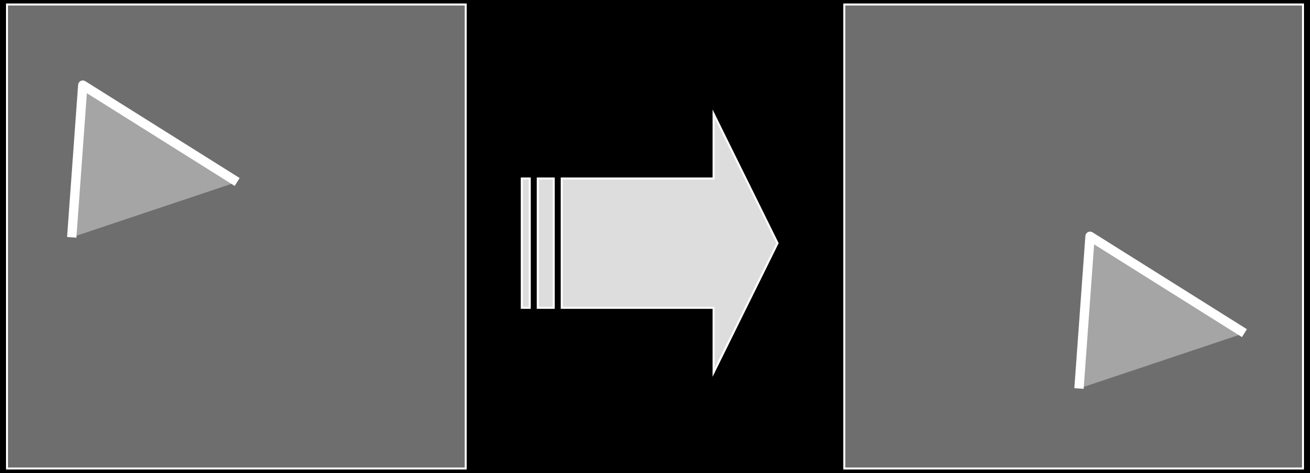
Intensity scaling: $I \rightarrow a\,I$



Partially invariant to affine intensity change

Image translation

via Svetlana

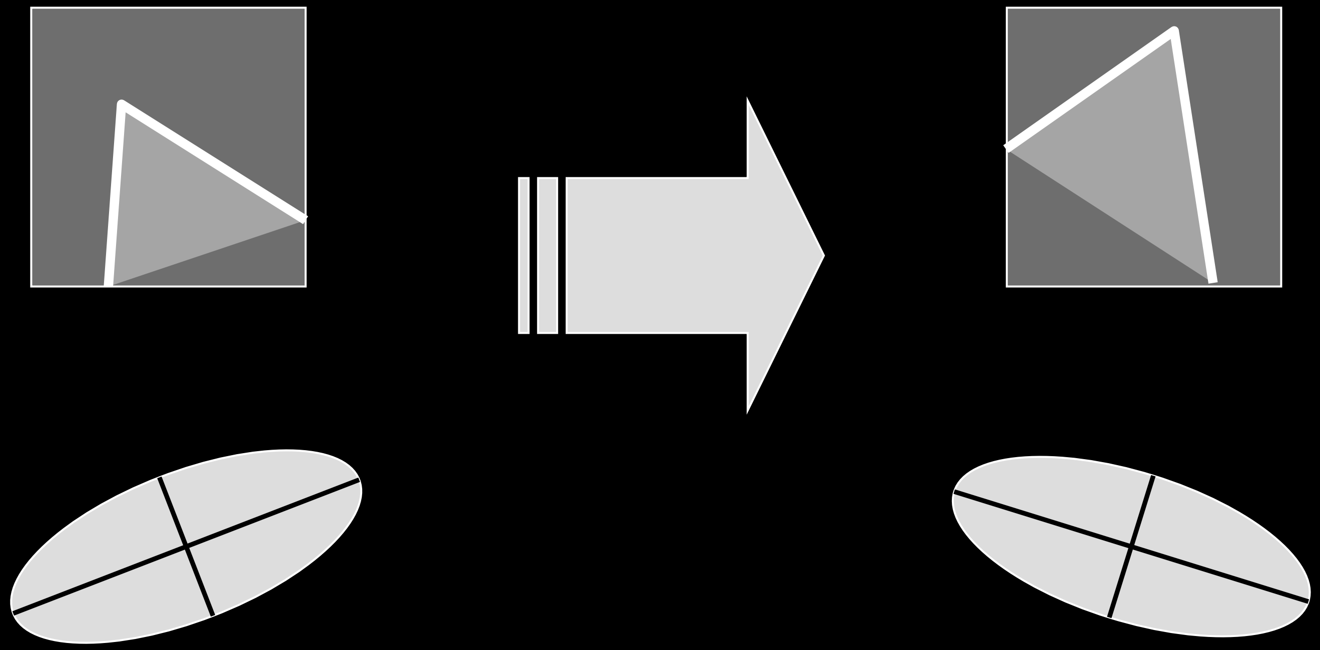


Derivatives and window function are shift-invariant

Corner location is *covariant* w.r.t. translation

via Svetlana

Image rotation

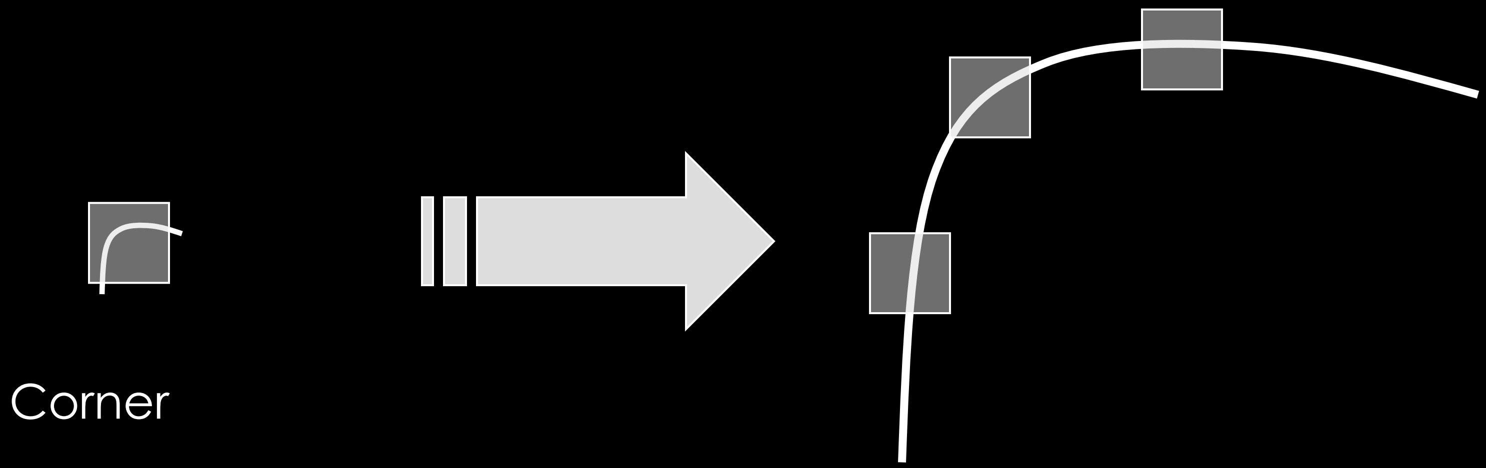


Second moment ellipse rotates but its shape (i.e. eigenvalues) remains the same

Corner location is covariant w.r.t. rotation

Scaling

via Svetlana



All points will be classified as edges

Corner location is not covariant w.r.t. scaling!

Scale & Rotation Covariance are Required for Long Range Matching

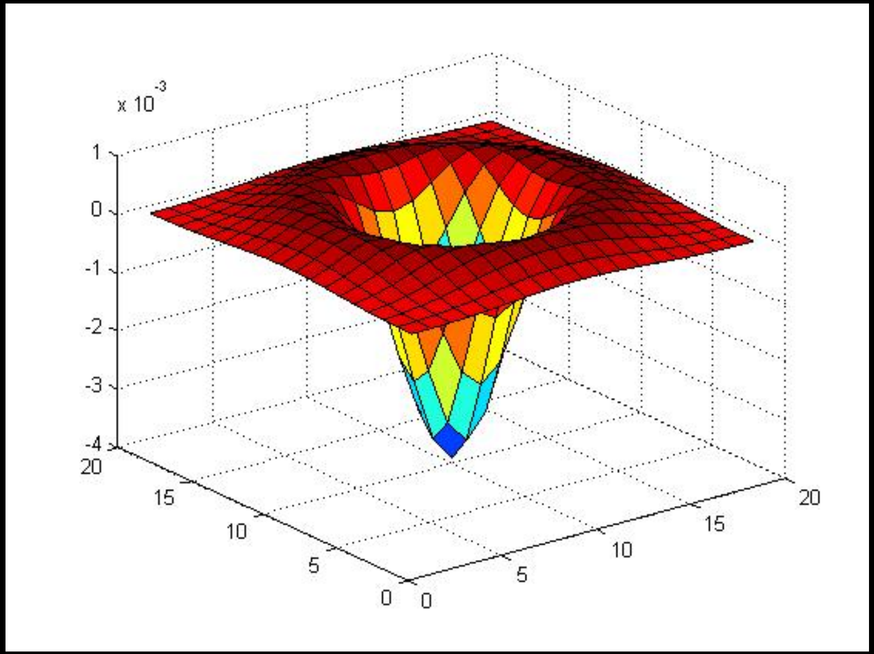
- Independently detect corresponding locations in scaled, rotated versions of an imaged scene

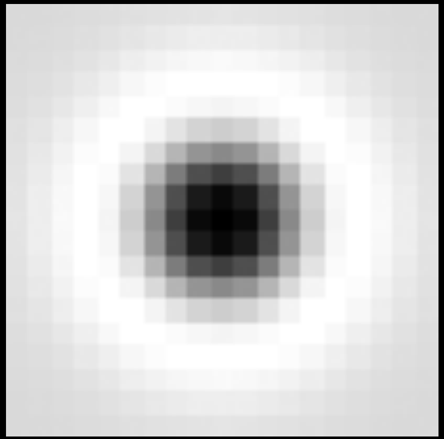
- Need scale covariant detector and rotation and scale normalization

via Svetlana

Blob detection in 2D

- Laplacian of Gaussian: Circularly symmetric operator for blob detection in 2D



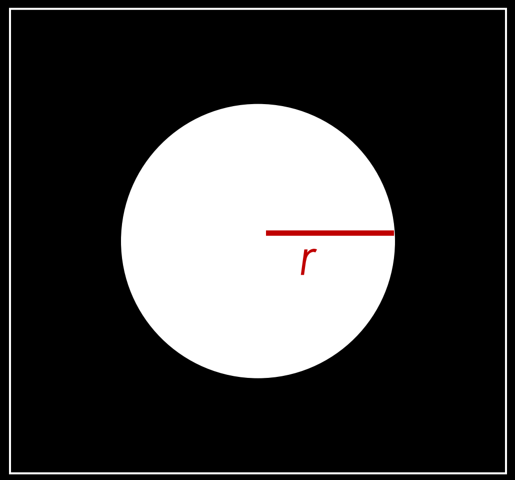


$$\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$$

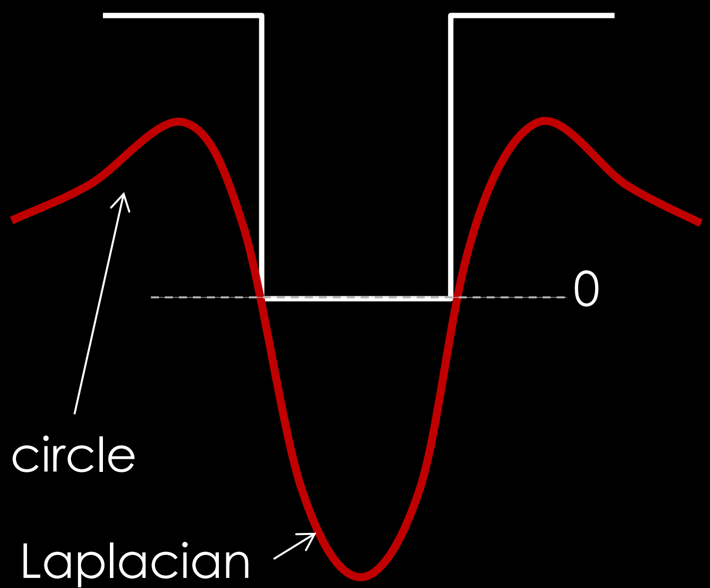
Scale selection

via Svetlana

- At what scale does the Laplacian achieve a maximum response to a blob of radius r?
- The Laplacian is given by (up to scale): $$(x^2 + y^2 - 2\sigma^2)e^{-(x^2+y^2)/2\sigma^2}$$
- Therefore, the maximum response occurs at $\sigma = r/\sqrt{2}$.



image

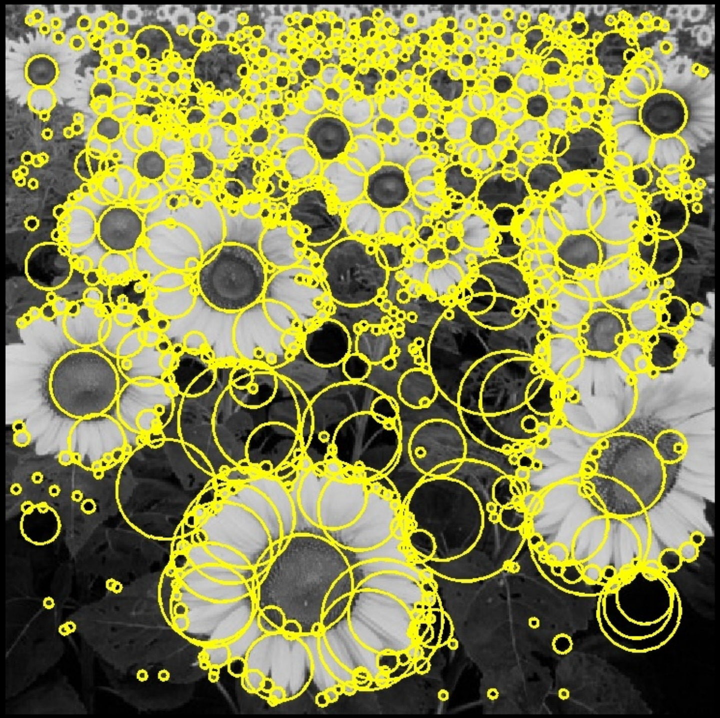


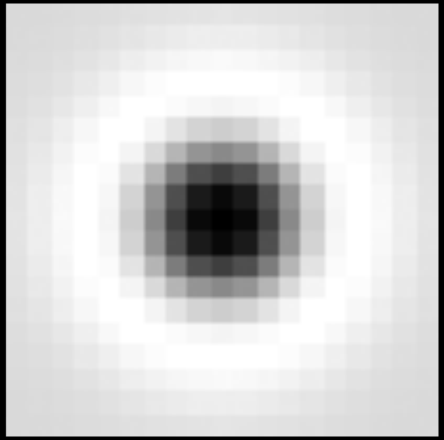
Basic idea

Convolve the image with a “blob filter” at multiple scales

and

Compute extrema of filter response in the resulting *scale space*

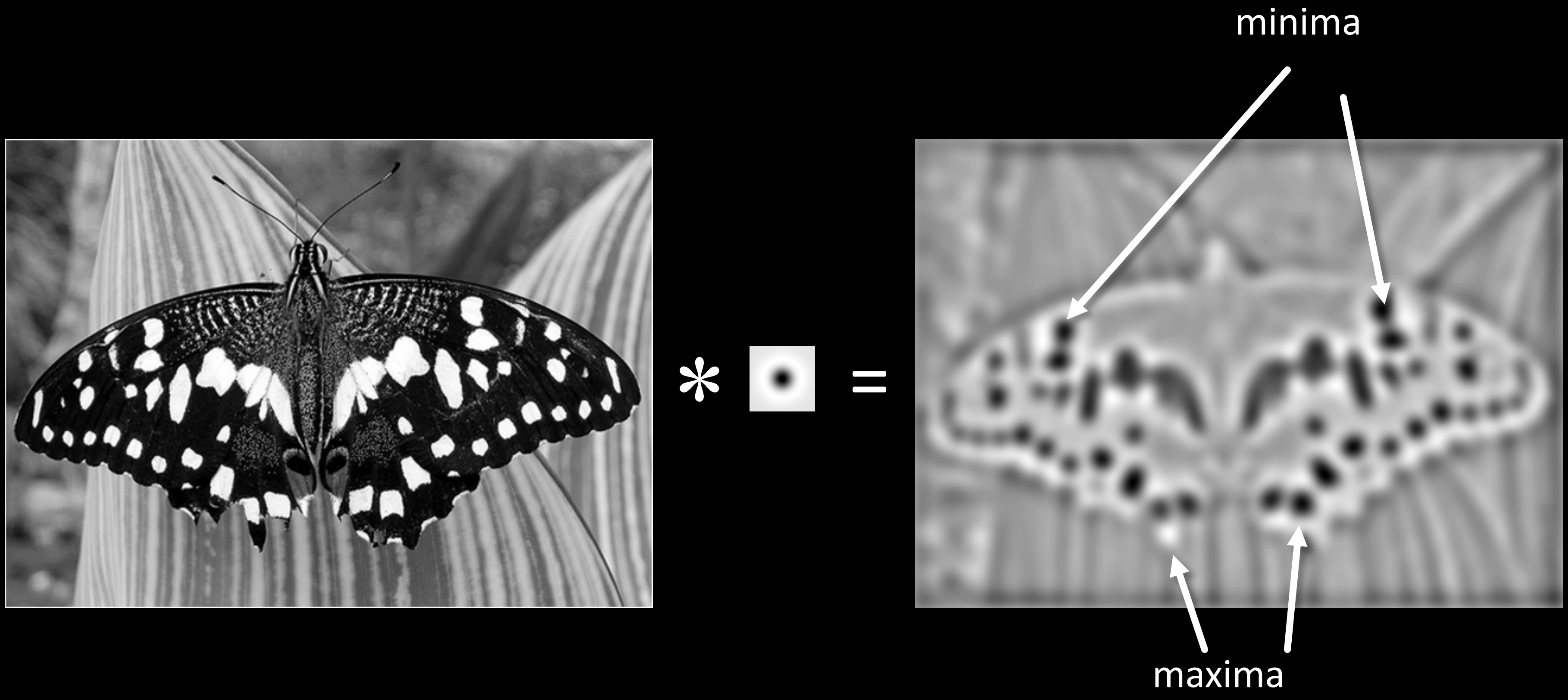




T. Lindeberg, Feature detection with automatic scale selection, *IJCV* 30(2), pp 77-116, 1998

via Svetlana

Blob detection

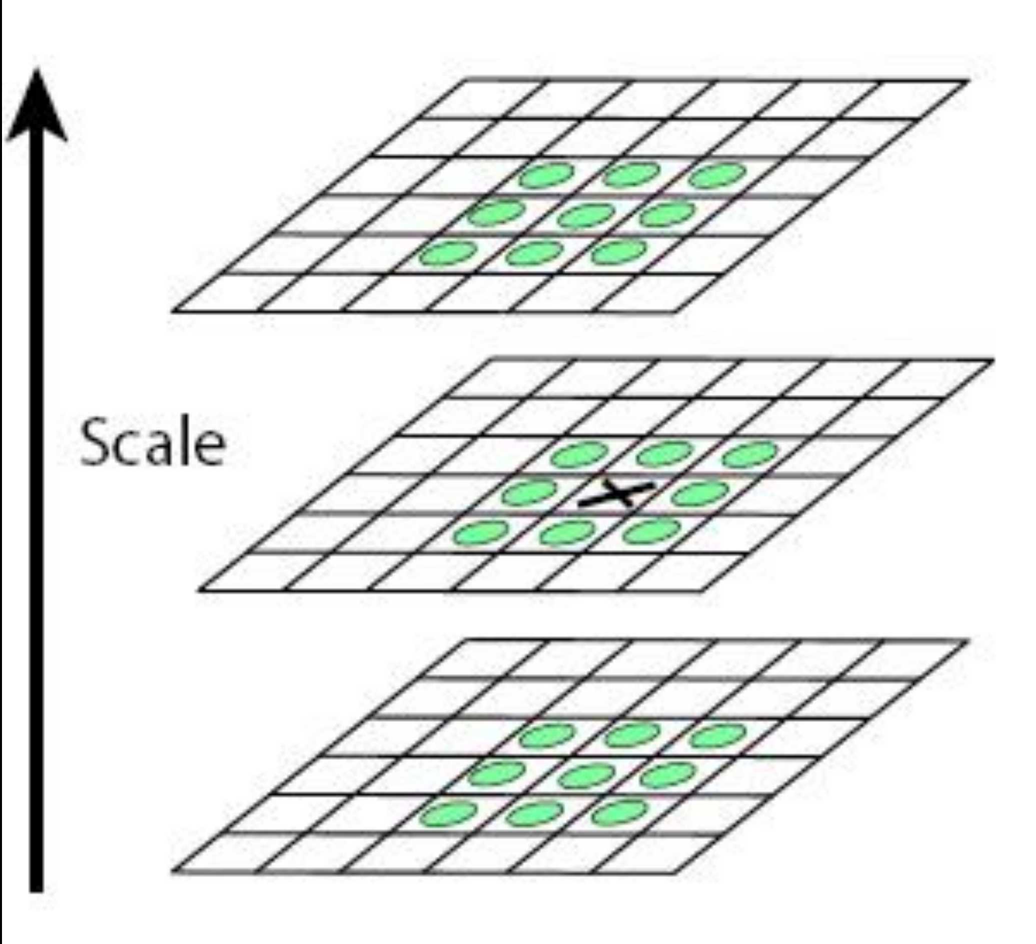


Find maxima *and minima* of blob filter response in space *and scale*

Source: N. Snavely

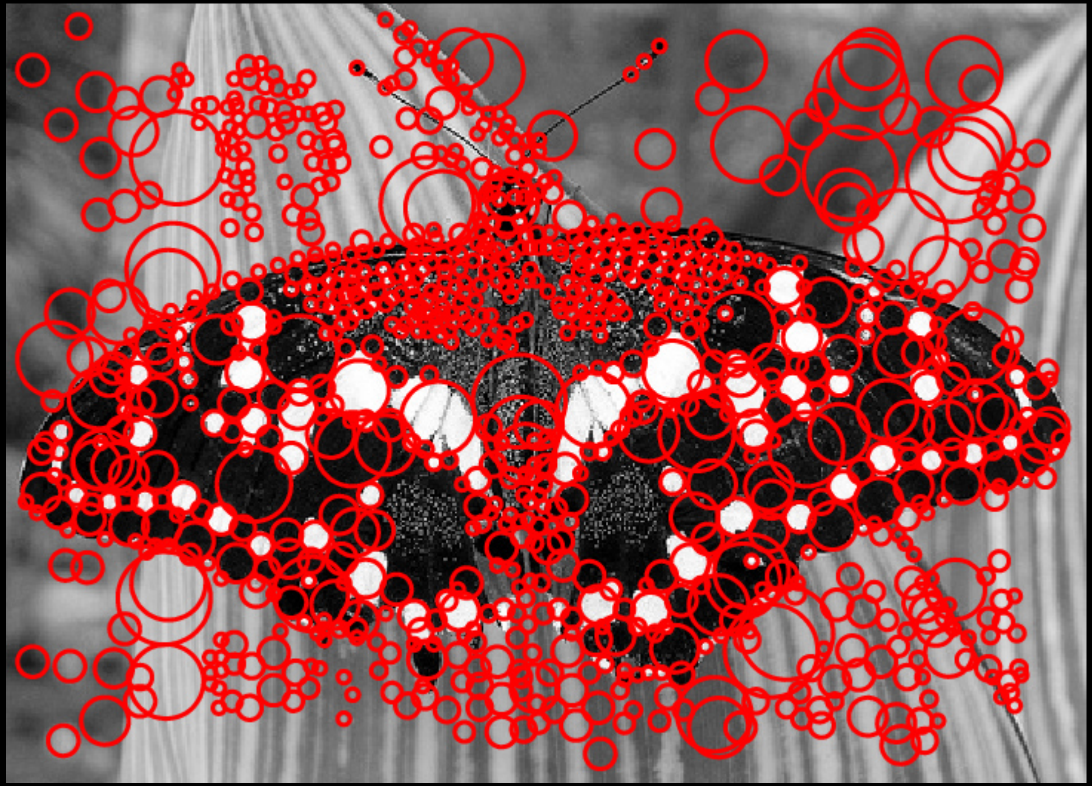
Scale-space blob detector

1. Convolve image with scale-normalized Laplacian at several scales
2. Find maxima of squared Laplacian response in scale-space



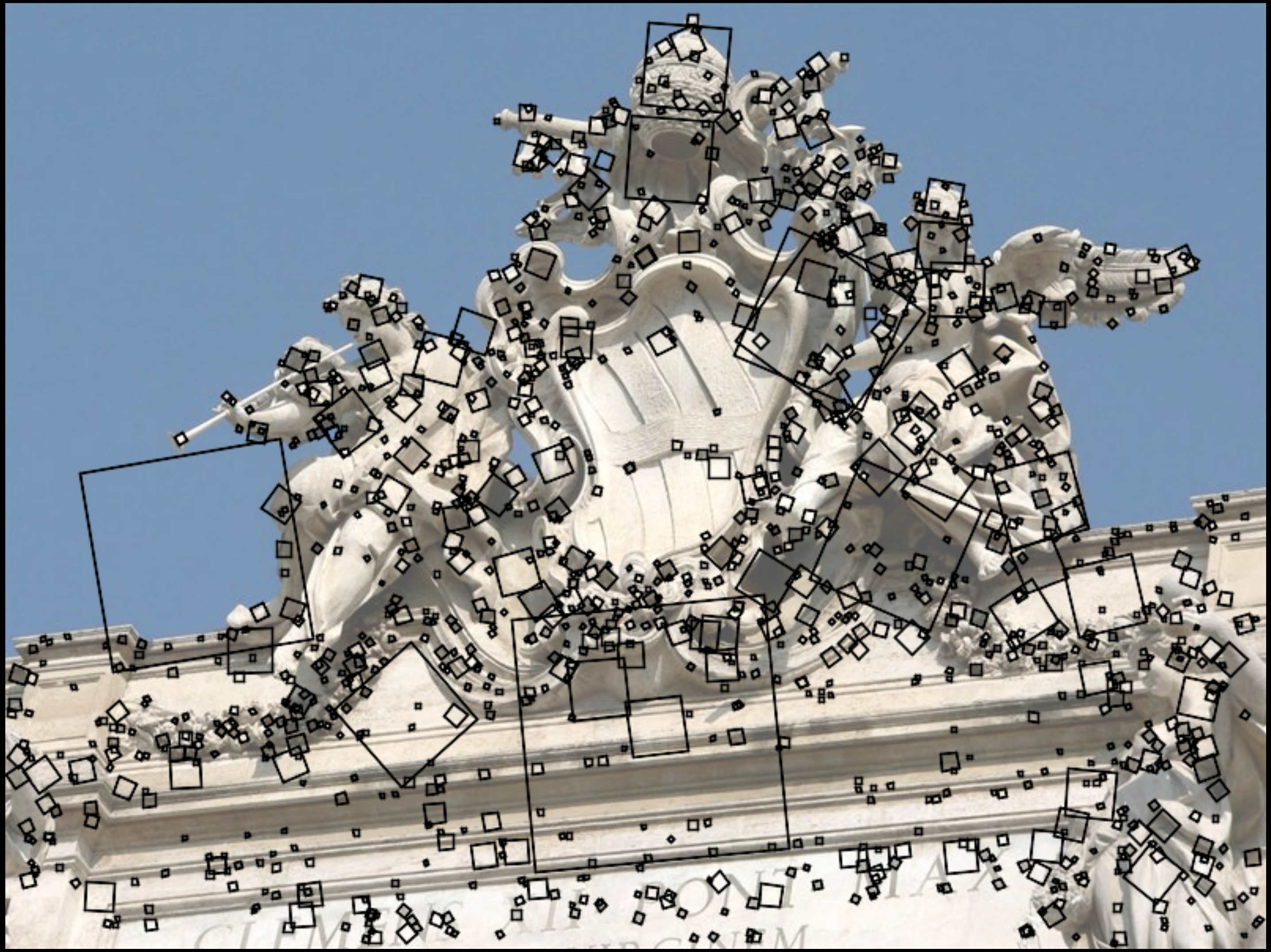
Scale-space blob detector: Example

via Svetlana



via Svetlana

Basis for SIFT Keypoint Detection



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

Efficient implementation

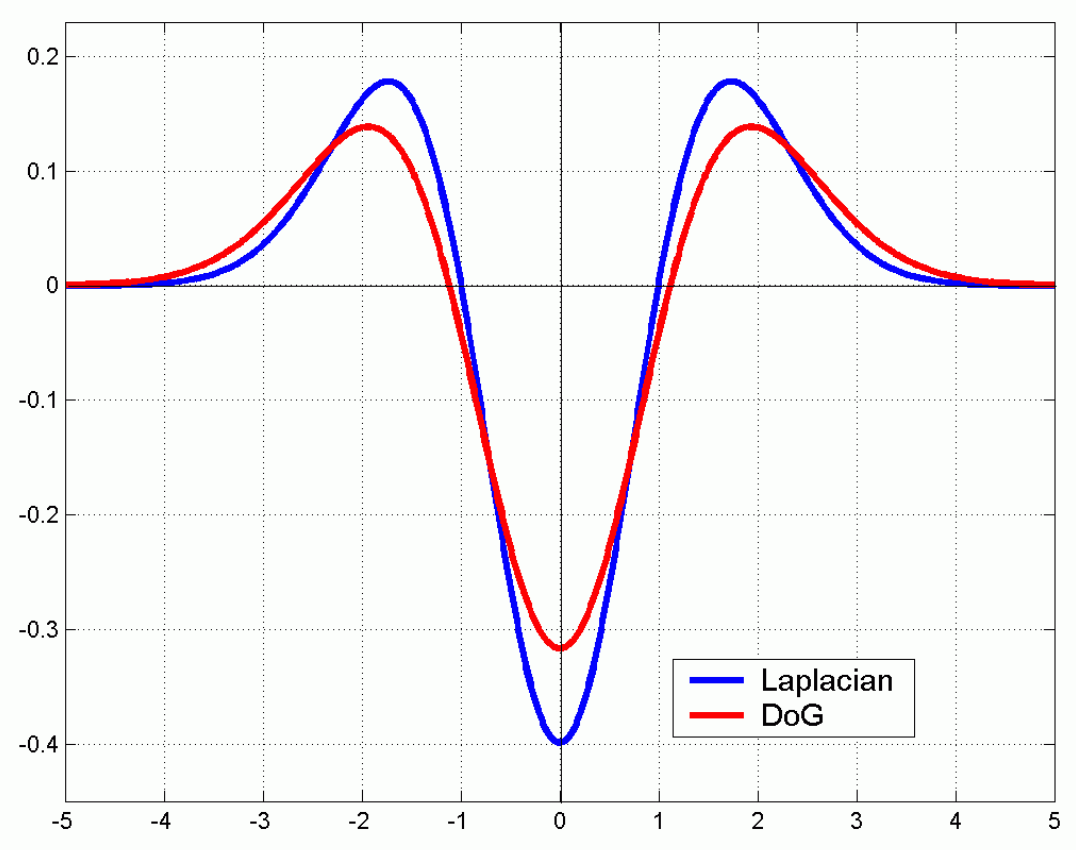
Approximating the Laplacian with a difference of Gaussians:

$$L = \sigma^2 \left(G_{xx}(x, y, \sigma) + G_{yy}(x, y, \sigma) \right)$$

(Laplacian)

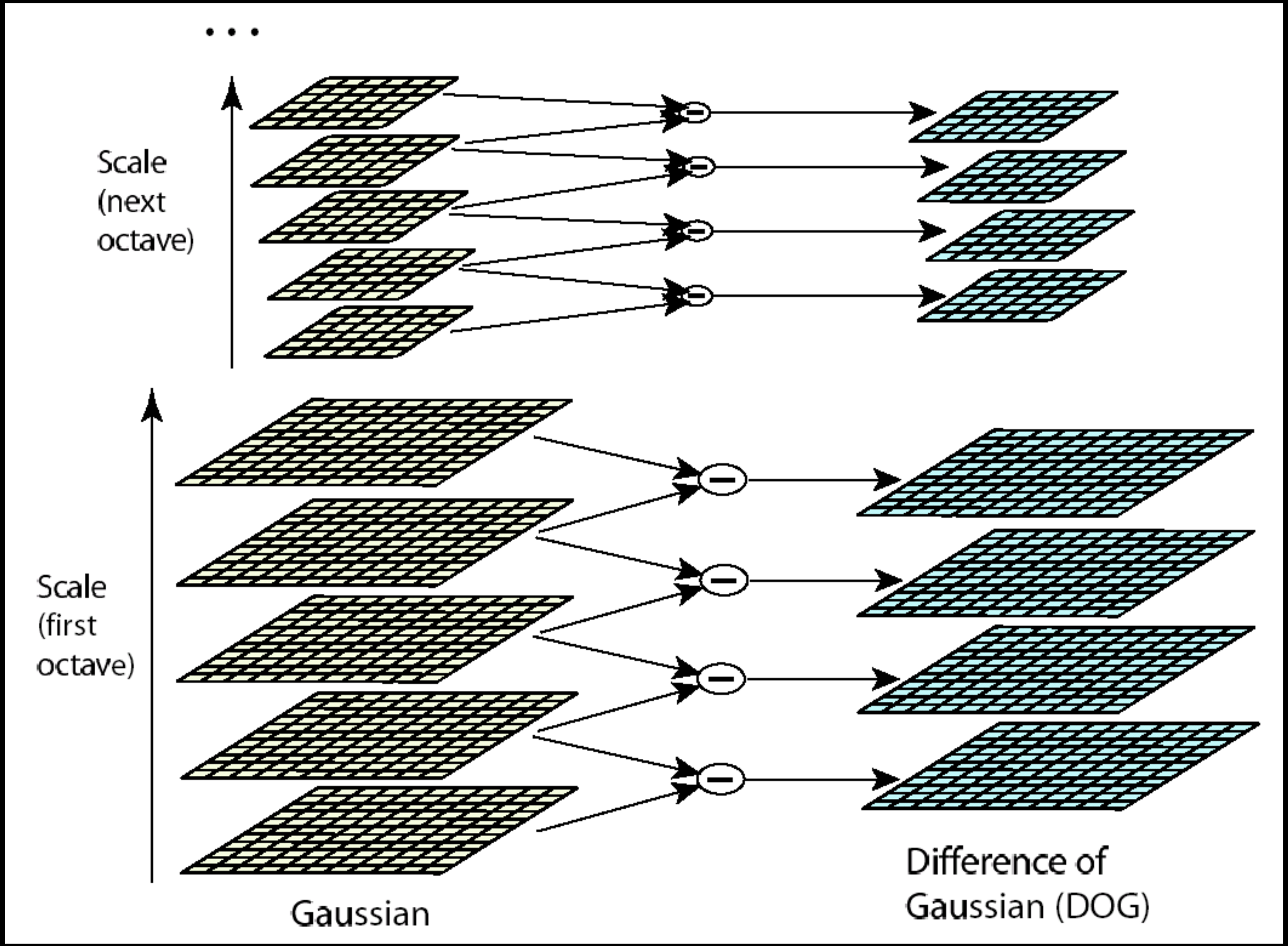
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)



Efficient implementation

via Svetlana

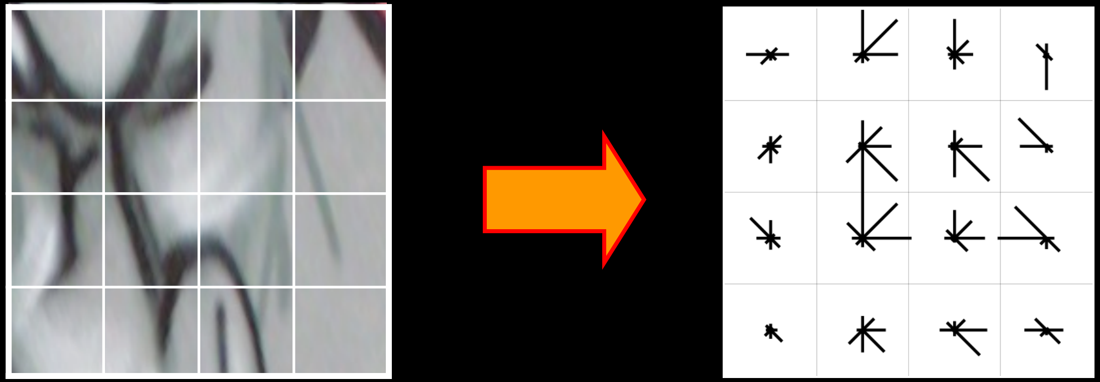


David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

via Svetlana

Feature descriptors: SIFT

- 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



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

Feature descriptors: SIFT

via Svetlana

- 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
- 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

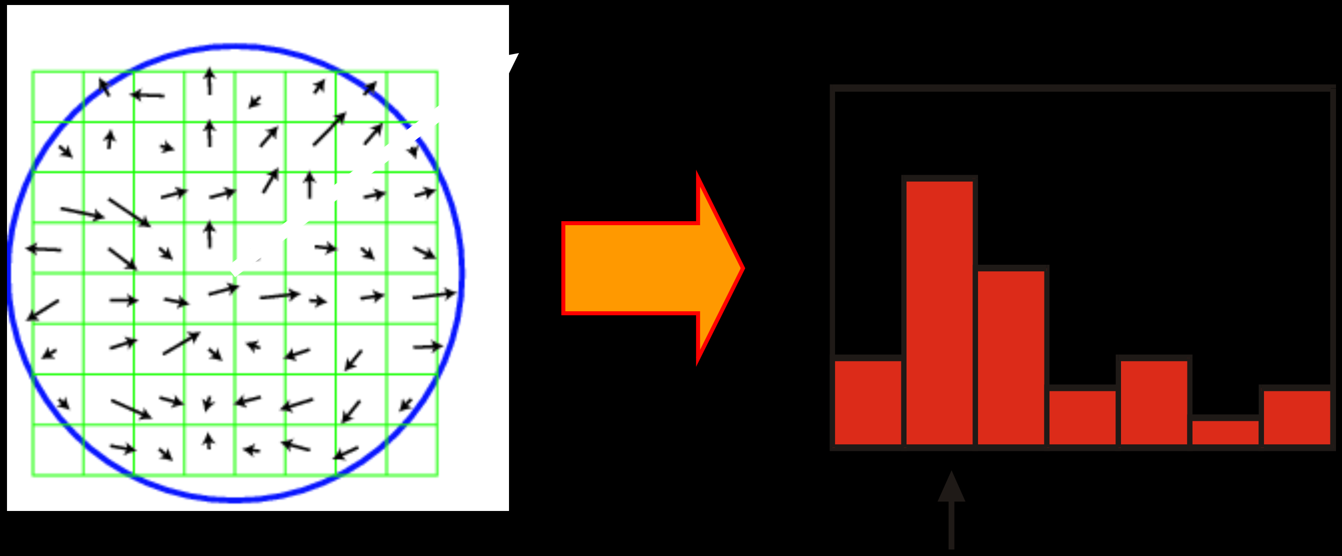
David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

via Svetlana

Rotational Normalization of SIFT Feature

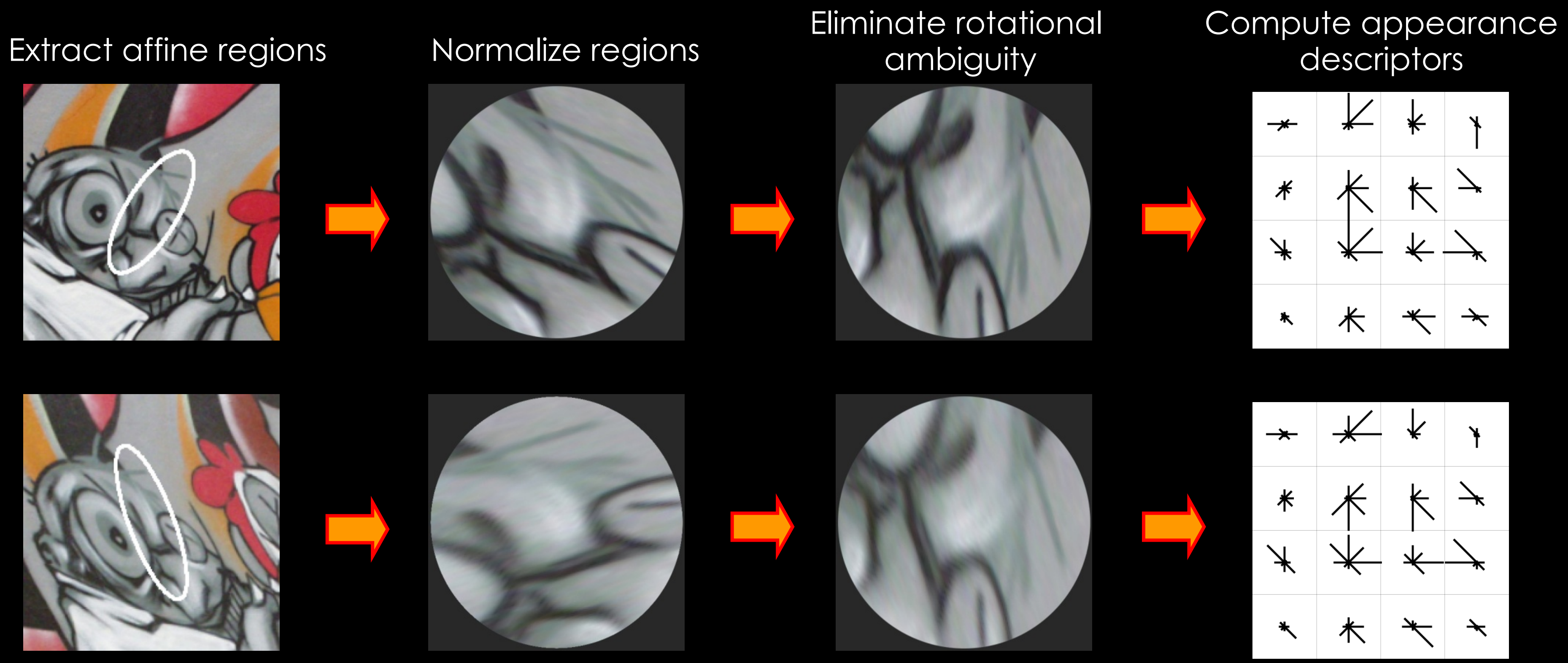
To assign a unique orientation to circular image windows:

- Create histogram of local gradient directions in the patch
- Assign canonical orientation at peak of smoothed histogram



via Svetlana

Normalization: From covariant regions to invariant features

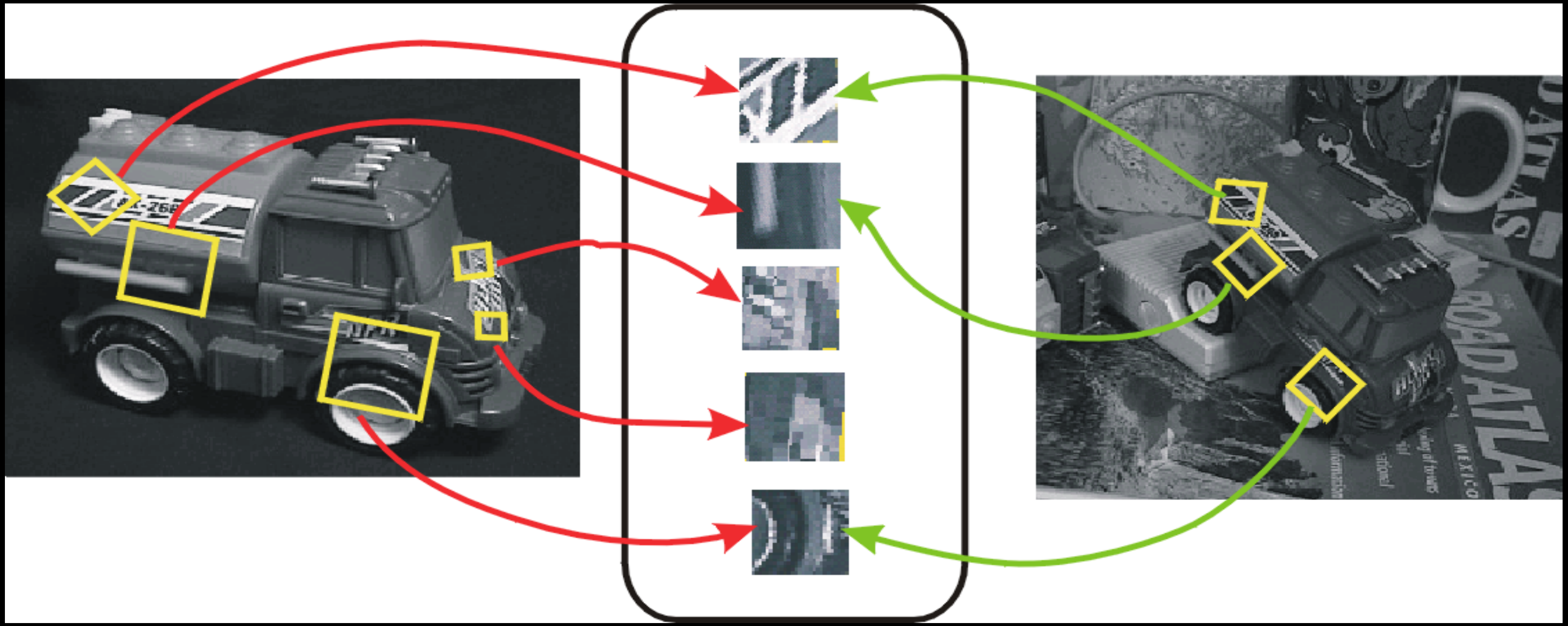


via Svetlana

Invariance vs. covariance

- **Invariance:**
  - features(transform(image)) = features(image)

- **Covariance:**
  - features(transform(image)) = transform(features(image))



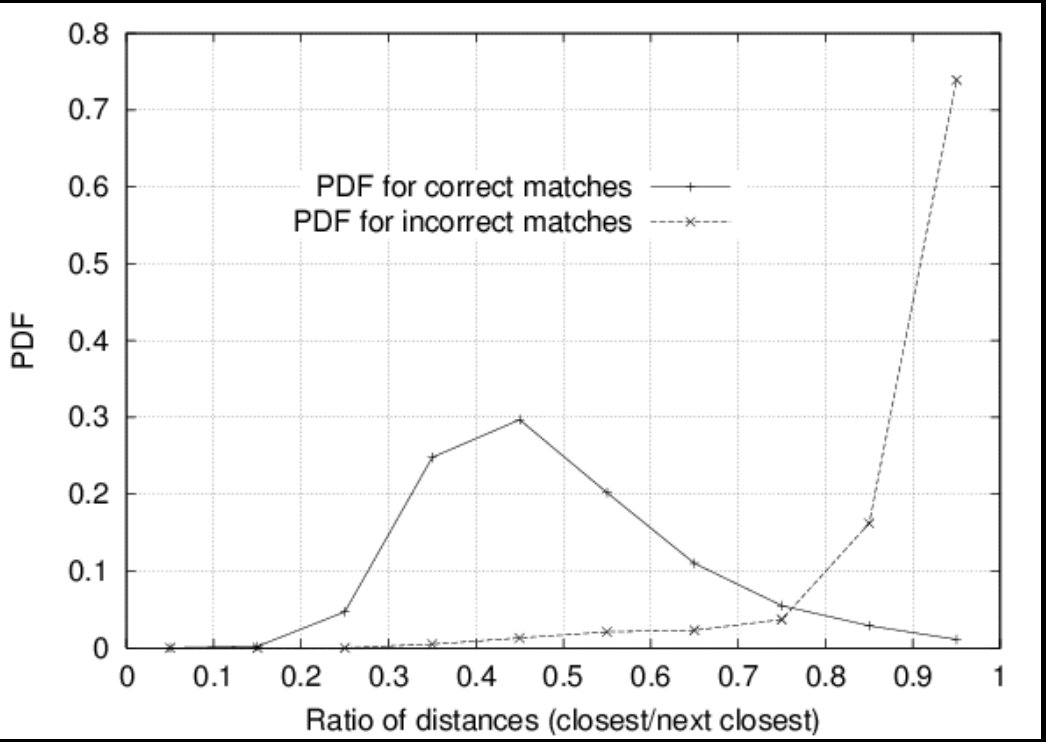
Covariant detection => invariant description

Problem: Ambiguous putative matches

Source: Y. Furukawa

Rejection of unreliable matches

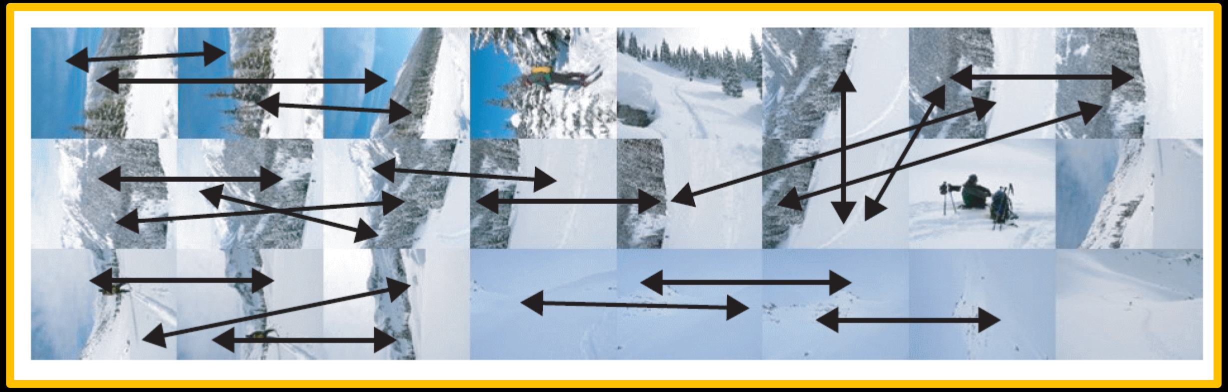
- How can we tell which putative matches are more reliable?
- Heuristic: compare distance of **nearest** neighbor to that of **second** nearest neighbor
  - Ratio of closest distance to second-closest distance will be *high* for features that are *not* distinctive

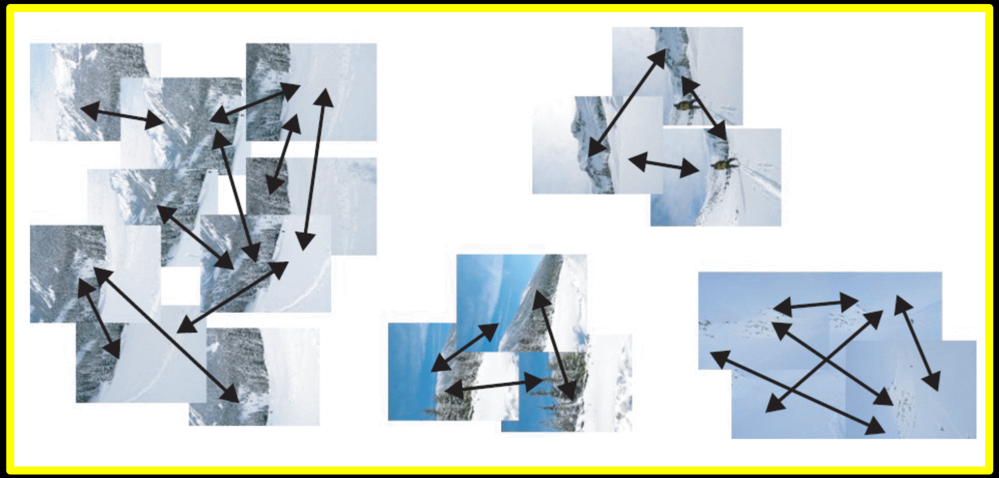


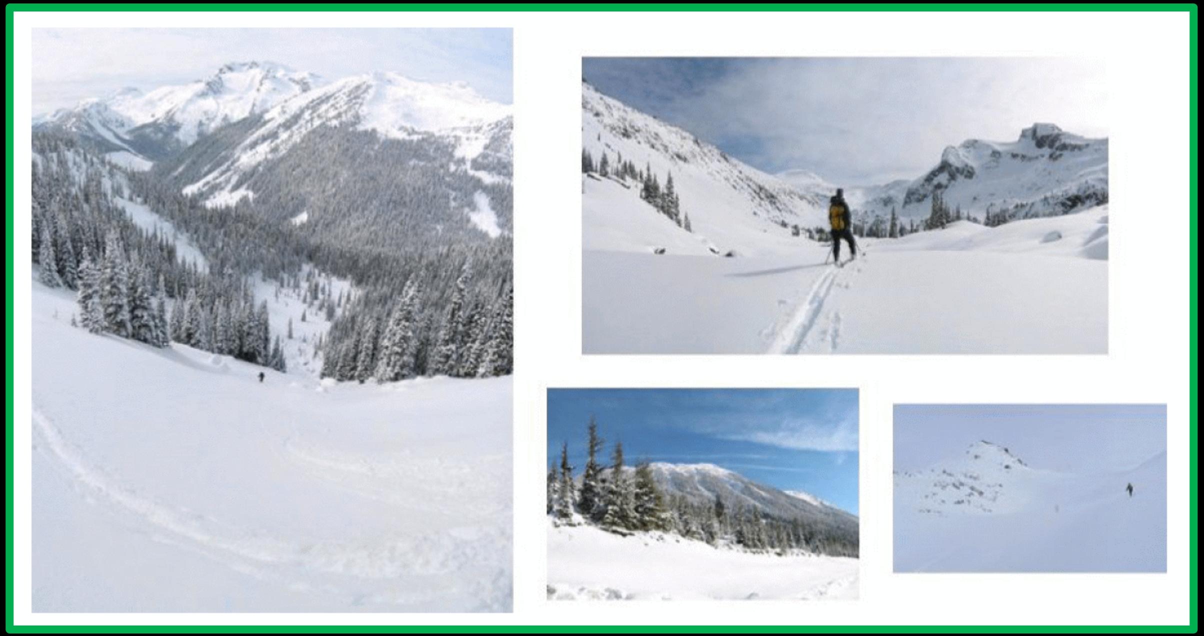
Threshold of 0.8 provides good separation

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

Application: Panoramas from a Jumble of Pictures







- Extract SIFT features from all images
- Find Pairwise Homographies
- Find Connected Components over the pair connections
- Bundle adjust the connected component to find image to panorama transformation
- Render panorama with blending

Application: Scalable Images based Search

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

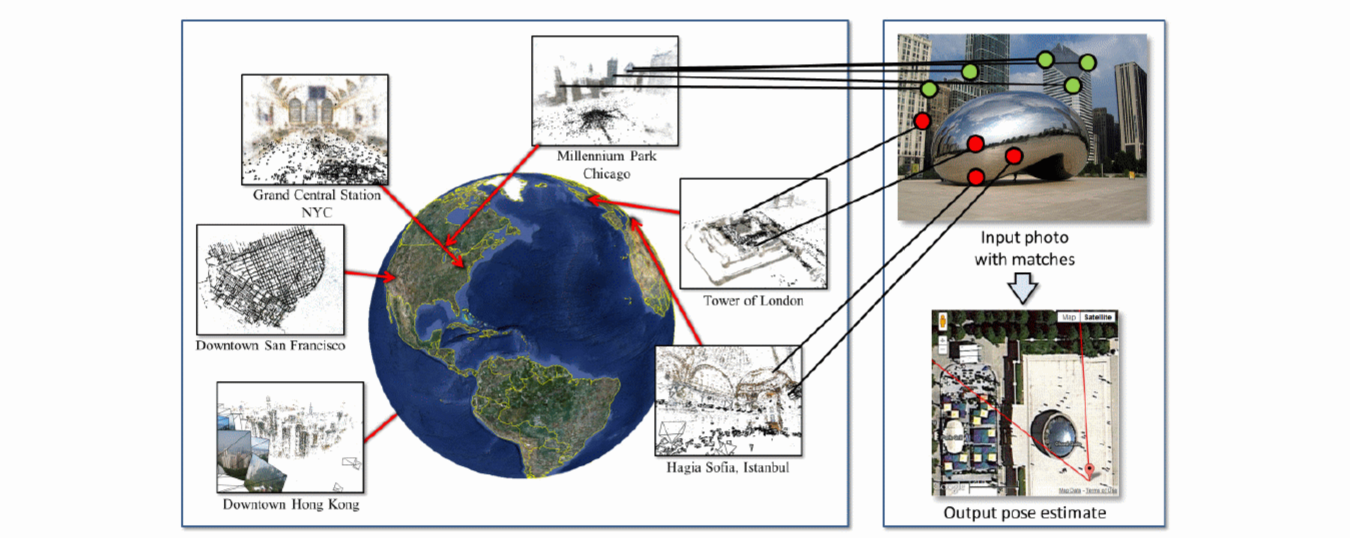


Fig. 1. *A worldwide point cloud database.* In order to compute the pose of a query image, we match it to a database of georeferenced structure from motion point clouds assembled from photos of places around the world. Our database (left) includes a street view image database of downtown

Worldwide Pose Estimation
Yunpeng Li, Noah Snavely, Dan Huttenlocher, Pascal Fua
ECCV 2012

- Find location of a Query image by matching against a large database of images indexed with their respective locations

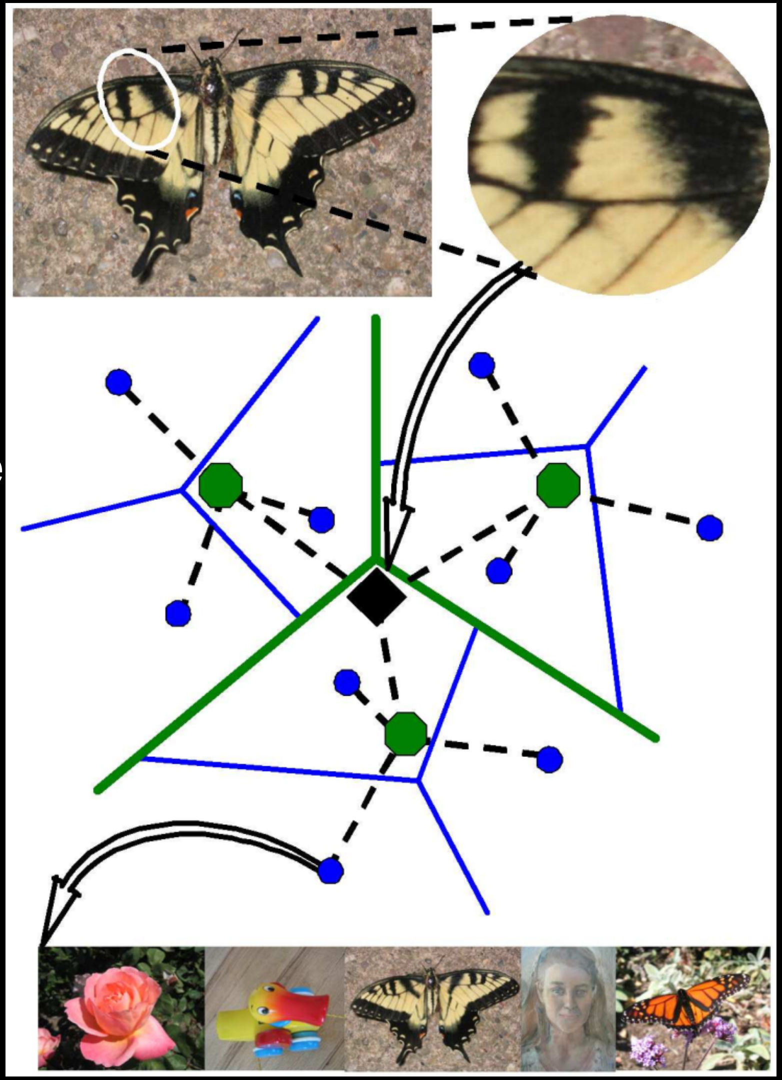
- Find instances of objects / images in a database of images

Efficient indexing technique: Vocabulary trees

Test image

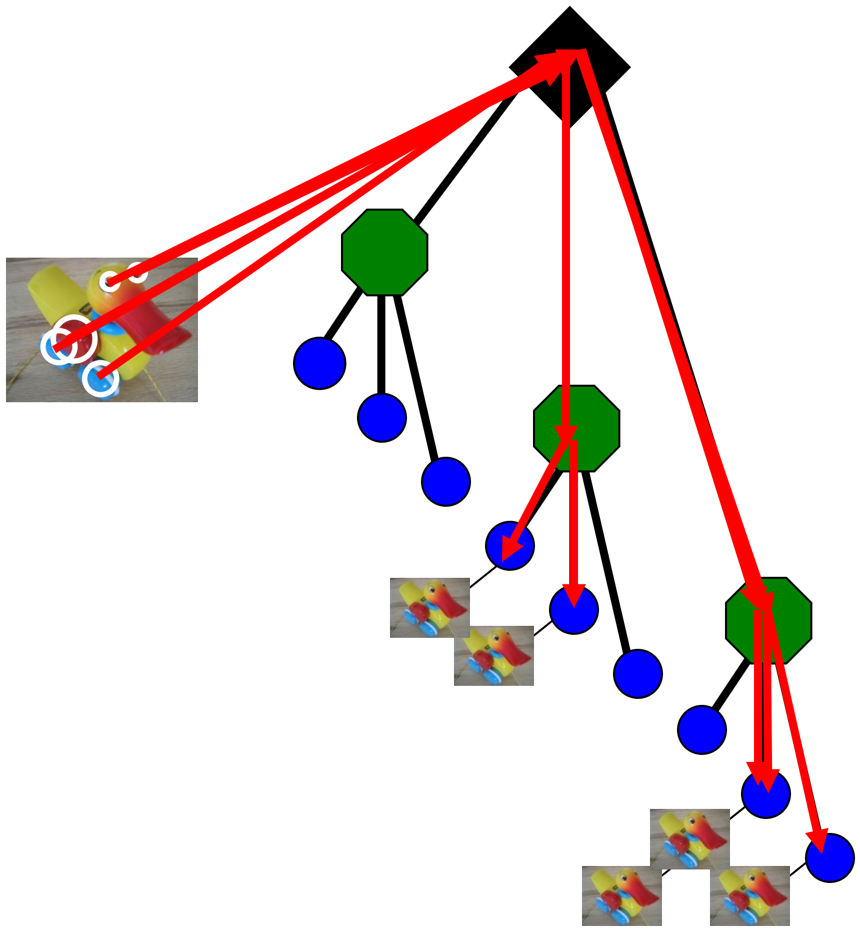
Vocabulary tree with inverted index

Database



D. Nistér and H. Stewénius, Scalable Recognition with a Vocabulary Tree, CVPR 2006

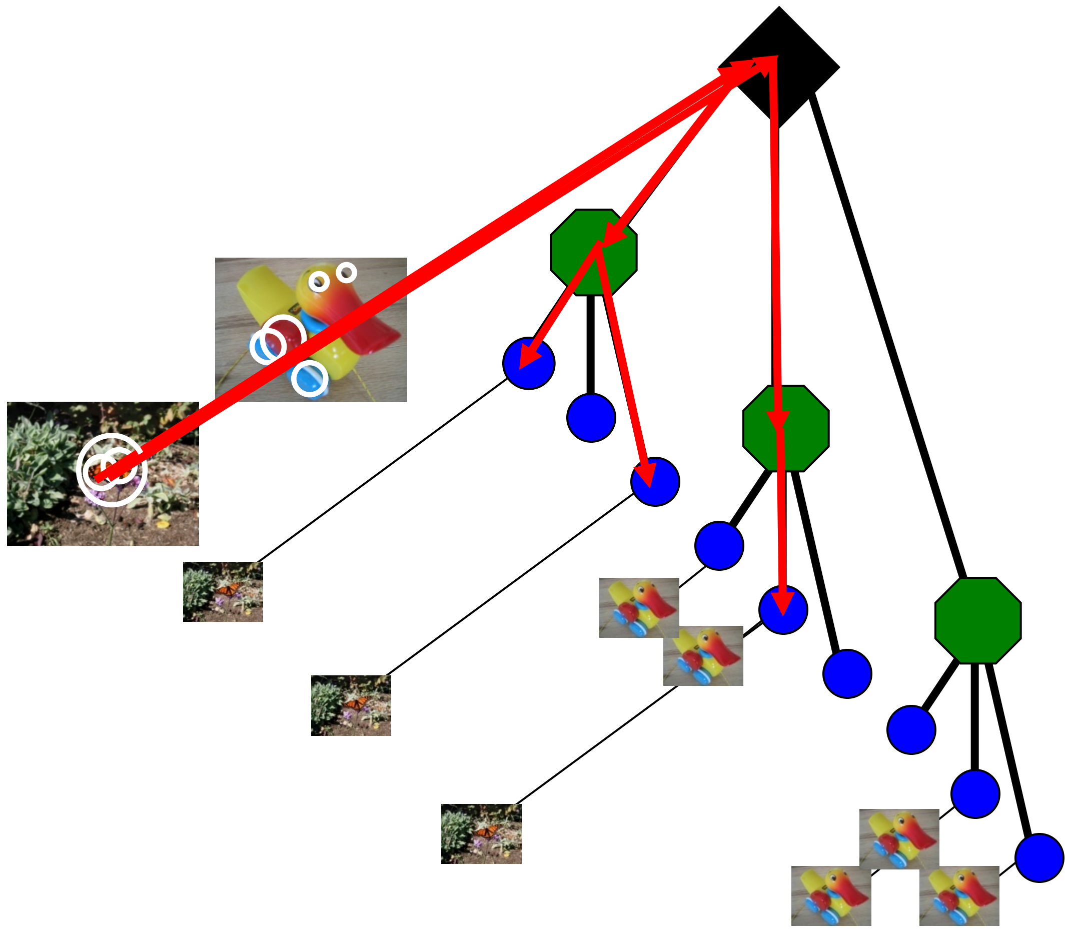
Model images



Populating the vocabulary tree/inverted index

Slide credit: D. Nister

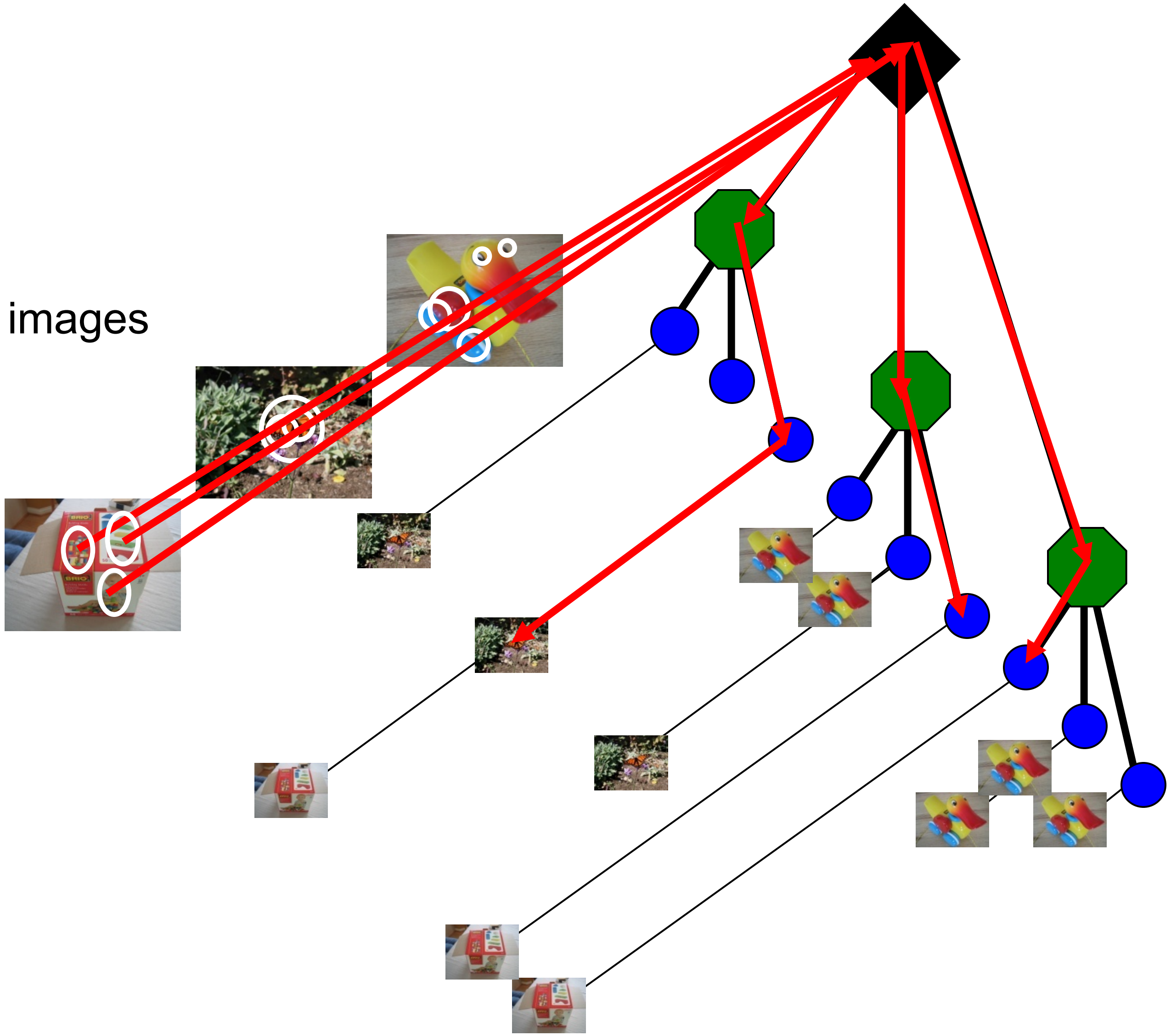
Model images



Populating the vocabulary tree/inverted index

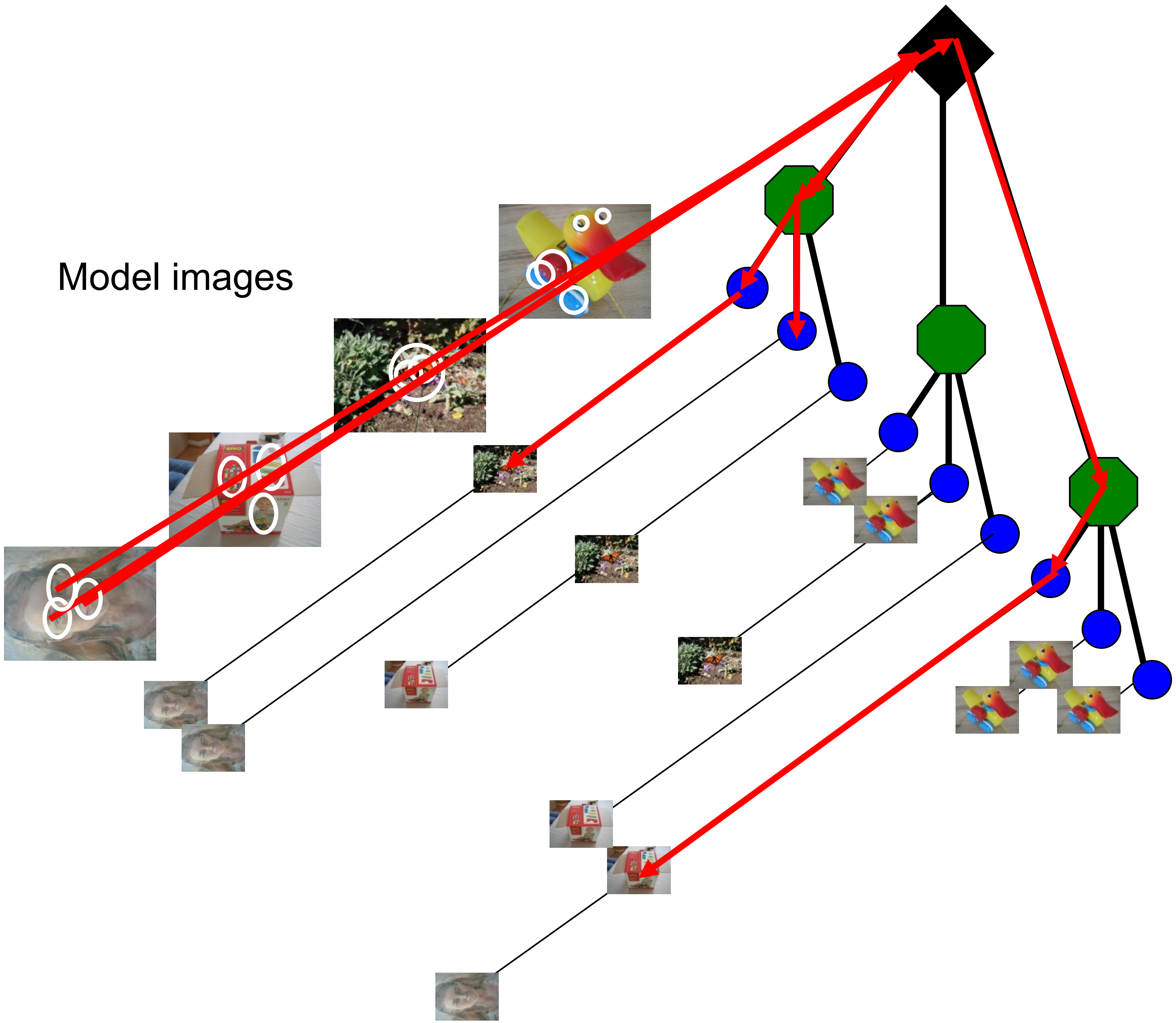
Slide credit: D. Nister

Model images



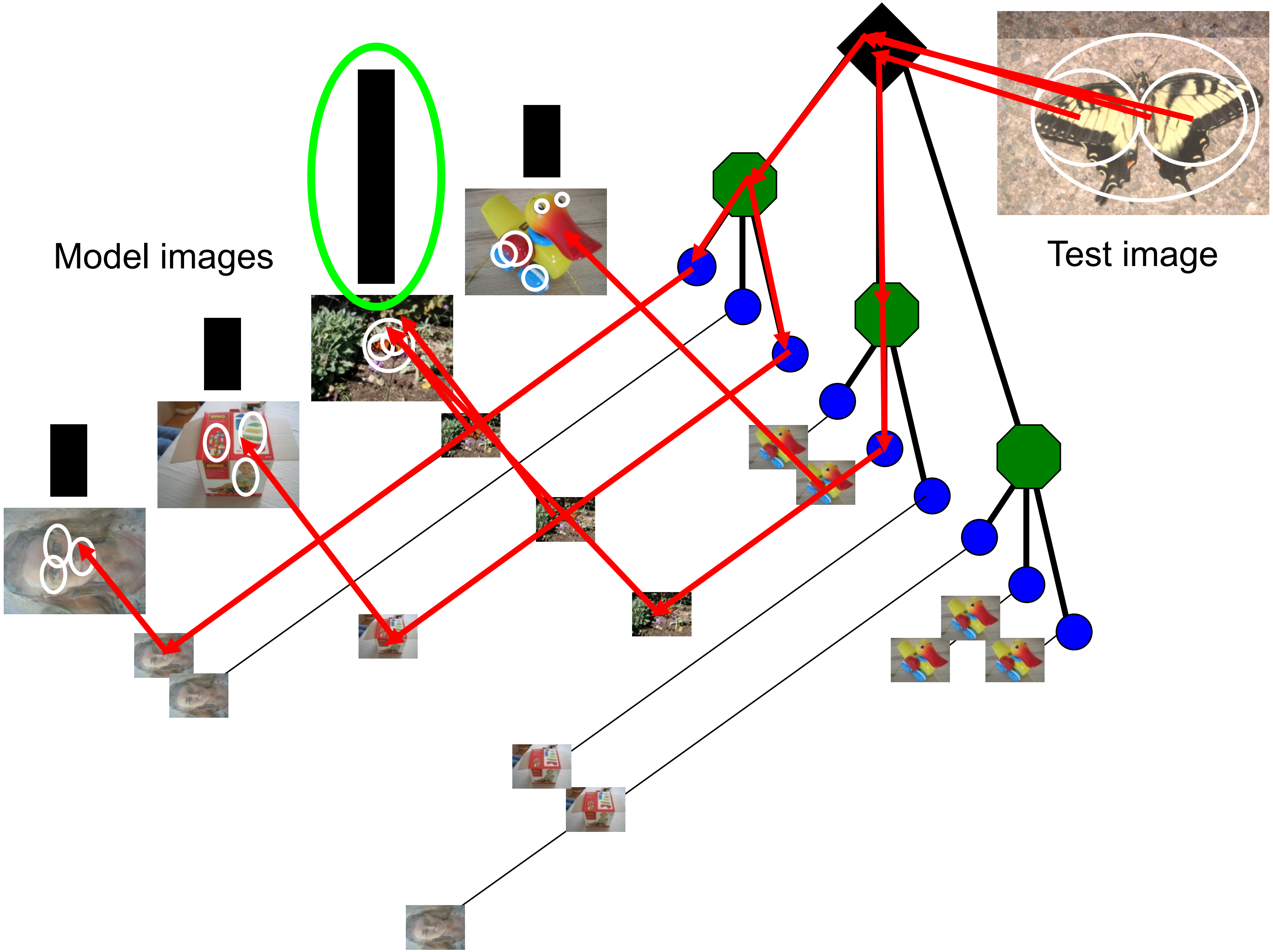
Populating the vocabulary tree/inverted index

Slide credit: D. Nister



Populating the vocabulary tree/inverted index

Slide credit: D. Nister

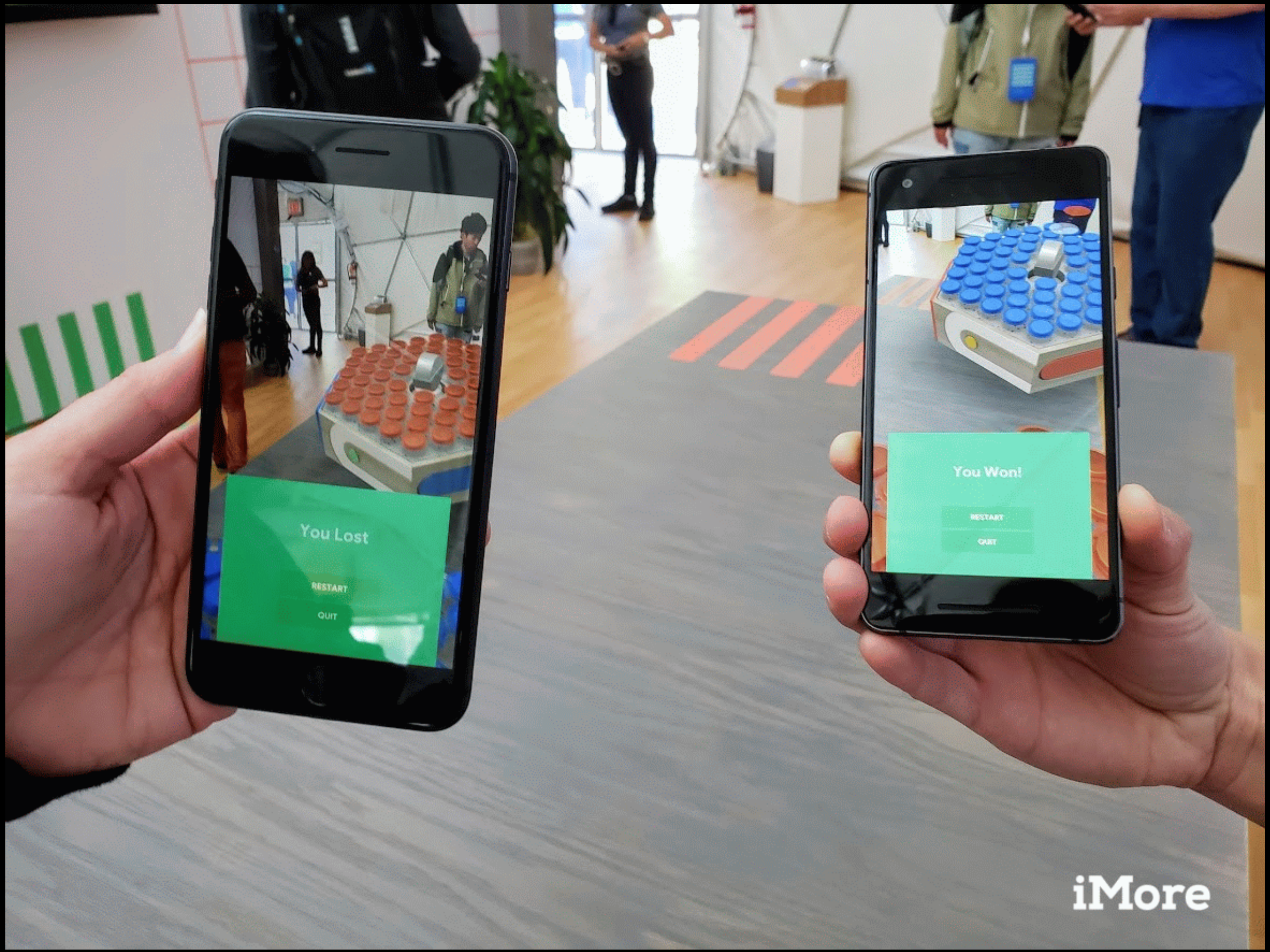


Looking up a test image

Slide credit: D. Nister

Google Cloud Anchors

- Google 'Cloud Anchors' will help synchronize group AR experiences across iOS and Android devices
- Employ Wide Baseline Matching
- https://mediafocus.biz/google-cloud-anchors-will-help-synchronize-group-ar-experiences-across-ios-and-android-devices/



Minecraft Earth via Azure Spatial Anchors

https://www.theverge.com/2019/11/12/20961639/minecraft-earth-now-available-early-access-us-ios-android

https://youtu.be/AQEizp-VrVU

Spatial Anchors are built on Wide Baseline Matching and SfM / SLAM

How do Hand-Crafted Features Compare with Learned Features?

Comparative Evaluation of Hand-Crafted and Learned Local Features

Johannes L. Schönberger[1] Hans Hardmeier[1] Torsten Sattler[1] Marc Pollefeys[1,2]
[1] Department of Computer Science, ETH Zürich [2] Microsoft Corp.
{jsch,harhans,sattlert,pomarc}@inf.ethz.ch

CVPR 2017

- “Hand-crafted features still perform on par or better than recent learned features for image-based reconstruction.
- The current generation of learned descriptors shows a high variance across different datasets and applications.
- The next generation of learned descriptors needs more training data.”

How do Hand-Crafted Features Compare with Learned Features?

Image Matching across Wide Baselines: From Paper to Practice

Yuhe Jin[1] Dmytro Mishkin[2] Anastasiia Mishchuk[3]
Jiří Matas[2] Pascal Fua[3] Kwang Moo Yi[1] Eduard Trulls[4]
[1]University of Victoria [2]Czech Technical University in Prague [3]École Polytechnique Fédérale de Lausanne [4]Google Research

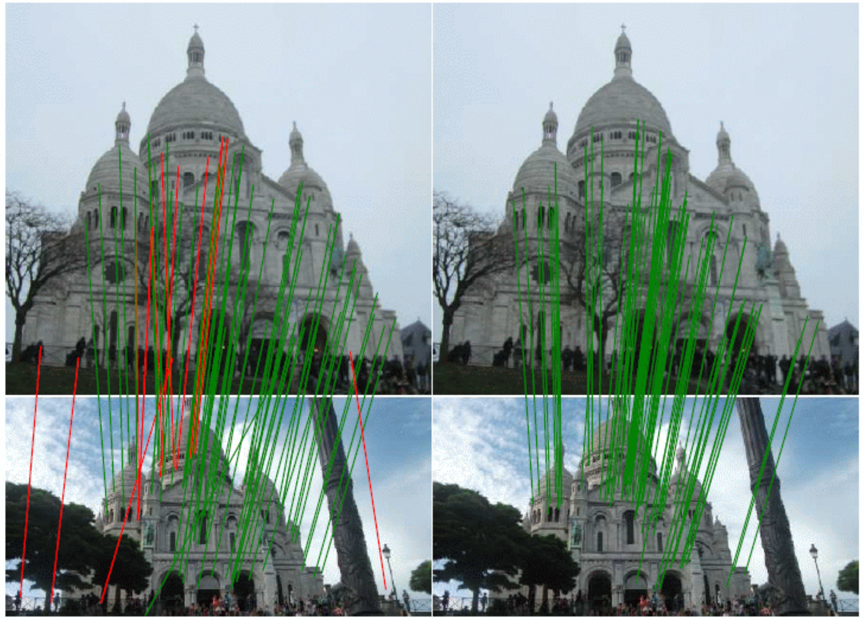


Figure 1. Every paper claims to outperform the state of the art. Is this possible, or an artifact of insufficient validation? On the left, we show stereo matches obtained with **D2-Net** (2019) [33], a state-of-the-art local feature, using OpenCV RANSAC with its default settings. On the right, we show **SIFT** (1999) [48] with a carefully tuned MAGSAC [29] – notice how the latter performs much better. We fill this gap with a new, modular benchmark for sparse image matching, with dozens of built-in methods.

Contributions

- Dataset with 30k images with depth maps and ground truth poses
- A modular pipeline incorporating dozens of methods for feature extraction and matching, and pose estimation
- Two downstream tasks – stereo and multi-view reconstruction – evaluated with downstream and intermediate metrics
- A thorough study of dozens of methods and techniques, hand-crafted and learned, and their combination, along with a procedure for hyper-parameter selection

“Hot Topic”

Image Matching: Local Features & Beyond

CVPR 2020 Workshop

Viewpoint and Illumination Variations Dataset

HPatches: A benchmark and evaluation of handcrafted and learned local descriptors

Vassileios Balntas*
Imperial College London
v.balntas@imperial.ac.uk

Karel Lenc*
University of Oxford
karel@robots.ox.ac.uk

Andrea Vedaldi
University of Oxford
vedaldi@robots.ox.ac.uk

Krystian Mikolajczyk
Imperial College London
k.mikolajczyk@imperial.ac.uk

https://github.com/hpatches/hpatches-dataset

Figure 1. Examples of image sequences; note the diversity of scenes and nuisance factors, including viewpoint, illumination, focus, reflections and other changes.

- Reproducible, patch-based: Descriptor evaluation should be done on patches to eliminate the detector related factors.
- Diverse: Representative of many different scenes and image capturing conditions.
- Real: Real data more challenging than a synthesized one due to nuisance factors that cannot be modelled in image transformations.
- Large: For accurate and stable evaluation; to provide substantial training sets for learning based descriptors.
- Multitask: Use cases, from matching image pairs to image retrieval.

A Contemporary Example of Learned Features

SuperPoint: Self-Supervised Interest Point Detection and Description

CVPR 2018

- Self-supervised framework for training interest point detectors and descriptors

- Fully-convolutional model operates on full-sized images and jointly computes pixel-level interest point locations and associated descriptors in one forward pass.

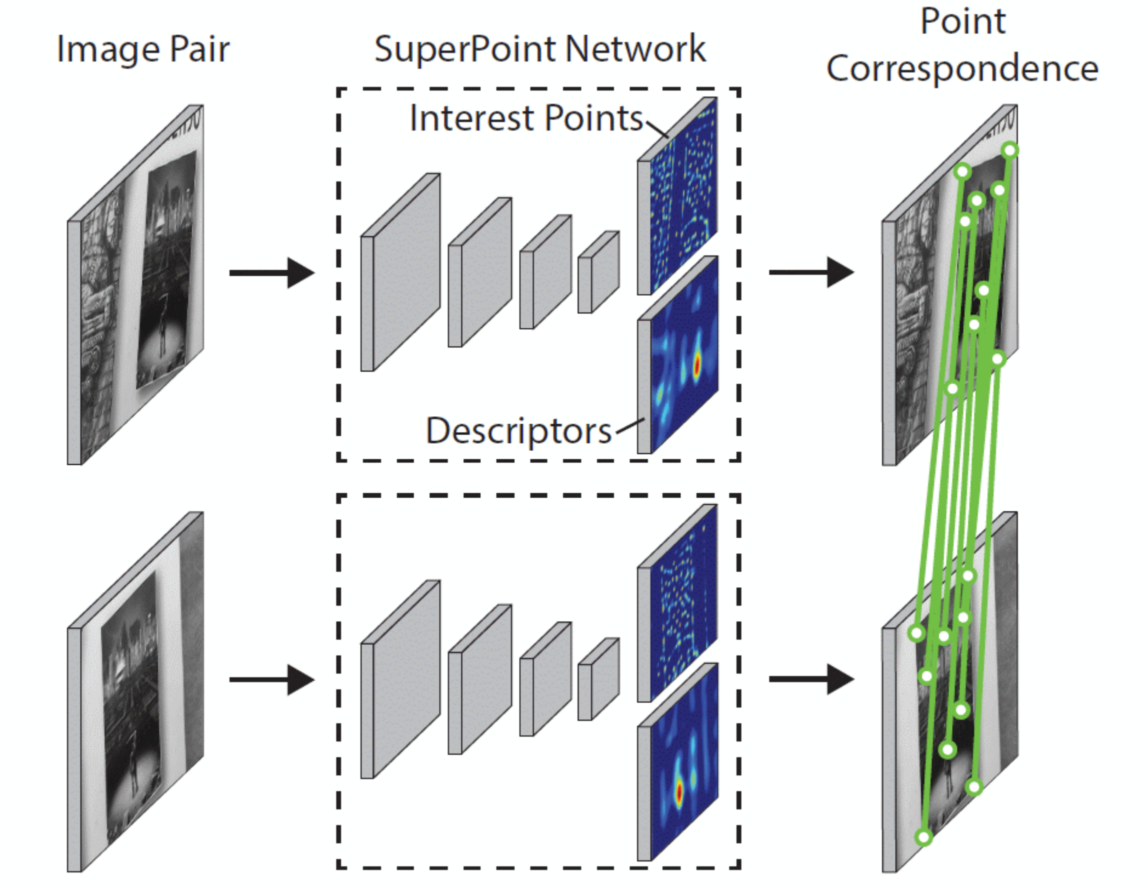


Figure 1. **SuperPoint for Geometric Correspondences.** We present a fully-convolutional neural network that computes SIFT-like 2D interest point locations and descriptors in a single forward pass and runs at 70 FPS on 480×640 images with a Titan X GPU.

Self-Supervised Training

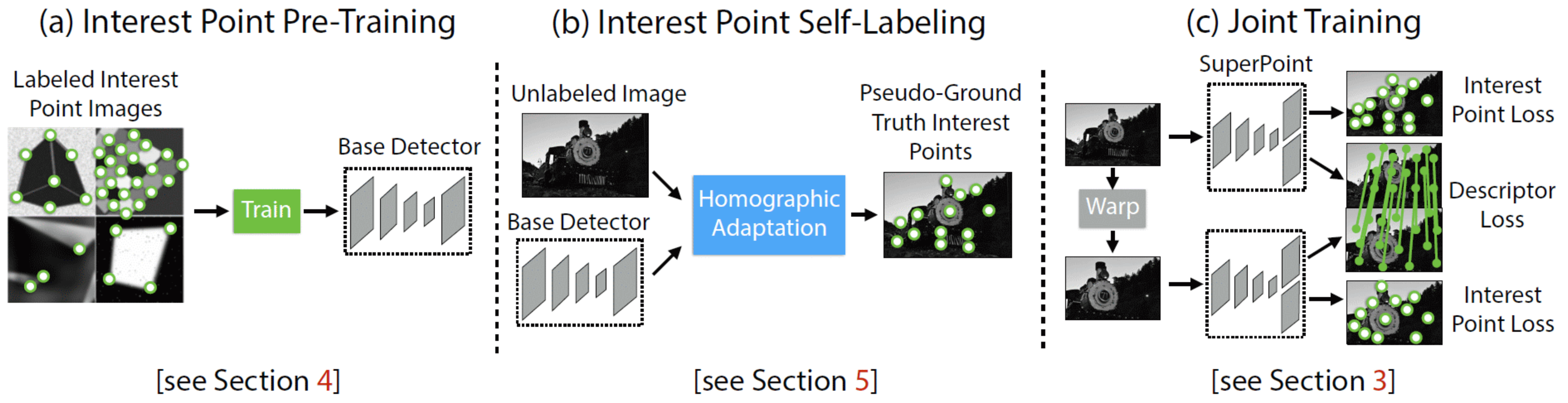


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to (c) train a fully-convolutional network that jointly extracts interest points and descriptors from an image.

Superpoint Architecture

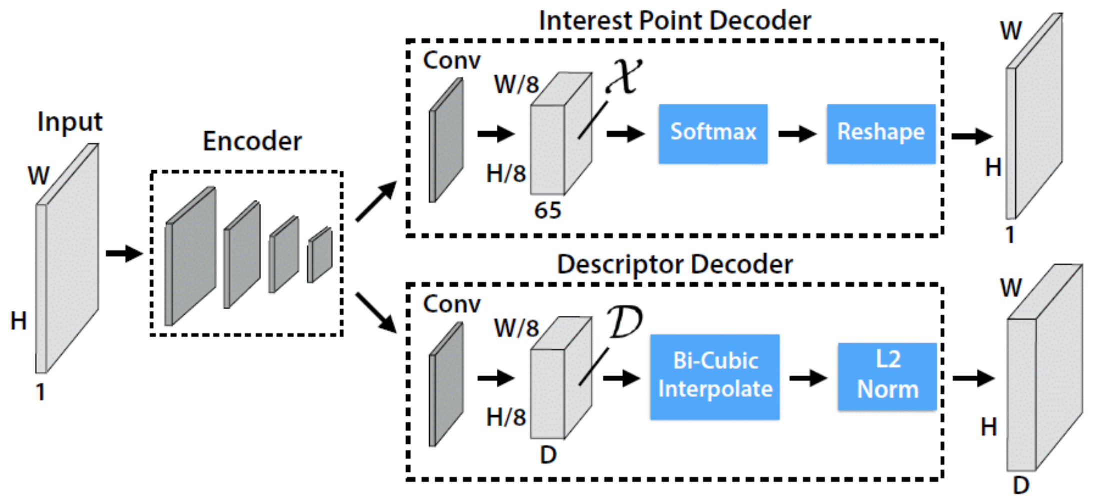


Figure 3. **SuperPoint Decoders.** Both decoders operate on a shared and spatially reduced representation of the input. To keep the model fast and easy to train, both decoders use non-learned upsampling to bring the representation back to $\mathbb{R}^{H \times W}$.

- The interest point detector head computes Hc x Wc x 65 and outputs a tensor sized H x W.
- The 65 channels correspond to local, non-overlapping 8 x8 grid regions of pixels plus an extra "no interest point" dustbin.
- After a channel-wise softmax, the dustbin dimension is removed and Hc x Wc x 64 ➔ H x W reshape is performed.
- The descriptor head computes Hc x Wc x D and outputs a tensor sized H x W x D.
- To output a dense map of L2-normalized fixed length descriptors, first output a semi-dense grid of descriptors (e.g., one every 8 pixels).
- The decoder then performs bicubic interpolation of the descriptor and then L2-normalizes to unit length.

Joint Geometric and Classification Loss

$$\mathcal{L}(\mathcal{X}, \mathcal{X}', \mathcal{D}, \mathcal{D}'; Y, Y', S) = \mathcal{L}_p(\mathcal{X}, Y) + \mathcal{L}_p(\mathcal{X}', Y') + \lambda \mathcal{L}_d(\mathcal{D}, \mathcal{D}', S).$$

- The interest point detector loss function Lp is a fully convolutional cross-entropy

$$\mathcal{L}_p(\mathcal{X}, Y) = \frac{1}{H_c W_c} \sum_{\substack{h=1 \\ w=1}}^{H_c, W_c} l_p(\mathbf{x}_{hw}; y_{hw}),$$

where

$$l_p(\mathbf{x}_{hw}; y) = -\log\left(\frac{\exp(\mathbf{x}_{hwy})}{\sum_{k=1}^{65} \exp(\mathbf{x}_{hwk})}\right).$$

- The descriptor loss is applied to all pairs of descriptor cells, (h, w) and (h’, w’)
- The homography-induced correspondence between the (h, w) cell and the (h’, w’) cell can be written as:

$$s_{hwh'w'} = \begin{cases} 1, & \text{if } ||\widehat{\mathcal{H}\mathbf{p}_{hw}} - \mathbf{p}_{h'w'}|| \leq 8 \\ 0, & \text{otherwise} \end{cases}$$

- The descriptor loss is given by:

$$\mathcal{L}_d(\mathcal{D}, \mathcal{D}', S) = \frac{1}{(H_c W_c)^2} \sum_{\substack{h=1 \\ w=1}}^{H_c, W_c} \sum_{\substack{h'=1 \\ w'=1}}^{H_c, W_c} l_d(\mathbf{d}_{hw}, \mathbf{d}'_{h'w'}; s_{hwh'w'}),$$

where

$$l_d(\mathbf{d}, \mathbf{d}'; s) = \lambda_d * s * \max(0, m_p - \mathbf{d}^T \mathbf{d}') + (1 - s) * \max(0, \mathbf{d}^T \mathbf{d}' - m_n).$$

Comparative Results

	57 Illumination Scenes		59 Viewpoint Scenes	
	NMS=4	NMS=8	NMS=4	NMS=8
SuperPoint	**.652**	**.631**	.503	**.484**
MagicPoint	.575	.507	.322	.260
FAST	.575	.472	.503	.404
Harris	.620	.533	**.556**	.461
Shi	.606	.511	.552	.453
Random	.101	.103	.100	.104

Table 3. **HPatches Detector Repeatability**. SuperPoint is the most repeatable under illumination changes, competitive on viewpoint changes, and outperforms MagicPoint in all scenarios.

	Homography Estimation			Detector Metrics		Descriptor Metrics	
	$\epsilon = 1$	$\epsilon = 3$	$\epsilon = 5$	Rep.	MLE	NN mAP	M. Score
SuperPoint	.310	**.684**	**.829**	.581	1.158	**.821**	**.470**
LIFT	.284	.598	.717	.449	1.102	.664	.315
SIFT	**.424**	.676	.759	.495	**0.833**	.694	.313
ORB	.150	.395	.538	**.641**	1.157	.735	.266

Table 4. **HPatches Homography Estimation.** SuperPoint outperforms LIFT and ORB and performs comparably to SIFT using various ϵ thresholds of correctness. We also report related metrics which measure detector and descriptor performance individually.

- SIFT performs well for sub-pixel precision homographies and has the lowest mean localization error (MLE).
- SuperPoint scores strongly in descriptor-focused metrics such as nearest neighbor mAP and matching score (M. Score)