Local Features: From SIFT to Differentiable Methods

Vasileios Balntas
Facebook Reality Labs

Dmytro Mishkin
Czech Technical University

Edgar Riba
Computer Vision Center

Tutorial Programme Overview

Classical & modern methods

From paper to practice

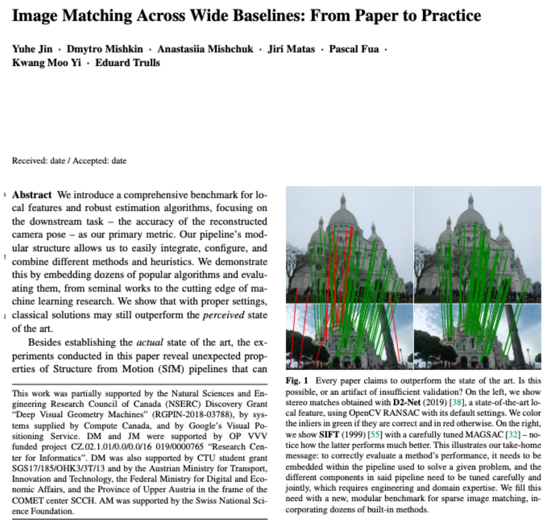


Image Matching Across Wide Baselines: From Paper to Practice

Yuhe Jin · Dmytro Mishkin · Anastasiia Mishchuk · Jiri Matas · Pascal Fua · Kwang Moo Yi · Eduard Trulls

Received: date / Accepted: date

Abstract We introduce a comprehensive benchmark for local features and robust estimation algorithms, focusing on the downstream task – the accuracy of the reconstructed camera pose – as our primary metric. Our pipeline's modular structure allows us to easily integrate, configure, and combine different methods and heuristics. We demonstrate this by embedding dozens of popular algorithms and evaluating them, from seminal works to the cutting edge of machine learning research. We show that with proper settings, classical solutions may still outperform the *perceived* state of the art.

Besides establishing the *actual* state of the art, the experiments conducted in this paper reveal unexpected properties of Structure from Motion (SfM) pipelines that can

This work was partially supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant "Deep Visual Geometry Machines" (RGPIN-2018-03788), by systems supplied by Compute Canada, and by Google's Visual Positioning Service. DM and JM were supported by OP VVV funded project CZ.02.1.01/0.0/0.0/16 019/0000765 "Research Center for Informatics". DM was also supported by CTU student grant SGS17/185/OHK3/3T/13 and by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry for Digital and Economic Affairs, and the Province of Upper Austria in the frame of the COMET center SCCH. AM was supported by the Swiss National Science Foundation.

Fig. 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) [38], a state-of-the-art local feature, using OpenCV RANSAC with its default settings. We color the inliers in green if they are correct and in red otherwise. On the right, we show **SIFT** (1999) [55] with a carefully tuned MAGSAC [32] – notice how the latter performs much better. This illustrates our take-home message: to correctly evaluate a method's performance, it needs to be embedded within the pipeline used to solve a given problem, and the different components in said pipeline need to be tuned carefully and jointly, which requires engineering and domain expertise. We fill this need with a new, modular benchmark for sparse image matching, incorporating dozens of built-in methods.

Kornia library

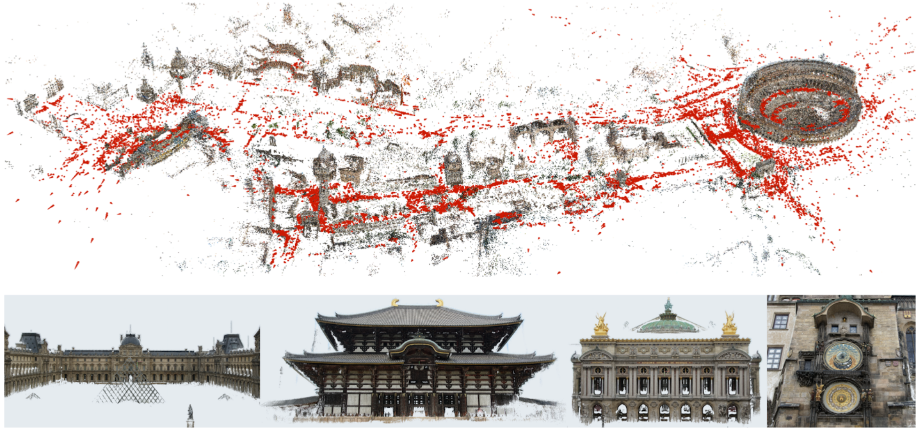
Programme

- 09:00 – 10:00 Overview of classical & end-to-end methods
- 10:00 – 11:00 Local features: from paper to practice
- 11:00 – 12:00 Kornia introduction & hands-on Session

https://local-features-tutorial.github.io/

What is image matching?

Why is image matching useful?



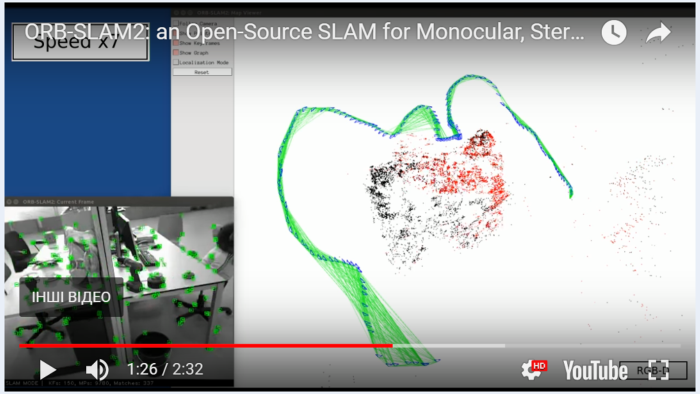
SfM

L. Schonberger and J.-M. Frahm,
Structure-from-Motion Revisited, 2016
COLMAP

Why is image matching useful?

ORB-SLAM2 for Monocular, Stereo and RGB-D Cameras

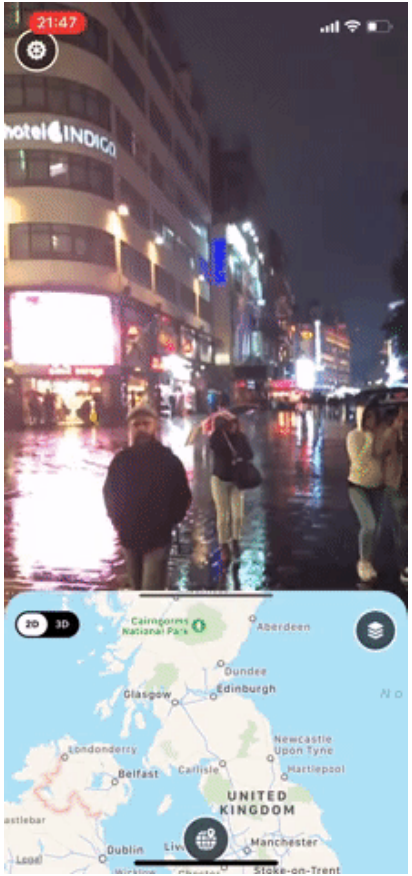
Code: https://github.com/raulmur/ORB_SLAM2 .
Paper: Raúl Mur-Artal, and Juan D. Tardós. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. ArXiv preprint arXiv:1610.06475, 2016



SLAM

R. Mur-Artal, and J. D. Tardós.
ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras, arXiv 2016

Localisation



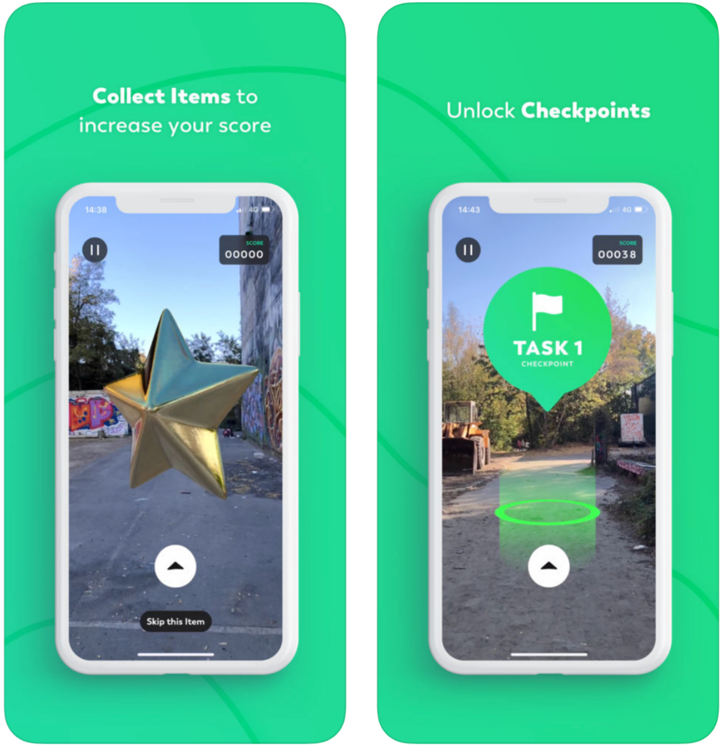
SCAPE

Why is image matching useful?

SLAM

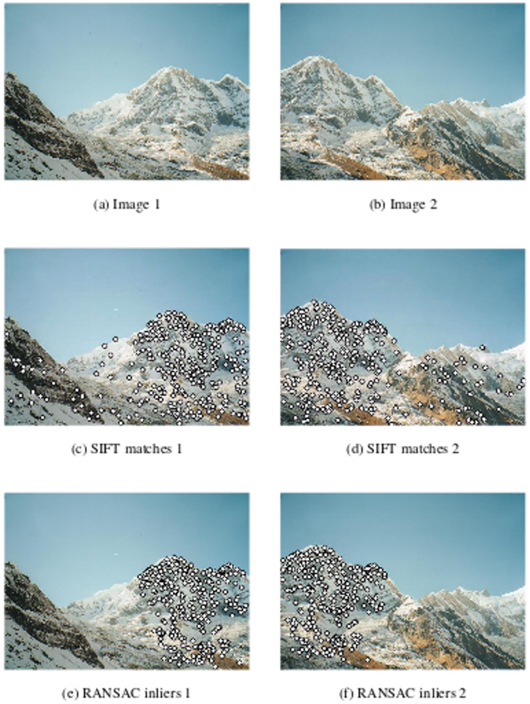
Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich, Superpoint.
MagicLeap SLAM

Why is image matching useful?



Augmented Reality
ScavengAR App

Why is image matching useful?



Panoramas

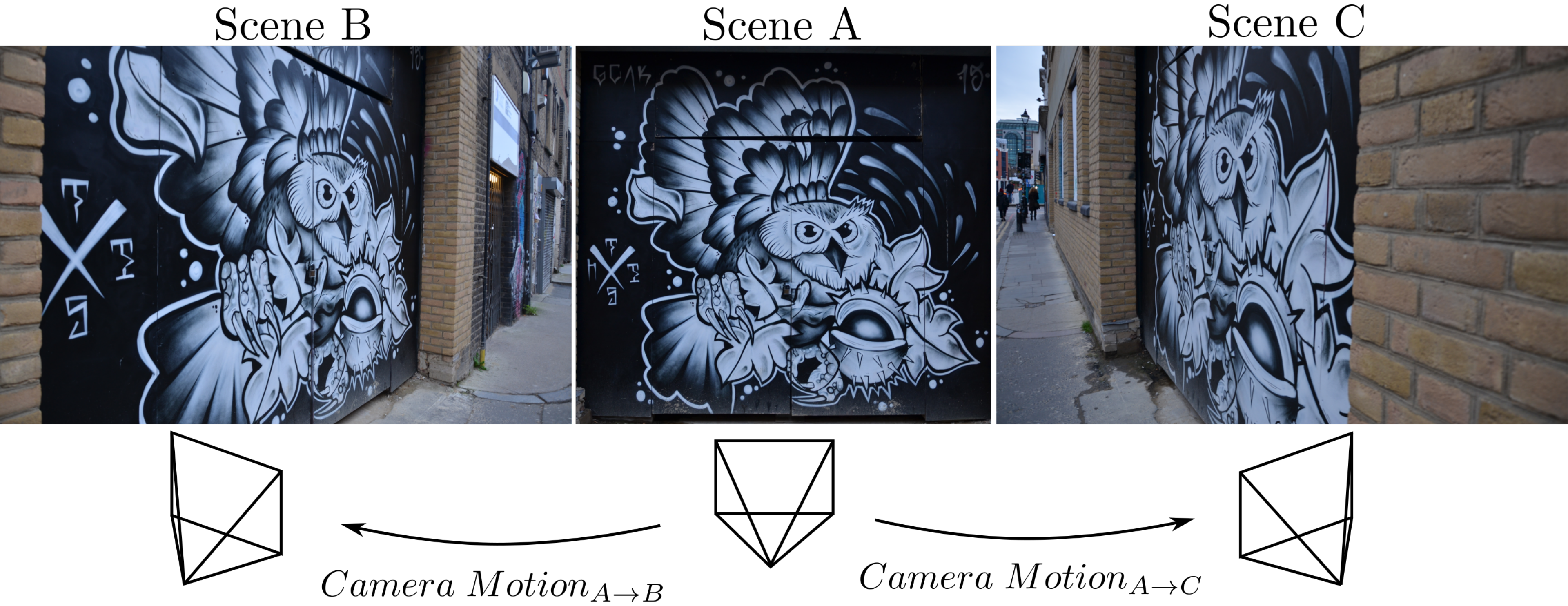
Brown and Lowe, **Automatic panoramic image stitching using invariant image features**

Image: Rick Szeliski

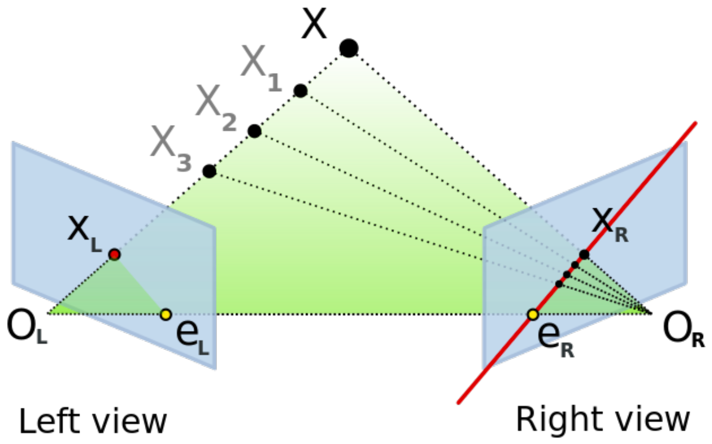
Image Matching - Practicality

- Matching a set of images enables us to “recover” the geometry of the world from individual images.

- To understand why, we need to quickly discuss a few things about cameras.


Scene B
Scene A
Scene C
$Camera\ Motion_{A \to B}$
$Camera\ Motion_{A \to C}$

Point correspondences for triangulation

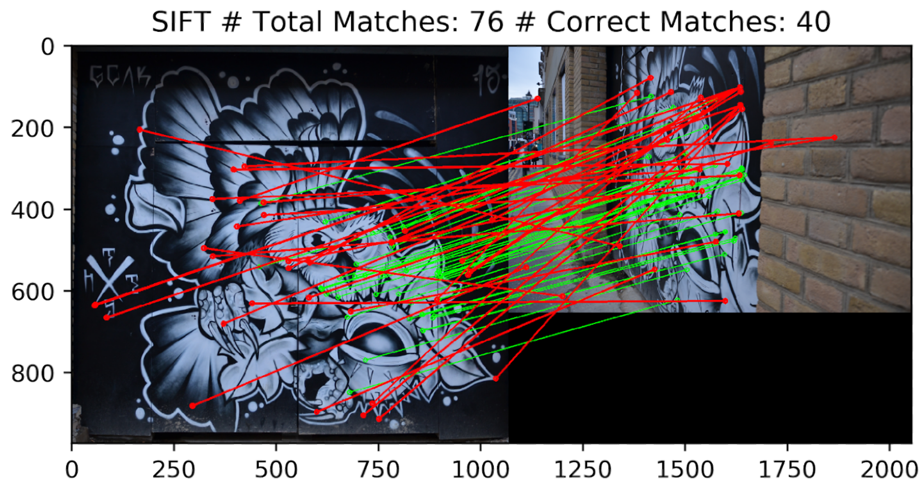


- One left-view to right-view match is not enough
- Min number of matches defined by theory and algorithms (e.g. 8-point algorithm)
- Practically we aim for a higher number of matches than the theoretical (e.g. > 100)

[1]Hartley and Zisserman, *Multiple view geometry in computer vision.*

Image: Wikipedia

Matching points - why do we need a lot of them?



RANSAC: fitting the data with gross outliers

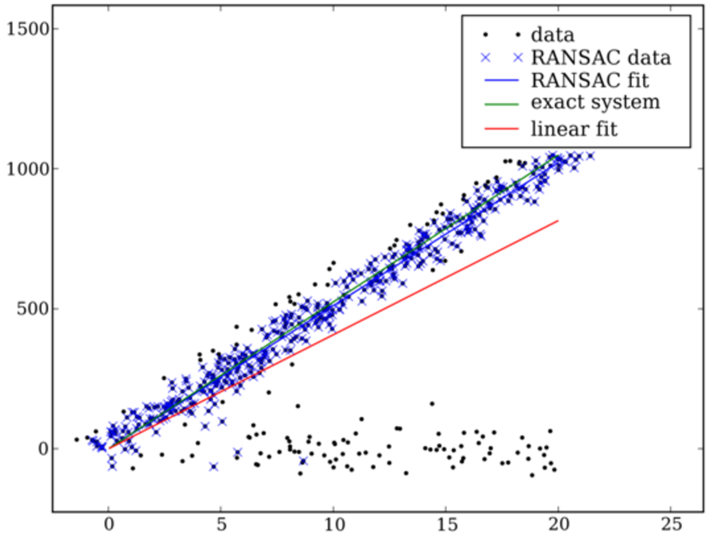
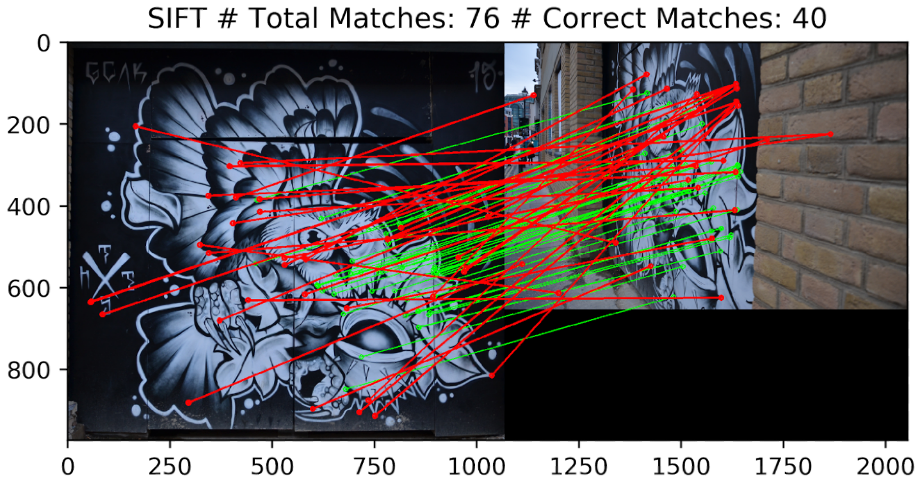


Image credit: https://scipy-cookbook.readthedocs.io/items/RANSAC.html

https://github.com/ducha-aiki/pyransac

More details & info: Dmytro's talk at 10:00am

Matching points - why do we need a lot of them?



RANSAC: image matching example

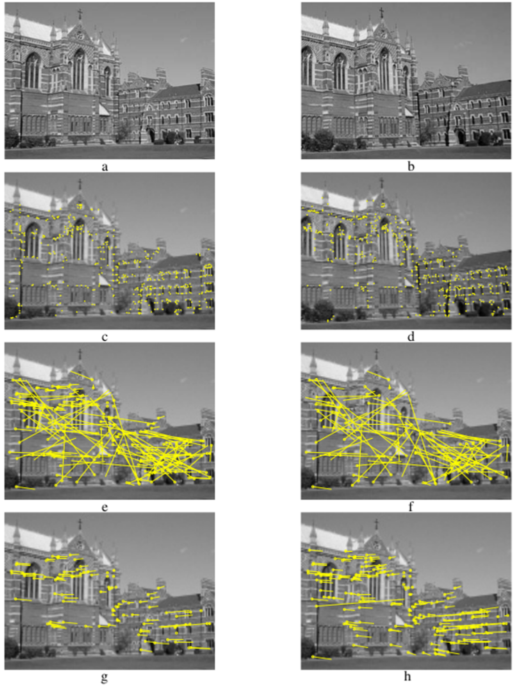


Fig. 11.4. **Automatic computation of the fundamental matrix between two images using RANSAC.** *(a) (b) left and right images of Keble College, Oxford. The motion between views is a translation and rotation. The images are 640×480 pixels. (c) (d) detected corners superimposed on the images. There are approximately 500 corners on each image. The following results are superimposed on the left image: (e) 188 putative matches shown by the line linking corners, note the clear mismatches; (f) outliers – 89 of the putative matches. (g) inliers – 99 correspondences consistent with the estimated $\mathtt{F}$; (h) final set of 157 correspondences after guided matching and MLE. There are still a few mismatches evident, e.g. the long line on the left.*

Multiple View Geometry in Computer Vision

Hartley & Zisserman

Recap

Better ways to match points between two images

↓

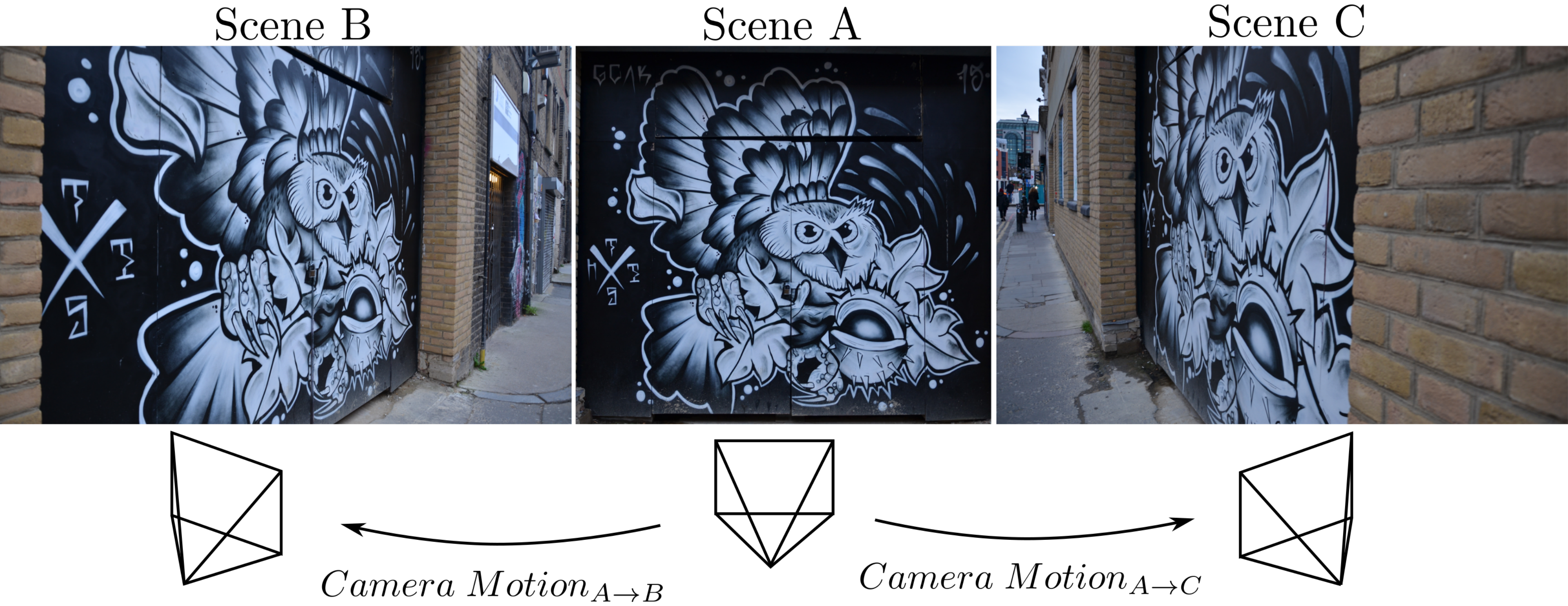
Easier job for relative camera pose estimators

↓

Better 3D models, panoramas, AR apps etc

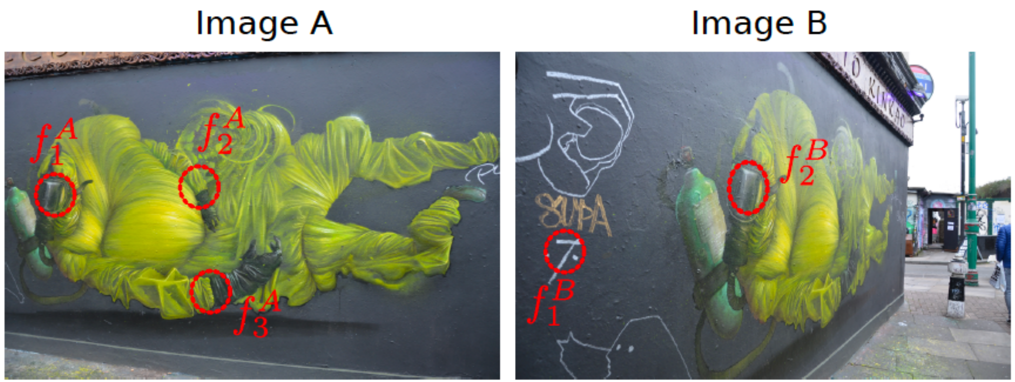
Classical pipeline

Image Credit: Wikipedia


Scene B
Scene A
Scene C
$Camera\ Motion_{A \to B}$
$Camera\ Motion_{A \to C}$

Classical pipeline

The classical image matching pipeline

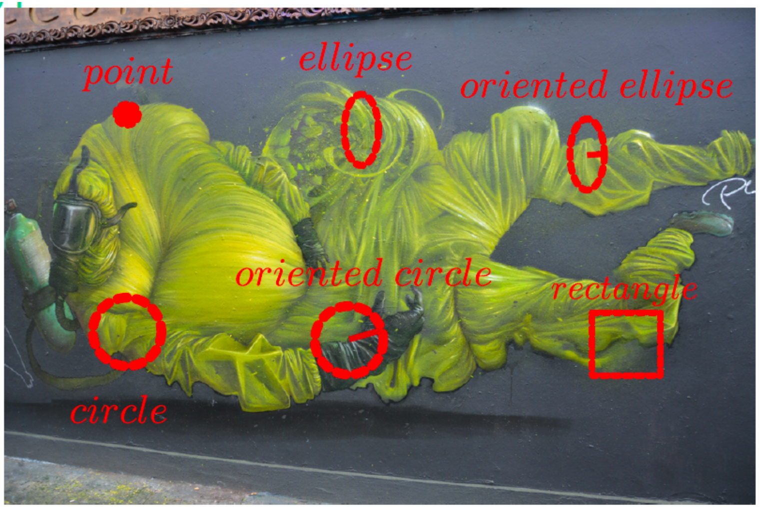


Step 1 *Detection:* Choose "interesting" points

Step 2 *Description:* Convert the points to a suitable mathematical representation (descriptor)

Step 3 *Matching:* Match the point descriptors between the two images

Common types of feature frames



- *Point:* x, y
- *Circle:* x, y, ρ
- *Rectangle:* x, y, w, h
- *Oriented Circle:* x, y, ρ, θ
- *Ellipse:* x, y, a, b
- *Oriented Ellipse:* x, y, a, b, θ

Feature frame/keypoint – Simplest Definition

$$f(x,y) = \sum_{(x_k,y_k)\in W} \left(I(x_k,y_k) - I(x_k+\Delta x, y_k+\Delta y)\right)^2$$

$$f(x,y) \approx \sum_{(x,y)\in W} \left(I_x(x,y)\Delta x + I_y(x,y)\Delta y\right)^2$$

Feature frame/keypoint – Simplest Definition

$$f(x,y) \approx \begin{pmatrix} \Delta x & \Delta y \end{pmatrix} M \begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix}$$

$$M = \begin{bmatrix} \sum\limits_{(x,y)\in W} I_x^2 & \sum\limits_{(x,y)\in W} I_x I_y \\ \sum\limits_{(x,y)\in W} I_x I_y & \sum\limits_{(x,y)\in W} I_y^2 \end{bmatrix}$$

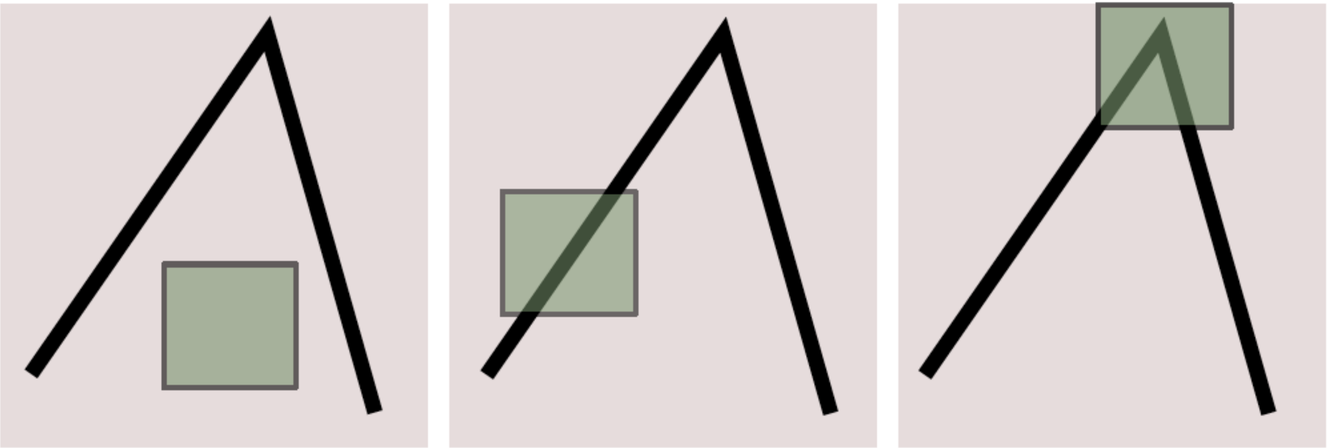
Feature frame/keypoint – Simplest Definition

$$M = \begin{bmatrix} \sum_{(x,y)\in W} I_x^2 & \sum_{(x,y)\in W} I_x I_y \\ \sum_{(x,y)\in W} I_x I_y & \sum_{(x,y)\in W} I_y^2 \end{bmatrix}$$

λ_1, λ_2: Eigenvalues of M

- $\lambda_1, \lambda_2 \approx 0$
- $\lambda_1 \gg \lambda_2$
- $\lambda_1 \ll \lambda_2$
- $\lambda_1 \approx \lambda_2 \gg 0$

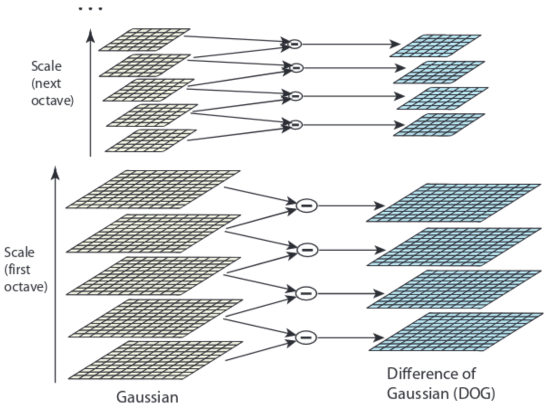
Feature frame/keypoint – Simplest Definition

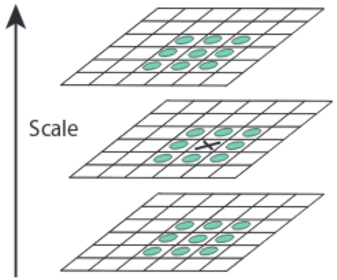


$\lambda_1 \approx \lambda_2 \approx 0$ $\qquad$ $\lambda_1 \gg \lambda_2$ $\qquad$ $\lambda_1 \approx \lambda_2 \gg 0$

Adding scale estimation

SIFT Detector





[4]

[4] Lowe, "Distinctive image features from scale-invariant keypoints".

SURF

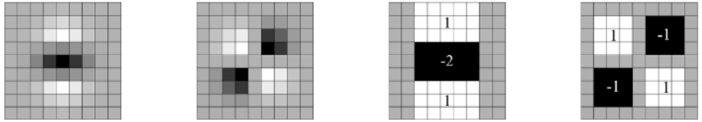


Fig. 1. Left to right: the (discretised and cropped) Gaussian second order partial derivatives in y-direction and xy-direction, and our approximations thereof using box filters. The grey regions are equal to zero. [5]

[5] Bay, Tuytelaars, and Van Gool, "Surf: Speeded up robust features".

Edge Foci

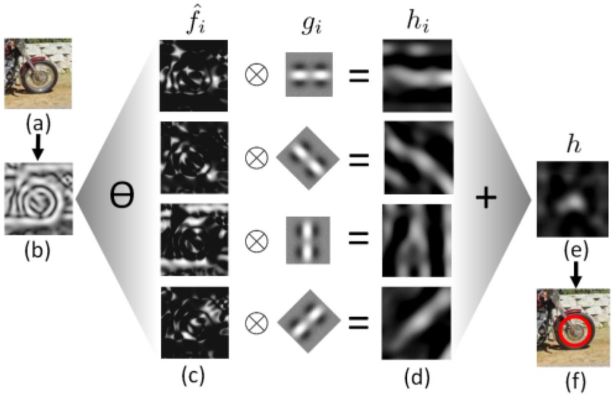
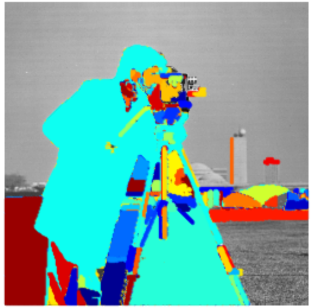


Figure 2. Flow diagram of the detector: (a) input image, (b) normalized gradient $\hat{f}$, (c) normalized gradients separated into orientations $\hat{f}_i$, (d) responses after applying oriented filter $h_i = \hat{f}_i \otimes g_i$, (e) the aggregated results h, and (f) detected interest point. [7]

[7] Zitnick and Ramnath, "Edge foci interest points".

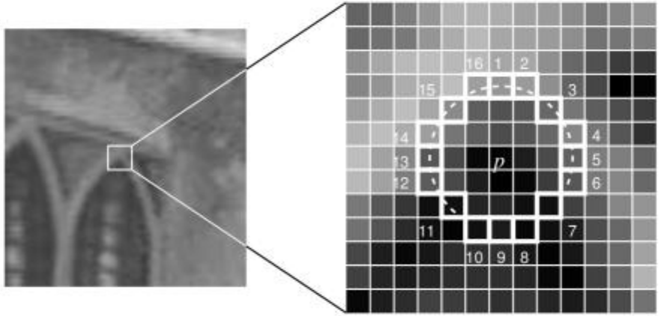
MSER



[8]

[8] Matas et al., “Robust wide-baseline stereo from maximally stable extremal regions”.

FAST



[9]

[9] Rosten and Drummond, “Machine learning for high-speed corner detection”.

- Many possibilities for types of feature frames
- Might include scale & orientation

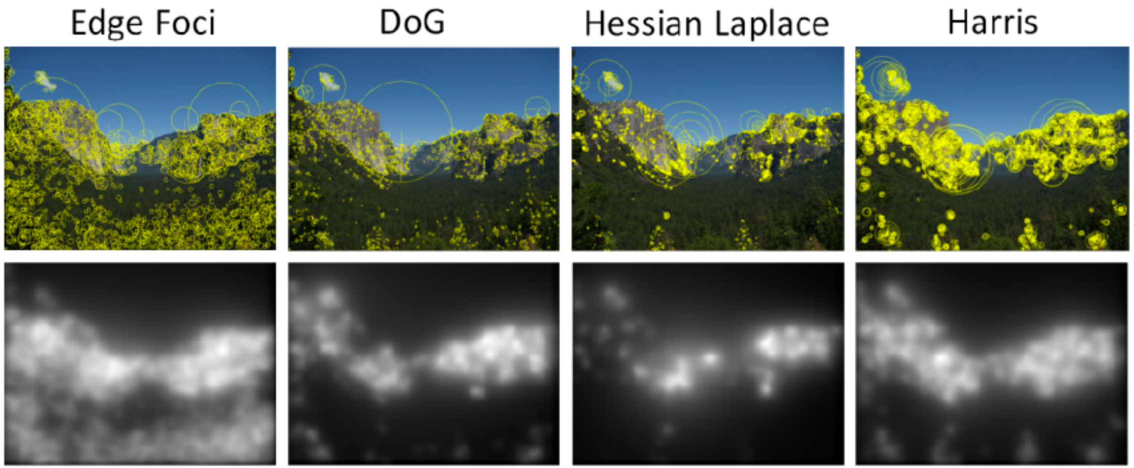
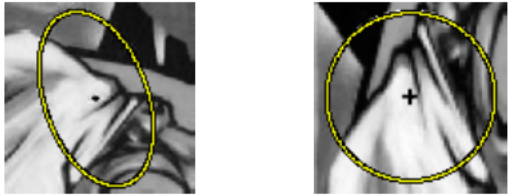


Figure 8. Visualization of the interest points and their spatial distributions for various detectors on Yosemite image.

Source: Rick Szeliski

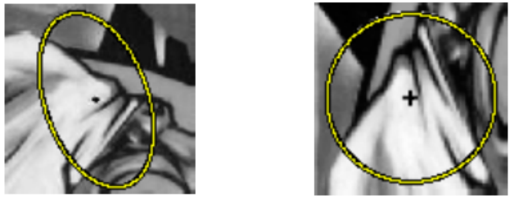
From feature frames to patches

Detect Regions



Rectify patch around feature frame

Detect Regions

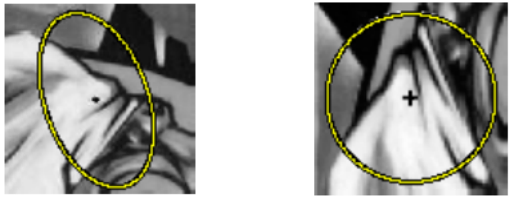


Rectify patch around feature frame

Local Descriptor

A *vectorial representation* of the patch around a feature frame which is more a discriminative and robust than the patch.

Detect Regions

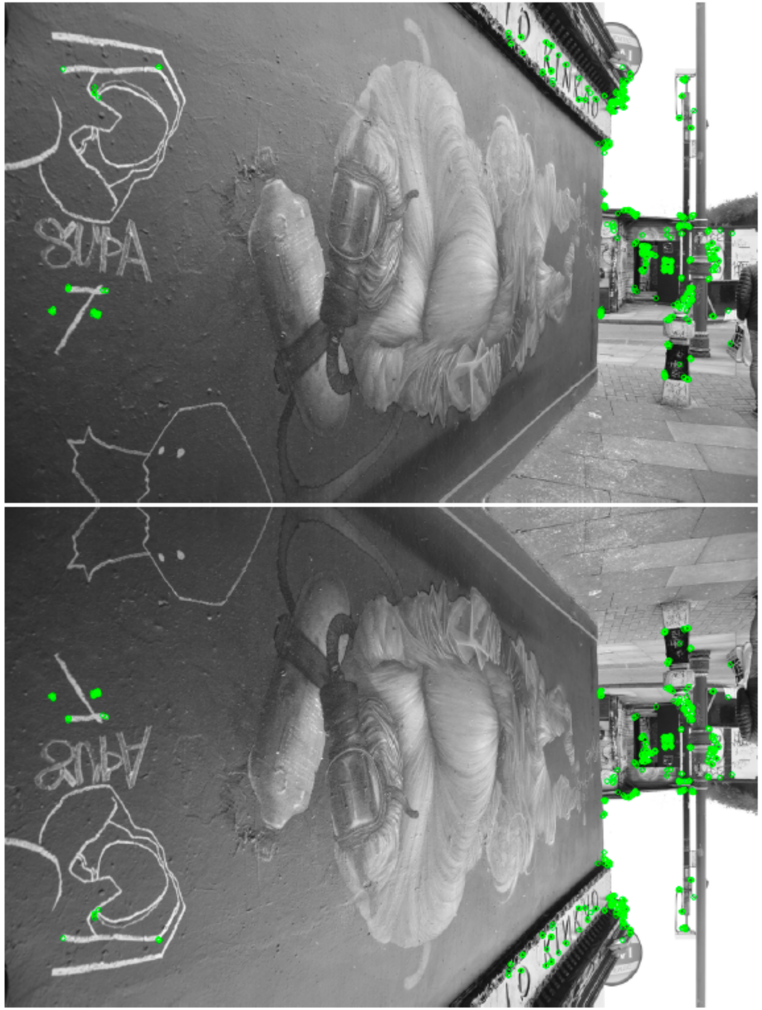


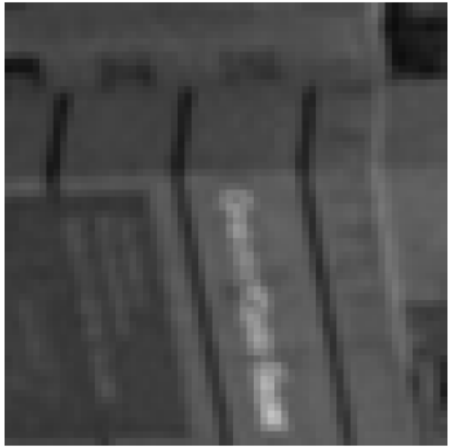
Rectify patch around feature frame

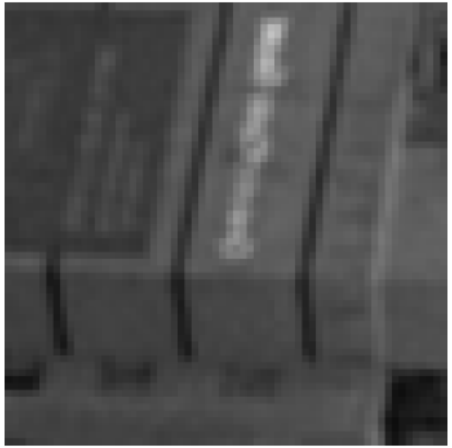
Local Descriptor

A *vectorial representation* of the patch around a feature frame which is more a discriminative and robust than the patch.

Orientation

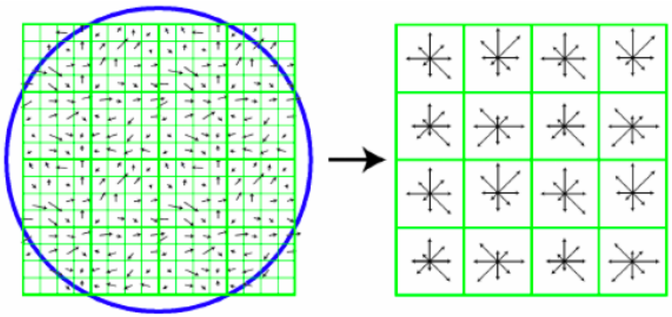






How to describe patches

SIFT

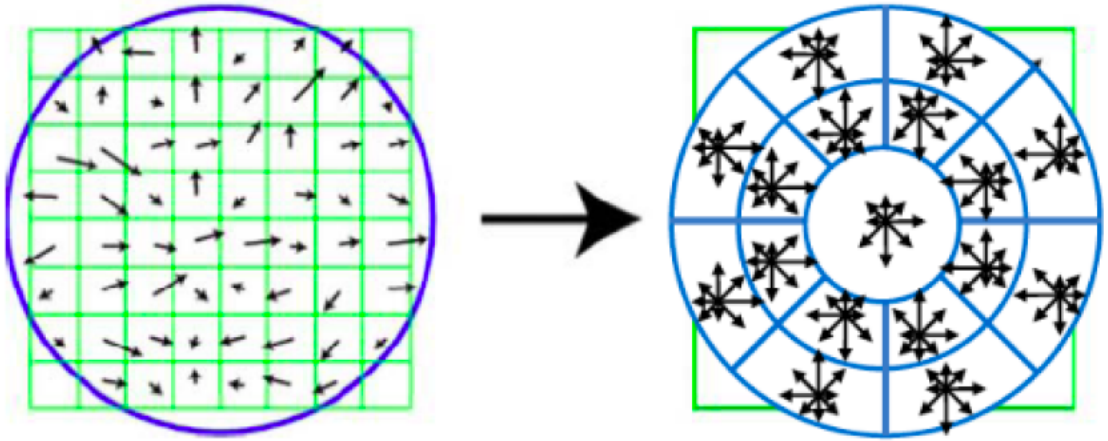


- The local spatial pooling of the descriptor is based on a rectangular grid that partitions the patch into several regions.
- Assuming the patch is divided into M rectangular areas, and the gradients are quantised to K angle bins, the resulting K dimensional histograms concatenated from M areas, will be represented by a point in the $\mathbb{R}^{M*K}$ space.
- In the case of the original implementation of SIFT, 16 grid quanta were combined with 8 angular bins, resulting in final dimensionality of 128.

Lowe: Distinctive Image Features from Scale-Invariant Keypoints

Image: vlfeat.org

GLOH

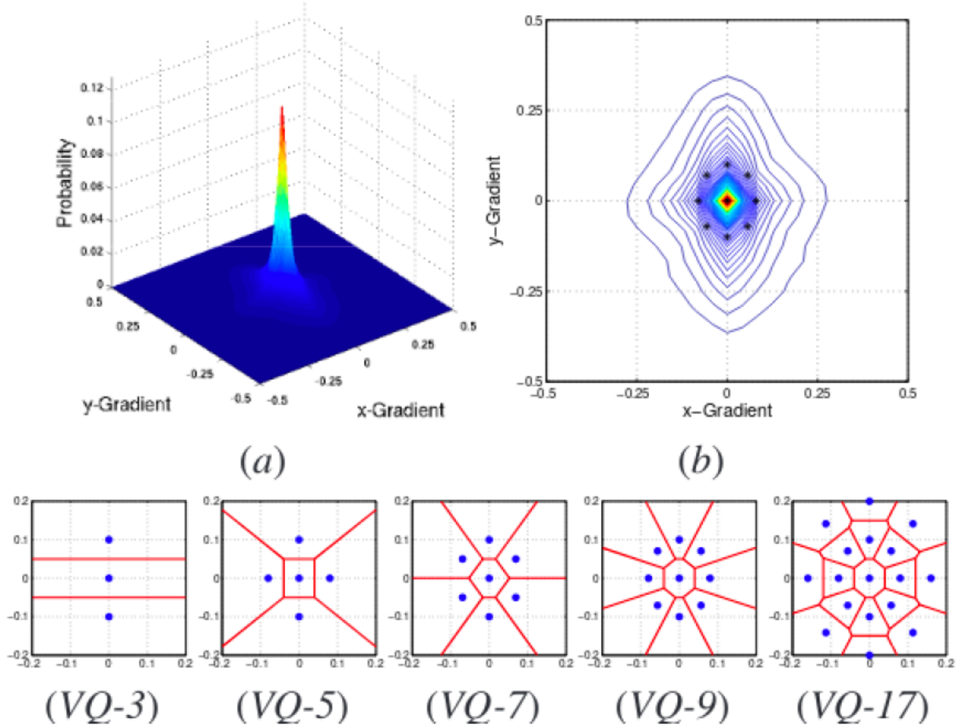


(a) image gradients

(b) keypoint descriptor

Mikolajczyk & Schmidt: **A Performance Evaluation of Local Descriptors**

CHoG



Chandrasekhar et al. CHoG: Compressed histogram of gradients a low bit-rate feature descriptor

DAISY

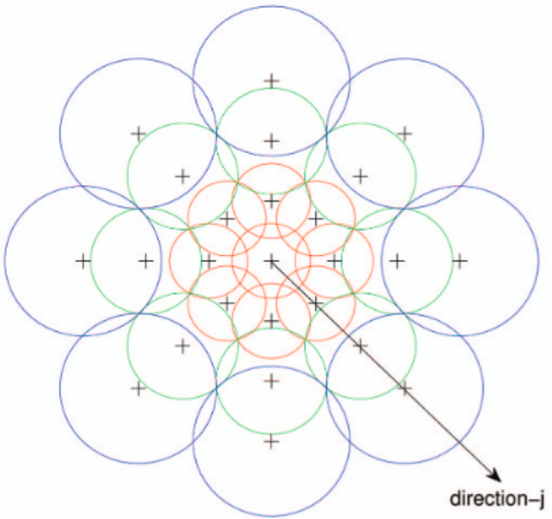
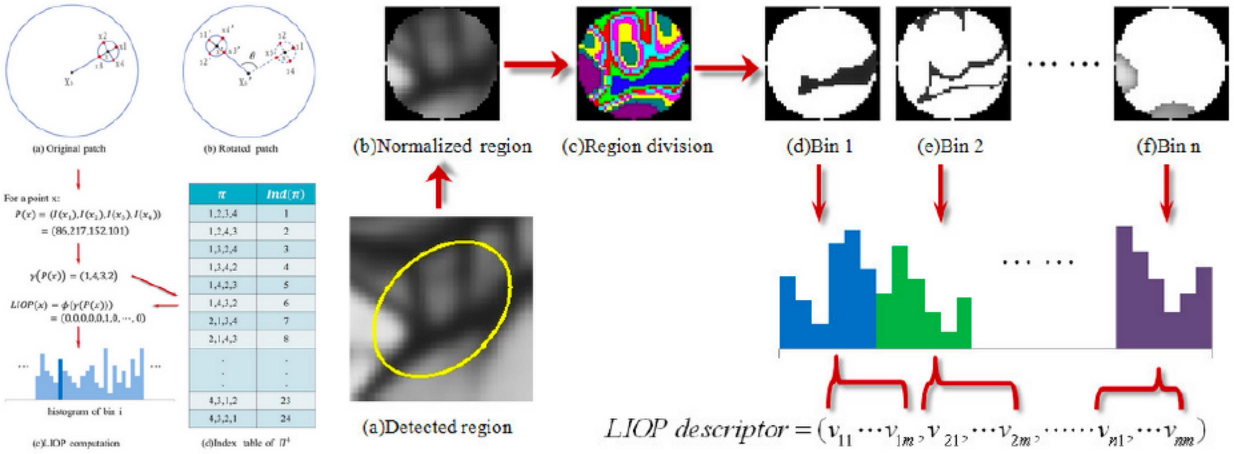


Fig. 6. The DAISY descriptor: Each circle represents a region where the radius is proportional to the standard deviations of the Gaussian kernels and the "+" sign represents the locations where we sample the convolved orientation maps center being a pixel location where we compute the descriptor. By overlapping the regions, we achieve smooth transitions between the regions and a degree of rotational robustness. The radii of the outer regions are increased to have an equal sampling of the rotational axis, which is necessary for robustness against rotation.

Winder et al. **Picking The Best Daisy**

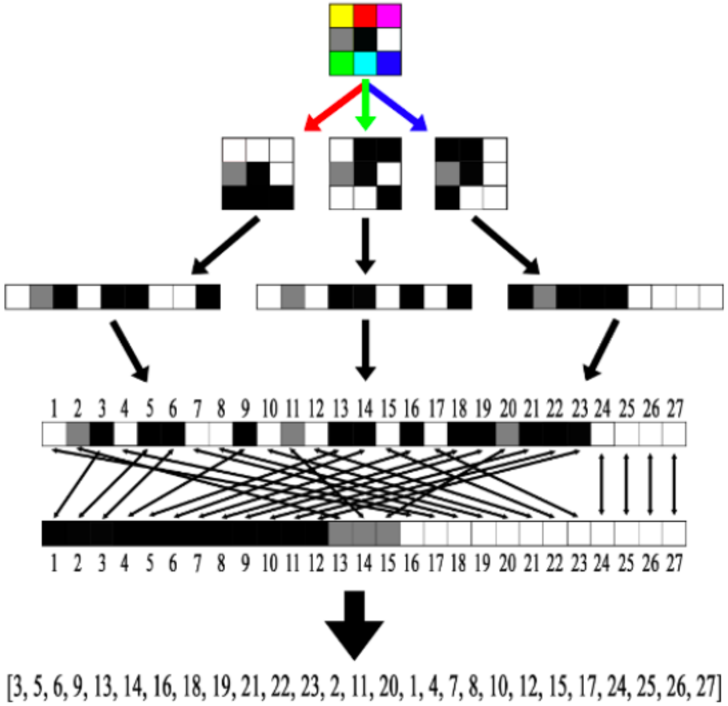
LIOP



Wang et al. **Local Intensity Order Pattern for Feature Description**

LUCID

```
[~, desc1] = sort(p1(:));
[~, desc2] = sort(p2(:));
distance = sum(desc1 ~= desc2);
```

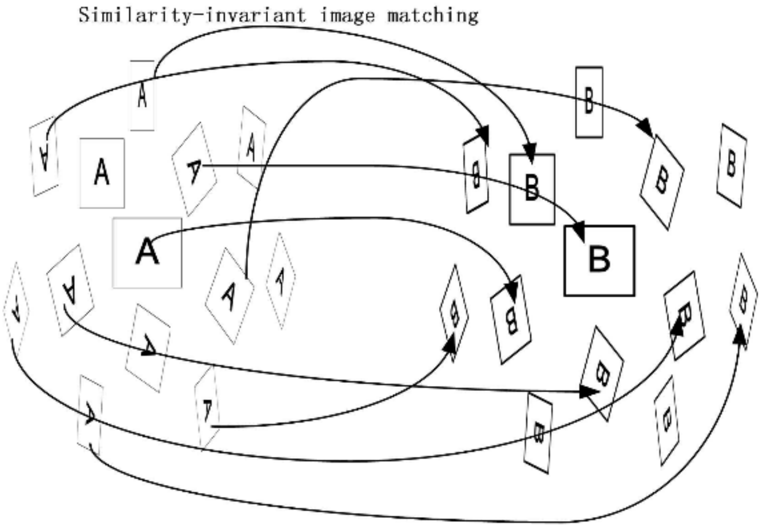


Ziegler et al. **Locally Uniform Comparison Image Descriptor**

Aggregation across scales and viewpoints

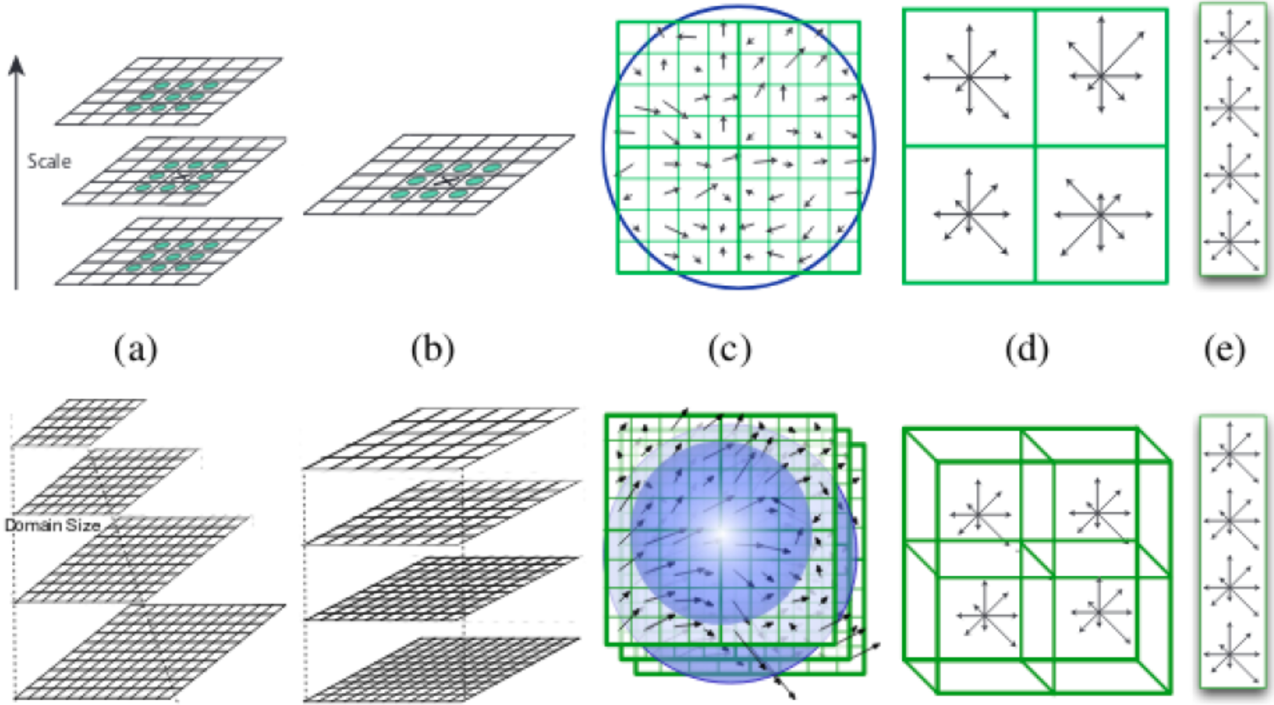
Several methods identified that aggregation across different scales or different viewpoints into a single feature vector can improve the discriminative power of the descriptor, albeit at the price of much higher computational cost.

ASIFT



Morel and Yu. **Affine-Sift**

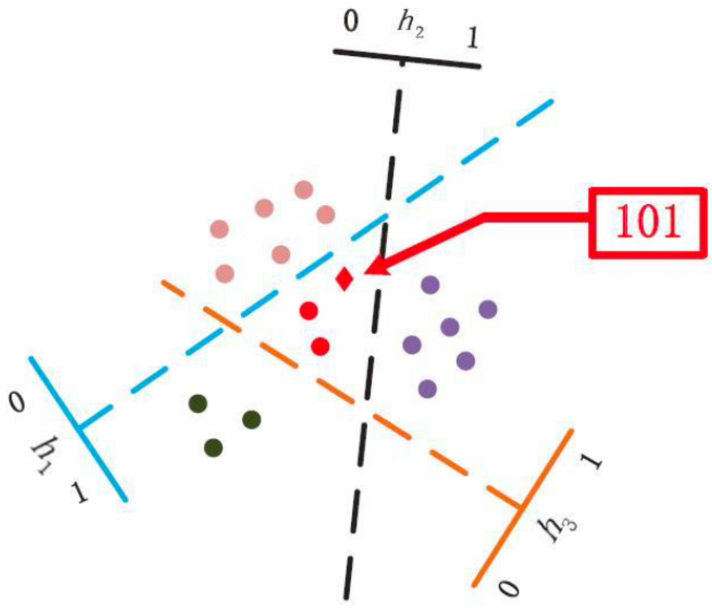
DSP-SIFT



Dong and Soatto. **Domain-Size Pooling in Local Descriptors: DSP-SIFT**

Binary descriptors

Hashing SIFT



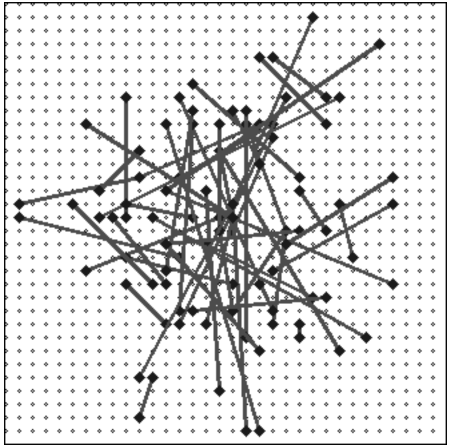
1819

Image from Haisheng Li.

Terasawa and Tanaka, **Spherical LSH for approximate nearest neighbour search on unit hypersphere.**

Strecha et al., **LDAHash: Improved matching with smaller descriptors.**

BRIEF



Calonder et al. **Brief: Binary robust independent elementary features**

Learning-based descriptors

- From 2005 and on, more and more machine learning was utilised

PCA-SIFT

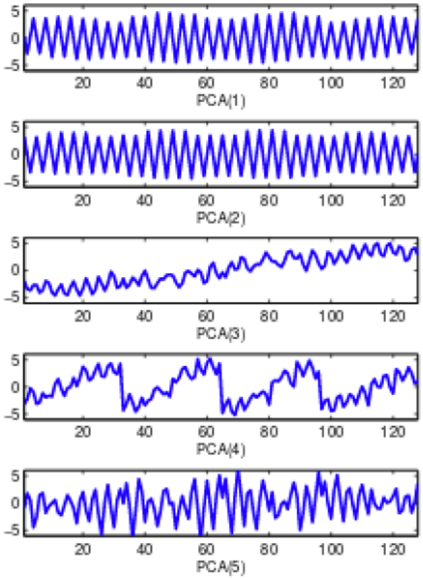
Collect a matrix $X \in \mathbb{R}^{N \times D}$ with N descriptors of dimensionality D

$$C = X^T X$$

$$C = U \Sigma V$$

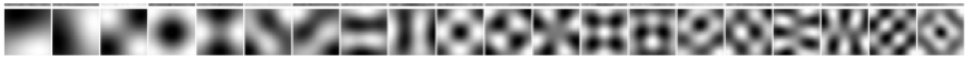
Use the first K eigenvectors from U to project X to a new descriptor of size K. $X_k = U_k X$[23]

Ke and Sukthankar, **PCA-SIFT: A more distinctive representation for local image descriptors**


5
0
-5
20 40 60 80 100 120
PCA(1)
5
0
-5
20 40 60 80 100 120
PCA(2)
5
0
-5
20 40 60 80 100 120
PCA(3)
5
0
-5
20 40 60 80 100 120
PCA(4)
5
0
-5
20 40 60 80 100 120
PCA(5)

PCA

PCA



Linear Projections

$$\mathbf{u}_{\mathbf{LDP}} = \arg\max_{\mathbf{u}} \frac{\sum_{(i,j)\in\mathcal{D}} \|\mathbf{u}^T\mathbf{x}_i - \mathbf{u}^T\mathbf{x}_j\|^2}{\sum_{(i,j)\in\mathcal{S}} \|\mathbf{u}^T\mathbf{x}_i - \mathbf{u}^T\mathbf{x}_j\|^2} = \arg\max_{\mathbf{u}} \frac{\mathbf{u}^T C_{\mathcal{D}} \mathbf{u}}{\mathbf{u}^T C_{\mathcal{S}} \mathbf{u}} \quad (2)$$

Where $C_{\mathcal{D}}$ and $C_{\mathcal{S}}$ represent the inter- and intra-class covariance matrices of differently labeled points (unmatched features in image descriptor space) and same labeled points (matched features), respectively.

$$C_{\mathcal{D}} \stackrel{\text{def}}{=} \sum_{(i,j)\in\mathcal{D}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \quad (3)$$

$$C_{\mathcal{S}} \stackrel{\text{def}}{=} \sum_{(i,j)\in\mathcal{S}} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^T \quad (4)$$

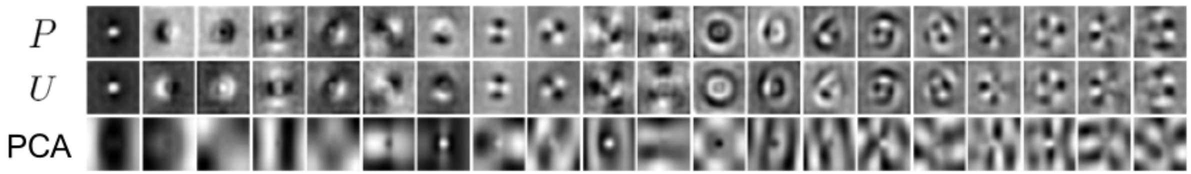
Note that these are not the same matrices as the between-class S_B and within-class scatters S_W in equation (1) for LDA, although they are related (see section 3.3) . The solution is the generalized eigenvectors:

$$U = \text{eig}(C_{\mathcal{S}}^{-1} C_{\mathcal{D}}) \quad (5)$$

The projection matrix is $U \in \mathbb{R}^{m\times m'}$, with $m' \leq m$ eigenvectors corresponding to the m' largest eigenvalues.

Cai, Mikolajczyk, and Matas, **Learning linear discriminant projections for dimensionality reduction of image descriptors**

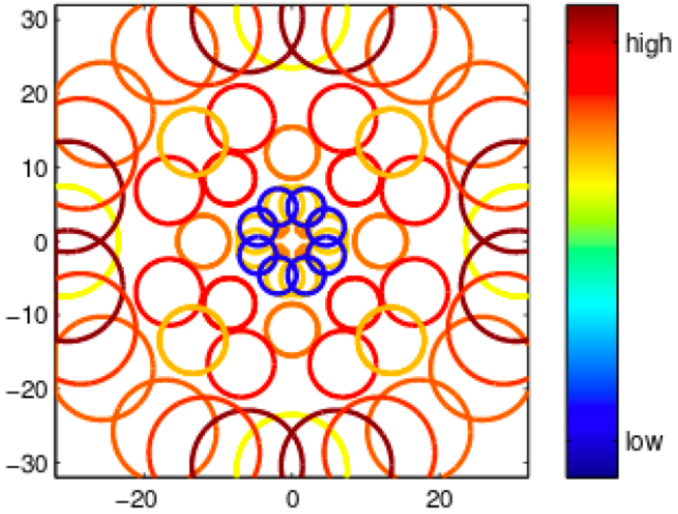
Linear Projections



Convex optimisation for learning descriptors

Learn optimal configuration of gaussian filters s.t.

$$\min_{\mathbf{y}\in P(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{y}) < \min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{u}),$$



Simonyan, Vedaldi, and Zisserman, **Learning Local Feature Descriptors Using Convex Optimisation**

Deep Learning Era

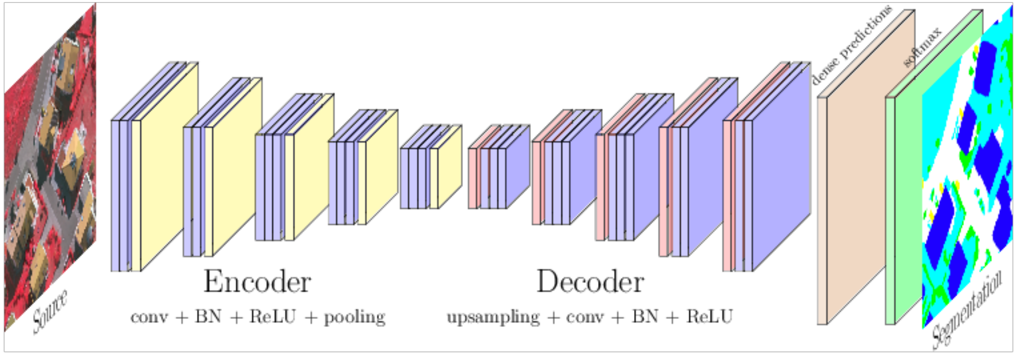
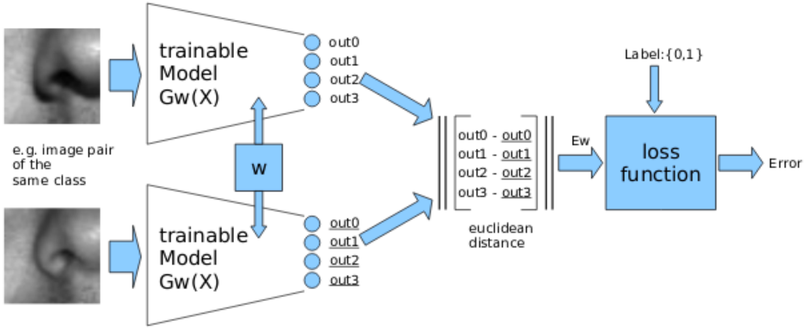


Image: Nicolas Audebert

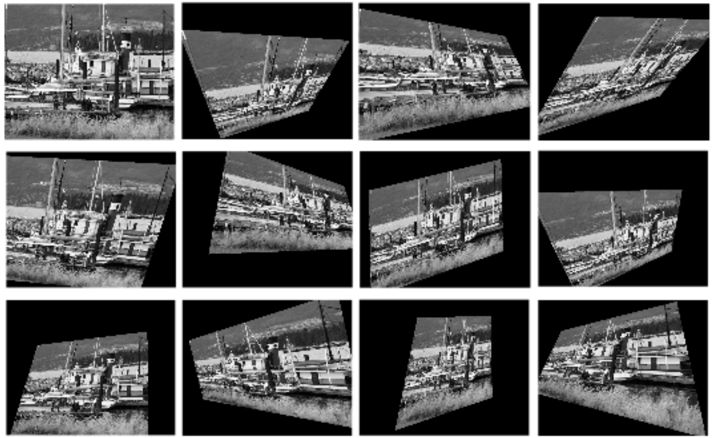
Early work (2008)

- Early work on learning convolutional neural networks as feature descriptors specifically for local patches, but was not immediately followed



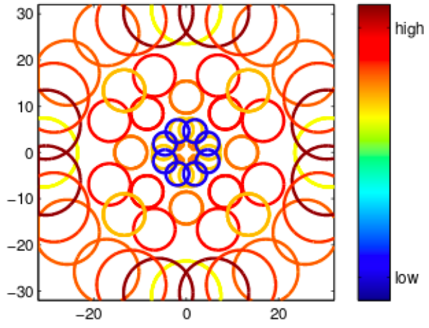
Jahrer, Grabner, and Bischof. **Learned local descriptors for recognition and matching**.

Early work (2008)

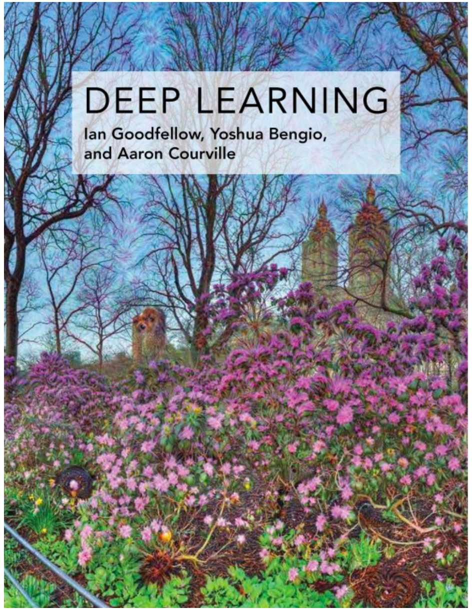


Learn optimal configuration of gaussian filters s.t.

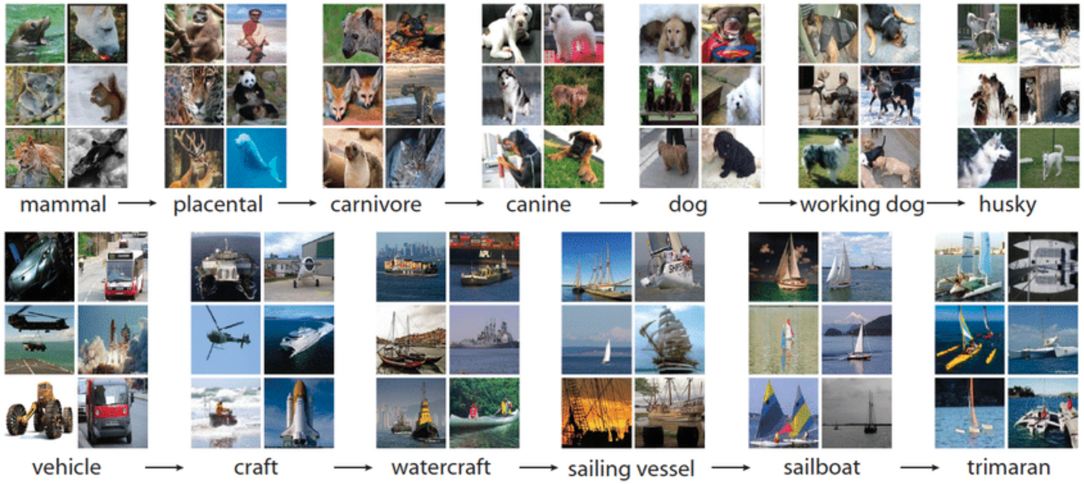
$$\min_{\mathbf{y}\in P(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{y}) < \min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{u}),$$



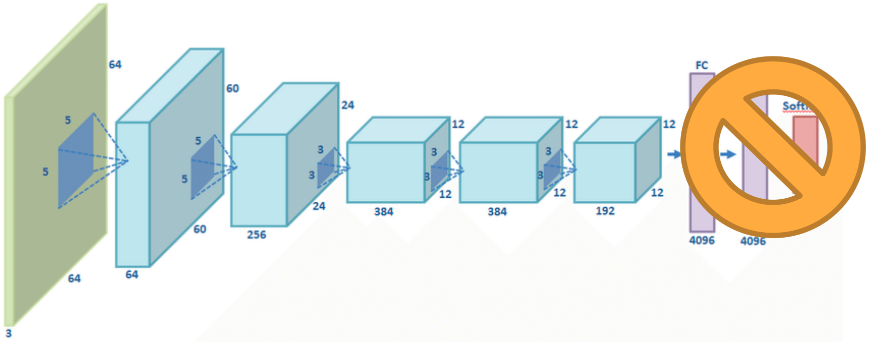
27



The first “deep” success

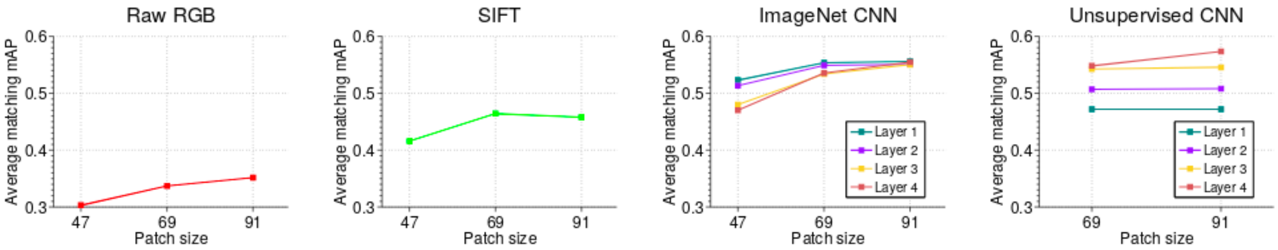


Get a network pre-trained on ImageNet



Remove FC layers & use features

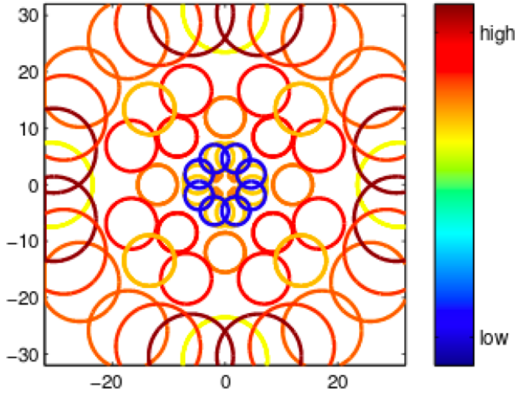
The first “deep” success



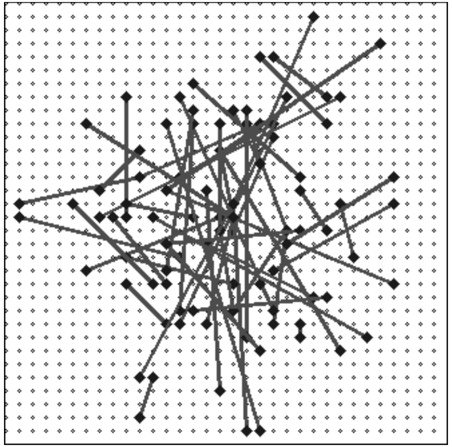
Fischer, Dosovitskiy, and Brox, **Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT**

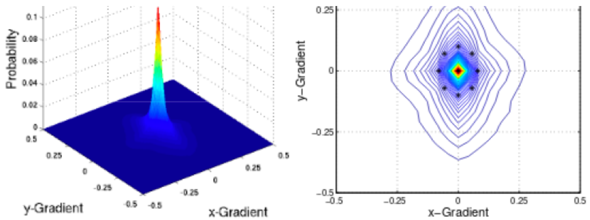
Learn optimal configuration of gaussian filters s.t.

$$\min_{\mathbf{y}\in P(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{y}) < \min_{\mathbf{u}\in N(\mathbf{x})} d_\eta(\mathbf{x},\mathbf{u}),$$

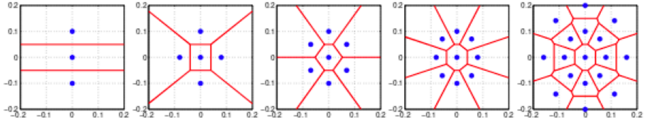


27

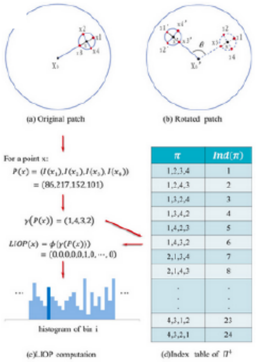


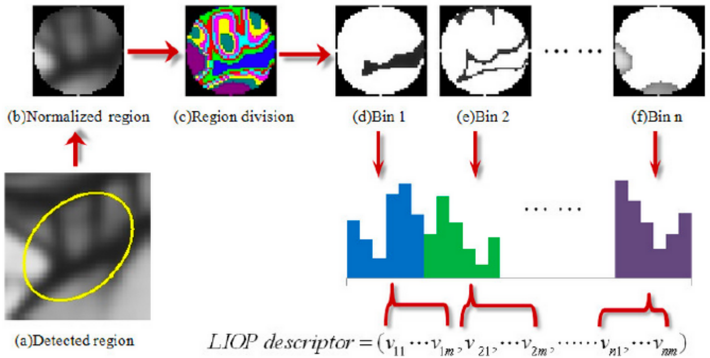


(a) (b)



(VQ-3) (VQ-5) (VQ-7) (VQ-9) (VQ-17)

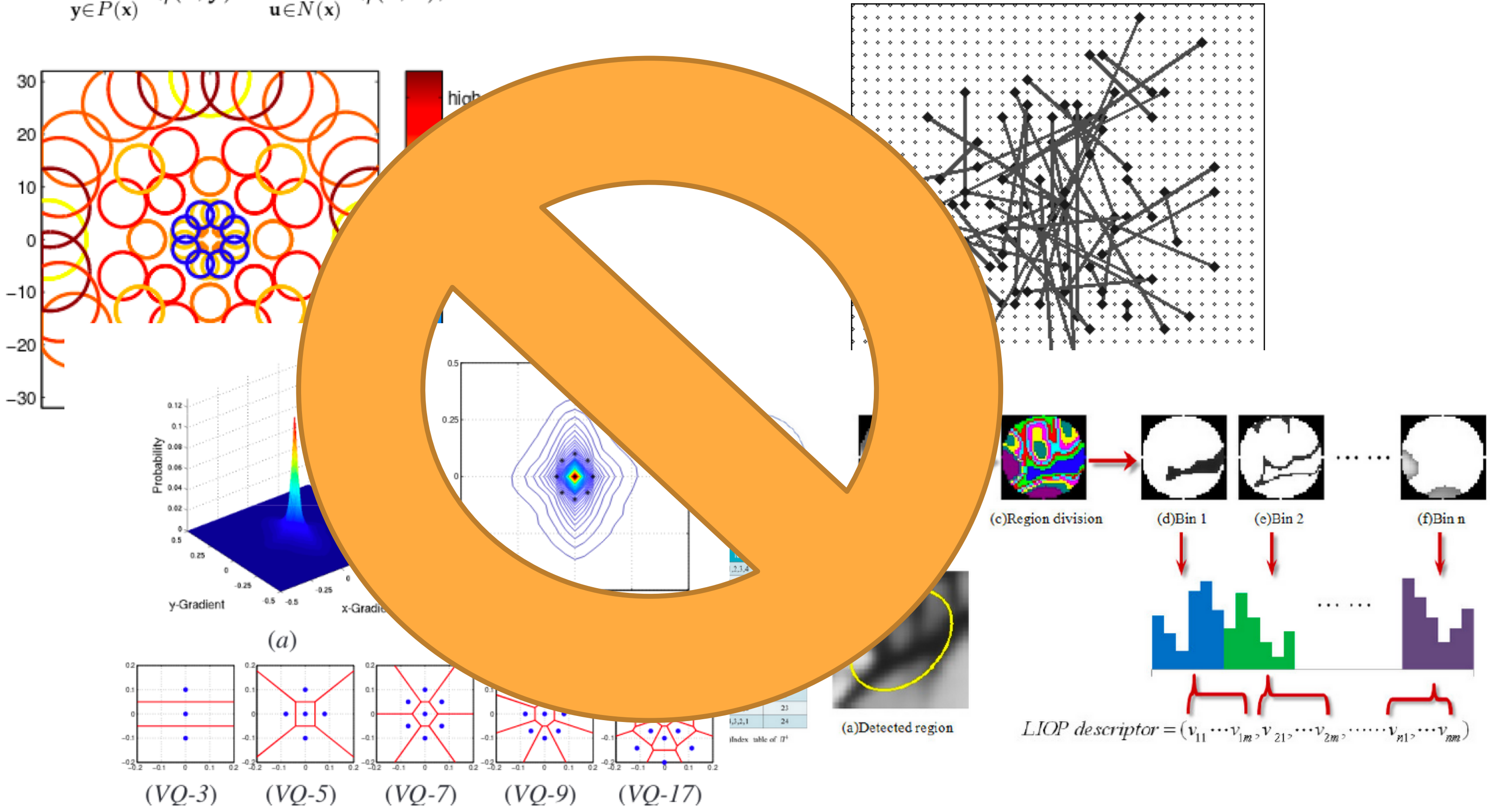




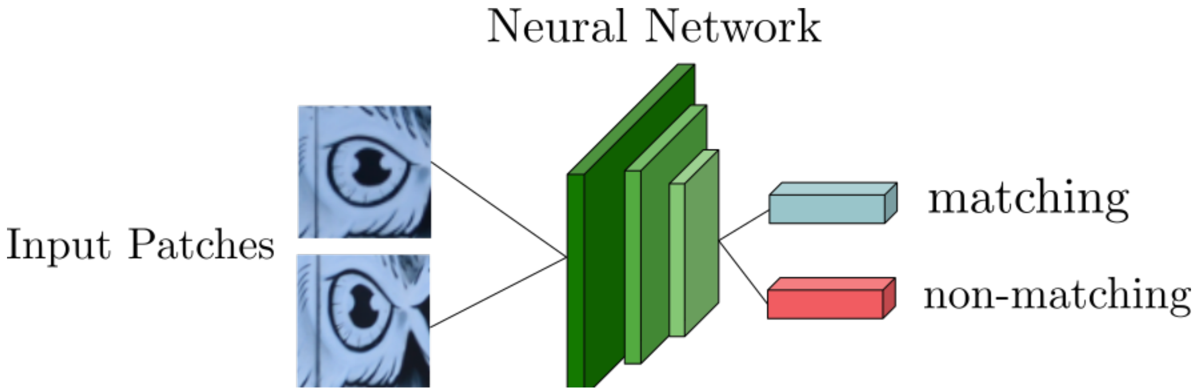
$LIOP\ descriptor = (v_{11} \cdots v_{1m}, v_{21}, \cdots v_{2m}, \cdots\cdots v_{n1}, \cdots v_{nm})$

Learn optimal configuration of gaussian filters s.t.

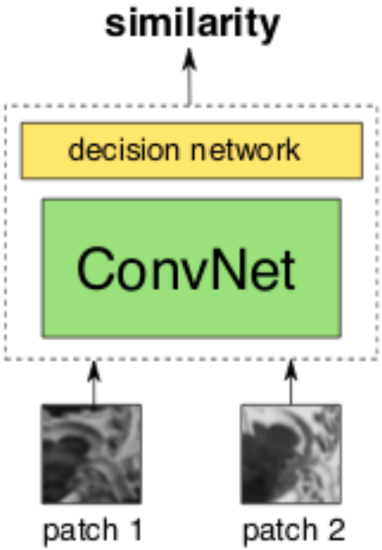
$$\min_{\mathbf{y} \in P(\mathbf{x})} d_\eta(\mathbf{x}, \mathbf{y}) < \min_{\mathbf{u} \in N(\mathbf{x})} d_\eta(\mathbf{x}, \mathbf{u}),$$



Deep learned descriptors



DeepCompare



$$\min_{w} \frac{\lambda}{2}\|w\|_2 + \sum_{i=1}^{N} \max(0, 1 - y_i o_i^{net})$$

Zagoruyko and Komodakis, **Learning to Compare Image Patches via Convolutional Neural Networks**

DeepCompare

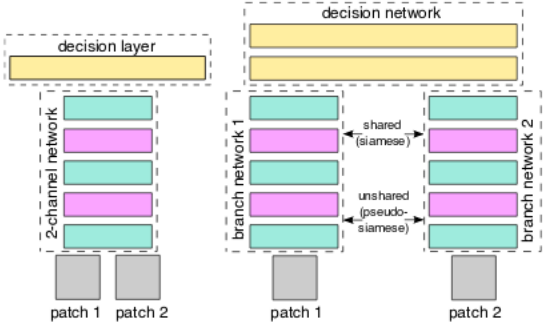


Figure 2. Three basic network architectures: 2-channel on the left, siamese and pseudo-siamese on the right (the difference between siamese and pseudo-siamese is that the latter does not have shared branches). Color code used: cyan = Conv+ReLU, purple = max pooling, yellow = fully connected layer (ReLU exists between fully connected layers as well).

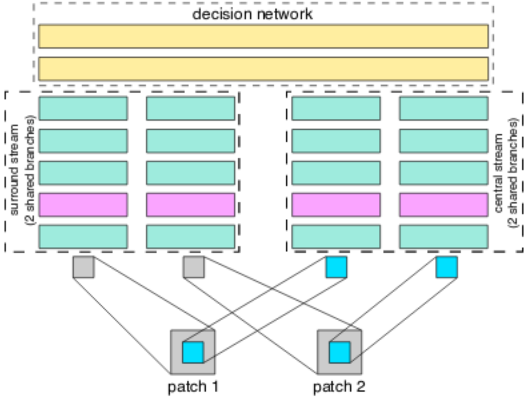
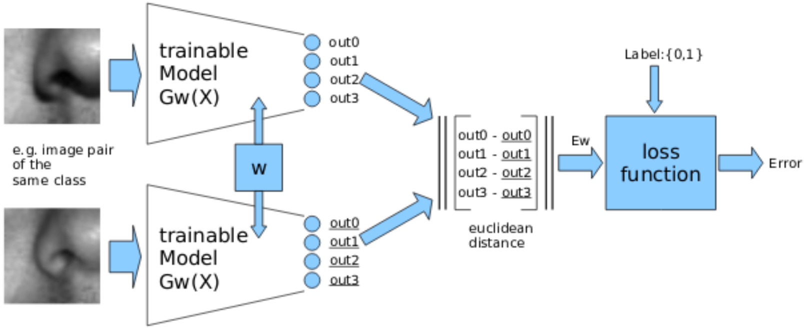


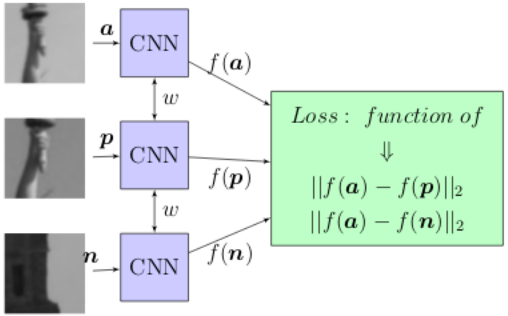
Figure 3. A central-surround two-stream network that uses a siamese-type architecture to process each stream. This results in 4 branches in total that are given as input to the top decision layer (the two branches in each stream are shared in this case).

Reminder: Early work (2008)



Jahrer, Grabner, and Bischof. **Learned local descriptors for recognition and matching**.

TFeat



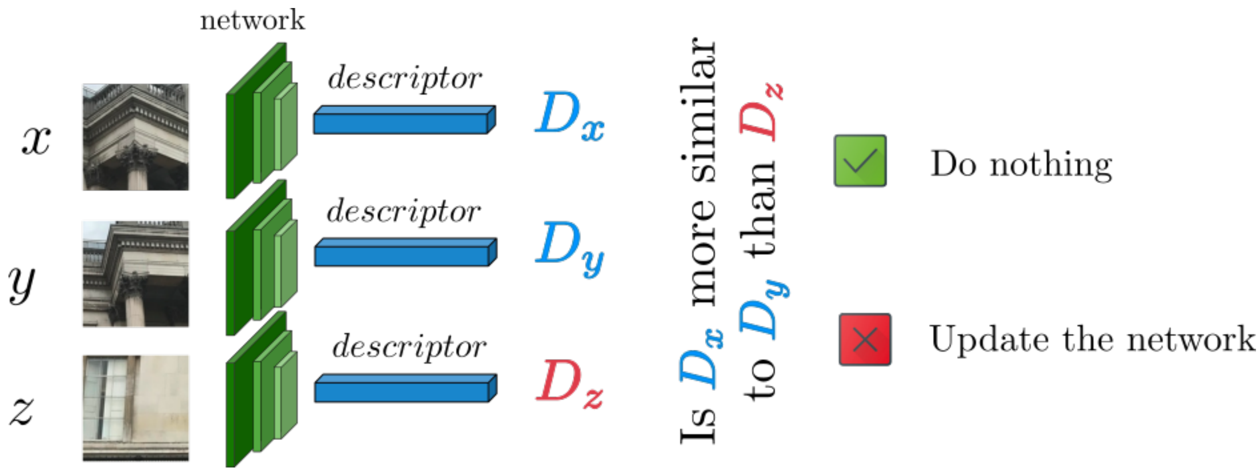
$$\sum_{i=1}^{N} l_{rank}(\delta_+, \delta_-) + \lambda \cdot ||\boldsymbol{w}||_2^2$$

where

$$l_{rank}(\delta_+, \delta_-) = max(0, \mu + \delta_+ - \delta_-)$$

Vassileios Balntas et al., **Learning local feature descriptors with triplets and shallow convolutional neural networks**

Triplet Learning



L2-Net

$$E_1 = -\frac{1}{2}\left(\sum_i \log s_{ii}^c + \sum_i \log s_{ii}^r\right)$$

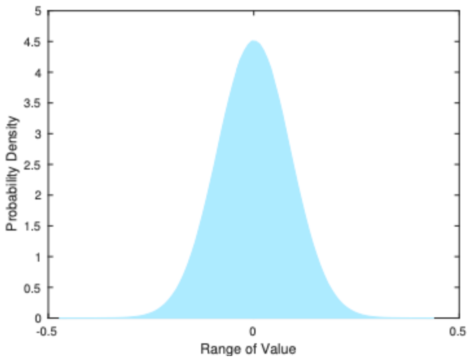
$$E_2 = \frac{1}{2}\left(\sum_{i\neq j} \left(r_{ij}^1\right)^2 + \sum_{i\neq j} \left(r_{ij}^2\right)^2\right)$$

$$E_3 = -\frac{1}{2}\left(\sum_i \log v_{ii}^c + \sum_i \log v_{ii}^r\right)$$

- E_1: Similarity loss
- E_2: Compactness loss
- E_3: Intermediate feature maps loss

Tian, Fan, and Wu. L2-Net: **Deep Learning of Discriminative Patch Descriptor in Euclidean Space**

Binary L2-Net

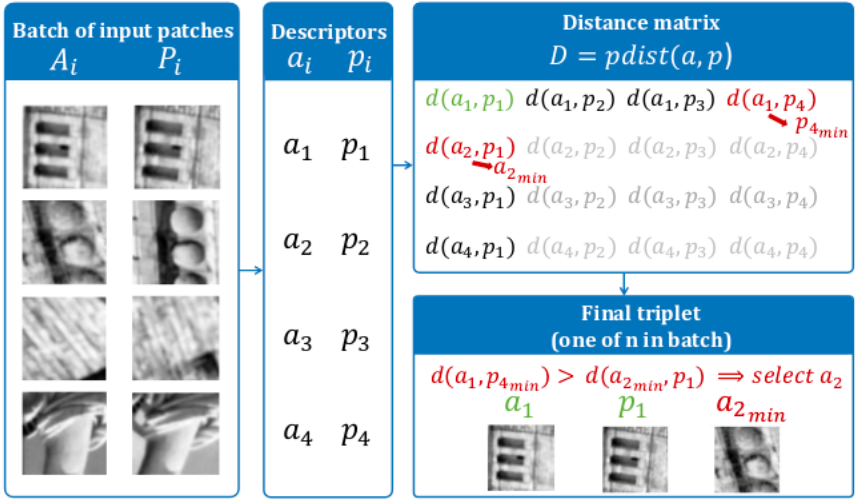


use sign of output

Test	Liberty	Notredame	Yosemite	Mean
L2-Net	4.16	1.54	4.41	3.37
L2-Net+	3.2	1.3	3.6	2.7
CS L2-Net	2.43	0.92	2.58	1.97
CS L2-Net+	**1.9**	**0.73**	**1.85**	**1.49**
Binary L2-Net	12.4	6.4	13.16	10.65
Binary L2-Net+	10.74	5.44	11.07	9.08
Binary CS L2-Net	6.43	2.88	6.91	5.4
Binary CS L2-Net+	**5.4**	**2.44**	**5.88**	**4.57**

Table 2. Performance of networks on the Brown dataset when they are trained on HPatches dataset .

HardNet



Mishchuk et al. **Working hard to know your neighbour's margins: Local descriptor learning loss**

HardNet

Table 1: Patch correspondence verification performance on the Brown dataset. We report false positive rate at true positive rate equal to 95% (FPR95). **Some papers report false discovery rate (FDR) instead of FPR due to bug in the source code**. For consistency we provide FPR, either obtained from the original article or re-estimated from the given FDR (marked with *). The best results are in **bold**.

Training	Notredame	Yosemite	Liberty	Yosemite	Liberty	Notredame	Mean	
Test	Liberty		Notredame		Yosemite		FDR	FPR
SIFT [9]	29.84		22.53		27.29			26.55
MatchNet*[14]	7.04	11.47	3.82	5.65	11.6	8.7	7.74	8.05
TFeat-M* [23]	7.39	10.31	3.06	3.8	8.06	7.24	6.47	6.64
PCW [33]	7.44	9.84	3.48	3.54	6.56	5.02		5.98
L2Net [24]	3.64	5.29	1.15	1.62	4.43	3.30		3.24
HardNetNIPS	3.06	4.27	0.96	1.4	3.04	2.53	3.00	2.54
HardNet	**1.47**	**2.67**	**0.62**	**0.88**	**2.14**	**1.65**		**1.57**
Augmentation: flip, 90° random rotation								
GLoss+[31]	3.69	4.91	0.77	1.14	3.09	2.67		2.71
DC2ch2st+[15]	4.85	7.2	1.9	2.11	5.00	4.10		4.19
L2Net+ [24] +	2.36	4.7	0.72	1.29	2.57	**1.71**		2.23
HardNet+NIPS	2.28	3.25	0.57	0.96	2.13	2.22	1.97	1.9
HardNet+	**1.49**	**2.51**	**0.53**	**0.78**	**1.96**	1.84		**1.51**

Mishchuk et al. **Working hard to know your neighbour's margins: Local descriptor learning loss**

SOSNet

First Order Similarity Loss

$$\mathcal{L}_{\text{FOS}} = \frac{1}{N}\sum_{i=1}^{N} \max\left(0, t + d_i^{\text{pos}} - d_i^{\text{neg}}\right)^2,$$
$$d_i^{\text{pos}} = d(\boldsymbol{x}_i, \boldsymbol{x}_i^+),$$
$$d_i^{\text{neg}} = \min_{\forall j, j \neq i}(d(\boldsymbol{x}_i, \boldsymbol{x}_j), d(\boldsymbol{x}_i, \boldsymbol{x}_j^+), d(\boldsymbol{x}_i^+, \boldsymbol{x}_j), d(\boldsymbol{x}_i^+, \boldsymbol{x}_j^+)),$$

Second Order Similarity Loss

$$d^{(2)}(\boldsymbol{x}_i, \boldsymbol{x}_i^+) = \sqrt{\sum_{j \neq i}^{N} (d(\boldsymbol{x}_i, \boldsymbol{x}_j) - d(\boldsymbol{x}_i^+, \boldsymbol{x}_j^+))^2},$$

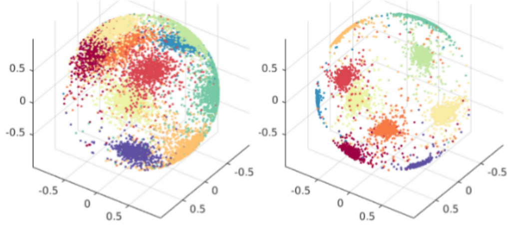
$$\mathcal{R}_{\text{SOS}} = \frac{1}{N}\sum_{i=1}^{N} d^{(2)}(\boldsymbol{x}_i, \boldsymbol{x}_i^+).$$

$$\mathcal{L}_{\text{T}} = \mathcal{L}_{\text{FOS}} + \mathcal{R}_{\text{SOS}},$$

SOSNet: Second Order Similarity Regularization for Local Descriptor Learning

Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, Vassileios Balntas

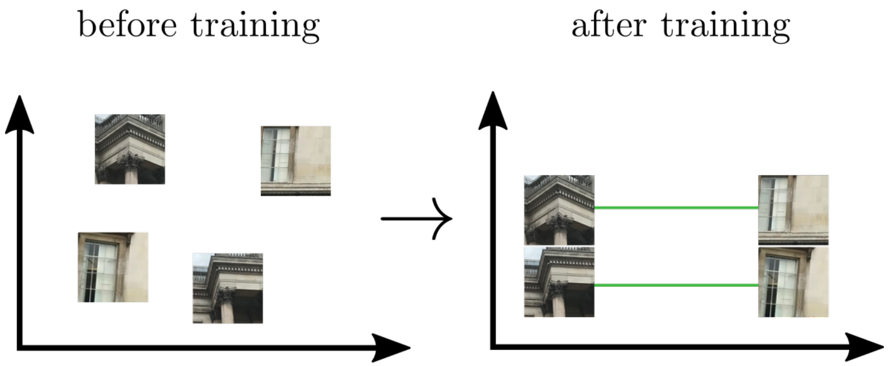
SOSNet



(a) Triplet Loss (b) Triplet Loss + SOSR

Figure 1. Qualitative results of our proposed SOSR on features learned for the 10 digits of the MNIST [19] dataset. Each digit is represented by a different colour on the unit sphere. We can observe that by using our SOSR method that encourages second order similarity, more compact individual clusters are learned compared to standard triplet loss.

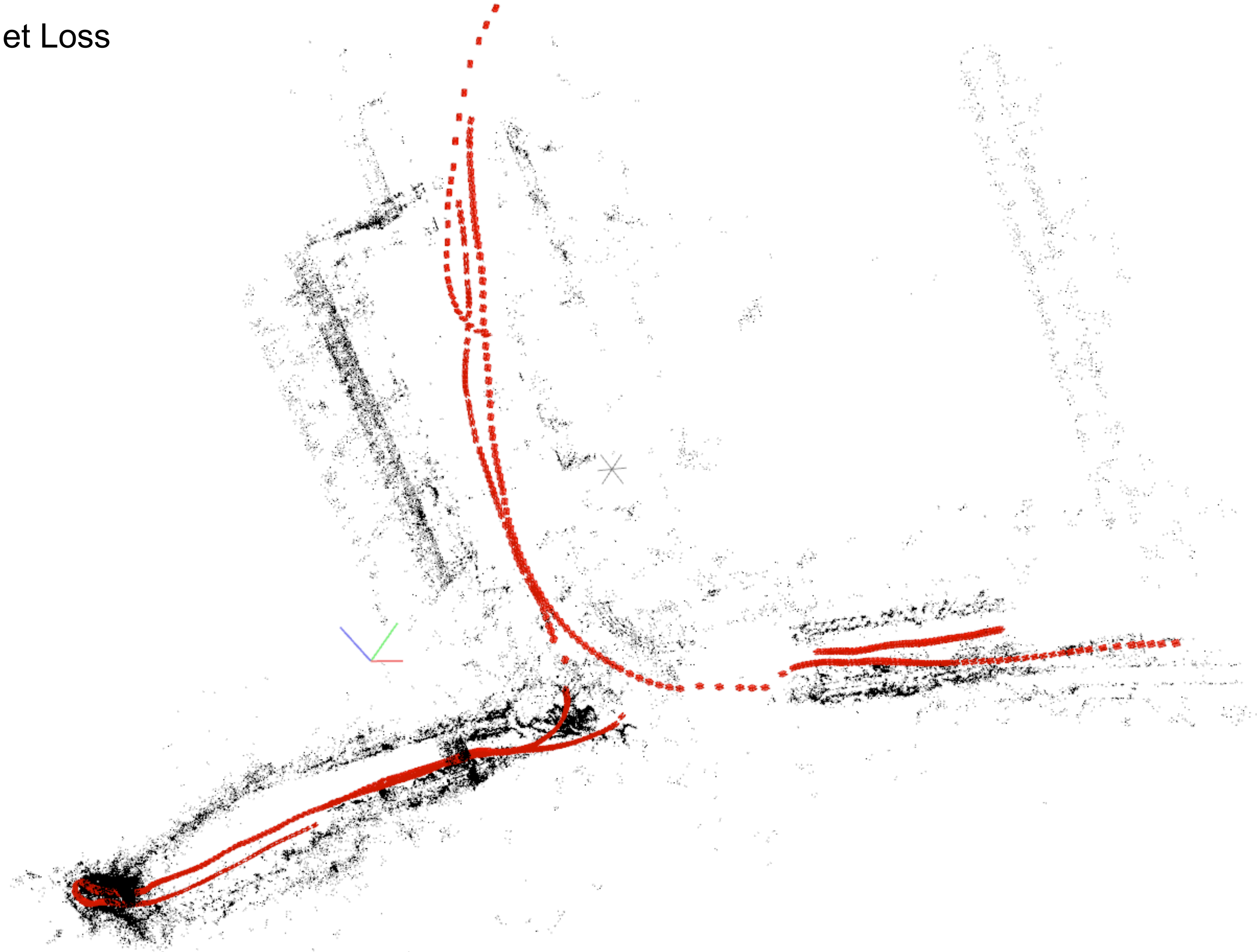
Second order consistency between classes



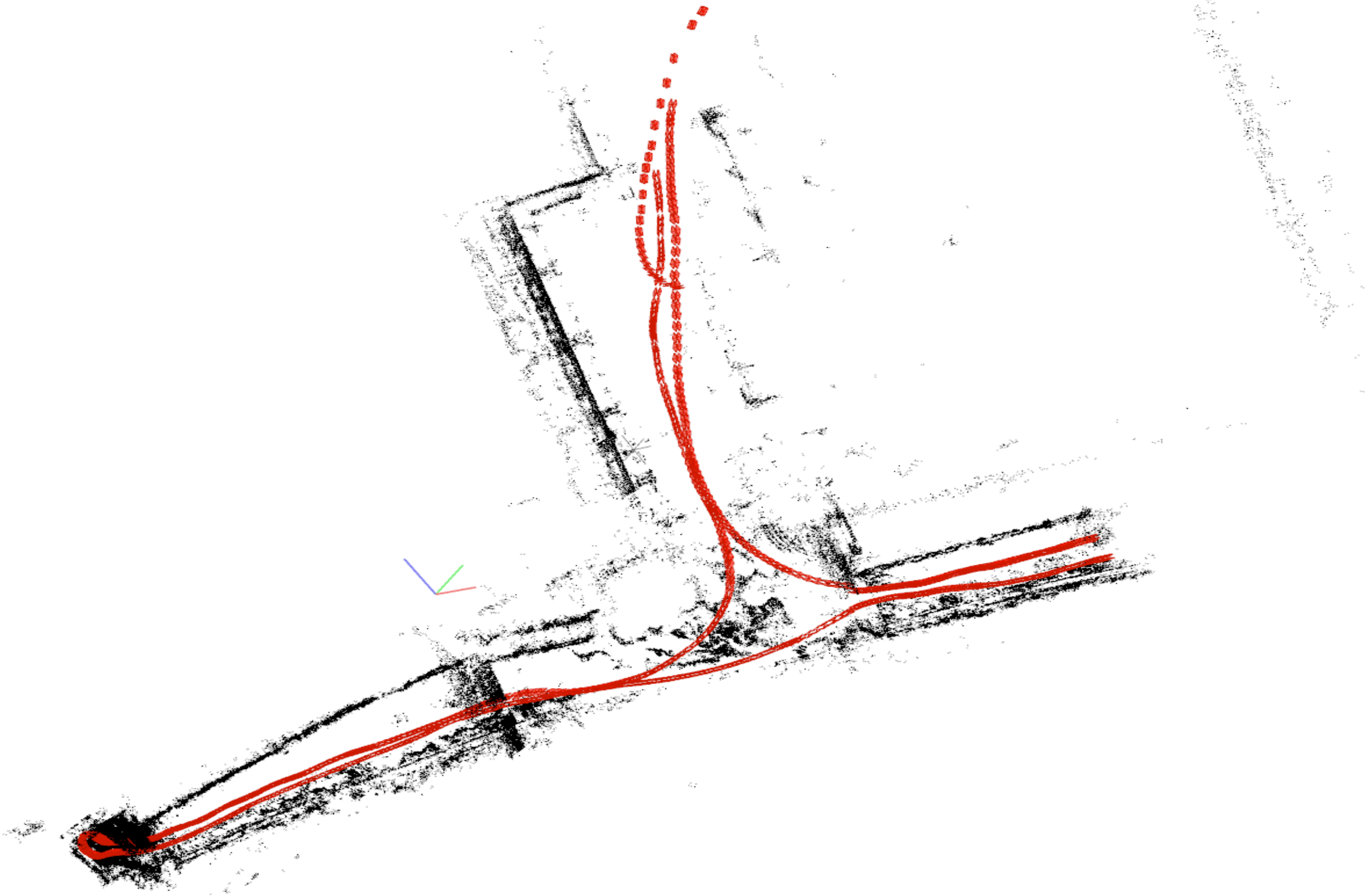
SOSNet: Second Order Similarity Regularization for Local Descriptor Learning

Yurun Tian, Xin Yu, Bin Fan, Fuchao Wu, Huub Heijnen, Vassileios Balntas

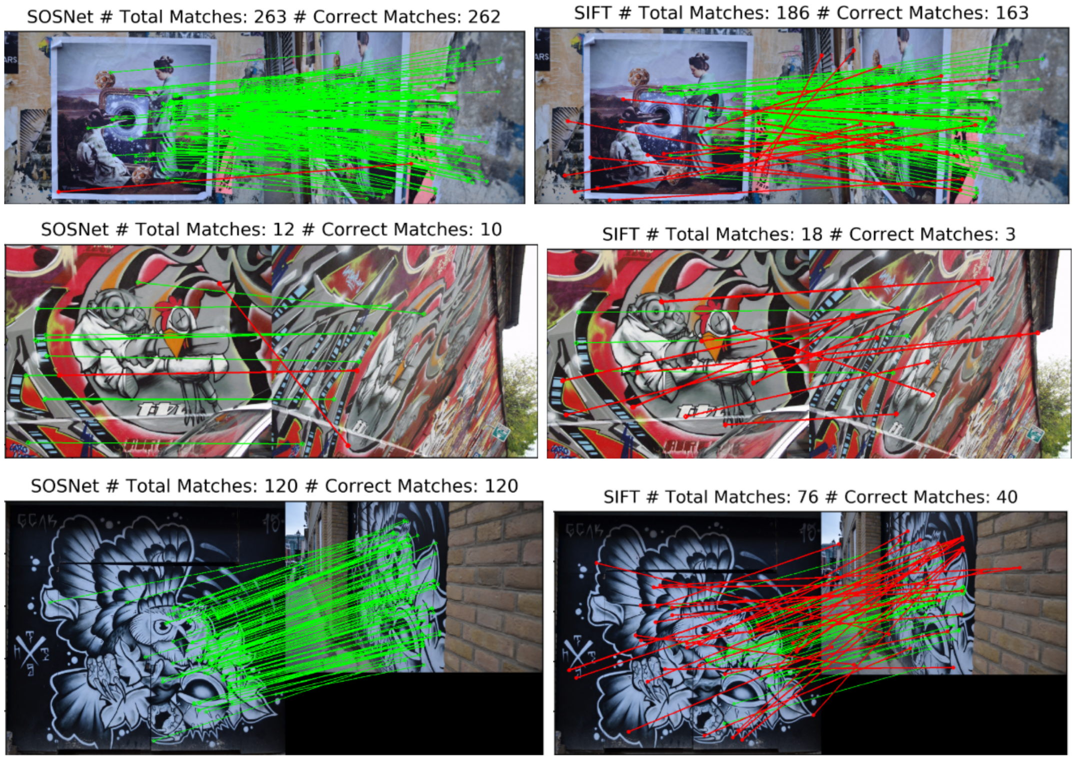
Triplet Loss



SOS



Current status: classical pipeline



Limits of the “classical pipeline”

Classical pipeline

Classical pipeline replacement?

Classical pipeline replacement?

*Maybe some parts only

Limits of the “classical pipeline”

- New methods are needed that are based on modern networks, including end to end training of networks
- Need to abstract more than the “keypoint” & “patch” paradigms.

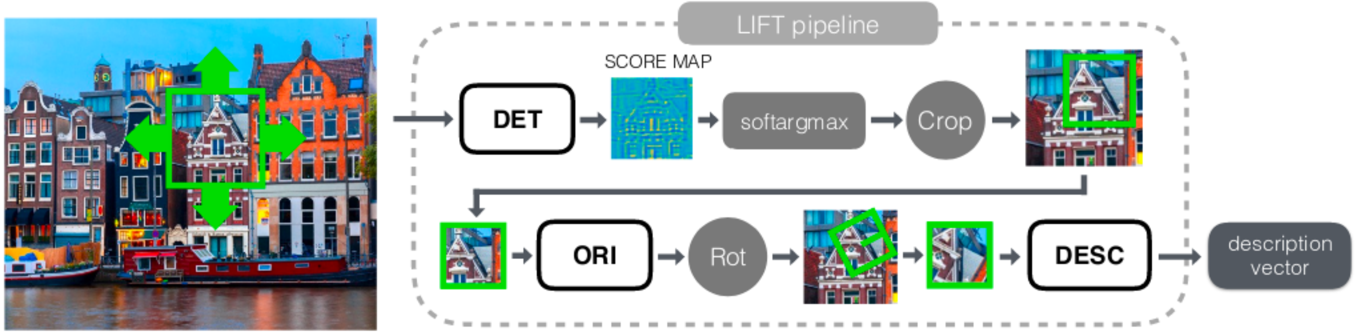
“Modern” Methods

Image: Wikipedia

Modern methods

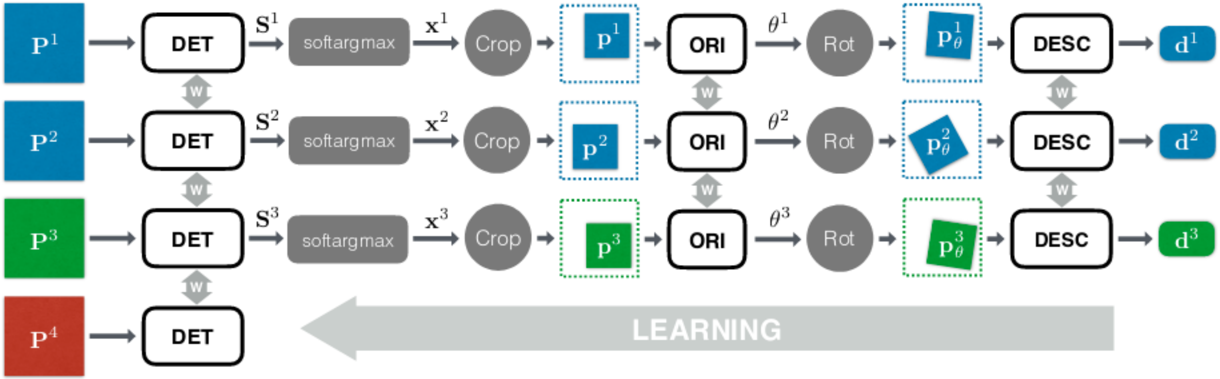
- Replace some/all parts of the classical pipeline
- Focus on training as much as possible end-to-end
- Focus on new matching methods, other than *argmins* of distance matrix

LIFT



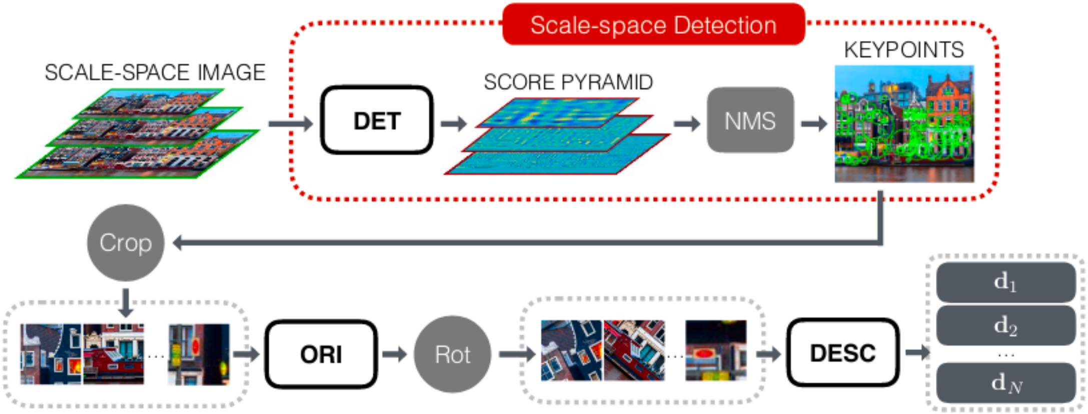
Yi et al., **LIFT: Learned Invariant Feature Transform**

LIFT



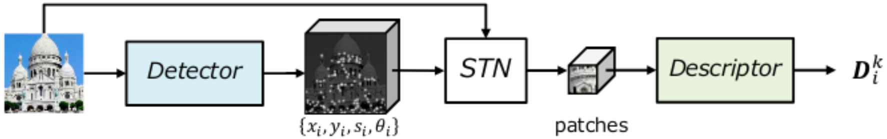
P^1
DET
S^1
softargmax
x^1
Crop
p^1
ORI
θ^1
Rot
p^1_θ
DESC
d^1
P^2
DET
S^2
softargmax
x^2
Crop
p^2
ORI
θ^2
Rot
p^2_θ
DESC
d^2
P^3
DET
S^3
softargmax
x^3
Crop
p^3
ORI
θ^3
Rot
p^3_θ
DESC
d^3
P^4
DET
W
LEARNING

LIFT

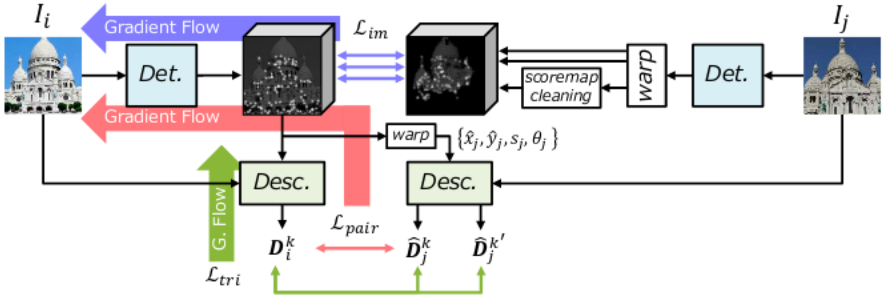


Scale-space Detection
SCALE-SPACE IMAGE
DET
SCORE PYRAMID
NMS
KEYPOINTS
Crop
ORI
Rot
DESC
$\mathbf{d}_1$
$\mathbf{d}_2$
...
$\mathbf{d}_N$

LF-Net

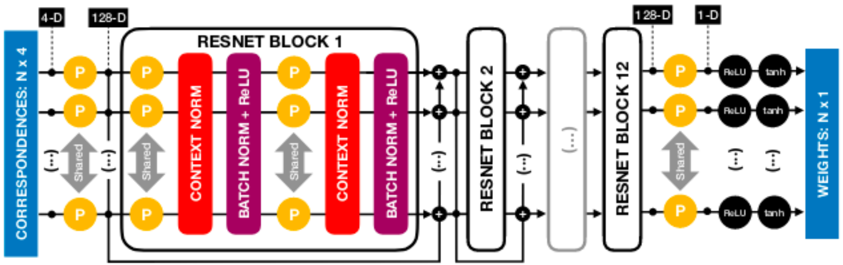


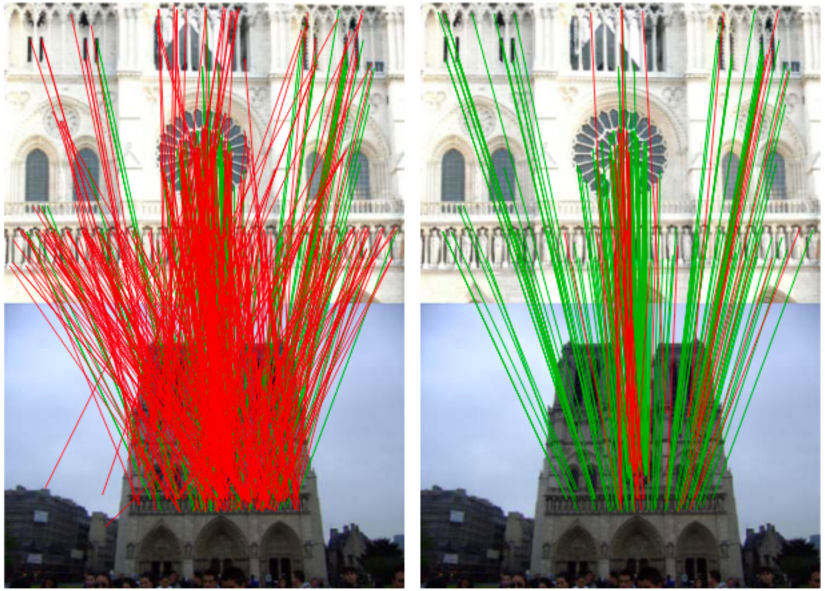
(a) The LF-Net architecture. The *detector* network generates a scale-space score map along with dense orientation estimates, which are used to select the keypoints. Image patches around the chosen keypoints are cropped with a differentiable sampler (STN) and fed to the *descriptor* network, which generates a descriptor for each patch.



Ono et al., **LF-Net: Learning Local Features from Images**

Learning correspondences





(a) RANSAC (b) Our approach

Yi et al., **Learning to Find Good Correspondences**

Superpoint

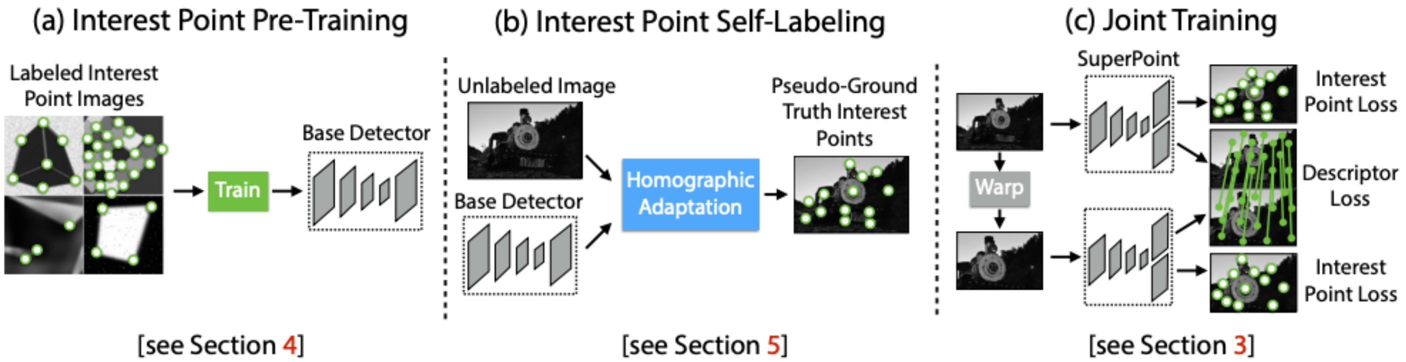
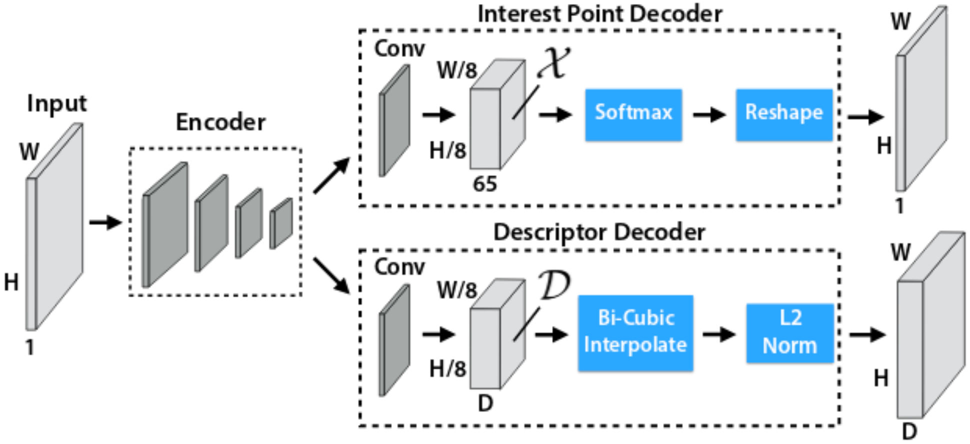


Figure 2. **Self-Supervised Training Overview.** In our self-supervised approach, we (a) pre-train an initial interest point detector on synthetic data and (b) apply a novel Homographic Adaptation procedure to automatically label images from a target, unlabeled domain. The generated labels are used to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

DeTone, Malisiewicz, and Rabinovich, SuperPoint: Self-Supervised
Interest Point Detection and Description

Superpoint



DeTone, Malisiewicz, and Rabinovich, SuperPoint: Self-Supervised
Interest Point Detection and Description

Implicitly Matched Interest Points (IMIPs)

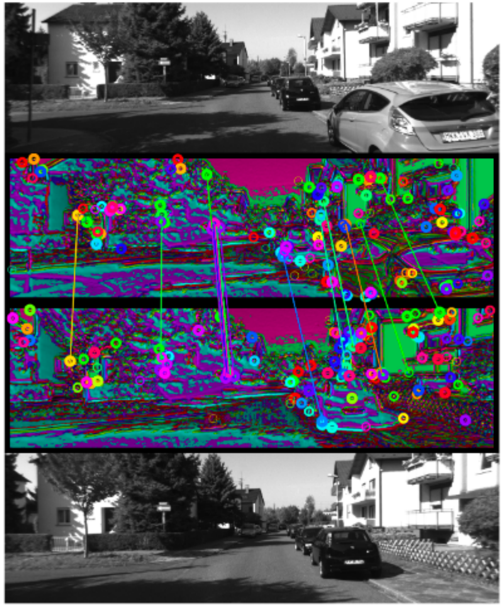
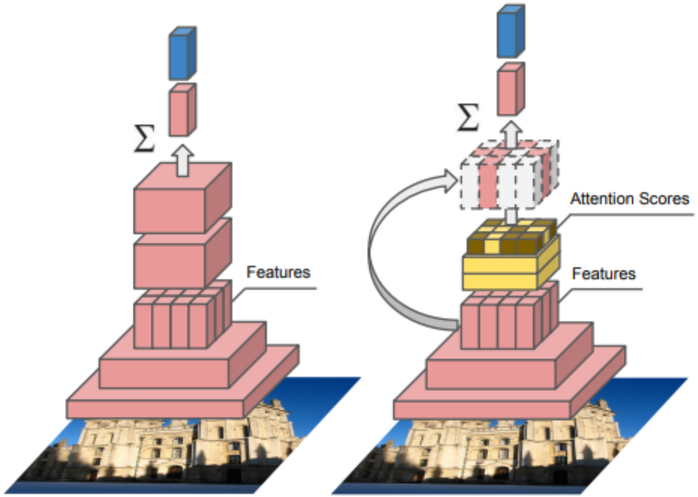


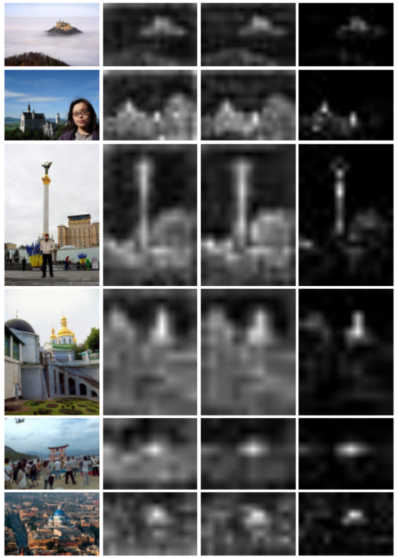
Figure 1. We propose a CNN interest point detector which provides implicitly matched interest points — descriptors are not needed for matching. This image illustrates the output of the final layer, which determines the interest points. Hue indicates which channel has the strongest response for a given pixel, and brightness indicates that response. Circles indicate the 128 interest points, which are the global maxima of each channel, circle thicknesses indicate confidence in a point. Lines indicate inlier matches after P3P localization.

Cieslewski, Bloesch, and Scaramuzza, **Matching Features without Descriptors: Implicitly Matched Interest Points**

DELF



(a) Descriptor Fine-tuning (b) Attention-based Training



(a) (b) (c) (d)

Figure 9: Comparison of keypoint selection methods. (a) Input image (b) L_2 norm scores using the pretrained model (DELF-noFT) (c) L_2 norm scores using fine-tuned descriptors (DELF+FT) (d) Attention-based scores (DELF+FT+ATT). Our attention-based model effectively disregards clutter compared to other options.

“Attention” as weighting for global descriptor

Noh et al., SuperPoint: Large-Scale Image Retrieval with Attentive Deep Local Features
ICCV 2017

D2Net

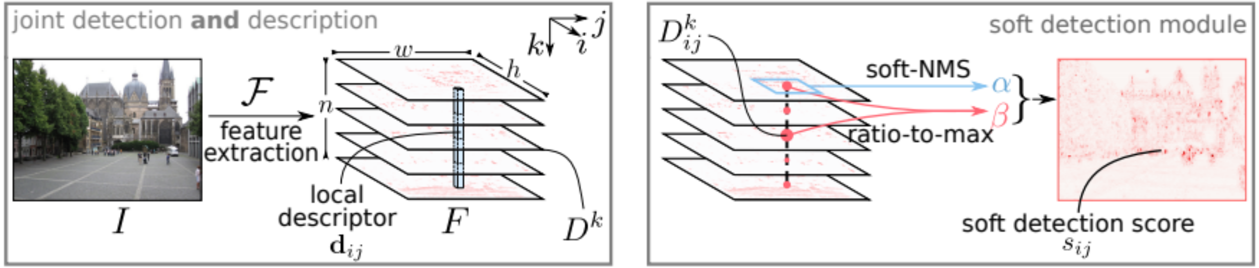
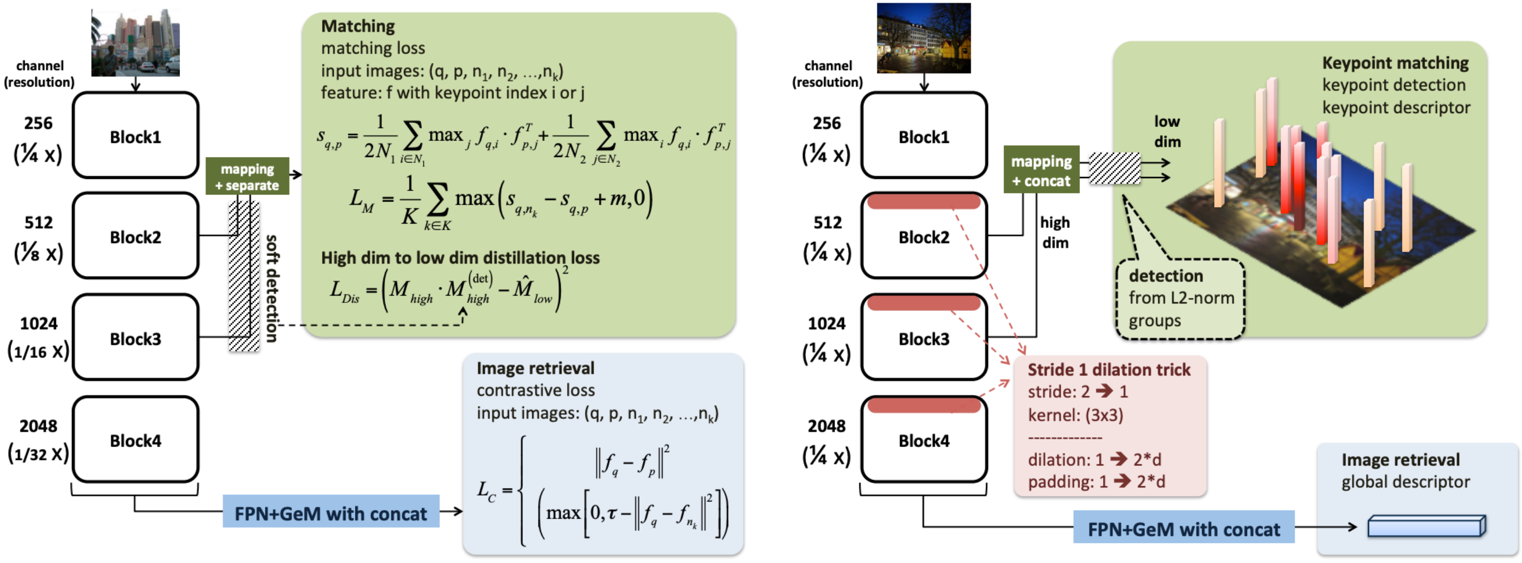


Figure 3: **Proposed detect-and-describe (D2) network.** A feature extraction CNN $\mathcal{F}$ is used to extract feature maps that play a dual role: (i) local descriptors $\mathbf{d}_{ij}$ are simply obtained by traversing all the n feature maps D^k at a spatial position (i, j); (ii) detections are obtained by performing a non-local-maximum suppression on a feature map followed by a non-maximum suppression across each descriptor - during training, keypoint detection scores s_{ij} are computed from a soft local-maximum score α and a ratio-to-maximum score per descriptor β.

$$\mathcal{L}(I_1, I_2) = \sum_{c \in \mathcal{C}} \frac{s_c^{(1)} s_c^{(2)}}{\sum_{q \in \mathcal{C}} s_q^{(1)} s_q^{(2)}} m(p(c), n(c)) \ ,$$

Dusmanu et al, D2-Net: A Trainable CNN for Joint Description and Detection of Local Features CVPR 2019

UR2KiD



$$s_{q,p} = \frac{1}{2N_1}\sum_{i\in N_1}\max_j f_{q,i}\cdot f_{p,j}^T + \frac{1}{2N_2}\sum_{j\in N_2}\max_i f_{q,i}\cdot f_{p,j}^T$$

$$L_M = \frac{1}{K}\sum_{k\in K}\max\left(s_{q,n_k} - s_{q,p} + m, 0\right)$$

$$L_{Dis} = \left(M_{high}\cdot M_{high}^{(\mathrm{det})} - \hat{M}_{low}\right)^2$$

$$L_C = \begin{cases} \|f_q - f_p\|^2 \\ \left(\max\left[0, \tau - \|f_q - f_{n_k}\|^2\right]\right) \end{cases}$$

(a) Training (b) Testing

Figure 2. Overview of the proposed method.

Yang et al. **UR2KiD: Unifying Retrieval, Keypoint Detection, and Keypoint Description without Local Correspondence Supervision**

UR2KiD

(a) Pre-trained ResNet101

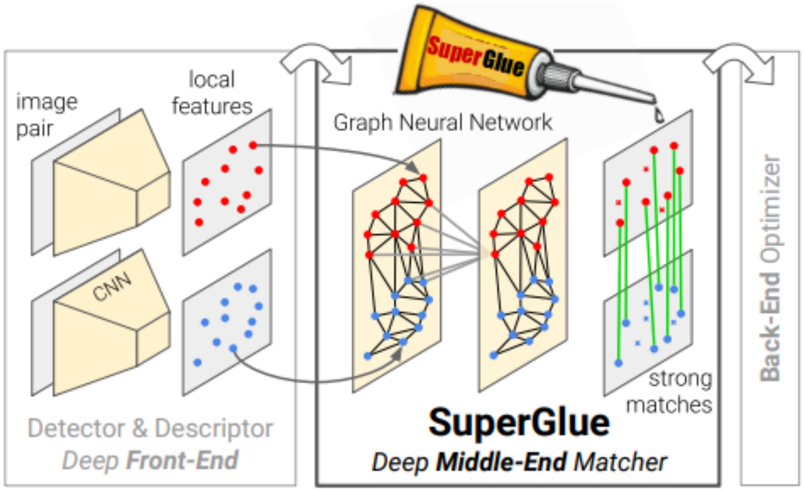
(b) D2-Net [15]

(c) **UR2KID (ours)**

Figure 1. Extremely challenging image matching scenario with severe scale change and significant scene difference between day and night. The proposed UR2KID method is able to utilize a common network structure to achieve state-of-the-art results.

Yang et al. **UR2KiD: Unifying Retrieval, Keypoint Detection, and Keypoint Description without Local Correspondence Supervision**

SuperGlue



Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich: SuperGlue: Learning Feature Matching with Graph Neural Networks – CVPR 2020

SuperGlue

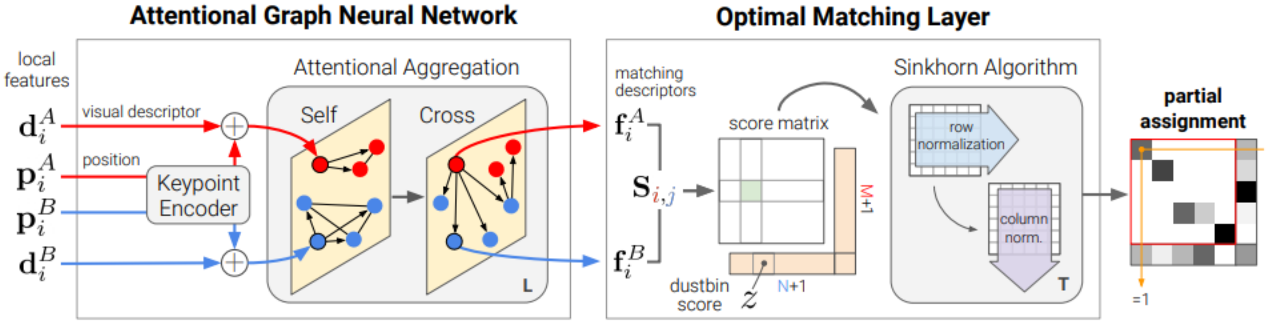
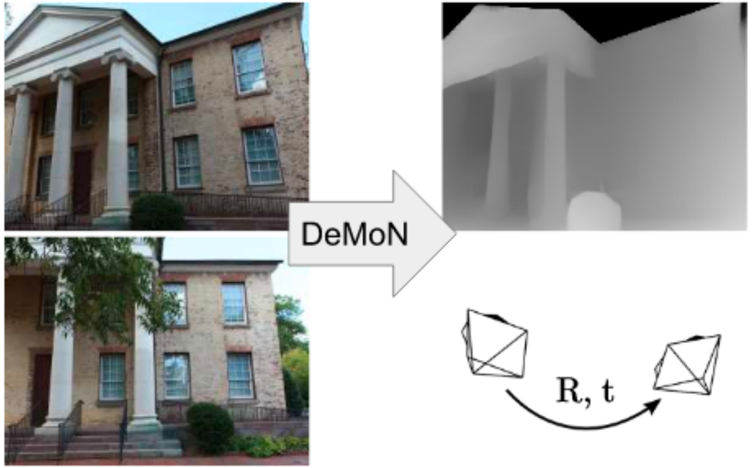


Figure 3: **The SuperGlue architecture.** SuperGlue is made up of two major components: the *attentional graph neural network* (Section 3.1), and the *optimal matching layer* (Section 3.2). The first component uses a *keypoint encoder* to map keypoint positions $\mathbf{p}$ and their visual descriptors $\mathbf{d}$ into a single vector, and then uses alternating self- and cross-attention layers (repeated L times) to create more powerful representations $\mathbf{f}$. The optimal matching layer creates an M by N score matrix, augments it with dustbins, then finds the optimal partial assignment using the Sinkhorn algorithm (for T iterations).

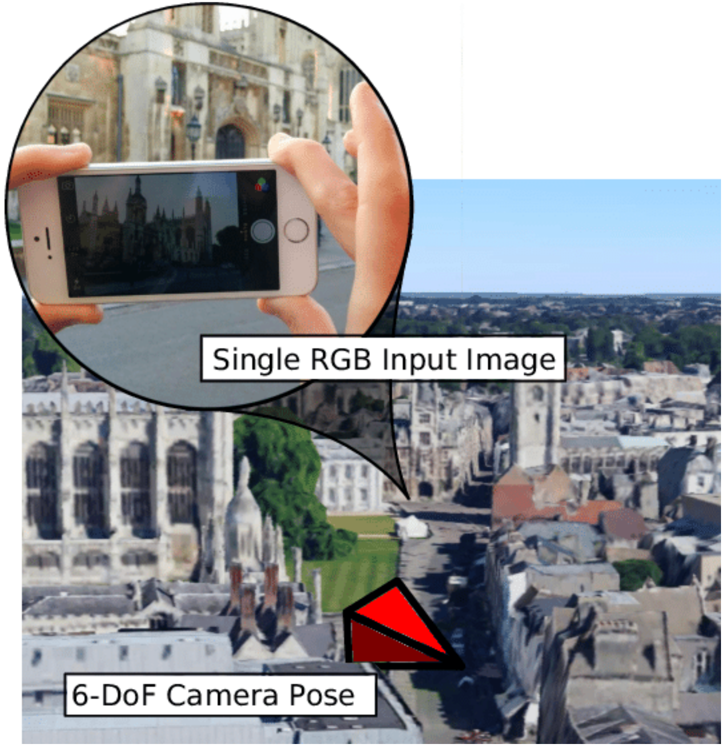
Paul-Edouard Sarlin, Daniel DeTone, Tomasz Malisiewicz, Andrew Rabinovich: SuperGlue: Learning Feature Matching with Graph Neural Networks – CVPR 2020

DeMoN



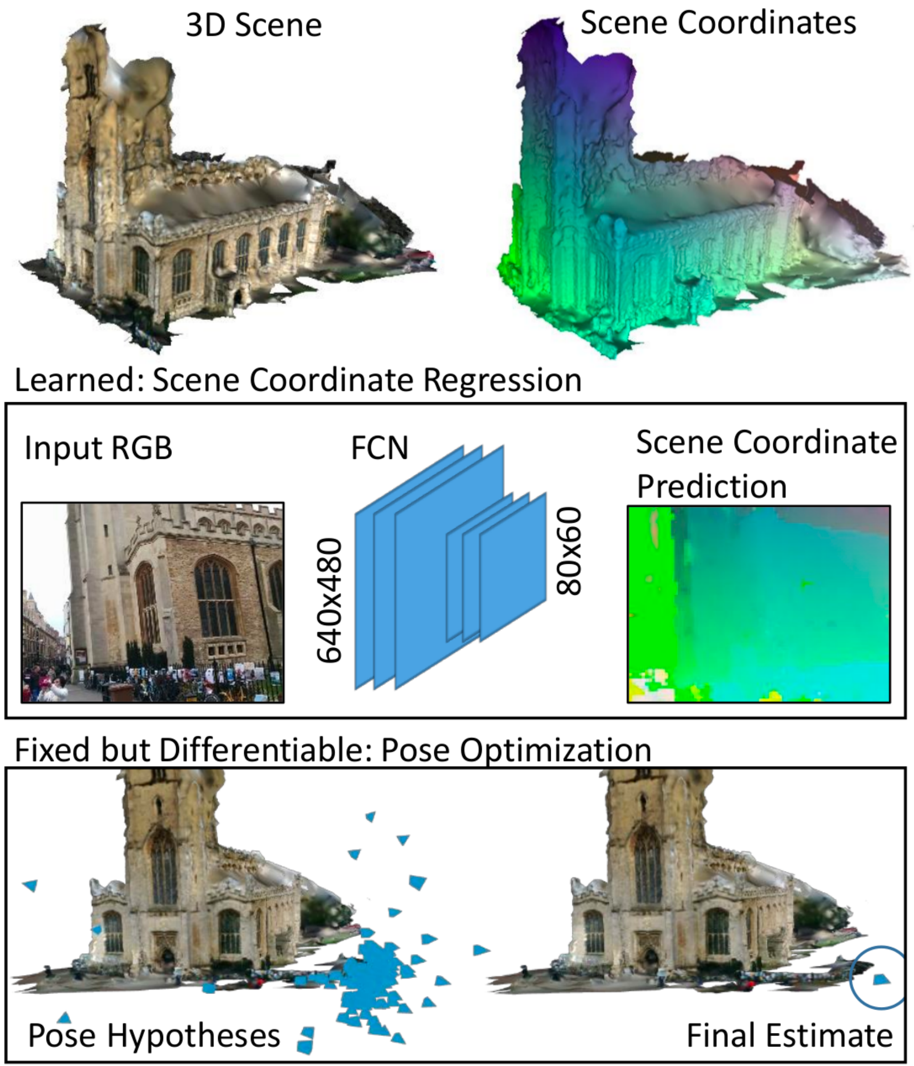
Ummenhofer et al., **DeMoN: Depth and Motion Network for Learning Monocular Stereo**

PoseNet



Kendall, Grimes, and Cipolla, **PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization**

Local scene coordinates

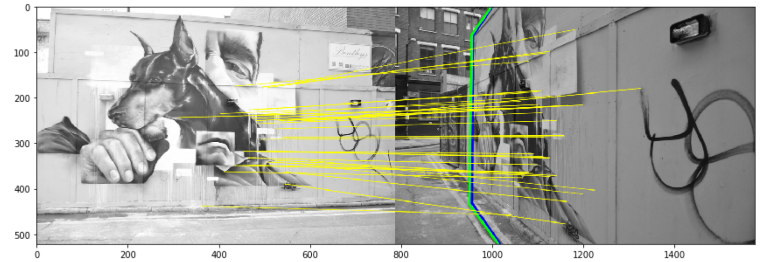


Brachmann and Rother. **Learning Less is More - 6D Camera Localization via 3D Surface Regression**

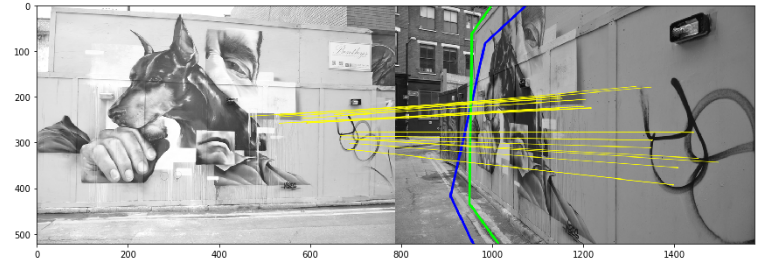
How good are modern methods?


FOOD

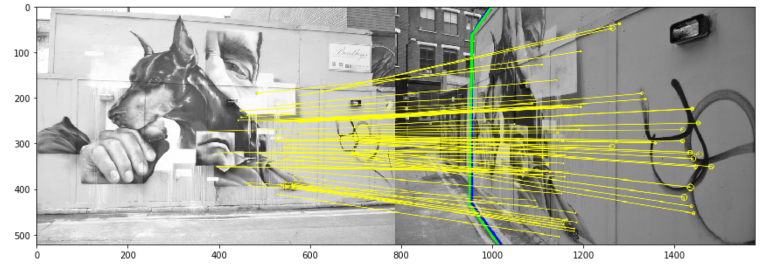
“Classical” methods are still quite strong



SuperPoint: 51 inliers



D2Net: 26 inliers, incorrect geometry



DoG + HardNet: 123 inliers

More later @ Dmytro’s talk!

State-of-the art & future challenges - open questions

- How can the current matching paradigm be improved?
- Do we still need local features?
- Are dense descriptors using FCN needed?
- Are attention models related to detectors?
- Is end-to-end learning of every stage the best solution?
- How to add semantics into the pipeline?

Programme

- 09:00 – 10:00 Overview of classical & end-to-end methods
- 10:00 – 11:00 Local features: from paper to practice
- 11:00 – 12:00 Kornia introduction & hands-on Session