Local Features:

Past, Present & Future

WACV 2019 Tutorial

Vassileios Balntas
Scape Technologies

vbalnt.github.io
vassileios@scape.io

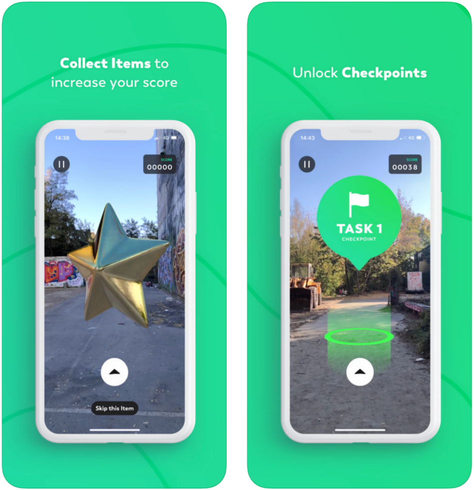
SCAPE

Matching images

Applications

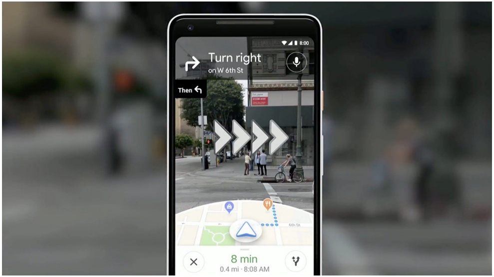
- 3D Reconstructions
- Self-driving cars
- Augmented Reality
- Assistance for Visually Impaired

Augmented Reality



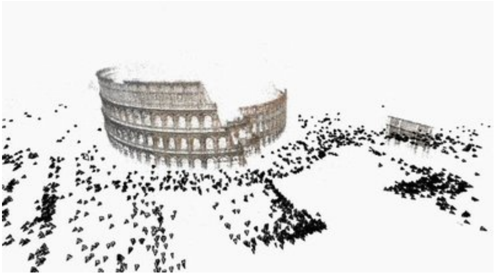
ScavengAR App

Assistance



Google Maps AR

Building Rome in a few hours



Building Rome in a Day - *University of Washington & Microsoft Research*

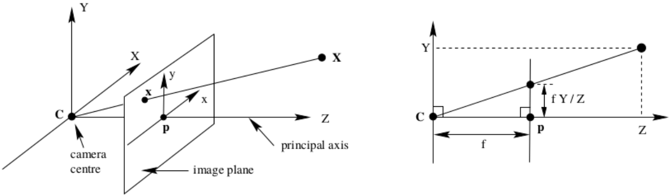
Image Matching - Practicality

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

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 discuss a few things about cameras.

Pinhole Camera Model



World Point

$$\boldsymbol{X} = (X, Y, Z)^T$$

Image Point

$$\boldsymbol{x} = (\frac{fX}{Z}, \frac{fY}{Z}, f)^T$$

[1]

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

Pinhole Camera Model

Homogeneous Coordinates Mapping

$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Pinhole Camera Model

World Point

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

Non-zero principal point

$$\boldsymbol{x} = \left(\frac{fX}{Z} + p_x, \frac{fY}{Z} + p_y, f\right)^T$$

Homogeneous Coordinates Mapping

$$\begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{pmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}$$

Forward and Backward Projections

Forward Projection (world point to image point)

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

Backward Projection (image point to world point)

$$\boldsymbol{X}(\lambda) = P^{+}\boldsymbol{x} + \lambda\boldsymbol{C}$$
$$P^{+}P = I$$

Epipolar Geometry

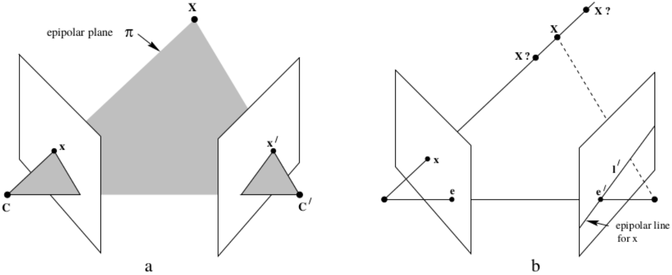


Fig. 9.1. **Point correspondence geometry.** *(a) The two cameras are indicated by their centres* $\mathbf{C}$ *and* $\mathbf{C}'$ *and image planes. The camera centres, 3-space point* $\mathbf{X}$*, and its images* $\mathbf{x}$ *and* $\mathbf{x}'$ *lie in a common plane* $\boldsymbol{\pi}$*. (b) An image point* $\mathbf{x}$ *back-projects to a ray in 3-space defined by the first camera centre,* $\mathbf{C}$*, and* $\mathbf{x}$*. This ray is imaged as a line* $\mathbf{l}'$ *in the second view. The 3-space point* $\mathbf{X}$ *which projects to* $\mathbf{x}$ *must lie on this ray, so the image of* $\mathbf{X}$ *in the second view must lie on* $\mathbf{l}'$*.*

Epipolar Geometry

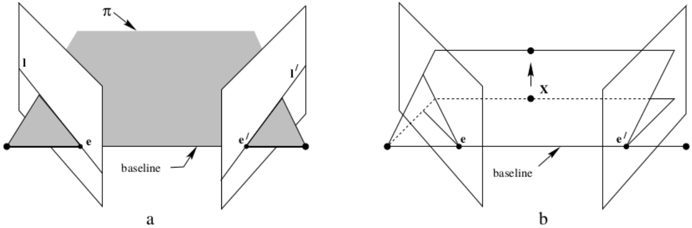


Fig. 9.2. **Epipolar geometry.** *(a) The camera baseline intersects each image plane at the epipoles* $\mathbf{e}$ *and* $\mathbf{e}'$. *Any plane* π *containing the baseline is an epipolar plane, and intersects the image planes in corresponding epipolar lines* $\mathbf{l}$ *and* $\mathbf{l}'$. *(b) As the position of the 3D point* $\mathbf{X}$ *varies, the epipolar planes "rotate" about the baseline. This family of planes is known as an epipolar pencil. All epipolar lines intersect at the epipole.*

Fundamental Matrix F

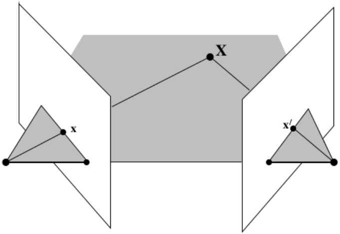


Fig. 10.1. **Triangulation.** *The image points* $\mathbf{x}$ *and* $\mathbf{x}'$ *back project to rays. If the epipolar constraint* $\mathbf{x}'^{\mathsf{T}}\mathrm{F}\mathbf{x} = 0$ *is satisfied, then these two rays lie in a plane, and so intersect in a point* $\mathbf{X}$ *in 3-space.*

For all corresponding points $x \leftrightarrow y$ in two images,

$$x^T F y = 0$$

We can find F by only using pairs of matching points.

3D Reconstruction

Given a set of N correspondences $\boldsymbol{x}_i \leftrightarrow \boldsymbol{x}_i^{'}$, find camera matrices P and $P^{'}$ and the 3D points $\boldsymbol{X}_i$ s.t.

$$\begin{aligned} \boldsymbol{x}_i &= P\boldsymbol{X}_i \\ \boldsymbol{x}_i^{'} &= P^{'}\boldsymbol{X}_i^{'} \\ \forall i &\in [1, N] \end{aligned}$$

3D Reconstruction

- Get point correspondences
- Compute F
- Compute camera matrices
- For each point correspondence, compute the point in space that projects to the two image points

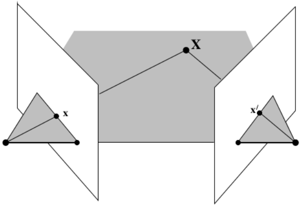


Fig. 10.1. **Triangulation.** *The image points* $\mathbf{x}$ *and* $\mathbf{x}'$ *back project to rays. If the epipolar constraint* $\mathbf{x}'^{\mathsf{T}} \mathrm{F} \mathbf{x} = 0$ *is satisfied, then these two rays lie in a plane, and so intersect in a point* $\mathbf{X}$ *in 3-space.*

Computation of the Fundamental Matrix F

Objective

Given $n \geq 8$ image point correspondences $\{\mathbf{x}_i \leftrightarrow \mathbf{x}'_i\}$, determine the fundamental matrix F such that $\mathbf{x}'^{\mathsf{T}}_i \mathtt{F} \mathbf{x}_i = 0$.

Algorithm

(i) **Normalization:** Transform the image coordinates according to $\hat{\mathbf{x}}_i = \mathtt{T}\mathbf{x}_i$ and $\hat{\mathbf{x}}'_i = \mathtt{T}'\mathbf{x}'_i$, where T and T′ are normalizing transformations consisting of a translation and scaling.

(ii) Find the fundamental matrix $\hat{\mathtt{F}}'$ corresponding to the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ by

(a) **Linear solution:** Determine $\hat{\mathtt{F}}$ from the singular vector corresponding to the smallest singular value of $\hat{\mathtt{A}}$, where $\hat{\mathtt{A}}$ is composed from the matches $\hat{\mathbf{x}}_i \leftrightarrow \hat{\mathbf{x}}'_i$ as defined in (11.3).

(b) **Constraint enforcement:** Replace $\hat{\mathtt{F}}$ by $\hat{\mathtt{F}}'$ such that $\det \hat{\mathtt{F}}' = 0$ using the SVD (see section 11.1.1).

(iii) **Denormalization:** Set $\mathtt{F} = \mathtt{T}'^{\mathsf{T}} \hat{\mathtt{F}}' \mathtt{T}$. Matrix F is the fundamental matrix corresponding to the original data $\mathbf{x}_i \leftrightarrow \mathbf{x}'_i$.

Algorithm 11.1. *The normalized 8-point algorithm for* F.

Computation of the Fundamental Matrix F

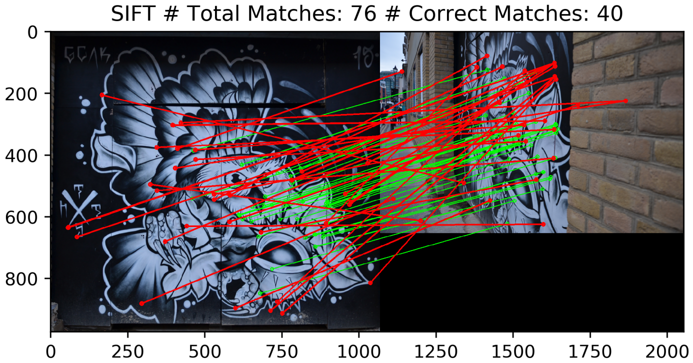
In theory

8 correspondences are enough for computing F

Practically

We rely on matching (lots of) interest points between images

Matching interest points



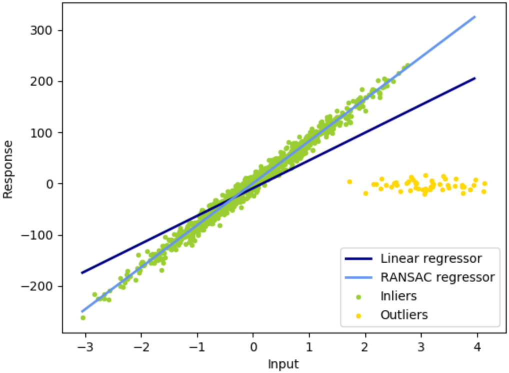
Robust estimation of good correspondences

Fischler and Bolles

Algorithm 1 RANSAC

1: Select randomly the minimum number of points required to determine the model parameters.
2: Solve for the parameters of the model.
3: Determine how many points from the set of all points fit with a predefined tolerance ϵ.
4: If the fraction of the number of inliers over the total number points in the set exceeds a predefined threshold τ, re-estimate the model parameters using all the identified inliers and terminate.
5: Otherwise, repeat steps 1 through 4 (maximum of N times).

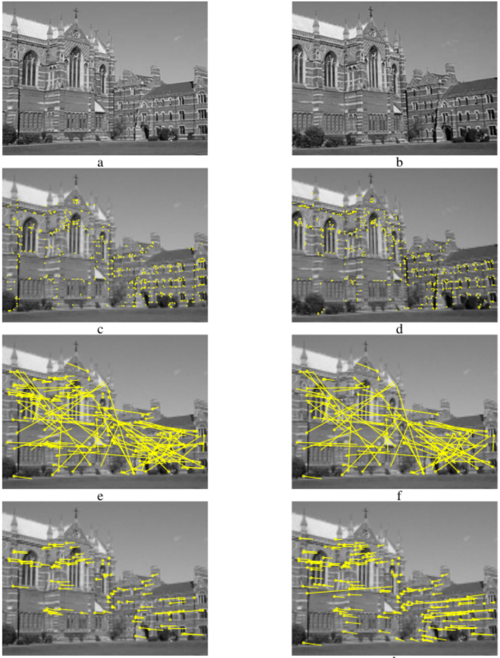
Robust estimation of good correspondences



`sklearn`

Estimated coefficients (true, linear regression, RANSAC):
82.1903908407869 [54.17236387] [82.08533159]

Robust estimation of good correspondences



Homography

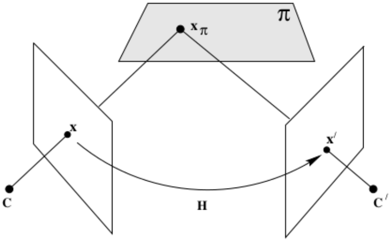


Fig. 13.1. **The homography induced by a plane.** *The ray corresponding to a point* $\mathbf{x}$ *is extended to meet the plane* $\boldsymbol{\pi}$ *in a point* $\mathbf{x}_\pi$*; this point is projected to a point* $\mathbf{x}'$ *in the other image. The map from* $\mathbf{x}$ *to* $\mathbf{x}'$ *is the homography induced by the plane* $\boldsymbol{\pi}$*. There is a perspectivity,* $\mathbf{x} = \mathtt{H}_{1\pi}\mathbf{x}_\pi$*, between the world plane* $\boldsymbol{\pi}$ *and the first image plane; and a perspectivity,* $\mathbf{x}' = \mathtt{H}_{2\pi}\mathbf{x}_\pi$*, between the world plane and second image plane. The composition of the two perspectivities is a homography,* $\mathbf{x}' = \mathtt{H}_{2\pi}\mathtt{H}_{1\pi}^{-1}\mathbf{x} = \mathtt{H}\mathbf{x}$*, between the image planes.*

Homography

F generic case

Each point in one image, is matched with a line in the other image

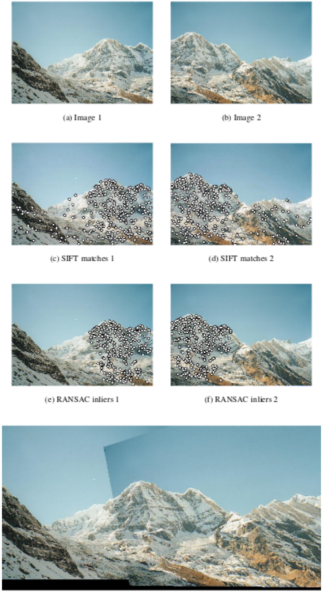
Homography special property

Each point in one image, is matched with a single point in the other image

Image Matching

Some examples & applications

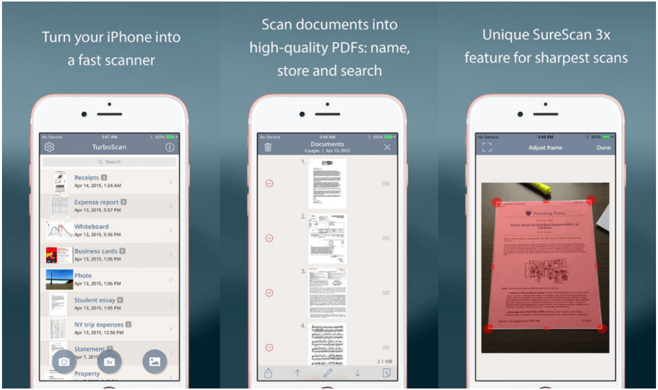
Panorama



[2]

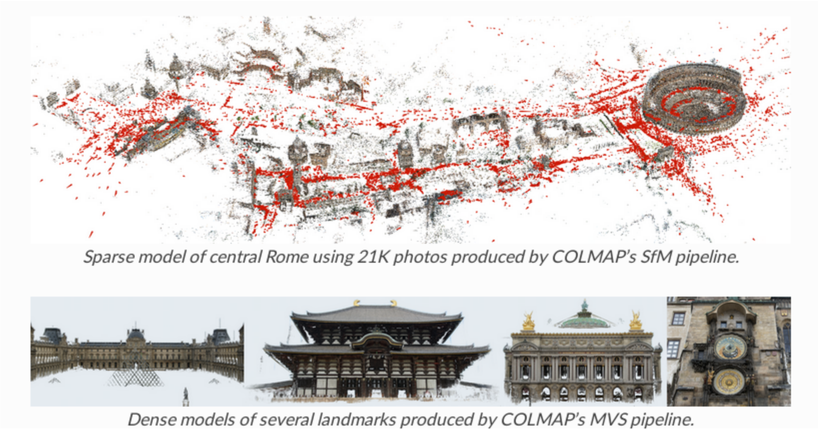
[2] Brown and Lowe, "Automatic panoramic image stitching using invariant

Image rectification



Turboscan App

3d Models



Colmap `https://colmap.github.io/`

Recap

Better ways to matching points between two images

$\downarrow$

Easier job for RANSAC

$\downarrow$

Better 3D models, panoramas, AR apps etc

Classical matching methods

Classical matching methods

The “classical” image matching pipeline

Image A   Image B

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

Terminology

Literature terms

Features, Keypoints, Local features, Interest points

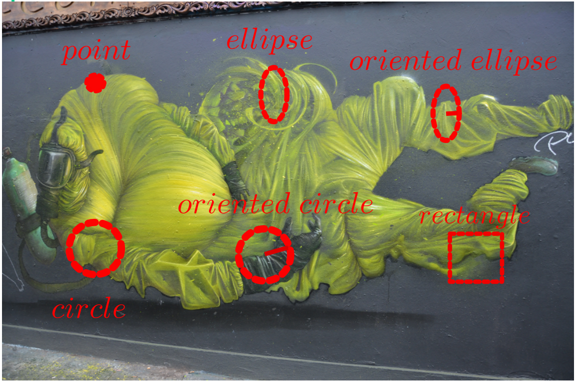
Our terminology

Feature frame[3]

a representation of a specific area/sub-region of an image, characterised by location and shape

[3] Vedaldi and Fulkerson, *VLFeat: An Open and Portable Library of Computer Vision Algorithms*.

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

Interest Points

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

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

Interest Points

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

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

Harris Corners

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

λ_1, λ_2: Eigenvalues of M

- $\lambda_1, \lambda_2 \approx 0$

Harris Corners

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

λ_1, λ_2: Eigenvalues of M

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

Harris Corners

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

λ_1, λ_2: Eigenvalues of M

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

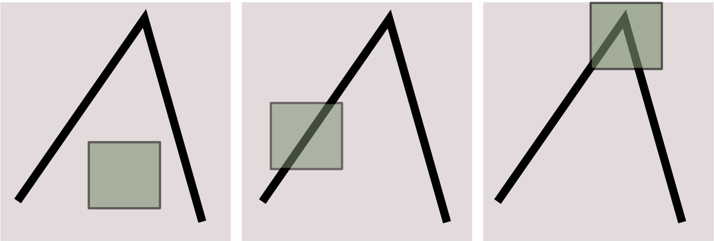
Harris Corners

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

λ_1, λ_2: Eigenvalues of M

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

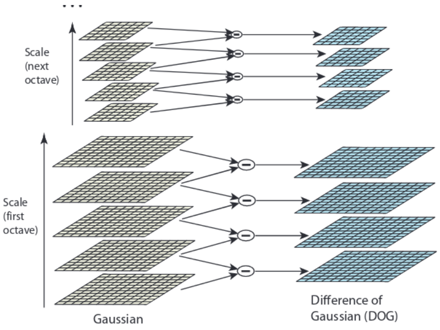
Harris Criterion

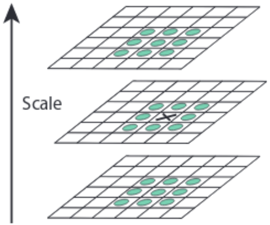


$\lambda_1 \approx \lambda_2 \approx 0$ $\lambda_1 \gg \lambda_2$ $\lambda_1 \approx \lambda_2 \gg 0$

Adding scale estimation

SIFT Detector





[4]

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

SURF

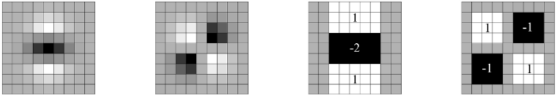


Fig. 1. Left to right: the (discretised and cropped) Gaussian second order partial derivatives in y-direction and xy-direction, and our approximations thereof using box filters. The grey regions are equal to zero. [5]

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

KAZE

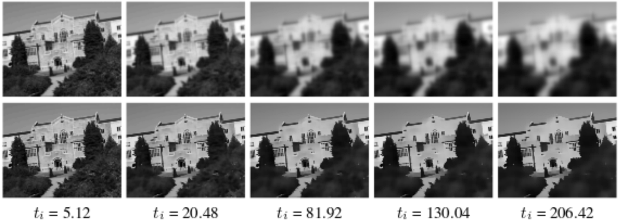


Fig. 2. Comparison between the Gaussian and nonlinear diffusion scale space for several evolution times t_i. First Row: Gaussian scale space (linear diffusion). The scale space is formed by convolving the original image with a Gaussian kernel of increasing standard deviation. Second Row: Nonlinear diffusion scale space with conductivity function g_3. [6]

[6] Alcantarilla, Bartoli, and Davison, "KAZE features".

Edge Foci

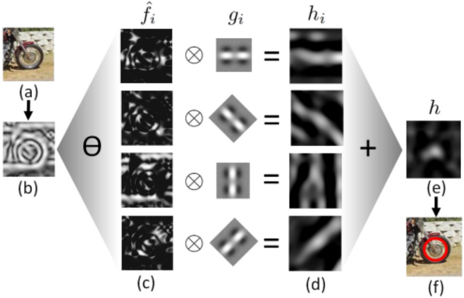
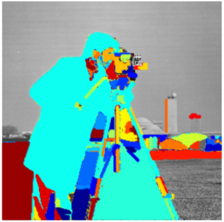


Figure 2. Flow diagram of the detector: (a) input image, (b) normalized gradient $\hat{f}$, (c) normalized gradients separated into orientations $\hat{f}_i$, (d) responses after applying oriented filter $h_i = \hat{f}_i \otimes g_i$, (e) the aggregated results h, and (f) detected interest point. [7]

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

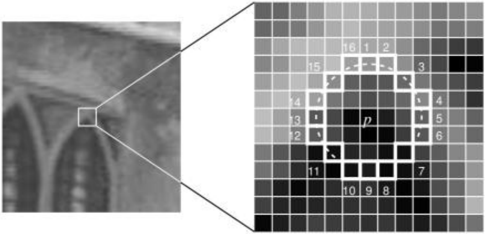
MSER



[8]

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

FAST



[9]

Rosten and Drummond, "Machine learning for high-speed corner detection".

Feature Frame Detectors - Recap

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

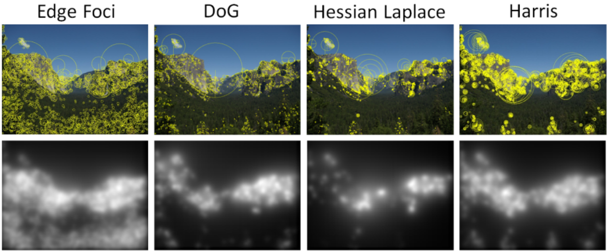
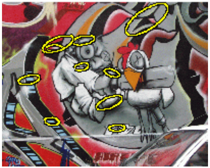
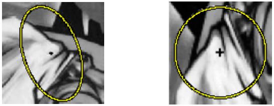


Figure 8. Visualization of the interest points and their spatial distributions for various detectors on Yosemite image.

From points to descriptors

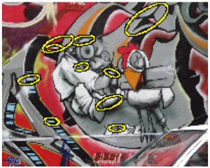


Detect Regions

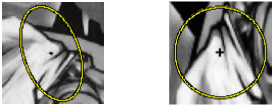


Rectify patch around feature frame

From points to descriptors



Detect Regions

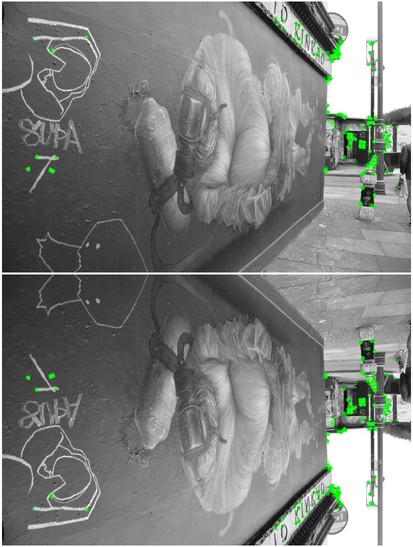


Rectify patch around feature frame

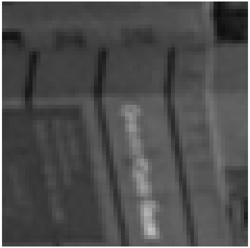
Local Descriptor

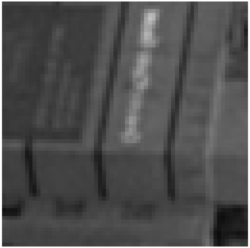
A *vectorial representation* of the patch around a feature frame which is more a discriminative and robust than the patch.

Importance of orientation



Importance of orientation





From points to descriptors

ZMUV descriptor

- Zeroed-mean-unit-variance patch ($ZMUV$) normalisation, which is defined as $\hat{\boldsymbol{p}} = \frac{mean(\boldsymbol{p})}{std(\boldsymbol{p})}$.
- not invariant to simple geometric deformations.
- In addition, the dimensionality of such a descriptor can be very high even for very small normalised patches e.g. it can reach 2^{10} for a 32×32 patch.

Descriptor definition

Given a patch $\boldsymbol{x} \in \mathbb{R}^{N \times N}$, a descriptor is the result $f_{\boldsymbol{x}} \in \mathbb{R}^{D}$ of a function f

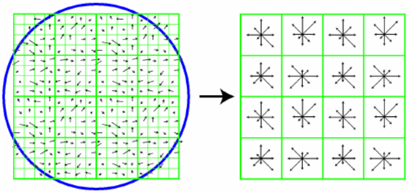
with $D < N \times N$ (ideally) and $f_{\boldsymbol{x}}$ more robust to geometric noise than the vector $\boldsymbol{x}$ (flattened list of pixel illuminations).

Descriptor Categorisation

- Output type
  - Floating point
  - Binary
- Hand-crafted vs. learning
  - Engineered / Hand Crafted Methods
  - Learning-based methods

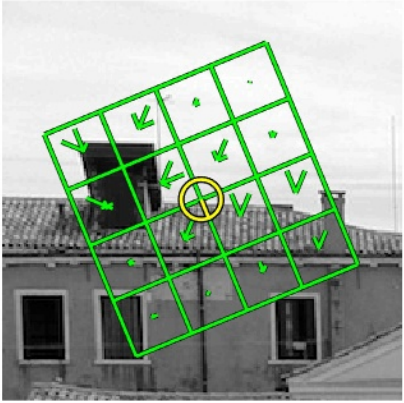
Hand-crafted floating point descriptors

SIFT Descriptor

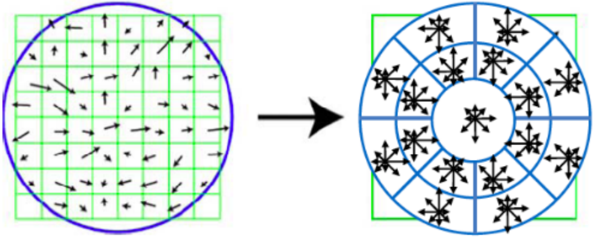


- The local spatial pooling of the descriptor is based on a rectangular grid that partitions the patch into several regions.
- Assuming the patch is divided into M rectangular areas, and the gradients are quantised to K angle bins, the resulting K dimensional histograms concatenated from M areas, will 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.

SIFT Descriptor



GLOH

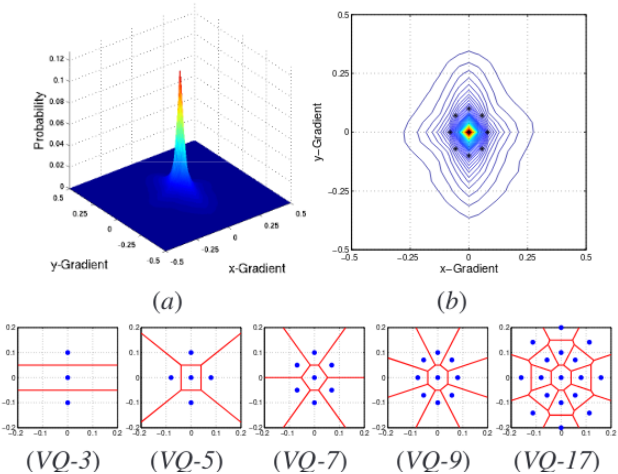


(a) image gradients

(b) keypoint descriptor [10]

[10] Mikolajczyk and Schmid, "A performance evaluation of local descriptors".

CHoG



[11]

[11] Chandrasekhar et al., "CHoG: Compressed histogram of gradients a low bit-rate feature descriptor".

Daisy

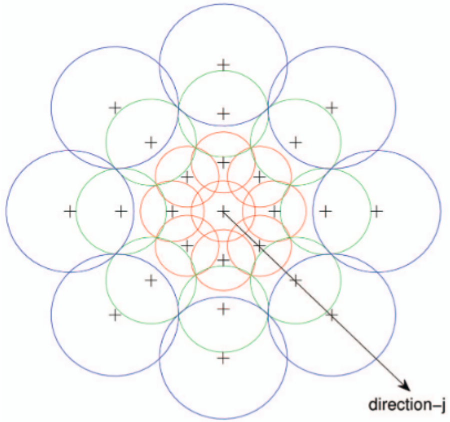
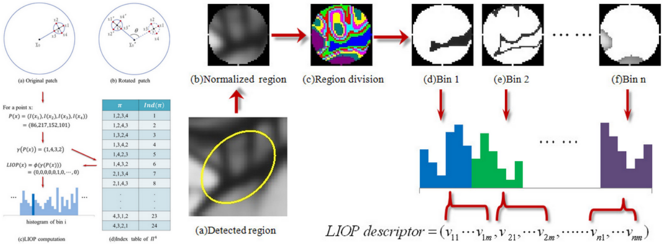


Fig. 6. The DAISY descriptor: Each circle represents a region where the radius is proportional to the standard deviations of the Gaussian kernels and the "+" sign represents the locations where we sample the convolved orientation maps center being a pixel location where we compute the descriptor. By 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. [12]

[12] Tola, Lepetit, and Fua, "Daisy: An efficient dense descriptor applied to wide-baseline stereo".

LIOP

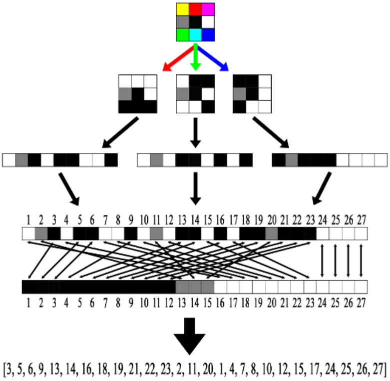


[13]

[13] Zhenhua Wang and Wu, "Local Intensity Order Pattern for Feature Description".

LUCID

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



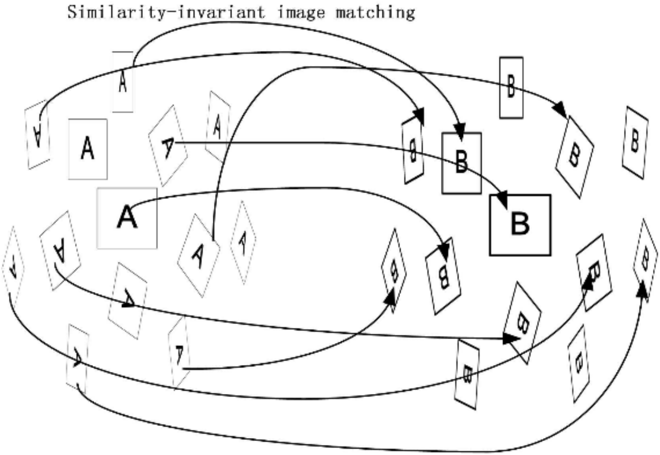
[14]

[14] Ziegler et al., "Locally uniform comparison image descriptor".

Aggregation across scales and viewpoints

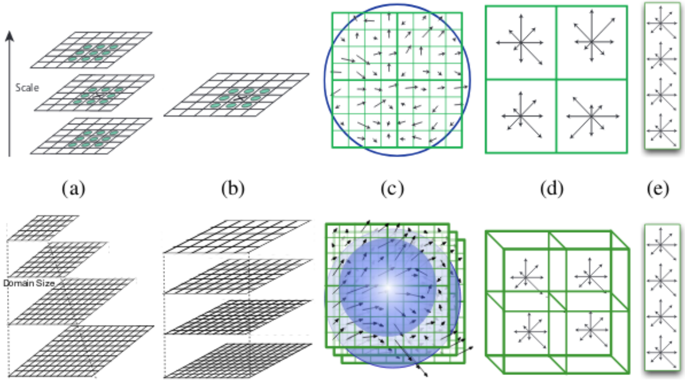
Several methods identified that aggregation across different scales or different affine viewpoints into a single feature vector can improve the discriminative power of the descriptor, albeit at the price of much higher computational cost

ASIFT



[15] Yu and Morel, "ASIFT: An algorithm for fully affine invariant comparison".

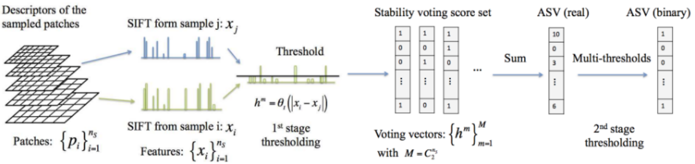
DSP-SIFT



16

Dong and Soatto, "Domain-size pooling in local descriptors: DSP-SIFT".[16]

ASV



17

[17] Yang, Lin, and Chuang, "Accumulated Stability Voting: A Robust Descriptor From Descriptors of Multiple Scales".

Hand-crafted binary descriptors

Hashing SIFT

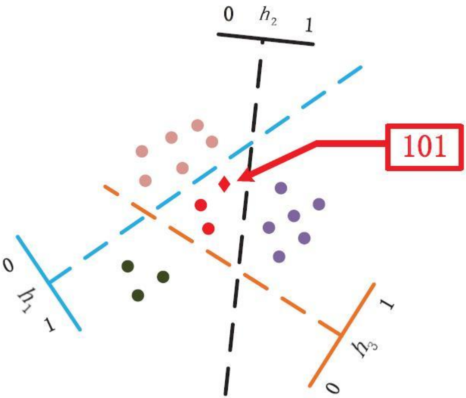


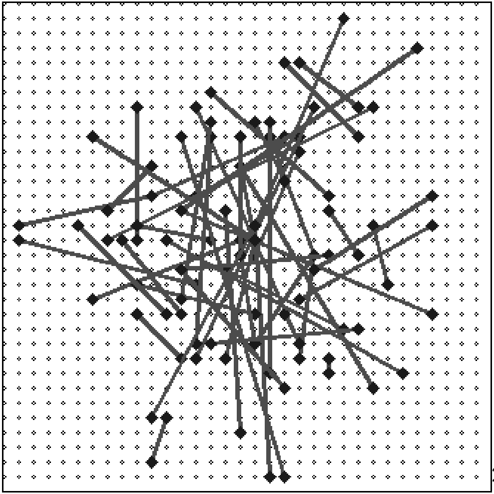
Image from Haisheng Li.

[18][19]

[18] Terasawa and Tanaka, "Spherical lsh for approximate nearest neighbor search on unit hypersphere".

[19] Strecha et al., "LDAHash: Improved matching with smaller descriptors".

BRIEF

[20]

[20] Calonder et al., "Brief: Binary robust independent elementary features".

BRISK

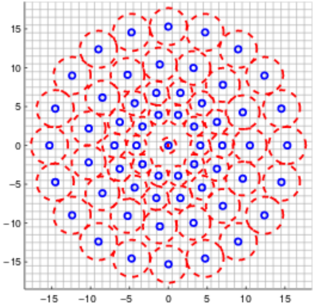


Figure 3. The BRISK sampling pattern with $N = 60$ points: the small blue circles denote the sampling locations; the bigger, red dashed circles are drawn at a radius σ corresponding to the standard deviation of the Gaussian kernel used to smooth the intensity values at the sampling points. The pattern shown applies to a scale of $t = 1$. [21]

[21] Leutenegger, Chli, and Siegwart, "BRISK: Binary robust invariant scalable keypoints".

FREAK

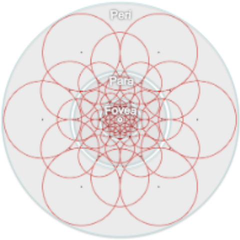


Figure 4: Illustration of the FREAK sampling pattern similar to the retinal ganglion cells distribution with their corresponding receptive fields. Each circle represents a receptive field where the image is smoothed with its corresponding Gaussian kernel. [22]

[22] Alahi, Ortiz, and Vandergheynst, "Freak: Fast retina keypoint".

Learning-based floating point descriptors

PCA-SIFT

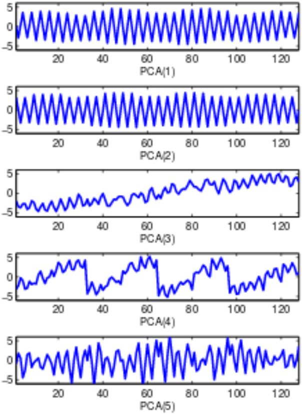
Collect a matrix $X \in \mathbb{R}^{N \times D}$ with N descriptors of dimensionality D

$$C = X^T X$$
$$C = U \Sigma V$$

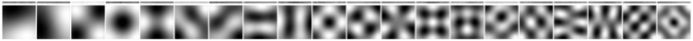
Use the first K eigenvectors from U to project X to a new descriptor of size K. $X_k = U_k X$[23]

[23] Ke and Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors".

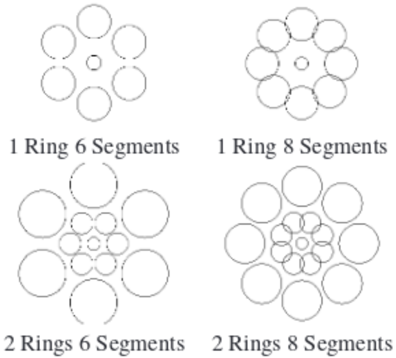
PCA-SIFT



PCA

PCA 

Picking the best Daisy



24

24 Calonder et al., "Brief: Binary robust independent elementary features".

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.

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

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

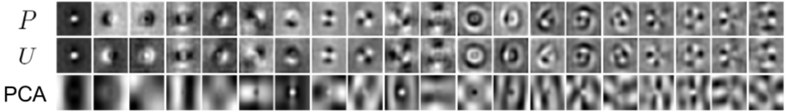
Note that these are not the same matrices as the between-class S_B and within-class scatters S_W in equation (1) for LDA, although they are related (see section 3.3) . The solution is the generalized eigenvectors:

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

The projection matrix is $U \in \mathbb{R}^{m \times m'}$, with $m' \leq m$ eigenvectors corresponding to the m' largest eigenvalues. [25]

[25] Cai, Mikolajczyk, and Matas, "Learning linear discriminant projections for dimensionality reduction of image descriptors".

Linear projections



[26]

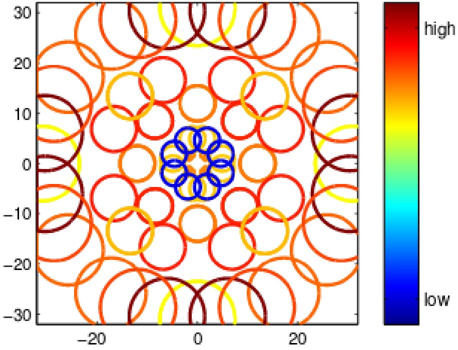
Cai, Mikolajczyk, and Matas, "Learning linear discriminant projections for dimensionality reduction of image descriptors".

Convex optimisation for learning descriptors

Learn optimal configuration of gaussian filters s.t.

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

.

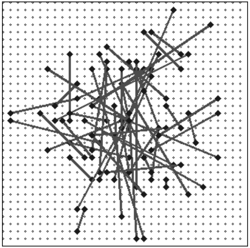


27

[27] Simonyan, Vedaldi, and Zisserman, "Learning Local Feature Descriptors Using Convex Optimisation."

Learning-based binary descriptors

ORB



- Instead of random intensity tests (as in BRIEF), select tests based on data
- Choose tests with maximum variance across different samples & minimum correlation between them.
- No need for pairs of labelled positive and negative patches [28]

[28] Rublee et al., "ORB: An efficient alternative to SIFT or SURF".

ORB

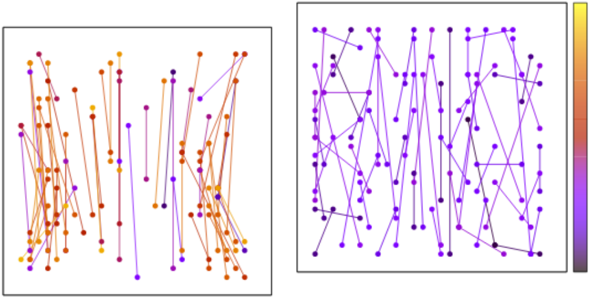
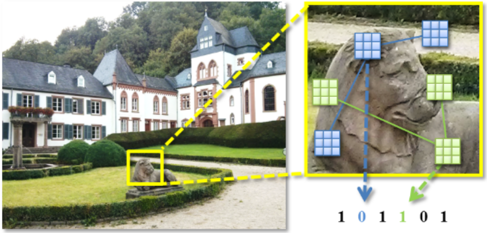


Figure 6. A subset of the binary tests generated by considering high-variance under orientation (left) and by running the learning algorithm to reduce correlation (right). Note the distribution of the tests around the axis of the keypoint orientation, which is pointing up. The color coding shows the maximum pairwise correlation of each test, with black and purple being the lowest. The learned tests clearly have a better distribution and lower correlation.

LATCH

Triplets of comparisons instead of pairs



29

[29] Levi and Hassner, "LATCH: learned arrangements of three patch codes".

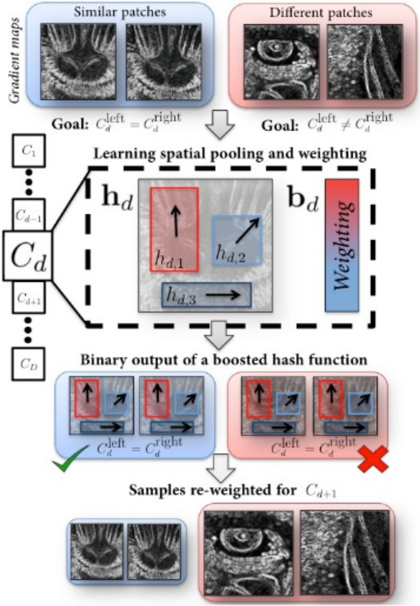
DBRIEF

$$\forall_{i \in 1,\dots,N} \quad b_i = \mathrm{sign}(\mathbf{w}_i^\top \mathbf{x} + \tau_i)$$

Learning of w_i and t_i
[30]

[30] Trzcinski and Lepetit, "Efficient Discriminative Projections for Compact Binary Descriptors".

Boosting



31

[31] Trzcinski et al., "Boosting Binary Keypoint Descriptors".

Boosting

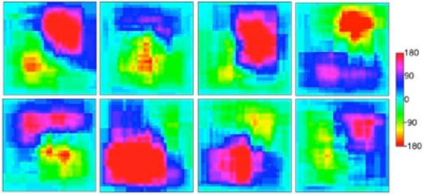
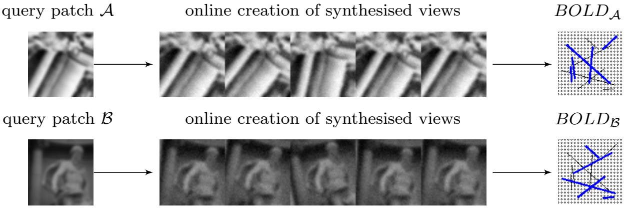


Figure 4. Visualization of the selected weak learners for the first 8 bits learned on 200k pairs of 32×32 patches from the Notre Dame dataset (best viewed on screen). For each pixel of the figure we show the average orientation weighted by the weights of the weak learners $\mathbf{b}_d$. For different bits, the weak learners cluster about different regions and orientations illustrating their complementary nature.

BOLD

Overview of the proposed BOLD descriptor



[32]

---

[32] Balntas, Tang, and Mikolajczyk, "BOLD - Binary Online Learned Descriptor For Efficient Image Matching".

BOLD

Typical descriptors

Patch		Descriptor
$\mathcal{A}$		$\mathcal{D}_{\mathcal{A}}=[d_{\mathcal{A}1}, d_{\mathcal{A}2} \dots d_{\mathcal{A}D}]$
$\mathcal{B}$		$\mathcal{D}_{\mathcal{B}}=[d_{\mathcal{B}1}, d_{\mathcal{B}2} \dots d_{\mathcal{B}D}]$

Distance

$\Delta(\mathcal{D}_{\mathcal{A}}, \mathcal{D}_{\mathcal{B}})$

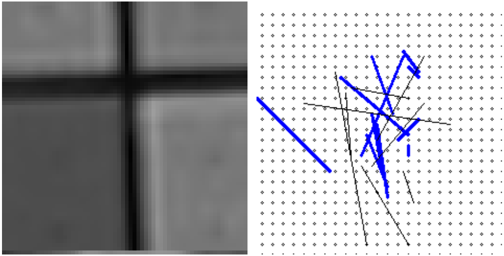
Locally learned BOLD

Patch		Descriptor	Stable descriptor dimensions
$\mathcal{A}$		$\mathcal{D}_{\mathcal{A}}=[d_{\mathcal{A}1}, d_{\mathcal{A}2} \dots d_{\mathcal{A}D}]$	$\mathcal{M}_{\mathcal{A}} = [0, 1 \dots 1]$
$\mathcal{B}$		$\mathcal{D}_{\mathcal{B}}=[d_{\mathcal{B}1}, d_{\mathcal{B}2} \dots d_{\mathcal{B}D}]$	$\mathcal{M}_{\mathcal{B}} = [1, 0 \dots 0]$

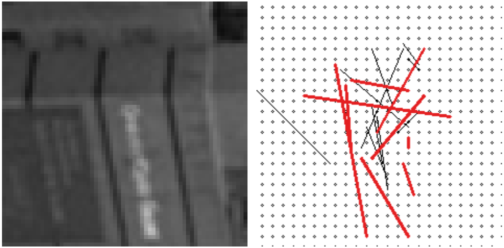
Distance

$\Delta(\mathcal{D}_{\mathcal{A}}, \mathcal{M}_{\mathcal{A}}, \mathcal{D}_{\mathcal{B}}, \mathcal{M}_{\mathcal{B}})$

BOLD



BOLD



Deep Learning Era

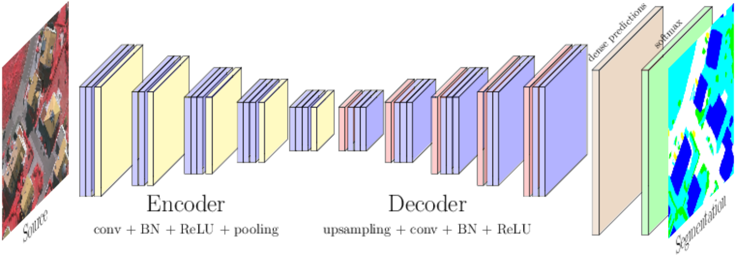
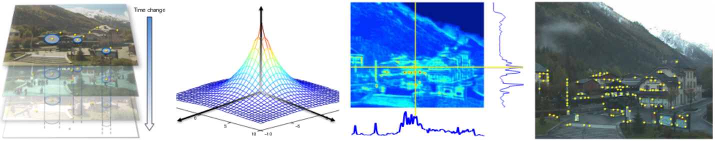


Image: Nicolas Audebert

Deep learning: detectors

TILDE



(a) Stack of training images
(b) Desired response on positive samples
(c) Regressor response for a new image
(d) Keypoints detected in the new image [33]

[33] Verdie et al., "TILDE: A Temporally Invariant Learned DEtector".

Learning a detector by ranking

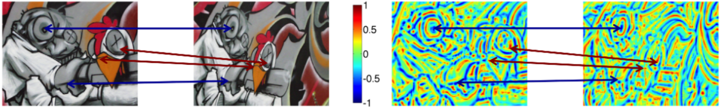


Figure 1. Left: an image undergoes a perspective change transformation. Right: our learned response function, visualized as a heat map, produces a ranking of image locations that is reasonably invariant under the transformation. Since the resulting ranking is largely repeatable, the top/bottom quantiles of the response function are also repeatable (examples of interest points are shown by arrows). [34]

[34] Savinov et al., *Quad-networks: unsupervised learning to rank for interest point detection.*

Learning a detector by ranking

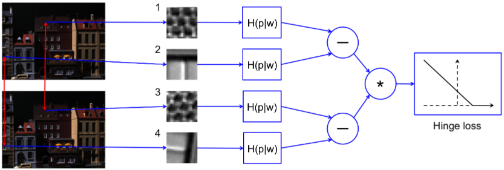


Figure 2. Quad-network forward pass on a training quadruple. Patches $(1, 3)$ and $(2, 4)$ are correspondence pairs between two different images, so $1, 2$ come from the first image and $3, 4$ come from the second image. All of the patches are extracted with a random rotation. [35]

[35] Savinov et al., *Quad-networks: unsupervised learning to rank for interest point detection.*

Learning covariant detectors

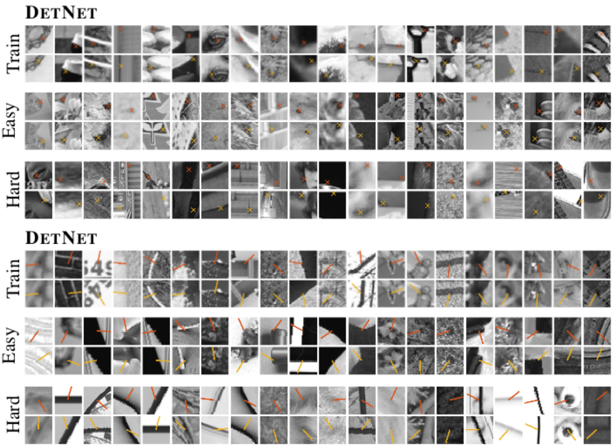
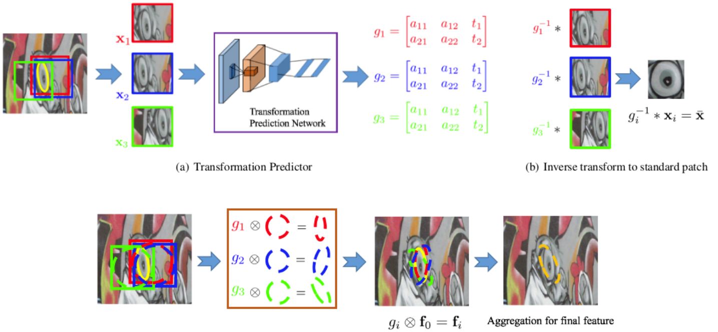


Fig. 4: *Training and validation patches.* Example of training triplets $(\mathbf{x}_1, \mathbf{x}_2, g)$ ($\mathbf{x}_1$ above and $\mathbf{x}_2 = g\mathbf{x}_1$ below) for different detectors. The figure also shows "easy" and "hard" patch pairs, extracted from the validation set based on the value of the loss (16). The crosses and bars represent respectively the detected translation and orientation, as learned by DETNET-L and ROTNET-L. [36]

[36] Lenc and Vedaldi, *Learning Covariant Feature Detectors*

Learning Discriminative and Transformation Covariant Local Feature Detectors

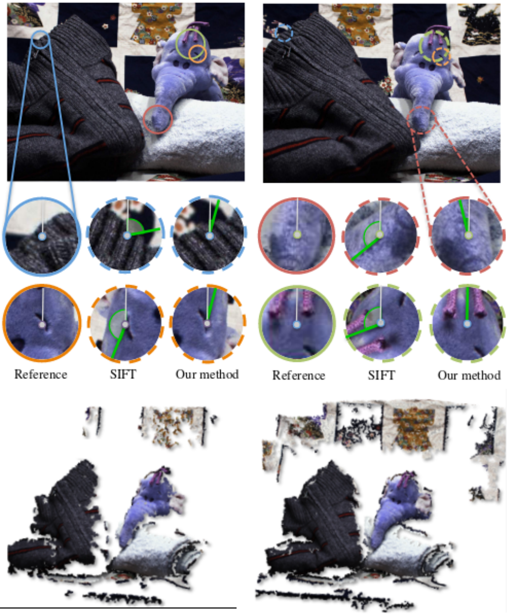


37

---

[37] Lenc and Vedaldi, *Learning Covariant Feature Detectors*.

Learning to assign orientations

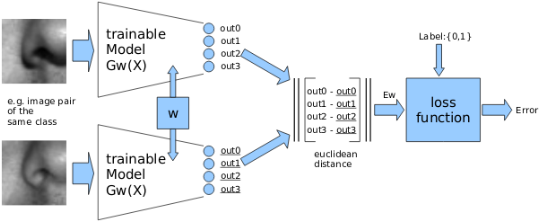


[38] Yi et al., "Learning to Assign Orientations to Feature Points".

Deep learning: descriptors

2008 work

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



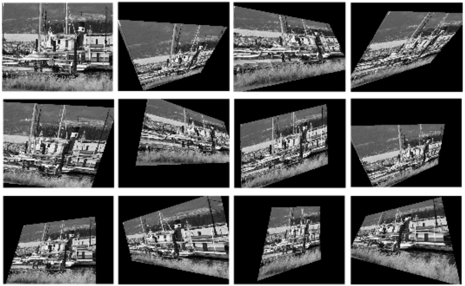
39

---

[39] Jahrer, Grabner, and Bischof, "Learned local descriptors for recognition and matching".

2008 work

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

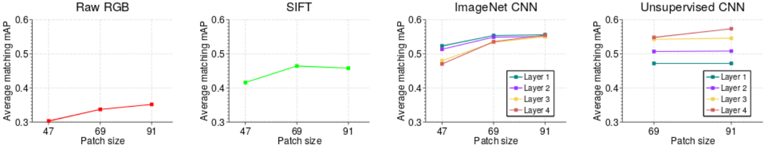


[40]

[40] Jahrer, Grabner, and Bischof, "Learned local descriptors for recognition and matching".

2014 work

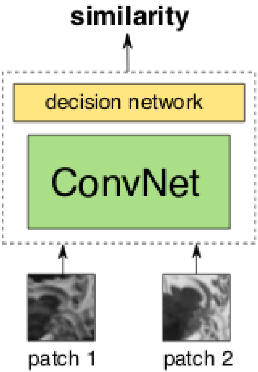
Results shown in[41] that the features from the last layer of a convolutional deep network trained on ImageNet dataset collected for general objects classification can outperform SIFT.



Such features outperform the performance of descriptors resulting from convex optimisation.

[41] Fischer, Dosovitskiy, and Brox, "Descriptor Matching with Convolutional Neural Networks: a Comparison to SIFT".

DeepCompare



$$\min_w \frac{\lambda}{2}\|w\|_2 + \sum_{i=1}^{N} \max(0, 1 - y_i o_i^{net})$$ [42]

[42] Zagoruyko and Komodakis, "Learning to Compare Image Patches via Convolutional Neural Networks".

DeepCompare

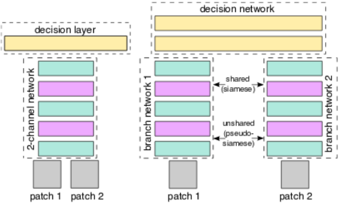


Figure 2. Three basic network architectures: 2-channel on the left, siamese and pseudo-siamese on the right (the difference between siamese and pseudo-siamese is that the latter does not have shared branches). Color code used: cyan = Conv+ReLU, purple = max pooling, yellow = fully connected layer (ReLU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

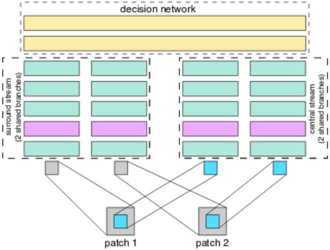


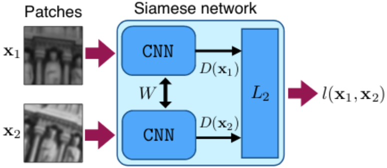
Figure 3. A central-surround two-stream network that uses a siamese-type architecture to process each stream. This results in 4 branches in total that are given as input to the top decision layer (the two branches in each stream are shared in this case).

[43]

---

[43] Zagoruyko and Komodakis, "Learning to Compare Image Patches via Convolutional Neural Networks".

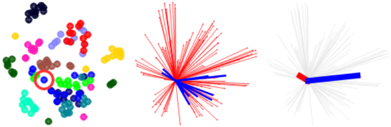
DeepDesc



$$l(\mathbf{x}_1, \mathbf{x}_2) = \begin{cases} \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2, & p_1 = p_2 \\ \max(0, C - \|D(\mathbf{x}_1) - D(\mathbf{x}_2)\|_2), & p_1 \neq p_2 \end{cases}$$ [44]

[44] Simo-Serra et al., "Discriminative Learning of Deep Convolutional Feature Point Descriptors".

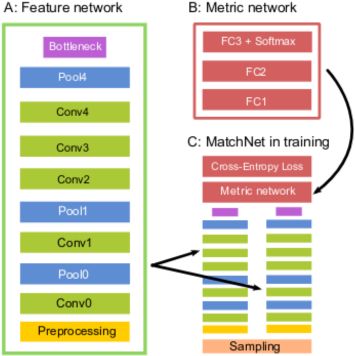
DeepDesc



(a) Data (b) All pairs (c) "Hard" pairs [45]

[45] Simo-Serra et al., "Discriminative Learning of Deep Convolutional Feature Point Descriptors".

MatchNet

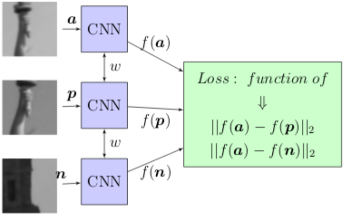


Name	Type	Output Dim.	PS	S
Conv0	C	$64 \times 64 \times 24$	7×7	1
Pool0	MP	$32 \times 32 \times 24$	3×3	2
Conv1	C	$32 \times 32 \times 64$	5×5	1
Pool1	MP	$16 \times 16 \times 64$	3×3	2
Conv2	C	$16 \times 16 \times 96$	3×3	1
Conv3	C	$16 \times 16 \times 96$	3×3	1
Conv4	C	$16 \times 16 \times 64$	3×3	1
Pool4	MP	$8 \times 8 \times 64$	3×3	2
Bottleneck	FC	B	-	-
FC1	FC	F	-	-
FC2	FC	F	-	-
FC3	FC	2	-	-

[46]

[46] Han et al., "MatchNet: Unifying Feature and Metric Learning for Patch-Based Matching".

TFeat



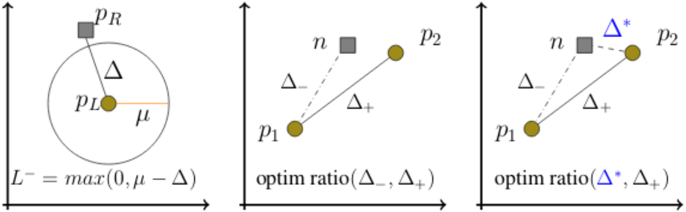
$$\sum_{i=1}^{N} l_{rank}(\delta_+, \delta_-) + \lambda \cdot ||\boldsymbol{w}||_2^2$$

where

$$l_{rank}(\delta_+, \delta_-) = max(0, \mu + \delta_+ - \delta_-)$$ [47]

[47] Vassileios Balntas and Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks".

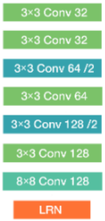
TFeat



[48]

[48] Vassileios Balntas and Mikolajczyk, "Learning local feature descriptors with triplets and shallow convolutional neural networks".

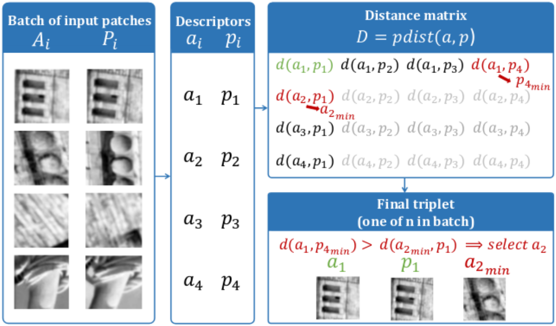
L2-Net



- Distance matrix loss: $\sqrt{2(1 - Y_1^T Y_2)}$
- De-corellation loss: $Y_1^T Y_1$ [49]

[49] Tian, Fan, and Wu, "L2-Net: Deep Learning of Discriminative Patch Descriptor in Euclidean Space".

HardNet



50

[50] Mishchuk et al., "Working hard to know your neighbor's margins: Local descriptor learning loss".

Spread out descriptor

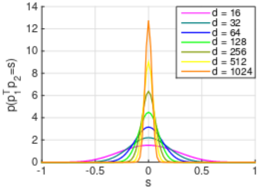
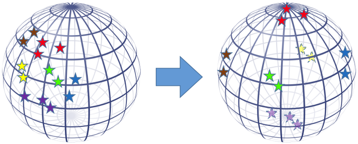


Figure 2. Probability density of inner product of two points which are independently and uniformly sampled from the unit sphere in d-dimensional space. We can see that, in high dimensional space, most pairs are close to orthogonal.



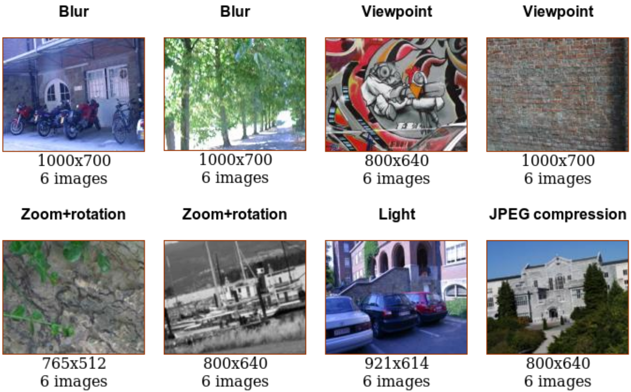
Not Spread-Out (Bad) Spread-Out (Good) [51]

[51] Zhang et al., "Learning Spread-out Local Feature Descriptors".

Datasets & Benchmarks

Oxford Matching Benchmark

- Measures descriptor performance in image matching task
- NN matching



Oxford Matching Protocol

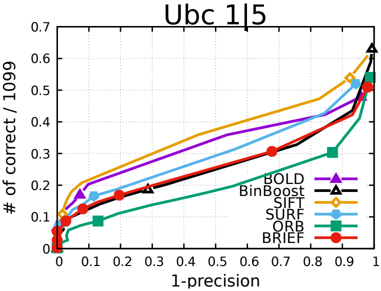
Two local frames A and B are matched if $||D_A - D_B||_2^2 < \tau$

- $recall = \dfrac{\#correct\ matches}{\#correspondences}$
- $1\text{-}precision = \dfrac{\#false\ matches}{\#correct\ matches + \#false\ matches}$

Performance curves

$$||D_A - D_B||_2^2 < \tau$$

Varying τ leads to performance curves



Inconsistency in evaluation results - Oxford Benchmark

LIOP outperforms SIFT	SIFT outperforms LIOP
[Miksik and Mikolajczyk, 2012]	[Tsun-Yi Yang and Chuang, 2016]
[Wang et al., 2011b]	

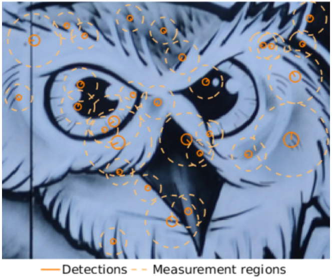
BRISK outperforms SIFT	SIFT outperforms BRISK
Leutenegger et al. [2011]	[Levi and Hassner, 2016]
Miksik and Mikolajczyk [2012]	

ORB outperforms SIFT	SIFT outperforms ORB
Rublee et al. [2011]	Miksik and Mikolajczyk [2012]

BinBoost outperforms SIFT	SIFT outperforms BinBoost
[Levi and Hassner, 2016]	[Balntas et al., 2015]
[T. Trzcinski and Lepetit, 2013]	[Tsun-Yi Yang and Chuang, 2016]

ORB outperforms BRIEF	BRIEF outperforms ORB
[Rublee et al., 2011]	[Levi and Hassner, 2016]

Inconsistency in evaluation results - Oxford Benchmark



- no strict protocol for patch extraction and normalisation
- no strict protocol for detector configuration
- no standardised measurement region

Inconsistency in evaluation results - Oxford Benchmark

mAP: mean area under performance curves

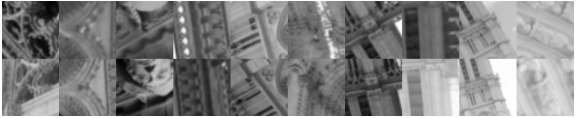
descr	1|2	1|3	1|4
SIFT `vl_sift`	0.47	0.40	**0.46**
SIFT `vl_covdet`	0.32	0.14	0.18

method	**paper**
`vl_sift`	***ASV*** [CVPR 2016], ***DSP-SIFT*** [CVPR 2015]
`vl_covdet`	***BinBoost*** [PAMI 2015], ***BOLD*** [CVPR 2015]

From images to patches

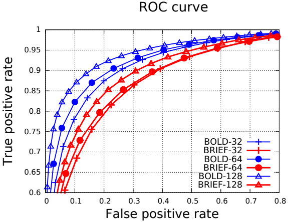
Phototourism Patch Datasets

Pre-extracted patches arranged in matching and non-matching pairs



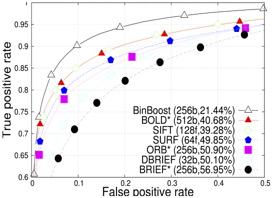
Phototourism Patch Datasets - Evaluation

Pair	Label	Distance
	Pos	0.3
	Pos	0.5
	Pos	0.8
	Neg	0.7
	Neg	1.7

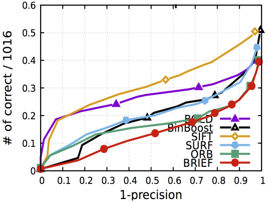


Phototourism Patch Datasets - Evaluation Issues

ROC from positive and negative pairs of patches

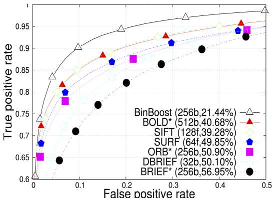


PR Curves from NN matching

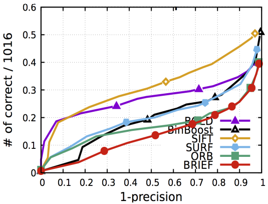


Phototourism Patch Datasets - Evaluation Issues

ROC from positive and negative pairs of patches



PR Curves from NN matching



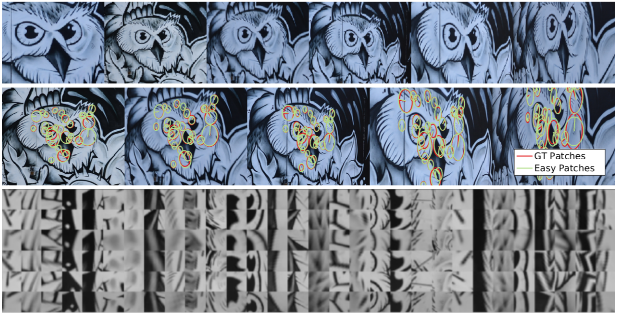
- Patch verification (yes/no) different problem than matching (match all with all)
- No single task should be used for evaluating a method

HPatches Dataset

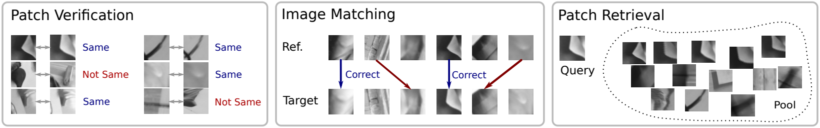
[52]

[52] Balntas et al., "HPatches: A Benchmark and Evaluation of Handcrafted and Learned Local Descriptors".

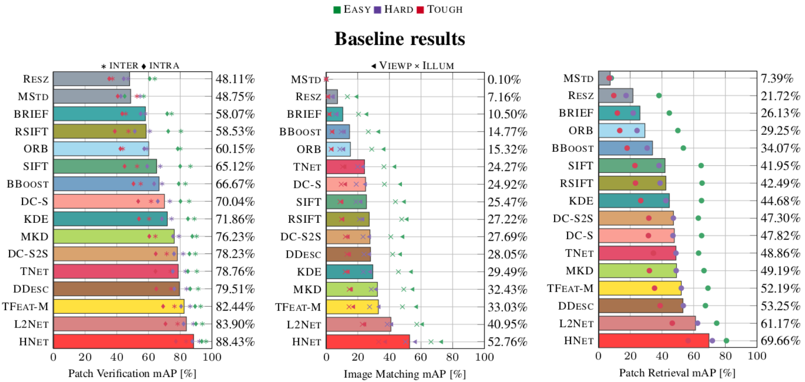
HPatches Dataset



HPatches tasks



HPatches results



SfM Benchmark

Sparse model of central Rome using 21K photos produced by COLMAP's SfM pipeline.

Dense models of several landmarks produced by COLMAP's MVS pipeline. [53]

[53] Schönberger et al., "Comparative Evaluation of Hand-Crafted and Learned Local Features".

SfM Benchmark

		# Images	# Registered	# Sparse Points	# Observations	Track Length	Reproj. Error	# Inlier Pairs	# Inlier Matches	# Dense Points
Fountain	*SIFT*	11	11	10,004	44K	4.49	**0.30px**	49	76K	2,970K
	SIFT-PCA		11	**14,608**	**70K**	**4.80**	0.39px	**55**	**124K**	**3,021K**
	DSP-SIFT		11	**14,785**	**71K**	**4.80**	0.41px	54	**129K**	2,999K
	ConvOpt		11	**14,179**	**67K**	**4.75**	0.37px	**55**	**114K**	**2,999K**
	DeepDesc		11	13,519	61K	4.55	**0.35px**	**55**	93K	2,972K
	TFeat		11	13,696	64K	4.68	**0.35px**	54	103K	2,969K
	LIFT		11	10,172	46K	4.55	0.59px	**55**	83K	**3,019K**
Herzjesu	*SIFT*	8	8	4,916	19K	4.00	**0.32px**	27	28K	2,373K
	SIFT-PCA		8	**7,433**	**31K**	**4.19**	0.42px	**28**	**47K**	2,372K
	DSP-SIFT		8	**7,760**	**32K**	**4.19**	0.45px	**28**	**50K**	**2,376K**
	ConvOpt		8	6,939	28K	**4.13**	0.40px	**28**	42K	2,375K
	DeepDesc		8	6,418	25K	3.92	**0.38px**	**28**	34K	**2,380K**
	TFeat		8	6,606	27K	4.09	**0.38px**	**28**	38K	**2,377K**
	LIFT		8	**7,834**	**30K**	3.95	0.63px	**28**	**46K**	2,375K
South Building	*SIFT*	128	128	62,780	353K	5.64	**0.42px**	1K	1,003K	1,972K
	SIFT-PCA		128	**107,674**	**650K**	**6.04**	0.54px	**3K**	**2,019K**	1,993K
	DSP-SIFT		128	**110,394**	**664K**	**6.02**	0.57px	3K	**2,079K**	**1,994K**
	ConvOpt		128	**103,602**	**617K**	5.96	0.51px	**4K**	**1,856K**	**2,007K**
	DeepDesc		128	101,154	558K	5.53	**0.48px**	**6K**	1,463K	**2,002K**
	TFeat		128	94,589	566K	**5.99**	**0.49px**	3K	1,567K	1,960K
	LIFT		128	74,607	399K	5.35	0.78px	3K	1,168K	1,975K
Madrid Metropolis	*SIFT*	1,344	440	62,729	416K	**6.64**	**0.53px**	14K	1,740K	435K
	SIFT-PCA		**465**	**119,244**	**702K**	5.89	0.57px	**27K**	**3,597K**	**537K**
	DSP-SIFT		**476**	**107,028**	**681K**	**6.36**	0.64px	**21K**	**3,155K**	**570K**
	ConvOpt		**455**	**115,134**	**634K**	5.51	0.57px	**29K**	**3,148K**	**561K**
	DeepDesc		377	68,110	348K	5.11	**0.53px**	19K	1,570K	516K
	TFeat		439	90,274	512K	5.68	**0.54px**	18K	2,135K	522K
	LIFT		430	52,755	337K	**6.40**	0.76px	13K	1,498K	450K
Gendarmenmarkt	*SIFT*	1,463	950	169,900	1,010K	**5.95**	**0.64px**	28K	3,292K	1,104K
	SIFT-PCA		**953**	272,118	**1,477K**	**5.43**	0.69px	**43K**	**5,137K**	1,240K
	DSP-SIFT		**975**	**321,846**	**1,732K**	**5.38**	0.74px	**56K**	**7,648K**	**1,505K**
	ConvOpt		945	**341,591**	**1,601K**	4.69	0.70px	**56K**	**6,525K**	**1,342K**
	DeepDesc		809	244,925	949K	3.88	**0.68px**	31K	2,849K	921K
	TFeat		**953**	**297,266**	1,445K	4.86	**0.66px**	39K	4,685K	1,181K
	LIFT		942	180,746	964K	5.34	0.83px	27K	2,495K	**1,386K**

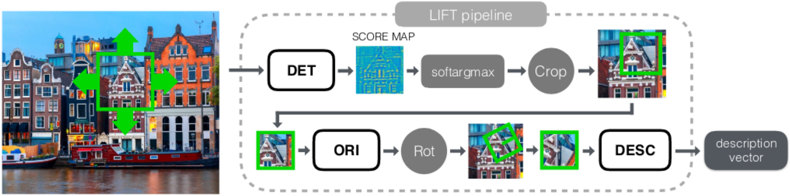
[54]

[54] Schönberger et al., "Comparative Evaluation of Hand-Crafted and Learned Local Features".

Current trends & future challenges

Matching without local features

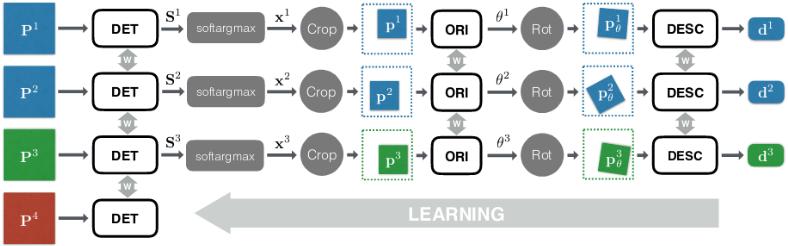
LIFT



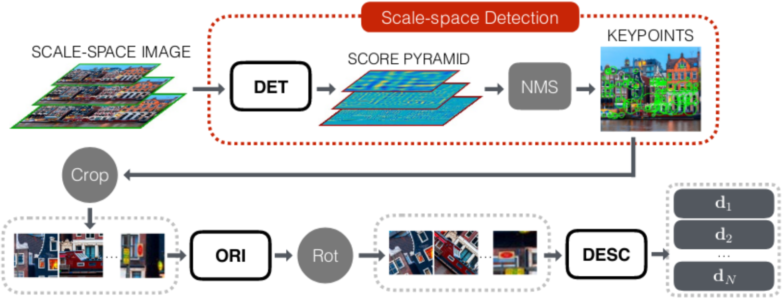
[55]

[55] Yi et al., "LIFT: Learned Invariant Feature Transform".

LIFT

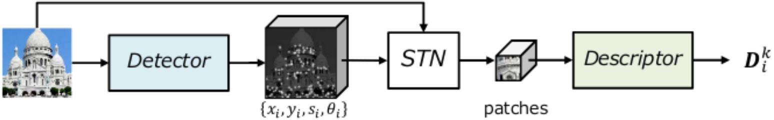

P^1
P^2
P^3
P^4
DET
DET
DET
DET
W
W
W
S^1
S^2
S^3
softargmax
softargmax
softargmax
x^1
x^2
x^3
Crop
Crop
Crop
p^1
p^2
p^3
ORI
ORI
ORI
θ^1
θ^2
θ^3
Rot
Rot
Rot
p_θ^1
p_θ^2
p_θ^3
DESC
DESC
DESC
d^1
d^2
d^3
LEARNING

LIFT

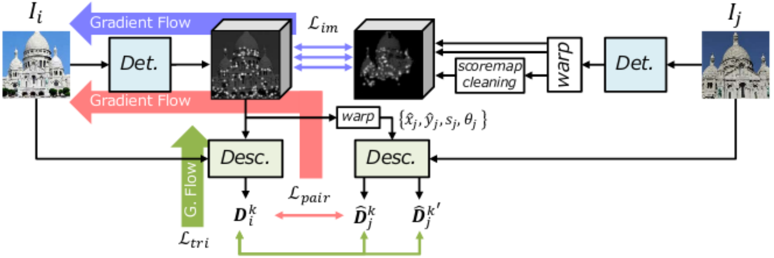


Scale-space Detection
SCALE-SPACE IMAGE
DET
SCORE PYRAMID
NMS
KEYPOINTS
Crop
ORI
Rot
DESC
d_1
d_2
...
d_N

LF-Net

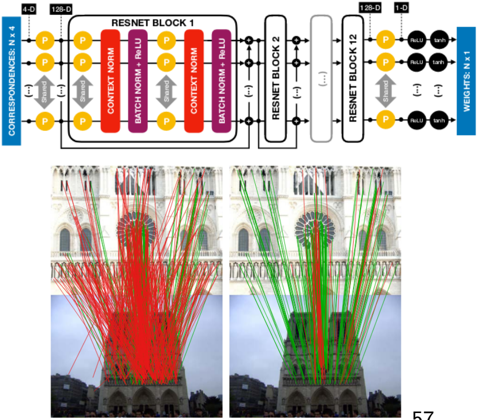


(a) The LF-Net architecture. The *detector* network generates a scale-space score map along with dense orientation estimates, which are used to select the keypoints. Image patches around the chosen keypoints are cropped with a differentiable sampler (STN) and fed to the *descriptor* network, which generates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



[56] Ono et al., "LF-Net: Learning Local Features from Images".

Learning correspondences

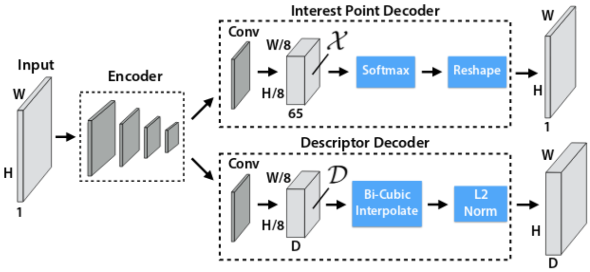


(a) RANSAC (b) Our approach [57]

---

[57] Yi et al., "Learning to Find Good Correspondences".

Superpoint



[58]

[58] DeTone, Malisiewicz, and Rabinovich, "SuperPoint: Self-Supervised Interest Point Detection and Description".

Implicitly Matched Interest Points (IMIPs)

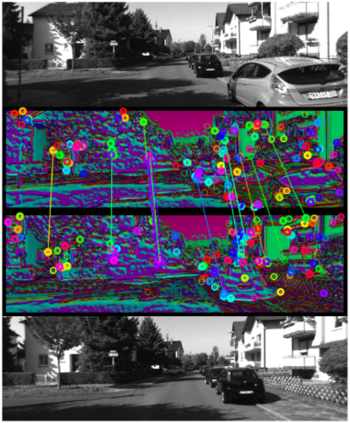
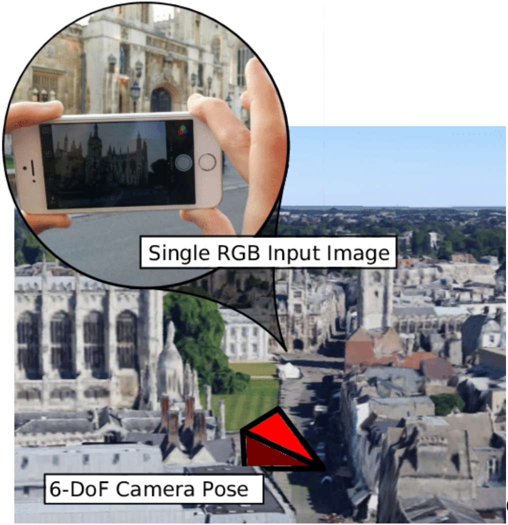


Figure 1. We propose a CNN interest point detector which provides implicitly matched interest points — descriptors are not needed for matching. This image illustrates the output of the final layer, which determines the interest points. Hue indicates which channel has the strongest response for a given 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. [59]

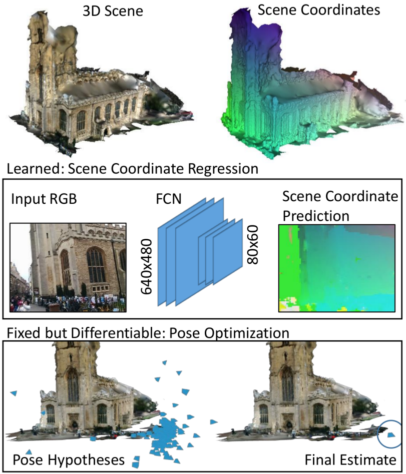
[59] Cieslewski, Bloesch, and Scaramuzza, "Matching Features without Descriptors: Implicitly Matched Interest Points (IMIPs)".



[60]

[60] Kendall, Grimes, and Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization".

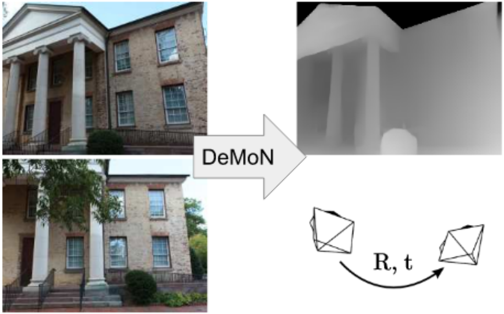
Local scene coordinates



61

[61] Brachmann and Rother, "Learning Less is More - 6D Camera Localization via 3D Surface Regression".

DeMoN

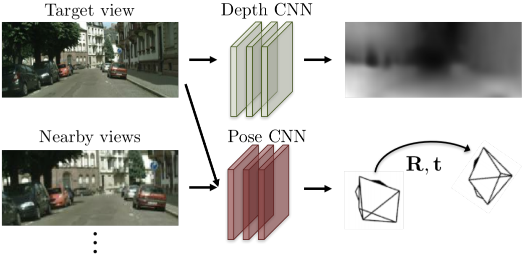


[62]

[62] Ummenhofer et al., "DeMoN: Depth and Motion Network for Learning Monocular Stereo".

Unsupervised learning of camera transformation

(a) Training: unlabeled video clips.



(b) Testing: single-view depth and multi-view pose estimation.[63]

[63] Zhou et al., "Unsupervised Learning of Depth and Ego-Motion from Video".

Open questions - Benchmarking

- Are matching benchmarks representative?
- How can we correctly evaluate methods by eliminating other nuisance factors?

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?

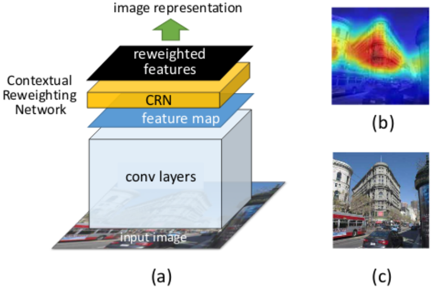
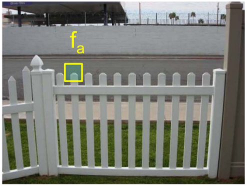
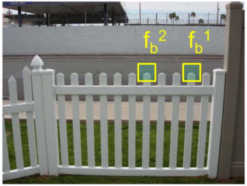


Figure 1. Image representation with contextual feature reweighting. (a) A contextual reweighting network takes convolutional features of a deep CNN as input to produce a spatial weighting mask (b) based on the learned contexts. The mask is used for weighted aggregation of input features to produce the representation of the input image (c). [64]

[64] Kim, Dunn, and Frahm, "Learned Contextual Feature Reweighting for Image Geo-Localization".


f_a


f_b^2
f_b^1

Related CVPR 2019 Workshops

Long-Term Visual Localization under Changing Conditions

T.Sattler, **V. Balntas**, M. Pollefeys, K. Mikolajczyk, J. Sivic, T. Pajdla, L. Hammarstrand, H. Heijnen, F. Kahl, W. Maddern, C. Toft, A. Torii

Includes a Challenge on Local Features

Image Matching: Local Features and Beyond

V. Balntas, E. Trulls, K.M. Yi, J. Shonberger, V. Lepetit

Includes a Challenge on Local Features

The End - Thanks

Please consider taking part in the CVPR 2019 workshop challenges!