计算机视觉表征与识别

Chapter 8: Interest Points: Descriptor

王利民

媒体计算课题组

http://mcg.nju.edu.cn/

Submission Requirements and Description (Very Important !)

Format requirements

(i) Please use the provided *Latex template* to write your report, and the report should contain your name, student ID, and e-mail address;

(ii) You should choose between Matlab and python to write your code, and provide a README file to describe how to execute the code;

(iii) Pack your **report**.pdf, **code** and **README** into a zip file, named with your student ID, like MG1833001.zip. If you have an improved version, add an extra '_' with a number, like MG1833001_1.zip. We will take the final submitted version as your results.

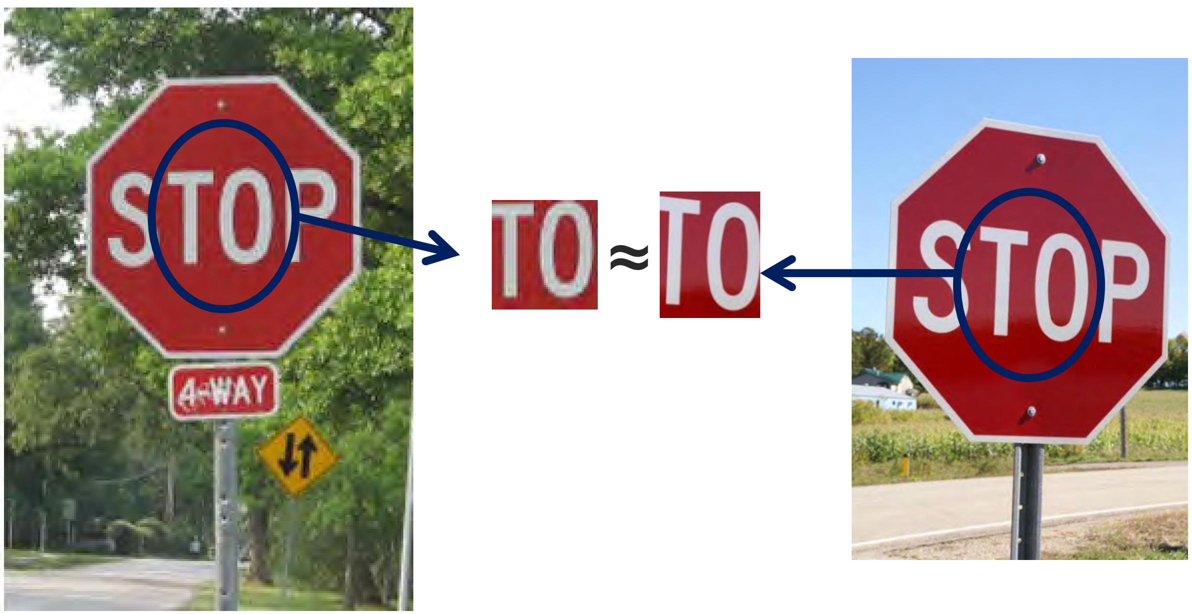
Submission Way

(i) Please submit your results to email nju.cvcourse@gmail.com , the email subject is "**Assignment 3**";

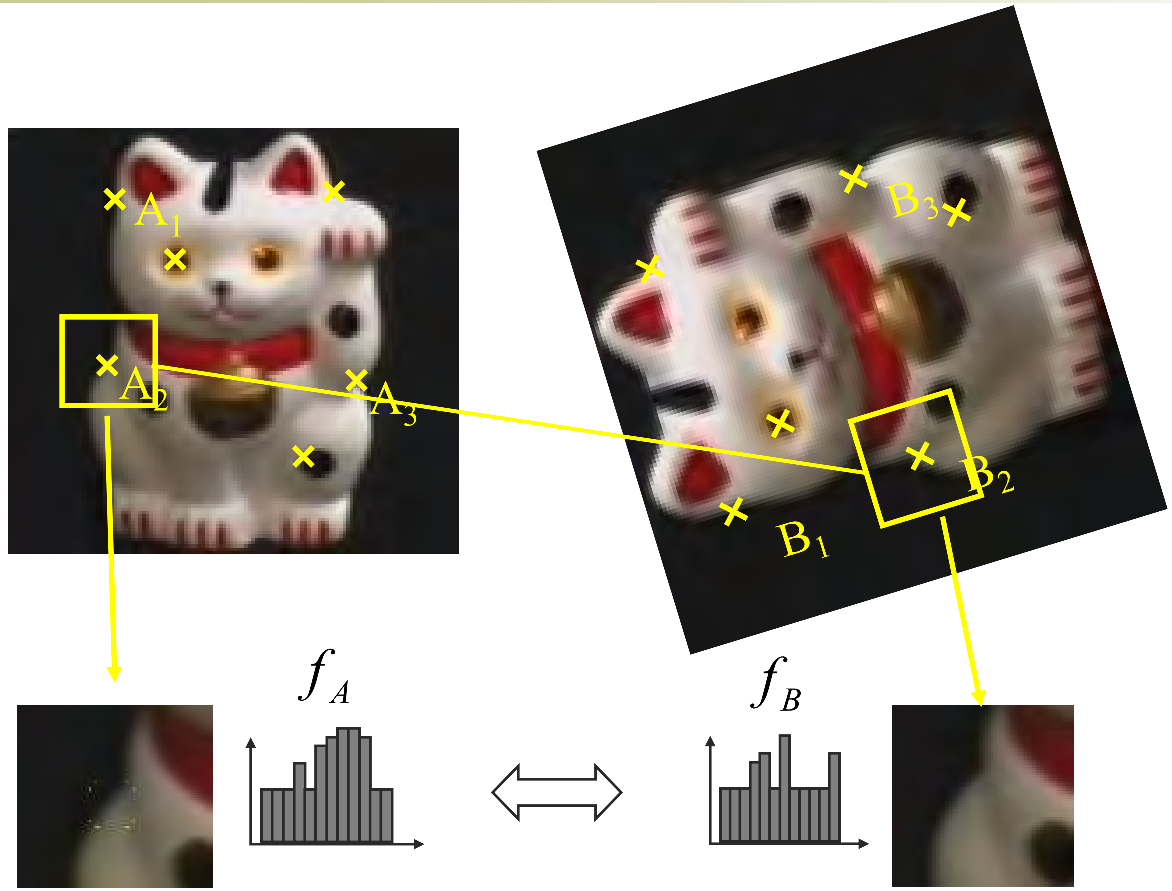
(ii) The deadline is **23:59 on June 21, 2021**. No submission after this deadline is acceptable.

Correspondence and alignment

Correspondence: matching points, patches, edges, or regions across images



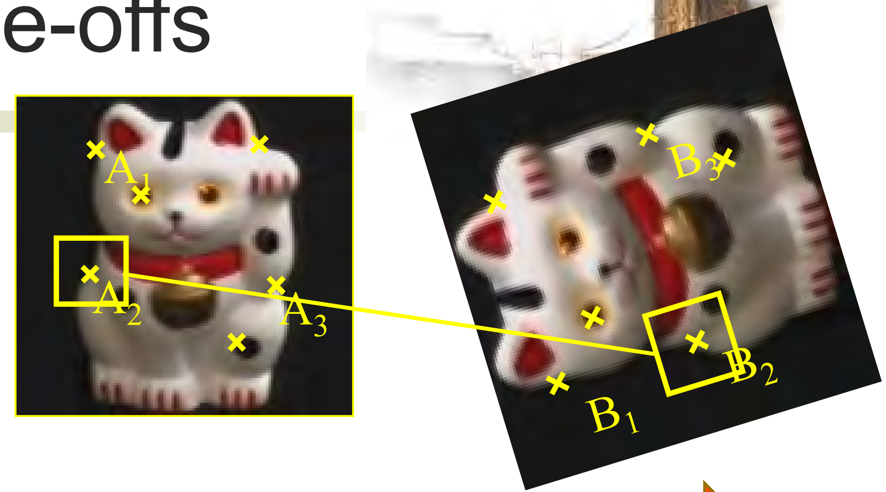
Recap: Keypoint Matching



$$d(f_A, f_B) < T$$

1. Find a set of distinctive key-points
2. Define a region around each keypoint
3. Extract and normalize the region content
4. Compute a local descriptor from the normalized region
5. Match local descriptors

Recap: Key trade-offs



Detection

More Repeatable

Robust detection
Precise localization

More Points

Robust to occlusion
Works with less texture

Description

More Distinctive

Minimize wrong matches

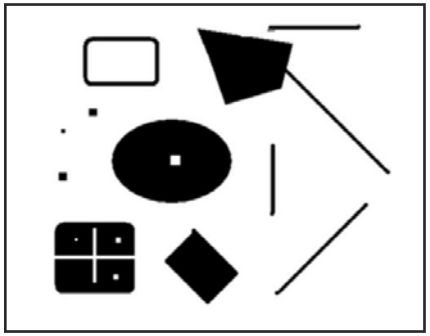
More Flexible

Robust to expected variations
Maximize correct matches

Harris Detector [Harris88]

- Second moment matrix

$$\mu(\sigma_I, \sigma_D) = g(\sigma_I) * \begin{bmatrix} I_x^2(\sigma_D) & I_x I_y(\sigma_D) \\ I_x I_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

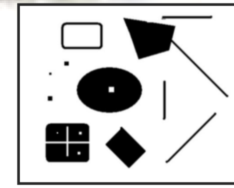


Intuition: **Search for local neighborhoods where the image content has two main directions (eigenvectors).**

C.Harris and M.Stephens. "A Combined Corner and Edge Detector.“ *Proceedings of the 4th Alvey Vision Conference*, 1988.

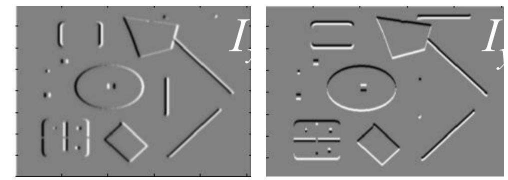
Harris Detector [Harris88]

- Second moment matrix



$$\mu(\sigma_I,\sigma_D)=g(\sigma_I)*\begin{bmatrix} I_x^2(\sigma_D) & I_xI_y(\sigma_D) \\ I_xI_y(\sigma_D) & I_y^2(\sigma_D) \end{bmatrix}$$

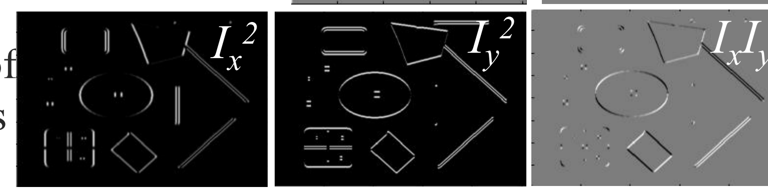
1. Image derivatives (optionally, blur first)



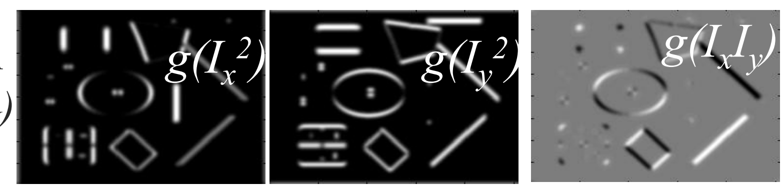
$$\det M = \lambda_1\lambda_2$$

$$\operatorname{trace} M = \lambda_1+\lambda_2$$

2. Square of derivatives



3. Gaussian filter $g(\sigma_I)$



4. Cornerness function – both eigenvalues are strong

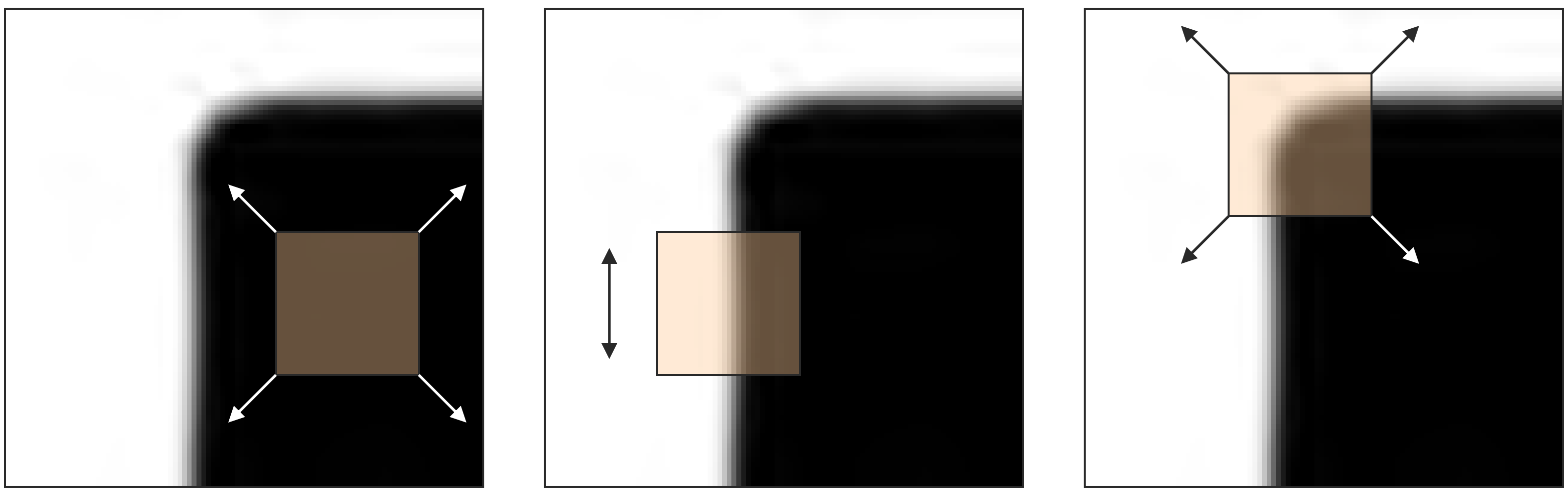
$$har=\det[\mu(\sigma_I,\sigma_D)]-\alpha[\operatorname{trace}(\mu(\sigma_I,\sigma_D))^2]=$$

$$g(I_x^2)g(I_y^2)-[g(I_xI_y)]^2-\alpha[g(I_x^2)+g(I_y^2)]^2$$

5. Non-maxima suppression

Corners as distinctive interest points

- We should easily recognize the point by looking through a small window
- Shifting a window in *any direction* should give *a large change* in intensity



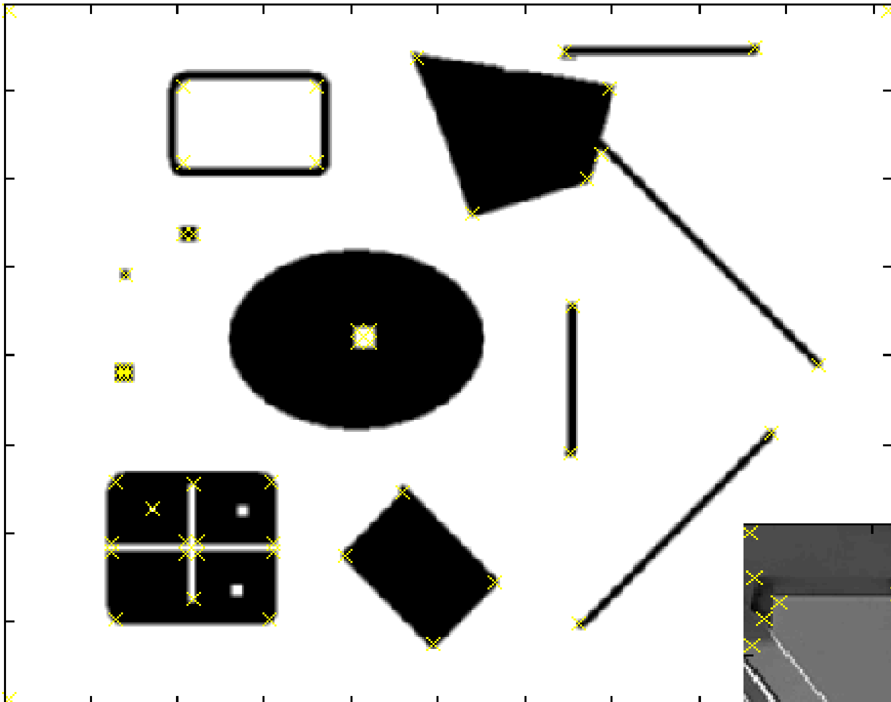
"flat" region: no change in all directions

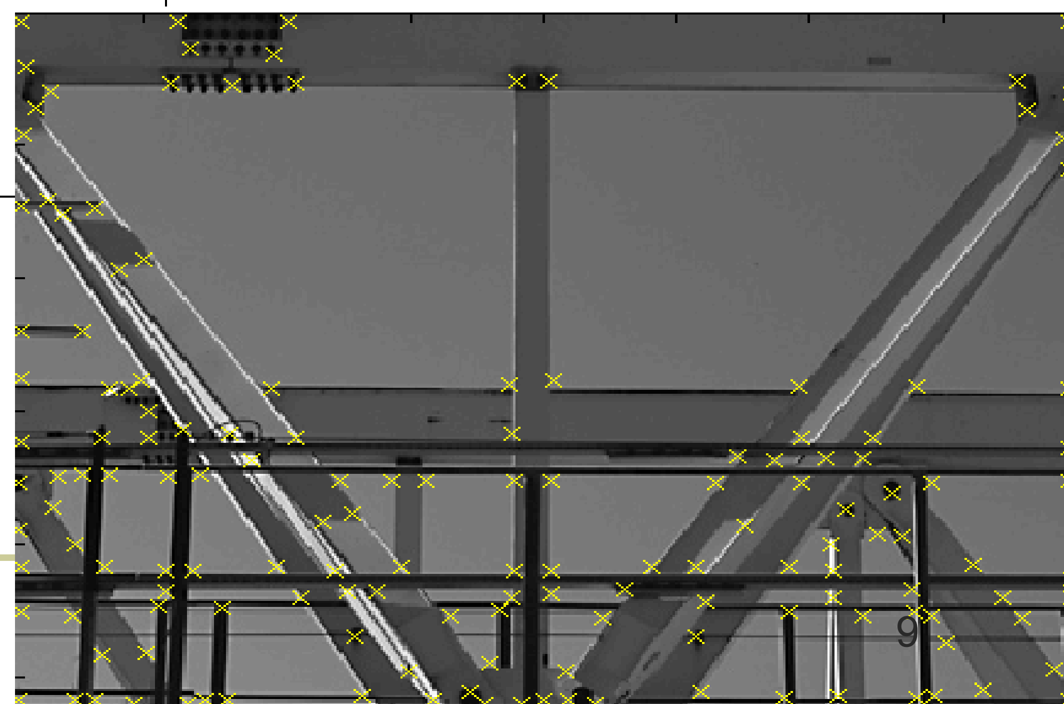
"edge": no change along the edge direction

"corner": significant change in all directions

Slide credit: Alyosha Efros, Darya Frolova, Denis Simakov

Harris Detector – Responses [Harris88]



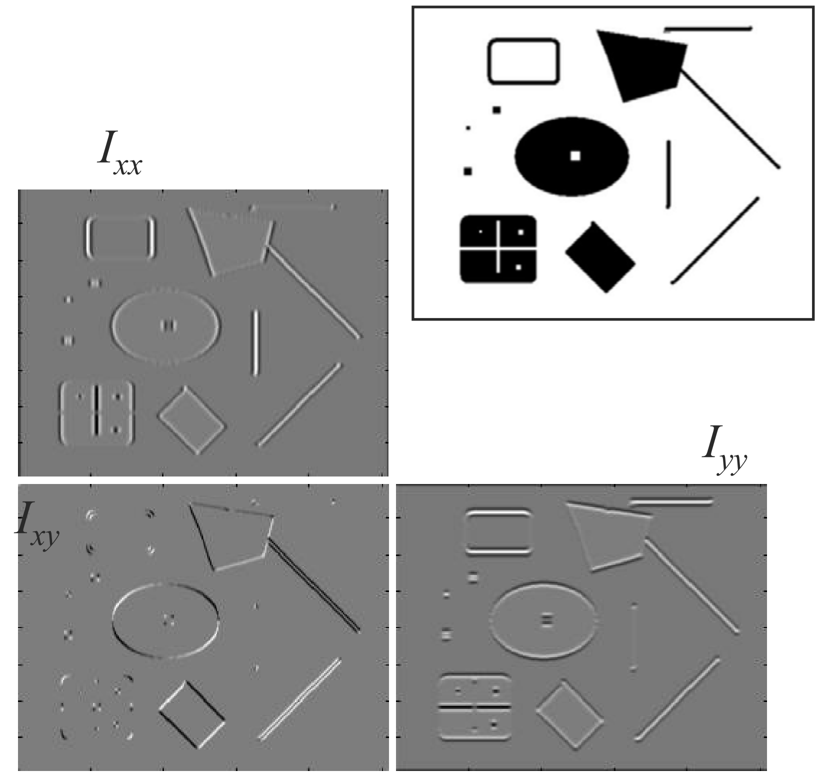


Effect: **A very precise corner detector.**

Hessian Detector [Beaudet78]

- Hessian determinant

$$Hessian(I) = \begin{bmatrix} I_{xx} & I_{xy} \\ I_{xy} & I_{yy} \end{bmatrix}$$



Intuition: **Search for strong curvature in two orthogonal directions**

Hessian Detector [Beaudet78]

- Hessian determinant

$$Hessian\,(x,\sigma) = \begin{bmatrix} I_{xx}(x,\sigma) & I_{xy}(x,\sigma) \\ I_{xy}(x,\sigma) & I_{yy}(x,\sigma) \end{bmatrix}$$

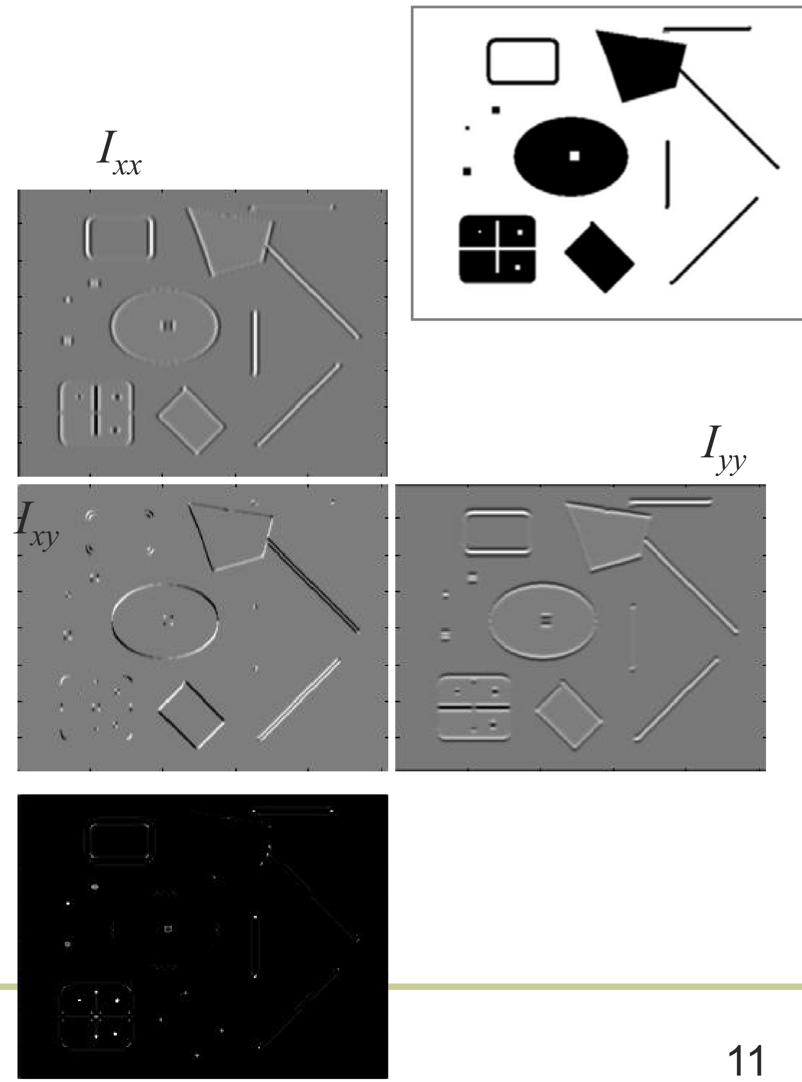
$$\det M = \lambda_1 \lambda_2$$
$$\operatorname{trace} M = \lambda_1 + \lambda_2$$

Find maxima of determinant

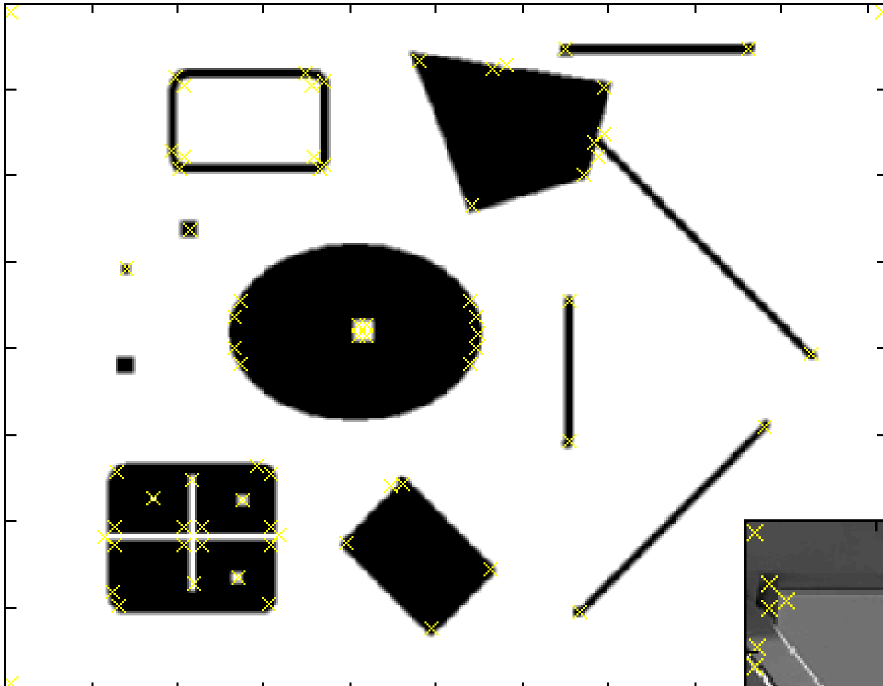
$$\det(Hessian(x)) = I_{xx}(x) I_{yy}(x) - I_{xy}^2(x)$$

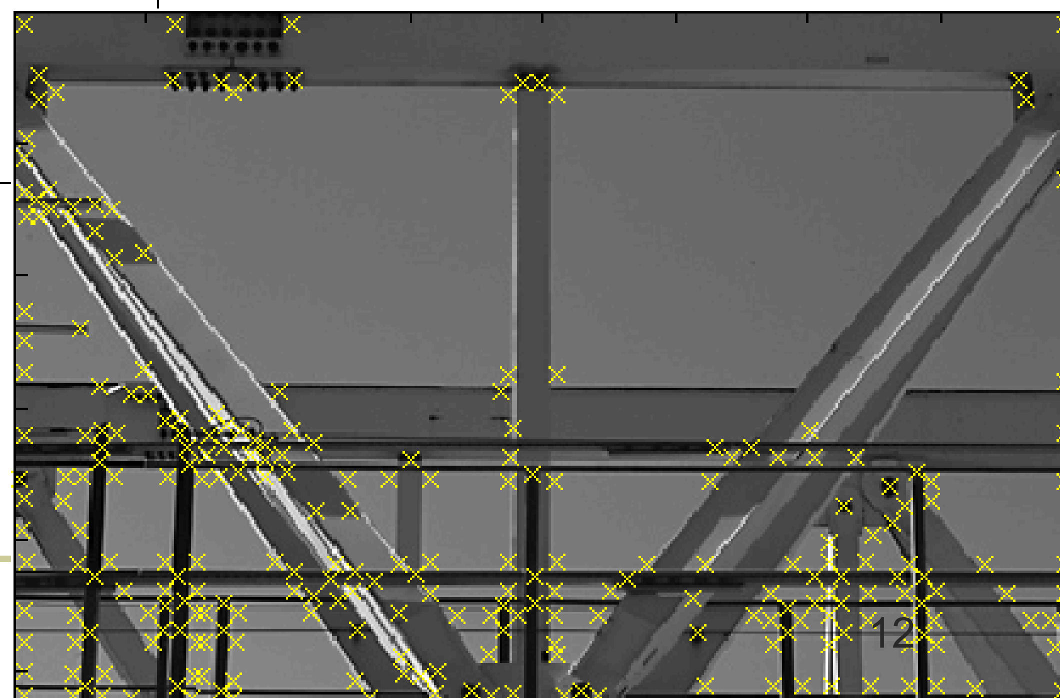
In Matlab:

$$I_{xx} .* I_{yy} - (I_{xy})\text{^}2$$



Hessian Detector – Responses [Beaudet78]

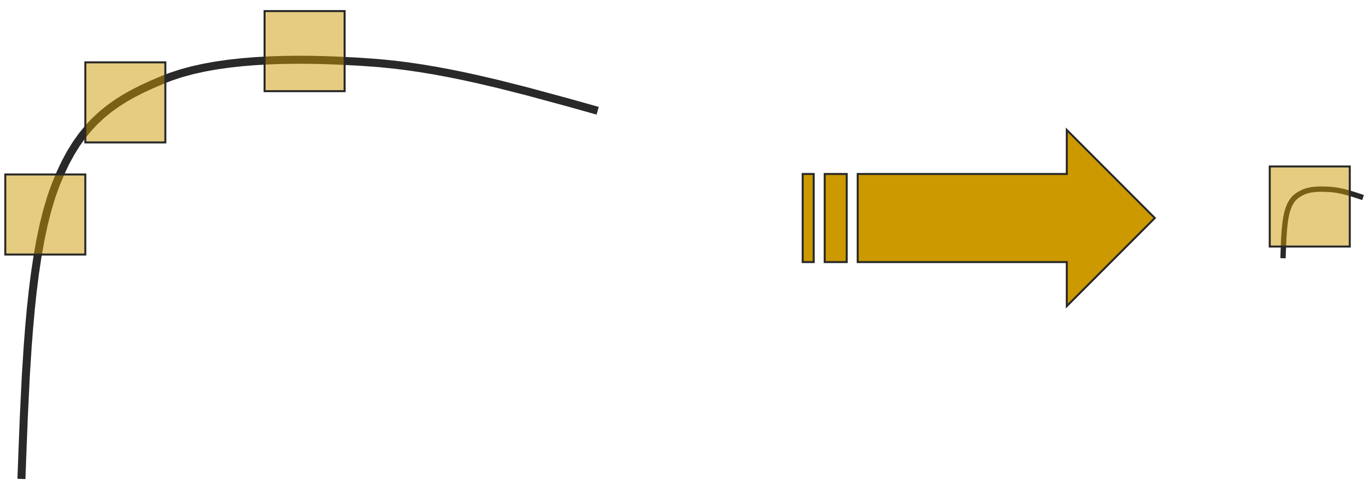




Effect: **Responses mainly on corners and strongly textured areas.**

Scale invariance?

- Scale invariant? No



All points will be classified as edges

Corner !

From points to regions

- **The Harris and Hessian operators define interest points.**
  - **Precise localization**
  - **High repeatability**

- **In order to compare those points, we need to compute a descriptor over a region.**
  - **How can we define such a region in a scale invariant manner?**

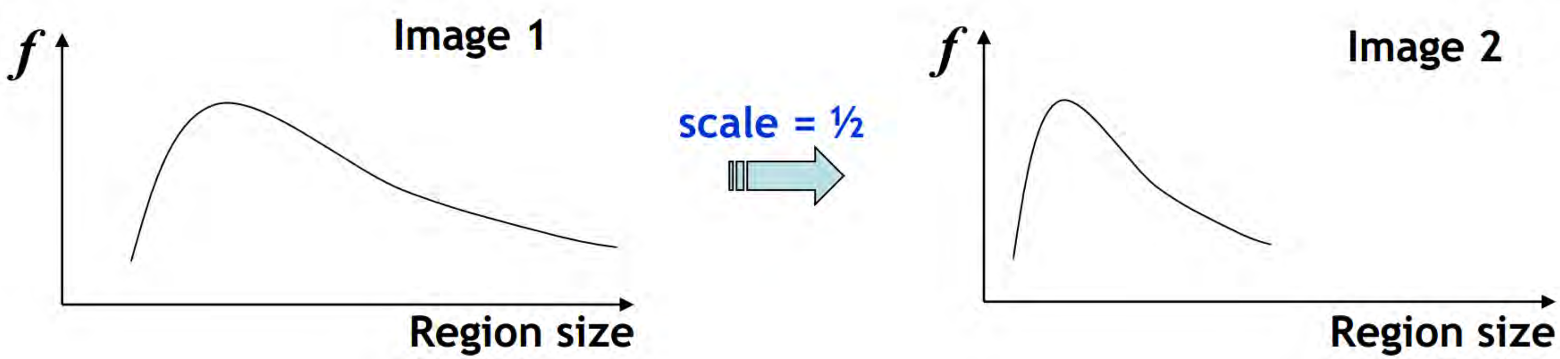
- ***I.e. how can we detect scale invariant interest regions?***

Automatic scale selection

- **Solution:**
  - Design a function on the region, which is "scale invariant" (*the same for corresponding regions, even if they are at different scales*)

    Example: average intensity. For corresponding regions (even of different sizes) it will be the same.

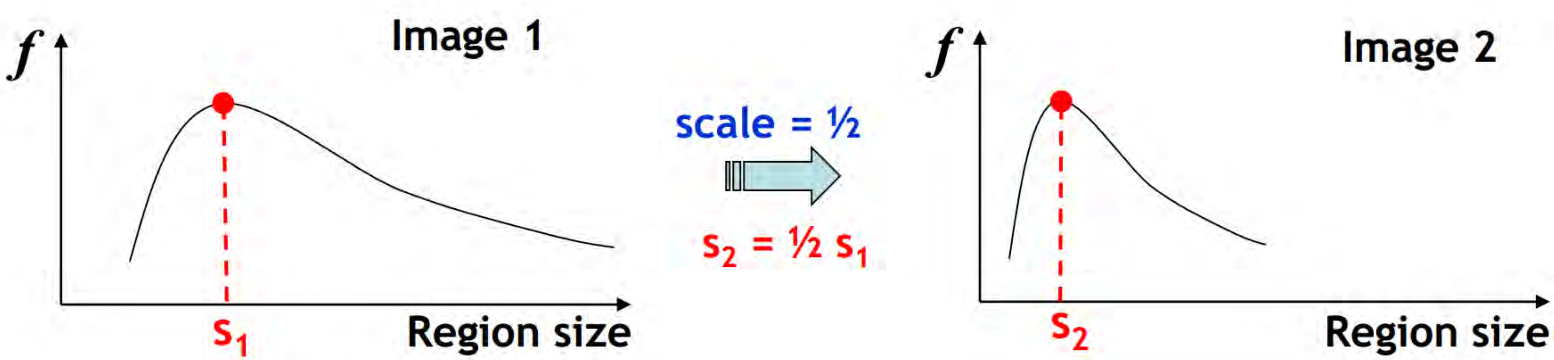
  - For a point in one image, we can consider it as a function of region size (patch width)



Automatic scale selection

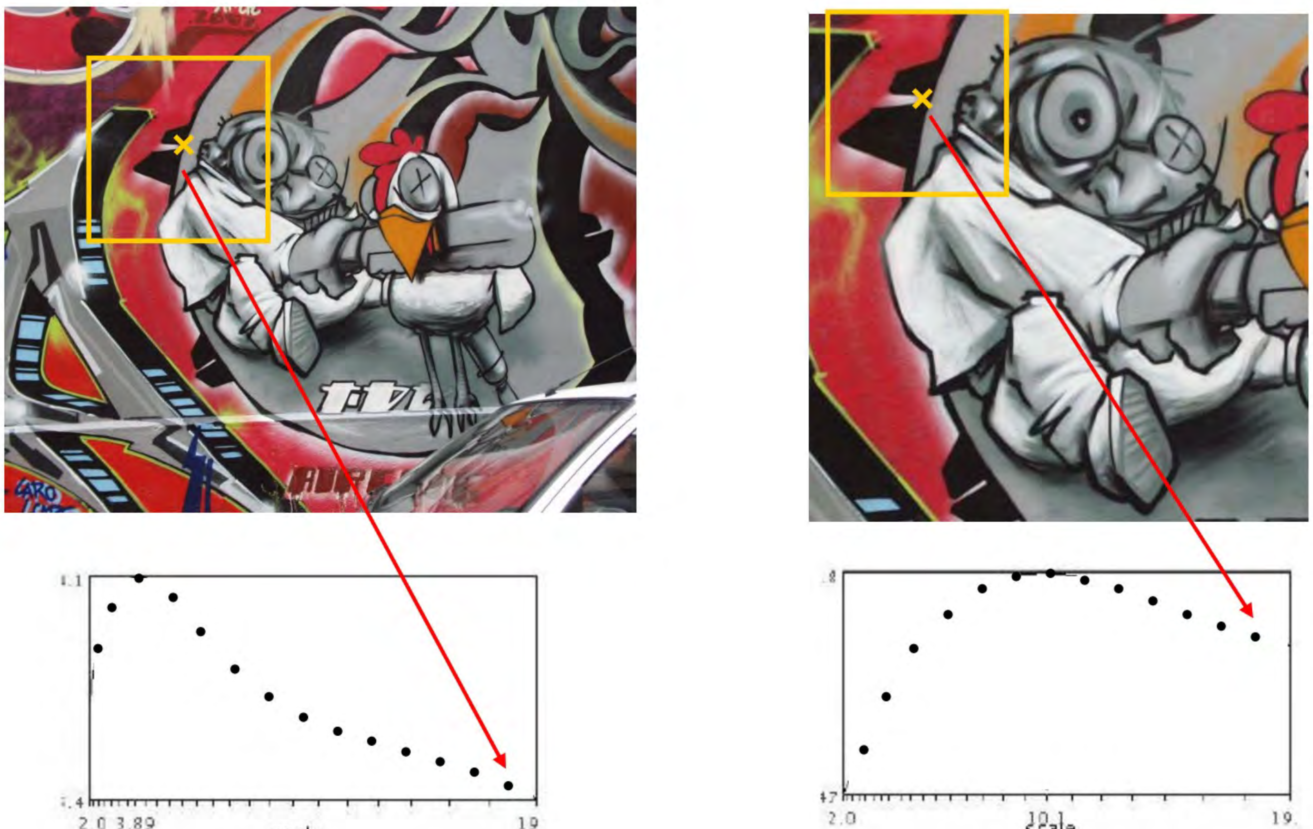
- **Common approach:**
  - **Take a local maximum of this function.**
  - **Observation: region size for which the maximum is achieved should be *invariant* to image scale.**

Important: this scale invariant region size is found in each image independently!



Automatic scale selection

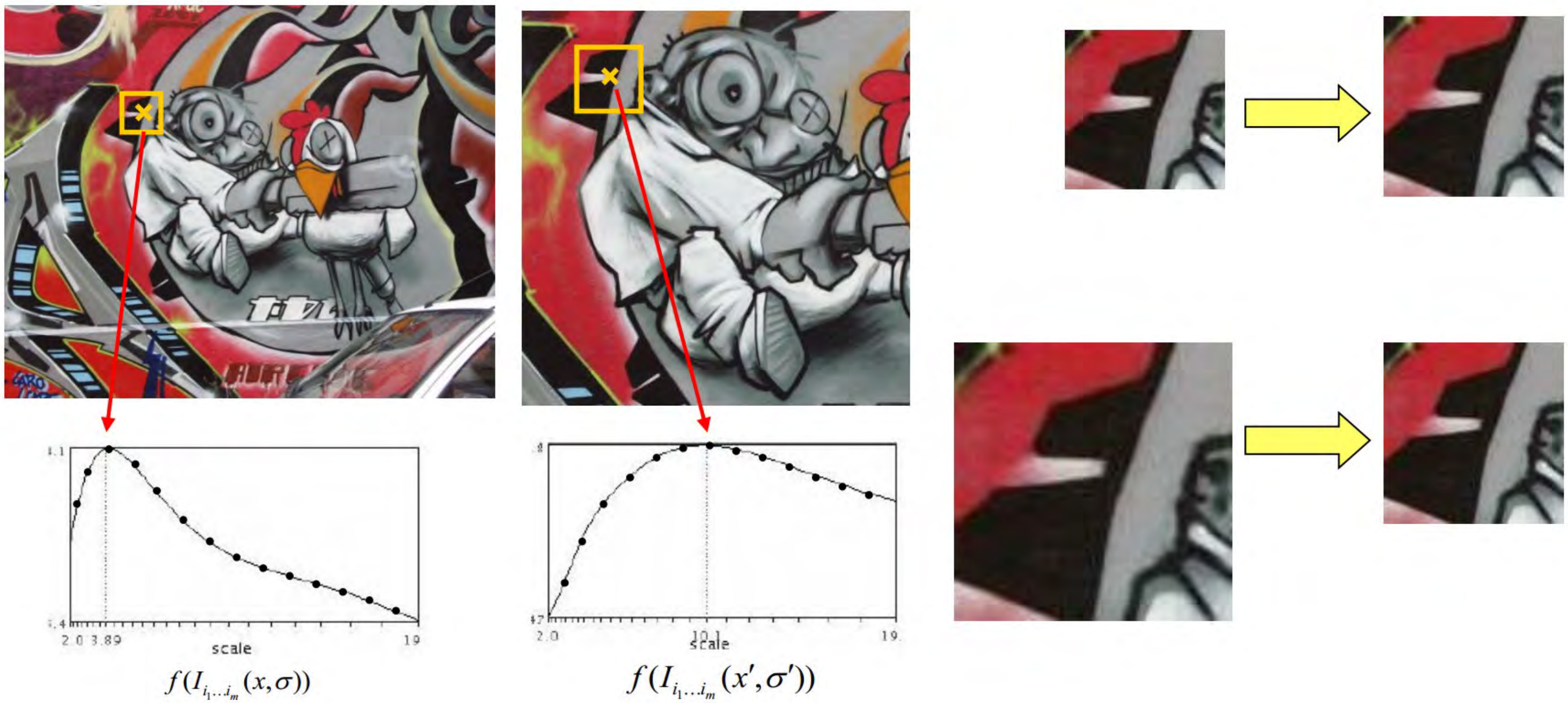
- **Function responses for increasing scale (scale signature)**



$f(I_{i_1 \dots i_m}(x,\sigma))$ $f(I_{i_1 \dots i_m}(x',\sigma))$

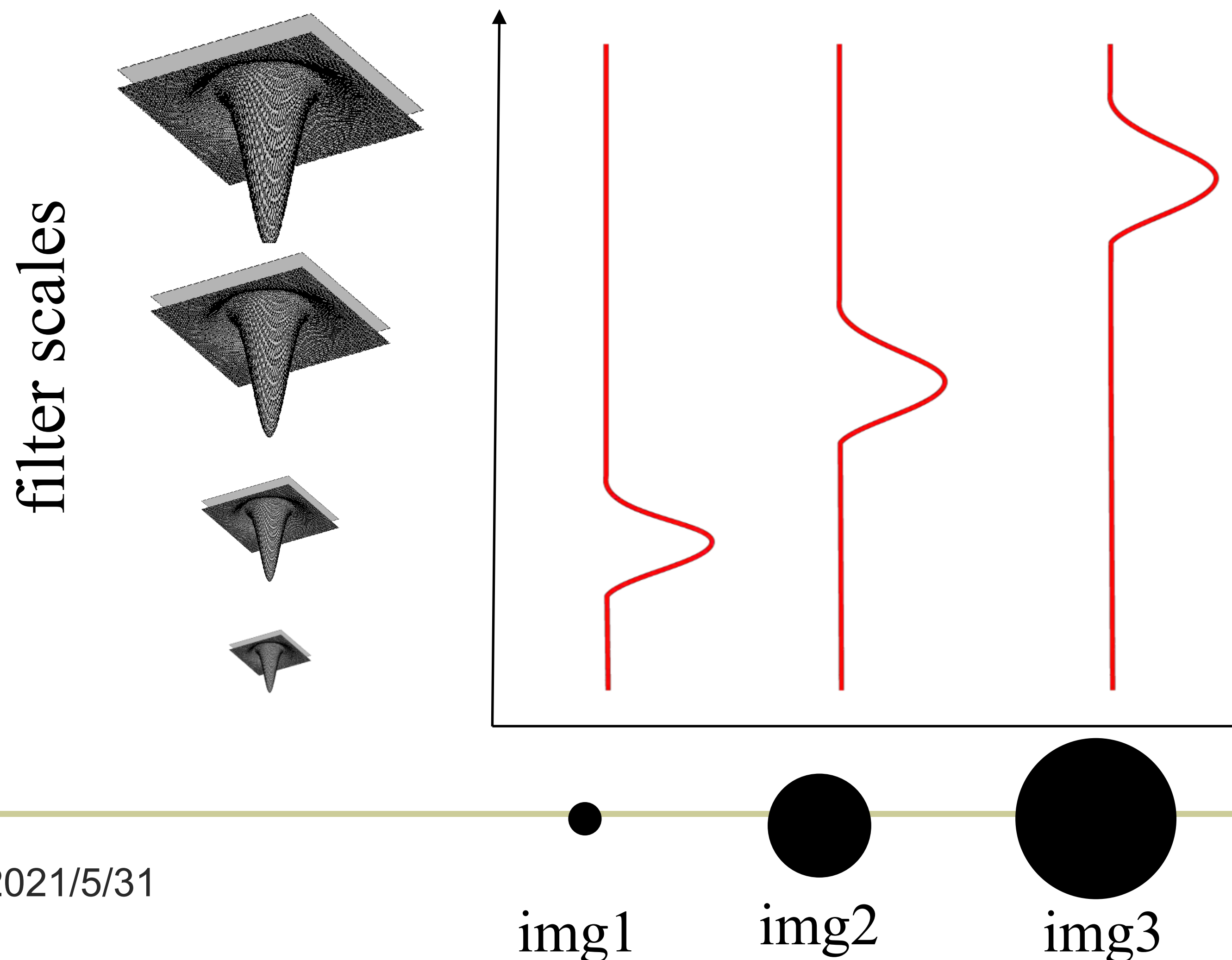
Automatic scale selection

- **Normalize: Rescale to fixed size**



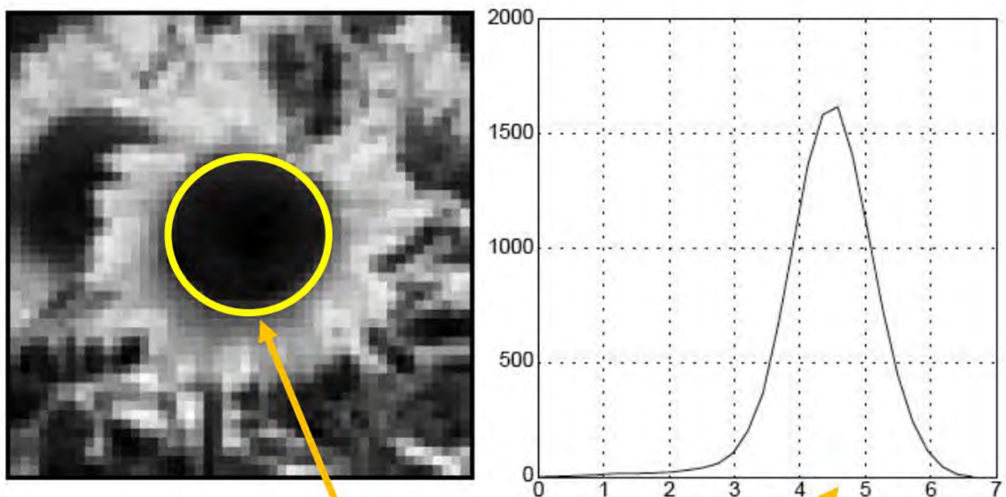
Blob detection in 2D

- Laplacian-of-Gaussian = "blob" detector $\nabla^2 g = \frac{\partial^2 g}{\partial x^2} + \frac{\partial^2 g}{\partial y^2}$



Characteristic scale

- We define the *characteristic scale* as the scale that produces peak of Laplacian response

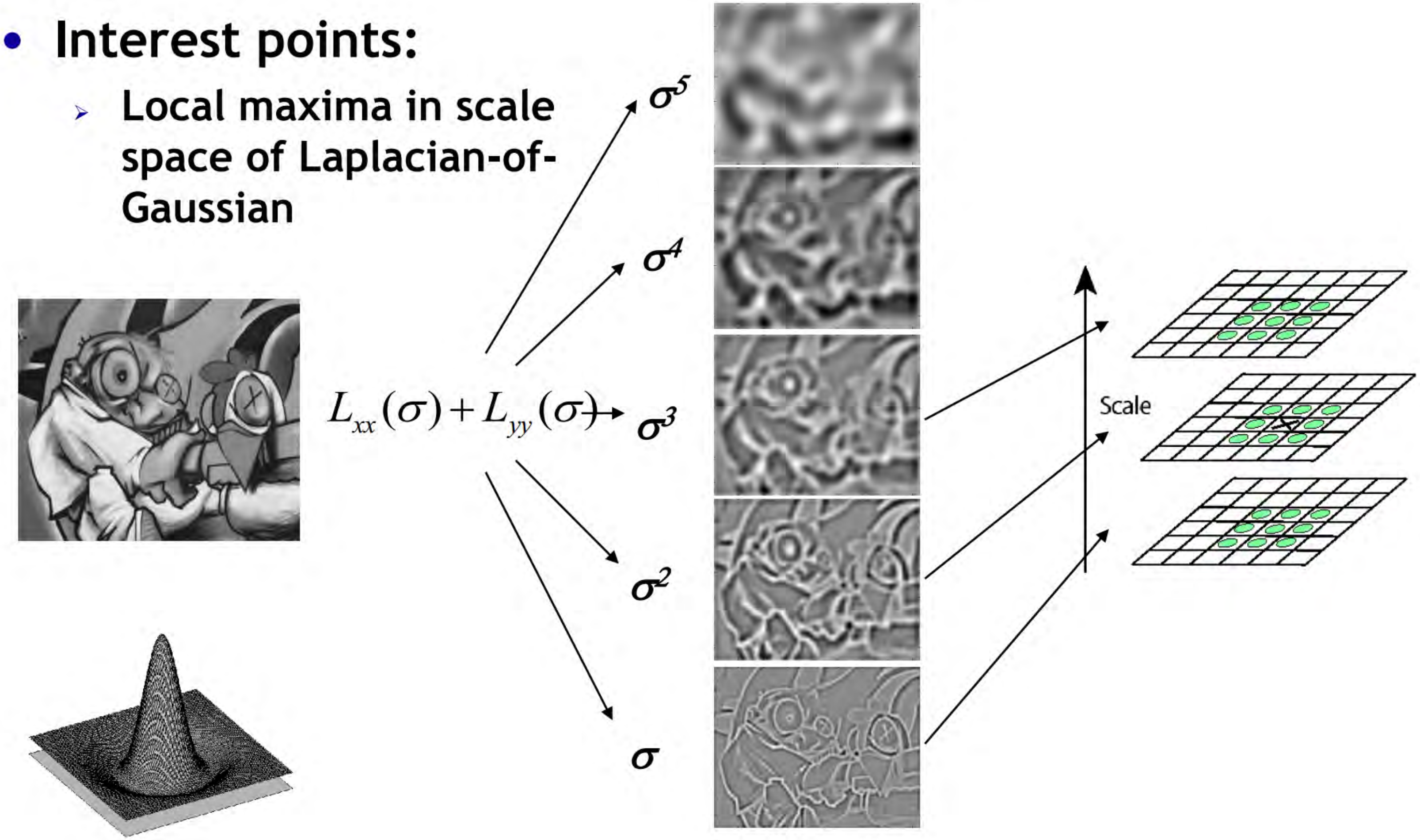


Characteristic scale

T. Lindeberg (1998). "Feature detection with automatic scale selection." *International Journal of Computer Vision* 30 (2): pp 77--116.

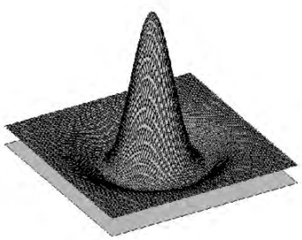
Laplacian-of-Gaussian (LoG)

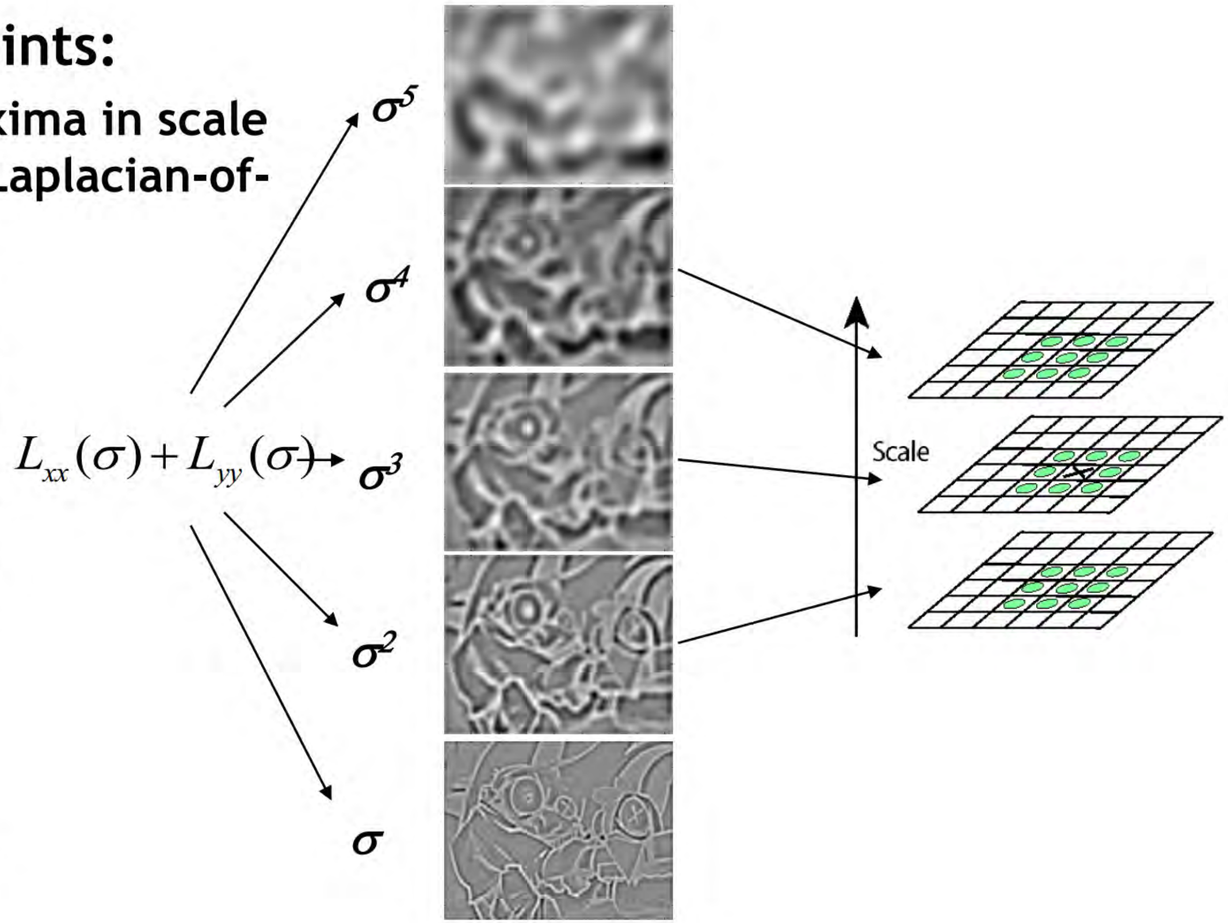
- **Interest points:**
  - **Local maxima in scale space of Laplacian-of-Gaussian**



Laplacian-of-Gaussian (LoG)

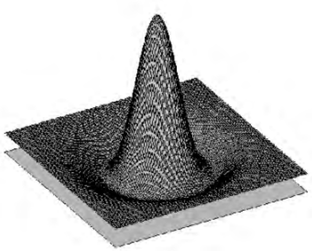
- **Interest points:**
  - **Local maxima in scale space of Laplacian-of-Gaussian**

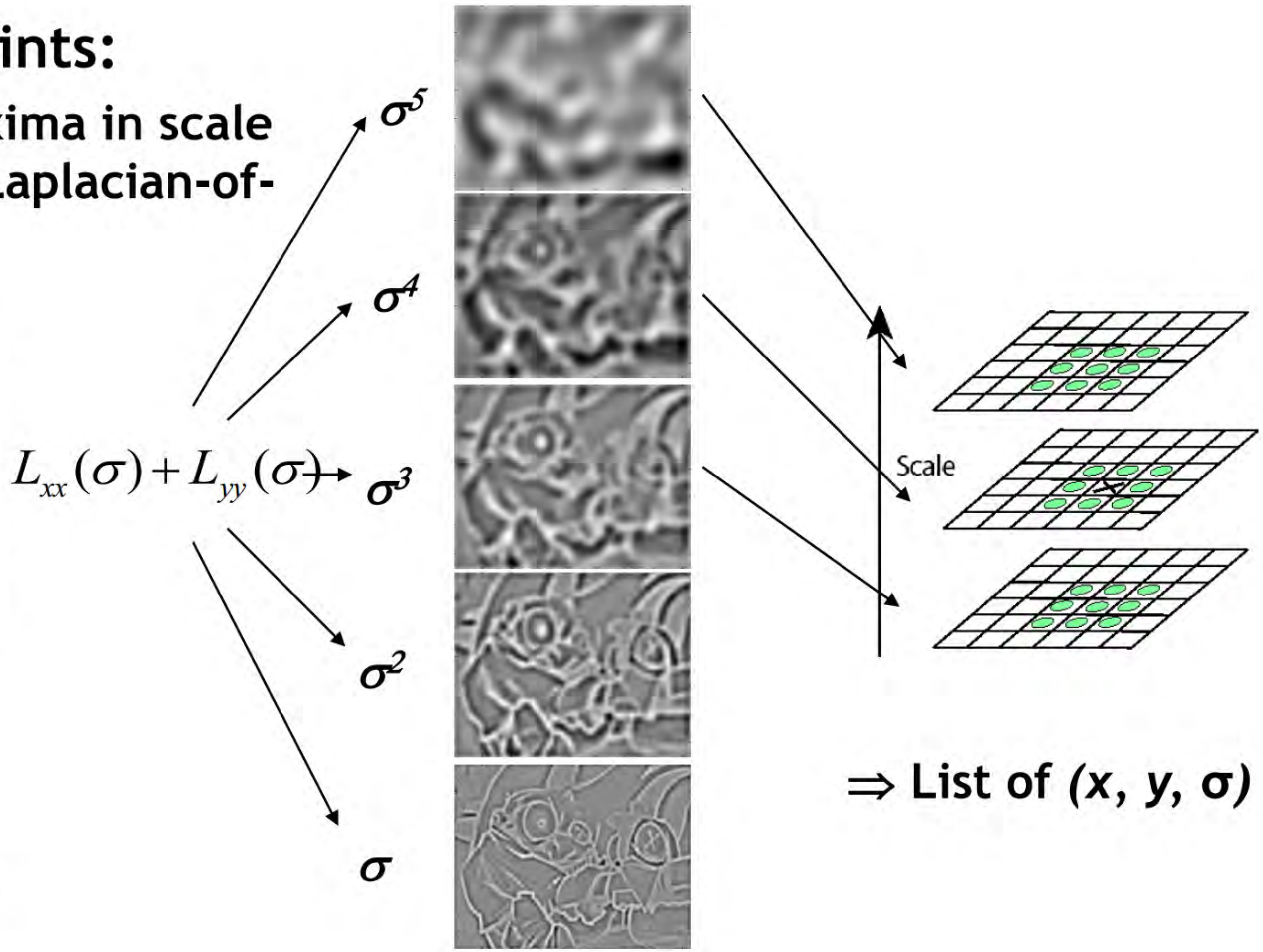




Laplacian-of-Gaussian (LoG)

- **Interest points:**
  - **Local maxima in scale space of Laplacian-of-Gaussian**





⇒ List of *(x, y, σ)*

Technical detail

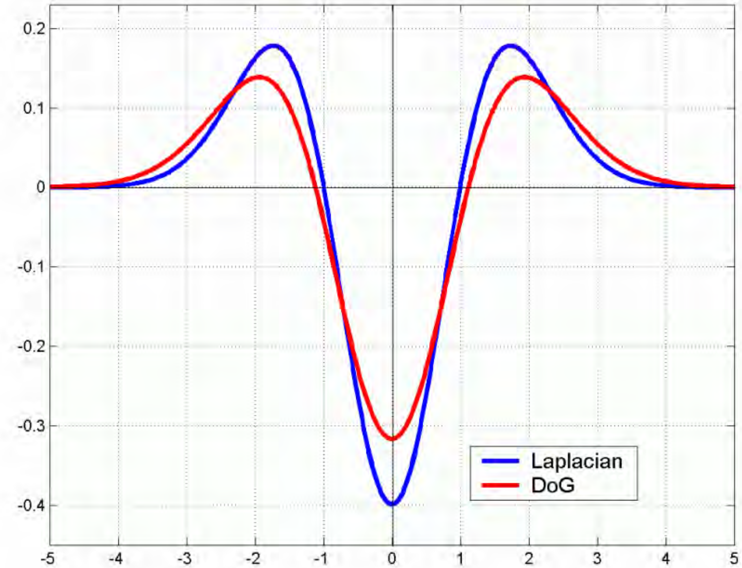
- **We can efficiently approximate the Laplacian with a difference of Gaussians:**

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

(Laplacian)

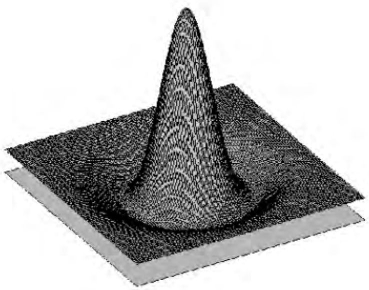
$$DoG = G(x, y, k\sigma) - G(x, y, \sigma)$$

(Difference of Gaussians)



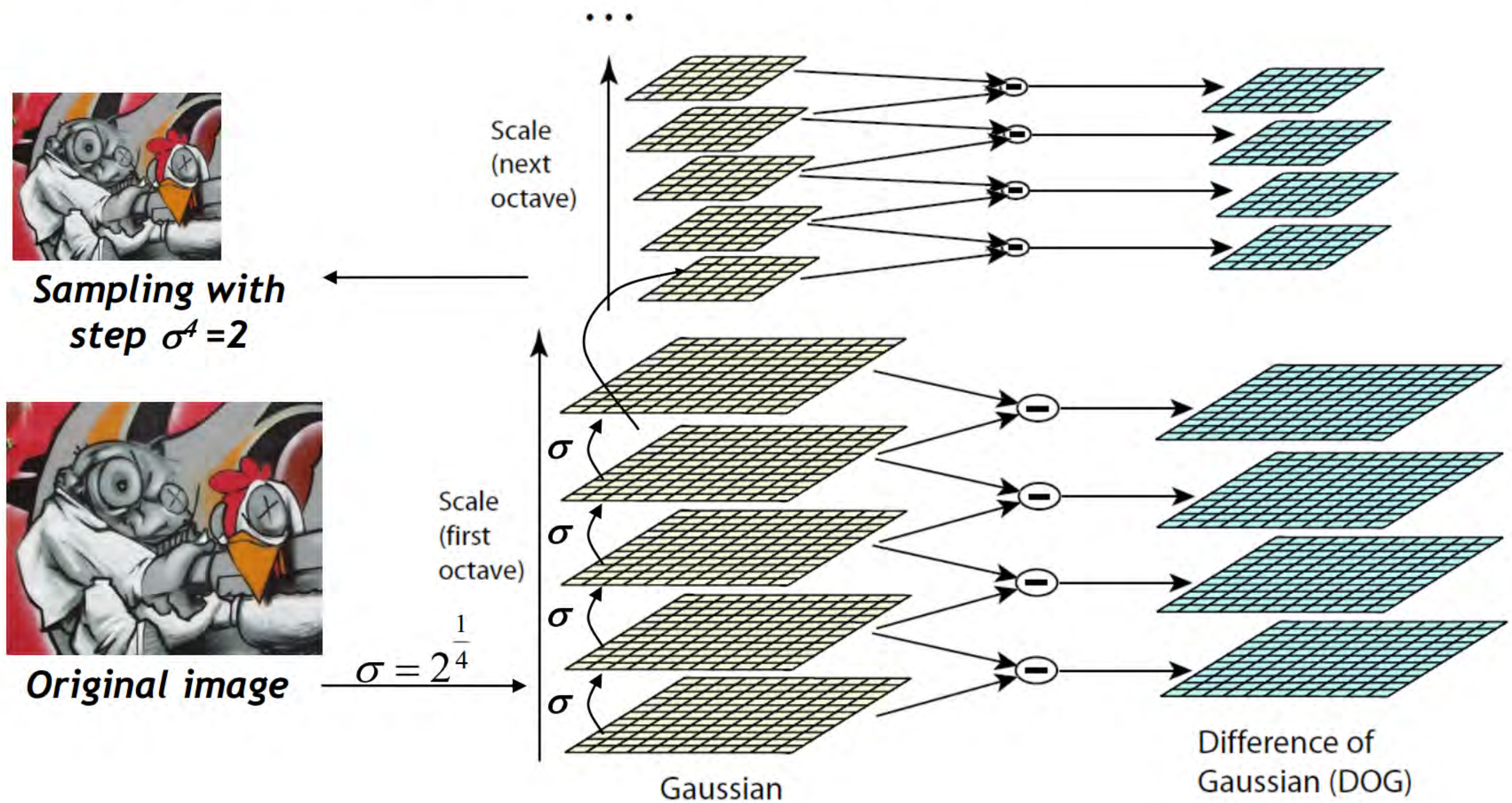
Difference-of-Gaussian(DoG)

- **Difference of Gaussians as approximation of the LoG**
  - **This is used e.g. in Lowe's SIFT pipeline for feature detection.**
- **Advantages**
  - **No need to compute 2nd derivatives**
  - **Gaussians are computed anyway, e.g. in a Gaussian pyramid.**



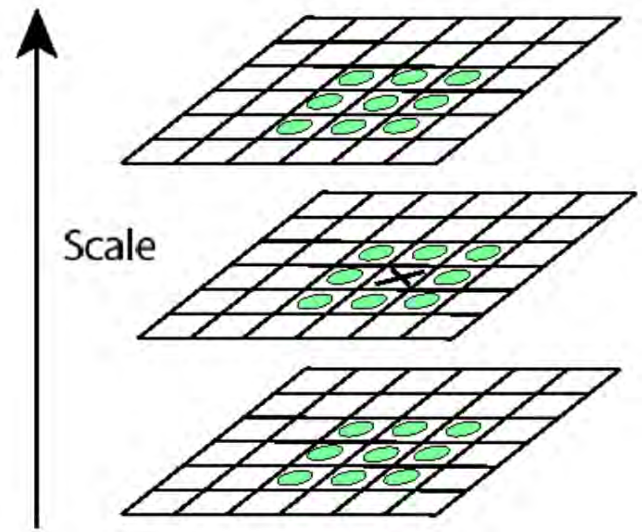
DoG: Efficient implementation

- **Computation in Gaussian scale pyramid**



Keypoint localization with DoG

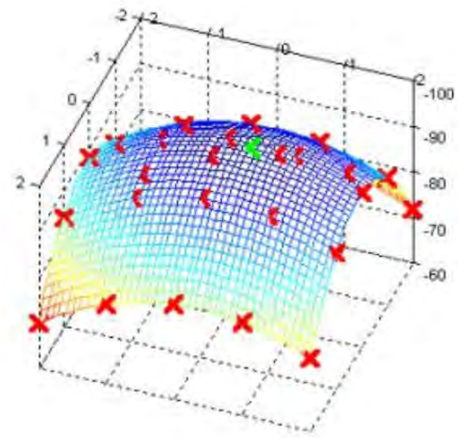
- Detect maxima of difference-of-Gaussian (DoG) in scale space
- Then reject points with low contrast (threshold)
- Eliminate edge responses



Candidate keypoints: list of (x,y,σ)

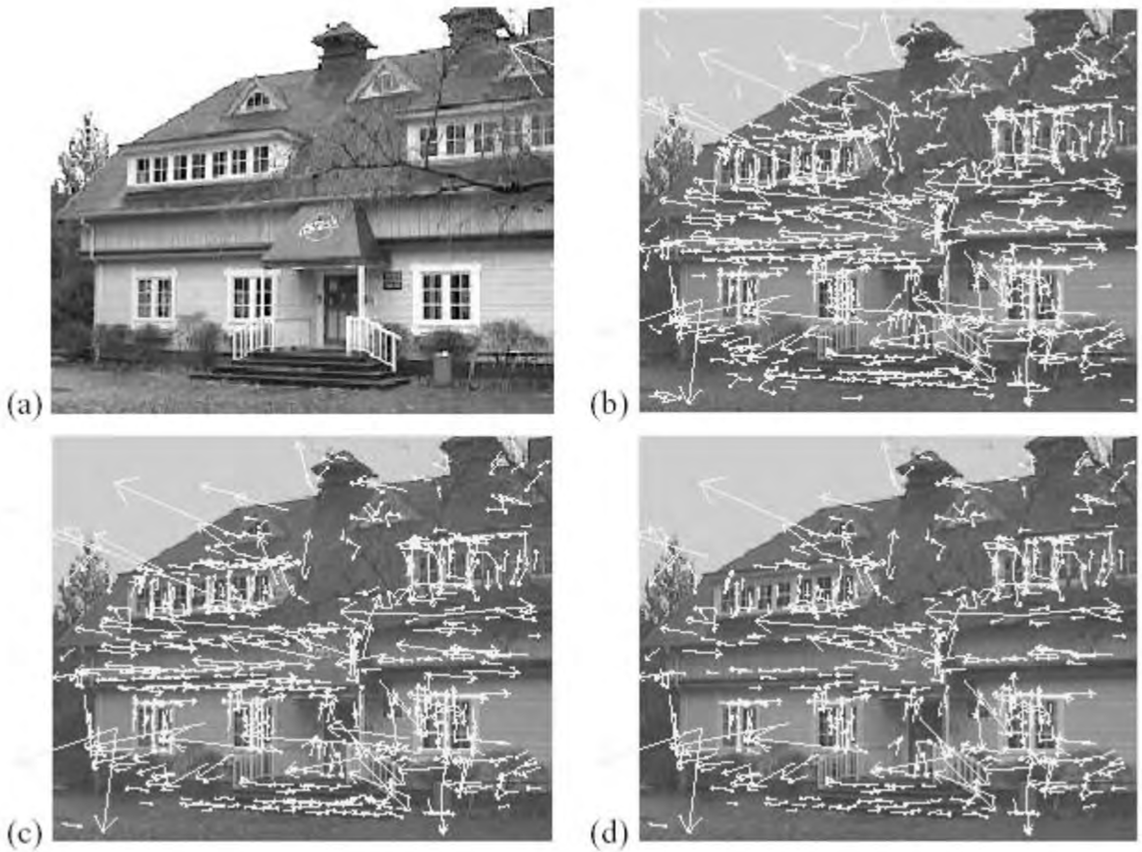
Keypoint Refinement

- **Refine**:
  - Fit a 3D (x,y,scale) curve to the initial keypoint, and find the peak in the curve as the refined keypoint.



- **Elimination**:
  - Discard keypoints with low refined DoG response.
  - Discard keypoints with high edge response.

Example of Keypoint Detection



(a) 233x189 image

(b) 832 DoG extrema

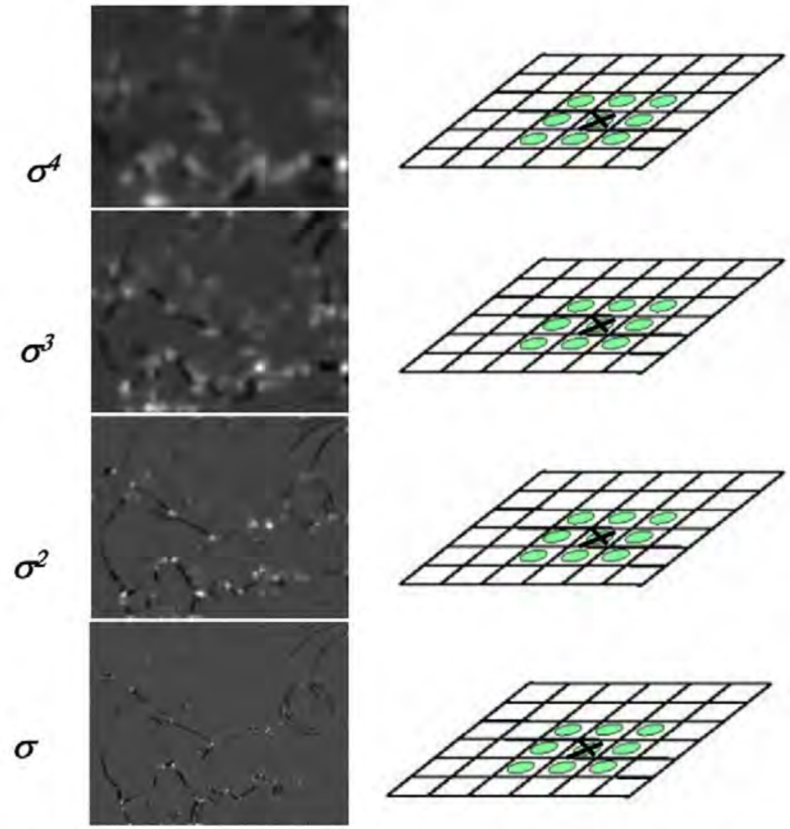
(c) 729 left after peak value threshold

(d) 536 left after testing ratio of principle curvatures (removing edge responses)

Results: Lowe's DoG

Harris-Laplace [Mikolajczyk ‘01]

1. Initialization: Multiscale Harris corner detection



Computing Harris function **Detecting local maxima**

Slide adapted from Krystian Mikolajczyk

Harris-Laplace [Mikolajczyk '01]

1. Initialization: Multiscale Harris corner detection
2. Scale selection based on Laplacian
   (same procedure with Hessian ⇒ Hessian-Laplace)

Harris points

Harris-Laplace points

Summary: Scale Invariant Detection

- **Given:** Two images of the same scene with a large *scale difference* between them.
- **Goal:** Find *the same* interest points *independently* in each image.
- **Solution:** Search for *maxima* of suitable functions in *scale* and in *space* (over the image).

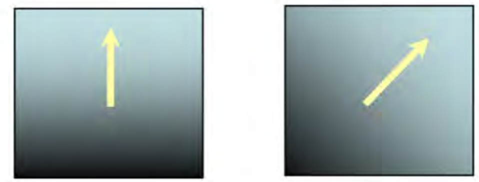
- Two strategies
  - Laplacian-of-Gaussian (LoG)
  - Difference-of-Gaussian (DoG) as a fast approximation
  - *These can be used either on their own, or in combinations with single-scale keypoint detectors (Harris, Hessian).*

Today's class

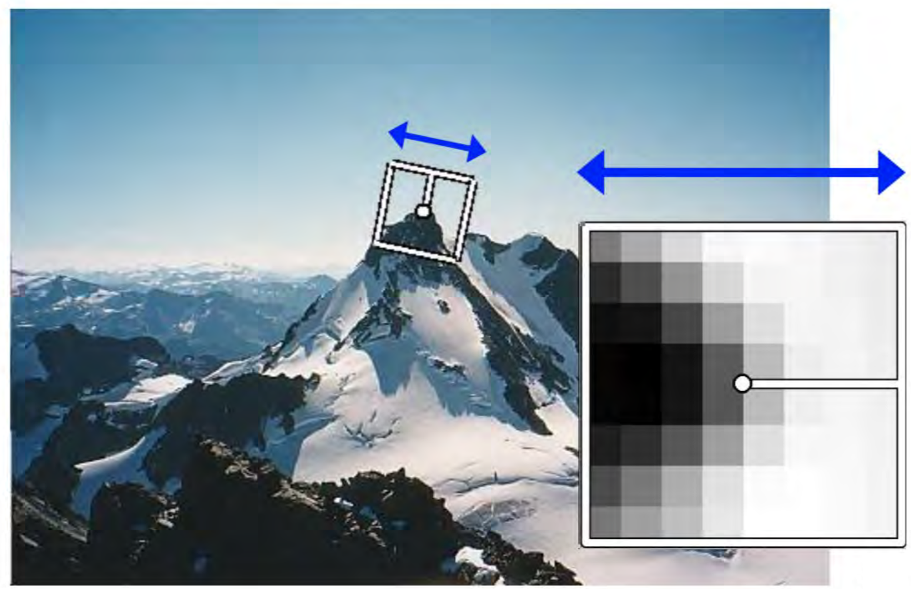
- Normalization
  - Orientation normalization
  - Affine invariant feature extraction
- Local descriptor
  - SIFT, SURF, GIST
- Binary descriptor
  - LBP, BRIEF
- CNN based descriptor
  - MatchNet, DeepCompare, DeepDesc, LIFT

Rotation Invariant Descriptors

- **Find local orientation**
  - Dominant direction of gradient for the image patch



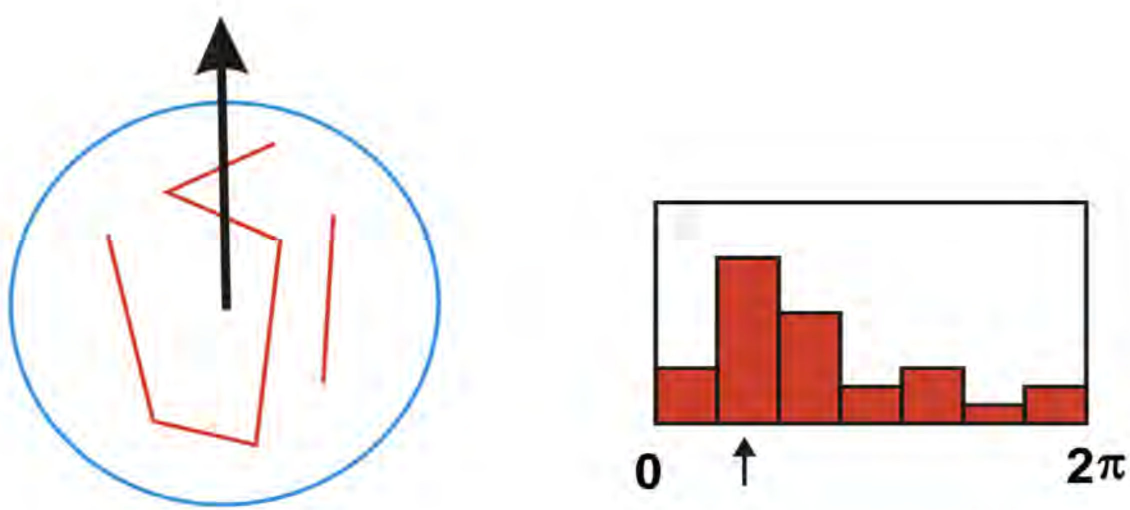
- **Rotate patch according to this angle**
  - This puts the patches into a canonical orientation.



Orientation Normalization

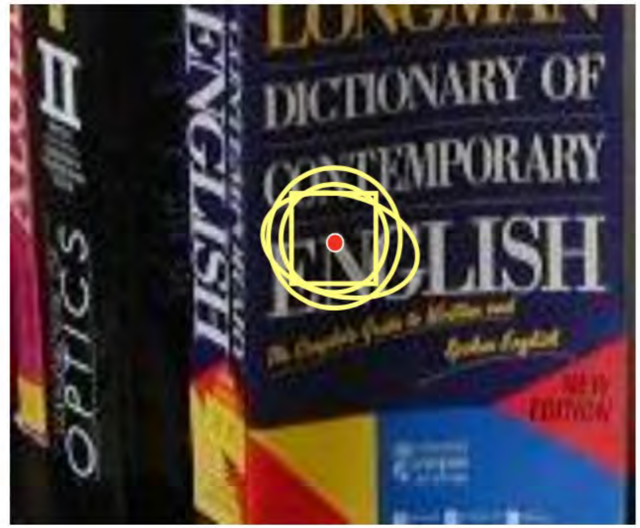
- Compute orientation histogram
- Select dominant orientation
- Normalize: rotate to fixed orientation

[Lowe, SIFT, 1999



Slide adapted from David Lowe

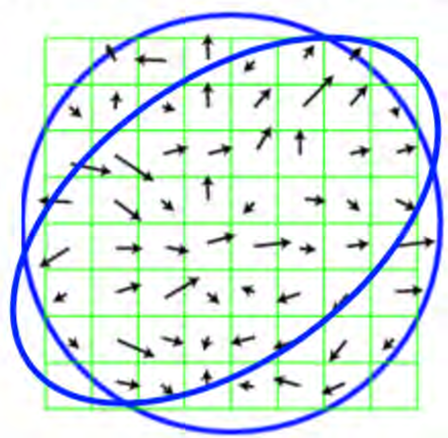
The Need for Invariance



- Up to now, we had invariance to
  - Translation
  - Scale
  - Rotation
- Not sufficient to match regions under viewpoint changes
  - For this, we need also affine adaptation

Affine Adaption

- **Problem:**
  - Determine the characteristic shape of the region.
  - Assumption: shape can be described by "local affine frame".
- **Solution: iterative approach**
  - Use a circular window to compute second moment matrix.
  - Compute eigenvectors to adapt the circle to an ellipse.
  - Recompute second moment matrix using new window and iterate…



Iterative Adaption

1. Detect keypoints, e.g. multi-scale Harris
2. Automatically select the scales
3. Adapt affine shape based on second order moment matrix
4. Refine point location

K. Mikolajczyk and C. Schmid, Scale and affine invariant interest point detectors, IJCV 60(1):63-86, 2004.

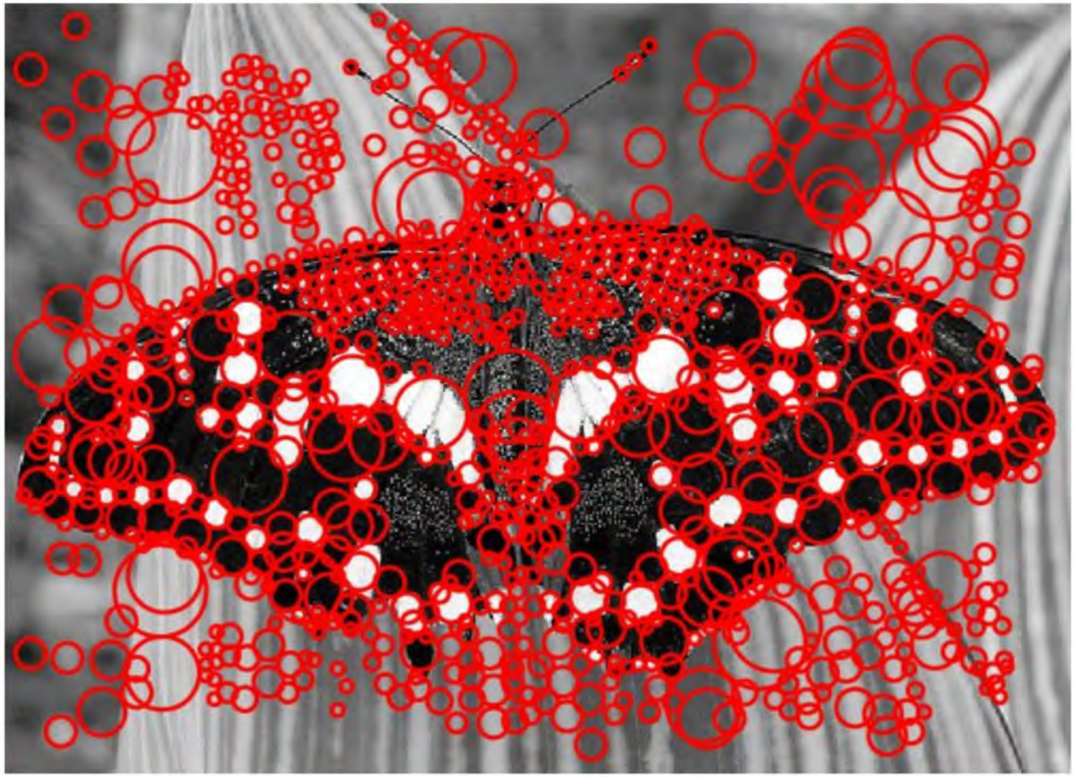
56

Slide credit: Tinne Tuytelaars

Affine Normalization

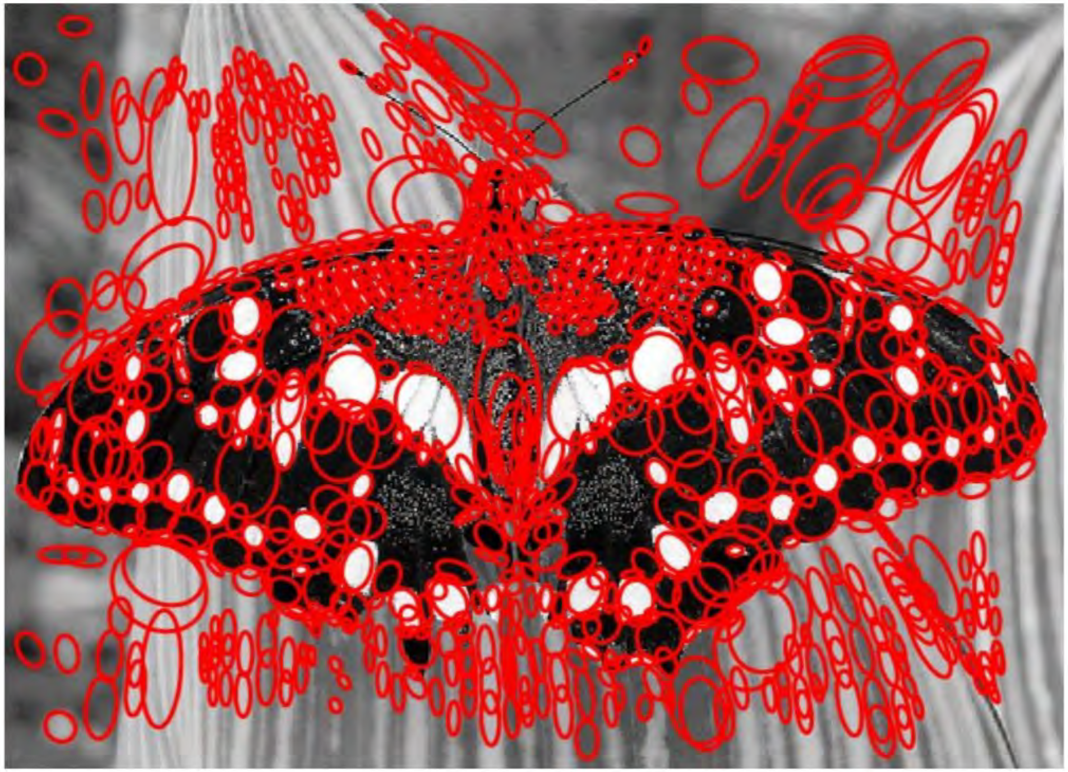
- **Steps**
  - Rotate the ellipse's main axis to horizontal
  - Scale the x axis, such that it forms a circle

Affine Adaption Example



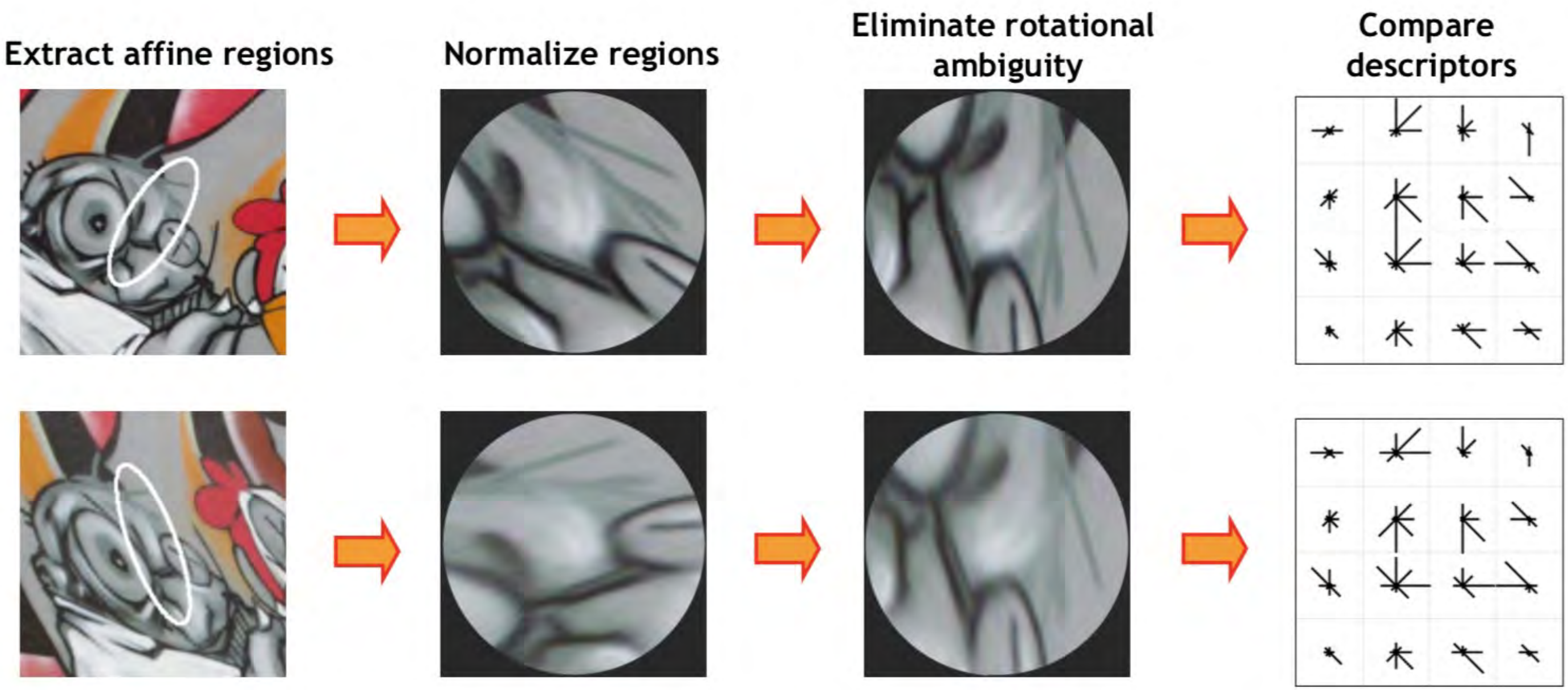
Scale-invariant regions (blobs)

Affine Adaption Example



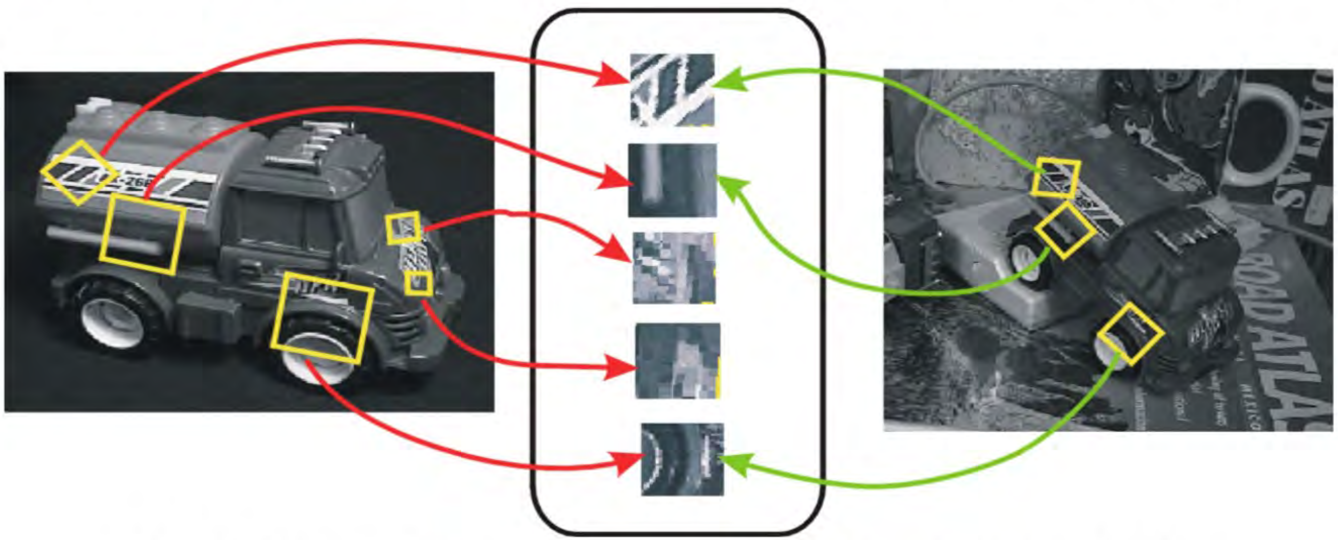
Affine-adapted blobs

Summary：Affine Invariance



Invariance vs. Covariance

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



Covariant detection ⇒ invariant description

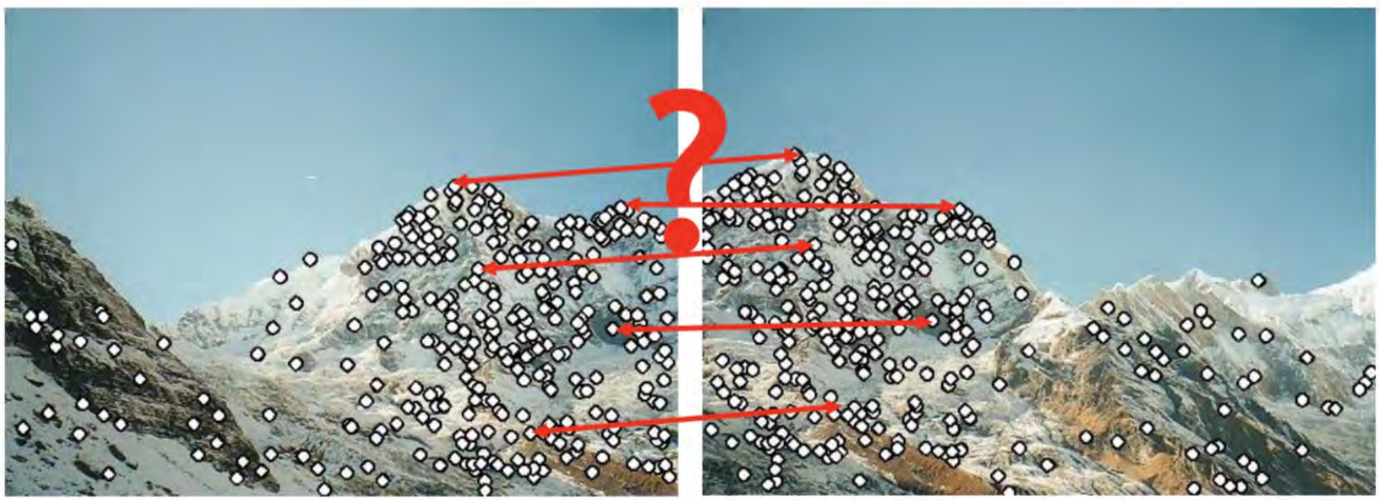
Today’s class

- Normalization
  - Orientation normalization
  - Affine invariant feature extraction
- **Local descriptor**
  - SIFT, SURF, GIST
- Binary descriptor
  - LBP, BRIEF
- CNN based descriptor
  - MatchNet, DeepCompare, DeepDesc, LIFT

Local Descriptor

- **We know how to detect points**
- **Next question:**

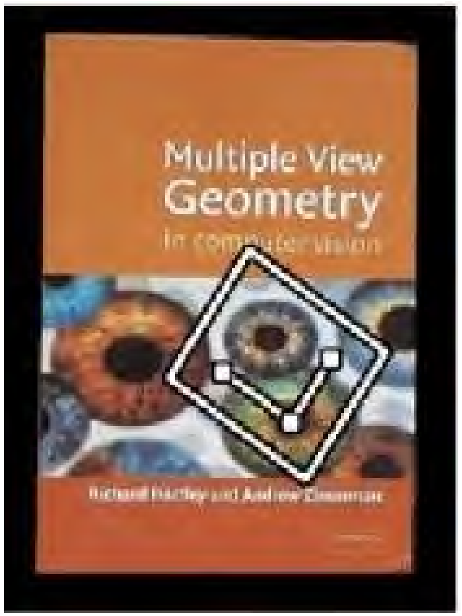
How to *describe* them for matching?



Point descriptor should be:

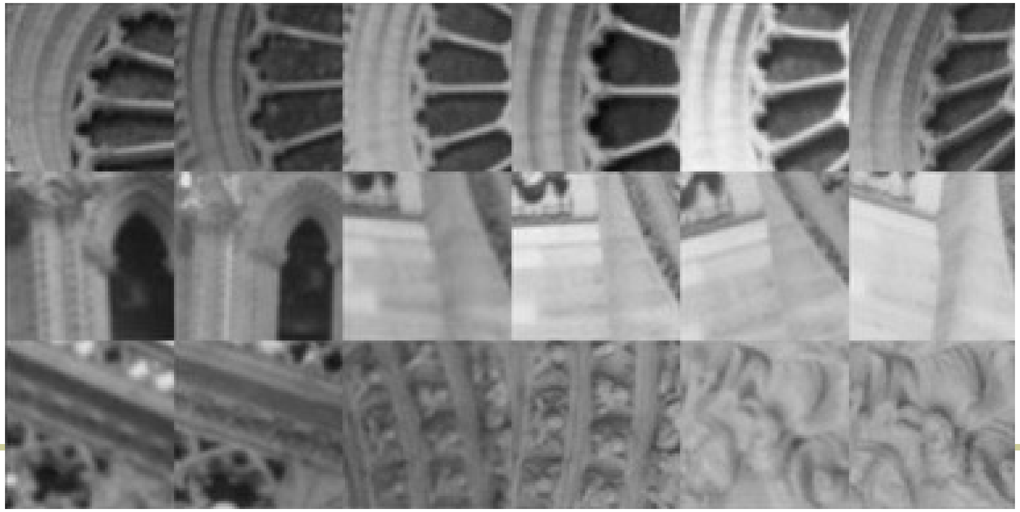
1. **Invariant**
2. **Distinctive**

Geometric transformations



Photometric transformations

What is the best descriptor for an image feature?



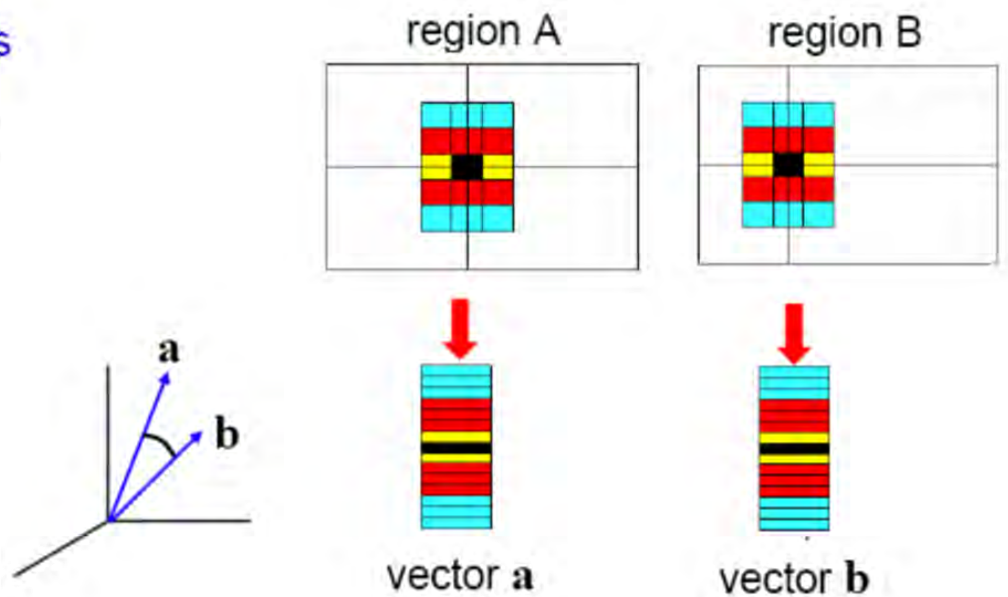
Local Descriptor

- Simplest descriptor: list of intensities within a patch.
- What is this going to be invariant to?

Perfectly fine if geometry and appearance is unchanged (a.k.a. template matching)

Write regions as vectors

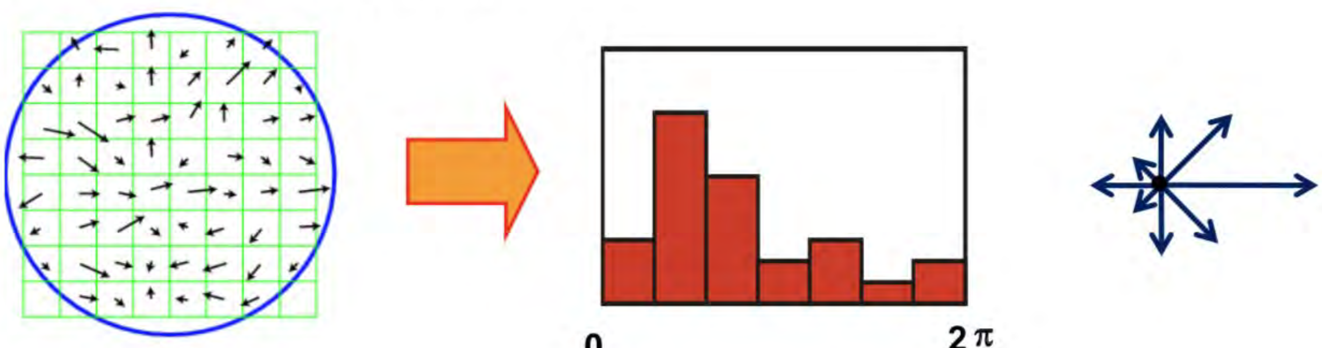
$$A \rightarrow \mathbf{a}, \ B \rightarrow \mathbf{b}$$



Feature Descriptor

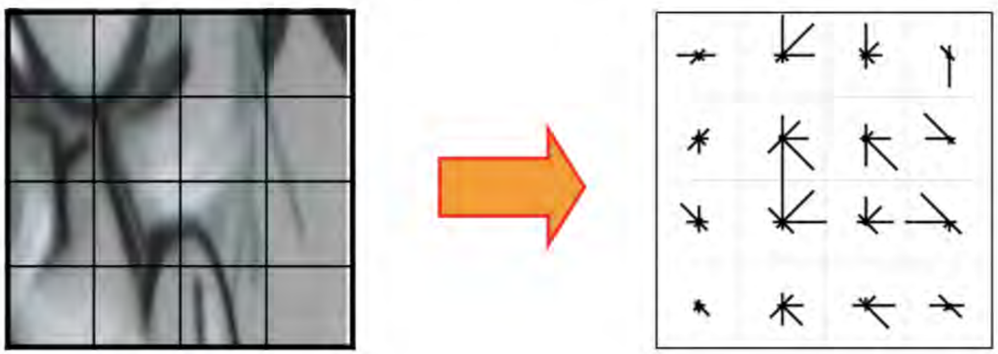
- **Disadvantage of patches as descriptors:**
  - Small shifts can affect matching score a lot

- **Solution: histograms**



Feature Descriptor：SIFT

- Scale Invariant Feature Transform
- Descriptor computation:
  - Divide patch into 4x4 sub-patches: 16 cells
  - Compute histogram of gradient orientations (8 reference angles) for all pixels inside each sub-patch
  - Resulting descriptor: 4x4x8 = 128 dimensions

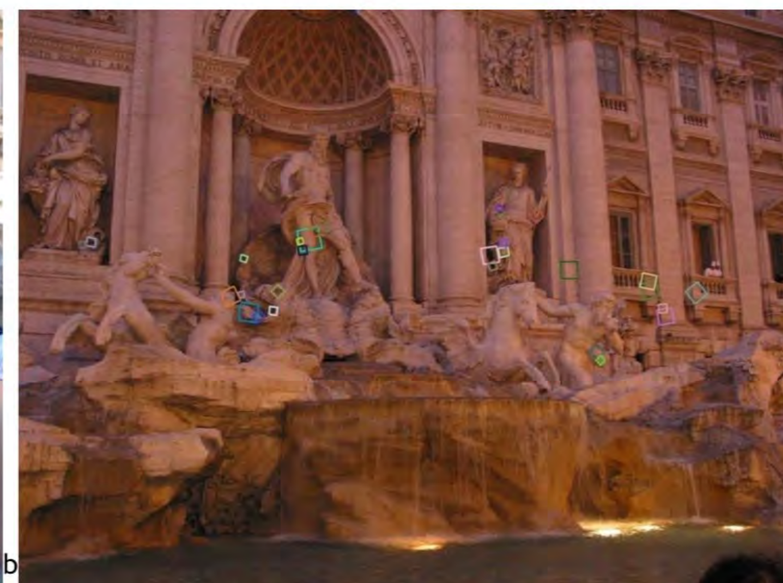


David G. Lowe. "Distinctive image features from scale-invariant keypoints." *IJCV* 60 (2), pp. 91-110, 2004.

SIFT Properties

- Extraordinarily robust matching technique
  - Can handle changes in viewpoint up to ~60 deg. out-of-plane rotation
  - Can handle significant changes in illumination
    - Sometimes even day vs. night (below)
  - Fast and efficient—can run in real time
  - Lots of code available
    - http://people.csail.mit.edu/albert/ladypack/wiki/index.php/Known_implementations_of_SIFT

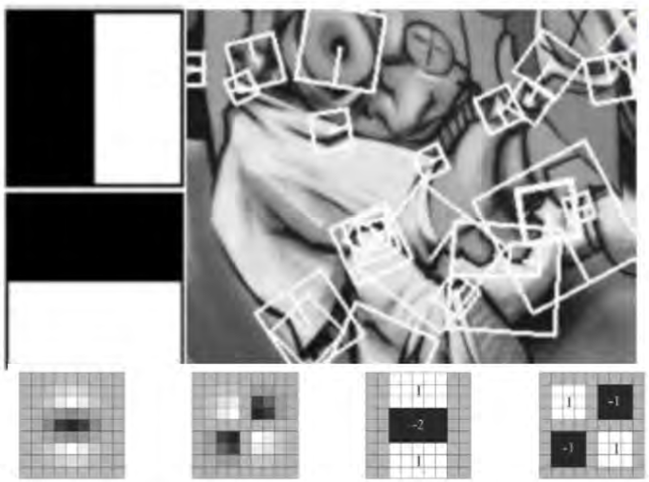
b



Summary：SIFT

- **One image yields:**
  - n **2D points giving positions of the patches**
    - **[n x 2 matrix]**
  - n **scale parameters specifying the size of each patch**
    - **[n x 1 vector]**
  - n **orientation parameters specifying the angle of the patch**
    - **[n x 1 vector]**
  - n **128-dimensional descriptors: each one is a histogram of the gradient orientations within a patch**
    - **[n x 128 matrix]**

Feature Descriptor: SURF



- **Fast approximation of SIFT idea**
  - Efficient computation by 2D box filters & integral images
    ⇒ 6 times faster than SIFT
  - Equivalent quality for object identification
  - http://www.vision.ee.ethz.ch/~surf

Well received!

- More than 8000 citations.
- CVIU Most Cited Paper
- Koenderink Prize of ECCV'16

GPU implementation available

- Feature extraction @ 100Hz (detector + descriptor, 640×480 img)
- http://homes.esat.kuleuven.be/~ncorneli/gpusurf/

Herbert Bay et al., SURF: Speeded Up Robust Feature, in ECCV 2006

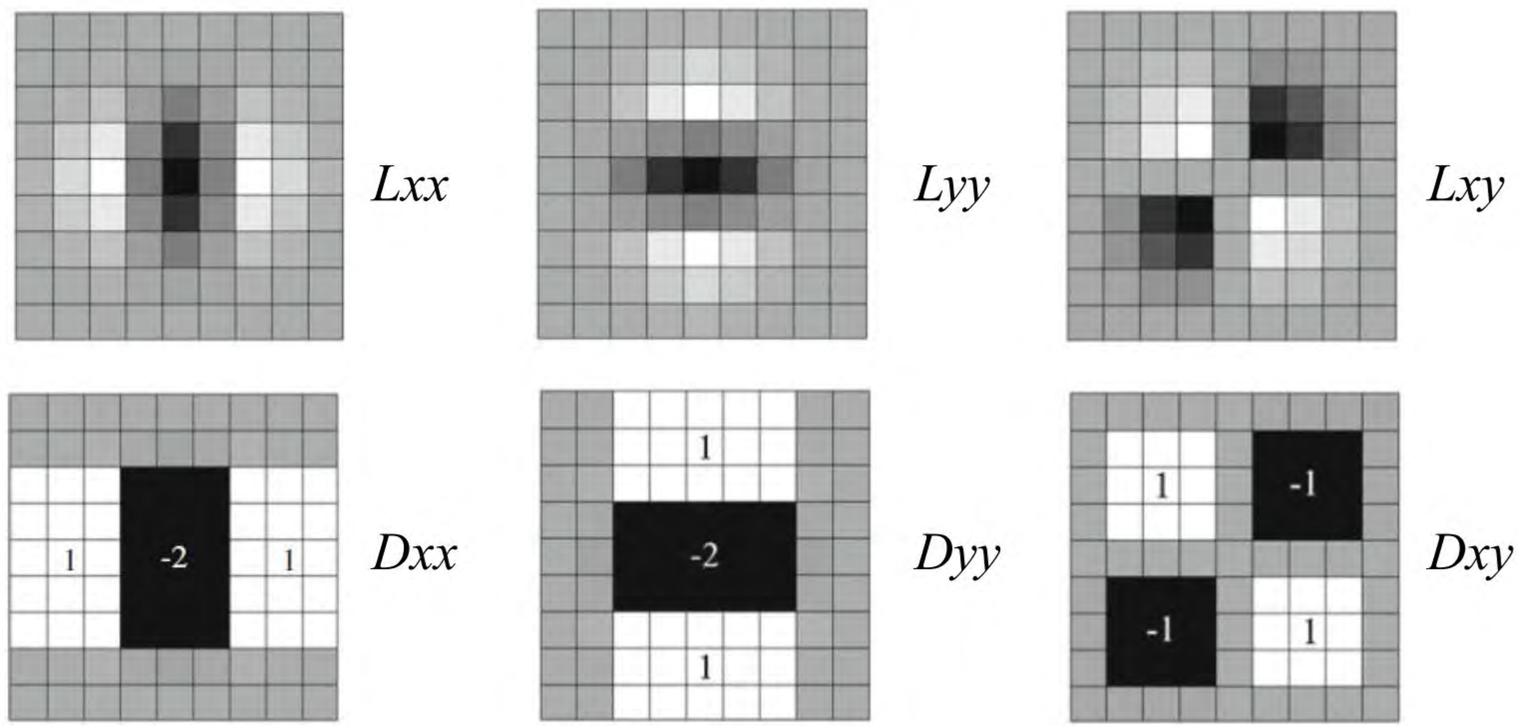
SURF: Keypoint Detection

- Uses determinant of Hessian matrix
- Approximate 2nd derivatives in Hessian matrix with box filters

$$H(x,\sigma) = \begin{bmatrix} L_{xx}(x,\sigma) & L_{xy}(x,\sigma) \\ L_{xy}(x,\sigma) & L_{yy}(x,\sigma) \end{bmatrix} \Rightarrow \hat{H}(x,\sigma) = \begin{bmatrix} D_{xx}(x,\sigma) & D_{xy}(x,\sigma) \\ D_{xy}(x,\sigma) & D_{yy}(x,\sigma) \end{bmatrix}$$

$$R(x,\sigma) = L_{xx}L_{yy} - L_{xy}^2 \approx D_{xx}D_{yy} - (0.9D_{xy}^2)$$

SURF: Keypoint Detection



Integral Image

$I(x, y)$

original image

1	5	2
2	4	1
2	1	1

$A(x, y)$

integral image

1	6	8
3	12	15
5	15	19

$$A(x, y) = \sum_{x' \le x, y' \le y} I(x', y')$$

Integral Image

$I(x,y)$

original image

1	5	2
2	4	1
2	1	1

$A(x,y)$

integral image

1	6	8
3	12	15
5	15	19

$$A(x,y) = \sum_{x' \le x, y' \le y} I(x', y')$$

Can find the **sum** of any block using **3** operations

$$A(x_1, y_1, x_2, y_2) = A(x_2, y_2) - A(x_1, y_2) - A(x_2, y_1) + A(x_1, y_1)$$

What is the sum of the bottom right 2x2 square?

$$A(x_1, y_1, x_2, y_2) = A(x_2, y_2) - A(x_1, y_2) - A(x_2, y_1) + A(x_1, y_1)$$

$I(x, y)$

1	5	2
2	4	1
2	1	1

image

$A(x, y)$

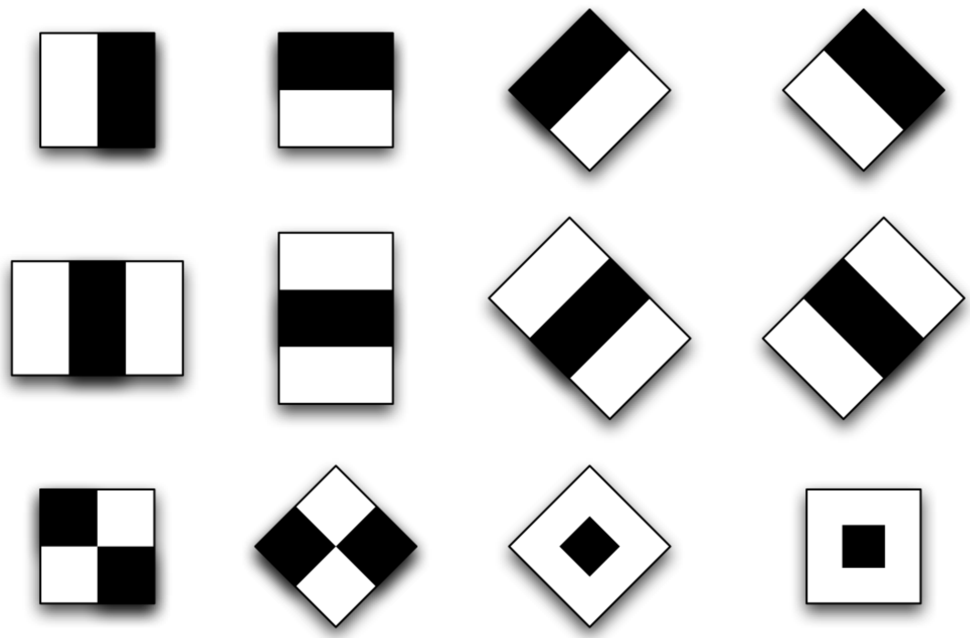
1	6	**8**
3	12	15
5	15	**19**

integral image

$$\begin{aligned} A(1,1,3,3) &= A(3,3) - A(1,3) - A(3,1) + A(1,1) \\ &= 19 - 8 - 5 + 1 \\ &= 7 \end{aligned}$$

Haar Wavelets (actually, Haar-like features)

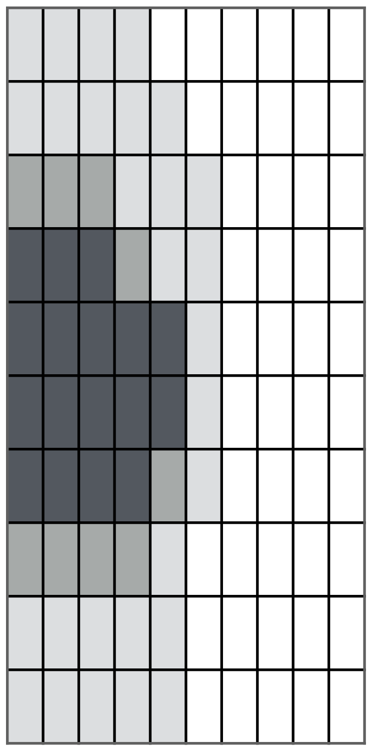
Use responses of a bank of filters as a descriptor



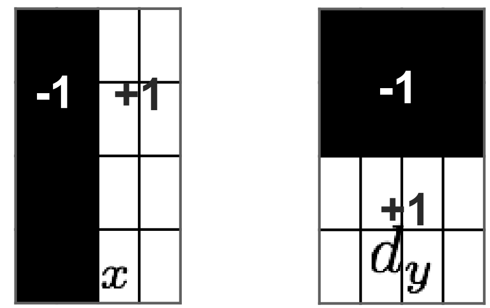
How to compute Haar wavelet responses **efficiently** (in constant time) with integral images

Haar wavelet responses can be computed with filtering

image patch

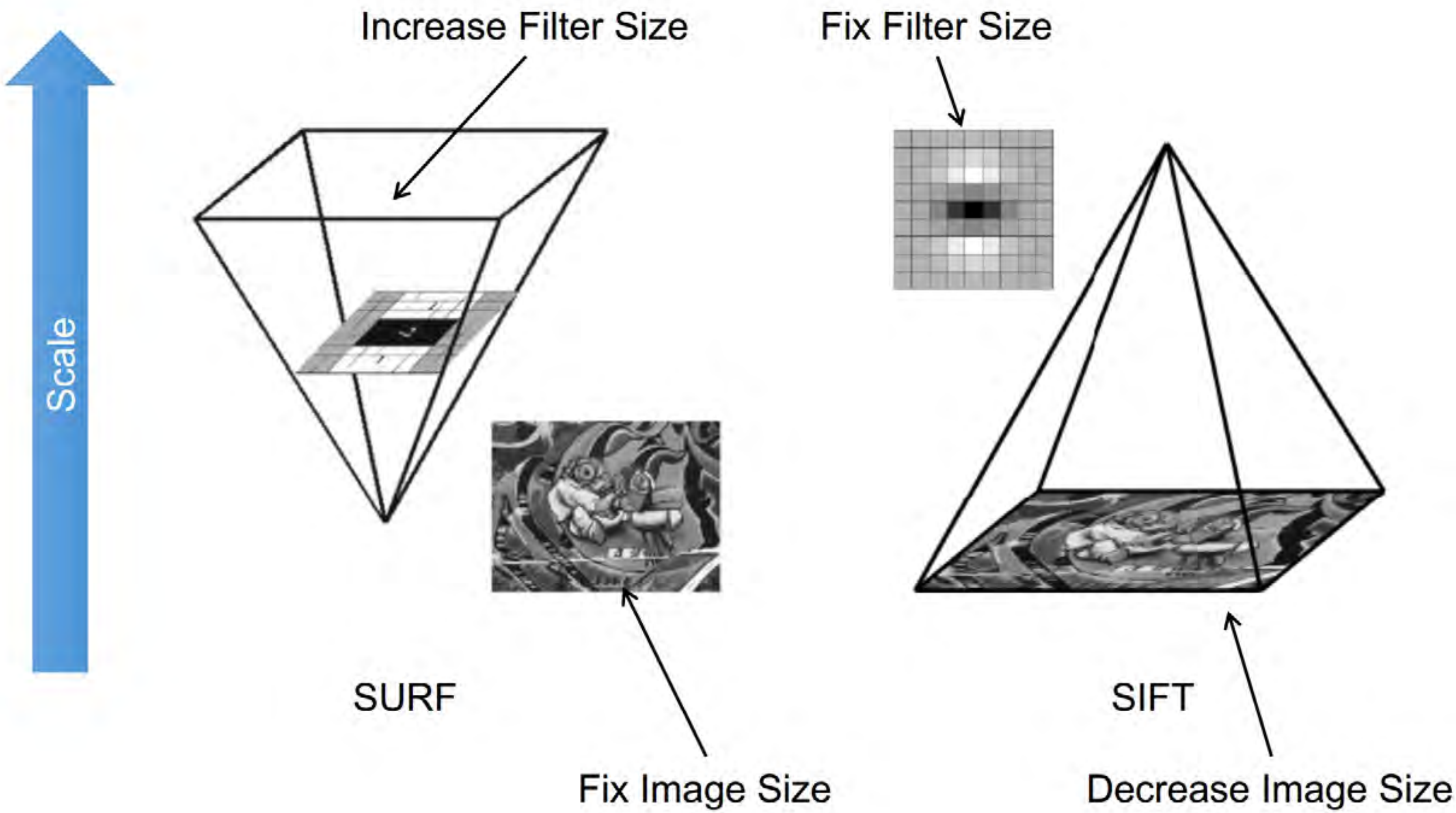


Haar wavelets filters

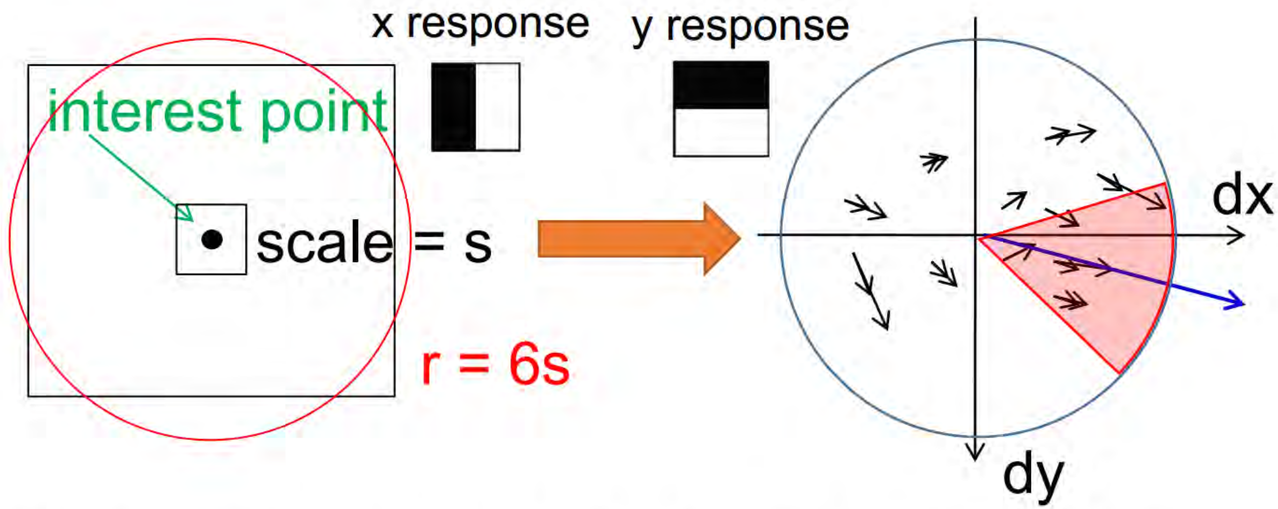


Haar wavelet responses can be computed **efficiently** (in constant time) with integral images

SURF vs. SIFT: Scale Space

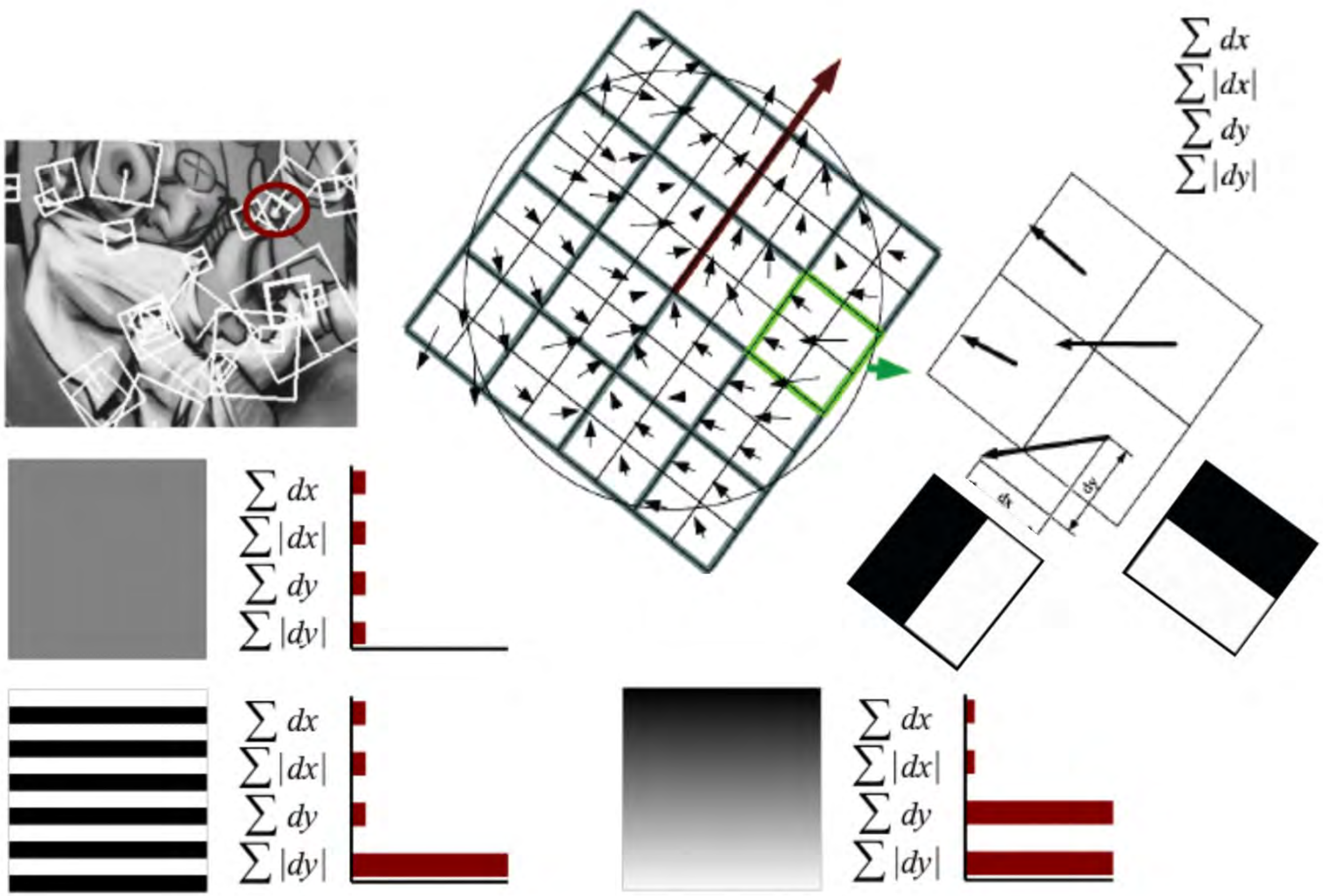


Dominant Orientation Estimation



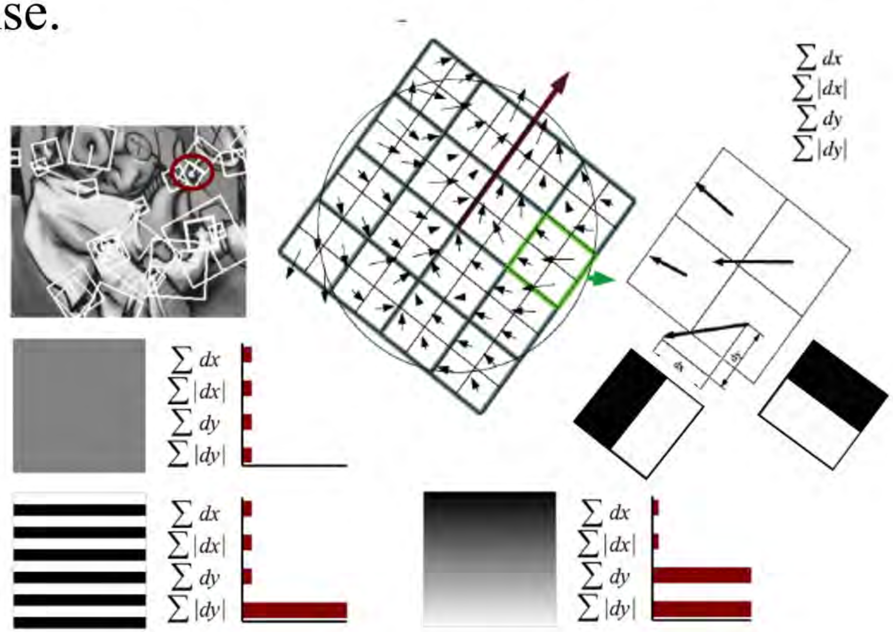
- The Haar wavelet responses (x and y) are represented as vectors.
- Sum all responses within a sliding orientation window covering an angle of 60 degree.
- The longest vector is the dominant orientation

SURF: Descriptor Extraction



SURF: Descriptor Extraction

1. Split the interest region (20s x 20s) into 4 x 4 square sub-regions.
2. Calculate Haar wavelet responses dx and dy, and weight the responses with a Gaussian kernel.
3. Sum the response over each sub-region for dx and dy, then sum the absolute value of response.
4. Concatenate summation results in all sub-regions, forming a 64D SURF descriptor.

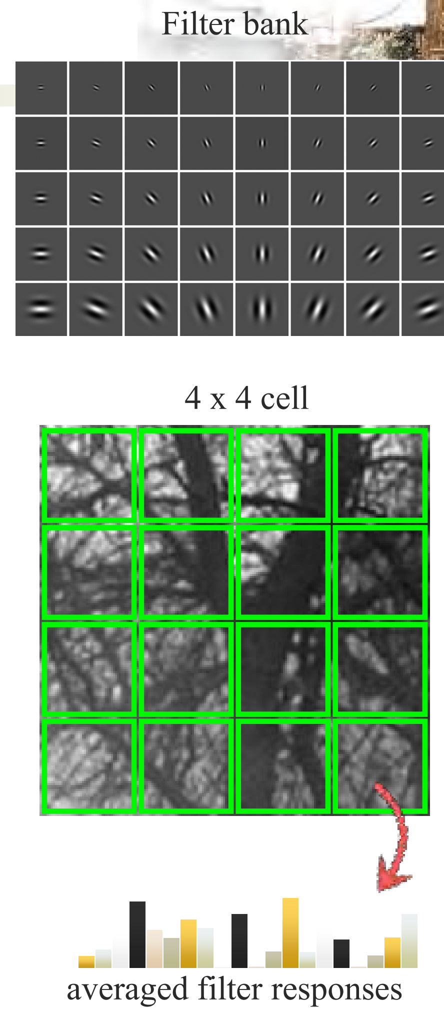


Summary: SURF

- Approximation yet can be computed much faster.
  - relying on integral images for image convolutions
  - building on the strengths of the leading existing detectors and descriptors
  - simplifying these methods to the essential
- Combination of novel detection, description, and matching steps.

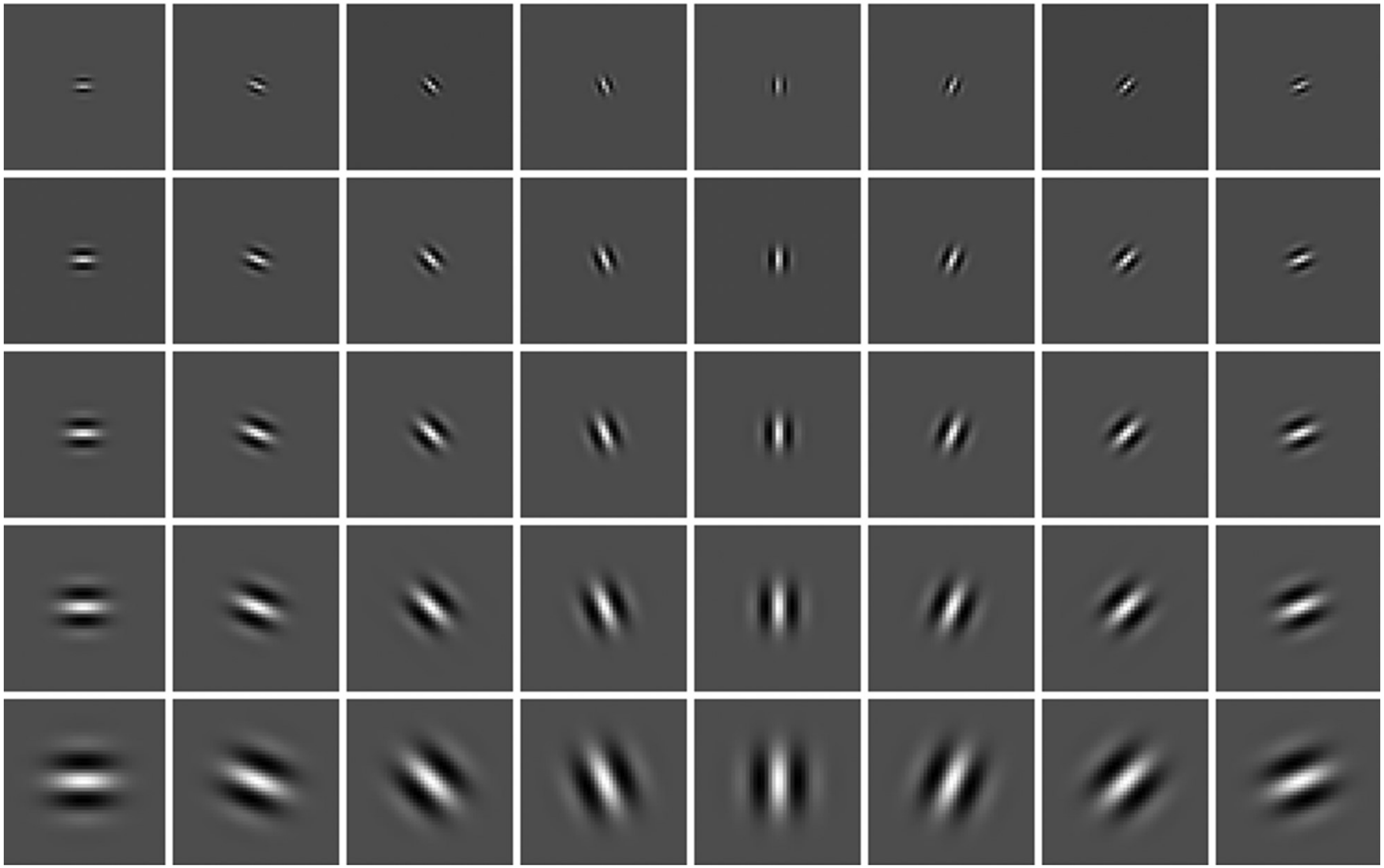
GIST

1. Compute filter responses (filter bank of Gabor filters)
2. Divide image patch into 4 x 4 cells
3. Compute filter response averages for each cell
4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank



Oliva, A., & Torralba, A., Modeling the Shape of the Scene: a Holistic Representation of the Spatial Envelope. IJCV 2001.

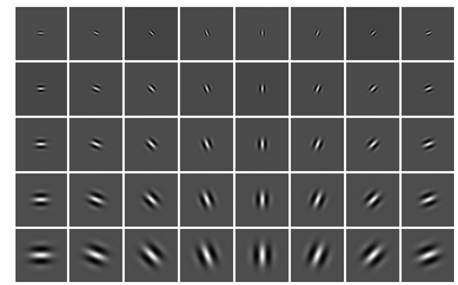
Directional edge detectors



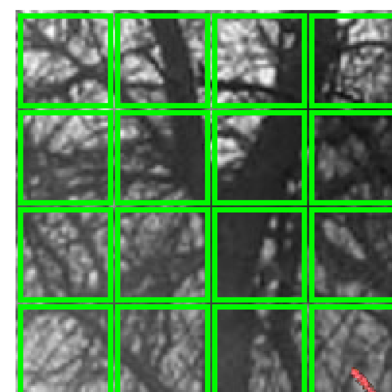
Summary: GIST

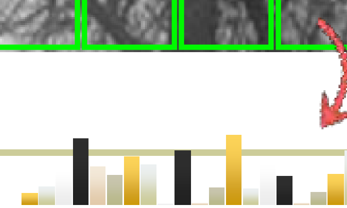
1. Compute filter responses (filter bank of Gabor filters)
2. Divide image patch into 4 x 4 cells
3. Compute filter response averages for each cell
4. Size of descriptor is 4 x 4 x N, where N is the size of the filter bank

Filter bank



4 x 4 cell





averaged filter responses

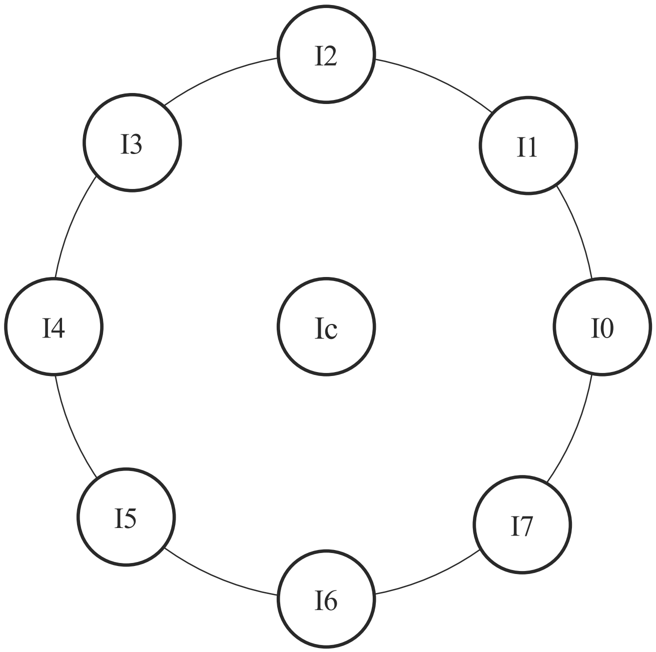
What is the GIST descriptor encoding?

Rough spatial distribution of image gradients

Today's class

- Normalization
  - Orientation normalization
  - Affine invariant feature extraction
- Local Descriptor
  - SIFT, SURF, GIST
- **Binary descriptor**
  - LBP, BRIEF
- CNN Based descriptor
  - MatchNet, DeepCompare, DeepDesc, LIFT

Center Symmetric Local Binary Patterns



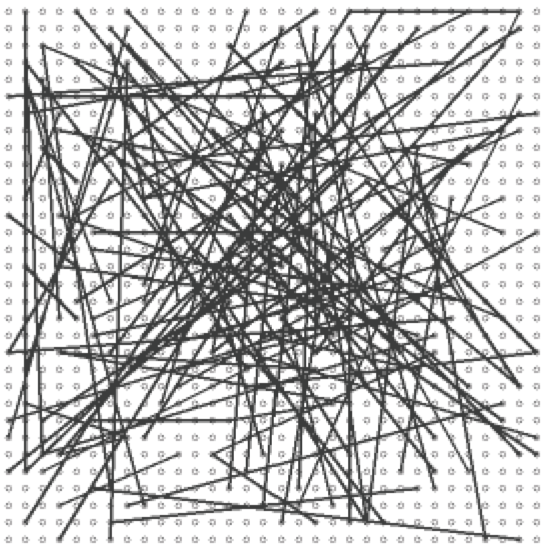
CS-LBP=
$\text{sign}(I0-I4-t)2^0 +$
$\text{sign}(I1-I5-t)2^1 +$
$\text{sign}(I2-I6-t)2^2 +$
$\text{sign}(I3-I7-t)2^3$

LBP=
$\text{sign}(I0-Ic)2^0 +$
$\text{sign}(I1-Ic)2^1 +$
$\text{sign}(I2-Ic)2^2 +$
$\text{sign}(I3-Ic)2^3 +$
$\text{sign}(I4-Ic)2^4 +$
$\text{sign}(I5-Ic)2^5 +$
$\text{sign}(I6-Ic)2^6 +$
$\text{sign}(I7-Ic)2^7$

Robustness to illumination.

CSLBP [Heikkilä '09]

Binary Robust Independent Elementary Features



Descriptor construction: intensity test between given point pairs：

$$f_n(P) = \left[\tau(P; x_1, y_1), \cdots, \tau(P; x_n, y_n)\right] \in \{0,1\}^n$$

$$\tau(P; x, y) = \begin{cases} 1, P(x) > P(y) \\ 0, P(x) \le P(y) \end{cases}$$

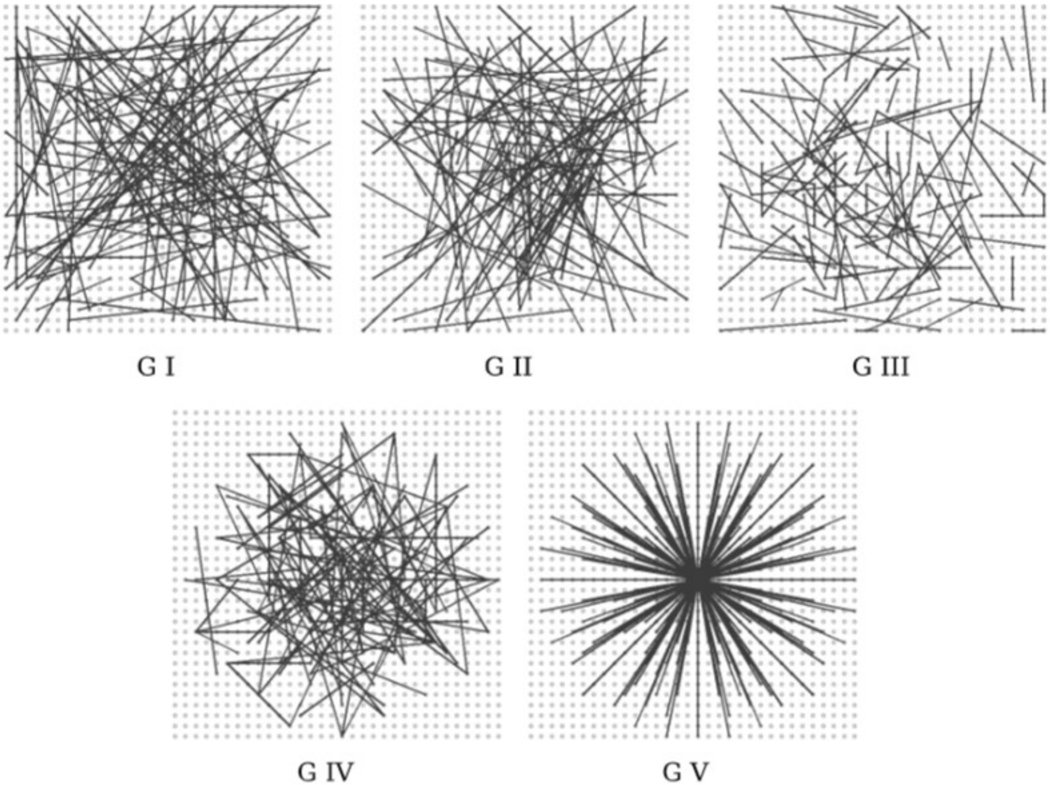
Point pairs：given by random sampling

$$x \sim G(0, \frac{S^2}{25}), y \sim G(0, \frac{S^2}{25})$$

- Simple, fast, moderate performance
- First binary descriptor for patch matching

BRIEF [Calonder ’10]

Binary Robust Independent Elementary Features



Binary Robust Independent Elementary Features

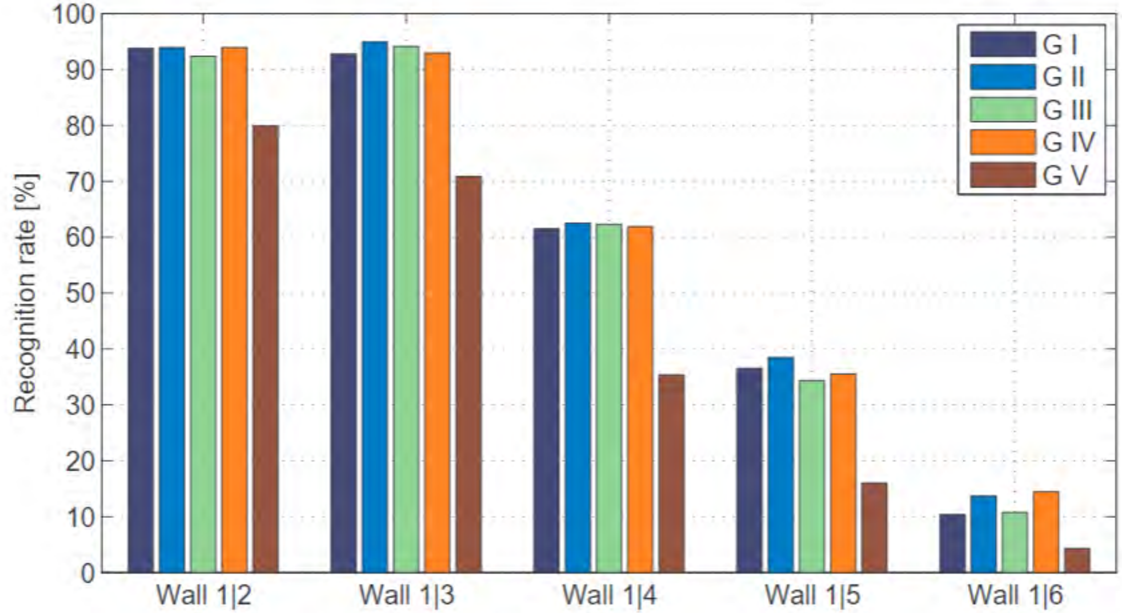
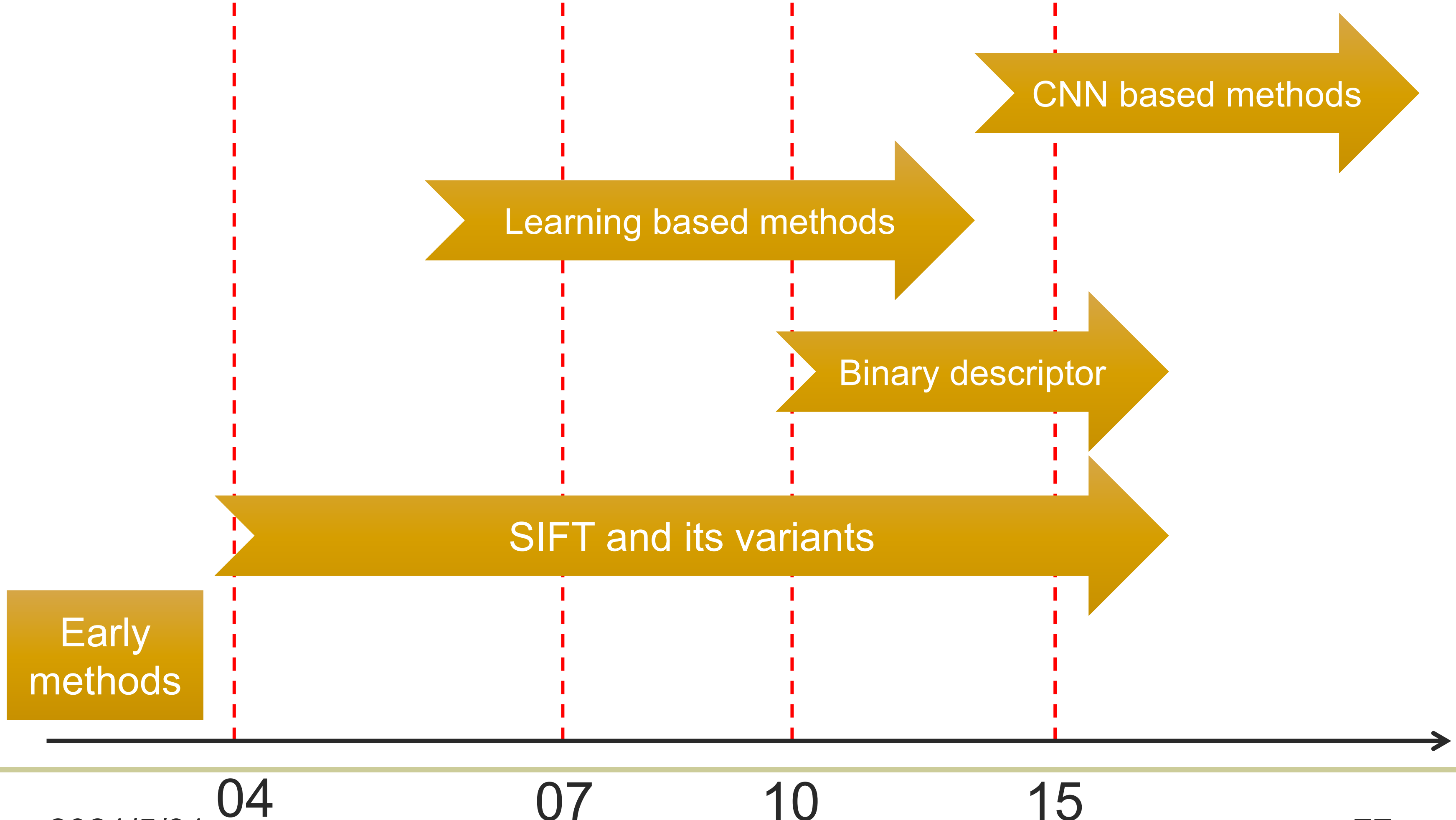


Fig. 3. Recognition rate for the five different test geometries introduced in section 3.2.

Today's class

- Normalization
  - Orientation normalization
  - Affine invariant feature extraction
- Local descriptor
  - SIFT, SURF, GIST
- Binary descriptor
  - LBP, BRIEF
- **CNN Based descriptor**
  - MatchNet, DeepCompare, DeepDesc, LIFT

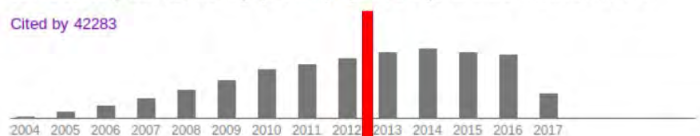
Local Descriptors: Trend



A Deep Casualty?

Distinctive image features from scale-invariant keypoints

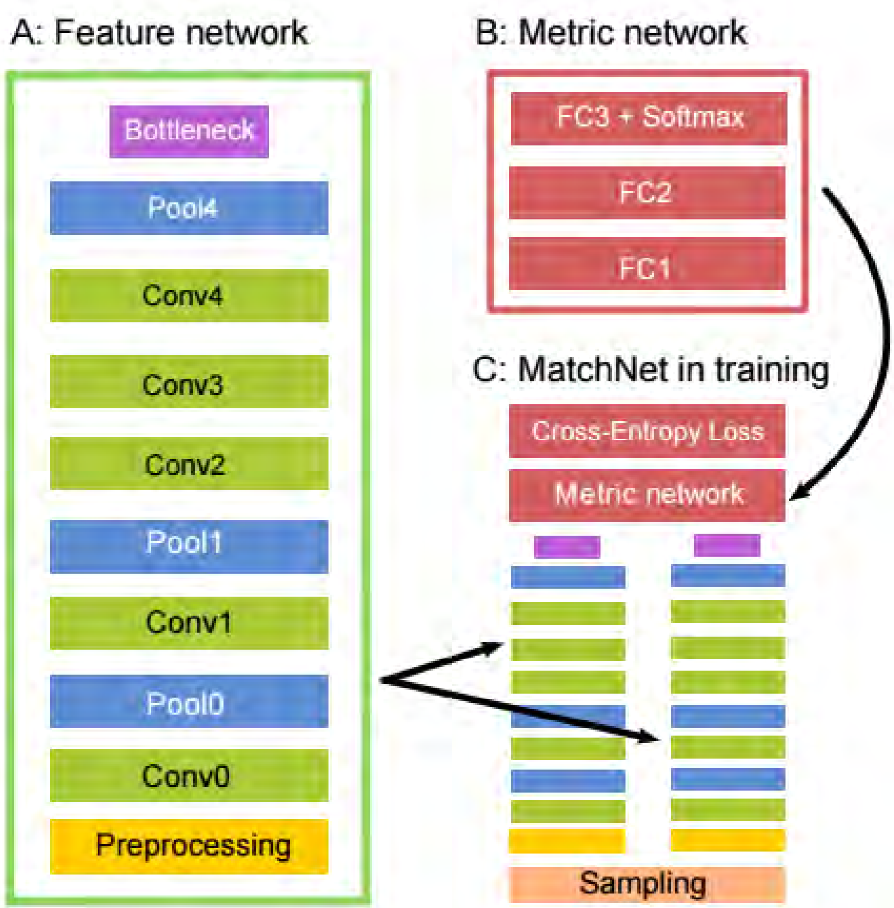
Authors	David G Lowe
Publication date	2004/11/1
Journal	International journal of computer vision
Volume	60
Issue	2
Pages	91-110
Publisher	Springer Netherlands
Description	This paper presents a method for extracting distinctive invariant features from images that can be used to perform reliable matching between different views of an object or scene. The features are invariant to image scale and rotation, and are shown to provide robust matching across a substantial range of affine distortion, change in 3D viewpoint, addition of noise, and change in illumination. The features are highly distinctive, in the sense that a single feature can be correctly matched with high probability against a large database of features from ...
Total citations	Cited by 42283



Scholar articles: Distinctive image features from scale-invariant keypoints
DG Lowe - International journal of computer vision, 2004
Cited by 42283 - Related articles - All 196 versions

- The SIFT paper is the most cited computer vision paper **ever**.
- But it’s not as dominant as it once was.

MatchNet



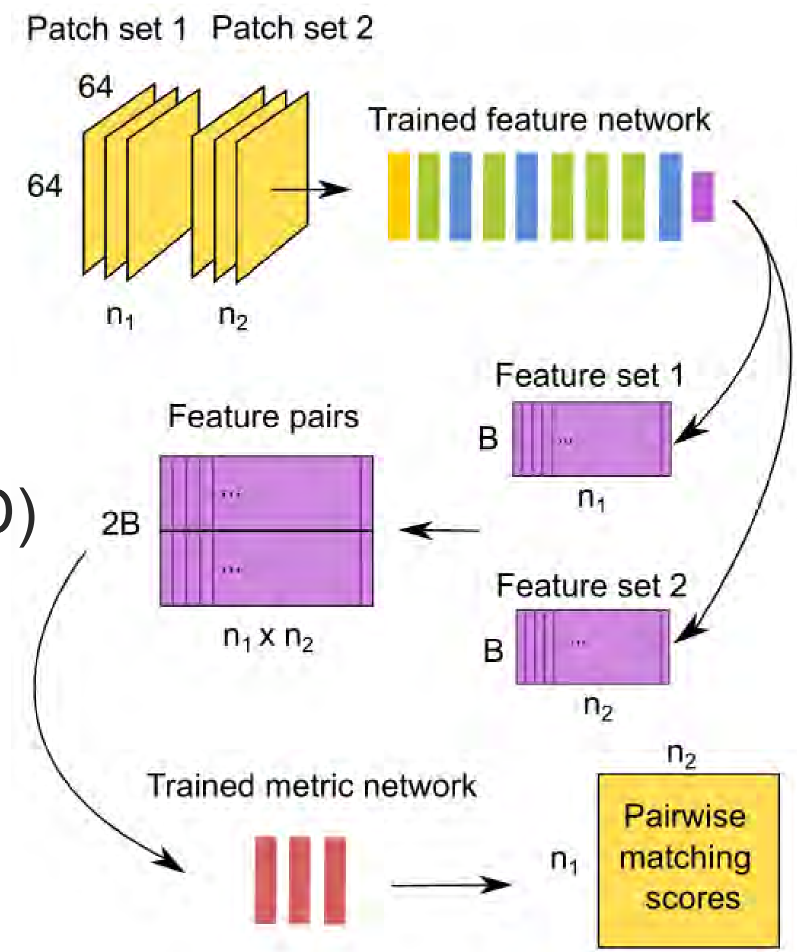
- Simultaneously learn the descriptor and the metric
- Siamese Feature descriptor network
- Metric network on top
- Cross-entropy loss, transfer matching problem to classification problem
- Train time：1 day – 1 week

Training MatchNet

- Cross-entropy error

$$E = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right]$$

- Stochastic gradient descent (SGD)
- A special reservoir sampler for negative sampling

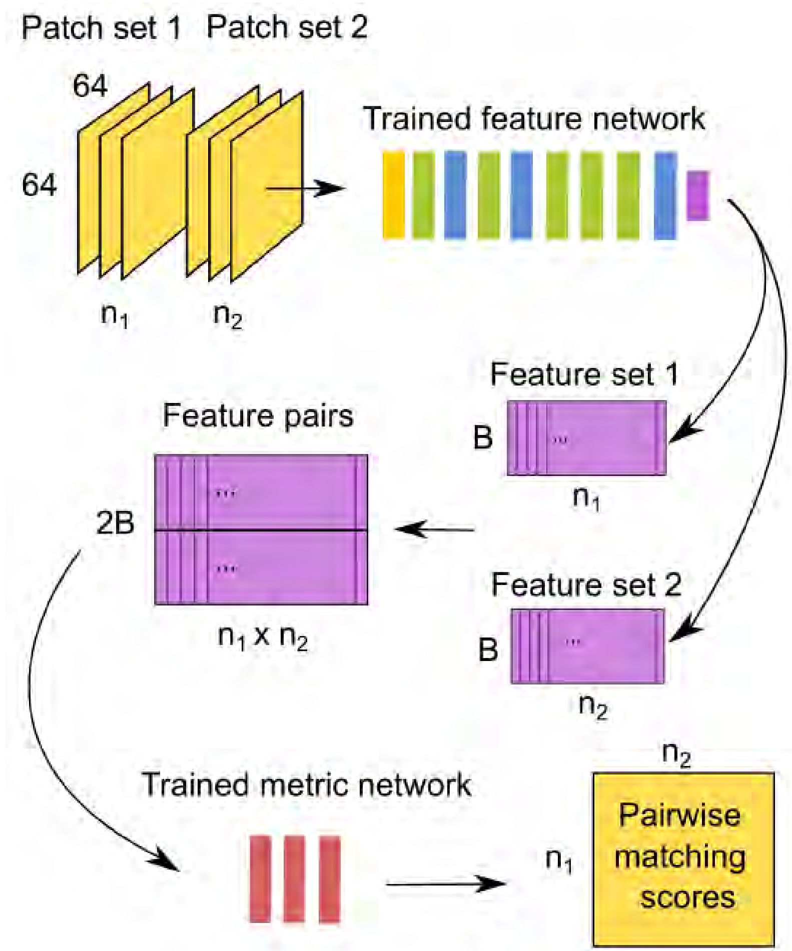


Testing MatchNet

A two-stage prediction pipeline:

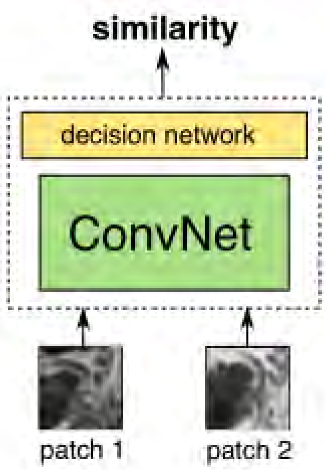
1. Generate feature descriptors for all patches.

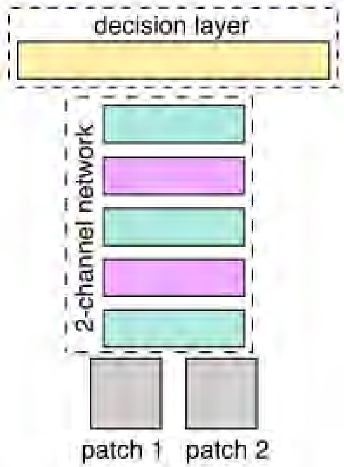
2. Pair the features and push them through the metric network to get the scores.



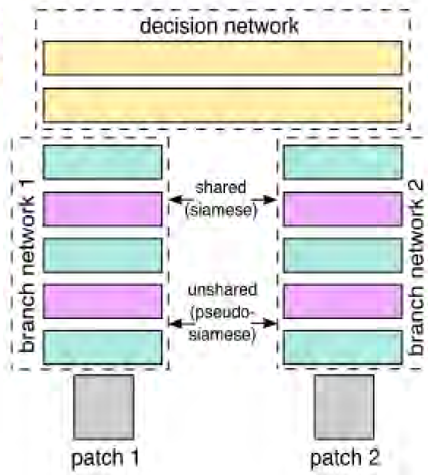
S. Zagoruyko et al. Learning to compare image patches via convolutional neural networks. In CVPR 2015.

DeepCompare

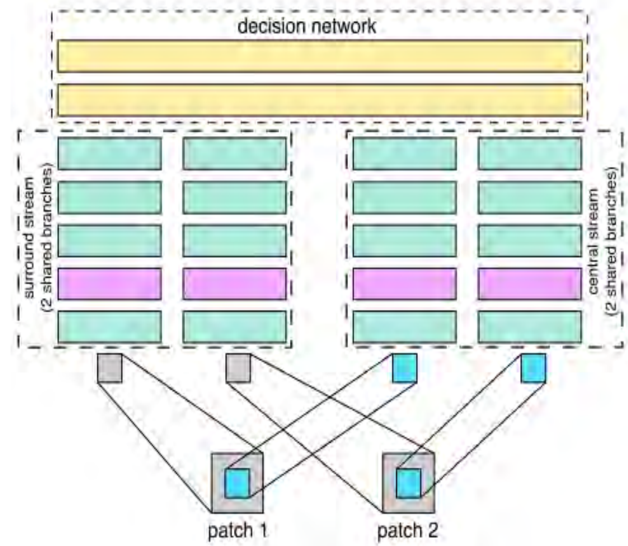




2 - channel



Siamese

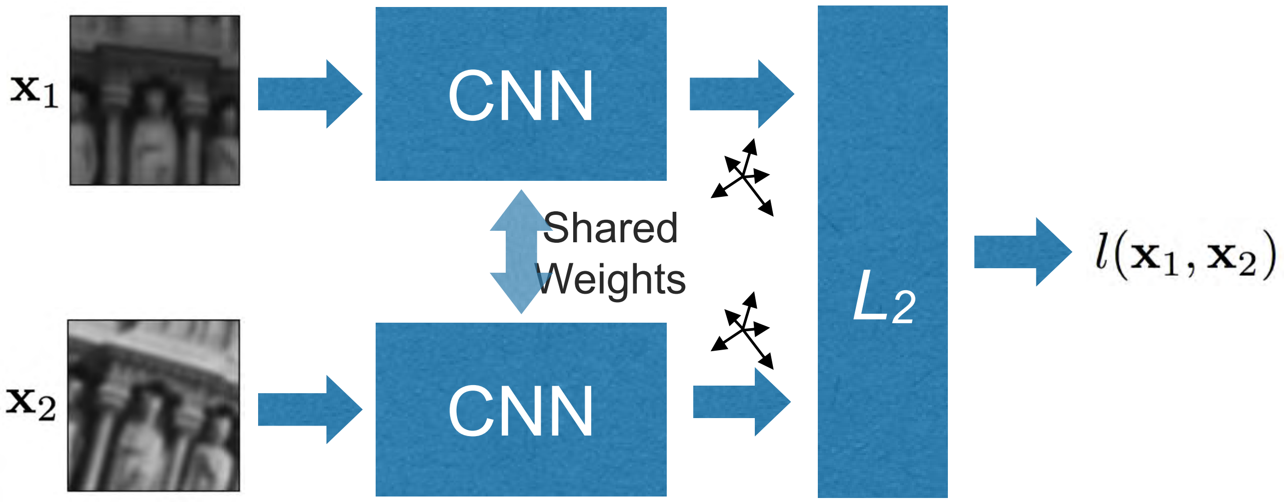


2 - channel 2-stream

DeepCompare

- Architecture
  - 2-channel structure
    - No direct notion of descriptor in the 2-channel architecture. It simply considers the two patches of an input pair as a 2-channel image, which is directly fed to the first convolutional layer of the network.
  - Central-surround two-stream network
    - Consists of two separate streams, central and surround, allowing the network to process at two different resolutions.
- Drawback
  - Pair-wise operation, can not re-use descriptor of each patch

DeepDesc

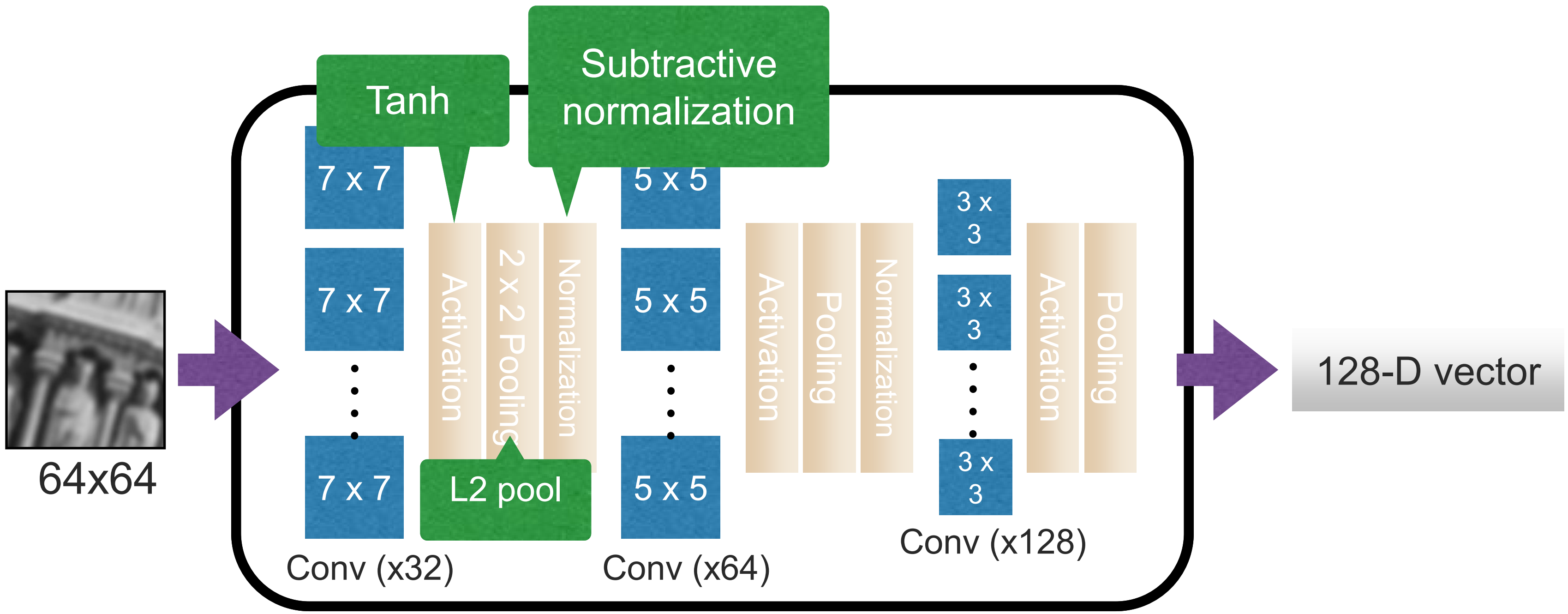


- Use Euclidean distance， direct substitution of SIFT
- loss： minimize pairwise hinge loss

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

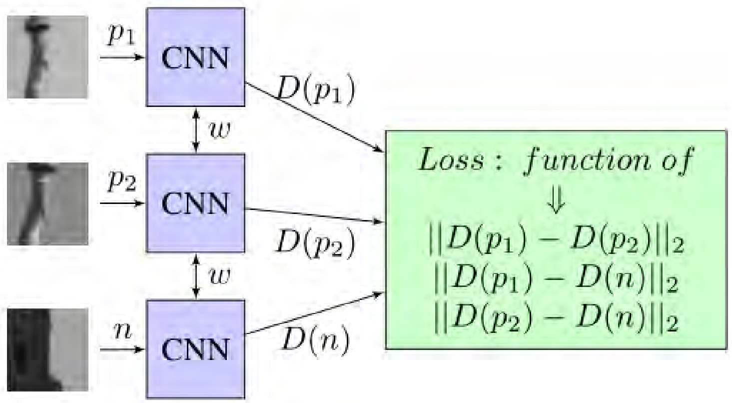
Edgar Simo-Serra et al. Discriminative learning of deep convolutional feature point descriptors. In ICCV, 2015.

DeepDesc Network Architecture



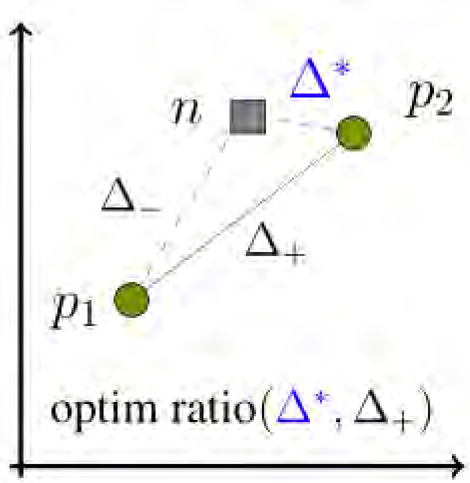
- Only 3 convolutional layers, simple.
- Use hard negative mining to alleviate the problem of imbalanced positive and negative samples, key to good performance.

PN-Net, TFeat



CNN Structure

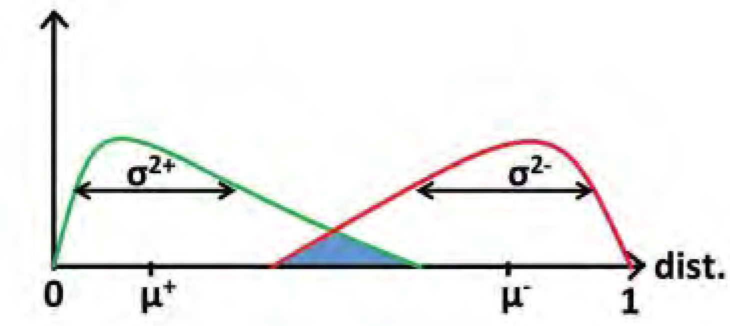
Layer #	Description
1	Spatial Convolution(7,7) $\rightarrow$ 32
2	Tanh
2	MaxPooling(2,2)
3	Spatial Convolution(6,6) $\rightarrow$ 64
4	Tanh
5	Linear $\rightarrow \{128, 256\}$
6	Tanh



Triplet Network: Smallest negative distance within the triplet should be larger than the positive distance.

V. Kumar B G et al. Learning local image descriptors with deep siamese and triplet convolutional networks by minimising global loss functions. In CVPR 2016

GLoss Net



Objective: Reduce the proportion of false positive and false negative, i.e., blue shaded area. (A global loss)

- **Global Loss**
  - Minimize the variance of the two distributions and the mean value of the distances between matching pairs.
  - Maximize the mean value of the distances between non-matching pairs.
- **Four models**
  - Metric learning: SNet-GLoss, CS SNet-GLoss (with Siamese Network)
  - L2 norm: TNet-TGLoss, TNet-TLoss (with Triplet Network)

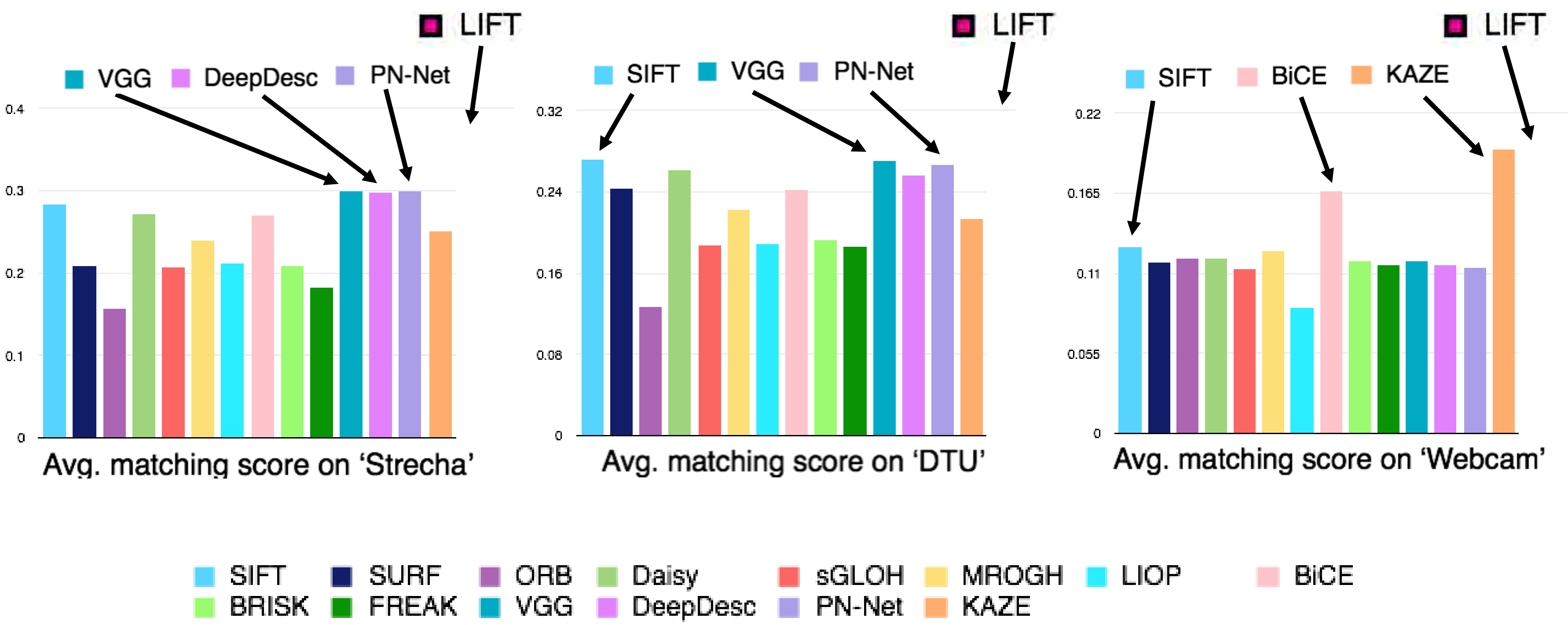
Performance Comparison of these CNN based Methods

Training	Metric	Feature	Notredame	Yosemite	Liberty	Yosemite	Liberty	Notredame
Test	Learning	Dim	Liberty		Notredame		Yosemite	
Float Descriptors								
SIFT		128	29.84		22.53		27.29	
MatchNet	Yes	4096	6.9	10.77	3.87	5.67	10.88	8.39
DeepCompare 2ch-2stream	Yes	256	4.85	7.20	1.90	2.11	5.00	4.10
DeepCompare 2ch-deep	Yes	256	4.55	7.40	2.01	2.52	4.75	4.38
SNet-GLoss	Yes	384	6.39	8.43	1.84	2.83	6.61	5.57
CS SNet-GLoss	Yes	384	**3.69**	**4.91**	**0.77**	**1.14**	**3.09**	**2.67**
TNet-TGLoss	No	256	9.91	13.45	3.91	5.43	10.65	9.47
TNet-TLoss	No	256	10.77	13.90	4.47	5.58	11.82	10.96
PN-Net	No	256	8.13	9.65	3.71	4.23	8.99	7.21
DeepDesc	No	128	10.9		4.40		5.69	

- Note: these results are on the Brown dataset.

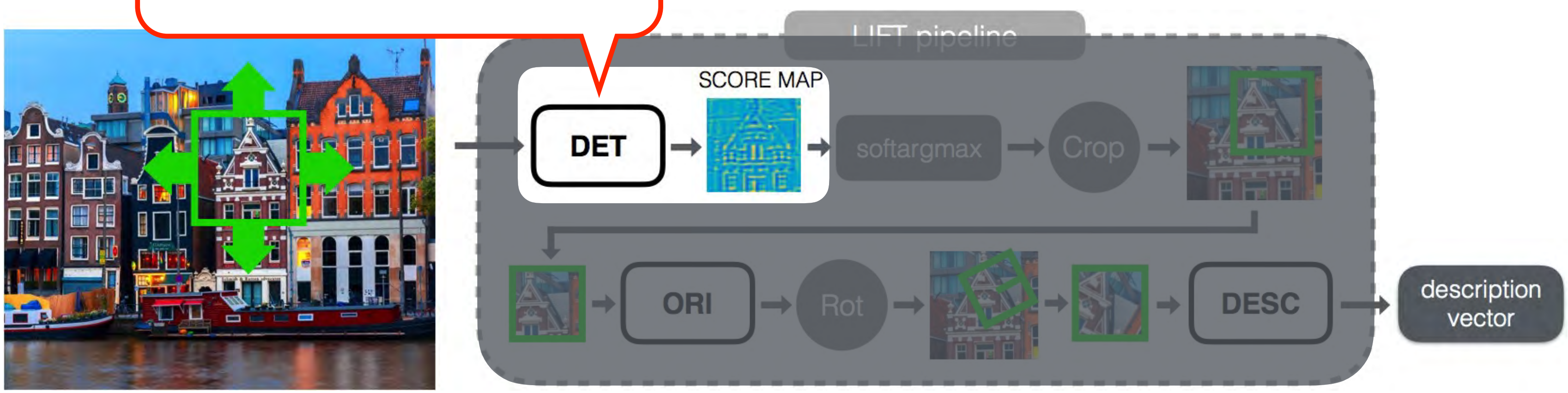
Moreover…

- Descriptors are affected by keypoints & orientations



The LIFT Network

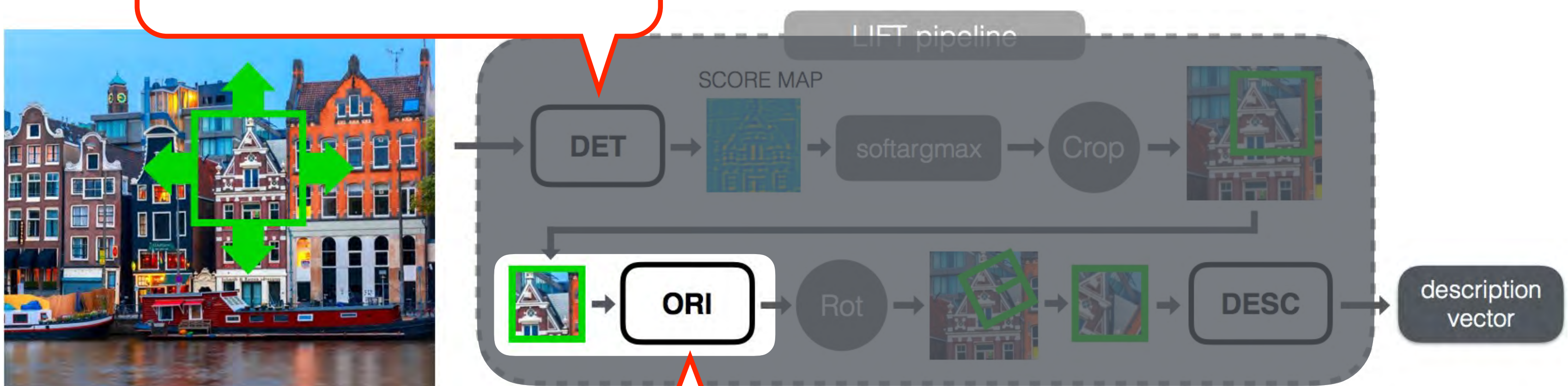
Y. Verdie, K.M. Yi, P. Fua, V. Lepetit: "TILDE: A Temporally Invariant Learned DEtector", CVPR 2015.



Kwang Moo Yi et al. LIFT: Learned invariant feature transform. In ECCV 2016.

The LIFT Network

Y. Verdie, K.M. Yi, P. Fua, V. Lepetit: "TILDE: A Temporally Invariant Learned DEtector", CVPR 2015.

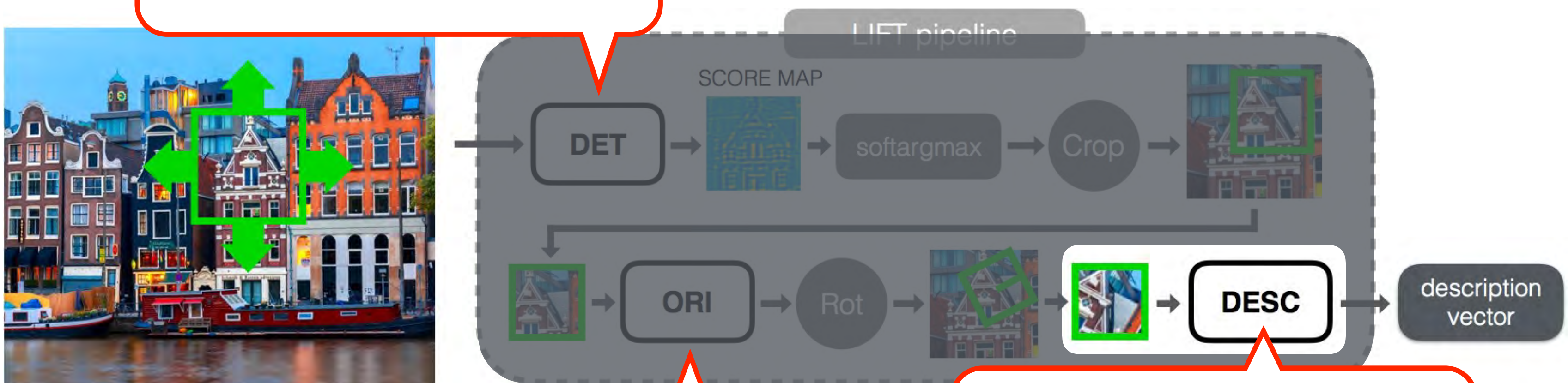


K.M. Yi, Y. Verdie, V. Lepetit, P. Fua : "Learning to Assign Orientations to Feature Points", CVPR 2016.

Kwang Moo Yi et al. LIFT: Learned invariant feature transform. In ECCV 2016.

The LIFT Network

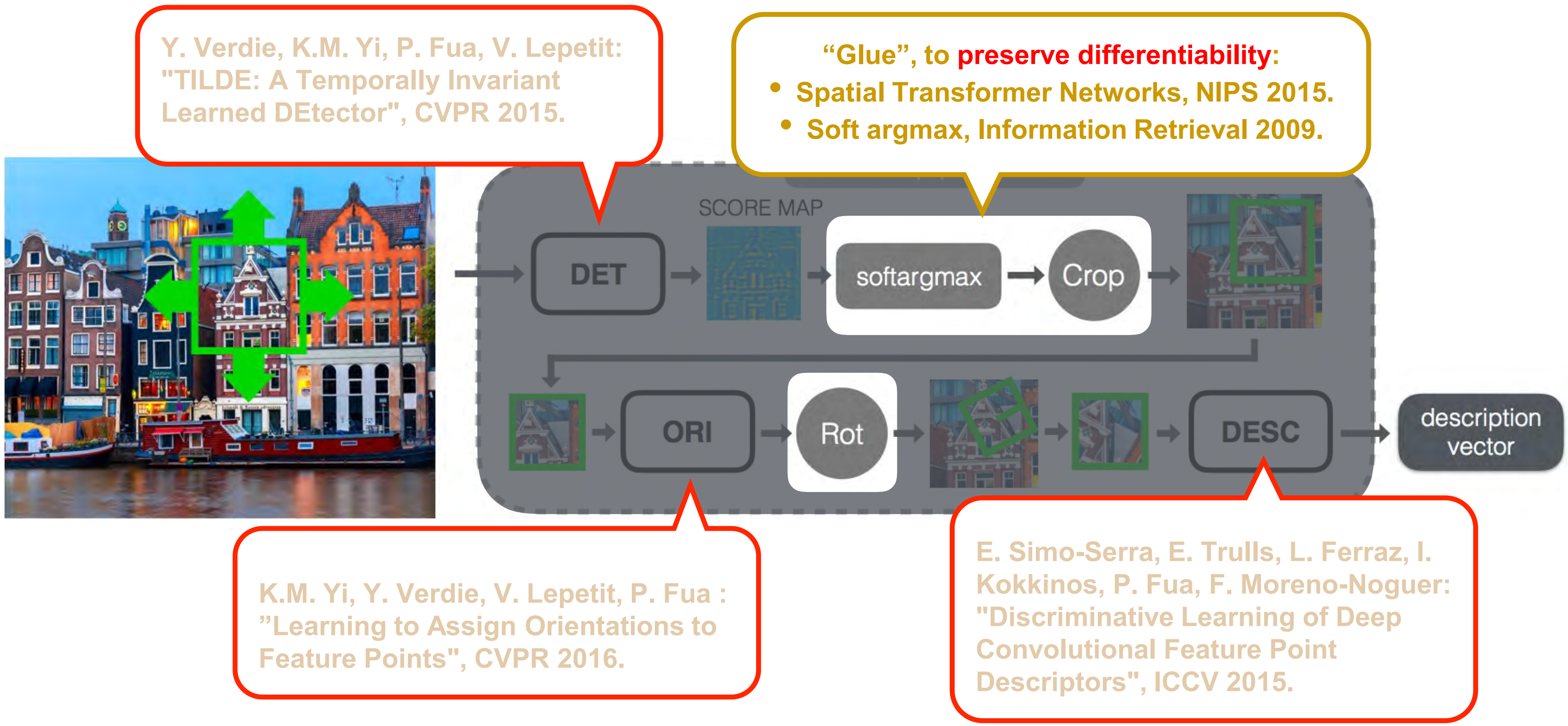
Y. Verdie, K.M. Yi, P. Fua, V. Lepetit: "TILDE: A Temporally Invariant Learned DEtector", CVPR 2015.



K.M. Yi, Y. Verdie, V. Lepetit, P. Fua : "Learning to Assign Orientations to Feature Points", CVPR 2016.

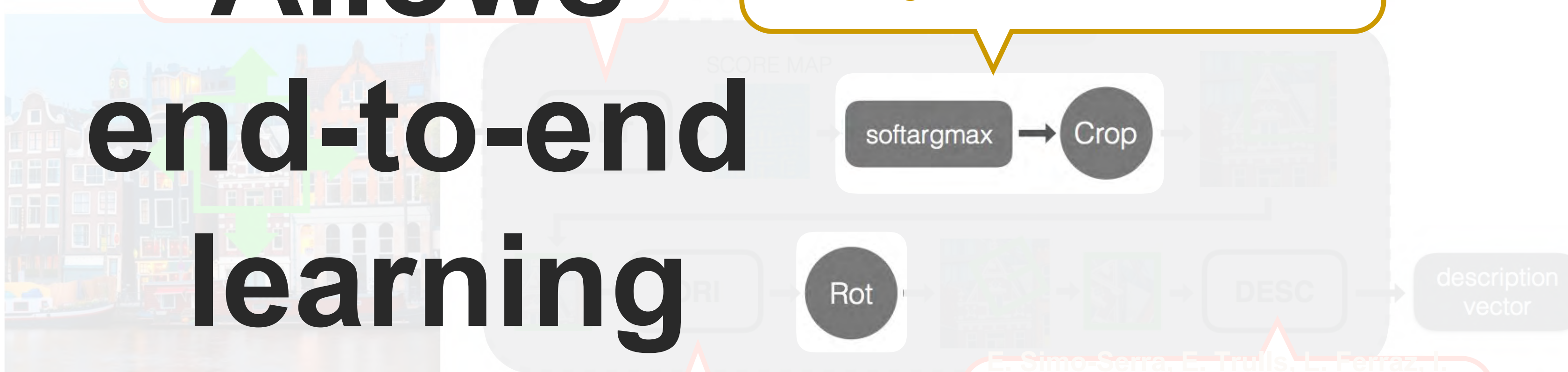
E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, F. Moreno-Noguer: "Discriminative Learning of Deep Convolutional Feature Point Descriptors", ICCV 2015.

The LIFT Network



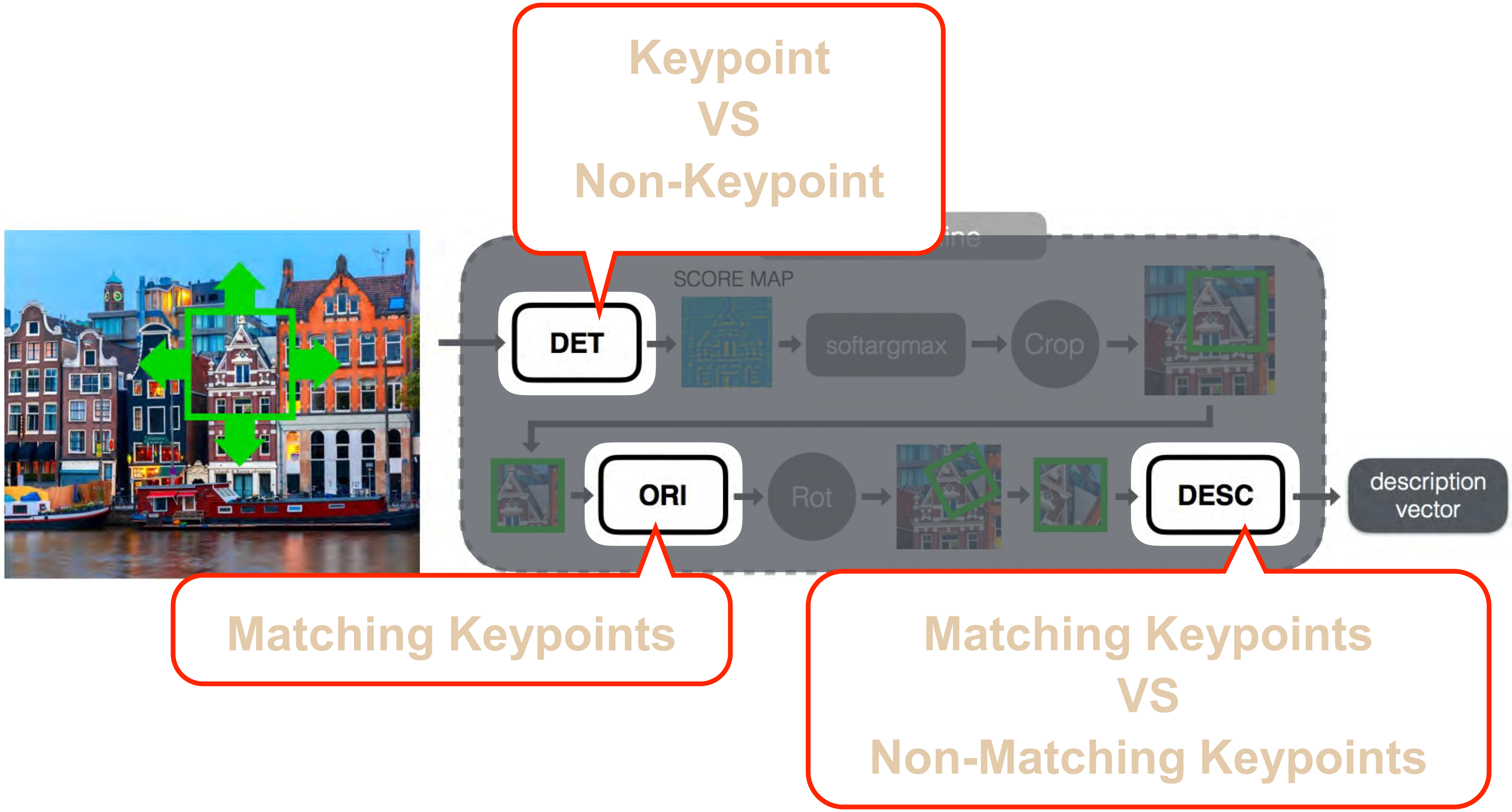
The LIFT Network

“Glue”, to **preserve differentiability**:
- Spatial Transformer Networks, NIPS 2015.
- Soft argmax, Information Retrieval 2009.

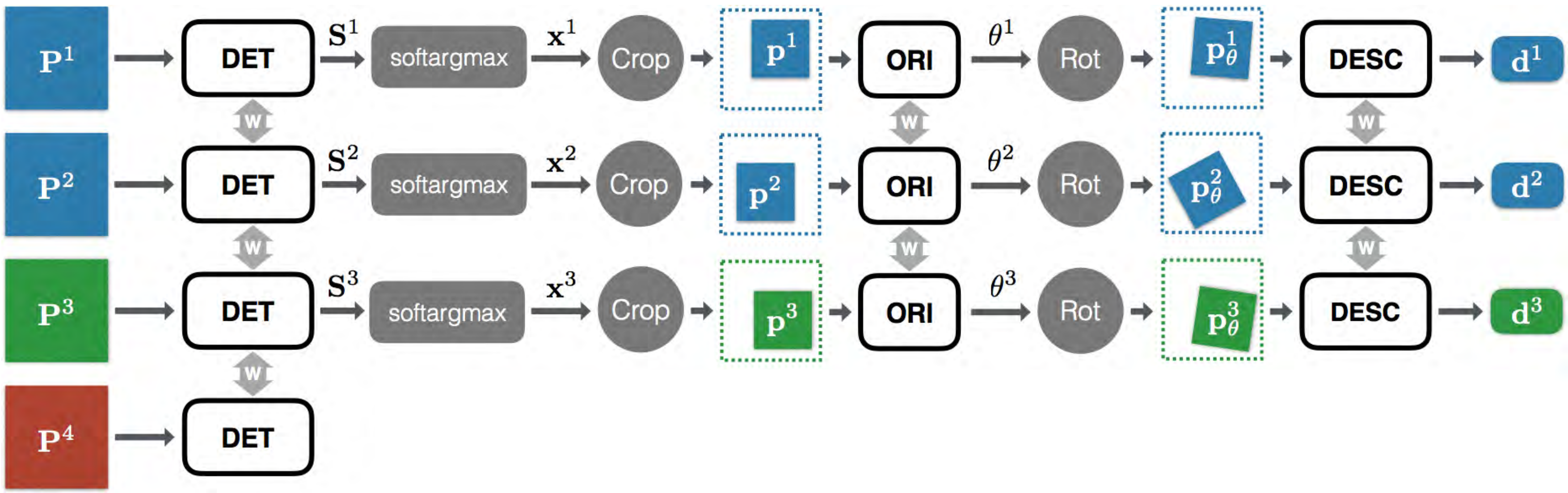


Allows end-to-end learning

Training requires various patches



Quadruplet Siamese Network



P$_1$, P$_2$: corresponding keypoints.

P$_3$: non-corresponding keypoint.

P$_4$: non-keypoint.

A single, global cost function

$$\min_{\{f_\mu, g_\phi, h_\rho\}} \sum_{\{(\mathbf{P}_1,\mathbf{P}_2,\mathbf{P}_3,\mathbf{P}_4)\}} \gamma \mathcal{L}_{class}(\mathbf{P}^1,\mathbf{P}^2,\mathbf{P}^3,\mathbf{P}^4) + \mathcal{L}_{pair}(\mathbf{P}^1,\mathbf{P}^2)$$

detector
orientation
descriptor

detector

$$\mathcal{L}_{\text{class}}(\mathbf{P}^1,\mathbf{P}^2,\mathbf{P}^3,\mathbf{P}^4) = \sum_{i=1}^{4} \alpha_i \max\left(0, \left(1 - \operatorname{softmax}\left(f_\mu\left(\mathbf{P}^i\right)\right) y_i\right)\right)^2$$

detector

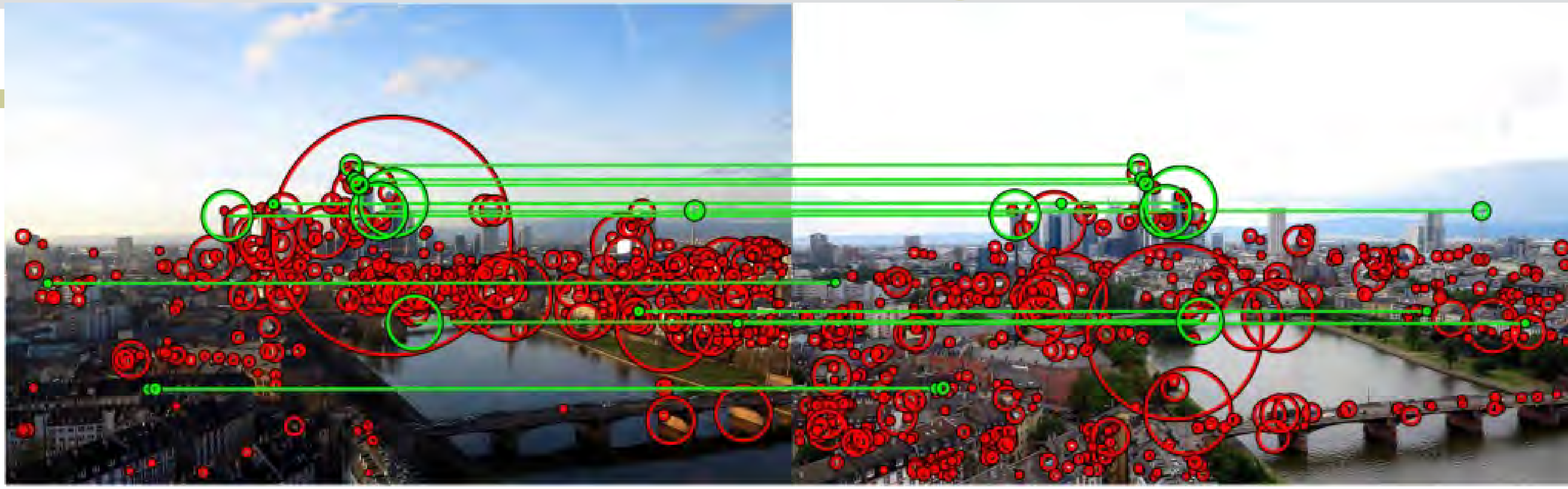
$$\begin{aligned}\mathcal{L}_{\text{pair}}(\mathbf{P}^1,\mathbf{P}^2) = \| \;& h_\rho(G(\mathbf{P}^1, \operatorname{softargmax}(f_\mu(\mathbf{P}^1)))) - \\ & h_\rho(G(\mathbf{P}^2, \operatorname{softargmax}(f_\mu(\mathbf{P}^2)))) \quad \|_2\end{aligned}$$

descriptor

$$G(\mathbf{P},\mathbf{x}) = \operatorname{Rot}(\mathbf{P},\mathbf{x}, g_\phi(\operatorname{Crop}(\mathbf{P},\mathbf{x})))$$

orientation

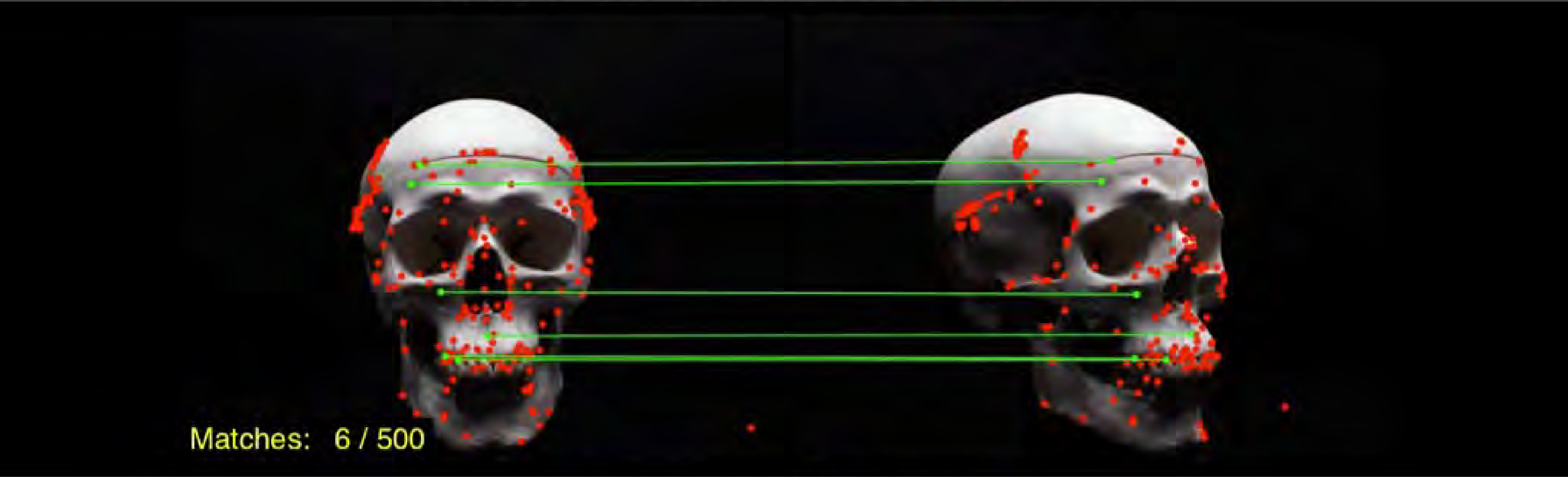
Matching features on **'Webcam'**, sequence **'Frankfurt'**.
Correct matches shown with green lines.



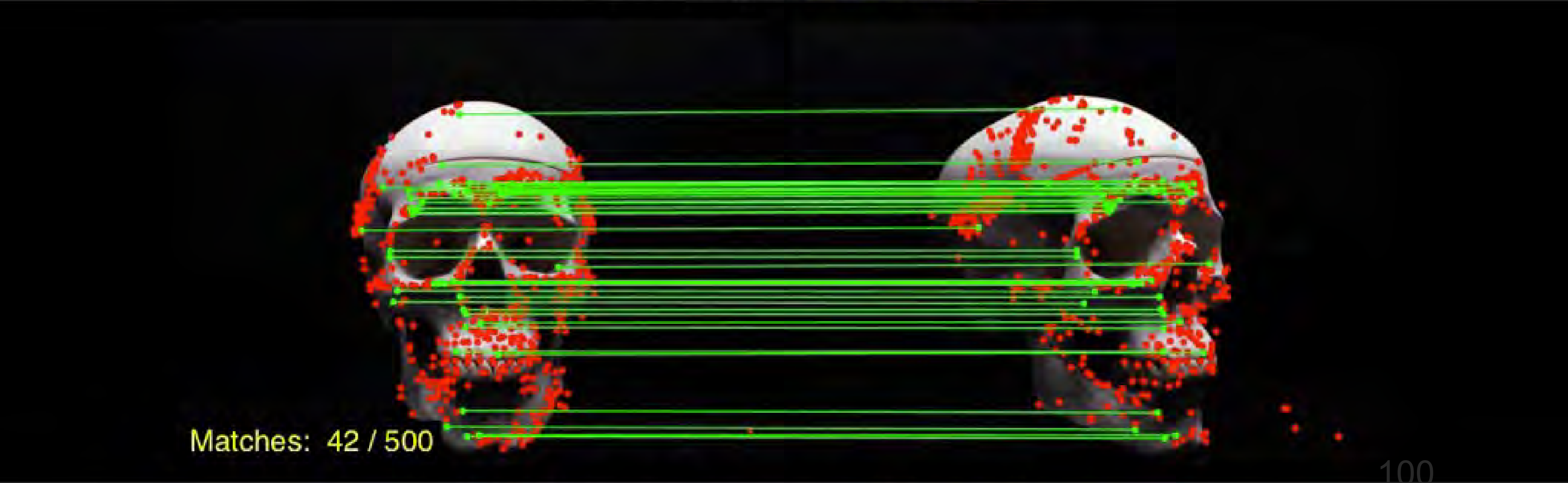
SIFT. Average: **23.1** matches

LIFT (Ours). Average: **60.6** matches

Matching features on **'DTU'**, sequence **#19**.
Correct matches shown with **green** lines.

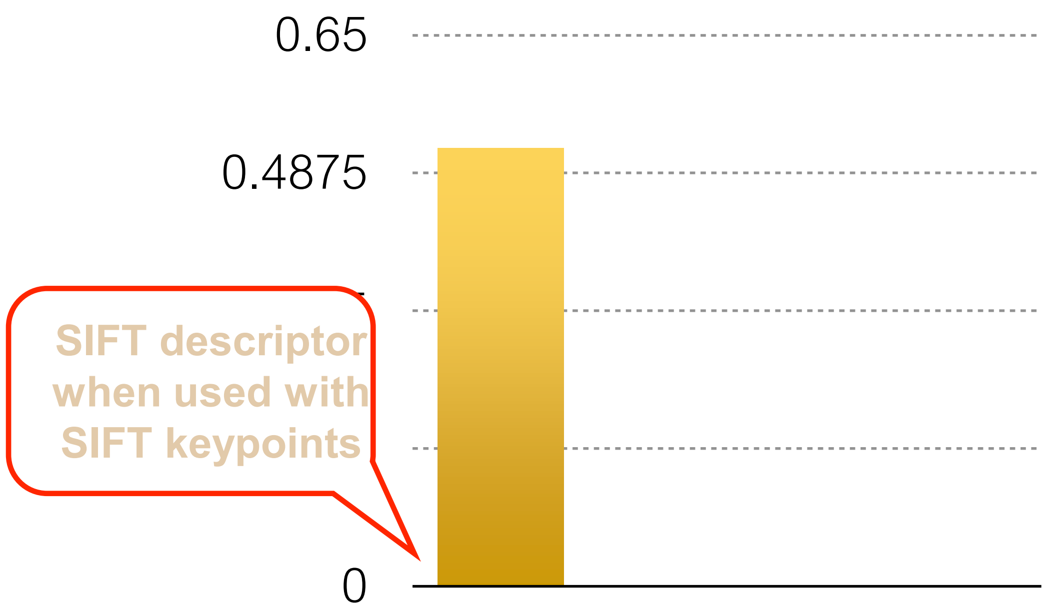


SIFT. Average: **34.1** matches



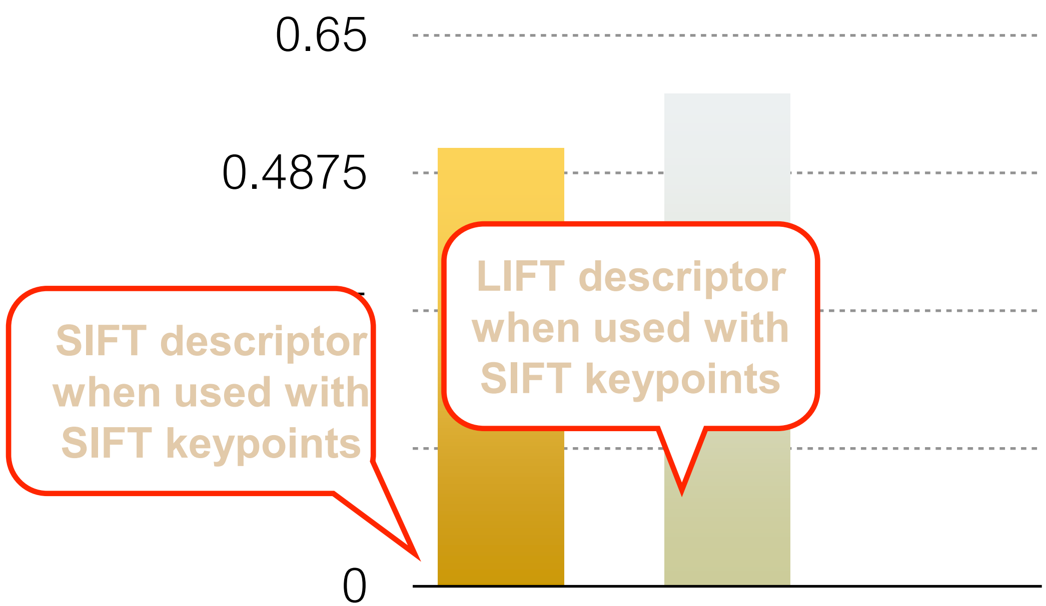
LIFT (Ours). Average: **98.5** matches

Each component is ***meant for*** each other



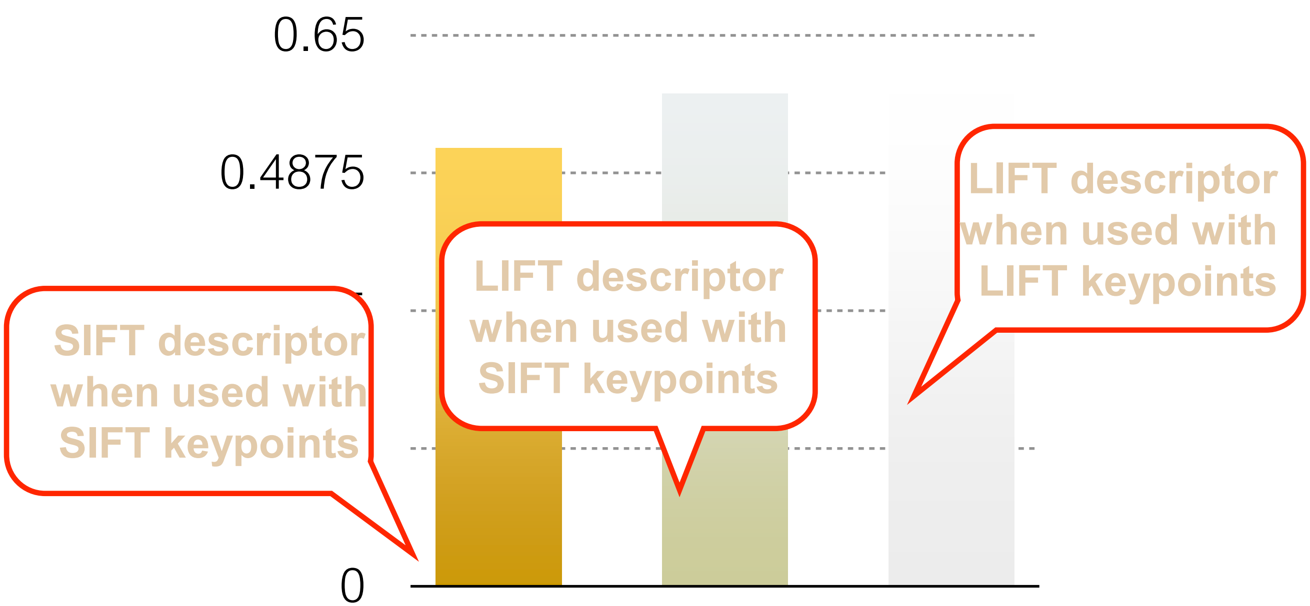
Descriptor performance (NN mAP)

Each component is ***meant for*** each other



Descriptor performance (NN mAP)

Each component is ***meant for*** each other



Descriptor performance (NN mAP)

Floating Point Descriptors: A Summarization

- For patch level datasets, learning based methods generally outperform hand-crafted ones.
- For image level dataset, performance gap between learning based methods and hand-crafted methods is not significant, except for LIFT.
- For domain adaptation (e.g. visible to IR), hand-crafted descriptors are more adaptable.
- CNN based methods are dominant in the learning based methods.
- CNN based methods operating on the Euclidean space is highly required for wider application.

Software

- OpenCV: http://opencv.org/
  - SIFT, SURF, BRISK, BRIEF, ORB, FREAK
- VLFeat: http://www.vlfeat.org/
  - SIFT, LIOP, Covariant Feature Detectors
- Authors' pages, Github, etc.
- Supplementary containing implementation information
  - Learned Orientations: https://infoscience.epfl.ch/record/217982/files/0141-supp.pdf
  - LIFT: https://documents.epfl.ch/groups/c/cv/cvlab-unit/www/data/keypoints/lift/paper_1377_supplementary.pdf

Image Matching Benchmark and Challenge

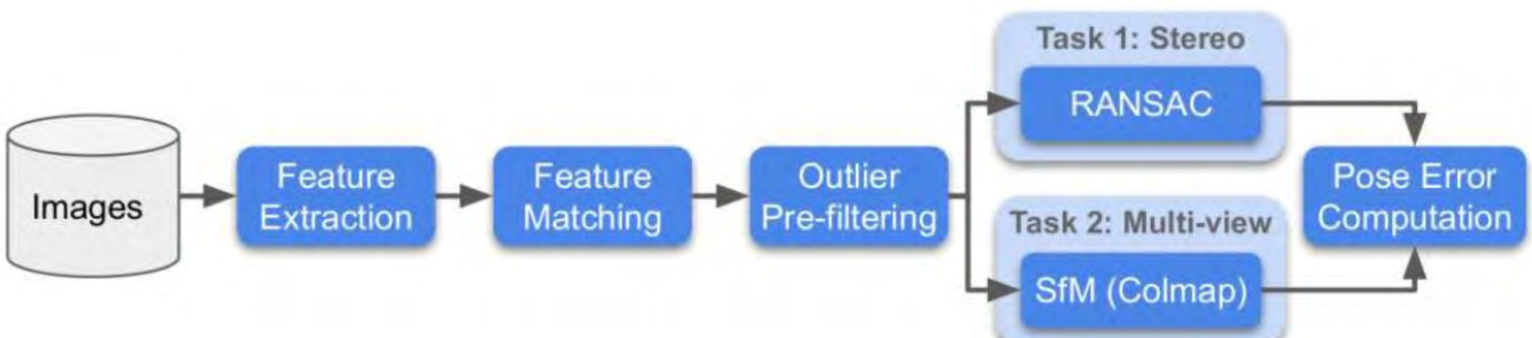


Image Matching Benchmark and Challenge

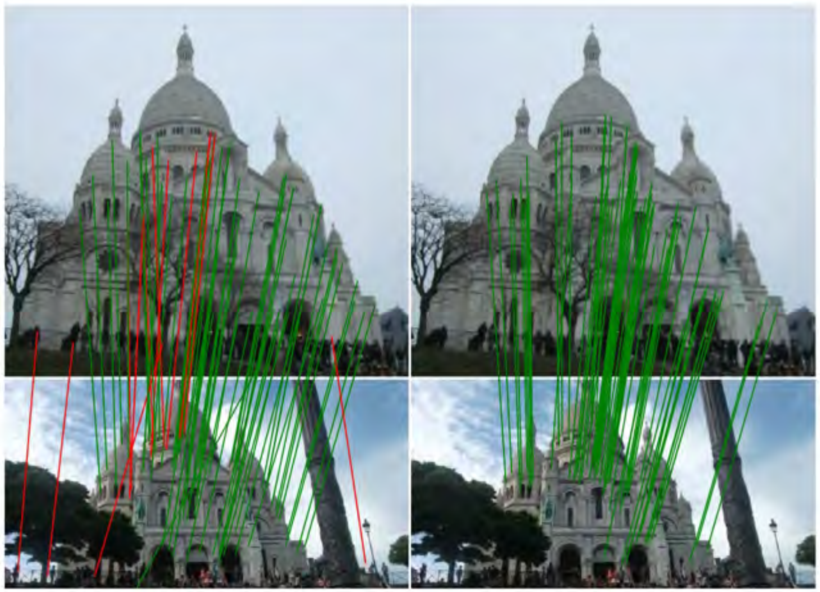


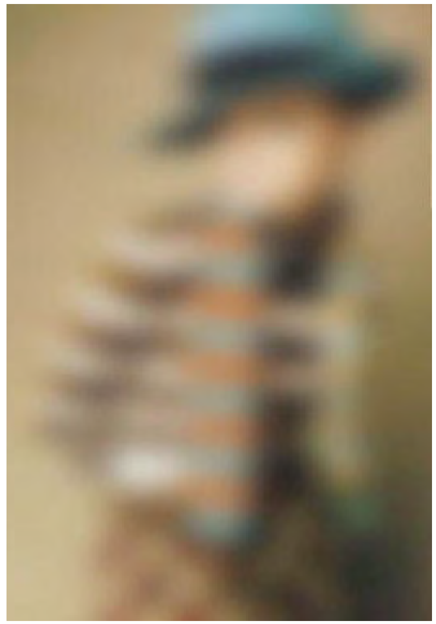
Figure 1. Every paper claims to outperform the state of the art. Is this possible, or an artifact of insufficient validation? On the left, we show stereo matches obtained with **D2-Net** (2019) [33], a state-of-the-art local feature, using OpenCV RANSAC with its default settings. On the right, we show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

		PyRANSAC		DEGENSAC		MAGSAC		
Method	NF	NI↑	mAP(10^o)↑	NI↑	mAP(10^o)↑	NI↑	mAP(10^o)↑	Rank
CV-SIFT	7879.0	153.9	.4160	222.7	.4608	270.6	.4586	13
VL-SIFT	7901.0	166.2	.4137	241.2	.4643	301.0	.4638	12
VL-Hessian-SIFT	8000.0	186.5	.3915	264.0	.4489	318.0	.4394	15
VL-DoGAff-SIFT	7910.9	218.6	.4049	229.7	.4653	291.9	.4624	11
VL-HesAffNet-SIFT	8000.0	190.8	.4081	271.6	.4659	328.5	.4581	10
CV-$\sqrt{\text{SIFT}}$	7884.0	176.3	.4348	257.4	.4921	317.4	.4891	6
SURF	7749.0	113.0	.2326	117.8	.2452	136.1	.2481	19
AKAZE	7879.8	184.3	.2738	232.7	.3142	284.4	.3054	17
ORB	7128.2	113.1	.1381	136.6	.1723	163.2	.1632	22
DoG-HardNet	7884.1	229.9	.4668	342.2	**.5286**	404.0	.5147	1
DoG-HardNetAmos+	7884.1	213.6	.4511	316.9	.5125	373.5	.5011	3
L2Net	7884.8	190.2	.4478	280.9	.4971	329.1	.4884	5
Key.Net-HardNet	7998.1	353.3	.3990	375.2	.4700	636.5	.4529	9
Geodesc	7884.3	179.7	.4183	264.0	.4787	340.3	.4753	8
ContextDesc	4811.1	248.8	.4283	261.4	.4856	356.1	.4662	7
SOSNet	7884.3	215.1	.4595	319.8	.5233	418.5	.5177	2
LogPolarDesc	7884.3	243.5	.4495	366.0	.5080	461.0	.5001	4
SuperPoint (2k)	1178.9	88.1	.2359	84.7	.2669	113.2	.2620	18
LF-Net (2k)	2024.8	95.1	.1945	100.8	.2253	134.2	.2164	20
D2-Net (SS)	5540.7	273.5	.1432	241.4	.1639	428.0	.1560	23
D2-Net (MS)	6806.3	193.8	.1690	322.8	.1836	505.1	.1731	21

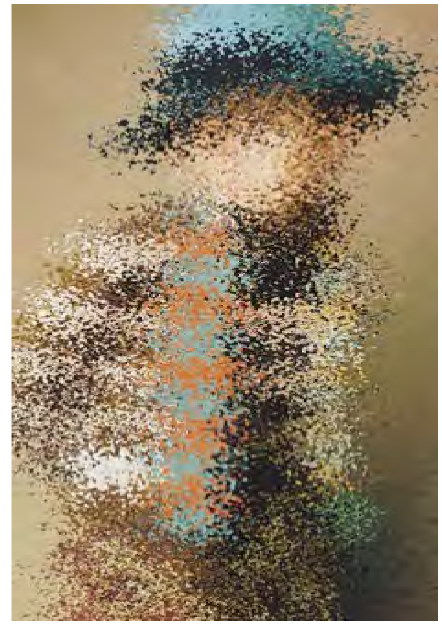
Table 1. **Stereo – Test set.** We report: **(NF)** Number of Features; **(NI)** Number of Inliers produced by RANSAC; and **mAP(10^o)**. Top three methods by mAP marked in red, green and blue.

Discriminative power

Raw pixels



Sampled



Locally orderless

Global histogram

Generalization power

Summary: Value of Local Features

- **Advantages**
  - Critical to find distinctive and repeatable local regions for multi-view matching.
  - Complexity reduction via selection of distinctive points.
  - Describe images, objects, parts without requiring segmentation; robustness to clutter & occlusion.
  - Robustness: similar descriptors in spite of moderate view changes, noise, blur, etc.

- **How can we use local features for such applications?**
  - Next: matching and recognition

Mikolajczyk, K. et al. A comparison of affine region detectors. IJCV, 2005
Mikolajczyk, K. et al. A performance evaluation of local descriptors. T-PAMI, 2005.