BELID: Boosted Efficient Local Image Descriptor

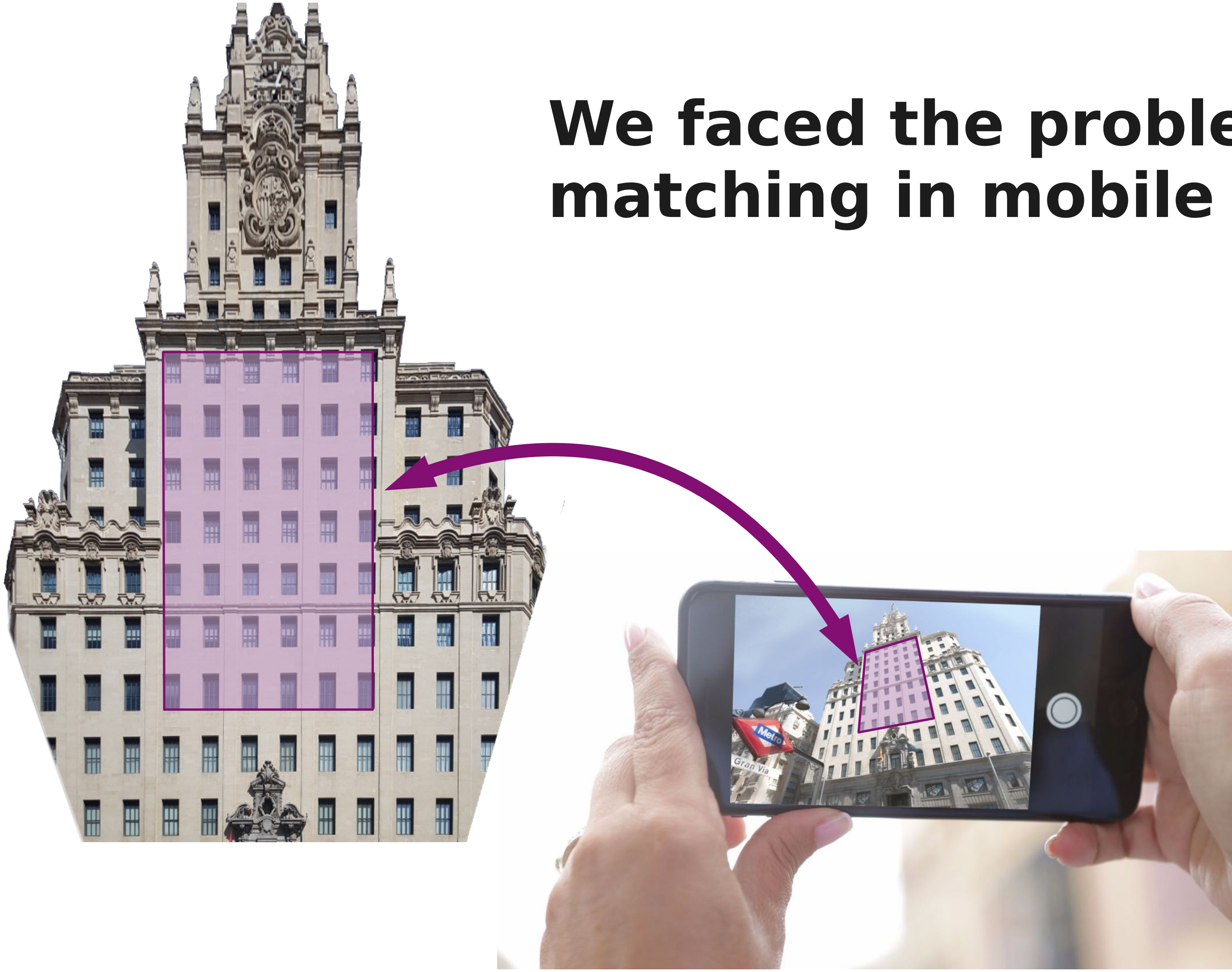
Iago Suárez, Ghesn Sfeir, José M. Buenaposada, Luis Baumela

Computer Vision and Aerial Robotics Group

Motivation

Motivation

We faced the problem image matching in mobile phones:



Motivation: Feature Matching

Input Image

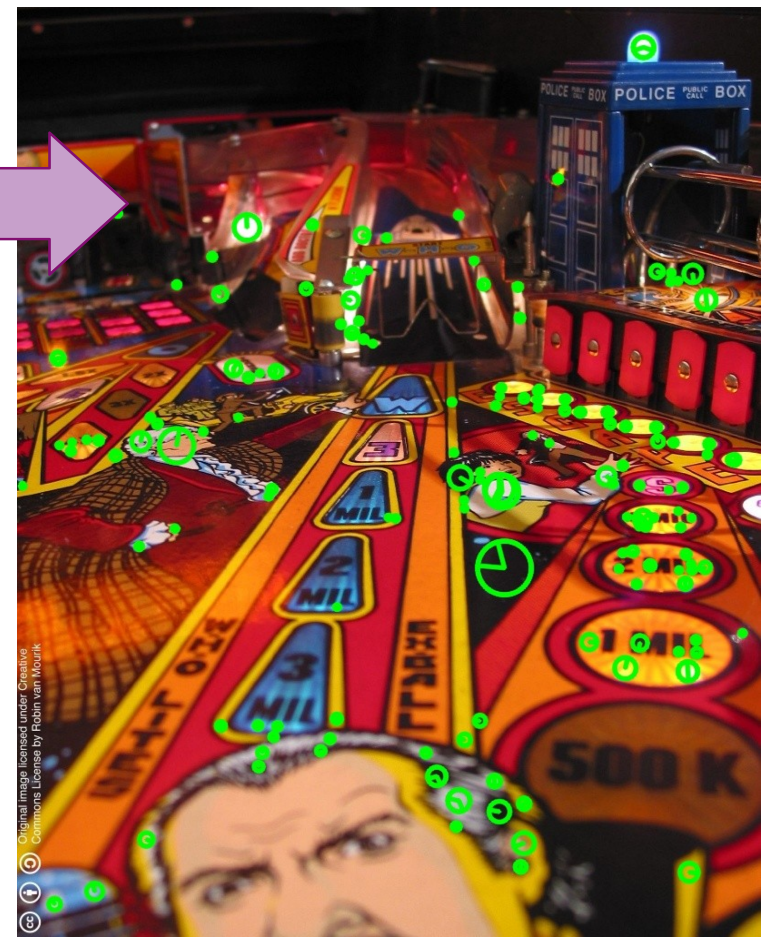
Pattern

https://www.github.com/artoolkit/artoolkit5

Motivation: Feature Matching

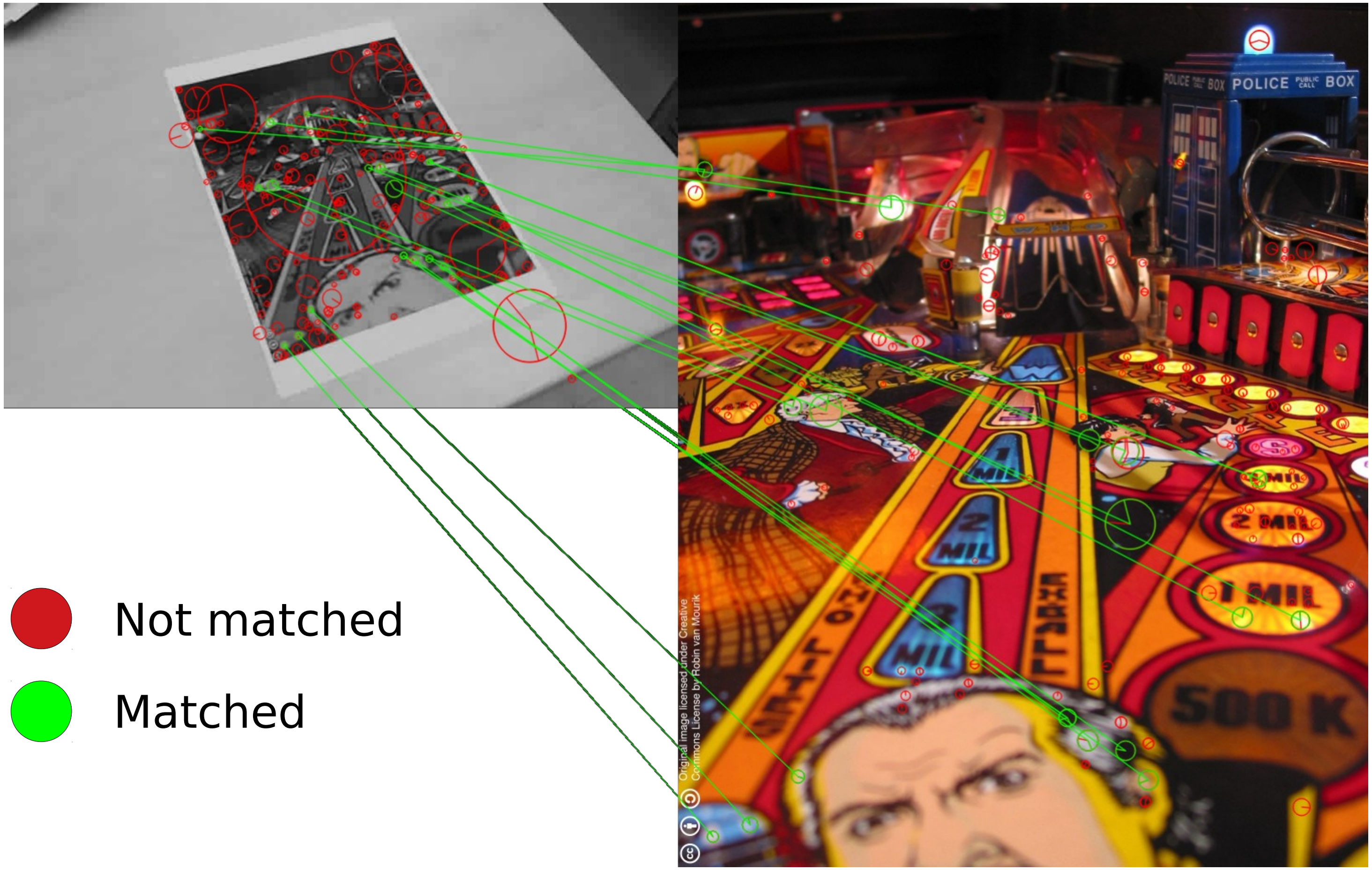
Input Image

Pattern



Motivation: Feature Matching

Matching of the image features with the pattern features

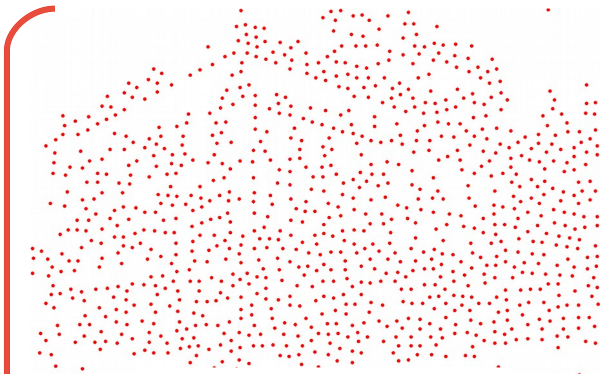


- Not matched
- Matched

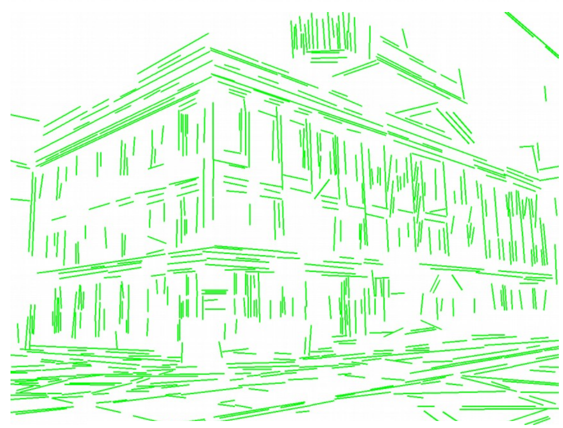
Motivation: Feature Matching

We first need to detect feature like:

Input image



Feature corners



Feature segments

Corner detectors:

- Harris (Harris, 1988)
- FAST(Rosten, 1906)

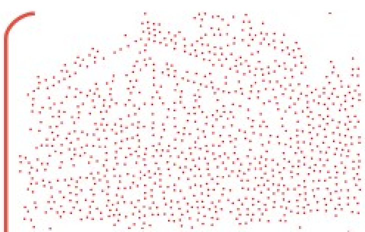
Line segment detectors:

- LSD (Von Gioi, 2010)
- EDLines (Akinlar, 2011)

Motivation: Feature Matching

Other applications:

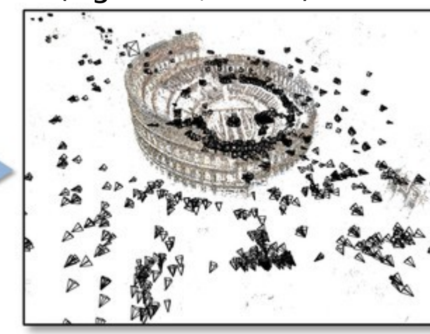
Input image



Feature corners

Feature segments

Structure from Motion (Agarwal, 2009)



SLAM

robots.ox.ac.uk/~vgg/blog/mapping-environments-with-deep-networks.html

Previous Work: SIFT Descriptor

SIFT(Lowe, 1999) is the most widely used descriptor:

Previous Work: SIFT Descriptor

SIFT(Lowe, 1999) is the most widely used descriptor:

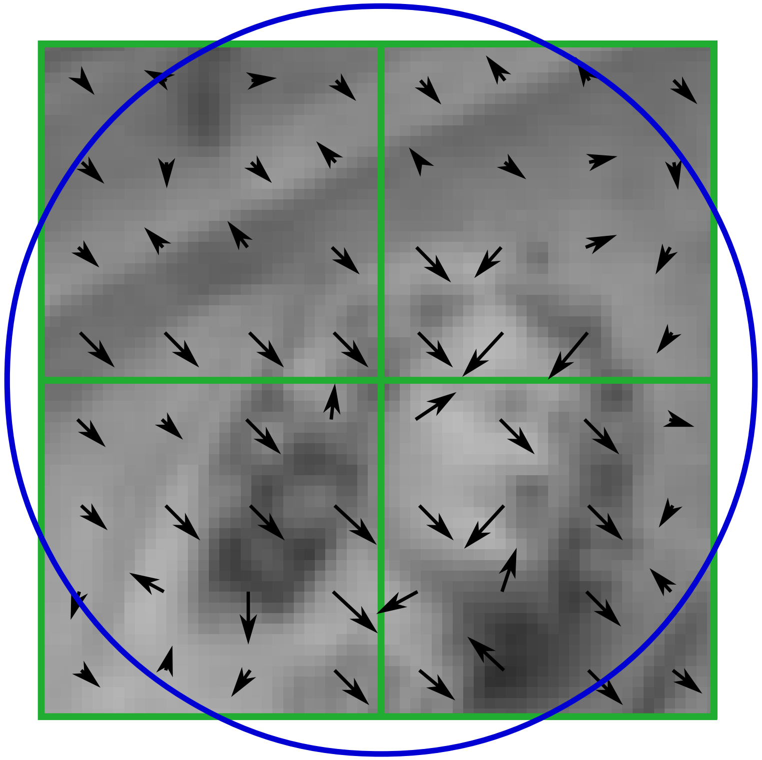


Image gradients

Previous Work: SIFT Descriptor

Uses the histograms of gradients in a fixed grid:

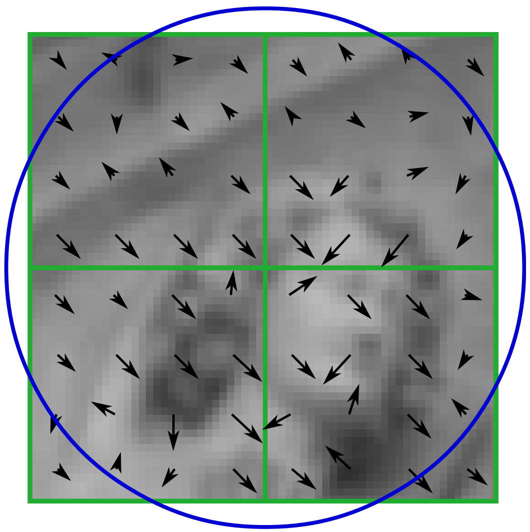
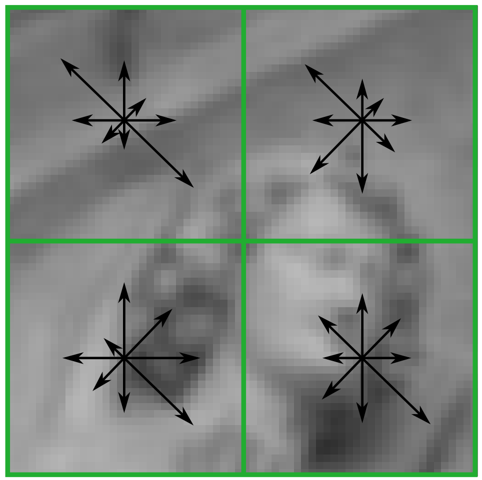


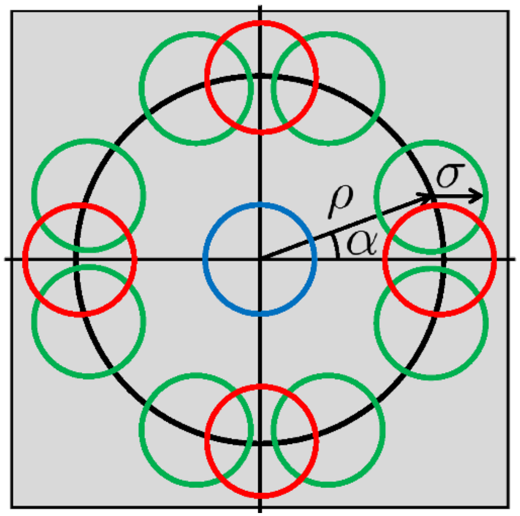
Image gradients



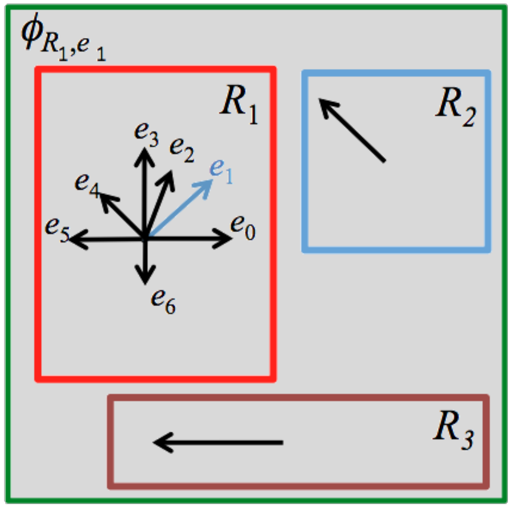
Gradient orientation histograms:
- **Fixed grid**: 2 x 2
- **Fixed scale**: The cell size
- **Dense**: Uses all patch pixels

Previous Work: Inefficient learnt descriptors

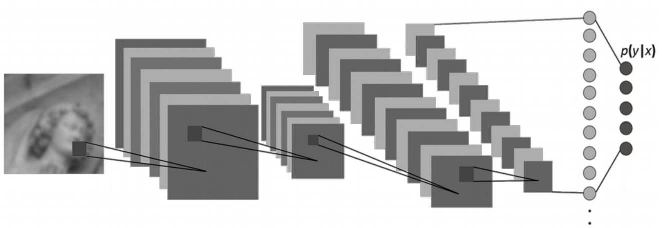
Instead of fixing the grid, the scale and the gradient as information, this can be learnt:



DLCO – VGG
(Simonyan, 2014)

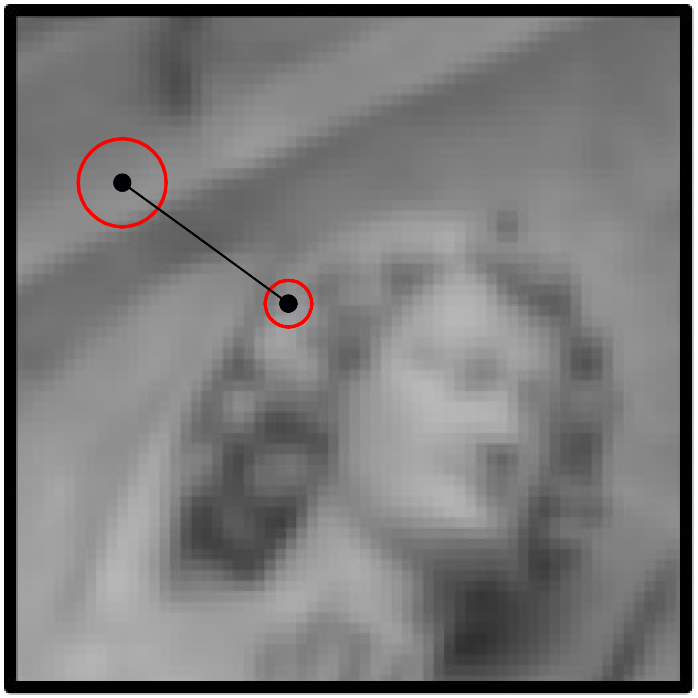


BinBoost & FP-Boost
(Trzcinski, 2015)



Deep Learning descriptors

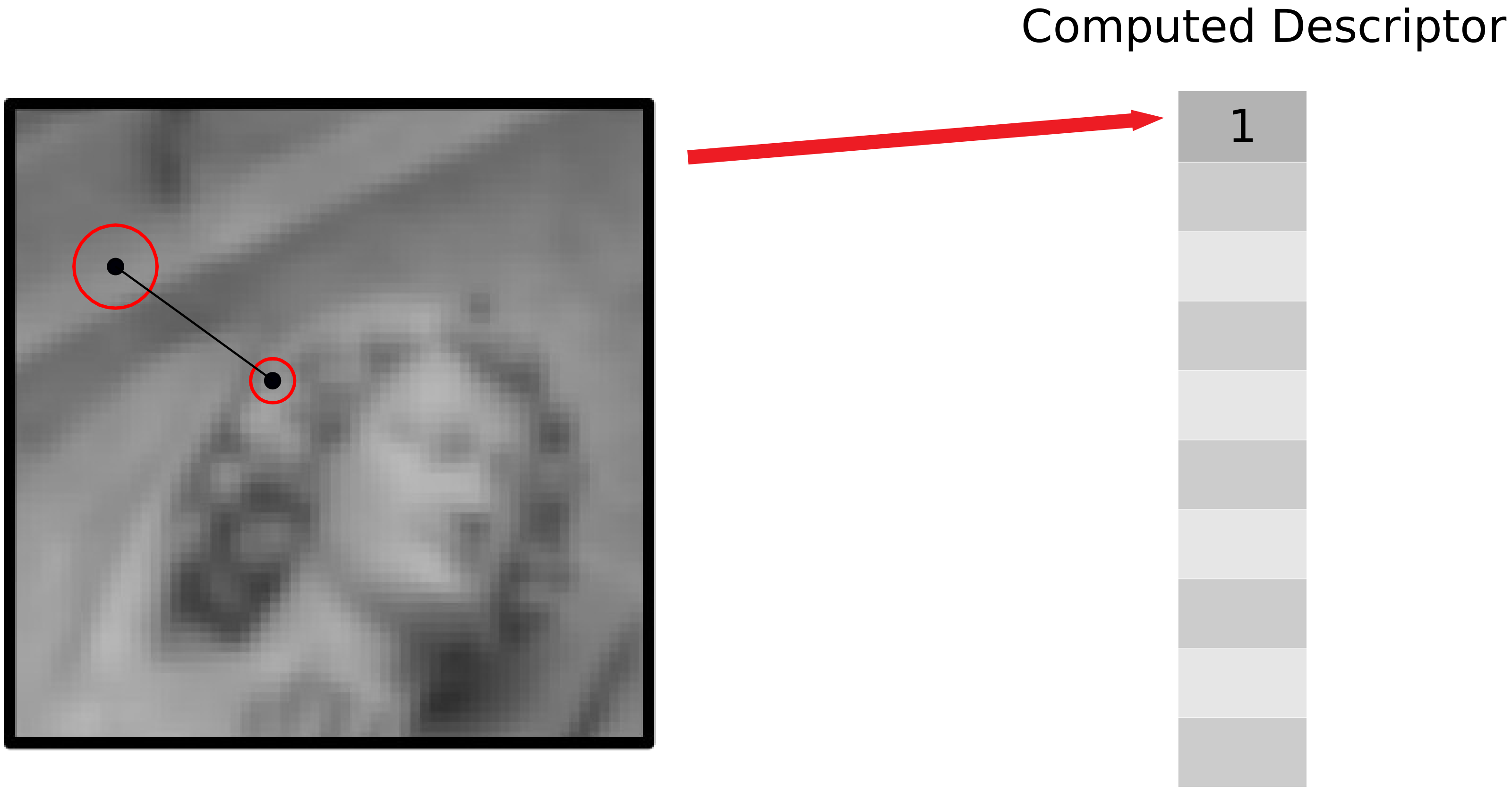
Previous Work: Efficient descriptors



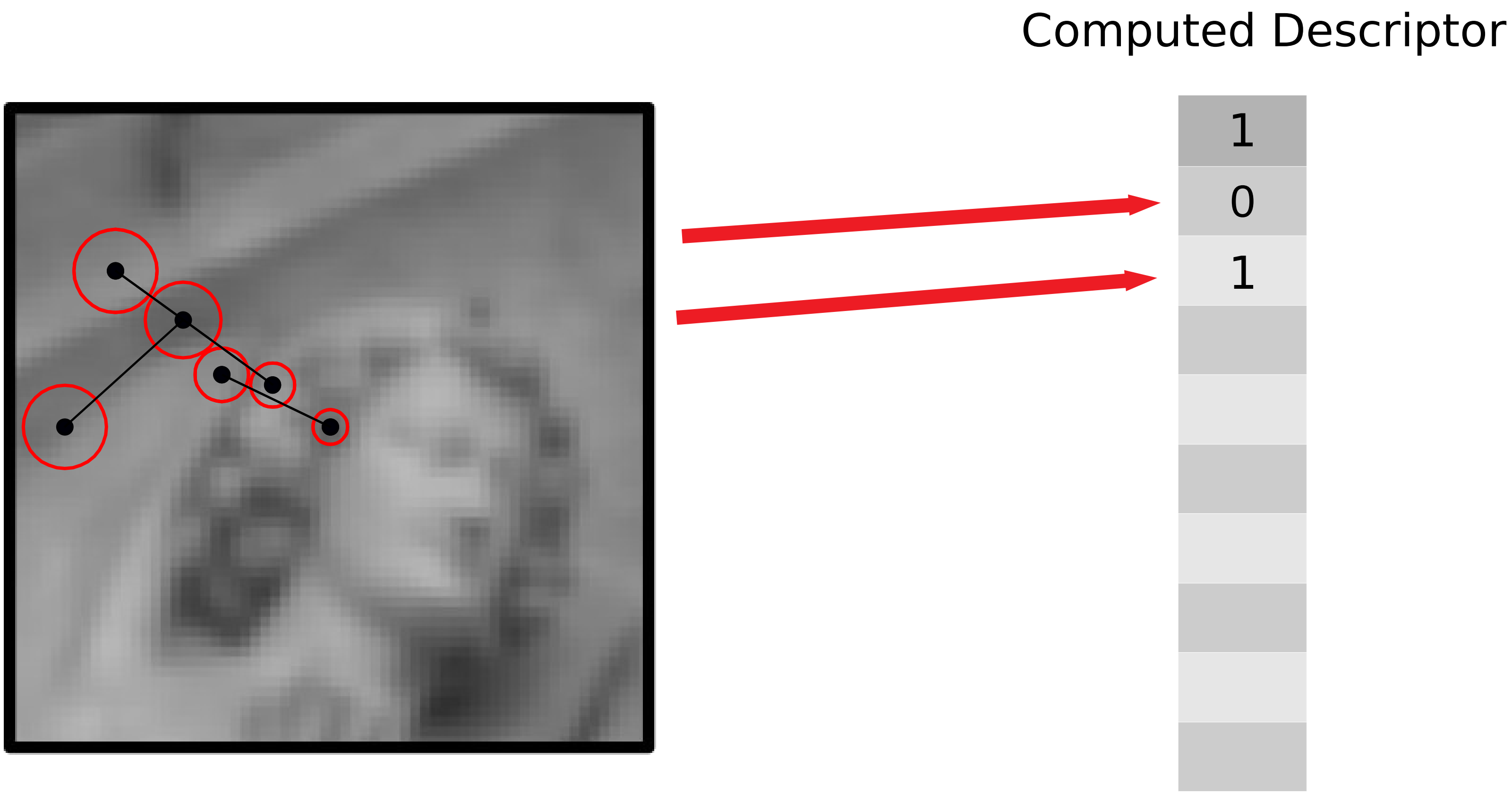
Efficient alternatives to SIFT are based in **aproximate gradient by comparing pixel intensities**

- Sparse approach → Fast

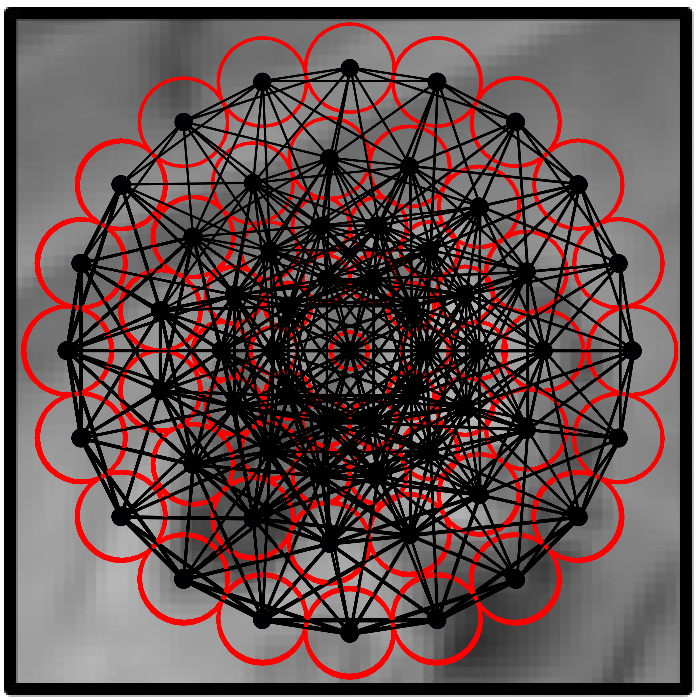
Previous Work: Efficient descriptors



Previous Work: Efficient descriptors



Previous Work: Efficient descriptors

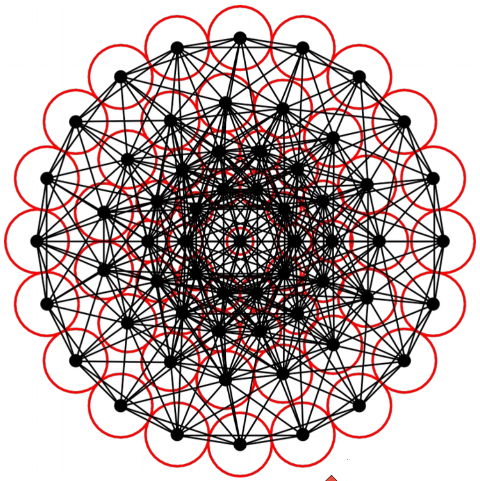


BRISK

Computed Descriptor

1
0
1
.
.
.
1
0

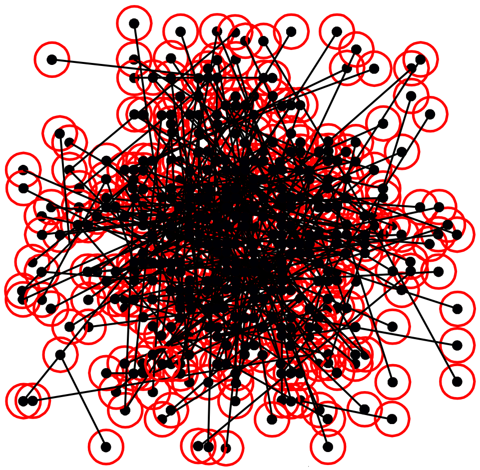
Previous Work: Descriptor comparison



Pattern	**Grid**	BRISK	
	Random		
	Learnt		

Leutenegger, S., Chli, M., & Siegwart, R. (2011). BRISK: Binary robust invariant scalable keypoints. In 2011 IEEE international conference on computer vision (ICCV) (pp. 2548-2555). Ieee.

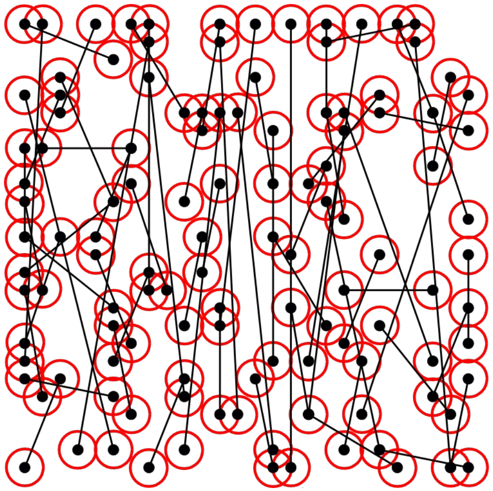
Previous Work: Descriptor comparison



Pattern			
	Grid	BRISK	
	Random	BRIEF	
	Learnt		

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010, September). Brief: Binary robust independent elementary features. In European conference on computer vision (pp. 778-792). Springer, Berlin, Heidelberg.

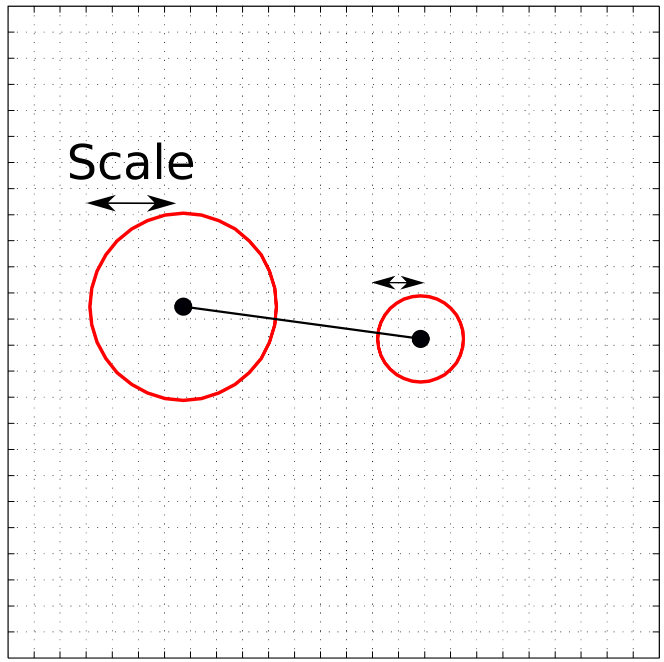
Previous Work: Descriptor comparison



Pattern	**Grid**	BRISK	
	Random	BRIEF	
	Learnt	ORB	

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. R. (2011, November). ORB: An efficient alternative to SIFT or SURF. In ICCV (Vol. 11, No. 1, p. 2).

Previous Work: Descriptor comparison



		Measurement Scale	
		Fixed	**Learnt**
Pattern	**Grid**	BRISK	
	Random	BRIEF	
	Learnt	ORB	

Previous Work: Descriptor comparison

BELID is a efficient descriptor that learns both:

- Descriptor Pattern
- Descriptor Measurements Scale

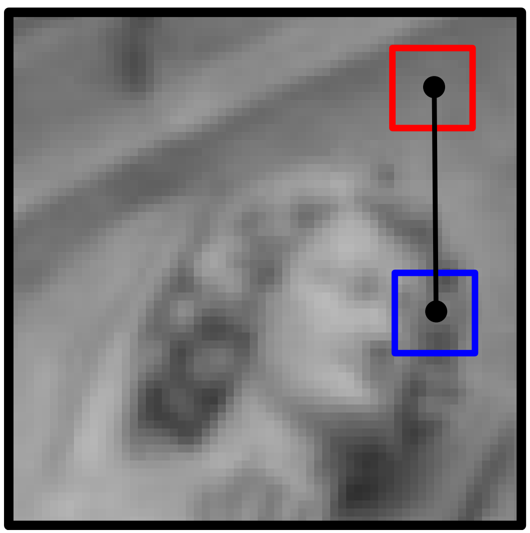
		Measurement Scale	
		Fixed	**Learnt**
Pattern	**Grid**	BRISK	
	Random	BRIEF	
	Learnt	ORB	**BELID**

The key of BELID:
A new measurement function

Our efficient measurement function

The function is the difference of the average gray level in two boxes

$\mathbf{x}$

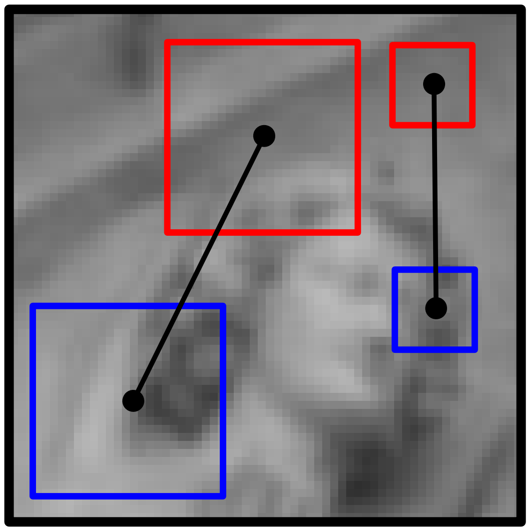


Original Patch to describe

Our efficient measurement function

The function is the difference of the average gray level in two boxes

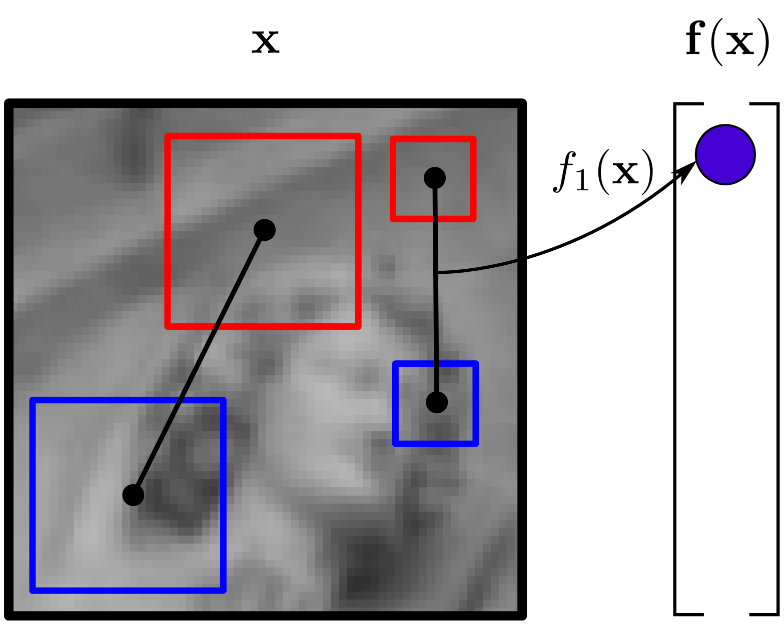
$\mathbf{x}$



Original Patch to describe

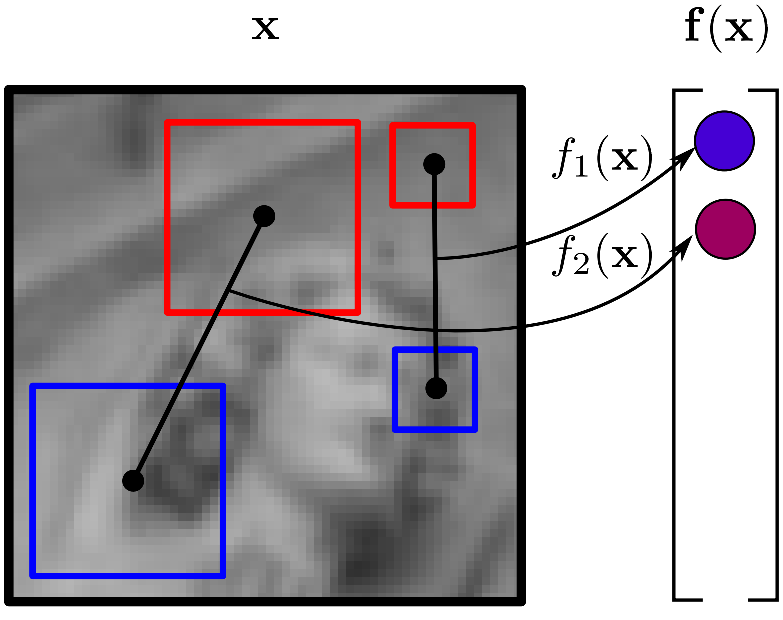
Our efficient measurement function

The function is the difference of the average gray level in two boxes



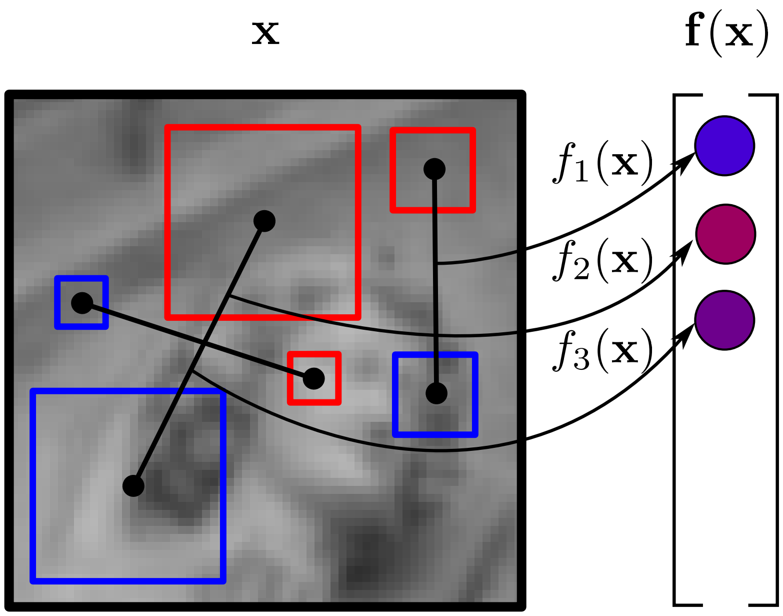
Our efficient measurement function

The function is the difference of the average gray level in two boxes



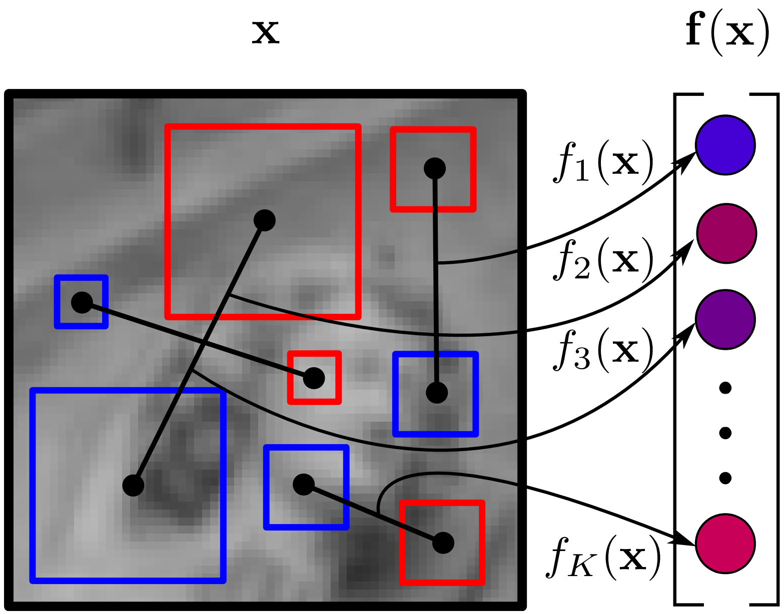
Our efficient measurement function

The function is the difference of the average gray level in two boxes



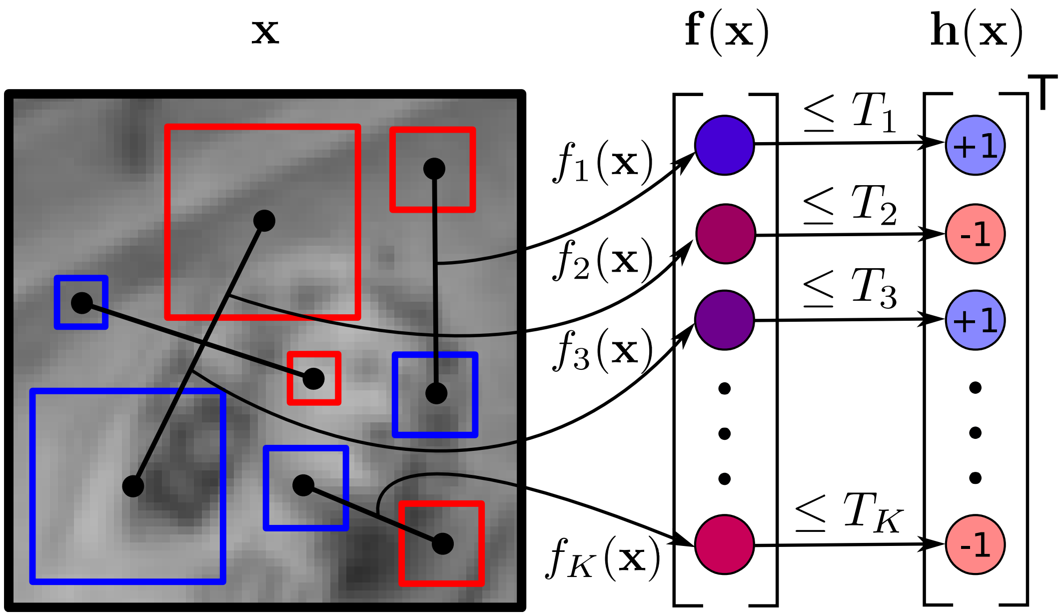
Our efficient measurement function

The function is the difference of the average gray level in two boxes



Our efficient measurement function

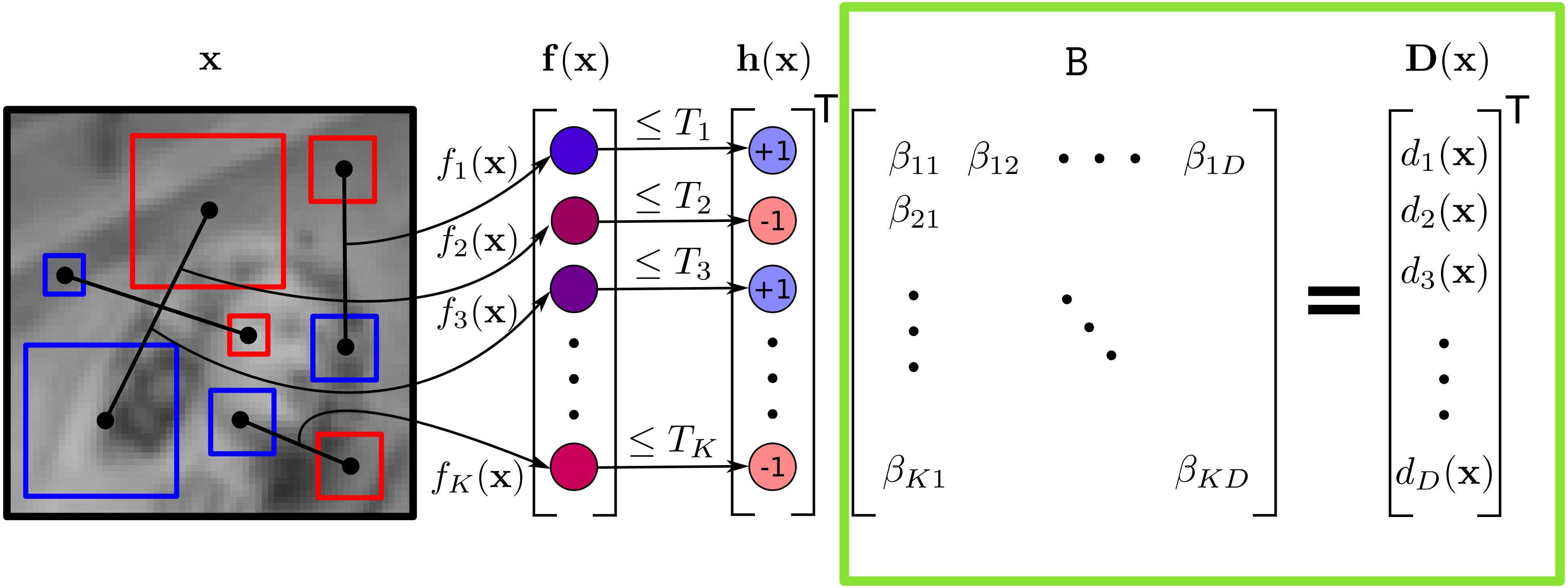
The function is the **thresholded** difference of the average gray level in two boxes.



Our efficient measurement function

Linear Transformation:

- Give more weight to the best h(x).
- Models the correlation between the h's.



How to select a good set of measurement functions?

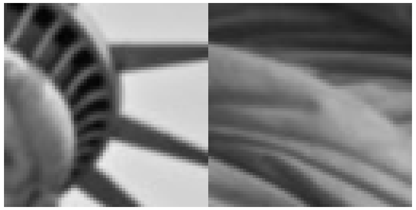
Boosting!

The training process

- **Training on Brown Balanced Dataset (Winder, 2007) of patch pairs where labels are:**
  - +1: Patches from the same 3D point
  - -1: Patches from different 3D point

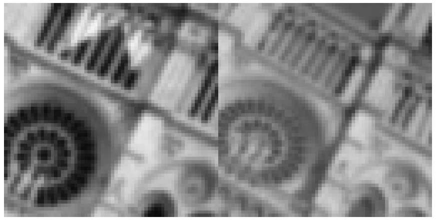
Liberty:

+1

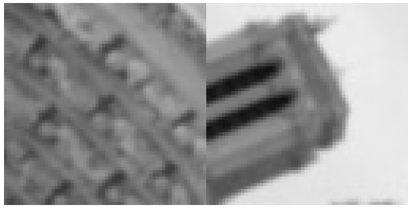


-1

Notre Dame:



+1



-1

BELID in the Boosting Framework

- Therefore we define our descriptors as:

$$\mathrm{H} : \mathrm{X} \rightarrow \left[\sqrt{\alpha_1} h_1(\mathbf{x}), \ldots, \sqrt{\alpha_M} h_M(\mathbf{x})\right]$$

0,34	-0.12	...	-0.0065

- And we train for a binary classification problem:

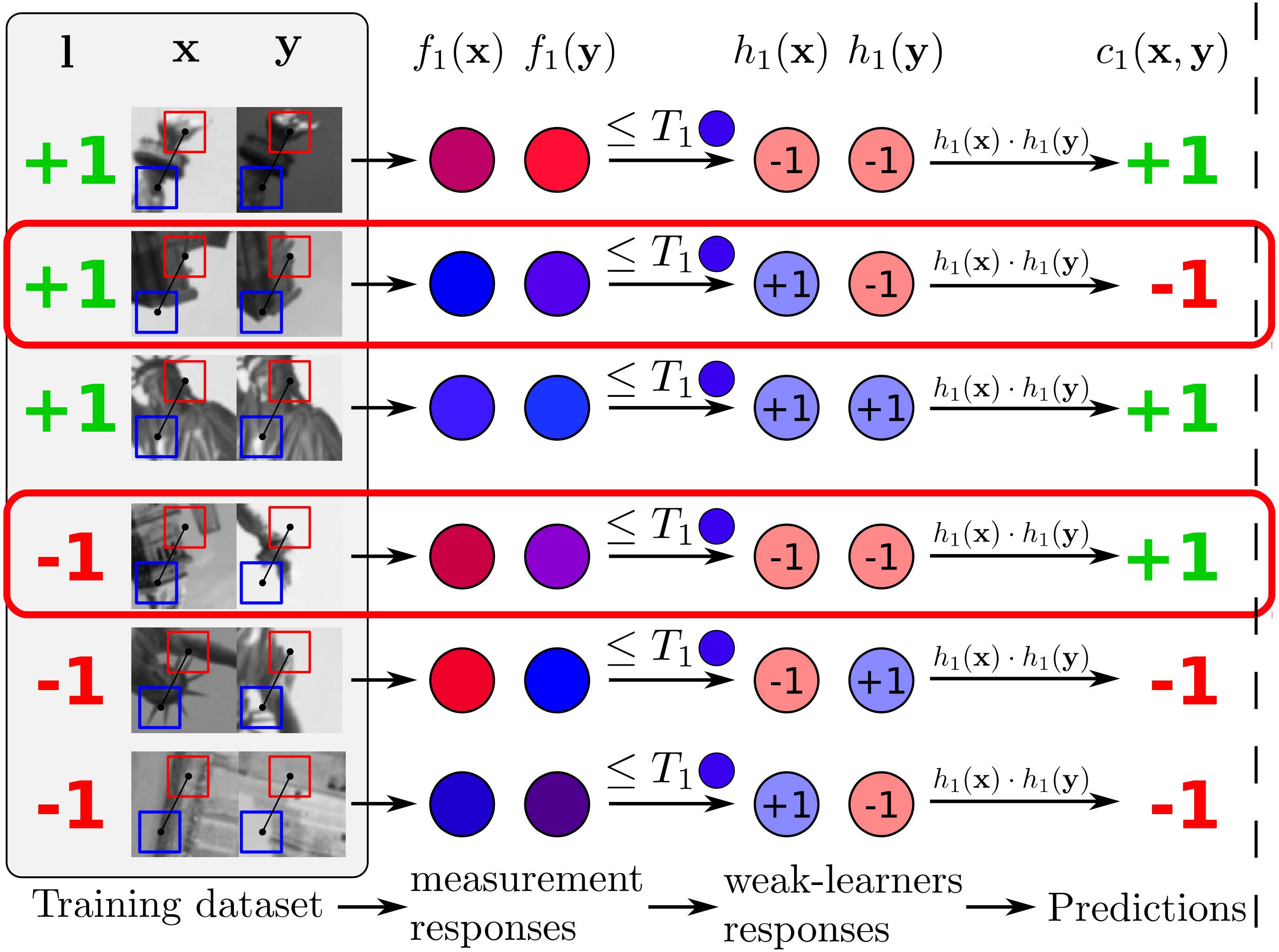
$$\mathrm{C} : (\mathrm{X}, \mathrm{Y}) \rightarrow \; sign\left(\alpha_1 c_1(\mathbf{x}, \mathbf{y}) + \ldots + \alpha_K c_K(\mathbf{x}, \mathbf{y})\right) \rightarrow \; \pm 1$$

+1 / -1

$$c_i(\mathbf{x}, \mathbf{y}) = h_i(\mathbf{x}) \cdot h_i(\mathbf{y})$$

-1 = -1 · +1

BELID in the Boosting Framework (first WL)



1. Find the weak-learner $\mathbf{h}$ that maximizes the **Youden's index**.

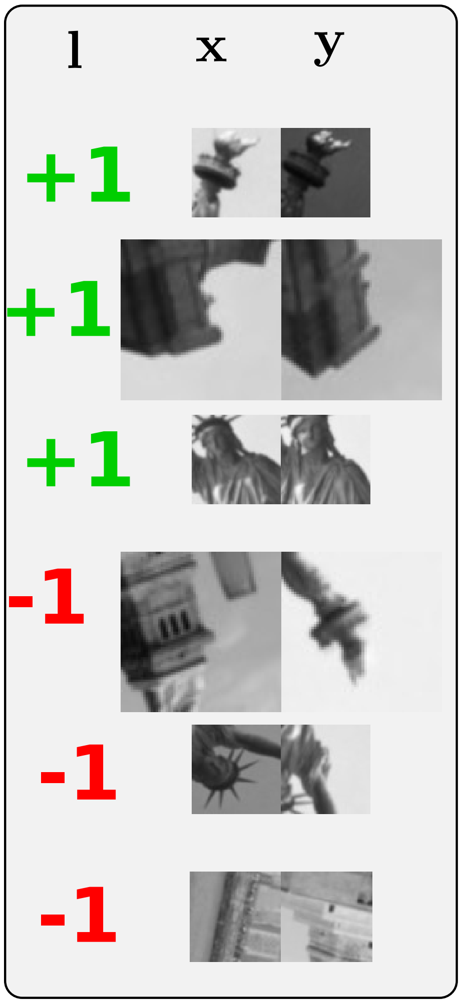
2. Calculate the error

$$\epsilon_1 = \sum_{\substack{i=1 \\ c_1(\mathbf{x}_i,\mathbf{y}_i)\neq l_i}}^{n} w_{i,1}$$

3. Choose the $\mathbf{h}$ weight:

$$\alpha_1 = \frac{1}{2}\ln\left(\frac{1-\epsilon_1}{\epsilon_1}\right)$$

BELID in the Boosting Framework (second WL)



Training dataset

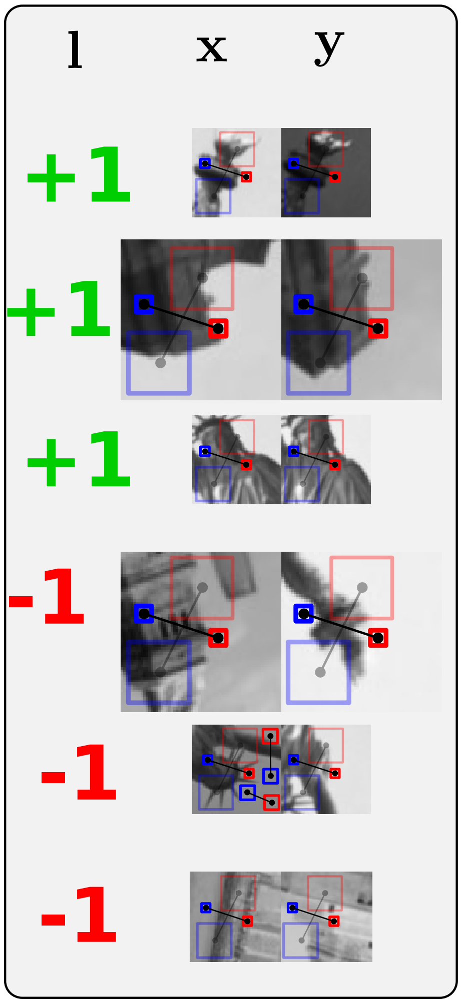
1. Find the weak-learner **h** that maximizes the **Youden's index**.

2. Calculate the error
$$\epsilon_1 = \sum_{\substack{i=1 \\ c_1(\mathbf{x}_i, \mathbf{y}_i) \neq l_i}}^{n} w_{i,1}$$

3. Choose the **h** weight:
$$\alpha_1 = \frac{1}{2} \ln\left(\frac{1-\epsilon_1}{\epsilon_1}\right)$$

4. Re-weight the samples:
$$w_{i,2} = w_{i,1} e^{-l_i \alpha_1 c_1(\mathbf{x}_i, \mathbf{y}_i)}$$

BELID in the Boosting Framework (n-th WL)

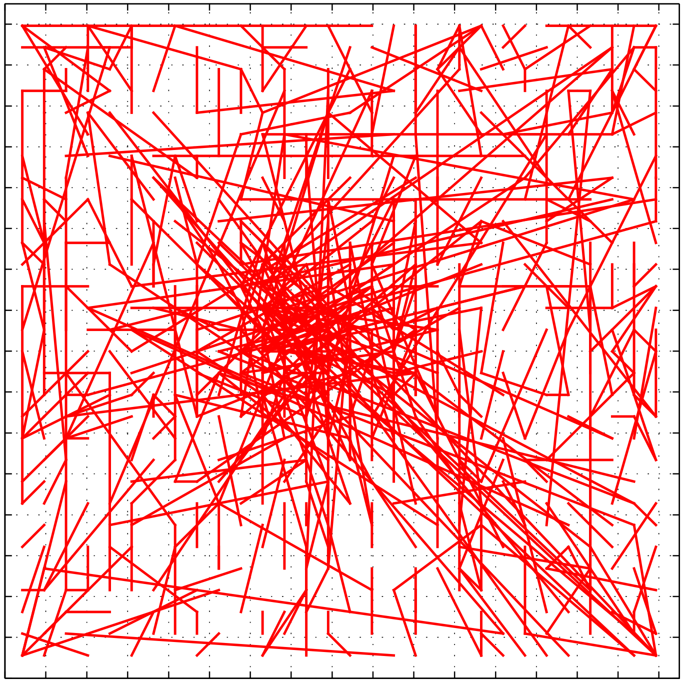


Training dataset → measurement responses → weak-learners responses → Predictions

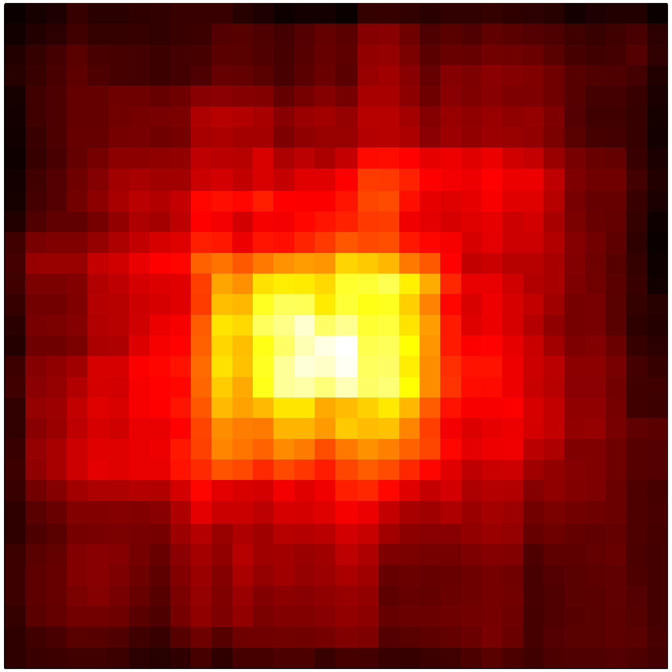
1. Find the weak-learner **h** that maximizes the **Youden's index**.
2. Calculate the error
$$\epsilon_t = \sum_{\substack{i=1 \\ c_t(\mathbf{x}_i, \mathbf{y}_i) \neq l_i}}^{n} w_{i,t}$$
3. Choose the **h** weight:
$$\alpha_t = \frac{1}{2} \ln \left(\frac{1-\epsilon_t}{\epsilon_t} \right)$$
4. Re-weight the samples:
$$w_{i,t+1} = w_{i,t} e^{-l_i \alpha_t c_t(\mathbf{x}_i, \mathbf{y}_i)}$$

Result of the training process

Selected measurement regions



heatmap of the weighted meassurement regions

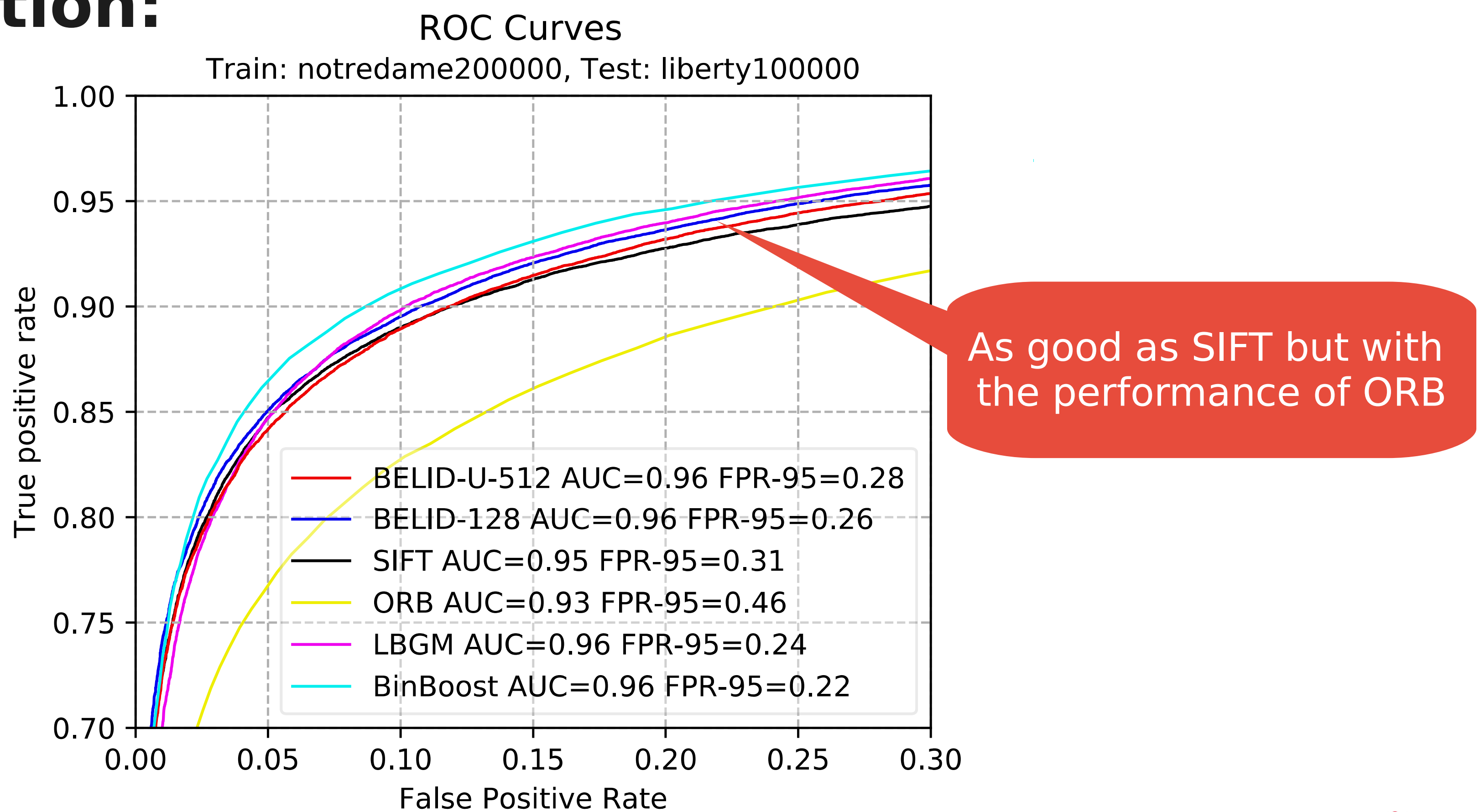


Training results for 512 weak-learners

Evaluation Results

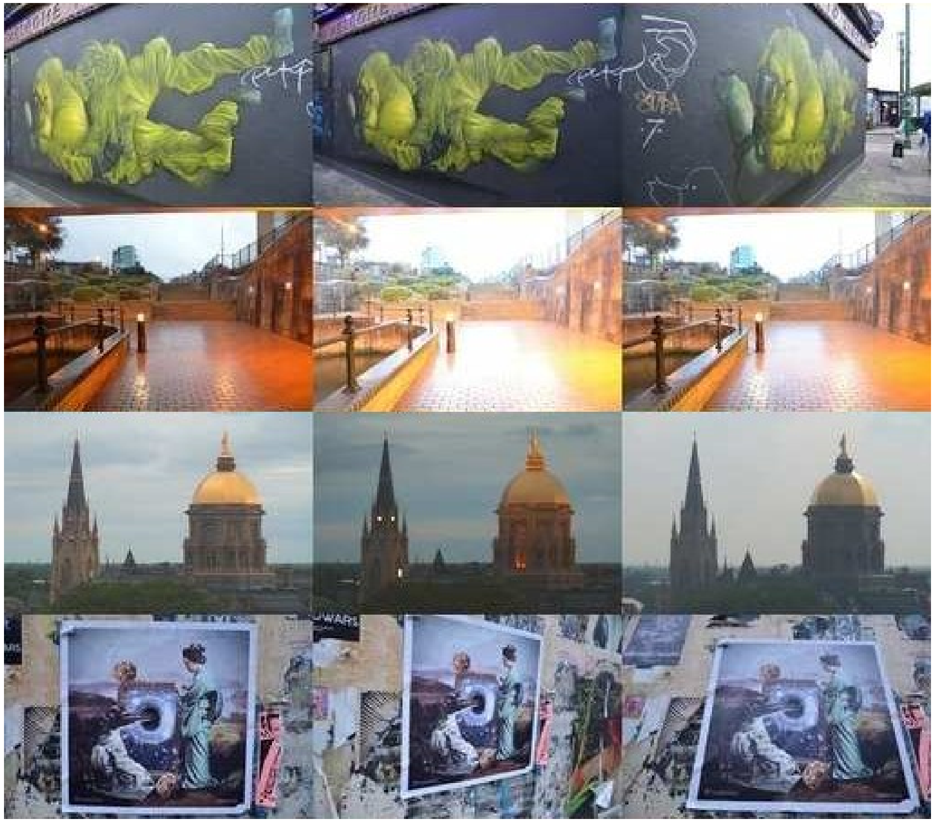
Evaluation Results: Brown Datasets

Evaluation results in the problem of binary classification:



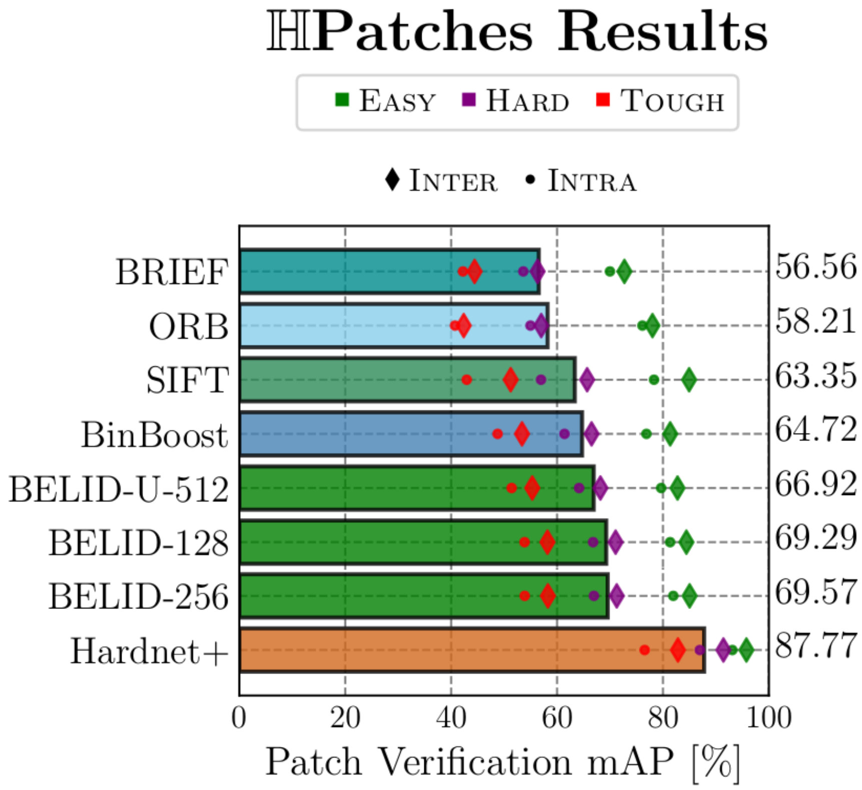
Evaluation Results: Hpatches

Hpatches is a multitask dataset of patches:



(Balntas, 2017)

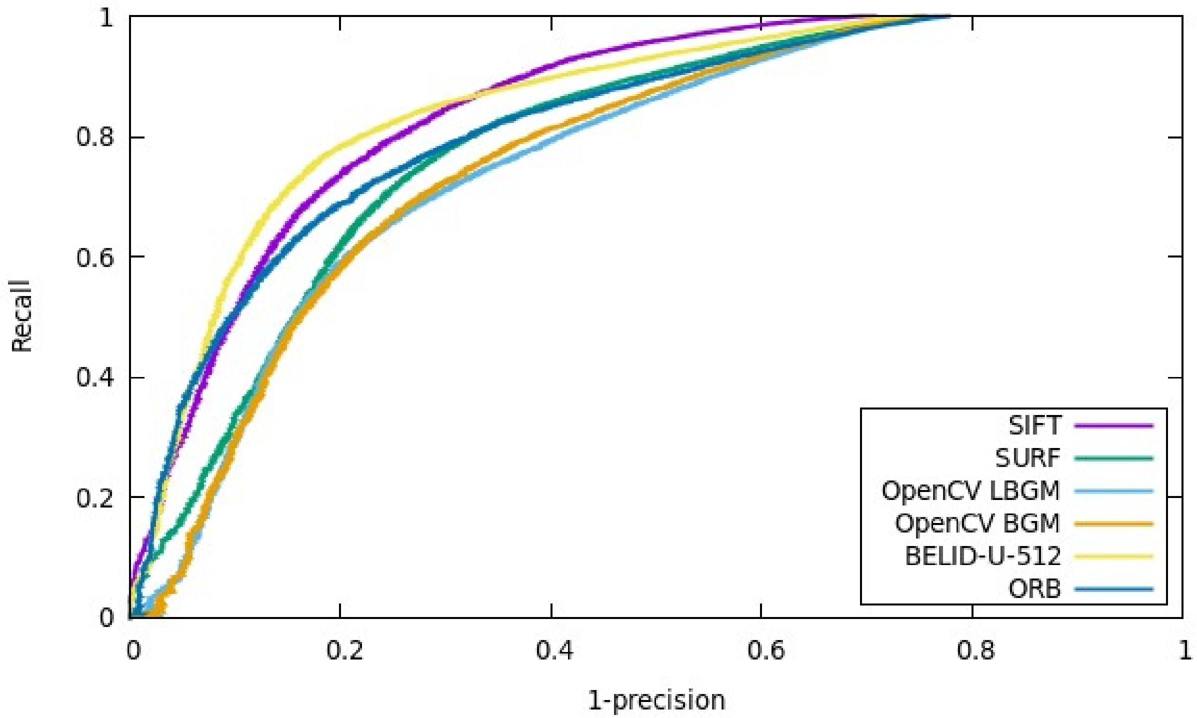
Results for the verification task:



Evaluation Results: Mikolajczyk Dataset

Mikolajczyk(Mikolajczyk, 2003) dataset is a set of 48 images widely used in feature matching.

Precision – Recall curve



Results (Times)

Times are quite important for us!

	Size	Intel Core i7	Exynox Octa
SIFT	128f	22.22 ms	163.2 ms
ORB	256b	0.44 ms	6.49 ms
LBGM	64f	19.77 ms	64.24 ms
BinBoost	256b	12.57 ms	42.39 ms
BELID-128	128f	3.08 ms	17.13 ms
BELID-U	512f	**0.41 ms**	**2.54 ms**

https://giphy.com/gifs/drone-vfZ7EwPyLnnRm

Implementation

- Training code →
- Production code → +
  - Parallel code compatible with `cv::Feature2D`
  - Only one integral image computed from the input image
  - Not rectified patches, measures computed using the integral image
  - Optimized matrix multiplication

Conclusions and Future Work

- **Conclusions**
  - BELID learns both, description pattern and scale
  - As accurate as SIFT, as fast as ORB!
  - BELID is a good compromise between speed and accuracy
- **Future work**
  - **Binarize** BELID
  - Train BELID for **unbalanced problems**
  - Integrate BELID in **high-level applications**

BELID: Boosted Efficient Local Image Descriptor

Iago Suárez, Ghesn Sfeir, José M. Buenaposada, Luis Baumela

Computer Vision and Aerial Robotics Group