Evaluation of Local Detectors and Descriptors for Fast Feature Matching

Liu TaiChen @imorpheus.ai && Stefan Haller

Friday 01/12/2017

12:00PM (GMT+8)

IMORPHEUS.AI

Outlines

- Introduction of features
- Methods
- Comparison
- References

IMORPHEUS.AI

Introduction of features

Image Features

***achievement*:**

- *detect* image features: *unique* interest *points*
- *describe* image feature: *sample region* of image patch around point

***applications*:**

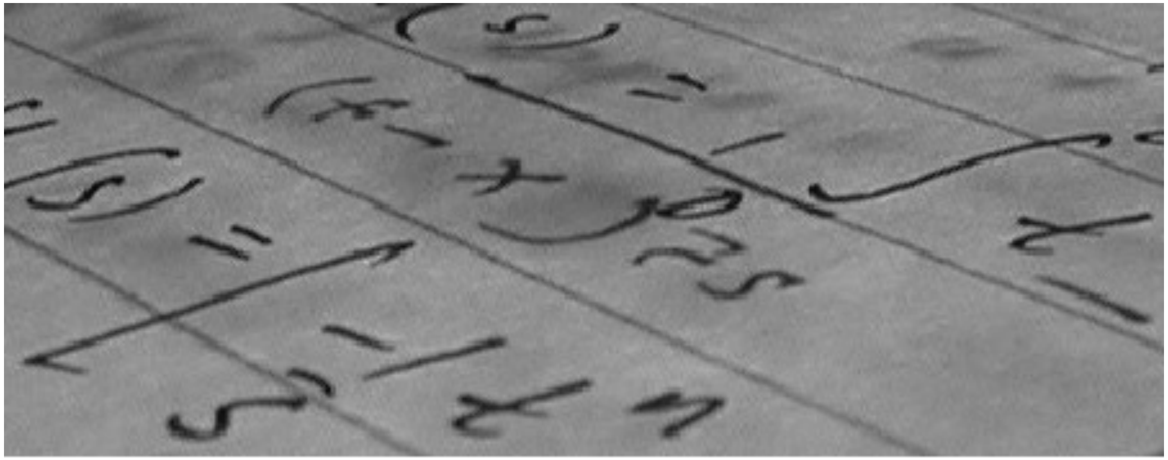
- object recognition and tracking
- image matching and stitching
- robotic mapping and 3D modeling

***keypoints*:**

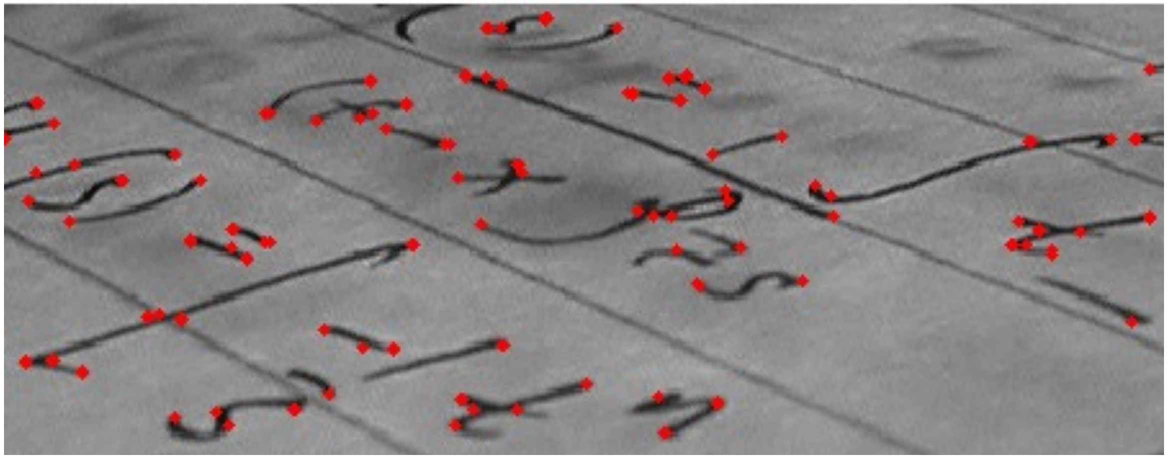
- repeatability and reliability
- good performance (description, matching)
- speed

Introduction of features

1st step: feature detection

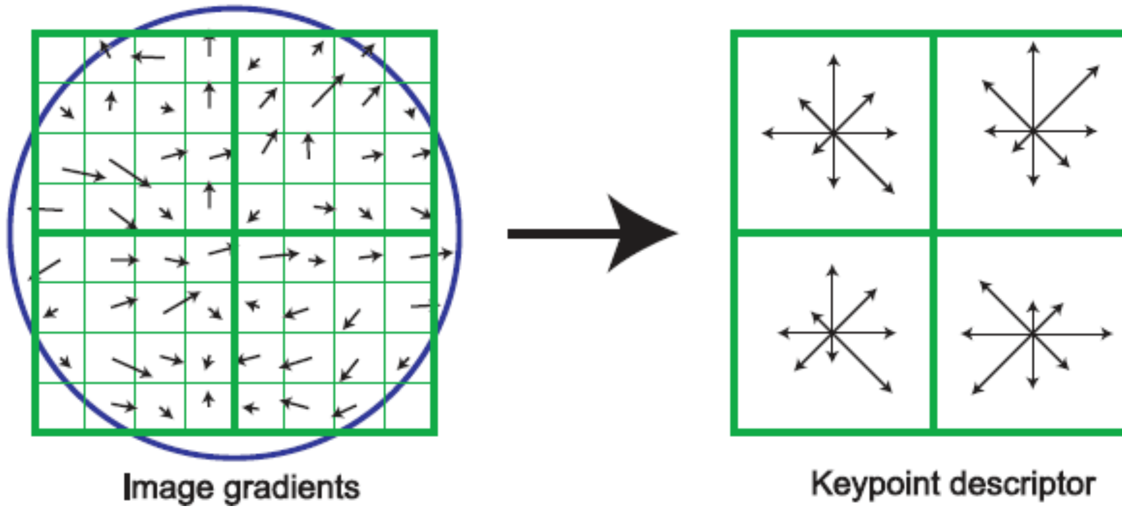


By Retardo-Wikimedia Commons



Introduction of features

2nd step: feature description



Bay, H., Tuytelaars, T., & Van Gool, L., 2006

Lowe, D. G., 2004

IMORPHEUS.AI

Introduction of features

History

Traditional (slower, accurate):

1999 **S**cale **I**nvariant **F**eature **T**ransform-SIFT

2006 **S**peeded **U**p **R**obust **F**eatures-SURF

Binary (faster, real time):

2010 **B**inary **R**obust **I**ndependent **E**lementary **F**eatures-BRIEF

2011 **O**riented FAST and **R**otated **B**RIEF-ORB

2011 **B**inary **R**obust **I**nvariant **S**calable **K**eypoints-BRISK

IMORPHEUS.AI

Outlines

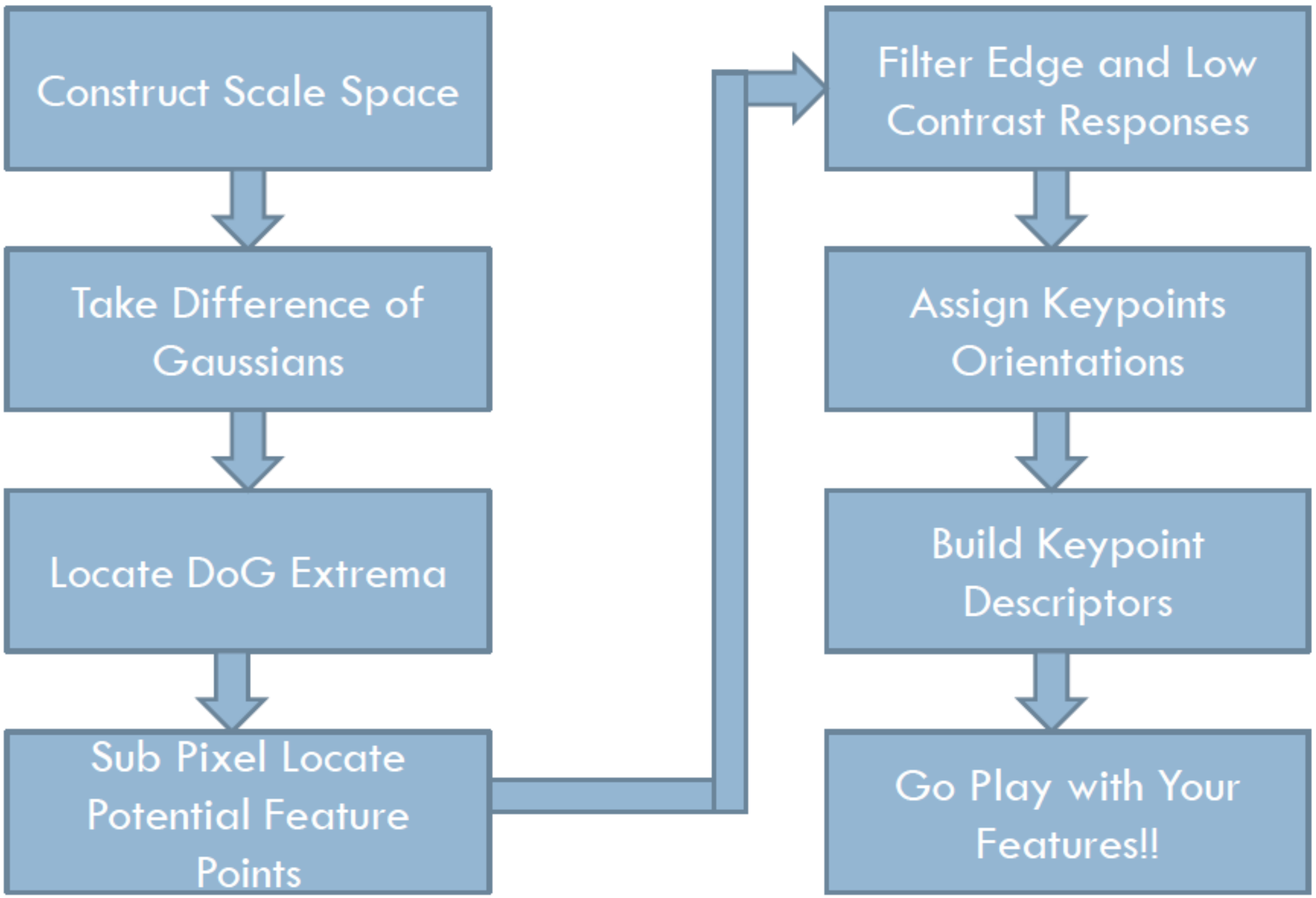
- Introduction of features
- Methods
- Comparison
- References

IMORPHEUS.AI

Scale **I**nvariant **F**eature **T**ransform-SIFT

Methods

SIFT



IMORPHEUS.AI

Methods

SIFT

1-Construct Scale Space

Gaussian kernel used to create scale space

-> Only possible scale space kernel (Lindberg)

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y),$$

where

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2}.$$

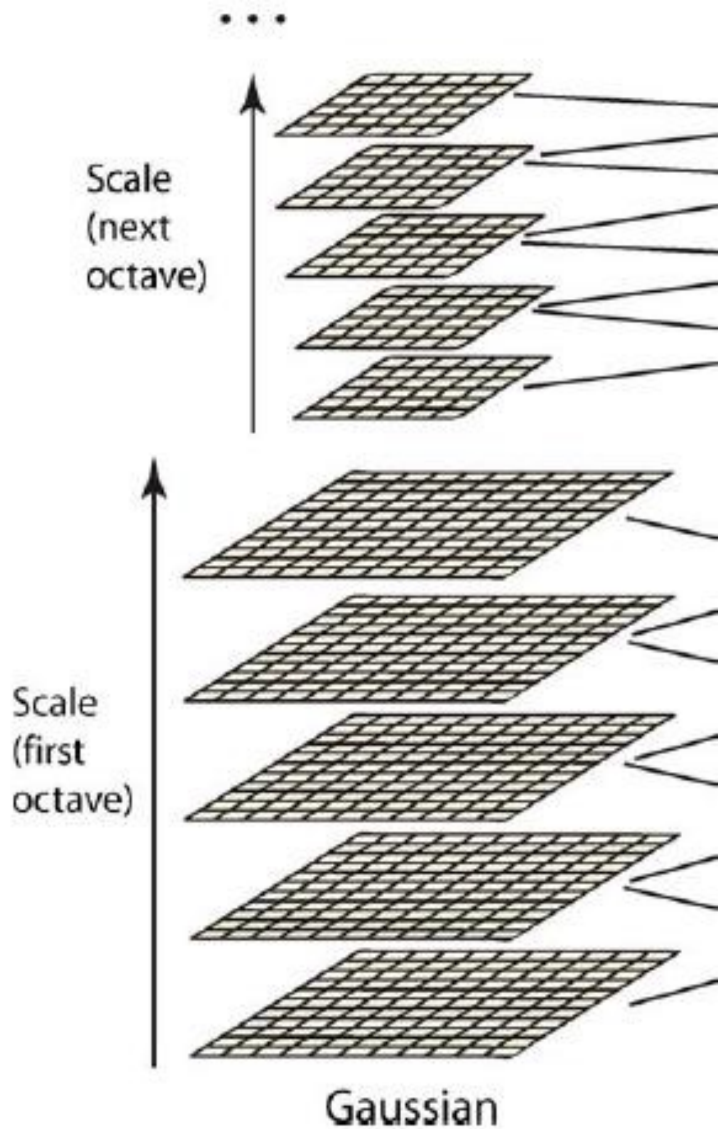
IMORPHEUS.AI

Methods

SIFT

1-Construct Scale Space

Image Pyramid



IMORPHEUS.AI

Methods

SIFT

2-Take Difference of Gaussians

Approximation of Laplacianof Gaussians

$$\sigma \nabla^2 G = \frac{\partial G}{\partial \sigma} \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma}$$

$$G(x, y, k\sigma) - G(x, y, \sigma) \approx (k - 1)\sigma^2 \nabla^2 G.$$

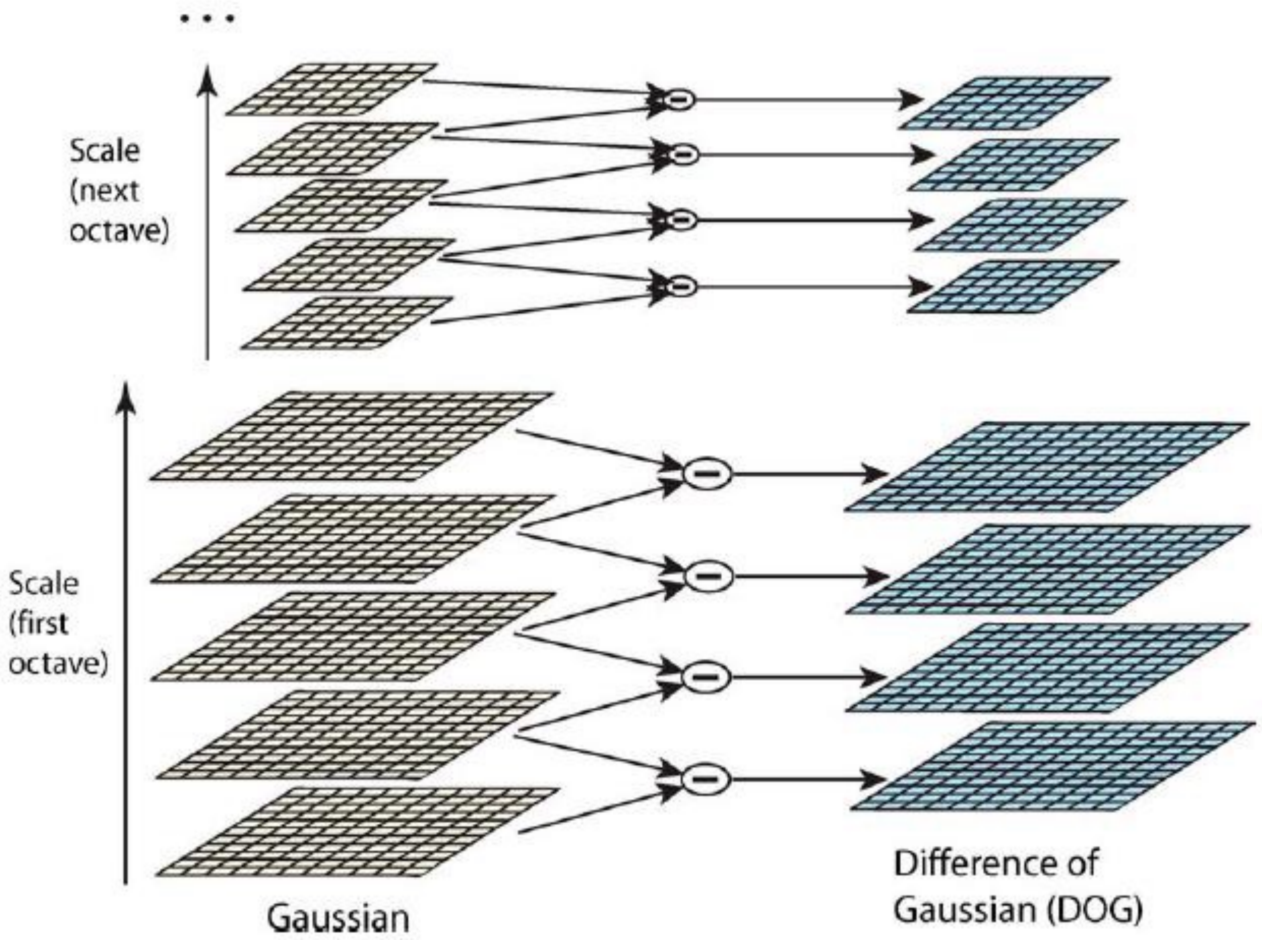
$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma). \end{aligned} \tag{1}$$

IMORPHEUS.AI

Methods

SIFT

2-Take Difference of Gaussians

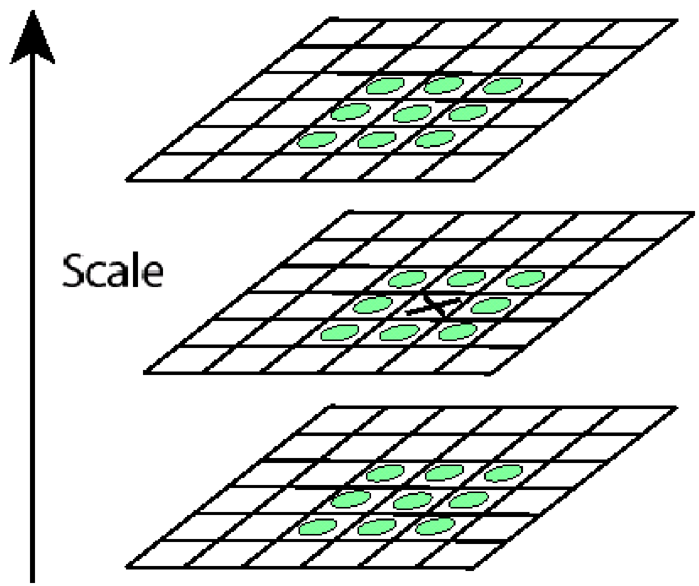


Methods

SIFT

3-Locate DoG Extrema

- Scan each DOG image
  - Look at all neighboring points (including scale)
  - Identify Min and Max
    - 26 Comparisons



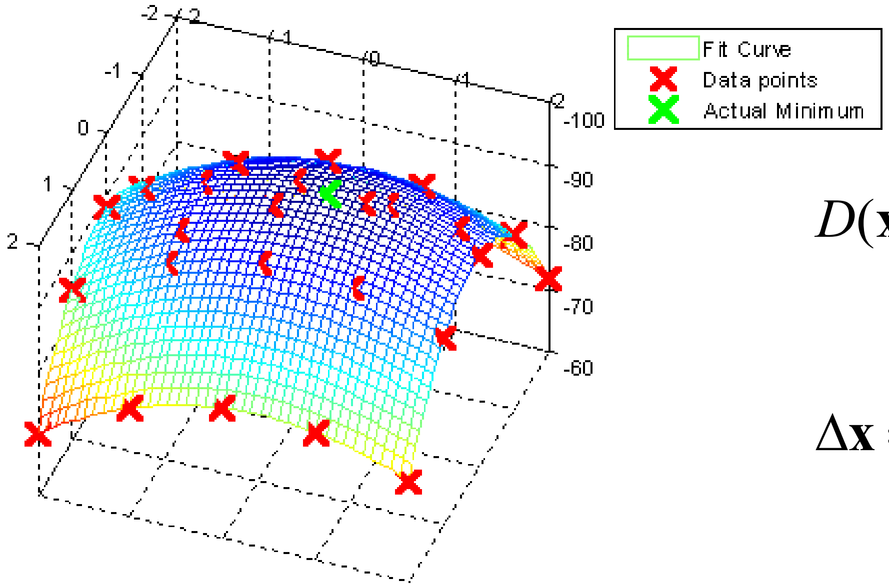
IMORPHEUS.AI

Methods

SIFT

4-Quit

4.1-Sub Pixel Locate Potential Feature Points



$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \Delta\mathbf{x} + \frac{1}{2} \Delta\mathbf{x}^T \frac{\partial^2 D^T}{\partial \mathbf{x}^2} \Delta\mathbf{x}$$

$$\Delta\mathbf{x} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D(\mathbf{x})}{\partial \mathbf{x}}$$

$$D(\hat{\mathbf{x}}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \mathbf{x}} \hat{\mathbf{x}} \quad \&\& \quad |D(\hat{\mathbf{x}})| \geq T$$

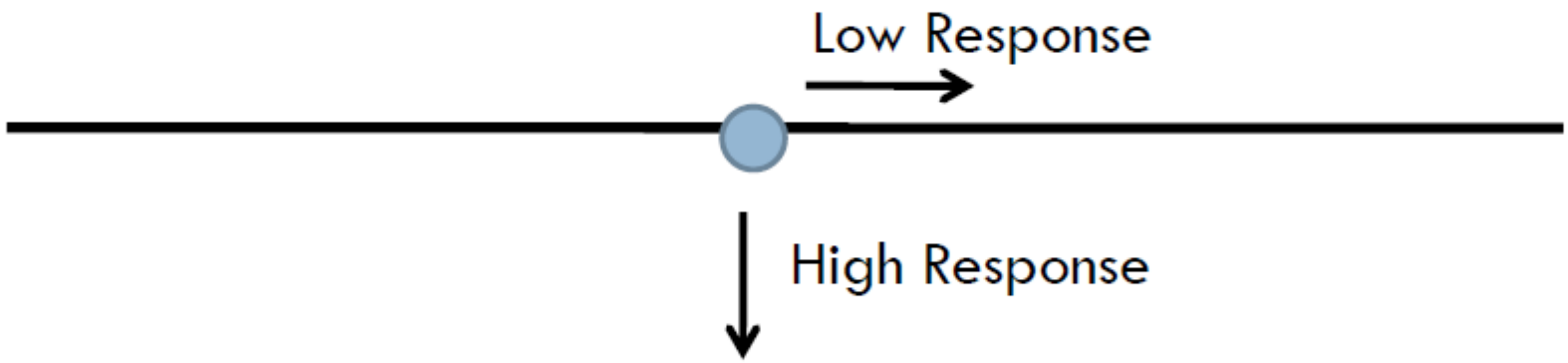
IMORPHEUS.AI

Methods

SIFT

4-Quit

4.2-Filter Edge and Low Contrast Responses



$$H = \begin{bmatrix} D_{xx} & D_{yx} \\ D_{xy} & D_{yy} \end{bmatrix} \qquad \frac{Tr(H)^2}{Det(H)} > \frac{(r+1)^2}{r}$$

IMORPHEUS.AI

Methods

SIFT

5-Assign Keypoints Orientations

Compute Gradient for each blurred image

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}$$
$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1))/(L(x+1, y) - L(x-1, y)))$$

For region around keypoint

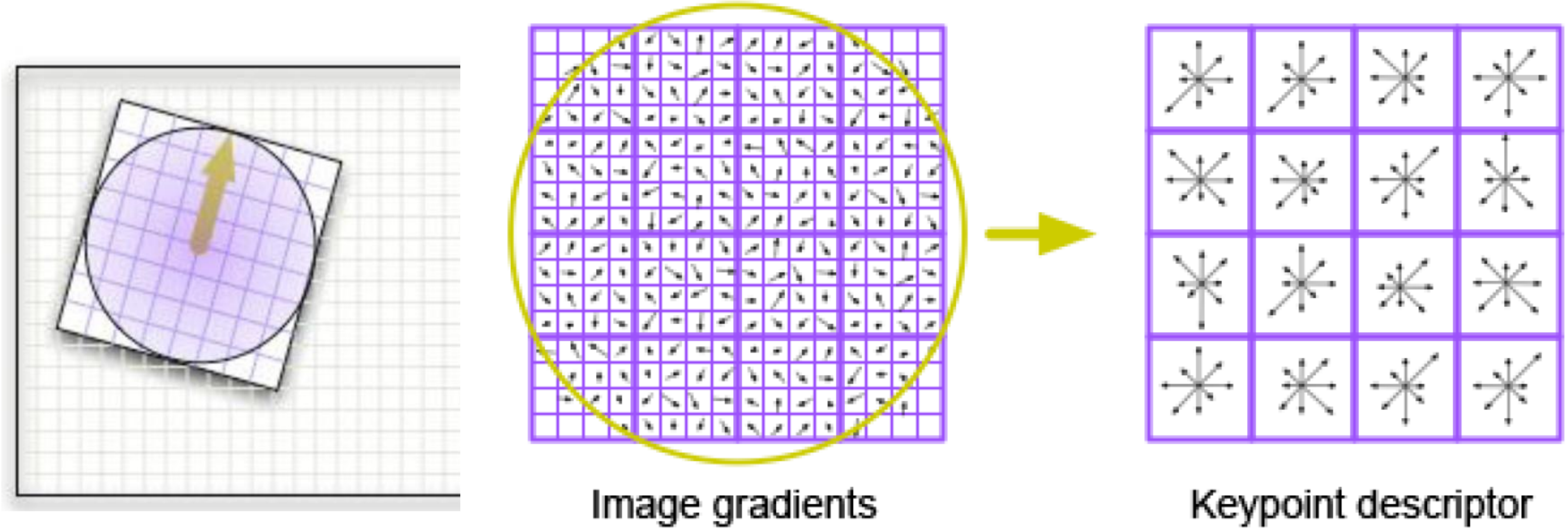
- Create Histogram with 36 bins for orientation
- Weight each point with Gaussian window of 1.5σ
- Create key point for all peaks with value>=.8 max bin

IMORPHEUS.AI

Methods

SIFT

6-Build Keypoint Descriptors



Actual implementation uses 4x4 descriptors from 16x16 which leads to a 4x4x8=128 element vector

IMORPHEUS.AI

Speeded **U**p **R**obust **F**eatures-SURF

Methods

SURF vs SIFT

Using laplacian of Gaussian, one could obtain scale invariant features.

SIFT uses **difference of Gaussian** to approximate laplacian of Gaussian.

SURF uses **Hessian-laplacian** to approximate laplacian of Gaussian.

Methods

SURF-detection

$$H(\underbrace{x, y}_{\text{position}}, \underbrace{\delta}_{\text{scale}}) = \begin{bmatrix} L_{xx}(x, y, \delta) & L_{yx}(x, y, \delta) \\ L_{xy}(x, y, \delta) & L_{yy}(x, y, \delta) \end{bmatrix}$$

$$\det(H_{approx}) = D_{xx} D_{yy} - (\omega D_{xy})^2$$

Box filter

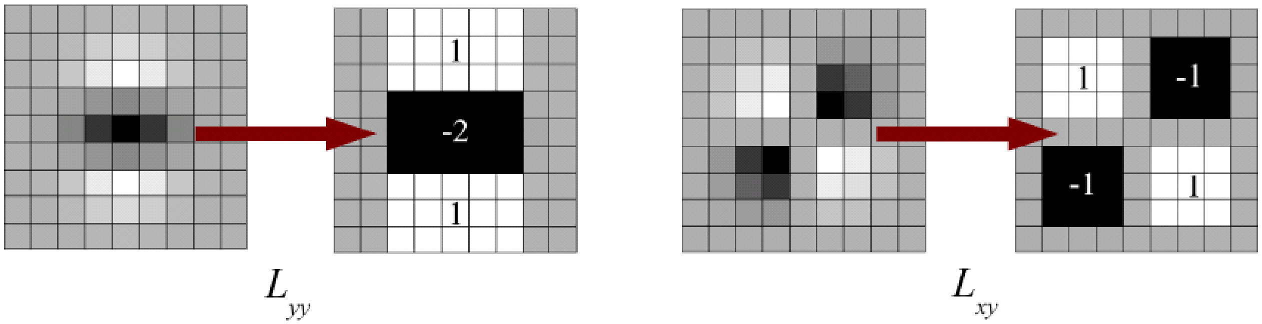
Integral Images

IMORPHEUS.AI

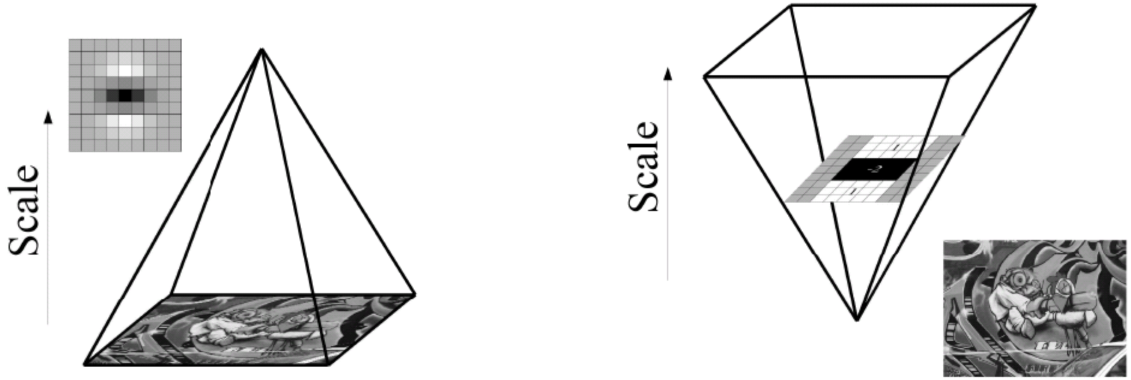
Methods

SURF-detection

Approximated second order derivatives with box filters.



Scale analysis with constant image size.



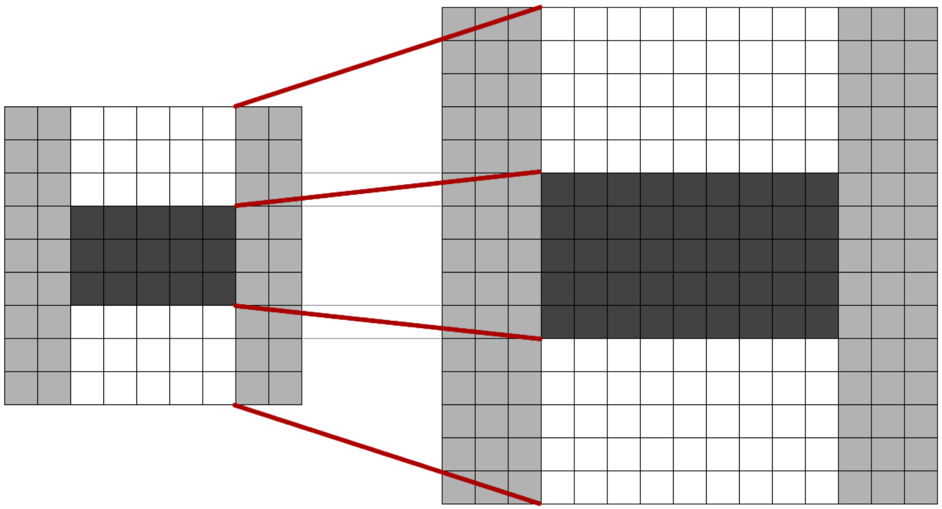
IMORPHEUS.AI

SIFT->Image Size

SURF->Filter Size

Methods

SURF-detection



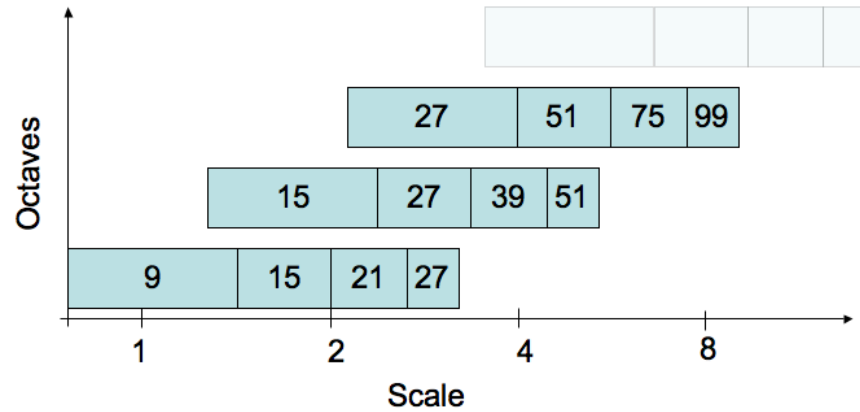
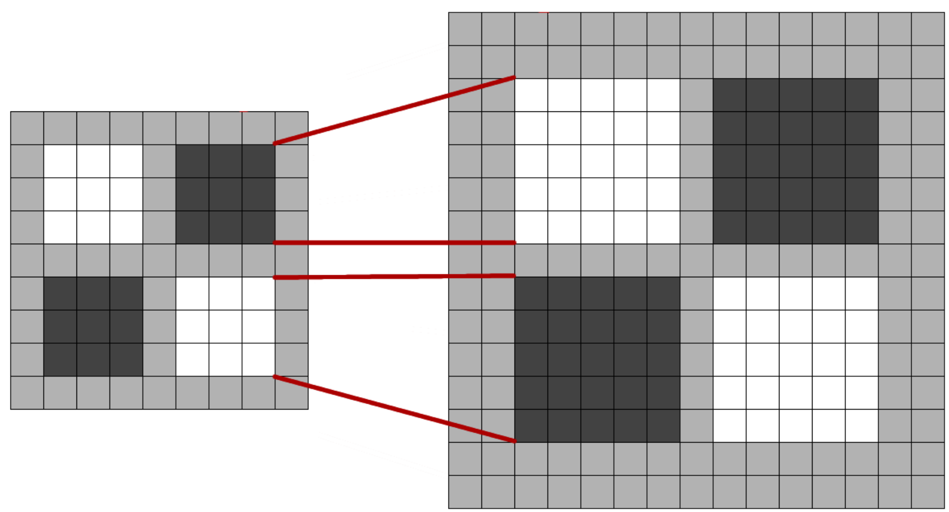


Fig. 5. Filters D_{yy} (top) and D_{xy} (bottom) for two successive scale levels (9×9 and 15×15). The length of the dark lobe can only be increased by an even number of pixels in order to guarantee the presence of a central pixel (top).



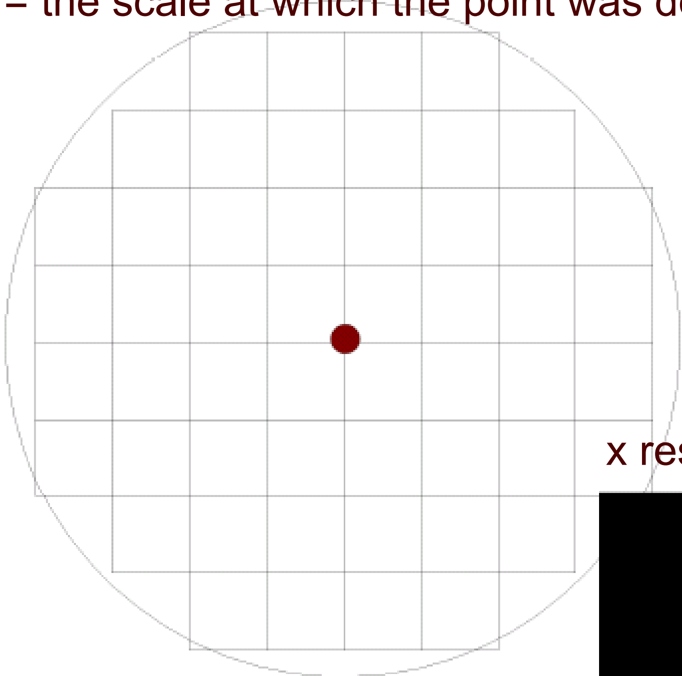
IMORPHEUS.AI

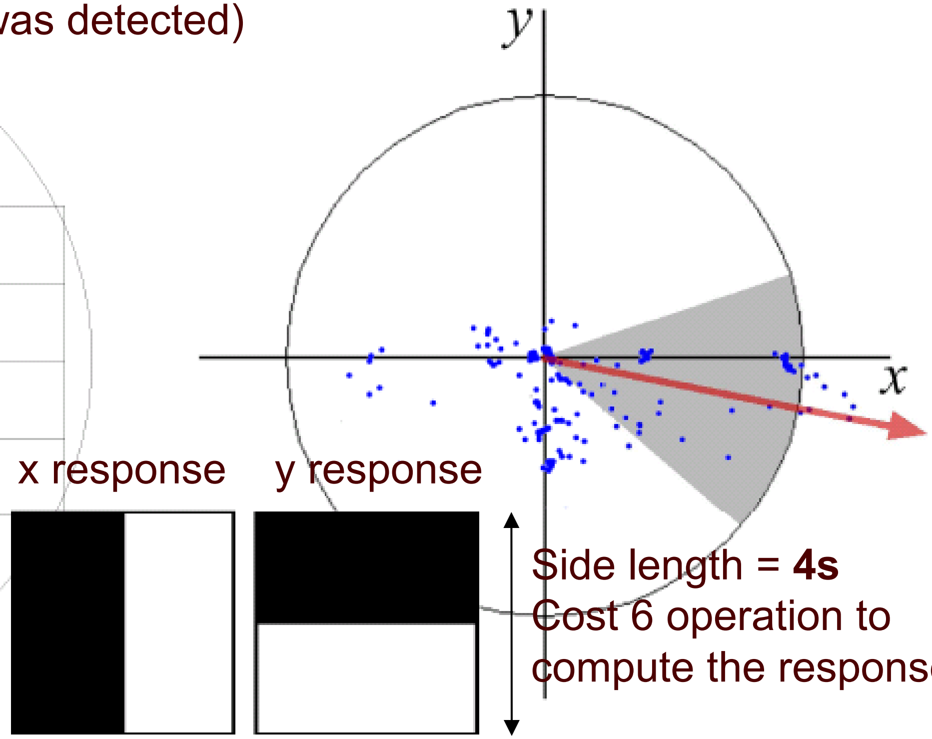
Methods

SURF-**Description**

Orientation Assignment-**Haar wavelet response**

Circular neighborhood of
radius **6s** around the interest point
(**s** = the scale at which the point was detected)





Side length = **4s**
Cost 6 operation to
compute the response

IMORPHEUS.AI

Methods

SURF-Description

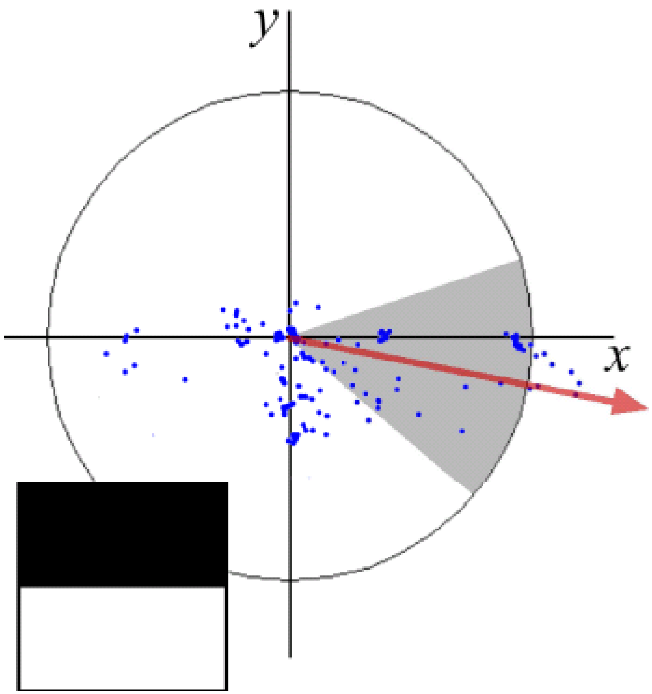
Dominant orientation

•The Haar wavelet responses are represented as vectors

•Sum all responses within a sliding orientation window covering an angle of 60 degree

•The two summed response yield a new vector

•The longest vector is the dominant orientation



IMORPHEUS.AI

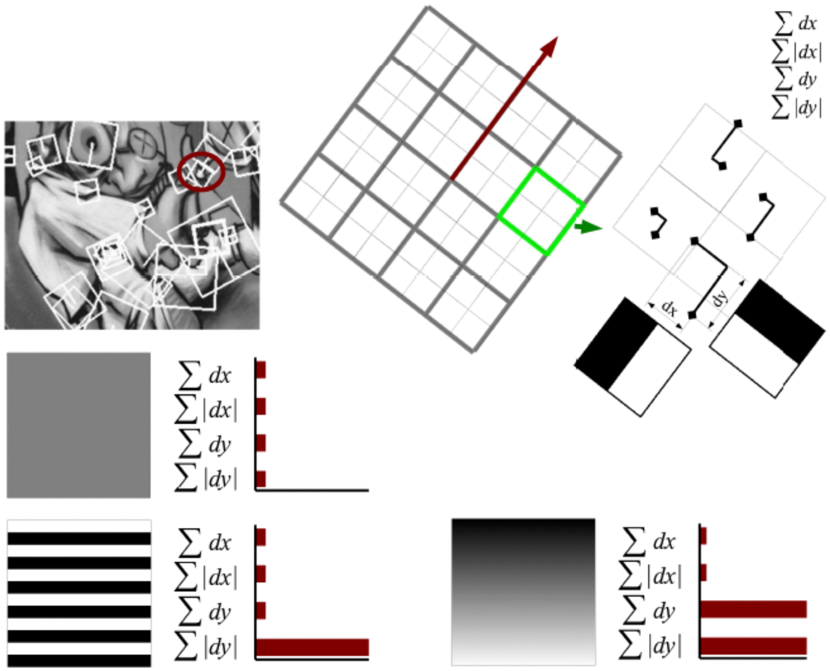
Methods

SURF-Description

•Split the interest region (20s x 20s) up into 4 x 4 square sub-regions.

•Calculate Haar wavelet response d_x and d_y and weight the response with a Gaussian kernel.

•Sum the response over each sub-region for d_x and d_y, then sum the absolute value of resp-onse.



•Normalize the vector into unit length

IMORPHEUS.AI

Oriented FAST and **R**otated **B**RIEF-ORB

Methods

ORB-detection

1. oFAST detector+ rBRIEF description

    1. Oriented FAST

    2. Rotated RBIEF

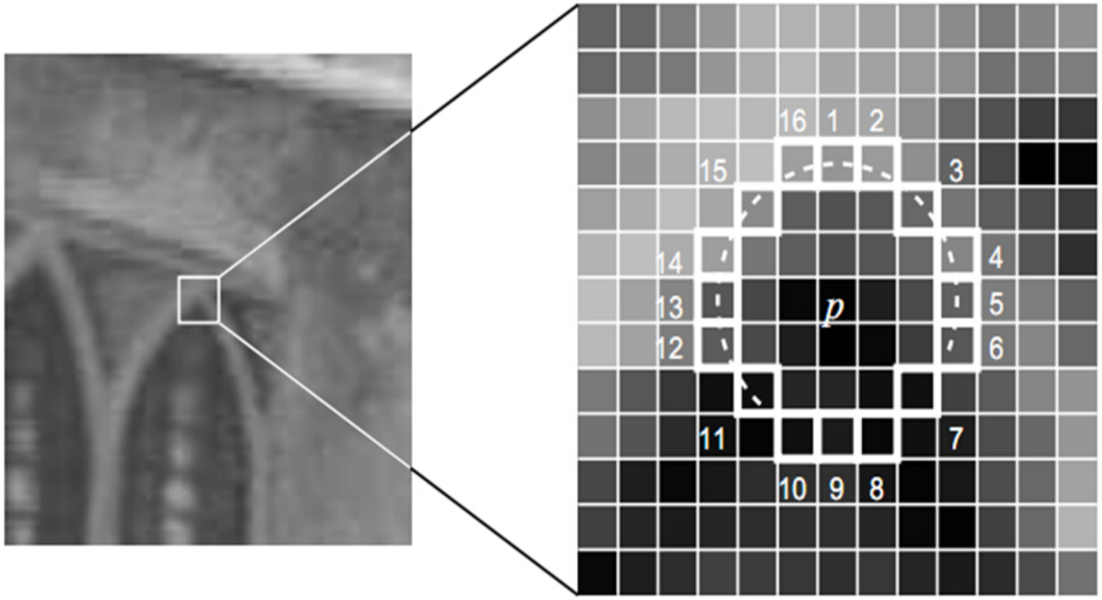
IMORPHEUS.AI

Methods

ORB-detection

1. oFAST detector

FAST detector



$$N = \sum_{x \forall (circle(p))} \left| I(x) - I(p) \right| > \varepsilon_d$$

IMORPHEUS.AI

Methods

ORB-detection

1. oFAST detector

shortcomings of FAST:

a) no quality measure ("cornerness")

solution: use Harris cornerness measure

b) not scale invariant

solution: use scale pyramid (like SIFT)

c) not rotation invariant

solution: calculate intensity centroid

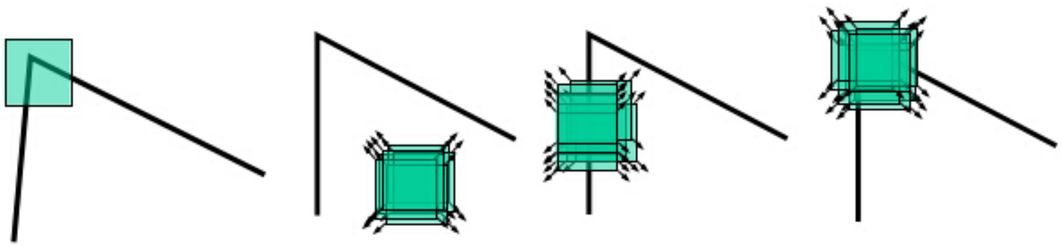
IMORPHEUS.AI

Methods

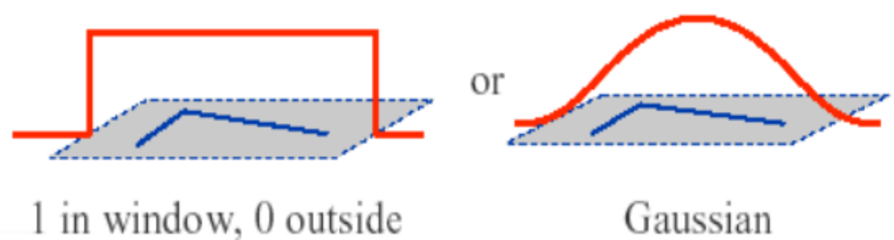
ORB-detection

1. oFAST detector

Harris cornerness measure



$$E(u,v) = \sum_{x,y} w(x,y)[I(x+u,y+v) - I(x,y)]^2 \qquad w(x,y) -> rectangle, \quad gaussian\ window$$



Methods

ORB-detection

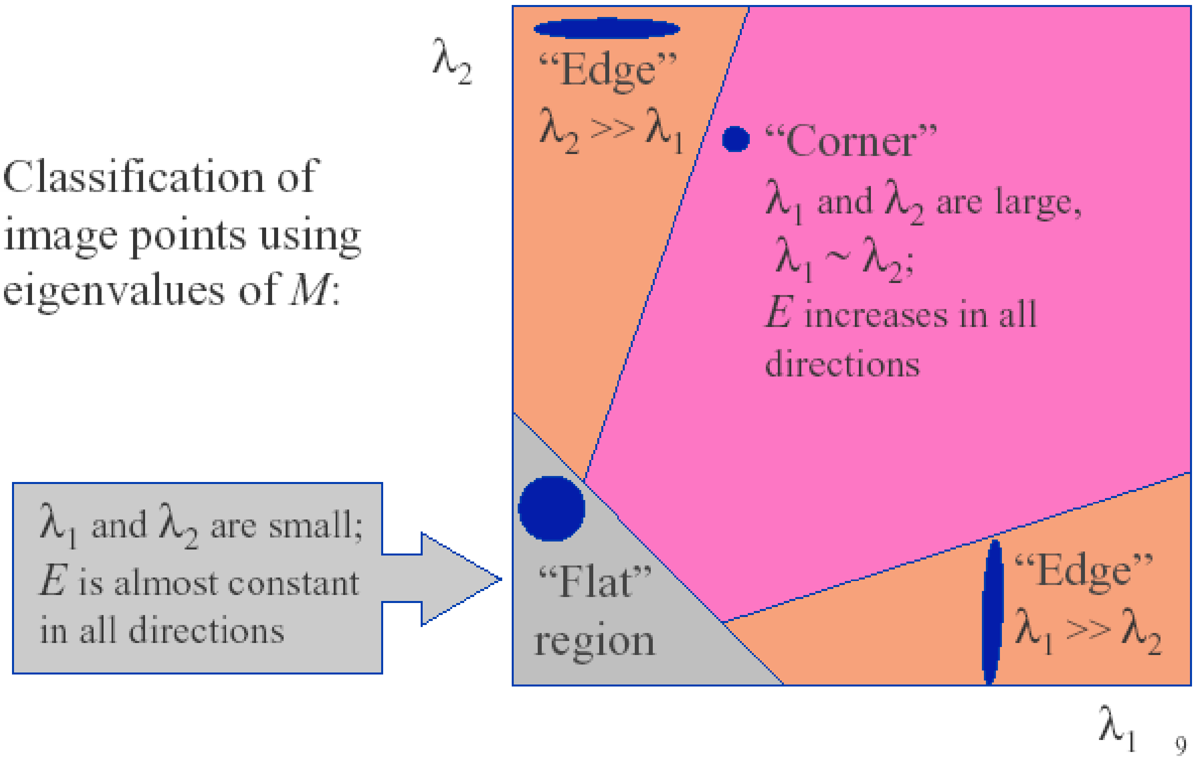
1. oFAST detector

$$E(u,v) \cong [u,v] M \begin{bmatrix} u \\ v \end{bmatrix}, \; M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

Harris cornerness measure

$$R = Det(M) - k * trace(M) * trace(M)$$

Classification of image points using eigenvalues of M:



Methods

ORB-detection

1. oFAST detector

assumption: corner's intensity is offset from center

$$\forall \underbrace{p, q \in \{0,1\}}_{\substack{\text{binary selector}\\ \text{for x and y}\\ \text{direction}}} : \qquad m_{pq} = \sum_{\underbrace{x,y}_{\substack{\text{circular}\\ \text{window}}}} \underbrace{x^p \quad y^q}_{\substack{\text{weighted}\\ \text{by}\\ \text{coordinate}}} \underbrace{I(x,y)}_{\substack{\text{image}\\ \text{function}}}$$

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right)$$

$$\text{dominant direction } \overrightarrow{OC}: \quad \theta = \operatorname{atan2}(m_{01}, m_{10})$$

IMORPHEUS.AI

Methods

ORB-descriptor

1. rBRIEF

Binary test

$$\tau(\mathbf{p};\mathbf{x},\mathbf{y}) = \begin{cases} 1 & \textit{if} \;\; \mathbf{p}(\mathbf{x}) < \mathbf{p}(\mathbf{y}) \\ 0 & \textit{otherwise} \end{cases}$$

BRIEF descriptor

$$f_{n_d}(p) := \sum_{1 \le i \le n_d} 2^{i-1} \tau(\mathbf{p};\mathbf{x},\mathbf{y})$$

For each S*S patch

1. Smooth it
2. Pick pixels using pre-defined binary tests
3. P->distribution function

IMORPHEUS.AI

Methods

ORB-descriptor

1. rBRIEF

5 different distribution functions

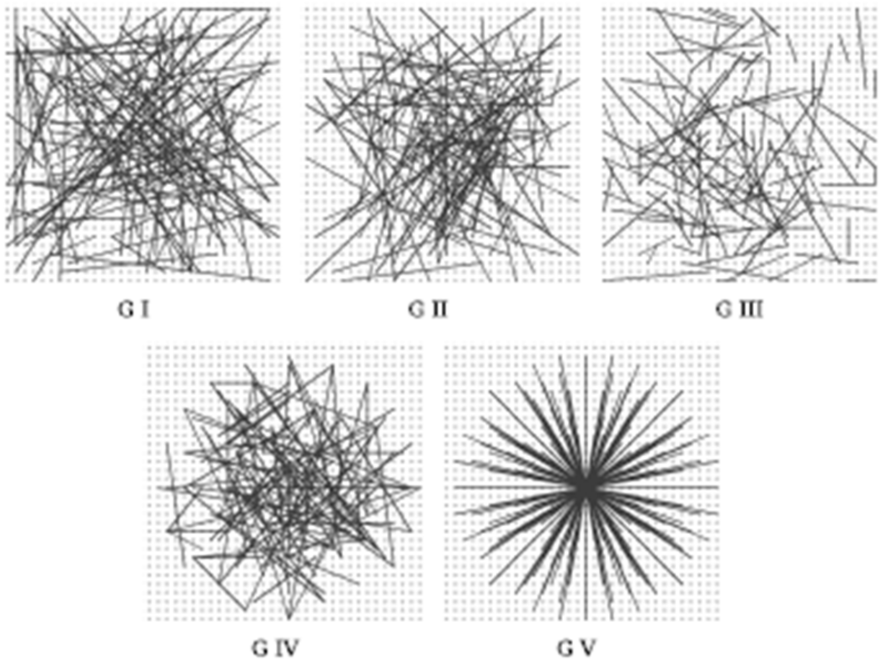


Fig. 2. Different approaches to choosing the test locations. All except the righmost one are selected by random sampling. Showing 128 tests in every image.

IMORPHEUS.AI

Methods

ORB-descriptor

1. rBRIEF

Steered BRIEF

$$\mathbf{S} = \begin{pmatrix} x_1, \cdots, x_n \\ y_1, \cdots, y_n \end{pmatrix}$$

$$\mathbf{S}_\theta = \mathbf{R}_\theta \mathbf{S}$$

We discretize the angle to increments of $2\pi/30$ (12 degrees), and construct a lookup table of precomputed BRIEF patterns. As long at the keypoint orientation θ is consistent across views, the correct set of points S_θ will be used to compute its descriptor.

$$g_n(\mathbf{p}, \theta) := f_n(\mathbf{p}) \mid (x_i, y_i) \in \mathbf{S}_\theta$$

IMORPHEUS.AI

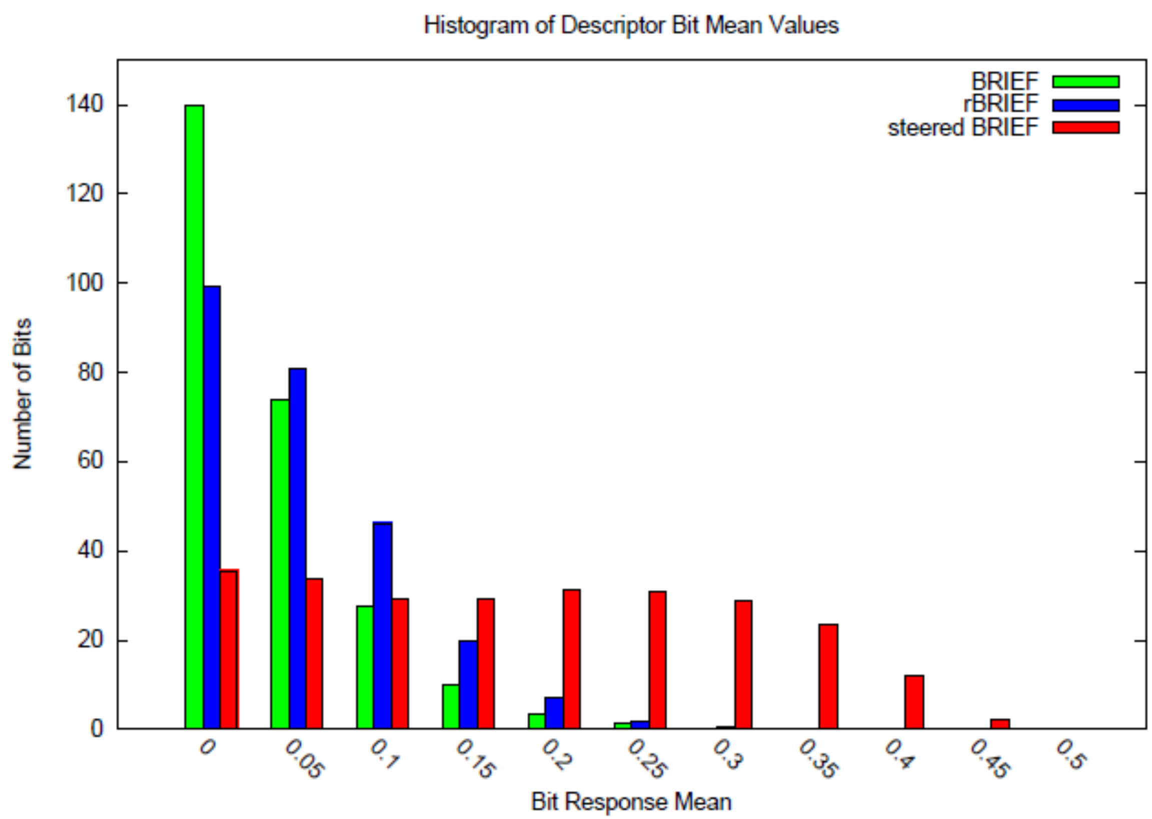
Methods

ORB-descriptor

1. rBRIEF

Steered BRIEF

Related in different S-BRIEF descriptors



IMORPHEUS.AI

Methods

ORB-descriptor

1. rBRIEF

1. Run each test against all training patches.
2. Order the tests by their distance from a mean of 0.5, forming the vector T.
3. Greedy search:
   (a) Put the first test into the result vector R and remove it from T.
   (b) Take the next test from T, and compare it against all tests in R. If its absolute correlation is greater than a threshold, discard it; else add it to R.
   (c) Repeat the previous step until there are 256 tests in R. If there are fewer than 256, raise the threshold and try again.

Binary **R**obust **I**nvariant **S**calable **K**eypoints

-

BRISK

Methods

BRISK-detector

Scale Space Keypoint Detection

N octaves c_i

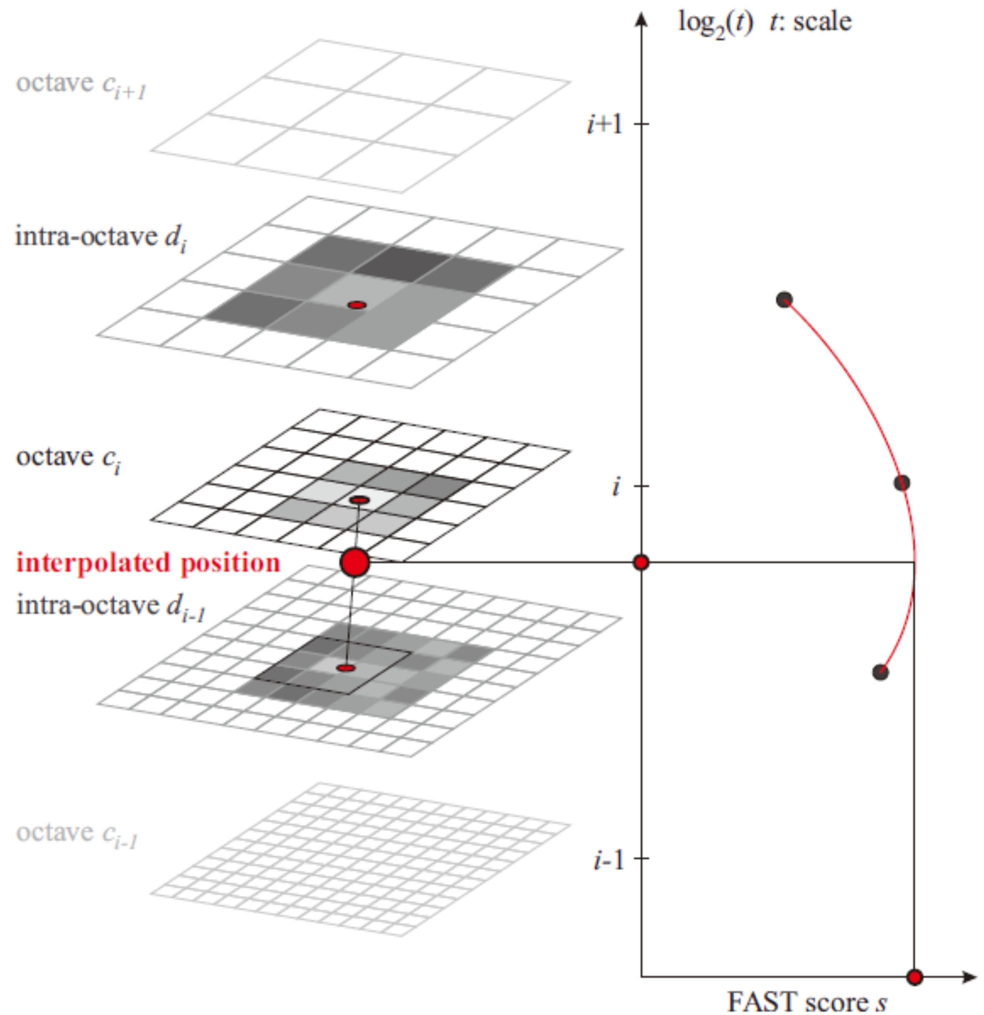
$t(c_i) = 2^i$

N intra-octaves d_i

$t(d_i) = 2^i \cdot 1.5$

typically n = 4.

Fast detector is applied.



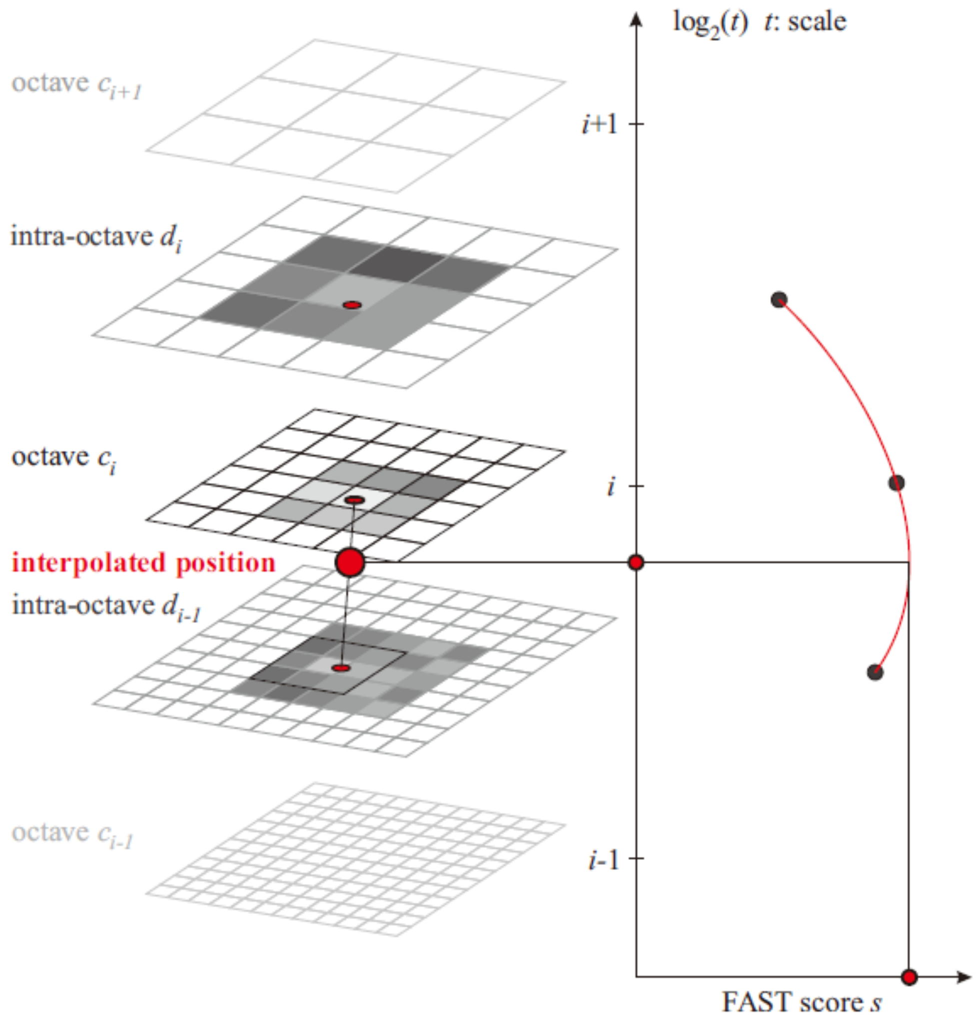
Methods

BRISK-detector

Scale Space Keypoint Detection

Img	Height-h	Width-w
c_0	H	W
d_0	2/3*H	2/3*W
c_1	1/2*H	1/2*W
d_1	1/3*H	1/3*W
c_2	1/4*H	1/4*W
d_2	1/6*H	1/6*W
c_3	1/8*H	1/8*W
d_3	1/12*H	1/12*H

IMORPHEUS.AI



Methods

BRISK-detector

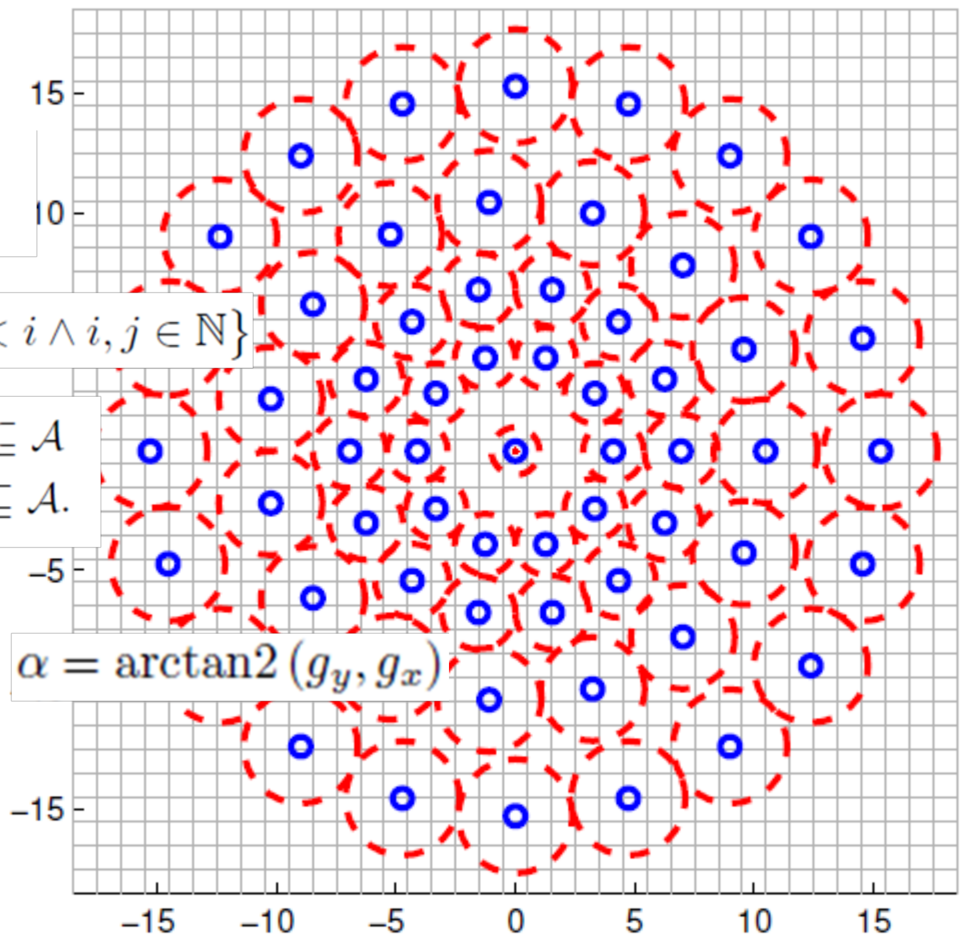
Scale Space Keypoint Detection

$$\mathbf{g}(\mathbf{p}_i, \mathbf{p}_j) = (\mathbf{p}_j - \mathbf{p}_i) \cdot \frac{I(\mathbf{p}_j, \sigma_j) - I(\mathbf{p}_i, \sigma_i)}{\|\mathbf{p}_j - \mathbf{p}_i\|^2}.$$

$$\mathcal{A} = \left\{(\mathbf{p}_i, \mathbf{p}_j) \in \mathbb{R}^2 \times \mathbb{R}^2 \mid i < N \wedge j < i \wedge i, j \in \mathbb{N}\right\}$$

$$\mathcal{S} = \{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{A} \mid \|\mathbf{p}_j - \mathbf{p}_i\| < \delta_{max}\} \subseteq \mathcal{A}$$
$$\mathcal{L} = \{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{A} \mid \|\mathbf{p}_j - \mathbf{p}_i\| > \delta_{min}\} \subseteq \mathcal{A}.$$

$$\mathbf{g} = \begin{pmatrix} g_x \\ g_y \end{pmatrix} = \frac{1}{L} \cdot \sum_{(\mathbf{p}_i, \mathbf{p}_j) \in \mathcal{L}} \mathbf{g}(\mathbf{p}_i, \mathbf{p}_j)$$



$$\alpha = \arctan2\left(g_y, g_x\right)$$

IMORPHEUS.AI

Methods

BRISK-description

BRIEF

$$b = \begin{cases} 1, & I(\mathbf{p}_j^\alpha, \sigma_j) > I(\mathbf{p}_i^\alpha, \sigma_i) \\ 0, & \text{otherwise} \end{cases}$$

$$\forall (\mathbf{p}_i^\alpha, \mathbf{p}_j^\alpha) \in \mathcal{S}$$

required to be locally consistent. With the sampling pattern and the distance thresholds as shown above, we obtain a bit-string of length 512. The bit-string of BRIEF64 also contains 512 bits, thus the matching for a descriptor pair will be performed equally fast by definition.

IMORPHEUS.AI

Methods

SUMMARY

SIFT

SURF

Construct scale space

1. Gaussian Kernel
2. Image Pyramid
3. Difference of Gaussian

Direction of detector

a) Gradient based;
b) Haar wavelet based;

Description

a) Gradient based->byte unit && Euclidean Distance;

IMORPHEUS.AI

Methods

SUMMARY

ORB

BRISK

Construct scale space

1. Gaussian Kernel
2. Image Pyramid

Detector

a) FAST detector based;
b) Calculation of direction;

Description

a) BRIEF based-> bit unit&&Hamming Distance;

IMORPHEUS.AI

Outlines

- Introduction of features
- Methods
- Comparison
- References

IMORPHEUS.AI

Comparison

Implementation Details

- matching: nearest neighborhood search (NN)
  1. kd-trees (geometric, partition of dimensions) for real-valued descriptors
  2. hashing (project similar data into same bucket, Hamming space) for binary descriptors
- ϵ-ANN: approximated NN

- image data set: standard graffiti image sequences

- 24 × 3.47 GHz, 12 MB cache, Intel Xeon X5690, 99 GB RAM, x64 Ubuntu 10.04

- SSE 4.2 enabled (fast POPCNT)

- algorithms used with default parameters (except FAST)

Comparison

Table 1. Averaged computation times for the different detectors

Detector	Run time [ms.]	Speed-up [-]	# keypoints
SURF	176	1.9	2 911
DoG	338	1.0	1 552
FAST	**2**	169.0	**5 158**
STAR	17	19.9	849
MSER	60	5.6	483
BRISK	**10**	33.8	1 874
ORB	**7**	48.3	594

Miksik, O., & Mikolajczyk, K. (2012, November). Evaluation of local detectors and descriptors for fast feature matching. In Pattern Recognition (ICPR), 2012 21st International Conference on (pp. 2681-2684). IEEE.

Comparison

Table 2. Computation times for the different descriptors for 1000 SURF keypoints

Descriptor	Run time [ms.]	Speed-up [-]
SURF	117.1	3.83
SIFT	448.6	1.00
BRIEF	**3.8**	118.05
BRISK	**10.6**	42.32
ORB	**4.2**	106.80
LIOP	1 801.1	0.25
MROGH	2 976.8	0.15
MRRID	5 625.1	0.08

Miksik, O., & Mikolajczyk, K. (2012, November). Evaluation of local detectors and descriptors for fast feature matching. In Pattern Recognition (ICPR), 2012 21st International Conference on (pp. 2681-2684). IEEE.

Comparison

Table 3. Speed-up over the sequential matching of SURF-SURF for the different descriptors and $N = 40$, $\epsilon = 3$

Descriptor	Size [bytes]	Time [ms.]	Speed-up (S_T)
SURF	64	390	859.4
SIFT	128	2 095	160.1
BRIEF	32	**370**	**905.9**
BRISK	64	524	640.1
ORB	32	**370**	**905.9**
LIOP	144	3 466	96.8
MROGH	192	15 317	21.9
MRRID	256	7 363	45.6

Miksik, O., & Mikolajczyk, K. (2012, November). Evaluation of local detectors and descriptors for fast feature matching. In Pattern Recognition (ICPR), 2012 21st International Conference on (pp. 2681-2684). IEEE.

Comparison

Performance Measures

$$recall = \frac{true\ \ positive}{true\ \ positive + false\ negative}$$

$$precision = \frac{true\ \ positive}{true\ \ positive + false\ positive}$$

Comparison

Performance Measures

Table 4. Precision/Recall for the different descriptors and $N = 40$, $\epsilon = 3$

Detector	Descriptor	Precision	Recall	MAP
SURF	SURF	0.485	0.513	0.334
SURF	SIFT	0.525	0.533	0.491
SURF	BRIEF	0.517	0.546	0.514
SURF	ORB	0.448	0.470	0.437
SURF	LIOP	0.581	0.597	0.568
SURF	MROGH	0.540	0.567	0.527
SURF	MRRID	0.550	0.569	0.510
SURF	BRISK	0.536	0.553	0.530
BRISK	BRISK	0.504	0.527	0.492
ORB	ORB	0.493	0.495	0.463
FAST	SIFT	0.366	0.376	0.336

Miksik, O., & Mikolajczyk, K. (2012, November). Evaluation of local detectors and descriptors for fast feature matching. In Pattern Recognition (ICPR), 2012 21st International Conference on (pp. 2681-2684). IEEE.

Outlines

- Introduction of features
- Methods
- Comparison
- References

IMORPHEUS.AI

References

Lowe, D. G. (2004). Distinctive image features from scale-invariant key-points. International journal of computer vision, 60(2), 91-110.

Rosten, E., & Drummond, T. (2006). Machine learning for high-speed corner detection. In Computer Vision–ECCV 2006 (pp. 430-443). Springer Berlin Heidelberg.

Juan, L., & Gwun, O. (2009). A comparison of sift, pca-sift and surf. International Journal of Image Processing (IJIP), 3(4), 143-152.

Calonder, M., Lepetit, V., Strecha, C., & Fua, P. (2010). Brief: Binary robust independent elementary features. In Computer Vision–ECCV 2010 (pp. 778-792). Springer Berlin Heidelberg.

Bekele, D., Teutsch, M., & Schuchert, T. (2013). Evaluation of binary keypoint descriptors. In Image Processing (ICIP), 2013 20th IEEE International Conference (pp. 3652-3656). doi: 10.1109/ICIP.2013.6738753

Figat, J., Kornuta, T., & Kasprzak, W. (2014). Performance Evaluation of Binary Descriptors of Local Features. In Computer Vision and Graphics (pp. 187-194). Springer International Publishing.

IMORPHEUS.AI

References

Bay, H., Tuytelaars, T., & Van Gool, L. (2006). Surf: Speeded up robust features. In Computer Vision–ECCV 2006 (pp. 404-417). Springer Berlin Heidelberg.

Rublee, E., Rabaud, V., Konolige, K., & Bradski, G. (2011, November). ORB: an efficient alternative to SIFT or SURF. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2564-2571). IEEE.

Leutenegger, S., Chli, M., & Siegwart, R. Y. (2011, November). BRISK: Binary robust invariant scalable keypoints. In Computer Vision (ICCV), 2011 IEEE International Conference on (pp. 2548-2555). IEEE.

Alahi, A., Ortiz, R., & Vandergheynst, P. (2012, June). Freak: Fast retina keypoint. In Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on (pp.510-517).IEEE.

Miksik, O., & Mikolajczyk, K. (2012, November). Evaluation of local detectors and descriptors for fast feature matching. In Pattern Recognition (ICPR), 2012 21st International Conference on (pp. 2681-2684). IEEE.

IMORPHEUS.AI

iMorpheus Journal Club

Next Friday, 8/12/2017 12:00PM GMT+8:

基于结构光的立体视觉 Structured-Light-Based Stereo vision

关键词：Stereo Vision, Structured Light, Machine Vision

微信 Wechat

Website : **http://imorpheus.ai**
Email Address : **live@imorpheus.ai**

IMORPHEUS.AI