Feature descriptors : Analysis and comparison of image feature extraction methods: SIFT and SURF

Nourane BOUZAD
Álvaro ESTEBAN MUÑOZ
Nwokorie ONEYMAUCHE

Outlines

- Introduction
- Theory
- Implementation
- Experimentation and results
- Conclusion

Introduction

- The SIFT and SURF are technologies which allow the reconstruction or recovery of images, and are the key points in image processing and computer science. They are based on detector and descriptor.

- The recovery of an image and its appropriate representation is a recurring problem that can be solved by algorithm application allowing analysis and synthesis of an image or a video.

- The algorithm applications uses tools called descriptors and detectors, which have become indispensable in this field of image representation.

Theory : How the descriptors work?

SIFT

- The method works in two steps:

  - Detections of feature point

  - Feature description.

- The SIFT used DoG to locate and describe feature points.

SURF

- The SURF approach uses box filters, thus enabling real-time application such as tracking and object recognition.

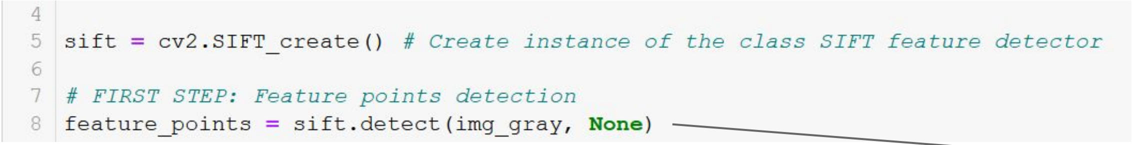
- Firstly, it extracts the strongest image features using speeded up robust features.

- Furthermore, the SURF algorithm constructs the visual vocabulary by extracting 80% of the strongest features and cluster them into visual words using a K -means clustering algorithm.

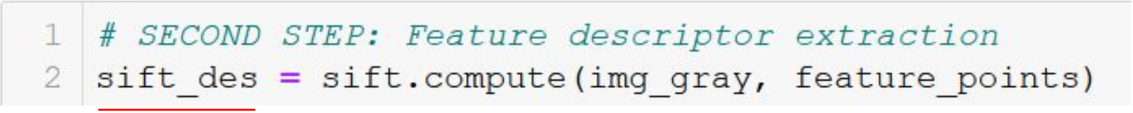
Implementation

To deep more in the algorithm’s implementation, we made two scripts from scratch.

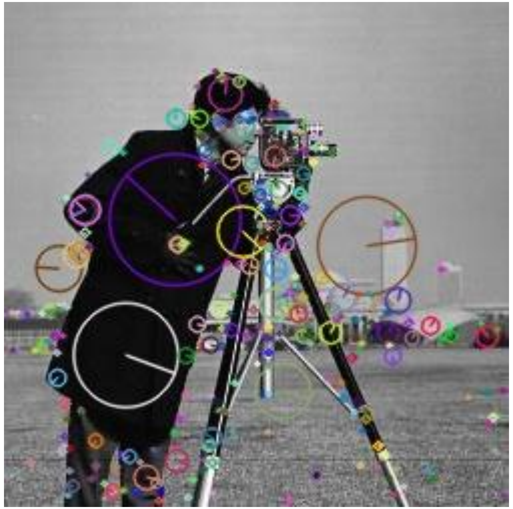
Computing of SIFT descriptor in Python:



```

sift = cv2.SIFT_create()  # Create instance of the class SIFT feature detector

# FIRST STEP: Feature points detection
feature_points = sift.detect(img_gray, None)
```



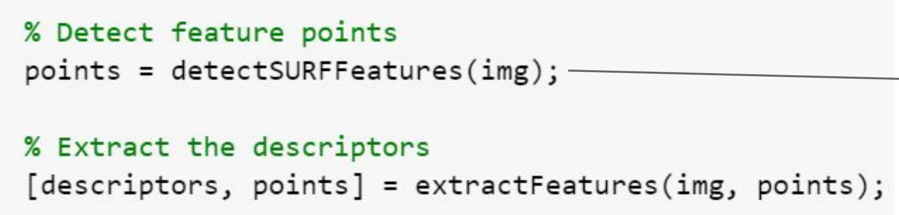
```
# SECOND STEP: Feature descriptor extraction
sift_des = sift.compute(img_gray, feature_points)
```



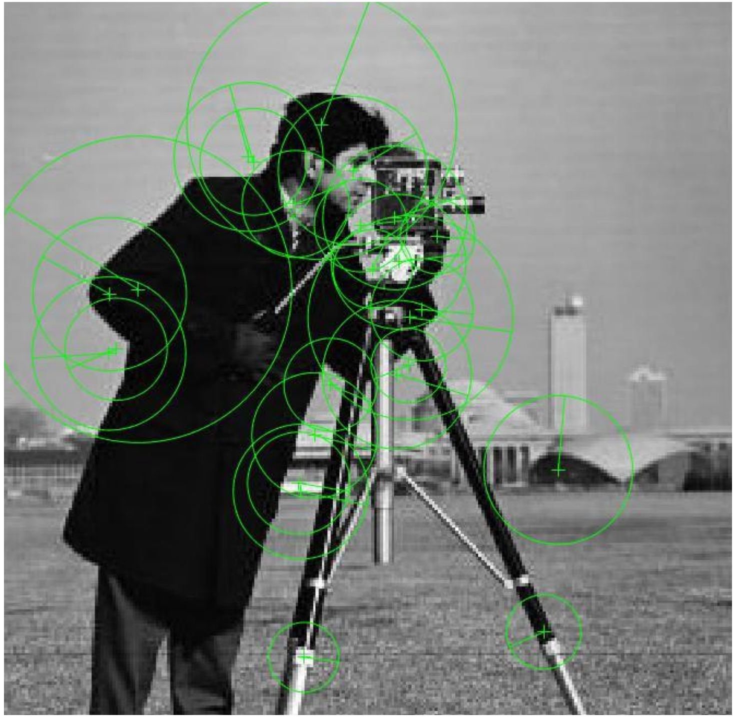
sift_des now contains the descriptors of the features points shown in the image

Implementation

Computing of SURF descriptor in Matlab



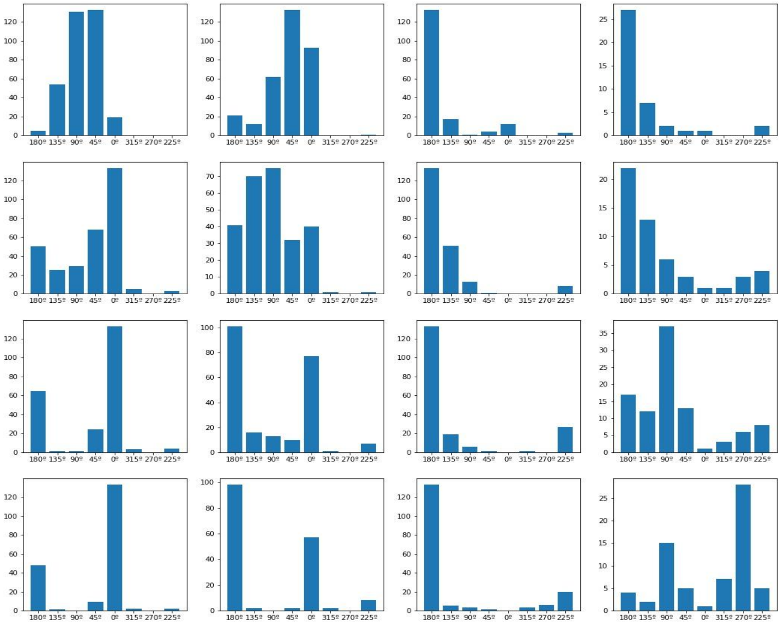
```
% Detect feature points
points = detectSURFFeatures(img);

% Extract the descriptors
[descriptors, points] = extractFeatures(img, points);
```



descriptors contains now the SURF descriptors of our keypoints

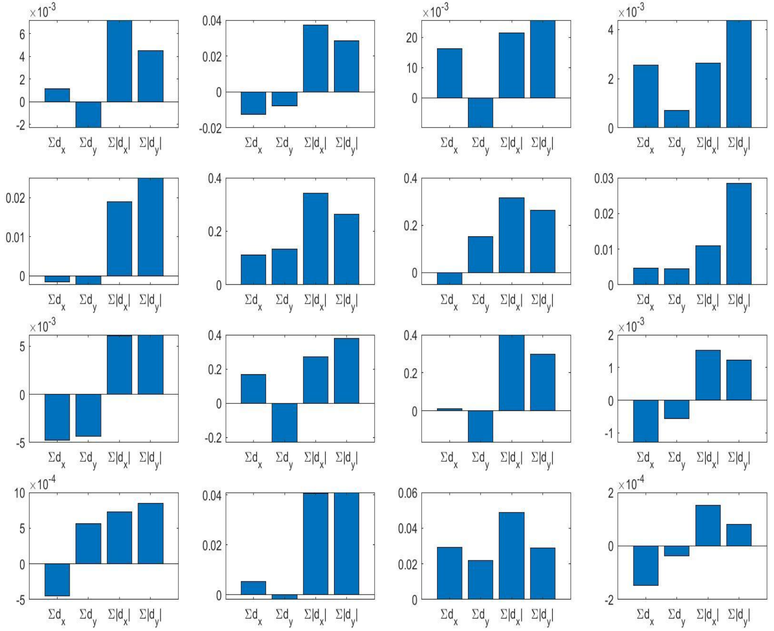
Implementation

SIFT Descriptor



128-dimension vector per feature point.

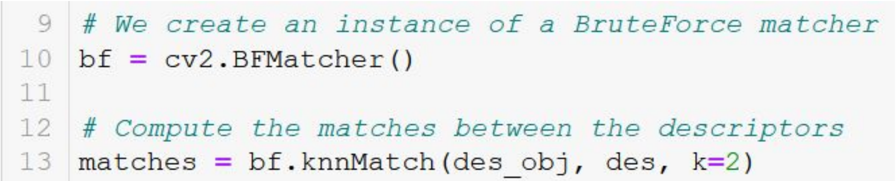
SURF Descriptor



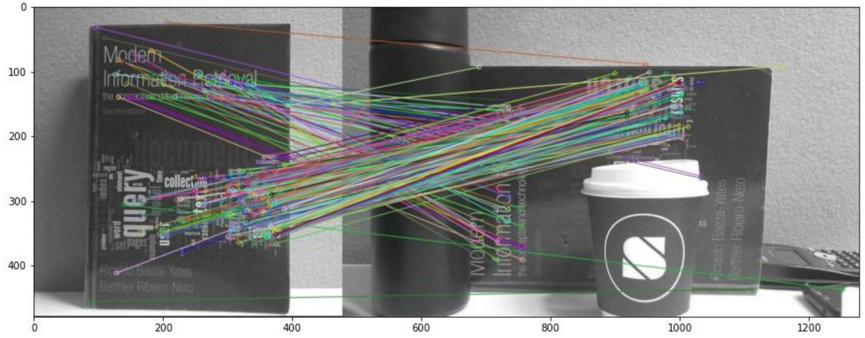
64-dimension vector per feature point.

Implementation

Python match computing

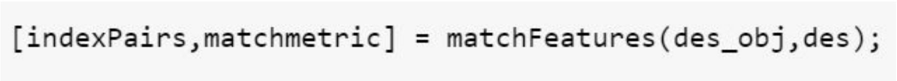


```python
# We create an instance of a BruteForce matcher
bf = cv2.BFMatcher()

# Compute the matches between the descriptors
matches = bf.knnMatch(des_obj, des, k=2)
```

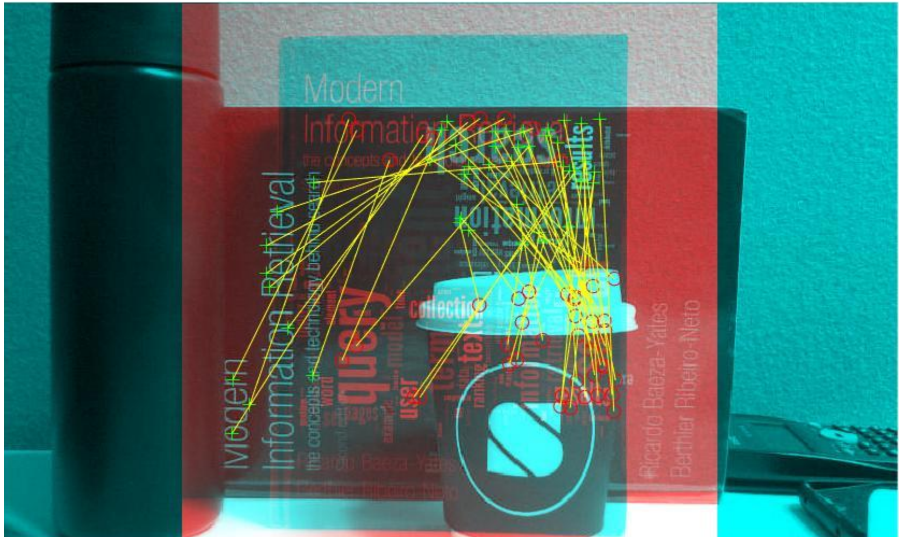


Notice, computation does not depend on the descriptor

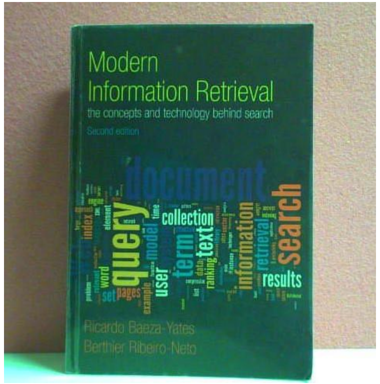
Matlab match computing



```matlab
[indexPairs,matchmetric] = matchFeatures(des_obj,des);
```



First experiment : Test and Compare



- 4 completely different images that can be used for different purpose
    - ○ Description of object or people
    - ○ Satellite imagery

First experiment : Test and Compare

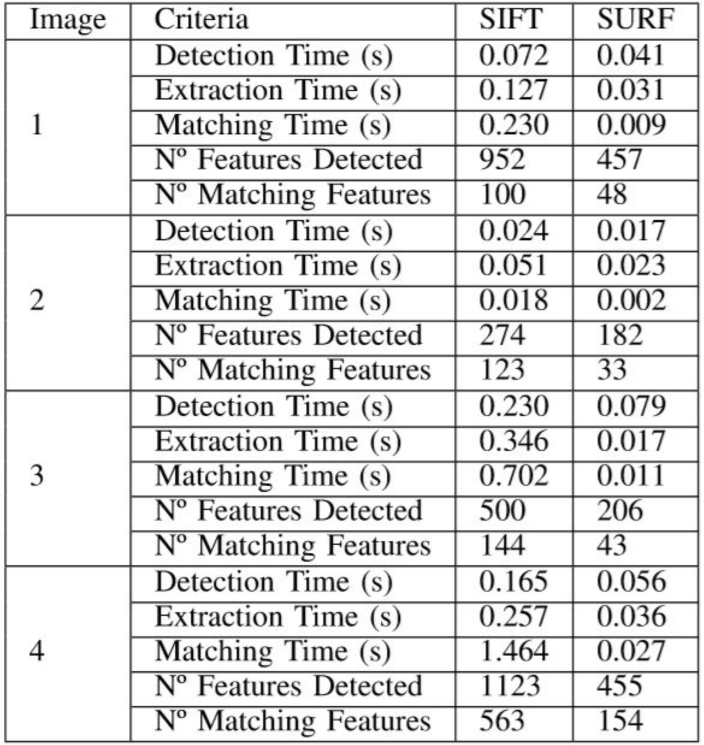


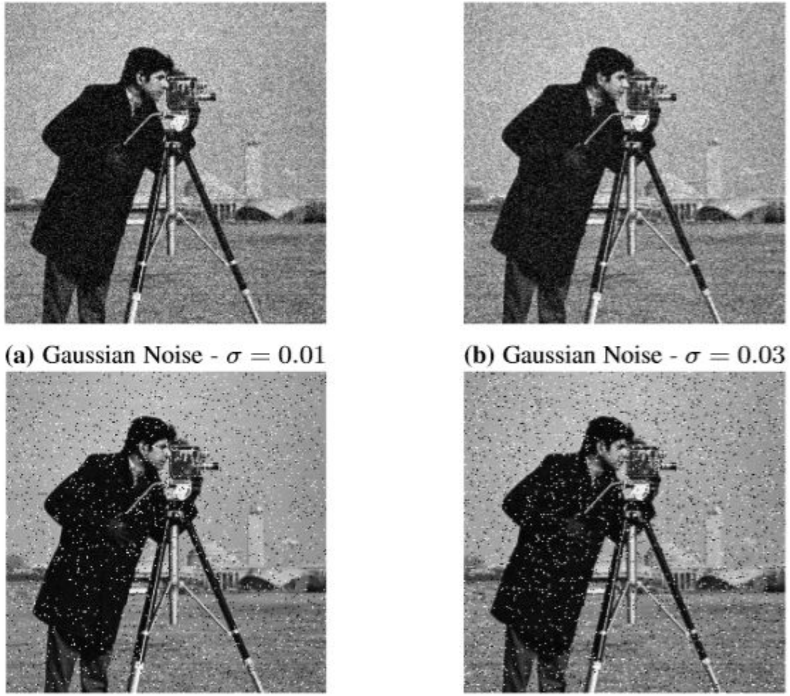
Image	Criteria	SIFT	SURF
1	Detection Time (s)	0.072	0.041
	Extraction Time (s)	0.127	0.031
	Matching Time (s)	0.230	0.009
	Nº Features Detected	952	457
	Nº Matching Features	100	48
2	Detection Time (s)	0.024	0.017
	Extraction Time (s)	0.051	0.023
	Matching Time (s)	0.018	0.002
	Nº Features Detected	274	182
	Nº Matching Features	123	33
3	Detection Time (s)	0.230	0.079
	Extraction Time (s)	0.346	0.017
	Matching Time (s)	0.702	0.011
	Nº Features Detected	500	206
	Nº Matching Features	144	43
4	Detection Time (s)	0.165	0.056
	Extraction Time (s)	0.257	0.036
	Matching Time (s)	1.464	0.027
	Nº Features Detected	1123	455
	Nº Matching Features	563	154

- SIFT detects more features than SURF and so more matching features

  - It doesn’t mean that those of features are useful

- For detection,extraction and matching SURF is faster than SIFT

Second experiment: performance on a noisy image

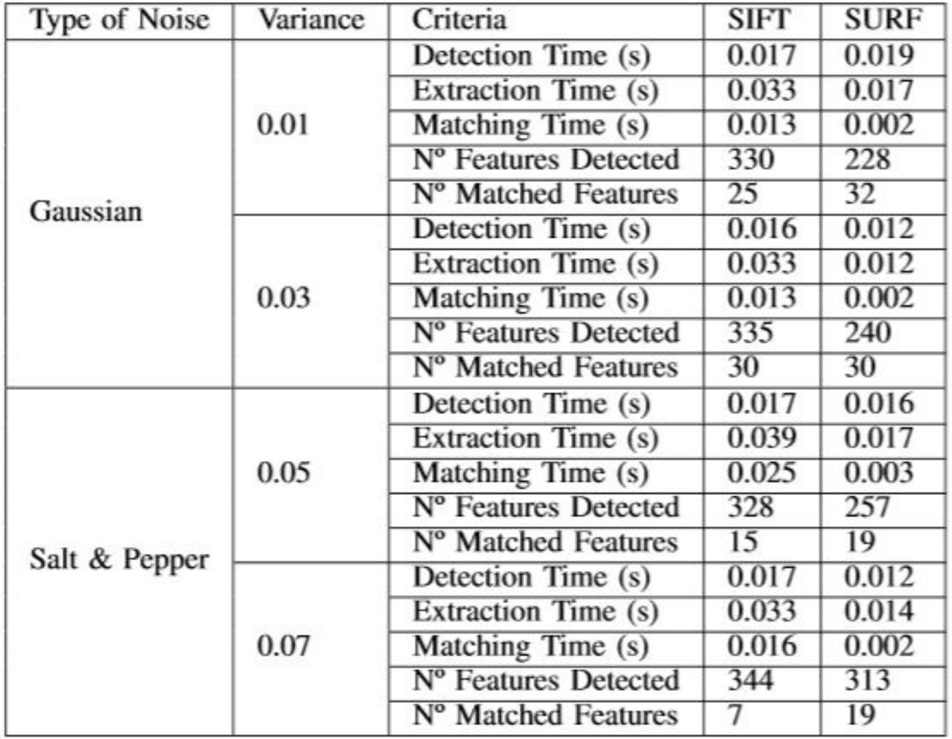


(a) Gaussian Noise - $\sigma = 0.01$

(b) Gaussian Noise - $\sigma = 0.03$

(c) Salt and Pepper Noise - $\sigma = 0.05$

(d) Salt and Pepper Noise - $\sigma = 0.07$



Type of Noise	Variance	Criteria	SIFT	SURF
Gaussian	0.01	Detection Time (s)	0.017	0.019
		Extraction Time (s)	0.033	0.017
		Matching Time (s)	0.013	0.002
		Nº Features Detected	330	228
		Nº Matched Features	25	32
	0.03	Detection Time (s)	0.016	0.012
		Extraction Time (s)	0.033	0.012
		Matching Time (s)	0.013	0.002
		Nº Features Detected	335	240
		Nº Matched Features	30	30
Salt & Pepper	0.05	Detection Time (s)	0.017	0.016
		Extraction Time (s)	0.039	0.017
		Matching Time (s)	0.025	0.003
		Nº Features Detected	328	257
		Nº Matched Features	15	19
	0.07	Detection Time (s)	0.017	0.012
		Extraction Time (s)	0.033	0.014
		Matching Time (s)	0.016	0.002
		Nº Features Detected	344	313
		Nº Matched Features	7	19

Conclusion

- The number of key feature points extracted for SIFT is significantly higher than SURF for the different images tested

- SURF is computationally more efficient

- SIFT becomes less robust when we add noise, as we have seen previously even if it detects more features, SURF has much more matching features