كلية الهندسة بشبرا
FACULTY OF ENGINEERING AT SHOUBRA
ERJ
PRINT ISSN 3009-6049
ONLINE ISSN 3009-6022

ENGINEERING RESEARCH JOURNAL (ERJ)
Volume (53) Issue (2)
April 2024, pp:136-143
https://erjsh.journals.ekb.eg

# SfM ToAEval: Trade-off Aware Evaluation of Feature Extraction Algorithms in Structure from Motion

**Dina M. Taha*[1], Shady Y. El-Mashad[1], Hala H. Zayed[2,3]**

[1] Department of Electrical Engineering, Faculty of Engineering at Shoubra, Benha University, Cairo, Egypt.
[2] Faculty of Computers and Artificial Intelligence, Benha University, Benha, Egypt.
[3] Faculty of Engineering, Egypt University of Informatics (EUI), Cairo, Egypt .
[*] Corresponding Author.
E-mail: : dina.taha@feng.bu.edu.eg, shady.elmashad@feng.bu.edu.eg, hala.zayed@fci.bu.edu.eg

**Abstract** :This paper presents SfM ToAEval, a framework for evaluating different feature extraction algorithms in Structure from Motion (SfM) pipelines. SfM ToAEval allows automatically evaluating the effect of using different feature detectors and descriptors combinations on the quality of the 3D reconstruction from a given collection of image sequences. In addition, SfM ToAEval evaluates the 3D reconstruction without the need for ground truth. Moreover, SfM ToAEval is aware of the reconstruction density-accuracy trade-off, and it supports visualizing it in order to allow deciding the "best" reconstruction transparently. Furthermore, SfM ToAEval allows quantifying the quality of each 3D reconstruction compared to others. SfM ToAEval was used to evaluate 98 feature detectors and descriptors combinations on six image sequences, and it was able to identify four promising combinations. Experimental results comparing the proposed combinations with related work are presented in this research. The complete source code of the proposed framework as well as a minimal Jupyter Notebook demonstrating how different functionalities can be used are released under the MIT license.

**Keywords:** Feature Detector; Feature Descriptor; Feature Correspondence; Structure from Motion (SfM); 3D Reconstruction.

## 1. INTRODUCTION

Recently, there has been a lot of interest in the 3D reconstruction of objects and scenes from 2D images due to its numerous practical applications [1]. Structure from Motion (SfM) is a widely used pipeline for reconstructing a sparse 3D point cloud of stationary objects and scenes from a given sequence of images that are taken for the object or scene from different viewpoints. The SfM pipeline starts with detecting and describing features in each image of the given sequence of images. The second stage is to match the detected features in pairs of images in order to identify corresponding features. In subsequent stages, the 2D location of the corresponding features is used for estimating the 3D location of the actual point in the sparse point cloud [2,3]. Consequently, the accuracy of the 2D location of the detected features (the responsibility of the feature detector) as well as the accuracy of the description and matching of the detected features (the responsibility of the feature descriptor) significantly affect the accuracy of the estimated 3D location. Therefore, a lot of attention has been paid during the past decade to the problem of deciding good feature detectors and descriptors combinations for SfM and it has become a hot area of research [4–12].

When using synthetic data (like SfM Flow does in [13]) is not an option, and the ground truth of the real data is not available, evaluation of the 3D reconstruction becomes very challenging as relying on the value of the reconstruction density (number of 3D points in the sparse point cloud) and accuracy (average reprojection error of the 3D points on the images) can be misleading [14].

The recent interest in the evaluation of different feature detectors and descriptors combinations in SfM, the challenge of evaluating 3D reconstruction in the absence of ground truth, and the increasing number of published feature extraction algorithms (for example, as of 2022, OpenCV implements about 30 different feature extraction algorithms [15]) motivated the development of SfM ToAEval.

The remaining sections of this paper are organized as follows: Section 0 briefly summarizes the related work. Section 0 provides a concise overview of different feature detection and feature extraction algorithms that were evaluated in this paper. Section 0 presents the framework proposed for automatically evaluating different feature detectors and descriptors combinations. Section 0 presents the experimental results obtained while section 0 discusses them. In Section 0, the conclusion and future work are reported.

## 2.RELATED WORK

To the best of the authors' knowledge, SfM ToAEval is the only published framework that fully automates the process of transparently evaluating different feature detectors and descriptors combinations on real datasets without the need for ground truth. On the other hand, SfM Flow [13] allows evaluating 3D reconstruction on synthetic datasets. It supports some incremental SfM pipelines (including COLMAP), but it does not support using different feature extraction algorithms.

In [4], Govender evaluated HCD, KLY, SIFT, and SURF. They reported that SIFT resulted in the minimum error. In [10], Urban et al. evaluated AKAZE + M-SURF, ORB, SIFT, SURF, and SURF + BinBoost. They reported that SURF is faster than other algorithms. In [6], Chien et al. evaluated AKAZE, ORB, SIFT, and SURF. They reported that SURF resulted in the largest number of features, AKAZE requires the smallest storage, and ORB was the fastest. In [16] Pusztai et al. evaluated AGAST, AKAZE, BRISK, FAST, GFTT, KAZE, MSER, ORB, SIFT, STAR, and SURF. They reported that SURF resulted in the largest number of features (Inliers). In [5], Schönberger et al. evaluated ConvOpt, DeepDesc, DSP-SIFT, LIFT, SIFT, SIFT- PCA, and TFeat. They reported that SIFT resulted in the minimum error, required the smallest storage, and was the fastest. In [8], Cao et al. evaluated BRISK, KAZE, ORB, SIFT, and SURF. They reported that SURF resulted in the minimum error (ROS) and BRISK was the fastest. In [9], Gao et al. evaluated AKAZE, DeepCompare, LF-Net, ORB, SIFT, SuperPoint, and SURF. They reported that SURF resulted in the denser reconstruction, AKAZE resulted in the minimum error, and ORB was the fastest. In [7], Yusefi et al. evaluated FAST, ORB, SIFT, STAR, and SURF. They reported that FAST resulted in the minimum error and STAR was the fastest.

To the best of the authors' knowledge, the experiment presented in this paper is the most comprehensive published experiment for evaluating feature extraction algorithms in SfM. In this experiment, all possible combinations of seven feature detectors and fourteen feature descriptors (a total of 98 combinations) were evaluated on six image sequences (a total of 588 3D reconstructions).

## 3.FEATURE DETECTORS AND DESCRIPTORS

Feature extraction algorithms can be designed for the purpose of feature detection only (like AGAST [17], FAST [18], and STAR [19]), feature description only (like BEBLID [20], BRIEF [21], DAISY [22], FREAK [23], LATCH [24], LUCID [25], TEBLID [26], and VGG [27]), or both feature detection and description (like AKAZE [28], BRISK [29], KAZE [30], ORB [31], SIFT [32], and SURF [33]). In this paper, algorithms that were designed for feature detection are combined with algorithms that were designed for feature description. This allowed SfM ToAEval to evaluate 126 (9 × 14) different combinations in the experimental work reported in section 0. In this paper, seventeen feature extraction algorithms (listed in Table 1) were evaluated.

## 4.PROPOSED FRAMEWORK

In this section, SfM ToAEval, the proposed framework for automatically evaluating different feature detectors and descriptors combinations in SfM is presented. SfM ToAEval is a stand-alone cross-platform software that is entirely developed using Python 3 and can be easily used on Google Colab or in a Jupyter Notebook on Windows, Linux, or macOS. SfM ToAEval not only automates the process of evaluation but also employs size-error curves proposed by Taha et al. in [14] for visualizing the reconstruction density-accuracy trade-off in order to allow deciding the best reconstruction transparently. As the visualization may become crowded when the number of combinations being evaluated grows as shown in Fig. 3, we propose quantifying the quality of each 3D reconstruction compared to others according to equation (1 and the algorithm shown in Fig. 1

$$S_i = \sum_{j \in A-\{i\}} \mathrm{H}(\varepsilon_j^{\min(|PC_i|,|PC_j|)} - \varepsilon_i^{\min(|PC_i|,|PC_j|)}) \qquad (1)$$

Where:

- $S_i$ is the quality score of reconstruction $i$,
- $A$ is the set of reconstructions being evaluated,
- $H(x)$ is the unit step function,
- $|PC_i|$ is the size of the point cloud in reconstruction $i$,
- $\varepsilon_i^n$ is the average reprojection error of the best $n$ 3D point in the point cloud in reconstruction $i$

The formula is based on counting the number of size-error curves that are asymptotically faster than the curve corresponding to reconstruction $i$ which gives the number of reconstructions that reconstruction $i$ outperforms.

Table 1 Evaluated Feature Extraction Algorithms

| | | Detector | Descriptor | Both |
|---|---|---|---|---|
| AGAST | [17] | X | | |
| AKAZE | [28] | | | X |
| BEBLID | [20] | | X | |
| BRIEF | [21] | | X | |
| BRISK | [29] | | | X |
| DAISY | [22] | | X | |
| FAST | [18] | X | | |
| FREAK | [23] | | X | |
| KAZE | [30] | | | X |
| LATCH | [24] | | X | |
| LUCID | [25] | | X | |
| ORB | [31] | | | X |
| SIFT | [32] | | | X |
| STAR | [19] | X | | |
| SURF | [33] | | | X |
| TEBLID | [26] | | X | |
| VGG | [27] | | X | |

```
Algorithm Quality Score Estimation (PC)
Input : list of point clouds PC, each point cloud is
      composed of a list of 3D points (x, y, z, e)
      sorted in non-decreasing order of e
Output: list of quality scores

S = [ ]
for i in range(len(PC)):
   s = 0
   for j in range(len(PC)):
      if i != j:
         m = min(len(PC[i]), len(PC[j]))
         if (sum([p[-1] for p in PC[j][0:m]]) >
            sum([p[-1] for p in PC[i][0:m]])):
            s += 1
   S.append(s)

return S
```

Fig. 1. Proposed algorithm for quantifying the quality of 3D reconstructions

SfM ToAEval framework employs OpenCV [34] for feature detection, description, and matching (the first stage of the SfM pipeline). For the rest of the SfM pipeline, SfM ToAEval framework employs COLMAP [35,36]. The architecture of the proposed framework is illustrated in Fig. 2. SfM ToAEval provides three main functionalities:

### I. 3D Reconstruction

3D reconstruction is automatically performed for each possible combination of the feature detectors and feature descriptors provided to the proposed framework on each sequence (subfolder) in the dataset folder.

1. Feature Extraction
   Feature detection and feature description are automatically performed using OpenCV for all possible combinations of the feature detectors and feature descriptors provided to the proposed framework as illustrated in Figure 3. Out of the features detected by each algorithm, the subsequent stage receives only a fixed number in order to reduce the evaluation time and ensure fairness [14].
2. Feature Matching
   Feature extraction is automatically performed using OpenCV brute force matcher with cross-check enabled and the default norm for each feature descriptor. Out of the features matched by each algorithm, the subsequent stage receives only a fixed number in order to reduce the evaluation time and ensure fairness [14].
3. Sparse Point Cloud Reconstruction
   Sparse point clouds are automatically reconstructed using COLMAP command line interface with the default parameters.

### II. Analysis

SfM ToAEval analyses the reconstructed sparse point clouds and generates the following statistics in an SQLite database:

1. Total Number of Extracted Features
2. Total Descriptors Size (in Bytes)
3. Feature Extraction Time (in Seconds)
4. Feature Matching Time (in Seconds)
5. Reprojection Error (in Pixels)
6. Point Cloud Size
7. Data of Size-Error Curves
8. Quality Scores

### III. Visualization

SfM ToAEval currently supports two types of visualization: Radar charts for visualizing the point cloud size or the reprojection error and size-error curves for visualizing the trade-off between them.

1. Radar Chart
   It is a convenient way to illustrate the relative quality of the reconstructed sparse point clouds (as shown in Fig. 6).
2. Size-Error Curves
   They visualize the 3D reconstruction density-accuracy trade-off (as shown in Fig. 5). This allows transparently deciding the combination corresponding to the best reconstruction (the one with the slowest growth rate of reprojection error with the point cloud size).

All extracted and matched features are stored in a single SQLite database. In addition, for each sequence and feature detector and descriptor combination, SfM ToAEval builds an

SQLite database and passes it to COLMAP. Moreover, all reconstructed point clouds are analysed, and the analysis results are stored in a single SQLite database. SfM ToAEval stores all intermediate and final results to allow:

1. Pausing and resuming the evaluation
2. Verifying the results
3. Performing more analysis
4. Using the results for building machine learning or deep learning models

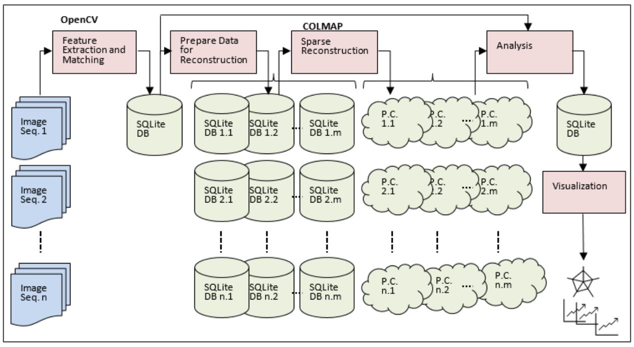


Fig. 2. Architecture of the proposed framework

## 5.EXPERIMENTAL WORK

### 1. Dataset

A widely used and publicly available collection of image sequences was used as a benchmark for evaluating different feature detectors and descriptors combinations in SfM. The collection contains six different sequences of 3072 × 2028-pixel images that were previously corrected for radial distortion. More information about the dataset can be found in [37,38].

### 2. Setup

A virtual machine (provided by Google Collaborate Pro) was used for empirically evaluating the quality of different feature extraction algorithms. The virtual machine is powered with a 2 x Core Intel(R) Xeon(R) CPU @ 2.20 GHz and equipped with 13.00 GB of RAM. OpenCV 4.6.0 was used for feature detection, description, and matching. It was built with OPENCV_EXTRA_MODULES_PATH and OPENCV_ENABLE_NONFREE in order to enable proprietary and patented feature extraction algorithms. COLMAP 3.8 was used for the remaining stages of the SfM pipeline. It was built with the default configuration.

### 3. Results

In this experiment, 98 feature detectors and descriptors combinations were evaluated in constructing sparse point clouds from six image sequences resulting in a total of 588 sparse point clouds. The 588 size error curves are shown in Fig. 3. Although the visualization may be crowded, it is clear that SIFT feature detector is the best for entry-P10, fountain-P11, Herz-Jesus-P25, and Herz-Jesus-P8 image sequences while SURF feature detector is the best for castle-P19 and castle-P30 image sequences. The reprojection error and point cloud size of the 98 combinations are shown in Fig. 4. Although the radar chart may be crowded, it is clear that ORB feature detector resulted in the sparsest point cloud and the highest reprojection error.

From this comprehensive experiment, we were able to identify four promising combinations that were compared with combinations suggested in related work. The results of the comparison are summarized in Table 2 which shows the quality score of each combination with each sequence, Fig. 5 which shows size-error curves, and Fig. 6 which shows radar charts of point cloud size as well as reprojection error of each evaluated combination with each image sequence. It is clear that the four proposed combinations outperform all other combinations suggested in related work.

From the size-error curves shown in Fig. 5, the 3D reconstruction with the slowest growth rate of reprojection error with the point cloud size can be identified as the best reconstruction regardless of the value of point cloud size and reprojection error even if it is not the denser (the one with the largest number of points in the point cloud) or not the most accurate (the one with the smallest reprojection error). For example, the 3D reconstruction of Herz-Jesus-P25 using SURF – VGG combination is the best. This is also confirmed by the quality score reported in Table 2. The quality score of SURF – VGG is eight which means that it outperforms the other eight combinations that were evaluated on Herz-Jesus-P25 image sequence. The same applies to other sequences.

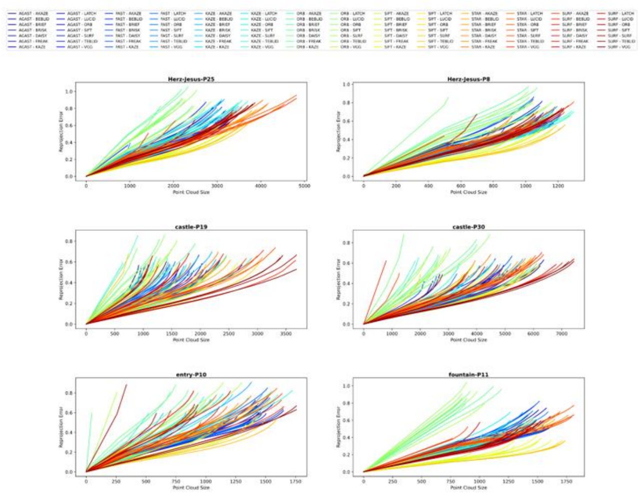


Fig. 3. Reprojection error vs. the point cloud size for 98 feature detectors and descriptors combinations

Table 2 Quality score of the proposed combinations compared to related work.

| Author(s) | | Algorithm | castle-P19 | castle-P30 | entry-P10 | fountain-P11 | Herz-Jesus-P25 | Herz-Jesus-P8 | Best |
|---|---|---|---|---|---|---|---|---|---|
| Chien et al.<br>Gao et al. | [6]<br>[9] | AKAZE - AKAZE | 4 | 3 | 3 | 2 | 2 | 5 | 5 |
| Cao et al. | [8] | BRISK - BRISK | 0 | 0 | 2 | 1 | 1 | 1 | 2 |
| Chien et al.<br>Gao et al. | [6]<br>[9] | ORB - ORB | 1 | 1 | 1 | 0 | 0 | 0 | 1 |
| Govender<br>Schönberger et al. | [4]<br>[5] | SIFT - SIFT | 6 | 6 | 4 | 6 | 6 | 6 | 6 |
| Cao et al.<br>Gao et al.<br>Pusztai et al.<br>Urban et al. | [8]<br>[9]<br>[16]<br>[10] | SURF - SURF | 2 | 2 | 0 | 3 | 3 | 2 | 3 |
| Proposed | | SIFT - TEBLID | 5 | 5 | 5 | 7 | **8** | 7 | 8 |
| | | SIFT - VGG | 3 | 4 | **8** | **8** | 7 | **8** | 8 |
| | | SURF - TEBLID | 7 | **8** | 6 | 5 | 4 | 4 | 8 |
| | | SURF - VGG | **8** | 7 | 7 | 4 | 5 | 3 | 8 |
| Best | | | 8 | 8 | 8 | 8 | 8 | 8 | 8 |

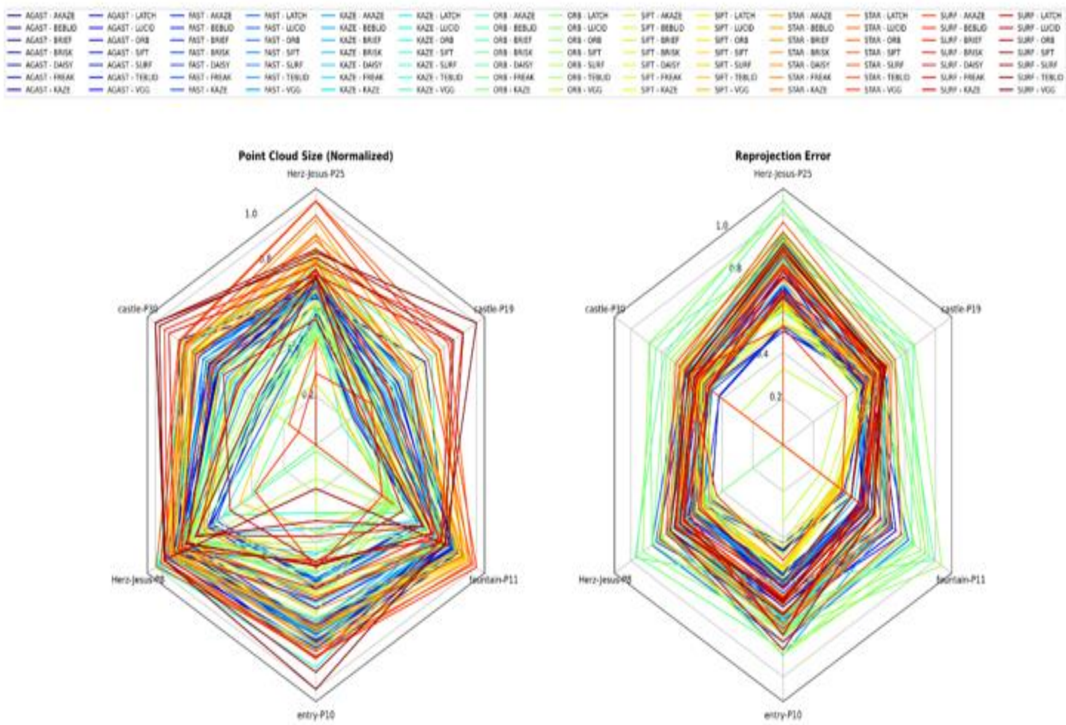


Fig. 4. Point cloud size (on the left) and reprojection error (on the right) radar charts for 98 feature detectors and descriptors combinations

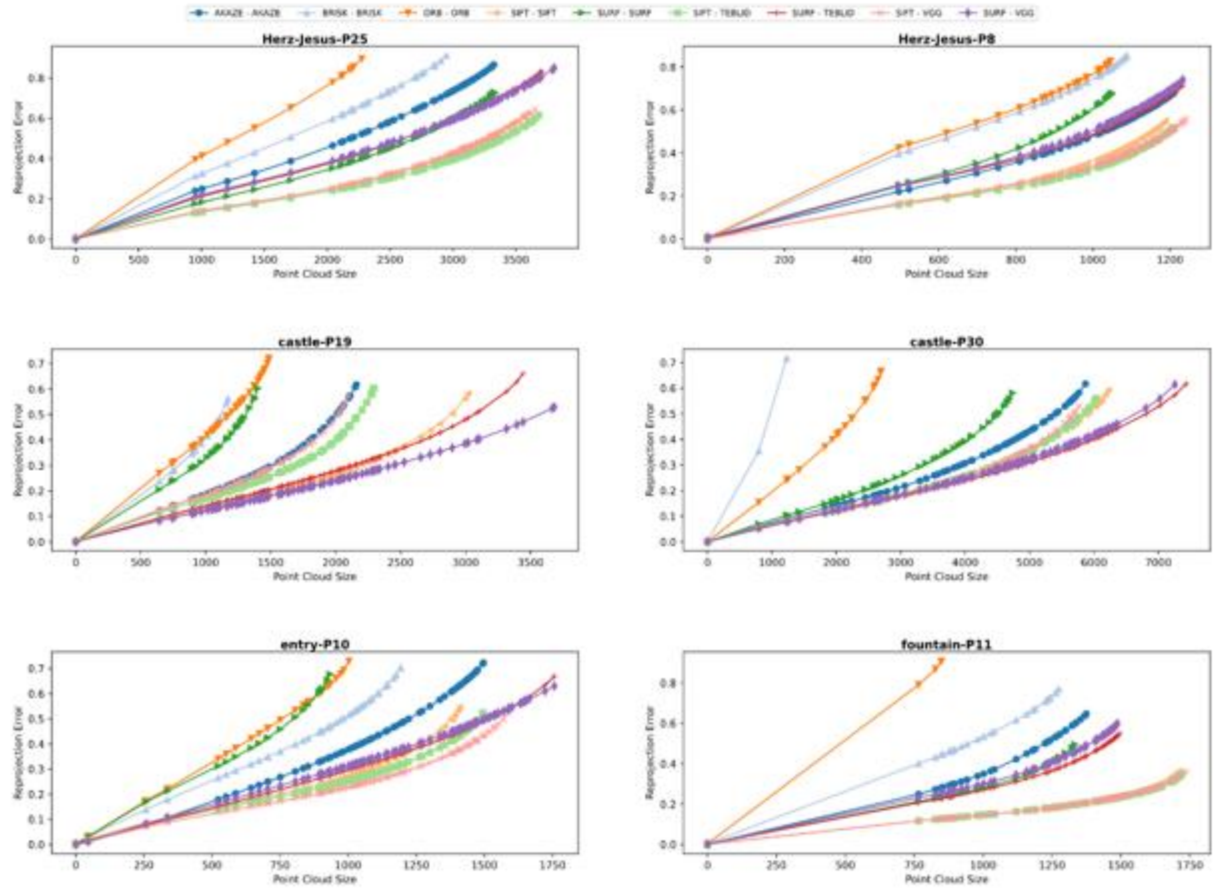


Fig. 5. Reprojection error as vs the point cloud size for the proposed combinations compared to related work

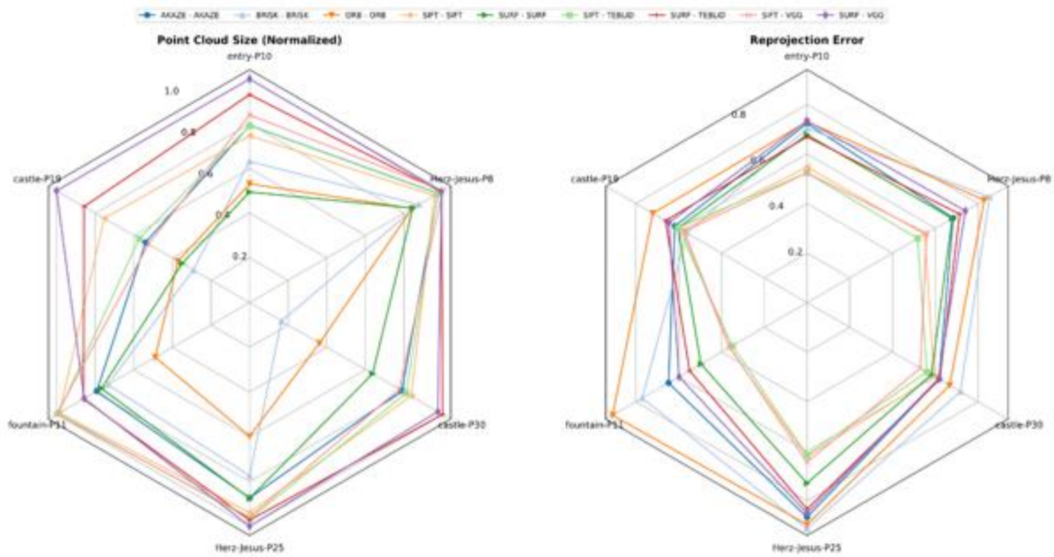


Fig. 6. Point cloud size (on the left) and reprojection error (on the right) radar charts for the proposed combinations compared to related work

## 6.DISCUSSION

In the Computer Vision domain, new feature extraction algorithms are continuously developed especially after the recent advances in the field of Deep Learning. SfM ToAEval can lead to discovering interesting feature detectors and descriptors combinations for 3D reconstruction and direct the research to improve potential combinations as shown in Table 2. In addition, SfM ToAEval can be used for automatically generating large amounts of experimental results that can be used for training machine learning models that recommend a suitable feature detector and descriptor combination for a given image sequence from the sequence characteristics without the need to try all the combinations. Moreover, even larger amounts of generated experimental results can be used to build deep learning-based feature detectors and feature descriptors that are specialized in 3D reconstruction. That is why all experimental results are automatically stored in SQLite databases.

On the other hand, SfM ToAEval can be used before applying 3D reconstruction to large image sequences in order to efficiently decide a suitable feature detector and descriptor combination in limited reconstructions before the actual reconstruction. This can dramatically reduce the reconstruction cost without sacrificing the quality.

## 7.CONCLUSION AND FUTURE WORK

This paper introduced SfM ToAEval, a Python-based open-source framework for automatically evaluating different feature detectors and descriptors combinations in the 3D reconstruction of stationary objects and scenes from a given sequence of images taken from different viewpoints. SfM ToAEval framework is aware of the reconstruction density-accuracy trade-off, and it allows visualizing this trade-off for a set of reconstructions and selecting the best reconstruction transparently. In addition, SfM ToAEval framework allows quantifying the quality of each 3D reconstruction compared to others based on the quality score proposed in this paper. This can be very helpful when the number of reconstructions is large, and the visualization is very crowded.

SfM ToAEval framework was used for evaluating 98 different feature detectors and descriptors combinations. Based on the results of the experimental work, SfM ToAEval framework managed to identify four promising combinations that outperform other combinations suggested in related works.

SfM ToAEval framework supports 100+ different feature detectors and descriptors combinations out of the box, and it can be easily extended to support any feature extraction algorithm implemented in OpenCV. We are currently allowing SfM ToAEval framework to be extended to support handcrafted and learned feature extraction algorithms that are not implemented in OpenCV.

## REFERENCES

[1]. Y.M. Wei, L. Kang, B. Yang, L. da Wu, Applications of structure from motion: A survey, Journal of Zhejiang University: Science C. 14 (2013).

[2]. O. Özyeşil, V. Voroninski, R. Basri, A. Singer, A survey of structure from motion, Acta Numerica. 26 (2017).

[3]. K. Häming, G. Peters, The structure-from-motion reconstruction pipeline - A survey with focus on short image sequences, Kybernetika. 46 (2010).

[4]. N. Govender, Evaluation of Feature Detection Algorithms for Structure from Motion, Csir. (2009).

[5]. J.L. Schönberger, H. Hardmeier, T. Sattler, M. Pollefeys, Comparative evaluation of hand-crafted and learned local features, in: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, 2017.

[6]. H.J. Chien, C.C. Chuang, C.Y. Chen, R. Klette, When to use what feature? SIFT, SURF, ORB, or A-KAZE features for monocular visual odometry, in: International Conference Image and Vision Computing New Zealand, 2016.

[7]. A. YUSEFI, A. DURDU, C. SUNGUR, Performance and Trade-off Evaluation of SIFT, SURF, FAST, STAR and ORB feature detection algorithms in Visual Odometry, European Journal of Science and Technology. (2020).

[8]. M. Cao, L. Cao, W. Jia, Y. Li, Z. Lv, L. Zheng, X. Liu, Evaluation of local features for structure from motion, Multimed Tools Appl. 77 (2018).

[9]. K. Gao, H. Aliakbarpour, J. Fraser, K. Nouduri, F. Bunyak, R. Massaro, G. Seetharaman, K. Palaniappan, Local feature performance evaluation for structure-from-motion and multi-view stereo using simulated city-scale aerial imagery, IEEE Sens J. 21 (2021).

[10]. S. Urban, M. Weinmann, Finding a good feature detector-descriptor combination for the 2d keypoint-based registration of tls point clouds, in: ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, 2015.

[11]. S. Jiang, W. Jiang, B. Guo, L. Li, L. Wang, Learned Local Features for Structure from Motion of UAV Images: A Comparative Evaluation, IEEE J Sel Top Appl Earth Obs Remote Sens. 14 (2021).

[12]. S.Y. El-Mashad, A. Shoukry, Evaluating the robustness of feature correspondence using different feature extractors, in: 2014 19th International Conference on Methods and Models in Automation and Robotics, MMAR 2014, 2014.

[13]. D. Marelli, S. Bianco, G. Ciocca, SfM Flow: A comprehensive toolset for the evaluation of 3D reconstruction pipelines, SoftwareX. 17 (2022).

[14]. D. Taha, H. Zayed, S. El-Mashad, Towards a Fair Evaluation of Feature Extraction Algorithms Robustness in Structure from Motion, International Journal of Advanced Computer Science and Applications(IJACSA). 13 (2022).

[15]. Intel Corporation, Willow Garage, Itseez, Open Source Computer Vision Library, (2022). https://github.com/opencv/opencv (accessed November 30, 2022).

[16]. Z. Pusztai, L. Hajder, Quantitative Comparison of Feature Matchers Implemented in OpenCV3, in: 21 St. Computer Vision Winter Workshop, Rimske Toplice, 2016.

[17]. E. Mair, G.D. Hager, D. Burschka, M. Suppa, G. Hirzinger, Adaptive and generic corner detection based on the accelerated segment test, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2010.

[18]. E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2006.

[19]. 19 A. Patel, D. R. Kasat, S. Jain, V. M. Thakare, Performance Analysis of Various Feature Detector and Descriptor for Real-Time Video based Face Tracking, Int J Comput Appl. 93 (2014).

[20]. I. Suárez, G. Sfeir, J.M. Buenaposada, L. Baumela, BEBLID: Boosted efficient binary local image descriptor, Pattern Recognit Lett. 133 (2020).

[21]. M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: Binary robust independent elementary features, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2010.

[22]. E. Tola, V. Lepetit, P. Fua, DAISY: An efficient dense descriptor applied to wide-baseline stereo, IEEE Trans Pattern Anal Mach Intell. 32 (2010).

[23]. A. Alahi, R. Ortiz, P. Vandergheynst, FREAK: Fast retina keypoint, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2012.

[24]. G. Levi, T. Hassner, LATCH: Learned arrangements of three patch codes, in: 2016 IEEE Winter Conference on Applications of Computer Vision, WACV 2016, 2016.

[25]. A. Ziegler, E. Christiansen, D. Kriegman, S. Belongie, Locally uniform comparison image descriptor, in: Adv Neural Inf Process Syst, 2012.

[26]. I. Suarez, J.M. Buenaposada, L. Baumela, Revisiting Binary Local Image Description for Resource Limited Devices, IEEE Robot Autom Lett. 6 (2021).

[27]. K. Simonyan, A. Vedaldi, A. Zisserman, Learning local feature descriptors using convex optimisation, IEEE Trans Pattern Anal Mach Intell. 36 (2014).

[28]. P.F. Alcantarilla, J. Nuevo, A. Bartoli, Fast explicit diffusion for accelerated features in nonlinear scale spaces, in: BMVC 2013 - Electronic Proceedings of the British Machine Vision Conference 2013, 2013.

[29]. S. Leutenegger, M. Chli, R.Y. Siegwart, BRISK: Binary Robust invariant scalable keypoints, in: Proceedings of the IEEE International Conference on Computer Vision, 2011.

[30]. P.F. Alcantarilla, A. Bartoli, A.J. Davison, KAZE features, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012.

[31]. E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An efficient alternative to SIFT or SURF, in: Proceedings of the IEEE International Conference on Computer Vision, 2011.

[32]. D.G. Low, Distinctive image features from scale-invariant keypoints, Int J Comput Vis. (2004).

[33]. H. Bay, T. Tuytelaars, L. van Gool, SURF: Speeded up robust features, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2006.

[34]. G. Bradski, The OpenCV Library, Dr. Dobb's Journal of Software Tools. (2000).

[35]. J.L. Schonberger, J.M. Frahm, Structure-from-Motion Revisited, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2016.

[36]. J.L. Schönberger, E. Zheng, J.M. Frahm, M. Pollefeys, Pixelwise view selection for unstructured multi-view stereo, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016.

[37]. C. Strecha, W. von Hansen, L. van Gool, P. Fua, U. Thoennessen, On benchmarking camera calibration and multi-view stereo for high resolution imagery, in: 26th IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2008.

[38]. A. Ley, R. Hänsch, O. Hellwich, Syb3r: A realistic synthetic benchmark for 3D reconstruction from images, in: Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2016.