69th International Astronautical Congress (IAC), Bremen, Germany, 1-5 October 2018. Copyright © 2018 by D. Rondao, N. Aouf, O. Dubois-Matra. Published by the IAF, with permission and released to the IAF to publish in all forms.
https://iafastro.directory/iac/paper/id/43502/summary/

IAC–18–C1.IP.3

Multispectral Image Processing for Navigation Using Low Performance Computing

Duarte Rondao[a]*, Nabil Aouf[a] , Olivier Dubois-Matra[b]

[a] *Centre for Electronic Warfare, Information and Cyber, Cranfield University, Defence Academy of the United Kingdom, Shrivenham, SN6 8LA, United Kingdom*, {d.rondao, n.aouf}@ cranfield.ac.uk
[b] *European Space Agency, ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, Netherlands*, olivier.dubois-matra@esa.int
* Corresponding Author

Abstract

Space debris represents a growing threat for both current spacecraft and future launches. This is exceptionally alarming in the case of low Earth orbits, where chain impacts of existing debris generate even more fragments, increasing the probability of further collisions. The now defunct satellite Envisat represents one of the largest objects classified as space debris. The e.Deorbit mission will demonstrate active debris removal (ADR) technology to successfully decommission Envisat and other non-functional target spacecraft in orbit. Relative navigation solutions shall be achieved using image processing algorithms, which implies the detection and matching of two-dimensional regions of interest. In this work, multiple pattern recognition techniques are investigated for the detection and description of these features. This analysis of feature perception is achieved for the first time in the context of space non-cooperative rendezvous (NCRV) across two different modalities: the visible (0.39-0.70 μm) and the thermal infrared (8-14 μm). The assessed algorithms are implemented in a dedicated, space-appropriate hardware processor to benchmark their real-time capabilities.
Keywords: Active Debris Removal, Multispectral, Feature Detection, Local Descriptor, Space Navigation, Real-Time

Acronyms/Abbreviations

ADR active debris removal
BBB BeagleBone(R) Black
BRISK Binary Robust Invariant Scalable Keypoints
CenSurE Centre Surround Extrema
DoG Difference of Gaussians
FAST Features from Accelerated Segment Test
FREAK Fast Retina Keypoint
GFTT Good Features To Track
IP image processing
IR infrared
LIOP Local Intensity Order Pattern
LoG Laplacian of Gaussians
LVLH local-vertical-local-horizontal
NCRV non-cooperative rendezvous
NNDR nearest-neighbour distance ratio
ORB Oriented FAST and Rotated BRIEF
ROC receiver operating characteristic
SFM structure from motion
SIFT Scale-Invariant Feature Transform
SLAM simultaneous localisation and mapping
SURF Speeded-Up Robust Features

1. Introduction

Space debris consists of the collective inoperative man-made objects currently orbiting the Earth. It represents a growing threat for both current spacecraft and future launches, as it is estimated that nearly $85\,\%$ of current space objects are classified as space waste [1]. Indeed, it has now been shown that this growth is being fuelled by chain reactions of collisions among objects that started as far back as 2007, a phenomenon which is termed *Kessler syndrome*. This fact justifies that debris mitigation strategies must be applied efficiently, whereas international rules state that at least five large space objects per year must be de-orbited in order to ensure long-term space operations [2].

One such strategy is termed ADR, where a chaser spacecraft is deployed to perform a NCRV with the target object in order to capture and de-orbit it. The *e.Deorbit mission* is set to be the first ADR mission to be carried out by the European Space Agency (ESA), demonstrating the removal of a large ESA-owned object from its current orbit and performing a controlled re-entry into the atmosphere. The Envisat spacecraft, launched on 1 March 2002 and non-functional since 9 May 2012, is a possible

© The Authors. Published by: International Astronautical Federation. This is the publisher Version of Record Manuscript.
Please refer to any applicable publisher terms of use.

target as it is one of the few ESA-owned debris in low Earth orbit, has a heavy mass, and is located in a crowded orbit [3].

e.Deorbit is part of ESA's CleanSpace initiative, which is focused on outlining the required technology for this domain, including advanced image processing (IP) for the relative navigation aspect of the NCRV. Using low-power-low-cost monocular camera-based systems, two-dimensional *features* of the target image can be identified and extracted to yield a relative navigation solution; in particular, when the target is not known accurately, such as in the case of a defunct satellite damaged by collisions with debris, these features are often *tracked* in time using matching algorithms and processed with visual mapping methods, such as structure from motion (SFM) or simultaneous localisation and mapping (SLAM). As the space environment may prove hostile to solutions in the visible wavelength due to illumination, approaches to ADR in other spectra have been proposed, such as the thermal infrared (IR) [4].

Whereas studies comparing the general performance of IP algorithms in the visible [5–7] and in the thermal infrared [8] are present in the literature, fewer exist which apply it in a space NCRV context [9], and none were found to do so in the infrared domain. Therefore, the purpose of this paper is to benchmark the performance of IP techniques adjusted towards multispectral camera setups than can be inserted in space rendezvous missions using affordable, low performance computing.

The structure of this paper is organised as follows. In Section 2, a background of image processing given, which includes a description of the assessed algorithms as well as the figures of merit used to benchmark them. Section 3 describes the devised experimental setup, namely the details of a new multispectral dataset generated for this experiment and the outline of the implementation of the performance analysis tool. Section 4 showcases and discusses the attained results with the tool. Lastly, Section 5 presents the conclusions of the study.

2. Background

Each relative navigation mission using imaging systems must consider performance figures to assess the viability of the IP algorithms used. This section analyses these figures of merit for the selected feature detectors and descriptors operating on the visible and thermal infrared spectra in the devised scenario, and provides a theoretical background for these algorithms.

2.1 Feature Detectors

The analysed detectors can be classified into two groups. The first group consists of *corner detectors*, i.e. algorithms that extract points defined as the intersection of two edges. The Harris corner detector [10], a historically influential algorithm in computer vision, works through the minimisation of the auto-correlation function that compares an image patch against itself shifted for small increments. Harris and Stephens showed that this is equivalent to testing the pair of eigenvalues of the auto-correlation matrix for a candidate point: if both are sufficiently large, it corresponds to a minimum in the function and hence to a corner that can be tracked reliably. The test itself is implemented through a corner response function that depends on an empirical constant and quadratically on the eigenvalues; the point is then selected if the response is greater than a given threshold. The Good Features To Track (GFTT) algorithm [11] improves on the previous method by defining a different corner response function: since the larger uncertainty component in the location of a matching patch is in the direction corresponding to the smallest eigenvalue, the proposed corner response function is merely dependent on it. Research continued to evolve in the sense of devising swift interest point detection methods. One such algorithm is Features from Accelerated Segment Test (FAST) [12], which is based on analysing the intensity of a fixed number of pixels around a candidate. FAST makes use of machine learning techniques by having built a decision tree from alternative training images that suggests which pixels should be assessed first on the test images in order to exclude a large number of non-corners, hence improving detection speed.

Conversely, the second group considers *blob detectors*, which extract points taking into account a support neighbouring region. This class of algorithms attempts to tackle many of the drawbacks of simple corner detectors, such as invariance to scale changes. The Laplacian of Gaussians (LoG) operator is often utilised to this end as the resulting function is sensitive to corners and edges [13]. However, the LoG involves the computation of second-order derivatives which are both sensitive to noise and computationally expensive. The Difference of Gaussians (DoG) [14] detector uses its namesake technique to approximate the LoG: it computes the difference between two Gaussian blurs of the same image with different standard deviations σ separated by a constant factor, and successive blurrings are done until the last layer is transformed with a value of twice the initial σ. Once a complete octave is processed, this layer is downsampled by a factor of 2, marking the start of the following octave. The resulting structure is then searched in scale and space for stable points (Fig. 1). The two other blob detectors benchmarked in this paper, Fast-Hessian [15] and Centre Surround Extrema (CenSurE) [16], are focused on further approximat-

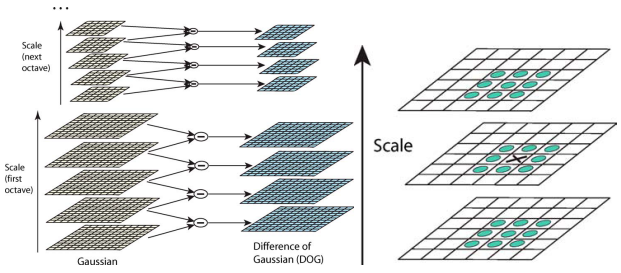


Fig. 1: The difference of Gaussian pyramid structure. (Reproduced from [14]) (a) Adjacent Gaussian images are subtracted to produce the DoG images, each octave is characterised by downsampling the previous one by a factor of 2 (b) Features are selected in the DoG images by comparing a candidate point (X) to its neighbours (green)

ing the LoG. The former utilises box filters, which can be evaluated swiftly independently of size using integral images, whereas the latter employs bi-level center-surround filters that mitigate subsampling accuracy effects at larger scales.

2.2 Feature Descriptors

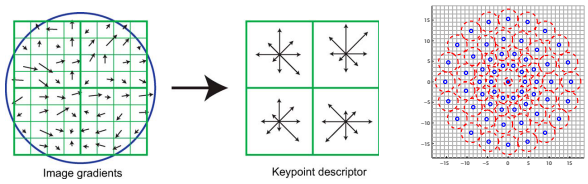
The present work considers three *floating point* type, or distribution-based, descriptors and three *binary* type descriptors. Distribution-based descriptors are called as such since they encode (in a floating point vector) how certain elements of the support region to the feature point are distributed around it. The first considered descriptor of this type, Scale-Invariant Feature Transform (SIFT) [14], calculates a histogram of local oriented gradients for each 4×4 subregion around the keypoint. Each histogram has a resolution of 8 bins, giving the descriptor vector a size of 128 elements (Fig. 2a). The Speeded-Up Robust Features (SURF) algorithm [15] proposes a refinement to this process by computing instead the Haar wavelet responses and storing them in a four-dimensional descriptor vector for each subregion, making up for a total of 64 elements. The more recent Local Intensity Order Pattern (LIOP) descriptor [17] is also benchmarked: the overall intensity order (i.e. the order acquired by sorting pixels by increasing intensity) is used to divide the local patch into subregions labelled ordinal bins; a LIOP of each point is defined based on the relationships among the intensities of its neighbouring sample points inside each bin; lastly, the descriptor for the patch is constructed by concatenating the LIOPs of each bin together.

The second type of considered local feature descriptor differs from the previous one in the sense that, instead of using a floating point vector representation, each descriptor consists of a binary string. For each feature point, a binary descriptor typically samples sets of pixel pairs $(\boldsymbol{x}_1, \boldsymbol{x}_2)_i, i \in n$ from the support patch, and performs a simple intensity comparison, where the result is 1 if $I(\boldsymbol{x}_1) < I(\boldsymbol{x}_2)$, and 0 otherwise, generating an n-dimensional bit string. To this end, Oriented FAST and Rotated BRIEF (ORB) [18] favours an isotropic Gaussian distribution pattern, rotated according to the orientation of the extracted feature. The algorithm used a training phase consisting of a greedy search algorithm to go through all the possible binary tests and select those that maximise the variance of the descriptor vector. Binary Robust Invariant Scalable Keypoints (BRISK) [19] applies a sampling pattern made up of n locations equally spaced on circles concentric with the interest point (Fig. 2b). Two subsets are defined in accordance with two scale-proportional thresholds: one of short-distance pairings and another of long-distance pairings. The gradients of the long-distance pairs are used to compute the overall characteristic pattern direction of the feature. After that, the pattern is rotated accordingly and the binary descriptor string is assembled by performing all the short-distance intensity comparisons of pixel pairs. This allows a single point to take part in more comparisons, limiting the complexity of the intensity values look-up process. Fast Retina Keypoint (FREAK) [20], on the other hand, adopts the human retinal sampling grid as the sampling pattern for the pixel intensity comparisons: it is a circular geometry where the density of points drops exponentially from the centre outwards, mimicking the spatial distribution of ganglion cells in the eye. A coarse-to-fine pair is shown to yield the largest variance and uncorrelation between pairs, i.e. the first selected pairs compare sampling points in the outer circles and the last pairs compare points in the inner circles. The first 16 bytes encode the coarse information, which is applied as a triage in the matching process, accelerating the procedure even further.

Table 1 highlights the differences between the descriptor types. Note that many of these algorithms were designed for detection as well as description. Indeed, DoG and Fast-Hessian are part of SIFT and SURF, respectively, and ORB and BRISK both use a FAST-based method for feature detection in their original implementations.

2.3 Performance Metrics

In order to evaluate the algorithms, the concept of *correspondence* is first defined: two regions, a and b, each from a different image, are said to be correspondences if the second region, when mapped to the first image, has an overlap with the first region higher than a defined thresh-



(a) (b)

Fig. 2: Distribution-based and binary description. (a) SIFT: the gradient magnitude and orientation at each subregion are weighted by a Gaussian window (blue) and accumulated into a histogram (reproduced from [14]) (b) BRISK: sampling locations (blue) and Gaussian kernels to smooth intensity values (red) for $n = 60$ points (reproduced from [19])

Table 1: Descriptors characteristics. (Adapted from [7])

Descriptor	Data Type	# Elements	Size [bytes]	Matching Type
SIFT	Floating point	128	512	Euclidean norm
SURF	Floating point	64	256	Euclidean norm
LIOP	Floating point	144	576	Euclidean norm
ORB	Binary	256	32	Hamming norm†
BRISK	Binary	512	64	Hamming norm
FREAK	Binary	512	64	Hamming norm

† The Hamming distance between two strings of equal length is defined as the minimum number of substitutions required to convert one into the other.

old (Fig. 3). Formally, the following condition must hold:

$$1 - \frac{R_{\boldsymbol{M}_a} \cap R_{(\boldsymbol{H}^T \boldsymbol{M}_b \boldsymbol{H})}}{R_{\boldsymbol{M}_a} \cup R_{(\boldsymbol{H}^T \boldsymbol{M}_b \boldsymbol{H})}} < \varepsilon_0, \tag{1}$$

where $R_{\boldsymbol{M}}$ represents the elliptic region defined by $\boldsymbol{x}^T \boldsymbol{M} \boldsymbol{x} = 1$ and ε_0 is the overlap error threshold. This mapping, the ground truth, can be given by a 3×3 homography matrix $\boldsymbol{H}$, assuming a pinhole camera model and that the two related images represent same planar surface in space.

Consequentially, the *repeatability score* for a given pair of images is calculated as the ratio between the number of correspondences and the number of total features presented in the reference image:

$$\text{repeatability} = \frac{C^+}{C}. \tag{2}$$

A second type of testing performed is based on the *matching score*. This test verifies how well the regions can be algorithmically matched, thus assessing the distinctiveness of the detected regions. To this end, a descriptor for the regions is computed and the total matches M^* provided by it are checked to see if they agree with the correspondences obtained with $\boldsymbol{H}$. If a matched pair

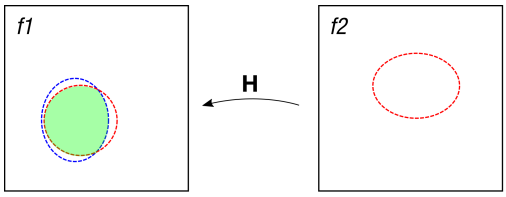


Fig. 3: The homography ground truth $\boldsymbol{H}$ maps the red feature from frame $f2$ to frame $f1$. The overlap area with the original blue feature from $f1$ is shown in green. If the amount of green is above a certain defined threshold, then the two features correspond.

is also a correspondence, then it is deemed a correct match M^+, contributing to the matching score as

$$\text{matching score} = \frac{C^+ \cap M^*}{C} = \frac{M^+}{C}. \tag{3}$$

To put it in short, features are desired to be repeatable, i.e. the same features should be observed regardless of how the target is manipulated, but they should also be distinctive enough so that they can be matched regardless of those transforms.

To evaluate the performance of feature descriptors, the figures of recall and precision are used. *Recall* is defined as the ratio of correct matches to the number of correspondences between a pair of frames:

$$\text{recall} = \frac{M^+}{C^+}. \tag{4}$$

On the other hand, *precision* is the ratio of correct matches to the total number of matches:

$$\text{precision} = \frac{M^+}{M^*}. \tag{5}$$

This performance metric is occasionally represented as its complement, i.e. 1-precision, the ratio of false matches to the total matches. For the ideal case, the recall and the precision would both be close to 1, meaning that the descriptor would return a great number of matches, all labelled correctly. A descriptor with high recall and low precision would translate into a great number of matches but many of them are false positives. Lastly, a descriptor with low recall and high precision would mean a small number of returned matches, but most of them are correct.

Note that the definition of a match is dependent on the chosen strategy. For the scope of this work, the concept of nearest-neighbour distance ratio (NNDR) is used: two regions are a match if the ratio of the distance to the first and

to the second nearest neighbouring descriptors is below a certain threshold μ:

$$\text{NNDR} = \frac{\|D_b - D_a\|}{\|D_c - D_a\|} < \mu, \tag{6}$$

where D_b, D_c are the first and second nearest neighbours to D_a, respectively. Results from [6, 8] show that using a NNDR improves the matching precision relative to other strategies.

The performance of different descriptors is often compared by generating for each one sets of recall and 1-precision values with varying values of μ. The plotted points result in a receiver operating characteristic (ROC) curve [21]. The larger the area under a descriptor's ROC curve, the better its performance, providing an intuitive way to benchmark descriptors.

The average computation times per extracted and described feature are benchmarked, respectively, for each detection and description algorithm. This assumes a proportionality between the required time and the computation burden, which can then be of interest to make an informed choice on the algorithm for a given application.

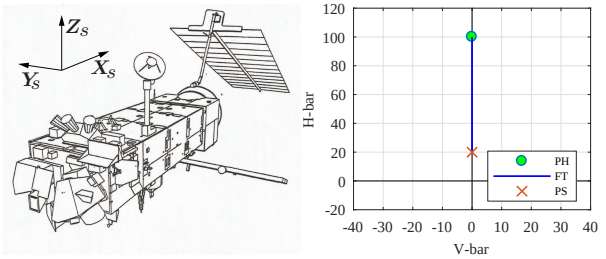
3. Experimental Setup

In this section, the experimental setup arranged to evaluate the performance of the IP algorithms is described. The generated datasets are delineated, and details of the implementation of the algorithms are outlined.

3.1 Dataset

A chaser spacecraft is assumed to approach the target with the translational profile relative to the local-vertical-local-horizontal (LVLH) reference frame illustrated in Fig. 4. The spin axis of the target in the body frame is aligned with the $+Ys-$axis, and the spin axis in the LVLH frame is aligned with the $+H-$bar axis. The chaser assumes a constant orientation with regards to the target's LVLH frame. The sequence begins with the chaser in a hold point (PH) $100\,\mathrm{m}$ away from the target. The rendezvous sequence is performed through a forced translation $H-$bar approach (FT) with the target until a stop point (PS) is reached at $20\,\mathrm{m}$ distance, after which the sequence ends. This ensures an approximate planarity of the scene between consecutive image frames, allowing the computation of the ground truth using a homography (Section 2).

The current orbit of Envisat is estimated using the two-line element (TLE) data of 30 October 2017.* This corresponds to a situation where the target spacecraft is in full sunlight.



(a) (b)

Fig. 4: Scenario specifications for dataset generation. (a) Envisat body reference frame axes (b) Chaser trajectory relative to target LVLH frame

Table 2: Camera properties.

Parameter	**Specification**
Resolution [px^2]	320×256
Focal length [mm]	5
Field-of-view [deg^2]	51×40
Measurement rate [Hz]	1

The chaser has two body-mounted cameras in a monocular setup facing the target, one capturing images in the visible band (0.39-0.70 μm) and the other in the thermal infrared (8-14 μm). A 3D computer aided design (CAD) model of Envisat is used to generate the synthetic dataset. The original textured model was heavily modified to guarantee a realistic simulation in the visible spectrum. This included re-meshing the main body of the spacecraft to emulate a "crumpled" effect for the multi-layer insulation and adding reflective effects to the solar panel. A second model was created embedded with material temperature and emissivity for thermal infrared simulation based on the analysis by Ref. [4], where it is assumed that the temperature of each component has reached a steady state.

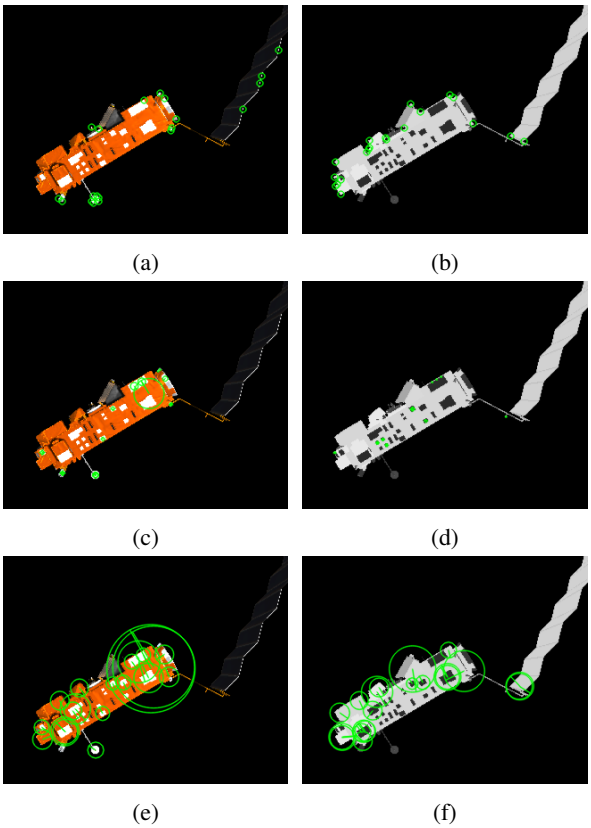
The relative trajectory is imaged using the Astos Camera Simulator software. In total, a camera sequence of 200 frames is generated for each of the two modalities. The characteristics of the cameras are displayed in Table 2. Samples of generated images and of detected features are illustrated in Fig. 5. The number of plotted features is limited to 20 for clarity.

3.2 Implementation

The performance analysis framework was coded in the C++ programming language. The OpenCV† library, ver-

*TLE data obtained from NORAD Two-Line Element Sets Current Data `http://www.celestrak.com/NORAD/elements/`

†`http://http://opencv.org/`



(a) (b) (c) (d) (e) (f)

Fig. 5: Examples of detected features (green) on frames from the dataset. Left: visible modality. Right: thermal infrared modality. (a, b) Harris (c, d) DoG (e, f) Fast-Hessian

sion 3.3, was used for computer vision and image processing related functions.

The native implementations for the Harris, GFTT, DoG, Fast-Hessian, and FAST algorithms are used. For CenSurE, an OpenCV emulation of the original algorithm termed "Star", was considered. For the descriptors, the native implementations were used for all but LIOP, which was not available. Instead, the original author's own code[‡] was ported manually into the present framework.

The ground truth homographies mapping each frame to the previous one in the sequence are computed *a priori* using the enhanced correlation coefficient (ECC) maximization algorithm [22].

To verify the computing performance of the IP methods, these were implemented and tested on a BeagleBone(R) Black (BBB) single-board computer with a 1 GHz ARM Cortex-A8 processor and 512 MB DDR3 RAM (Fig. 6 and Table 3).

[‡]https://github.com/foelin/IntensityOrderFeature

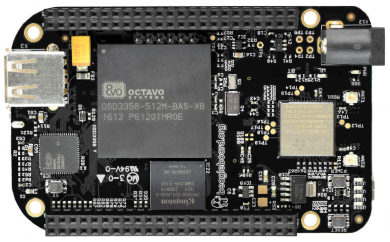


Fig. 6: The BeagleBone(R) Black wireless single board computer.

Table 3: BeagleBone(R) Black properties.

Parameter	Specification
System on a Chip (SoC)	AM3358/9
CPU	Cortex-A8 1 GHz
Digital Signal Processor	N/A
On-board storage	8 bit eMMC (running Ubuntu 16.04), microSD card 3.3 V supported
Memory	512 MB DDR3
Size	86.40 mm × 53.3 mm
Power ratings	210–460 mA at 5 V

4. Results and Analysis

In this section, the results of the adopted framework to determine the performance of the algorithms on the multispectral dataset are delineated.

The different parameters that have a potential influence on the algorithms' benchmarking setup was analysed *a priori*. In particular, the following variables were tuned: the overlap error threshold, the normalised region size, and the number of extracted features. The significance of the first variable stands on the fact that it decides how strict the definition of a correspondence is. The importance of the second parameter is related to the fact that the tool could favour detectors with large regions in terms of the repeatability score, so normalising each region size prior to the computation of the features of merit ensures a fairer setup to compare the detectors. The last property is considered to avoid bias towards dense responses, since each detector would extract a different number of features if this tuning was otherwise neglected.

4.1 Benchmarking of Feature Detectors

For this test, the repeatability scores obtained by each detector for each full sequence is analysed. In addition, the matching scores are computed. This is done using the

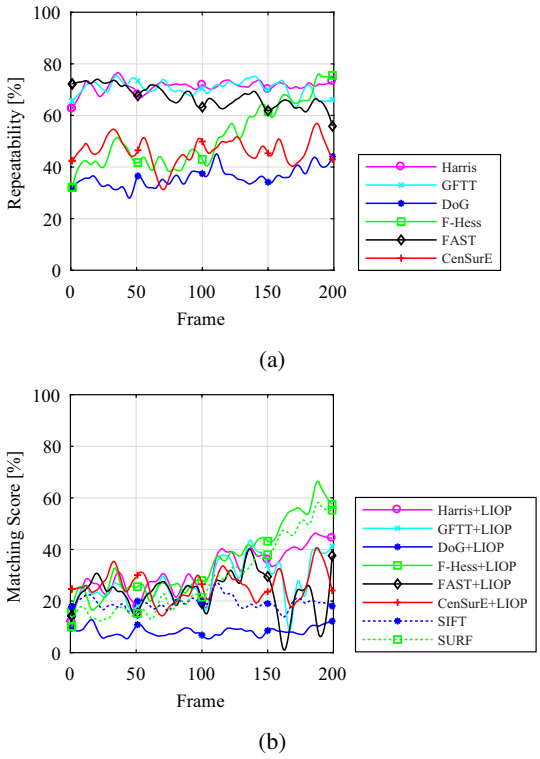


Fig. 7: Performance for rendezvous sequence: visible band. (a) Repeatability (b) Matching score. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor

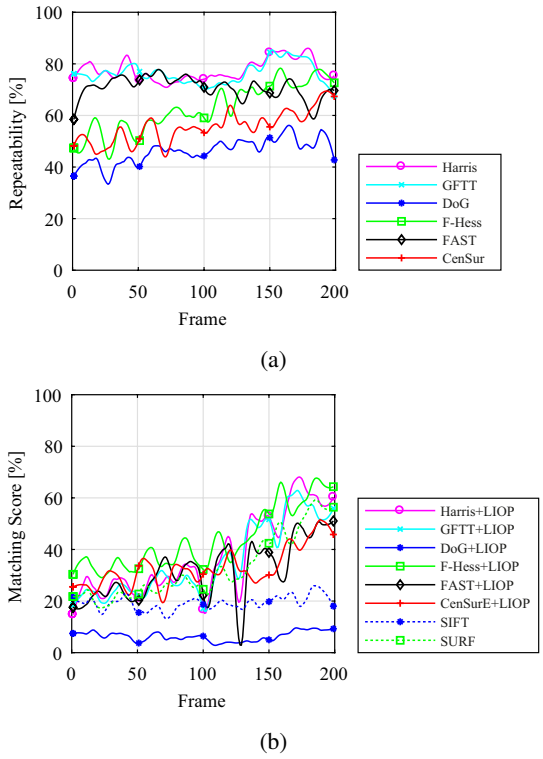


Fig. 8: Performance for rendezvous sequence: thermal infrared band. (a) Repeatability (b) Matching score. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor

LIOP descriptor. This descriptor was chosen as it is independent from all the detectors considered. Since the goal is to study the performance of the different feature extraction processes, this avoids any bias towards a specific detector, allowing for the examination of the features' distinctiveness regardless of the chosen descriptor. For added comparison, the performance using the original descriptors for DoG and Fast-Hessian (SIFT and SURF, respectively) is also showcased to benchmark the full original algorithms and provide a baseline.

An overlap error threshold of 30%, a normalised region size of 7.5 px, and a fixed number of 75 extracted features for each detector are considered.

The benchmarks are plotted in Figs. 7 and 8. Consecutive image transforms, which are commonly done in SFM and visual SLAM algorithms, are analysed; in this case, a value pertaining to frame k in the plot is referent to the transformation between frames k and $k+1$.

Visible Modality Fig. 7 showcases the performance of the detection algorithms for the approach sequence in the visible wavelength during a sunlight period. Harris, GFTT, and FAST achieve the highest repeatability scores. However, in terms of matching scores, they are comparable to Fast-Hessian and CenSurE, where the former actually outperforms the rest towards the end of the sequence, showing a bias in favour of shorter target ranges. Conversely, the matching scores when using GFTT and FAST actually decrease as the chaser nears the target, meaning that the high repeatability attained was likely due to accidental overlapping of features. This could represent a problem when using these detectors with visible imagery at close proximity. CenSurE is the most consistent algorithm throughout. Note from Fig. 7b that Fast-Hessian shows a better performance when coupled with LIOP than when combined with its native descriptor. On the other hand, DoG when used with its native SIFT is shown to perform the worst.

Thermal IR Modality Fig. 8 shows the results attained for the rendezvous sequence observed in the thermal infrared band during sunlight. The algorithms suggest robustness in this modality with high repeatability scores overall (notably in the case of the blob detectors: DoG,

Fast-Hessian, and CenSurE) and matching scores increasing with time. Note that FAST shows significant declines in the matching score despite its high repeatability, illustrating lower feature distinctiveness when compared with the other corner detectors. Fast-Hessian again scores one of the highest benchmarks in general.

4.1.1 Discussion

Despite being imaged in two different modalities, the two sequences analysed include common relative motion. Therefore, some similarities in the results are expected. The repeatability, in particular, is comparable for both sequences. The same cannot be said about the matching scores, however: despite scoring generally lower than the repeatability, they vary in trend and relative ranking between sequences. This highlights the importance in using descriptors to compute matches instead of relying on the geometry overlap only, and implies different degrees of distinctiveness in extracted features depending on the detector and wavelength considered.

Corner detectors score higher in repeatability. However, they are often equalled or even surpassed by the blob detectors in terms of matching score. Despite high repeatability, FAST is one of the least distinctive algorithms across both tests. Fast-Hessian performs well in terms of matching scores in both cases despite average repeatability. This suggests an extraction of quite distinctive features, which confirms what was stated in the thermal IR analysis of Ref. [8] and extends the conclusions to the visible spectrum. This is an important finding as it is desirable to have a detector that works well in both spectra. DoG shows low scores regardless of the wavelength, but seems to perform worse on the IR. It performs better with SIFT than with LIOP in every situation, whereas Fast-Hessian usually performs better with LIOP than SURF. This reiterates the importance of testing detectors and descriptors separately to avoid any cause of bias.

Corner detectors are shown to lose in performance when the target is closer on the visible. This could signify that they are more sensitive to noise coming from the multi layer insulation, for example, as they perform quite well on the textureless IR. In the latter case, the "physical" corners are more evident and impervious to illumination changes.[§] On the same note, performance is generally better for the thermal IR case: CenSurE and Fast-Hessian, in particular, are comparable to the visible case, but the former performs better than its visible counterpart in the end of the sequence where the latter does so in the beginning of it.

[§]The reader is reminded that a steady-state thermal condition was considered for the IR sequence.

4.2 Benchmarking of Feature Descriptors

For this test, the performance of the descriptors is assessed. To this end, a comparison is done using the same feature detector for all the descriptors in order to reduce the influence of the former on the results. Similar settings as in the previous experiments were used, that is, an error threshold of 30% and a fixed number of 75 extracted features. The regions are not normalised in the computation of the descriptors.

The efficiency of the algorithms is evaluated by computing their ROC, or recall/1-precision, curves. Two sets of results are shown for each sequence: the first is a descriptor benchmark for short (sucessive) image transformations using the DoG detector; and the second repeats the same experiment using Fast-Hessian. This allows insight into if and how different detector-descriptor combinations affect the outcomes. These are plotted in Figs. 9–10.

Visible Modality Fig. 9 illustrates the attained ROC curves for the visible modality during the sunlight period. It can be seen that the performance of the descriptors depends on the feature detection algorithm used: Fast-Hessian features are shown to yield better precision.

It is interesting to note that SIFT performs better with Fast-Hessian features (Fig. 9b) than with DoG features (Fig. 9a). Indeed, when DoG features are used, SIFT performs best, followed by ORB and BRISK, and the performance of the three descriptors is comparable.

For Fast-Hessian features, BRISK, FREAK, and LIOP give the best results. Overall the results obtained for SURF are sub-par, showing that combining a feature detector with a non-native descriptor can yield better results.

Thermal IR Modality Here, the descriptors are compared for the case of the thermal infrared imaging of the sequence during sunlight conditions; the results are shown in Fig. 10. The performance computed on DoG features follows the same trend as for the visible case, albeit with a lower yielded precision, which signifies more false matches.

On the other hand, when using Fast-Hessian features the descriptors perform better than for the visible case, in general. FREAK obtains the highest score, but as in the visible case, it behaves quite similarly to BRISK, SIFT, and LIOP. SURF and ORB plot similar curves to their visible counterparts, indicating that they are robust to changes in frequency when computed on Fast-Hessian features.

4.2.1 Discussion

The general conclusion from this analysis seems to be that the performance of the descriptors is dependent on the feature they are applied on, regardless of descriptor type.

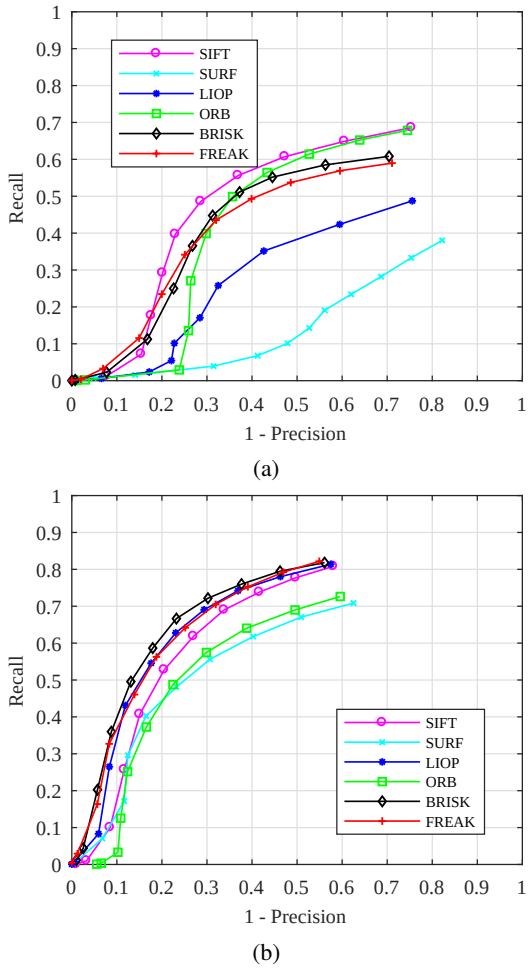


Fig. 9: Descriptor ROC curves for the visible band, hot case. (a) DoG features (b) Fast-Hessian features

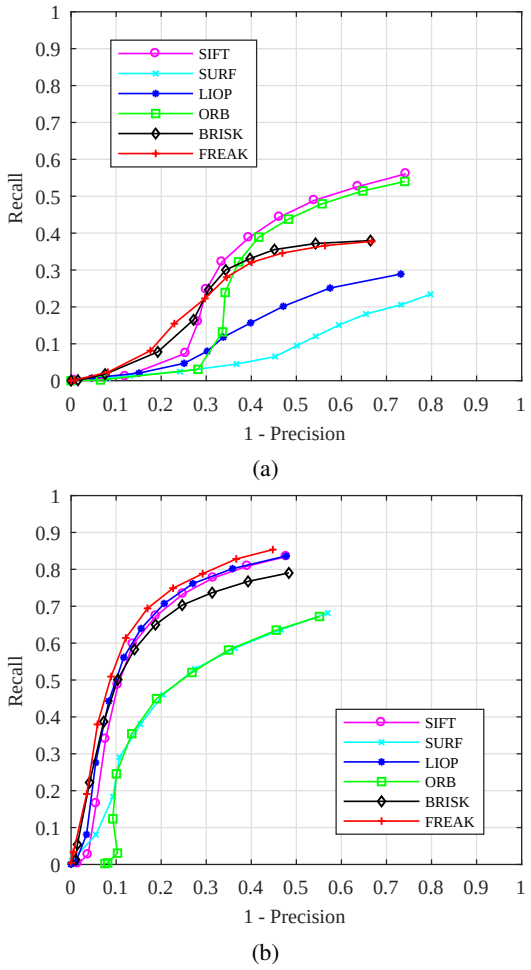


Fig. 10: Descriptor ROC curves for the thermal IR band, hot case. (a) DoG features (b) Fast-Hessian features

Fast-Hessian performs better in general both in terms of recall and precision scores, regardless of the modality. As theorised by Ref. [8], a possible explanation for this could be the fact that Fast-Hessian usually extracts larger blobs than DoG, so a larger support area is considered in the computation of the descriptor, capturing in principle a larger signal variation. In can be seen by inspecting Fig. 5 that this is also the case for the analysed dataset.

LIOP was shown to perform better when computed on Fast-Hessian features, both on the visible, and as reported in Ref. [8] on the IR. It can be ranked amongst the best descriptors when used with this type of feature.

Overall, SIFT, BRISK, and FREAK are ranked among the best descriptors for all cases.

4.3 Computation Times

In this subsection, the IP algorithms are benchmarked in terms of their computational performance. These tests are ran on the single board computer setup, allowing for the examination of their real-time capacity on a low performance embedded system. The recorded benchmarks account only for the core tasks of detection or description.

Table 4 portrays the average extraction time per feature for each detector. DoG scores the slowest detection time, at $21.2\,\mathrm{ms}$ per feature. To better compare their performance, in addition to the absolute computation times, the relative speed-up factors with respect to the heaviest algorithm are also displayed. FAST is the quickest algorithm to run, being almost three orders of magnitude swifter than DoG. As expected (see Section 2), CenSurE is faster than Fast-Hessian, which is in turn faster than DoG. Surprisingly, GFTT is recorded having a higher execution time than Harris.

It is clear from Table 4 that, given the minimum number of features considered (75), DoG is the only detector to violate the given computational budget of $1000\,\mathrm{ms}$,

Table 4: Average detection times per feature.

Detector	Time [ms]	Speed-up
FAST	0.0261	814
CenSurE	1.3189	16
Harris	1.4248	15
GFTT	1.4933	14
F-Hess	2.6338	8
DoG	21.2249	1

Table 5: Average description times per feature.

Descriptor	Time [ms]	Speed-up
ORB	0.1627	103
BRISK	0.2057	81
SURF	0.7676	22
SIFT	9.4847	2
LIOP	14.5418	1
FREAK	16.7328	1

whereas the other algorithms execute leaving a considerable amount of time left for allocating other tasks.

Analogously, Table 5 shows the benchmarked computation times for the descriptors averaged per feature. While the list is topped by two of the binary descriptors, FREAK is actually the slowest algorithm, costing $16.7328\,\mathrm{ms}$ per feature on average. The high computation time is unusual for a binary descriptor and contradicts the findings of Ref. [8]. The authors of the cited research work benchmark the descriptor computation times on an Intel(R) 64-bit processor, which could suggest limitations or different levels of optimisation of the algorithms in different architectures.

LIOP is similar in performance to FREAK, while SIFT is two times faster. The former two algorithms exceed the computational budget, though, while the latter takes up $71\,\%$ of it. Surprisingly, the performance of SURF is in the same order of magnitude as ORB and BRISK.

5. Conclusions

This paper found that electing a key detector-description combination that outperforms all others is not an straightforward task, as it comes down to a trade-off mainly between robustness and complexity. However, given the hardware limitations it is possible to point out which IP algorithms are more adequate for the studied scenario. A combination of Fast-Hessian with BRISK is capable of providing adequate performance, both in terms of matching performance and computational efficiency, as it was shown to perform inside the boundaries of the considered frame-rate, taking up little over $20\,\%$ of the computational budget. Hence, this combination could possibly be used for a multispectral navigation algorithm, analysing a frame of each modality per cycle, and it would still achieve detection and description with less than half of the budget. Furthermore, the benchmark of Fast-Hessian + BRISK is comparable in both spectra, which is an added advantage as a reduced number of algorithms means reduced memory usage.

Given the conducted analysis, it should be noted that other detector-descriptor combinations that comply with the hardware requirements are possible. Recommendations include additional experimentation with algorithms besides the ones tested herein, e.g. ORB with its native descriptor. Future work includes analysing the performance of IP algorithms under additional perturbations (e.g. sunlight vs. eclipse) and expanding the analysis to benchmark large image transformations.

6. Acknowledgements

The research work detailed in the present paper has been funded by ESA contract no. 4000117583/16/NL/HK/as.

The authors would like to thank ESA for access to the Astos Camera Simulator.

References

[1] M. Andrenucci, P. Pergola, and A. Ruggiero, "Active Removal of Space Debris: Expanding Foam Application for Active Debris Removal," tech. rep., University of Pisa, Aerospace Engineering Department, Italy, 2011.

[2] C. Bonnal, J.-M. Ruault, and M.-C. Desjean, "Active Debris Removal: Recent Progress and Current Trends," *Acta Astronautica*, vol. 85, pp. 51–60, 2013.

[3] R. Biesbroek, L. Innocenti, A. Wolahan, and S. M. Serrano, "e.Deorbit–ESA's Active Debris Removal Mission," in *7th European Conference on Space Debris*, ESA Space Debris Office, 2017.

[4] Ö. Yılmaz, N. Aouf, E. Checa, L. Majewski, and M. Sanchez-Gestido, "Thermal Analysis of Space Debris For Infrared-Based Active Debris Removal," *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 2017.

[5] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A Comparison of Affine Region Detectors," *International Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[6] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[7] O. Miksik and K. Mikolajczyk, "Evaluation of Local Detectors and Descriptors for Fast Feature Matching," in *Pattern Recognition (ICPR), 2012 21st International Conference on*, pp. 2681–2684, IEEE, 2012.

[8] T. Mouats, N. Aouf, D. Nam, and S. Vidas, "Performance Evaluation of Feature Detectors and Descriptors Beyond the Visible," *Journal of Intelligent & Robotic Systems*, pp. 1–31, 2018.

[9] N. Takeishi, A. Tanimoto, T. Yairi, Y. Tsuda, F. Terui, N. Ogawa, and Y. Mimasu, "Evaluation of Interest-Region Detectors and Descriptors for Automatic Landmark Tracking on Asteroids," *Transactions of the Japan Society for Aeronautical and Space Sciences*, vol. 58, no. 1, pp. 45–53, 2015.

[10] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," in *Alvey Vision Conference*, vol. 15, p. 50, Citeseer, 1988.

[11] J. Shi and C. Tomasi, "Good Features To Track," in *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 593–600, IEEE, 1994.

[12] E. Rosten and T. Drummond, "Machine Learning for High-Speed Corner Detection," in *European Conference on Computer Vision*, pp. 430–443, Springer, 2006.

[13] T. Lindeberg, "Scale-Space Theory: A Basic Tool for Analyzing Structures at different scales," *Journal of Applied Statistics*, vol. 21, no. 1-2, pp. 225–270, 1994.

[14] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *International Journal of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[15] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded Up Robust Features," in *European Conference on Computer Vision*, pp. 404–417, Springer, 2006.

[16] M. Agrawal, K. Konolige, and M. R. Blas, "Censure: Center Surround Extremas for Realtime Feature Detection and Matching," in *European Conference on Computer Vision*, pp. 102–115, Springer, 2008.

[17] Z. Wang, B. Fan, and F. Wu, "Local Intensity Order Pattern for Feature Description," in *2011 International Conference on Computer Vision*, pp. 603–610, IEEE, 2011.

[18] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in *2011 International Conference on Computer Vision*, pp. 2564–2571, IEEE, 2011.

[19] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary Robust Invariant Scalable Keypoints," in *2011 International Conference on Computer Vision*, pp. 2548–2555, IEEE, 2011.

[20] A. Alahi, R. Ortiz, and P. Vandergheynst, "FReAK: Fast Retina Keypoint," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pp. 510–517, Ieee, 2012.

[21] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010.

[22] G. D. Evangelidis and E. Z. Psarakis, "Parametric Image Alignment using Enhanced Correlation Coefficient Maximization," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1858–1865, 2008.



https://dspace.org/

Multispectral image processing for navigation using low performance computing

Rondao, Duarte; Aouf, Nabil; Dubois-Matra, Olivier

2018-10
International Astronautical Federation (IAF)
https://iafastro.directory/iac/paper/id/43502/summary/
Duarte Rondao, Nabil Aouf and Olivier Dubois-Matra. Multispectral image processing for navigation using low performance computing International Astronautical Congress 2018 (IAC ' 18), 1-5 October 2018, Bremen, Germany.

Downloaded from a DSpace repository. https://dspace.org/