Acta Astronautica, Volume 172, July 2020, pp. 100-122
DOI:10.1016/j.actaastro.2020.03.049

Benchmarking of Local Feature Detectors and Descriptors for Multispectral Relative Navigation in Space

Duarte Rondao[a,1,*], Nabil Aouf[b,2], Mark A. Richardson[a,3], Olivier Dubois-Matra[c,4]

[a] *Cranfield University, Defence Academy of the United Kingdom, SN6 8LA Shrivenham, United Kingdom*
[b] *City, University of London, ECV1 0HB London, United Kingdom*
[c] *European Space Agency, ESTEC, Keplerlaan 1, 2201 AZ Noordwijk, Netherlands*

Abstract

Optical-based navigation for space is a field growing in popularity due to the appeal of efficient techniques such as Visual Simultaneous Localisation and Mapping (VSLAM), which rely on automatic feature tracking with low-cost hardware. However, low-level image processing algorithms have traditionally been measured and tested for ground-based exploration scenarios. This paper aims to fill the gap in the literature by analysing state-of-the-art local feature detectors and descriptors with a taylor-made synthetic dataset emulating a Non-Cooperative Rendezvous (NCRV) with a complex spacecraft, featuring variations in illumination, rotation, and scale. Furthermore, the performance of the algorithms on the Long Wavelength Infrared (LWIR) is investigated as a possible solution to the challenges inherent to on-orbit imaging in the visible, such as diffuse light scattering and eclipse conditions. The Harris, GFTT, DoG, Fast-Hessian, FAST, CenSurE detectors and the SIFT, SURF, LIOP, ORB,

*Corresponding author
Email addresses: d.rondao@cranfield.ac.uk (Duarte Rondao), nabil.aouf@city.ac.uk (Nabil Aouf), m.a.richardson@cranfield.ac.uk (Mark A. Richardson), olivier.dubois-matra@esa.int (Olivier Dubois-Matra)
[1]PhD Candidate, Centre for Electronic Warfare, Information and Cyber.
[2]Professor of Robotics and Autonomous Systems, Department of Electrical and Electronic Engineering.
[3]Professor of Electronic Warfare, Centre for Electronic Warfare, Information and Cyber.
[4]Engineer, Guidance, Navigation and Control Section.

Published by Elsevier. This is the Author Accepted Manuscript issued with: Creative Commons Attribution Non-Commercial No Derivatives License (CC:BY:NC:ND 4.0).
The final published version (version of record) is available online at DOI:10.1016/j.actaastro.2020.03.049. Please refer to any applicable publisher terms of use.

BRISK, FREAK descriptors are benchmarked for images of Envisat. It was found that a combination of Fast-Hessian with BRISK was the most robust, while still capable of running on a low resolution and acquisition rate setup. For large baselines, the rate of false-positives increases, limiting their use in model-based strategies.

Keywords: Benchmarking, Feature detectors, Feature descriptors, Multispectral imaging, Space relative navigation
2010 MSC: 68T45

1. Introduction

In 2007, the number of catalogued space objects orbiting the Earth suddenly grew by approximately 26 % [1]. It has now been shown that this hike was due to a major event in low Earth orbit which resulted in the exponentiation of the number of fragmentation debris, a phenomenon which had been predicted as far back as 1978 by Kessler and Cour-Palais [2]. The "Kessler Syndrome", as it is designated, suggests that space debris can grow irrespective of newer spacecraft launches simply due to cascading collisions between orbiting, most likely derelict, spacecraft. Such a phenomenon is capable of precipitating the arrival of a point of no return beyond which human intervention becomes futile, rendering space operations permanently unfeasible. However, the number of Earth-orbiting debris had been steadily growing even before 2007. In fact, they now outnumber active spacecraft by more than 5 to 1, inhabiting mainly the orbits commonly targeted for launches, i.e. Low Earth Orbit (LEO) and Geostationary Orbit (GEO) [3].

A potential chain reaction trigger is the Envisat spacecraft: a sizeable spacecraft in LEO weighing over 8000 kg, launched on 1 March 2002 and non-functional since 9 May 2012. The existence of such space objects 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 [4].

One such mitigation strategy is termed Active Debris Removal (ADR), where a chaser spacecraft is deployed to perform a Non-Cooperative Rendezvous (NCRV) with the target object in order to capture and de-orbit it. The e.Deorbit mission is set out to be the first ADR mission to be carried out by the European Space Agency (ESA), demonstrating the removal of a large object from its current orbit and performing a controlled re-entry into the atmosphere. As one of the few ESA-owned debris in LEO, Envisat is a possible target for the mission [5].

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 rendezvous operations. A smaller scale in-orbit demonstration mission using CubeSats to test IP algorithms has been proposed: e.Inspector, as it is called, would visually inspect Envisat to determine its tumbling rate and axis. This data could then be used for validation purposes to use with e.Deorbit [6]. Indeed, using low-power-low-cost camera-based systems, two-dimensional features of the target image can be identified and extracted to yield a relative navigation solution. 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 Long Wavelength Infrared (LWIR), also known as thermal infrared [7].

Although studies comparing the general performance of IP algorithms in the visible and in LWIR are present separately in the literature, benchmarks performed in a space NCRV context are scarce. Furthermore, no LWIR IP comparisons for NCRV were found to exist, to the best of the authors' knowledge. Therefore, the purpose of this paper is to benchmark the performance of IP techniques adjusted towards multispectral camera setups than can be inserted in ADR missions using affordable, low performance computing (Fig. 1).

The outline of the paper is as follows. Section 2 offers the motivation behind the study, as well as a review of related work. Section 3 defines the theoretical aspects of the algorithms to benchmark and the figures of merit used herein. Section 4 delineates the experimental setup. Section 5 illustrates the attained

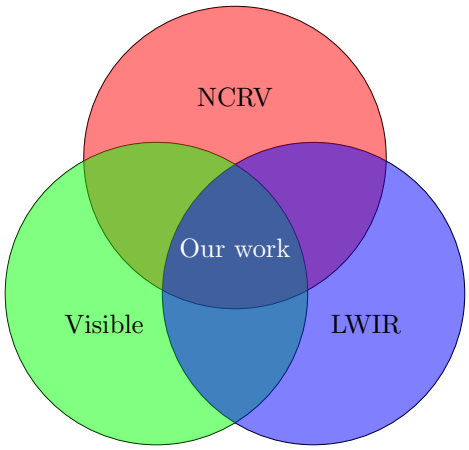


Figure 1: The domain of the present paper. While literature exists on one or the intersection of two domains represented by each circle, this study introduces a connection between all three.

results and discusses them. Lastly, Section 6 concludes the paper.

2. Background and Related Works

2.1. The Camera as a Navigation Sensor

The Attitude and Orbit Control System (AOCS) is a fundamental subsystem of many spacecraft. Typically, space missions are designed with inertial navigation in mind, working in reference frames centred on the Sun or the Earth. For instance, a geostationary satellite might be required to point an antenna accurately at a target on the surface of the Earth, while simultaneously engaging in routine north-south station-keeping manoeuvres to prevent drifts in the orbit's inclination [8].

When dealing with a rendezvous manoeuvre, however, relative navigation aspects must be considered. In this case, the scenario is commonly defined in terms of an active spacecraft, termed the chaser, which attempts to approach another called the target. Inertial navigation is still needed, as the two bodies typically start out in separate orbits at large distances; it is at a distance to the target between 1 km and 100 m that the professed proximity operations stage

commences and the switch must be made to the relative navigation on-board sensors [9].

Proximity operations usually rely on precise, active sensors, such as Light Detection and Ranging (LIDAR). Although mature in the context of orbital operations, this technology is not free from drawbacks: the operational range is limited, allowing only for for close range rendezvous, and they are characterised by a relatively high mass, power consumption, and cost. On the contrary, passive sensors such as cameras are witnessing an increase in popularity due to their lighter frame and cheaper cost; as more efficient IP algorithms are developed, camera-based navigation solutions for space are a promising technology [10].

Camera-based navigation strategies draw inspiration from two parallel, but closely related, fields [11]: Structure-From-Motion (SFM) in computer vision, involving the joint estimation of camera motions and 3D structure, and Simultaneous Localisation and Mapping (SLAM) in robotics, consisting in navigating a robot through an unknown environment while building a map of it. The union of these two techniques is often termed Visual Simultaneous Localisation and Mapping (VSLAM) [12]. Although a stereo setup yields scene depth information directly by triangulating features detected in each camera, it can also be estimated using a monocular configuration fused with different sensors or the assumption of pre-known information. VSLAM was shown to be applicable to the relative navigation with a tumbling satellite by modelling the uncertainty as originating from the moving "map" [13]. It is also important to mention an alternative technique for use when one is not interested in reconstructing the scene per se, but rather the robot's own ego-motion. This approach, Visual Odometry (VO), has been adopted for the navigation of Mars exploration rovers [14]. Lastly, when the target is assumed to be known, an offline training phase can be included in which its appearance is learned and condensed into a database to be matched on-the-go to the features detected during the actual mission in order to solve for the relative pose. This is challenging as the discretisation of a 3D object into a 2D representation warrants a feature matching process that is robust to large rotation, scale, and illumination baselines [15].

Such a model-based approach is more arduous in the LWIR domain due to the extensive number of variables that influence the target's thermal signature, and hence its database representation. In this way, studies in the literature have focused on the use of features that are invariant to an object's thermal profile, such as its contour edges [16].

With the recent increase of hardware acceleration capabilities in consumer-grade computers and consequent expansion of deep learning to computer vision problems, Convolutional Neural Networks (CNNs) have demonstrated potential in local feature detection and description applied to relative navigation in space [17, 18]. However, these algorithms are often characterised by a supervised selection of landmarks in the pre-training stage which is specific to the target, and therefore are outside of the scope of this paper.

2.2. Evaluating Image Processing Algorithms

Detection and matching of features is the fulcrum of navigation solutions based on computer vision. A "feature" is purely a distinguishable part of an image. In particular, Szeliski describes interest points as one of the fundamental and most popular types of features [11]. These point features, or keypoints, are the subject of the present study; however, different types of features exist and are suitable for IP-based navigation, such as edges, circles, or even colours. Nonetheless, for the context of this work, one shall assume features unequivocally refer to interest points.

The evaluation of feature detectors goes back to before the turn of the century, when interest points were reduced to any point in an image for which the signal changed two-dimensionally, encompassing the traditional "L-corners", "T-junctions", and "Y-junctions"; a small image patch (the template) around the detected corner would then be extracted and matched for in the target image using correlation [19]. By then, however, there was not yet a clear consensus on how a proper evaluation framework should be set up. In fact, some authors resorted to subjective visual inspection methods to evaluate the quality of detection (e.g. [20]).

A few years later, with the advent of algorithms capable of detecting invariant features, such as SIFT [21], criteria such as repeatability and matching scores became commonplace in evaluative frameworks. These concepts and others are described in Section 3. These algorithms would automatically extract a support region around the feature and encode it into a numerical descriptor, allowing it to be matched without searching the whole image. Arguably, the most well-known examples in the computer vision literature are the studies by Mikolajczyk and Schmid on detectors [22] and descriptors [23]. This change in paradigm potentiated new developments in VSLAM; hence, the contemporary studies included benchmarks regarding transformations that one would expect to experience in that context, such as scale, rotation, illumination, among others [24].

With the onset of binary descriptors, the focus of study began to include the computational advantage these and others presented in the face of the more traditional, already established, algorithms. One such notable study is the one by Miksik and Mikolajczyk [25], which highlights the speed of FAST [26] for detection, and of BRISK [27] and ORB [28] for detection and description, in the face of the classical DoG/SIFT and Fast-Hessian/SURF [29]. However, like their preceding studies, the authors evaluate the methods on fixed, sparse image sequences, each one benchmarking a different transform, such as the Oxford dataset[5], rather than application-specific data. There have been some publications focused on the latter, such as visual tracking for UAVs [30] and grid map matching [31]; regarding space NCRV, the only reference in the literature found by the authors was the study by Takeishi et al. [32] on the benchmarking of the aforementioned IP algorithms for automatic landmark tracking on the Itokawa asteroid in the context of the Hayabusa mission. In it, the authors analyse their performance on a tumbling target navigation dataset in the visible wavelength, where they found that the algorithms suffer from low recalls in terms of corresponding interest regions when the angle of the asteroid shifts

[5] `http://www.robots.ox.ac.uk/~vgg/research/affine/`.

more than 20 deg and that the matching precision scores decline sharply after 10 deg.

Studies in the LWIR are certainly fewer in number, but they have been the object of recent study. Ricaurte et al. [33] evaluate the behaviour of classic descriptors in a cross-modality outdoor dataset, finding that many of the algorithms are actually more robust to changes in rotation and scaling in the LWIR than in the visible. Johansson et al. [34], and more recently Mouats et al. [35], highlight the importance of experimenting with different combinations of detectors and descriptors in the LWIR, as these often outperformed the native setups.

3. Image Processing Benchmarking Framework

Each ADR application, or more generally rendezvous mission, using imaging systems must consider performance figures to assess the viability of the image processing algorithms used. Feature detectors search an image for locations that are probable to match well in other images, and feature descriptors convert each region around the detected keypoint locations into a condensed vector that can be matched against other descriptors [11]. This section analyses these figures of merit for the selected detectors and operating on the visible and thermal infrared spectra in the devised scenario, and provides a theoretical background for these algorithms.

3.1. 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. Conversely, the second group considers blob detectors, which extract points taking into account a supporting 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 [36]. However, the LoG involves the computation of

second-order derivatives which are both sensitive to noise and computationally expensive.

Harris Corner Detector. Harris and Stephens [37] assembled their historically influential computer vision algorithm from the mathematical formalisation of Moravec's work [38] through the minimisation of the auto-correlation function that compares an image patch against itself shifted for small increments:

$$E(\mathbf{u}) = \sum_i w(\mathbf{x}_i) \left[I(\mathbf{x_i} + \mathbf{u}) - I(\mathbf{x}_i) \right]^2, \tag{1}$$

where I is the intensity of the image, $\mathbf{x} = (x, y)$ is the pixel position vector in I, $\mathbf{u} = (u, v)$ is the displacement vector, and $w(\mathbf{x})$ is a weighting function. For small variations in position $\mathbf{u} = \Delta\mathbf{u}$, it is shown that Eq. (1) can be written using a Taylor series approximation as

$$E(\Delta\mathbf{u}) \approx \Delta\mathbf{u}^\top \mathtt{A}\, \Delta\mathbf{u}, \tag{2}$$

where $\mathtt{A}$ is the auto-correlation matrix:

$$\mathtt{A} = w(\mathbf{x}) * \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}, \tag{3}$$

with $(*)$ representing the convolution operator and $I_x \equiv \partial I/\partial x$, $I_y \equiv \partial I/\partial y$. The matrix $\mathtt{A}$ contains the information on how stable the auto-correlation function is at a given point. Consider the two eigenvalues of $\mathtt{A}$, (λ_1, λ_2). If both eigenvalues are small, that translates into an approximately constant intensity profile within a window. A small and a large eigenvalue are equivalent to a unidirectional texture pattern, i.e. the surface of $E(\Delta\mathbf{u})$ is flat along that direction. If the two eigenvalues are sufficiently large, it corresponds to a minimum in $E(\Delta\mathbf{u})$ and to a corner or other pattern that can be tracked reliably. Harris and Stephens propose a corner response function given by

$$R = \det(\mathtt{A}) - k\,\mathrm{tr}(\mathtt{A})^2 = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2)^2, \tag{4}$$

where k is an empirical constant. The region is then considered a corner based whether the size of the response R is greater than a given threshold.

Good Features To Track. Shi and Tomasi [39] attempt to further improve Harris' work by proposing a different measure to determine what are Good Features To Track (GFTT). 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:

$$R = \min(\lambda_1, \lambda_2). \tag{5}$$

Difference of Gaussians. Although invariant to rotation, corner detectors such as Harris and GFTT employ a fixed window size which makes interest point detection sensitive to scale changes. In his Scale Invariant Feature Transform (SIFT) algorithm [21], Lowe makes use of scale-space filtering to tackle this issue. A Difference of Gaussians (DoG) is used to approximate the LoG; it is obtained by computing the difference between two Gaussian blurs of the same image with different standard deviations separated by a constant factor, i.e. σ and $k\sigma$. 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 down-sampled by a factor of 2, marking the start of the following octave. Once all the DoG are found, the resulting structure is searched for extrema in space ($\mathbf{x}$) and scale (σ): each sample point is compared to its eight neighbours in the current image and nine neighbours in the scale (Fig. 2). It is selected as a potential feature if it is either larger or smaller than all of them.

As a further refinement, each potential feature is subjected to a rejection process based on a contrast threshold value. Additionally, in order to reject edges, a process similar to the Harris corner detector is employed by computing the 2×2 Hessian matrix $\mathcal{H}$ of the difference image D at the location and scale of the interest point

$$\mathcal{H}(\mathbf{x}, \sigma) = \begin{bmatrix} D_{xx}(\mathbf{x}, \sigma) & D_{xy}(\mathbf{x}, \sigma) \\ D_{yx}(\mathbf{x}, \sigma) & D_{yy}(\mathbf{x}, \sigma) \end{bmatrix} \tag{6}$$

and submitting its ratio of principal curvatures to an edge threshold. The quantities D_{xx}, etc., are the second-order derivatives of D, estimated by taking differences of neighbouring sample points.

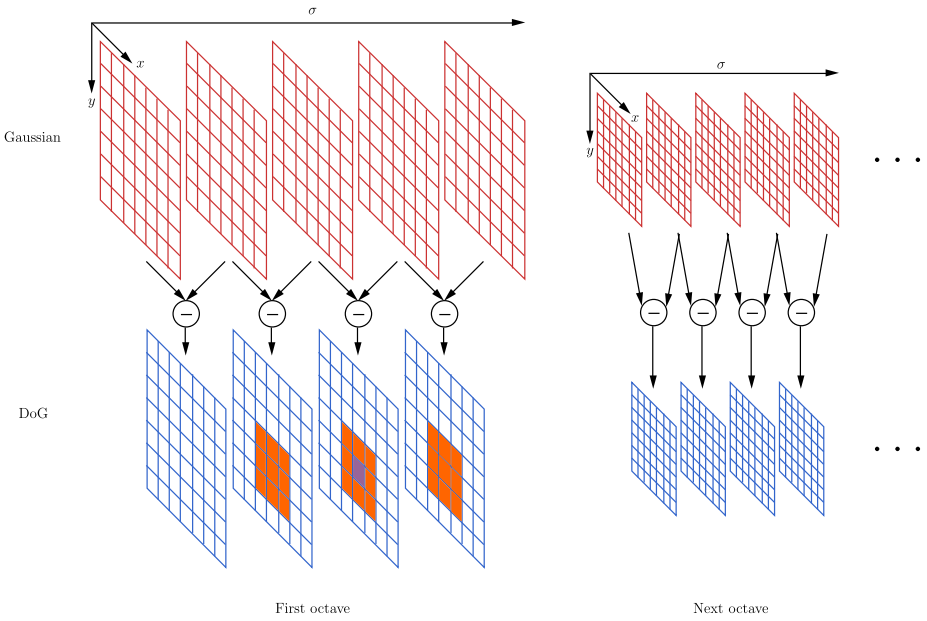


Figure 2: The difference of Gaussian pyramid structure (adapted from [21]). Adjacent Gaussian images are subtracted to produce the DoG images, each octave is characterised by downsampling the previous one by a factor of 2. Features are selected in the DoG images by comparing a candidate point (purple) to its neighbours (orange) in scale and space.

Fast-Hessian. The Fast-Hessian detector was introduced as part of the Speeded-Up Robust Features (SURF) algorithm [29], which aimed to provide a computationally faster version of SIFT. The Fast-Hessian detector makes use of a further approximation of the LoG by using box filters, which can be evaluated swiftly independently of size using integral images. The box filters are used to compute approximations to the derivatives D_{xx}, etc. For instance, 9×9 box filters are approximations for Gaussian second order derivatives with $\sigma = 1.2$. These approximations are consequently used to produce an estimation of the determinant of the hessian $\mathcal{H}$, which is used as a threshold for candidate features.

Features from Accelerated Segment Test. The Features from Accelerated Segment Test (FAST) algorithm [26] was developed with the purpose of creating a high-speed feature detector for real-time applications, such as SLAM. FAST first selects a pixel $\mathbf{x}_i$ in the image as an interest point candidate. A circle

of 16 pixels around $\mathbf{x}_i$ and a threshold t are defined. If there exists a set of n contiguous pixels in the circle which are all brighter than $I(\mathbf{x}_i) + t$ or all darker than $I(\mathbf{x}_i) - t$, then $\mathbf{x}_i$ is classified as a corner. The detection process is robustified through an offline machine learning stage, where a decision tree is built from alternative training images that is used in deciding which pixels should be assessed first on the test images in order to exclude a large number of non-corners, hence improving detection speed. The algorithm also makes use of non-maximal suppression to avoid detecting multiple features adjacent to one another. Bearing a greater resemblance to corner detectors rather than blob detectors, FAST is not natively scale- or rotation-invariant.

Centre Surround Extrema. For SIFT and SURF, responses are not computed at all pixels for larger scales. At each successive octave, the sub-sampling is increased, so the accuracy of features at larger scales is sacrificed. One solution to tackle this problem in scale-space filtering is to approximate the LoG using bi-level centre-surround filters, as proposed for the CenSurE algorithm [40]. This allows for the achievement of full spatial resolution at every scale.

Bi-level filters multiply the image intensity value by either -1 or 1. The circular bi-level filter is shown to be the most faithful to the LoG, but the hardest to compute. Other filter shapes can be computed briskly with integral images, with decreasing cost from octagon to hexagon to box filter. After computing the filter responses, candidate features are subjected to a non-maximal suppression over the scale space in a $3 \times 3 \times 3$ neighbourhood. Lastly, the Harris measure from Eq. (3) at the particular scale is used to filter out edge-like responses.

3.2. 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 second type of considered local feature descriptor differs from the previous

Table 1: Descriptors characteristics (adapted from [25]).

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

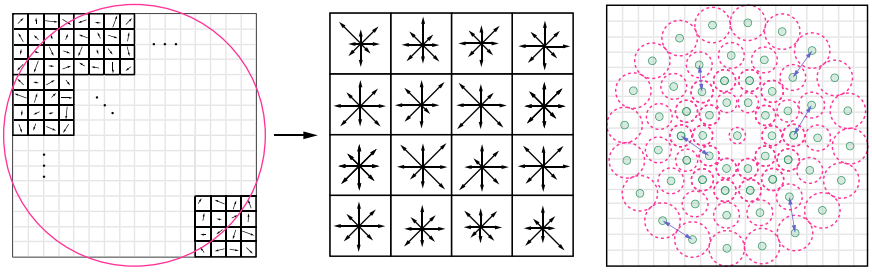
one in the sense that, instead of using a floating point vector representation, each descriptor consists of a binary string. For each feature point, a binary descriptor typically samples sets of pixel pairs $(\mathbf{x}_1, \mathbf{x}_2)_i, i \in n$ from the support patch, and performs a simple intensity comparison, where the result is 1 if $I(\mathbf{x}_1) < I(\mathbf{x}_2)$, and 0 otherwise, generating an n-dimensional bit string.

Using binary descriptors is advantageous as feature matching can be performed with resort to the Hamming distance[6], which provides better runtime performance with respect to the Euclidean distance test used with floating point descriptors: it consists only of applying the Exclusive-OR (XOR) logical operator followed by a bit count.

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.

Scale Invariant Feature Transform. For the SIFT algorithm [21], each keypoint is conferred an orientation by sampling the gradient magnitude and direction

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



(a) (b)

Figure 3: Distribution-based and binary description. **(a, left)** SIFT: the gradient magnitude and orientation at each subregion are weighted by a Gaussian window (pink) and **(a, right)** accumulated into a histogram (adapted from [21]). **(b)** BRISK: sampling locations (green) and Gaussian kernels to smooth intensity values (pink) for $n = 60$ points and some pairwise comparisons (purple) between them (adapted from [27]).

in a neighbourhood around it with a size dependant on the scale. The creation of the descriptor itself starts with the computation of the gradient magnitudes and orientations of a 16×16 sample array around the location of the detected interest point. The orientations are computed with respect to the keypoint's own orientation in order to achieve rotation invariance. To avoid abrupt changes in the descriptor, the computed quantities are weighted by a Gaussian window. Then, the samples of each 4×4 subregion are aggregated into an orientation histogram, each orientation weighted by the corresponding magnitude. The descriptor is finally formed from a vector that holds the magnitudes of all the orientation histogram entries. Each histogram has 8 bins, giving the descriptor vector a size of 128 elements (Fig. 3a) .

Speeded-Up Robust Features. For SURF [29], the orientation of each extracted region is assigned by computing instead the Haar wavelet responses [41] in a circular neighbourhood of radius equal to six times the scale, which are then weighted with a Gaussian window centred at the feature point. The first step in building the descriptor itself is defining a square region of size twenty times the scale centred around the interest point and oriented along the previously defined direction. This area is divided into smaller 4×4 subregions, and for each

of them the Haar wavelet responses are again computed, in the horizontal and vertical directions with respect to the orientation, and weighed with a Gaussian function. The sum of the wavelet responses and of their absolute values are stored in a four-dimensional descriptor vector for each subregion, making up for a total of 64 elements. The sign of the Laplacian distinguishes bright blobs on dark backgrounds from the reverse situation and is therefore conserved to allow for faster matching and an increase in performance.

Local Intensity Order Pattern. LIOP [42] is an algorithm for feature description designed to grant not only invariance to rotation and scale but also to complex illumination changes. As indicated by its name, it is based on order patterns, i.e. the order acquired by sorting the pixels of selected image patches by increasing intensity. It operates on the principle that this relative order remains unaltered in the case of monotonic intensity changes. First, the image is smoothed by a Gaussian filter as the relative order is sensitive to noise. Then, the size of each feature is normalised to a fixed diameter. The descriptor is constructed in an orientation-independent fashion, making it inherently invariant to rotation; therefore, the local patch is not rotated according to the local orientation as in SIFT. Afterwards, the overall intensity order 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.

Oriented FAST and Rotated BRIEF. Oriented FAST and Rotated BRIEF (ORB) is a method supporting both feature detection and description [28]. It applies a pyramidal representation of FAST for multi-scale feature detection combined with a Harris corner filter for edge rejection. An orientation is assigned to the feature through the intensity centroid method—the assumption that a corner's intensity is offset from its centre, where the direction of the vector from the interest point to this centroid yields the orientation. The feature description procedure is based upon the Binary Robust Independent Elementary Features

(BRIEF) mechanism [43], i.e. the pixel pairs are sampled from an isotropic Gaussian distribution. The original BRIEF algorithm is not rotation-invariant though, so ORB first steers the computed descriptor according to the feature orientation. However, this causes a loss of variance in each descriptor string, which is undesirable as high variance makes a feature more discriminative since it responds distinctively to inputs. In order to recover from the loss of performance of steered BRIEF, a greedy search algorithm is employed to look through all possible binary tests to find sets that both have high variance and are uncorrelated, resulting in a description processed coined "rBRIEF".

Binary Robust Invariant Scalable Keypoints. As ORB, Binary Robust Invariant Scalable Keypoints (BRISK) [27] also employs a scale-space modification of FAST for feature detection. Likewise, the description process yields a binary string and is based on pixel intensity comparison tests. The key concept of the descriptor is the sampling pattern used: n locations equally spaced on circles concentric with the interest point (Fig. 3b). 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. When sampling the image intensities for each pair, Gaussian smoothing is applied with a standard deviation proportional to their distance.

There are three main distinctions between BRISK and ORB. Firstly, BRISK's uniform sampling pattern prevents accidental distortion of brightness comparison between pairs after Gaussian smoothing. Secondly, in BRISK a single point takes part in more comparisons, limiting the complexity the intensity values look-up process. Lastly, the comparisons are restricted spatially such that the brightness variations are only required to be locally consistent.

Fast Retina Keypoint. FREAK [44] is a binary feature description algorithm which takes inspiration in the design of the human retina. The method adopts

the 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. These are segmented into four different areas; this is believed to result in a body resource optimization, where a higher resolution is captured in the fovea (inner-most circle), while lower acuity images are formed in the perifovea (outer-most circle). To match this biological model, the algorithm uses different kernel sizes for the Gaussian smoothing of every sample point in each receptive field, where these overlap for added redundancy leading to increased discriminative power. To determine which pairs of pixels to compare, the authors defend that a coarse-to-fine pair selection 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. This is interestingly consistent with modern understanding of the retina, where the perifoveal fields are first used to estimate the location of a point of interest and the validation is then performed with the densely distributed foveal receptive fields. Effectively, to describe a (even static) scene, the eye moves around with discontinuous individual movements called saccades. As such, FREAK emulates this process by parsing the computed descriptor in a way that the first 16 bytes represent coarse information, which is applied as a triage in the matching process. This way, a cascade of comparisons is performed, accelerating the procedure even further. For rotation-invariance, the orientation of the feature is estimated using local gradients similarly to BRISK.

3.3. 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 threshold (Fig. 4). Formally,

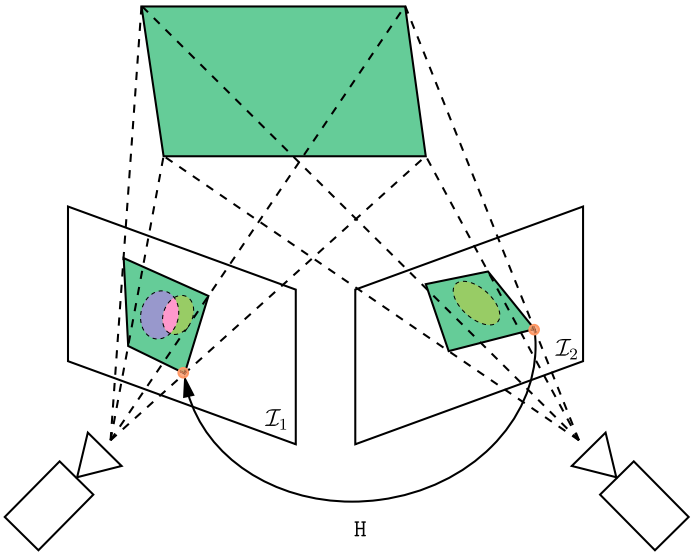


Figure 4: The homography ground truth $\mathtt{H}$ maps the light green feature from frame $\mathcal{I}_2$ to frame $\mathcal{I}_1$. The overlap area with the original purple feature from $\mathcal{I}_1$ is shown in pink. If the amount of overlap is above a certain defined threshold, then the two features correspond.

the following condition must hold:

$$1 - \frac{R_{\mathtt{M}_a} \cap R_{(\mathtt{H}^\top \mathtt{M}_b \mathtt{H})}}{R_{\mathtt{M}_a} \cup R_{(\mathtt{H}^\top \mathtt{M}_b \mathtt{H})}} < \varepsilon_0, \tag{7}$$

where $R_{\mathtt{M}}$ represents the elliptic region defined by $\mathbf{x}^\top \mathtt{M} \mathbf{x} = 1$, with $\mathtt{M}$ being the 2×2 symmetric matrix of ellipse coefficients, and ε_0 is the overlap error threshold. This mapping, the ground truth, can be given by a 3×3 homography matrix $\mathtt{H}$, 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:

$$\text{repeatability} \coloneqq \frac{C^+}{C}. \tag{8}$$

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 `H`. If a matched pair is also a correspondence, then it is deemed a correct match M^+, contributing to the matching score as

$$\text{matching score} \coloneqq \frac{C^+ \cap M^*}{C} = \frac{M^+}{C}. \tag{9}$$

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:

$$\text{recall} \coloneqq \frac{M^+}{C^+}. \tag{10}$$

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

$$\text{precision} \coloneqq \frac{M^+}{M^*}. \tag{11}$$

This performance metric is occasionally represented as its complement, i.e. $1 - \text{precision}$, 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. Ref. [23] defines three different ones. The first one is termed threshold-based matching, where two regions are matched if the distance between their descriptors is below a certain threshold μ. The second one is the Nearest Neighbour (NN)

based matching: regions a and b are matched if the descriptor $\mathbf{d}_b$ is the nearest neighbour to $\mathbf{d}_a$ and

$$\mathrm{NN} = \|\mathbf{d}_b - \mathbf{d}_a\| < \mu. \tag{12}$$

For the scope of this study, the third and last 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:

$$\mathrm{NNDR} = \frac{\|\mathbf{d}_b - \mathbf{d}_a\|}{\|\mathbf{d}_c - \mathbf{d}_a\|} < \mu, \tag{13}$$

where $\mathbf{d}_b, \mathbf{d}_c$ are the first and second nearest neighbours to $\mathbf{d}_a$, respectively. While for threshold-based matching a descriptor can have several matches—and several of them might be correct—for the NN and NNDR-based techniques, a descriptor only has one match. The former strategy can be attractive for real-time applications due to low computational effort. However, setting the threshold value μ proves to be a difficult task, as a fixed value may bias the results towards a given region of interest, whereas for the other strategies, μ is relative to each pair. Results from Refs. [23, 35] show that the NN strategy results in high precision, as all matches below μ are rejected, diminishing the number of false matches; using NNDR improves the precision even further.

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 [11]. 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.

4. 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.

4.1. Dataset

A dataset comprised of synthetically generated computer images was specifically designed for the proposed experiments. Two cameras were reproduced using the Astos Camera Simulator software: (i) one operating on the visible wavelength (0.39-0.70 µm), based on the mvBlueFOX-MLC 202b[7] camera with a Field of View (FOV) of 51 deg × 40 deg; (ii) and one operating on the LWIR wavelength (8-14 µm) based on the FLIR Tau2[8] camera with a scene temperature range from −40 °C to 160 °C, where the FOV was matched to the visible camera to ensure the scene is imaged similarly. Both cameras had their resolution scaled down to 320 px × 256 px and acquisition rate set to 1 Hz to run the image processing functions on a low performance hardware board. The generated images simulate a rendezvous approach with Envisat, capturing realistic variations in illumination, rotation, and scale. Two mission scenarios are considered: (i) a "Hot Case", where the spacecrafts are in a sunlit section of their orbit; (ii) and a "Cold Case", where they are in eclipse, under no direct illumination from the Sun. This yields a total of four different imaging sequences for the benchmarking of the IP algorithms, with 200 frames per sequence.

4.2. Ground Truth

To evaluate the proposed performance metrics (see Section 3.3), the ground truth relating the changing of the scene between frames must be established. Generally, calibrating a 3D scene, or the motion of a 3D object as in the present

[7] `https://www.matrix-vision.com/USB2.0-single-board-camera-mvbluefox-mlc.html`.
[8] `https://www.flir.co.uk/products/tau-2`.

case, would require the back-projection of the detected features into rays, computing their intersection with a 3D mesh of the target, and transferring the feature and its support region to a differently rotated and translated scene, similarly to the work of Ref. [32], resulting in a highly complex and possibly time-consuming framework.

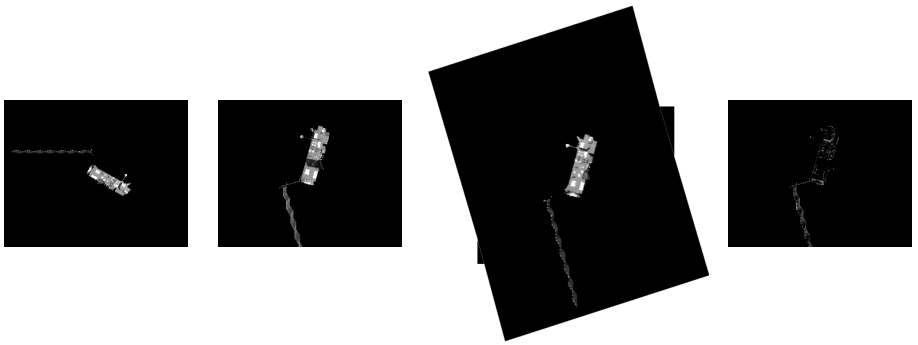
However, this can be greatly simplified by simulating a planar scene. This is achieved by modeling a rendezvous approach such that the same facet of the target is constantly visible. In this case, the ground truth can be computed just from the dataset itself, without resorting to the Computer Aided Design (CAD) model of the target, via a 3×3 homography matrix $\mathtt{H}$ [45]. 2D points $\mathbf{x}_i$ in one frame are related to those $\mathbf{x}'_i$ in another frame as:

$$\mathbf{x}'_i = \mathtt{H}\,\mathbf{x}_i, \tag{14}$$

where the points $\mathbf{x}_i = (x, y, 1)^\top$, $\mathbf{x}'_i = (x', y', 1)^\top$ are expressed in homogeneous coordinates.

The homography matrices can be computed directly from feature point correspondences between each frame (see, for example, Ref. [45]); however, a different approach is taken to avoid biasing the algorithms to be tested, similarly to Ref. [35]. First, putative feature matches between the two frames are obtained using a detector and descriptor not included in the benchmarks. This work uses Accelerated KAZE (AKAZE)[9] [46] features for this purpose. Then, an initial homography $\hat{\mathtt{H}}$ is estimated from these matches using RANdom SAmple Consensus (RANSAC) [47] to reject outliers. Finally, $\hat{\mathtt{H}}$ is used to initialise a forward additive Enhanced Correlation Coefficient (ECC) algorithm [48] to compute a refined $\mathtt{H}$. Figure 5 illustrates the ground truth computation for a pair of frames in the dataset.

[9] Here, "kaze" is not an acronym, but the romanisation of the Japanese word "風", meaning "wind", an allusion to the algorithm's speed.



(a) (b) (c) (d)

Figure 5: Homography `H` computation example for a pair of wide baseline frames. **(a)** Source image. **(b)** Destination image. **(c)** Source image mapped to destination using `H`. **(d)** Difference image between destination and mapped source.

4.2.1. Planarity Assumption

The facet of the spacecraft's main body that is constantly observed by the chaser is approximately flat (Fig. 5) and hence well modelled by a plane π parallel to $\mathbf{t}_2 - \mathbf{t}_3$ in the target frame (see the frames of reference defined in Fig. 6). This represents the dominant plane based on which the planar homography in Eq. (14) is computed.

The solar panel is not contained in this plane, meaning that Eq. (14) would normally not model the ground truth adequately. However, for this particular motion a valid planar assumption is upheld as follows. Since the motion of the chaser is always parallel to π (Section 4.3), the only apparent transformation experienced by the solar panel not explained by `H` is due to perspective projection (i.e. the dimension along $\mathbf{t}_1$ appears lengthier the closer the target is to the camera). In the computation of the homography between consecutive frames, due to the reduced motion the changes in the solar panel caused by perspective projection are not observed, thus producing a stable `H`. When computing it for larger baselines, the number of frames is limited so as to maximise the length of the sequence for benchmarking while minimising the perspective projection deformations and hence keeping the stability of `H`. In this way, features detected

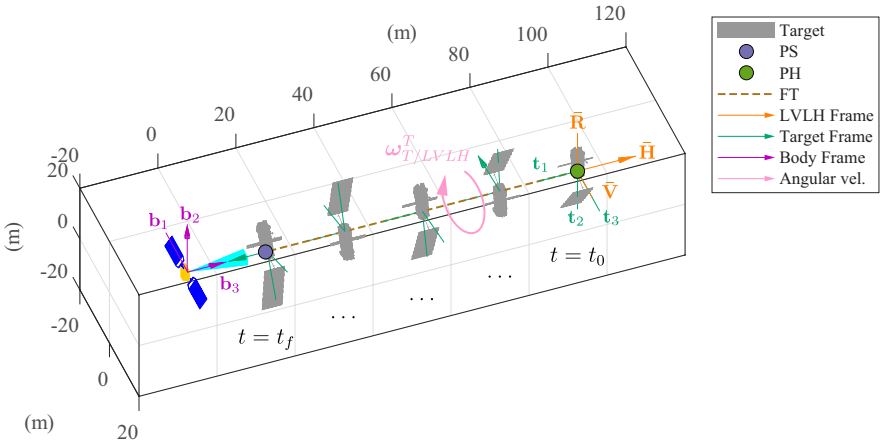


Figure 6: Scenario specifications for dataset generation, centred in the chaser's body frame $\mathcal{F}_{\mathcal{B}}$. The relative configurations of the target frame $\mathcal{F}_{\mathcal{T}}$ and of the LVLH frame are shown.

on both the main body and the solar panel can be accurately benchmarked without violating the planarity assumption and the validity of the ground truth.

4.3. Orbital Dynamics

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. 6. The spin axis of the target in the target frame, $\mathcal{F}_{\mathcal{T}}$, is aligned with the $+\mathbf{t}_1$-axis, and the spin axis in the LVLH frame is aligned with the $+\mathbf{H}$-bar axis; the rotation rate is 3.5 deg /s. The chaser ($\mathcal{F}_{\mathcal{B}}-$frame) assumes a constant orientation with regards to the target's LVLH frame. The sequence begins with the chaser in a hold point ("PH") 100 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 m distance, after which the sequence ends.

The current orbit of Envisat is estimated using the Two-Line Element (TLE) data of 30th October 2017[10]; the corresponding orbital elements are shown in

[10]TLE data obtained from NORAD Two-Line Element Sets Current Data at `http://www.`

Table 2: Envisat set of orbital elements at $t = t_0$ for evaluation dataset generation.

Element	**Dimensions**	**Symbol**	**Value**
Eccentricity	-	e	7.6112×10^{-4}
Semimajor axis	km	a	7.1427×10^{3}
Inclination	deg	i	98.2156
Right ascension of the ascending node	deg	Ω	343.0760
Argument of periapsis	deg	ϖ	189.5264
True anomaly	deg	θ	3.0109

Table 2. This corresponds to a situation where the target spacecraft is in full sunlight. To obtain the Cold Case trajectories, the true anomaly θ is altered.

4.4. Implementation

4.4.1. Modelling

The orbital states of both chaser and target, the camera parameters, and a 3D CAD model of Envisat are used as inputs to the Astos Camera Simulator to generate the dataset. The original textured model was obtained from the free astronomy software Celestia[11], a program which allows for the real-time 3D visualisation of space.

Visible Model. For use in the present study, the 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 (MLI) to properly emulate the diffuse reflection of light, as well as adding reflective properties to the solar panel; the differences between both models can be seen in Fig. 7. Additionally, an image of a laboratory mock-up of Envisat is also included for a qualitative comparison.

`celestrak.com/NORAD/elements` (accessed October 2017).

[11] `http://celestia.space`.

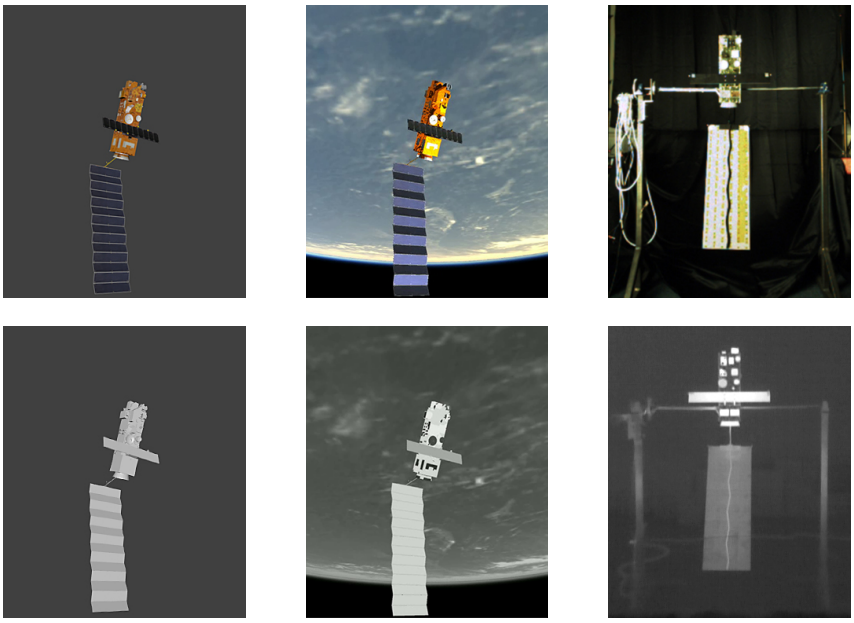
Thermal Model. The creation of thermal spacecraft model involved a different procedure. A thermal testing campaign was performed using a scaled-down replica of Envisat with surface coatings of similar types to those used in the real spacecraft, from which the temperature and emissivity of each component were obtained [7]. Two steady-state profiles were determined (Section 4.1). Then, the CAD model was stripped of texture and the collected data was incorporated as follows. First, the in-band radiance of each component was calculated by integrating the spectral radiance, given by Planck's law:

$$L_c(\lambda, T_c, \epsilon_c) = \epsilon_c \int_{\lambda-\delta}^{\lambda+\delta} \frac{2hc^2}{\lambda'^5} \frac{1}{e^{hc/(\lambda' k_B T_c)} - 1} \, \mathrm{d}\lambda', \tag{15}$$

where λ is the sampled wavelength; T_c, ϵ_c are the component's temperature and emissivity, respectively; k_B is the Boltzmann constant; h is the Planck constant; c is the speed of light in vacuum; and δ is a small neighbourhood around λ. The LWIR band was sampled at $\lambda = (8,\ 11 \text{ and } 14)\,\mu\text{m}$. The reader is directed to Ref. [49] for details on the closed-form computation of the integral in Eq. (15). Then, the computed radiances were normalised to $[0, 1]$ according to the modelled thermal camera's scene temperature range, yielding a 3-tuple analogous to the Red, Green, Blue (RGB) values in the visible. Due to this normalisation step, the obtained values become insensitive to the choice of δ. It was found that for the given band and range of temperatures, the solution was stable for any $\delta < 1\,\mu\text{m}$. Each 3-tuple was logged in a material file to accompany the mesh file as inputs to the camera simulator. Finally, the spectral response coefficients γ_{λ_1} of the camera for each of the three sampled wavelengths are also added as inputs; the software then rendered the single-channel thermal images with intensity equal to

$$I_c = \gamma_{\lambda_1} L_{c_{\lambda_1}} + \gamma_{\lambda_2} L_{c_{\lambda_2}} + \gamma_{\lambda_3} L_{c_{\lambda_3}}. \tag{16}$$

The generation of synthetic thermal data according to the aforementioned process entails some approximations, which are a reflection of the limitations of the software. Concretely, a fixed thermal signature for each sequence and



(a) (b) (c)

Figure 7: Multispectral modelling of Envisat. **(top)** Visible. **(bottom)** LWIR. **(a)** Original CAD mesh. **(b)** Modified mesh and materials running on the Astos engine. **(c)** Laboratory mock-up.

the assignment of a solid colour to each component, instead of gradients, are assumed. Nonetheless, this does not affect the validity of the presented results: the first approximation is justified by the fact that the dataset considers short duration sequences in thermal steady-state; the second approximation is upheld based on the distances between chaser and target and the low camera resolution. Figure 7 also features a synthetic image rendered by Astos on the LWIR alongside a real thermal image of the mock-up as captured by a FLIR Tau2 in laboratory. Despite the different thermal signatures, it can be seen that the representation of the target in both images is comparable.

4.5. Software and Hardware

The performance analysis framework was coded in the C++ programming language. The OpenCV[12] library, version 3, was used for computer vision and image processing related functions.

The implementations of every detector and descriptor used are publicly available from OpenCV, except for LIOP, where the author's original open-source code[13] was used. The implementation of DoG+SIFT is based on the code of Rob Hess[14]. Fast-Hessian+SURF, Harris, and GFTT are direct adaptations of the original papers [29, 37, 39]. FAST, FREAK, BRISK, and ORB are ports of the authors' own implementations. Lastly, for CenSurE, the OpenCV implementation is termed STAR and it is an altered version of the original algorithm [40] for added computational stability and speed.

Issue 1.4 (August 2014) of the Astos Camera Simulator was used to generate the dataset. The 3D visible and thermal models of the targets are input as Wavefront `.obj` and `.mtl` files, along with a text file containing the 6D pose of the chaser and target at each time-step, specified either in the inertial or relative frames. The simulator is also capable of automatically propagating the objects' trajectories in space given the initial orbital elements; however, these have been generated manually in the present study for better control. A separate configuration file is also supplied, specifying the Julian date for the start of the simulation, the frames of references used, the camera parameters, and the graphical settings (reflections, light glare, shadows, etc.). The frames are then rendered as imaged by the synthetic cameras with the placement of the Earth, Sun, and Moon defined from their true ephemeris and the input date.

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

[12] `http://opencv.org`.

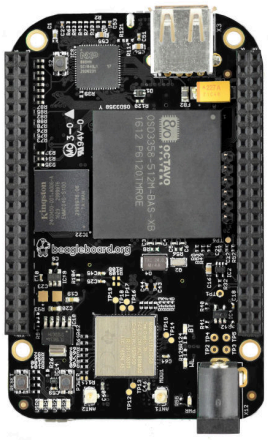
[13] `https://github.com/foelin/IntensityOrderFeature`.

[14] `http://robwhess.github.io/opensift/`.

Table 3: The BeagleBone(R) Black wireless single board computer. **(a)** Hardware properties. **(b)** Image of the board.

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

(a)



(b)

5. 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 pipeline is based on the works of Refs. [22, 23, 35] with some modifications given the nature of the dataset.

The first one refers to the considered dataset itself. Unlike common studies which consider scenes where the detected features are distributed over the whole image, for the images in the present dataset the background is featureless and the target may occupy a relatively small area. This imposes a limitation on the number of features that can be extracted from each frame. Since it is desirable to have a number of detections that is constant across frames and sequences for a balanced basis of comparison, this implies adequately tuning the sensitivity thresholds of each algorithm instead of relying on the default values. Samples of generated images and of detected features are illustrated in Figs. 8 and 9; the number of plotted features is limited to 40 for clarity.

The second modification, also related to the dataset, considers the particular

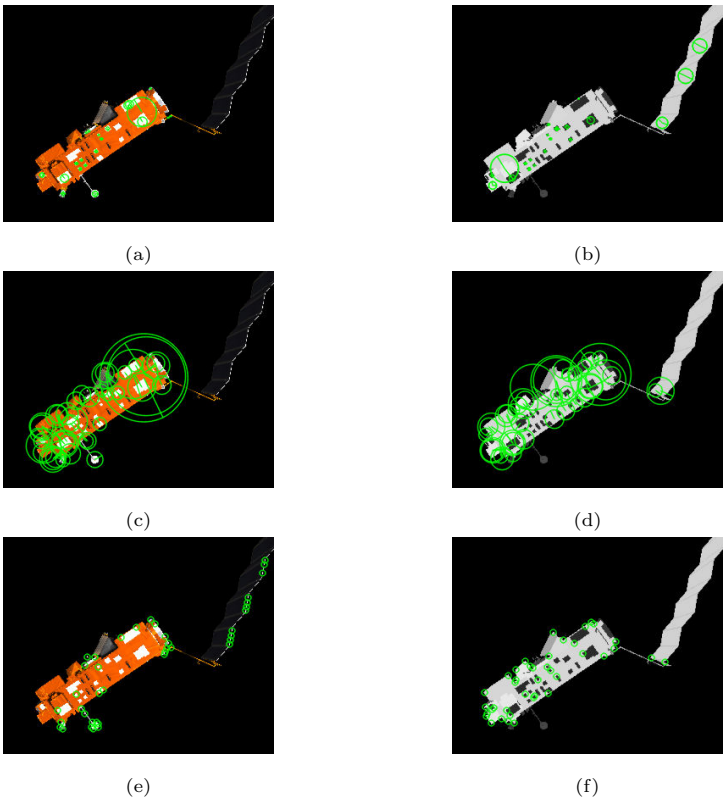


Figure 8: Examples of detected features (green) on hot case frames from the dataset. **(left)** Visible wavelength. **(right)** LWIR. **(a, b)** DoG. **(c, d)** Fast-Hessian. **(e, f)** Harris.

case of the eclipse in the visible band. For this sequence, the target is barely visible, as the only source of illumination is light reflected by the Earth's atmosphere. This is illustrated in Fig. 10 (left), where it can be seen on the histogram of image pixel intensities that the values are concentrated to the left of the spectrum, next to the largest bar representing the background. This has a limiting effect on the number of features that can be detected, which is a problem since it is intended to compare the IP algorithms under similar conditions. To enhance the visualisation of the target in these conditions, adaptive histogram equalisation is employed: the image is automatically divided into different sections (the default in OpenCV is a tile size of 8×8) and a histogram is computed for each one. The pixel intensities in each histogram are then equalised, improving

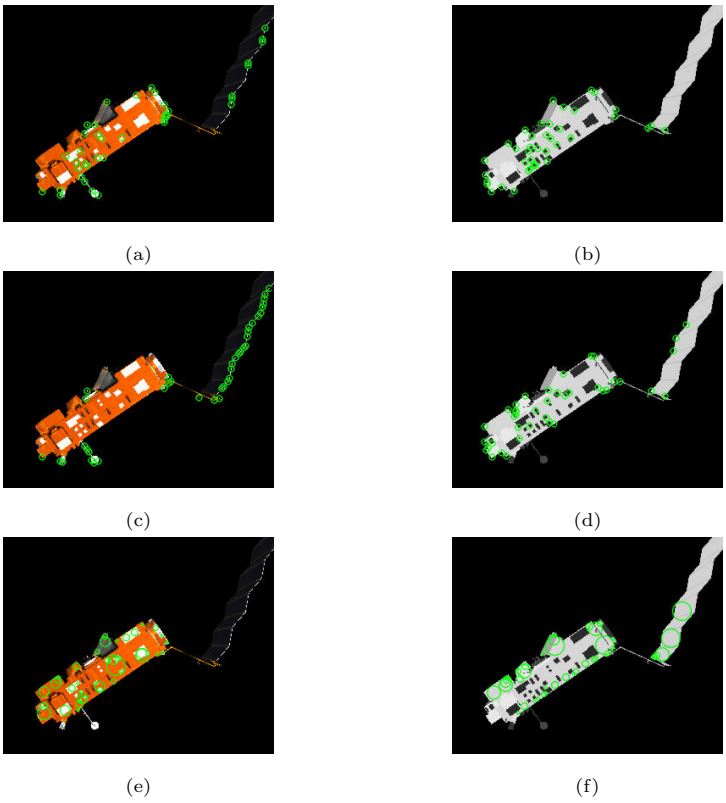


Figure 9: Examples of detected features (green) on hot case frames from the dataset. **(left)** Visible wavelength. **(right)** LWIR. **(a, b)** GFTT. **(c, d)** FAST. **(e, f)** CenSurE.

the contrast and the edges (and hence, corners). To prevent overshooting that could amplify noise, the output contrast is limited, in what is called Contrast Limited Adaptive Histogram Equalisation (CLAHE). After equalisation, bilinear interpolation is used to cull artefacts on tile borders. The result is displayed in Fig. 10 (right).

The last aspect concerns the implementation of the algorithms. Apart from differing internal mechanics, the computational code of each algorithm has been developed by different authors. As such, the parameters used to tune each one are not uniform. Consider, for example, the non-maximum suppression functionality: the process of removing multiple interest points that were detected in adjacent locations, leaving only the most distinctive ones. For Harris, GFTT

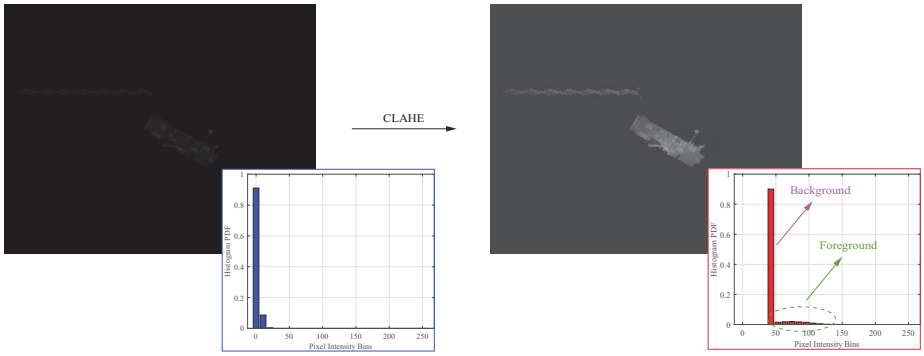


Figure 10: Effect of CLAHE on the visible cold case. **(left)** The original image, where the foreground is virtually indistinguishable from the background. **(right)** The image after application of CLAHE, where the intensity of the foreground pixels becomes more spread out over the range of possible values, resulting in the enhancement of the target.

and CenSurE, it is possible to set the suppression window size as an input parameter; for FAST it is only possible to toggle the functionality on or off; whereas for DoG and Fast-Hessian it is not controllable at all. In general terms, the smaller the suppression window, the more features are obtained, but the less distinctive they will be. These differences in interface make it difficult, to guarantee that each processed sequence will have the same ratio of feature number to feature distinctiveness. From the authors' observations, turning off non-maximum suppression on a feature detector, when given the option, leads to a sharp drop in performance when compared to the others. Therefore, it is ensured that non-maximum suppression is activated for a fair benchmark. Another aspect to consider is the implementation performance of each algorithm.

5.1. Tuning of Benchmark Parameters

Firstly, the different parameters that have a potential influence on the algorithms' benchmarking setup is analysed. A pair of common frames from each sequence, corresponding to an original image and a transformed one, is selected. The resulting data are averaged over all sequences for each feature detector and plotted in Figs. 11a–11c.

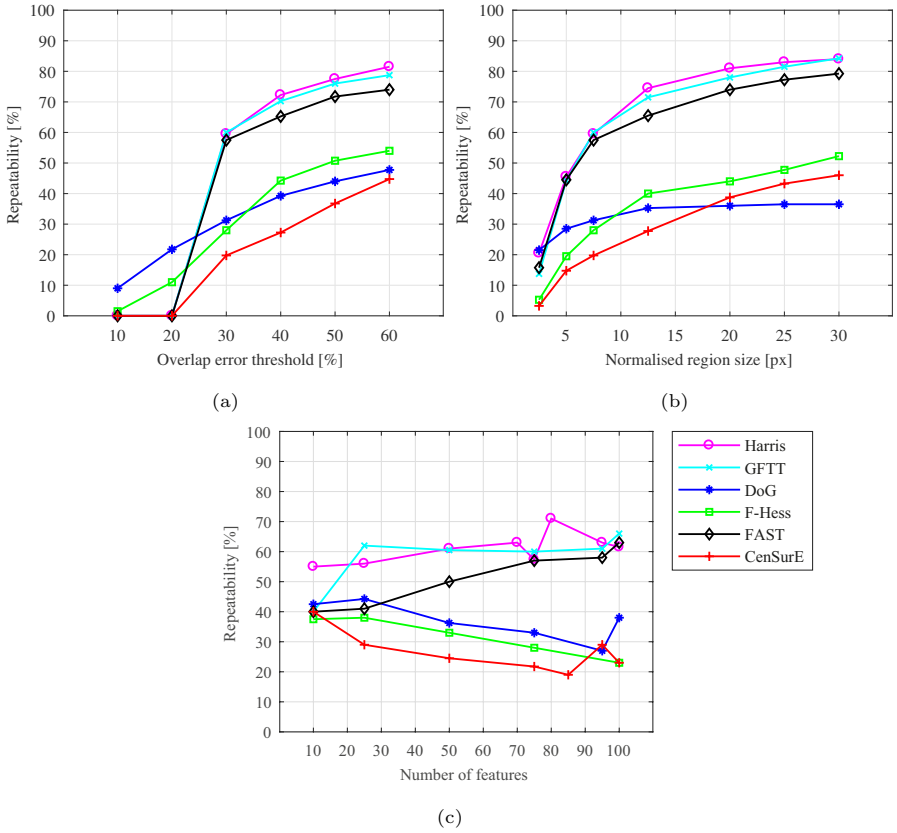


Figure 11: Repeatability scores as a function of different benchmark parameters. **(a)** Repeatability vs overlap error threshold. **(b)** Repeatability vs normalised region size. **(c)** Repeatability vs region density.

5.1.1. Accuracy of the Detectors

Fig. 11a illustrates the repeatability as a function of the overlap error threshold for the two considered bands. As the overlap error, i.e. the requirement to qualify two regions as corresponding, is relaxed the repeatability score goes up. For strict thresholds (10–20%), the three corner detectors demonstrate a null repeatability, but become the highest-ranking ones as the threshold is relaxed. This shows that these detectors are less accurate than the others for this type of scenario. CenSurE exhibits a similar behaviour but does not change its order relative to the others, scoring below the remaining two blob detectors. An

overlap error threshold of 30% is selected to ensure non-zero repeatability scores.

5.1.2. Normalised Region Size

Secondly, the effect of the choice on the normalised region size is studied; the results are displayed in Fig. 11b. This test was conducted with a fixed overlap error threshold of 30%. The relative ordering of the feature detectors remains the same, save for DoG: for the minimum considered radius, it ranks first in repeatability score, but as soon as this parameter is increased, it is surpassed by the corner detectors and begins to saturate beyond 12.5 px. Choosing a normalised region size of 7.5 px will limit the bias in further evaluations.

5.1.3. Region Density

For this test, the effect of increasing the number of features on the repeatability of the detectors is considered. This is achievable by altering the tuning parameters for each algorithm, allowing them to be compared when they output a similar amount of interest points. This is plotted in Fig. 11c, where the overlap error threshold and the normalised region size were set to 30% and 7.5 px, respectively. It can be seen that the corner detectors (i.e. Harris, GFTT, and FAST) tend to improve their repeatability scores when the number of features is increased, whereas the opposite is observed for the blob detectors (i.e. DoG, Fast-Hessian, and CenSurE). Note that the scores of DoG and CenSurE slightly increase towards the maximum considered number of detections, which indicates that these algorithms could possibly be less robust to noise: the quality threshold of the extracted features must be lowered to increase detections, which can lead to more false positives.

5.2. Benchmarking of Feature Detectors

For this test, the repeatability and number of correspondences obtained by each detector for each full sequence is analysed. In addition, the matching scores and number of matches is computed. This is done using the 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. 12–19. As in Ref. [35], two plots are provided for each sequence: (i) the first benchmarks consecutive image transforms, which is commonly done in SFM and VSLAM algorithms; in this case, a value pertaining to frame $\mathcal{I}_k$ in the plot is referent to the transformation between frames $\mathcal{I}_k$ and $\mathcal{I}_{k+1}$, while (ii) the second plot demonstrates the behaviour of the detectors when faced with large image transformations, which is usually the case encountered when applying model-based navigation strategies; the number "0" is used to represent the reference image (a frame from the middle of each sequence is chosen), whereas positive numbers represent transformations in posterior frames with respect to that reference and negative numbers represent those prior to it.

Visible Modality Hot Case. Figs. 12–13 showcase 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 correct matches when using GFTT and FAST actually decrease as the chaser nears the target, meaning that the high number of obtained correspondences likely stems from accidental overlap. 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. 12b that Fast-Hessian shows a better performance when coupled with LIOP than when combined with its

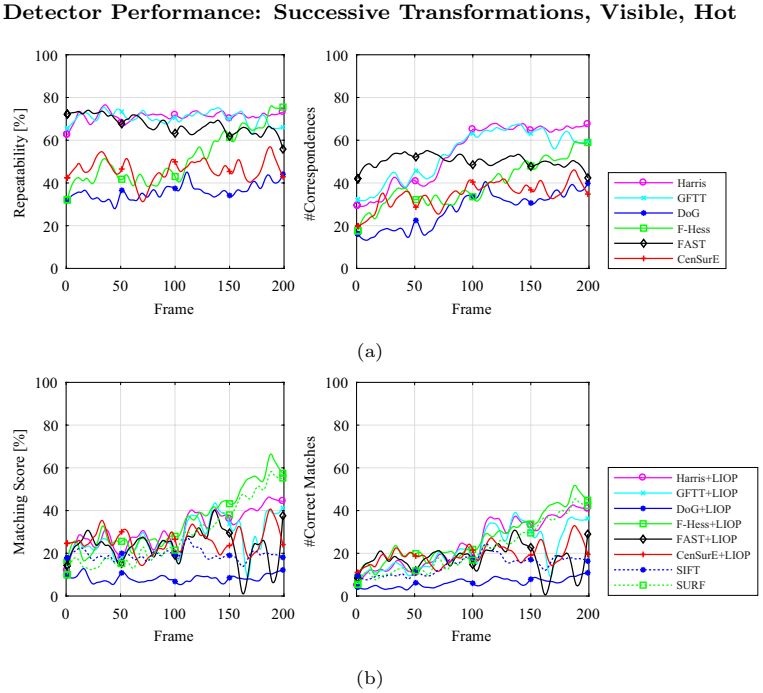


Figure 12: Performance for rendezvous sequence: successive transformations, visible band, hot case. **(a)** Repeatability and number of correspondences. **(b)** Matching score and number of correct matches. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

native descriptor. From Fig. 13 it can be seen that the detection algorithms are in general less resilient to large image transformations. In spite of a high repeatability for variations relatively close to the baseline, the number of correct matches of the three corner detectors drops rapidly; for sufficiently large transformations, they produce no correspondences at all. Interestingly, DoG when used with its native SIFT is shown to be the most robust in terms of matching score for large variations, when it performed the worst for small variations.

Visible Modality Cold Case. Figs. 14–15 represent the results obtained for the detectors in the visible eclipse case. Generally, the repeatability scores are quite similar to the hot case both in trend and magnitude. In opposition, the matching scores are now seen to decrease with time; the exception is Fast-

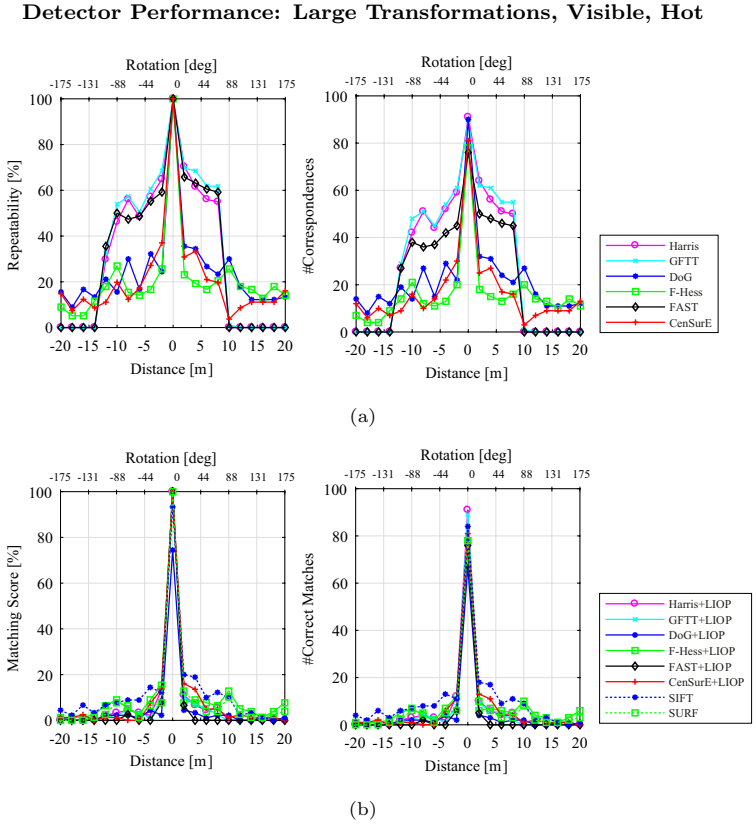


Figure 13: Performance for rendezvous sequence: large transformations, visible band, hot case. **(a)** Repeatability and number of correspondences. **(b)** Matching score and number of correct matches. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

Hessian combined with LIOP, which remains consistent, and the same detector use with SURF, which actually increases performance with time. The decreasing number of matches when the correspondences are increasing indicates that as the sequence progresses the features are becoming less distinctive to be correctly matched. In terms of large variations, the matching score decreases more sharply than in the hot case; this could be explained by a decreased consistency in the target pixels' intensity values due to CLAHE between the reference and query frames.

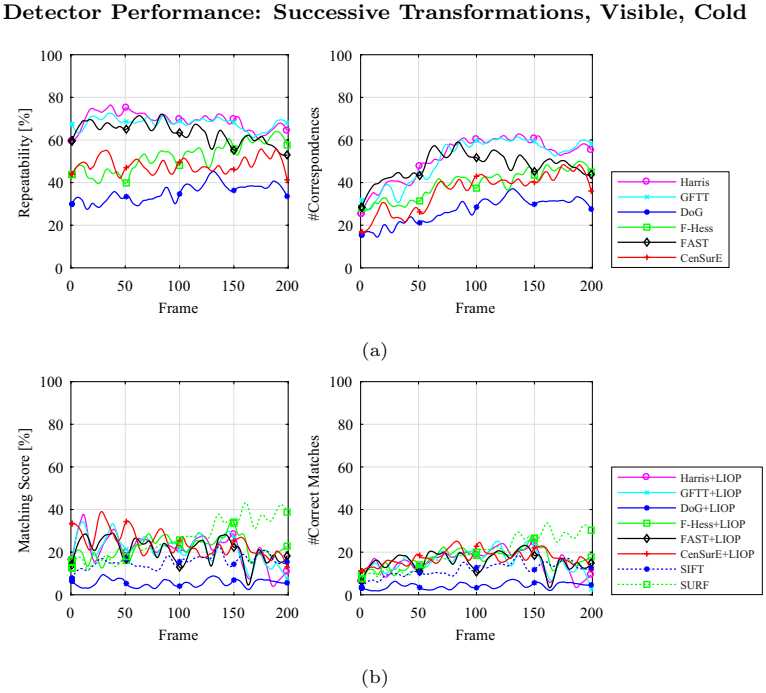


Figure 14: Performance for rendezvous sequence: successive transformations, visible band, cold case. **(a)** Repeatability and number of correspondences. **(b)** Matching score and number of correct matches. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

Thermal Infrared Modality Hot Case. Figs. 16–17 show 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. From Fig. 17 the behaviour of the detectors is less consistent: FAST and DoG outperform the other algorithms in medium transformations (up to ± 10 m and ± 90 deg baselines) with respect to matching score, whereas Fast-Hessian pro-

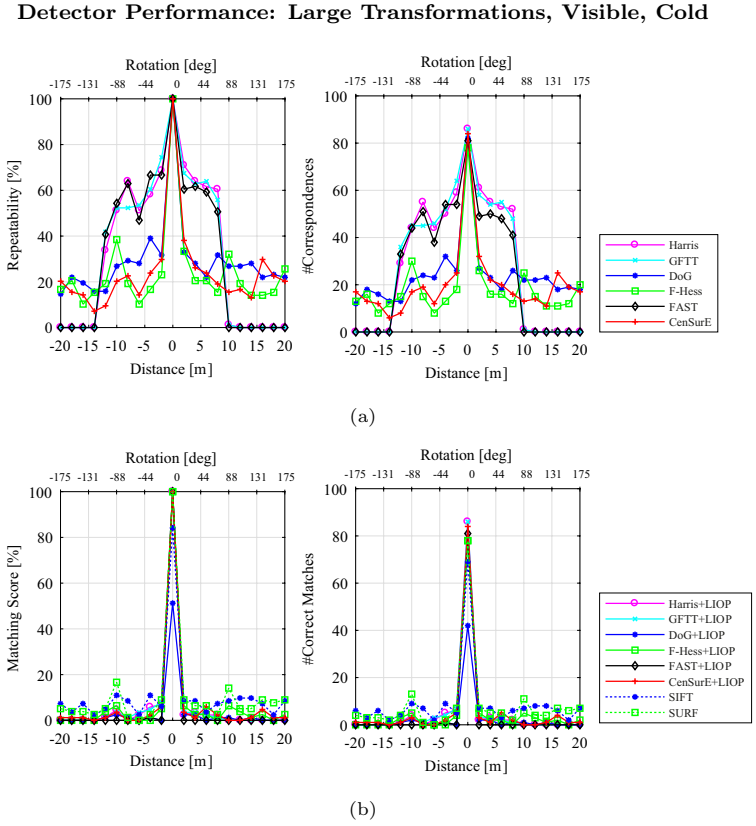


Figure 15: Performance for rendezvous sequence: large transformations, visible band, cold case. **(a)** Repeatability and number of correspondences. **(b)** Matching score and number of correct matches. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

vides the best performance for larger variations.

Thermal Infrared Modality Cold Case. Figs. 18–19 illustrate the detector performance for the thermal infrared during eclipse. For both consecutive and large transformations, the number of correct matches is generally lower than for the hot case in the same band; in spite of that, the repeatability scores are similar, which suggests that the cold case generates less distinctive features. This is more noticeable in the case of FAST, whereas Fast-Hessian and CenSurE are more impervious to the changes in temperature. It is however important to note

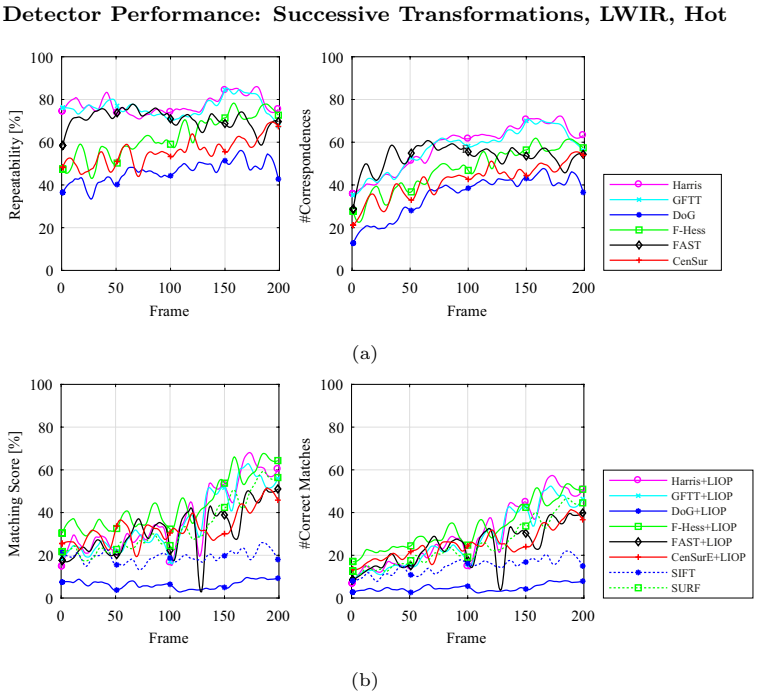


Figure 16: Performance for rendezvous sequence: successive transformations, thermal infrared band, hot case. **(a)** Repeatability and number of correspondences. **(b)** Matching score and number of correct matches. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

that Harris and GFTT recover greatly towards the end of the sequence in terms of correct matches for successive transformations, outperforming the remaining detectors (Fig. 18b).

5.2.1. Discussion

Despite being imaged in two different modalities, the simulated sequences feature a common relative motion. Therefore, some similarities in the results are expected. The repeatability trend for the successive transformations, in particular, is similar for all four sequences: corner detectors tend to be the most robust and for blob detectors the score tends to increase with the inverse of the distance to the target. For large transformations, the repeatability of

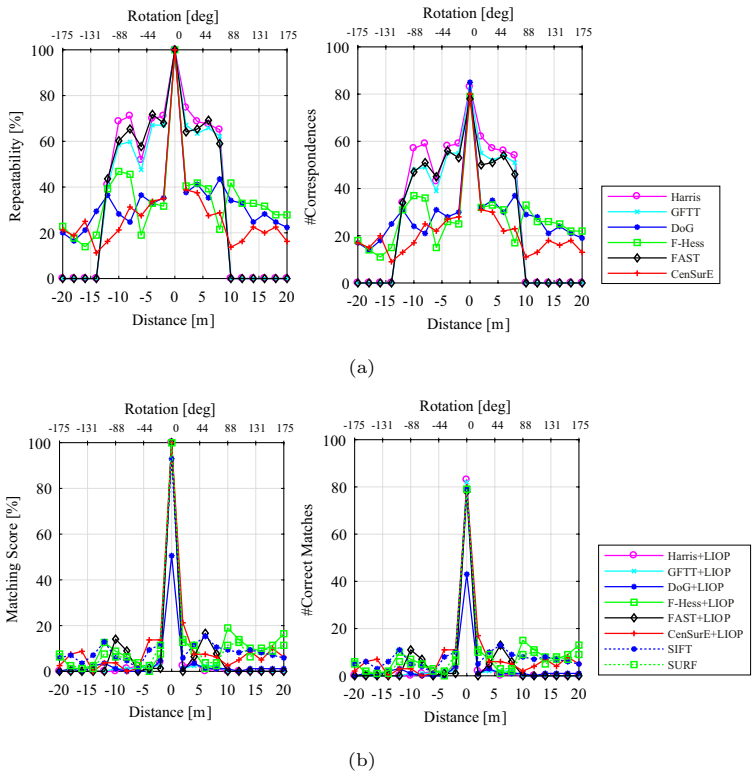


Figure 17: Performance for rendezvous sequence: large transformations, thermal infrared band, hot case. **(a)** Repeatability and number of correspondences. **(b)** matching score and number of correct matches. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor

corner detectors drops to zero after a certain point, whereas blob detectors are resilient. 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, wavelength, and illumination condition considered.

Despite their high repeatability, corner detectors are are often equalled or even surpassed by the blob detectors in terms of matching score. Despite high

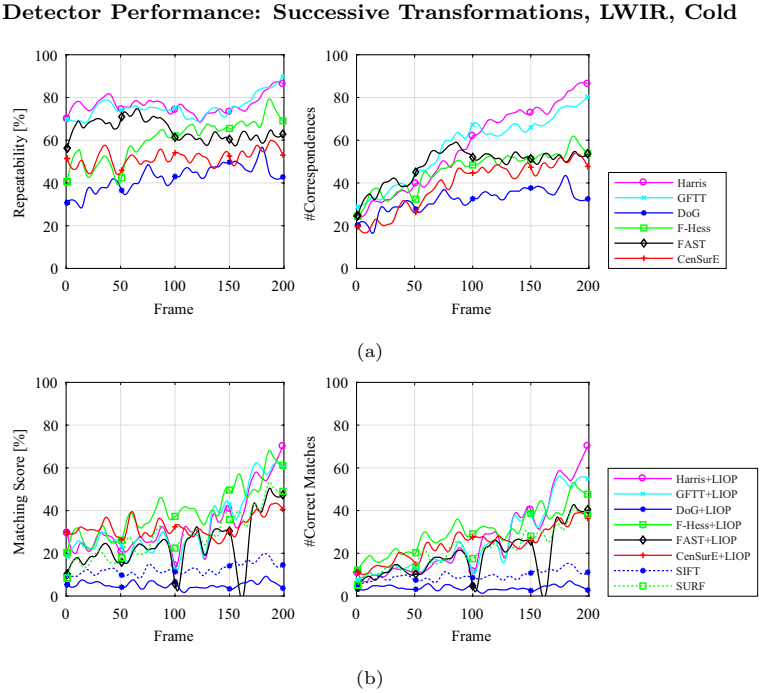


Figure 18: Performance for rendezvous sequence: successive transformations, thermal infrared band, cold case. **(a)** Repeatability and number of correspondences. **(b)** Matching score and number of correct matches. The raw data is presented smoothed with markers added for readability. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

repeatability, FAST is one of the least distinctive algorithms across all tests. Fast-Hessian performs well in terms of matching scores in most cases despite average repeatability; the exception is the visible cold case, where there is a generalised loss of performance, but it still maintains a good ranking in relative terms. This suggests an extraction of quite distinctive features, which confirms what was stated in the LWIR analysis of Ref. [35] 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 for successive transformations regardless of the wavelength and illumination, but seems to perform worse on the LWIR cold case. On the other hand, its performance is

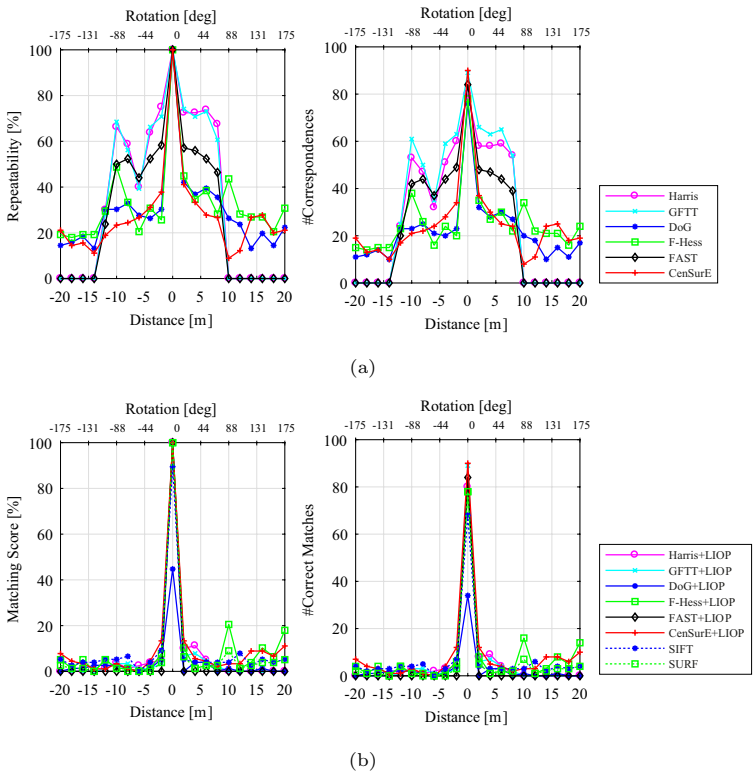


Figure 19: Performance for rendezvous sequence: large transformations, thermal infrared band, cold case. **(a)** Repeatability and number of correspondences. **(b)** Matching score and number of correct matches. The dashed lines show the results for DoG and Fast-Hessian with their original descriptor.

comparable to the other blob detectors when dealing with large transformations. It performs better with SIFT than with LIOP in every situation, whereas Fast-Hessian usually performs better with LIOP than SURF, the exception for the latter being the visible cold case. This reiterates the importance of testing detectors and descriptors separately to avoid any cause of bias.

In the benchmarking of successive transforms, 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 inherent to the MLI, for example, as their matching scores are better on the textureless LWIR. In the latter case, the actual

corners are better defined and impervious to illumination changes. On the same note, performance is generally better for the LWIR sequences: for the hot cases, 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. In the visible eclipse sequence, the efficiency of the algorithms is greatly diminished. This finding suggests that an artificial solution such as CLAHE to tackle the cold case is not a feasible solution. It does allow for the detection of more features, but these are not distinctive enough to guarantee an acceptable matching score. The use of a thermal infrared camera is a better approach in this case according to the results.

Regarding the benchmarking of large transformations, the results show a quasi-symmetrical pattern around the baseline. The matching scores are generally biased towards the right, meaning that larger scales (shorter distances between chaser and target) are favourable. This is a judicious hypothesis since, due to the low resolution of the dataset, bigger distances quickly translate into less details. On the other hand, there is a bias towards the left in repeatability, which is explained by the fact that smaller scales with a constant region sizee lead to more overlaps. The lack of scale invariance in the corner detectors is evident from the abrupt decline of the associated number of matches when varying the distance to the target. In general, the performance of the detectors is quite low for large baselines as opposed to successive transformations, which can make their use difficult in model-based pose estimation pipelines.

5.3. 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, i.e. 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. For each of the four sequences, and similarly to Ref. [35], two sets of results are shown: the first representing a descriptor benchmark for short (sucessive) and large 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. 20–23.

However, a different procedure is adopted regarding which frames from the dataset are used. Ref. [35] considers only the matching between features from two frames (one pair with a short baseline, another pair with a large one) for this test. This is because the authors benchmark image transform variations in an isolated way, i.e. one test for rotation variation, one for scale change, and so on. In the case of the present paper, the used datasets have in common a fixed trajectory where more than one transform is present. Since the aim is to assess the performance for the whole rendezvous manoeuvre, the ROC curves are computed using the average values for every pair of frames; in particular, for the large transformations set, the reference used is a frame located at the middle point of the sequence, i.e. when the target is 60 m away from the chaser, and the test includes variations in the range of $\pm 20\,\mathrm{m}/\pm 175\,\mathrm{deg}$ relative to the reference.

As mentioned in Section 3, a NNDR-based matching strategy is considered.

Visible Modality Hot Case. Fig. 20 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 can also be seen that the performance of the algorithms is degraded for large transformations comparatively to sequential ones.

It is interesting to note that SIFT performs better with Fast-Hessian features (Fig. 20c) than with DoG features (Fig. 20a) in the case of short transformations. However, the opposite is true for large transformations (Figs. 20b and

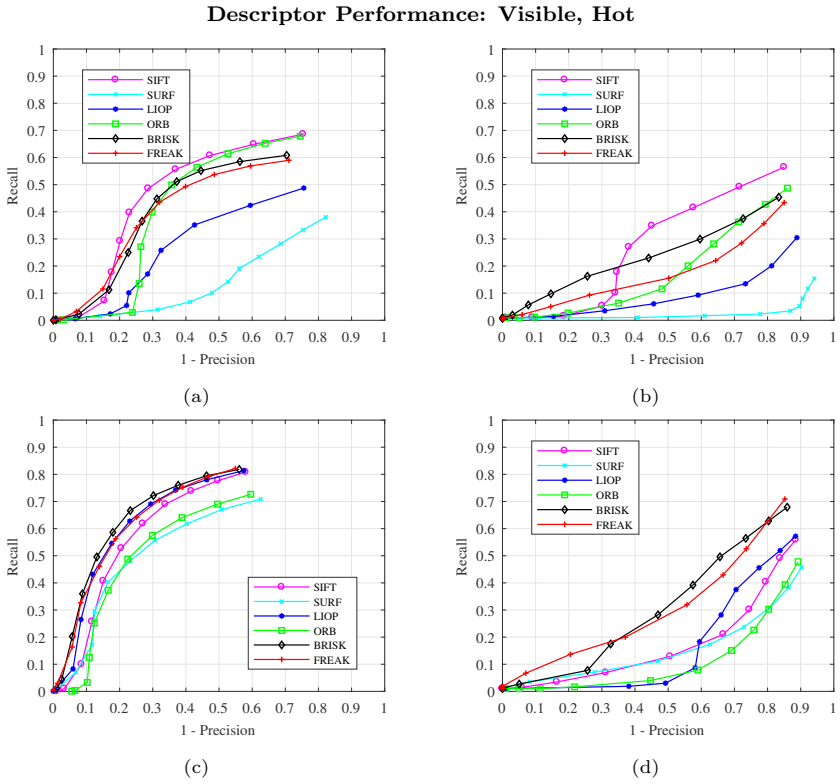


Figure 20: Descriptor ROC curves for the visible band, hot case. **(a, b)** DoG features with small and large transformations. **(c, d)** Fast-Hessian features with small and large transformations.

20d). Indeed, when DoG features are used, SIFT performs best, followed by ORB and BRISK. For small transformations, the performance of the three descriptors are comparable, whereas for large transformations, BRISK obtains the best results if $1 - \text{precision} < 0.35$ but SIFT dominates for values above that.

For Fast-Hessian features, BRISK, FREAK, and LIOP give the best results in the case of small variations; in the case of large variations the performance of the latter one degrades considerably, which seems to agree with the observations of Ref. [35] regarding the monotonic intensity changes of LIOP's rotation invariant sampling not holding for large angles. Overall the results obtained for SURF are sub-par, showing that combining a feature detector with a non-native

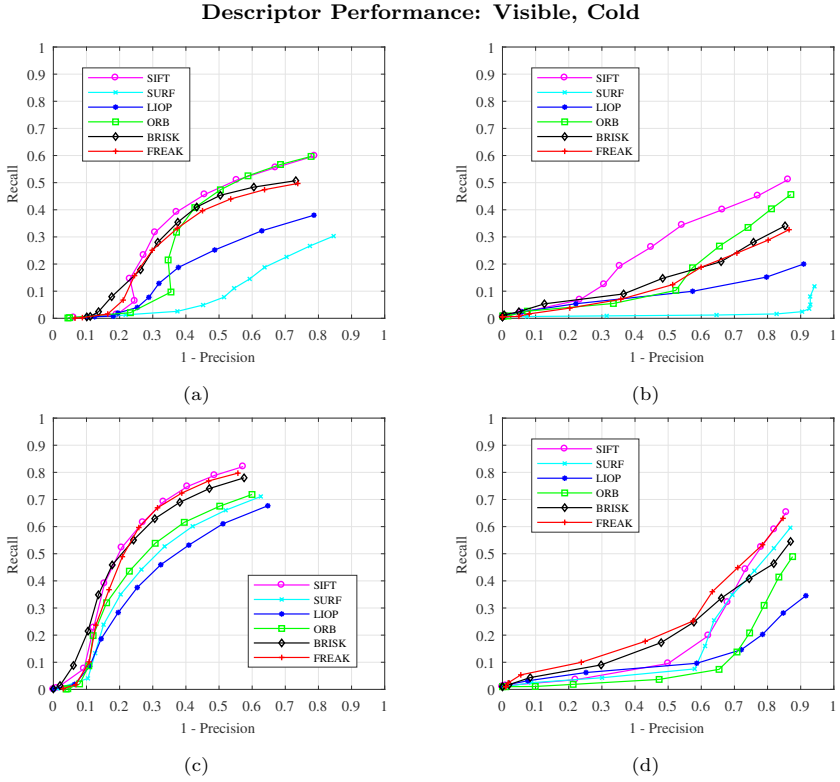


Figure 21: Descriptor ROC curves for the visible band, cold case. **(a, b)** DoG features with small and large transformations. **(c, d)** Fast-Hessian features with small and large transformations.

descriptor can yield better results.

Visible Modality Cold Case. Fig. 21 shows the descriptors' performance for the visible in eclipse. The algorithms are affected by the low illumination case more than the sunlit scenario for this spectrum. The precision can be shown to be relatively lower, particularly for larger variations, which means the descriptors incur more frequently in false matches. The relative ranking of the algorithms is similar to the previous case, save for small variations computed on Fast-Hessian features, where SIFT shows the best performance (close to FREAK and BRISK) and LIOP performs the worst. This is in agreement with the plot of Fig. 14, where there is a drop in the matching score of Fast-Hessian + LIOP, but it still

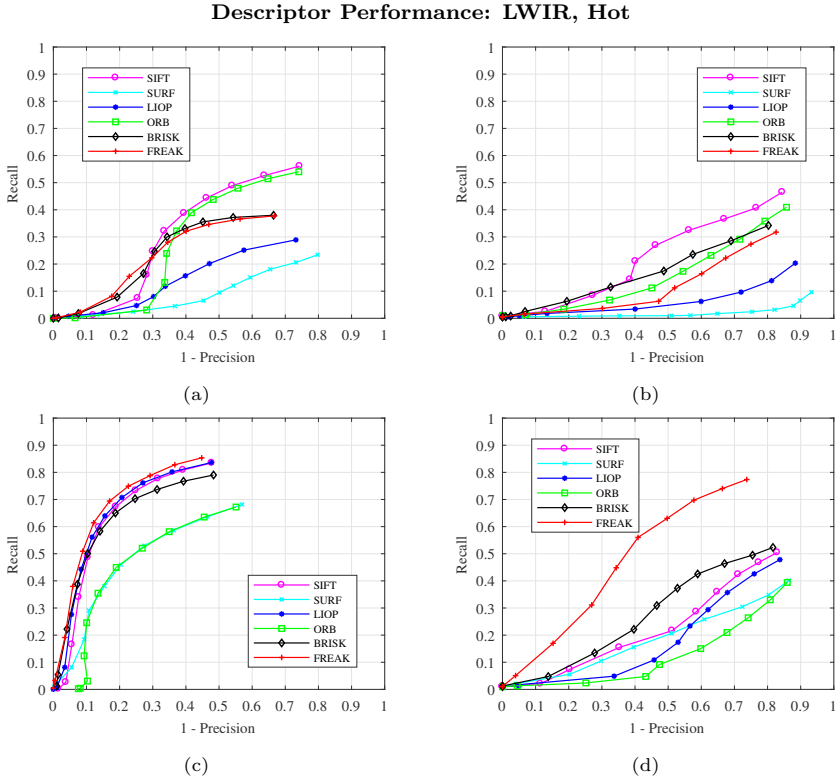


Figure 22: Descriptor ROC curves for the thermal infrared band, hot case. **(a, b)** DoG features with small and large transformations. **(c, d)** Fast-Hessian features with small and large transformations

remains higher than that of SIFT.

Thermal Infrared Modality Hot Case. 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. 22. The performance computed on DoG features follows the same trend as for the visible case, albeit with a yielded precision lower than the eclipse case.

On the other hand, when using Fast-Hessian features the descriptors perform better than both visible cases. For short transform variations, FREAK obtains the higher score, but as in the analogous visible case, it behaves quite similarly to BRISK, SIFT, and LIOP. With respect to larger transformations, FREAK

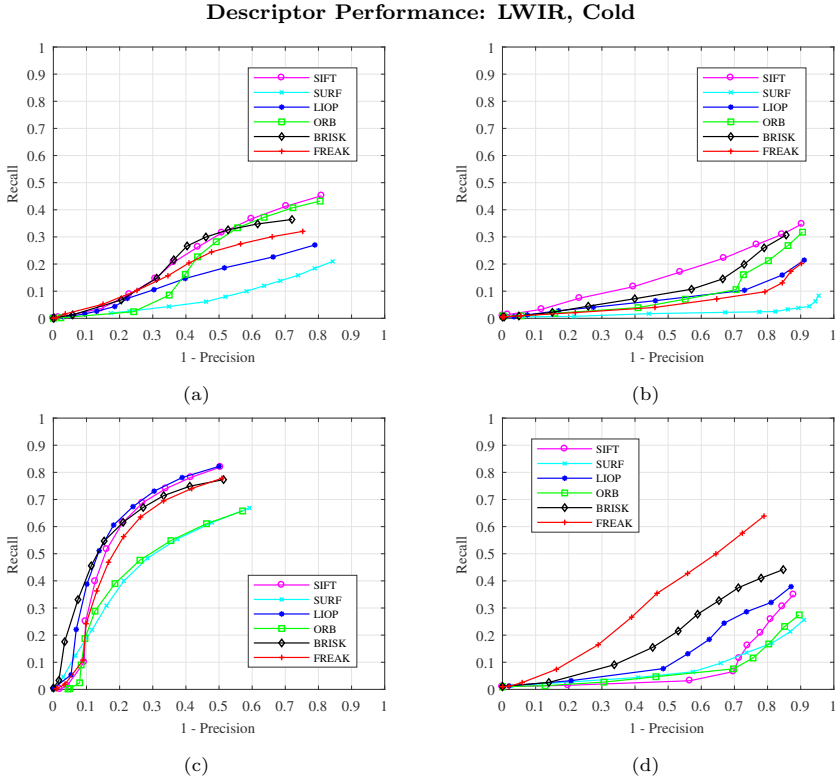


Figure 23: Descriptor ROC curves for the thermal infrared band, cold case. **(a, b)** DoG features with small and large transformations. **(c, d)** Fast-Hessian features with small and large transformations.

performs best by a large margin. The other algorithms are also less affected by these variations than in the visible case. This means that, for the same relative motion, the descriptors are more affected by the dynamic effects present in the visible modality—such as textural noise, glare, shadows—than by a textureless scene.

Thermal Infrared Modality Cold Case. Lastly, Fig. 23 illustrates the benchmarking of the descriptors in the eclipse case for the thermal infrared sequence. As in the visible case, the algorithms are more affected by these transformations than in the hot case.

When DoG features are used, the descriptors perform worse than in the

visible cold case. The precision attained by the algorithms is quite low, which is in line with the observations from Subsection 5.2 regarding the low number of correct matches for this detector in the thermal infrared modality.

Conversely, descriptors computed on Fast-Hessian features in this scenario are actually comparable to the performance attained for the visible hot case; for small transformations, LIOP achieves the best performance, however it is again degraded in the case of larger transform variations.

5.3.1. Discussion

The presented results suggest that the performance of the descriptors is dependent on the feature they are applied on, regardless of descriptor type. Fast-Hessian performs better in general both in terms of recall and precision scores, regardless of the modality, although the gap is narrower in the benchmarking of large transformations. As theorised by Ref. [35], 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. 8 that this is also the case for the analysed dataset.

Overall, SIFT as a whole obtained very good scores. However, its performance is degraded substantially in the case of large transformations (particularly on Fast-Hessian features).

LIOP was shown to perform better when computed on Fast-Hessian features, both on the visible, and as reported in Ref. [35] on the LWIR. It can be ranked amongst the best descriptors when used with this type of feature for successive transformations. The exception is the visible cold case, where it is ranked last. Furthermore, when considering large transforms, its performance declines, which is in line with the analysis made for the detectors in Section 5.2.

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

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

5.4. 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. All values are averaged between the four sequences for each algorithm.

Table 4 portrays the average extraction time per feature for each detector. This type of analysis is useful in shifting awareness towards the computation time, which can be limiting depending on the application, and is particularly important for those involving low performance computing. DoG scores the slowest detection time, at 21.2 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, CenSurE is faster than Fast-Hessian, which is in turn faster than DoG. Surprisingly, GFTT is recorded having a higher execution time than Harris.

Fig. 24 displays the average computation times of the detectors per frame. DoG is the clear outlier, being the only detector that does not fit in the computational budget of 1 Hz. In addition, the average execution time per frame for CLAHE in the case of the visible cold case sequence are also shown (in red).

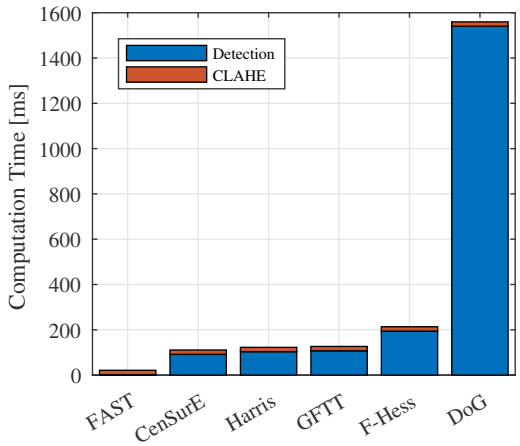


Figure 24: Comparison of average feature detection times per frame.

This function does not depend on the detector and the mean execution time was 19.22 ms, accounting for less than 2 % of the allocated budget. Note, however, that the average detection time per frame of FAST was 1.9 ms, which is faster than the preprocessing step by a factor of 10.

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 16.7 ms per feature on average. The high computation time is unusual for a binary descriptor and contradicts the findings in the literature. LIOP is similar in performance, while SIFT is two times faster. Surprisingly, the performance of SURF is in the same order of magnitude as ORB and BRISK.

Fig. 25 illustrates the average computation times of the descriptors per frame. The matching times are represented in purple. As expected, the matching times for the binary descriptors are the fastest, scoring and average of 2.5 ms per frame (75 features). ORB features are the fastest to be matched at an average of 1.9 ms per frame. The distribution-based descriptors are on average one order of magnitude slower in terms of matching speed, at 24 ms; SURF features are the fastest of the kind, scoring 14.5 ms on average.

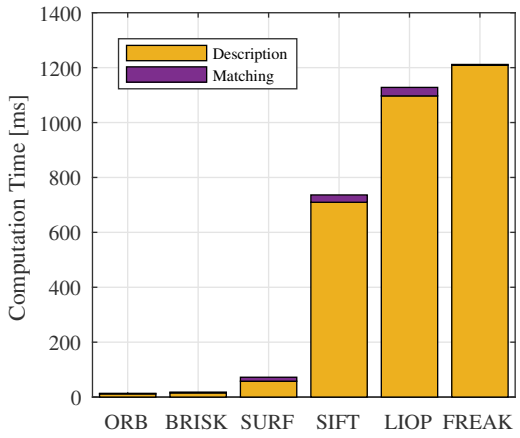


Figure 25: Comparison of average feature description and matching times per frame.

5.4.1. FREAK Results

Given that the previous experiments recorded abnormally high execution times for the FREAK descriptor, the benchmarks are repeated, this time on a desktop workstation with an Intel(R) Core(TM) i7-6700 processor (x64) at 3.40 GHz. Fig. 26 compares the speed-up times for the six descriptors and two processors relative to the heaviest test run. It can be seen that the relative ranking of the algorithm changes for the x64 processor, where FREAK totals as the third fastest descriptor. It is almost two orders of magnitude faster than LIOP, whereas the execution times are identical on the ARM processor. The relative ranking of the distribution-based descriptors is the same on both processors, and they maintain approximately the same proportions in terms of runtime. However, BRISK is the fastest running descriptor for the x64 processor, being 243 % faster than ORB; for the ARM processor it was 21 % slower. This seems to suggest implementation issues in the case of the binary descriptors, i.e. the algorithms are optimised differently depending on the architecture.

6. Conclusion

In this paper, several state-of-the-art feature detectors and descriptors have been benchmarked in the context of an ADR application. To this end, a novel

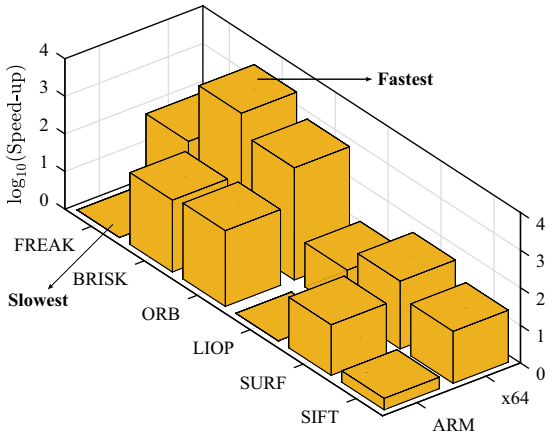


Figure 26: Comparison of descriptor speed-up factors in different processors.

dataset was created using a realistic camera simulator for space scenarios, featuring a rendezvous with the defunct spacecraft Envisat. This dataset encompasses two different trajectories, one during a sunlight period and one during eclipse, imaged in two different modalities, the visible and the LWIR, yielding four different scenarios. The performance of the IP algorithms has then been benchmarked for these scenarios, providing a multispectral evaluation of the low-level processes in computer vision required for a further integration in a vision-based navigation system.

The presented benchmarks have shown that features in the LWIR domain are generally more repeatable than in the visible. In terms of matching score, the difference between the two modalities is smaller when the target appears small in the FOV of the camera, and greater at shorter distances in favour of the LWIR, meaning that the shadows and noise in the visible become more noticeable and the algorithms become more sensitive, which could be a limitation of this modality for short ranges. Conversely, this could mean an advantage of using LWIR imaging as a way to bypass the difficulties of optical navigation relative to a complex spacecraft bearing non-imaging-friendly components such as MLI.

In terms of the analysis of feature descriptors, it was found that the perfor-

mance depends on the type of feature used: when DoG features were used, the performance is better on the visible, but the performance became better on the LWIR with Fast-Hessian features. The latter were shown to be larger in radius than the former, hence capturing a larger support region.

The results also shown the advantage of thermal imaging in eclipse sequences. Using visible imaging, all detectors have shown a decline in matching score, and the benchmarking of the descriptors resulted in a lower number of matches and elevated false positives. Regardless of the sequence, the IP algorithms have performed substantially worse when testing large baseline transformations, which could hinder the development of model-based visual navigation pipelines when only feature points are used.

With respect to computation times, it was found that, for a fixed number of 75 features per frame, only one of the detectors (DoG) and two of the descriptors (LIOP, FREAK) exceed the computation budget of 1000 ms. FAST has shown the largest speedup factor (814) with respect to the traditional DoG, and in general the corner detectors were faster to compute than the blob detectors. As expected, the binary descriptors (ORB, BRISK) demonstrated lower running times with respect to SURF, SIFT, LIOP; the exception was FREAK, although its large processing time was subsequently shown to be related to its current OpenCV implementation in the ARM architecture.

The benchmarks have additionally provided an interesting insight into the state-of-the-art baseline algorithms such as SIFT and SURF. The latter, for instance, provided higher scores with Fast-Hessian features than with its native detector, DoG. In general, the results have motivated combining different detectors and descriptors to boost performance. Overall, a combination of Fast-Hessian with FREAK is capable of providing adequate performance for a vision-based navigation in the context of ADR. However, it is currently compromised by its current implementation in the low-performance ARM processor. Fast-Hessian + BRISK offers similar performance and is computationally efficient, as it was shown to run inside the boundaries of the considered low acquisition frame-rate, taking up slightly over 20 % of the computational budget, leaving

the remaining 80 % open for the relative pose estimation taks. Furthermore, the benchmark of Fast-Hessian + BRISK is comparable in both spectra, meaning it could potentially be used for a multispectral navigation algorithm, analysing a frame of each modality per cycle, and it would still perform the detection and description tasks in less than half of the budget with lower 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. A potential direction for future work would involve an investigation of the improvement of the performance of IP algorithms for model-based navigation.

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] D. H. McKnight, Pay Me Now or Pay Me More Later: Start the Development of Active Orbital Debris Removal Now, 2010.

[2] D. J. Kessler, B. G. Cour-Palais, Collision Frequency of Artificial Satellites: The Creation of a Debris Belt, Journal of Geophysical Research 83 (A6) (1978) 2637. doi:10.1029/ja083ia06p02637.

[3] M. Andrenucci, P. Pergola, A. Ruggiero, Active Removal of Space Debris: Expanding Foam Application for Active Debris Removal, Tech. rep., ESA, ESTEC, Noordwijk, NL (2011).

[4] C. Bonnal, J.-M. Ruault, M.-C. Desjean, Active Debris Removal: Recent Progress and Current Trends, Acta Astronautica 85 (2013) 51–60. doi:10.1016/j.actaastro.2012.11.009.

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

[6] R. Biesbroek, A. Wolahan, S. M. Serrano, e.Inspector: Clean Space Industrial Days, https://indico.esa.int/event/181/contributions/1378/attachments/1305/1530/e.Inspector_SARA.pdf, [Online; accessed February 2020] (2017).

[7] Özgün Yılmaz, N. Aouf, E. Checa, L. Majewski, 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 233 (3) (2017) 811–822. doi:10.1177/0954410017740917.

[8] P. V. Anderson, D. S. McKnight, F. Pentino, H. Schaub, Operational Considerations of GEO Debris Synchronization Dynamics, in: 66th International Astronautical Congress, Jerusalem, Israel, Vol. 6, p. 7.

[9] J. R. Wertz, R. Bell, Autonomous Rendezvous and Docking Technologies: Status and Prospects, in: J. Peter Tchoryk, J. Shoemaker (Eds.), Space Systems Technology and Operations, Vol. 5088, SPIE, 2003, pp. 20–31. doi:10.1117/12.498121.

[10] M. Macdonald, V. Badescu (Eds.), The International Handbook of Space Technology, Springer Berlin Heidelberg, 2014, pp. 355–356. doi:10.1007/978-3-642-41101-4.

[11] R. Szeliski, Computer Vision: Algorithms and Applications, Springer London, 2011, pp. 324–325, 183, 184, 202. doi:10.1007/978-1-84882-935-0.

[12] The vSLAM Algorithm for Robust Localization and Mapping, in: Proceedings of the 2005 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2005, pp. 24–29. doi:10.1109/robot.2005.1570091.

[13] S. Augenstein, S. M. Rock, Improved Frame-to-Frame Pose Tracking During Vision-Only SLAM/SFM with a Tumbling Target, in: 2011 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2011. doi:10.1109/icra.2011.5980232.

[14] M. Maimone, Y. Cheng, L. Matthies, Two Years of Visual Odometry on the Mars Exploration Rovers, Journal of Field Robotics 24 (3) (2007) 169–186. doi:10.1002/rob.20184.

[15] D. Rondao, N. Aouf, Multi-View Monocular Pose Estimation for Spacecraft Relative Navigation, in: 2018 AIAA Guidance, Navigation, and Control Conference, American Institute of Aeronautics and Astronautics, 2018. doi:10.2514/6.2018-2100.

[16] M. Gansmann, O. Mongrard, F. Ankersen, 3D Model-Based Relative Pose Estimation for Rendezvous and Docking Using Edge Features, in: 10^{th} International ESA Conference on Guidance, Navigation and Control Systems, ESA, Salzburg, Austria, 2017.

[17] L. P. Cassinis, R. Fonod, E. Gill, I. Ahrns, J. G. Fernandez, CNN-Based Pose Estimation System for Close-Proximity Operations Around Uncooperative Spacecraft, in: AIAA Scitech 2020 Forum, American Institute of Aeronautics and Astronautics, 2020. doi:10.2514/6.2020-1457.

[18] A. Harvard, V. Capuano, E. Y. Shao, S.-J. Chung, Pose Estimation of Uncooperative Spacecraft from Monocular Images Using Neural Network Based Keypoints, in: AIAA Scitech 2020 Forum, American Institute of Aeronautics and Astronautics, 2020. doi:10.2514/6.2020-1874.

[19] C. Schmid, R. Mohr, C. Bauckhage, Evaluation of Interest Point Detectors,

International Journal of Computer Vision 37 (2) (2000) 151–172. doi:10.1023/A:1008199403446.

[20] A. M. López, F. Lumbreras, J. Serrat, J. J. Villanueva, Evaluation of Methods for Ridge and Valley Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (4) (1999) 327–335. doi:10.1109/34.761263.

[21] D. G. Lowe, Distinctive Image Features from Scale-Invariant Keypoints, International Journal of Computer Vision 60 (2) (2004) 91–110. doi:10.1023/b:visi.0000029664.99615.94.

[22] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, L. Van Gool, A Comparison of Affine Region Detectors, International Journal of Computer Vision 65 (1-2) (2005) 43–72. doi:10.1007/s11263-005-3848-x.

[23] K. Mikolajczyk, C. Schmid, A Performance Evaluation of Local Descriptors, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (10) (2005) 1615–1630. doi:10.1109/tpami.2005.188.

[24] A. Gil, O. M. Mozos, M. Ballesta, O. Reinoso, A Comparative Evaluation of Interest Point Detectors and Local Descriptors for Visual SLAM, Machine Vision and Applications 21 (6) (2009) 905–920. doi:10.1007/s00138-009-0195-x.

[25] O. Miksik, K. Mikolajczyk, Evaluation of Local Detectors and Descriptors for Fast Feature Matching, in: Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), IEEE, 2012, pp. 2681–2684.

[26] E. Rosten, T. Drummond, Machine learning for high-speed corner detection, in: Computer Vision – ECCV 2006, Springer Berlin Heidelberg, 2006, pp. 430–443. doi:10.1007/11744023_34.

[27] S. Leutenegger, M. Chli, R. Y. Siegwart, BRISK: Binary Robust Invariant Scalable Keypoints, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2548–2555. doi:10.1109/ICCV.2011.6126542.

[28] E. Rublee, V. Rabaud, K. Konolige, G. Bradski, ORB: An Efficient Alternative to SIFT or SURF, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 2564–2571. doi:10.1109/ICCV.2011.6126544.

[29] H. Bay, T. Tuytelaars, L. Van Gool, SURF: Speeded Up Robust Features, Springer Berlin Heidelberg, 2006, pp. 404–417. doi:10.1007/11744023_32.

[30] B. Cowan, N. Imanberdiyev, C. Fu, Y. Dong, E. Kayacan, A Performance Evaluation of Detectors and Descriptors for UAV Visual Tracking, in: 2016 14$^{\text{th}}$ International Conference on Control, Automation, Robotics and Vision (ICARCV), IEEE, 2016. doi:10.1109/icarcv.2016.7838649.

[31] J. L. Blanco, J. Gonzalez, J. A. Fernández-Madrigal, An Experimental Comparison of Image Feature Detectors and Descriptors Applied to Grid Map Matching, Tech. rep., University of Malaga, Spain (2010).

[32] N. Takeishi, A. Tanimoto, T. Yairi, Y. Tsuda, F. Terui, N. Ogawa, 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 58 (1) (2015) 45–53. doi:10.2322/tjsass.58.45.

[33] P. Ricaurte, C. Chilán, C. Aguilera-Carrasco, B. Vintimilla, A. Sappa, Feature Point Descriptors: Infrared and Visible Spectra, Sensors 14 (2) (2014) 3690–3701. doi:10.3390/s140203690.

[34] J. Johansson, M. Solli, A. Maki, An Evaluation of Local Feature Detectors and Descriptors for Infrared Images, in: G. Hua, H. Jégou (Eds.), Computer Vision – ECCV 2016 Workshops, Springer International Publishing, 2016, pp. 711–723. doi:10.1007/978-3-319-49409-8_59.

[35] T. Mouats, N. Aouf, D. Nam, S. Vidas, Performance evaluation of feature detectors and descriptors beyond the visible, Journal of Intelligent & Robotic Systems 92 (1) (2018) 33–63. doi:10.1007/s10846-017-0762-8.

[36] T. Lindeberg, Scale-Space Theory: A Basic Tool for Analyzing Structures at Different Scales, Journal of Applied Statistics 21 (1-2) (1994) 225–270. doi:10.1080/757582976.

[37] C. Harris, M. Stephens, A Combined Corner and Edge Detector, in: Alvey Vision Conference 1988, Alvey Vision Club, 1988. doi:10.5244/c.2.23.

[38] H. P. Moravec, Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover, Ph.D. thesis, Stanford, CA, USA, AAI8024717 (1980).

[39] J. Shi, C. Tomasi, Good Features To Track, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition CVPR-94, IEEE Comput. Soc. Press, 1994. doi:10.1109/cvpr.1994.323794.

[40] M. Agrawal, K. Konolige, M. R. Blas, CenSurE: Center Surround Extremas for Realtime Feature Detection and Matching, in: D. Forsyth, P. Torr, A. Zisserman (Eds.), Computer Vision – ECCV 2008, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 102–115. doi:10.1007/978-3-540-88693-8_8.

[41] R. S. Stanković, B. J. Falkowski, The Haar Wavelet Transform: its Status and Achievements, Computers & Electrical Engineering 29 (1) (2003) 25–44. doi:10.1016/s0045-7906(01)00011-8.

[42] Z. Wang, B. Fan, F. Wu, Local Intensity Order Pattern for Feature Description, in: 2011 International Conference on Computer Vision, IEEE, 2011, pp. 603–610. doi:10.1109/iccv.2011.6126294.

[43] M. Calonder, V. Lepetit, C. Strecha, P. Fua, BRIEF: Binary Robust Independent Elementary Features, in: Computer Vision – ECCV 2010, Springer Berlin Heidelberg, 2010, pp. 778–792. doi:10.1007/978-3-642-15561-1_56.

[44] A. Alahi, R. Ortiz, P. Vandergheynst, FREAK: Fast Retina Keypoint, in: 2012 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012. doi:10.1109/cvpr.2012.6247715.

[45] R. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, 2nd Edition, Cambridge University Press, Cambridge, UK, 2004, pp. 32–33, 88–93. doi:10.1017/cbo9780511811685.

[46] P. Alcantarilla, J. Nuevo, A. Bartoli, Fast Explicit Diffusion for Accelerated Features in Nonlinear Scale Spaces, in: Procedings of the British Machine Vision Conference (BMVC) 2013, British Machine Vision Association, 2013. doi:10.5244/c.27.13.

[47] M. A. Fischler, R. C. Bolles, Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography, Communications of the ACM 24 (6) (1981) 381–395. doi:10.1145/358669.358692.

[48] G. D. Evangelidis, E. Z. Psarakis, Parametric Image Alignment using Enhanced Correlation Coefficient Maximization, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (10) (2008) 1858–1865. doi:10.1109/tpami.2008.113.

[49] W. K. Widger, M. P. Woodall, Integration of the Planck Blackbody Radiation Function, Bulletin of the American Meteorological Society 57 (10) (1976) 1217–1219. doi:10.1175/1520-0477(1976)057<1217:IOTPBR>2.0.CO;2.



https://dspace.org/

Benchmarking of local feature detectors and descriptors for multispectral relative navigation in space

Rondao, Duarte; Aouf, Nabil; Richardson, Mark A.; Dubois-Matra, Olivier

2020-04-07
Elsevier
https://doi.org/10.1016/j.actaastro.2020.03.049
Rondao D, Aouf N, Richardson MA, Dubois-Matra O. (2020) Benchmarking of local feature detectors and descriptors for multispectral relative navigation in space, Acta Astronautica, Volume 172, July 2020, pp. 100-122

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