# Image processing techniques for vision-based navigation around small bodies

Tesi di Laurea Magistrale in
Space Engineering - Ingegneria Spaziale

Author: **Lorenzo Beccari**

Student ID: 941980
Advisor: Francesco Topputo
Co-advisors: Paolo Panicucci, Mattia Pugliatti
Academic Year: 2021-22

# Abstract

The increasing interest in small bodies such as asteroids is due to both scientific and commercial motivations. The scientific community believes that studying small bodies could provide useful information regarding how life developed on our planet. In addition, asteroids contain metals, some of which are valuable and scarce on Earth. Finally, the interest in those bodies is due to planetary protection reasons. As a consequence, the space industry has growing interest in producing spacecrafts that are as cheap and as autonomous as possible to explore these bodies. In this framework, it is necessary to develop navigation techniques that allow to operate autonomously in such environments. A promising approach consists in vision-based navigation, that would allow to estimate the state of a spacecraft from images acquired from a navigation camera. In this work, feature tracking and matching techniques for vision-based navigation are investigated. These techniques consist in extracting reference points, or features, from the images, and tracking or matching them between different frames. The apparent displacement of the features on the image plane can be used to reconstruct the state of the spacecraft by means of localization algorithms. The performances of traditional techniques, i.e. not based on machine learning, are compared to a convolutional neural network specifically designed for features extraction and tracking. Results show lower errors and less outliers for traditional techniques, while faster computations for the CNN.

**Keywords:** Asteroid, Vision-based navigation, Feature tracking, Features extractors, CNN

# Abstract in lingua italiana

Il crescente interesse in corpi celesti di piccole dimensioni, come gli asteroidi, è attribuibile a ragioni sia scientifiche che commerciali. La comunità scientifica ritiene che lo studio di tali corpi possa fornire importanti informazioni riguardanti lo sviluppo della vita sul nostro pianeta. Inoltre, gli asteroidi contengono grandi quantità di metalli, alcuni dei quali sono preziosi e presenti sulla Terra solo in scarse quantità. Infine, l'interesse per questi oggetti è dovuto a motivi di protezione planetaria. Di conseguenza, l'industria spaziale ha un crescente interesse nel produrre veicoli spaziali che siano il più economici e autonomi possibile, in modo da poterli usare per l'esplorazione di corpi di questo tipo. In questo scenario, è necessario studiare tecniche di navigazione che consentano di operare in autonomia in ambienti di questo tipo. Un approccio promettente consiste nelle tecniche di navigazione ottica, che consentono di stimare lo stato del veicolo a partire dalle immagini acquisite da una fotocamera di navigazione. In questo lavoro sono state studiate tecniche di feature tracking e matching per la navigazione ottica. Queste tecniche consentono di estrarre dei punti di riferimento dalle immagini e tracciarli attraverso i vari frames. Tali punti possono poi essere utilizzati per stimare lo stato del veicolo, usando algoritmi di determinazione della posizione. Le prestazioni delle tecniche tradizionali, ovvero che non utilizzano machine learning, sono confrontate con le prestazioni di una rete neurale convoluzionale sviluppata specificamente per l'estrazione e il tracciamento di tali punti di riferimento. Si osservano errori minori e meno outliers con le tecniche tradizionali, mentre la rete neurale risulta più rapida dal punto di vista computazionale.

**Parole chiave:** Asteroide, Navigazione ottica, Tracciamento di features, Estrazione di features, CNN

# Contents

# Conventions and nomenclature

## Scalars, vectors and matrices

- A scalar is indicated with lower-case font, such as $a$
- A vector is indicated with lower-case, bold font, such as $\boldsymbol{a}$
- A matrix is indicated with upper-case, bold font, such as $\boldsymbol{A}$
- A reference frame is indicated with mathematical font, such as $\mathcal{A}$

## Glossary

The following is a brief recap of the main acronyms used in the work:

- **NFT**: Natural Feature Tracking
- **NCC**: Normalized Cross Correlation
- **DEM**: Digital Elevation Map
- **NN**: Neural Network
- **CNN**: Convolutional Neural Network
- **IAU**: International Astronomical Union
- **AU**: Astronomical Unit
- **ESA**: European Space Agency
- **NASA**: National Aeronautics and Space Administration
- **ESA**: European Space Agency
- **AIDA**: Asteroid Impact and Deflection Assessment
- **DART**: Double Asteroid Redirection Test
- **NAVCAM**: Navigation Camera
- **OLA**: OSIRIS-REx Laser Altimeter
- **TM**: Target Marker
- **PDS**: Planetary Data System
- **SBN**: Small Bodies Node
- **KLT**: Kanade-Lucas-Tomasi
- **SIFT**: Scale Invariant Feature Transform
- **LoG**: Laplacian of Gaussian
- **DoG**: Difference of Gaussian
- **SURF**: Speeded-Up Robust Features

- **BRIEF**: Binary Robust Independent Elementary Features
- **Features from Accelerated Segment Test**
- **ORB**: Oriented FAST and Rotated BRIEF
- **BRISK**: Binary Robust Invariant Scalable Keypoints
- **AKAZE**: Accelerated KAZE
- **FED**: Fast Explicit Diffusion
- **CenSurE**: Center Surround Extremas for Realtime Feature Detection and Matching
- **LOS**: Line Of Sight
- **SLAM**: Simultaneous Localization And Mapping
- **BRDF**: Bidirectional Reflectance Distribution Function
- **HAFOV**: Horizontal Angular Field Of View
- **TAGCAMS**: Touch And Go Camera System
- **NDC**: Normalized Device Coordinates

## 0.1. Working environment

The work has been carried out on a laptop running Ubuntu 20.04.3 with the following hardware specifications:

- **CPU:** Intel Core i7-7700HQ
- **GPU:** NVIDIA GeForce GTX 1050 Mobile, 4040 MiB memory
- **RAM:** 16 GB

The work had been carried out for the vast majority in Python. Anaconda[1] had been used to simplify the management of the working environment. Anaconda is a Python distribution for scientific computing, with the big advantage that it embeds a package management tool, named "conda", that simplifies the management of package and libraries with respect to other tools. An Anaconda environment running Python 3.6 had been set up. This specific Python version had been chosen for compatibility reasons with external code that had been used in the work.
For the tasks directly related to the work packages, the following libraries have been used:

- OpenCV for image processing with "traditional" techniques.
- TensorFlow for deep learning tasks.

[1] `https://www.anaconda.com/`

Other standard libraries, not mentioned, have been used.
MATLAB had been used for those tasks that required data taken from the SPICE system.
For the creation of the model and the generation of the images, Blender 3.0[2] had been used. Blender is an open-source 3D modeling software with an embedded Python console, that allows to run Python scripts directly inside the program. This allows to automate tasks that require to be executed within the Blender environment.
Google Colab[3], a cloud computing platform developed by Google, has been used to train the neural network. However, the network was tested on the same hardware previously mentioned.

[2]https://www.blender.org/
[3]https://colab.research.google.com/?utm_source=scs-index

# 1 | Introduction

According to the IAU Resolution B5 [3], a small body is defined as a body that is not a planet, a dwarf planet nor a satellite. These objects can be found throughout the Solar System, but in some areas the concentration is higher:

- The main asteroid belt between the orbits of Mars and Jupiter.
- The Lagrange points of the orbits of the planets, particularly Jupiter. The asteroids found in the Lagrangian points are called Trojan asteroids.
- The Kuiper belt, located beyond the orbit of Neptune. These objects are called Trans-Neptunian objects.
- The Oort cloud, a spherical zone located between 20000 and 50000 AU.

The domain of small bodies includes the classes of comets and asteroids. Generally, an object is considered a comet[1] when:

- It manifests cometary activity, i.e. tails are formed when the object is in proximity of the Sun.
- It does not display cometary activity due to its orbit not being close enough to the Sun, but the composition suggests that it could become active.
- Objects with an open orbit with respect to the Sun, i.e. that will not return in the solar system.

Comets are composed by a cometary nucleus, an atmosphere and a tail. The nucleus consists in a solid body composed by ice, dust and other materials, with the dimensions generally in the order of 10 kilometers or lower. When the comet reaches the proximity of the Sun, the ice in the nucleus sublimates, forming an atmosphere and a tail that can reach a length of a million of kilometers.
Asteroid are bodies that can not be classified as comets. The scale of asteroids is strongly variable: the smallest known asteroids have dimensions less than 10 meters across, while

[1]`https://www.britannica.com/science/small-body`

the largest, generally of pseudo-spherical shape, have diameters of several hundreds of kilometers. The largest asteroid known, Vesta, has a diameter of about 530 kilometers[2]. The interest in these objects has seen a strong growth due to multiple reasons. Small bodies are essentially residuals of the pristine material of the process of formation of the planets during the formation of the solar system. For this reason, these objects could provide information regarding the formation of the solar system and the origin of life on Earth. To this purpose, missions have been and are being planned to collect samples of these objects and study them, in-situ or by bringing a sample to Earth. The ESA mission Rosetta [37], launched in 2004, extensively studied the comet 67P/Churyumov–Gerasimenko. The NASA mission OSIRIS-REx [15], launched in 2016, collected in 2020 a sample of the asteroid 101955 Bennu, and will return it back to Earth in 2023.
The Hayabusa [81] and Hayabusa2 [72] missions were two sample return missions developed by JAXA that targeted, respectively, the asteroids 25143 Itokawa and 162173 Ryugu. Both the missions successfully returned samples of the asteroids back to Earth.
Another important aspect is the possibility of exploiting the resources contained in small bodies. During the formation of the Earth, a large amount of precious materials such as gold, platinum, palladium, rhodium and other rare metals have been retained close to the core by the strong gravitational force. On the contrary, in asteroids these elements are distributed more evenly, and hence more easily accessible from the surface[3]. Many organizations are considering to exploit these bodies to access large amount of extremely valuable resources.
Finally, the focus on these objects is also due to planetary protection reasons. The impact of a large object with the Earth could have a catastrophic effect for the human civilization. There is hence interest in monitoring potentially hazardous objects, and developing strategies to reduce the risk of a possible collision. The Asteroid Impact and Deflection Assessment (AIDA) [22] mission is an international cooperation between ESA and NASA, and will target the binary system composed by the asteroid 65803 Didymos and its satellite 65803 I Dimorphos. NASA will provide the Double Asteroid Redirection Test (DART) mission, which will perform a kinetic impactor to deflect the satellite 65803 I Dimorphos, and ESA will provide the HERA mission, which will monitor the deflection and study the Didymos system.
In order to enable missions targeting those bodies, vehicles should be equipped with systems that allow an high degree of autonomy. An approach consists in vision-based navigation techniques, which enable to estimate the spacecraft state by means of images taken by a navigation camera, that are processed in order to extract useful information.

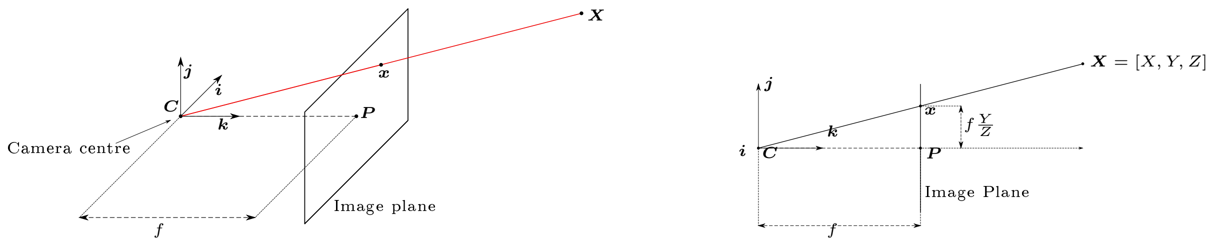
[2]`https://solarsystem.nasa.gov/asteroids-comets-and-meteors/asteroids/in-depth/`
[3]`https://web.mit.edu/12.000/www/m2016/finalwebsite/solutions/asteroids.html`

This work consists in a comparison of the performance of algorithms aimed to detect points from images that can be used as measurements to estimate the spacecraft state.

# 2 | State of the art

## 2.1. Camera fundamentals

The simplest model for a camera is the pinhole camera model, as described in Hartley and Zisserman [42]. In the pinhole camera, rays are directly projected on the image plane, and no distortion due to the lenses is considered. The geometry of the pinhole camera model is shown in Figure 2.1.



(a) Spatial view. (b) Side view.

Figure 2.1: Geometry of the pinhole camera model. Note that this convention, used in Hartley and Zisserman [42], is different from the convention used by Blender, in which the $\boldsymbol{i}$ and $\boldsymbol{k}$ axes are in the opposite directions. Image modified from Hartley and Zisserman [42].

Consider a reference frame in which the directions $\boldsymbol{i}$ and $\boldsymbol{j}$ define the image plane, while $\boldsymbol{k}$ is perpendicular to the image plane. This frame is called camera coordinate frame $\mathcal{C}$. The origin of the camera coordinate frame is called camera center $C$. The intersection between $\boldsymbol{k}$ and the image plane is called principal point $P$. The $\boldsymbol{k}$ direction is also referred to as principal axis (note that, in Blender, the principal axis is in $-\boldsymbol{k}$ direction). Let $f$ be the distance of the camera centre from the image plane in the principal axis direction, also known as focal length, and let $(X, Y, Z)$ be a point. The projection of the point in the image plane will be:

$$\begin{cases} x = f\frac{X}{Z} \\ y = f\frac{Y}{Z} \end{cases} \tag{2.1}$$

Where x and y are the coordinates of the projected point on the sensor device.
Hence, the pinhole camera model is essentially a function that maps points from world coordinates to image plane coordinates. It is important to point out that, if the spatial coordinates of the point are not known, it is not possible to uniquely determine its true spatial position. If the coordinates on the image plane $(x, y)$ are available, only $\frac{X}{Z}$ and $\frac{Y}{Z}$ can be determined, according to Equation (2.1). Hence, the point lies on the line of sight (LOS) passing through the camera center and the projected point, but can't be uniquely placed on the LOS. As a consequence, images obtained from a single pinhole camera can't be used to uniquely reconstruct the scene, and the scale of the scene remains undefined.
More generically, Equation (2.1) can be written, less than a scale factor $Z$, as:

$$\begin{bmatrix} Zx \\ Zy \\ Z \end{bmatrix} = \begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} = \boldsymbol{P} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{2.2}$$

The $3 \times 4$ matrix in Equation (2.2), called camera matrix $\boldsymbol{P}$, is decomposed as $[\boldsymbol{K}|\boldsymbol{0}]$. The matrix $\boldsymbol{K}$ is called camera intrinsic matrix, and for a pinhole camera with no principal point offset assumes the form:

$$\boldsymbol{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.3}$$

In general, the position of a point or an object is known in the world coordinate frame $\mathcal{W}$ that is generally distinct from the camera coordinate frame $\mathcal{C}$. Let $\boldsymbol{X}_{\mathcal{W}}$ be a vector that defines a point in the world coordinate frame, and $\boldsymbol{X}_{\mathcal{C}}$ a vector that defines the same point in the camera coordinate frame. The relation between $\boldsymbol{X}_{\mathcal{W}}$ and $\boldsymbol{X}_{\mathcal{C}}$ is:

$$\boldsymbol{X}_{\mathcal{C}} = \boldsymbol{R}_{\mathcal{CW}} \left( \boldsymbol{X}_{\mathcal{W}} - \tilde{\boldsymbol{c}} \right) \tag{2.4}$$

In equation (2.4), $\tilde{\boldsymbol{c}}$ indicates the camera centre in the world coordinate frame, and $\boldsymbol{R}_{\mathcal{CW}}$ is a rotation matrix from the world coordinate frame to the camera coordinate frame. The matrix $\boldsymbol{P}$ can be applied to the vector $\boldsymbol{X}_{\mathcal{C}}$, or alternatively it can be corrected to account for the point expressed in the world coordinate frame:

$$\boldsymbol{P} = \boldsymbol{K} \left[ \boldsymbol{R}_{\mathcal{CW}} | - \boldsymbol{R}_{\mathcal{CW}} \tilde{\boldsymbol{c}} \right] \tag{2.5}$$

The calibration matrix $\boldsymbol{K}$, reported in Equation (2.3), can represent more complex camera

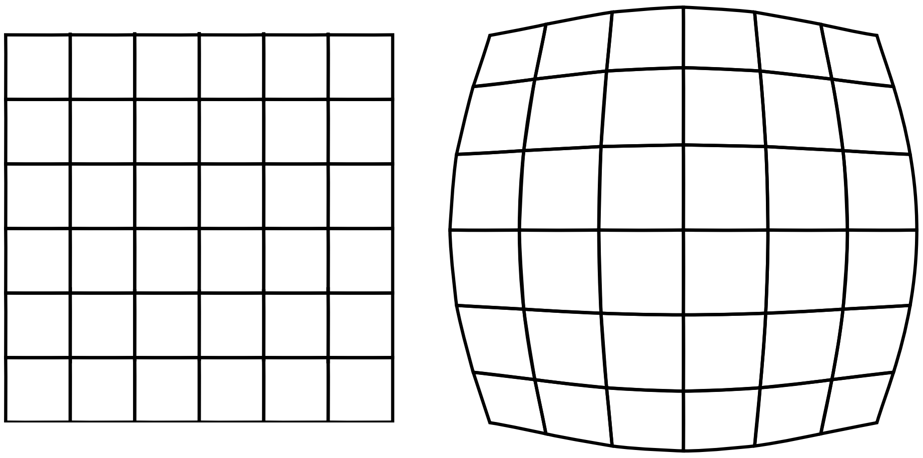
models with additional terms, that account for factors such as principal point offset with respect to the image reference frame. However, the camera models expressed in this form are called linear, since no distortion is present: the rays that form the image on the image plane remain straight. This is not the case of real world cameras, since the presence of lenses deviates the rays.
Real world cameras are not well described by the linear model, since the presence of lenses causes a deflection of the light rays. This is due to refraction, i.e. the deviation that a wave undergoes when the transmission medium changes. Distortion needs to be addressed, since it changes the relation between 3D coordinates of the points and 2D coordinates on the pixels on the image plane. The effect of distortion is shown in Figure 2.2, and a comparison between a linear and non-linear camera is shown in Figure 2.3 for barrel distortion.
Distortion is corrected by converting the actual measurements in those measurements that would be obtained with a linear camera model. More specifically, a distortion function $L(r)$ is determined during calibration. The linear coordinates $(x, y)$ are determined as:

$$\begin{bmatrix} x \\ y \end{bmatrix} = L(\tilde{r}) \begin{bmatrix} x_d \\ y_d \end{bmatrix} \tag{2.6}$$

Where $x_d$ and $y_d$ are the measured distorted coordinates, and $\tilde{r}$ is the distance from the center of distortion. The distortion function can be computed starting from an image whose lines are known to be straight, and comparing with the acquired lines distorted by the lenses. The procedure is described in more detail in Hartley and Zisserman [42].

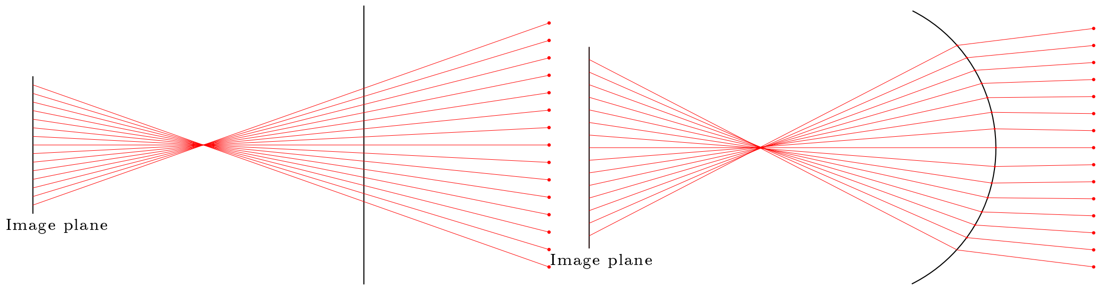
## 2.2. Optical navigation

Optical navigation techniques consist essentially in methods capable of estimating the spacecraft state, intended in this work as position, with images taken by a navigation camera.
When the spacecraft is relatively far from the asteroid, horizon-based optical navigation techniques are used [67]. These techniques essentially consist in determining the body-spacecraft relative position by processing the image of the entire body, by using global morphological information. These techniques are suitable when the spacecraft is far from the body and it is not possible to use local information of the small body morphology. A survey and development of these techniques, using both AI-based and non AI-based methods, is carried out in Pavoni [64].
When morphological features can be distinguished, an alternative approach can be used.



(a) Linear camera (no distortion). (b) Real camera (with distortion).

Figure 2.2: Comparison between a the image of a grid taken with a linear camera with no distortion due to lenses, shown in Figure 2.2a, and a real camera with barrel distortion due to lenses, shown in Figure 2.2b.



(a) Pinhole camera (no distortion). (b) Real camera (with distortion).

Figure 2.3: Comparison between a pinhole camera and a non-linear camera. In the pinhole camera, shown in Figure 2.3a, the rays are not distorted. In a non-linear camera, shown in Figure 2.3b, the rays are distorted by the lens.

A convenient navigation solution is feature based navigation: if geomorphological features are resolved on the image, it is possible to use those which are considered particularly significant and recognizable as measurements to estimate the spacecraft state. These features can be effectively meaningful physical characteristics, such as craters (Owen et al. [60]), or simply considered significant by algorithms specialized in their extraction. Once these points are identified, the spacecraft state is estimated (up to a scale factor) on the base of the movement on these points on the image. These tasks can be carried out using non-AI-based methods, that rely on various image processing techniques to localize possible features, or methods based on deep learning, particularly convolutional neural networks, that can be specialized in recognizing those points with an appropriate training.

## State of the art

Feature-based navigation techniques have been used in past missions and are planned to be used in future missions. Here, some examples are provided.

**NEAR** The Near-Earth Asteroid Rendezvous mission (NEAR) [21], launched on on February 17, 1996, targeted the asteroid 433 Eros, orbited it for about one year and carried out scientific operations. According to Owen et al. [60], images of the surface were taken and the craters found were modeled as circles, and added to a database which stored their position with respect to the center of mass of the asteroid, radius and orientation. The process required human intervention to identify the craters to add to the database. The centers of those circles were used as measurements to estimate the spacecraft state.

**HERA** The HERA mission [22], to be launched in 2024, is part of the Asteroid Impact and Deflection Assessment mission (AIDA). The goal of AIDA is to investigate the possibility of deviating a small body using a kinetic impactor (DART) on the satellite of the Didymos binary system. The HERA spacecraft will then be sent to study the effect of such impact. The navigation technique of the HERA mission is based on the Kanade-Lucas-Tomasi (KLT) feature tracker [65]. The algorithm consists in extracting features from the image, and following their motion in the different frames [51, 75, 79].

**OSIRIS-REx** OSIRIS-REx [15] is a NASA mission launched on the September 8, 2016. On October 20, 2020 the spacecraft collected a sample from asteroid 101955 Bennu. A dedicated navigation software, called Natural Feature Tracking (NFT), has been developed by Lockheed Martin [49]. The procedure followed by the NFT software consisted, conceptually, in generating a digital model of the asteroid topography using Stereophotoclinometry (SPC) [34]. This model is then used, together with the information

regarding the direction of the Sun vector, to predict the appearance of the terrain.
More precisely, a Digital Elevation Map (DEM) of the asteroid, that comprehends an elevation map and an albedo map, is generated [12]. DEMs are computed from images through SPC , using the information acquired by the navigation camera (NAVCAM) and the laser altimeter (OLA), and are stored onboard.
Once the DEM is available, if the Sun vector is known, the appearance of the terrain can be rendered. Rendering is performed on board. The predicted appearance of certain features is compared with the current image taken by the navigation camera through cross-correlation [45].

**Hayabusa2** Hayabusa2 [72] is a Japanese sample return mission launched in 2014, that visited the asteroid 162173 Ryugu in 2018. The navigation system used by Hayabusa2 differs from other missions since, in this case, artificial landmarks, called Target Markers (TM), were used. These elements, made by a reflective material, were released on the asteroid's surface. They reflected the light of a flash lamp on the spacecraft, and were hence easily recognizable by the navigation camera. The navigation algorithm is based on the detection and tracking of the bright TM on the asteroid surface [58].

**Rosetta** The Rosetta mission [37], designed by ESA and launched in 2004, had an objective the study of the comet 67P/Churyumov-Gerasimenko. According to Santayana and Lauer [73], maplets consisting in albedo and elevation of landmarks are generated using SPC. The landmark appearance is then predicted taking into account the illumination conditions, and correlation is used to localize the projections of the landmarks on the actual images taken by the NAVCAM. The process is carried out by the ground segment, with human intervention.

### 2.2.1. Traditional image processing techniques

Traditional optical techniques are those which do not use deep learning. From a general standpoint, these techniques consist in image processing algorithms that enable to locate features in the image.

#### Feature extraction and tracking

features extractors are image processing algorithms that are able to localize points in the image that are particularly distinctive due to the pixel intensities in the neighborhood of the point. In some cases a descriptor, i.e. a vector that embeds properties of the feature, is added in order to allow to recognize features through different images. In this case,

the term "feature" refers to both the point and the descriptor. In some cases, the term "keypoint" is used to indicate the location of a feature.
The interest in detecting features in an image is motivated by the fact that these points can be recognized under certain changes in the scene, such as the change of the viewpoint, and the movement can be reconstructed based on the displacement of the features in the image. According to Morrell et al. [55], two approaches are generally used to follow or recognize features between different images:

- Feature matching consists in extracting separately the features from different images, along with the descriptors. The descriptors associated to the features are compared, and matched if they are found similar.
- Feature tracking consists in following the features in subsequent images, based on local changes of image intensity. It is not necessary to have descriptors, since there is no matching. However, the tracked features are searched only in the neighborhood of the previous features. Hence, these algorithms struggle to work if images are not similar enough.

An introduction to the main features extraction and description algorithms is presented below. The algorithms used in the work are presented more in detail in Section 4.1.1.
The simplest features extractors, also referred to as corner detectors, operate on the eigenvalues of the Hessian of the image, i.e. the second derivative of the image. Examples are the Harris corner detector [6] and the Shi-Tomasi [5] corner detector. Both are based on the extraction of the eigenvalues of a matrix that contains image derivatives. A "score" $R$ is calculated, in a slightly different way in the two algorithms, that determines if a region of the image is flat, an edge or a corner. Corners are generally the easiest regions to track, due to the high gradients in both the directions.
SIFT (Scale Invariant Feature Transform) [50] is able to extract features that are invariant to scaling. In order to detect the edges of an image, Laplacian filters are commonly used, and their use is discussed in Haralock and Shapiro [39]. However, since this operation is computationally expensive, SIFT approximates the LoG with the Difference of Gaussian (DoG). The DoG is obtained by blurring the image with Gaussian filters, at different scales ($\sigma$), and by computing the difference between consecutive scales. features are detected as maxima in the DoG. An orientation is assigned to each feature in order to achieve orientation invariance, and the descriptor is created taking into account the orientations of patches in the neighborhood of the feature location.
SURF (Speeded-Up Robust Features) [13] further approximates the Laplacian of Gaussian by means of box filters, i.e. filters that approximate the second derivatives of the Gaussian function with boxes of constant value. The convolution of box filters with the image is

computed, and a Hessian matrix function is built for each point of the image by means of the results of the convolution. features are maxima in the Hessian matrix determinant. The descriptor is built according to the response to Haar wavelets of patches taken from the neighborhood of the feature. The algorithm is less accurate than SIFT, but according to the authors is about three times faster thanks to the approximation of box filters.

BRIEF (Binary Robust Independent Elementary Features), presented by Calonder et al. [20], is an algorithm that computes descriptors for pre-existing features locations. The descriptors are built by performing a certain number of intensity comparisons between pixels sampled from a patch built around the feature location. The accuracy in terms of detection rate is comparable to SURF descriptors, but the whole process can be about two order of magnitude faster (depending on the feature detector used) [19]. In fact, the detection can be performed with any algorithm, such as a corner detector or an algorithm faster than SURF or SIFT. In addition, description and matching are faster, according to Calonder et al. [19].

FAST (Features from Accelerated Segment Test), presented by Rosten et al. [70], is an algorithm developed specifically for real-time applications in robotics. Features are located by means of intensity tests performed in the neighborhood of a feature point candidate. The algorithm can be improved with a machine learning approach based on decision trees. Feature description is not provided.

ORB (Oriented FAST and Rotated BRIEF), presentet by Rublee et al. [71], is an open source algorithm developed by the OpenCV team, as alternative to SURF and SIFT. The algorithm consist, conceptually, in extracting the features with a modified implementation of FAST in order to take into account the orientation, and attaching BRIEF descriptors. The result is an algorithm that, according to the authors, is much faster than SURF and SIFT.

BRISK (Binary Robust Invariant Scalable Features) [44] uses AGAST (Mair et al. [52]), essentially an extension of FAST for accelerated performance, as feature detector, and a variant of BRIEF to build the descriptors.

AKAZE (Accelerated KAZE), presentet by Alcantarilla et al. [10], is an algorithm for both feature detection and description. The procedure is conceptually similar to SIFT and SURF. However, instead of Gaussian blur, a function named FED (Fast Explicit Diffusion) is applied.

STAR is the OpenCV implementation of a feature detector based on CenSurE (Center Surround Extremas for Realtime Feature Detection and Matching), presented by Agrawal et al. [9]. The basic idea is that a source of error introduced by algorithms such as SIFT is caused by the down-sampling of the image in the construction of the DoG. In CenSurE, the image is not down-sampled, and the Laplacian of Gaussian is approximated by bi-

level filters of different geometric shapes. Such filters can be applied by means of integral images, hence the computation is fast even if the image is not down-sampled.
Comparison between these algorithms in a space navigation framework is carried out by Morrell et al. [55]. The comparison is between six algorithms that constitute a complete detection and tracking system: SIFT, SURF, KLT, ORB, AKAZE, BRISK.
In general, in order to evaluate feature tracking algorithms for small bodies, the following criticalities have to be taken into account:

- The changing of illumination condition can be responsible of features that disappear, or that are not matched.
- For certain bodies, it can be particularly difficult to detect good features because of the surface texture.
- False matches can degrade the quality of the navigation. Techniques for outlier rejection such as RANSAC are therefore required.

As highlighted in that work, four characteristics are desirable:

- The error should be limited.
- The error rate, i.e. the rate with which the error increases during tracking, should be low.
- For position determination purposes, it is preferable that the features are persistent, i.e. the algorithm tracks them for a long time before losing them.
- It is preferable that the error produced has a Gaussian distribution, since many navigation filters rely on this hypotesis.

From Morrell et al. [55], KLT results the best tracking algorithm in terms of feature persistence. This results in better performance and lower computation time for Simultaneous Localization And Mapping (SLAM) algorithms. However, KLT tends to accumulate higher errors during the tracking, and the error rate is quite high.
BRISK and SIFT have a low persistence, but errors are kept at lower levels. The error rate is low.
SURF and ORB have comparable performances regarding both the persistence and the error behaviour. The persistence is comparable to BRISK and SIFT, but the error is larger.
According to Morrell et al. [55], AKAZE has a better persistance than SIFT, SURF, BRISK and ORB, but behaves poorly in terms of error rate.
The error distribution is almost Gaussian for SIFT, ORB and BRISK. AKAZE and SURF

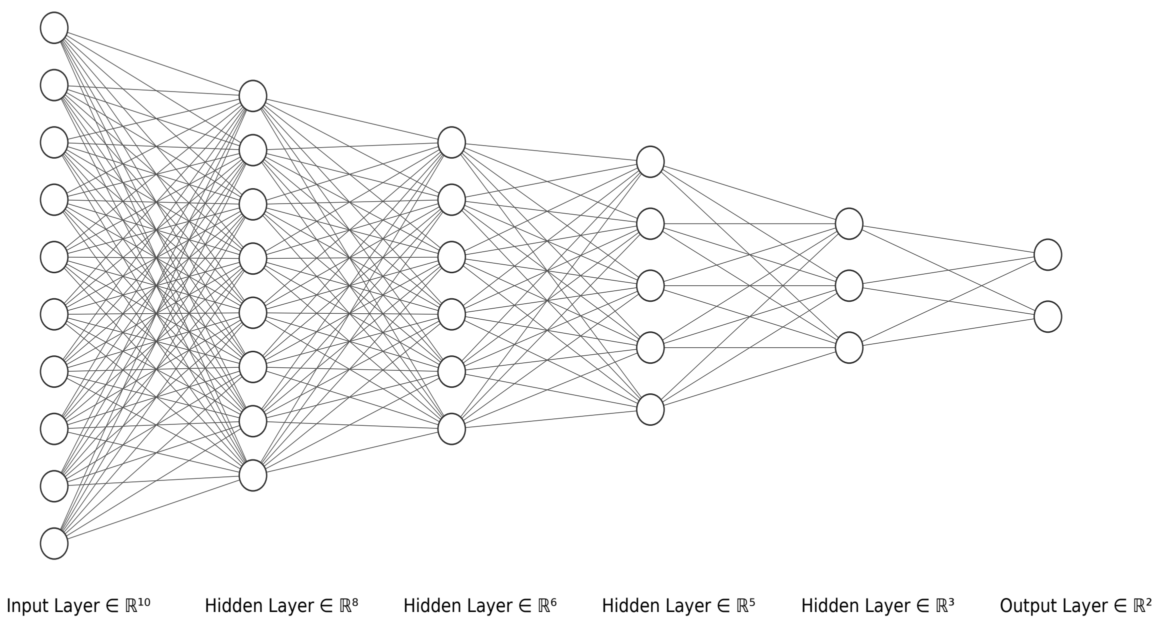


Figure 2.4: Example of the structure of a neural network.

do not have a Gaussian distribution, but the bias is small and manageable. KLT produces errors that are far from being Gaussian.

## 2.3. Navigation with convolutional neural networks

### 2.3.1. Introduction to neural networks

Neural networks (NN) are machine learning algorithms based on a collection of nodes, called neurons, that vaguely resemble the neurons in an animal brain. These algorithms can be trained using known data with the goal of making predictions in unknown, but similar, situations.

A neural network essentially consists in a set of layers. Each layer is composed by a certain number of elements, called neurons. The first layer, called input layer. receives the input given to the neural network. The last layer, called output layer, gives the result of the network, obtained by processing the input. Between input and output layer, a certain number of hidden layers is present. Each neuron of the network has an input and an output. The input of a neuron (except for the neurons of the input layer) consists essentially in the outputs of previous neurons, manipulated by parameters called weights and biases. A weight is a multiplication factor that links the output values of two neurons of consecutive layers, and is generically indicated as $w$. To this multiplication, a bias is summed. A bias is a parameter defined for each neuron, generically indicated as $b$.

Let $w^l_{jk}$ be the weight that connects the neuron $k$ of the layer $(l-1)$ to the neuron $j$ of the layer $l$. Similarly, let $b^l_j$ the bias value of the neuron $j$ of layer $l$. The value assumed by the neuron j of layer $l$, called activation, will be Equation (2.7):

$$a^l_j = \sigma\left(\sum_k w^l_{jk} a^{l-1}_k + b^l_j\right) \tag{2.7}$$

Or, in a vectorized form (2.8):

$$\boldsymbol{a}^l = \sigma\left(\boldsymbol{W}^l \boldsymbol{a}^{l-1} + \boldsymbol{b}^l\right) \tag{2.8}$$

Where $\boldsymbol{a}^l$ and $\boldsymbol{a}^{l-1}$ are vectors of output values of layers $l$ and $l-1$, $\mathbf{W}^l$ is the weight matrix that connects each activation of layer $l$ to each activation of layer $l-1$, and $\boldsymbol{b}^l$ is the bias vector for layer $l$. $\sigma$ is the activation function, used to introduce non-linearity in the network and to avoid that the output values of the neurons assume extremely large or extremely small values as the network progresses. An example of commonly used activation function is the sigmoid function:

$$\sigma(z) = \frac{1}{1+e^{-z}} \tag{2.9}$$

Other commonly used activation functions are ReLu and tanh [74]. In some of the following equations the argument of the activation function, namely $\left(\boldsymbol{W}^l \boldsymbol{a}^{l-1} + \boldsymbol{b}^l\right)$, will be generically referred to as $z$.
As stated before, the neural network is able to learn through a training process. Training a neural network consists in adjusting the weights and the biases in order to minimize the error that the network makes when providing an output on a set of data named training set. The training set is made by data that have a known output, or label.
The training process consists in two phases: a forward phase, in which data from the training set are processed and the result is compared to the labels, and a backward phase, in which the parameters of the network are modified to improve the results.
More specifically, in the forward phase, data from the training set are given as input to the neural network. These data are processed by the network by means of the weights and biases, and the output values of all the hidden layers are computed along with the output of the network. The output is then compared with the labels of the training data, and a cost function that represents this error is computed. Various types of cost functions are available, and the choice generally depends on the type of network. A very simple

example is the quadratic cost function, given by:

$$C = \frac{1}{2n} \sum_x ||\boldsymbol{y}(x) - \boldsymbol{a}^L(x)||^2 \tag{2.10}$$

Where $C$ is the cost function, $n$ the number of training examples, $\boldsymbol{y}(x)$ is the label of the training example $x$ and $\boldsymbol{a}^L(x)$ is the output value of the network referred to the input $x$. The result of the forward phase is the cost $C$. In order to make the network learn, the cost function is minimized during the backward phase by means of a gradient descent algorithm. More specifically, the goal of the backward phase is to compute the gradients of the cost function with respect to the weights and biases, and use them to update the parameters of the network. The gradients are computed starting from the output layer and proceeding backwards: for this reason, the algorithm is called back-propagation.
The back-propagation algorithm is detailed in Nielsen [57]. Conceptually, back-propagation consists in determining how weights and biases affect the cost function, i.e. in computing $\frac{\partial C}{\partial w^l_{jk}}$ and $\frac{\partial C}{\partial b^l_j}$. Nielsen [57] defines the error $\delta^l_j$ of neuron $j$ in layer $l$ as:

$$\delta^l_j = \frac{\partial C}{\partial z^l_j} \tag{2.11}$$

The first step of the back-propagation algorithm consists in computing the error of the output layer $\delta^L$ as:

$$\delta^L_j = \frac{\partial C}{\partial z^L_j} = \frac{\partial C}{\partial a^L_j}\sigma'(z^L_j) \tag{2.12}$$

By means of the back-propagation algorithm, the error of the layer $l$ can be computed knowing the error of the layer $l+1$ as:

$$\boldsymbol{\delta}^l = ((\boldsymbol{W}^{l+1})^T \boldsymbol{\delta^{l+1}}) \odot \sigma'(\boldsymbol{z}^l) \tag{2.13}$$

In Equation (2.13), the symbol $\odot$ indicates the Hadamard product (element-wise product) between two vectors. By combining Equations (2.13) and (2.12), all the $\boldsymbol{\delta}^l$ can be computed: the error on the last neuron is computed with (2.12), and all the errors on the other neurons are computed backwards with (2.13). According to Nielsen [57], the gradients of the cost function with respect to weights and biases are computed as:

$$\frac{\partial C}{\partial w^l_{jk}} = a^{l-1}_k \delta^l_j \tag{2.14}$$

$$\frac{\partial C}{\partial b^l_j} = \delta^l_j \tag{2.15}$$

Weights and biases are then modified using the gradient descent method. In the simplest form of gradient descent, weights and biases are updated as:

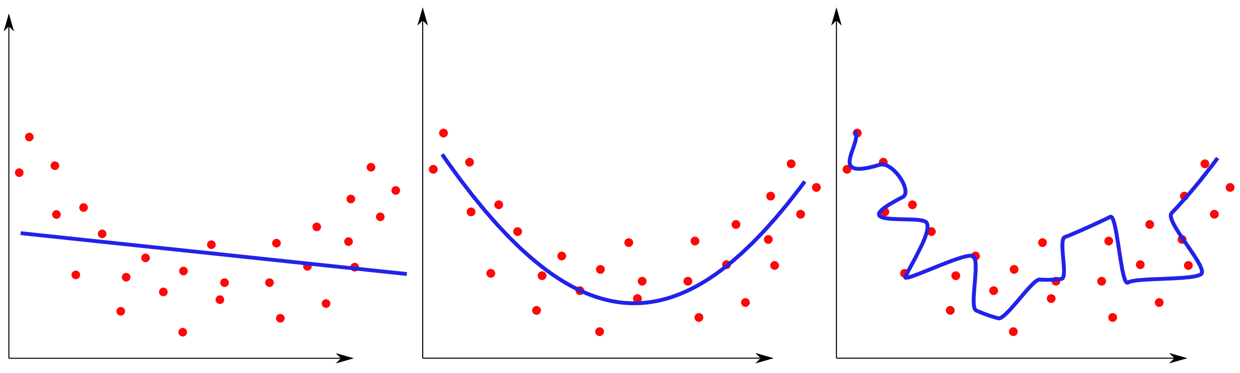
$$\partial w_{jk}^{l} = \partial w_{jk}^{l} - \alpha \frac{\partial C}{\partial w_{jk}^{l}} \tag{2.16}$$

$$\partial b_{j}^{l} = \partial b_{j}^{l} - \alpha \frac{\partial C}{\partial b_{j}^{l}} = \delta_{j}^{l} \tag{2.17}$$

In order to achieve convergence of weights and biases, more complex gradient descent algorithms could be required. A survey is provided by Aggarwal [8].
As stated above, a proper training is fundamental to generalize the output of the network to previously unseen data that have the same characteristics of the training set. If the network is not trained enough, the error is likely to be large both on the training data and on previously unseen data. This condition, in which the data of the training set are not fit well by the model, is called under-fitting, and consists conceptually in the model not capturing the logic behind the data.
A more complex situation can arise in cases in which the cost function is very low, the data of the training set are well fit by the model, but the network does not generalize well on previously unseen cases. This condition is called over-fitting, and consists conceptually in a model that is too complex and focused too well on the training set, to the point that it is not able to generalize to new examples that are not already known to the network. Ways to prevent over-fitting are:

- Stopping the training earlier.
- Improving the data quality and using more data.
- Using regularization techniques, that force the model to be less complex.
- Using dropouts, i.e. dropping some connections between neurons at different steps of the training.

More details are provided by Aggarwal [8]. In the context of linear regression, underfitting, correct fitting and overfitting are shown in Figure 2.5. Once the model has been trained on the training set, the performance of the network is evaluated on a test set, that should be made of data that were not part of the training set.
A validation set is typically used in addition. The validation set consists in data that are not part of the training set nor of the test set, used to compute periodically the evaluation metrics during the training phase. In this way, the progression of the training is monitored. While the network is training, both the losses on the test and the validation sets decrease. While the training loss keeps decreasing, the validation loss reaches a minimum when



(a) Underfitting. (b) Correct fitting. (c) Overfitting.

Figure 2.5: Underfitting, correct fitting and overfitting. If underfititng occurs, the model is not complex enough, and does not fit the data properly. If the fitting is correct, the model fits the data but not the noise on the data. If overfitting occurs, the model fits the data and the noise in the data, and does not generalize well. Image modified from [1].

the network is trained optimally, then increases when overfitting occurs. This behavior is shown in Figure 2.6.

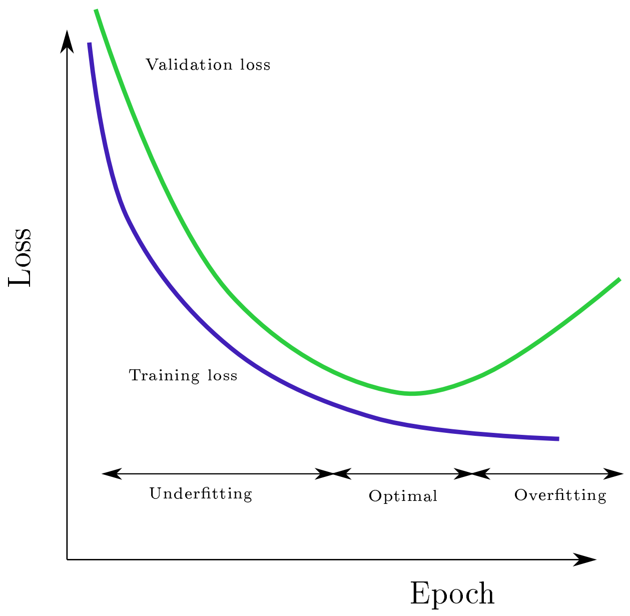


Figure 2.6: Training and validation losses. When both decrease, the network is training. When the validation loss has a minimum, the training is optimal. When the validation loss starts rising, the network is overfitting. On the other hand, the training loss keeps decreasing also when overfitting occurs.

## Convolutional neural networks

In computer vision, and hence in optical navigation, traditional neural networks are rarely used. Instead, the standard for computer vision tasks is Convolutional Neural Networks (CNN). Convolutional neural networks are a class of neural networks whose name is inspired to the convolution operation, a type of product between a grid-structured input and a matrix, named kernel, described in Equations (2.18) and (2.19). According to Yi [80], the use of kernels allows to have less trainable parameters with respect to fully connected architectures, typical of neural networks. This is important because, if traditional NN architectures were used on images, the number of trainable parameters would be enormous, since all the pixels of the image would have to be unwrapped and each of them would constitute an input unit. This would likely result in excessive memory requirements and computational cost, especially for large image resolutions. The architecture of CNNs allows to manage two-dimensional (in case of gray scale images) or three-dimensional (in case of RGB images) inputs. In addition, the use of convolutions and filters was a standard technique used in image processing even before the introduction of neural networks. An example is edge detection by means of Laplacian of Gaussian filters (Haralock and Shapiro [39]). This is possible thanks to some operations used in CNNs, here presented.

**Convolution** Convolution is a type of operation widely use in signal processing. The convolution between two multi-variable functions $f(x, y)$ and $h(x, y)$ is defined as the following integral:

$$g(x,y) = f(x,y) \star h(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\tau_u,\tau_v)h(x-\tau_u, y-\tau_v)d\tau_u d\tau_v \tag{2.18}$$

Equation (2.18) can be discretized as follows:

$$g(x,y) = \sum_{\tau_u}\sum_{\tau_v} f(\tau_u,\tau_v)h(x-\tau_u, y-\tau_v) \tag{2.19}$$

Conceptually, convolution is a type of product between a grid-structured input, such as an image, and a grid-structured set of weights and biases, called filter or kernel. The convolution operation places the filter in each possible position in the input, or in a certain sequence of positions, computes the element wise product and adds the bias, if present. It is shown, for the top-left element, in Figure 2.7. If the convolution is performed by positioning an $m \times n$ filter in each possible position of an $M \times N$ input, the output size is $(M - m + 1) \times (N - n + 1)$. It is also possible to apply the filter in different ways

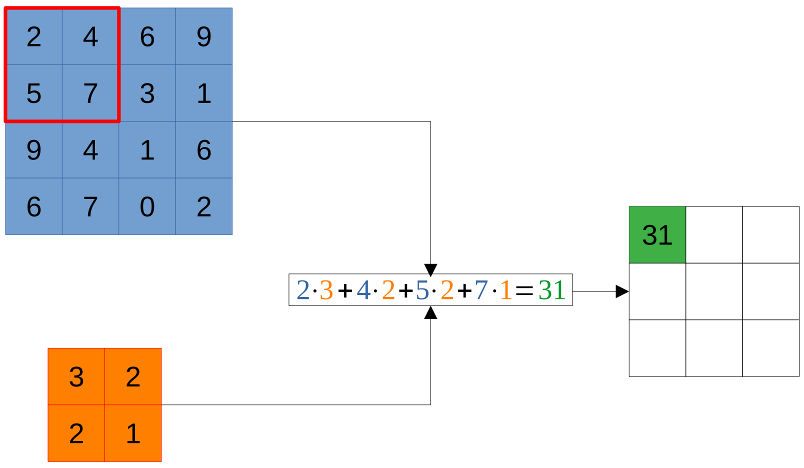


Figure 2.7: Example of the convolution operation on the top-left element, highlighted in red. Image modified from [64].

by using a padding, i.e. a certain number of elements that are placed at the boundaries of the input in order to increase the size of the output, and a stride, i.e. the number of positions of which the filter is translated at each step of the convolution. More details are provided by [8].

**Pooling** Pooling is used as down-sampling technique to reduce the dimension of an input. This allows to reduce the number of parameters, and hence the complexity of the model and the computational load, and has an effect on preventing overfitting. Four types of pooling operations are mostly used:

- Max pooling reduces a window to a single value, equal to the maximum value in the original window.
- Average pooling reduces a window to a single value, equal to the average value of the elements in the original window.
- Global max pooling reduces the whole input to a single value, equal to the maximum value in the input.
- Global average pooling reduces the whole input to a single value, equal to the average of all the values in the input.

Schemes of down-sampling operations are shown in figure 2.8.

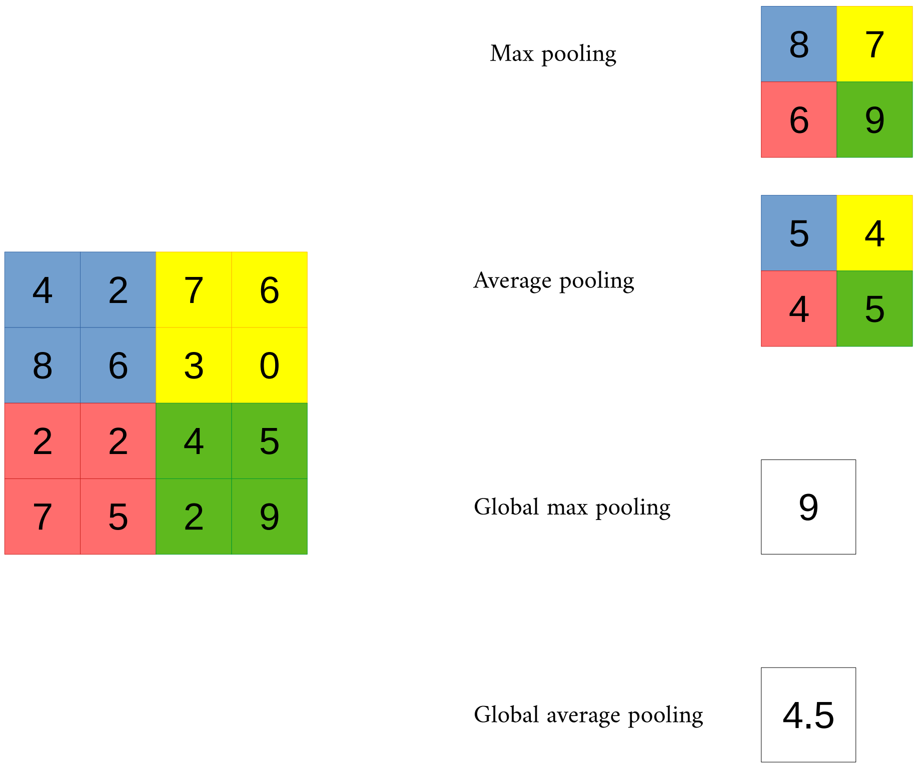


Figure 2.8: Down-sampling operations. Image modified from [64]

**Flattening** Flattening consists in converting a multi-dimensional tensor into a one-dimensional vector, that can be given as input to a traditional Neural Network. An example of flattening operation is reported in Figure 2.9.

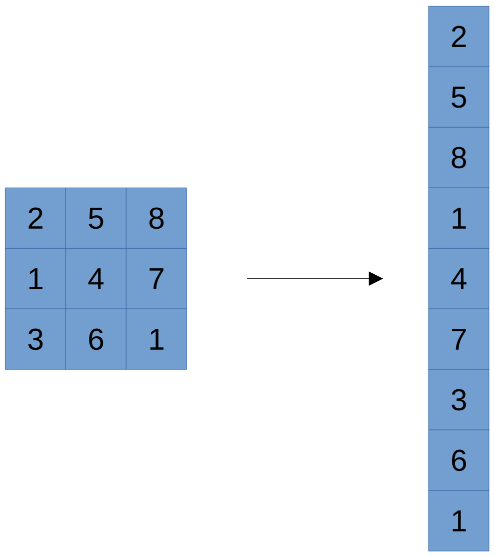


Figure 2.9: Two-dimensional flattening operation. A $3 \times 3$ matrix is reduced to a vector with 9 elements.

In a CNN, the operations previously seen are used in a sequential and hierarchical structure, forming a layered structure. In each layer, an operation is performed. An example of a full CNN is shown in Figure 2.10.

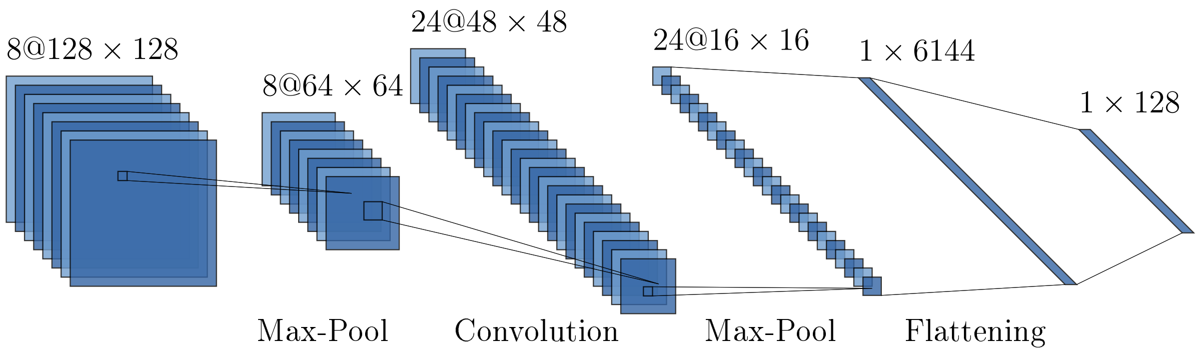


Figure 2.10: Example of a CNN architecture.

### 2.3.2. Feature based navigation with Convolutional Neural Networks

If CNNs are used, two approaches to the navigation problem are possible:

- The CNN is used to extract and match features, that are used as measurements for navigation. Those networks are trained to recognize specific landmarks [26, 76], such as craters, or features provided of descriptors [24, 32].

- The CNN is used to carry out the whole navigation process. It receives the images as input, and outputs the state of the spacecraft. Examples are Pugliatti and Topputo [66] and Pugliatti et al. [67].

## Crater detection using convolutional neural networks

If craters are present, they could be used as reference points for navigation. Neural networks have been developed for autonomous crater detection and classification.

LunaNet, presented in Downes et al. [26], is a convolutional neural network developed for lunar vision-based navigation applications. The input is the image, while the output is a grey scale image in which the pixel brightness indicate if a pixel is part of a crater rim or not. Tests show that the algorithm has better tracking persistence and robustness against illumination changes with respect to non CNN-based algorithms, specifically the Trinary Edge Detector [26]. Also the position reconstruction error using EKF is lower.

DeepMoon, presented in Silburt et al. [76], is a CNN that detects craters, receiving as input the digital elevation map (DEM) of the terrain. The size of the input is $256 \times 256$. The output is a $256 \times 256$ image, also referred as "target", that consist in a black background on which craters are highlighted in white. It demonstrates that transfer learning can be applied to crater detection because, while being trained on datasets regarding the Moon, it performs well also with other bodies. However, it is required to have a DEM as input.

LunaNet, DeepMoon and PyCDA[1] are compared, along with the Trinary Edge Detector, in Downes et al. [28] and Downes et al. [27]. The comparison takes into account different performance metrics.

In terms of computational time, LunaNet and the Trinary Edge Detector are the best solutions. While DeepMoon and PyCDA take an average of 4-5 seconds to detect a crater, LunaNet and the Trinary Edge Detector take less then a second (the results have been obtained on an NVIDIA K80 GPU). LunaNet has a lower localization error in terms of pixels, and also appears to be more robust against noise in the image and reduced brightness.

When position estimation is taken into account, Downes et al. [28] report that LunaNet and PyCDA are the only algorithms for which the estimation error is bounded. For the other algorithms, it tends to diverge.

The possibility of applying these techniques obviously depends on the presence of craters. Asteroids such as 101955 Bennu and 162173 Ryugu do not have craters, or at least craters that are particularly distinct from other surface features, and therefore crater-based navigation wouldn't likely work well. Other asteroid, such as 4 Vesta, present

[1]`https://github.com/AlliedToasters/PyCDA`

craters, and hence this navigation technique would be more suitable. A comparison of the textures is shown in Figure 2.11.

### Feature matching using convolutional neural networks

The algorithms cited above can be used to detect craters in the image. However, many small bodies do not have craters, or they are present only in a small number. Techniques that are more independent from morphological features are hence necessary. In addition to the classical feature extractors, also feature extractors that use CNNs are available.
SuperPoint [24] is a fully convolutional neural network with a double pipeline architecture. The input is an $H \times W$ image. First, a common encoder downsamples the image. After the image has been encoded, the CNN develops in two pipelines: the first pipeline is the feature point decoder. The output is an $H \times W$ matrix in which each element corresponds to the probability of a pixel in the input image to be an feature point. The second pipeline is the descriptor decoder: the output is an $H \times W \times D$ tensor, in which the depth dimension corresponds to the length of the descriptors of the feature points. The architecture is reported in Figure 2.12.

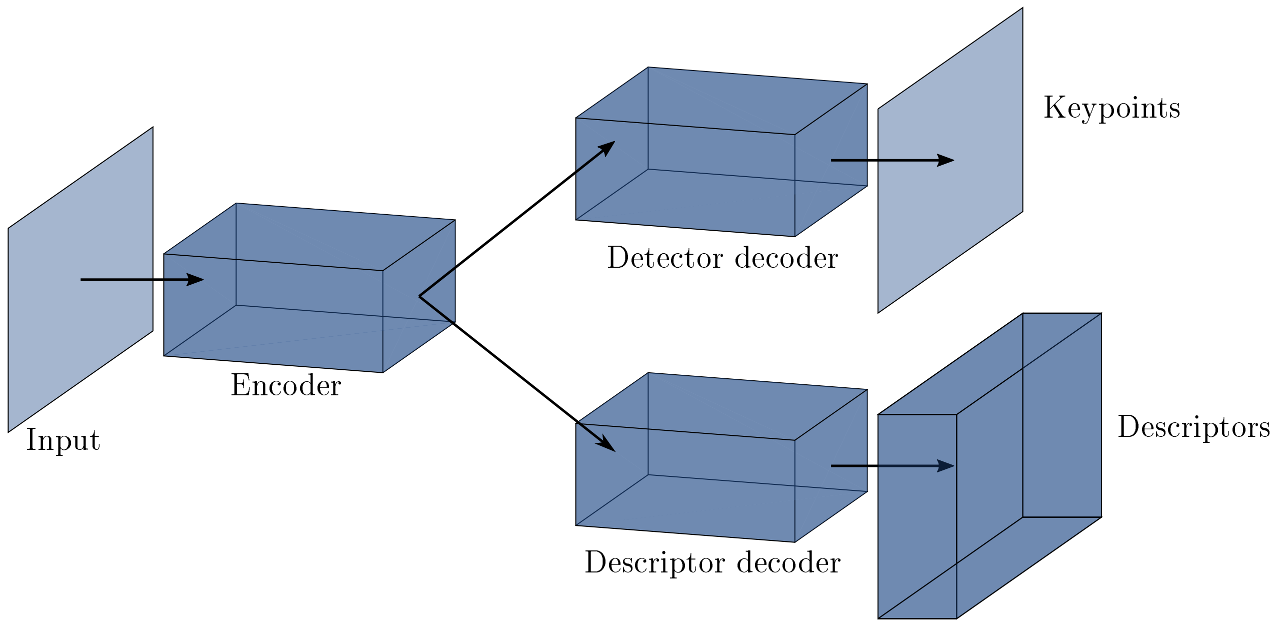
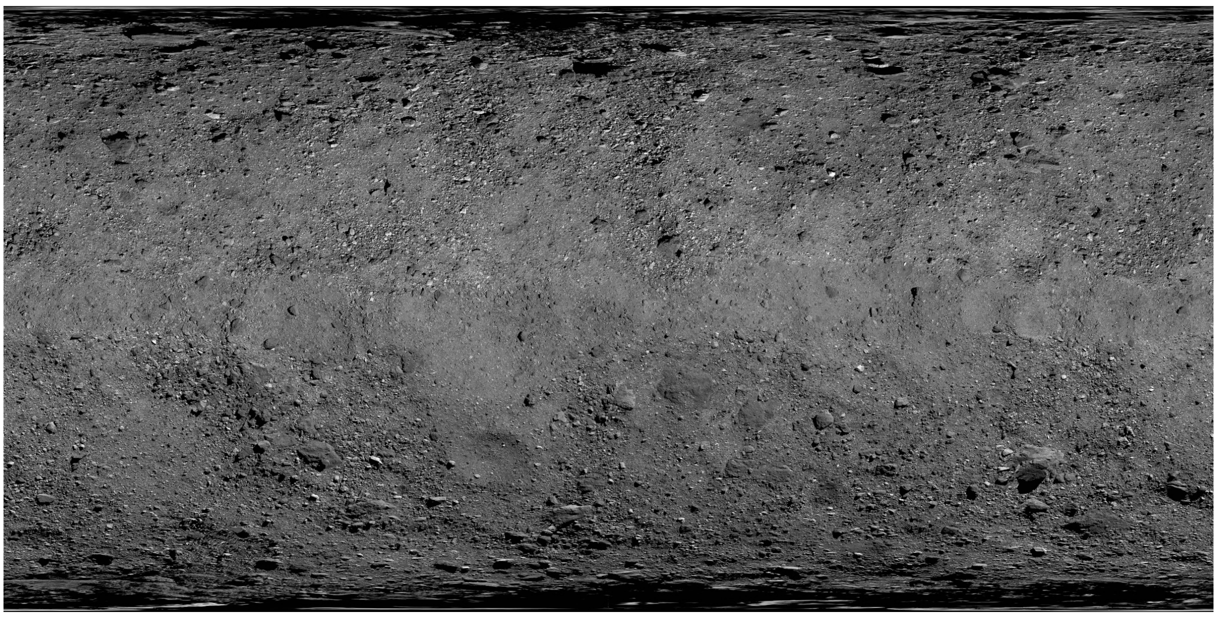
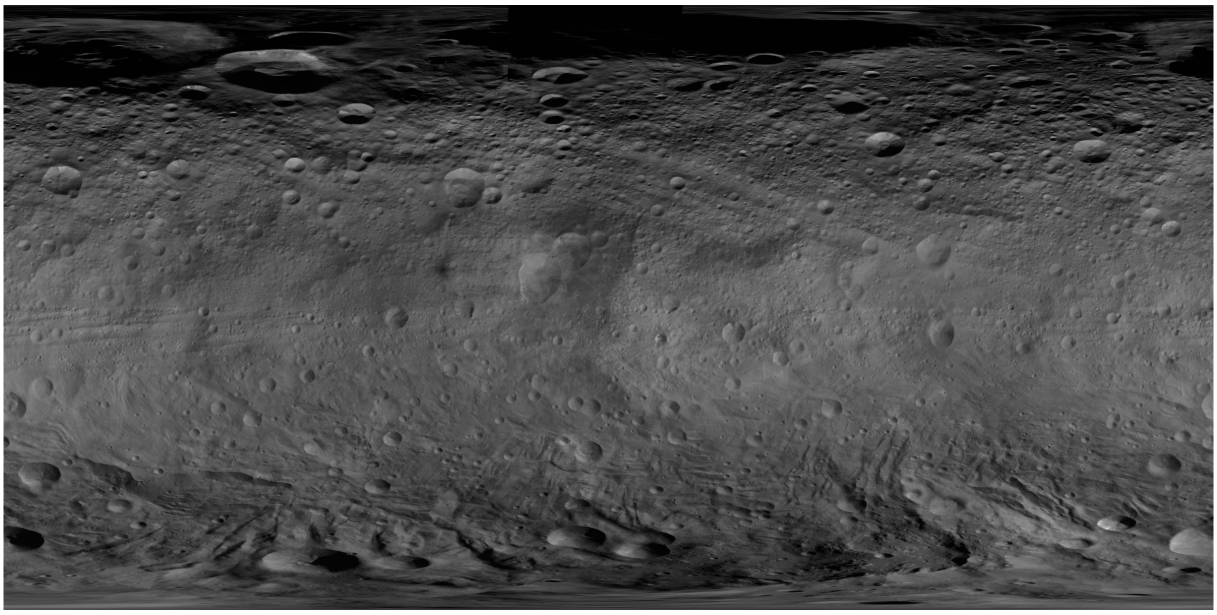


Figure 2.12: SuperPoint architecture.

D2-Net [32] is a fully convolutional neural network, with a single pipeline architecture, that simultaneously extracts feature locations and descriptors. Comparisons show that, if the accuracy required (in terms of pixels) is very strict, D2-Net performs worse than SuperPoint in terms of number of correct matches. However, if less accuracy is required,



(a) Texture of the asteroid 101955 Bennu (Bennett [14]).



(b) Texture of the asteroid 4 Vesta (Roatsch et al. [68]).

Figure 2.11: Comparison between the textures of the asteroids 101955 Bennu (Figure 2.11a) and 4 Vesta (Figure 2.11b). On the asteroid 4 Vesta, craters are clearly visible, and they could probably be used for optical navigation. On the contrary, on 101955 Bennu, craters are not clearly visible, or they blend with the rest of the texture.

D2-Net outperforms SuperPoint in terms of correct matches. Hence, the best choice depends on the matching accuracy required. If the input is a $H \times W$ image, the output is a $H \times W \times D$ tensor, which embeds both detector and descriptor. As in the case of SuperPoint, the output is dense.
LF-Net [59] is a convolutional neural network based on sparse matching, meaning that there is not a one-by-one pixel correspondence between image and detector/descriptor. For this reason, the computational burden should be lower than the previous methods. In terms of matching score, it is shown to outperform classic optical techniques, and even SuperPoint in most of the scenarios.
It is worth noting that these algorithms have been tested on generic data sets, but have not been used for space applications. Hence, the behaviour might be different in the case of textures typical of small bodies. One of the main scopes of this work is to investigate how these networks perform for such applications.

## Navigation using convolutional neural networks

As stated above, it is possible to push further the use of CNNs, performing the whole navigation task. In this case, the CNN receives as input the images taken by the navigation camera, and gives as output the position of the spacecraft.
In the work carried out by Linares et al. [46], the position of a spacecraft is estimated starting from the simulated image of a navigation camera, extracted as a subset of a larger scenario. The scenario consisted in a $1024 \times 1024$ nadir pointed image of the Apollo 16 landing site. The training set was obtained by sampling $128 \times 128$ images from the scenario, simulating different illumination conditions. The input of the CNN is hence a $128 \times 128$ image. During the simulation, motion is restricted in one dimension. Since the whole scenario is $1024 \times 1024$, and being the motion one-dimensional, there are 1024 possible classes. Results show that the CNN is able to predict the position within an error of a few pixels.
A more complex work is carried out by Pugliatti and Topputo [66], a CNN is used to localize the spacecraft around a small body. Since the CNN naturally manages classification problems, the spherical space around the small body is discretized, and each portion of space consists in a class. There are a total of 1176 classes, and they are defined in polar coordinates. The CNN does not receive directly the output images coming from the navigation camera, but it works using segmentation maps, that are less dependent on illumination conditions. Once the area in which the spacecraft is located is roughly determined, the position estimation is refined using NCC.
The task can be pushed even further by integrating also control laws. In the work by

Furfaro et al. [33], a CNN is developed in order to achieve optimal control of the thrusters in a lunar landing scenario. The network takes a sequence of three images as input. Two cases are considered: a vertical lunar landing, and a 2D planar landing. In the vertical landing, only the component relative to the altitude changes. In this case, the network used is made by various convolutional layers, followed by fully connected layers, and the output consists in two classes, that indicate if the thruster is ON or OFF. In the planar case, the architecture is more complex: after the first part, that is convolutional, a fully connected layer flattens the output, and a LSTM (Long-Short Term Memory) network is present. This type of network contains feedback connections, and is used to process sequences of data. After the LSTM, the network develops into two pipelines: one computes the magnitude of the thrust, while the other one computes the thrust angle.

## 2.4. Position determination

Once the features have been extracted, they can be used to estimate the position of the spacecraft. In robotics, the problem of performing navigation in an unknown environment is labeled SLAM (Simultaneous Localization And Mapping). SLAM regards the simultaneous determination of the external environment (mapping) and the position from which the environment is observed (localization). Dylan et al. [29] describe an approach to the solution of the SLAM problem for a spacecraft equipped with a monocular camera. Note that a monocular camera does not allow to resolve the scale of the problem: to do that, a stereo camera or additional sensors are needed.
Initially, algorithm has to be initialized. This consists in starting from two sequential images, detecting and matching the features, and reconstructing the three dimensional position of these points through the triangulation procedure (Hartley and Sturm [41]). Two cameras with the same intrinsic matrix that observe the same point are linked by the so-called Essential matrix, indicated with **E**. This matrix can be estimated, if at least eight point matches are known, using the Eight Point Algorithm (Longuet-Higgins [48]). With the singular value decomposition of the essential matrix, it is possible to extract the translation (up to a scale factor) and rotation of the camera between the two views.
Panicucci [61] and Panicucci et al. [62] propose a SLAM solution based on SURF features tracked by the KLT tracker, that includes the probe localization, the shape reconstruction of the body, the characterization of the rotational dynamics of the body and the definition of the body fixed reference frame. The work also proposes a strategy for loop closure.
In the work carried out by Cocaud and Kubota [23], Monte Carlo methods are applied to find candidates for possible camera motions. Dor et al. [25] propose a SLAM solution

based on ORB features.

## 2.5. Research question

During the literature study, increasing interest emerged in the application of feature extraction and tracking techniques, particularly based on artificial intelligence, in the space field. The present work is based on the following research question:

*How do feature extraction and tracking techniques based on machine learning perform, in terms of accuracy and robustness, compared to traditional techniques, for the purpose of navigating around a small body?*

In order to answer to the main research question, the following sub-questions are formulated:

1. *How do classical optical techniques perform in the scenario of a small body?*
2. *How do artificial intelligence techniques perform in the scenario of a small body?*
3. *Which category of techniques has the better performance, in terms of the characteristics desired to solve a localization problem?*

# 3 | Creation of the models and generation of the images

## 3.1. Creation of the model

The asteroid models used in this work are essentially composed by three main elements: a shape model, a texture and a BRDF (Bidirectional Reflectance Distribution Function), that determines the response of the model to illumination.
A shape model is a file that contains information regarding the vertices and the faces composing the 3D mesh. Shape models are generally produced using data obtained by radar observations, images or lidar measurements. Shape models generated from images or lidar data are more accurate, but available for less bodies, since a mission in proximity of the body would be necessary.
Shape models of various bodies, obtained with different techniques and of different resolutions, can be obtained using the Small Body Mapping Tool[1] (SBMT). A shape model of the asteroid 101955 Bennu, based on lidar measurements, is shown in Figure 3.1. Textures are mosaics obtained by merging images of the body surface. An example of texture of the asteroid 101955 Bennu is shown in Figure 3.2. The shape models and the textures, along with their source, used to generate the models of the asteroids are reported in Table 3.1.
Since a texture is essentially the surface of the asteroid unwrapped in two dimensions, certain areas will present inaccuracies due to projection methods. This is evident in the top and bottom areas of the texture, where black regions are present. As a consequence,

[1] `https://sbmt.jhuapl.edu/`

Table 3.1: Shape models and textures used for the asteroid models.

| Asteroid | Shape model | Texture |
|---|---|---|
| Bennu | SPC v42 [2] | Bennett [14] |
| Itokawa | Gaskell et al. [36] (.obj from SBMT) | Stooke [78] |
| Eros | Gaskell [35] (.obj from SBMT) | Stooke [78] |

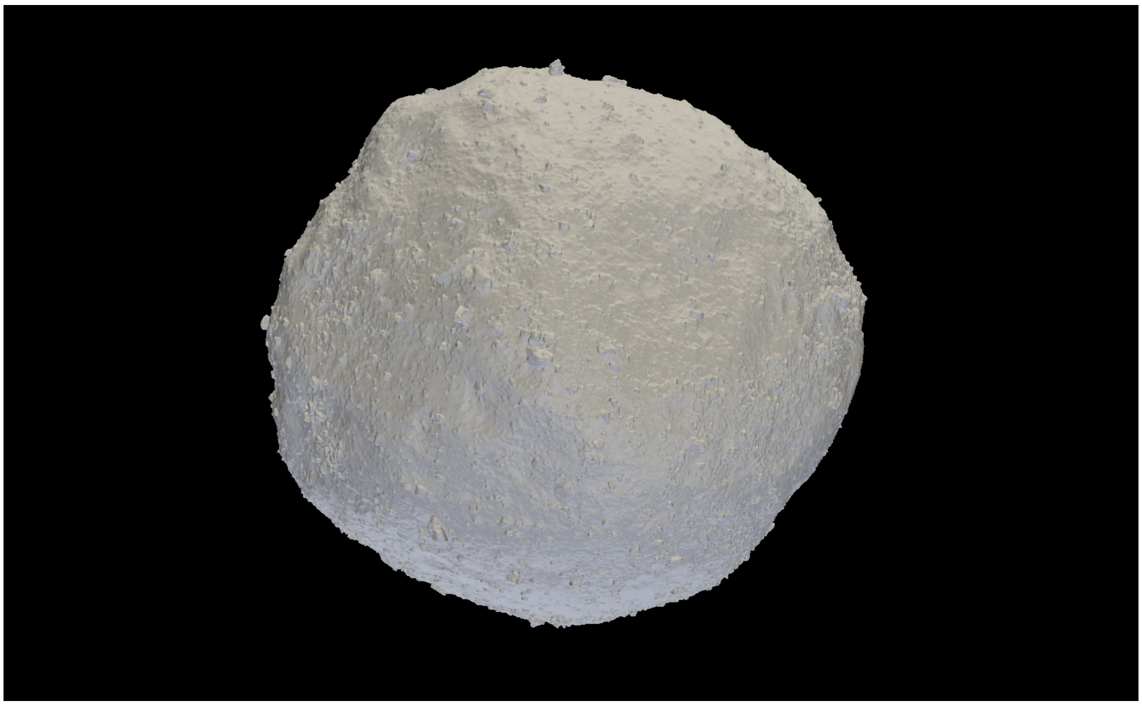


Figure 3.1: Shape model of the asteroid 101955 Bennu, based on lidar measurements. Source: PDS SBN (Shape model SPC v42 [2])

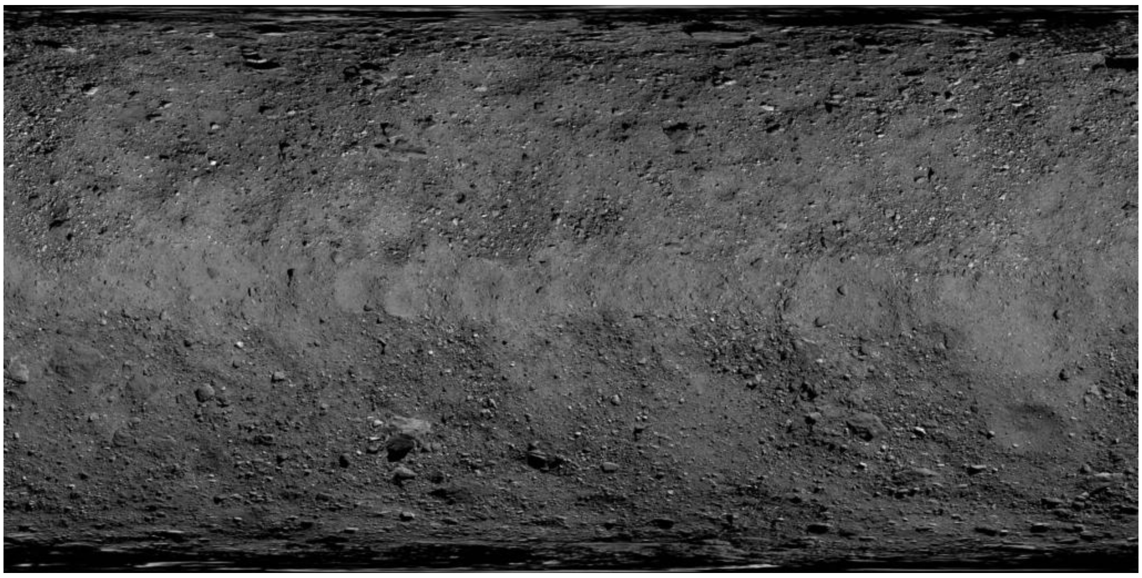


Figure 3.2: Texture of the asteroid 101955 Bennu. Source: Bennett [14].

the model will not reproduce accurately these areas of the body. This has to be considered when the images are rendered.
Finally, the BRDF of the model has to be adjusted in order to have a realistic response to light. This can be done by visually comparing rendered and real images of the body, and determining when the response to illumination appears more accurate. Section 3.4 provides details about the validation process, which allows to determine if the response to light is accurate or not.
It should be considered that some factors prevent from obtaining extremely realistic models, in particular:

- High resolution textures are available only for bodies targeted by recent missions, such as Bennu. The textures available for bodies targeted by older missions, such as Eros, are less detailed.
- In this work, the BRDF is assumed to be constant on the entire model for performance reasons. However, this might be not realistic, since photometric properties are likely not constant on the entire surface. For instance, according to Golish et al. [38], in the case of Bennu the heterogeneous surface presents variations in reflectance, and this makes photometric modeling difficult. As a consequence, the response to light of the model might differ from reality.
- A lower limit on the distance from the model at which images can be rendered is posed either by the texture, which however accurate has a limited resolution, or by the shape model, which however detailed consists in a discrete mesh whose polygons become visible at close range.

## 3.2. Rendering principles

According to Marschner and Shirley [53], rendering is a process that takes as input a set of objects and produces as output an array of pixels. Fundamentally, the process consists in determining in which way each object influences each pixel, and computing the value of that pixel.
This procedure is carried out by an algorithm named ray tracer. It consists, conceptually, in three steps:

- The viewing rays are computed for each pixel. The viewing rays determine the objects that influence the appearance (i.e. the color) of each pixel.
- The closest object to the camera intersecting the viewing ray is determined.

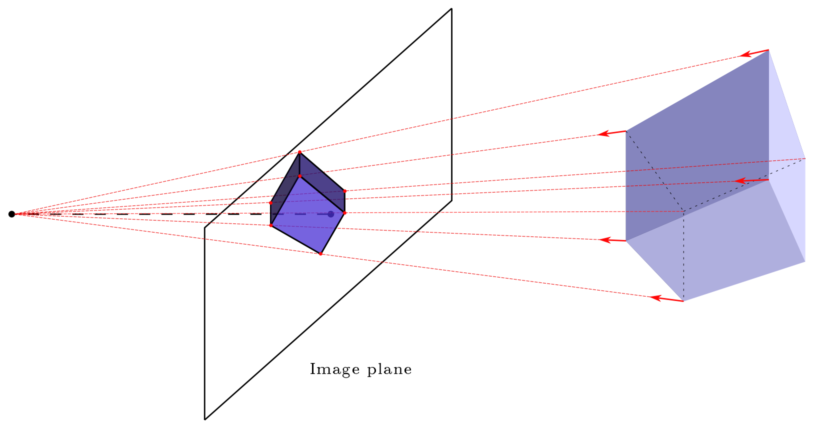


Figure 3.3: Scheme of the ray tracing procedure. Rays are traced from the elements in the scene towards the camera, determining which pixels are influenced by the objects. The values of the pixels are computed by means of the shader model.

- The pixel value is computed using a shading model, that determines the color of the pixels on the base of the points hit by the viewing rays.

The first two steps involve the procedure of ray casting. The procedure, described in detail in Section 4.3, consists in determining the pixels influenced by the elements in the scene by tracing rays from the objects towards the camera, and verifying in which pixels the intersections are. In these steps, the camera model plays a major role, since it directly influences how the pixels are influenced by the scene. In Blender, a pinhole camera model, presented in Section 2.1, is used. A scheme of the procedure is shown in Figure 3.3. The focus goes there on the third step. The process of shading consists in determining the value of a pixel, based on the amount of light reflected by the objects that reaches the pixel. For the purpose, a shading model is used. A shading model is a model designed to reproduce light reflection.
The simplest shading model is the Lambertian model [53]. The basic assumption is that the amount of energy received by a surface depends on the angle between the normal to the surface $\boldsymbol{n}$ and the illumination direction $\boldsymbol{l}$:

$$L = k_d I \cdot max(0, \boldsymbol{n} \cdot \boldsymbol{l}) \tag{3.1}$$

Where $L$ is the pixel color, $k_d$ is the diffuse coefficient, I the intensity of the light source, $\boldsymbol{n}$ the surface normal and $\boldsymbol{l}$ the light direction.
The Lambertian model is simple, but not accurate in the case of real surfaces: for most of real surfaces, in fact, the response will depend not only from the illumination direction,

but also from the viewpoint. This dependence is introduced in the Blinn-Phong shading model [53], described by the following equation:

$$\boldsymbol{h} = \frac{\boldsymbol{v} + \boldsymbol{l}}{||\boldsymbol{v} + \boldsymbol{l}||} \tag{3.2}$$

$$L = k_d I \cdot max(0, \boldsymbol{n} \cdot \boldsymbol{l}) + k_s I \cdot max(0, \boldsymbol{n} \cdot \boldsymbol{h}) \tag{3.3}$$

Where the vector $\boldsymbol{h}$ determines the point of view and $\boldsymbol{n}$ is the normal to the surface.
According to Marschner and Shirley [53], more realistic and physically based renders can be obtained with the BRDF model (Bidirectional Reflectance Distribution Function). The BRDF is defined as the ratio between the radiance measured in an outgoing direction $\boldsymbol{k}_o$, and the irradiance coming from a direction $\boldsymbol{k}_i$. The BRDF is expressed as function of these two directions;

$$f_r(\boldsymbol{k}_o, \boldsymbol{k}_i) = \frac{L_s(\boldsymbol{k}_o)}{H(\boldsymbol{k}_i)} \tag{3.4}$$

Real-world BRDF models are extremely complex. However, it is possible to build empirically BRDF models for different classes of materials. Some of these models are presented in Marschner and Shirley [53].
Analogously to BRDF, the concept of BTDF (Bidirectional Transmittance Distribution Function), described by Asmail [11], accounts for the light that is transmitted through the surface.
Blender uses a shading model called Principled BSDF (Bidirectional Scattering Distribution Function). BSDF is a generalization of the concepts of BRDF and BTDF, that accounts for both the reflected and transmitted radiation.
The Principled BSDF is developed on the base of the BSDF model proposed by Burley [18]. BSDF has been developed as a physical based framework (i.e. that respects energy conservation) that accounts account for all the reflection, transmission and scattering effects.
In Blender two rendering engines are implemented: Cycles and EEVEE. An overview is provided by Koteswara Rao et al. [43]. Cycles is a physically based rendering engine based on ray tracing. EEVEE is a real-time rendering engine, meaning that the focus is less on the quality of the renders and more on the speed. It uses a technique called rasterization, that consists in determining all the pixels of an image that are occupied by an object in the scene. The color of the pixels is then computed with specific algorithms. Cycles tends to produce more realistic results than EEVEE, but the rendering is more computationally expensive.

## 3.3. Generation of the images

All the images were rendered using a pinhole camera with resolution $1600\times1200$ pixels and an horizontal angular field of view (HAFOV) of 44 deg. The resolution is arbitrary, while the HAFOV is the same of the OSIRIS-REx Touch And Go Camera System (TAGCAMS), according to Bos et al. [16].
In order to simulate the operation of a camera mounted on a spacecraft, images of the model have to be rendered from the viewpoint of the spacecraft that moves on different trajectories and with different attitudes. The sets of images are produced by changing parameters whose effect has been considered important to investigate. In particular, different sets of images differ in:

- Distance from the body.
- Positions of the camera around the body. In order to add more variety to the sets, the camera moves along trajectories defined by different orbital inclinations and that approach the asteroid at different rates.
- Illumination conditions, in terms of sun direction and light intensity.

From a practical standpoint the sequences of positions and orientations of the camera and the directions of the light (expressed through the sub-solar point) are generated and stored in a file, that is referred as "trajectory" in a broad sense.
The sequence of positions is obtained by superimposing a motion around a circular orbit with a given inclination and a motion towards the center of the body. The result is a spiral-like trajectory.
In its motion around the sequence of positions, the camera is constrained to point towards the asteroids. This assumption was introduced to avoid to compute, at each step, a camera inclination that was coherent with the position, the inclination history and pointing the asteroid.
For each camera position and orientation, a picture of the asteroid is rendered with different values of illumination. In this work, two light intensities of $1\frac{W}{m^2}$ and $5\frac{W}{m^2}$ have been used.
The number of images generated by the procedure is:

$$n_{i,tot} = n_t \cdot n_l \cdot n_{i,set} \tag{3.5}$$

Where $n_{i,tot}$ is the total number of images that are generated, $n_t$ is the number of trajectories, $n_l$ is the number of light intensities used, and $n_{i,set}$ is the number of images

Table 3.2: Initial values from which the trajectories have been generated.

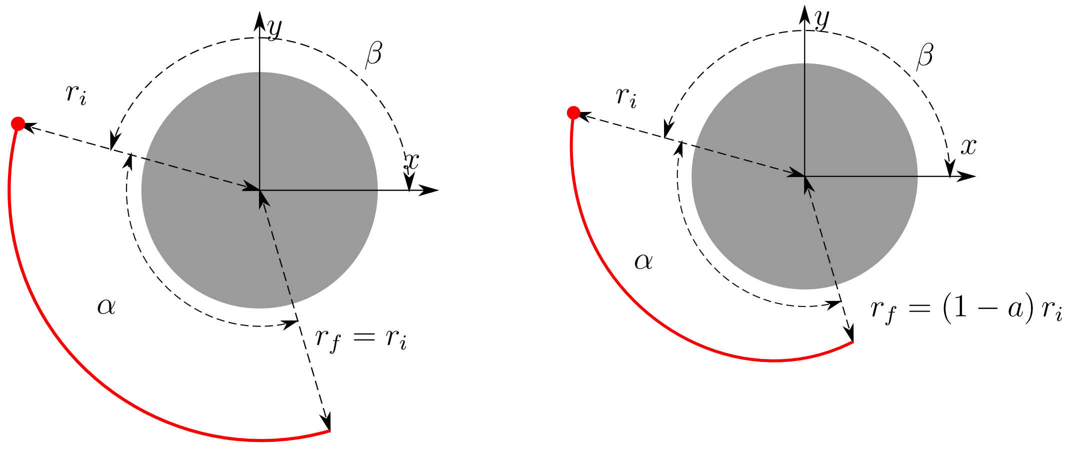
| Asteroid | $r_i$ $[Km]$ | $\alpha\,[rad]$ | i $[deg]$ | $\beta\,[deg]$ | I $\frac{W}{m^2}$ | $a[-]$ |
|---|---|---|---|---|---|---|
| Bennu | $1; 2$ | $\frac{\pi}{6}; \frac{\pi}{2}$ | $0; 40$ | $0; \pi$ | $1.0; 5.0$ | $0; 0.2$ |
| Itokawa | $0.7; 1.5$ | $\frac{\pi}{6}; \frac{\pi}{2}$ | $0; 40$ | $0; \pi$ | $1.0; 5.0$ | $0; 0.2$ |
| Eros | $50; 70$ | $\frac{\pi}{6}; \frac{\pi}{2}$ | $0; 40$ | $0; \pi$ | $1.0; 5.0$ | $0; 0.2$ |

that are rendered for a trajectory with a selected light intensity.
The distance from the asteroid and the angle covered by the movement of the camera, together with the number of images acquired, have been determined in order to achieve an acquisition frequency of approximately one image per minute. This frequency was arbitrary selected to match both the necessity of having sufficiently similar images (especially for satisfying the hypothesis behind feature tracking algorithms, as explained in Chapter 4.1) and a sufficiently large camera movement to acquire different regions of the asteroid. Anyway, a less frequent image acquisition can be simulated by processing images less frequently: for example, instead of processing sequential images, an image can be skipped at each step.
Table 3.2 reports the initial values of the variables used to generate the trajectories: the (initial) orbital radius $r$, the angle $\alpha$ that defines the width of the arc on which the camera moves during a sequence of acquisitions, the orbital inclination $i$, the angle $\beta$ between the $x$ axis and the beginning of the trajectory, the light intensity $I$ and the approach rate $a$, that defines the fraction of radius of which the camera approaches the asteroid during an acquisition. Figure 3.4 clarifies the nomenclature used.

At each combination of position, orientation of the camera, orientation of the asteroid and direction of the sun, the angle between the camera and the sun is computed. If this angle is too large the image is not rendered because, as shown in Figure 3.5, the asteroid would be between the sun and the camera, and the image would be almost completely black.

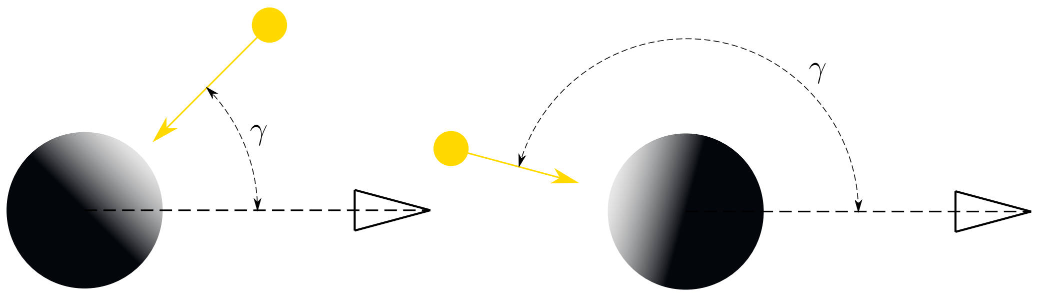
## 3.4. Validation of the results

The model obtained can be validated by comparing it to actual images taken by spacecrafts. Images and other scientific data collected by past missions to small bodies are available in the NASA PDS Small Body Node (SBN). Each image is coupled to an header that contains information such as the specific time at which the picture was taken and the instrument that was used.
The images obtained from the NASA PDS SBN can be compared to the images of the



(a) $a = 0$ (b) $a > 0$

Figure 3.4: Scheme of the nomenclature used in Table 3.2. The trajectories starts in the red dot at a distance $r_i$ from the origin and an angle $\beta$ from the $x$ axis, and ends after an arc of amplitude $\alpha$ at a distance of $r_f = (1-a)\,r_i$ from the origin. Figure 3.4a shows the case in which $a = 0$, in which the trajectory is an arc of a circle, while 3.4b shows the case in which $a > 0$, in which the trajectory is an arc of a spiral.



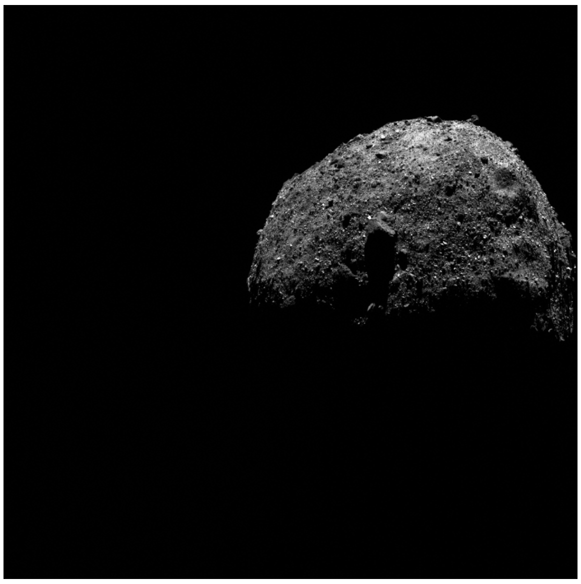
(a) Small $\gamma$ angle. (b) Large $\gamma$ angle.

Figure 3.5: Figure 3.5a represents the case in which the angle $\gamma$ between the sun and the camera is small: in this case, the region of the asteroid exposed to the camera is in light, hence the image is rendered. In Figure 3.5b, the angle $\gamma$ is large, asteroid is between the sun and the camera, and the region of the asteroid exposed to the camera is in shadow. The image would be almost completely black, hence it is not rendered.

model generated, taken reproducing the same conditions of the original one. These conditions can be easily obtained using the SPICE system, that allows to obtain information on the state in which a spacecraft was at a specific time. In particular, the information needed to reproduce the original conditions of the spacecraft are:

- The position of the spacecraft in the inertial reference frame of the body.
- The orientation of the instrument (in this specific case, a camera) with respect to the inertial reference frame of the body.
- The illumination conditions. In this case, various approaches can be used. In this work, the sub-solar point has been calculated, and the sun has been forced to point the body through the sub-solar point.
- The resolution and the field of view of the instrument.

These data are inserted in Blender, and the result is compared to the original image. Some results are shown below. The original images are on the left, while the rendered images on the right.


(a) Original picture.

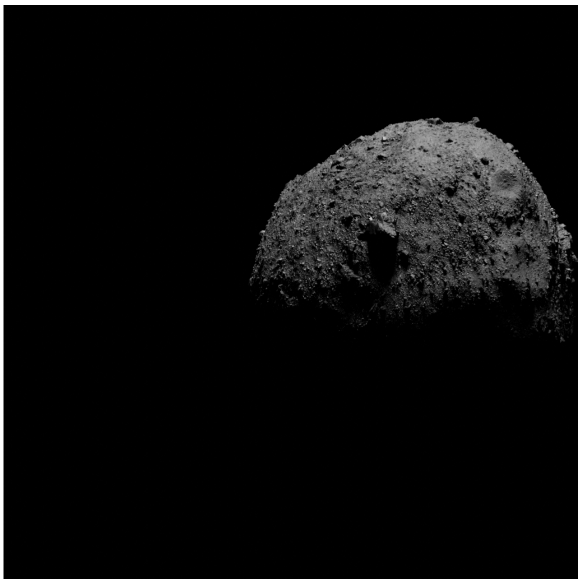
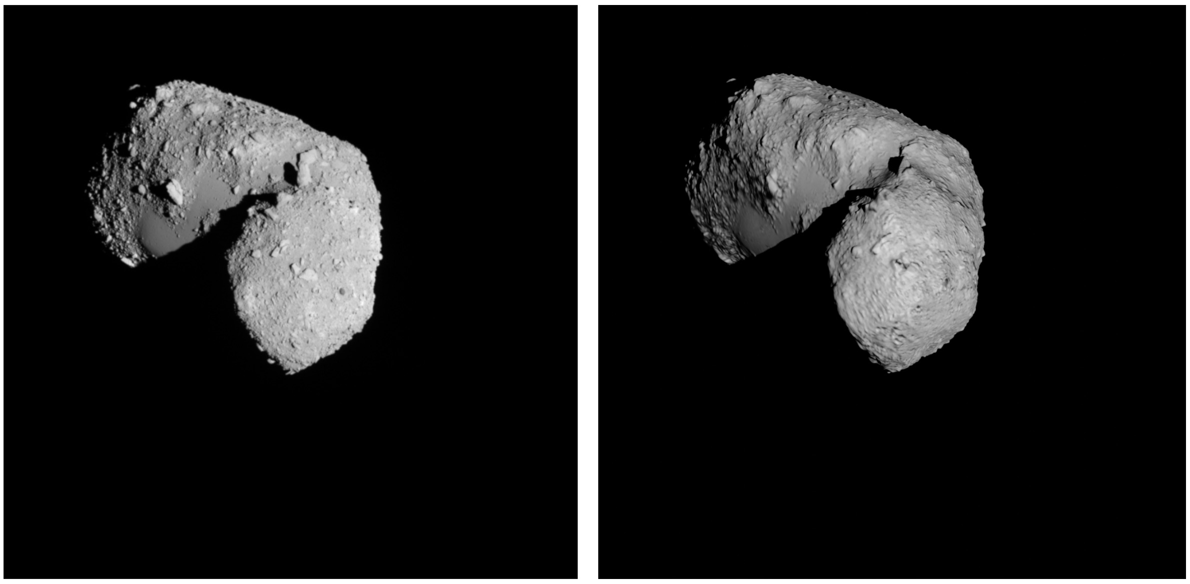
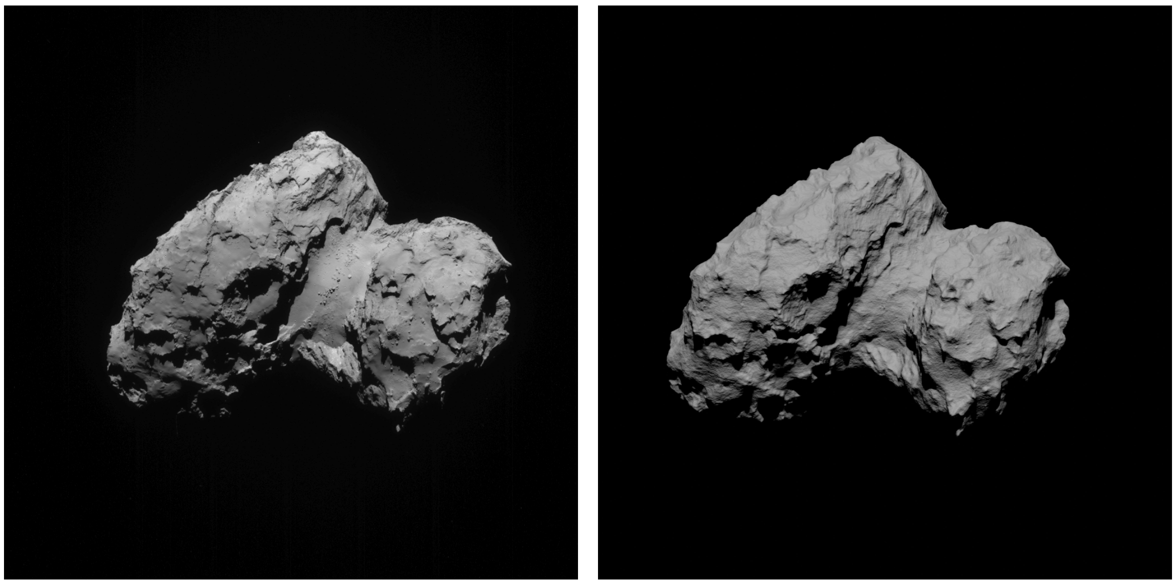

(b) Rendered picture.

Figure 3.6: Comparison between the picture taken by the OSIRIS-REx MAPCAM to the asteroid Bennu on the 16/12/2018 at 03:27:55 (left) and the model of Bennu (right).



(a) Original picture. (b) Rendered picture.

Figure 3.7: Comparison between the picture taken by the Hayabusa AMICA instrument of the asteroid 25143 Itokawa on the 18/10/2005 at 16:23:49 (left) and the model of 25143 Itokawa (right).



(a) Original picture. (b) Rendered picture.

Figure 3.8: Comparison between the picture taken by the Rosetta NAVCAM of the comet 67P/Churyumov-Gerasimenko on the 19/08/2014 at 19:07:18 (left) and the model of 67P (right).

## 3.5. Datasets

Test sets have been produced for each asteroid. All the algorithms and the CNN models were tested on the same test sets. In addition, train and validation sets have been produced for the CNN.
The test sets are composed by images generated sequentially, with the procedure explained in Section 3.3. The test sets composition are reported in Table 3.3.

Table 3.3: Test sets composition.

| Set name | Number of images |
|---|---|
| Bennu | 9715 |
| Itokawa | 9451 |
| Eros | 5903 |

The number of images is different for each asteroid since, due to the different shapes, some images required to be eliminated manually due to bad illumination conditions. The Eros dataset has far less images due to rendering issues that occurred for unknown reasons.
The training and validation sets used for SuperPoint were produced by rendering images from random camera positions, without following pre-defined trajectories. Due to time reasons, only training sets of Bennu and Eros have been produced, and these sets have a different number of images, 10000 for Bennu and 5000 for Eros. The train-validation split is reported in Table 3.4.

Table 3.4: Train and validation sets composition.

| Set name | Test size | Validation size |
|---|---|---|
| Bennu | 8976 | 1024 |
| Itokawa | 4488 | 512 |

# 4 | Methodology

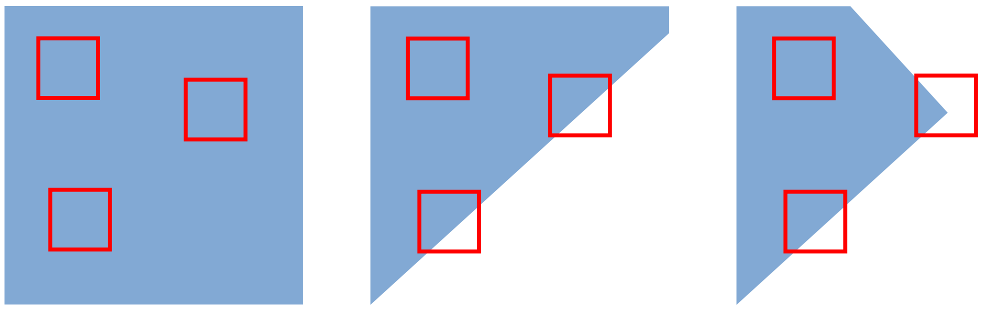
## 4.1. Features extraction and tracking with traditional techniques

As explained in Section 2.2, optical navigation can be performed by using features as reference points to determine the state of the spacecraft. Features are points on the image that are easily recognizable in case of variation of the viewpoint or on the environmental conditions, such as the light intensity and direction.
The concept of "easily recognizable" is quite arbitrary. However, some regions in an image certainly do not fit the definition. For example, as shown in Figure 4.1, points collocated in low contrast regions are certainly hard to precisely recognize in case of variations. In the same way, it might be difficult to recognize points located on an edge, since the contrast is strong only in one direction. On the other hand, regions which have a strong contrast in two perpendicular directions, i.e. corners, are certainly more distinctive than uniform regions or edges. As a consequence, many features extractors specifically search for regions that contain a corner, and hence are named corner detectors. Once the features are extracted, their recognition between different images can be carried out with techniques based on matching them according to their description, as in feature matching techniques, or following them, as in feature tracking techniques. Finally, outliers rejection techniques are used to reject false matches.
An overview of the features extractors, descriptor and tracking algorithms that have been used is there provided.

### 4.1.1. Algorithms overview

#### Features extractors and descriptors

Here is presented an overview of the traditional techniques that have been used.



(a) Flat region. (b) Edge. (c) Corner.

Figure 4.1: Comparison between a flat region, an edge and a corner. The red boxes are observation windows. In a flat region (Figure 4.1a), if the observation window is shifted, there are no distinctive characteristic that are recognizable. In an edge (Figure 4.1b), there is no ambiguity in the direction perpendicular to the edge, but if the window is shifted along the edge all the points are similar. In a corner (Figure 4.1c), there is no ambiguity: the corner is easily recognizable, and if the window is shifted in any direction no similar points are present in the immediate neighborhood. Hence, a corner is considered a good feature.

**SIFT** SIFT, detailed in Lowe [50], is an algorithm that extracts features that are invariant to scale and rotation. Invariance to scale is achieved by using a staged filtering approach.
The edges in an image can be highlighted with the convolution[1] of the image with a Laplacian filter. According to Haralock and Shapiro [39], a possible discretization of a Laplacian filter is:

$$\boldsymbol{L} = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \tag{4.1}$$

However, this filter is extremely sensitive to noise, and hence of little application in real world images. This sensitivity can be reduced by smoothing the window with a Gaussian filter, and then applying the Laplacian filter. The two steps are put together with the Laplacian of Gaussian (LoG). The analytical expression of the LoG filter is:

$$LoG(x, y) = -\frac{1}{2\pi\sigma^4}\left[2 - \frac{x^2 + y^2}{\sigma^2}\right] e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.2}$$

[1]`https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm`

The LoG filter can be implemented with a discrete approximation such as the following:

$$\boldsymbol{LoG} = \begin{bmatrix} 2 & -1 & 2 \\ -1 & -4 & -1 \\ 2 & -1 & 2 \end{bmatrix} \tag{4.3}$$

In SIFT, the Laplacian of Gaussian is replaced with the Difference of Gaussian (DoG), that consists in computing the difference between images repeatedly blurred with Gaussian filters. According to Lindeberg [47], the DoG is an approximation of the LoG. The DoG is computed starting from the scale space of the image. The scale space is built by progressively blurring the image with Gaussian filters. Let $L(x, y, \sigma)$ be the scale space representation of the image:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{4.4}$$

Where $I(x, y)$ is the image and $G(x, y, \sigma)$ is a Gaussian filter, whose expression is:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \tag{4.5}$$

The scale space representation is computed with equation 4.4, for different values of the scale parameter $\sigma$. Each scale of the image is characterized by a value of $\sigma$. The operator $*$ indicates the convolution operation.
In SIFT, the scale space representation is created for the original image, and for different sizes of the original image. In other words, after a certain number of filters have been applied, the image is resized, and the scale space representation of the resized image is computed again according to equation 4.4. The scale space representation for an image at a particular size is called octave. The sequence of octaves at different image sizes is called image pyramid.
Once the scale space representations are available, the Differences of Gaussians (DoG) are computed as difference between two consecutive scales. Let $D(x, y, \sigma)$ be the DoG at a scale $\sigma$. The expression of $D(x, y, \sigma)$ is:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{4.6}$$

Where $\sigma$ and $k\sigma$ are two consecutive scales, separated by a constant multiplicative factor $k$.
The procedure consists in building the scale space at different sizes and computing the Difference of Gaussian is shown in Figure 4.2. Potential features are determined by

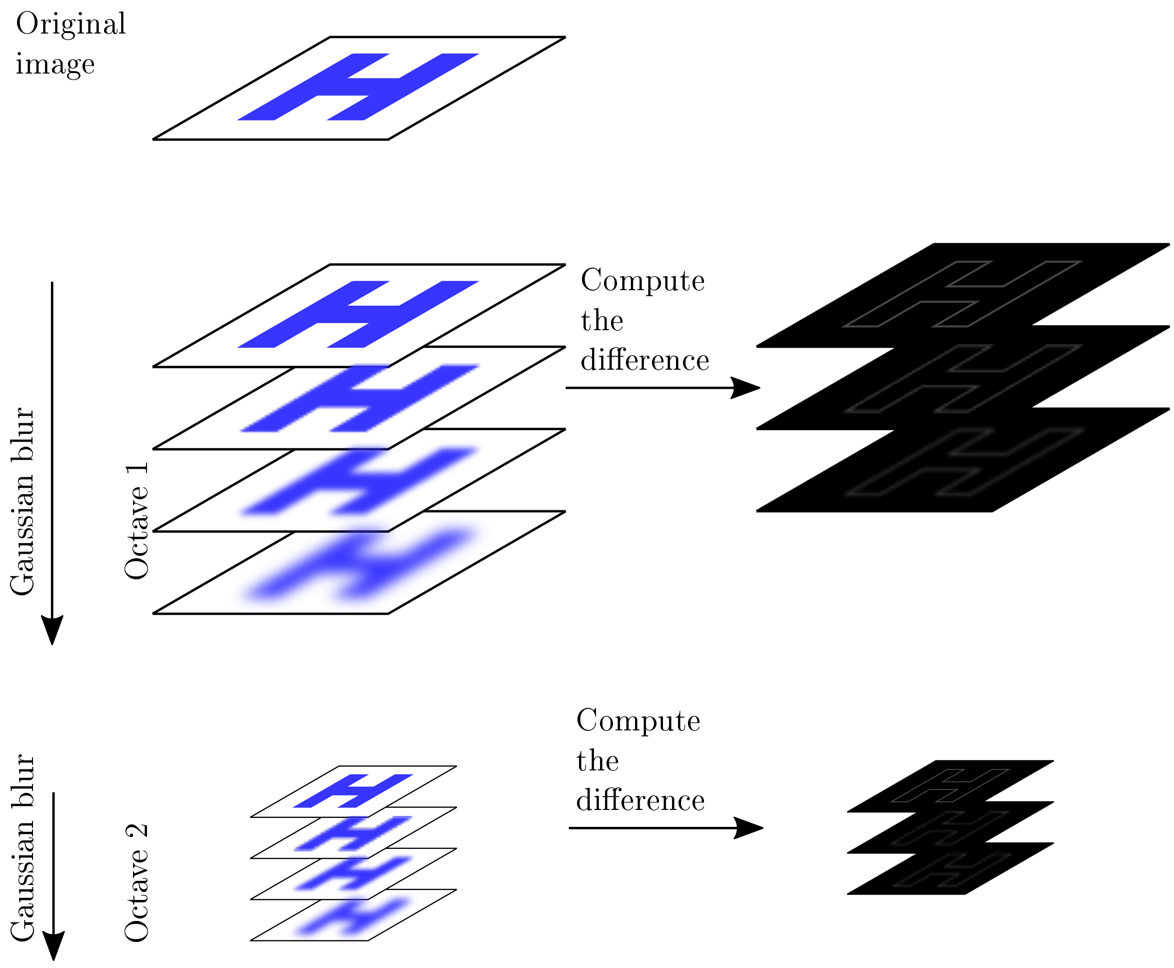


Figure 4.2: Illustration of the procedure of computation of the scale space and the DoG. The original image is progressively blurred and downsampled. The difference between sequential scales is computed. The result is a series of images in which the edges are visible. Image modified from [4].

searching for local extrema in the DoG. Local extrema are found by comparing the values of sample points with the neighbor points in both scale and location on the image.
Once local extrema have been identified, it is necessary to reject those points that are poorly recognizable. Ideally, a corner is a good feature, because it has high contrast in two directions. Points that have a low contrast in both the directions, i.e. flat regions, and those points that are poorly localized along an edge, with high contrast only in one direction, are not considered good features. In fact, those points are poorly recognizable. Low contrast points are removed using a Taylor expansion of the DoG, and excluding the regions whose contrast is below a certain threshold.
Edges are rejected using the Hessian matrix. The Hessian matrix **H**, at the location and scale of the feature, is computed:

$$\boldsymbol{H} = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{yx} & D_{yy} \end{bmatrix} \tag{4.7}$$

The derivatives $D_{xx}, D_{xy}, D_{yx}$ and $D_{yy}$ are estimated using finite differences, at the location and scale of the feature. The feature is considered valid if:

$$\frac{Tr(\boldsymbol{H})}{Det(\boldsymbol{H})} < \frac{(r+1)^2}{r} \tag{4.8}$$

Where $Tr(\boldsymbol{H})$ is the trace of $\boldsymbol{H}$, $Det(\boldsymbol{H})$ is the determinant of $\boldsymbol{H}$ and $r$ is a threshold parameter.
The orientation of the feature is determined based on the orientation of the intensity gradient. A neighborhood is considered around the feature point, and the gradient is computed for each pixel. The gradient direction for each pixel is added to a 36-bin histogram. The peak of the histogram, as well as the values above the 80% of the peak, are used to compute the feature orientation.
The descriptor is created by selecting a $16\times16$ block around the feature, and dividing it in 16 sub-blocks with size $4\times4$. For each sub-block, an 8-bin orientation histogram is created. Hence, a total of 128 values are available, given by each bin of the histogram for each of the 16 sub-blocks in which the feature neighborhood was divided. The feature orientation is subtracted from the orientation of each sub-block: in this way, the orientations of the sub-blocks are relative to the orientation of the feature. These values are converted into integers, and they constitute the descriptor, that is invariant to rotations.

**SURF** SURF (Bay et al. [13]) is a feature extractor and descriptor that enhances speed by making use of box filters and integral images.
In SIFT, the Laplacian of Gaussian was approximated with the Difference of Gaussian.

In SURF, the Laplacian of Gaussian is even further approximated by means of box filters, i.e. filters that have a constant value over a patch. A box filter can be computed efficiently by using integral images [13].
In an integral image, the value of a pixel $(x, y)$ is the sum of all the pixels between the origin and $(x, y)$ in the original image. Let $I_{\boldsymbol{\Sigma}}$ be the integral image and $I$ be the original image:

$$I_{\boldsymbol{\Sigma}}(x,y) = \sum_{i=0}^{i\leq x} \sum_{j=0}^{j\leq y} I(i,j) \tag{4.9}$$

This method allows to compute much faster sums of rectangular patches of constant value, such as box filters.
In SURF, the detector is based on the Hessian matrix. The Hessian matrix $\boldsymbol{H}$ is defined in each point of the image as:

$$\boldsymbol{H}(\boldsymbol{x},\sigma) = \begin{bmatrix} L_{xx}(\boldsymbol{x},\sigma) & L_{xy}(\boldsymbol{x},\sigma) \\ L_{yx}(\boldsymbol{x},\sigma) & L_{yy}(\boldsymbol{x},\sigma) \end{bmatrix} \tag{4.10}$$

Where:

$$\begin{aligned} L_{xx}(\boldsymbol{x},\sigma) &= \frac{\partial}{\partial x^2} G(x,y,\sigma) * I(x,y) \\ L_{xy}(\boldsymbol{x},\sigma) &= \frac{\partial}{\partial x \partial y} G(x,y,\sigma) * I(x,y) \\ L_{yx}(\boldsymbol{x},\sigma) &= \frac{\partial}{\partial y \partial x} G(x,y,\sigma) * I(x,y) \\ L_{yy}(\boldsymbol{x},\sigma) &= \frac{\partial}{\partial y^2} G(x,y,\sigma) * I(x,y) \end{aligned} \tag{4.11}$$

Where $G(x, y, \sigma)$ is a Gaussian filter at scale $\sigma$ (Equation (4.5)), and $I(x, y)$ is the original image. The partial derivatives of the Gaussian filter are approximated with box filters.
The process consists in computing a convolution between the original image and each of the box filters that approximate the derivatives of $G(x, y, \sigma)$. This can be done extremely quickly by means of integral images. After this operation, the function $\boldsymbol{H}(\boldsymbol{x}, \sigma)$ is available for all the points in the image. The approximations of $L_{xx}$, $L_{xy}$, $L_{yx}$ and $L_{yy}$ are denoted with $D_{xx}$, $D_{xy}$, $D_{yx}$ and $D_{yy}$. According to Bay et al. [13], in order to balance the approximation introduced with box filters, the determinant of $\boldsymbol{H}$ is corrected as:

$$Det(\boldsymbol{H})_{approx} = D_{xx}D_{yy} - 0.9D_{xy}^2 \tag{4.12}$$

The maxima of the determinant of the Hessian matrix are features candidates.
To compute the descriptors, a squared area is built around each feature, and sub-divided in 16 regions (4 for each side). The vertical and horizontal Haar wavelets responses

(Papageorgiou et al. [63]), respectively $dx$ and $dy$, are computed for each of the 16 regions. For each region, the following vector is built:

$$\boldsymbol{v} = \left[\sum dx, \sum dy, \sum |dx|, \sum |dy|\right] \tag{4.13}$$

This results in 16 4-dimensional vectors, that are put together into a 64-dimensional descriptor.

**BRIEF** BRIEF, described in Calonder et al. [20], is a descriptor developed to speed-up both computation and matching. BRIEF is a binary descriptor: this allows to use the Hamming distance (instead of the L2-norm) for efficient computations in the matching phase. The Hamming distance is a method to compare binary strings, and consists in the number of positions in which the binary value is different.
The idea behind BRIEF consists in building a descriptor from a set of intensity comparisons within a patch of the image. Calonder et al. [20] define a test $\tau$ as follows:

$$\tau(p, \boldsymbol{x}, \boldsymbol{y}) = \begin{cases} 1 & \text{if } p(\boldsymbol{x}) < p(\boldsymbol{y}) \\ 0 & \text{otherwise} \end{cases} \tag{4.14}$$

Where $p(\boldsymbol{x})$ and $p(\boldsymbol{y})$ are pixel intensities in the locations $\boldsymbol{x}$ and $\boldsymbol{y}$ within the patch. The descriptor is built by repeating the test $\tau$ multiple times, in locations determined by a sampling distribution, (typically 128, 256 or 512). The sequence of the results of the tests is the descriptor.
Calonder et al. [20] propose various sampling patterns, generated from different random distributions.

**FAST** FAST, described in Rosten et al. [70], is a corner detector that uses a machine learning approach based on decision trees.
FAST builds a circle made of 16 pixels around a central pixel, that is a feature candidate. Let $I_p$ be the intensity of the central pixel. The intensities of the surrounding pixels are compared to the intensity of the center, and the center is classified as a corner if:

- A certain number of contiguous surrounding pixels has intensity higher than $I_p + t$.
- A certain number of contiguous surrounding pixels has intensity lower than $I_p - t$.

Where $t$ is a threshold value. In practice, the comparison is done for only 4 of the 16 surrounding pixels, and the central point is classified as a corner if 3 pixels over 4 have

the same result in the comparison.
The procedure is initially carried out for all the pixels of all the images of a training set. A decision tree is built by minimizing an entropy function, and the result is used to detect corners in previously unseen images.
Non-maximal suppression is used to avoid the detection of multiple adjacent features. The score function used in non-maximal suppression is based on the intensity difference between the center and the surrounding pixels.

**ORB** ORB, presented in Rublee et al. [71], is a features extractor and matcher based on FAST and BRIEF. During the features extraction with FAST, a computation of the feature orientation is introduced in order to achieve rotation invariance. This FAST variant is denominated oFAST. Moreover, to compute the feature orientation, the intensity centroid of the FAST corner is computed. According to Rosin [69], the moments of a patch of order $p, q$ are:

$$m_{pq} = \sum_{x,y} x^p y^q I(x,y) \tag{4.15}$$

Where $I(x,y)$ is the image intensity. The centroid $C$ is determined as:

$$C = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}}\right) \tag{4.16}$$

The patch orientation is:

$$\theta = atan(m_{01}, m_{10}) \tag{4.17}$$

Where $atan$, in this case, refers the quadrant-aware operation. The patch orientation $\theta$ is used to achieve rotation invariance with the BRIEF descriptor. This variant of the BRIEF descriptor is denominated rBRIEF.
Consider a feature and the corresponding patch around the feature. A matrix $\boldsymbol{S}$ containing the locations of the test performed to compute the BRIEF descriptor is built:

$$\boldsymbol{S} = \begin{bmatrix} x_1 & x_2 & \dots & x_n \\ y_1 & y_2 & \dots & y_n \end{bmatrix} \tag{4.18}$$

This matrix can be rotated, according to the patch rotation $\theta$, using a rotation matrix $\boldsymbol{R(\theta)}$:

$$\boldsymbol{S_\theta} = \boldsymbol{R(\theta)S} \tag{4.19}$$

A set of pre-computed BRIEF test locations patterns is built. These locations are rotated each time according to the rotation of the patch, and the correct $\boldsymbol{S_\theta}$ is used. In this way, the sampling points are consistent.

**BRISK** BRISK, presented in Leutenegger et al. [44], is a feature detector-descriptor based on a FAST-based detector and a binary descriptor.
A scale space of the image is built, and the features are extracted using the FAST method at different scales. Non-maximum suppression is applied to select the best feature in the scale space and within a neighborhood of the point.
The descriptor construction is conceptually similar to BRIEF. However, the sampling pattern used by BRISK is deterministic. In addition, a Gaussian smoothing is applied before the sampling to reduce aliasing. In order to achieve rotation invariance, the sampling pattern is rotated by $\alpha = atan(g_x, g_y)$, where the quadratic-aware version of $atan$ is used, while $g_x$ and $g_y$ are the $x$ and $y$ components of the local gradient $\boldsymbol{g}$ of the feature, that can be estimated from the intensities of the points in the sampling pattern. In this way, the sampling pattern used for the construction of the descriptors is each time rotated according to the orientation determined by the gradient of the feature, and as a consequence the feature is invariant to rotations.
Hence, the working principle of BRISK is similar to ORB. The main difference is the way the orientation of the feature is computed.

**Shi-Tomasi** The Shi-Tomasi corner detector, described by Shi and Tomasi [75], is a corner detector based on the work presented by Harris and Stephens [40].
In the Harris corner detector, the underlying idea is that a point in an image represents a corner if the following expression is maximized:

$$E(u,v) = \sum_{x,y} w(x,y) \left[I(x+u, y+v) - I(x,y)\right]^2 \tag{4.20}$$

where $w(x,y)$ is a window function, $I(x,y)$ is the image and $(u,v)$ is a small displacement.
If the expression of $E(u,v)$ is linearized, it can be written as:

$$E(u,v) = \begin{bmatrix} u & v \end{bmatrix} M \begin{bmatrix} u \\ v \end{bmatrix} \tag{4.21}$$

where:

$$\boldsymbol{M} = \sum_{x,y} w(x,y) \begin{bmatrix} I_x I_x & I_x I_y \\ I_x I_y & I_y I_y \end{bmatrix} \tag{4.22}$$

It results that the function $E(u,v)$ is maximized, and hence the point is a corner, if the following score is maximized:

$$R = \lambda_1\lambda_2 - k(\lambda_1 + \lambda_2) \tag{4.23}$$

While the Harris corner detector score function (Equation (4.23)) is based on an arbitrary notion of "good feature", i.e. a corner, Shi and Tomasi [75] consider instead as "good feature" a feature that can be easily tracked by the tracking algorithm proposed in [75], that is the KLT feature tracker. With this goal, they formulate the following scoring function, used in the Shi-Tomasi corner detector:

$$R = min(\lambda_1, \lambda_2) \tag{4.24}$$

**STAR** STAR is a feature detector implemented in OpenCV and derived from CenSurE (Center Surround Extremas for Realtime Feature Detection and Matching), described in Agrawal et al. [9]. According to the authors, the use of the image pyramid in algorithms such as SIFT, i.e. the progressive reduction of the image size after a certain number of scales, is responsible of a loss in accuracy, since features are extracted from smaller images. However, in SIFT the use of the image pyramid is necessary to maintain an acceptable computational cost. Agrawal et al. [9] propose to extract features candidates at each scale and at every pixel of the image, using center-surrounded filters that can be computed quickly and independently from the filter size, thanks to integral images. The extrema across scale and location are then selected.
In SIFT, the Laplacian was approximated by means of Difference of Gaussian. CenSurE uses bi-level filters, that multiply the image either by 1 or $-1$. According to Agrawal et al. [9], the best result would be achieved by using circular filters, the closest to a laplacian filter. However, in order to speed-up the computations, the circle can be approximated with polygons. In the case of a square, the filter is basically a two dimensional Haar wavelet. However, the authors find that the circle is better approximated with an octagon, due to stronger resemblance of the shapes. The authors use a scale space made by 7 scales, and the response to the filter is computed at each pixel of each scale. The pixels that have maximum response in a neighborhood over both location and scale are selected as features. In particular, a $3 \times 3 \times 3$ neighborhood is used. A threshold is applied to the magnitude of the filter response, to avoid selecting "weak" points. Finally, features that lie along an edge are removed.
In the OpenCV implementation the STAR algorithm, instead of using octagonal filters, uses a filter with the shape of two overlapping squares, one rotated by 45 with respect to

the other.

## Features tracking and matching

Two approaches are available to follow a feature in sequential frames: matching and tracking.

**Features Matching** Features matching consists in extracting the features separately from two images, and matching them by means of the descriptors. The simplest technique is the so-called Brute Force matching [7]: given two sets of features along with their descriptors, a distance calculation is performed for each descriptor of the first set, with respect to each descriptor of the second set. The closest match is returned. For floating point descriptors (SIFT, SURF), the L2 norm of the Euclidean distance is used. Let $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ be the descriptors. The L2 norm of their distance is:

$$||\boldsymbol{d}_1 - \boldsymbol{d}_2||_{L2} = \sum_i (d_{1,i} - d_{2,i})^2 \tag{4.25}$$

For binary descriptors (ORB, BRIEF, BRISK), the Hamming distance is used.
Matches are preliminarily filtered by applying the Lowe's ratio test described in Lowe [50]. The test consists in comparing the distances of the closest and second-closest matches, and accepting the closest match only if the difference is sufficiently large.
An alternative approach to Brute Force matching is FLANN-based matching. FLANN (Fast Library for Approximate Nearest Neighbors) (Muja and Lowe [56]) is a library for performing approximate nearest neighbor searches in high dimensional spaces. The result is approximate, but the method is expected to be more computationally efficient than Brute Force matching, especially if large numbers of descriptors are involved.
Since features matching approach relies on extracting features and descriptors separately, matching is possible also with a relatively low images acquisition frequency, as long as the descriptors are not excessively different. On the other side, the main disadvantage consists in the computation of the descriptors, which can demand a relevant time and memory, particularly when large numbers of features are involved. In addition, depending on the descriptor used, the robustness to certain factors such as changes in illumination and rotations could not be satisfactory.

**Feature tracking** Feature tracking consists conceptually in following a feature, based on image intensity under the hypothesis that small changes are present in the images. The approach used in this work, implemented in OpenCV and described in Bouguet [17],

consists in a pyramidal representation of the classical Lucas-Kanade algorithm (Shi and Tomasi [75]).
In Shi and Tomasi [75], the authors observe that images taken at near time instants are strongly related to each other. Images taken at time $t$ and $t + \tau$ are assumed to be correlated according to the following expression:

$$I(x, y, t + \tau) \approx I(x - \xi(x, y, t, \tau), y - \eta(x, y, t, \tau)) \tag{4.26}$$

This means that an image taken at time $t+\tau$ can be obtained by displacing every point of the image taken at time $t$ by a suitable amount $\xi(x, y, t, \tau)$ in $x$ direction and $\eta(x, y, t, \tau)$ in $y$ direction. In case of large intervals between image acquisition, or more in general of large displacements of the acquisition source, Equation 4.26 is violated due to changes in viewpoint and illumination conditions.
The problem of tracking a feature is formalized in Bouguet [17] as determining a displacement vector $\boldsymbol{d}$, which represents the displacement of a patch from the first image to the second image. If $I$ and $J$ are two sequential images, the point $\boldsymbol{x}$ in image $I$ is moved to the position $\boldsymbol{x} + \boldsymbol{d}$ in image $J$. Hence, to track a feature consists in finding a vector $\boldsymbol{d}$ such that:

$$J(\boldsymbol{Ax} + \boldsymbol{d}) = I(\boldsymbol{x}) \tag{4.27}$$

where $\boldsymbol{A}$ is an homography, equal to the identity matrix in case of pure translation.
Because of noise and second order effects, variations in illuminations and, more in general, variations in the environment, Equation 4.26 is not exactly satisfied. The problem can then be reformulated as finding $\boldsymbol{A}$ and $\boldsymbol{d}$ that minimize the error:

$$\epsilon = \int\int_W [J(\boldsymbol{Ax} + \boldsymbol{d}) - I(\boldsymbol{x})]^2 \, w(\boldsymbol{x}) d\boldsymbol{x} \tag{4.28}$$

Where $W$ is an acquisition window and $w(\boldsymbol{x})$ a weighting term.
Under the assumption of small variations in the image (or, alternatively, of small displacements of the camera and small changes in the environmental conditions) and pure translation, (4.28) can be reduced to a linear system with $\boldsymbol{d}$ as the only unknown variable.
Bouguet [17] suggest an implementation of the algorithm based on image pyramid representation.
Pyramids, presented in Edward et al. [30], consist in a method of representing an image based on progressive filtering and downsampling. This algorithm is implemented in the OpenCV library.
The biggest advantage of features tracking methods is the fact that descriptors are not required: for this reasons, features tracking algorithms are fast and do not require large amounts of memory to store the descriptors. On the other side, a sufficiently high

acquisition frequency is required to ensure that the linear hypothesis holds. From a practical stand-point, the OpenCV implementation of the Lucas-Kanade algorithm builds a window of a fixed size around the feature location. If, in the second frame, the displacement of the corresponding point is too large and it falls outside the window, the tracking does not work.

## Outliers rejection

During the features matching process, a preliminary filtering is done using Lowe's test. Lowe's test, described in [50], consists in determining, for each feature, the two best matches, and accepting the best match only if the ratio between the two distances is lower than a ratio:

$$||\boldsymbol{d}_1|| \leq k \cdot ||\boldsymbol{d}_2|| \tag{4.29}$$

where $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$ are the distances referred respectively to the best and the second-best matches, and $k$ is a factor that, in this work, has been set to 0.7. In this way, a match is considered of good quality only if it is well distinct from the others. However, if high accuracy is required, an outliers rejection method might be needed to remove false matches that are possibly still present.
In this work, matches are filtered using the RANSAC algorithm, described in Hartley and Zisserman [42]. The logic behind the RANSAC algorithm is to estimate a model from a subset of the available data, enlarging this set with consistent data, and attempting to eliminate invalid data. OpenCV offers various implementations of the RANSAC algorithm. The implementation used in this work is based on the estimation of the essential matrix, which embeds information about the cameras relative poses (Hartley and Zisserman [42]).

### 4.1.2. Method

The combinations of algorithms presented in the previous section that have been used are reported in Table 4.1. At each step, two sequential images are considered.
If Feature Matching is used, the features and descriptors are extracted from the two images, and they are matched.
If feature tracking is used, the features are extracted from the first image, and tracked in the following images. The features have been re-initialized every 30 frames, in order to avoid ending up without any more of them.
The features matched or tracked are filtered using the RANSAC filter, as explained in

Table 4.1: Algorithms used. For each algorithm the feature extractor and, if applicable, the descriptor, the matcher and the tracker have been reported.

| Algorithm | Extractor | Descriptor | Matcher | Tracker |
|---|---|---|---|---|
| SIFT | SIFT | SIFT | Brute Force | - |
| SURF | SURF | SURF | Brute Force | - |
| ORB | oFAST | rBRIEF | Brute Force | - |
| BRISK | BRISK | BRISK | Brute Force | - |
| KLT | Shi-Tomasi | - | - | KLT |
| STAR + BRIEF | STAR | BRIEF | Brute Force | - |

Section 4.1.1.
In order to investigate the effect that a more or less frequent image acquisition has on the tracking, the procedure is repeated more times, varying the parameter $s$, that indicates the frequency of the images that are matched: the image $n$ is matched with the image $n + s$. Hence, $s - 1$ images are skipped.

## 4.2. Features extraction and tracking with convolutional neural networks

As explained in Section 2.3, optical navigation can be performed using techniques based on deep learning. In particular, for computer vision tasks, Convolutional Neural Networks (CNN) are particularly suitable, since they can naturally manage grid-structured inputs such as images by means of operators such as convolutions and sampling.

### 4.2.1. SuperPoint

SuperPoint, presented by DeTone et al. [24], is a CNN developed to extract features and descriptors from an image. The implementation used in this work is developed by Rémi Pautrat and Paul-Edouard Sarlin[2], and is based on the original paper. Minor modifications to the code were introduced to ensure compatibility with the rest of the work. SuperPoint was chosen among the networks presented in Chapter 2 because it resulted relatively user-friendly, fast and required a reasonable amount of data to work.

[2]`https://github.com/rpautrat/SuperPoint`

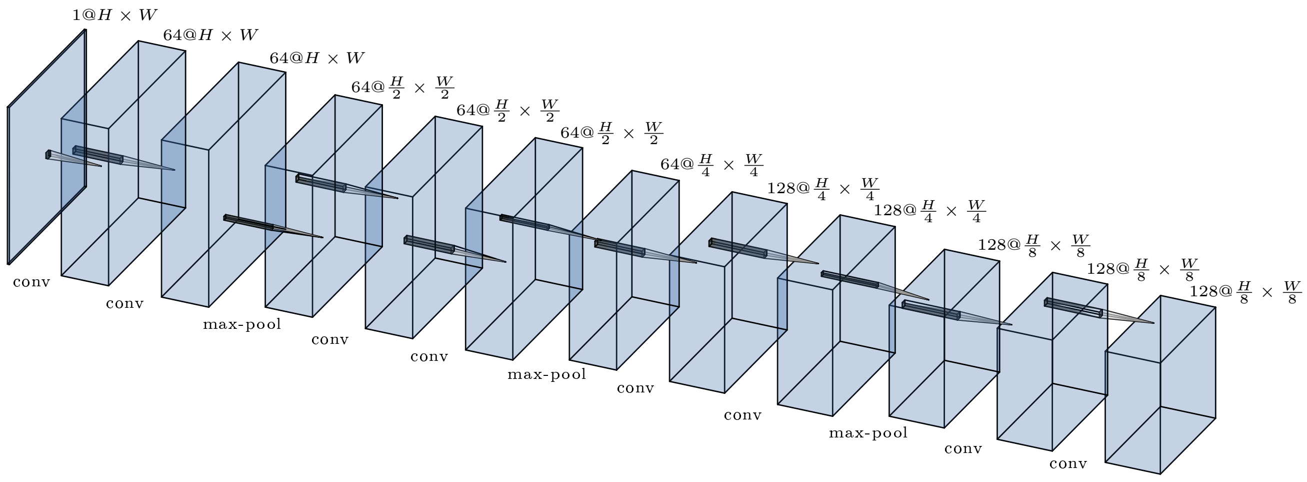

$1@H \times W$
$64@H \times W$
$64@H \times W$
$64@\frac{H}{2} \times \frac{W}{2}$
$64@\frac{H}{2} \times \frac{W}{2}$
$64@\frac{H}{2} \times \frac{W}{2}$
$64@\frac{H}{4} \times \frac{W}{4}$
$128@\frac{H}{4} \times \frac{W}{4}$
$128@\frac{H}{4} \times \frac{W}{4}$
$128@\frac{H}{8} \times \frac{W}{8}$
$128@\frac{H}{8} \times \frac{W}{8}$
$128@\frac{H}{8} \times \frac{W}{8}$
conv
conv
max-pool
conv
conv
max-pool
conv
conv
max-pool
conv
conv

Figure 4.3: Architecture of the SuperPoint Encoder.

## Network structure

SuperPoint's architecture is fully convolutional, and it is composed by an encoder and a double pipeline decoder: one line is for the extraction of the features, the other for the computation of the descriptors.

**Encoder** An encoder is a network that takes an input, in this case an image, and outputs a tensor that embeds information extracted from the input. Superpoint uses a VGG-style encoder [77] to reduce the dimensionality of the image. The characteristics of the VGG-style encoder, based on the work by Simonyan and Zisserman [77], are a large network depth and a very small size of the convolution filters, $3 \times 3$ in this case. The encoder is constituted by four blocks. Each one of the first two blocks is made by two convolutional layers using $3 \times 3$ kernels with 64 channels, and a $2 \times 2$ max pooling. The third block is made by two convolutional layers using $3 \times 3$ kernels with 128 channels, and a $2 \times 2$ max pooling layer. The fourth block is made by two convolutional layers using $3 \times 3$ kernels with 128 channels.
As a result, an input image sized $H \times W$ is reduced to a tensor of size $\frac{H}{8} \times \frac{W}{8} \times 128$. Because of that, the input image should have dimensions that are multiple of 8. According to the authors, the network should work also if this condition is not satisfied, but performance might be worse. Note that $H \times W$ might not be the original size of the image: in order to keep the computations sustainable for hardware, the original image is typically resized during the preprocessing. More details about preprocessing are reported in Section 4.2.2. The architecture of the encoder is illustrated in Figure 4.3.

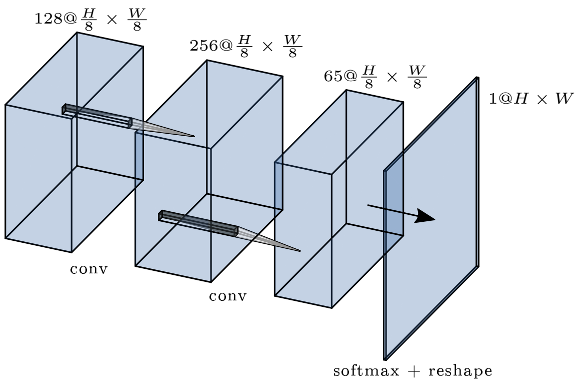

$128@\frac{H}{8} \times \frac{W}{8}$
$256@\frac{H}{8} \times \frac{W}{8}$
$65@\frac{H}{8} \times \frac{W}{8}$
$1@H \times W$
conv
conv
softmax + reshape

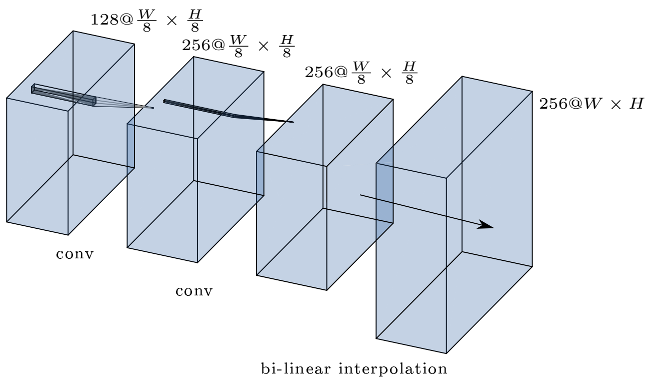
Figure 4.4: Architecture of the SuperPoint detection decoder.

**Features detection decoder** The detector decoder is used for the extraction of the features. The features detection decoder takes as input the output tensor of the encoder. First, the input is processed by a convolutional layer that uses $3 \times 3$ filters with 256 channels, followed by a convolutional layer that uses $1 \times 1$ filters with 65 channels. The output is a $\frac{H}{8} \times \frac{W}{8} \times 65$ tensor, called $\mathcal{X}$. The 65 channels correspond to 64 $8 \times 8$ non-overlapping regions of pixels, and a "no interest points" dustbin. A channel-wise softmax is applied, the dustbin is removed and the output is reshaped into a $H \times W \times 1$ tensor, hence the same size of the input image. Each element of the output corresponds to the probability that the corresponding pixel in the original image is a feature. The elements with a value above a certain threshold are selected as features.
The architecture of the detection decoder is illustrated in Figure 4.4.

**Descriptors decoder** The second line of the decoder is used to produce the descriptors. The descriptors decoder takes as input the output tensor of the encoder. The input is processed by a convolutional layer that uses $3 \times 3$ filters with 256 channels, followed by convolutional layer that uses $1 \times 1$ kernels with 256 channels. As a result, a $\frac{H}{8} \times \frac{W}{8} \times 256$ tensor is produced. This tensor, called $\mathcal{D}$, constitutes a semi-dense map of descriptors, one every $8 \times 8$ patch. After that, a $H \times W \times 256$ tensor is produced using bi-linear interpolation (in the original DeTone et al. [24], however, bi-cubic interpolation was used). In this way, a dense map of descriptors is produced. Finally, the descriptors are L2-normalized.
The architecture of the detection decoder is illustrated in Figure 4.5.

## Training procedure

The training procedure of SuperPoint is self-supervised: that means that it does not require a labeled dataset. This is a big advantage since, according to DeTone et al. [24], no dataset of labeled interest points currently exist, and its creation would require a



$128@\frac{W}{8} \times \frac{H}{8}$
$256@\frac{W}{8} \times \frac{H}{8}$
$256@\frac{W}{8} \times \frac{H}{8}$
$256@W \times H$
conv
conv
bi-linear interpolation

Figure 4.5: Architecture of the SuperPoint descriptor decoder.

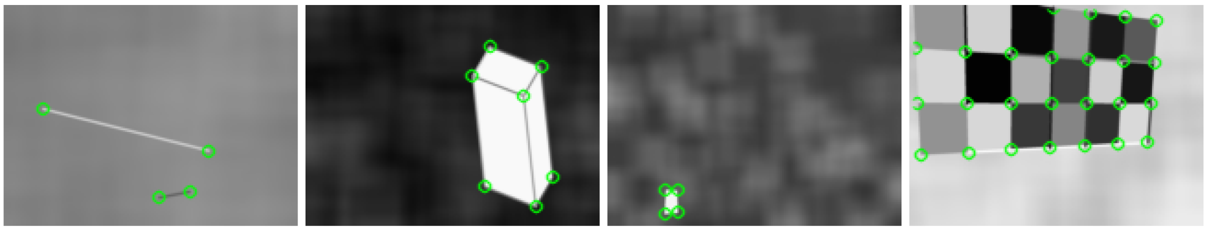


Figure 4.6: Examples of images in the synthetic shapes dataset, along with interest points labels (in green).

massive amount of work.
The training labels are produced by MagicPoint, a network composed by the SuperPoint encoder and the detection decoder. This network is initially trained on a dataset composed by basic geometric shapes, called synthetic shapes, that are synthetically generated and whose interest points such as corners are known a-priori. Examples of images in the synthetic shapes dataset, along with the labels, are reoorted in Figure 4.6. Then, the model is improved on real world images. The labels produced can be used to train SuperPoint.

**Loss function** During the training, pairs of synthetically warped images are used. For each image, the pseudo-ground truth location of the interest points and the homography matrix $\boldsymbol{H}$ that relates the two images are known.
The SuperPoint loss is expressed as:

$$L(\boldsymbol{X}, \boldsymbol{X'}, \boldsymbol{D}, \boldsymbol{D'}, \boldsymbol{Y}, \boldsymbol{Y'}, S) = L_p(\boldsymbol{X}, \boldsymbol{Y}) + L_p(\boldsymbol{X'}, \boldsymbol{Y'}) + \lambda L_d(\boldsymbol{D}, \boldsymbol{D'}, S) \tag{4.30}$$

Where $\boldsymbol{X}$ and $\boldsymbol{X'}$ are the results of the second convolutional layer of the features detection decoder for the two images, $\boldsymbol{D}$ and $\boldsymbol{D'}$ are the results of the second convolutional layer of the descriptors decoder for the two images, $\boldsymbol{Y}$ and $\boldsymbol{Y'}$ are the pseudo ground truths

(known for the two images). The latter are binary images with 1 in the locations of the features, and 0 elsewhere[3].
$L_p$ is the part of loss function related to the features. It is formulated as:

$$L_p(\boldsymbol{X},\boldsymbol{Y}) = \frac{1}{H_cW_c}\sum_{h=1,w=1}^{H_c,W_c} l_p(\boldsymbol{x}_{hw}, y_{hw}) \tag{4.31}$$

$$l_p(\boldsymbol{x}_{hw}, y) = -log\left(\frac{exp(x_{hwy})}{\sum_{k=1}^{65} exp(x_{hwk})}\right) \tag{4.32}$$

In equations (4.31) and (4.32), $\boldsymbol{x}_{hw}$ is the vector at coordinates $(h, w, :)$ of the tensor $\boldsymbol{X}$, $y_{hw}$ is the element at position $(h, w)$ of the pseudo-ground truth $\boldsymbol{Y}$, $x_{hwy}$ is the element $y$ of the vector $\boldsymbol{x}_{hw}$, $H_c = \frac{H}{8}$ and $W_c = \frac{W}{8}$. The 65th element in the summatory refers to the no interest point dustbin, used if no features are detected in the patch.
The pseudo ground truth $\boldsymbol{Y}$ is an $H_c \times W_c \times 1$ tensor, whose elements are integers from 0 to 64[4]. The integer represents at which element of the vector $\boldsymbol{x}_{hw}$ the feature is located, i.e. in which pixel of the $8 \times 8$ patch represented by the vector $\boldsymbol{x}_{hw}$ the point is located.
The descriptor loss $L_d(\boldsymbol{D}, \boldsymbol{D'}, \mathrm{S})$ is applied to pairs of descriptors to the first and second images.
First, the authors define the homography-induced correspondence between the cells $(h, w)$ and $(h', w')$ as:

$$s_{hwh'w'} = \begin{cases} 1 & \text{if} \quad \|\widehat{\boldsymbol{H}\boldsymbol{p}_{hw}} - \boldsymbol{p}_{h'w'}\| \leq 8 \\ 0 & \text{otherwise} \end{cases} \tag{4.33}$$

Where the subscripts $h$ and $w$ indicate the central pixel in the cell $(h, w)$, and $\widehat{\boldsymbol{H}\boldsymbol{p}_{hw}}$ the same quantity with the homography applied.

This part of loss function is given by:

$$L_d(\boldsymbol{D}, \boldsymbol{D'}, \mathrm{S}) = \frac{1}{(H_cW_c)^2}\sum_{h=1,w=1}^{H_c,W_c}\sum_{h'=1,w'=1}^{H_c,W_c} l_d(\boldsymbol{d}_{hw}, \boldsymbol{d'}_{h'w'}, s_{hwh'w'})$$

$$l_d(\boldsymbol{d}, \boldsymbol{d'}, s) = \lambda_d \cdot s \cdot max(0, m_p - \boldsymbol{d}^T\boldsymbol{d'}) + (1-s)\cdot max(0, \boldsymbol{d}^T\boldsymbol{d'} - m_n)$$

Where $m_p$ and $m_n$ are a positive and a negative margin, and the term $\lambda_d$ is used to balance the fact that the negative correspondences are more than the positive ones.

[3]Source: https://github.com/rpautrat/SuperPoint/issues/95
[4]Source: https://github.com/rpautrat/SuperPoint/issues/114

**Training procedure** As stated in DeTone et al. [24], currently no labeled dataset for features exists. Hence, a self-supervised learning approach has been developed for SuperPoint.
In order to produce labels, a network called MagicPoint is used. MagicPoint consists essentially in SuperPoint with the descriptor decoder removed. Hence, it is used for the labeling of the images, but not for the computation of the descriptors.
In order to train MagicPoint, a synthetic dataset called Synthetic Shapes, made by simple shapes such as lines and polygons oriented in space, is produced. Since the images are synthetic, the positions of the corners are known.
The authors found out that MagicPoint trained on Synthetic Shapes can perform reasonably well on real world images. However, in order to improve the performance, the training is carried out also using real world images.
In order to train on real world images, the process of Homographic Adaptation [24] is used. Homographic Adaptation consists in sampling homographies from an image, extracting the features from the homographies, and projecting the features back on the original image. According to DeTone et al. [24], this approach increases the number of detections and improves the repeatability.
Once the results obtained with MagicPoint are considered sufficiently good, SuperPoint can be trained with the same images used for MagicPoint. In this phase, the network improves its ability of computing the descriptors.

### 4.2.2. Datasets and models

#### Creation of the dataset

Since SuperPoint's training works with single images, it is not necessary to train on sequential images that reproduce an actual image acquisition. For this reason, it is possible to randomly generate the training set. This approach also grants an higher data entropy than producing images according to a pre-determined trajectory.
Camera position, orientation, illumination direction and light intensity are randomly determined from pre-defined ranges, and images are rendered using these conditions.
As explained in Chapter 3, scenes with a large angle between the sun and the spacecraft are not rendered, to avoid completely or almost completely black images.
The images produced are split into training set and validation set. According to the SuperPoint developers, the size of the validation set is required[5] to be a multiple of $(batch\quad size) \cdot (GPUs\quad number)$. Since MagicPoint has been trained on a single GPU

[5]https://github.com/rpautrat/SuperPoint/issues/17

with a batch size of 32, it was required to have a validation set size multiple of 32. Around the 10% of the images produces were used for the validation, and the rest constituted the train set. The models were then tested on the same dataset used for traditional algorithms. More details about the datasets used and their generation are reported in Section 3.5.

## Models

In order to train the network, the Google Colab [6] platform has been used. Colab is a cloud computing platform developed by Google that allows to execute python code on powerful hardware.
The following models were selected among the trained:

- **SPV6**: provided by the authors, it is trained on the COCO dataset[7], containing around 80000 generic images.
- **SuperPoint Base**: trained with 300000 iterations on the COCO dataset.
- **SuperPoint Bennu**: the model SuperPoint Base was finetuned with 50000 iterations on the dataset composed by 10000 images of Bennu. The MagicPoint model used to generate the labels
- **SuperPoint Eros**: the model SuperPoint Base was finetuned with 50000 iterations on the dataset composed by 5000 images of Eros. The MagicPoint model used to generate the label was trained first with 18000 iterations from Synthetic Shapes, and then with 25000 iterations from the Eros dataset.

During the training $H = 320$ and $W = 240$ were used, hence the images were downsized from a resolution of $1600 \times 1200$ pixels to $320 \times 240$ pixels during the preprocessing. This corresponds to reducing the size of a factor of 5. This parameter was kept from the original implementation of the CNN.

## Testing

The models were tested on the same datasets used for traditional techniques. As Figure 4.7 shows, the original image sized $1600 \times 1200$ pixels is scaled by a factor $s$ during the preprocessing, and the network effectively receives an input sized $(H, W) = \left(\frac{1600}{s}, \frac{1200}{s}\right)$. As a consequence, the features are extracted an located on the resized image. In order to have results that are comparable with the traditional techniques, the output of the CNN

[6] `https://colab.research.google.com/`
[7] `https://cocodataset.org`

is brought again to the original size, and the features coordinates on the resized images are multiplied by the scale factor to project them on the original image.
It was observed that, contrary to expectations, resizing the original image to the same size used during the training (320×240 pixels) is not beneficial, since during the postprocessing the errors are effectively multiplied by the scale factor $s$, and using a large $s$ results in the growth of the errors. On the other side, giving the image at the original size as input to the CNN or using a small scale factor (such as $s = 2$) required an excessive amount of memory and resulted in errors or performance losses. As a consequence, $s = 4$ was effectively used.

## 4.3. Ray Casting

The fact that an algorithm recognizes two points in different images as the same physical feature does not mean that the result is actually correct. In fact, false matches can occur. When the performance of tracking methods is investigated, it is necessary to apply a method that allows to determine how the algorithm is performing.
In this work, a method based on ray casting has been applied. The coordinates of the physical features on the asteroid model that correspond to the features extracted on the image are determined, and these coordinates are used to determine the correct location of the corresponding features on the following images, independently from the tracking algorithm. The procedure was inspired to the work by Morrell et al. [55], and developed independently.

### 4.3.1. Localization on the features on the 3D model

When the a set of features is extracted from an image, their 2D coordinates on the image are given. These features indicate physical features on 3D model of the asteroid. The 3D coordinates of the features can be determined using ray casting.
Ray casting consists in casting a ray from an origin in a given direction, and determining the location in which the ray intersects a target object.
The ray casting procedure is here explained in more detail.
For each feature $i$, let $\mathbf{u}_i$ be the unit vector that goes from the the camera center to the feature on the image plane. Let $(x_i, y_i)$ be the coordinates of the feature $i$ on the image plane.
Let $\mathcal{C}$ be the camera coordinate frame. The position of the feature $i$ in the coordinate

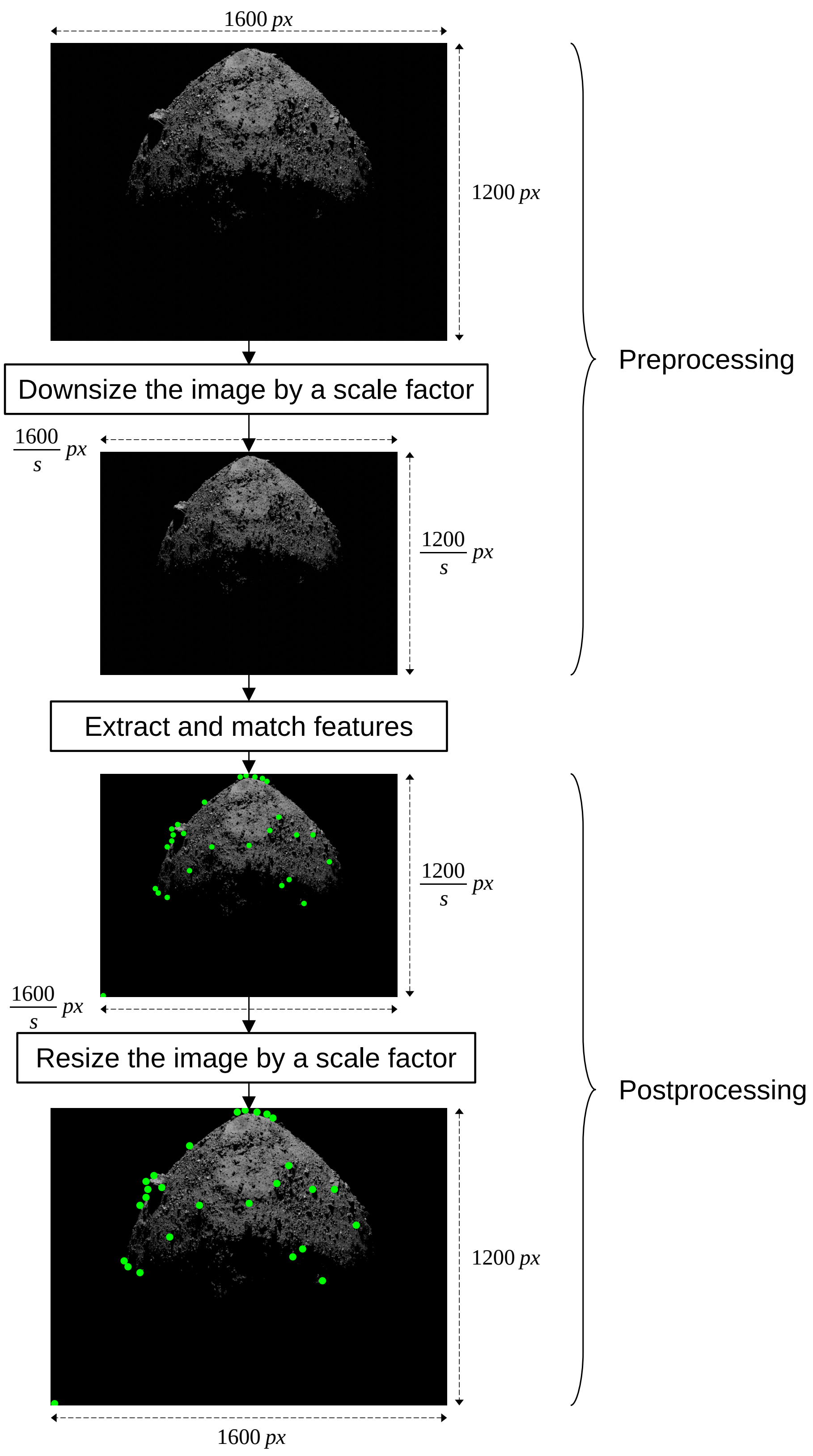


Figure 4.7: Scheme of the testing procedure.

frame $\mathcal{C}$ is:

$$\boldsymbol{x}_i = \begin{bmatrix} x_i \\ y_i \\ f \end{bmatrix} \tag{4.34}$$

Where $f$ is the focal length. The unit vector $\boldsymbol{u}_i$ is obtained normalizing $\boldsymbol{x}_i$.
The vector $\boldsymbol{u}_i$ is in the reference frame $\mathcal{C}$. The same vector, in the world coordinates frame $\mathcal{W}$, is:

$$\boldsymbol{u}_{i,\mathcal{W}} = \boldsymbol{R}_{\mathcal{CW}} \boldsymbol{u}_{i,\mathcal{C}} \tag{4.35}$$

Where $\boldsymbol{R}_{\mathcal{CW}}$ is the rotation matrix from the camera coordinate frame to the world coordinate frame, determined by the orientation of the camera.
At this point, the ray casting procedure consists in determining the point in which $\boldsymbol{u}_{i,\mathcal{W}}$ intersects the model. In blender, a built-in function is available. The intersection point indicates the 3D location of the physical feature that was localized by the feature $i$.
The Blender implementation of the ray casting algorithm consists in a bounding box (or bounding volume) test to exclude preliminarily those rays that do not intersect the target, followed by a Bounding Volume Hierarchy [31] tree to compute the actual intersection.
According to Ericson [31], a bounding volume is a volume encapsulating an object. The idea behind bounding volumes is to have volumes that are simpler than the target object, that can be used as a cheap test to check if the ray is likely to intersect the target. The intersections of the rays with the target are computed only if an intersection with the bounding volume exists, otherwise they are excluded. Cheap algorithms [31] are available to filter out the rays that do not intersect the bounding volume, and hence do not intersect the target.
The following step is to actually compute the intersection with the target. Bounding Volume Hierarchies (BVH) are used to this purpose. Ericson [31] defines BVH trees as sets of bounding volumes that are organized hierarchically. Larger volumes encapsulate smaller volumes. Having smaller volumes allows to compute the intersection with the target object more precisely. However, intersections with smaller volumes need to be checked only if an intersection with the larger volume encapsulating them exists. This hierarchical structure allows to reduce the number of computations required from an exponential to a logarithmic progression.

### 4.3.2. Reprojection of the feature on the camera

Once the 3D coordinates of the physical feature $i$ are available, they are reprojected back as 2D coordinates on the image plane. In this way, it is possible to compare the

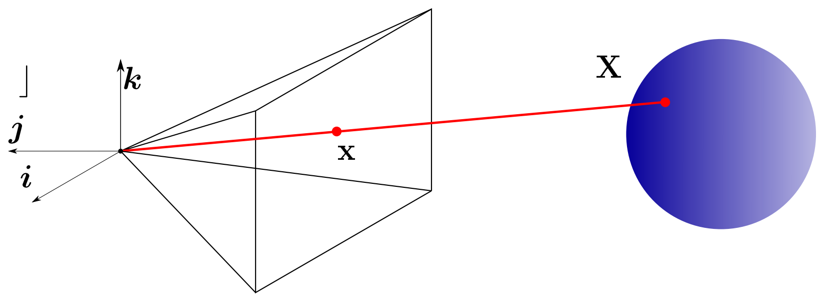


Figure 4.8: Illustration of the ray casting procedure. The red ray is casted from the camera center $\mathbf{C}$ in the direction of the pixel $\mathbf{x}$, and hits the model (represented by the blue sphere) in the point $\mathbf{X}$. The axes use the same convention used in Blender.

location where the tracking algorithm locates the feature, i.e. the coordinates obtained by the matching or tracking of the features, and the true coordinates in which the physical feature is projected on the image plane.
Let $\boldsymbol{X}_{\mathcal{W},i}$ be the coordinates of the feature $i$ in the world coordinate frame $\mathcal{W}$. Let $\boldsymbol{R}_{\mathcal{WC}} = \boldsymbol{R}_{\mathcal{CW}}^{-1}$ be the rotation matrix from the world frame $\mathcal{W}$ to the camera frame $\mathcal{C}$. The coordinates of the feature in the camera frame, hence, are $\boldsymbol{X}_{\mathcal{C},i} = \boldsymbol{R}_{\mathcal{WC}}\boldsymbol{X}_{\mathcal{W},i}$.
According to Hartley and Zisserman [42], for a pinhole camera, the mapping from a point in the camera frame to a point on the image plane is:

$$\begin{bmatrix} X_i \\ Y_i \\ Z_i \end{bmatrix} \rightarrow \begin{bmatrix} f\frac{X}{Z} \\ f\frac{Y}{Z} \end{bmatrix} \tag{4.36}$$

Hence, given the coordinates $\boldsymbol{X}_{\mathcal{C},i}$ of the feature, the correct location of the feature on the image plane is determined according to 4.36.
The Blender implementation[8] of the aforementioned procedure is slightly different. Blender uses a coordinate system in which $\boldsymbol{k}$ is directed backwards with respect to the principal axis, such as Figure 4.8. Consider a point $X_{\mathcal{C},i} = [X_i, Y_i, Z_i]$ in the camera coordinate frame. Consider the camera sensor, shown in Figure 4.9. Let $x_{max}$ and $x_{min}$ be, respectively, the maximum and minimum $x$ coordinates of the corners of the camera sensor, and $y_{max}$ and $y_{min}$ be, respectively, the maximum and minimum $y$ coordinates of the corners of the camera sensor. The Normalized Device Coordinates[9] (NDC), ranging from $(0,0)$ for the lower-left corner to $(1,1)$ for the upper-right corner. are computed as:

[8]`https://github.com/blender/blender/blob/594f47ecd2d5367ca936cf6fc6ec8168c2b360d0/release/scripts/modules/bpy_extras/object_utils.py`, function *world_to_camera_view*
[9]`https://www.ncl.ucar.edu/Document/Graphics/ndc.shtml`

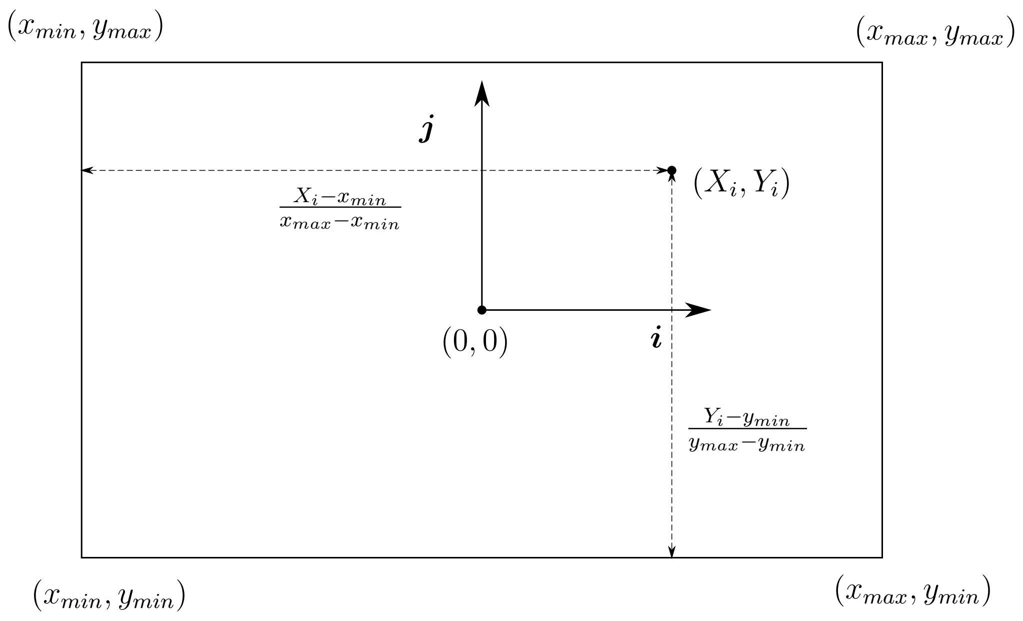


Figure 4.9: Blender implementation of the projection of a point on the camera plane. The spatial coordinates in the camera frame $(X_i, Y_i)$ are converted in Normalized Device Coordinates (NDC). The lower-left corner corresponds to $(0,0)$, while the upper-right corner corresponds to $(1,1)$. The pixel coordinates are computed by scaling the NDC by the camera resolution.

$$NDC = \left( \frac{X_i - x_{min}}{x_{max} - x_{min}}, \frac{Y_i - y_{min}}{y_{max} - y_{min}} \right) \tag{4.37}$$

The procedure is shown in Figure 4.9. Once the NDCs are available, the corresponding pixel coordinates are found by multiplying the NDCs by the resolution.

The true coordinates obtained are compared to the coordinates where the corresponding feature is located, according to the tracking algorithm.
The complete procedure is illustrated in Figure 4.10.
It should be pointed out that the procedure introduces intrinsically an error: even if a point is projected on the model and projected back on a camera that kept the exact same state, the projection is not in the exact same position of the original point. However, this error is usually of the order of $10^{-4}px$, several orders of magnitude below the typical error ranges of the algorithms. Hence, it has been considered negligible.

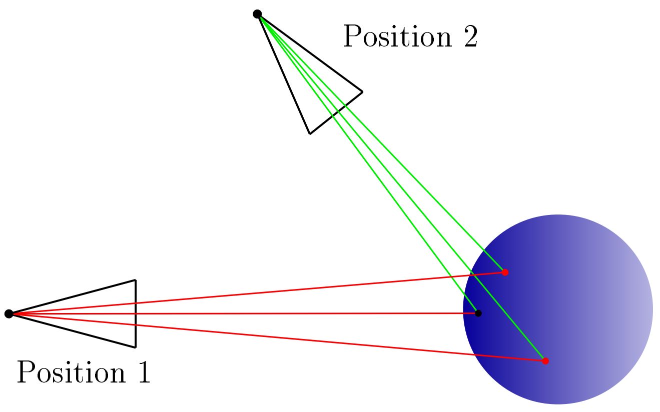


Figure 4.10: Illustration of the process of verification of the matches. When the camera is in position 1, the points are ray casted on the model (red lines). When the camera moves in position 2, the 3D points located on the model are projected back on the image plane (green lines).

# 5 | Analysis of the results

## 5.1. Evaluation method

### 5.1.1. Evaluation metrics

The metrics used to evaluate the performances of the methods used are inspired from Morrell et al. [55]. In particular, the following have been considered:

- The magnitude of the errors committed by the algorithms between two sequential images.
- The distribution of the errors. State estimation algorithms often assume that the distribution of the errors is Gaussian. Hence, an error distribution close to a Gaussian distribution is desirable.
- The percentage of features detected within the asteroid. It was observed that, sometimes, features are extracted in the black space around the asteroid. These points are of course meaningless, hence their amount should be minimum.
- The persistence of the features. For state estimation algorithms, it is desirable that the same features are tracked for a long time, instead of having a large number of new features at each image acquisition.
- The analysis of the outliers. A feature becomes an outlier when, during the tracking, the error with respect to the position in which it was originally located goes above a threshold, set in this work to 5 pixels. It is desirable to have the minimum possible number of outliers.
- The number of points that are matched. In this work, the number of features that are used has been limited to 200 for those algorithms for which the number is directly controllable, and measures have been taken to retain a similar number of keypoints for those algorithms for which such number can't be directly set. However, it can happen that not enough features are extracted, and part of them are removed in the filtering phase. Hence, how many features are effectively available is considered

an important metric.

- The processing time. In general, low computational times are preferred. This is true in particular if algorithms that require frequent image acquisition, such as KLT, are used: in this case, having a low processing time allows a more frequent image acquisition. If feature matching algorithms are used, the acquisition frequency can be lower, and slower algorithms are acceptable.

### 5.1.2. Implementation details

All the performance metrics are computed separately for each value of illumination intensity, image processing frequency and trajectory radius. In this way, the influence of the three factors described in Section 3.3 is investigated.
The distribution of the errors, which inherently includes also the magnitude of the errors, is determined by comparing, for each couple of images processed, the positions in which the features are located in the second image, and the projection on the second image of the features extracted from the first image. The latter is determined through ray casting, described in Chapter 4.3.
The analysis of the outliers is carried out by following the positions in which a point is tracked through the frames. For each feature, the 2D positions in which it is located through the frames are recorded. In addition, the first time a feature is extracted, the 3D location is stored. At frame $n$, the features locations are compared with the projection of the original 3D location on the image plane relative to frame $n$. Hence, a history of the evolution of the error with respect to the original location of a feature is recorded.

## 5.2. Analysis of the results

The results obtained following the procedures described in Chapters 4.1, 4.2 and 4.3 are analyzed in this section.
In the current section, only the results obtained with the asteroid 101955 Bennu are reported for the sake of brevity. The results for the other asteroids are reported in the appendices.

### 5.2.1. Error magnitude and distribution

The average errors for all the algorithms are reported for each illumination intensity, camera distance and image processing step in Table 5.1 for Bennu,. For Itokawa and Eros, results are reported in Appendix B. The average errors are generally low for features matching algorithm. In general, SIFT achieves the best accuracy. BRISK has similar accuracy if the RANSAC algorithm is not used, but the filter appears less effective on BRISK than SIFT. The other traditional algorithms have generally slightly larger errors. With neural network errors are larger. As expected, training a model on an asteroid leads to better performance on that asteroid. In addition, it is observed an improvement of performance also for models not trained with that specific asteroid. In fact, both the models trained on Bennu and Eros have better performance than the pretrained model SPV6 and the Base model. This indicates that SuperPoint is able to generalize well also on previously unseen shapes and textures. In addition, the network trained on the Bennu dataset seems having better performance on Eros than the network trained on the Eros dataset. It is worth to point out that, since SuperPoint downsizes the input image during the preprocessing, the matches are computed on an image that is actually smaller by a scale factor, therefore the errors are multiplied when the original image is considered. In addition, the use of semi-dense descriptors could be a factor that makes the network even less precise, since descriptors are not computed specifically for each features (like traditional algorithms). It was observed that, if the input image is kept at a larger size during the preprocessing, the error is greatly reduced.

SIFT and BRISK appear less robust with respect to variations in illumination, but the effect is small and strongly attenuated by the outliers rejection. The rest of the algorithms result almost unaffected by the illumination intensity.

The distance of the camera from the asteroid has a rather large effect on the average error. This is likely due to the fact that, if the image acquisition rate is constant, a smaller distance results in a larger linear velocity, and hence two consecutive frames will be more similar if acquired from a large distance, and less similar if acquired from a small distance. This gives space to another observation: in case of irregular shape of the body, it might happen that, during the tracking, the average error is greater when some regions of the asteroid are approached. This behavior is reported in Figure 5.1. Bennu has a spheroid-like shape, hence the distance from the surface does not depend on the position around the asteroid. For this reason, the average error along the frames changes due to the noise, but does not have a regular behavior. For the KLT, the spikes are likely due to the features that are occasionally lost as the asteroid rotates, and are re-initialized after a fixed number of frames. This behavior is reported in Figure 5.1a. In the case of Itokawa, the shape is not a spheroid, but is strongly elongated along one axis. The trend of the errors present, especially with ORB, BRISK and KLT, higher values at the beginning, a decrease after the first 50 frames and a growth after 200 frames. The intervals in which the error is larger correspond to the frames in which the camera is positioned above the

Table 5.1: Average norm of the error (in pixels) committed on the matches on the asteroid Bennu. Each table shows the errors for a value of illumination and distance from the asteroid. The first three columns refer to the processing without the RANSAC filter, the last three columns to the processing with the RANSAC filter. Each column refers to a processing frequency (the higher, the less frequent).

(a) Average error (in pixels) committed on the matches on the asteroid Bennu, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.291217 | 0.743627 | 1.03101 | 0.0731754 | 0.118926 | 0.155445 |
| SURF | 0.484176 | 1.73506 | 2.99815 | 0.242747 | 0.356075 | 0.43683 |
| ORB | 0.956696 | 1.18814 | 1.65021 | 0.771635 | 0.807276 | 0.832606 |
| BRISK | 0.305668 | 0.44394 | 0.562708 | 0.269507 | 0.299123 | 0.316808 |
| KLT | 3.10621 | 17.1042 | 34.498 | 2.45052 | 7.96557 | 20.0635 |
| STAR + BRIEF | 0.446622 | 0.594781 | 0.919536 | 0.440504 | 0.525326 | 0.588074 |
| SPV6 | 6.47323 | 7.38328 | 8.17899 | 3.82061 | 4.22307 | 4.7556 |
| SuperPoint Base | 5.34922 | 5.8157 | 6.52354 | 3.823 | 4.09876 | 4.51877 |
| SuperPoint Eros | 5.11747 | 5.5061 | 5.69546 | 3.79096 | 4.05302 | 4.28907 |
| SuperPoint Bennu | 5.14753 | 5.51475 | 5.89969 | 3.76922 | 3.99517 | 4.29543 |

(b) Average error (in pixels) committed on the matches on the asteroid Bennu, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 2.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| IP step | W/o outliers rejection | | | W/ outliers rejection | | |
| | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.206203 | 0.433173 | 0.685509 | 0.140666 | 0.219289 | 0.295027 |
| SURF | 0.424673 | 1.2168 | 2.01292 | 0.303044 | 0.428635 | 0.518706 |
| ORB | 0.783203 | 0.994513 | 1.15083 | 0.678865 | 0.77582 | 0.813902 |
| BRISK | 0.261214 | 0.342329 | 0.447961 | 0.249789 | 0.285481 | 0.301315 |
| KLT | 0.839056 | 2.77351 | 6.84449 | 0.818261 | 2.22574 | 4.36437 |
| STAR + BRIEF | 0.407013 | 0.5631 | 0.767915 | 0.402778 | 0.523516 | 0.607799 |
| SPV6 | 6.91258 | 8.43501 | 9.38469 | 4.49334 | 5.3283 | 6.04738 |
| SuperPoint Base | 5.88752 | 7.05161 | 7.72411 | 4.5657 | 5.39983 | 5.92458 |
| SuperPoint Eros | 5.66651 | 6.56686 | 6.95788 | 4.38574 | 4.98591 | 5.32741 |
| SuperPoint Bennu | 4.98761 | 5.85312 | 6.4039 | 3.79882 | 4.30387 | 4.79177 |

Table 5.1

(c) Average error (in pixels) committed on the matches on the asteroid Bennu, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.459926 | 1.14708 | 2.07697 | 0.0825204 | 0.129573 | 0.175069 |
| SURF | 0.473546 | 1.82762 | 3.32643 | 0.24422 | 0.36659 | 0.465163 |
| ORB | 1.03311 | 1.39962 | 1.9695 | 0.792761 | 0.839975 | 0.885064 |
| BRISK | 0.435053 | 0.561525 | 0.855947 | 0.305671 | 0.340313 | 0.370095 |
| KLT | 3.17982 | 16.5829 | 34.3111 | 2.48651 | 7.47934 | 18.2594 |
| STAR + BRIEF | 0.45168 | 0.600282 | 0.967989 | 0.446705 | 0.535308 | 0.604364 |
| SPV6 | 6.06332 | 7.13475 | 8.2888 | 3.73962 | 4.17553 | 4.7351 |
| SuperPoint Base | 5.11357 | 5.65702 | 6.13489 | 3.68813 | 4.02613 | 4.35203 |
| SuperPoint Eros | 4.93294 | 5.27625 | 5.49798 | 3.72224 | 3.95876 | 4.18565 |
| SuperPoint Bennu | 5.01912 | 5.3874 | 5.80319 | 3.66304 | 3.88311 | 4.18814 |

(d) Average error (in pixels) committed on the matches on the asteroid Bennu, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 2.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.254388 | 0.527442 | 0.986757 | 0.147581 | 0.23546 | 0.311097 |
| SURF | 0.451502 | 1.49278 | 2.26055 | 0.320028 | 0.46313 | 0.551726 |
| ORB | 0.804428 | 1.04065 | 1.27803 | 0.693443 | 0.80106 | 0.856363 |
| BRISK | 0.305121 | 0.446191 | 0.511166 | 0.285335 | 0.33192 | 0.356105 |
| KLT | 0.855882 | 2.81985 | 6.93393 | 0.815626 | 2.17114 | 4.33124 |
| STAR + BRIEF | 0.430399 | 0.602837 | 0.848284 | 0.425907 | 0.552566 | 0.645872 |
| SPV6 | 6.28856 | 7.90026 | 9.22177 | 4.18023 | 5.06226 | 5.99557 |
| SuperPoint Base | 5.57356 | 6.49439 | 7.32907 | 4.25681 | 4.94445 | 5.63262 |
| SuperPoint Eros | 5.38009 | 6.13746 | 6.47982 | 4.11945 | 4.67623 | 5.05809 |
| SuperPoint Bennu | 4.83544 | 5.70156 | 6.26917 | 3.66907 | 4.22166 | 4.67057 |

areas where the shape of the asteroid is longer, and hence closer to the camera. Since the asteroid is longer, the camera has a larger relative speed with respect to the asteroid, and hence the difference between two sequential frame becomes significant. As a result, the errors in these areas are larger, especially with features tracking algorithms, that are more sensitive to large variations between sequential images.

As expected, all the algorithms have worse performance if the (simulated) image acquisition is less frequent. However, this effect appears to be smaller for the CNN, quite significant for some features matching algorithm (SIFT ad SURF), and extreme for the KLT algorithm.

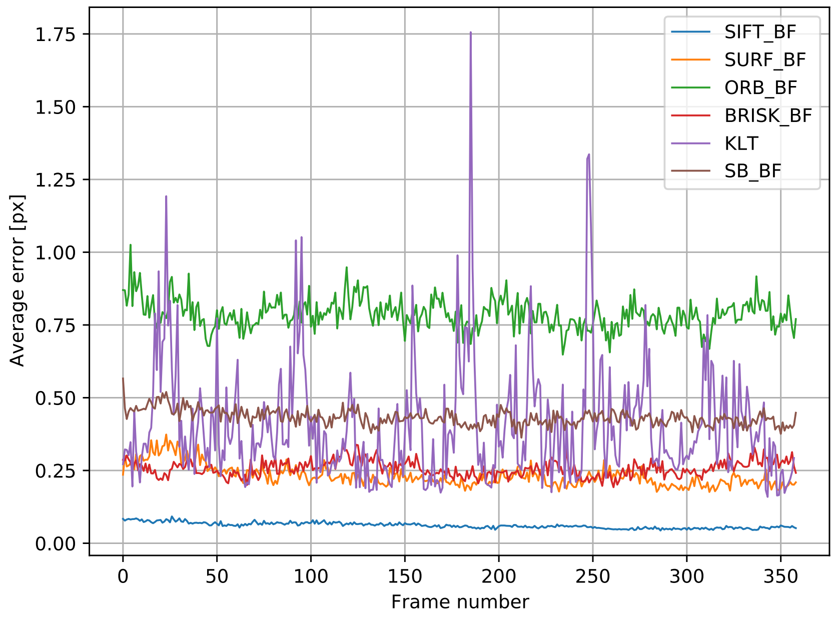
The distributions of the error in $x$ and $y$ directions for all the algorithms are reported in Appendix C. The windows are centered on the mean value, with a width of 3 standard deviations. Hence, a portion of the values are not shown. Traditional algorithms generate a Gaussian error distribution, while the distribution produced by the SuperPoint models present peaks in $x$ direction. Those peaks are located at multiples of the scaling factor used to resize the input images.

The covariance matrices, that give more details about the distribution of the errors, are reported in Appendix A.

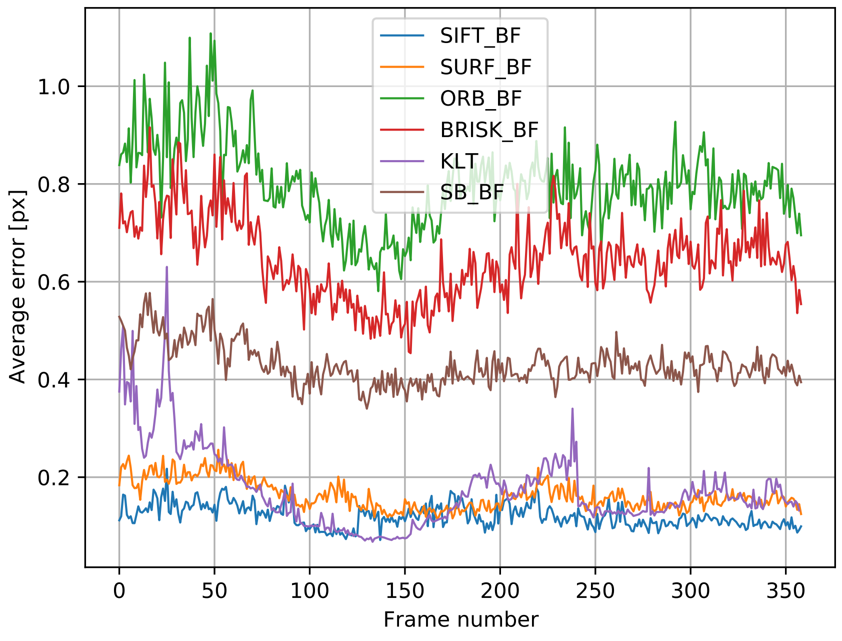
### 5.2.2. Features located within the asteroid

It might happen that algorithms extract points that do not actually belong to the asteroid, but lie in the black area around the body. These points are meaningless, since they do not allow ray casting and hence the computation of the performance metrics, and the tracking is likely to fail.

Table 5.2 reports the percentage of points that are detected within the asteroid Bennu. The same results for Itokawa and Eros are reported in Appendix D.



(a) Bennu



(b) Eros

Figure 5.1: Comparison of the average error history during the tracking, for each frame, for the traditional algorithms.

Table 5.2: Percentage of the features that located within the asteroid Bennu. Each table refers to an illumination and a distance value. The first three columns refer to the processing without the RANSAC filter, the last three with the RANSAC filter. Each column refers to a processing frequency (the higher, the less frequent).

(a) Percentage of features located within the asteroid Bennu, divided by image processing frequency, for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of 1 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 100 | 100 | 100 | 100 | 100 | 100 |
| SURF | 99.6989 | 99.6732 | 99.6318 | 99.7178 | 99.7659 | 99.8251 |
| ORB | 99.5314 | 99.8089 | 99.8882 | 99.5971 | 99.8496 | 99.9215 |
| BRISK | 99.8549 | 99.9494 | 99.97 | 99.8581 | 99.9515 | 99.9758 |
| KLT | 94.8785 | 92.0944 | 94.2793 | 94.9174 | 90.7536 | 91.4766 |
| STAR + BRIEF | 99.9567 | 99.963 | 99.9728 | 99.964 | 99.9692 | 99.9831 |
| SPV6 | 66.5309 | 65.875 | 63.9176 | 66.3147 | 66.0485 | 64.279 |
| SuperPoint Base | 79.8384 | 80.9151 | 81.4316 | 80.1717 | 81.7168 | 82.4844 |
| SuperPoint Eros | 76.2012 | 76.37 | 76.4232 | 76.3823 | 77.0379 | 77.3436 |
| SuperPoint Bennu | 79.2447 | 83.3872 | 84.9007 | 79.3207 | 84.3099 | 86.195 |

(b) Percentage of features located within the asteroid Bennu, divided by image processing frequency, for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of 2 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.9387 | 99.9437 | 99.9451 | 99.9402 | 99.9481 | 99.9513 |
| SURF | 97.3978 | 97.733 | 97.9832 | 97.5395 | 98.1867 | 98.633 |
| ORB | 97.824 | 98.7902 | 99.2085 | 97.9195 | 98.9313 | 99.3678 |
| BRISK | 99.3203 | 99.6841 | 99.8449 | 99.3265 | 99.6925 | 99.8535 |
| KLT | 93.5906 | 83.5251 | 81.8353 | 93.5957 | 83.4625 | 81.1078 |
| STAR + BRIEF | 97.7899 | 98.1833 | 98.5621 | 97.8471 | 98.4392 | 98.9384 |
| SPV6 | 49.1267 | 46.7171 | 44.4567 | 47.2443 | 43.4564 | 40.5215 |
| SuperPoint Base | 64.4889 | 62.771 | 61.9993 | 64.4121 | 62.2869 | 61.532 |
| SuperPoint Eros | 57.4166 | 54.0643 | 52.0778 | 56.7722 | 53.2051 | 50.817 |
| SuperPoint Bennu | 59.6154 | 56.4143 | 56.7044 | 58.4247 | 54.8155 | 55.3712 |

Table 5.2

(c) Percentage of features located within the asteroid Bennu, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.9964 | 99.9959 | 99.9965 | 99.9964 | 99.9959 | 99.9965 |
| SURF | 99.5822 | 99.5493 | 99.464 | 99.6013 | 99.71 | 99.7545 |
| ORB | 98.6253 | 99.4597 | 99.6628 | 98.8154 | 99.5709 | 99.7345 |
| BRISK | 99.4205 | 99.7883 | 99.9178 | 99.4417 | 99.8088 | 99.9378 |
| KLT | 94.5946 | 92.1653 | 93.471 | 94.6341 | 90.6124 | 89.946 |
| STAR + BRIEF | 99.9711 | 99.9718 | 99.9795 | 99.9725 | 99.9767 | 99.9832 |
| SPV6 | 67.8218 | 67.0642 | 64.6797 | 67.7849 | 67.4075 | 64.2941 |
| SuperPoint Base | 80.2182 | 81.1839 | 81.5996 | 80.6549 | 82.2133 | 82.9043 |
| SuperPoint Eros | 78.3643 | 78.8588 | 79.1472 | 78.4232 | 79.6184 | 80.0823 |
| SuperPoint Bennu | 81.1177 | 84.8344 | 86.0128 | 81.4274 | 85.638 | 87.2278 |

(d) Percentage of features located within the asteroid Bennu, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.9789 | 99.9831 | 99.9757 | 99.9788 | 99.9839 | 99.9826 |
| SURF | 96.3783 | 96.9696 | 97.1901 | 96.5428 | 97.4587 | 98.0096 |
| ORB | 96.0899 | 97.8516 | 98.7419 | 96.2203 | 98.0748 | 99.0179 |
| BRISK | 98.284 | 99.173 | 99.6091 | 98.3073 | 99.2091 | 99.6441 |
| KLT | 94.1359 | 86.3787 | 83.6869 | 94.1427 | 86.3236 | 83.1072 |
| STAR + BRIEF | 97.0375 | 97.4034 | 98.0067 | 97.1006 | 97.7144 | 98.445 |
| SPV6 | 49.8125 | 46.6215 | 43.341 | 48.1285 | 43.5971 | 39.2115 |
| SuperPoint Base | 65.5974 | 63.2765 | 61.3782 | 65.3346 | 63.1039 | 60.8891 |
| SuperPoint Eros | 62.3973 | 59.1738 | 56.5892 | 61.6489 | 58.1551 | 55.9291 |
| SuperPoint Bennu | 63.469 | 60.932 | 61.024 | 62.6245 | 59.9308 | 60.2409 |

Most of the traditional algorithms detect the vast majority of the points within the model of the asteroid, hence the computation can be considered valid. With KLT, as expected,

the performance decreases as the image processing step increases, hence less points are correctly detected if images are processed less frequently.
SuperPoint models perform significantly worse than traditional algorithms in these terms. Even with models specifically trained on the asteroid that is being analyzed, the percentage of points detected within the asteroid is significantly lower than traditional algorithms. It is found out, by visual inspection, that the SuperPoint models tend to locate many features on the contour of the asteroid shape, and many of them fall outside of the shape and are located in the black region surrounding the body. A possible explanation is that these points are located on the contour of the asteroid on the preprocessed (and resized) image, and end outside the contour when, in the postprocessing, the image is brought back to the original size and the feature locations are projected on the original image. Hence, the large fraction of invalid points might be caused by the postprocessing, and the points could actually be valid for navigation. However, them being out of the contour of the body prevents from using the ray casting algorithm and evaluating the quality of the results.
The percentage for both the classes of algorithms is similar if the RANSAC filter is or is not applied. No significant difference is observed if the illumination intensity vary. SuperPoint models seem to perform far worse if the distance from the asteroid increases.

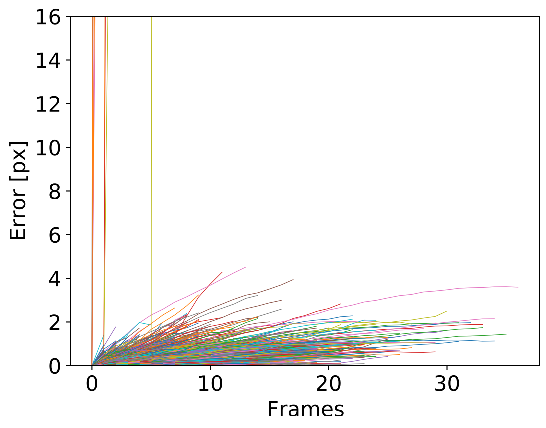
### 5.2.3. Features persistence and outliers

In order to investigate the persistence of the features and analyze the outliers, the history of the error is computed for each features.
Table 5.3 reports the percentage of features that, at some point, become outliers, for each algorithm. The table contains the results obtained for Bennu, for all the distances from the asteroid and the light intensities. The results on Itokawa and Eros are reported in Appendix E.
The percentage of points that become outliers is lower for traditional algorithms, except KLT, and higher for neural networks. In the case of KLT, the question deserves to be investigated. Figure 5.2 shows an example of the behavior of the outliers with SIFT (similar to the other feature matching algorithms) and KLT. Each line correspond to a point. The $x$ axis reports the number of frames through which each features is tracked, and the $y$ axis report the norm of the error with respect to the original position of the features. A features is considered an outlier when the error rises above 5 pixels.
In the case of SIFT (Figure 5.2a), and in general, all the feature matching algorithms, outliers are formed gradually, due to small errors that accumulate during the tracking, or

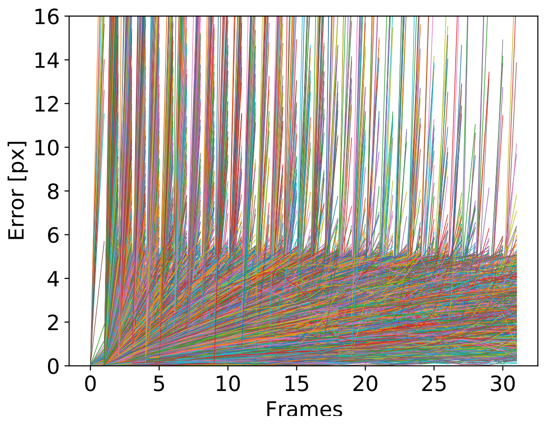
suddenly, due to large errors on a single match, usually happening in the first frames. The latter, as shown, generally happens in the first frames. In the case of KLT (Figure 5.2b), the behavior is different, since outliers form suddenly also after they have been tracked for a large number of frames with a low error. By visual inspection of the matches, it can be seen that KLT tends to produce large errors in areas of transition between light and shadow, where the features tend to disappear from the screen and change their appearance quicker than the areas in which the illumination is in zenith direction. Hence, the KLT algorithm appears to struggle in these areas.

In the case of Bennu, it appears that more points become outliers if the distance from the asteroid grows. This behavior, however, is not confirmed by the results on Itokawa and Eros, in Appendix E. Hence, it probably isn't a systematic feature of the algorithms, and is dictated by some other factor.

An increase in illumination results in a larger percentage of outliers for all the algorithms. The mean persistence of the features, reported in Table 5.4 for Bennu and in Appendix E for Eros and Itokawa, is between 3 and 4 frames for most of the traditional algorithms, except KLT that has a large average persistence, generally between 10 and 20 frames. SuperPoint seems to have a larger persistence than traditional algorithms, that is between 4 and 5 in most of the cases. Variation of the illumination intensity and the radius do not have a significant effect on the average persistence of traditional features matching algorithms. On the other side, the average persistence of the KLT algorithm increases when the distance from the asteroid increases. The same effect is observed for SuperPoint. In this case, the pre-trained model, that was not trained on images from asteroids, has a larger features persistence.



(a) SIFT.



(b) KLT.

Figure 5.2: Comparison of the behavior of the outliers with SIFT and KLT.

Table 5.3: Percentage of points that, at some point, become outliers, for the asteroid Bennu. The first three columns refer to the processing without the RANSAC filter, the last three with the RANSAC filter. Each column refers to a processing frequency (the higher, the less frequent).

(a) Percentage of points that, at some point, become outliers, for the asteroid Bennu, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.065425 | 0.218469 | 0.497454 | 0 | 0.0168209 | 0.0237614 |
| SURF | 0.533977 | 2.00953 | 2.36415 | 0.18356 | 0.36674 | 0.40608 |
| ORB | 1.02308 | 1.06727 | 1.00174 | 0.109111 | 0.102284 | 0.234082 |
| BRISK | 0.0672495 | 0.175824 | 0.113293 | 0.0177651 | 0.0369113 | 0.0127291 |
| KLT | 59.0937 | 91.3311 | 92.582 | 33.4775 | 57.8778 | 77.5448 |
| STAR + BRIEF | 0.0810373 | 0.576132 | 1.85497 | 0.0814996 | 0.082713 | 0.428449 |
| SPV6 | 39.6875 | 43.0508 | 40.7801 | 27.3356 | 33.5907 | 31.0204 |
| SuperPoint Base | 50.4823 | 50.6536 | 50.5703 | 41.3559 | 32.0158 | 42.562 |
| SuperPoint Eros | 47.8827 | 48.3444 | 46.0993 | 36.1204 | 43.299 | 35.5102 |
| SuperPoint Bennu | 36.1371 | 42.623 | 38.4354 | 29.5203 | 32.9787 | 30.9963 |

(b) Percentage of points that, at some point, become outliers, for the asteroid Bennu, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 1.56582 | 1.98831 | 2.17735 | 1.40833 | 1.22639 | 0.923498 |
| SURF | 1.4765 | 2.97892 | 3.7129 | 0.618622 | 0.906217 | 0.989144 |
| ORB | 1.2139 | 1.38889 | 1.74448 | 0.393529 | 0.447227 | 0.443459 |
| BRISK | 0.0719709 | 0.18582 | 0.347264 | 0.0468637 | 0.0792393 | 0.133463 |
| KLT | 30.1314 | 74.7288 | 86.3887 | 24.0065 | 36.9777 | 37.5455 |
| STAR + BRIEF | 0.519931 | 1.20846 | 4.33018 | 0.172117 | 0.445765 | 1.81311 |
| SPV6 | 37.9147 | 32.1244 | 32.5581 | 29.6512 | 24.2105 | 23.2877 |
| SuperPoint Base | 41.6667 | 39.2638 | 42.3841 | 39.4444 | 41.1392 | 32.3308 |
| SuperPoint Eros | 45.4082 | 44.8454 | 40.3509 | 30.8989 | 32.1839 | 32.1429 |
| SuperPoint Bennu | 30.8571 | 27.4194 | 23.3333 | 26.3473 | 17.6471 | 23.2877 |

Table 5.3

(c) Percentage of points that, at some point, become outliers, for the asteroid Bennu, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.188147 | 0.617284 | 1.19626 | 0.00448612 | 0.0413189 | 0.0478354 |
| SURF | 0.464897 | 2.16249 | 2.67094 | 0.258756 | 0.518672 | 0.554939 |
| ORB | 0.914867 | 1.37261 | 1.54827 | 0.275786 | 0.143421 | 0.584795 |
| BRISK | 0.141063 | 0.216752 | 0.310252 | 0.037176 | 0.062898 | 0.0430849 |
| KLT | 62.6222 | 90.3978 | 91.5858 | 37.168 | 55.8444 | 74.8123 |
| STAR + BRIEF | 0.082713 | 0.699913 | 1.49813 | 0.0838926 | 0.352734 | 0.647549 |
| SPV6 | 41.8006 | 42.1769 | 36.8821 | 30.1325 | 31.0714 | 30.4721 |
| SuperPoint Base | 44.8071 | 45.614 | 43.1734 | 33.218 | 33.9768 | 34.1564 |
| SuperPoint Eros | 48.7395 | 48.5623 | 48.0427 | 35.7143 | 40.0722 | 36.8217 |
| SuperPoint Bennu | 37.3041 | 42.4051 | 47.0149 | 29.5139 | 31.9703 | 39.3443 |

(d) Percentage of points that, at some point, become outliers, for the asteroid Bennu, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 1.52849 | 1.90316 | 2.50508 | 1.37186 | 1.16392 | 0.849013 |
| SURF | 1.55218 | 3.29884 | 4.24087 | 0.71178 | 0.930295 | 0.992467 |
| ORB | 0.349498 | 1.69861 | 2.26316 | 0 | 0.698649 | 0.518135 |
| BRISK | 0.117874 | 0.306321 | 0.322465 | 0.0715538 | 0.0632067 | 0.0364609 |
| KLT | 39.4053 | 76.1813 | 85.742 | 29.1379 | 36.6654 | 36.4214 |
| STAR + BRIEF | 0.167504 | 1.12994 | 4.41558 | 0.169492 | 0.710227 | 2.43572 |
| SPV6 | 37.395 | 36.612 | 33.1492 | 31.1765 | 27.9221 | 23.6025 |
| SuperPoint Base | 44.5545 | 36.5854 | 38.9535 | 38.3333 | 27.2152 | 30.3704 |
| SuperPoint Eros | 43.5897 | 45.4545 | 37.037 | 38.3234 | 33.5227 | 28 |
| SuperPoint Bennu | 39.8773 | 36.9942 | 32.1637 | 30.7692 | 30.2632 | 21.7391 |

Table 5.4: Mean persistence of the features, for the asteroid Bennu. The first three columns refer to the processing without the RANSAC filter, the last three with the RANSAC filter. Each column refers to a processing frequency (the higher, the less frequent).

(a) Mean persistence of the features, for the asteroid Bennu, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.94177 | 3.79649 | 3.39974 | 3.94184 | 3.79647 | 3.40228 |
| SURF | 3.6922 | 3.27795 | 2.86263 | 3.68053 | 3.27336 | 2.86507 |
| ORB | 3.92681 | 3.07794 | 2.59321 | 3.38898 | 2.86158 | 2.49438 |
| BRISK | 3.71815 | 3.28029 | 2.87689 | 3.69853 | 3.27078 | 2.87347 |
| KLT | 14.1018 | 4.08481 | 3.2699 | 13.7167 | 3.74924 | 2.27002 |
| STAR + BRIEF | 3.90276 | 4.08395 | 3.82125 | 3.88753 | 3.81141 | 3.65381 |
| SPV6 | 10.3781 | 6.48136 | 5.11702 | 10.3495 | 6.74517 | 4.98776 |
| SuperPoint Base | 4.14791 | 3.18954 | 2.68821 | 3.61017 | 3.03953 | 2.71074 |
| SuperPoint Eros | 4.5114 | 3.33113 | 2.89007 | 3.85619 | 3.20619 | 2.70612 |
| SuperPoint Bennu | 3.01246 | 2.63279 | 2.44218 | 2.80443 | 2.65957 | 2.37269 |

(b) Mean persistence of the features, for the asteroid Bennu, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.84404 | 3.74241 | 3.41719 | 3.84129 | 3.74761 | 3.42184 |
| SURF | 3.54201 | 3.092 | 2.72689 | 3.50814 | 3.06906 | 2.72328 |
| ORB | 3.81708 | 3.21338 | 2.7255 | 3.65282 | 2.97764 | 2.60255 |
| BRISK | 3.69603 | 3.44897 | 3.03868 | 3.68035 | 3.43027 | 3.03021 |
| KLT | 22.6188 | 10.2808 | 5.45712 | 21.7785 | 9.45451 | 4.97273 |
| STAR + BRIEF | 3.94974 | 3.88973 | 3.79838 | 3.92083 | 3.91382 | 3.8145 |
| SPV6 | 14.9336 | 9.73575 | 7.15116 | 16.5814 | 9.44737 | 7.59589 |
| SuperPoint Base | 6.55556 | 4.53374 | 3.65563 | 4.97222 | 3.43671 | 2.84211 |
| SuperPoint Eros | 6.07653 | 5.30928 | 3.63743 | 5.0618 | 3.88506 | 3.25714 |
| SuperPoint Bennu | 4.77143 | 3.23656 | 2.59333 | 4.46108 | 3.01961 | 2.59589 |

Table 5.4

(c) Mean persistence of the features, for the asteroid Bennu, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 4.04408 | 3.76831 | 3.30451 | 4.0467 | 3.77332 | 3.3126 |
| SURF | 3.67603 | 3.24373 | 2.81603 | 3.66653 | 3.24142 | 2.81842 |
| ORB | 3.82135 | 2.9682 | 2.51138 | 3.31412 | 2.75152 | 2.42495 |
| BRISK | 3.38786 | 2.93126 | 2.61684 | 3.36223 | 2.92617 | 2.61468 |
| KLT | 13.264 | 4.23938 | 3.40494 | 12.6039 | 3.87061 | 2.35387 |
| STAR + BRIEF | 3.69644 | 3.66317 | 3.5412 | 3.63255 | 3.54145 | 3.40888 |
| SPV6 | 10.2315 | 6.48299 | 5.20152 | 9.9404 | 6.28571 | 5.06438 |
| SuperPoint Base | 3.97329 | 2.93684 | 2.61993 | 3.73702 | 2.84556 | 2.61728 |
| SuperPoint Eros | 4.45098 | 3.19489 | 2.6548 | 3.71769 | 2.85921 | 2.6124 |
| SuperPoint Bennu | 3.06583 | 2.75 | 2.39925 | 3.01736 | 2.6803 | 2.34836 |

(d) Mean persistence of the features, for the asteroid Bennu, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.83461 | 3.71509 | 3.39145 | 3.83392 | 3.72244 | 3.4089 |
| SURF | 3.50963 | 3.05336 | 2.70614 | 3.47789 | 3.03701 | 2.70704 |
| ORB | 3.71909 | 3.19904 | 2.65053 | 3.46526 | 2.99301 | 2.54865 |
| BRISK | 3.70321 | 3.26699 | 2.85035 | 3.67994 | 3.2451 | 2.84115 |
| KLT | 20.057 | 9.15252 | 5.14026 | 19.1078 | 8.35911 | 4.57856 |
| STAR + BRIEF | 3.75712 | 3.88559 | 3.82987 | 3.73559 | 3.8196 | 3.75913 |
| SPV6 | 14.7983 | 9.55738 | 6.90608 | 16.9294 | 10.2078 | 6.51553 |
| SuperPoint Base | 6.06931 | 4.03049 | 3.59884 | 4.48889 | 3.32278 | 3.11111 |
| SuperPoint Eros | 8.62051 | 5.26136 | 3.25309 | 5.12575 | 3.54545 | 2.85333 |
| SuperPoint Bennu | 4.33742 | 3 | 2.73099 | 4.10059 | 2.93421 | 2.56522 |

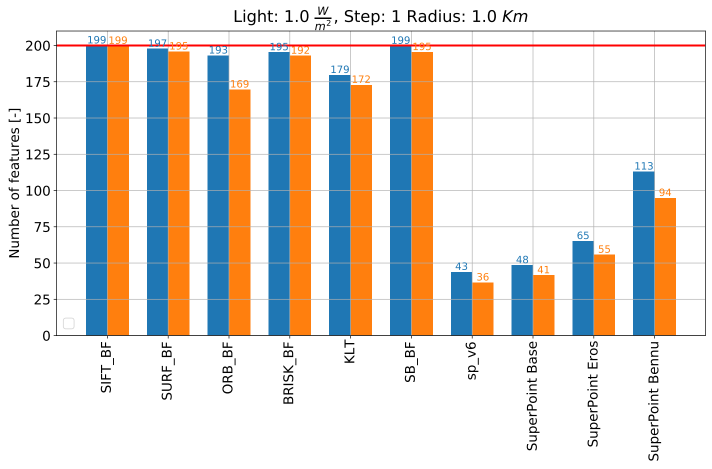
### 5.2.4. Number of points tracked

The RANSAC algorithm can significantly reduce the number of points available. In this work, the maximum number of features followed for each couple of frames has been limited to 200 for performance reasons. Figure 5.3 reports the average number of features tracked, rounded to the closest integer, for each algorithm, both the filtered and not filtered cases and for all the values of illumination and processing frequency. The results refer to a dataset containing only images of Bennu. Only the variation of points matched with respect to the image processing step is shown. In general, the loss of features due to outliers rejection is not large. As expected, more points are lost as the image processing step grows, and hence images are processed less frequently, since the accuracy tends to decrease and the filter removes more points, as explained in Section 5.2.1.
Traditional algorithms track a number of points close to the limit, particularly when the processing of the images is frequent. On the contrary, SuperPoint models track a relatively small number of points, and even for the SuperPoint Bennu model, specifically trained for Bennu, the number of points is low.

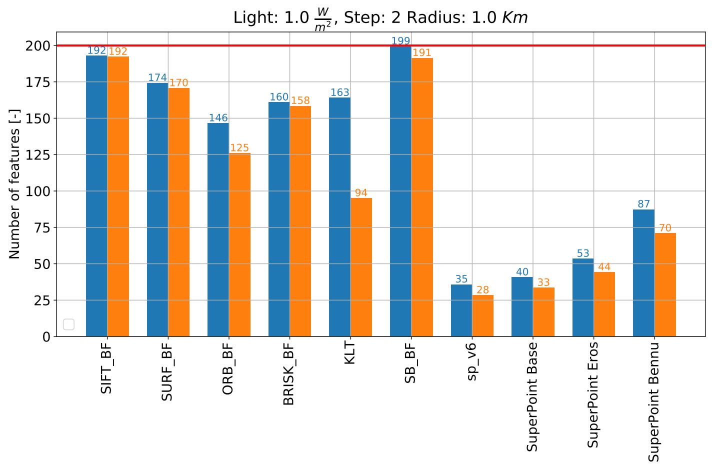
### 5.2.5. Processing time

Figure 5.4 shows the processing time required by the algorithms. For features matching algorithms, the time was decomposed in extraction and matching time. The extraction time includes the extraction of both features and descriptors, while the matching time includes the time required to match all the descriptors in an image. For features tracking algorithms, only the time required to track all the features between two sequential images has been considered. The extraction time has not been included, since it is less frequent with features tracking algorithms.
It is worth to point out that the results here presented should be interpreted as an approximate comparisons between the algorithms, and not as indications of the absolute processing time, that will likely be different in a real application due to different implementation, different programming language and different hardware.
Some observations can be made:

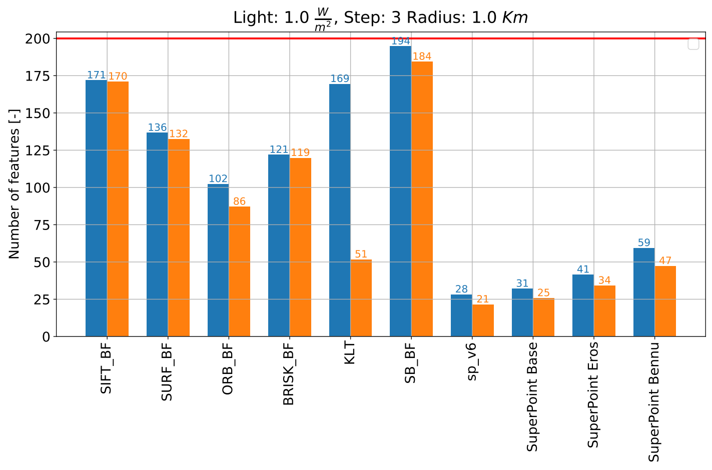
- With features matching algorithms, the processing time is far greater than with features tracking algorithms. Since the matching time is relatively low, this difference is most likely due to the computation of the descriptors.
- Using binary descriptors is faster. BRISK, ORB and STAR + BRIEF use binary descriptors, and have a lower matching time than SIFT and SURF. The comparison with SuperPoint is not really meaningful, since the number of features matched is



(a) Light: $5.0\frac{W}{m^2}$, Step: 1, Radius: $1.0Km$



(b) Light: $5.0\frac{W}{m^2}$, Step: 2, Radius: $1.0Km$



(c) Light: $5.0\frac{W}{m^2}$, Step: 3, Radius: $1.0Km$

Figure 5.3: Number of points matched by the algorithms. The horizontal red line represents the maximum number of matches kept (limited to 200 for performance reasons)

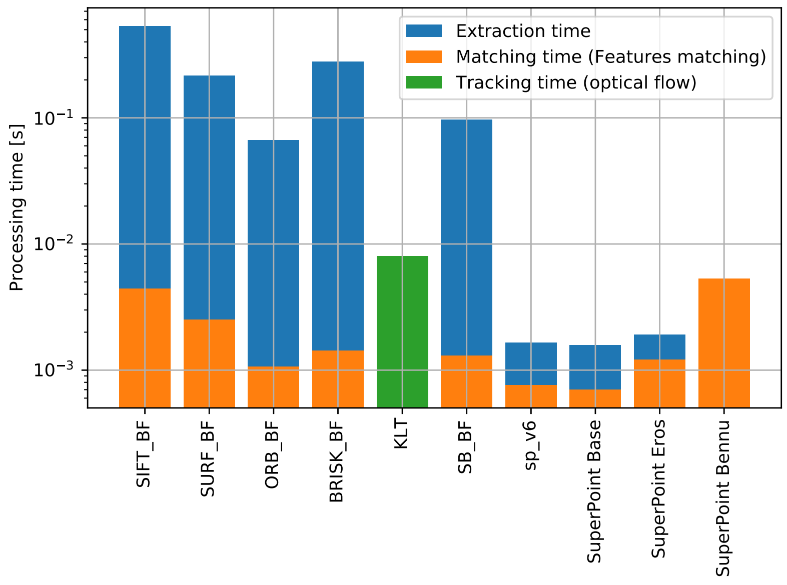


Figure 5.4: Image processing time, for each algorithm. For features matching algorithm, the total time is decomposed in extraction and matching time. For KLT, only the tracking time is considered.

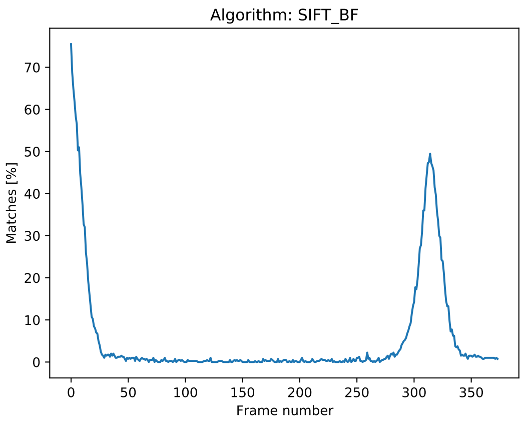
very different. According to Calonder et al. [20] the Hamming distance, used to compare binary descriptors, is more efficient than the L2 norm, used to compare floating point descriptors, hence a faster matching is expected. However, it was not possible to verify if the computation of the descriptor is faster.

- Neural networks result extremely fast, but this can be at least partially attributed to the fact that the quantity of features extracted with neural networks is low, as explained in Section 5.2.4.

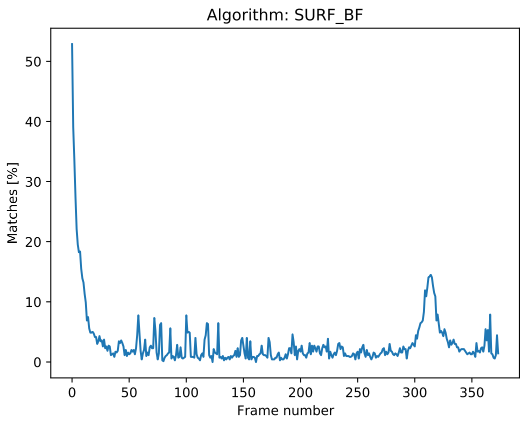
## 5.3. Loop closure performance

Loop closure indicates the capacity of an algorithm of determining when a full orbit around the asteroid has been concluded. A strategy based on descriptors has been developed to detect loop closure. The descriptors from the first image are kept in memory. At each frame, the descriptors extracted from the current frame are matched with the descriptors from the first frame, and the matches are filtered using Lowe's test (Lowe [50]). The ratio between the number of matches and the total number of descriptors (and hence of features) from the first frame is computed. The ratio, high at the beginning, will decrease when no correspondences are found any more. When the ratio rises again, it means that more "good" matches (according to Lowe's test) are found, and hence the loop is closing.

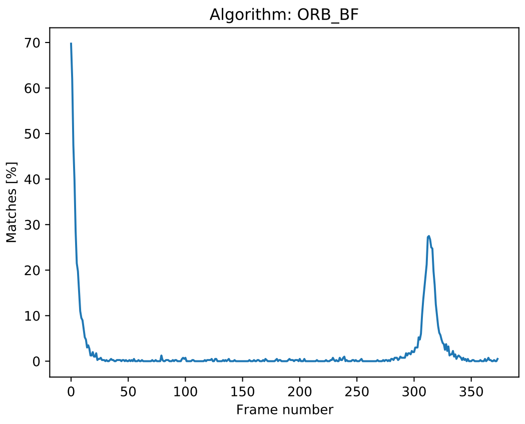
This approach requires the computation of the descriptors, hence it can't be used with features tracking methods. Since the method developed to check the loop closure requires the computation of the descriptors, results for the KLT algorithm are not available.
At each frame, the descriptors obtained from that frame are matched with the descriptors obtained from the first frame. A peak is expected when the trajectory around the body closes, since the initial frame is theoretically encountered again.
The performance of the algorithms in the detection of the loop closure is reported in Figure 5.5. All the algorithms shows, in different measures, a peak where the loop is closed. The intensity of the peak is lower with respect to the beginning, since variations in position, orientation and illumination occur and the frames will not correspond exactly. The analysis was done only for traditional algorithms, since they appeared to have better characteristics in the previous sections.



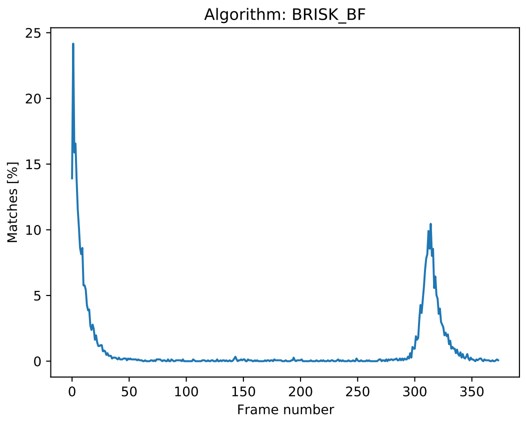
(a) SIFT with Brute Force matcher.



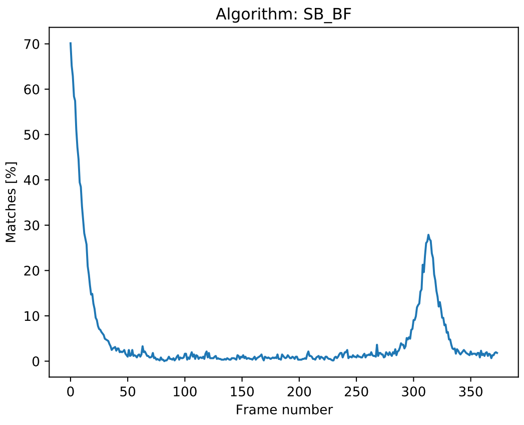
(b) SURF with Brute Force matcher.



(c) ORB with Brute Force matcher.



(d) BRISK with Brute Force matcher.

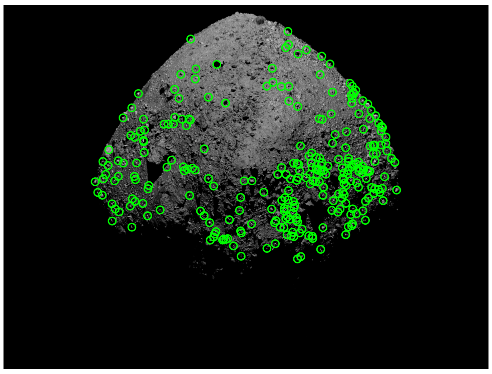


(e) STAR + BRIEF with Brute Force matcher.

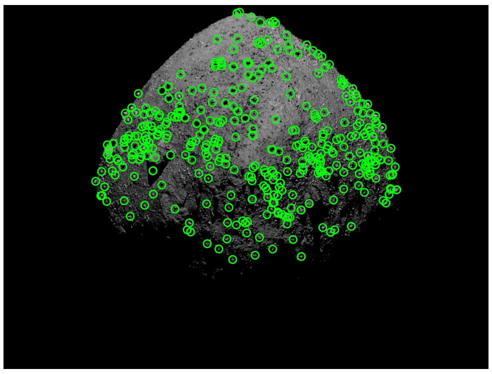
Figure 5.5: Comparison of the loop closure performance of the algorithms.

## 5.4. Detections

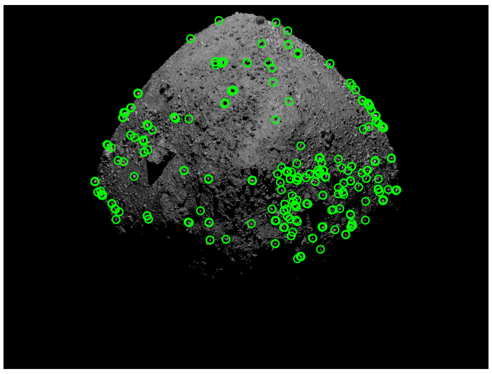
Examples of the features extracted by the different algorithms are here provided.



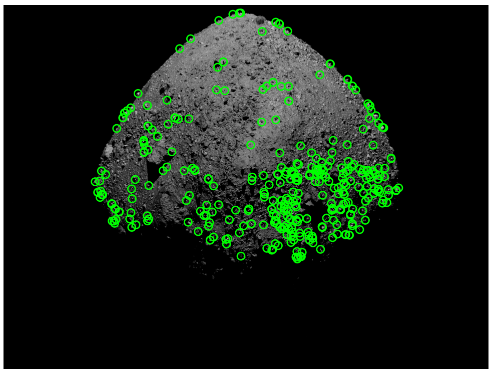
(a) SIFT.



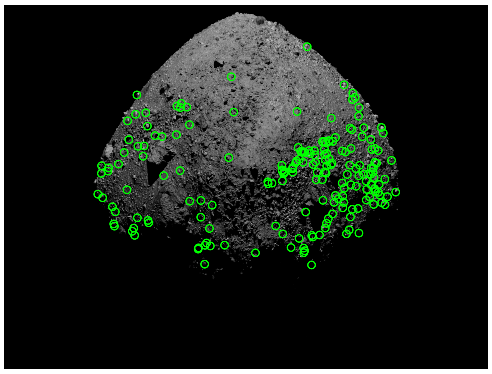
(b) SURF.



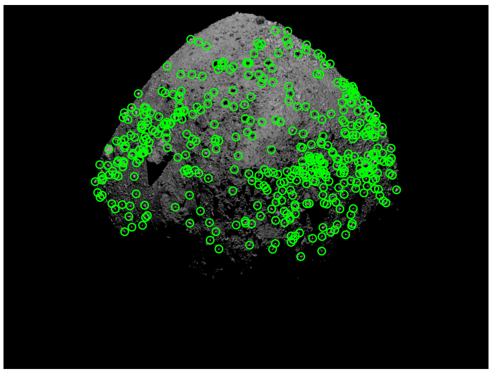
(c) ORB.



(d) BRISK.

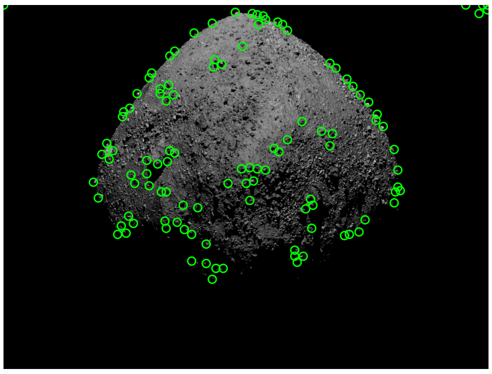


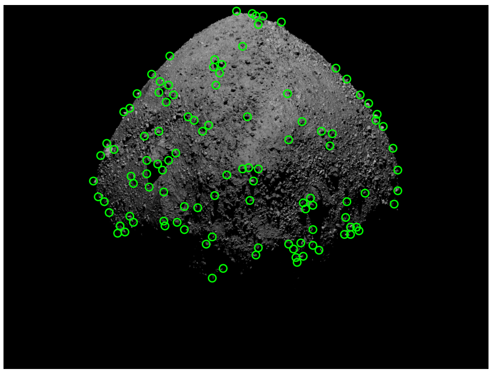
(e) KLT.

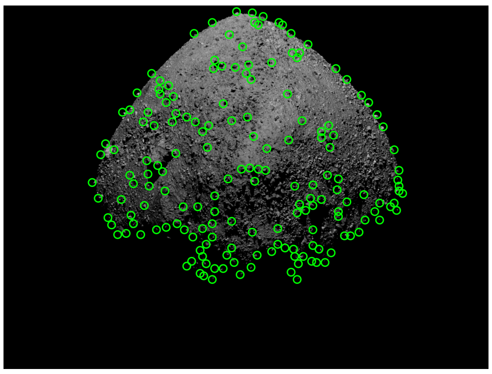


(f) STAR + BRIEF.

Figure 5.6: Detections on the asteroid Bennu.


(g) SPV6.


(h) SuperPoint Base.


(i) SuperPoint Eros.

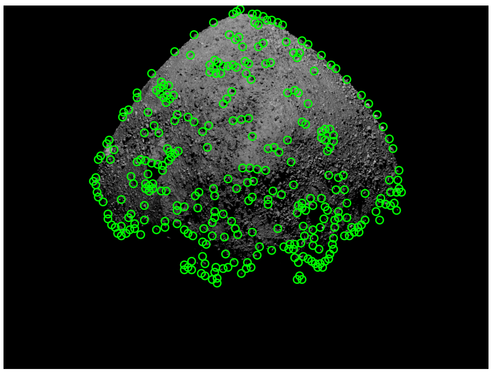

(j) SuperPoint Bennu.

Figure 5.6: Detections on the asteroid Bennu.

# 6 | Conclusions

## 6.1. Final considerations

The performance of traditional algorithms has been compared with the performance of the SuperPoint neural network. The following performance metrics have been considered:

- Magnitude of the error between consecutive frames.
- Distribution of the errors.
- Percentage of valid points, i.e. actually detected within the asteroid model and not in the surrounding black area.
- Persistence of the features.
- Analysis of the outliers.
- Number of features matched between consecutive frames.
- Processing time.
- Loop closure.

Regarding those points, the following considerations have been made:

- Feature matching algorithms maintain the error at lower levels with respect to the neural network. SIFT and BRISK have the best performance in these terms, followed by STAR + BRIEF, SURF and ORB. KLT has higher errors than feature matching algorithms, and the performances greatly degrade if the (simulated) image acquisition frequency decreases. However, it has to be pointed out that errors with KLT tend to be higher in those areas in which the illumination gradient is stronger, such as transition areas between light and shadow. SuperPoint models carry large errors, that however improve if larger images are given as input to the network.
- The distribution of the errors is Gaussian for all the traditional algorithms. For neural network, the distributions present peaks at the multiples of the scale factor used to resize the images in preprocessing.

- With traditional algorithms, most of the features are located within the contour of the body, and hence can be considered "valid". With SuperPoint, many features fall outside of the contour. This is likely due to the projection of the feature locations, computed for the downsized image, on the original image.
- The persistence of traditional algorithms, except KLT, is between 3 and 4 frames, and is not much affected by factors such as illumination and the distance from the asteroid. KLT has an high features persistence, that increases as the distance from the asteroid increases. Neural networks have a larger persistence than traditional features matching techniques, but a lower persistence than KLT. The persistence is larger for the pre-trained model, that is not trained on asteroid images.
- With traditional features matching algorithms, only a small percentage of the points become outliers during the tracking. With SuperPoint, many features become outliers, mostly in the first frames in which they are tracked. With KLT, many points become outliers, but this happens both at the beginning of the tracking and after many frames. The reason is that KLT tends to produce large errors, and hence outliers, in areas of transition from light to shadow. With SuperPoint, a vast majority of the features become outliers in the first frames.
- Traditional algorithms extract and track a number of features close to the maximum allowed. On the contrary, SuperPoint extracts a small quantity of features also with specific training.
- Features matching algorithms take a significant time to extract and process the features. Binary descriptor seem to reduce the processing time with respect to floating point descriptors. SIFT is the slowest algorithm, followed by BRISK, SURF, ORB and STAR + BRIEF. KLT is faster, since no descriptors are computed. SuperPoint models are extremely fast.
- All the traditional algorithms, except SURF, are able to detect clearly when the loop around the asteroid is closed.

Table 6.1 summarizes the positive and negative aspects emerged for all the algorithms.
If the primary objective is the accuracy, a small error and a small number of outliers are desirable. In this case, the choice would fall on SIFT or BRISK, preferring SIFT if having a minimum error is more important or BRISK if having less outliers is more important. These algorithms require a larger computational time than the others, however since with features matching methods a low image acquisition frequency is acceptable having computational time that is larger by a few tenths of a second shouldn't be an

issue. Eventual robustness issues, for instance to light and distance, appear to be well addressed by the outliers rejection algorithm. However, these methods have a low feature persistence.
Other feature matching algorithms, such as SURF, ORB and STAR + BRIEF, tend to be less accurate both in terms of average error and outliers. SURF, ORB and STAR + BRIEF show improved robustness with respect to variations in the distance from the asteroid. However, the effect is less evident after the outliers rejection algorithm is applied.
If a large feature persistence is mandatory, the choice would fall on KLT. It is expected that KLT would have also better accuracy characteristics with more frequent image processing. However, the study shows poor performance in those areas with a strong illumination gradient, such as transition areas from light to shadow.
The SuperPoint models performed poorly with respect to traditional algorithms. However, it is safe to say that the performances have been greatly affected by the resize of the images in postprocessing. In particular, this causes a growth of the errors (that are multiplied) and is likely responsible for the large percentage of feature points that fall outside the contour of the asteroid. In particular, the latter implies that the performances relative of many feature points that were possibly valid could not have been computed.

Table 6.1: Positive and negative aspects emerged for the algorithms.

| Algorithm | Pros | Cons |
|---|---|---|
| SIFT | • Lowest matching errors.<br>• Quite low percentage of outliers.<br>• High percentage of valid points.<br>• Gaussian errors.<br>• Quite robust with respect to variations in acquisition frequency. | • Largest computational time.<br>• Low feature persistence.<br>• Not very robust with respect to variations in distance.<br>• Not very robust with respect to variations in illumination. |
| SURF | • Low matching errors.<br>• Low percentage of outliers.<br>• High percentage of valid points.<br>• Gaussian errors.<br>• Quite robust with respect to variations in distance.<br>• Quite robust with respect to variations in illumination.<br>• Quite robust with respect to variations in acquisition frequency. | • Quite large computational time.<br>• Low feature persistence. |
| ORB | • Low matching errors.<br>• Low percentage of outliers.<br>• High percentage of valid points.<br>• Gaussian errors.<br>• Small computational time (between features matchers).<br>• Quite robust with respect to variations in distance.<br>• Quite robust with respect to variations in illumination.<br>• Quite robust with respect to variations in acquisition frequency. | • Low feature persistence. |
| BRISK | • Very low matching errors.<br>• Lowest percentage of outliers.<br>• High percentage of valid points.<br>• Gaussian errors.<br>• Quite robust with respect to variations in distance.<br>• Quite robust with respect to variations in illumination.<br>• Very robust with respect to variations in acquisition frequency. | • Low feature persistence.<br>• Quite large computational time. |
| KLT | • Highest feature persistence.<br>• Low computational time.<br>• Gaussian errors.<br>• Quite high percentage of valid points.<br>• Very robust with respect to variations in illumination. | • Almost all the points become outliers.<br>• Large errors in transition areas between light and shadow.<br>• Requires high image acquisition frequency.<br>• Not robust with respect to variations in distance.<br>• Not robust with respect to variations in acquisition frequency. |

Table 6.1

| | | |
|---|---|---|
| STAR + BRIEF | • Low matching errors.<br>• Low computational time.<br>• High percentage of valid points.<br>• Gaussian errors.<br>• Quite low percentage of outliers.<br>• Quite robust with respect to variations in distance.<br>• Quite robust with respect to variations in acquisition frequency. | • Low feature persistence. |
| SPV6 | • Very low computational time.<br>• Large features persistence.<br>• Quite robust with respect to variations in distance.<br>• Very robust with respect to variations in illumination.<br>• Very robust with respect to variations in acquisition frequency. | • Very high errors between matches.<br>• Very high percentage of outliers.<br>• Non Gaussian errors.<br>• High percentage of invalid points. |
| SuperPoint Base | • Very low computational time.<br>• Large features persistence.<br>• Quite robust with respect to variations in distance.<br>• Very robust with respect to variations in illumination.<br>• Very robust with respect to variations in acquisition frequency. | • High errors between matches.<br>• Very high percentage of outliers.<br>• Non Gaussian errors.<br>• High percentage of invalid points. |
| SuperPoint Bennu | • Very low computational time.<br>• Large features persistence.<br>• Quite robust with respect to variations in distance.<br>• Very robust with respect to variations in illumination.<br>• Very robust with respect to variations in acquisition frequency. | • High errors between matches.<br>• Very high percentage of outliers.<br>• Non Gaussian errors.<br>• High percentage of invalid points. |
| SuperPoint Eros | • Very low computational time.<br>• Large features persistence.<br>• Quite robust with respect to variations in distance.<br>• Very robust with respect to variations in illumination.<br>• Very robust with respect to variations in acquisition frequency. | • High errors between matches.<br>• Very high percentage of outliers.<br>• Non Gaussian errors.<br>• High percentage of invalid points. |

## 6.2. Future work

During the analysis of the results, two critical aspects emerged regarding KLT and SuperPoint. With KLT, it was noticed that the errors grow in transition areas from light to shadow, likely due to the large illumination gradient. In addition, features tend to disappear from the frame, and hence their number is reduced progressively. Both the behaviors should be addressed to improve performances. With SuperPoint, it emerged that the resize factor in preprocessing and postprocessing was more important than expected, and had a large impact on almost all the performance metrics. Keeping the size of the input image large results in better performances. On the other side, using large images results in a large computational and memory cost. Both the benefits of using larger images and the performances of position estimation algorithms on small images should be investigated to address a solution.
In this work, only the detection and the tracking of reference points has been considered. However, in a real application, these results will be at one point used by the navigation software to estimate the spacecraft state. This part of the navigation task, not considered in the work, influences inevitably also the feature tracking procedure. Depending on the algorithm, in fact, the desirable characteristics of the features used for navigation might be different. In this work, the number of correspondences between features determined from a couple of images has been limited to 200, for speed and memory reasons. In case a larger number is required by the navigation software, two observations arise:

- In this work, for features matching algorithms, the 200 best matches (according to the performance metrics measured by the matcher) were retained, and the rest discarded. For KLT, the best 200 features (according to the performance metrics measured by the extractor) were retained, and the rest discarded. Hence, in both the cases, only the 200 best correspondences are available at the end, at least according to the metrics computed by the algorithms. If more features and matches are retained, these can be expected to have a worse quality, hence it is likely that a degradation of the performance will occur with respect to the results of this work.

- In the case of features matching algorithms, if more features are used, the matching time is expected to grow. If the Brute Force matcher is used, with $n$ features and descriptors extracted from each image, $n^2$ comparison between the descriptors are required. This might become computationally expensive if large numbers of features are involved. Approximate matchers, such as the FLANN-Based matcher, can speed-up the computations, but the quality of the matches is likely to be lower, and a further degradation of the performance can be expected.

Hence, it is plausible that the performances will be worse if large quantities of features are involved. A trade-off between the benefits brought by a larger number of features and the performance degradation expected might be necessary.
In this work, the OpenCV implementation of the traditional image processing algorithms has been used. The OpenCV library is developed in C++ [1]. In a real application, however, the implementation of the algorithms might be different from the implementation provided in OpenCV, and hence the speed of the computations might differ. The same considerations are valid for the neural networks: different architectures and/or different frameworks are likely to result in different performances. In addition, hardware has to be considered: space-hardened avionics, or radiation-hardened avionics, is designed to operate reliably in the space environment and to consume a small amount of power, but in general is much slower than modern common consumer hardware, and even more so of high performance computing hardware. According to Mehlitz and Penix [54], the performance gap between radiation hardened and consumer hardware is at least of an order of magnitude. For this reason it can be expected that, other conditions being equal, image processing will be much slower if performed on-board.

[1] `https://github.com/opencv/opencv/wiki/Coding_Style_Guide`

# Bibliography

[1] Aws developer guide - model fit: Underfitting vs. overfitting. URL `https://docs.aws.amazon.com/machine-learning/latest/dg/model-fit-underfitting-vs-overfitting.html`.

[2] 101955 bennu. URL `https://sbmt.jhuapl.edu/Object-Template.php?obj=77`.

[3] Iau resolution b5. URL `https://www.iau.org/static/resolutions/Resolution_GA26-5-6.pdf`.

[4] Opencv documentation - introduction to sift (scale-invariant feature transform), . URL `https://docs.opencv.org/4.x/da/df5/tutorial_py_sift_intro.html`.

[5] Opencv documentation - shi-tomasi corner detector & good features to track, . URL `https://docs.opencv.org/4.5.2/d4/d8c/tutorial_py_shi_tomasi.html`.

[6] Opencv documentation - harris corner detection, . URL `https://docs.opencv.org/4.5.2/d4/d8c/tutorial_py_shi_tomasi.html`.

[7] Opencv documentation - feature matching, . URL `https://docs.opencv.org/4.x/dc/dc3/tutorial_py_matcher.html`.

[8] C. C. Aggarwal. *Neural Networks and Deep Learning*. Springer, 2018. ISBN 978-3-319-94462-3. doi: 10.1007/978-3-319-94463-0.

[9] M. Agrawal, K. Konolige, and M. R. Blas. Censure: Center surround extremas for realtime feature detection and matching. In D. Forsyth, P. Torr, and A. Zisserman, editors, *Computer Vision – ECCV 2008*, pages 102–115, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[10] P. Alcantarilla, J. Nuevo, and A. Bartoli. Fast explicit diffusion for accelerated features in nonlinear scale spaces. In *BMVC*, 2013. doi: http://dx.doi.org/10.5244/C.27.13.

[11] C. Asmail. Bidirectional scattering distribution function (bsdf): A systematized

bibliography. *National Institute of Standards and Technology Journal of Research*, 96:215–223, 03 1991. doi: 10.6028/jres.096.010.

[12] O. S. Barnouin et al. Digital terrain mapping by the osiris-rex mission. *Planetary and Space Science*. doi: https://doi.org/10.1016/j.pss.2019.104764.

[13] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision - ECCV 2006*, 3951:404–417, 2006. doi: https://doi.org/10.1007/11744023_32.

[14] C. e. a. Bennett. A high-resolution global basemap of (101955) bennu. *Icarus*, 357:113690, 2021. ISSN 0019-1035. doi: https://doi.org/10.1016/j.icarus.2020.113690. URL `https://www.sciencedirect.com/science/article/pii/S0019103520300816`.

[15] E. Beshore et al. The osiris-rex asteroid sample return mission. In *2015 IEEE Aerospace Conference*, pages 1–14, 2015. doi: 10.1109/AERO.2015.7118989.

[16] B. J. Bos et al. Touch And Go Camera System (TAGCAMS) for the OSIRIS-REx Asteroid Sample Return Mission. *Space Science Reviews*, 214(1):37, Feb. 2018. doi: 10.1007/s11214-017-0465-2.

[17] J. Bouguet. Pyramidal implementation of the lucas kanade feature tracker. *Intel Corporation, Microprocessor Research Labs*, 2000.

[18] B. Burley. Extending the disney brdf to a bsdf with integrated subsurface scattering. 2015.

[19] Calonder et al. Brief: Computing a local binary descriptor very fast. 34(7):1281–1298, 2012. doi: 10.1109/TPAMI.2011.222.

[20] M. Calonder, L. V., S. C., and F. P. Brief: Binary robust independent elementary features. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, pages 778–792, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. doi: https://doi.org/10.1007/978-3-642-15561-1_56.

[21] A. F. Cheng et al. Near-earth asteroid rendezvous: Mission overview. *Journal of Geophysical Research: Planets*, 102(E10):23695–23708, 1997. doi: https://doi.org/10.1029/96JE03364. URL `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/96JE03364`.

[22] A. F. Cheng et al. Aida dart asteroid deflection test: Planetary defense and science objectives. *Planetary and Space Science*, 157:104–115, 2018. ISSN 0032-0633.

doi: https://doi.org/10.1016/j.pss.2018.02.015. URL `https://www.sciencedirect.com/science/article/pii/S0032063317304579`.

[23] C. Cocaud and T. Kubota. Slam-based navigation scheme for pinpoint landing on small celestial body. *Advanced Robotics*, 26(15):1747–1770, 2012. doi: 10.1080/01691864.2012.685227. URL `https://doi.org/10.1080/01691864.2012.685227`.

[24] D. DeTone, T. Malisiewicz, and A. Rabinovich. Superpoint: Self-supervised interest point detection and description. pages 337–33712, 06 2018. doi: 10.1109/CVPRW.2018.00060.

[25] M. Dor et al. Visual slam for asteroid relative navigation. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 2066–2075, 2021. doi: 10.1109/CVPRW53098.2021.00235.

[26] L. Downes, T. Steiner, and J. How. Lunar terrain relative navigation using a convolutional neural network for visual crater detection. *2020 American Control Conference (ACC)*, pages 4448–4453, 2020. doi: https://doi.org/10.23919/ACC45564.2020.9147595.

[27] L. Downes, T. Steiner, and J. How. Deep learning crater detection for lunar terrain relative navigation. 01 2020. doi: 10.2514/6.2020-1838.

[28] L. M. Downes, T. J. Steiner, and J. P. How. Neural network approach to crater detection for lunar terrain relative navigation. *Journal of Aerospace Information Systems*, 18(7):391–403, 2021. doi: 10.2514/1.I010884.

[29] T. Dylan, K. Scott, and B. Jonathan. A monocular slam method for satellite proximity operations. In *2016 American Control Conference (ACC)*, pages 4035–4040, 2016. doi: 10.1109/ACC.2016.7525555.

[30] H. A. Edward et al. Pyramid methods in image processing. 1984.

[31] C. Ericson. *Real-Time Collision Detection*. CRC Press, Inc., USA, 2004. ISBN 1558607323.

[32] D. M. et al. D2-net: A trainable cnn for joint description and detection of local features. pages 8084–8093, 06 2019. doi: 10.1109/CVPR.2019.00828.

[33] R. Furfaro et al. Deep learning for autonomous lunar landing. volume 167, pages 3285–3306, 2018. URL `https://www.scopus.com/inward/record.uri?eid=2-s2.0-85069471365&partnerID=40&md5=634f5e2ec8a39075fd83de52abd17237`. cited By 9.

[34] Gaskell et al. Characterizing and navigating small bodies with imaging data. *The Meteoritical Society*, 43(6):1049–1061, 2008. doi: https://doi.org/10.1111/j.1945-5100.2008.tb00692.x.

[35] R. Gaskell. Gaskell eros shape model. doi: 10.26033/d0gq-9427, 2008. Accessed: 2022-04-03.

[36] R. Gaskell et al. Gaskell itokawa shape model. doi: 10.26033/3b2j-yy57, 2008. Accessed: 2022-04-03.

[37] K. H. Glassmeier et al. The rosetta mission: Flying towards the origin of the solar system. *Space Science Reviews*, 128:1–21, 2007. doi: https://doi.org/10.1007/S11214-006-9140-8.

[38] D. Golish et al. Disk-resolved photometric modeling and properties of asteroid (101955) bennu. *Icarus*, 357:113724, 2021. ISSN 0019-1035. doi: https://doi.org/10.1016/j.icarus.2020.113724. URL `https://www.sciencedirect.com/science/article/pii/S0019103520301159`.

[39] R. M. Haralock and L. G. Shapiro. *Computer and Robot Vision*, chapter 8. Addison-Wesley Longman Publishing Co., Inc., USA, 1st edition, 1991. ISBN 0201108771.

[40] C. Harris and M. J. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, 1988.

[41] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1997. ISSN 1077-3142. doi: https://doi.org/10.1006/cviu.1997.0547. URL `https://www.sciencedirect.com/science/article/pii/S1077314297905476`.

[42] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[43] G. Koteswara Rao et al. Comparing 3d rendering engines in blender. In *2021 2nd International Conference on Smart Electronics and Communication (ICOSEC)*, pages 489–495, 2021. doi: 10.1109/ICOSEC51865.2021.9591800.

[44] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *2011 International Conference on Computer Vision*, pages 2548–2555, 2011. doi: 10.1109/ICCV.2011.6126542.

[45] J. P. Lewis. Fast normalized cross-correlation, 1995.

[46] R. Linares, T. Campbell, R. Furfaro, and D. Gaylor. A deep learning approach for optical autonomous planetary relative terrain navigation. 02 2017.

[47] T. Lindeberg. Image matching using generalized scale-space interest points. volume Springer LNCS volume 7893, pages 355–367, 06 2013. ISBN 978-3-642-38266-6. doi: 10.1007/978-3-642-38267-3_30.

[48] H. Longuet-Higgins. A computer algorithm for reconstructing a scene from two projections. In M. A. Fischler and O. Firschein, editors, *Readings in Computer Vision*, pages 61–62. Morgan Kaufmann, San Francisco (CA), 1987. ISBN 978-0-08-051581-6. doi: https://doi.org/10.1016/B978-0-08-051581-6.50012-X. URL `https://www.sciencedirect.com/science/article/pii/B978008051581650012X`.

[49] D. A. Lorenz et al. Lessons learned from osiris-rex autonomous navigation using natural feature tracking. *2017 IEEE Aerospace Conference*. doi: https://doi.org/10.1109/AERO.2017.7943684.

[50] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110, 2004. doi: https://doi.org/10.1023/B:VISI.0000029664.99615.94.

[51] B. D. Lucas, T. Kanade, et al. An iterative image registration technique with an application to stereo vision. Vancouver, British Columbia, 1981.

[52] E. Mair et al. Adaptive and generic corner detection based on the accelerated segment test. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Computer Vision – ECCV 2010*, pages 183–196, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg. doi: https://doi.org/10.1007/978-3-642-15552-9_14.

[53] S. Marschner and P. Shirley. Fundamentals of computer graphics, fourth edition. 2016.

[54] P. Mehlitz and J. Penix. *Expecting the Unexpected - Radiation Hardened Software*. doi: 10.2514/6.2005-7088. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2005-7088`.

[55] B. J. Morrell et al. *Autonomous Feature Tracking for Autonomous Approach to a Small Body*. doi: 10.2514/6.2020-4151. URL `https://arc.aiaa.org/doi/abs/10.2514/6.2020-4151`.

[56] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.

[57] M. Nielsen. *How the backpropagation algorithm works*, chapter 2. URL `http://neuralnetworksanddeeplearning.com/chap2.html`.

[58] N. Ogawa et al. *Image-based Autonomous Navigation of Hayabusa2 using Artificial Landmarks: Design and In-Flight Results in Landing Operations on Asteroid Ryugu.* doi: 10.2514/6.2020-0225.

[59] Y. Ono, E. Trulls, P. Fua, and K. M. Yi. Lf-net: Learning local features from images. In *NeurIPS*, 2018.

[60] W. J. Owen et al. Near optical navigation at eros. 109:1075–1087, 01 2002. URL `http://hdl.handle.net/2014/12992`.

[61] P. Panicucci. *Autonomous vision-based navigation and shape reconstruction of an unknown asteroid during approach phase.* PhD thesis, ISAE-SUPAERO, Tolouse, France, Mar. 2021.

[62] P. Panicucci et al. Localization and mapping merging silhouttes information and feature tracking for small body applications. 06 2021.

[63] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 555–562, 1998. doi: 10.1109/ICCV.1998.710772.

[64] M. Pavoni. Small bodies centroiding via image processing and convolutional neural network, 2020.

[65] A. Pellacani et al. Hera vision based gnc and autonomy. In *Proceedings of the 8th European Conference for Aeronautics and Space Sciences (EUCASS), Madrid, Spain*, pages 1–4, 2019.

[66] M. Pugliatti and F. Topputo. Navigation about irregular bodies through segmentation maps. 02 2021. doi: http://hdl.handle.net/11311/1163932.

[67] M. Pugliatti, V. Franzese, and F. Topputo. Data-driven image processing for onboard optical navigation around a binary asteroid. *Journal of Spacecraft and Rockets*, 0(0): 1–17, 0. doi: 10.2514/1.A35213. URL `https://doi.org/10.2514/1.A35213`.

[68] T. Roatsch et al. Dawn fc2 derived vesta global mosaics v1.0. `https://sbn.psi.edu/pds/archive/maps.html`, 2015. Accessed: 2022-04-03.

[69] P. L. Rosin. Measuring corner properties. *Comput. Vis. Image Underst.*, 73:291–307, 1999.

[70] E. Rosten et al. Machine learning for high-speed corner detection. In A. Leonardis, H. Bischof, and A. Pinz, editors, *Computer Vision – ECCV 2006*, pages 430–443, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg. doi: https://doi.org/10.1007/11744023_34.

[71] E. Rublee et al. Orb: An efficient alternative to sift or surf. In *2011 International Conference on Computer Vision*, pages 2564–2571, 2011. doi: 10.1109/ICCV.2011.6126544.

[72] W. S. et al. Hayabusa2 mission overview. *Space Science Reviews*, 208:1–14, 07 2017. doi: 10.1007/s11214-017-0377-1.

[73] R. P. D. Santayana and M. Lauer. Optical measurements for rosetta navigation near the comet. 2015.

[74] S. Sharma. Activation functions in neural networks. URL `https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6`.

[75] J. Shi and C. Tomasi. Good features to track. In *1994 Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 593–600, 1994. doi: 10.1109/CVPR.1994.323794.

[76] A. Silburt et al. Lunar crater identification via deep learning. *Icarus*, 317:27–38, 2019. doi: https://doi.org/10.1016/j.icarus.2018.06.022.

[77] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.

[78] P. Stooke. Stooke small bodies maps v3.0. multi-sa-multi-6-stookemaps-v3.0. nasa planetary data system. `https://sbn.psi.edu/pds/resource/stookemaps.html`, 2015. Accessed: 2022-04-03.

[79] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.

[80] Z. Yi. Evaluation and implementation of convolutional neural networks in image recognition. *Journal of Physics: Conference Series*, 1087:062018, sep 2018. doi: 10.1088/1742-6596/1087/6/062018. URL `https://doi.org/10.1088/1742-6596/1087/6/062018`.

[81] M. Yoshikawa, J. Kawaguchi, A. Fujiwara, and A. Tsuchiyama. Chapter 6 - the hayabusa mission. In A. Longobardo, editor, *Sample Return Missions*, pages 123–146. Elsevier, 2021. ISBN 978-0-12-818330-4. doi: https://doi.org/10.1016/

B978-0-12-818330-4.00006-9. URL https://www.sciencedirect.com/science/article/pii/B9780128183304000069.

# A | Appendix A - Covariance matrices

In this appendix, the covariance matrices of the distributions of the errors are reported.

## A.0.1. Bennu

Table A.1: Covariance matrices (Bennu).

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of $1.0\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}110.37 & 3.72\\ 3.72 & 8.18\end{bmatrix}$ | $\begin{bmatrix}291.79 & -8.04\\ -8.04 & 34.38\end{bmatrix}$ | $\begin{bmatrix}376.34 & -7.72\\ -7.72 & 60.91\end{bmatrix}$ |
| SURF | $\begin{bmatrix}86.8 & -5.01\\ -5.01 & 16.64\end{bmatrix}$ | $\begin{bmatrix}459.22 & -15.24\\ -15.24 & 183.69\end{bmatrix}$ | $\begin{bmatrix}775.44 & -20.44\\ -20.44 & 466.79\end{bmatrix}$ |
| ORB | $\begin{bmatrix}26.83 & 1.12\\ 1.12 & 8.33\end{bmatrix}$ | $\begin{bmatrix}84.01 & -0.32\\ -0.32 & 35.5\end{bmatrix}$ | $\begin{bmatrix}271.96 & -14.92\\ -14.92 & 85.18\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}8.15 & -0.48\\ -0.48 & 6.18\end{bmatrix}$ | $\begin{bmatrix}41.5 & 5.79\\ 5.79 & 24.72\end{bmatrix}$ | $\begin{bmatrix}73.03 & 6.56\\ 6.56 & 41.65\end{bmatrix}$ |
| KLT | $\begin{bmatrix}92.12 & 149.17\\ 149.17 & 1396.71\end{bmatrix}$ | $\begin{bmatrix}246.82 & 317.56\\ 317.56 & 3188.1\end{bmatrix}$ | $\begin{bmatrix}424.32 & 450.45\\ 450.45 & 4662.08\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.43 & 0.23\\ 0.23 & 0.31\end{bmatrix}$ | $\begin{bmatrix}10.1 & 1.7\\ 1.7 & 6.48\end{bmatrix}$ | $\begin{bmatrix}60.66 & -0.26\\ -0.26 & 35.47\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}161.55 & 3.09\\ 3.09 & 80.27\end{bmatrix}$ | $\begin{bmatrix}185.51 & 0.01\\ 0.01 & 97.31\end{bmatrix}$ | $\begin{bmatrix}188.23 & -13.65\\ -13.65 & 112.19\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}91.54 & -7.82\\ -7.82 & 40.78\end{bmatrix}$ | $\begin{bmatrix}76.2 & -9.3\\ -9.3 & 50.27\end{bmatrix}$ | $\begin{bmatrix}180.99 & -18.52\\ -18.52 & 66.65\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}35.98 & -0.83\\ -0.83 & 29.61\end{bmatrix}$ | $\begin{bmatrix}48.26 & -2.13\\ -2.13 & 33.54\end{bmatrix}$ | $\begin{bmatrix}46.23 & -3.04\\ -3.04 & 35.05\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}27.39 & -0.68\\ -0.68 & 25.78\end{bmatrix}$ | $\begin{bmatrix}30.12 & -1.12\\ -1.12 & 30.53\end{bmatrix}$ | $\begin{bmatrix}33.63 & -0.44\\ -0.44 & 36.75\end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of $1.0\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}0.01 & 0.0\\ 0.0 & 0.01\end{bmatrix}$ | $\begin{bmatrix}0.01 & 0.01\\ 0.01 & 0.4\end{bmatrix}$ | $\begin{bmatrix}0.02 & 0.0\\ 0.0 & 0.15\end{bmatrix}$ |
| SURF | $\begin{bmatrix}0.06 & -0.05\\ -0.05 & 0.4\end{bmatrix}$ | $\begin{bmatrix}0.07 & -0.03\\ -0.03 & 1.2\end{bmatrix}$ | $\begin{bmatrix}0.09 & -0.0\\ -0.0 & 0.27\end{bmatrix}$ |
| ORB | $\begin{bmatrix}0.34 & 0.0\\ 0.0 & 0.53\end{bmatrix}$ | $\begin{bmatrix}0.37 & 0.0\\ 0.0 & 0.59\end{bmatrix}$ | $\begin{bmatrix}0.39 & 0.01\\ 0.01 & 0.64\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}0.07 & 0.0\\ 0.0 & 0.07\end{bmatrix}$ | $\begin{bmatrix}0.08 & -0.01\\ -0.01 & 1.0\end{bmatrix}$ | $\begin{bmatrix}0.08 & 0.0\\ 0.0 & 0.1\end{bmatrix}$ |
| KLT | $\begin{bmatrix}81.5 & 145.25\\ 145.25 & 1396.74\end{bmatrix}$ | $\begin{bmatrix}147.58 & 273.29\\ 273.29 & 3340.77\end{bmatrix}$ | $\begin{bmatrix}199.09 & 214.0\\ 214.0 & 3977.44\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.12 & 0.0\\ 0.0 & 0.13\end{bmatrix}$ | $\begin{bmatrix}0.16 & 0.0\\ 0.0 & 0.21\end{bmatrix}$ | $\begin{bmatrix}0.2 & 0.0\\ 0.0 & 0.36\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}11.85 & -1.51\\ -1.51 & 23.89\end{bmatrix}$ | $\begin{bmatrix}9.91 & 0.53\\ 0.53 & 35.33\end{bmatrix}$ | $\begin{bmatrix}14.54 & -1.91\\ -1.91 & 38.7\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}10.48 & -0.66\\ -0.66 & 18.5\end{bmatrix}$ | $\begin{bmatrix}9.39 & -0.42\\ -0.42 & 21.58\end{bmatrix}$ | $\begin{bmatrix}24.32 & 2.58\\ 2.58 & 28.71\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}8.78 & -0.48\\ -0.48 & 17.15\end{bmatrix}$ | $\begin{bmatrix}8.34 & -0.36\\ -0.36 & 20.2\end{bmatrix}$ | $\begin{bmatrix}8.46 & -0.65\\ -0.65 & 23.32\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}8.5 & -0.25\\ -0.25 & 16.11\end{bmatrix}$ | $\begin{bmatrix}8.19 & -0.26\\ -0.26 & 19.42\end{bmatrix}$ | $\begin{bmatrix}9.2 & -0.36\\ -0.36 & 23.27\end{bmatrix}$ |

Table A.1

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of $1.0\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 16.18 & -0.87 \\ -0.87 & 3.95 \end{bmatrix}$ | $\begin{bmatrix} 43.68 & -3.58 \\ -3.58 & 13.5 \end{bmatrix}$ | $\begin{bmatrix} 65.69 & 2.1 \\ 2.1 & 27.73 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 17.7 & -0.66 \\ -0.66 & 6.55 \end{bmatrix}$ | $\begin{bmatrix} 127.14 & -3.0 \\ -3.0 & 68.79 \end{bmatrix}$ | $\begin{bmatrix} 243.25 & 3.84 \\ 3.84 & 138.58 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 1.98 & -0.04 \\ -0.04 & 1.22 \end{bmatrix}$ | $\begin{bmatrix} 16.63 & 2.16 \\ 2.16 & 5.78 \end{bmatrix}$ | $\begin{bmatrix} 36.04 & 3.54 \\ 3.54 & 11.74 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 0.86 & -0.09 \\ -0.09 & 0.33 \end{bmatrix}$ | $\begin{bmatrix} 7.88 & -0.35 \\ -0.35 & 2.72 \end{bmatrix}$ | $\begin{bmatrix} 24.5 & -2.15 \\ -2.15 & 10.21 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 11.14 & 0.92 \\ 0.92 & 154.17 \end{bmatrix}$ | $\begin{bmatrix} 32.88 & 0.29 \\ 0.29 & 392.33 \end{bmatrix}$ | $\begin{bmatrix} 69.83 & -2.1 \\ -2.1 & 676.07 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.16 & 0.04 \\ 0.04 & 0.17 \end{bmatrix}$ | $\begin{bmatrix} 2.08 & 0.13 \\ 0.13 & 1.02 \end{bmatrix}$ | $\begin{bmatrix} 11.94 & -0.37 \\ -0.37 & 5.65 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 94.75 & -11.07 \\ -11.07 & 79.74 \end{bmatrix}$ | $\begin{bmatrix} 137.88 & -16.69 \\ -16.69 & 110.77 \end{bmatrix}$ | $\begin{bmatrix} 149.13 & -20.1 \\ -20.1 & 122.03 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 66.39 & -12.46 \\ -12.46 & 54.75 \end{bmatrix}$ | $\begin{bmatrix} 98.64 & -21.48 \\ -21.48 & 80.76 \end{bmatrix}$ | $\begin{bmatrix} 111.65 & -13.72 \\ -13.72 & 92.91 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 44.04 & -3.42 \\ -3.42 & 35.58 \end{bmatrix}$ | $\begin{bmatrix} 62.61 & -6.38 \\ -6.38 & 46.86 \end{bmatrix}$ | $\begin{bmatrix} 60.56 & -9.99 \\ -9.99 & 58.31 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 28.61 & -1.1 \\ -1.1 & 25.1 \end{bmatrix}$ | $\begin{bmatrix} 36.14 & -2.17 \\ -2.17 & 35.51 \end{bmatrix}$ | $\begin{bmatrix} 42.54 & -3.05 \\ -3.05 & 44.67 \end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of $1.0\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 0.01 & -0.01 \\ -0.01 & 0.42 \end{bmatrix}$ | $\begin{bmatrix} 0.02 & -0.01 \\ -0.01 & 1.38 \end{bmatrix}$ | $\begin{bmatrix} 0.03 & -0.02 \\ -0.02 & 3.25 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 0.06 & -0.0 \\ -0.0 & 0.3 \end{bmatrix}$ | $\begin{bmatrix} 0.1 & -0.01 \\ -0.01 & 0.55 \end{bmatrix}$ | $\begin{bmatrix} 0.15 & -0.1 \\ -0.1 & 2.77 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 0.19 & 0.0 \\ 0.0 & 0.53 \end{bmatrix}$ | $\begin{bmatrix} 0.28 & 0.01 \\ 0.01 & 0.63 \end{bmatrix}$ | $\begin{bmatrix} 0.32 & 0.01 \\ 0.01 & 0.69 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 0.05 & 0.0 \\ 0.0 & 0.06 \end{bmatrix}$ | $\begin{bmatrix} 0.07 & 0.0 \\ 0.0 & 0.08 \end{bmatrix}$ | $\begin{bmatrix} 0.07 & 0.0 \\ 0.0 & 0.09 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 10.96 & 0.93 \\ 0.93 & 154.51 \end{bmatrix}$ | $\begin{bmatrix} 28.44 & 1.06 \\ 1.06 & 399.58 \end{bmatrix}$ | $\begin{bmatrix} 51.28 & 6.18 \\ 6.18 & 668.13 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.09 & 0.0 \\ 0.0 & 0.13 \end{bmatrix}$ | $\begin{bmatrix} 0.16 & 0.0 \\ 0.0 & 0.22 \end{bmatrix}$ | $\begin{bmatrix} 0.2 & 0.0 \\ 0.0 & 0.33 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 28.03 & -3.2 \\ -3.2 & 36.44 \end{bmatrix}$ | $\begin{bmatrix} 34.28 & -6.37 \\ -6.37 & 53.99 \end{bmatrix}$ | $\begin{bmatrix} 44.75 & -10.42 \\ -10.42 & 64.03 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 31.26 & -8.65 \\ -8.65 & 33.74 \end{bmatrix}$ | $\begin{bmatrix} 38.58 & -10.77 \\ -10.77 & 48.61 \end{bmatrix}$ | $\begin{bmatrix} 52.67 & -5.6 \\ -5.6 & 58.23 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 19.98 & -1.23 \\ -1.23 & 22.84 \end{bmatrix}$ | $\begin{bmatrix} 22.46 & -0.56 \\ -0.56 & 31.82 \end{bmatrix}$ | $\begin{bmatrix} 26.44 & -5.92 \\ -5.92 & 42.5 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 12.08 & -0.8 \\ -0.8 & 15.79 \end{bmatrix}$ | $\begin{bmatrix} 12.72 & -1.37 \\ -1.37 & 22.45 \end{bmatrix}$ | $\begin{bmatrix} 16.1 & -1.55 \\ -1.55 & 28.46 \end{bmatrix}$ |

Table A.1

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of $1.0\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}174.81 & -0.12\\ -0.12 & 17.37\end{bmatrix}$ | $\begin{bmatrix}433.02 & -12.76\\ -12.76 & 68.28\end{bmatrix}$ | $\begin{bmatrix}753.92 & 2.64\\ 2.64 & 191.14\end{bmatrix}$ |
| SURF | $\begin{bmatrix}82.35 & 3.65\\ 3.65 & 15.45\end{bmatrix}$ | $\begin{bmatrix}524.66 & -13.43\\ -13.43 & 178.22\end{bmatrix}$ | $\begin{bmatrix}939.97 & 30.66\\ 30.66 & 431.14\end{bmatrix}$ |
| ORB | $\begin{bmatrix}46.14 & -0.14\\ -0.14 & 12.47\end{bmatrix}$ | $\begin{bmatrix}166.37 & 8.56\\ 8.56 & 57.63\end{bmatrix}$ | $\begin{bmatrix}349.85 & 6.79\\ 6.79 & 154.57\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}43.11 & -3.45\\ -3.45 & 18.94\end{bmatrix}$ | $\begin{bmatrix}80.75 & -3.19\\ -3.19 & 28.5\end{bmatrix}$ | $\begin{bmatrix}200.13 & 11.87\\ 11.87 & 76.54\end{bmatrix}$ |
| KLT | $\begin{bmatrix}104.91 & 167.64\\ 167.64 & 1368.8\end{bmatrix}$ | $\begin{bmatrix}277.04 & 354.88\\ 354.88 & 3106.12\end{bmatrix}$ | $\begin{bmatrix}475.33 & 505.51\\ 505.51 & 4608.1\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.13 & 0.0\\ 0.0 & 0.14\end{bmatrix}$ | $\begin{bmatrix}11.58 & -1.53\\ -1.53 & 3.38\end{bmatrix}$ | $\begin{bmatrix}72.33 & -6.16\\ -6.16 & 28.51\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}83.82 & -0.91\\ -0.91 & 54.71\end{bmatrix}$ | $\begin{bmatrix}143.35 & -2.09\\ -2.09 & 71.5\end{bmatrix}$ | $\begin{bmatrix}243.79 & -31.63\\ -31.63 & 116.5\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}65.47 & -0.23\\ -0.23 & 32.73\end{bmatrix}$ | $\begin{bmatrix}62.68 & -2.21\\ -2.21 & 37.55\end{bmatrix}$ | $\begin{bmatrix}100.65 & -7.12\\ -7.12 & 49.59\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}28.28 & -0.49\\ -0.49 & 26.71\end{bmatrix}$ | $\begin{bmatrix}31.64 & -0.83\\ -0.83 & 31.68\end{bmatrix}$ | $\begin{bmatrix}37.89 & 0.2\\ 0.2 & 35.88\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}27.55 & -0.53\\ -0.53 & 25.55\end{bmatrix}$ | $\begin{bmatrix}29.54 & -0.89\\ -0.89 & 29.31\end{bmatrix}$ | $\begin{bmatrix}33.59 & -1.58\\ -1.58 & 37.85\end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of $1.0\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}0.01 & 0.0\\ 0.0 & 0.04\end{bmatrix}$ | $\begin{bmatrix}0.02 & 0.0\\ 0.0 & 0.02\end{bmatrix}$ | $\begin{bmatrix}0.03 & -0.08\\ -0.08 & 2.08\end{bmatrix}$ |
| SURF | $\begin{bmatrix}0.04 & -0.0\\ -0.0 & 0.09\end{bmatrix}$ | $\begin{bmatrix}0.07 & -0.0\\ -0.0 & 0.44\end{bmatrix}$ | $\begin{bmatrix}0.13 & -0.26\\ -0.26 & 2.36\end{bmatrix}$ |
| ORB | $\begin{bmatrix}0.34 & -0.01\\ -0.01 & 1.27\end{bmatrix}$ | $\begin{bmatrix}0.38 & 0.0\\ 0.0 & 0.68\end{bmatrix}$ | $\begin{bmatrix}0.41 & -0.0\\ -0.0 & 5.87\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}0.08 & -0.01\\ -0.01 & 0.16\end{bmatrix}$ | $\begin{bmatrix}0.1 & 0.0\\ 0.0 & 0.11\end{bmatrix}$ | $\begin{bmatrix}0.1 & 0.0\\ 0.0 & 0.19\end{bmatrix}$ |
| KLT | $\begin{bmatrix}95.61 & 167.41\\ 167.41 & 1372.87\end{bmatrix}$ | $\begin{bmatrix}167.82 & 288.29\\ 288.29 & 3139.63\end{bmatrix}$ | $\begin{bmatrix}196.67 & 231.1\\ 231.1 & 3572.49\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.12 & 0.0\\ 0.0 & 0.14\end{bmatrix}$ | $\begin{bmatrix}0.17 & 0.0\\ 0.0 & 0.21\end{bmatrix}$ | $\begin{bmatrix}0.2 & 0.0\\ 0.0 & 0.31\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}9.67 & -0.69\\ -0.69 & 21.88\end{bmatrix}$ | $\begin{bmatrix}16.46 & 1.04\\ 1.04 & 28.99\end{bmatrix}$ | $\begin{bmatrix}12.79 & -2.0\\ -2.0 & 38.89\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}9.02 & -0.38\\ -0.38 & 16.76\end{bmatrix}$ | $\begin{bmatrix}8.72 & -0.53\\ -0.53 & 20.94\end{bmatrix}$ | $\begin{bmatrix}9.04 & -0.59\\ -0.59 & 25.56\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}8.28 & -0.27\\ -0.27 & 16.43\end{bmatrix}$ | $\begin{bmatrix}7.7 & -0.4\\ -0.4 & 19.09\end{bmatrix}$ | $\begin{bmatrix}8.18 & -0.43\\ -0.43 & 21.97\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}7.98 & -0.22\\ -0.22 & 15.44\end{bmatrix}$ | $\begin{bmatrix}7.68 & -0.32\\ -0.32 & 18.22\end{bmatrix}$ | $\begin{bmatrix}8.46 & -0.52\\ -0.52 & 22.38\end{bmatrix}$ |

Table A.1

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.0 $Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 20.0 & -0.69 \\ -0.69 & 11.42 \end{bmatrix}$ | $\begin{bmatrix} 51.91 & 0.77 \\ 0.77 & 23.23 \end{bmatrix}$ | $\begin{bmatrix} 125.06 & -1.06 \\ -1.06 & 49.43 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 17.02 & -0.31 \\ -0.31 & 10.07 \end{bmatrix}$ | $\begin{bmatrix} 173.42 & 2.67 \\ 2.67 & 97.89 \end{bmatrix}$ | $\begin{bmatrix} 291.33 & 17.77 \\ 17.77 & 151.49 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 3.37 & -0.1 \\ -0.1 & 1.56 \end{bmatrix}$ | $\begin{bmatrix} 19.79 & -0.16 \\ -0.16 & 7.78 \end{bmatrix}$ | $\begin{bmatrix} 50.05 & -0.06 \\ -0.06 & 23.17 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 1.74 & 0.19 \\ 0.19 & 0.53 \end{bmatrix}$ | $\begin{bmatrix} 17.97 & 0.77 \\ 0.77 & 9.96 \end{bmatrix}$ | $\begin{bmatrix} 27.04 & -0.75 \\ -0.75 & 9.8 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 12.29 & -0.74 \\ -0.74 & 139.46 \end{bmatrix}$ | $\begin{bmatrix} 36.13 & -3.84 \\ -3.84 & 350.06 \end{bmatrix}$ | $\begin{bmatrix} 76.68 & -9.18 \\ -9.18 & 613.36 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.12 & -0.0 \\ -0.0 & 0.16 \end{bmatrix}$ | $\begin{bmatrix} 1.73 & 0.01 \\ 0.01 & 1.28 \end{bmatrix}$ | $\begin{bmatrix} 18.14 & 0.47 \\ 0.47 & 6.25 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 79.59 & -10.55 \\ -10.55 & 57.75 \end{bmatrix}$ | $\begin{bmatrix} 113.32 & -16.17 \\ -16.17 & 87.54 \end{bmatrix}$ | $\begin{bmatrix} 164.49 & -21.03 \\ -21.03 & 111.81 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 49.37 & -7.25 \\ -7.25 & 44.91 \end{bmatrix}$ | $\begin{bmatrix} 64.29 & -10.65 \\ -10.65 & 60.84 \end{bmatrix}$ | $\begin{bmatrix} 79.64 & -16.44 \\ -16.44 & 76.38 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 37.15 & -4.58 \\ -4.58 & 35.78 \end{bmatrix}$ | $\begin{bmatrix} 43.58 & -5.86 \\ -5.86 & 44.28 \end{bmatrix}$ | $\begin{bmatrix} 44.18 & -5.58 \\ -5.58 & 50.0 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 27.5 & -0.81 \\ -0.81 & 24.97 \end{bmatrix}$ | $\begin{bmatrix} 34.95 & -1.86 \\ -1.86 & 34.56 \end{bmatrix}$ | $\begin{bmatrix} 40.71 & -1.91 \\ -1.91 & 42.8 \end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.0 $Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 0.01 & -0.01 \\ -0.01 & 0.5 \end{bmatrix}$ | $\begin{bmatrix} 0.02 & -0.02 \\ -0.02 & 1.87 \end{bmatrix}$ | $\begin{bmatrix} 0.03 & -0.02 \\ -0.02 & 3.79 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 0.07 & -0.01 \\ -0.01 & 0.31 \end{bmatrix}$ | $\begin{bmatrix} 0.12 & 0.02 \\ 0.02 & 1.19 \end{bmatrix}$ | $\begin{bmatrix} 0.14 & -0.09 \\ -0.09 & 3.66 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 0.19 & 0.0 \\ 0.0 & 0.57 \end{bmatrix}$ | $\begin{bmatrix} 0.29 & 0.02 \\ 0.02 & 0.75 \end{bmatrix}$ | $\begin{bmatrix} 0.34 & 0.01 \\ 0.01 & 0.82 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 0.07 & 0.0 \\ 0.0 & 0.08 \end{bmatrix}$ | $\begin{bmatrix} 0.09 & 0.0 \\ 0.0 & 0.15 \end{bmatrix}$ | $\begin{bmatrix} 0.1 & 0.0 \\ 0.0 & 0.13 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 11.93 & -0.74 \\ -0.74 & 139.85 \end{bmatrix}$ | $\begin{bmatrix} 30.07 & -2.96 \\ -2.96 & 353.92 \end{bmatrix}$ | $\begin{bmatrix} 54.42 & -0.11 \\ -0.11 & 602.48 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.1 & 0.0 \\ 0.0 & 0.15 \end{bmatrix}$ | $\begin{bmatrix} 0.17 & 0.0 \\ 0.0 & 0.26 \end{bmatrix}$ | $\begin{bmatrix} 0.21 & -0.01 \\ -0.01 & 0.57 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 24.8 & -3.93 \\ -3.93 & 28.29 \end{bmatrix}$ | $\begin{bmatrix} 28.02 & -7.86 \\ -7.86 & 45.52 \end{bmatrix}$ | $\begin{bmatrix} 43.3 & -10.91 \\ -10.91 & 63.06 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 22.84 & -3.95 \\ -3.95 & 26.82 \end{bmatrix}$ | $\begin{bmatrix} 25.79 & -5.37 \\ -5.37 & 34.15 \end{bmatrix}$ | $\begin{bmatrix} 39.88 & -8.23 \\ -8.23 & 46.76 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 16.82 & -2.07 \\ -2.07 & 21.14 \end{bmatrix}$ | $\begin{bmatrix} 17.23 & -2.77 \\ -2.77 & 28.44 \end{bmatrix}$ | $\begin{bmatrix} 19.32 & -4.88 \\ -4.88 & 35.81 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 11.15 & -0.74 \\ -0.74 & 15.43 \end{bmatrix}$ | $\begin{bmatrix} 12.21 & -1.01 \\ -1.01 & 22.32 \end{bmatrix}$ | $\begin{bmatrix} 14.71 & -1.67 \\ -1.67 & 28.46 \end{bmatrix}$ |

### A.0.2. Itokawa

Table A.2: Covariance matrices (Itokawa).

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of $0.7\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}20.89 & -0.29\\ -0.29 & 11.65\end{bmatrix}$ | $\begin{bmatrix}74.88 & -5.63\\ -5.63 & 55.48\end{bmatrix}$ | $\begin{bmatrix}191.99 & -15.05\\ -15.05 & 174.91\end{bmatrix}$ |
| SURF | $\begin{bmatrix}5.48 & 0.06\\ 0.06 & 7.39\end{bmatrix}$ | $\begin{bmatrix}29.08 & 3.81\\ 3.81 & 19.33\end{bmatrix}$ | $\begin{bmatrix}111.67 & -11.12\\ -11.12 & 58.16\end{bmatrix}$ |
| ORB | $\begin{bmatrix}16.32 & -4.06\\ -4.06 & 12.12\end{bmatrix}$ | $\begin{bmatrix}84.18 & -7.61\\ -7.61 & 64.15\end{bmatrix}$ | $\begin{bmatrix}124.67 & 11.27\\ 11.27 & 99.03\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}4.93 & -1.76\\ -1.76 & 4.69\end{bmatrix}$ | $\begin{bmatrix}22.07 & -5.16\\ -5.16 & 36.89\end{bmatrix}$ | $\begin{bmatrix}27.48 & -5.7\\ -5.7 & 29.04\end{bmatrix}$ |
| KLT | $\begin{bmatrix}78.54 & 93.19\\ 93.19 & 689.64\end{bmatrix}$ | $\begin{bmatrix}200.7 & 205.21\\ 205.21 & 1564.88\end{bmatrix}$ | $\begin{bmatrix}390.96 & 339.85\\ 339.85 & 2673.95\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.44 & -0.2\\ -0.2 & 0.33\end{bmatrix}$ | $\begin{bmatrix}1.52 & -1.67\\ -1.67 & 3.07\end{bmatrix}$ | $\begin{bmatrix}5.59 & -2.68\\ -2.68 & 4.63\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}24.21 & -0.27\\ -0.27 & 27.37\end{bmatrix}$ | $\begin{bmatrix}31.84 & -0.81\\ -0.81 & 33.69\end{bmatrix}$ | $\begin{bmatrix}39.49 & -1.86\\ -1.86 & 39.69\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}19.16 & 0.39\\ 0.39 & 19.21\end{bmatrix}$ | $\begin{bmatrix}29.66 & -2.62\\ -2.62 & 24.05\end{bmatrix}$ | $\begin{bmatrix}23.67 & -0.88\\ -0.88 & 28.74\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}18.22 & 0.74\\ 0.74 & 18.88\end{bmatrix}$ | $\begin{bmatrix}18.14 & -0.06\\ -0.06 & 22.71\end{bmatrix}$ | $\begin{bmatrix}20.26 & -0.64\\ -0.64 & 31.44\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}16.81 & -0.21\\ -0.21 & 16.12\end{bmatrix}$ | $\begin{bmatrix}19.06 & -0.39\\ -0.39 & 20.06\end{bmatrix}$ | $\begin{bmatrix}21.12 & -0.31\\ -0.31 & 23.61\end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of $0.7\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}0.02 & -0.0\\ -0.0 & 0.12\end{bmatrix}$ | $\begin{bmatrix}0.03 & -0.02\\ -0.02 & 0.39\end{bmatrix}$ | $\begin{bmatrix}0.05 & -0.03\\ -0.03 & 0.55\end{bmatrix}$ |
| SURF | $\begin{bmatrix}0.03 & -0.0\\ -0.0 & 0.06\end{bmatrix}$ | $\begin{bmatrix}0.04 & -0.01\\ -0.01 & 0.13\end{bmatrix}$ | $\begin{bmatrix}0.07 & -0.01\\ -0.01 & 0.21\end{bmatrix}$ |
| ORB | $\begin{bmatrix}0.25 & -0.01\\ -0.01 & 0.57\end{bmatrix}$ | $\begin{bmatrix}0.35 & -0.03\\ -0.03 & 0.75\end{bmatrix}$ | $\begin{bmatrix}0.39 & -0.04\\ -0.04 & 0.91\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}0.22 & -0.0\\ -0.0 & 0.44\end{bmatrix}$ | $\begin{bmatrix}0.25 & -0.01\\ -0.01 & 0.6\end{bmatrix}$ | $\begin{bmatrix}0.28 & -0.02\\ -0.02 & 0.75\end{bmatrix}$ |
| KLT | $\begin{bmatrix}77.15 & 93.18\\ 93.18 & 675.74\end{bmatrix}$ | $\begin{bmatrix}152.8 & 178.12\\ 178.12 & 1291.98\end{bmatrix}$ | $\begin{bmatrix}181.72 & 211.48\\ 211.48 & 1551.78\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.11 & 0.0\\ 0.0 & 0.13\end{bmatrix}$ | $\begin{bmatrix}0.2 & -0.21\\ -0.21 & 0.93\end{bmatrix}$ | $\begin{bmatrix}0.17 & -0.0\\ -0.0 & 0.25\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}5.96 & -0.4\\ -0.4 & 12.78\end{bmatrix}$ | $\begin{bmatrix}5.89 & -0.76\\ -0.76 & 16.45\end{bmatrix}$ | $\begin{bmatrix}5.77 & -0.91\\ -0.91 & 19.09\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}6.43 & -0.27\\ -0.27 & 10.18\end{bmatrix}$ | $\begin{bmatrix}6.19 & -0.53\\ -0.53 & 13.17\end{bmatrix}$ | $\begin{bmatrix}6.0 & -0.69\\ -0.69 & 16.51\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}6.32 & -0.25\\ -0.25 & 10.29\end{bmatrix}$ | $\begin{bmatrix}6.17 & -0.47\\ -0.47 & 13.93\end{bmatrix}$ | $\begin{bmatrix}6.05 & -0.81\\ -0.81 & 17.16\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}6.55 & -0.23\\ -0.23 & 10.07\end{bmatrix}$ | $\begin{bmatrix}6.38 & -0.34\\ -0.34 & 12.62\end{bmatrix}$ | $\begin{bmatrix}6.26 & -0.65\\ -0.65 & 14.74\end{bmatrix}$ |

Table A.2

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}0.81 & 0.26\\0.26 & 3.26\end{bmatrix}$ | $\begin{bmatrix}4.08 & 0.69\\0.69 & 11.43\end{bmatrix}$ | $\begin{bmatrix}11.74 & 3.89\\3.89 & 25.04\end{bmatrix}$ |
| SURF | $\begin{bmatrix}0.46 & 0.03\\0.03 & 1.51\end{bmatrix}$ | $\begin{bmatrix}3.35 & -0.33\\-0.33 & 4.06\end{bmatrix}$ | $\begin{bmatrix}12.58 & 1.22\\1.22 & 16.88\end{bmatrix}$ |
| ORB | $\begin{bmatrix}0.17 & -0.0\\-0.0 & 0.45\end{bmatrix}$ | $\begin{bmatrix}0.69 & -0.15\\-0.15 & 0.95\end{bmatrix}$ | $\begin{bmatrix}3.57 & 0.49\\0.49 & 6.23\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}3.9 & -2.7\\-2.7 & 6.67\end{bmatrix}$ | $\begin{bmatrix}0.75 & 0.82\\0.82 & 1.73\end{bmatrix}$ | $\begin{bmatrix}4.55 & -0.25\\-0.25 & 3.35\end{bmatrix}$ |
| KLT | $\begin{bmatrix}66.97 & 1.54\\1.54 & 210.39\end{bmatrix}$ | $\begin{bmatrix}71.77 & 6.88\\6.88 & 389.15\end{bmatrix}$ | $\begin{bmatrix}102.79 & -12.15\\-12.15 & 446.55\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.12 & 0.01\\0.01 & 0.17\end{bmatrix}$ | $\begin{bmatrix}0.78 & -0.04\\-0.04 & 1.57\end{bmatrix}$ | $\begin{bmatrix}0.32 & -0.07\\-0.07 & 2.98\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}23.63 & -0.1\\-0.1 & 26.45\end{bmatrix}$ | $\begin{bmatrix}28.33 & 0.42\\0.42 & 32.04\end{bmatrix}$ | $\begin{bmatrix}30.08 & -1.04\\-1.04 & 34.6\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}18.75 & 0.03\\0.03 & 20.95\end{bmatrix}$ | $\begin{bmatrix}22.04 & -0.42\\-0.42 & 23.26\end{bmatrix}$ | $\begin{bmatrix}24.13 & -1.03\\-1.03 & 27.93\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}17.81 & -0.96\\-0.96 & 18.24\end{bmatrix}$ | $\begin{bmatrix}21.23 & -1.34\\-1.34 & 20.79\end{bmatrix}$ | $\begin{bmatrix}21.95 & -1.51\\-1.51 & 24.22\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}14.83 & -0.41\\-0.41 & 14.25\end{bmatrix}$ | $\begin{bmatrix}17.18 & -0.44\\-0.44 & 17.11\end{bmatrix}$ | $\begin{bmatrix}19.19 & -0.97\\-0.97 & 20.76\end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}0.06 & -0.19\\-0.19 & 1.43\end{bmatrix}$ | $\begin{bmatrix}0.04 & -0.1\\-0.1 & 0.42\end{bmatrix}$ | $\begin{bmatrix}0.73 & 0.38\\0.38 & 8.11\end{bmatrix}$ |
| SURF | $\begin{bmatrix}0.04 & -0.0\\-0.0 & 0.08\end{bmatrix}$ | $\begin{bmatrix}0.06 & -0.0\\-0.0 & 0.15\end{bmatrix}$ | $\begin{bmatrix}0.42 & 0.31\\0.31 & 0.61\end{bmatrix}$ |
| ORB | $\begin{bmatrix}0.08 & -0.0\\-0.0 & 0.42\end{bmatrix}$ | $\begin{bmatrix}0.17 & -0.0\\-0.0 & 0.51\end{bmatrix}$ | $\begin{bmatrix}0.24 & -0.01\\-0.01 & 0.6\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}3.82 & -2.7\\-2.7 & 6.49\end{bmatrix}$ | $\begin{bmatrix}0.12 & 0.0\\0.0 & 0.22\end{bmatrix}$ | $\begin{bmatrix}0.14 & -0.0\\-0.0 & 0.27\end{bmatrix}$ |
| KLT | $\begin{bmatrix}17.96 & -1.75\\-1.75 & 160.67\end{bmatrix}$ | $\begin{bmatrix}34.75 & -4.74\\-4.74 & 332.09\end{bmatrix}$ | $\begin{bmatrix}32.14 & -9.32\\-9.32 & 373.74\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.06 & 0.01\\0.01 & 0.1\end{bmatrix}$ | $\begin{bmatrix}0.09 & 0.0\\0.0 & 0.14\end{bmatrix}$ | $\begin{bmatrix}0.12 & 0.0\\0.0 & 2.42\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}9.94 & -0.87\\-0.87 & 14.38\end{bmatrix}$ | $\begin{bmatrix}8.67 & -0.43\\-0.43 & 16.84\end{bmatrix}$ | $\begin{bmatrix}8.46 & -1.2\\-1.2 & 18.43\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}10.05 & -0.62\\-0.62 & 12.87\end{bmatrix}$ | $\begin{bmatrix}10.31 & -1.06\\-1.06 & 13.92\end{bmatrix}$ | $\begin{bmatrix}10.42 & -1.85\\-1.85 & 17.1\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}9.57 & -0.86\\-0.86 & 12.48\end{bmatrix}$ | $\begin{bmatrix}10.08 & -1.01\\-1.01 & 14.06\end{bmatrix}$ | $\begin{bmatrix}9.42 & -0.91\\-0.91 & 16.75\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}8.03 & -0.51\\-0.51 & 9.58\end{bmatrix}$ | $\begin{bmatrix}8.03 & -0.62\\-0.62 & 11.46\end{bmatrix}$ | $\begin{bmatrix}8.58 & -0.76\\-0.76 & 14.27\end{bmatrix}$ |

Table A.2

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}111.54 & -18.72\\ -18.72 & 67.72\end{bmatrix}$ | $\begin{bmatrix}302.11 & -48.3\\ -48.3 & 270.89\end{bmatrix}$ | $\begin{bmatrix}525.55 & -55.6\\ -55.6 & 558.7\end{bmatrix}$ |
| SURF | $\begin{bmatrix}19.23 & -0.61\\ -0.61 & 7.63\end{bmatrix}$ | $\begin{bmatrix}112.24 & -17.52\\ -17.52 & 75.76\end{bmatrix}$ | $\begin{bmatrix}274.38 & -40.14\\ -40.14 & 219.25\end{bmatrix}$ |
| ORB | $\begin{bmatrix}20.57 & -0.86\\ -0.86 & 8.54\end{bmatrix}$ | $\begin{bmatrix}59.45 & -5.7\\ -5.7 & 49.56\end{bmatrix}$ | $\begin{bmatrix}141.72 & -5.96\\ -5.96 & 156.55\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}9.28 & -1.78\\ -1.78 & 6.02\end{bmatrix}$ | $\begin{bmatrix}32.63 & -0.31\\ -0.31 & 37.22\end{bmatrix}$ | $\begin{bmatrix}26.19 & 0.82\\ 0.82 & 57.23\end{bmatrix}$ |
| KLT | $\begin{bmatrix}71.72 & 44.36\\ 44.36 & 544.64\end{bmatrix}$ | $\begin{bmatrix}189.52 & 94.22\\ 94.22 & 1265.21\end{bmatrix}$ | $\begin{bmatrix}383.9 & 157.4\\ 157.4 & 2216.59\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.18 & 0.05\\ 0.05 & 0.23\end{bmatrix}$ | $\begin{bmatrix}3.2 & -2.68\\ -2.68 & 4.6\end{bmatrix}$ | $\begin{bmatrix}11.46 & -0.55\\ -0.55 & 9.45\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}20.84 & -1.09\\ -1.09 & 25.96\end{bmatrix}$ | $\begin{bmatrix}26.75 & -1.92\\ -1.92 & 35.73\end{bmatrix}$ | $\begin{bmatrix}30.75 & -1.47\\ -1.47 & 36.34\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}17.5 & -0.64\\ -0.64 & 19.73\end{bmatrix}$ | $\begin{bmatrix}19.82 & -0.42\\ -0.42 & 22.99\end{bmatrix}$ | $\begin{bmatrix}23.67 & -0.97\\ -0.97 & 27.78\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}16.37 & -0.13\\ -0.13 & 18.2\end{bmatrix}$ | $\begin{bmatrix}20.2 & -0.56\\ -0.56 & 23.81\end{bmatrix}$ | $\begin{bmatrix}21.37 & -1.02\\ -1.02 & 28.35\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}16.73 & -0.34\\ -0.34 & 16.57\end{bmatrix}$ | $\begin{bmatrix}18.52 & -0.7\\ -0.7 & 21.1\end{bmatrix}$ | $\begin{bmatrix}22.44 & -0.59\\ -0.59 & 23.15\end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix}0.46 & -0.81\\ -0.81 & 1.66\end{bmatrix}$ | $\begin{bmatrix}0.05 & -0.02\\ -0.02 & 0.32\end{bmatrix}$ | $\begin{bmatrix}0.08 & 0.14\\ 0.14 & 6.19\end{bmatrix}$ |
| SURF | $\begin{bmatrix}0.04 & -0.01\\ -0.01 & 0.13\end{bmatrix}$ | $\begin{bmatrix}0.07 & -0.01\\ -0.01 & 0.27\end{bmatrix}$ | $\begin{bmatrix}0.1 & -0.04\\ -0.04 & 0.48\end{bmatrix}$ |
| ORB | $\begin{bmatrix}0.28 & -0.03\\ -0.03 & 0.64\end{bmatrix}$ | $\begin{bmatrix}0.39 & -0.03\\ -0.03 & 0.97\end{bmatrix}$ | $\begin{bmatrix}0.45 & -0.07\\ -0.07 & 1.07\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}0.23 & -0.04\\ -0.04 & 1.2\end{bmatrix}$ | $\begin{bmatrix}0.27 & -0.02\\ -0.02 & 0.72\end{bmatrix}$ | $\begin{bmatrix}0.34 & -0.52\\ -0.52 & 5.97\end{bmatrix}$ |
| KLT | $\begin{bmatrix}69.88 & 46.08\\ 46.08 & 531.34\end{bmatrix}$ | $\begin{bmatrix}140.29 & 89.67\\ 89.67 & 1079.07\end{bmatrix}$ | $\begin{bmatrix}144.04 & 113.05\\ 113.05 & 1386.29\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.13 & -0.0\\ -0.0 & 0.18\end{bmatrix}$ | $\begin{bmatrix}0.17 & -0.0\\ -0.0 & 0.38\end{bmatrix}$ | $\begin{bmatrix}0.2 & -0.02\\ -0.02 & 0.42\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}5.75 & -0.33\\ -0.33 & 12.08\end{bmatrix}$ | $\begin{bmatrix}5.7 & -0.57\\ -0.57 & 15.91\end{bmatrix}$ | $\begin{bmatrix}5.95 & -1.08\\ -1.08 & 17.85\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}6.18 & -0.4\\ -0.4 & 10.55\end{bmatrix}$ | $\begin{bmatrix}6.57 & -0.5\\ -0.5 & 13.46\end{bmatrix}$ | $\begin{bmatrix}6.34 & -0.79\\ -0.79 & 16.03\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}6.12 & -0.21\\ -0.21 & 10.53\end{bmatrix}$ | $\begin{bmatrix}6.05 & -0.53\\ -0.53 & 13.71\end{bmatrix}$ | $\begin{bmatrix}6.12 & -0.97\\ -0.97 & 19.2\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}6.68 & -0.21\\ -0.21 & 9.97\end{bmatrix}$ | $\begin{bmatrix}6.37 & -0.42\\ -0.42 & 12.26\end{bmatrix}$ | $\begin{bmatrix}6.41 & -0.59\\ -0.59 & 14.59\end{bmatrix}$ |

## Table A.2

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Covariance [$px^2$] | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 5.51 & 1.69 \\ 1.69 & 12.29 \end{bmatrix}$ | $\begin{bmatrix} 19.82 & 2.2 \\ 2.2 & 32.51 \end{bmatrix}$ | $\begin{bmatrix} 53.86 & 4.82 \\ 4.82 & 91.83 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 2.23 & 0.28 \\ 0.28 & 2.84 \end{bmatrix}$ | $\begin{bmatrix} 4.89 & -0.03 \\ -0.03 & 5.7 \end{bmatrix}$ | $\begin{bmatrix} 24.24 & 0.11 \\ 0.11 & 38.14 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 0.13 & -0.0 \\ -0.0 & 0.47 \end{bmatrix}$ | $\begin{bmatrix} 1.23 & -0.63 \\ -0.63 & 5.18 \end{bmatrix}$ | $\begin{bmatrix} 4.97 & 2.6 \\ 2.6 & 8.34 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 0.29 & 0.03 \\ 0.03 & 0.21 \end{bmatrix}$ | $\begin{bmatrix} 1.65 & -0.25 \\ -0.25 & 2.35 \end{bmatrix}$ | $\begin{bmatrix} 4.0 & -0.03 \\ -0.03 & 10.65 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 29.68 & 1.41 \\ 1.41 & 129.54 \end{bmatrix}$ | $\begin{bmatrix} 54.07 & 1.93 \\ 1.93 & 235.46 \end{bmatrix}$ | $\begin{bmatrix} 28.8 & 0.47 \\ 0.47 & 320.25 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.06 & 0.02 \\ 0.02 & 0.27 \end{bmatrix}$ | $\begin{bmatrix} 0.58 & -0.47 \\ -0.47 & 1.27 \end{bmatrix}$ | $\begin{bmatrix} 1.27 & -0.1 \\ -0.1 & 0.85 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 21.57 & 0.52 \\ 0.52 & 25.88 \end{bmatrix}$ | $\begin{bmatrix} 24.96 & 0.77 \\ 0.77 & 29.47 \end{bmatrix}$ | $\begin{bmatrix} 27.01 & -0.41 \\ -0.41 & 32.78 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 18.03 & 0.15 \\ 0.15 & 20.49 \end{bmatrix}$ | $\begin{bmatrix} 19.51 & 0.0 \\ 0.0 & 23.7 \end{bmatrix}$ | $\begin{bmatrix} 21.46 & -0.28 \\ -0.28 & 26.38 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 17.74 & -0.58 \\ -0.58 & 22.3 \end{bmatrix}$ | $\begin{bmatrix} 20.13 & -0.3 \\ -0.3 & 22.88 \end{bmatrix}$ | $\begin{bmatrix} 22.22 & 0.12 \\ 0.12 & 28.53 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 14.66 & -0.58 \\ -0.58 & 15.61 \end{bmatrix}$ | $\begin{bmatrix} 17.19 & -0.63 \\ -0.63 & 18.33 \end{bmatrix}$ | $\begin{bmatrix} 18.92 & -1.37 \\ -1.37 & 21.52 \end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Covariance [$px^2$] | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 0.01 & -0.0 \\ -0.0 & 0.05 \end{bmatrix}$ | $\begin{bmatrix} 0.25 & -0.05 \\ -0.05 & 0.16 \end{bmatrix}$ | $\begin{bmatrix} 0.03 & 0.08 \\ 0.08 & 4.98 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 0.04 & -0.0 \\ -0.0 & 0.1 \end{bmatrix}$ | $\begin{bmatrix} 0.07 & -0.0 \\ -0.0 & 0.22 \end{bmatrix}$ | $\begin{bmatrix} 0.09 & -0.01 \\ -0.01 & 0.39 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 0.06 & 0.0 \\ 0.0 & 0.45 \end{bmatrix}$ | $\begin{bmatrix} 0.11 & -0.01 \\ -0.01 & 0.75 \end{bmatrix}$ | $\begin{bmatrix} 0.17 & -0.01 \\ -0.01 & 0.7 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 0.09 & 0.0 \\ 0.0 & 0.18 \end{bmatrix}$ | $\begin{bmatrix} 0.11 & 0.0 \\ 0.0 & 0.27 \end{bmatrix}$ | $\begin{bmatrix} 0.13 & 0.0 \\ 0.0 & 0.36 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 10.51 & 0.26 \\ 0.26 & 114.11 \end{bmatrix}$ | $\begin{bmatrix} 19.27 & 1.08 \\ 1.08 & 204.52 \end{bmatrix}$ | $\begin{bmatrix} 25.22 & 0.79 \\ 0.79 & 303.91 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.05 & 0.0 \\ 0.0 & 0.25 \end{bmatrix}$ | $\begin{bmatrix} 0.15 & -0.2 \\ -0.2 & 0.88 \end{bmatrix}$ | $\begin{bmatrix} 0.13 & 0.0 \\ 0.0 & 0.24 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 8.6 & -0.38 \\ -0.38 & 13.82 \end{bmatrix}$ | $\begin{bmatrix} 7.25 & -0.4 \\ -0.4 & 16.1 \end{bmatrix}$ | $\begin{bmatrix} 7.44 & -0.71 \\ -0.71 & 18.49 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 9.32 & -0.57 \\ -0.57 & 12.63 \end{bmatrix}$ | $\begin{bmatrix} 8.95 & -0.8 \\ -0.8 & 14.97 \end{bmatrix}$ | $\begin{bmatrix} 8.93 & -1.3 \\ -1.3 & 15.63 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 9.35 & -0.59 \\ -0.59 & 11.86 \end{bmatrix}$ | $\begin{bmatrix} 8.96 & -0.52 \\ -0.52 & 13.48 \end{bmatrix}$ | $\begin{bmatrix} 9.18 & -0.69 \\ -0.69 & 16.48 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 7.9 & -0.53 \\ -0.53 & 9.89 \end{bmatrix}$ | $\begin{bmatrix} 8.28 & -0.37 \\ -0.37 & 11.95 \end{bmatrix}$ | $\begin{bmatrix} 8.44 & -0.8 \\ -0.8 & 14.12 \end{bmatrix}$ |

### A.0.3. Eros

Table A.3: Covariance matrices (Eros).

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of 50.0 $Km$.

| | Covariance [$px^2$] | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 5.29 & 0.88 \\ 0.88 & 1.65 \end{bmatrix}$ | $\begin{bmatrix} 16.88 & 1.0 \\ 1.0 & 16.14 \end{bmatrix}$ | $\begin{bmatrix} 51.65 & 13.82 \\ 13.82 & 31.47 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 4.52 & 3.49 \\ 3.49 & 32.23 \end{bmatrix}$ | $\begin{bmatrix} 50.29 & 16.26 \\ 16.26 & 112.88 \end{bmatrix}$ | $\begin{bmatrix} 64.37 & 3.4 \\ 3.4 & 87.82 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 10.54 & -0.22 \\ -0.22 & 9.79 \end{bmatrix}$ | $\begin{bmatrix} 32.17 & 10.47 \\ 10.47 & 39.6 \end{bmatrix}$ | $\begin{bmatrix} 59.03 & 13.02 \\ 13.02 & 71.92 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 6.19 & -0.88 \\ -0.88 & 12.19 \end{bmatrix}$ | $\begin{bmatrix} 19.71 & -3.01 \\ -3.01 & 40.38 \end{bmatrix}$ | $\begin{bmatrix} 19.09 & 0.05 \\ 0.05 & 28.16 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 55.59 & -33.85 \\ -33.85 & 150.42 \end{bmatrix}$ | $\begin{bmatrix} 295.04 & -85.74 \\ -85.74 & 528.81 \end{bmatrix}$ | $\begin{bmatrix} 802.64 & -162.48 \\ -162.48 & 1093.04 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.48 & -0.03 \\ -0.03 & 1.61 \end{bmatrix}$ | $\begin{bmatrix} 6.5 & 0.36 \\ 0.36 & 24.36 \end{bmatrix}$ | $\begin{bmatrix} 22.75 & -0.36 \\ -0.36 & 80.82 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 42.3 & 2.62 \\ 2.62 & 64.92 \end{bmatrix}$ | $\begin{bmatrix} 76.8 & 1.27 \\ 1.27 & 81.22 \end{bmatrix}$ | $\begin{bmatrix} 110.29 & 5.31 \\ 5.31 & 127.06 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 35.11 & 3.69 \\ 3.69 & 45.02 \end{bmatrix}$ | $\begin{bmatrix} 61.96 & 12.25 \\ 12.25 & 62.69 \end{bmatrix}$ | $\begin{bmatrix} 84.81 & 3.54 \\ 3.54 & 74.56 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 30.33 & 2.97 \\ 2.97 & 35.27 \end{bmatrix}$ | $\begin{bmatrix} 43.08 & 3.95 \\ 3.95 & 42.99 \end{bmatrix}$ | $\begin{bmatrix} 67.11 & 2.78 \\ 2.78 & 59.15 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 20.9 & 0.49 \\ 0.49 & 23.15 \end{bmatrix}$ | $\begin{bmatrix} 25.47 & 1.96 \\ 1.96 & 33.56 \end{bmatrix}$ | $\begin{bmatrix} 35.26 & -2.71 \\ -2.71 & 33.86 \end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of 50.0 $Km$.

| | Covariance [$px^2$] | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 0.88 & 0.27 \\ 0.27 & 0.88 \end{bmatrix}$ | $\begin{bmatrix} 0.25 & -0.97 \\ -0.97 & 5.53 \end{bmatrix}$ | $\begin{bmatrix} 0.09 & 0.15 \\ 0.15 & 2.18 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 0.04 & -0.0 \\ -0.0 & 4.69 \end{bmatrix}$ | $\begin{bmatrix} 0.48 & -0.87 \\ -0.87 & 8.78 \end{bmatrix}$ | $\begin{bmatrix} 1.82 & -0.27 \\ -0.27 & 1.01 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 0.42 & 0.59 \\ 0.59 & 2.52 \end{bmatrix}$ | $\begin{bmatrix} 0.38 & -0.61 \\ -0.61 & 6.75 \end{bmatrix}$ | $\begin{bmatrix} 2.29 & -0.28 \\ -0.28 & 0.79 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 0.21 & 0.0 \\ 0.0 & 2.5 \end{bmatrix}$ | $\begin{bmatrix} 1.3 & -1.2 \\ -1.2 & 12.39 \end{bmatrix}$ | $\begin{bmatrix} 0.43 & -0.17 \\ -0.17 & 0.74 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 29.33 & -30.79 \\ -30.79 & 117.41 \end{bmatrix}$ | $\begin{bmatrix} 40.93 & -40.52 \\ -40.52 & 161.31 \end{bmatrix}$ | $\begin{bmatrix} 82.77 & -43.05 \\ -43.05 & 275.74 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.12 & 0.0 \\ 0.0 & 1.01 \end{bmatrix}$ | $\begin{bmatrix} 0.2 & -0.08 \\ -0.08 & 1.39 \end{bmatrix}$ | $\begin{bmatrix} 0.27 & -0.16 \\ -0.16 & 2.92 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 8.01 & -0.37 \\ -0.37 & 25.3 \end{bmatrix}$ | $\begin{bmatrix} 17.04 & -3.38 \\ -3.38 & 30.38 \end{bmatrix}$ | $\begin{bmatrix} 8.1 & -1.1 \\ -1.1 & 34.87 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 8.97 & -0.26 \\ -0.26 & 17.37 \end{bmatrix}$ | $\begin{bmatrix} 21.19 & 0.03 \\ 0.03 & 24.52 \end{bmatrix}$ | $\begin{bmatrix} 10.33 & 0.39 \\ 0.39 & 37.64 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 8.27 & -0.22 \\ -0.22 & 15.97 \end{bmatrix}$ | $\begin{bmatrix} 8.77 & -0.78 \\ -0.78 & 21.47 \end{bmatrix}$ | $\begin{bmatrix} 9.1 & -0.85 \\ -0.85 & 28.3 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 7.0 & -0.06 \\ -0.06 & 13.75 \end{bmatrix}$ | $\begin{bmatrix} 6.98 & -0.21 \\ -0.21 & 17.2 \end{bmatrix}$ | $\begin{bmatrix} 7.33 & -0.38 \\ -0.38 & 21.03 \end{bmatrix}$ |

Table A.3

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 70.0 $Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 3.86 & 0.34 \\ 0.34 & 0.83 \end{bmatrix}$ | $\begin{bmatrix} 12.05 & 1.22 \\ 1.22 & 8.44 \end{bmatrix}$ | $\begin{bmatrix} 11.92 & 2.06 \\ 2.06 & 22.68 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 4.82 & 0.18 \\ 0.18 & 3.62 \end{bmatrix}$ | $\begin{bmatrix} 13.51 & 5.19 \\ 5.19 & 19.26 \end{bmatrix}$ | $\begin{bmatrix} 34.57 & 9.88 \\ 9.88 & 51.69 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 2.23 & 0.51 \\ 0.51 & 3.63 \end{bmatrix}$ | $\begin{bmatrix} 4.58 & 0.07 \\ 0.07 & 11.94 \end{bmatrix}$ | $\begin{bmatrix} 16.88 & 1.22 \\ 1.22 & 19.13 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 0.62 & -0.08 \\ -0.08 & 0.47 \end{bmatrix}$ | $\begin{bmatrix} 3.11 & -0.33 \\ -0.33 & 3.25 \end{bmatrix}$ | $\begin{bmatrix} 11.98 & 0.89 \\ 0.89 & 7.03 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 13.85 & -9.58 \\ -9.58 & 78.17 \end{bmatrix}$ | $\begin{bmatrix} 66.86 & -23.79 \\ -23.79 & 220.15 \end{bmatrix}$ | $\begin{bmatrix} 295.42 & -33.4 \\ -33.4 & 479.92 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.58 & -0.01 \\ -0.01 & 14.34 \end{bmatrix}$ | $\begin{bmatrix} 1.38 & 0.35 \\ 0.35 & 36.76 \end{bmatrix}$ | $\begin{bmatrix} 5.57 & 4.71 \\ 4.71 & 98.84 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 43.5 & 6.24 \\ 6.24 & 58.73 \end{bmatrix}$ | $\begin{bmatrix} 59.06 & 12.74 \\ 12.74 & 77.06 \end{bmatrix}$ | $\begin{bmatrix} 126.53 & 19.76 \\ 19.76 & 119.46 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 33.9 & 1.63 \\ 1.63 & 46.14 \end{bmatrix}$ | $\begin{bmatrix} 51.5 & 10.48 \\ 10.48 & 77.66 \end{bmatrix}$ | $\begin{bmatrix} 53.77 & 11.19 \\ 11.19 & 89.32 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 29.25 & -0.75 \\ -0.75 & 25.49 \end{bmatrix}$ | $\begin{bmatrix} 30.93 & -0.37 \\ -0.37 & 33.4 \end{bmatrix}$ | $\begin{bmatrix} 72.62 & -9.71 \\ -9.71 & 43.96 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 20.2 & 0.37 \\ 0.37 & 20.15 \end{bmatrix}$ | $\begin{bmatrix} 21.88 & 0.21 \\ 0.21 & 25.83 \end{bmatrix}$ | $\begin{bmatrix} 36.17 & 6.84 \\ 6.84 & 38.67 \end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 70.0 $Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 0.14 & 0.09 \\ 0.09 & 0.28 \end{bmatrix}$ | $\begin{bmatrix} 0.07 & 0.48 \\ 0.48 & 6.06 \end{bmatrix}$ | $\begin{bmatrix} 0.04 & 0.01 \\ 0.01 & 0.9 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 0.71 & 0.22 \\ 0.22 & 0.42 \end{bmatrix}$ | $\begin{bmatrix} 0.09 & -0.01 \\ -0.01 & 0.21 \end{bmatrix}$ | $\begin{bmatrix} 0.11 & 0.02 \\ 0.02 & 0.4 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 0.24 & -0.01 \\ -0.01 & 0.44 \end{bmatrix}$ | $\begin{bmatrix} 0.25 & -0.01 \\ -0.01 & 0.51 \end{bmatrix}$ | $\begin{bmatrix} 3.16 & 0.15 \\ 0.15 & 0.71 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 0.18 & 0.0 \\ 0.0 & 0.27 \end{bmatrix}$ | $\begin{bmatrix} 0.21 & 0.0 \\ 0.0 & 0.34 \end{bmatrix}$ | $\begin{bmatrix} 0.26 & -0.01 \\ -0.01 & 0.4 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 8.89 & -10.44 \\ -10.44 & 75.27 \end{bmatrix}$ | $\begin{bmatrix} 17.94 & -18.82 \\ -18.82 & 122.05 \end{bmatrix}$ | $\begin{bmatrix} 33.75 & -28.54 \\ -28.54 & 162.13 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.09 & 0.0 \\ 0.0 & 13.95 \end{bmatrix}$ | $\begin{bmatrix} 0.14 & 0.0 \\ 0.0 & 31.49 \end{bmatrix}$ | $\begin{bmatrix} 0.19 & 0.03 \\ 0.03 & 0.4 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 10.39 & 0.23 \\ 0.23 & 25.32 \end{bmatrix}$ | $\begin{bmatrix} 9.16 & -0.3 \\ -0.3 & 33.78 \end{bmatrix}$ | $\begin{bmatrix} 22.24 & 5.37 \\ 5.37 & 44.97 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 12.46 & -0.01 \\ -0.01 & 23.75 \end{bmatrix}$ | $\begin{bmatrix} 23.85 & 8.95 \\ 8.95 & 36.37 \end{bmatrix}$ | $\begin{bmatrix} 13.84 & 0.55 \\ 0.55 & 51.81 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 10.51 & -0.05 \\ -0.05 & 16.58 \end{bmatrix}$ | $\begin{bmatrix} 11.13 & 0.16 \\ 0.16 & 21.33 \end{bmatrix}$ | $\begin{bmatrix} 54.2 & -11.87 \\ -11.87 & 28.88 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 7.2 & 0.13 \\ 0.13 & 13.38 \end{bmatrix}$ | $\begin{bmatrix} 7.3 & 0.01 \\ 0.01 & 17.51 \end{bmatrix}$ | $\begin{bmatrix} 7.72 & -0.37 \\ -0.37 & 21.99 \end{bmatrix}$ |

Table A.3

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of $50.0\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 17.39 & -0.19 \\ -0.19 & 5.25 \end{bmatrix}$ | $\begin{bmatrix} 54.12 & 3.58 \\ 3.58 & 35.11 \end{bmatrix}$ | $\begin{bmatrix} 142.14 & 12.07 \\ 12.07 & 101.38 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 21.98 & 9.98 \\ 9.98 & 22.41 \end{bmatrix}$ | $\begin{bmatrix} 86.06 & 11.24 \\ 11.24 & 111.44 \end{bmatrix}$ | $\begin{bmatrix} 116.94 & 3.29 \\ 3.29 & 177.34 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 16.37 & 0.93 \\ 0.93 & 15.82 \end{bmatrix}$ | $\begin{bmatrix} 63.57 & 3.39 \\ 3.39 & 44.99 \end{bmatrix}$ | $\begin{bmatrix} 135.5 & -4.46 \\ -4.46 & 149.25 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 9.95 & 2.78 \\ 2.78 & 20.81 \end{bmatrix}$ | $\begin{bmatrix} 24.33 & 1.26 \\ 1.26 & 11.04 \end{bmatrix}$ | $\begin{bmatrix} 41.14 & 3.71 \\ 3.71 & 38.86 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 68.35 & -56.33 \\ -56.33 & 247.47 \end{bmatrix}$ | $\begin{bmatrix} 282.5 & -127.33 \\ -127.33 & 730.51 \end{bmatrix}$ | $\begin{bmatrix} 781.85 & -233.11 \\ -233.11 & 1401.55 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 2.12 & -0.12 \\ -0.12 & 4.53 \end{bmatrix}$ | $\begin{bmatrix} 12.09 & 7.13 \\ 7.13 & 35.06 \end{bmatrix}$ | $\begin{bmatrix} 48.75 & 12.7 \\ 12.7 & 122.79 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 37.99 & 0.92 \\ 0.92 & 50.95 \end{bmatrix}$ | $\begin{bmatrix} 52.45 & 0.67 \\ 0.67 & 68.44 \end{bmatrix}$ | $\begin{bmatrix} 72.9 & -4.19 \\ -4.19 & 85.82 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 36.76 & -3.3 \\ -3.3 & 39.89 \end{bmatrix}$ | $\begin{bmatrix} 48.71 & 1.26 \\ 1.26 & 51.86 \end{bmatrix}$ | $\begin{bmatrix} 65.09 & 7.78 \\ 7.78 & 73.22 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 26.54 & 2.0 \\ 2.0 & 33.61 \end{bmatrix}$ | $\begin{bmatrix} 35.58 & 5.56 \\ 5.56 & 41.3 \end{bmatrix}$ | $\begin{bmatrix} 66.97 & 3.48 \\ 3.48 & 58.03 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 22.07 & -0.44 \\ -0.44 & 22.65 \end{bmatrix}$ | $\begin{bmatrix} 28.31 & -1.42 \\ -1.42 & 27.16 \end{bmatrix}$ | $\begin{bmatrix} 29.73 & -0.66 \\ -0.66 & 33.33 \end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of $50.0\ Km$.

| | Covariance $[px^2]$ | | |
|---|---|---|---|
| | 1 | 2 | 3 |
| SIFT | $\begin{bmatrix} 0.03 & -0.01 \\ -0.01 & 0.3 \end{bmatrix}$ | $\begin{bmatrix} 1.11 & 0.1 \\ 0.1 & 0.89 \end{bmatrix}$ | $\begin{bmatrix} 0.09 & 0.01 \\ 0.01 & 1.58 \end{bmatrix}$ |
| SURF | $\begin{bmatrix} 0.06 & 0.01 \\ 0.01 & 0.19 \end{bmatrix}$ | $\begin{bmatrix} 0.11 & 0.01 \\ 0.01 & 1.28 \end{bmatrix}$ | $\begin{bmatrix} 1.32 & 0.43 \\ 0.43 & 0.88 \end{bmatrix}$ |
| ORB | $\begin{bmatrix} 0.93 & -0.04 \\ -0.04 & 0.78 \end{bmatrix}$ | $\begin{bmatrix} 0.38 & -0.02 \\ -0.02 & 0.83 \end{bmatrix}$ | $\begin{bmatrix} 0.45 & -0.64 \\ -0.64 & 8.18 \end{bmatrix}$ |
| BRISK | $\begin{bmatrix} 0.2 & 0.0 \\ 0.0 & 6.05 \end{bmatrix}$ | $\begin{bmatrix} 0.3 & -0.05 \\ -0.05 & 0.76 \end{bmatrix}$ | $\begin{bmatrix} 0.29 & -0.04 \\ -0.04 & 1.43 \end{bmatrix}$ |
| KLT | $\begin{bmatrix} 43.81 & -49.89 \\ -49.89 & 200.68 \end{bmatrix}$ | $\begin{bmatrix} 63.71 & -69.65 \\ -69.65 & 275.19 \end{bmatrix}$ | $\begin{bmatrix} 183.44 & -88.36 \\ -88.36 & 405.41 \end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix} 0.13 & -0.0 \\ -0.0 & 1.18 \end{bmatrix}$ | $\begin{bmatrix} 0.19 & 0.06 \\ 0.06 & 1.25 \end{bmatrix}$ | $\begin{bmatrix} 0.23 & 0.0 \\ 0.0 & 0.7 \end{bmatrix}$ |
| SPV6 | $\begin{bmatrix} 7.3 & 0.75 \\ 0.75 & 24.52 \end{bmatrix}$ | $\begin{bmatrix} 7.73 & -0.85 \\ -0.85 & 32.0 \end{bmatrix}$ | $\begin{bmatrix} 18.6 & -2.68 \\ -2.68 & 40.03 \end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix} 17.63 & -1.37 \\ -1.37 & 21.33 \end{bmatrix}$ | $\begin{bmatrix} 18.75 & -2.93 \\ -2.93 & 26.72 \end{bmatrix}$ | $\begin{bmatrix} 24.94 & 13.24 \\ 13.24 & 46.01 \end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix} 7.65 & -0.32 \\ -0.32 & 18.28 \end{bmatrix}$ | $\begin{bmatrix} 8.11 & -0.41 \\ -0.41 & 22.4 \end{bmatrix}$ | $\begin{bmatrix} 8.95 & 0.51 \\ 0.51 & 30.54 \end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix} 6.56 & -0.04 \\ -0.04 & 13.32 \end{bmatrix}$ | $\begin{bmatrix} 6.85 & -0.16 \\ -0.16 & 16.57 \end{bmatrix}$ | $\begin{bmatrix} 6.91 & -0.28 \\ -0.28 & 20.19 \end{bmatrix}$ |

Table A.3

(a) Covariance matrices (without outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 70.0 $Km$.

| | Covariance $[px^2]$ 1 | 2 | 3 |
|---|---|---|---|
| SIFT | $\begin{bmatrix}11.85 & 0.79\\0.79 & 2.13\end{bmatrix}$ | $\begin{bmatrix}24.83 & 2.47\\2.47 & 30.21\end{bmatrix}$ | $\begin{bmatrix}51.77 & 0.15\\0.15 & 38.15\end{bmatrix}$ |
| SURF | $\begin{bmatrix}5.14 & 0.7\\0.7 & 6.21\end{bmatrix}$ | $\begin{bmatrix}22.37 & 2.6\\2.6 & 43.83\end{bmatrix}$ | $\begin{bmatrix}47.98 & 3.14\\3.14 & 67.63\end{bmatrix}$ |
| ORB | $\begin{bmatrix}4.41 & 0.58\\0.58 & 1.43\end{bmatrix}$ | $\begin{bmatrix}14.55 & 6.19\\6.19 & 15.48\end{bmatrix}$ | $\begin{bmatrix}30.46 & 0.92\\0.92 & 30.6\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}2.85 & 0.85\\0.85 & 3.89\end{bmatrix}$ | $\begin{bmatrix}7.2 & 0.67\\0.67 & 22.0\end{bmatrix}$ | $\begin{bmatrix}17.8 & 1.91\\1.91 & 22.11\end{bmatrix}$ |
| KLT | $\begin{bmatrix}18.3 & -15.95\\-15.95 & 114.08\end{bmatrix}$ | $\begin{bmatrix}72.42 & -37.03\\-37.03 & 295.54\end{bmatrix}$ | $\begin{bmatrix}255.67 & -60.73\\-60.73 & 563.97\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}1.18 & 1.07\\1.07 & 15.47\end{bmatrix}$ | $\begin{bmatrix}5.1 & 3.64\\3.64 & 52.99\end{bmatrix}$ | $\begin{bmatrix}12.71 & 5.0\\5.0 & 101.37\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}35.98 & 2.16\\2.16 & 54.59\end{bmatrix}$ | $\begin{bmatrix}40.33 & -0.26\\-0.26 & 73.9\end{bmatrix}$ | $\begin{bmatrix}54.58 & 2.9\\2.9 & 74.46\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}24.85 & -0.08\\-0.08 & 39.51\end{bmatrix}$ | $\begin{bmatrix}29.79 & 0.22\\0.22 & 52.67\end{bmatrix}$ | $\begin{bmatrix}50.92 & 8.87\\8.87 & 82.49\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}23.46 & -0.04\\-0.04 & 24.5\end{bmatrix}$ | $\begin{bmatrix}30.23 & -0.49\\-0.49 & 31.38\end{bmatrix}$ | $\begin{bmatrix}42.85 & 0.06\\0.06 & 39.05\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}19.05 & 0.37\\0.37 & 20.04\end{bmatrix}$ | $\begin{bmatrix}23.42 & -0.08\\-0.08 & 26.09\end{bmatrix}$ | $\begin{bmatrix}26.44 & 0.65\\0.65 & 32.5\end{bmatrix}$ |

(b) Covariance matrices (with outliers rejection), for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 70.0 $Km$.

| | Covariance $[px^2]$ 1 | 2 | 3 |
|---|---|---|---|
| SIFT | $\begin{bmatrix}0.02 & 0.01\\0.01 & 0.28\end{bmatrix}$ | $\begin{bmatrix}0.04 & 0.02\\0.02 & 0.74\end{bmatrix}$ | $\begin{bmatrix}0.05 & -0.01\\-0.01 & 1.0\end{bmatrix}$ |
| SURF | $\begin{bmatrix}0.06 & -0.0\\-0.0 & 0.15\end{bmatrix}$ | $\begin{bmatrix}1.09 & 0.36\\0.36 & 5.69\end{bmatrix}$ | $\begin{bmatrix}0.16 & -0.04\\-0.04 & 5.48\end{bmatrix}$ |
| ORB | $\begin{bmatrix}0.54 & -0.05\\-0.05 & 0.5\end{bmatrix}$ | $\begin{bmatrix}0.3 & 0.01\\0.01 & 0.65\end{bmatrix}$ | $\begin{bmatrix}0.46 & 0.72\\0.72 & 8.73\end{bmatrix}$ |
| BRISK | $\begin{bmatrix}0.27 & 0.14\\0.14 & 0.54\end{bmatrix}$ | $\begin{bmatrix}0.22 & -0.07\\-0.07 & 0.77\end{bmatrix}$ | $\begin{bmatrix}0.23 & 0.06\\0.06 & 0.8\end{bmatrix}$ |
| KLT | $\begin{bmatrix}13.34 & -15.95\\-15.95 & 110.4\end{bmatrix}$ | $\begin{bmatrix}29.05 & -33.1\\-33.1 & 195.86\end{bmatrix}$ | $\begin{bmatrix}56.1 & -55.03\\-55.03 & 267.19\end{bmatrix}$ |
| STAR + BRIEF | $\begin{bmatrix}0.1 & 0.0\\0.0 & 11.97\end{bmatrix}$ | $\begin{bmatrix}0.15 & 0.01\\0.01 & 11.3\end{bmatrix}$ | $\begin{bmatrix}0.19 & 0.02\\0.02 & 59.36\end{bmatrix}$ |
| SPV6 | $\begin{bmatrix}9.2 & -0.1\\-0.1 & 26.36\end{bmatrix}$ | $\begin{bmatrix}8.82 & -0.82\\-0.82 & 37.29\end{bmatrix}$ | $\begin{bmatrix}8.8 & -0.24\\-0.24 & 37.54\end{bmatrix}$ |
| SuperPoint Base | $\begin{bmatrix}10.79 & 0.33\\0.33 & 27.14\end{bmatrix}$ | $\begin{bmatrix}11.69 & 0.21\\0.21 & 31.44\end{bmatrix}$ | $\begin{bmatrix}12.61 & -0.04\\-0.04 & 38.85\end{bmatrix}$ |
| SuperPoint Eros | $\begin{bmatrix}10.28 & 0.05\\0.05 & 15.7\end{bmatrix}$ | $\begin{bmatrix}9.85 & 0.43\\0.43 & 19.18\end{bmatrix}$ | $\begin{bmatrix}11.28 & 0.08\\0.08 & 25.93\end{bmatrix}$ |
| SuperPoint Bennu | $\begin{bmatrix}7.31 & 0.01\\0.01 & 12.99\end{bmatrix}$ | $\begin{bmatrix}7.37 & 0.26\\0.26 & 16.75\end{bmatrix}$ | $\begin{bmatrix}7.39 & -0.01\\-0.01 & 20.59\end{bmatrix}$ |

# B | Appendix B - Average errors

## Bennu

In this appendix, the tables containing the average errors are reported.

Table B.1: Average norm of the error (in pixels) committed on the matches on the asteroid Bennu.

(a) Average error committed on the matches on the asteroid Bennu, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.291217 | 0.743627 | 1.03101 | 0.0731754 | 0.118926 | 0.155445 |
| SURF | 0.484176 | 1.73506 | 2.99815 | 0.242747 | 0.356075 | 0.43683 |
| ORB | 0.956696 | 1.18814 | 1.65021 | 0.771635 | 0.807276 | 0.832606 |
| BRISK | 0.305668 | 0.44394 | 0.562708 | 0.269507 | 0.299123 | 0.316808 |
| KLT | 3.10621 | 17.1042 | 34.498 | 2.45052 | 7.96557 | 20.0635 |
| STAR + BRIEF | 0.446622 | 0.594781 | 0.919536 | 0.440504 | 0.525326 | 0.588074 |
| SPV6 | 6.47323 | 7.38328 | 8.17899 | 3.82061 | 4.22307 | 4.7556 |
| SuperPoint Base | 5.34922 | 5.8157 | 6.52354 | 3.823 | 4.09876 | 4.51877 |
| SuperPoint Eros | 5.11747 | 5.5061 | 5.69546 | 3.79096 | 4.05302 | 4.28907 |
| SuperPoint Bennu | 5.14753 | 5.51475 | 5.89969 | 3.76922 | 3.99517 | 4.29543 |

(b) Average error committed on the matches on the asteroid Bennu, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 2.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| IP step | W/o outliers rejection | | | W/ outliers rejection | | |
| | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.206203 | 0.433173 | 0.685509 | 0.140666 | 0.219289 | 0.295027 |
| SURF | 0.424673 | 1.2168 | 2.01292 | 0.303044 | 0.428635 | 0.518706 |
| ORB | 0.783203 | 0.994513 | 1.15083 | 0.678865 | 0.77582 | 0.813902 |
| BRISK | 0.261214 | 0.342329 | 0.447961 | 0.249789 | 0.285481 | 0.301315 |
| KLT | 0.839056 | 2.77351 | 6.84449 | 0.818261 | 2.22574 | 4.36437 |
| STAR + BRIEF | 0.407013 | 0.5631 | 0.767915 | 0.402778 | 0.523516 | 0.607799 |
| SPV6 | 6.91258 | 8.43501 | 9.38469 | 4.49334 | 5.3283 | 6.04738 |
| SuperPoint Base | 5.88752 | 7.05161 | 7.72411 | 4.5657 | 5.39983 | 5.92458 |
| SuperPoint Eros | 5.66651 | 6.56686 | 6.95788 | 4.38574 | 4.98591 | 5.32741 |
| SuperPoint Bennu | 4.98761 | 5.85312 | 6.4039 | 3.79882 | 4.30387 | 4.79177 |

Table B.1

(a) Average error committed on the matches on the asteroid Bennu, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.459926 | 1.14708 | 2.07697 | 0.0825204 | 0.129573 | 0.175069 |
| SURF | 0.473546 | 1.82762 | 3.32643 | 0.24422 | 0.36659 | 0.465163 |
| ORB | 1.03311 | 1.39962 | 1.9695 | 0.792761 | 0.839975 | 0.885064 |
| BRISK | 0.435053 | 0.561525 | 0.855947 | 0.305671 | 0.340313 | 0.370095 |
| KLT | 3.17982 | 16.5829 | 34.3111 | 2.48651 | 7.47934 | 18.2594 |
| STAR + BRIEF | 0.45168 | 0.600282 | 0.967989 | 0.446705 | 0.535308 | 0.604364 |
| SPV6 | 6.06332 | 7.13475 | 8.2888 | 3.73962 | 4.17553 | 4.7351 |
| SuperPoint Base | 5.11357 | 5.65702 | 6.13489 | 3.68813 | 4.02613 | 4.35203 |
| SuperPoint Eros | 4.93294 | 5.27625 | 5.49798 | 3.72224 | 3.95876 | 4.18565 |
| SuperPoint Bennu | 5.01912 | 5.3874 | 5.80319 | 3.66304 | 3.88311 | 4.18814 |

(b) Average error committed on the matches on the asteroid Bennu, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 2.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.254388 | 0.527442 | 0.986757 | 0.147581 | 0.23546 | 0.311097 |
| SURF | 0.451502 | 1.49278 | 2.26055 | 0.320028 | 0.46313 | 0.551726 |
| ORB | 0.804428 | 1.04065 | 1.27803 | 0.693443 | 0.80106 | 0.856363 |
| BRISK | 0.305121 | 0.446191 | 0.511166 | 0.285335 | 0.33192 | 0.356105 |
| KLT | 0.855882 | 2.81985 | 6.93393 | 0.815626 | 2.17114 | 4.33124 |
| STAR + BRIEF | 0.430399 | 0.602837 | 0.848284 | 0.425907 | 0.552566 | 0.645872 |
| SPV6 | 6.28856 | 7.90026 | 9.22177 | 4.18023 | 5.06226 | 5.99557 |
| SuperPoint Base | 5.57356 | 6.49439 | 7.32907 | 4.25681 | 4.94445 | 5.63262 |
| SuperPoint Eros | 5.38009 | 6.13746 | 6.47982 | 4.11945 | 4.67623 | 5.05809 |
| SuperPoint Bennu | 4.83544 | 5.70156 | 6.26917 | 3.66907 | 4.22166 | 4.67057 |

## Itokawa

Table B.2: Average norm of the error (in pixels) committed on the matches on the asteroid Itokawa.

(a) Average error committed on the matches on the asteroid Itokawa, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.216511 | 0.500041 | 1.03622 | 0.130287 | 0.191606 | 0.249063 |
| SURF | 0.198219 | 0.356744 | 0.701212 | 0.172244 | 0.24791 | 0.319711 |
| ORB | 0.866468 | 1.24862 | 1.46722 | 0.736803 | 0.849105 | 0.913377 |
| BRISK | 0.729858 | 0.937449 | 1.02478 | 0.602021 | 0.683984 | 0.740146 |
| KLT | 1.66213 | 4.9163 | 10.9012 | 1.56311 | 3.11704 | 4.34792 |
| STAR + BRIEF | 0.427202 | 0.50354 | 0.59004 | 0.422022 | 0.487181 | 0.539769 |
| SPV6 | 4.06176 | 4.57089 | 4.95134 | 2.94776 | 3.20498 | 3.37419 |
| SuperPoint Base | 3.71939 | 4.16691 | 4.48674 | 2.94647 | 3.19858 | 3.38488 |
| SuperPoint Eros | 3.73371 | 4.16607 | 4.48873 | 3.00793 | 3.29608 | 3.47788 |
| SuperPoint Bennu | 3.87324 | 4.23303 | 4.48453 | 3.06498 | 3.26883 | 3.40094 |

(b) Average error committed on the matches on the asteroid Itokawa, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.0959819 | 0.181005 | 0.307767 | 0.082344 | 0.113503 | 0.160701 |
| SURF | 0.211415 | 0.333669 | 0.497004 | 0.19594 | 0.277073 | 0.33943 |
| ORB | 0.579671 | 0.728393 | 0.87204 | 0.54639 | 0.647687 | 0.727437 |
| BRISK | 0.40224 | 0.454355 | 0.530847 | 0.375259 | 0.400452 | 0.443719 |
| KLT | 1.23195 | 2.12656 | 2.73716 | 0.812931 | 1.69602 | 1.98297 |
| STAR + BRIEF | 0.315305 | 0.411495 | 0.470778 | 0.313249 | 0.398307 | 0.457069 |
| SPV6 | 3.96514 | 4.43745 | 4.82024 | 3.06417 | 3.23624 | 3.47122 |
| SuperPoint Base | 3.70567 | 4.03612 | 4.44695 | 3.07423 | 3.2703 | 3.52305 |
| SuperPoint Eros | 3.6693 | 4.09709 | 4.46021 | 3.08095 | 3.33989 | 3.59237 |
| SuperPoint Bennu | 3.50888 | 3.92804 | 4.35763 | 2.94528 | 3.19975 | 3.51271 |

Table B.2

(a) Average error committed on the matches on the asteroid Itokawa, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.522039 | 1.38336 | 2.43583 | 0.167303 | 0.249085 | 0.336597 |
| SURF | 0.279478 | 0.724474 | 1.40887 | 0.220737 | 0.326611 | 0.419855 |
| ORB | 0.923269 | 1.27987 | 1.72348 | 0.772558 | 0.911664 | 0.99501 |
| BRISK | 0.768842 | 1.00299 | 1.14308 | 0.629646 | 0.734764 | 0.818935 |
| KLT | 1.5527 | 4.71249 | 10.3493 | 1.43882 | 2.98196 | 4.08278 |
| STAR + BRIEF | 0.471853 | 0.575168 | 0.702143 | 0.4669 | 0.540208 | 0.61109 |
| SPV6 | 3.92256 | 4.44329 | 4.76898 | 2.93293 | 3.20086 | 3.34633 |
| SuperPoint Base | 3.73924 | 4.14098 | 4.45122 | 2.9704 | 3.2214 | 3.39778 |
| SuperPoint Eros | 3.72502 | 4.13479 | 4.39737 | 3.01545 | 3.26842 | 3.42976 |
| SuperPoint Bennu | 3.83345 | 4.18489 | 4.44687 | 3.04008 | 3.24118 | 3.39402 |

(b) Average error committed on the matches on the asteroid Itokawa, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.156651 | 0.365164 | 0.748808 | 0.0973535 | 0.148335 | 0.198272 |
| SURF | 0.249052 | 0.408655 | 0.696337 | 0.221092 | 0.322918 | 0.410066 |
| ORB | 0.571685 | 0.72917 | 0.873525 | 0.544006 | 0.645096 | 0.725552 |
| BRISK | 0.377961 | 0.486252 | 0.584891 | 0.352977 | 0.427801 | 0.479933 |
| KLT | 0.793978 | 1.54538 | 2.24549 | 0.676755 | 1.27097 | 1.95084 |
| STAR + BRIEF | 0.31388 | 0.432465 | 0.517355 | 0.312415 | 0.423821 | 0.490404 |
| SPV6 | 3.88545 | 4.24121 | 4.63054 | 2.9922 | 3.14875 | 3.39517 |
| SuperPoint Base | 3.66707 | 3.99528 | 4.28972 | 3.01395 | 3.22272 | 3.3903 |
| SuperPoint Eros | 3.63929 | 4.01844 | 4.43017 | 3.03313 | 3.26269 | 3.53902 |
| SuperPoint Bennu | 3.48006 | 3.90333 | 4.27659 | 2.90009 | 3.18943 | 3.42174 |

## Eros

Table B.3: Average norm of the error (in pixels) committed on the matches on the asteroid Eros.

(a) Average error committed on the matches on the asteroid Eros, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 50.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.166904 | 0.2998 | 0.499904 | 0.136696 | 0.198016 | 0.25086 |
| SURF | 0.32792 | 0.671484 | 0.87701 | 0.252538 | 0.316314 | 0.396293 |
| ORB | 0.78984 | 1.0334 | 1.25892 | 0.671978 | 0.76441 | 0.810218 |
| BRISK | 0.770438 | 1.01297 | 0.980346 | 0.596897 | 0.721325 | 0.648652 |
| KLT | 2.04792 | 10.2333 | 20.0796 | 0.9852 | 2.24207 | 4.72877 |
| STAR + BRIEF | 0.439561 | 0.610468 | 0.891343 | 0.425861 | 0.517306 | 0.587384 |
| SPV6 | 5.1806 | 6.0667 | 6.92129 | 3.43742 | 3.84538 | 4.04555 |
| SuperPoint Base | 4.52114 | 5.26779 | 5.7865 | 3.40501 | 3.80327 | 4.07309 |
| SuperPoint Eros | 4.47755 | 5.1254 | 5.61076 | 3.43851 | 3.79871 | 4.05383 |
| SuperPoint Bennu | 4.3693 | 4.8708 | 5.18319 | 3.35586 | 3.62177 | 3.80805 |

(b) Average error committed on the matches on the asteroid Eros, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 70.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.142704 | 0.248869 | 0.359949 | 0.114917 | 0.174702 | 0.219369 |
| SURF | 0.269772 | 0.474669 | 0.751651 | 0.214046 | 0.296289 | 0.362233 |
| ORB | 0.678027 | 0.861085 | 1.04439 | 0.598798 | 0.69746 | 0.768455 |
| BRISK | 0.554994 | 0.66283 | 0.758003 | 0.467575 | 0.521285 | 0.555988 |
| KLT | 0.980769 | 4.72924 | 10.7867 | 0.731543 | 1.78983 | 3.28009 |
| STAR + BRIEF | 0.412139 | 0.557451 | 0.748599 | 0.391243 | 0.50893 | 0.526027 |
| SPV6 | 4.97129 | 5.78175 | 6.79339 | 3.46221 | 3.84015 | 4.03849 |
| SuperPoint Base | 4.43617 | 5.1873 | 5.76021 | 3.48812 | 3.89329 | 4.3506 |
| SuperPoint Eros | 4.26552 | 4.93519 | 5.51006 | 3.44709 | 3.80487 | 4.16842 |
| SuperPoint Bennu | 4.1835 | 4.66948 | 5.05882 | 3.29904 | 3.59963 | 3.83513 |

Table B.3

(a) Average error committed on the matches on the asteroid Eros, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 50.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.255914 | 0.53473 | 1.06223 | 0.168918 | 0.254692 | 0.330139 |
| SURF | 0.379516 | 0.878372 | 1.26718 | 0.252304 | 0.357192 | 0.447754 |
| ORB | 0.877889 | 1.23362 | 1.66343 | 0.719766 | 0.831467 | 0.90741 |
| BRISK | 0.814743 | 0.935305 | 1.14781 | 0.623961 | 0.660674 | 0.713481 |
| KLT | 2.22936 | 9.70402 | 19.1315 | 1.18378 | 2.36833 | 4.87352 |
| STAR + BRIEF | 0.480911 | 0.674001 | 1.09642 | 0.455849 | 0.553898 | 0.622843 |
| SPV6 | 5.05827 | 5.95383 | 6.697 | 3.42799 | 3.86898 | 4.18702 |
| SuperPoint Base | 4.59887 | 5.35351 | 5.92028 | 3.49799 | 3.90621 | 4.23514 |
| SuperPoint Eros | 4.36418 | 5.0835 | 5.5705 | 3.40054 | 3.7718 | 4.04487 |
| SuperPoint Bennu | 4.26211 | 4.73377 | 5.03716 | 3.2794 | 3.53198 | 3.69641 |

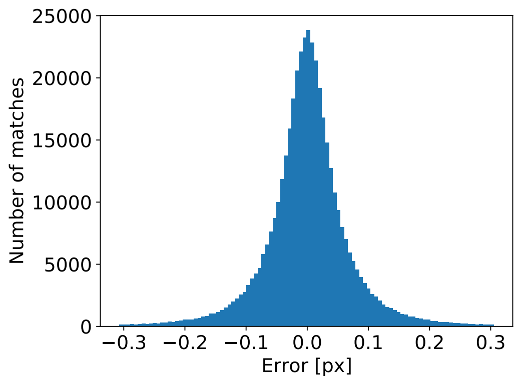
(b) Average error committed on the matches on the asteroid Eros, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 70.0 $Km$.

| | Average error [px] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.197792 | 0.410871 | 0.638861 | 0.134989 | 0.207849 | 0.262077 |
| SURF | 0.330402 | 0.648951 | 0.976322 | 0.2491 | 0.355459 | 0.43186 |
| ORB | 0.723495 | 0.97698 | 1.19219 | 0.632842 | 0.754208 | 0.832907 |
| BRISK | 0.603502 | 0.76885 | 0.883085 | 0.492271 | 0.560104 | 0.602964 |
| KLT | 1.14316 | 4.63908 | 10.2917 | 0.86187 | 2.03894 | 3.68129 |
| STAR + BRIEF | 0.443596 | 0.638476 | 0.868948 | 0.415329 | 0.511732 | 0.655313 |
| SPV6 | 4.9114 | 5.77157 | 6.54438 | 3.49097 | 3.91164 | 4.24268 |
| SuperPoint Base | 4.41909 | 5.07966 | 5.80822 | 3.49882 | 3.884 | 4.27625 |
| SuperPoint Eros | 4.24555 | 4.94103 | 5.42482 | 3.39844 | 3.73329 | 4.07079 |
| SuperPoint Bennu | 4.11402 | 4.63037 | 5.00949 | 3.25369 | 3.53415 | 3.74194 |

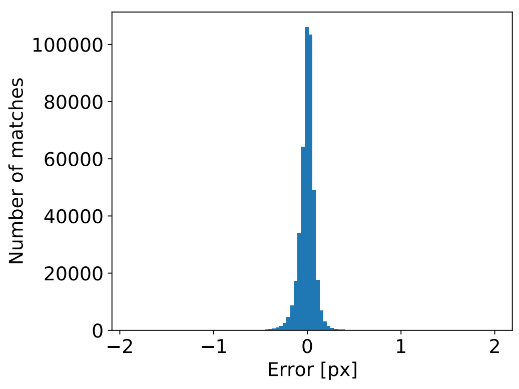
# C | Appendix C - Errors distributions

In this appendix, the distributions of the errors, in $x$ and $y$ directions, are reported.

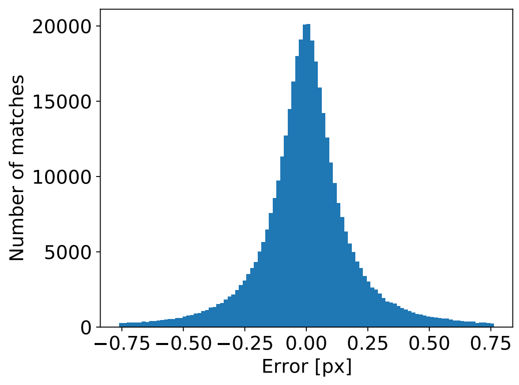
## C.0.1. Bennu



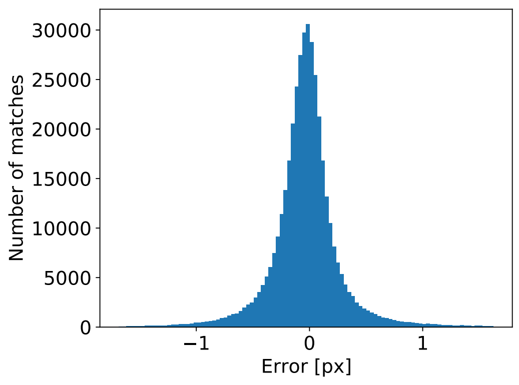
(a) SIFT, errors in $x$ direction.



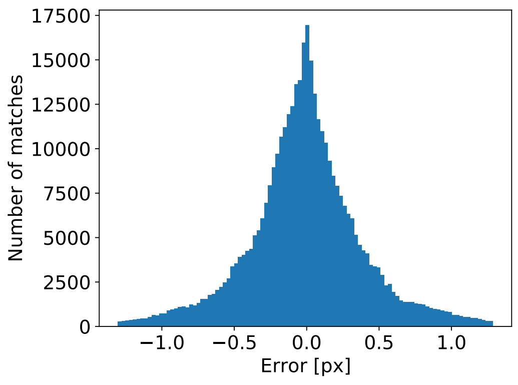
(b) SIFT, errors in $y$ direction.



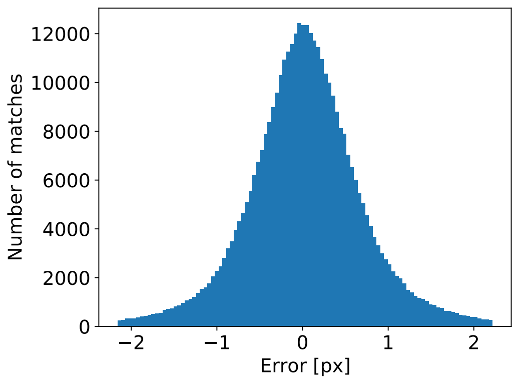
(c) SURF, errors in $x$ direction.



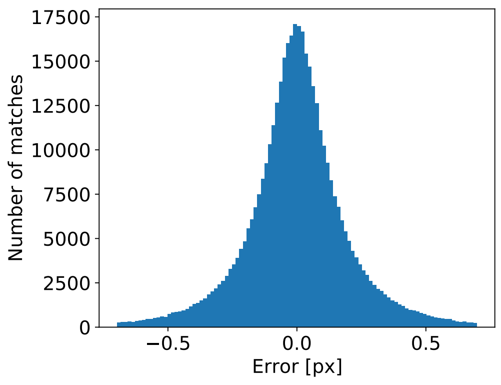
(d) SURF, errors in $y$ direction.



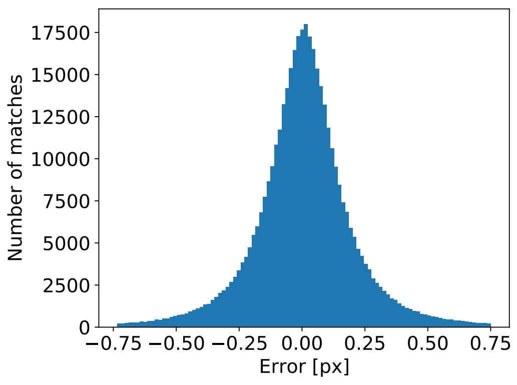
(e) ORB, errors in $x$ direction.



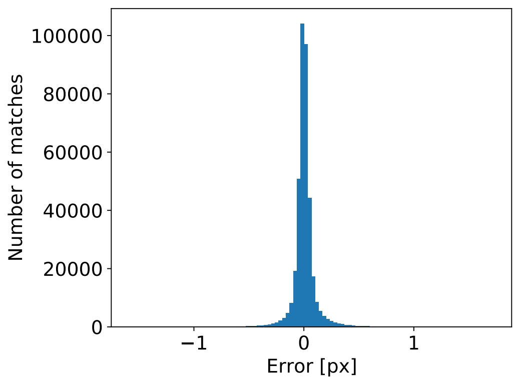
(f) ORB, errors in $y$ direction.



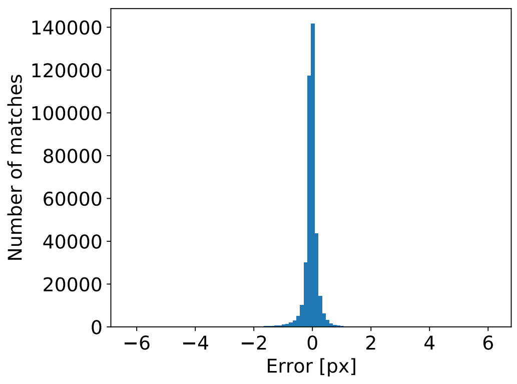
(g) BRISK, errors in $x$ direction.



(h) BRISK, errors in $y$ direction.

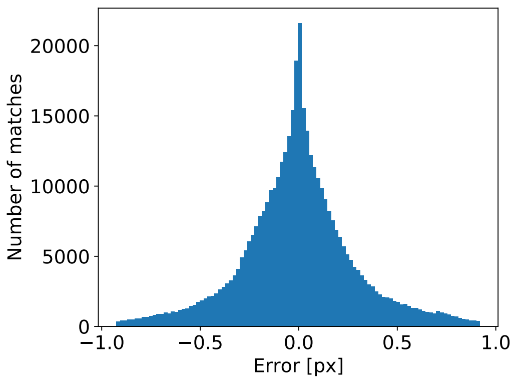


(i) KLT, errors in $x$ direction.

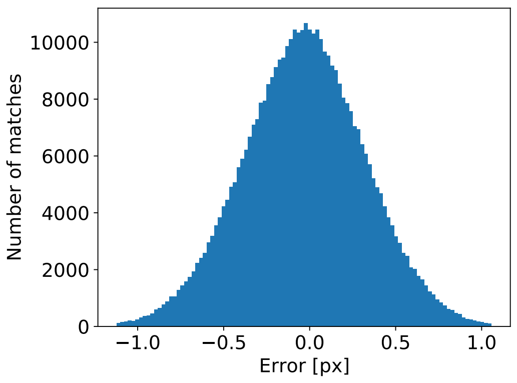


(j) KLT, errors in $y$ direction.

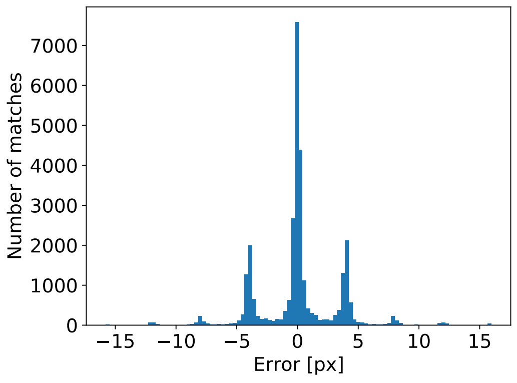
Figure C.0



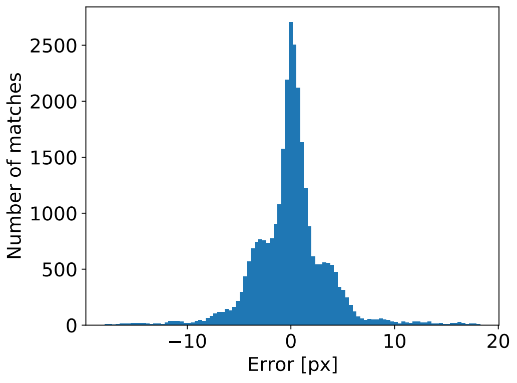
(k) STAR + BRIEF, errors in $x$ direction.



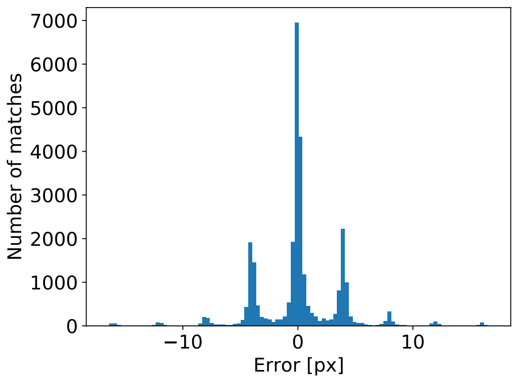
(l) STAR + BRIEF, errors in $y$ direction.



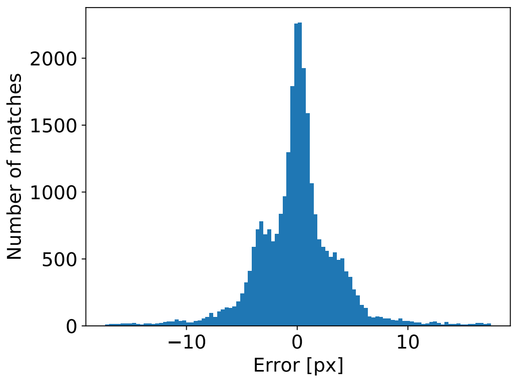
(m) SPV6, errors in $x$ direction.



(n) SPV6, errors in $y$ direction.

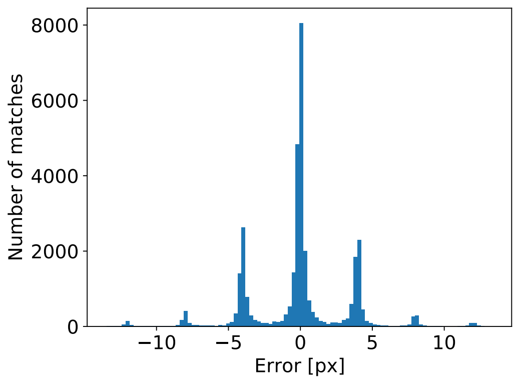


(o) SuperPoint Base, errors in $x$ direction.

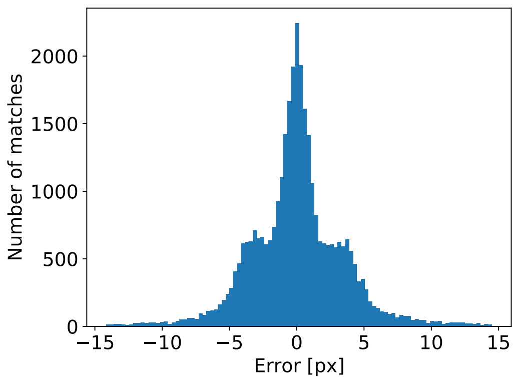


(p) SuperPoint Base, errors in $y$ direction.

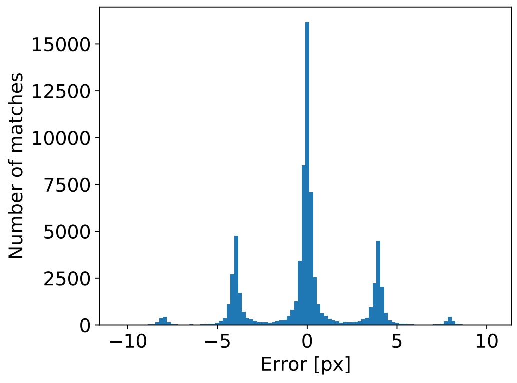
Figure C.0



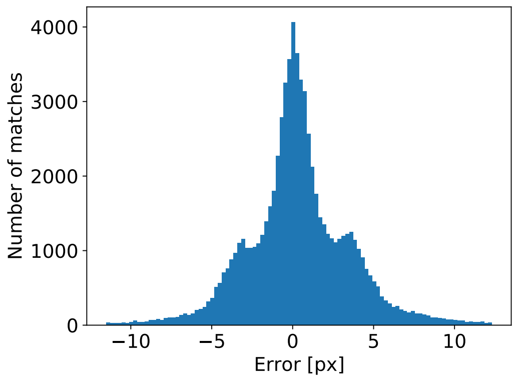
(q) SuperPoint Eros, errors in $x$ direction.



(r) SuperPoint Eros, errors in $y$ direction.



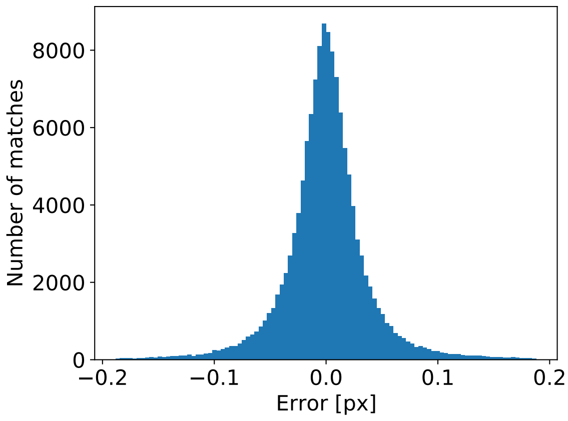
(s) SuperPoint Bennu, errors in $x$ direction.



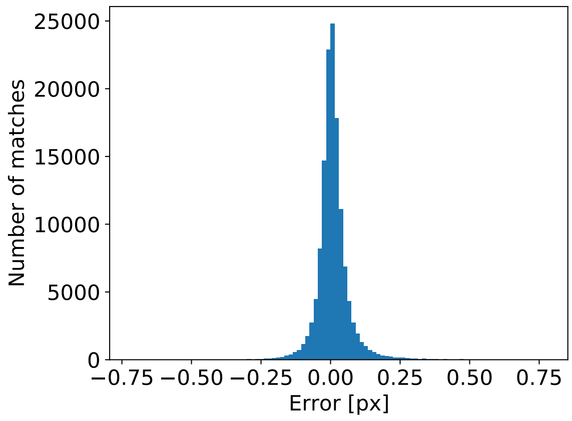
(t) SuperPoint Bennu, errors in $y$ direction.

Figure C.0: Error distribution in $x$ and $y$ direction for the asteroid Bennu. Ideally, the distributions should be Gaussian.

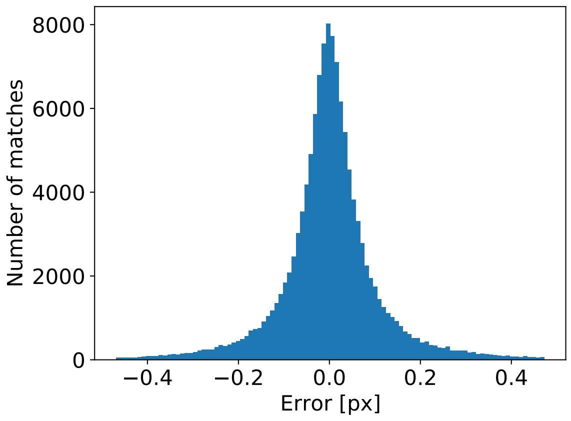
### C.0.2. Itokawa



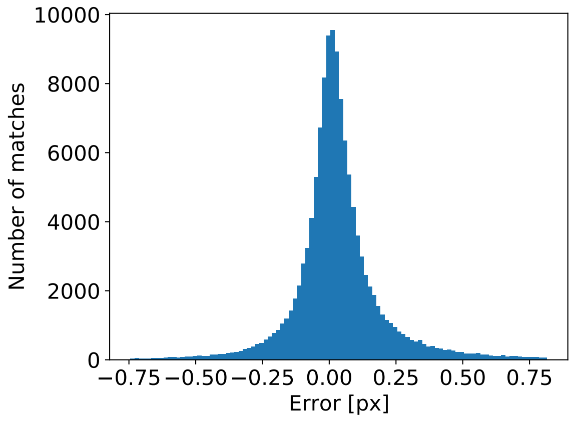
(a) SIFT, errors in $x$ direction.



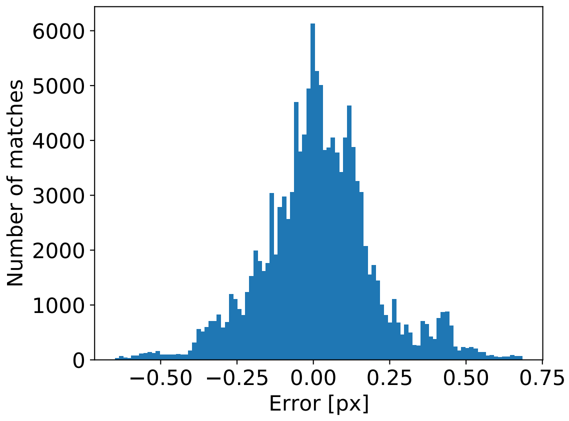
(b) SIFT, errors in $y$ direction.



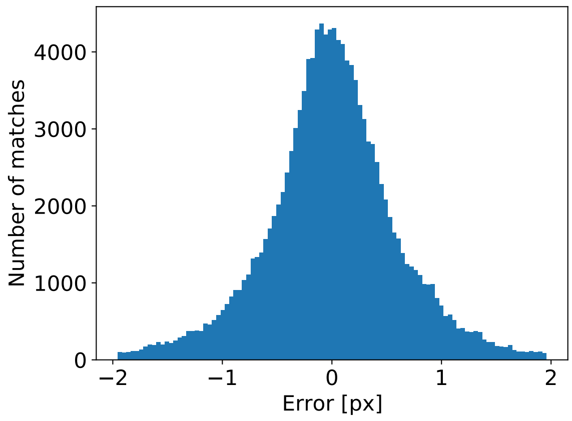
(c) SURF, errors in $x$ direction.



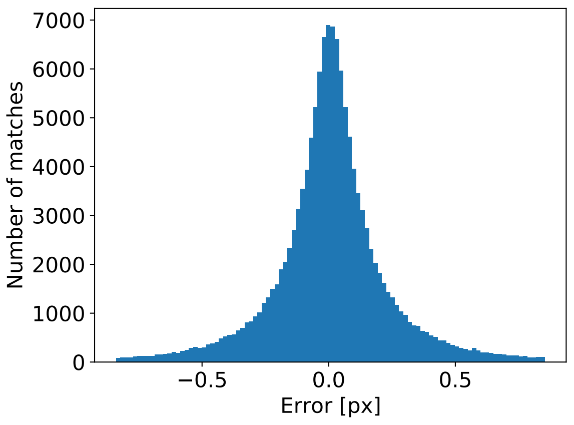
(d) SURF, errors in $y$ direction.



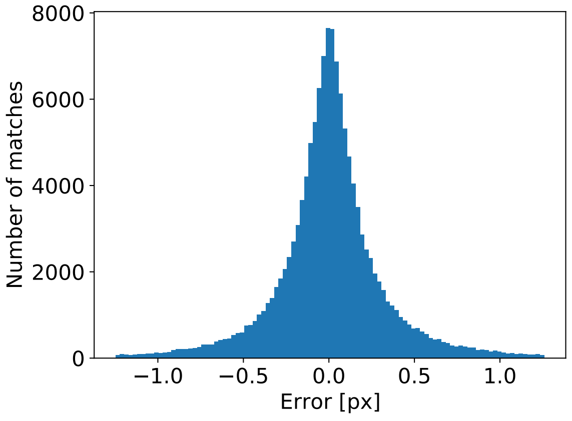
(e) ORB, errors in $x$ direction.



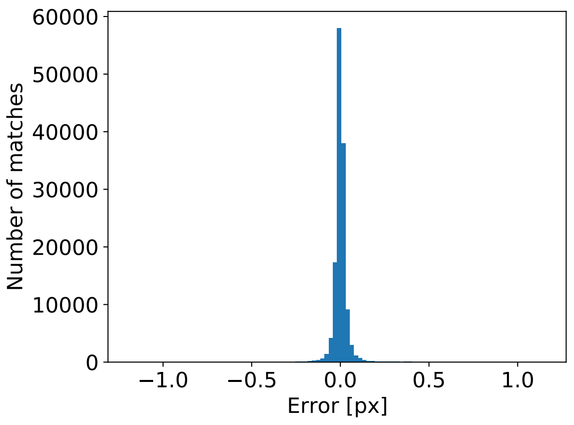
(f) ORB, errors in $y$ direction.



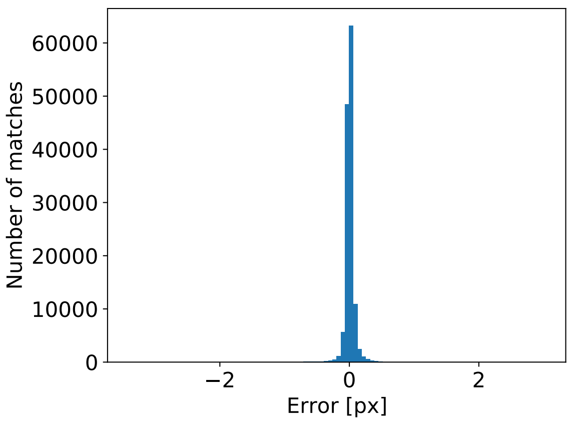
(g) BRISK, errors in $x$ direction.



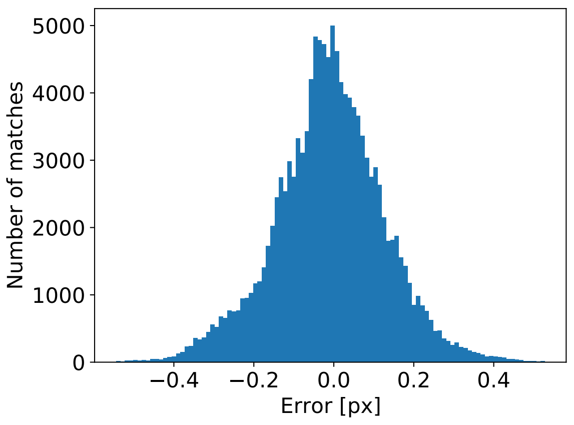
(h) BRISK, errors in $y$ direction.



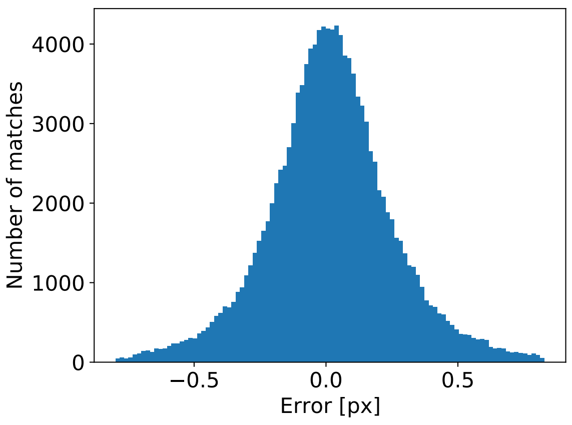
(i) KLT, errors in $x$ direction.



(j) KLT, errors in $y$ direction.

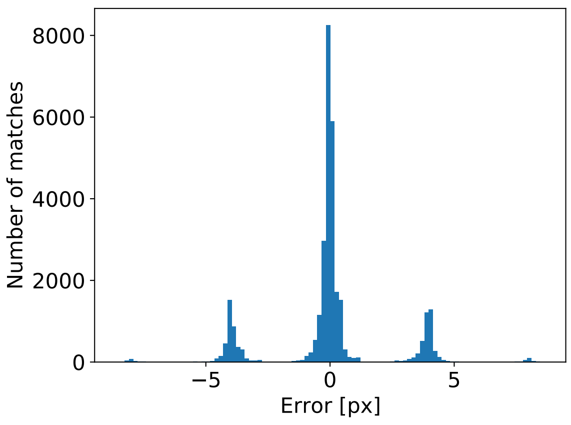


(k) STAR + BRIEF, errors in $x$ direction.

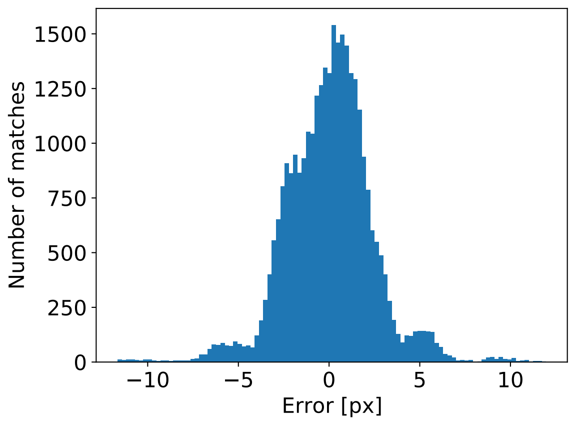


(l) STAR + BRIEF, errors in $y$ direction.

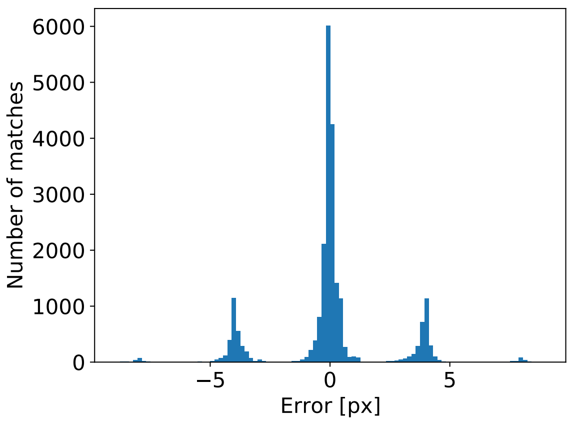
Figure C.0



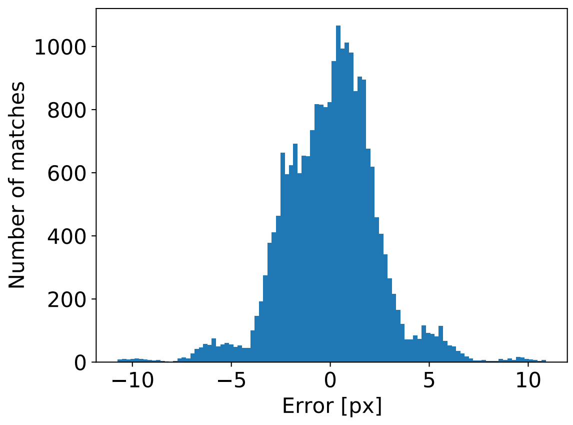
(m) SPV6, errors in $x$ direction.



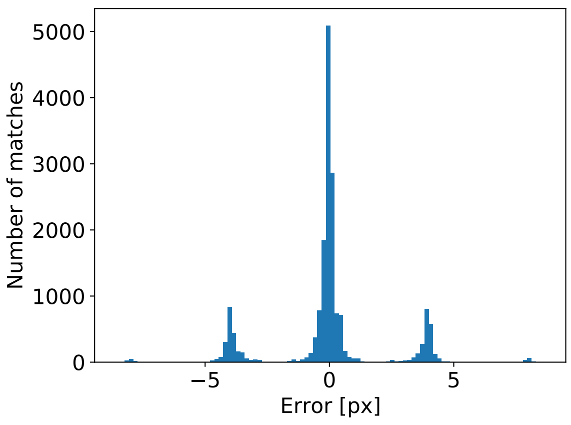
(n) SPV6, errors in $y$ direction.



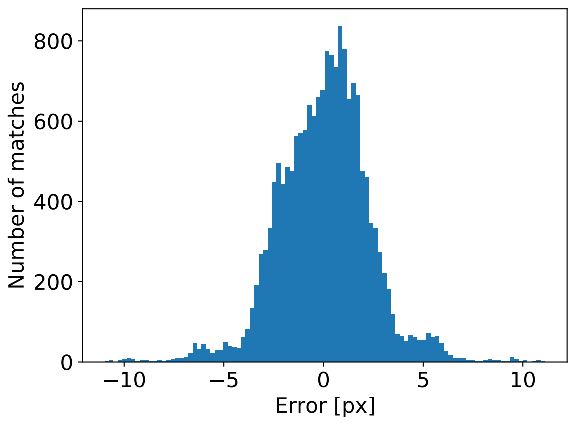
(o) SuperPoint Base, errors in $x$ direction.



(p) SuperPoint Base, errors in $y$ direction.

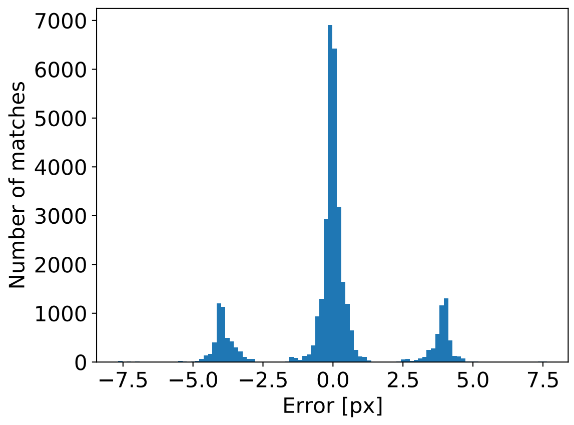


(q) SuperPoint Eros, errors in $x$ direction.

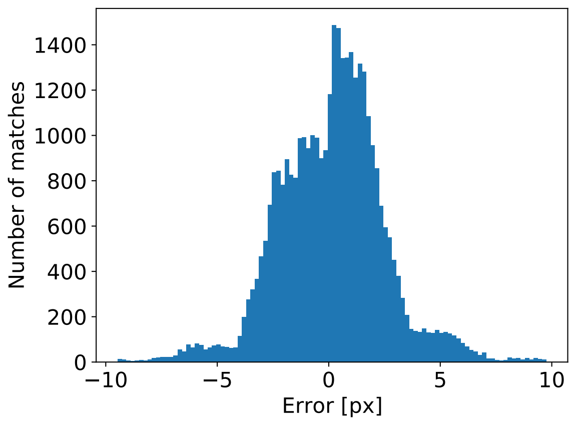


(r) SuperPoint Eros, errors in $y$ direction.

Figure C.0



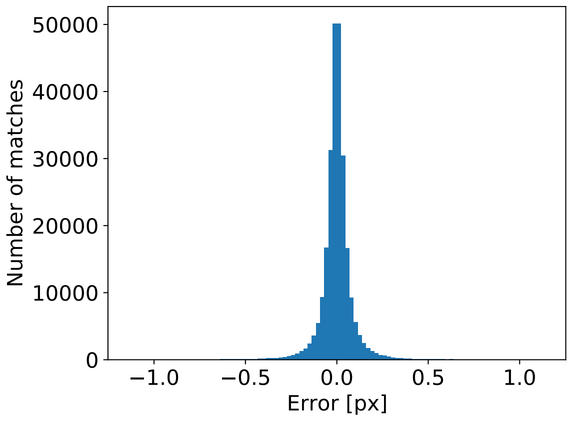
(s) SuperPoint Bennu, errors in $x$ direction.

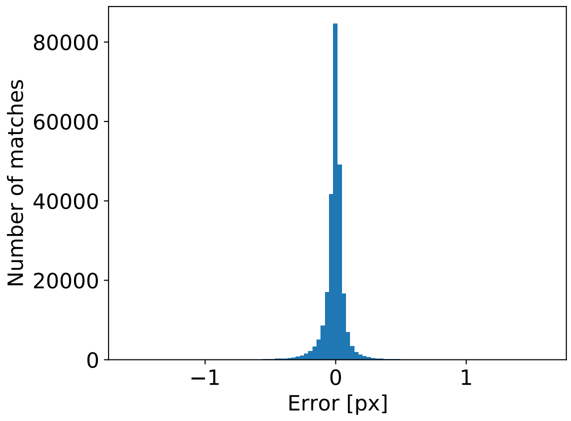


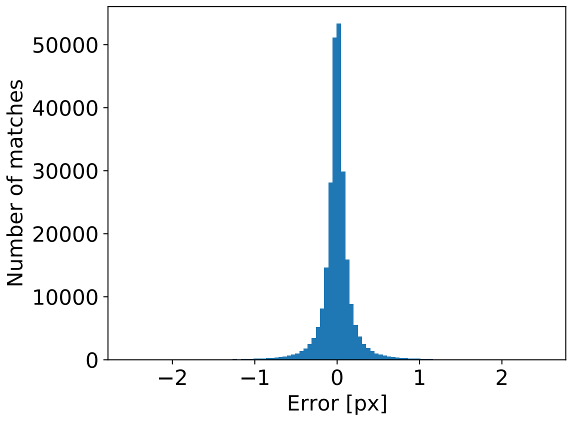
(t) SuperPoint Bennu, errors in $y$ direction.

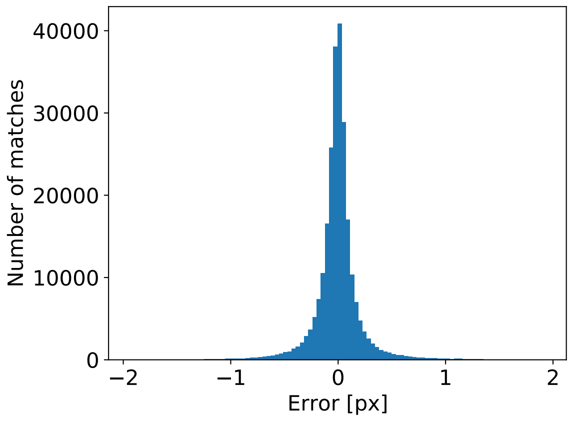
Figure C.0: Error distribution in $x$ and $y$ direction for the asteroid Bennu. Ideally, the distributions should be Gaussian.

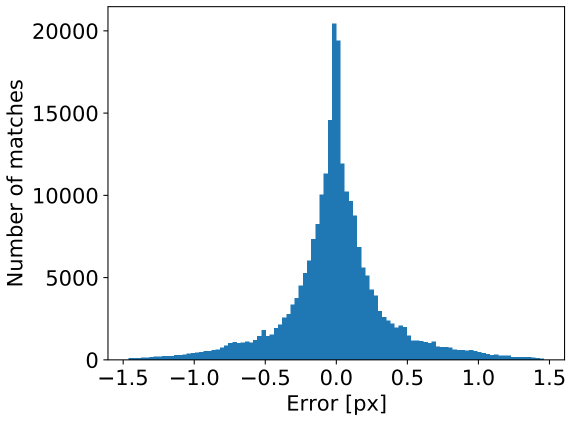
### C.0.3. Eros

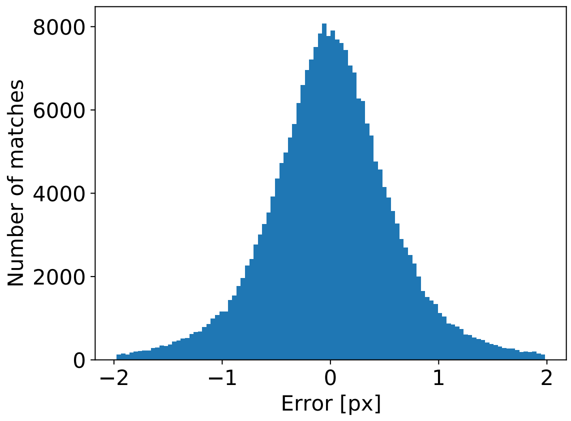

(a) SIFT, errors in $x$ direction.

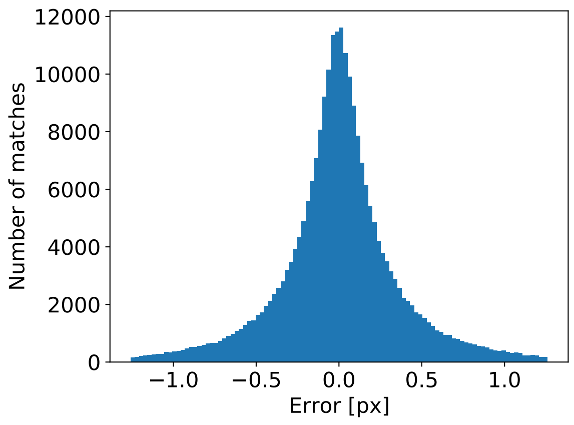

(b) SIFT, errors in $y$ direction.


(c) SURF, errors in $x$ direction.

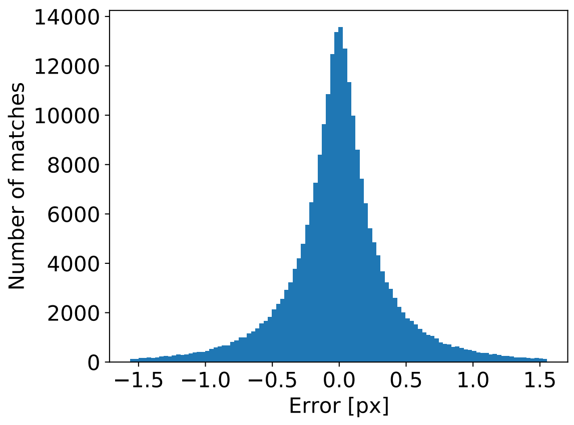

(d) SURF, errors in $y$ direction.


(e) ORB, errors in $x$ direction.

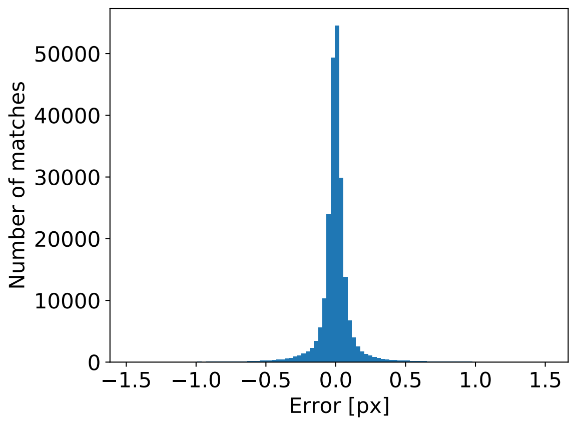

(f) ORB, errors in $y$ direction.



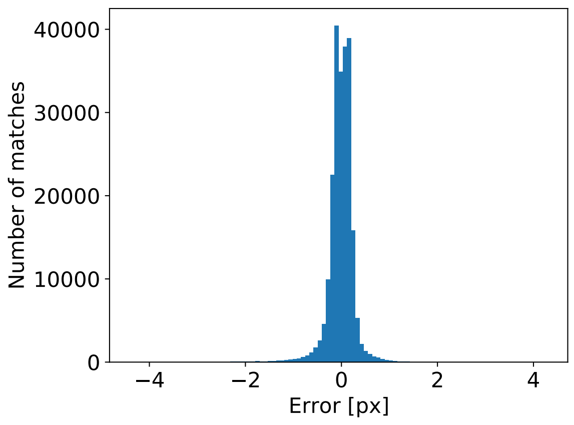
(g) BRISK, errors in $x$ direction.



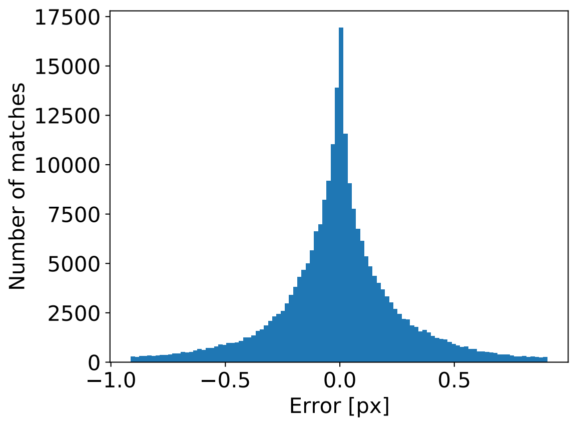
(h) BRISK, errors in $y$ direction.



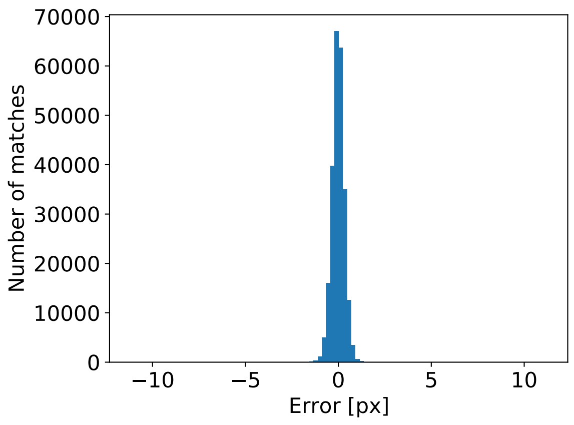
(i) KLT, errors in $x$ direction.



(j) KLT, errors in $y$ direction.

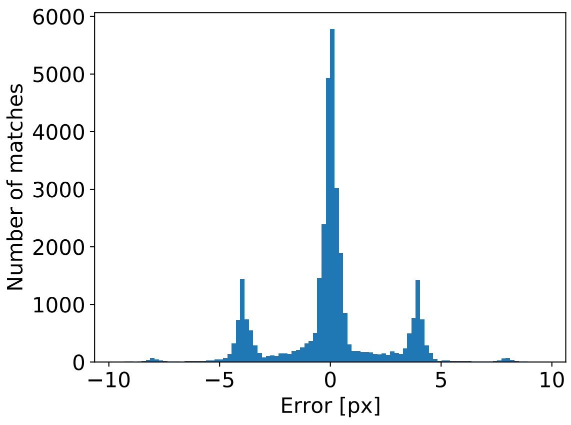


(k) STAR + BRIEF, errors in $x$ direction.

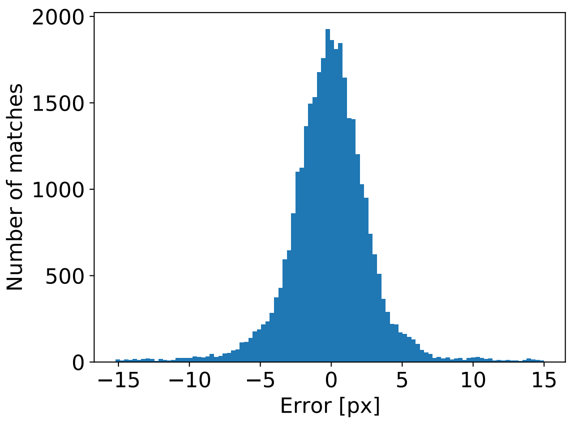


(l) STAR + BRIEF, errors in $y$ direction.

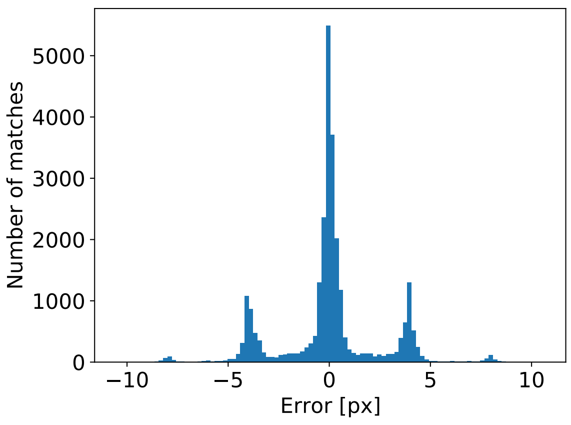
Figure C.0



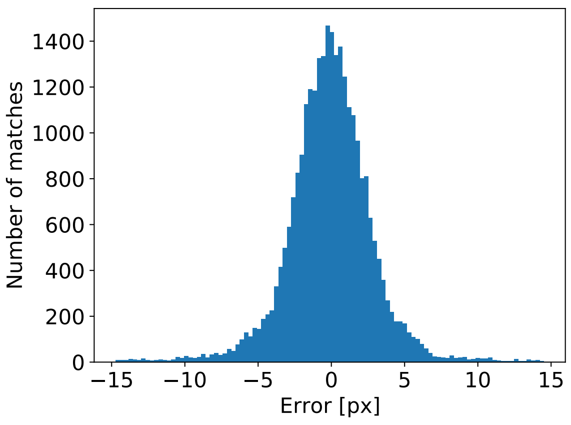
(m) SPV6, errors in $x$ direction.



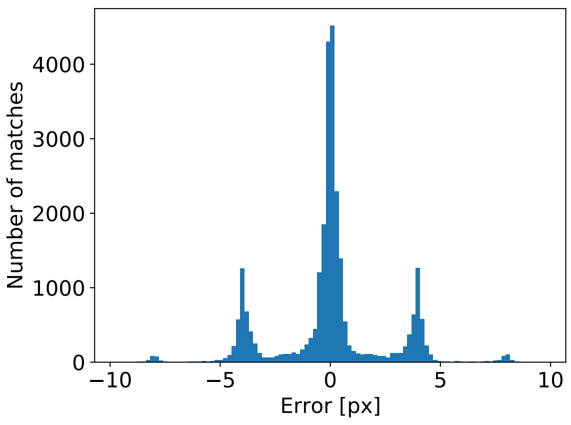
(n) SPV6, errors in $y$ direction.



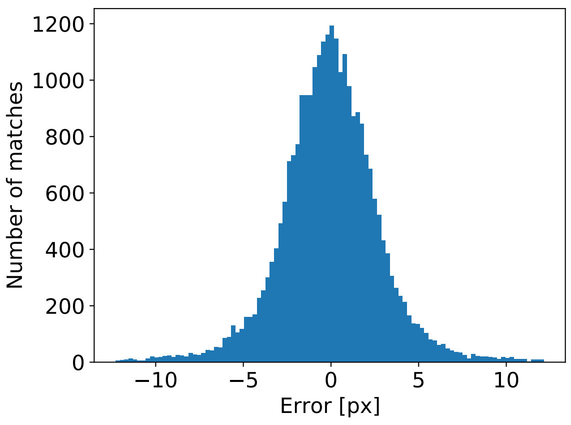
(o) SuperPoint Base, errors in $x$ direction.



(p) SuperPoint Base, errors in $y$ direction.

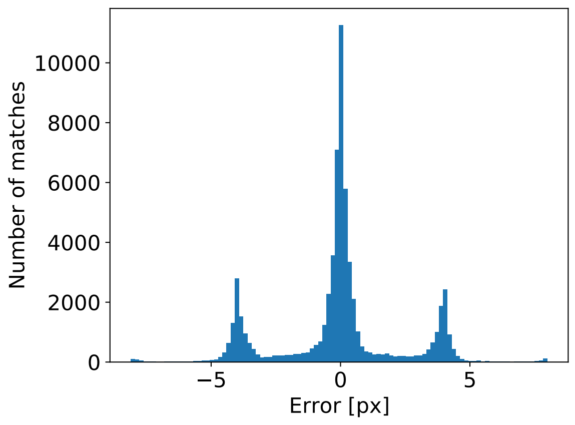


(q) SuperPoint Eros, errors in $x$ direction.

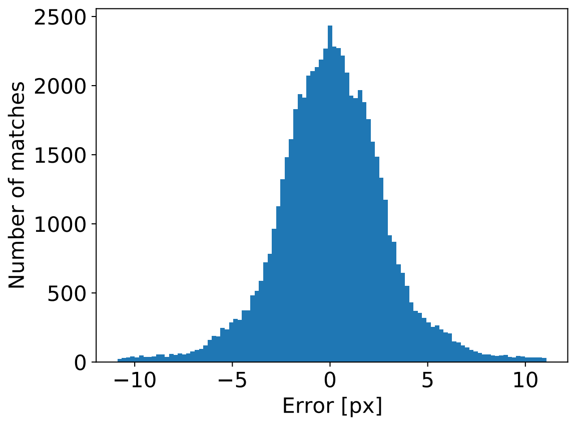


(r) SuperPoint Eros, errors in $y$ direction.

Figure C.0



(s) SuperPoint Bennu, errors in $x$ direction.



(t) SuperPoint Bennu, errors in $y$ direction.

Figure C.0: Error distribution in $x$ and $y$ direction for the asteroid Bennu. Ideally, the distributions should be Gaussian.

# D | Appendix D - Points within asteroid contour

Here are reported the tables with the percentages of points located within the contour of the asteroids.

Table D.1: Percentage of feature points located within the contour of the asteroid Bennu.

(a) Percentage of feature points located within the asteroid Bennu, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 100 | 100 | 100 | 100 | 100 | 100 |
| SURF | 99.6989 | 99.6732 | 99.6318 | 99.7178 | 99.7659 | 99.8251 |
| ORB | 99.5314 | 99.8089 | 99.8882 | 99.5971 | 99.8496 | 99.9215 |
| BRISK | 99.8549 | 99.9494 | 99.97 | 99.8581 | 99.9515 | 99.9758 |
| KLT | 94.8785 | 92.0944 | 94.2793 | 94.9174 | 90.7536 | 91.4766 |
| STAR + BRIEF | 99.9567 | 99.963 | 99.9728 | 99.964 | 99.9692 | 99.9831 |
| SPV6 | 66.5309 | 65.875 | 63.9176 | 66.3147 | 66.0485 | 64.279 |
| SuperPoint Base | 79.8384 | 80.9151 | 81.4316 | 80.1717 | 81.7168 | 82.4844 |
| SuperPoint Eros | 76.2012 | 76.37 | 76.4232 | 76.3823 | 77.0379 | 77.3436 |
| SuperPoint Bennu | 79.2447 | 83.3872 | 84.9007 | 79.3207 | 84.3099 | 86.195 |

(b) Percentage of feature points located within the asteroid Bennu, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.9387 | 99.9437 | 99.9451 | 99.9402 | 99.9481 | 99.9513 |
| SURF | 97.3978 | 97.733 | 97.9832 | 97.5395 | 98.1867 | 98.633 |
| ORB | 97.824 | 98.7902 | 99.2085 | 97.9195 | 98.9313 | 99.3678 |
| BRISK | 99.3203 | 99.6841 | 99.8449 | 99.3265 | 99.6925 | 99.8535 |
| KLT | 93.5906 | 83.5251 | 81.8353 | 93.5957 | 83.4625 | 81.1078 |
| STAR + BRIEF | 97.7899 | 98.1833 | 98.5621 | 97.8471 | 98.4392 | 98.9384 |
| SPV6 | 49.1267 | 46.7171 | 44.4567 | 47.2443 | 43.4564 | 40.5215 |
| SuperPoint Base | 64.4889 | 62.771 | 61.9993 | 64.4121 | 62.2869 | 61.532 |
| SuperPoint Eros | 57.4166 | 54.0643 | 52.0778 | 56.7722 | 53.2051 | 50.817 |
| SuperPoint Bennu | 59.6154 | 56.4143 | 56.7044 | 58.4247 | 54.8155 | 55.3712 |

## Bennu

Table D.1

(a) Percentage of feature points located within the asteroid Bennu, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.9964 | 99.9959 | 99.9965 | 99.9964 | 99.9959 | 99.9965 |
| SURF | 99.5822 | 99.5493 | 99.464 | 99.6013 | 99.71 | 99.7545 |
| ORB | 98.6253 | 99.4597 | 99.6628 | 98.8154 | 99.5709 | 99.7345 |
| BRISK | 99.4205 | 99.7883 | 99.9178 | 99.4417 | 99.8088 | 99.9378 |
| KLT | 94.5946 | 92.1653 | 93.471 | 94.6341 | 90.6124 | 89.946 |
| STAR + BRIEF | 99.9711 | 99.9718 | 99.9795 | 99.9725 | 99.9767 | 99.9832 |
| SPV6 | 67.8218 | 67.0642 | 64.6797 | 67.7849 | 67.4075 | 64.2941 |
| SuperPoint Base | 80.2182 | 81.1839 | 81.5996 | 80.6549 | 82.2133 | 82.9043 |
| SuperPoint Eros | 78.3643 | 78.8588 | 79.1472 | 78.4232 | 79.6184 | 80.0823 |
| SuperPoint Bennu | 81.1177 | 84.8344 | 86.0128 | 81.4274 | 85.638 | 87.2278 |

(b) Percentage of feature points located within the asteroid Bennu, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.9789 | 99.9831 | 99.9757 | 99.9788 | 99.9839 | 99.9826 |
| SURF | 96.3783 | 96.9696 | 97.1901 | 96.5428 | 97.4587 | 98.0096 |
| ORB | 96.0899 | 97.8516 | 98.7419 | 96.2203 | 98.0748 | 99.0179 |
| BRISK | 98.284 | 99.173 | 99.6091 | 98.3073 | 99.2091 | 99.6441 |
| KLT | 94.1359 | 86.3787 | 83.6869 | 94.1427 | 86.3236 | 83.1072 |
| STAR + BRIEF | 97.0375 | 97.4034 | 98.0067 | 97.1006 | 97.7144 | 98.445 |
| SPV6 | 49.8125 | 46.6215 | 43.341 | 48.1285 | 43.5971 | 39.2115 |
| SuperPoint Base | 65.5974 | 63.2765 | 61.3782 | 65.3346 | 63.1039 | 60.8891 |
| SuperPoint Eros | 62.3973 | 59.1738 | 56.5892 | 61.6489 | 58.1551 | 55.9291 |
| SuperPoint Bennu | 63.469 | 60.932 | 61.024 | 62.6245 | 59.9308 | 60.2409 |

## Itokawa

Table D.2: Percentage of feature points located within the contour of the asteroid Itokawa.

(a) Percentage of feature points located within the asteroid Itokawa, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 97.8628 | 98.0755 | 98.2191 | 97.9365 | 98.2819 | 98.5782 |
| SURF | 97.8533 | 98.169 | 98.3322 | 97.9221 | 98.3465 | 98.5916 |
| ORB | 92.6369 | 95.2628 | 96.849 | 92.6388 | 95.3621 | 97.1023 |
| BRISK | 96.3969 | 97.7279 | 98.5657 | 96.5349 | 97.952 | 98.7939 |
| KLT | 93.5197 | 90.3537 | 88.3231 | 93.6962 | 90.6125 | 88.0338 |
| STAR + BRIEF | 99.7157 | 99.8242 | 99.8178 | 99.7288 | 99.852 | 99.8557 |
| SPV6 | 80.4778 | 80.8127 | 80.7672 | 80.8645 | 81.8224 | 82.2008 |
| SuperPoint Base | 85.3258 | 85.6317 | 85.7808 | 85.5451 | 85.9856 | 86.3039 |
| SuperPoint Eros | 81.5824 | 81.6601 | 81.6906 | 81.6813 | 81.9113 | 81.9892 |
| SuperPoint Bennu | 84.4869 | 85.7322 | 86.4652 | 84.6362 | 86.1797 | 87.1415 |

(b) Percentage of feature points located within the asteroid Itokawa, divided by image processing frequency, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 96.8443 | 97.1181 | 97.1196 | 96.9333 | 97.2673 | 97.4338 |
| SURF | 92.9359 | 93.599 | 94.1251 | 93.053 | 93.9638 | 94.4783 |
| ORB | 95.4896 | 96.3504 | 96.8484 | 95.5651 | 96.6322 | 97.1075 |
| BRISK | 97.2749 | 97.532 | 97.9626 | 97.3504 | 97.8621 | 98.1687 |
| KLT | 97.1604 | 95.1773 | 91.0347 | 97.2135 | 95.2439 | 92.9082 |
| STAR + BRIEF | 94.391 | 96.3281 | 97.2457 | 94.4065 | 96.5762 | 97.627 |
| SPV6 | 63.5351 | 63.7898 | 63.1529 | 63.7016 | 63.4767 | 63.0706 |
| SuperPoint Base | 72.4801 | 73.1527 | 72.785 | 72.6471 | 73.4222 | 73.1037 |
| SuperPoint Eros | 65.5465 | 65.5109 | 64.7957 | 65.4709 | 65.3997 | 64.6033 |
| SuperPoint Bennu | 64.951 | 64.881 | 64.4224 | 64.7745 | 64.7238 | 64.1899 |

Table D.2

(a) Percentage of feature points located within the asteroid Itokawa, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 96.924 | 97.1698 | 97.3429 | 97.0991 | 97.6082 | 98.0255 |
| SURF | 96.0995 | 96.5507 | 96.8453 | 96.2254 | 96.846 | 97.2986 |
| ORB | 92.3573 | 94.7082 | 96.231 | 92.4634 | 94.8772 | 96.5804 |
| BRISK | 95.1977 | 97.0757 | 98.2484 | 95.3512 | 97.3572 | 98.4785 |
| KLT | 92.5849 | 89.4513 | 86.617 | 92.8206 | 89.8025 | 86.6492 |
| STAR + BRIEF | 99.0354 | 99.287 | 99.2786 | 99.0812 | 99.4004 | 99.4254 |
| SPV6 | 83.2612 | 83.5936 | 83.3245 | 83.5769 | 84.4513 | 84.7857 |
| SuperPoint Base | 86.3473 | 86.8989 | 87.13 | 86.6351 | 87.2723 | 87.7163 |
| SuperPoint Eros | 84.7047 | 84.7835 | 84.9575 | 84.755 | 85.0152 | 85.2604 |
| SuperPoint Bennu | 85.9347 | 87.0723 | 87.434 | 86.1042 | 87.5374 | 88.0814 |

(b) Percentage of feature points located within the asteroid Itokawa, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 93.1012 | 92.9945 | 93.3874 | 93.1978 | 93.3281 | 94.0067 |
| SURF | 87.9923 | 87.6858 | 88.5132 | 88.1519 | 88.0192 | 88.9415 |
| ORB | 91.4752 | 93.0785 | 94.2443 | 91.4987 | 93.0867 | 94.3192 |
| BRISK | 94.0403 | 95.0313 | 96.1363 | 94.1055 | 95.2286 | 96.3969 |
| KLT | 94.9207 | 92.3303 | 90.8563 | 94.9351 | 93.2754 | 90.9788 |
| STAR + BRIEF | 93.3667 | 95.9826 | 96.3256 | 93.3823 | 96.0188 | 96.3944 |
| SPV6 | 63.6039 | 63.4982 | 63.0145 | 63.2444 | 63.1842 | 62.4615 |
| SuperPoint Base | 69.7385 | 70.8034 | 70.284 | 69.8147 | 71.0043 | 70.5273 |
| SuperPoint Eros | 67.5944 | 67.4519 | 66.7125 | 67.5049 | 67.3895 | 66.5547 |
| SuperPoint Bennu | 64.2325 | 64.0754 | 63.9375 | 64.144 | 64.1032 | 63.9449 |

## Eros

Table D.3: Percentage of feature points located within the contour of the asteroid Eros.

(a) Percentage of feature points located within the asteroid Eros, divided by image processing frequency, for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of 50.0 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.7919 | 99.8312 | 99.8487 | 99.8293 | 99.8898 | 99.9101 |
| SURF | 94.1058 | 99.4245 | 99.6816 | 94.1212 | 99.5135 | 99.7175 |
| ORB | 99.9289 | 99.9611 | 99.9247 | 99.9414 | 99.9848 | 99.9838 |
| BRISK | 98.5086 | 98.655 | 99.9642 | 98.5127 | 98.6494 | 99.9727 |
| KLT | 96.2009 | 93.5447 | 91.7557 | 96.3235 | 93.8544 | 91.4245 |
| STAR + BRIEF | 98.3654 | 98.2524 | 98.2268 | 98.4079 | 98.3395 | 98.1984 |
| SPV6 | 74.0371 | 73.0501 | 71.7155 | 74.7144 | 74.5048 | 73.4391 |
| SuperPoint Base | 81.7041 | 82.0514 | 82.1653 | 82.398 | 82.8994 | 83.5351 |
| SuperPoint Eros | 76.9396 | 77.1762 | 77.1615 | 77.571 | 78.2999 | 78.5495 |
| SuperPoint Bennu | 77.7746 | 79.6423 | 80.8318 | 78.0882 | 80.3332 | 81.6812 |

(b) Percentage of feature points located within the asteroid Eros, divided by image processing frequency, for a light intensity of $1.0 \frac{W}{m^2}$ and a radius of 70.0 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.6967 | 99.7371 | 99.7278 | 99.7234 | 99.7946 | 99.8435 |
| SURF | 97.5645 | 98.5273 | 98.8839 | 97.7011 | 98.6875 | 99.1245 |
| ORB | 99.934 | 99.9631 | 99.9567 | 99.9393 | 99.9671 | 99.9713 |
| BRISK | 99.9008 | 99.9338 | 99.9596 | 99.918 | 99.947 | 99.9671 |
| KLT | 96.0253 | 92.0438 | 86.3461 | 96.0476 | 92.2564 | 85.8046 |
| STAR + BRIEF | 97.8324 | 97.8103 | 97.5684 | 97.9199 | 97.9785 | 97.6182 |
| SPV6 | 60.6907 | 61.0478 | 60.6105 | 60.8556 | 61.1531 | 60.4215 |
| SuperPoint Base | 71.0393 | 71.4212 | 72.0742 | 71.7047 | 72.4079 | 73.6584 |
| SuperPoint Eros | 65.1623 | 65.9249 | 66.3331 | 65.6474 | 66.8409 | 67.4669 |
| SuperPoint Bennu | 62.432 | 65.2251 | 67.7403 | 62.2117 | 65.4655 | 68.3622 |

Table D.3

(a) Percentage of feature points located within the asteroid Eros, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 50.0 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.4263 | 99.4794 | 99.5561 | 99.5502 | 99.6991 | 99.7939 |
| SURF | 97.8825 | 99.1322 | 99.3771 | 98.0391 | 99.2401 | 99.4736 |
| ORB | 99.8548 | 99.8545 | 99.8802 | 99.8794 | 99.8639 | 99.8989 |
| BRISK | 98.5363 | 99.8779 | 99.9201 | 98.5582 | 99.9099 | 99.942 |
| KLT | 96.5679 | 92.3065 | 90.8742 | 96.7398 | 91.7075 | 90.2114 |
| STAR + BRIEF | 98.4121 | 98.3278 | 98.2791 | 98.4363 | 98.2962 | 98.3772 |
| SPV6 | 74.4732 | 73.0275 | 71.1943 | 75.0015 | 74.1236 | 72.533 |
| SuperPoint Base | 80.0144 | 80.4272 | 80.0688 | 80.8208 | 81.6939 | 81.5283 |
| SuperPoint Eros | 78.6741 | 78.6537 | 78.406 | 79.3366 | 79.5587 | 79.7974 |
| SuperPoint Bennu | 79.4505 | 80.9994 | 81.5269 | 79.9097 | 82.0893 | 82.9823 |

(b) Percentage of feature points located within the asteroid Eros, divided by image processing frequency, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 70.0 $Km$.

| | Percentage of valid keypoints [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 99.3003 | 99.3525 | 99.399 | 99.3642 | 99.5044 | 99.6127 |
| SURF | 97.2117 | 97.9819 | 98.3732 | 97.3929 | 98.1836 | 98.6747 |
| ORB | 99.8325 | 99.8991 | 99.9011 | 99.8486 | 99.9169 | 99.9139 |
| BRISK | 99.7485 | 99.8279 | 99.887 | 99.7844 | 99.8533 | 99.917 |
| KLT | 96.4525 | 91.8039 | 86.168 | 96.5024 | 91.7884 | 85.807 |
| STAR + BRIEF | 97.8558 | 97.8075 | 97.5214 | 97.9417 | 97.917 | 97.715 |
| SPV6 | 60.4259 | 60.3992 | 59.2169 | 60.4452 | 60.2997 | 59.0058 |
| SuperPoint Base | 68.52 | 69.2157 | 69.2317 | 69.3836 | 70.3075 | 70.7156 |
| SuperPoint Eros | 67.642 | 68.0951 | 68.5062 | 68.0653 | 68.6801 | 69.2754 |
| SuperPoint Bennu | 64.331 | 66.6285 | 68.8143 | 64.5036 | 67.0432 | 69.9107 |

# E | Appendix E - Outliers

## E.1. Percentage of outliers

Here are reported the tables with the percentages of points that become outliers.

### E.1.1. Bennu

Table E.1: Percentage of points that, at some point, become outliers, for the asteroid Bennu.

(a) Percentage of points that, at some point, become outliers, for the asteroid Bennu, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.065425 | 0.218469 | 0.497454 | 0 | 0.0168209 | 0.0237614 |
| SURF | 0.533977 | 2.00953 | 2.36415 | 0.18356 | 0.36674 | 0.40608 |
| ORB | 1.02308 | 1.06727 | 1.00174 | 0.109111 | 0.102284 | 0.234082 |
| BRISK | 0.0672495 | 0.175824 | 0.113293 | 0.0177651 | 0.0369113 | 0.0127291 |
| KLT | 59.0937 | 91.3311 | 92.582 | 33.4775 | 57.8778 | 77.5448 |
| STAR + BRIEF | 0.0810373 | 0.576132 | 1.85497 | 0.0814996 | 0.082713 | 0.428449 |
| SPV6 | 39.6875 | 43.0508 | 40.7801 | 27.3356 | 33.5907 | 31.0204 |
| SuperPoint Base | 50.4823 | 50.6536 | 50.5703 | 41.3559 | 32.0158 | 42.562 |
| SuperPoint Eros | 47.8827 | 48.3444 | 46.0993 | 36.1204 | 43.299 | 35.5102 |
| SuperPoint Bennu | 36.1371 | 42.623 | 38.4354 | 29.5203 | 32.9787 | 30.9963 |

(b) Percentage of points that, at some point, become outliers, for the asteroid Bennu, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 1.56582 | 1.98831 | 2.17735 | 1.40833 | 1.22639 | 0.923498 |
| SURF | 1.4765 | 2.97892 | 3.7129 | 0.618622 | 0.906217 | 0.989144 |
| ORB | 1.2139 | 1.38889 | 1.74448 | 0.393529 | 0.447227 | 0.443459 |
| BRISK | 0.0719709 | 0.18582 | 0.347264 | 0.0468637 | 0.0792393 | 0.133463 |
| KLT | 30.1314 | 74.7288 | 86.3887 | 24.0065 | 36.9777 | 37.5455 |
| STAR + BRIEF | 0.519931 | 1.20846 | 4.33018 | 0.172117 | 0.445765 | 1.81311 |
| SPV6 | 37.9147 | 32.1244 | 32.5581 | 29.6512 | 24.2105 | 23.2877 |
| SuperPoint Base | 41.6667 | 39.2638 | 42.3841 | 39.4444 | 41.1392 | 32.3308 |
| SuperPoint Eros | 45.4082 | 44.8454 | 40.3509 | 30.8989 | 32.1839 | 32.1429 |
| SuperPoint Bennu | 30.8571 | 27.4194 | 23.3333 | 26.3473 | 17.6471 | 23.2877 |

Table E.1

(a) Percentage of points that, at some point, become outliers, for the asteroid Bennu, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.188147 | 0.617284 | 1.19626 | 0.00448612 | 0.0413189 | 0.0478354 |
| SURF | 0.464897 | 2.16249 | 2.67094 | 0.258756 | 0.518672 | 0.554939 |
| ORB | 0.914867 | 1.37261 | 1.54827 | 0.275786 | 0.143421 | 0.584795 |
| BRISK | 0.141063 | 0.216752 | 0.310252 | 0.037176 | 0.062898 | 0.0430849 |
| KLT | 62.6222 | 90.3978 | 91.5858 | 37.168 | 55.8444 | 74.8123 |
| STAR + BRIEF | 0.082713 | 0.699913 | 1.49813 | 0.0838926 | 0.352734 | 0.647549 |
| SPV6 | 41.8006 | 42.1769 | 36.8821 | 30.1325 | 31.0714 | 30.4721 |
| SuperPoint Base | 44.8071 | 45.614 | 43.1734 | 33.218 | 33.9768 | 34.1564 |
| SuperPoint Eros | 48.7395 | 48.5623 | 48.0427 | 35.7143 | 40.0722 | 36.8217 |
| SuperPoint Bennu | 37.3041 | 42.4051 | 47.0149 | 29.5139 | 31.9703 | 39.3443 |

(b) Percentage of points that, at some point, become outliers, for the asteroid Bennu, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 1.52849 | 1.90316 | 2.50508 | 1.37186 | 1.16392 | 0.849013 |
| SURF | 1.55218 | 3.29884 | 4.24087 | 0.71178 | 0.930295 | 0.992467 |
| ORB | 0.349498 | 1.69861 | 2.26316 | 0 | 0.698649 | 0.518135 |
| BRISK | 0.117874 | 0.306321 | 0.322465 | 0.0715538 | 0.0632067 | 0.0364609 |
| KLT | 39.4053 | 76.1813 | 85.742 | 29.1379 | 36.6654 | 36.4214 |
| STAR + BRIEF | 0.167504 | 1.12994 | 4.41558 | 0.169492 | 0.710227 | 2.43572 |
| SPV6 | 37.395 | 36.612 | 33.1492 | 31.1765 | 27.9221 | 23.6025 |
| SuperPoint Base | 44.5545 | 36.5854 | 38.9535 | 38.3333 | 27.2152 | 30.3704 |
| SuperPoint Eros | 43.5897 | 45.4545 | 37.037 | 38.3234 | 33.5227 | 28 |
| SuperPoint Bennu | 39.8773 | 36.9942 | 32.1637 | 30.7692 | 30.2632 | 21.7391 |

### E.1.2. Itokawa

Table E.2: Percentage of points that, at some point, become outliers, for the asteroid Itokawa.

(a) Percentage of points that, at some point, become outliers, for the asteroid Itokawa, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.739933 | 1.63894 | 2.96241 | 0.472675 | 0.854271 | 1.19776 |
| SURF | 0.428571 | 1.03809 | 2.23499 | 0.335959 | 0.582124 | 1.07282 |
| ORB | 0.801673 | 1.98675 | 2.77548 | 0.397399 | 0.705053 | 0.979955 |
| BRISK | 1.05199 | 2.36194 | 2.83702 | 0.399243 | 0.754665 | 0.943396 |
| KLT | 44.9637 | 70.4827 | 80.2792 | 38.5749 | 53.1414 | 55.717 |
| STAR + BRIEF | 0.251256 | 0.856531 | 2.14133 | 0 | 0.441014 | 0.896861 |
| SPV6 | 45.8194 | 49.3711 | 50.5226 | 36.4662 | 35.5469 | 38.8316 |
| SuperPoint Base | 58.6331 | 53.405 | 56.9288 | 45.1128 | 43.9331 | 51.0204 |
| SuperPoint Eros | 53.0405 | 51.5901 | 53.2374 | 46.2745 | 46.8 | 40.4255 |
| SuperPoint Bennu | 51.6717 | 42.1986 | 49.434 | 34.4086 | 33.0544 | 37.6106 |

(b) Percentage of points that, at some point, become outliers, for the asteroid Itokawa, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.493275 | 1.08271 | 1.79503 | 0.389766 | 0.623411 | 0.778877 |
| SURF | 0.844325 | 2.08491 | 3.15435 | 0.564774 | 1.08679 | 1.50381 |
| ORB | 0.572519 | 1.221 | 1.05556 | 0.25641 | 0.189036 | 0.297619 |
| BRISK | 0.352384 | 0.475207 | 0.7876 | 0.213817 | 0.153546 | 0.231705 |
| KLT | 40.2383 | 55.2395 | 66.2464 | 20.2437 | 31.934 | 40.8506 |
| STAR + BRIEF | 0.904977 | 1.16732 | 1.24417 | 0.453515 | 0.19084 | 0.165837 |
| SPV6 | 33.908 | 45.5026 | 44.9102 | 36.2573 | 38.3117 | 35.2941 |
| SuperPoint Base | 54.3011 | 53.5484 | 57.6159 | 51.049 | 47.4359 | 47.4453 |
| SuperPoint Eros | 59.1195 | 54.1096 | 54.7945 | 54.6763 | 47.1831 | 49.2857 |
| SuperPoint Bennu | 50.2674 | 45.3552 | 37.5723 | 40.1099 | 35.7576 | 32.7485 |

Table E.2

(a) Percentage of points that, at some point, become outliers, for the asteroid Itokawa, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 1.24956 | 3.23799 | 4.77053 | 0.722492 | 1.16452 | 1.54274 |
| SURF | 1.01807 | 2.60054 | 3.82603 | 0.771493 | 1.3936 | 1.86817 |
| ORB | 1.13355 | 3.05628 | 4.18719 | 0.74184 | 1.46072 | 1.87082 |
| BRISK | 1.58181 | 2.8932 | 3.58603 | 0.697865 | 1.3368 | 1.6066 |
| KLT | 49.976 | 70.9318 | 79.5291 | 43.4744 | 52.3609 | 53.6558 |
| STAR + BRIEF | 0.123305 | 0.931677 | 2.72177 | 0.123916 | 0.527426 | 1.44778 |
| SPV6 | 42.029 | 43.1095 | 46.6165 | 37.5479 | 36.0153 | 32.8244 |
| SuperPoint Base | 50.519 | 44.8148 | 54.3165 | 40.9962 | 32.6848 | 43.0328 |
| SuperPoint Eros | 53.2872 | 53.7671 | 50.9881 | 42.3077 | 45.6274 | 46.7391 |
| SuperPoint Bennu | 34.375 | 43.4457 | 43.1734 | 33.8346 | 38.4 | 36.7769 |

(b) Percentage of points that, at some point, become outliers, for the asteroid Itokawa, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.684452 | 1.95344 | 3.28738 | 0.464481 | 0.818808 | 0.864904 |
| SURF | 1.01642 | 2.5869 | 3.92615 | 0.75318 | 1.24555 | 1.49935 |
| ORB | 0.500939 | 1.36731 | 2.0552 | 0.190597 | 0.323625 | 0.628931 |
| BRISK | 0.262341 | 0.726141 | 1.24838 | 0.140876 | 0.377728 | 0.490798 |
| KLT | 26.4909 | 52.1939 | 66.0014 | 23.6617 | 36.3851 | 39.733 |
| STAR + BRIEF | 0.666667 | 1.13636 | 3.29068 | 0 | 0.957854 | 0.536673 |
| SPV6 | 39.3939 | 47.093 | 36.1446 | 34.5912 | 33.526 | 27.2109 |
| SuperPoint Base | 50.2857 | 52.1053 | 46.7033 | 41.6107 | 43.75 | 30.5556 |
| SuperPoint Eros | 52.6316 | 52.1472 | 47.3054 | 51.8919 | 40 | 32.8571 |
| SuperPoint Bennu | 42.3729 | 40.6977 | 39.6341 | 39.2405 | 27.3973 | 34.4595 |

## E.1.3. Eros

Table E.3: Percentage of points that, at some point, become outliers, for the asteroid Eros.

(a) Percentage of points that, at some point, become outliers, for the asteroid Eros, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 50 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.71736 | 1.39421 | 2.06531 | 0.586943 | 0.865199 | 0.85842 |
| SURF | 0.823674 | 2.14301 | 3.16951 | 0.452032 | 0.848122 | 1.19723 |
| ORB | 0.779423 | 1.73496 | 2.06186 | 0.369914 | 0.626714 | 0.475367 |
| BRISK | 1.42214 | 2.5091 | 3.27085 | 0.421985 | 0.713814 | 0.888979 |
| KLT | 68.0093 | 87.1737 | 91.2082 | 46.2475 | 59.7564 | 68.1773 |
| STAR + BRIEF | 0.795756 | 1.33185 | 2.37812 | 0.543478 | 1.00671 | 1.59705 |
| SPV6 | 43.6123 | 52.7132 | 47.7876 | 37.5587 | 38.7755 | 43.1373 |
| SuperPoint Base | 47.8448 | 50 | 52 | 40.1961 | 49.0385 | 40.4762 |
| SuperPoint Eros | 53.4188 | 49.1736 | 48.7179 | 40.7407 | 40.9524 | 35.4839 |
| SuperPoint Bennu | 37.4429 | 41.3462 | 42.723 | 32.0442 | 33.4975 | 31.1005 |

(b) Percentage of points that, at some point, become outliers, for the asteroid Eros, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 70 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 0.772446 | 1.3574 | 2.1284 | 0.699301 | 0.961762 | 1.17041 |
| SURF | 1.10978 | 2.29237 | 3.41657 | 0.67687 | 0.902278 | 1.24217 |
| ORB | 0.472813 | 1.55844 | 2.5987 | 0.123305 | 0.451977 | 0.60241 |
| BRISK | 0.967503 | 1.88261 | 2.3683 | 0.257555 | 0.433782 | 0.562852 |
| KLT | 54.4262 | 84.4816 | 89.4323 | 39.1012 | 54.249 | 63.2026 |
| STAR + BRIEF | 0.251889 | 0.963391 | 1.70455 | 0 | 0.380952 | 0.753296 |
| SPV6 | 39.9015 | 42.487 | 43.0939 | 36.2573 | 34.3195 | 29.4118 |
| SuperPoint Base | 47.3054 | 49.7143 | 49.3976 | 38.3117 | 45.3846 | 40.9396 |
| SuperPoint Eros | 43.9759 | 38.8535 | 45.8599 | 32.0261 | 36.0902 | 36.8 |
| SuperPoint Bennu | 27.8689 | 30.9677 | 31.0345 | 25.1613 | 24.4898 | 29.4872 |

Table E.3

(a) Percentage of points that, at some point, become outliers, for the asteroid Eros, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 1.44474 | 2.99771 | 4.84946 | 0.953883 | 1.38215 | 1.79548 |
| SURF | 1.92422 | 3.60109 | 4.63489 | 0.990508 | 1.47967 | 1.64911 |
| ORB | 1.15798 | 3.53331 | 4.09091 | 0.652174 | 1.63934 | 1.5343 |
| BRISK | 2.22581 | 3.54399 | 5.08475 | 0.764283 | 1.45104 | 1.86503 |
| KLT | 71.024 | 85.813 | 90.2459 | 45.9643 | 59.1406 | 68.7849 |
| STAR + BRIEF | 0.257069 | 1.76991 | 3.62637 | 0.262467 | 1.01925 | 1.96078 |
| SPV6 | 48.4581 | 50.4237 | 44.6281 | 37.8109 | 41.8848 | 39.0625 |
| SuperPoint Base | 46.6942 | 50.6224 | 49.5495 | 38.2979 | 37.6744 | 40.5941 |
| SuperPoint Eros | 51.8672 | 49.7436 | 45.1327 | 44.7964 | 38.5027 | 39.6907 |
| SuperPoint Bennu | 35.3535 | 37 | 38.2883 | 27.1739 | 31.4136 | 32.5 |

(b) Percentage of points that, at some point, become outliers, for the asteroid Eros, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Percentage of points that become outliers [%] | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 1.20616 | 2.3937 | 3.6672 | 0.971806 | 1.31589 | 1.54227 |
| SURF | 1.95932 | 3.65974 | 5.15464 | 1.04339 | 1.32643 | 1.61658 |
| ORB | 0.865801 | 2.47238 | 3.66922 | 0.645161 | 1.1534 | 1.25298 |
| BRISK | 1.30393 | 2.57461 | 2.64402 | 0.524378 | 0.924202 | 0.968992 |
| KLT | 61.1493 | 82.7449 | 88.1571 | 39.1645 | 50.548 | 60.4682 |
| STAR + BRIEF | 0.455581 | 1.66052 | 3.09811 | 0.229358 | 0.963391 | 0.881834 |
| SPV6 | 40.678 | 41.875 | 41.9355 | 31.0127 | 33.8235 | 40.3974 |
| SuperPoint Base | 49.3421 | 47.4026 | 49.3421 | 42.6573 | 40.4412 | 41.0256 |
| SuperPoint Eros | 55.1136 | 41.9118 | 48.4848 | 42.0382 | 36.9427 | 34.5324 |
| SuperPoint Bennu | 34.7826 | 24.5509 | 35.1852 | 30.6569 | 22.2222 | 32.5758 |

## E.2. Persistence of the features

Here are reported the tables with the average feature persistence.

### E.2.1. Bennu

Table E.4: Mean persistence of the features, for the asteroid Bennu.

(a) Mean persistence of the features, for the asteroid Bennu, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.94177 | 3.79649 | 3.39974 | 3.94184 | 3.79647 | 3.40228 |
| SURF | 3.6922 | 3.27795 | 2.86263 | 3.68053 | 3.27336 | 2.86507 |
| ORB | 3.92681 | 3.07794 | 2.59321 | 3.38898 | 2.86158 | 2.49438 |
| BRISK | 3.71815 | 3.28029 | 2.87689 | 3.69853 | 3.27078 | 2.87347 |
| KLT | 14.1018 | 4.08481 | 3.2699 | 13.7167 | 3.74924 | 2.27002 |
| STAR + BRIEF | 3.90276 | 4.08395 | 3.82125 | 3.88753 | 3.81141 | 3.65381 |
| SPV6 | 10.3781 | 6.48136 | 5.11702 | 10.3495 | 6.74517 | 4.98776 |
| SuperPoint Base | 4.14791 | 3.18954 | 2.68821 | 3.61017 | 3.03953 | 2.71074 |
| SuperPoint Eros | 4.5114 | 3.33113 | 2.89007 | 3.85619 | 3.20619 | 2.70612 |
| SuperPoint Bennu | 3.01246 | 2.63279 | 2.44218 | 2.80443 | 2.65957 | 2.37269 |

(b) Mean persistence of the features, for the asteroid Bennu, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.84404 | 3.74241 | 3.41719 | 3.84129 | 3.74761 | 3.42184 |
| SURF | 3.54201 | 3.092 | 2.72689 | 3.50814 | 3.06906 | 2.72328 |
| ORB | 3.81708 | 3.21338 | 2.7255 | 3.65282 | 2.97764 | 2.60255 |
| BRISK | 3.69603 | 3.44897 | 3.03868 | 3.68035 | 3.43027 | 3.03021 |
| KLT | 22.6188 | 10.2808 | 5.45712 | 21.7785 | 9.45451 | 4.97273 |
| STAR + BRIEF | 3.94974 | 3.88973 | 3.79838 | 3.92083 | 3.91382 | 3.8145 |
| SPV6 | 14.9336 | 9.73575 | 7.15116 | 16.5814 | 9.44737 | 7.59589 |
| SuperPoint Base | 6.55556 | 4.53374 | 3.65563 | 4.97222 | 3.43671 | 2.84211 |
| SuperPoint Eros | 6.07653 | 5.30928 | 3.63743 | 5.0618 | 3.88506 | 3.25714 |
| SuperPoint Bennu | 4.77143 | 3.23656 | 2.59333 | 4.46108 | 3.01961 | 2.59589 |

Table E.4

(a) Mean persistence of the features, for the asteroid Bennu, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 4.04408 | 3.76831 | 3.30451 | 4.0467 | 3.77332 | 3.3126 |
| SURF | 3.67603 | 3.24373 | 2.81603 | 3.66653 | 3.24142 | 2.81842 |
| ORB | 3.82135 | 2.9682 | 2.51138 | 3.31412 | 2.75152 | 2.42495 |
| BRISK | 3.38786 | 2.93126 | 2.61684 | 3.36223 | 2.92617 | 2.61468 |
| KLT | 13.264 | 4.23938 | 3.40494 | 12.6039 | 3.87061 | 2.35387 |
| STAR + BRIEF | 3.69644 | 3.66317 | 3.5412 | 3.63255 | 3.54145 | 3.40888 |
| SPV6 | 10.2315 | 6.48299 | 5.20152 | 9.9404 | 6.28571 | 5.06438 |
| SuperPoint Base | 3.97329 | 2.93684 | 2.61993 | 3.73702 | 2.84556 | 2.61728 |
| SuperPoint Eros | 4.45098 | 3.19489 | 2.6548 | 3.71769 | 2.85921 | 2.6124 |
| SuperPoint Bennu | 3.06583 | 2.75 | 2.39925 | 3.01736 | 2.6803 | 2.34836 |

(b) Mean persistence of the features, for the asteroid Bennu, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 2 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.83461 | 3.71509 | 3.39145 | 3.83392 | 3.72244 | 3.4089 |
| SURF | 3.50963 | 3.05336 | 2.70614 | 3.47789 | 3.03701 | 2.70704 |
| ORB | 3.71909 | 3.19904 | 2.65053 | 3.46526 | 2.99301 | 2.54865 |
| BRISK | 3.70321 | 3.26699 | 2.85035 | 3.67994 | 3.2451 | 2.84115 |
| KLT | 20.057 | 9.15252 | 5.14026 | 19.1078 | 8.35911 | 4.57856 |
| STAR + BRIEF | 3.75712 | 3.88559 | 3.82987 | 3.73559 | 3.8196 | 3.75913 |
| SPV6 | 14.7983 | 9.55738 | 6.90608 | 16.9294 | 10.2078 | 6.51553 |
| SuperPoint Base | 6.06931 | 4.03049 | 3.59884 | 4.48889 | 3.32278 | 3.11111 |
| SuperPoint Eros | 8.62051 | 5.26136 | 3.25309 | 5.12575 | 3.54545 | 2.85333 |
| SuperPoint Bennu | 4.33742 | 3 | 2.73099 | 4.10059 | 2.93421 | 2.56522 |

### E.2.2. Itokawa

Table E.5: Mean persistence of the features, for the asteroid Itokawa.

(a) Mean persistence of the features, for the asteroid Itokawa, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.71548 | 3.86905 | 3.8657 | 3.7121 | 3.86985 | 3.8792 |
| SURF | 3.84614 | 3.71928 | 3.63974 | 3.8316 | 3.7057 | 3.63004 |
| ORB | 3.93726 | 3.74614 | 3.58782 | 3.61127 | 3.26635 | 3.22851 |
| BRISK | 3.73562 | 3.6118 | 3.41459 | 3.59019 | 3.43105 | 3.22207 |
| KLT | 17.3614 | 9.5062 | 6.5383 | 15.9531 | 8.49211 | 6.00306 |
| STAR + BRIEF | 4.36307 | 4.61242 | 4.36296 | 4.32579 | 4.33297 | 4.25448 |
| SPV6 | 14.1037 | 8.7044 | 6.67596 | 14.5 | 8.85547 | 6.39863 |
| SuperPoint Base | 8.10072 | 5.65591 | 4.75281 | 7.62782 | 5.34728 | 4.54286 |
| SuperPoint Eros | 12.6351 | 6.88693 | 5.35612 | 10.4039 | 5.768 | 4.80426 |
| SuperPoint Bennu | 6.14286 | 3.89007 | 3.49057 | 4.94982 | 3.86192 | 3.65044 |

(b) Mean persistence of the features, for the asteroid Itokawa, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.57178 | 3.76042 | 3.76105 | 3.57172 | 3.76515 | 3.77272 |
| SURF | 3.69573 | 3.65108 | 3.6315 | 3.67652 | 3.6178 | 3.59304 |
| ORB | 3.6972 | 3.942 | 3.55222 | 3.65256 | 3.70825 | 3.27679 |
| BRISK | 3.74052 | 3.65861 | 3.56632 | 3.7109 | 3.61823 | 3.49649 |
| KLT | 20.6975 | 12.8523 | 10.4583 | 19.7833 | 11.8142 | 9.56223 |
| STAR + BRIEF | 6.28959 | 4.69066 | 4.17107 | 6.28345 | 4.58588 | 4.17745 |
| SPV6 | 20.3908 | 11.2751 | 9.13174 | 18.8772 | 11.8442 | 8.05882 |
| SuperPoint Base | 11.4247 | 7.72903 | 5.78146 | 8.85315 | 6.96154 | 5.29197 |
| SuperPoint Eros | 15.1887 | 9.06849 | 7.4863 | 11.6691 | 7.8662 | 6.21429 |
| SuperPoint Bennu | 13.2888 | 8.2459 | 5.63584 | 13.1319 | 7.09697 | 5.4386 |

Table E.5

(a) Mean persistence of the features, for the asteroid Itokawa, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.7289 | 3.77575 | 3.72029 | 3.72597 | 3.78477 | 3.74608 |
| SURF | 3.82681 | 3.65127 | 3.4461 | 3.80319 | 3.61853 | 3.42936 |
| ORB | 4.03578 | 3.7637 | 3.40312 | 3.68954 | 3.26609 | 3.07171 |
| BRISK | 3.73072 | 3.50567 | 3.29716 | 3.58043 | 3.33476 | 3.12775 |
| KLT | 15.5365 | 9.01837 | 6.38245 | 13.9873 | 7.74947 | 5.46306 |
| STAR + BRIEF | 4.30826 | 4.51656 | 4.24597 | 4.2912 | 4.21519 | 4.07653 |
| SPV6 | 14.0109 | 8.58657 | 6.39098 | 13.6935 | 8.26437 | 5.98092 |
| SuperPoint Base | 7.4083 | 5.0963 | 4.18705 | 6.18391 | 4.68872 | 3.84836 |
| SuperPoint Eros | 9.72318 | 6.57877 | 4.80632 | 8.04615 | 5.06084 | 4.41667 |
| SuperPoint Bennu | 5.03125 | 3.79026 | 3.11808 | 4.65414 | 3.68 | 3.25207 |

(b) Mean persistence of the features, for the asteroid Itokawa, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.70168 | 3.75602 | 3.70366 | 3.70281 | 3.76887 | 3.73444 |
| SURF | 3.81501 | 3.72588 | 3.53947 | 3.78661 | 3.66392 | 3.49609 |
| ORB | 4.02254 | 3.8614 | 3.65649 | 3.96442 | 3.62718 | 3.42516 |
| BRISK | 3.9087 | 3.78658 | 3.54409 | 3.87595 | 3.723 | 3.4863 |
| KLT | 22.5002 | 14.6694 | 11.031 | 21.5768 | 13.3204 | 9.81755 |
| STAR + BRIEF | 9.74222 | 5.66856 | 4.78793 | 9.49333 | 5.5 | 4.51521 |
| SPV6 | 22.0505 | 13.5581 | 9.04819 | 21.8553 | 12.2081 | 9.06122 |
| SuperPoint Base | 16.2629 | 8.83158 | 7.26374 | 12.7651 | 7.7375 | 5.09028 |
| SuperPoint Eros | 16.2053 | 9.54601 | 6.92216 | 11.4649 | 7.32667 | 5.95 |
| SuperPoint Bennu | 9.71186 | 6.31977 | 4.73171 | 8.91139 | 5.91096 | 4.13514 |

### E.2.3. Eros

Table E.6: Mean persistence of the features, for the asteroid Eros.

(a) Mean persistence of the features, for the asteroid Eros, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 50 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.64389 | 3.6585 | 3.63494 | 3.64306 | 3.65647 | 3.6404 |
| SURF | 3.75853 | 3.54934 | 3.29649 | 3.74014 | 3.5309 | 3.27289 |
| ORB | 3.86828 | 3.52898 | 3.16931 | 3.66132 | 3.29181 | 2.94123 |
| BRISK | 3.64884 | 3.38113 | 3.1433 | 3.4987 | 3.20833 | 2.99859 |
| KLT | 11.6056 | 5.4406 | 4.11491 | 11.2093 | 5.30921 | 3.5832 |
| STAR + BRIEF | 3.7679 | 3.74695 | 3.6956 | 3.72011 | 3.64541 | 3.58845 |
| SPV6 | 15.0749 | 8.18992 | 6.17257 | 15.3474 | 8.77041 | 6.22549 |
| SuperPoint Base | 6.49138 | 4.31304 | 3.575 | 5.32353 | 3.75962 | 3.19524 |
| SuperPoint Eros | 5.46154 | 4.34298 | 3.75641 | 4.89947 | 4.24762 | 3.51613 |
| SuperPoint Bennu | 4.01826 | 3.23077 | 2.94836 | 4.20994 | 3.2069 | 2.71292 |

(b) Mean persistence of the features, for the asteroid Eros, for a light intensity of $1.0\frac{W}{m^2}$ and a radius of 70 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.6037 | 3.67361 | 3.64407 | 3.60439 | 3.67277 | 3.64513 |
| SURF | 3.73722 | 3.54674 | 3.3287 | 3.71838 | 3.52412 | 3.31119 |
| ORB | 3.90603 | 3.64364 | 3.1914 | 3.76572 | 3.42429 | 2.95893 |
| BRISK | 3.65116 | 3.44593 | 3.17445 | 3.56504 | 3.33237 | 3.08364 |
| KLT | 15.7029 | 6.44002 | 4.65795 | 14.8696 | 6.01523 | 4.0683 |
| STAR + BRIEF | 3.86398 | 4.01541 | 3.89962 | 3.8475 | 3.95048 | 3.74011 |
| SPV6 | 17.0788 | 10.0518 | 7.35912 | 18.4444 | 10.4083 | 7.84967 |
| SuperPoint Base | 8.13174 | 5.29714 | 3.8253 | 6.44805 | 4.56923 | 3.42953 |
| SuperPoint Eros | 11.3795 | 6.10828 | 4.5414 | 7.75163 | 4.70677 | 3.856 |
| SuperPoint Bennu | 5.7541 | 3.68387 | 3.07471 | 4.94194 | 3.42857 | 2.84615 |

Table E.6

(a) Mean persistence of the features, for the asteroid Eros, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 0.7 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.73364 | 3.73234 | 3.55114 | 3.73303 | 3.74329 | 3.56219 |
| SURF | 3.75726 | 3.46961 | 3.20548 | 3.73451 | 3.4592 | 3.1964 |
| ORB | 3.65881 | 3.4516 | 2.92314 | 3.46478 | 3.20246 | 2.74188 |
| BRISK | 3.62481 | 3.32922 | 3.04746 | 3.48146 | 3.1689 | 2.91332 |
| KLT | 10.3592 | 5.32846 | 4.23224 | 9.77304 | 5.13018 | 3.57647 |
| STAR + BRIEF | 3.82776 | 4.00221 | 3.89231 | 3.80184 | 3.82446 | 3.80854 |
| SPV6 | 14.4714 | 8.14407 | 6.13636 | 14.7711 | 8.5445 | 5.30729 |
| SuperPoint Base | 6.64463 | 3.87137 | 3.41441 | 5.37872 | 3.5907 | 3.47525 |
| SuperPoint Eros | 6.36929 | 4.32821 | 3.45575 | 5.84163 | 3.83422 | 3.38144 |
| SuperPoint Bennu | 3.65657 | 3.095 | 2.84685 | 3.7663 | 3.05236 | 2.855 |

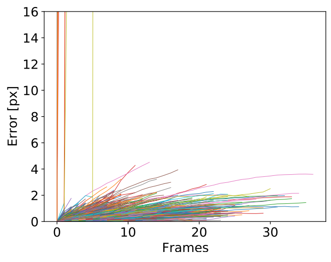
(b) Mean persistence of the features, for the asteroid Eros, for a light intensity of $5.0\frac{W}{m^2}$ and a radius of 1.5 $Km$.

| | Average feature persistence [frames]. | | | | | |
|---|---|---|---|---|---|---|
| | W/o outliers rejection | | | W/ outliers rejection | | |
| IP step | 1 | 2 | 3 | 1 | 2 | 3 |
| SIFT | 3.6712 | 3.72898 | 3.59581 | 3.66855 | 3.7314 | 3.60666 |
| SURF | 3.74029 | 3.50007 | 3.25742 | 3.70635 | 3.4848 | 3.24086 |
| ORB | 3.92517 | 3.55024 | 3.08434 | 3.78774 | 3.34025 | 2.95167 |
| BRISK | 3.66183 | 3.35218 | 3.11126 | 3.56841 | 3.24481 | 3.01795 |
| KLT | 13.2494 | 6.51914 | 4.68458 | 12.515 | 6.03903 | 3.8757 |
| STAR + BRIEF | 3.6492 | 3.84317 | 3.60241 | 3.7133 | 3.74952 | 3.44444 |
| SPV6 | 18.8475 | 10.7438 | 8.0129 | 18.8418 | 11.6176 | 7.64238 |
| SuperPoint Base | 7.28289 | 5.03896 | 3.78947 | 6.51748 | 4.15441 | 3.82051 |
| SuperPoint Eros | 9.33523 | 5.97059 | 4.43636 | 6.69427 | 5.35032 | 3.58273 |
| SuperPoint Bennu | 5.10145 | 3.65269 | 3.22222 | 4.43066 | 3.4 | 3.10606 |

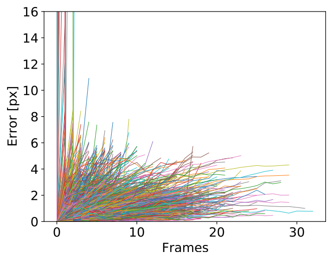
## E.3. Outliers evolution

Here are reported the charts that show the evolution of the outliers.

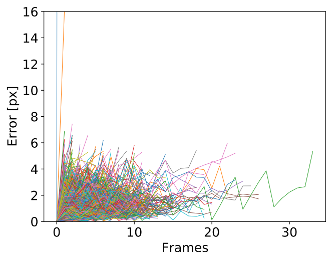
## E.3.1. Bennu



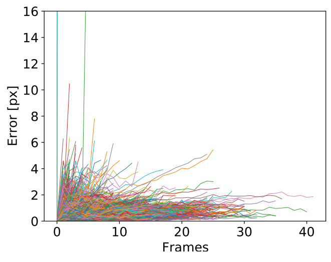
(a) SIFT



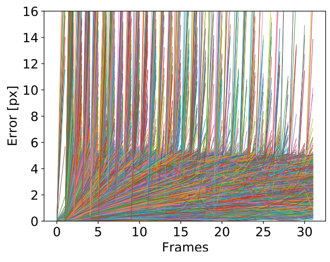
(b) SURF



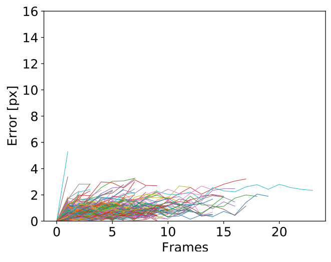
(c) ORB



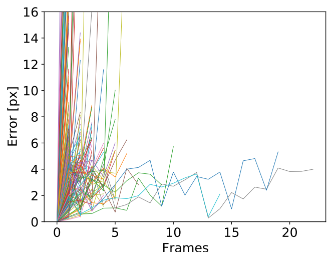
(d) BRISK



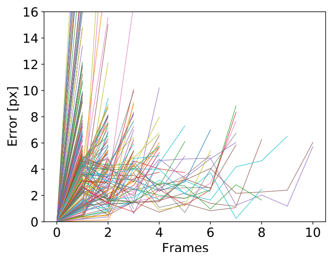
(e) KLT



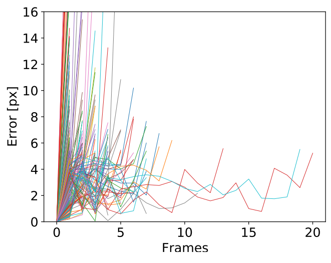
(f) STAR + BRIEF



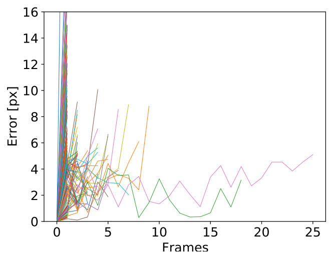
(g) SPV6



(h) SuperPoint Base

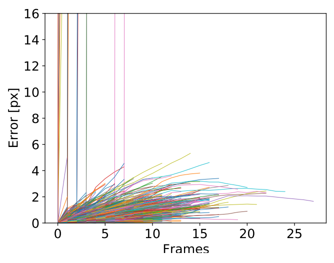


(i) SuperPoint Eros

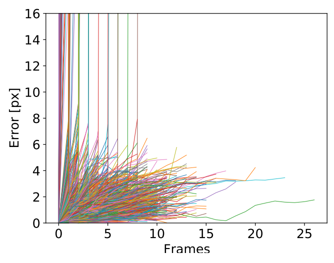


(j) SuperPoint Bennu

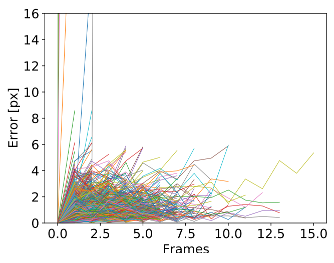
Figure E.1: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



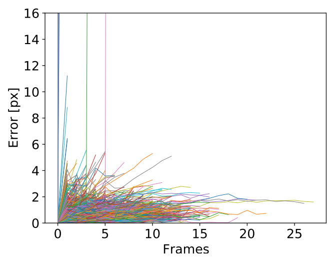
(a) SIFT



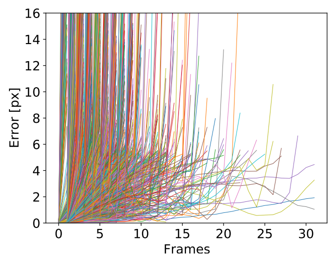
(b) SURF



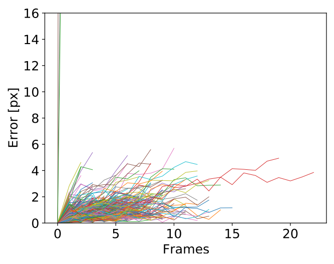
(c) ORB



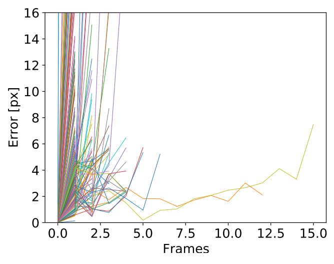
(d) BRISK



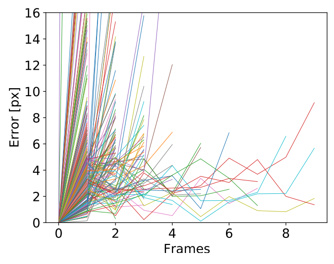
(e) KLT



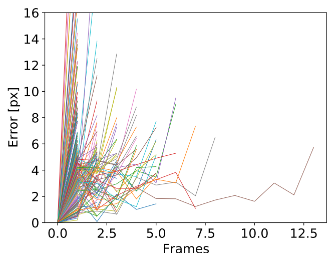
(f) STAR + BRIEF



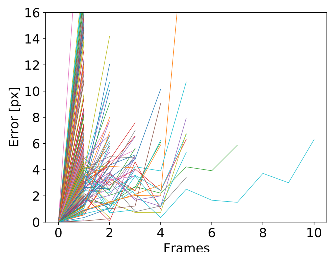
(g) SPV6



(h) SuperPoint Base

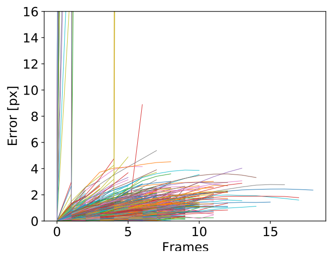


(i) SuperPoint Eros

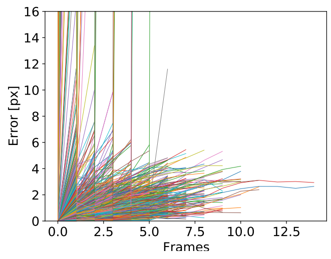


(j) SuperPoint Bennu

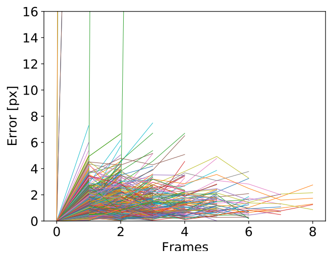
Figure E.2: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.



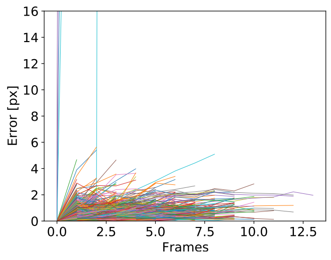
(a) SIFT



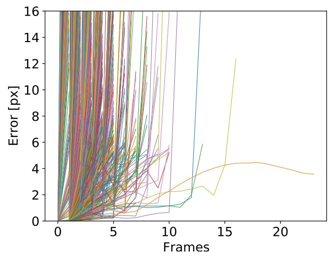
(b) SURF



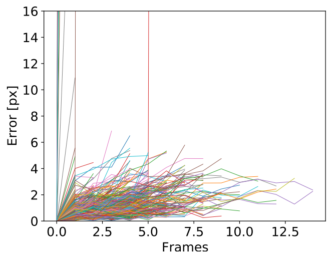
(c) ORB



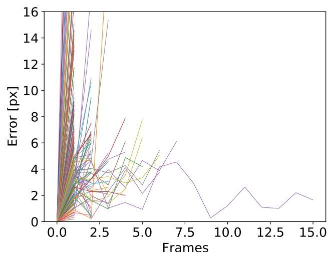
(d) BRISK



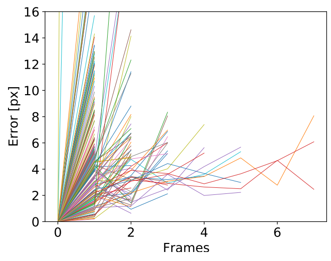
(e) KLT



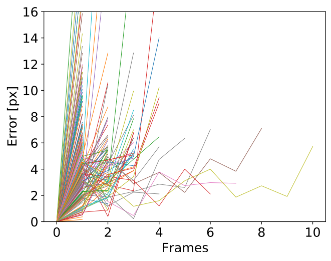
(f) STAR + BRIEF



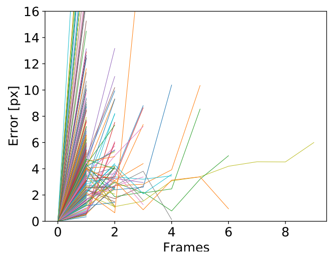
(g) SPV6



(h) SuperPoint Base

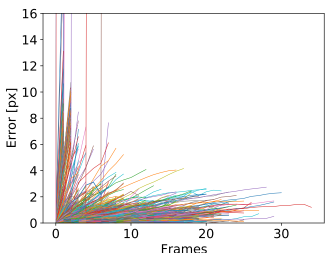


(i) SuperPoint Eros

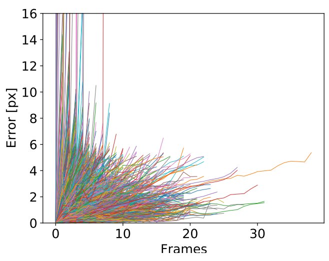


(j) SuperPoint Bennu

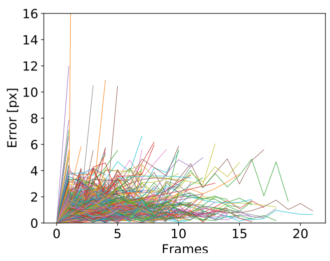
Figure E.3: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



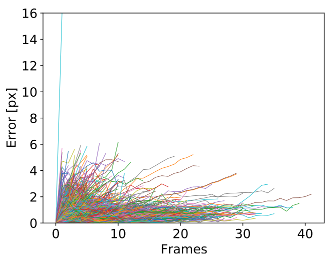
(a) SIFT



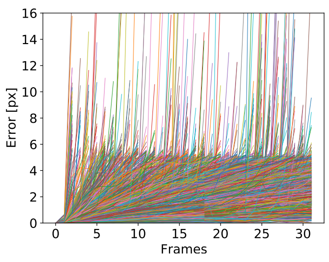
(b) SURF



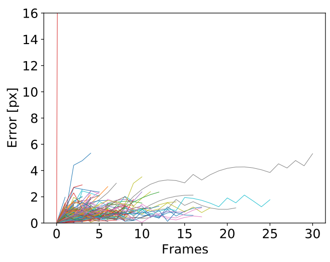
(c) ORB



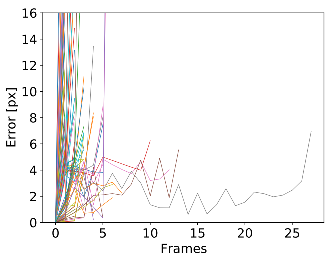
(d) BRISK



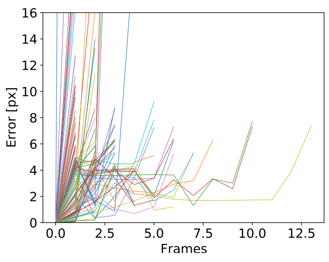
(e) KLT



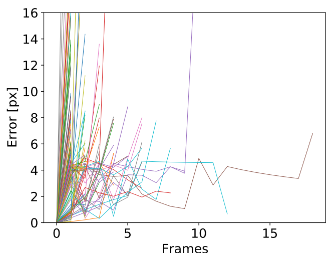
(f) STAR + BRIEF



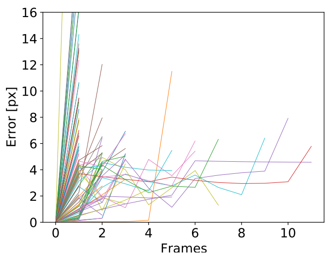
(g) SPV6



(h) SuperPoint Base

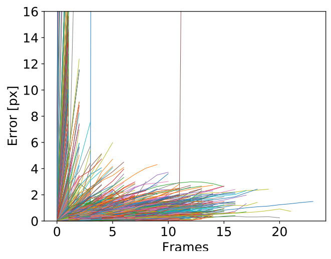


(i) SuperPoint Eros

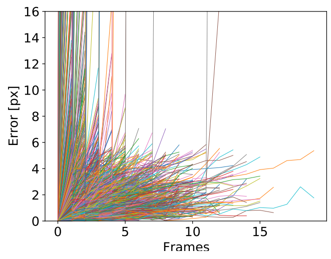


(j) SuperPoint Bennu

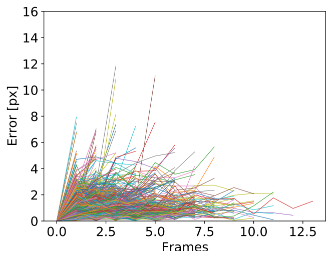
Figure E.4: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



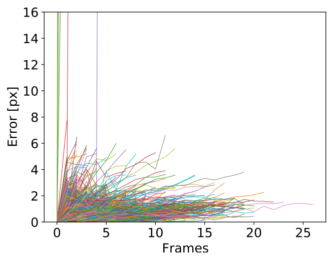
(a) SIFT



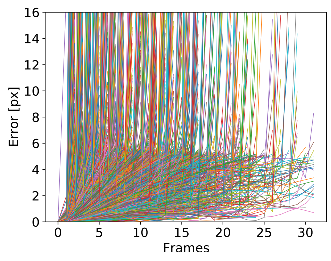
(b) SURF



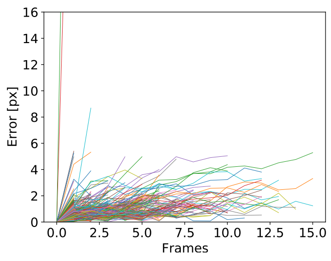
(c) ORB



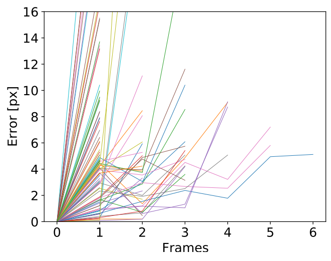
(d) BRISK



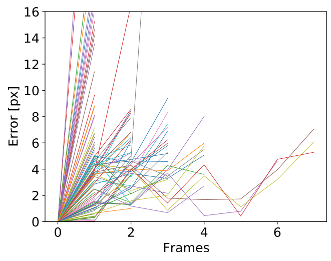
(e) KLT



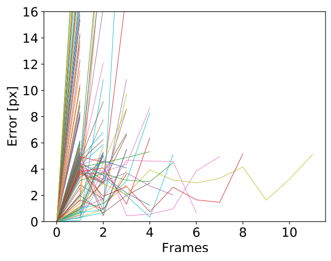
(f) STAR + BRIEF



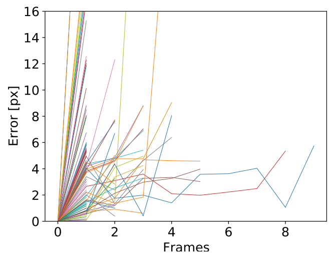
(g) SPV6



(h) SuperPoint Base

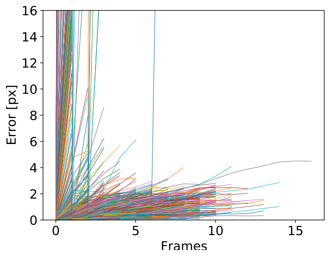


(i) SuperPoint Eros

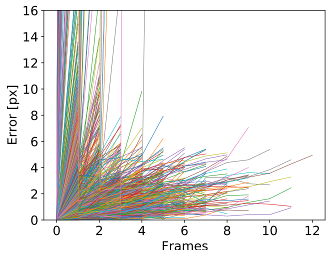


(j) SuperPoint Bennu

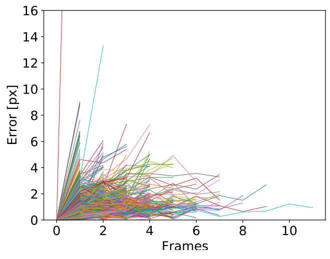
Figure E.5: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



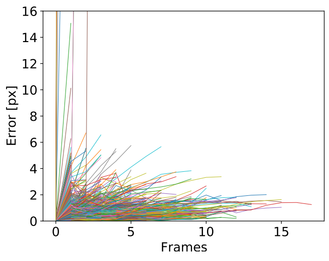
(a) SIFT



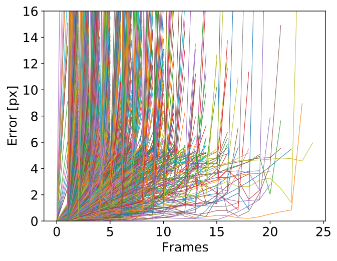
(b) SURF



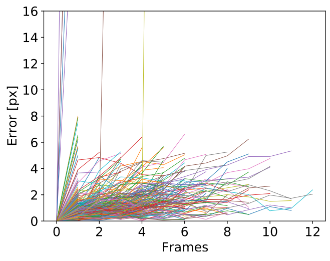
(c) ORB



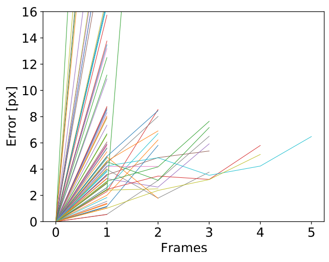
(d) BRISK



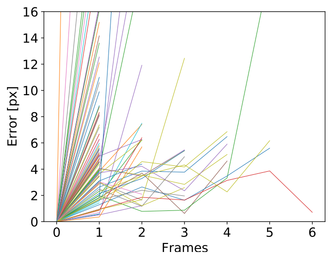
(e) KLT



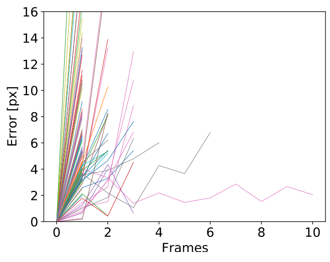
(f) STAR + BRIEF



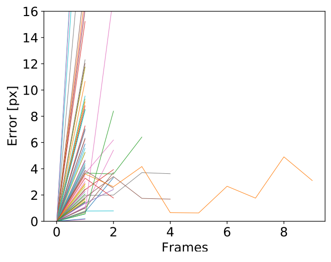
(g) SPV6



(h) SuperPoint Base

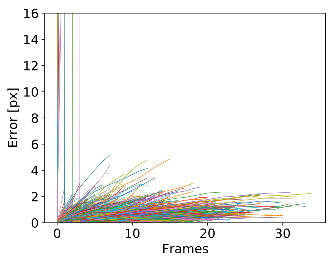


(i) SuperPoint Eros

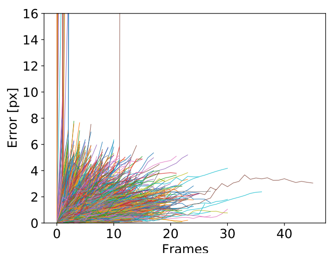


(j) SuperPoint Bennu

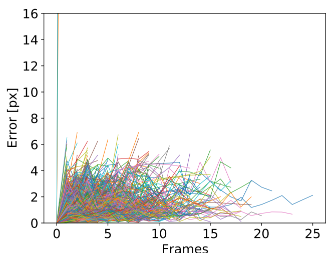
Figure E.6: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.



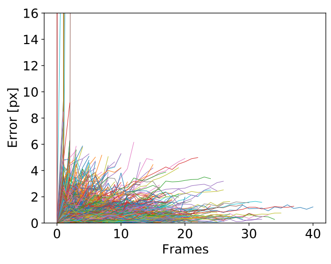
(a) SIFT



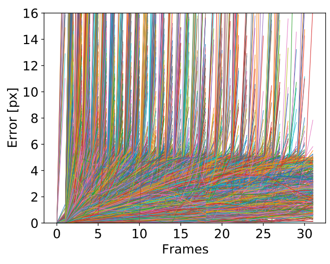
(b) SURF



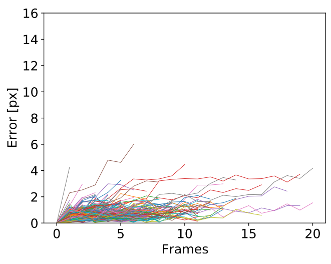
(c) ORB



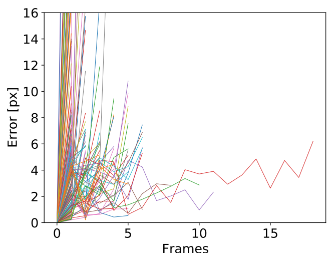
(d) BRISK



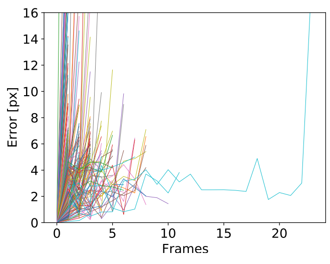
(e) KLT



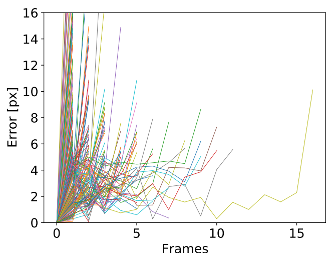
(f) STAR + BRIEF



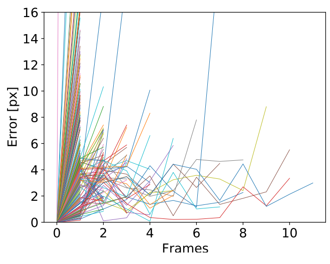
(g) SPV6



(h) SuperPoint Base

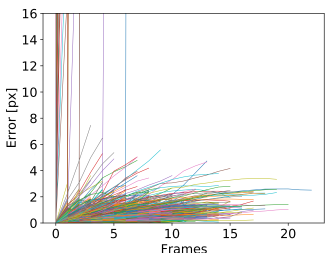


(i) SuperPoint Eros

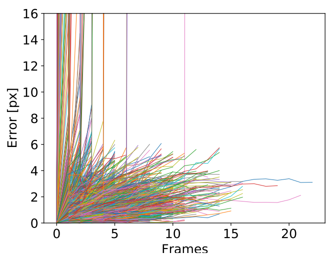


(j) SuperPoint Bennu

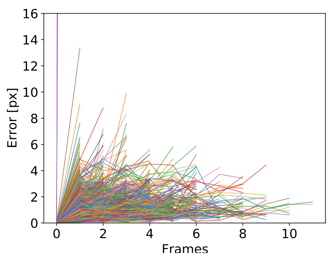
Figure E.7: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



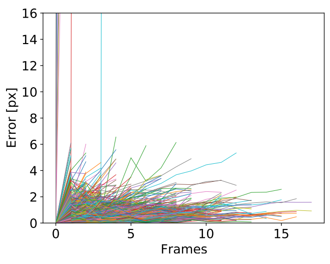
(a) SIFT



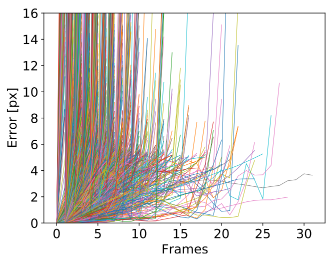
(b) SURF



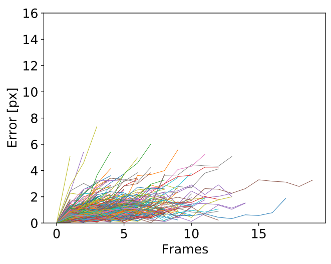
(c) ORB



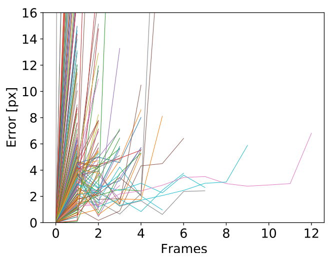
(d) BRISK



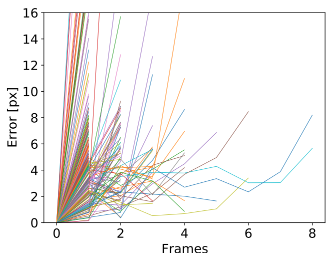
(e) KLT



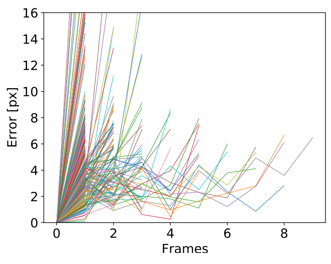
(f) STAR + BRIEF



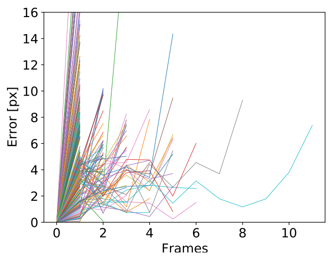
(g) SPV6



(h) SuperPoint Base

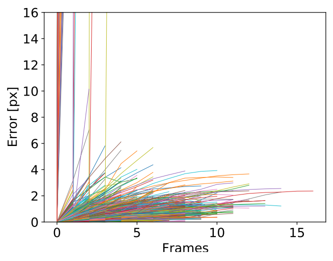


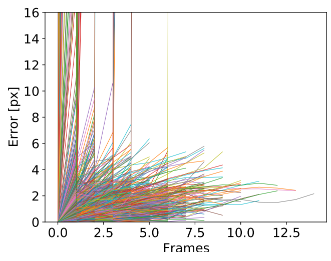
(i) SuperPoint Eros

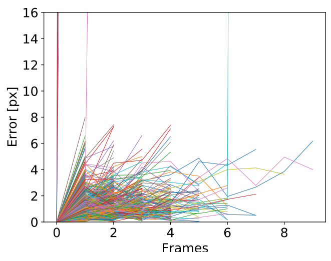


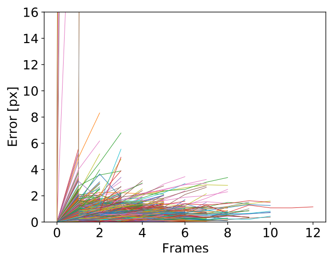
(j) SuperPoint Bennu

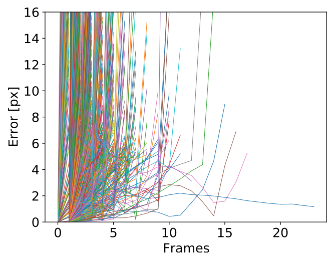
Figure E.8: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.

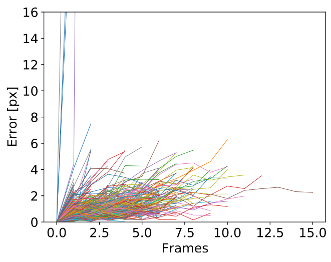

(a) SIFT

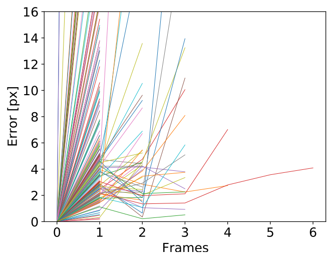

(b) SURF

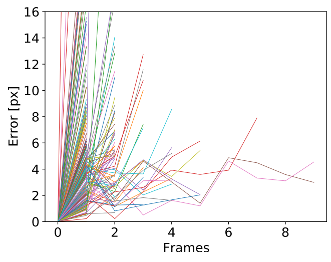

(c) ORB

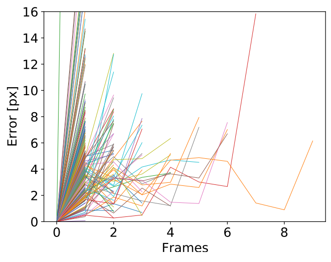

(d) BRISK


(e) KLT


(f) STAR + BRIEF


(g) SPV6


(h) SuperPoint Base


(i) SuperPoint Eros

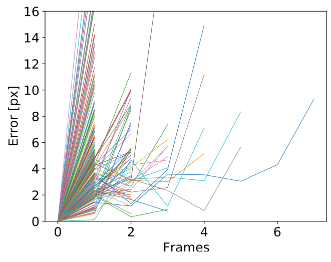
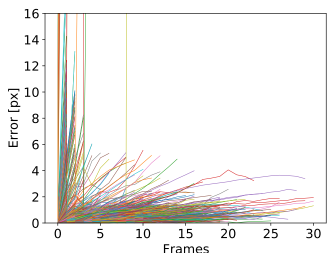
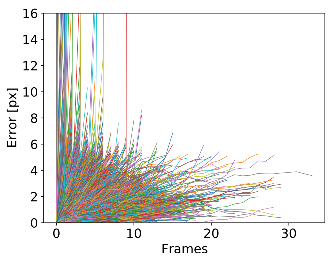
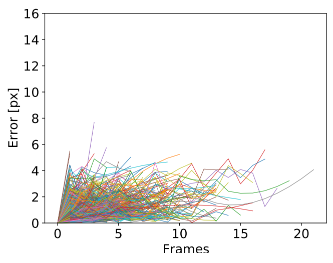
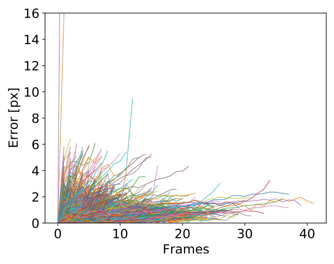

(j) SuperPoint Bennu

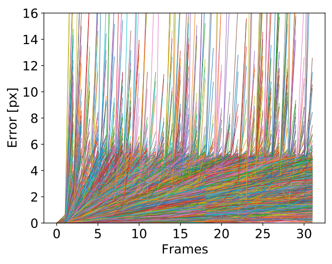
Figure E.9: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.

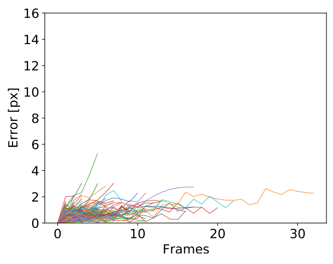

(a) SIFT

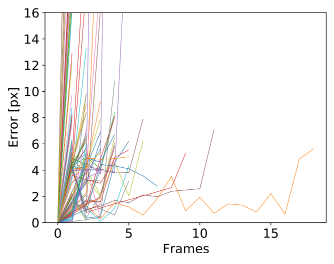

(b) SURF

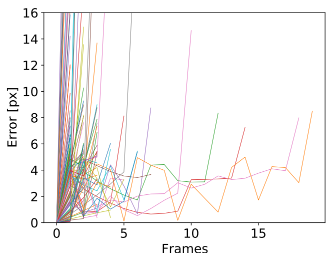

(c) ORB

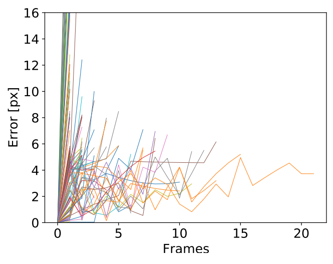

(d) BRISK


(e) KLT


(f) STAR + BRIEF


(g) SPV6


(h) SuperPoint Base


(i) SuperPoint Eros

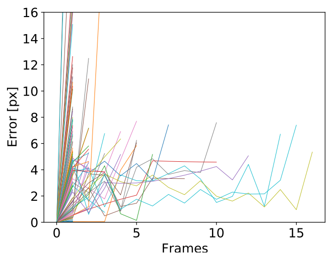
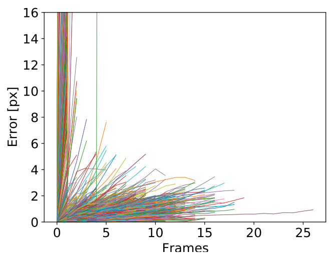
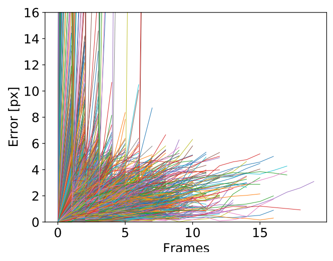

(j) SuperPoint Bennu

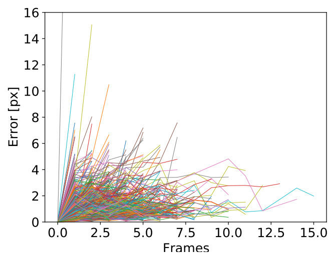
Figure E.10: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



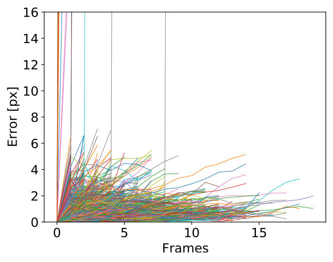
(a) SIFT



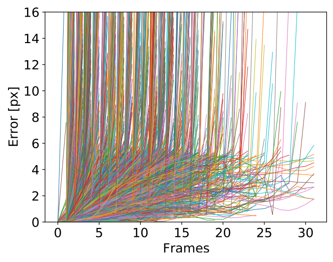
(b) SURF



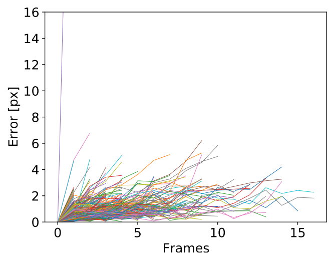
(c) ORB



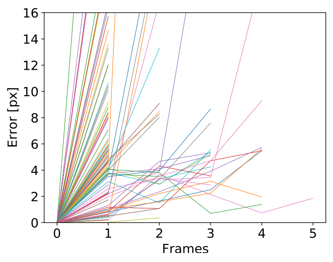
(d) BRISK



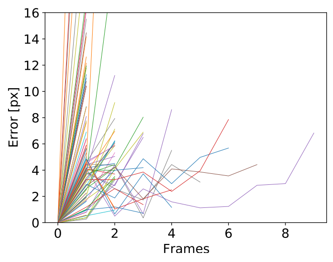
(e) KLT



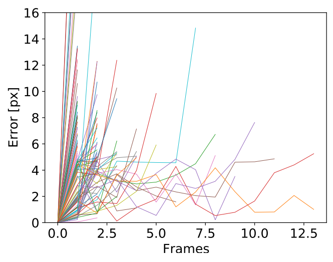
(f) STAR + BRIEF



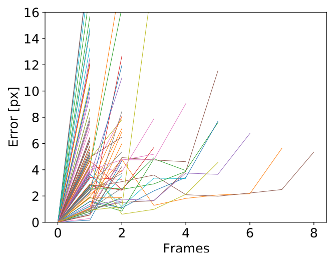
(g) SPV6



(h) SuperPoint Base

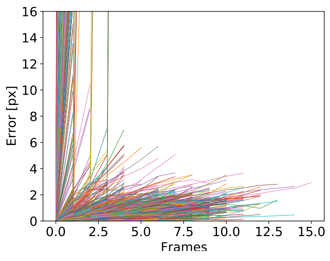


(i) SuperPoint Eros

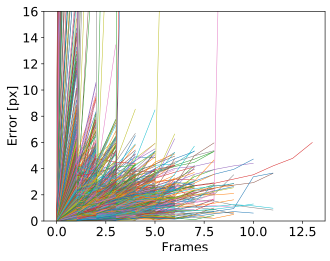


(j) SuperPoint Bennu

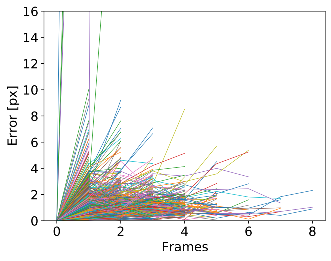
Figure E.11: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



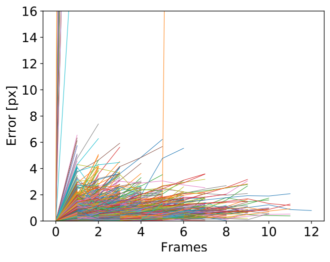
(a) SIFT



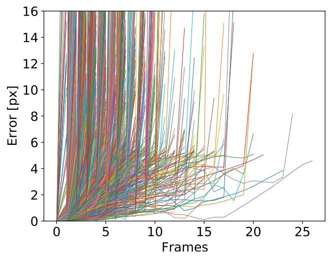
(b) SURF



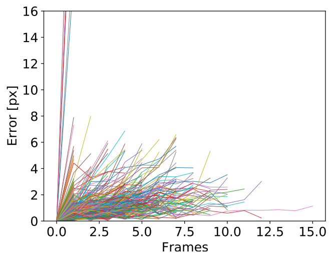
(c) ORB



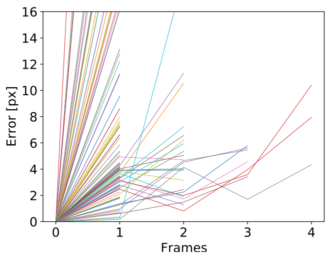
(d) BRISK



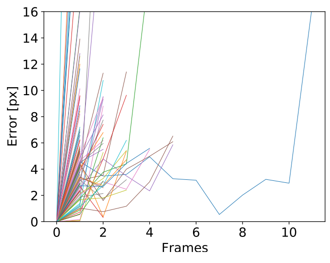
(e) KLT



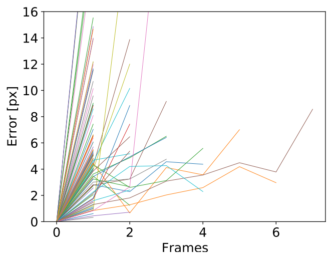
(f) STAR + BRIEF



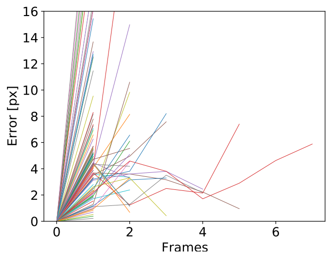
(g) SPV6



(h) SuperPoint Base



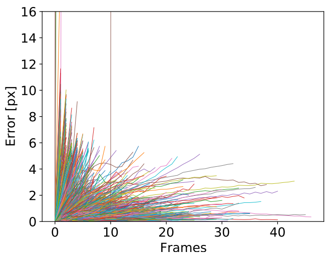
(i) SuperPoint Eros



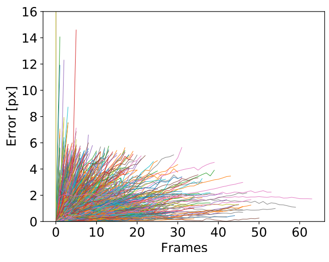
(j) SuperPoint Bennu

Figure E.12: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.

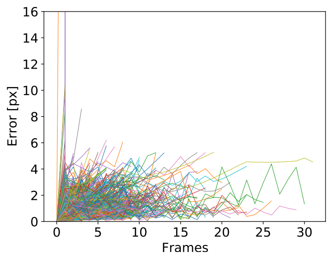
## E.3.2. Itokawa



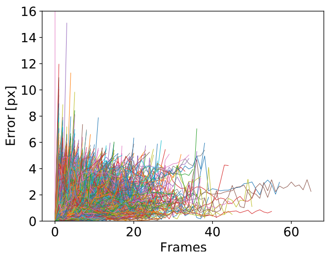
(a) SIFT



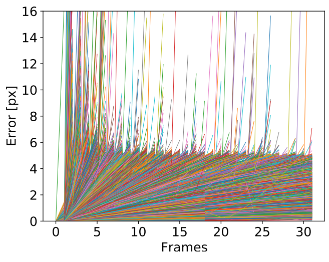
(b) SURF



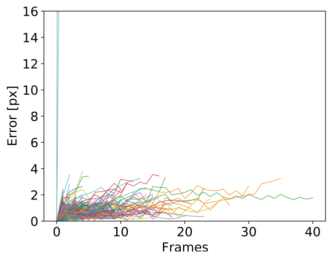
(c) ORB



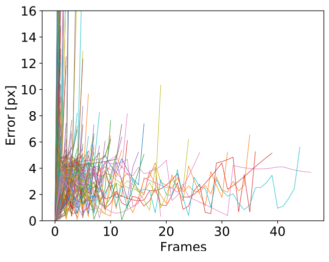
(d) BRISK



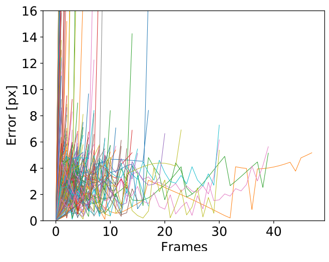
(e) KLT



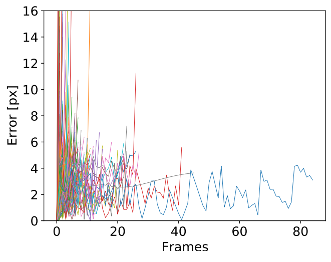
(f) STAR + BRIEF



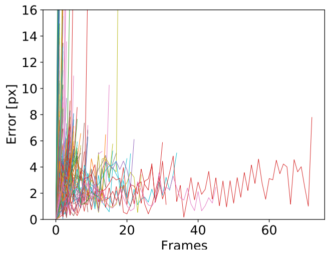
(g) SPV6



(h) SuperPoint Base

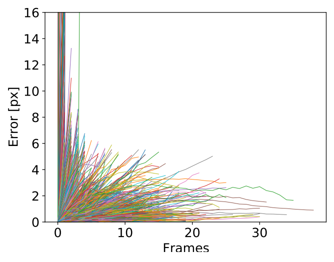


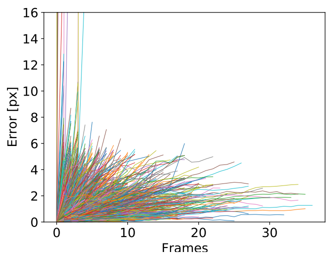
(i) SuperPoint Eros

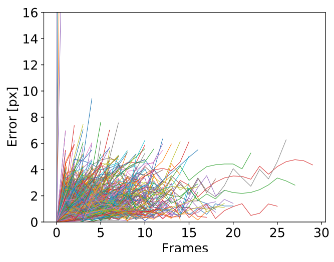


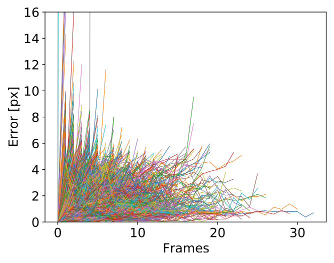
(j) SuperPoint Bennu

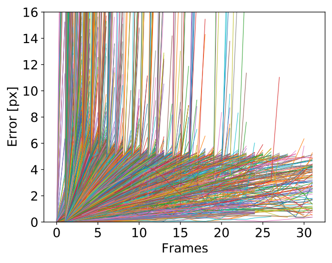
Figure E.13: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.

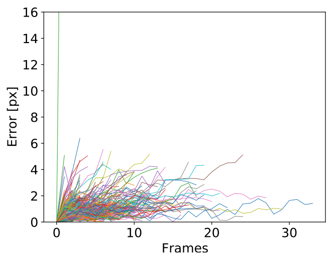

(a) SIFT

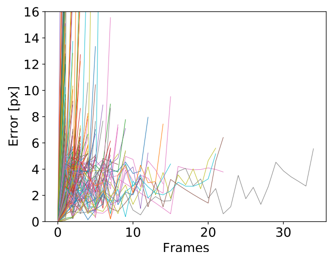

(b) SURF

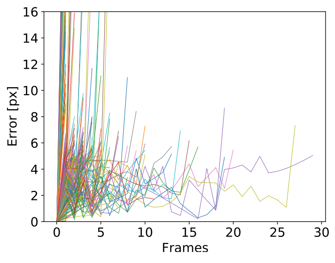

(c) ORB

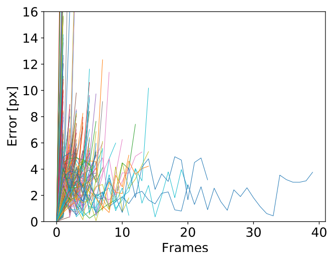

(d) BRISK


(e) KLT


(f) STAR + BRIEF


(g) SPV6


(h) SuperPoint Base


(i) SuperPoint Eros

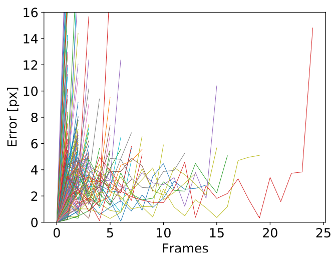
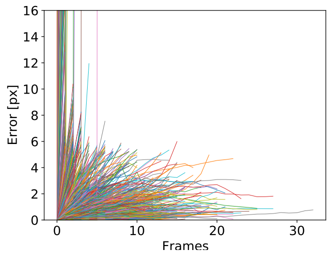
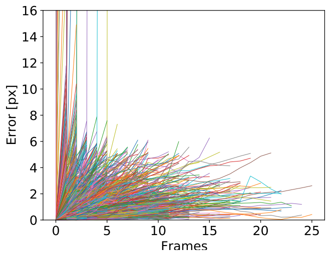

(j) SuperPoint Bennu

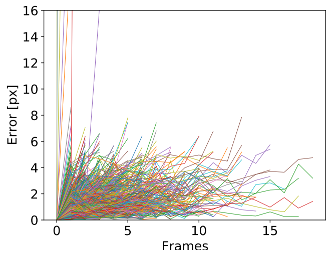
Figure E.14: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.



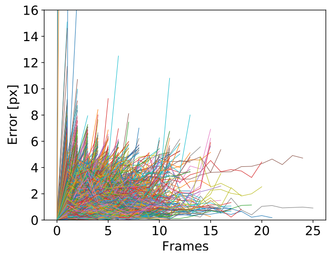
(a) SIFT



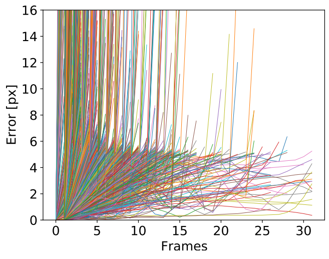
(b) SURF



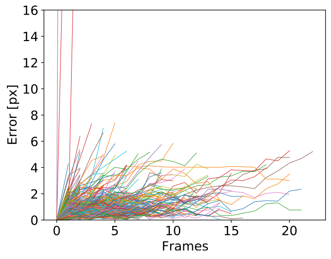
(c) ORB



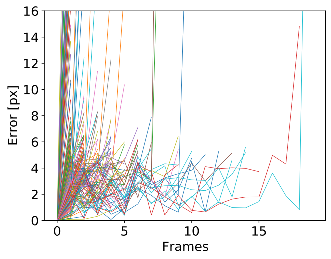
(d) BRISK



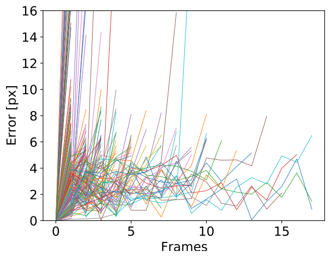
(e) KLT



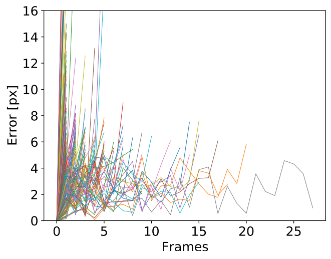
(f) STAR + BRIEF



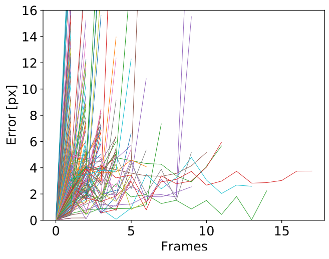
(g) SPV6



(h) SuperPoint Base

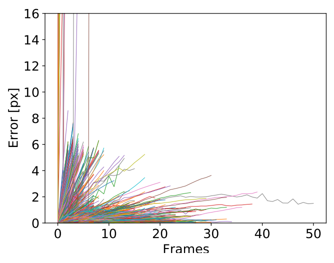


(i) SuperPoint Eros

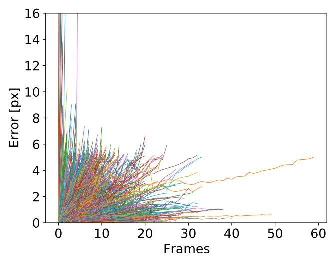


(j) SuperPoint Bennu

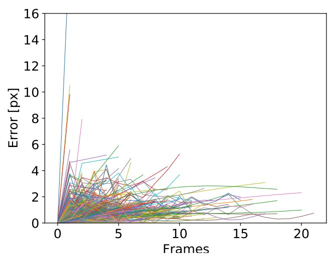
Figure E.15: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



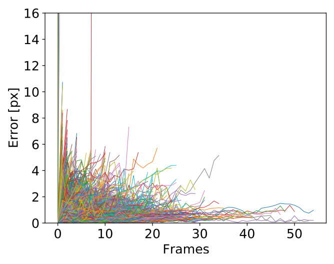
(a) SIFT



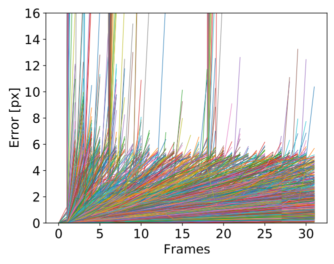
(b) SURF



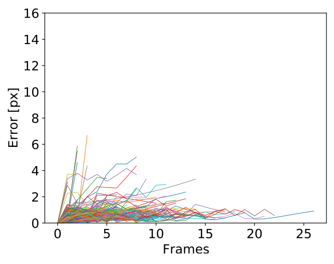
(c) ORB



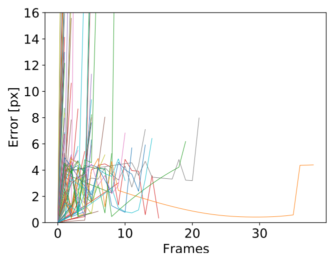
(d) BRISK



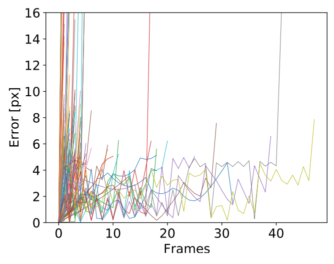
(e) KLT



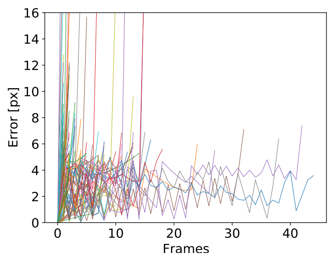
(f) STAR + BRIEF



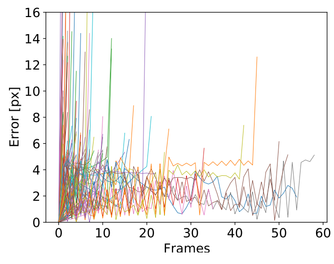
(g) SPV6



(h) SuperPoint Base

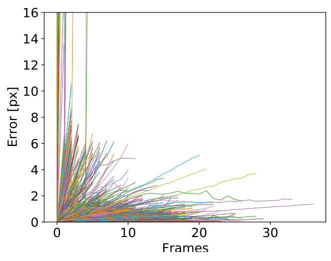


(i) SuperPoint Eros

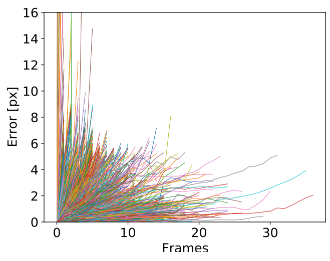


(j) SuperPoint Bennu

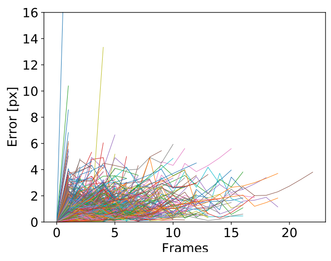
Figure E.16: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



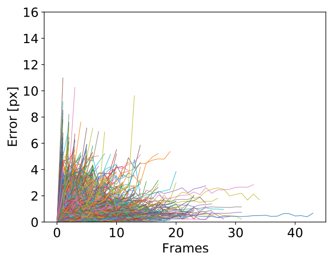
(a) SIFT



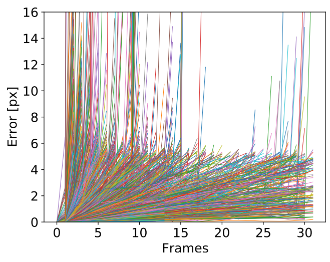
(b) SURF



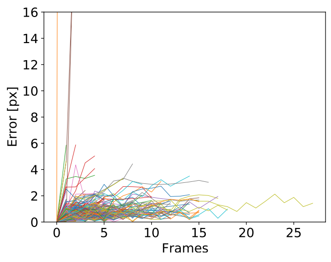
(c) ORB



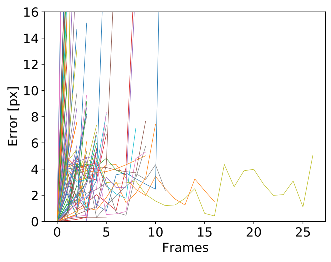
(d) BRISK



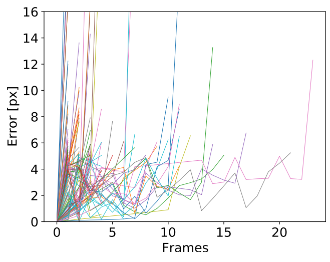
(e) KLT



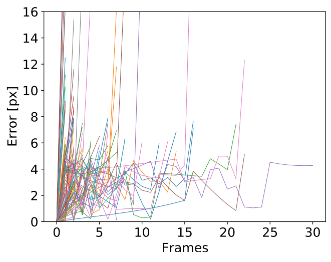
(f) STAR + BRIEF



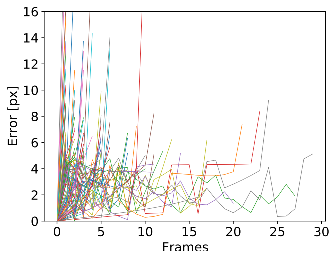
(g) SPV6



(h) SuperPoint Base

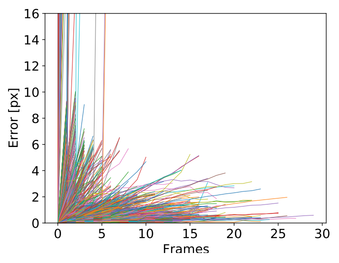


(i) SuperPoint Eros

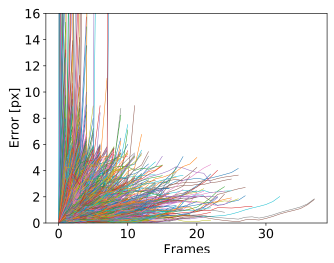


(j) SuperPoint Bennu

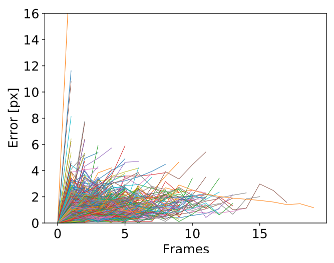
Figure E.17: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



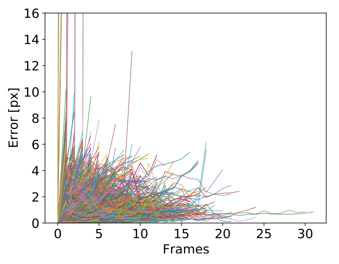
(a) SIFT



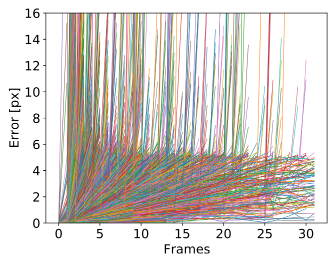
(b) SURF



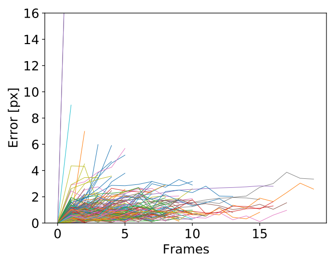
(c) ORB



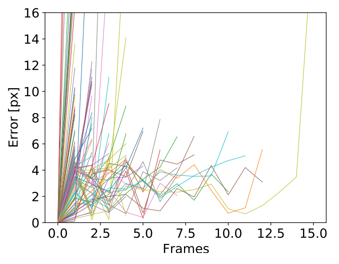
(d) BRISK



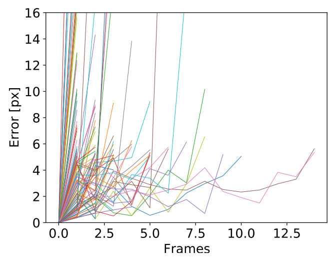
(e) KLT



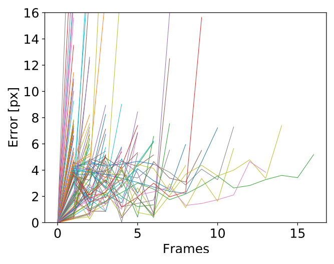
(f) STAR + BRIEF



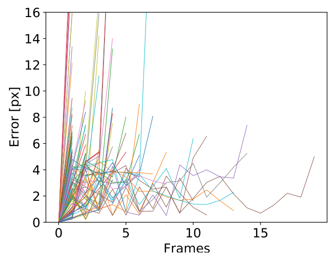
(g) SPV6



(h) SuperPoint Base

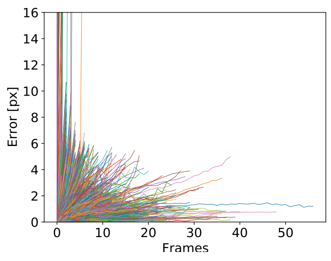


(i) SuperPoint Eros

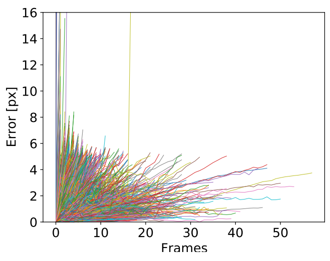


(j) SuperPoint Bennu

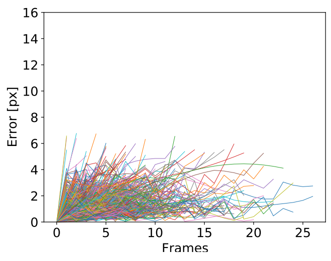
Figure E.18: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.



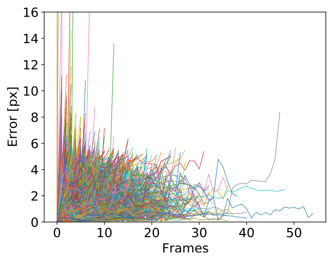
(a) SIFT



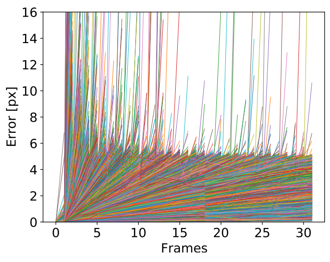
(b) SURF



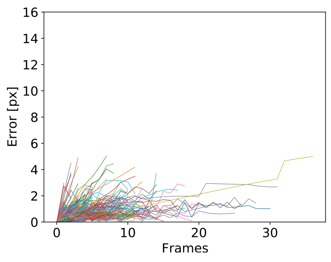
(c) ORB



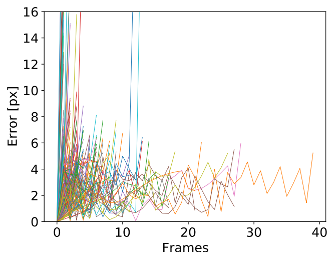
(d) BRISK



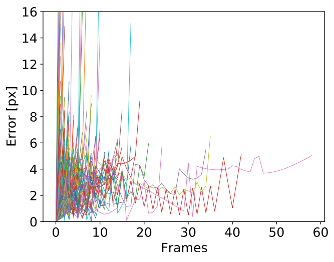
(e) KLT



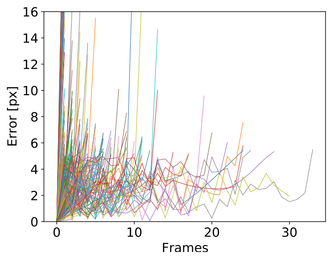
(f) STAR + BRIEF



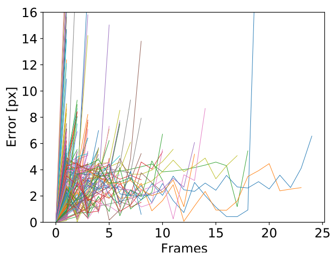
(g) SPV6



(h) SuperPoint Base

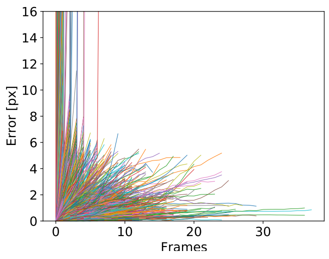


(i) SuperPoint Eros

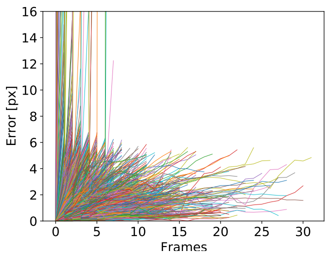


(j) SuperPoint Bennu

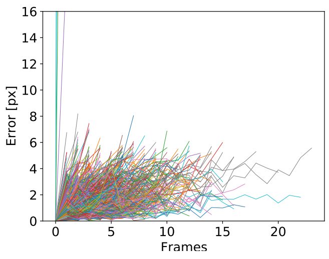
Figure E.19: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



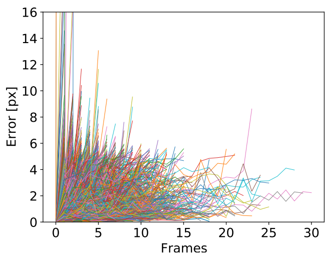
(a) SIFT



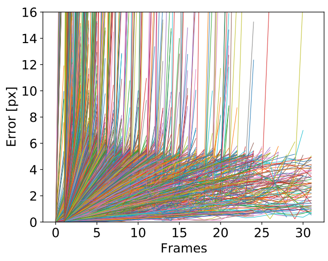
(b) SURF



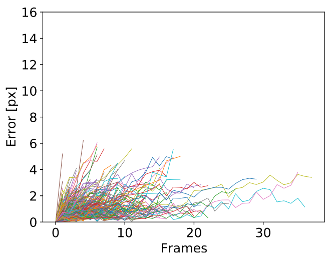
(c) ORB



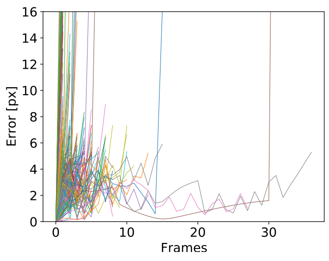
(d) BRISK



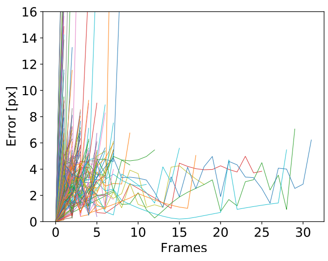
(e) KLT



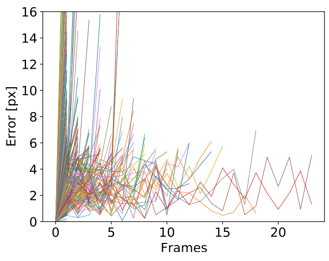
(f) STAR + BRIEF



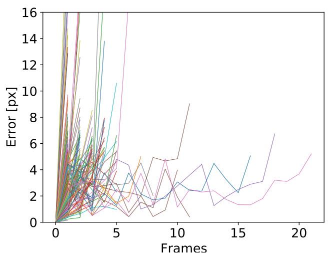
(g) SPV6



(h) SuperPoint Base

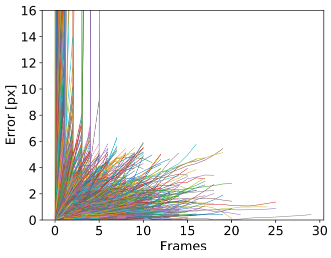


(i) SuperPoint Eros

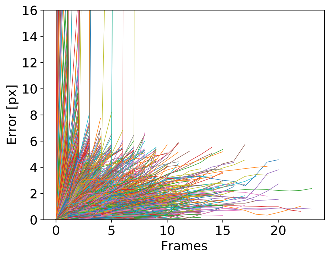


(j) SuperPoint Bennu

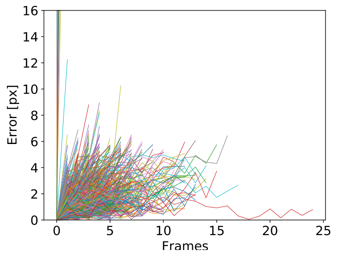
Figure E.20: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.



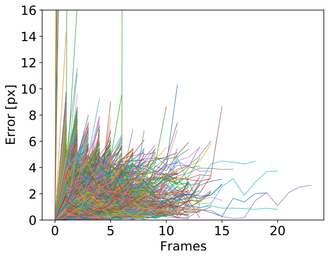
(a) SIFT



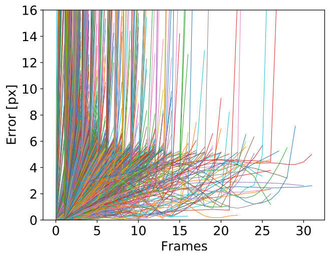
(b) SURF



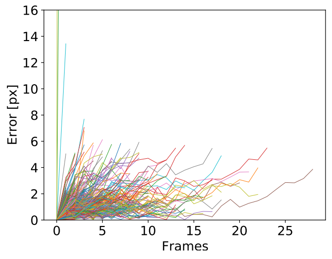
(c) ORB



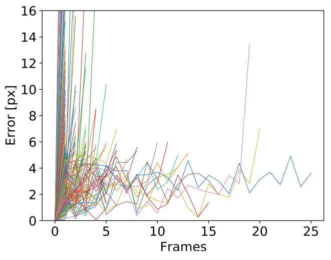
(d) BRISK



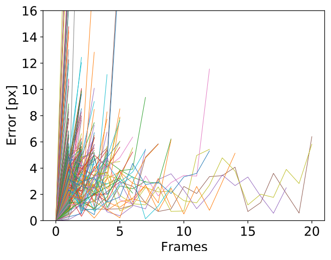
(e) KLT



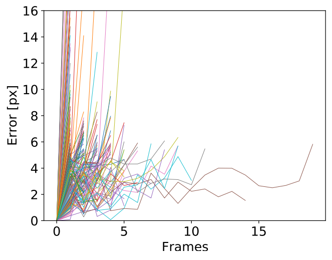
(f) STAR + BRIEF



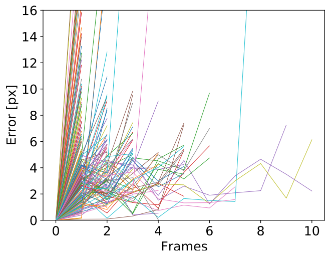
(g) SPV6



(h) SuperPoint Base

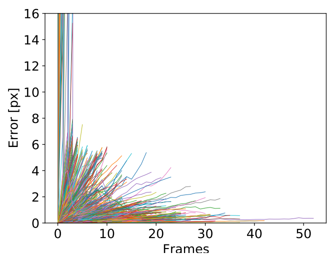


(i) SuperPoint Eros

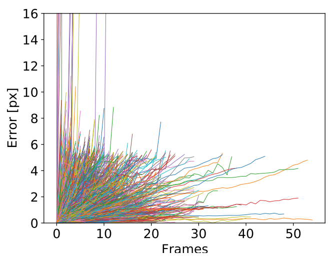


(j) SuperPoint Bennu

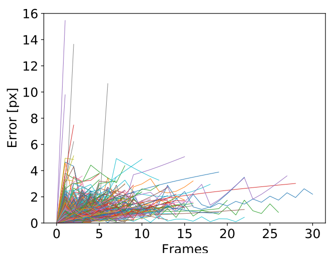
Figure E.21: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



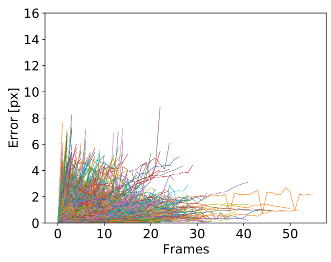
(a) SIFT



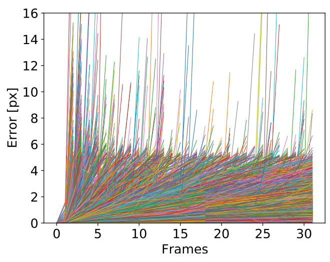
(b) SURF



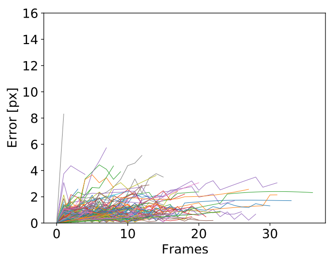
(c) ORB



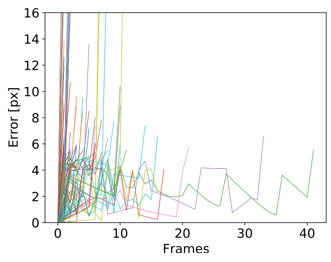
(d) BRISK



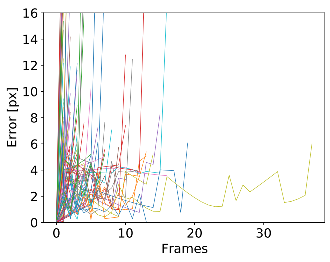
(e) KLT



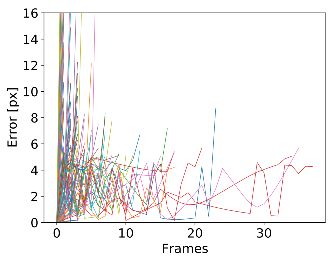
(f) STAR + BRIEF



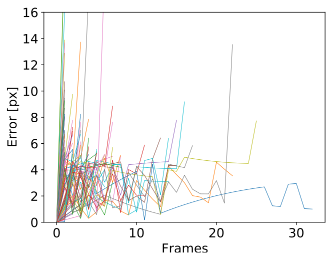
(g) SPV6



(h) SuperPoint Base

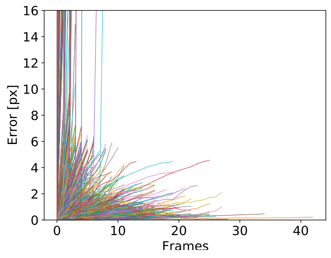


(i) SuperPoint Eros

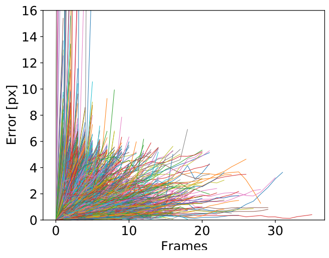


(j) SuperPoint Bennu

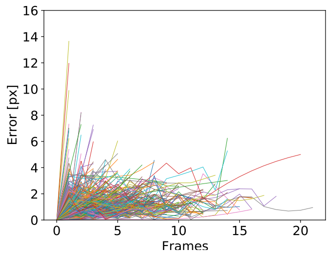
Figure E.22: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



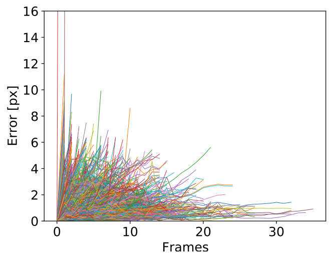
(a) SIFT



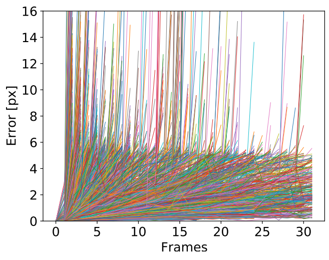
(b) SURF



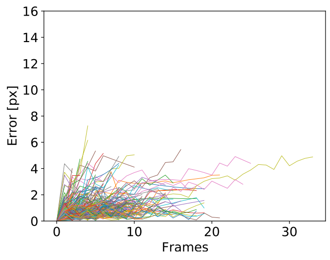
(c) ORB



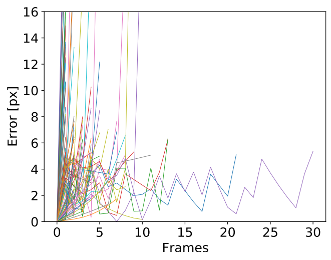
(d) BRISK



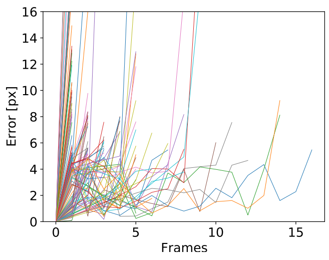
(e) KLT



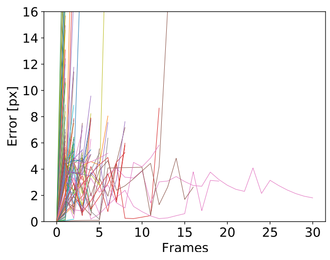
(f) STAR + BRIEF



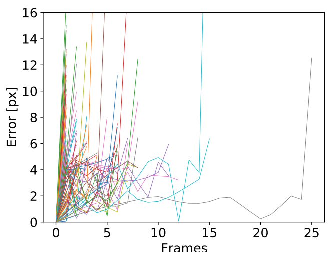
(g) SPV6



(h) SuperPoint Base

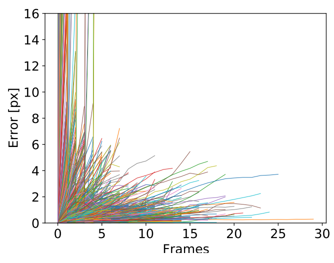


(i) SuperPoint Eros

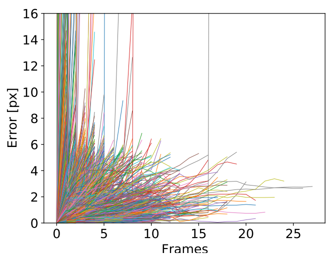


(j) SuperPoint Bennu

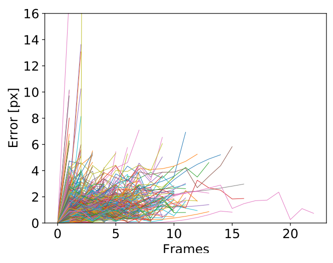
Figure E.23: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



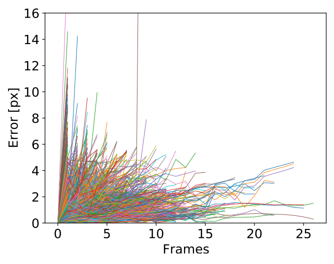
(a) SIFT



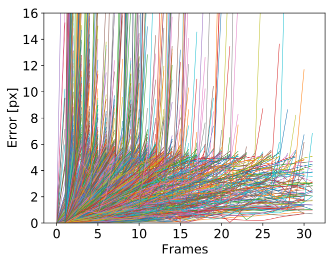
(b) SURF



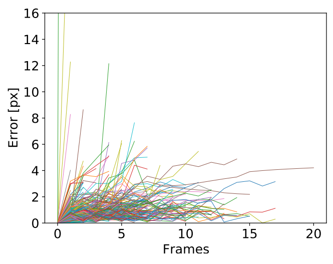
(c) ORB



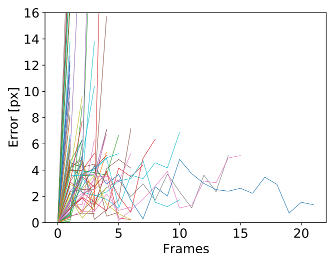
(d) BRISK



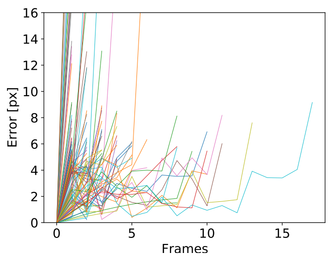
(e) KLT



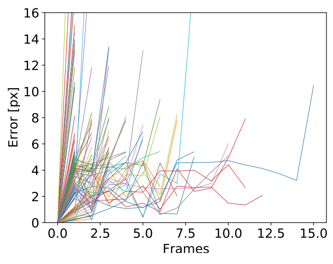
(f) STAR + BRIEF



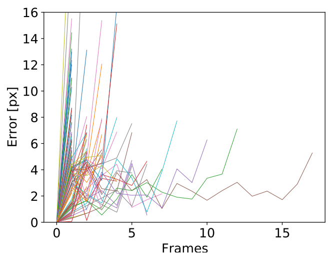
(g) SPV6



(h) SuperPoint Base



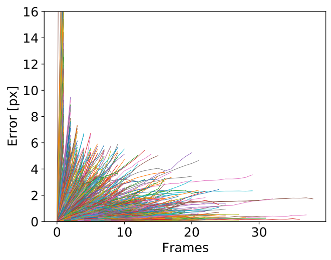
(i) SuperPoint Eros



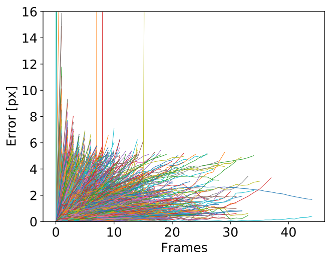
(j) SuperPoint Bennu

Figure E.24: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.

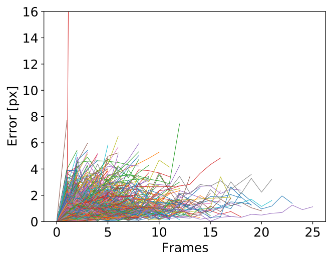
## E.3.3. Eros



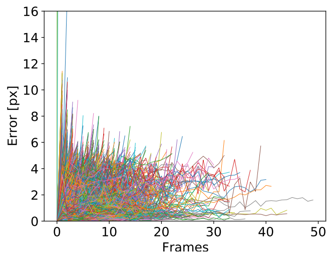
(a) SIFT



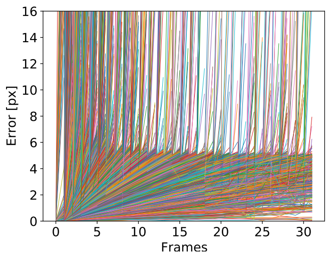
(b) SURF



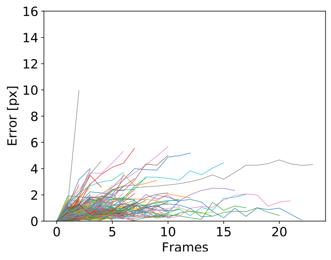
(c) ORB



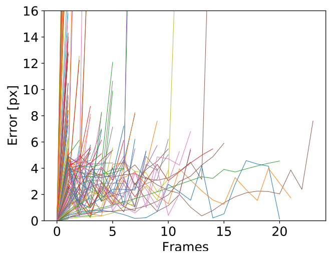
(d) BRISK



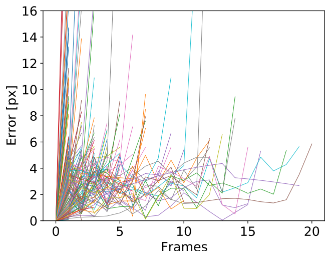
(e) KLT



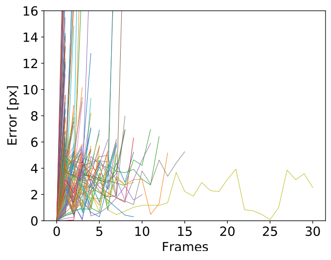
(f) STAR + BRIEF



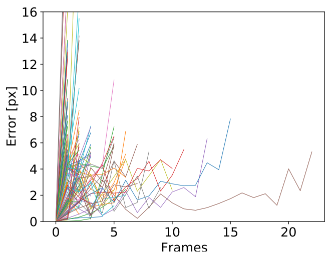
(g) SPV6



(h) SuperPoint Base

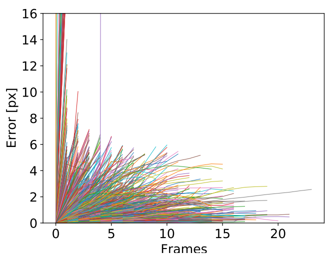


(i) SuperPoint Eros

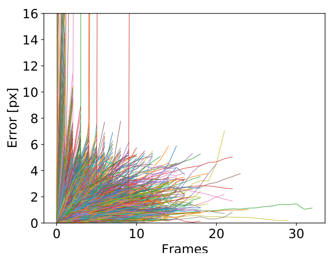


(j) SuperPoint Bennu

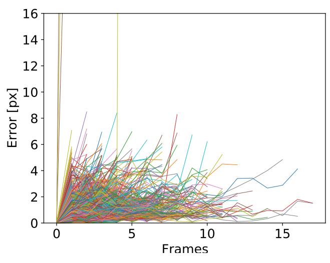
Figure E.25: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



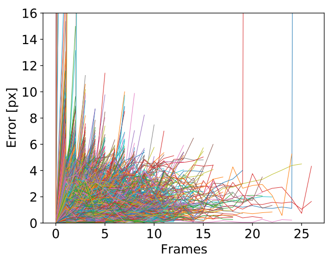
(a) SIFT



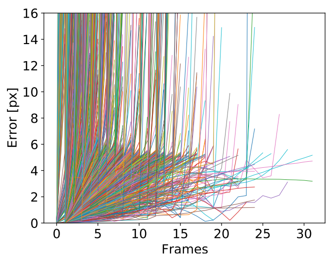
(b) SURF



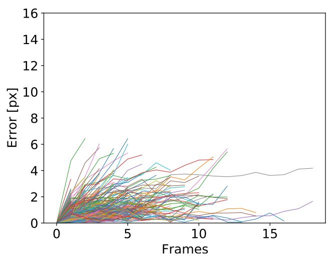
(c) ORB



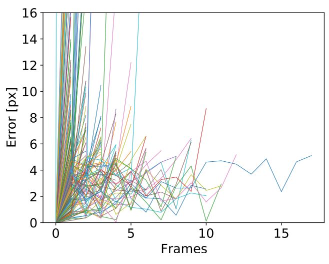
(d) BRISK



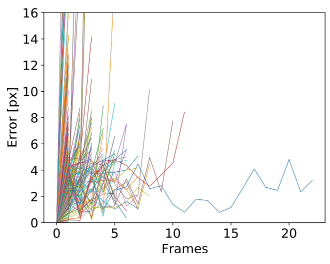
(e) KLT



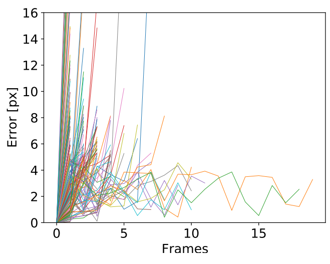
(f) STAR + BRIEF



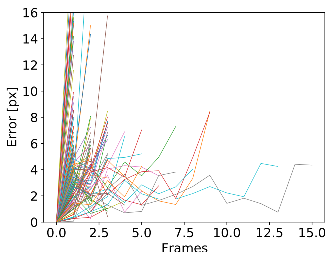
(g) SPV6



(h) SuperPoint Base

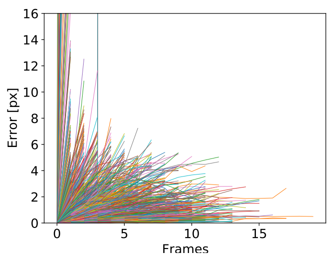


(i) SuperPoint Eros

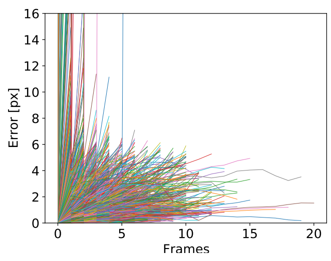


(j) SuperPoint Bennu

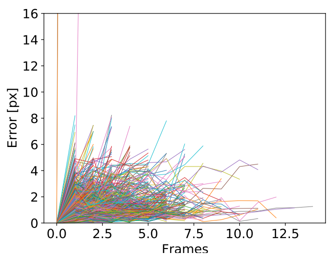
Figure E.26: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.



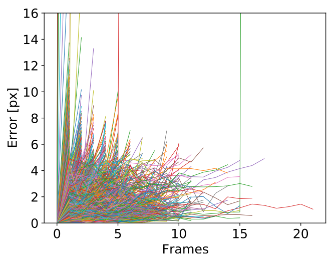
(a) SIFT



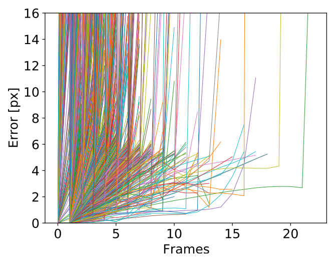
(b) SURF



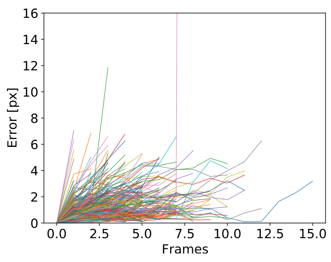
(c) ORB



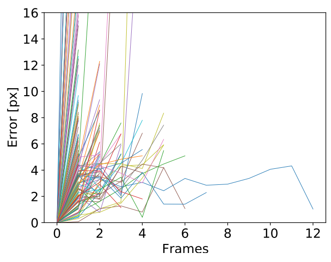
(d) BRISK



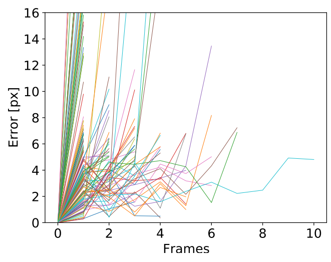
(e) KLT



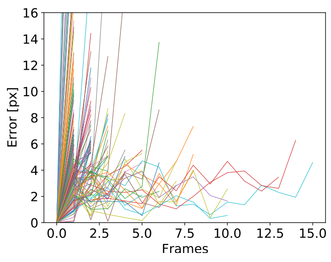
(f) STAR + BRIEF



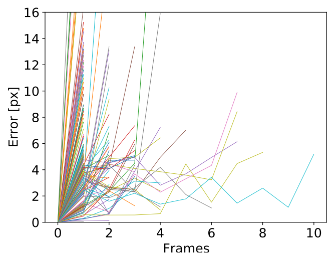
(g) SPV6



(h) SuperPoint Base

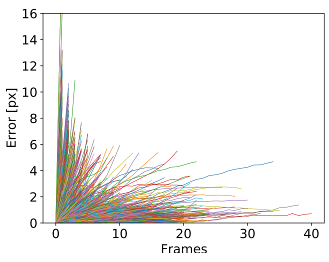


(i) SuperPoint Eros

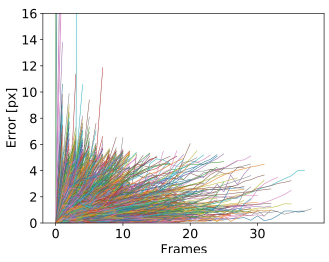


(j) SuperPoint Bennu

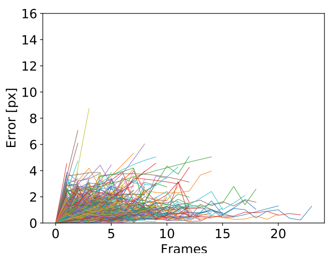
Figure E.27: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



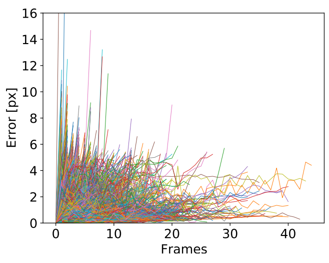
(a) SIFT



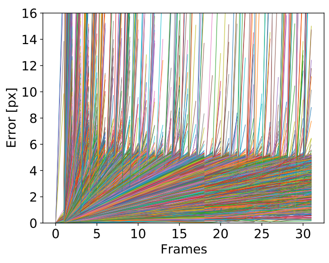
(b) SURF



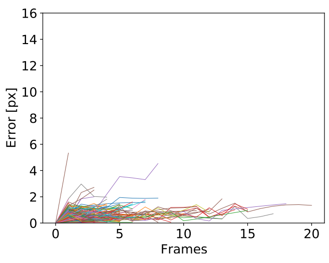
(c) ORB



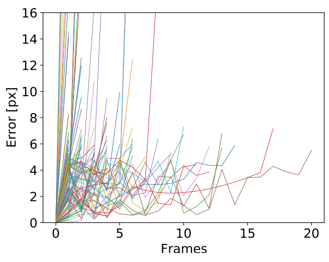
(d) BRISK



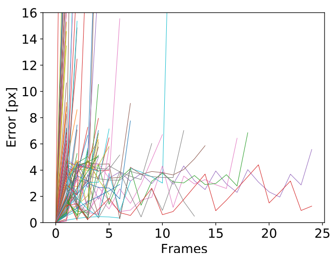
(e) KLT



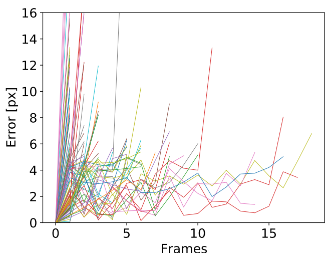
(f) STAR + BRIEF



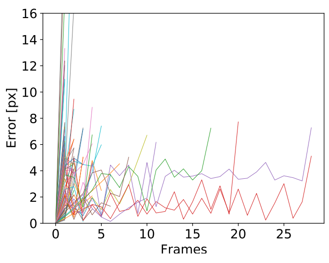
(g) SPV6



(h) SuperPoint Base

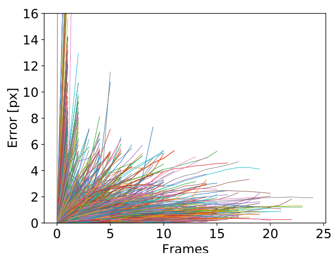


(i) SuperPoint Eros

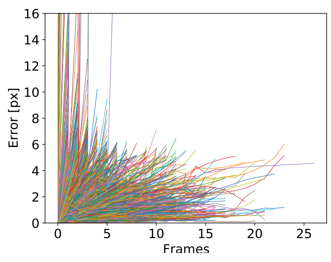


(j) SuperPoint Bennu

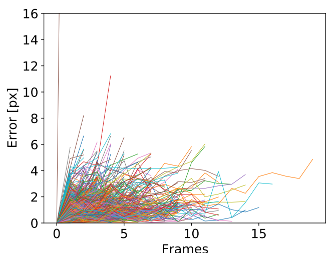
Figure E.28: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



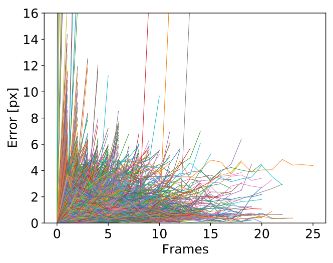
(a) SIFT



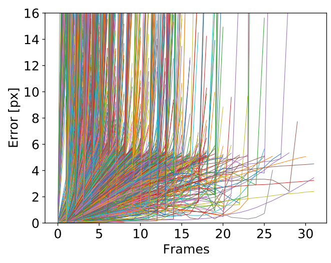
(b) SURF



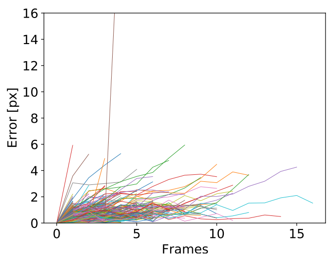
(c) ORB



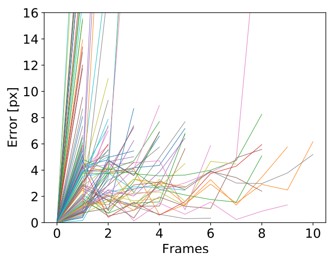
(d) BRISK



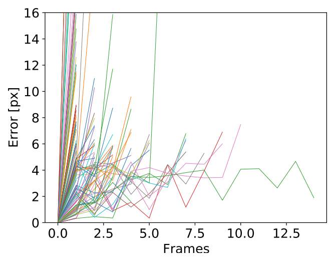
(e) KLT



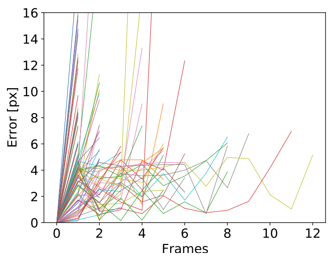
(f) STAR + BRIEF



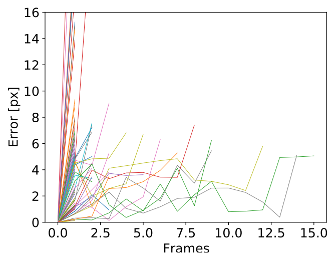
(g) SPV6



(h) SuperPoint Base

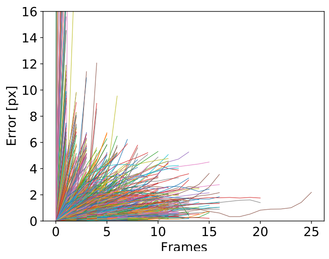


(i) SuperPoint Eros

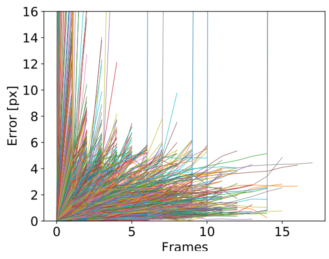


(j) SuperPoint Bennu

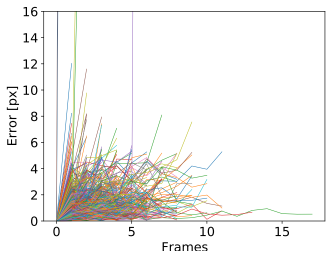
Figure E.29: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



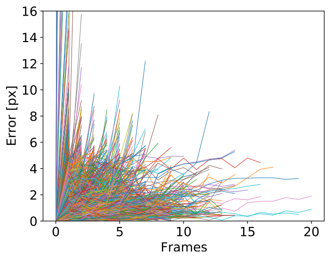
(a) SIFT



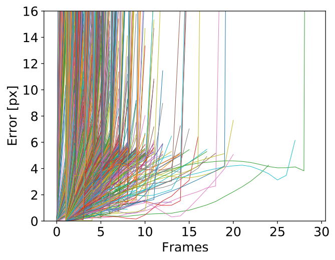
(b) SURF



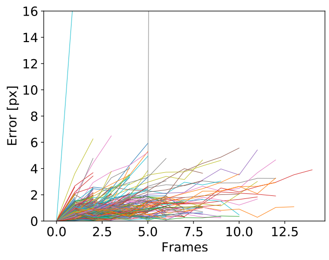
(c) ORB



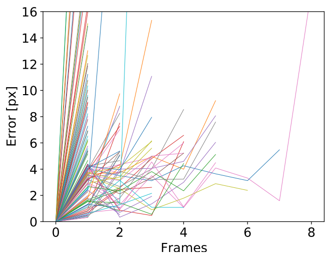
(d) BRISK



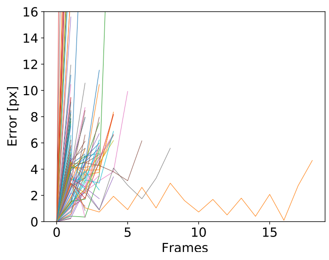
(e) KLT



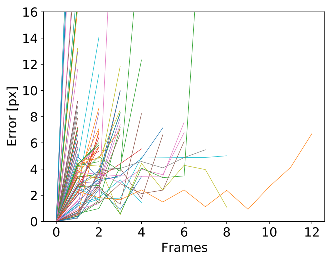
(f) STAR + BRIEF



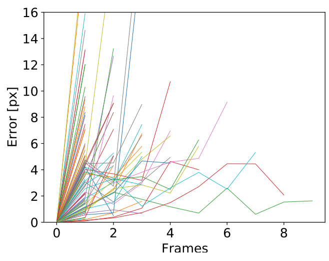
(g) SPV6



(h) SuperPoint Base

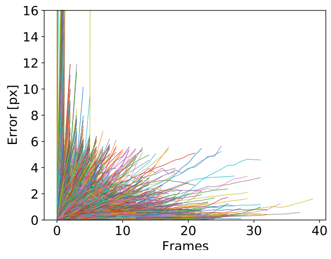


(i) SuperPoint Eros

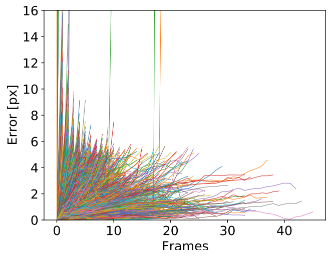


(j) SuperPoint Bennu

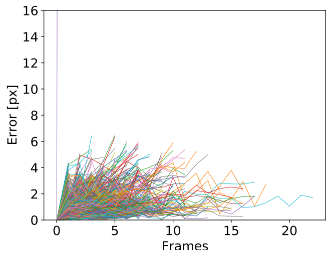
Figure E.30: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.



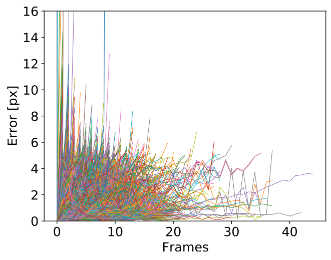
(a) SIFT



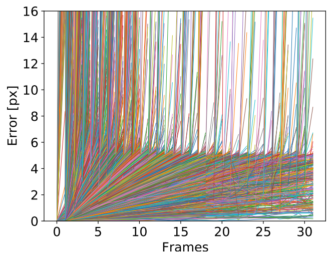
(b) SURF



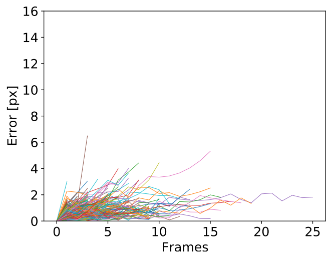
(c) ORB



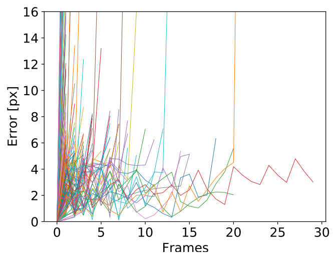
(d) BRISK



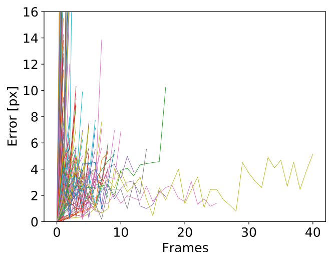
(e) KLT



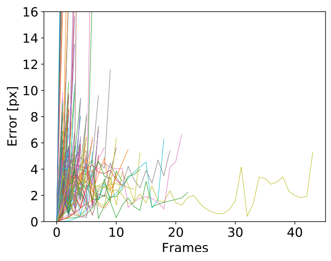
(f) STAR + BRIEF



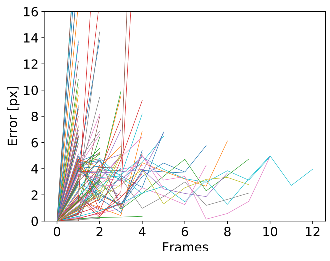
(g) SPV6



(h) SuperPoint Base

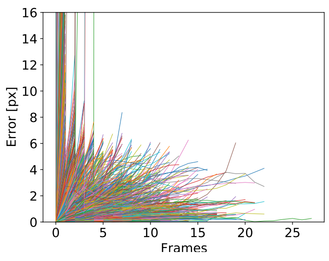


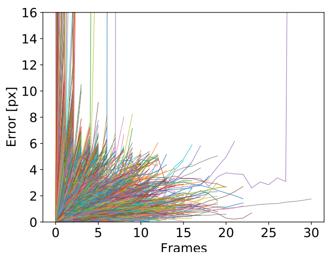
(i) SuperPoint Eros

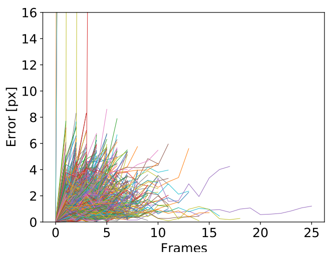


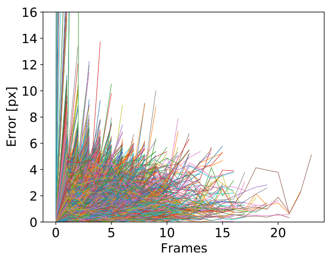
(j) SuperPoint Bennu

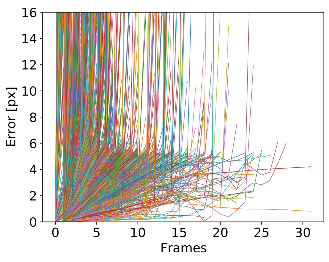
Figure E.31: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.

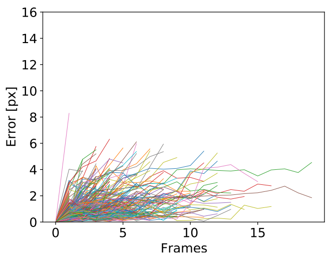

(a) SIFT

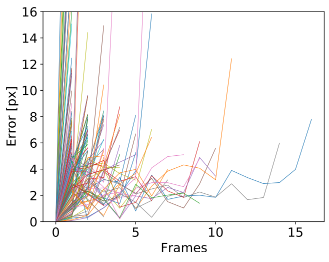

(b) SURF

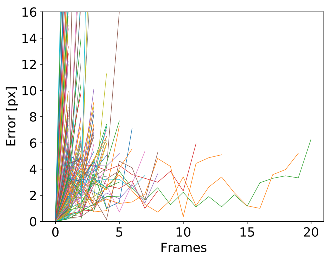

(c) ORB

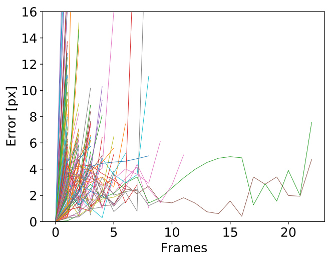

(d) BRISK


(e) KLT


(f) STAR + BRIEF


(g) SPV6


(h) SuperPoint Base


(i) SuperPoint Eros

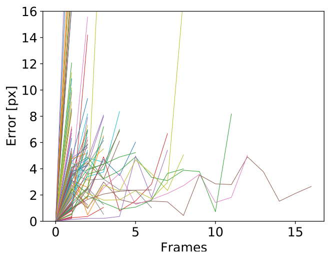
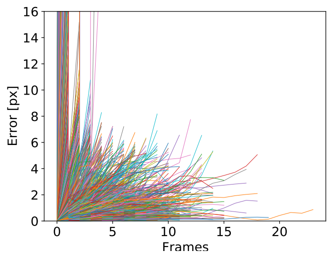
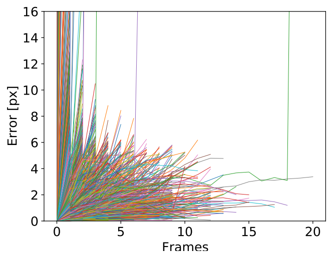
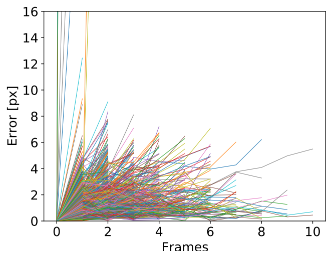
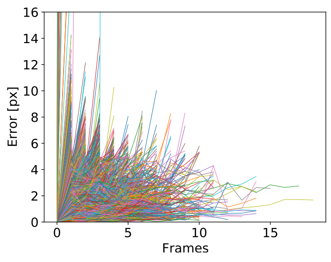

(j) SuperPoint Bennu

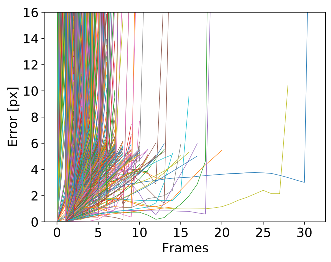
Figure E.32: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.

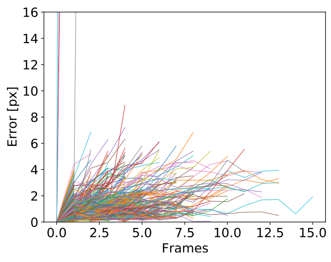

(a) SIFT

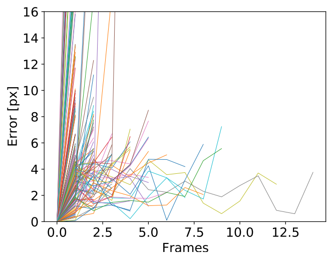

(b) SURF

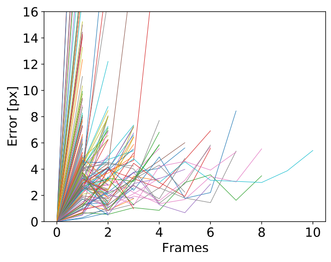

(c) ORB

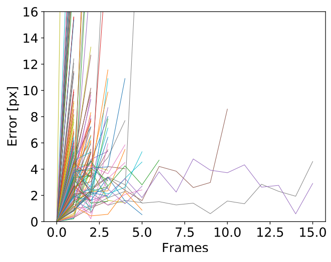

(d) BRISK


(e) KLT


(f) STAR + BRIEF


(g) SPV6


(h) SuperPoint Base


(i) SuperPoint Eros

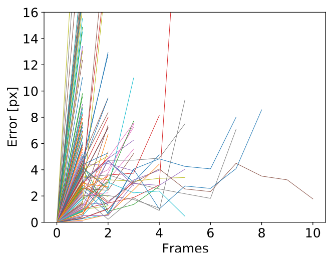
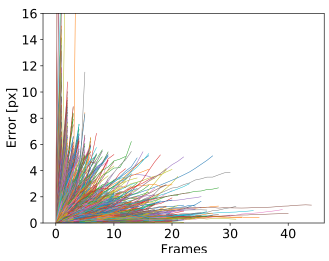
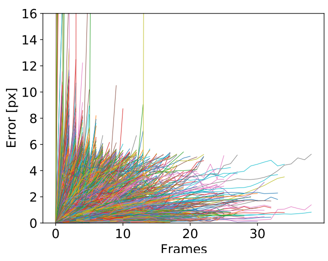

(j) SuperPoint Bennu

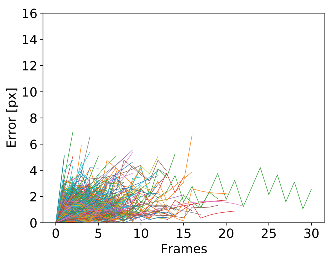
Figure E.33: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



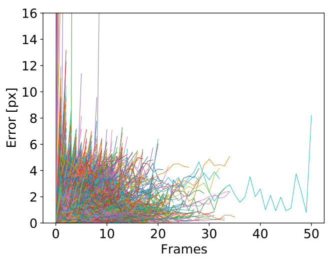
(a) SIFT



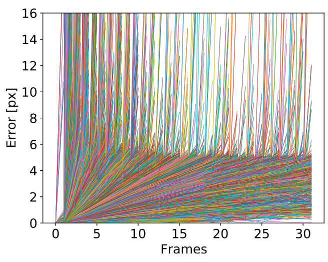
(b) SURF



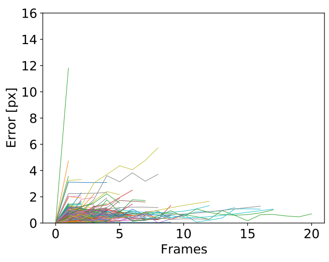
(c) ORB



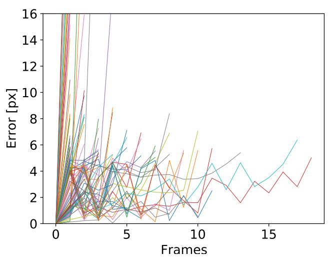
(d) BRISK



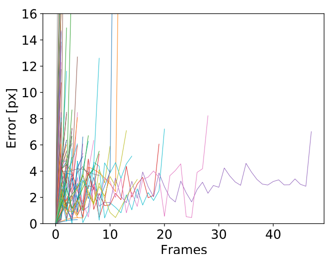
(e) KLT



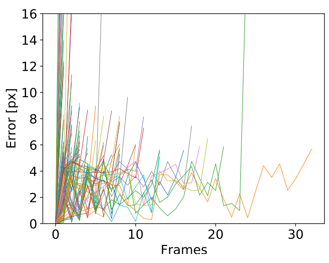
(f) STAR + BRIEF



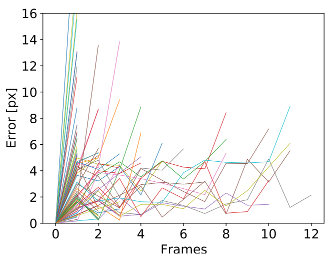
(g) SPV6



(h) SuperPoint Base

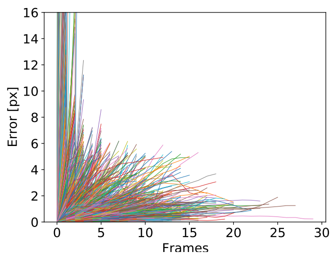


(i) SuperPoint Eros

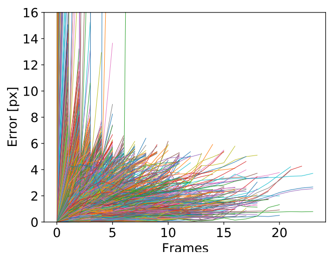


(j) SuperPoint Bennu

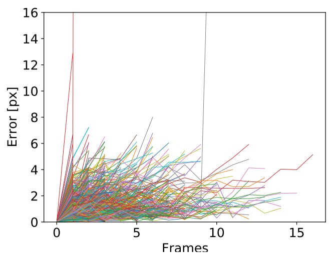
Figure E.34: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



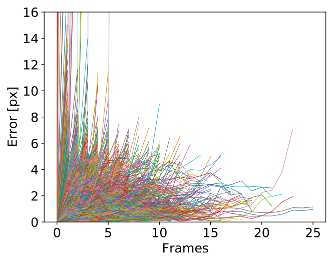
(a) SIFT



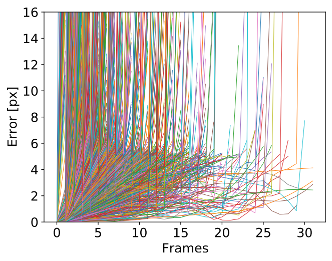
(b) SURF



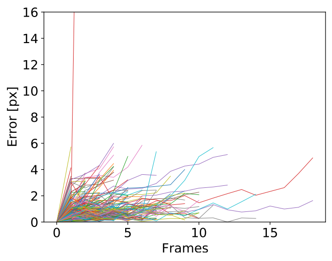
(c) ORB



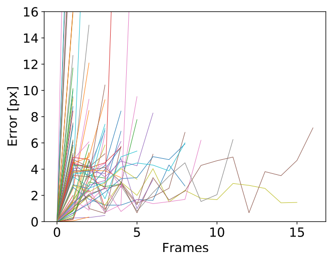
(d) BRISK



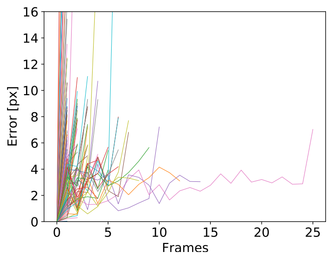
(e) KLT



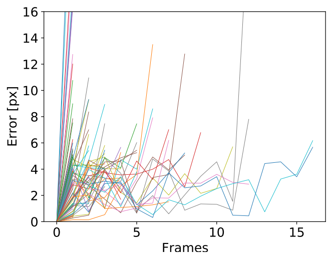
(f) STAR + BRIEF



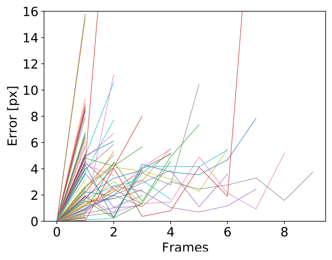
(g) SPV6



(h) SuperPoint Base

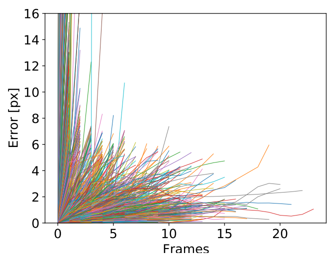


(i) SuperPoint Eros

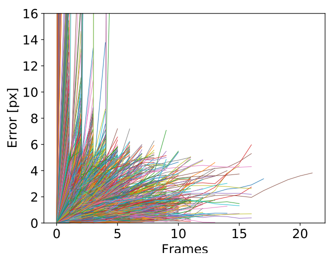


(j) SuperPoint Bennu

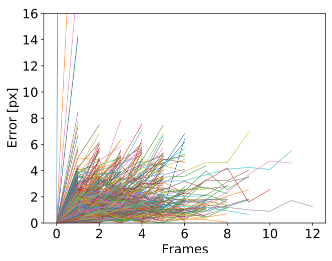
Figure E.35: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



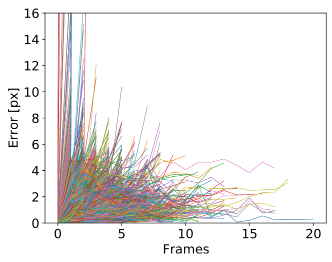
(a) SIFT



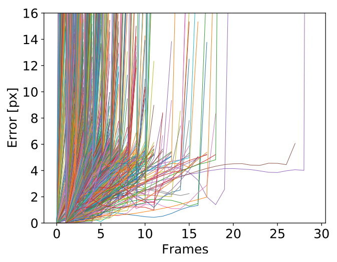
(b) SURF



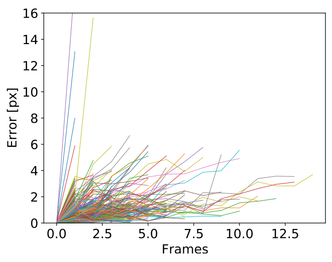
(c) ORB



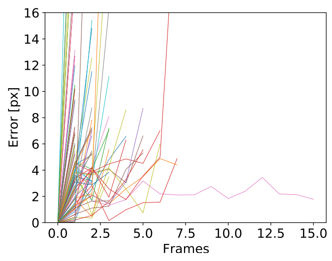
(d) BRISK



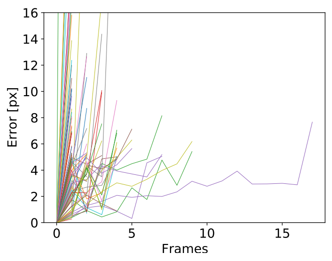
(e) KLT



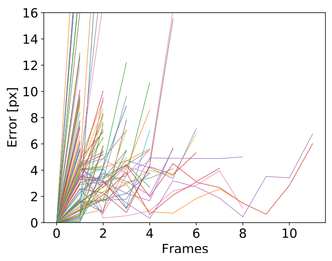
(f) STAR + BRIEF



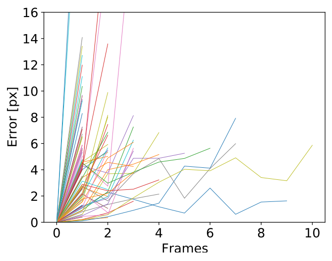
(g) SPV6



(h) SuperPoint Base



(i) SuperPoint Eros



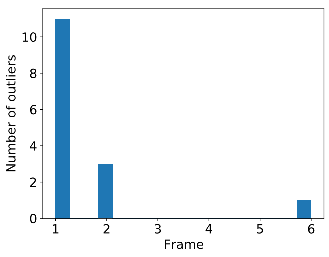
(j) SuperPoint Bennu

Figure E.36: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.

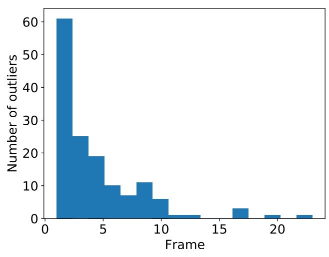
## E.4. Outliers histograms

Here are reported the histograms that show at which frames the points tend to become outliers.

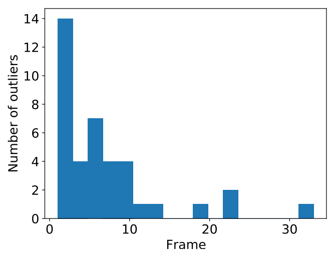
### E.4.1. Bennu



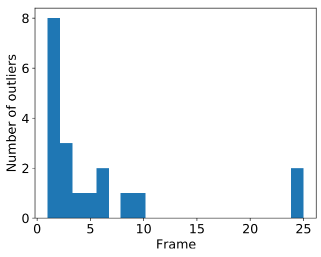
(a) SIFT



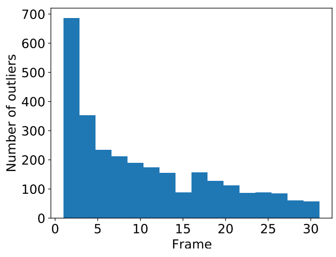
(b) SURF



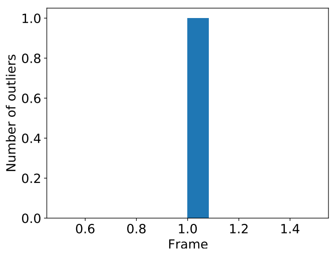
(c) ORB



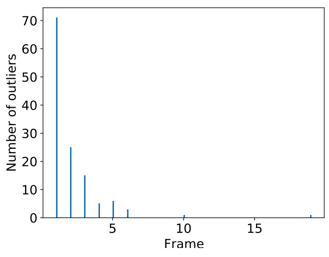
(d) BRISK



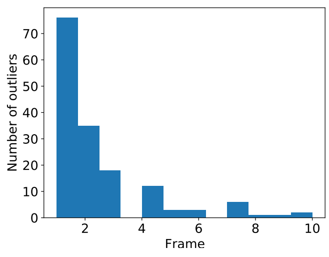
(e) KLT



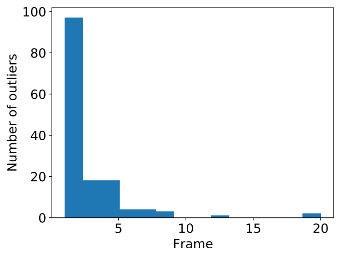
(f) STAR + BRIEF



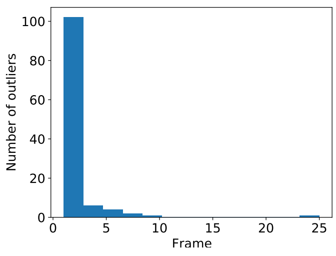
(g) SPV6



(h) SuperPoint Base

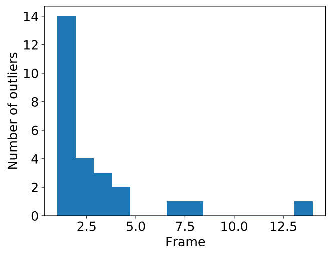


(i) SuperPoint Eros

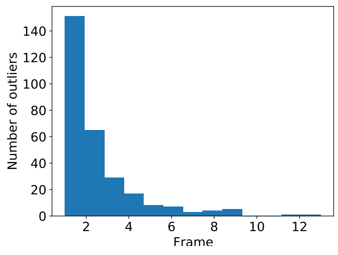


(j) SuperPoint Bennu

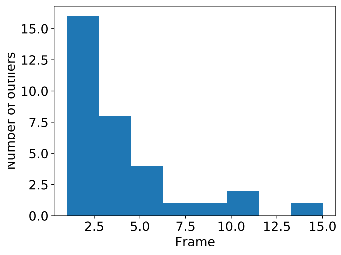
Figure E.37: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



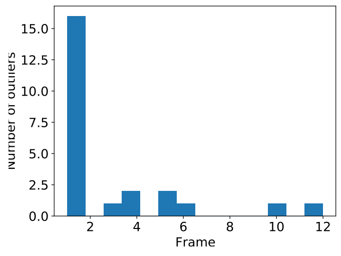
(a) SIFT



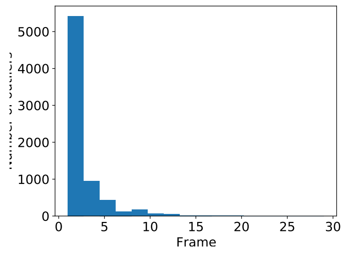
(b) SURF



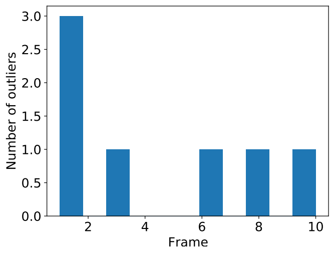
(c) ORB



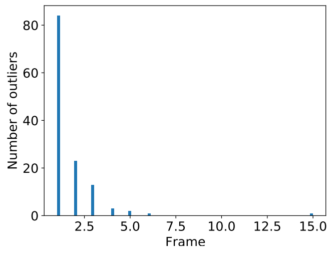
(d) BRISK



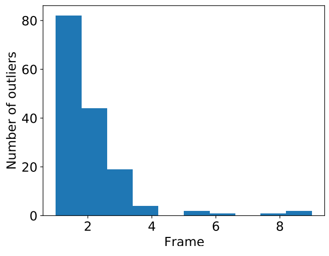
(e) KLT



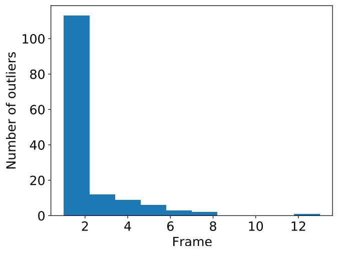
(f) STAR + BRIEF



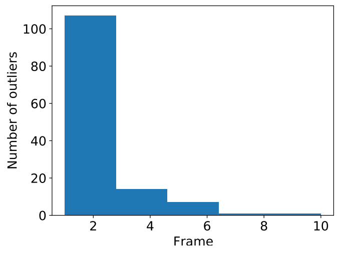
(g) SPV6



(h) SuperPoint Base

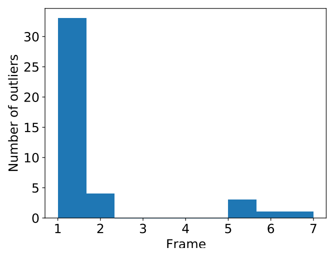


(i) SuperPoint Eros

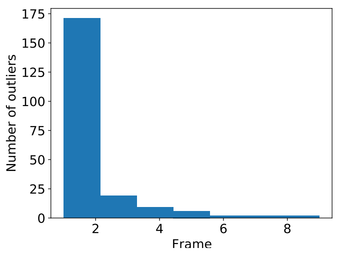


(j) SuperPoint Bennu

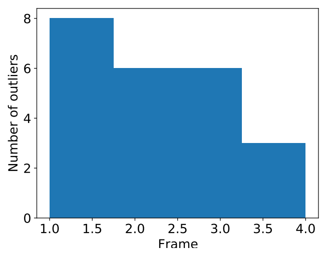
Figure E.38: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.



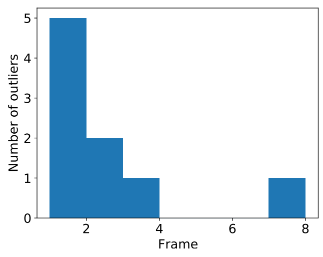
(a) SIFT



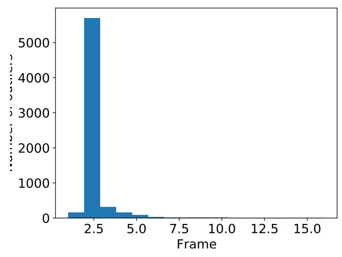
(b) SURF



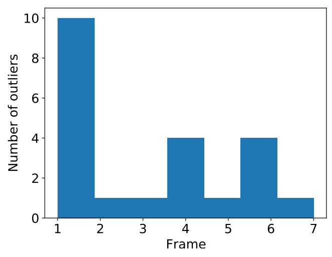
(c) ORB



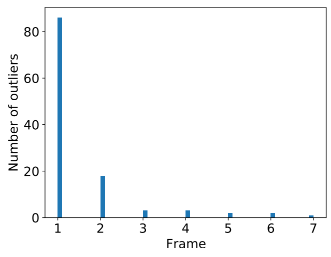
(d) BRISK



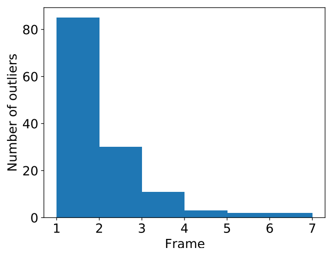
(e) KLT



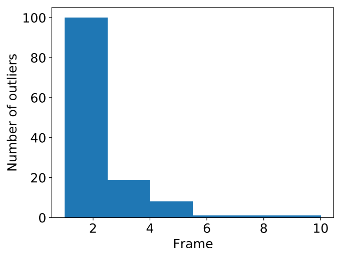
(f) STAR + BRIEF



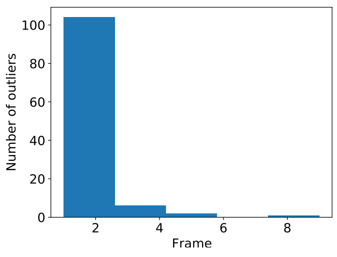
(g) SPV6



(h) SuperPoint Base

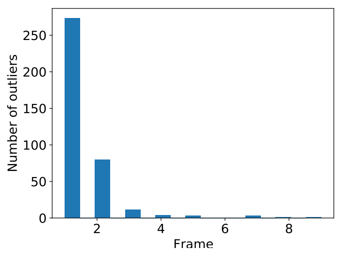


(i) SuperPoint Eros

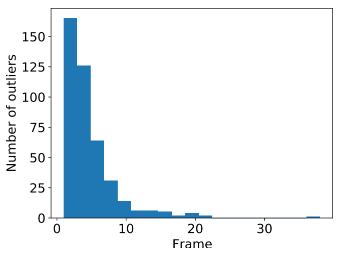


(j) SuperPoint Bennu

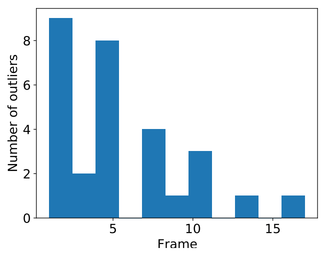
Figure E.39: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



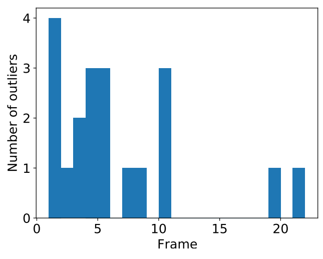
(a) SIFT



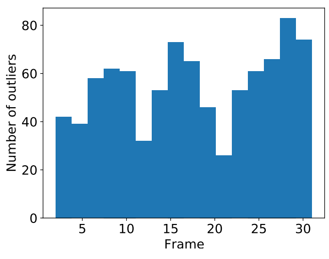
(b) SURF



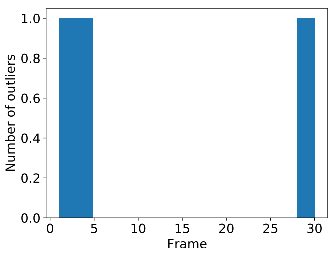
(c) ORB



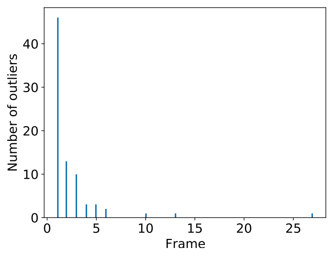
(d) BRISK



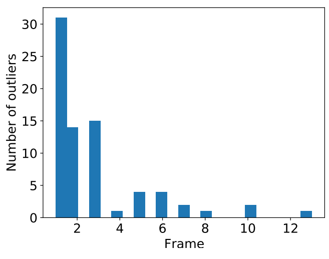
(e) KLT



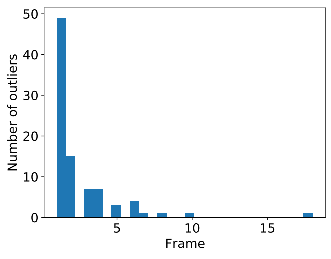
(f) STAR + BRIEF



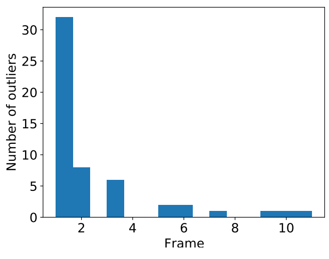
(g) SPV6



(h) SuperPoint Base

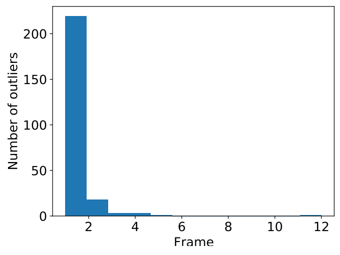


(i) SuperPoint Eros

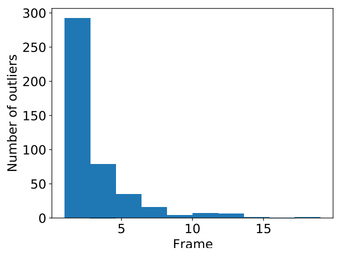


(j) SuperPoint Bennu

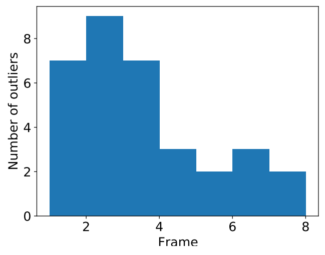
Figure E.40: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



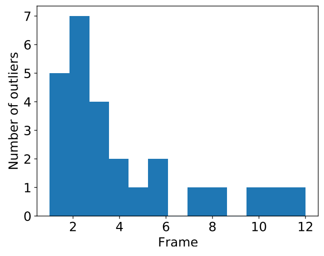
(a) SIFT



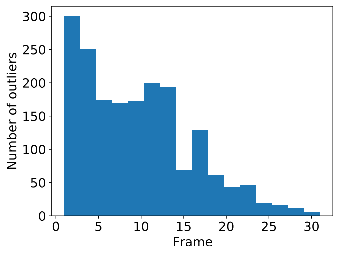
(b) SURF



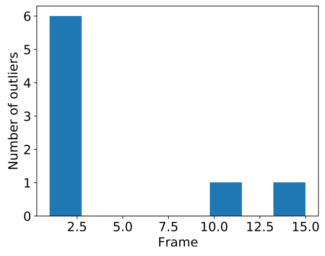
(c) ORB



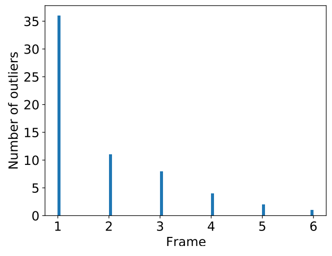
(d) BRISK



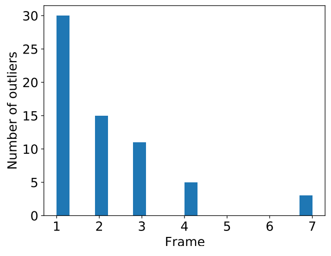
(e) KLT



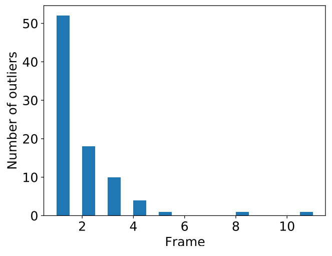
(f) STAR + BRIEF



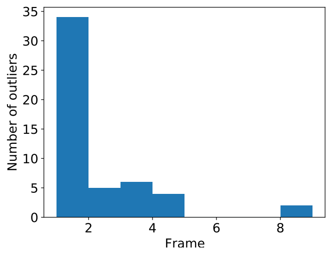
(g) SPV6



(h) SuperPoint Base

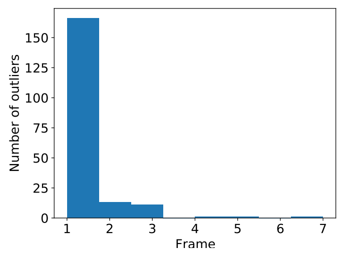


(i) SuperPoint Eros

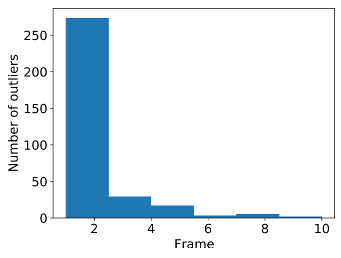


(j) SuperPoint Bennu

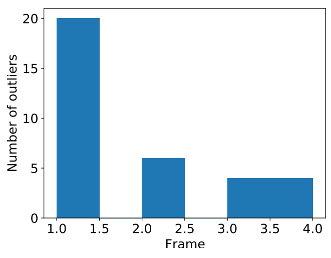
Figure E.41: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



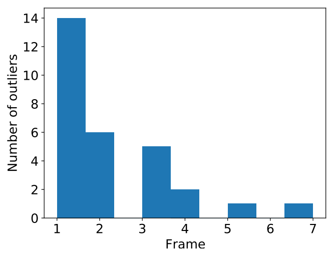
(a) SIFT



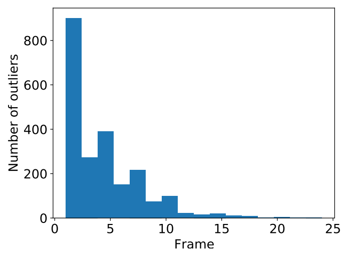
(b) SURF



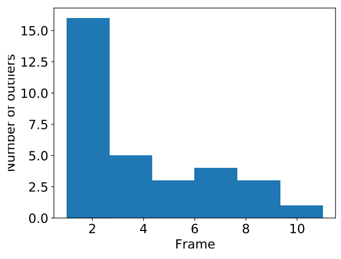
(c) ORB



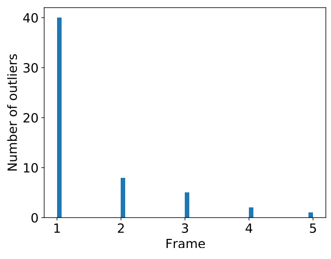
(d) BRISK



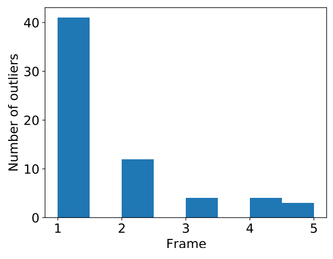
(e) KLT



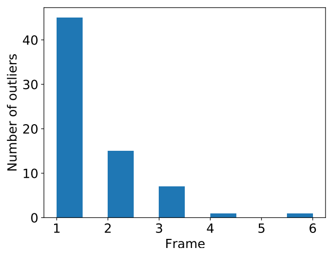
(f) STAR + BRIEF



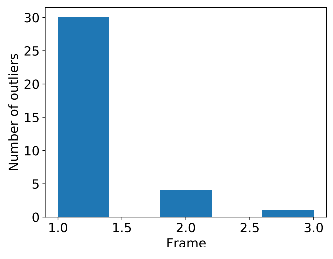
(g) SPV6



(h) SuperPoint Base

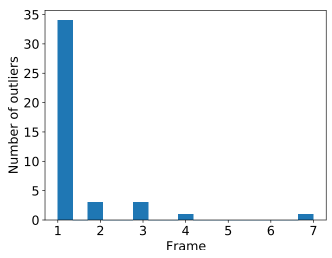


(i) SuperPoint Eros

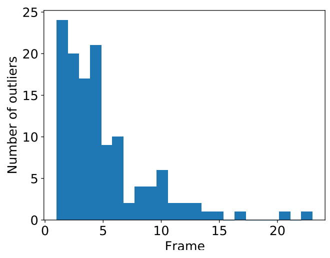


(j) SuperPoint Bennu

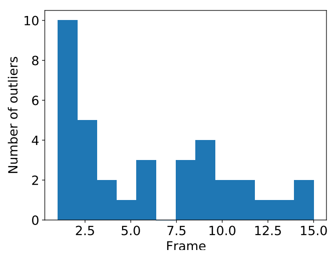
Figure E.42: Asteroid: Bennu. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.



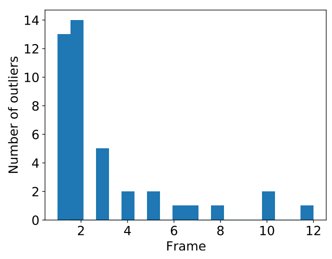
(a) SIFT



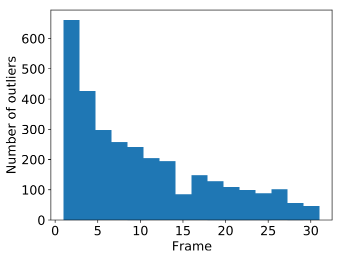
(b) SURF



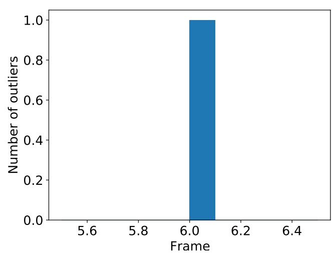
(c) ORB



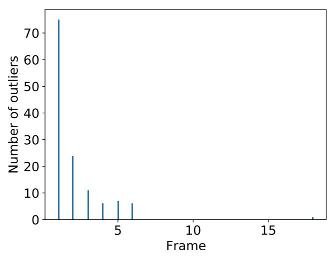
(d) BRISK



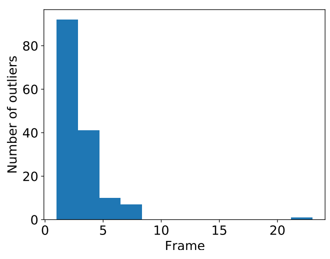
(e) KLT



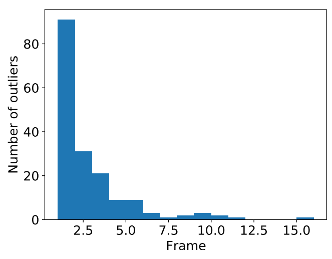
(f) STAR + BRIEF



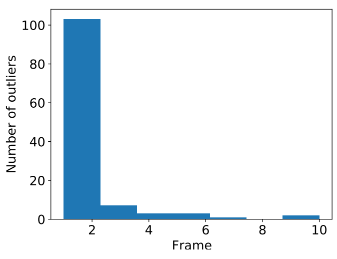
(g) SPV6



(h) SuperPoint Base

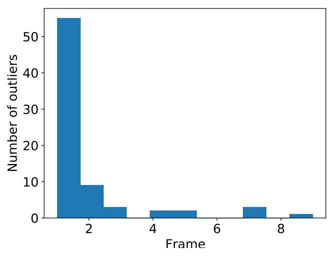


(i) SuperPoint Eros

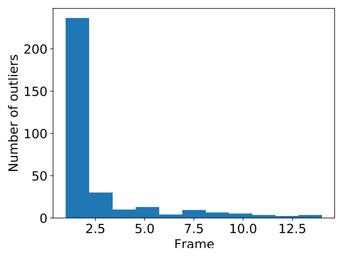


(j) SuperPoint Bennu

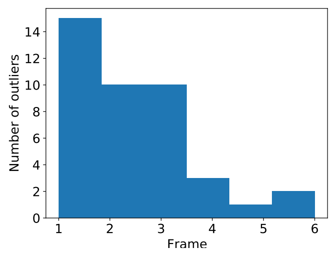
Figure E.43: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



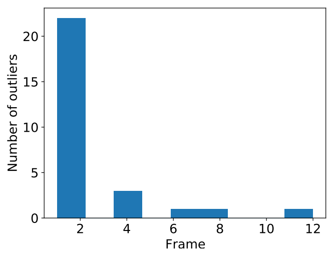
(a) SIFT



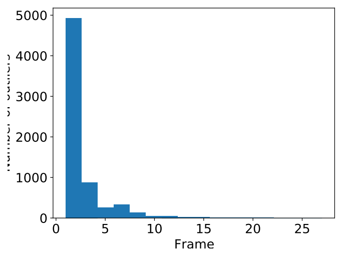
(b) SURF



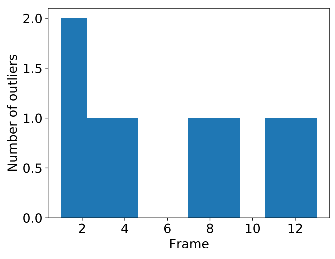
(c) ORB



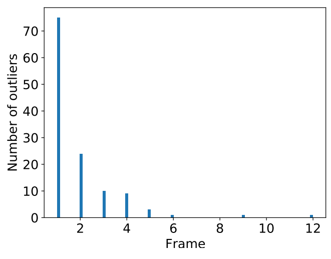
(d) BRISK



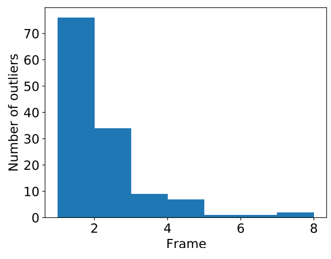
(e) KLT



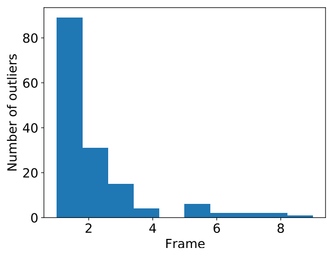
(f) STAR + BRIEF



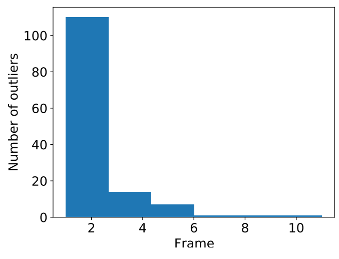
(g) SPV6



(h) SuperPoint Base

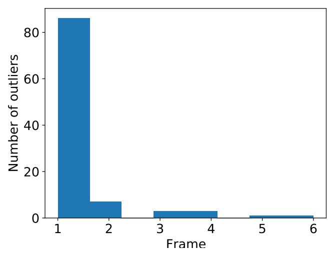


(i) SuperPoint Eros

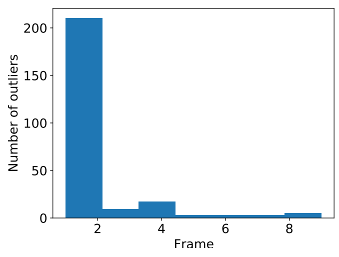


(j) SuperPoint Bennu

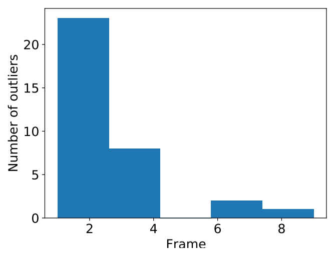
Figure E.44: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.



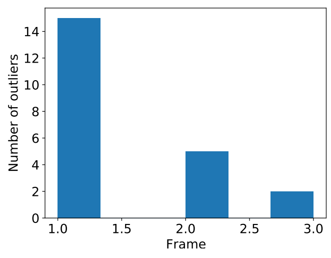
(a) SIFT



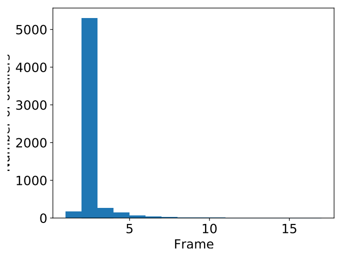
(b) SURF



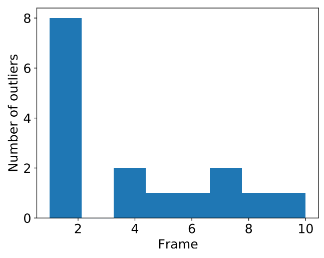
(c) ORB



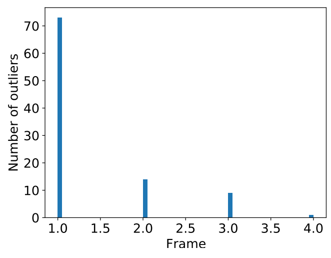
(d) BRISK



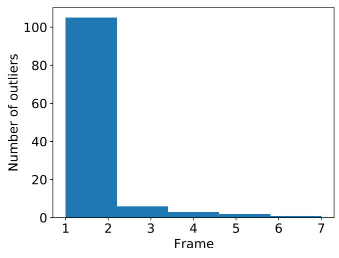
(e) KLT



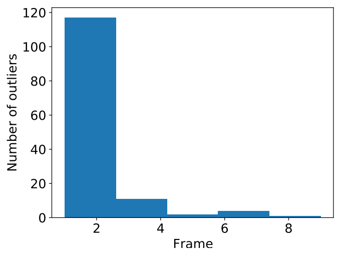
(f) STAR + BRIEF



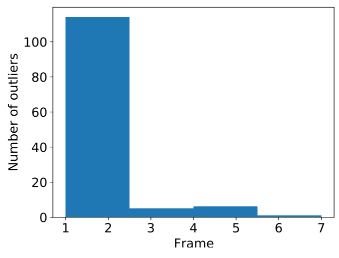
(g) SPV6



(h) SuperPoint Base

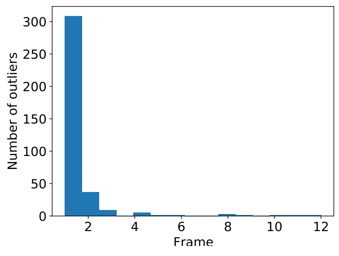


(i) SuperPoint Eros

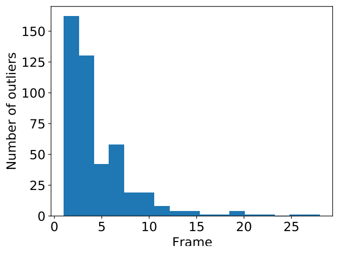


(j) SuperPoint Bennu

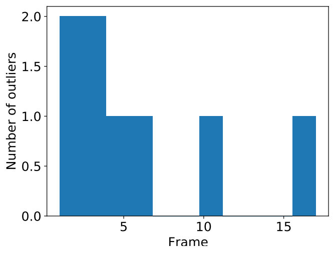
Figure E.45: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



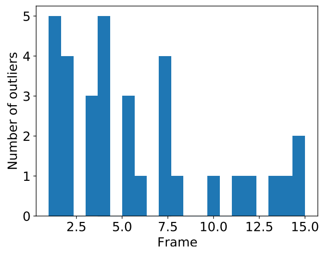
(a) SIFT



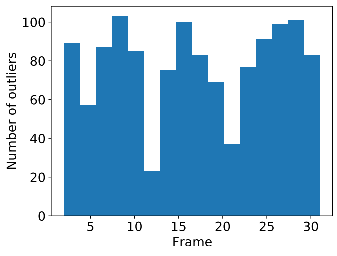
(b) SURF



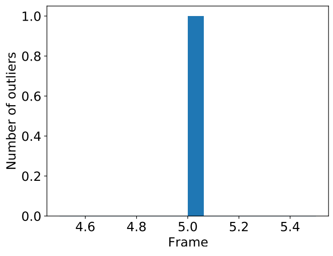
(c) ORB



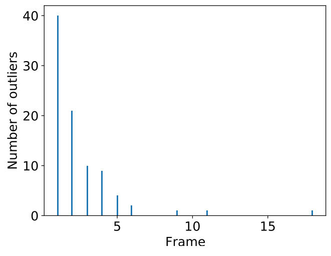
(d) BRISK



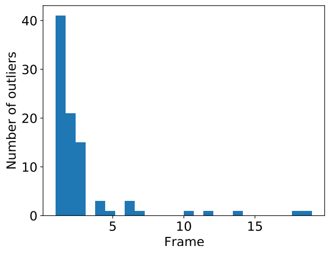
(e) KLT



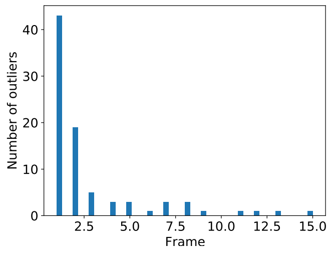
(f) STAR + BRIEF



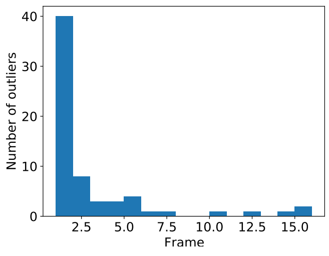
(g) SPV6



(h) SuperPoint Base

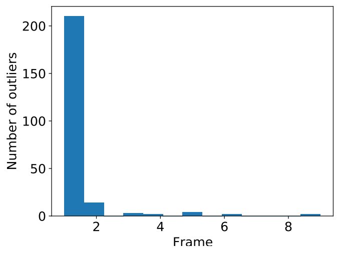


(i) SuperPoint Eros

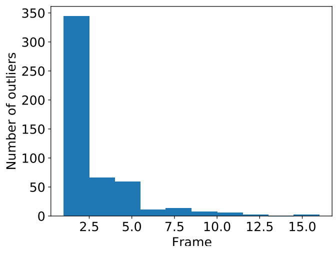


(j) SuperPoint Bennu

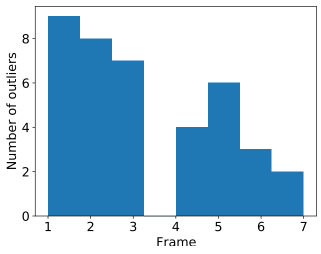
Figure E.46: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



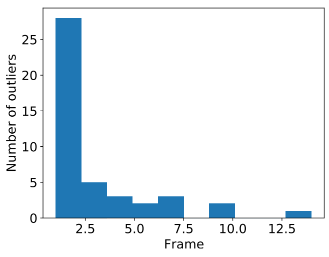
(a) SIFT



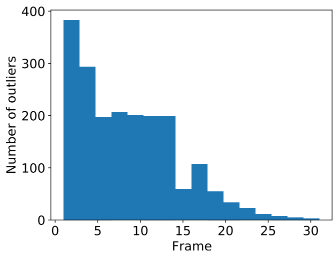
(b) SURF



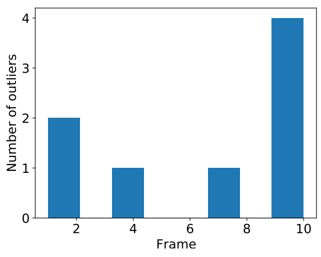
(c) ORB



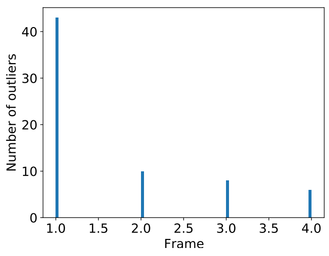
(d) BRISK



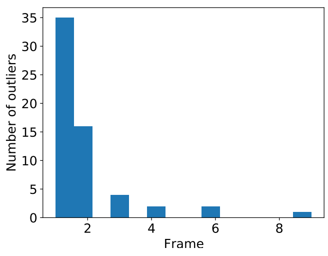
(e) KLT



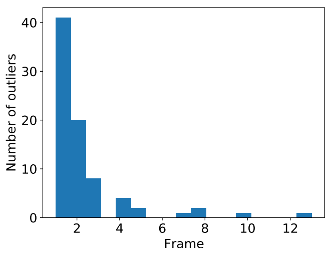
(f) STAR + BRIEF



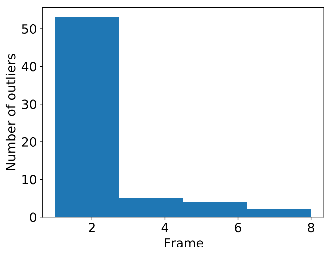
(g) SPV6



(h) SuperPoint Base

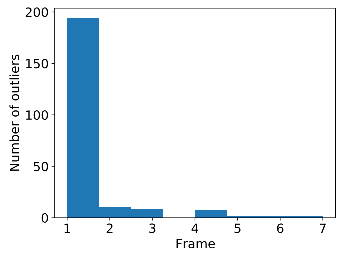


(i) SuperPoint Eros

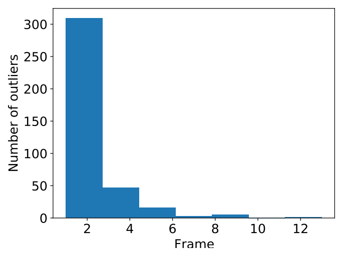


(j) SuperPoint Bennu

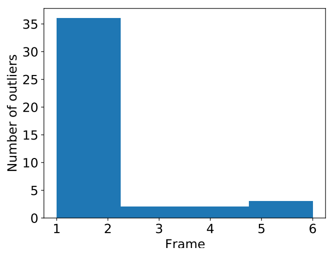
Figure E.47: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



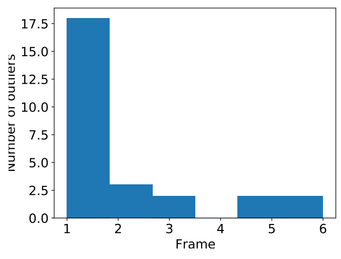
(a) SIFT



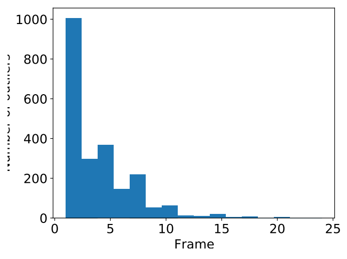
(b) SURF



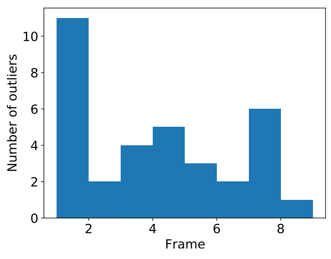
(c) ORB



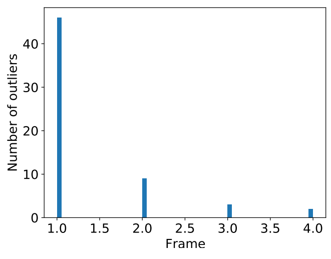
(d) BRISK



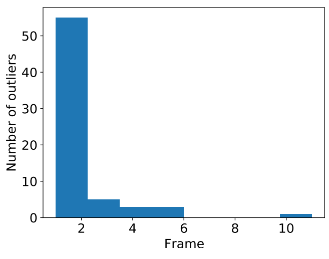
(e) KLT



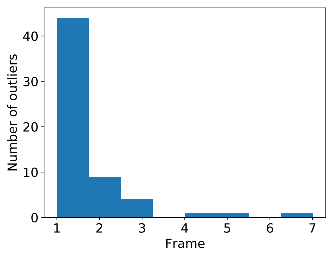
(f) STAR + BRIEF



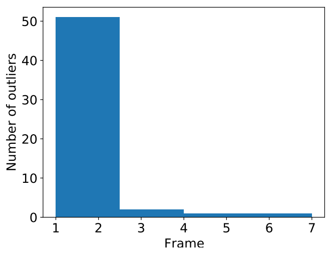
(g) SPV6



(h) SuperPoint Base



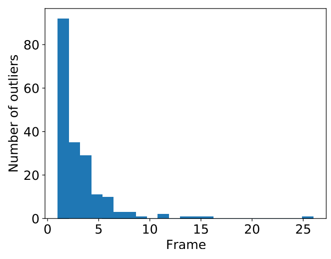
(i) SuperPoint Eros



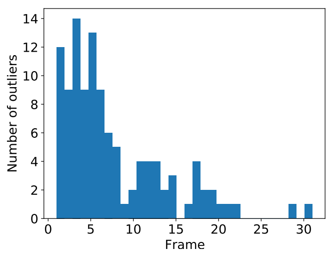
(j) SuperPoint Bennu

Figure E.48: Asteroid: Bennu. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.

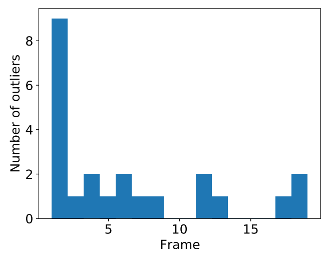
### E.4.2. Itokawa



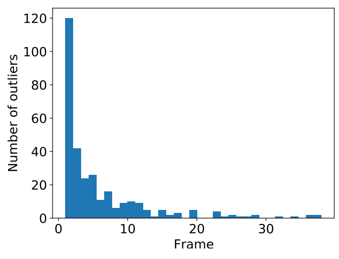
(a) SIFT



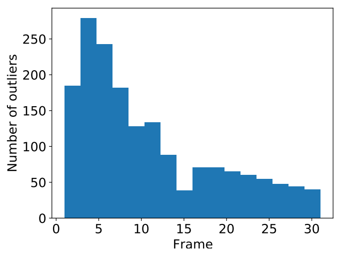
(b) SURF



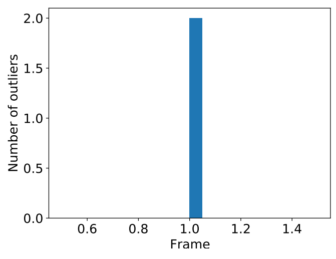
(c) ORB



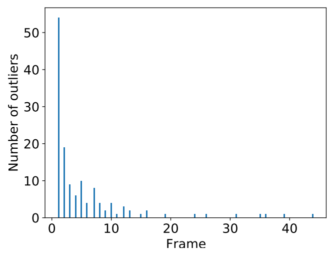
(d) BRISK



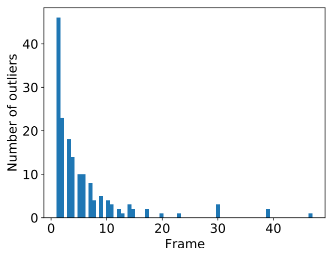
(e) KLT



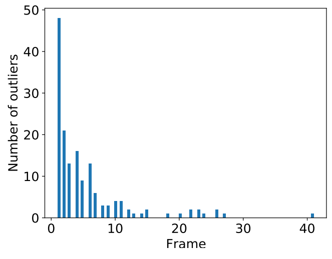
(f) STAR + BRIEF



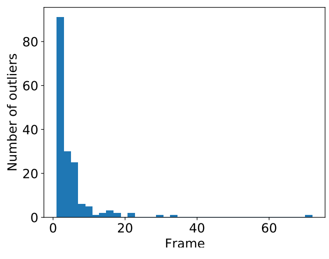
(g) SPV6



(h) SuperPoint Base

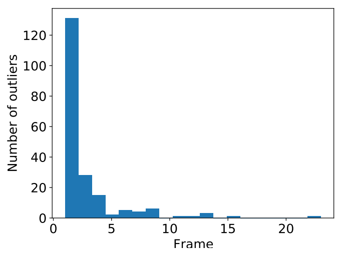


(i) SuperPoint Eros

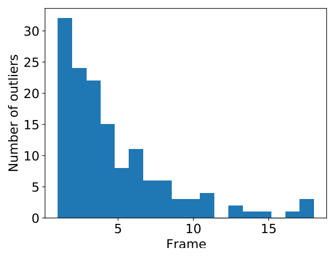


(j) SuperPoint Bennu

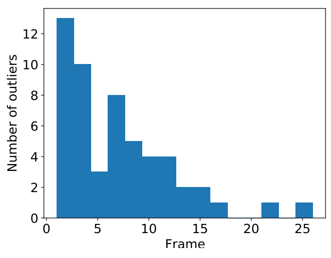
Figure E.49: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



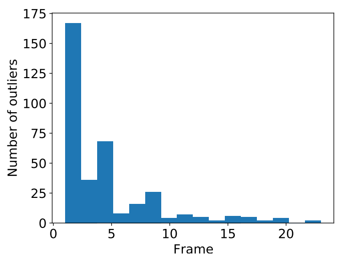
(a) SIFT



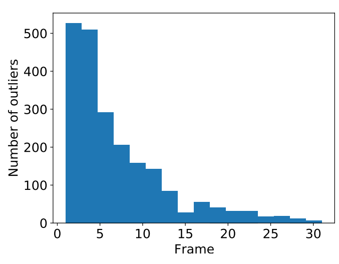
(b) SURF



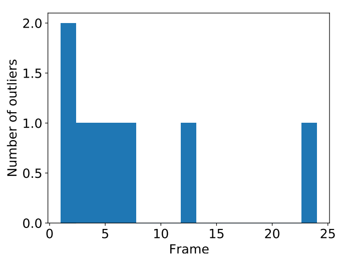
(c) ORB



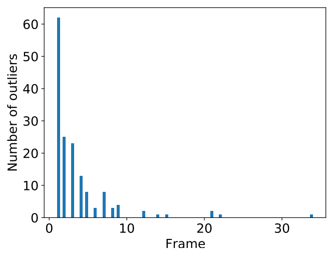
(d) BRISK



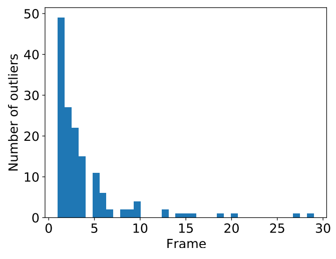
(e) KLT



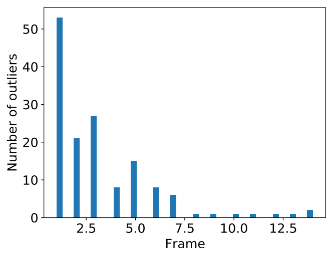
(f) STAR + BRIEF



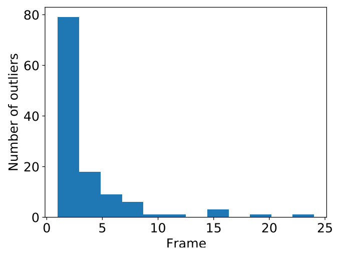
(g) SPV6



(h) SuperPoint Base

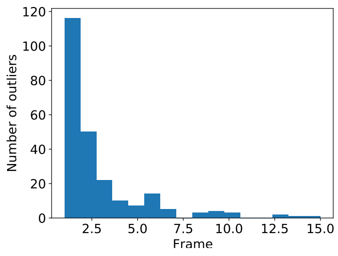


(i) SuperPoint Eros

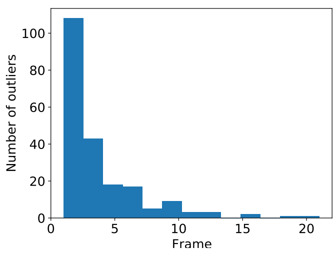


(j) SuperPoint Bennu

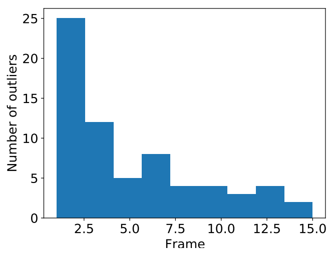
Figure E.50: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.



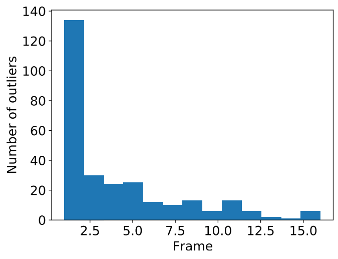
(a) SIFT



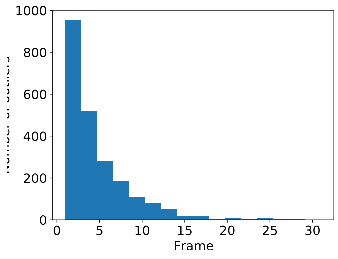
(b) SURF



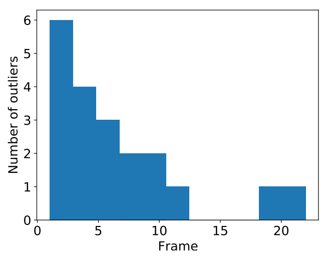
(c) ORB



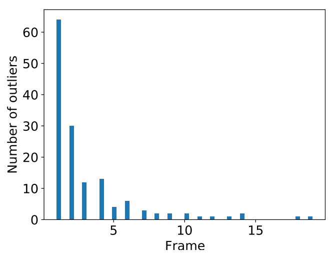
(d) BRISK



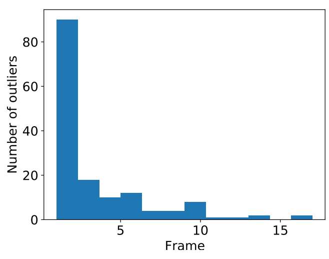
(e) KLT



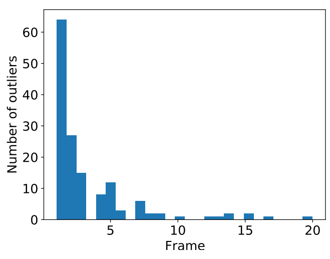
(f) STAR + BRIEF



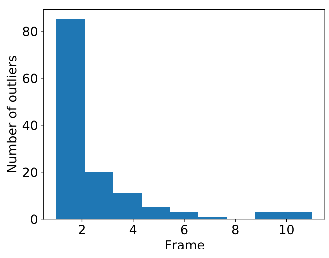
(g) SPV6



(h) SuperPoint Base

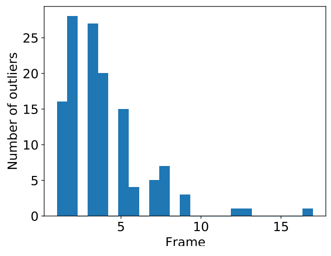


(i) SuperPoint Eros

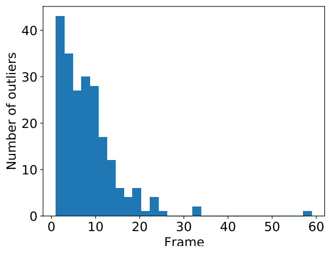


(j) SuperPoint Bennu

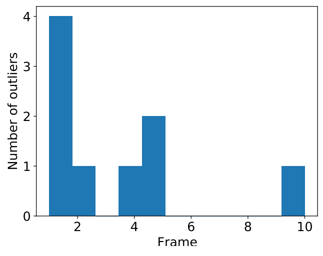
Figure E.51: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



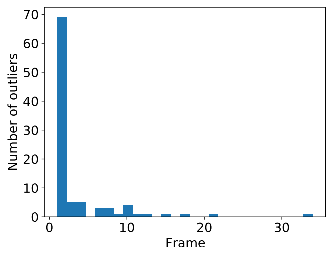
(a) SIFT



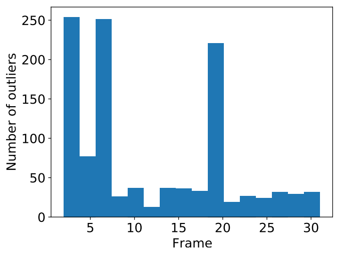
(b) SURF



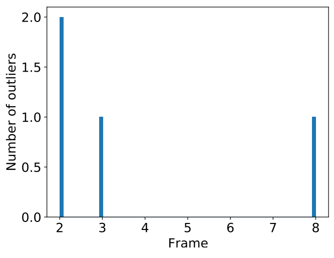
(c) ORB



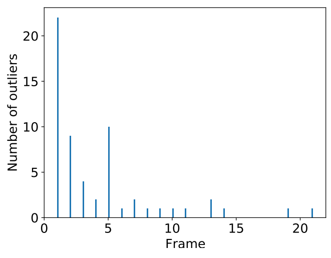
(d) BRISK



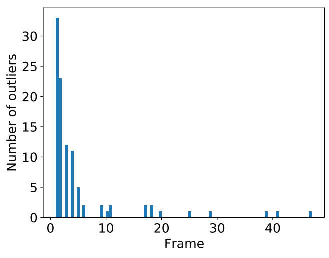
(e) KLT



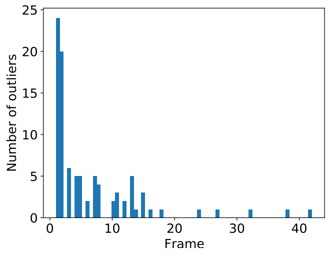
(f) STAR + BRIEF



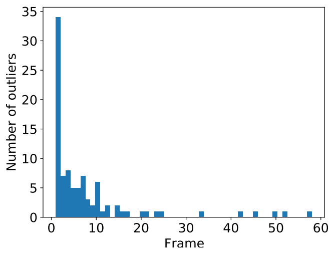
(g) SPV6



(h) SuperPoint Base

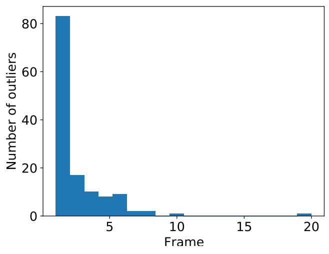


(i) SuperPoint Eros

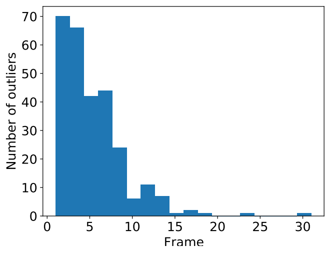


(j) SuperPoint Bennu

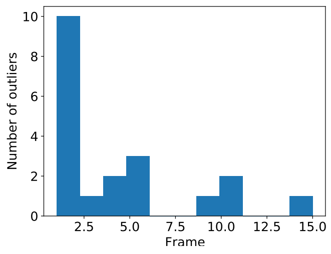
Figure E.52: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



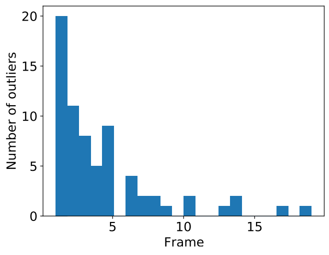
(a) SIFT



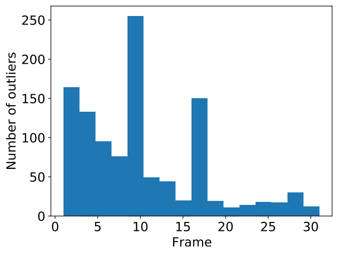
(b) SURF



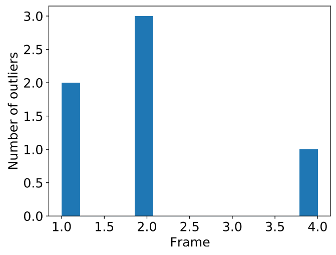
(c) ORB



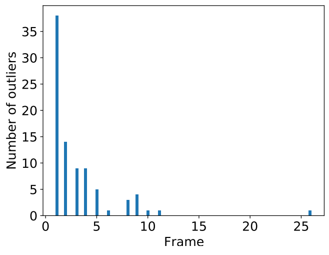
(d) BRISK



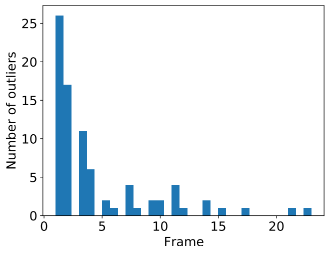
(e) KLT



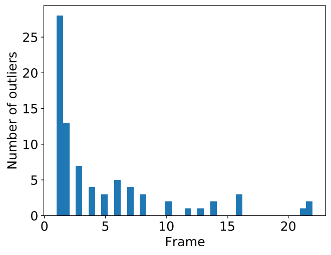
(f) STAR + BRIEF



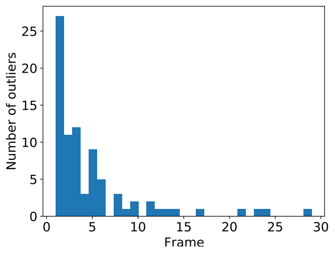
(g) SPV6



(h) SuperPoint Base

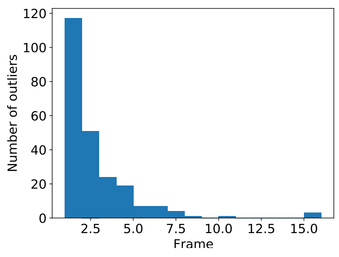


(i) SuperPoint Eros

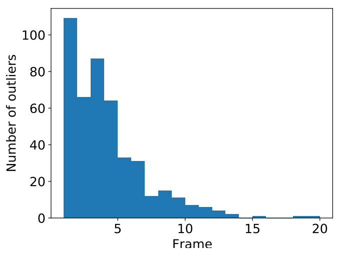


(j) SuperPoint Bennu

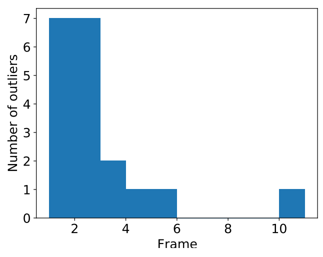
Figure E.53: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



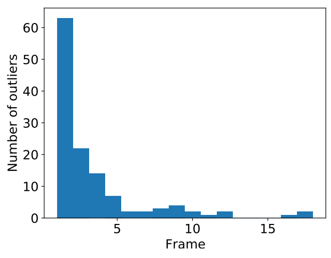
(a) SIFT



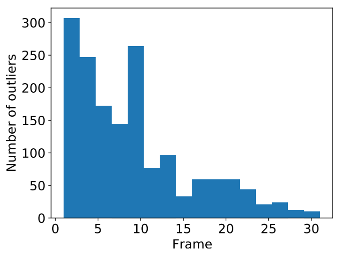
(b) SURF



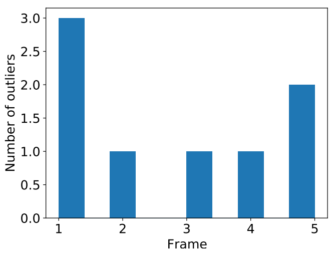
(c) ORB



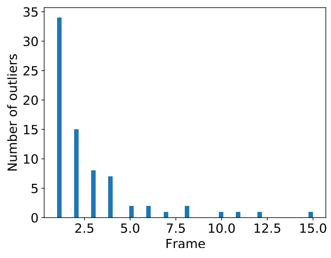
(d) BRISK



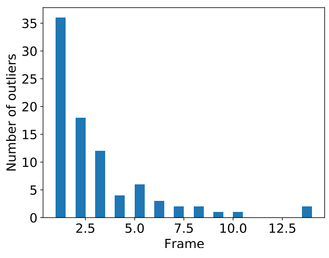
(e) KLT



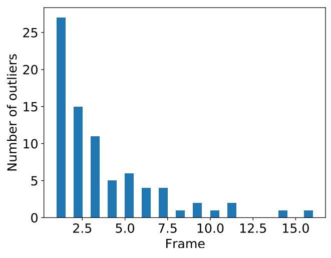
(f) STAR + BRIEF



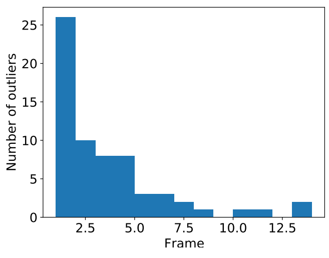
(g) SPV6



(h) SuperPoint Base

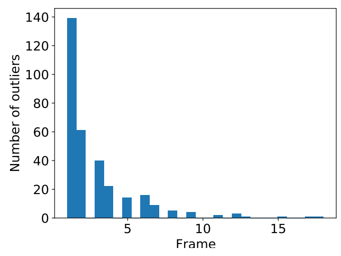


(i) SuperPoint Eros

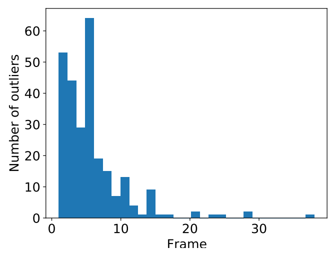


(j) SuperPoint Bennu

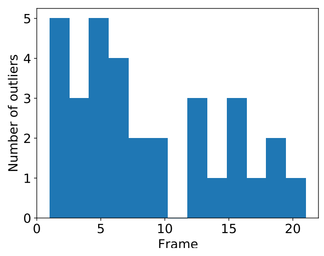
Figure E.54: Asteroid: Itokawa. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.



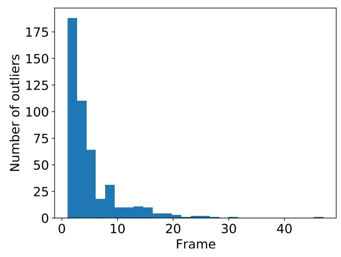
(a) SIFT



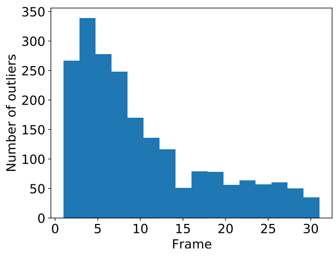
(b) SURF



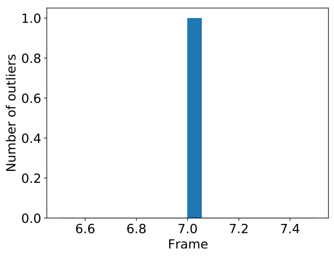
(c) ORB



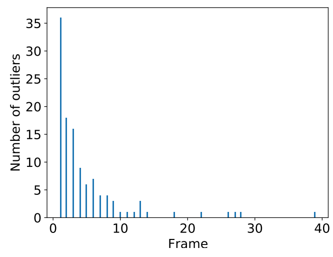
(d) BRISK



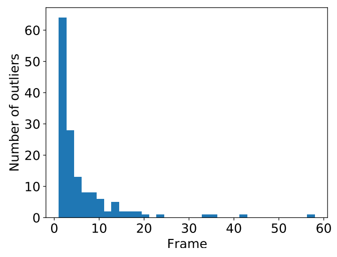
(e) KLT



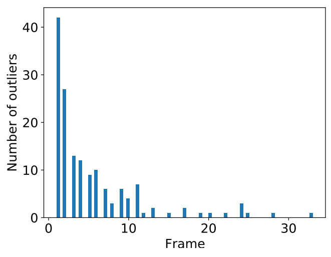
(f) STAR + BRIEF



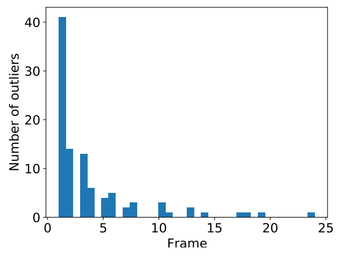
(g) SPV6



(h) SuperPoint Base

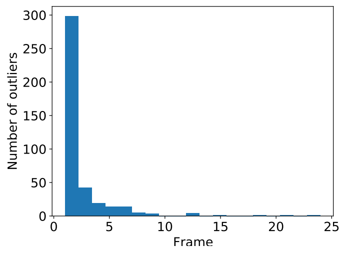


(i) SuperPoint Eros

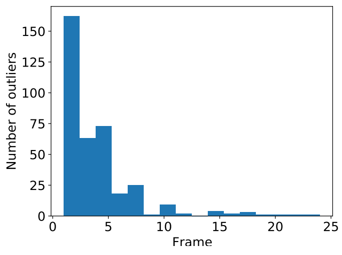


(j) SuperPoint Bennu

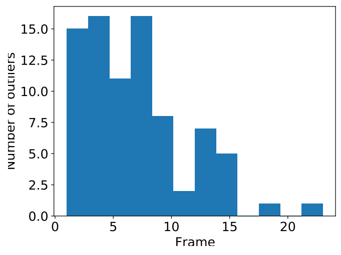
Figure E.55: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



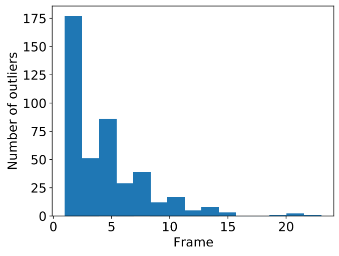
(a) SIFT



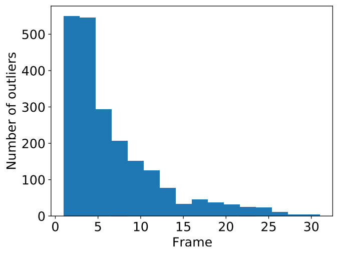
(b) SURF



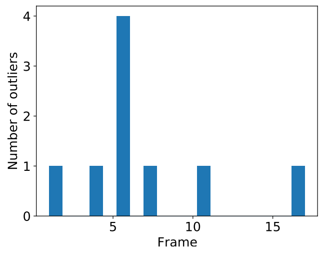
(c) ORB



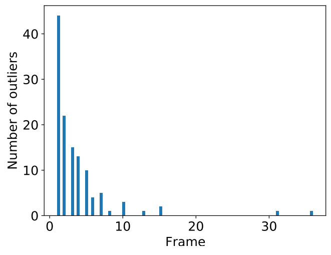
(d) BRISK



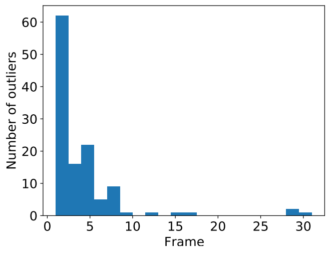
(e) KLT



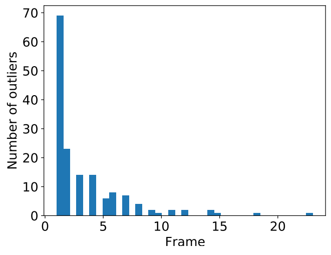
(f) STAR + BRIEF



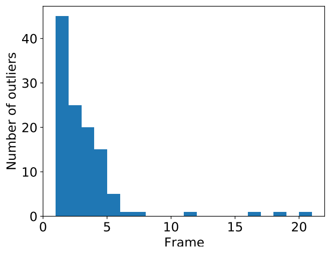
(g) SPV6



(h) SuperPoint Base

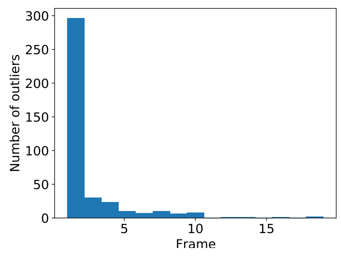


(i) SuperPoint Eros

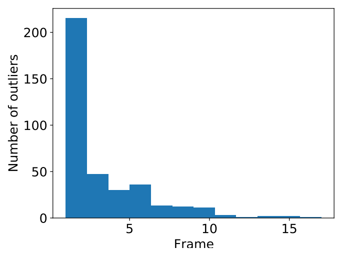


(j) SuperPoint Bennu

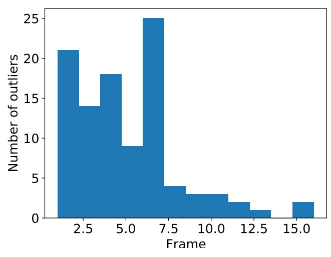
Figure E.56: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.



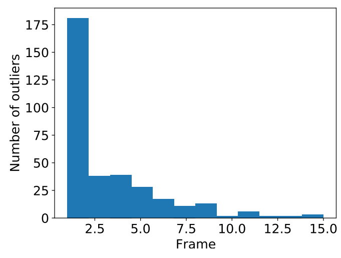
(a) SIFT



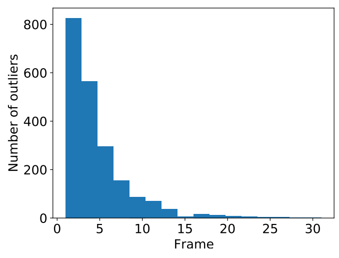
(b) SURF



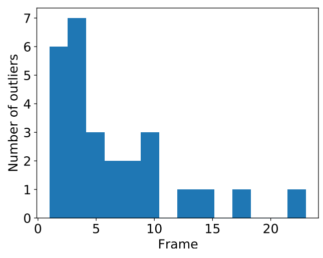
(c) ORB



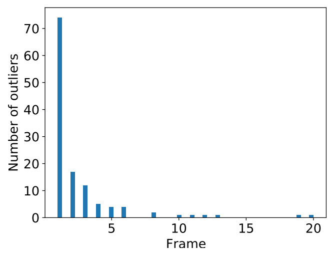
(d) BRISK



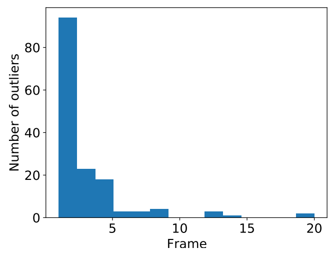
(e) KLT



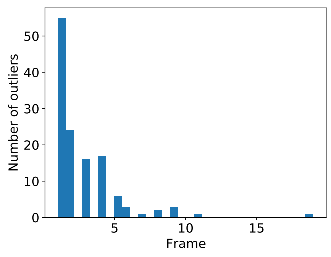
(f) STAR + BRIEF



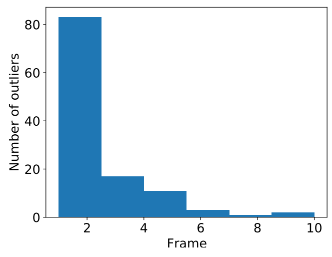
(g) SPV6



(h) SuperPoint Base

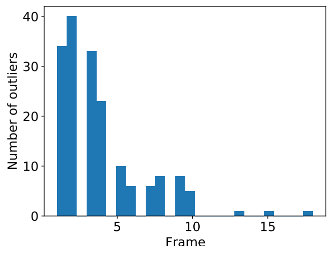


(i) SuperPoint Eros

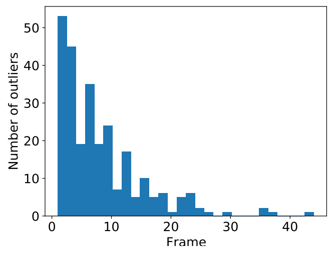


(j) SuperPoint Bennu

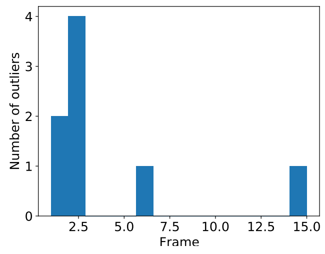
Figure E.57: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



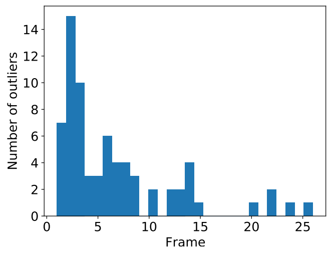
(a) SIFT



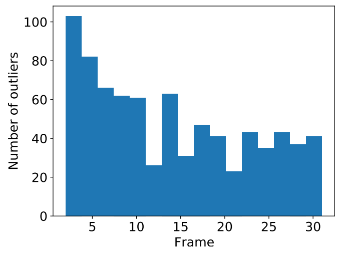
(b) SURF



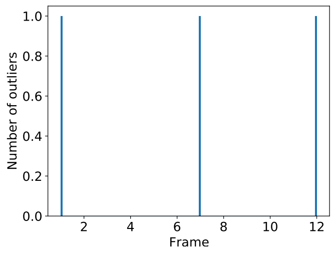
(c) ORB



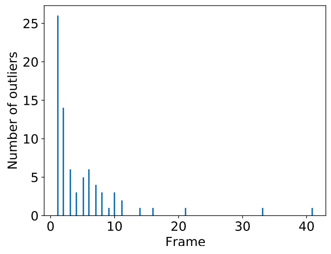
(d) BRISK



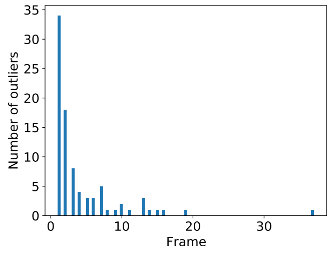
(e) KLT



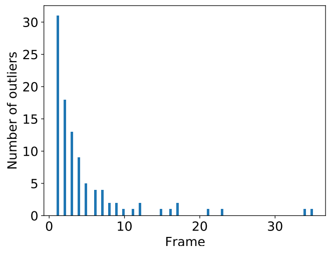
(f) STAR + BRIEF



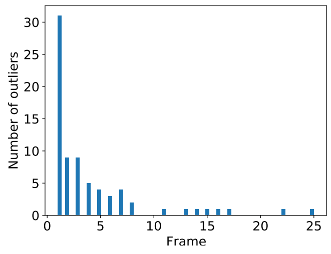
(g) SPV6



(h) SuperPoint Base

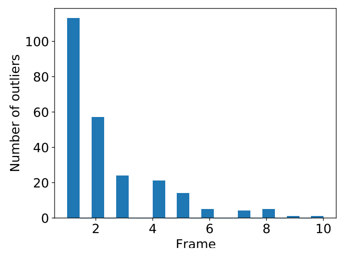


(i) SuperPoint Eros

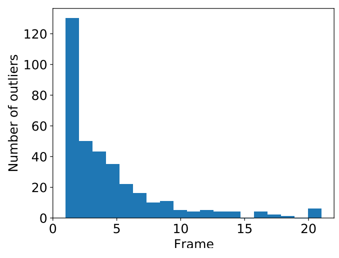


(j) SuperPoint Bennu

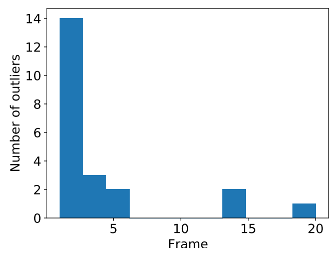
Figure E.58: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



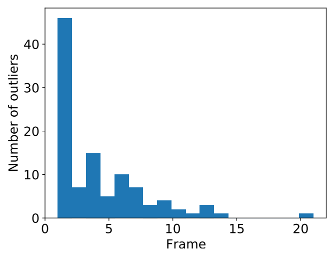
(a) SIFT



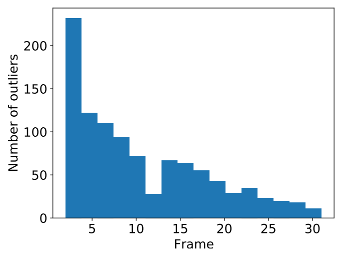
(b) SURF



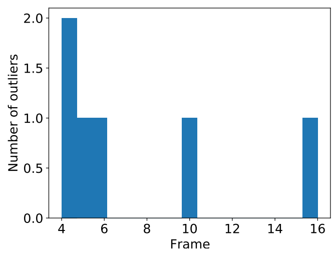
(c) ORB



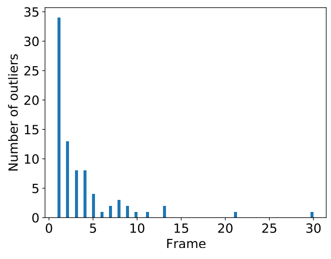
(d) BRISK



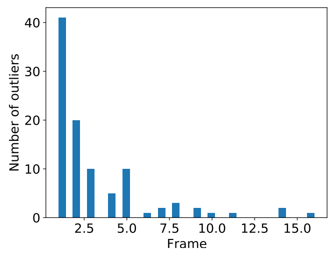
(e) KLT



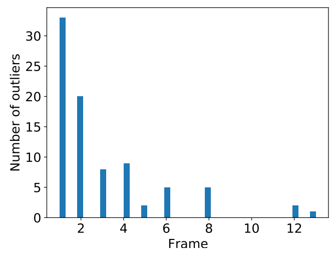
(f) STAR + BRIEF



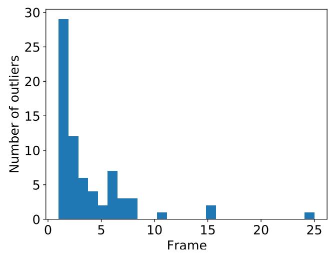
(g) SPV6



(h) SuperPoint Base

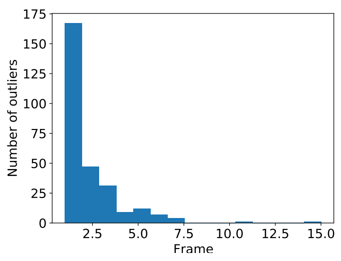


(i) SuperPoint Eros

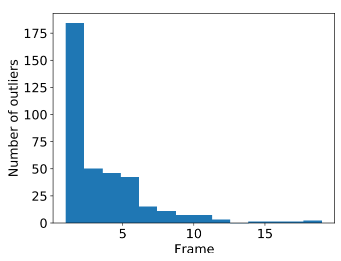


(j) SuperPoint Bennu

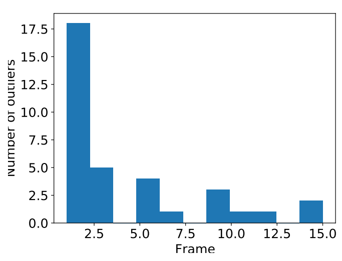
Figure E.59: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



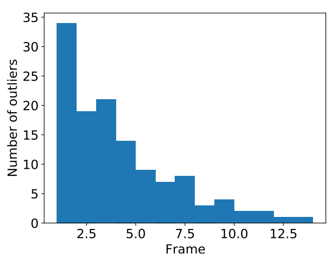
(a) SIFT



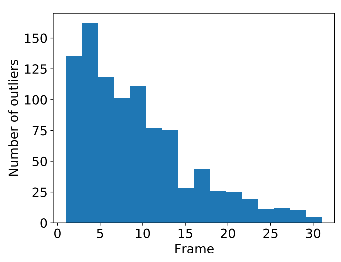
(b) SURF



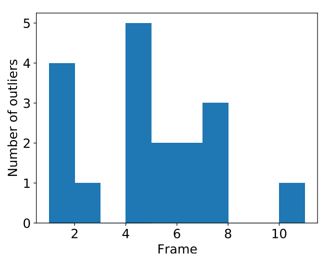
(c) ORB



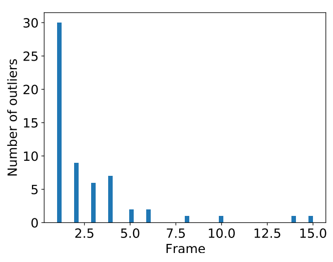
(d) BRISK



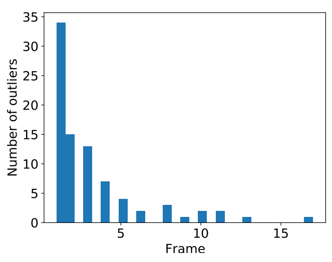
(e) KLT



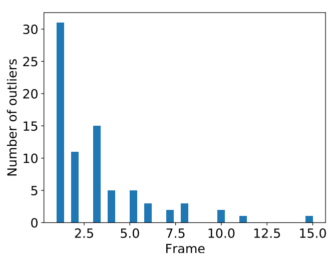
(f) STAR + BRIEF



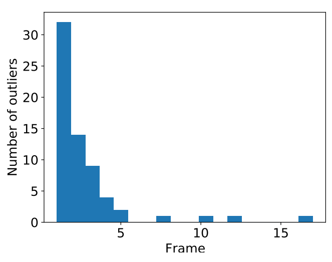
(g) SPV6



(h) SuperPoint Base



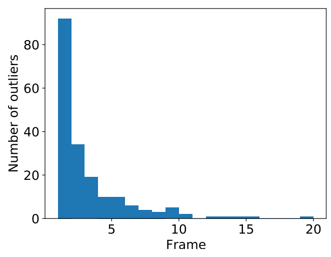
(i) SuperPoint Eros



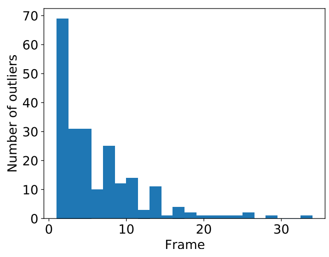
(j) SuperPoint Bennu

Figure E.60: Asteroid: Itokawa. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.

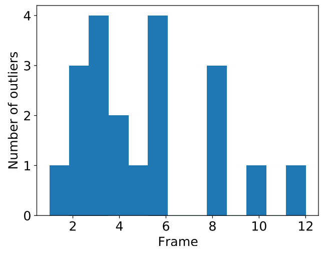
### E.4.3. Eros



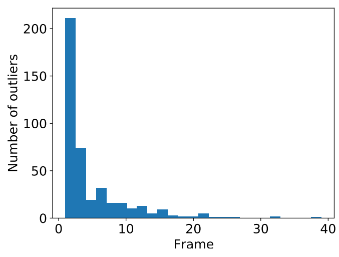
(a) SIFT



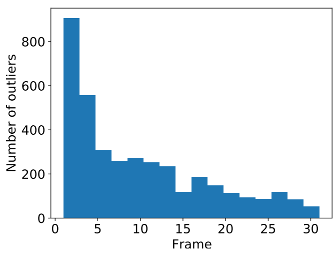
(b) SURF



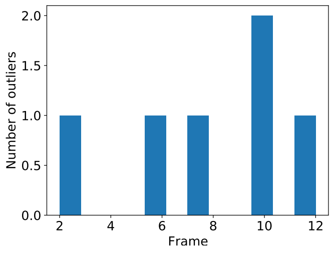
(c) ORB



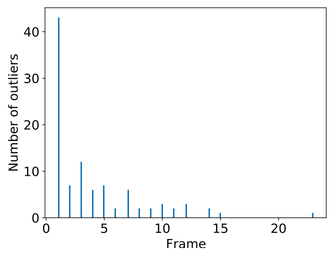
(d) BRISK



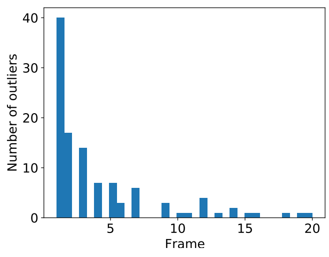
(e) KLT



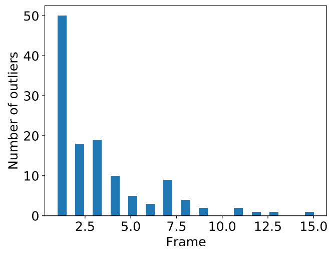
(f) STAR + BRIEF



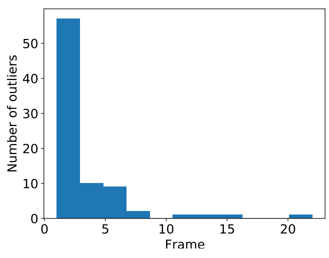
(g) SPV6



(h) SuperPoint Base

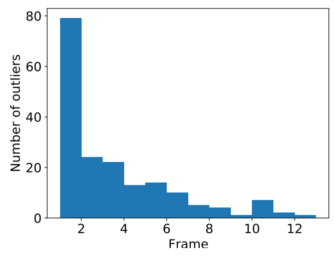


(i) SuperPoint Eros

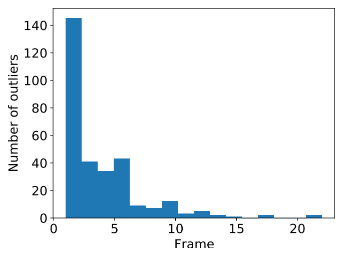


(j) SuperPoint Bennu

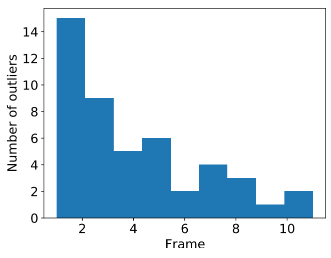
Figure E.61: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



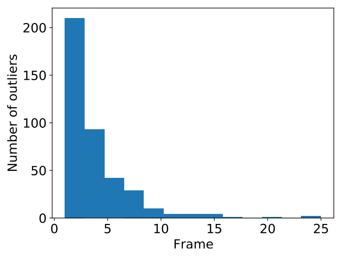
(a) SIFT



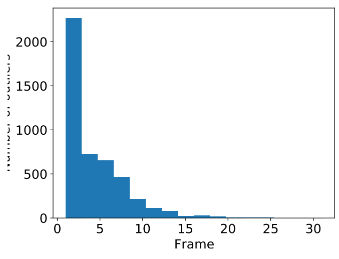
(b) SURF



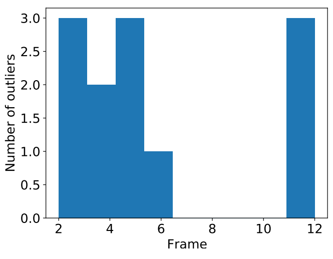
(c) ORB



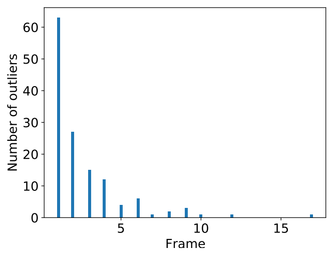
(d) BRISK



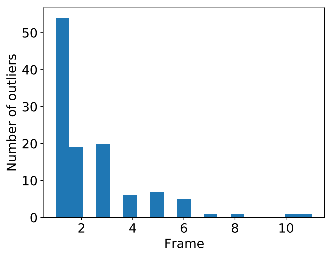
(e) KLT



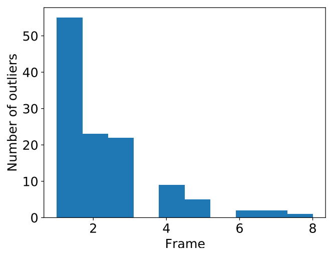
(f) STAR + BRIEF



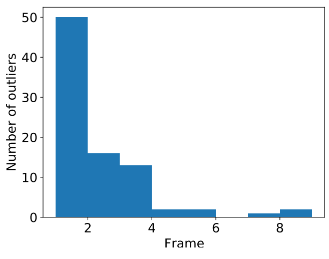
(g) SPV6



(h) SuperPoint Base

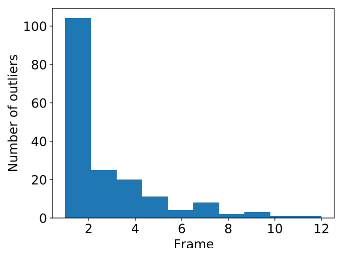


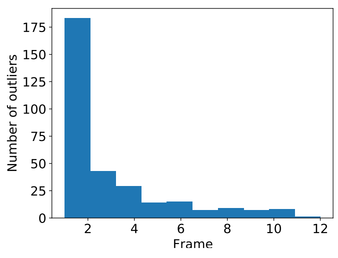
(i) SuperPoint Eros

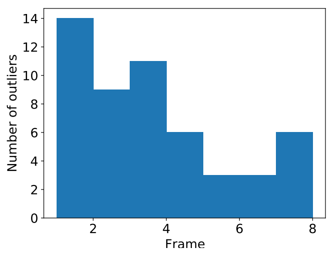


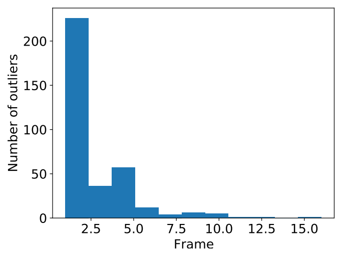
(j) SuperPoint Bennu

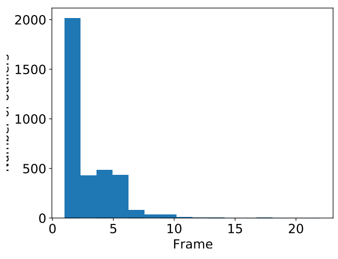
Figure E.62: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.

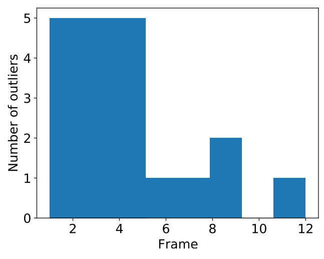

(a) SIFT

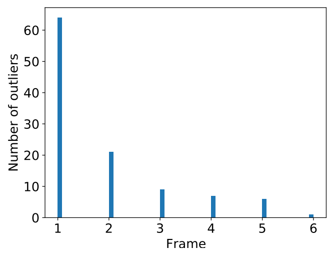

(b) SURF

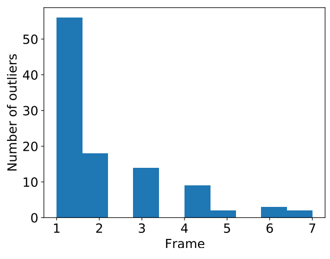

(c) ORB

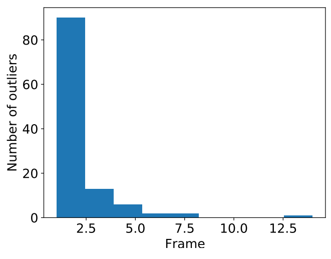

(d) BRISK


(e) KLT


(f) STAR + BRIEF


(g) SPV6


(h) SuperPoint Base


(i) SuperPoint Eros

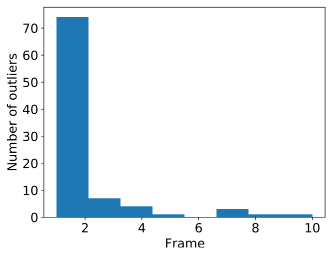
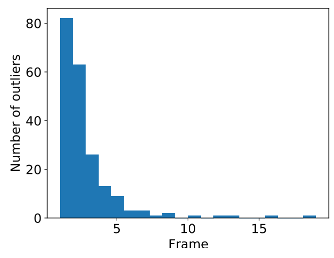
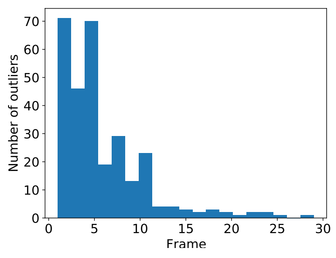

(j) SuperPoint Bennu

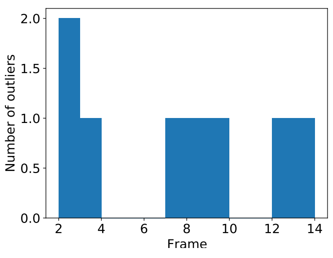
Figure E.63: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



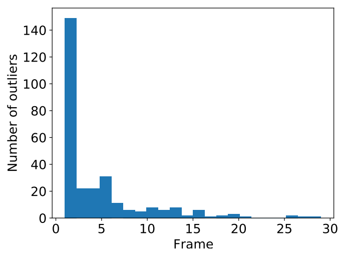
(a) SIFT



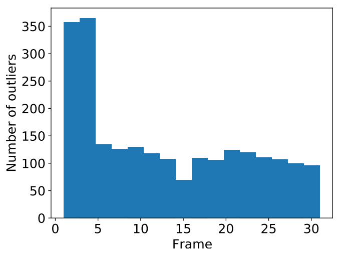
(b) SURF



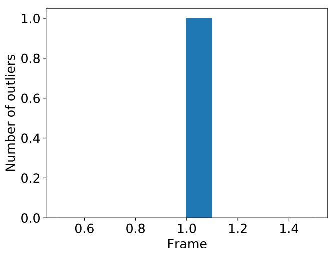
(c) ORB



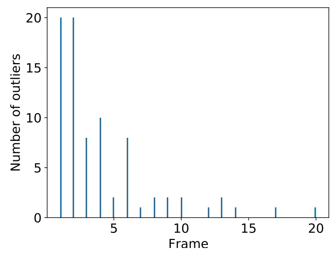
(d) BRISK



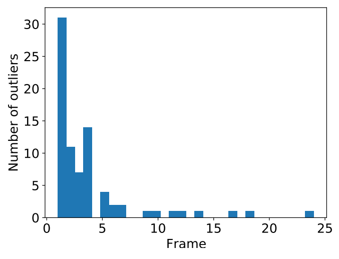
(e) KLT



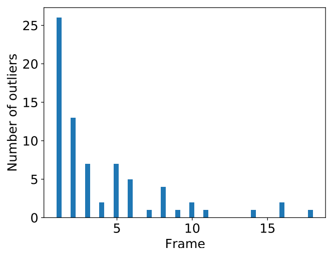
(f) STAR + BRIEF



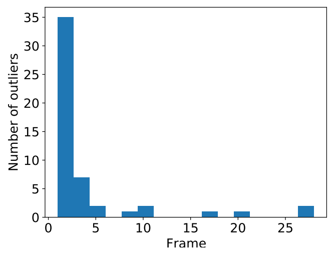
(g) SPV6



(h) SuperPoint Base

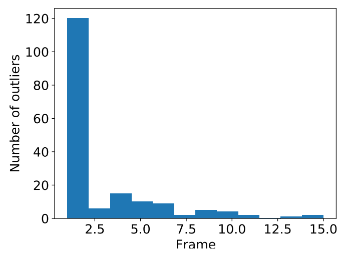


(i) SuperPoint Eros

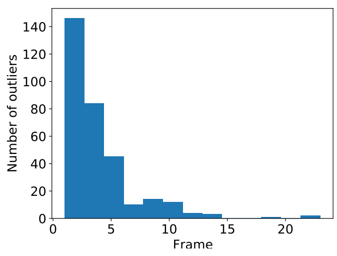


(j) SuperPoint Bennu

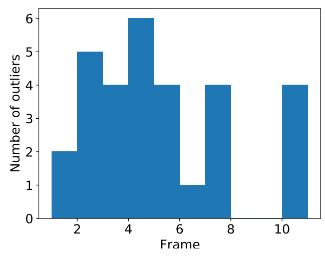
Figure E.64: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



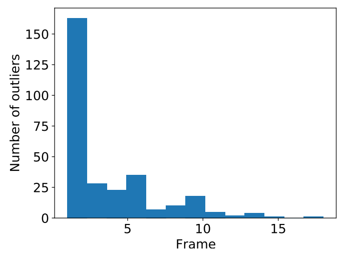
(a) SIFT



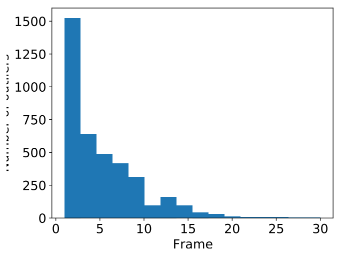
(b) SURF



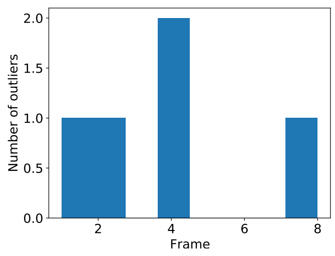
(c) ORB



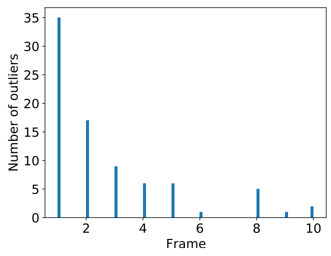
(d) BRISK



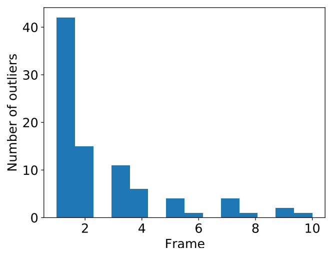
(e) KLT



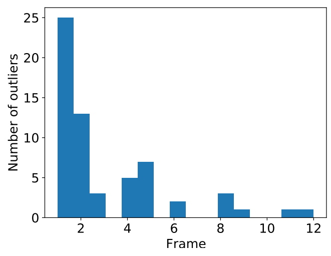
(f) STAR + BRIEF



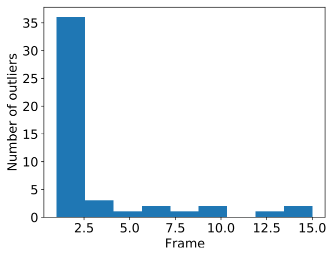
(g) SPV6



(h) SuperPoint Base

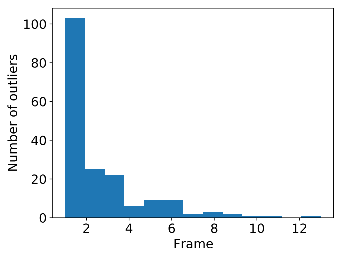


(i) SuperPoint Eros

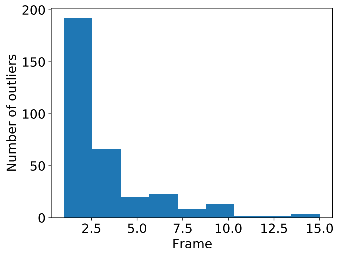


(j) SuperPoint Bennu

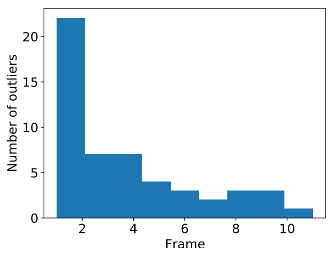
Figure E.65: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



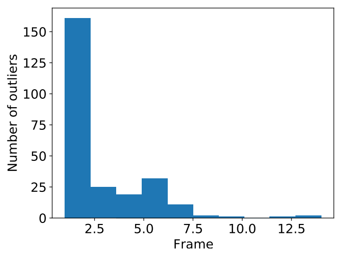
(a) SIFT



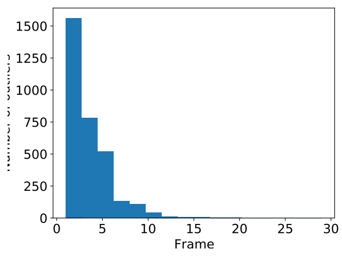
(b) SURF



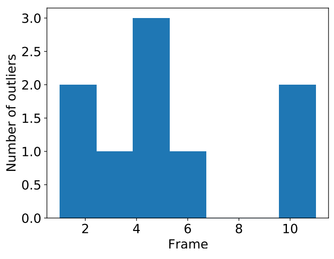
(c) ORB



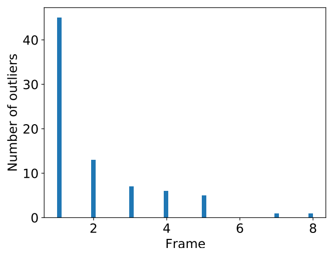
(d) BRISK



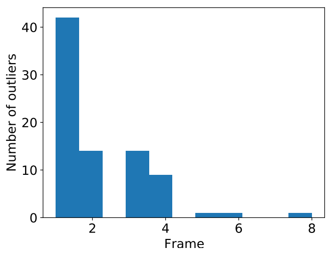
(e) KLT



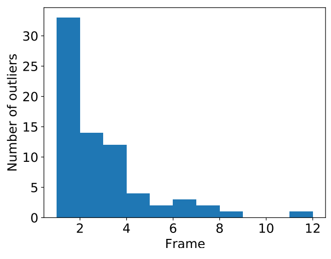
(f) STAR + BRIEF



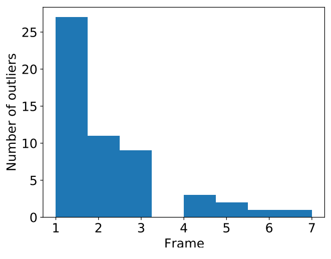
(g) SPV6



(h) SuperPoint Base

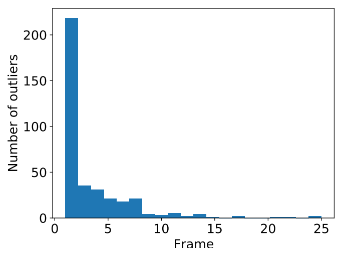


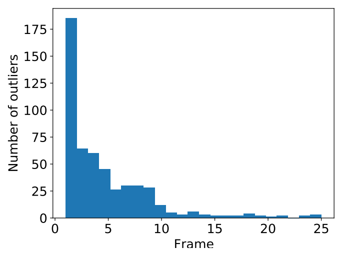
(i) SuperPoint Eros

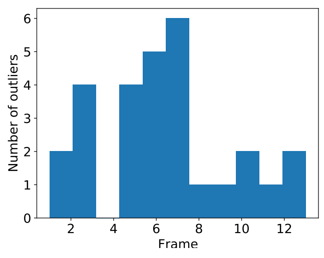


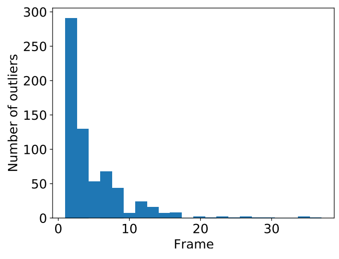
(j) SuperPoint Bennu

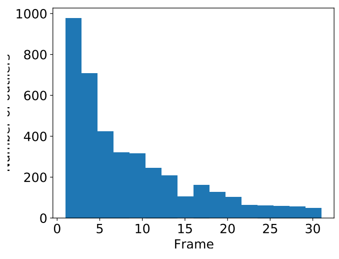
Figure E.66: Asteroid: Eros. Light: $1.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.

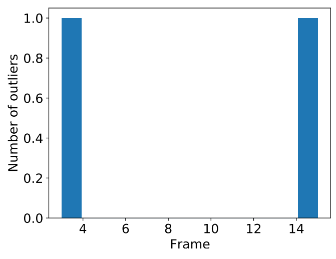

(a) SIFT

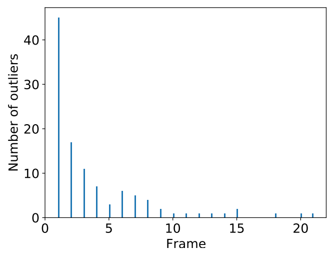

(b) SURF

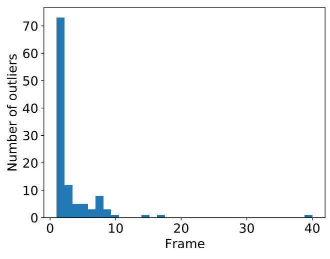

(c) ORB

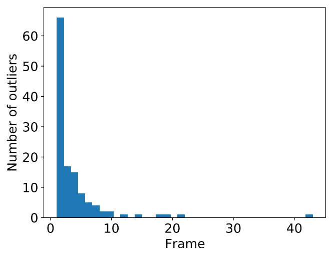

(d) BRISK


(e) KLT


(f) STAR + BRIEF


(g) SPV6


(h) SuperPoint Base


(i) SuperPoint Eros

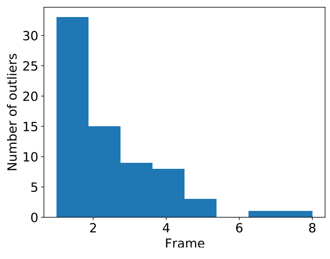
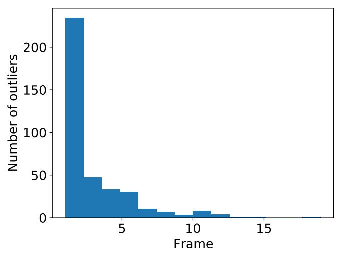
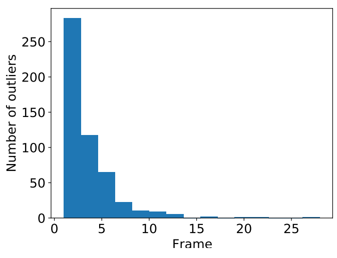

(j) SuperPoint Bennu

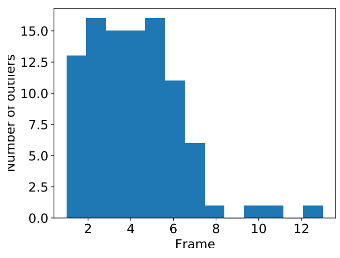
Figure E.67: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $1Km$.



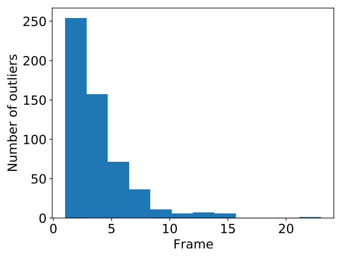
(a) SIFT



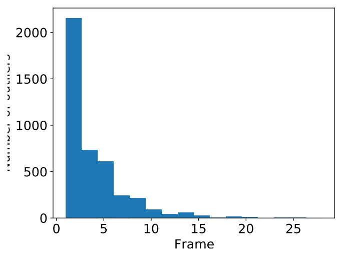
(b) SURF



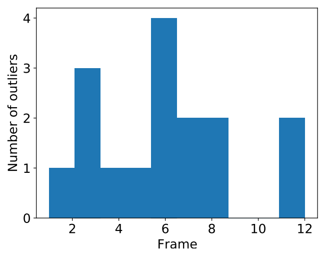
(c) ORB



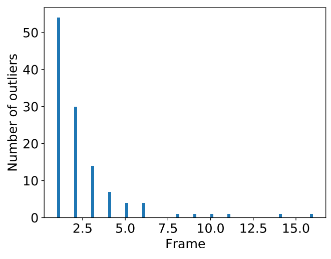
(d) BRISK



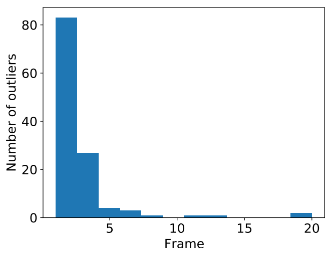
(e) KLT



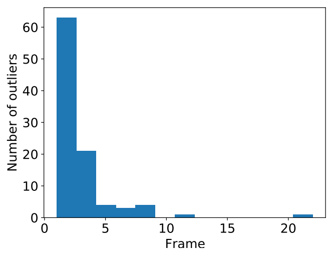
(f) STAR + BRIEF



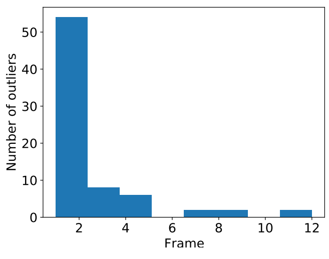
(g) SPV6



(h) SuperPoint Base

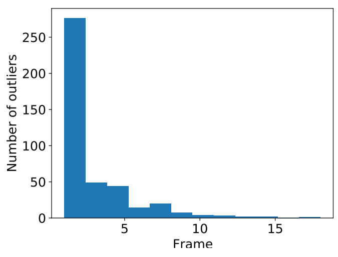


(i) SuperPoint Eros

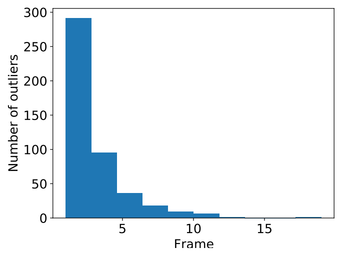


(j) SuperPoint Bennu

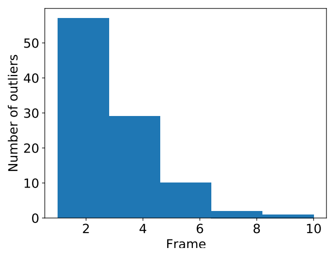
Figure E.68: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $1Km$.



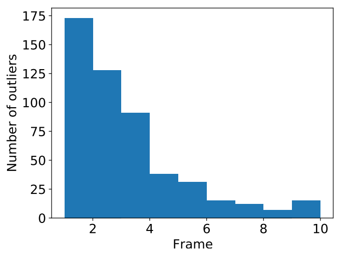
(a) SIFT



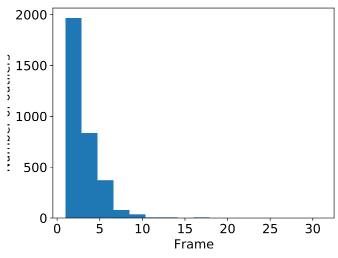
(b) SURF



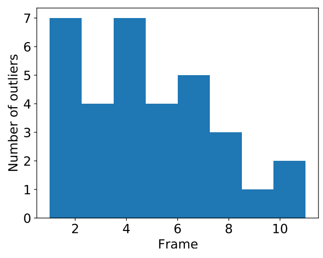
(c) ORB



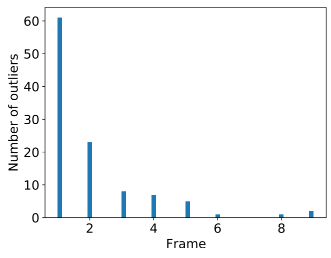
(d) BRISK



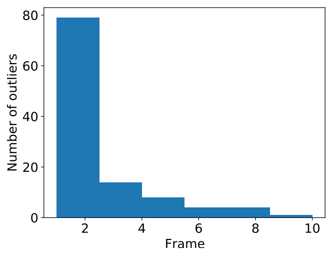
(e) KLT



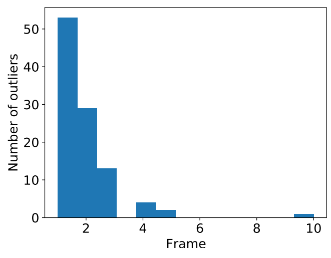
(f) STAR + BRIEF



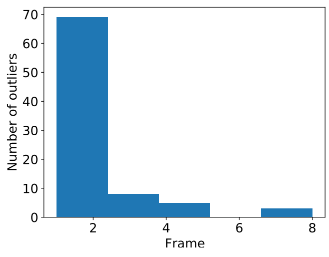
(g) SPV6



(h) SuperPoint Base

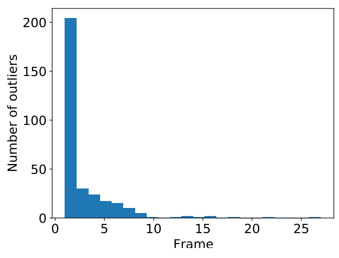


(i) SuperPoint Eros

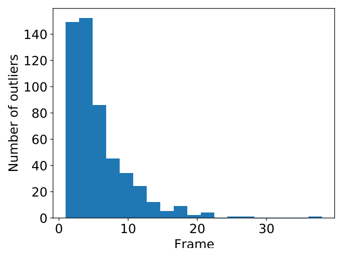


(j) SuperPoint Bennu

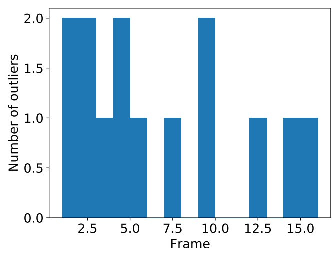
Figure E.69: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $1Km$.



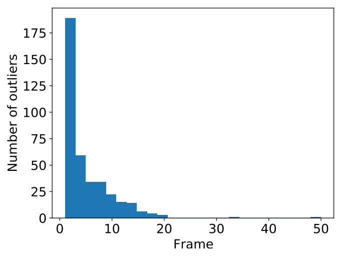
(a) SIFT



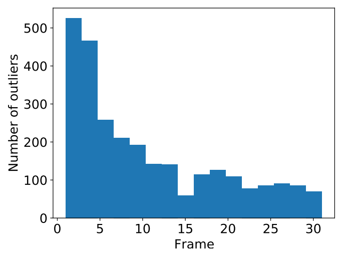
(b) SURF



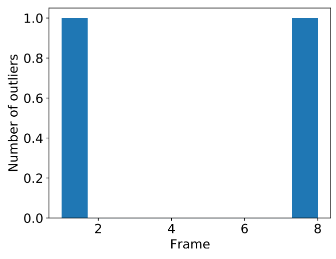
(c) ORB



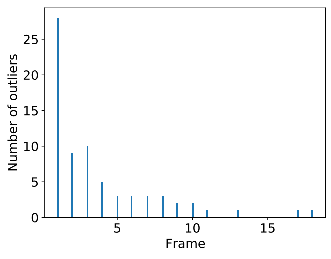
(d) BRISK



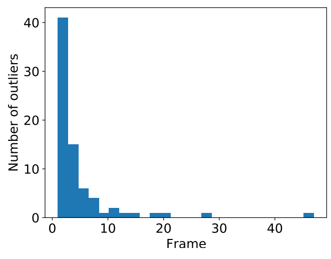
(e) KLT



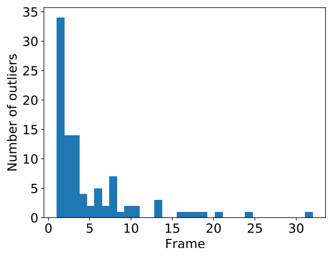
(f) STAR + BRIEF



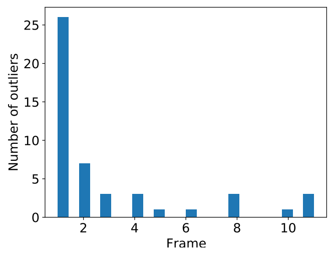
(g) SPV6



(h) SuperPoint Base

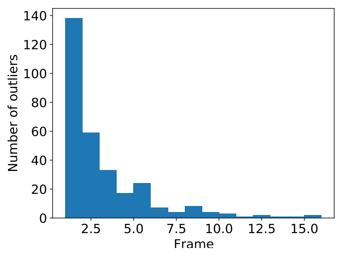


(i) SuperPoint Eros

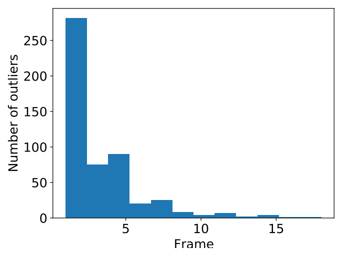


(j) SuperPoint Bennu

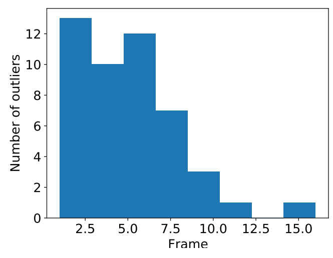
Figure E.70: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 1. Radius: $2Km$.



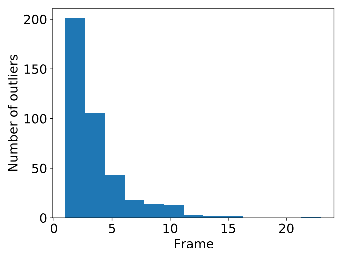
(a) SIFT



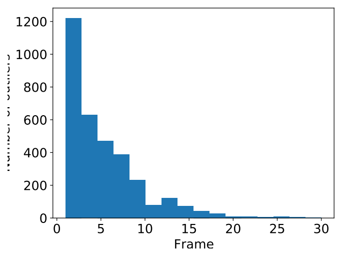
(b) SURF



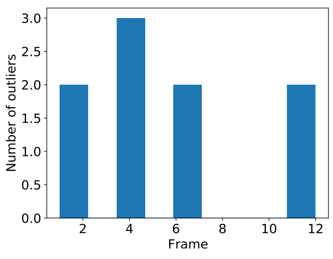
(c) ORB



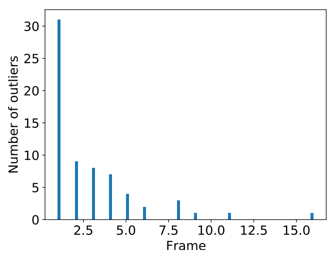
(d) BRISK



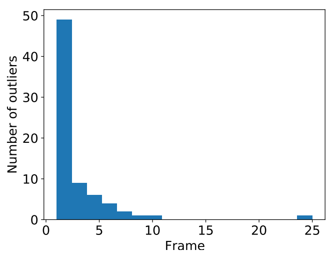
(e) KLT



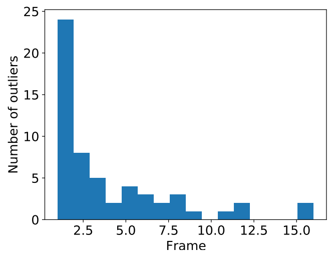
(f) STAR + BRIEF



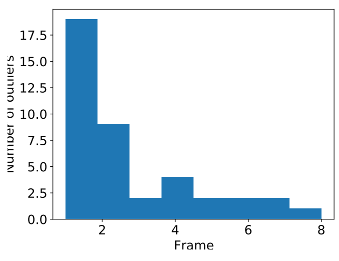
(g) SPV6



(h) SuperPoint Base

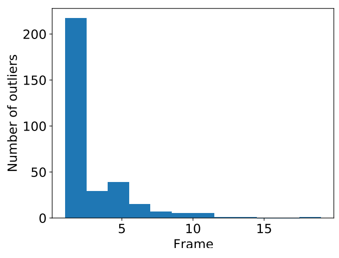


(i) SuperPoint Eros

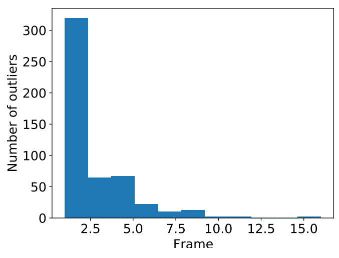


(j) SuperPoint Bennu

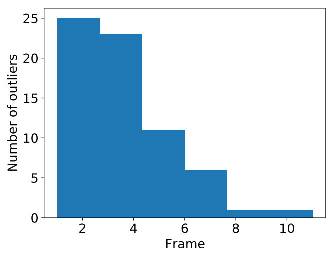
Figure E.71: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 2. Radius: $2Km$.



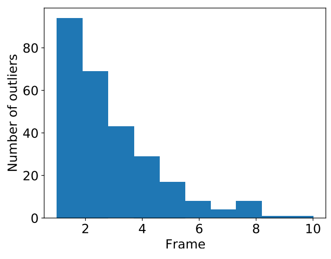
(a) SIFT



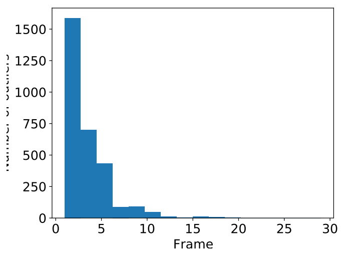
(b) SURF



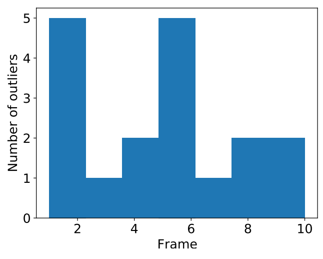
(c) ORB



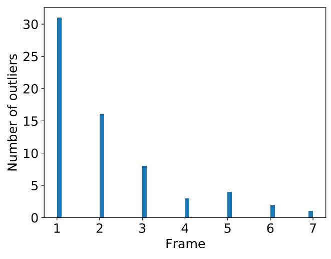
(d) BRISK



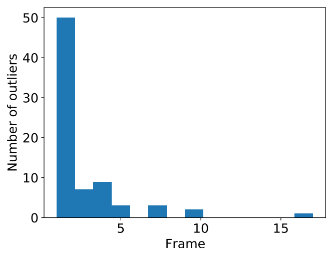
(e) KLT



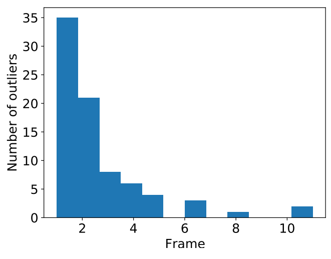
(f) STAR + BRIEF



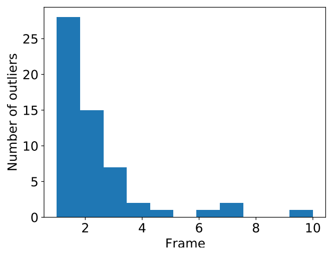
(g) SPV6



(h) SuperPoint Base



(i) SuperPoint Eros



(j) SuperPoint Bennu

Figure E.72: Asteroid: Eros. Light: $5.0\frac{W}{m^2}$. Step: 3. Radius: $2Km$.

# List of Figures

# List of Tables

# List of Symbols

| **Variable** | **Description** | S.I. unit (if applicable) |
|---|---|---|
| $\boldsymbol{P}$ | Camera matrix | $[px]$ |
| $\boldsymbol{K}$ | Camera intrinsic matrix | $[px]$ |
| $\mathcal{W}$ | World reference frame | n.a. |
| $\mathcal{C}$ | Camera reference frame | n.a. |
| $\boldsymbol{R}$ | Rotation matrix | $[-]$ |
| $r$ | Probe position (radius) | $[Km]$ |
| $i$ | Orbital inclination | $deg$ |
| $I$ | Light intensity | $\frac{W}{m^2}$ |
| $\boldsymbol{l}$ | Illumination direction | $[-]$ |
| $\boldsymbol{n}$ | Normal to the surface | $[-]$ |
| $C$ or $L$ | Cost / loss function | $[-]$ |
| $\sigma$ | Sigmoid function | n.a. |
| $\boldsymbol{a}$ | Neuron output | $[-]$ |
| $\boldsymbol{y}$ | Data label | $[-]$ |
| $\boldsymbol{W}$ | Weights matrix | $[-]$ |
| $\boldsymbol{b}$ | Bias vector | $[-]$ |

# Acknowledgements

Dopo quasi sei anni, sia brevi che lunghi, il mio percorso al Poli si conclude. E' stato un cammino duro, ma solamente ora comincio a rendermi conto di ciò che mi ha lasciato, sia dal punto di vista personale che in vista di una futura carriera.
Un primo ringraziamento va al professor Francesco Topputo, a Mattia Pugliatti e a Paolo Panicucci, che mi hanno dato l'opportunità di sviluppare questo lavoro e il cui contributo è stato essenziale.
Vorrei ringraziare i miei genitori, che mi hanno permesso di intraprendere questo percorso e che mi hanno sempre supportato e sostenuto in questi anni, ed il resto della mia famiglia, nonne, cugini e zii.
Sono grato a tutte le persone che ho conosciuto in questi anni al Poli, in particolare Lollo, Aspe ed Elia, con cui ho condiviso il percorso universitario, Cesa e Matte, che sono tornati con i piedi per terra ma continuano ad esserci, e tutti gli altri ragazzi conosciuti in triennale. Il ricordo che avrò di questi anni è dovuto in gran parte a voi.
Un ringraziamento speciale va a tutti gli amici, in particolare Francesco e Paolo, e a tutti gli altri compagni di serate e di vacanze. Siete stati l'ancora di salvezza quando avevo bisogno di una pausa dal Poli.
Infine, grazie a tutte le altre persone con cui in qualche modo ho condiviso parte di questo percorso, i compagni/colleghi conosciuti nei vari progetti, i compagni di avventura in Erasmus e tutte le altre persone che sicuramente avrò dimenticato ma che, in un modo o nell'altro, hanno contribuito a farmi arrivare dove sono ora.
Grazie a tutti.