



Detection of small features in flat-field images of astronomical cameras: application to the OSIRIS instrument on board the Rosetta (ESA) mission

Alicia D. Benítez ^(a), Luisa M. Lara ^(a)

^(a) Solar System Department, Instituto de Astrofísica de Andalucía, Consejo Superior de Investigaciones Científicas (IAA-CSIC), c/Glorieta de la Astronomía s/n, ES-18008, Granada, Spain

Article Information

Keywords:

small features,
detection and recognition,
internal calibration lamps,
clustering,
astrophysics.

Corresponding author:

Alicia D. Benítez
Tel.: +34 958 121311
Fax.: +34 958 814530
e-mail: abenitez@iaa.csic.es
Address: IAA-CSIC, c/Glorieta de la
Astronomía s/n, ES-18008, Granada,
Spain

Abstract

Flat field images acquired with astronomical cameras, both in space and groundbased observatories, do usually contain features which are originally placed on the CCD detector or on the filters that comprise the camera. The detection of these features and monitoring (i.e. time evolution of the number of these found features) are the main purposes of two codes developed in IDL. We have applied these codes to the flat-field images acquired with the Narrow Angle Camera (NAC) internal calibration lamps on board the Rosetta spacecraft launched in 2004. The features found in the images can be identified as: dust on the detector, dust on the filter, scratches on the detector, scratches on the filters or pinholes (characteristic light leakage is seen as the degradation of the filters' coatings). This paper presents two different software (named DustDetector and DustEvolution), which detect the above mentioned features by means of an algorithm based on the Roberts filter to detect, and a clustering algorithm to the recognition of the features. By means of an experimental study, we have observed the effectiveness of the methods in the analysis of the features and their evolution among different images acquired in interplanetary space since 2004 until 2011.

1 Introduction

Rosetta [1] is the name of an ESA (European Space Agency) interplanetary mission to study the comet 67P/Churyumov-Gerasimenko. It comprises an orbital and a landing module called Philae. The Rosetta spacecraft was launched from *Centre Spatial Guyanais* (CSG) in Kourou in March 2004. The mission has a nominal lifetime of 11 years, ending in December 2015. The mission analyses the physical and chemical processes occurring in the nucleus and surface of the comet 67P to know more about the accretion process in the protoplanetary disc, the physical and chemical conditions (temperature, pressure, molecular composition), the relationship between the original interstellar composition and the disk composition, and the variation of its properties with both time and heliocentric distance. The size distribution of planetesimals and the degree to which they come from different parts of the protoplanetary disc can be studied directly by images from which the heterogeneity of a cometary nucleus at all scales can be determined.

Several instruments are used by the mission to achieve its scientific objectives, and the Optical, Spectroscopic, and Infrared Remote Imaging System (OSIRIS) is one of them. OSIRIS is the scientific camera system onboard the Rosetta spacecraft which will characterise the comet shape and volume, its rotational

state and find a suitable landing spot for Philae, the Rosetta lander.

OSIRIS comprises a high resolution Narrow Angle Camera (NAC) unit and a Wide Angle Camera (WAC) unit accompanied by three electronics boxes. The NAC is designed to obtain high resolution images of the surface of comet 67P/Churyumov-Gerasimenko through 12 discrete filters over the wavelength range 250–1000 nm at an angular resolution of $18.6 \mu\text{rad px}^{-1}$. The NAC allows an initial detection of the nucleus, the study of its structure and rotation from relatively large distances (typically 10^4 km), the investigation of the mineralogy of the surface, and study the dust ejection processes. The instrument takes images with 2048 x 2048 pixels in grey scale and IMG format. For internal calibration purposes, the instrument has four lamps (two main and two redundant) to shine light on the optics and to obtain images with an even illumination of the detector through the filters. This is what we call flat-field images.

The OSIRIS instrument is required to operate over a period of 11 years in a harsh radiation environment after its launch date on the 2nd of March 2004. The geometric distortion of the cameras needs to be periodically remeasured in flight because it might have changed during launch and afterwards due to the different thermal environments the camera has been in addition to that, the detectors will be affected by energetic particles during cruise leading, amongst others, to changes in the detector dark current and charge transfer efficiency. The optics

might be sputtered by energetic particles, and the primary mirrors might eventually further deteriorate because of dust particle impacts. Pinholes might originate in the filters.

Thus, any feature detected, either on the filters and/or the CCD during the laboratory calibration, in-flight commissioning and operations, needs to be characterised as there are no means to replace either the filters or the CCD detectors after the mission launch. These features, depending on their nature, can lead to a wrong scientific interpretation of any physical process occurring on the comet nucleus and/or in the gas and dust coma.

Characterisation of these features comprises detection, location and determination of their shape. One of the most popular methods for detecting features in images is Roberts filter [2, 3]. Roberts filter is a convolution-based operator for extracting edges from images: two 2-D arrays run over an image separately to find gradient edges.

DustDetector software detects features on the flat field images acquired with the NAC internal calibration lamps. Once these features were detected, they were identified as: dust on the detector, dust on the filter, scratches on the detector, scratches on the filters, pinholes (characteristic light leakage is seen). The software implements an algorithm to detect each feature. For this purpose, a Roberts filter is used in order to detect the object frontiers in the image and a threshold is established to identify the sort of feature, either extended or one single pixel. The dust in the image clearly manifests as a set of pixels, whereas features with an isolated pixel do not represent dust but dead pixels on the CCD. Therefore, a clustering algorithm has been also implemented to know the groups of dust pixels existing in the image.

DustEvolution software is used for detecting the different coordinates of dust pixels between two flat-field images acquired with the camera: one of the images is a reference, whereas the other one is an image acquired any when since the mission launch (or at any time). This software is based on the DustDetector software which is applied to both images, the reference one and any other image acquired with the same filters combination. Once the coordinates of the features in both images are located and stored in two different data structures, an algorithm based on the search of different coordinates in both data structure is applied.

The NAC images (in the IMG format compatible with the ESA and NASA standards for space missions data delivery) were acquired either with the main or redundant internal calibration lamps with any filters combination. These images are restricted to the OSIRIS Team and can be only downloaded from a password protected server in the Max-Planck Institute fuer Sonnensystemforschung (MPS) in Katlenburg-Lindau (Germany).

According to that, this contribution is organized as follows: Section 2 describes the methodology of each software; Section 3 shows the effectiveness of the software applied to any CCD image (in particular we have focused our work on the NAC images which provide a more even illumination of the CCD detector); and finally, Section 4 contains some conclusions and future work.

2 Software description

Each software is described in the following subsections 2.1 and 2.2. Let us note that the implementation language has been IDL [5] by means of a Command Line Interface and some IDL routines specially developed to deal with data in the Planetary Data Systems (PDS). The PDS archives and distributes scientific data from NASA planetary missions, astronomical observations, and laboratory measurements. The PDS is sponsored by NASA's Science Mission Directorate. Its purpose is to ensure the long-term usability of NASA data and to stimulate advanced research. For more information about the PDS format, we refer the reader to [4].

2.1 DustDetector software

DustDetector Software is an IDL code designed to automate the detection of features in flat-field images and to obtain a summary with the main characteristics of these features in each analysed image. The summary contains the number of pixels with dust in each image and their coordinates, or the number of groups that conforms a feature. This is possible by means of two procedures: 1) Detection, and 2) recognition of pixels that constitute a feature. The Algorithm 1 has been designed for detection purpose.

Inputs: IM--> NAC image file IMG, τ --> threshold Outputs: pixel_coordinates --> (x,y) pixel dust coordinates
<pre> begin procedure I = ReadImage(IM); R = RobertsFilter(I); pixelCoordinates = {}; For each $R_{i,j} \in R$ if ($R_{i,j} > \tau$) then pixelCoordinates = pixelCoordinates U {$R_{i,j}$} endif endfor end procedure </pre>

Algorithm 1: Detection of dust pixels.

For detecting features, Roberts filter is applied to the image in order to obtain the edges of the objects in gray scale. A threshold is defined to select the features in the processed image by the Roberts filter. The criterion is set according to the value of the pixel in DNs (Note: as in every CCD image, the pixels contain information in the so-called "digital numbers", DNs). The choice of the threshold parameter has been done in a way that the number of features visually identified coincides with the numerical result of our algorithm. Figs. 1 shows an original CCD image (or NAC image) whereas the result of applying the Roberts filter to it can be seen in Fig. 2.

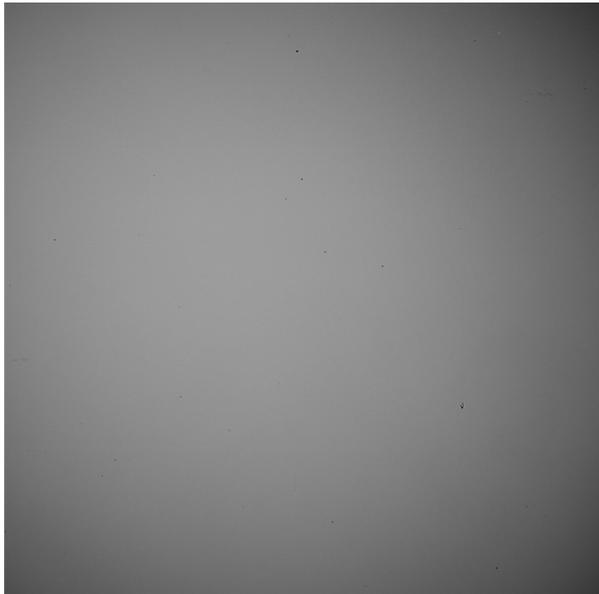


Fig. 1: Original NAC image

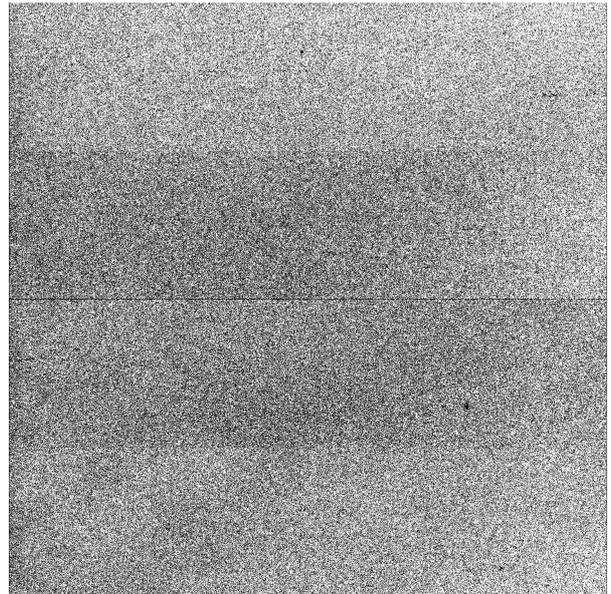


Fig. 3: $\tau = 1 * \text{average}(R)$

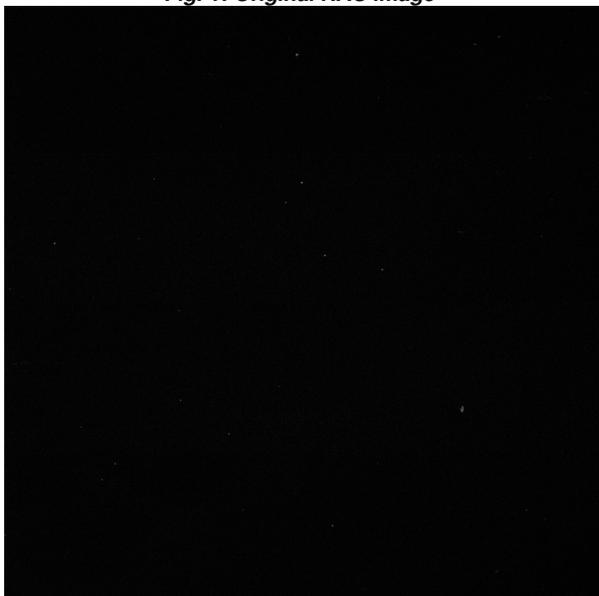


Fig. 2: Edge detection by Roberts filter

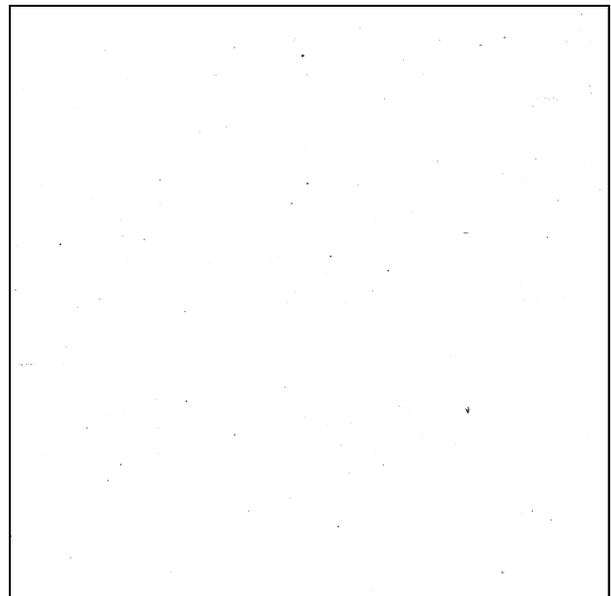


Fig. 4: $\tau = 5 * \text{average}(R)$

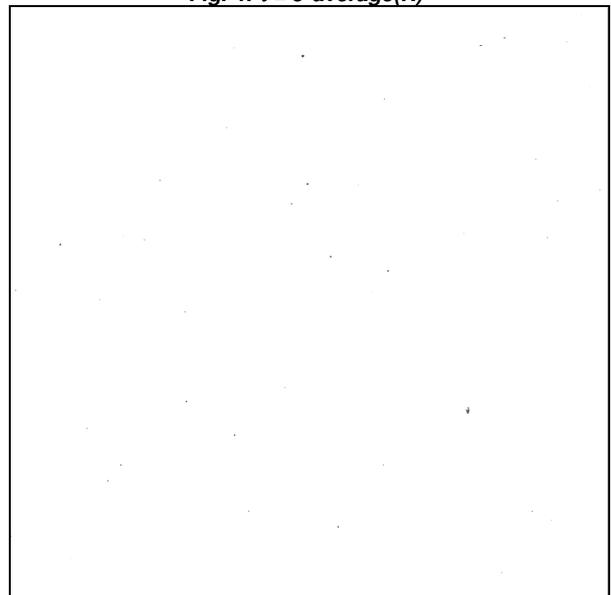


Fig. 5: $\tau = 10 * \text{average}(R)$

We define the threshold as

$$\tau = n * \text{average}(R) \quad (1)$$

where R is a 2-D array obtained from applying Roberts filter to the input image, average(R) is the average value of an array whose values are between 0 and 255 (in gray scale), and n is the parameter under study for defining the threshold τ .

Figs. 3-5 contain an example of the image after applying the algorithm 1 with $n = 1, 5$ and 10 , respectively.

These figures illustrate that the number of mapped detections depend on the threshold τ . For smaller τ , the number of features is rather large and it does not coincide with the features visually identified. In fact, Fig. 3 clearly shows that $n=1$ (i.e. small τ) shows the CCD imperfections as dead pixels and electronic noise. After a long investigation, we conclude that the best agreement between features visually identified and with algorithm 1 is obtained for $\tau = 10 \cdot \text{average}(R)$.

Fig. 6 shows how one of the features detected according to the different values of τ looks like. We can observe that the feature is identified with more precision with a higher τ .

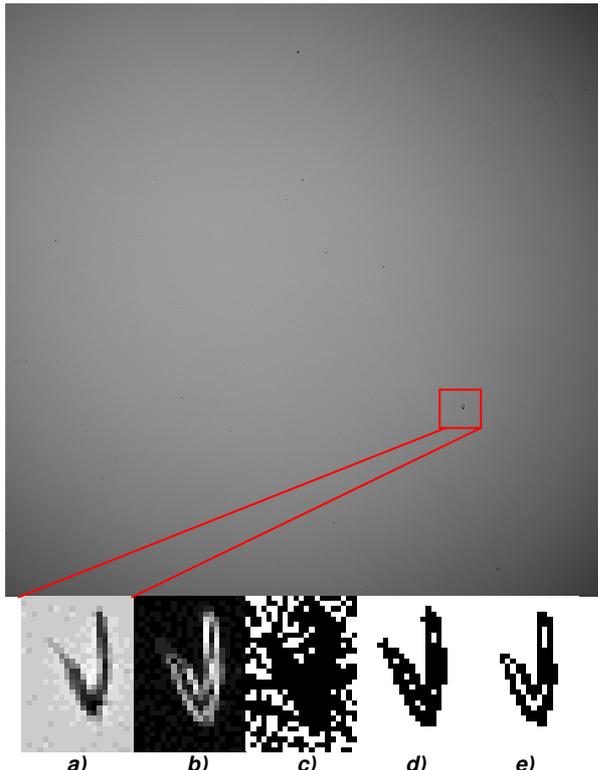


Fig. 6: Identification of one feature. a) Feature in the original image, b) the same feature after applying Roberts filter, c to e) visualization of the same feature when $\tau = 1 \cdot \text{average}(R)$, $\tau = 5 \cdot \text{average}(R)$, and $\tau = 10 \cdot \text{average}(R)$

Two parameters are received by Algorithm 1: a NAC image and a threshold, τ , which determines the granularity level of distinguishing features. The next step is to map those inhomogeneities and for this, the coordinates of the pixels forming the features clearly visible in the processed image are obtained. The mapping method works as follows: each feature enhanced by Roberts filter is analysed, and so, if the value is greater than τ , then there is a positive feature detection. If the number of pixels detected is greater than 20000, the image will be considered as corrupted after verifying that any problem (as a shutter locking, uneven CCD illumination, etc) prevented a homogeneous (i.e. flat) image acquisition.

When the features are determined, an algorithm for features recognition is applied. Let A and B be two pixels where (x_1, y_1) and (x_2, y_2) are their coordinates, respectively. The Euclidean distance is calculated as in

eq. 2. Fig. 6 illustrates the Euclidean distance between two points.

$$\text{EuclideanDistance} = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (2)$$

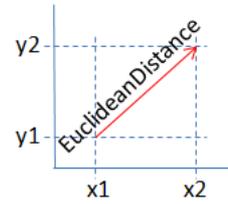


Fig. 7: Euclidean Distance calculation

This concept will be used by a clustering algorithm in order to detect groups of dust pixels. A group is defined as a set of consecutive pixels where their Euclidean distance is less than two. If the Euclidean distance is one, only the purple pixels (see Fig. 8) are considered as part of a feature, whereas the visual identification really indicated that the number of pixels was higher comprising the dark-blue pixels in Fig. 8. On the other hand, the number of elements (i.e. number of pixels with their coordinates) of each group is called granularity. Algorithm 2 shows the pseudocode.

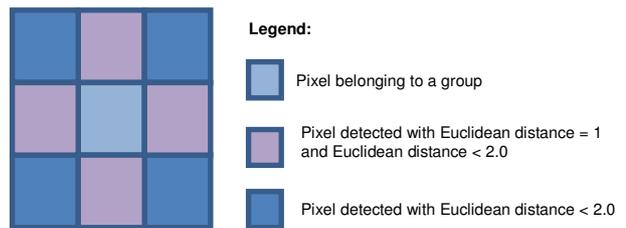


Fig. 8: Pixel clustering detection.

```

Inputs: P --> a set of (x,y) pixel dust coordinates
Outputs: G --> groups of pixel dust coordinates
begin procedure
  P = {p1, p2, ..., pn}; // List of dust pixels,
                        // where pi is (xi, yi) pixel coordinate
  G = {}; //List of groups of clusters
  g1 = {p1};
  P = P - {p1};
  G = G U {g1};

  For each pixel pi ∈ P
    For each group gj ∈ G and not belong to any group
      For each pixel pk ∈ gj and not belong to any
      group
        D = Euclidean_distance(pi, pk);
        If(D < 2.0) then
          gj = gj U {pi};
        endif
      endfor
    endfor

    If(pi does not belong to any group) then
      gnew = {pi};
      G = G U {gnew};
    endif

  endfor
end procedure
    
```

Algorithm 2: Recognition of dust pixels. Clustering algorithm.

When the clustering algorithm is applied, the pixel coordinates of the found features will be placed on a white background image (see Fig. 5). Depending on the granularity level indicated by the user on the command line, groups with size greater than the granularity value will be placed on a white background image (see Fig. 9-11 for the resulting images generated with granularity equal to one, four and eight).

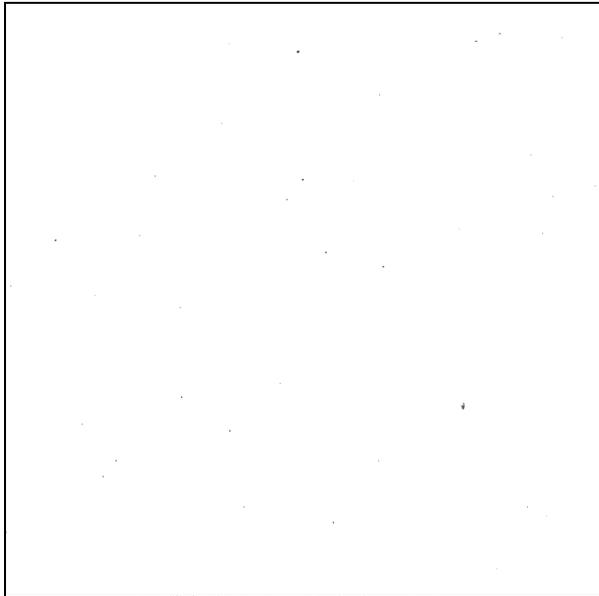


Fig. 9: NAC image, granularity 1.

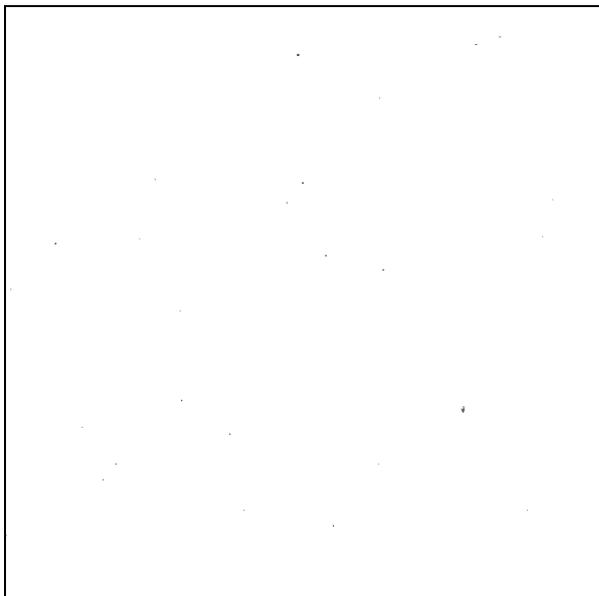


Fig. 10: NAC image, granularity 4

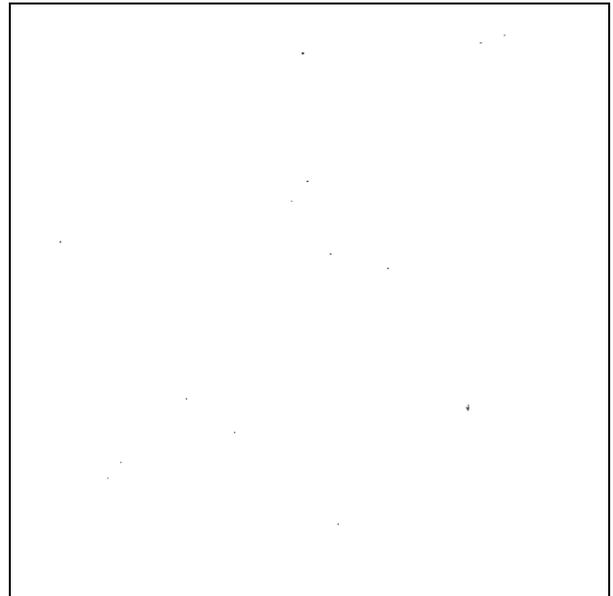


Fig. 11: NAC image, granularity 8

Fig. 12 illustrates that the granularity does not affect to the shape of the features detected.

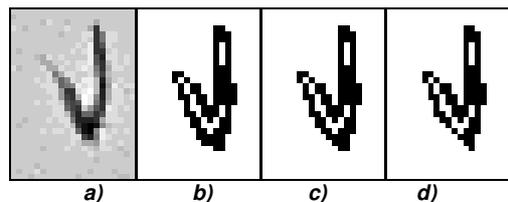


Fig. 12: Identification of one feature. a) Feature in the original image, b to d) feature displayed in a white background image choosing granularity 1, 4, and 8 after applying the algorithm to detect clusters.

Finally, the software generates the following files:

- 1) A file in PDS format with the original image.
- 2) A file in JPG format with the original image for visualization.
- 3) A file in JPG format with the original image with appropriate contrast for a better view of the features. On command line the user can choose the contrast level for the JPG image.
- 4) A file in JPG format with the image obtained after applying the Roberts filter (e.g. Fig. 2).
- 5) A JPG image resulting from Algorithm 1, which shows the dust pixels detected (Fig. 5).
- 6) A JPG image obtained during the recognition process by the Algorithm 2 (Fig. 9-11).
- 7) A JPG image with the subtraction of the image obtained from the detection and the recognition of dust pixels. As an example, let be X and Y two images where X is obtained by means of the

detection of dust pixels algorithm (Algorithm 1) and Y is produced by the recognition of dust pixels algorithm (Algorithm 2), the difference (Z) is calculated as eq. 3.

$$Z = X - Y \quad (3)$$

It allows us to check the accuracy of algorithm 1 and 2 to detect and recognize the features in the image. The method is highly accurate as Z turns out to be an image with white background (i.e. no feature remains undetected).

- 8) A text file for each image which contains the coordinates of the pixels forming a feature found by Algorithm 2.
- 9) A text file containing a summary of all images processed, listing the following information for each image: file name, acquisition time of the image, name of the filters used during the image acquisition, numbering of those filters, number of dust groups, and total number of dust pixels.
- 10) A text file which indicates the corrupted images which cannot be processed.

2.2 DustEvolution software

DustEvolution Software is designed to automate the detection of features between NAC images and a reference one. A summary with the main features is obtained, such as temporal evolution between the image coordinates for the same set of NAC filters, the number of pixels with dust for each image, or the total number of dust pixel in both images in order to check the increase or decrease of features with time.

The code is written such that the user must have two directory structures. One of them must save the reference images for each filter and the other one must contain any other taken images.

The study of these features is possible by means of the following processes:

- 1) File selection. The reference and any other image are selected and analysed for each filter from two different directories. Algorithm 3 has been designed for the selection of the most recent NAC image. The file selection algorithm receives a directory with NAC files as input and returns a list with the most recent NAC images for each NAC filter combination. At first, the algorithm obtains the NAC filters from the header of the images such that the comparison is done for the same set of filters both in the reference and in the analysed image. The next step is a file search based-on time criteria for consistent filter combinations.
- 2) Detection and recognition of dust pixels for each image by means of the algorithms used in the DustDetector software (see section 2.1).

Input: D --> NAC image directory Output: filesPerFilter --> List of recent NAC image for each Filter

```

begin procedure
  D = {d1, d2, d3, ..., dn}; //List of files
  F = {}; //List of filters
  filesPerFilter = {}; //List of recent files for each filter
  F = GetFilters(D);

  For each filter fi ∈ F
    For each file dj ∈ D and not analysed previously
      If (GetFilter(dj) == fi) then
        recentFile = {dj};
        For each dk ∈ D
          If (filterName(recentFile) == filterName(dk) and
              time(recentFile) > time(dk)) then
            recentFile = {dk};
          End if
        End for
      End if
      filesPerFilter = filesPerFilter U {recentFile};
    End for
  end procedure
    
```

Algorithm 3. File selection.

3) Detection of different feature coordinates between the reference and any other image taken with the camera. When the dust pixel coordinates have been determined in the two images, Algorithm 4 is applied for detecting the coordinates of different dust pixels. The algorithm obtains coordinates (x, y) of the dust pixel from both images by means of the algorithms implemented in the DustDetector Software. These coordinates are saved into a list. Within this list, if a coordinate is found twice, it means that the same feature is present in both images, that is, it has been maintained in time. Otherwise, the feature is new, it has disappeared, or it has moved to another location – which is not traceable with the current codes and it is beyond the scope of this work-.

Input: filesPerFilter --> List of recent files for each filter PIPE --> pipeline image directory Output: diffCoordList --> a set of (x,y) pixel dust different coordinates between pipeline and NAC images equalCoordList --> a set of (x,y) pixel dust common coordinates between pipeline and NAC images

```

begin procedure
  filesPerFilter = {f1, f2, ..., fn}; //List of recent files for each filter
  DPIPE={ dpipe1, dpipe2, dpipe3, ..., dpipen }; //List of
  //pipeline files
  coordinatesList= {};

  For each file dpipei ∈ DPIPE
    For each file fj ∈ filesPerFilter
      If (filterName(dpipei) == filterName(fj)) then
        coordinatesList= {};
        diffCoordList= {};
        equalCoordList= {};

        //See DustDetector (section 2.1)
        coordPipe= DustDetector(dpipei);
        coordNAC=DustDetector(fj);

        coordinatesList={coordPipe} U {coordNAC};

        For each file ck ∈ coordinatesList
          If (GetRepeatedPositions(ck) > 1 ) then
            diffCoordList= diffCoordList U{ ck };
          Else
            If (GetRepeatedPositions(ck) == 0 )
              then
                equalCoordList=
                  equalCoordList U{ ck };
            End if
          End if
        End for
      End if
    End for
  End procedure
    
```

```

                End if-else
            End for
        End if
    End for
End for
end procedure

```

Algorithm 4. Detection of different coordinates.

Finally, the software generates the following files:

- 1) A file PDS format with the recent NAC image.
- 2) A file in JPG format with the original reference and any other NAC image.
- 3) A file in JPG format with the image obtained by Roberts filter applied to every image.
- 4) A JPG image as result of the evolution (obtained by the application of the Algorithm 4).
- 5) A text file for each image which contains the different and equal pixel coordinates. Also, the file contains a summary with the different and equal coordinates of the features, the number of dusty pixels for each image, and the total number of dust pixel in both images.
- 6) A text file which indicates the corrupted images which cannot be processed (see Section 2.1).

3 Software application

The codes developed in this work are applicable to any flat-field astronomical image to detect features both in the optics of the system (i.e. mirrors, filters and lenses) and on the CCD detector. In the case of ground-based telescopes, where all of the subsystems can be reached, dismantled and repaired, the algorithms here presented can be used as diagnostic before attempting the cleaning of the mirrors, filters and/or the CCD.

However, in the case of interplanetary missions where the instruments are remotely checked, the only means to study the evolution of degradation of these instruments is developing routines and procedures as the one outlined in this work. For the particular case of the Rosetta spacecraft, in deep hibernation at distances larger than the orbit of Jupiter since the 30th of June 2011, this software will be applied to flat-field images acquired after wake-up in January 2014 to compare the cleanliness level of the filters and of the CCD with that just before hibernation or even during the laboratory calibration. Identification of these features is of crucial importance to avoid a wrong radiometric calibration of the instrument, or an over-interpretation of scientific images of the nucleus and coma of comet 67P/Churyumov-Gerasimenko (the target of the mission).

Acknowledgement

This work was supported by the Andalusian Government (Grant No. P08-FQM-03837) and research project AyA 2009-08011 awarded by the Spanish Ministry of Science and Innovation (MICINN). The OSIRIS imaging system on board Rosetta is managed by the

Max-Planck-Intitute for Solar System Research in Katlenburg-Lindau (Germany), thanks to an International collaboration between Germany, France, Italy, Spain, and Sweden.

References

- [1] H. U. Keller, C. Barbieri, P. Lamy, H. Rickman, R. Rodrigo, K.-P. Wenzel, H. Sierks, M. F. A'Hearn, F. Angrilli, M. Angulo, M. E. Bailey, P. Barthol, M. A. Barucci, J.-L. Bertaux, G. Bianchini, J.-L. Boit, V. Brown, J. A. Burns, I. Büttner, J. M. Castro, G. Cremonese, W. Curdt, V. Da Deppo, S. Debei, M. De Cecco, K. Dohlen, S. Fornasier, M. Fulle, D. Germerott, F. Gliem, G. P. Guizzo, S. F. Hviid, W.-H. Ip, L. Jorda, D. Koschny, J. R. Kramm, E. Kührt, M. Küppers, L. M. Lara, A. Llebaria, A. López, A. López-Jimenez, J. López-Moreno, R. Meller, H. Michalik, M. D. Michelena, R. Müller, G. Naletto, A. Origné, G. Parzianello, M. Pertile, C. Quintana, R. Ragazzoni, P. Ramous, K.-U. Reiche, M. Reina, J. Rodríguez, G. Rousset, L. Sabau, A. Sanz, J.-P. Sivan, K. Stöckner, J. Tabero, U. Telljohann, N. Thomas, V. Timon, G. Tomasch, T. Wittrock, M. Zaccariotto, et al. "OSIRIS – The Scientific Camera System Onboard Rosetta". *Space Science Reviews* 01/2007; 128(1):433-506. DOI:10.1007/s11214-006-9128-4.
- [2] Roberts filter website: http://www.digplanet.com/wiki/Roberts_Cross.
- [3] Roberts filter website: <http://www.packet.cc/files/mach-per-3D-solids.html>
- [4] Planetary Data System: <http://pds.nasa.gov>
- [5] IDL tutorial: http://www.geo.mtu.edu/geoschem/docs/IDL_Manuals/QUICK%20REFERENCE.pdf