**StraKLIP: A Novel Pipeline for Detection and Characterization of Close-in Faint Companions through the Karhunen–Loève Image Processing Algorithm**

Giovanni M. Strampelli\(^1\), Laurent Pueyo\(^2\), Jonathan Aguilar\(^1\), Antonio Aparicio\(^3\)\(^4\), Gaspard Duchène\(^5\)\(^6\), and Massimo Robberto\(^1\)\(^2\)

\(^1\) Johns Hopkins University, 3400 N. Charles St., Baltimore, MD 21218, USA; strampelligiovanni@jhu.edu
\(^2\) Space Telescope Science Institute, 3700 San Martin Dr., Baltimore, MD 21218, USA
\(^3\) Department of Astrophysics, University of La Laguna, Av. Astrofísico Francisco Sánchez, E-38200 San Cristóbal de La Laguna, Tenerife, Canary Islands, Spain
\(^4\) Instituto de Astrofísica de Canarias, C. Vía Láctea, E-38200. San Cristóbal de La Laguna, Tenerife, Canary Islands, Spain
\(^5\) Astronomy Department, University of California, Berkeley, CA 94720, USA
\(^6\) Université Grenoble Alpes/CNRS, Institut de Planétologie et d’Astrophysique de Grenoble, F-38000 Grenoble, France

Received 2022 June 21; revised 2022 July 28; accepted 2022 August 4; published 2022 September 19

**Abstract**

We present a new pipeline developed to detect and characterize faint astronomical companions at small angular separation from the host star using sets of wide-field imaging observations not specifically designed for high-contrast imaging analysis. The core of the pipeline relies on Karhunen–Loève truncated transformation of the reference point-spread function (PSF) library to perform PSF subtraction and identify candidates. Tests of reliability of detections and characterization of companions are made through simulation of binaries and generation of receiver operating characteristic curves for false-positive/true-positive analysis. The algorithm has been successfully tested on large HST/ACS and WFC3 data sets acquired for two HST Treasury Programs on the Orion Nebula Cluster. Based on these extensive numerical experiments we find that, despite being based on methods designed for observations of a single star at a time, our pipeline performs very well on mosaic space-based data. In fact, we are able to detect brown-dwarf-mass companions almost down to the planetary-mass limit. The pipeline is designed for observations of a single star at a time, our pipeline performs very well on mosaic space-based data. In principle, HCI is a most promising technique. However, even when the primary star can be masked out, the detection of faint close-in companions is heavily hampered by the dominating presence of bright quasi-static speckles mainly caused by imperfections in the optics (Schneider & Silverstone 2003; Biller et al. 2004; Marois et al. 2005).

To deal with this problem, and possibly take advantage of the quasi-static nature of the speckles, a number of point-spread function (PSF) modeling algorithms have been developed: LOCI (locally optimized combination of images algorithm; Lafrenière et al. 2007), NNMF (nonnegative matrix factorization: robust extraction of extended structures; Ren et al. 2018), pynpoint (Amara & Quanz 2012; Stolker et al. 2019), and KLIP (Karhunen–Loève image processing algorithm; Soummer et al. 2012).

In this paper we present an application of KLIP to detect faint astrophysical companions at small angular separation in regular wide-field images taken with the Hubble Space Telescope (HST), i.e., observations not specifically designed for HCI analysis. We focus in particular on data taken with the Wide Field Channel of the Advanced Camera for Surveys (ACS) at visible wavelengths and with the IR channel of WFC3, which can be considered representative of two different PSF sampling regimes. In Section 2 we present a general overview of the pipeline, while in Sections 3 to 8 we describe in more depth each individual step. The conclusions are presented in Section 9. Our results are expressed using two metrics: completeness/robustness of a detection using a receiver operating characteristic (ROC)-based approach and the photometric uncertainty on the measured brightness of discovered companions. In theory astrometric precision on a companion could be considered but precise orbit fitting is usually carried out with narrower-field observations and we thus omit it in the context of this paper.

**1. Introduction**

The ability to detect and analyze companions in binaries and multiple systems is essential for characterizing the frequency and physical properties of those objects. Linking their parameters to models of dynamical evolution from their original molecular cloud can provide insight into the physics at play during the earliest stages of star formation. For this purpose, the multiplicity of pre-main-sequence stars has been investigated over a broad range of environments: dense stellar clusters (e.g., Petr et al. 1998; Luhman et al. 2005; Köhler et al. 2006; Reipurth et al. 2007), young OB associations (e.g., Brown & Verschueren 1997; Shatsky & Tokovinin 2002; Kouwenhoven et al. 2007), and T associations (e.g., Kraus et al. 2008, 2011). Speckle interferometry, adaptive optics, aperture masking, and high-contrast imaging (HCI, e.g., coronography and nulling interferometry) are just some of the techniques that have been applied to obtain rich data sets.

In principle HCI is a most promising technique. However, even when the primary star can be masked out, the detection of faint close-in companions is heavily hampered by the dominating presence of bright quasi-static speckles mainly caused by imperfections in the optics (Schneider & Silverstone 2003; Biller et al. 2004; Marois et al. 2005; Masciadri et al. 2005).
2. Overview of the Pipeline

The core of the pipeline, coded in Python 3.7 (Van Rossum & Drake 1995), can be divided into nine basic steps, which will be briefly introduced here and described in more detail later in the following sections. Due to the highly interactive nature of the analysis, the default version of the pipeline (called StraKLIp) organizes the different steps in a series of Jupyter notebooks (hereafter simply notebooks).

The main objective is to detect faint companions by performing PSF subtraction using the KLIP algorithm, Step 6. This requires two types of input:

1. a set of initial flt HST images, as delivered by the standard HST pipelines
2. an initial catalog of primary targets to be searched for companions (note that the existence of an initial catalog is a soft requirement—see point 2 later in this section)

During the following discussion, we will distinguish between the initial catalog of target sources, typically the result of deriving a photometric source catalog, and the derived input catalog in the format actually accepted by the pipeline. A similar distinction applies to the images: The prefix initial refers to the flt HST images delivered by the standard pipeline and downloaded from the Space Telescope Science Institute (STScI) Mikulski Archive for Space Telescopes (MAST); they are different from the input images needed by the pipeline. These input images are small FITS images cut around each target source; their size defines the searching region S (see Section 6) for each target.

The nine steps of the pipeline are as follows (see also Figure 1):

1. Loading images: A preliminary step needed to adapt the initial images to the format expected by the pipeline. The pipeline requires a set of initial flt images in electrons per second, already multiplied by the pixel area map (PAM)8. This is an image where each pixel value describes the pixel’s area on the sky relative to the native plate scale, and it allows one to account for differences between on-sky pixel sizes across the field of view (FOV) in images that have not been distortion-corrected. For each filter, the pipeline rearranges the full set of images as a multidimensional data cube with the following layers:
   (1) an SCI-labeled image that will contain the actual data,  
   (2) a DQ-labeled image containing the data quality flags,  
   (3) an ERR-labeled image containing the uncertainties associated with the SCI image.

2. Generating a source catalog: If the initial catalog only provides the position of each source on each individual image, preliminary aperture photometry can be performed on all sources using the class photometryAP (see later in Section 7.1). If an initial catalog is not available, a collection of routines has been developed to create it using the TweakReg package from DrizzlePac10 to align the HST images to the Gaia reference catalog (Gaia Collaboration et al. 2018; or any other catalog of sources provided). This gives the pipeline the versatility needed to work regardless of the accuracy, or even the existence of an initial catalog.

3. Generating an input catalog: In this step the initial catalog of target sources is modified and reassembled in order to produce an input catalog for the pipeline in a suitable format.

4. Extracting tiles: Small tiles are created for each entry of the input catalog, trimmed so that the centroid of each star coincides with the center of its tile, setting up a search region S (see Section 6) for each target.

5. Grouping tiles: The tiles are grouped according to their position in the FOV of the instrument to be processed to determine suitable field-dependent PSF stars.

6. Subtracting PSFs: PSF subtraction is performed on each tile using the KLIP algorithm (Soummer et al. 2012).

7. Searching for residual outliers: Residual tiles produced by the previous step are analyzed for the presence of a previously undetected astronomical source.

8. Generating a catalog of candidates: An output catalog of candidate detections, with their photometry, is produced.

9. Inspecting candidates: A collection of tools is used to assess the reliability of the detections. They are based on extensive simulations where both false negatives (missed injected companions) and false positives (noise residuals classified as companions) are statistically analyzed.

The final product of the pipeline is represented by a pandas11 dataframe and stored in a hierarchical data format (HDF5; McKinney 2010). A library of notebooks allows one to analyze the output of the pipeline and produce the types of results shown in Strampelli et al. (2020) and G. M. Strampelli et al. (2022, in preparation). For example, ancillary routines have been developed to provide warnings with the capability of rejecting candidates that do not pass the reliability threshold for false-positive detections. Again, notebooks provide examples of the ancillary routines developed to analyze and interpret the output catalog, providing warnings with the capability of rejecting candidates that do not pass the reliability threshold for false-positive detections.

3. Step 1: Loading Images

The input images accepted by the pipeline are the flt HST images delivered by STScI MAST. While each FITS file is composed of five layers, the pipeline considers only three of them:

1. the science image, extension SCI, where the actual science data reside;
2. the data quality, extension DQ, containing coded information about bad pixels, saturated pixels, cosmic rays (CRs), etc.; and
3. the error layer, extension ERR, providing the variance of the science image.
The initial split_chip.py routine takes advantage of ACSmask/WFC3mask from the DOLPHOT package (Dolphin 2000)\(^\text{12}\) to mask out all pixels flagged as bad in the data quality image, so that they will be ignored in future steps (sky determination, photometry, etc.) and multiplies the initial image by the PAM. While the WFC3-IR data are single-chip image data cubes, in the case of ACS-WFC and WFC3-UVIS, the routine distinguishes between the two CCDs of each camera splitting the initial images into two .chipN FITS data cubes, where \(N\) represents the chip number 1 or 2. Both data cubes contain their relative SCI, DQ, and ERR sublayers. Before running this step it is best practice to back up all the initial images because the ACSmask/WFC3mask from the DOLPHOT package will alter them.

4. Steps 2 and 3: Catalogs

Assuming an initial catalog has been created, the pipeline generates four pandas dataframes:

1. A header dataframe storing all the global information the pipeline will need in order to deliver the final results. This includes project-dependent column labels, selected pipeline input options, and information about the detector like the pixel scale, detector dimensions, etc. This dataframe also provides a “recovery mode” that keeps track of the options selected during the different steps of the pipeline, allowing the user to go back and restart from any intermediate step.

2. A multiple visits dataframe. A multiple-visit target dataframe holds the original position and photometry of each source in the initial images as provided by the initial catalog, with their uncertainties. Ancillary information is also included, such as the program visit, the camera CCD, the telescope position angle \(\text{PA}_V3\), saturation, image name, and more. Sources detected in different visits have one entry for each detection, each detection receiving a unique label.

3. Similar to the previous multiple visits dataframe, the pipeline creates an average dataframe. The average target dataframe holds the average photometry and celestial coordinates of sources that are detected multiple times, along with their unique identifier: average ID.

4. A cross-match dataframe links together the IDs from the multiple visits dataframe to the average ones.

A type keyword is added to the average target dataframe to help during some crucial decision-making points along the pipeline (see Table 1). Sources flagged as type 0 are rejected by the pipeline due to possibly poor photometry, noisy images, the presence of spikes, CRs, image artifacts, or any other reasons that could bring about a wrong detection. Type 1 sources are the typical targets for the pipeline. They usually are isolated and have good photometry (within user-provided parameters). Type 2 labels close pairs that can be resolved but have not been reliably measured in the initial catalog (see Figure 3 for an example of a type 2 source). Type 3 sources are well-resolved pairs with individual photometry. The pipeline will ignore these targets unless told otherwise.

---

\(^{12}\) See the package handbook at http://americano.dolphinsim.com/dolphot/ for more details.
A type 2 flag is not added automatically by the pipeline; instead, it can be added manually through the Showroom routine that allows the user to visually inspect each produced tile and modify its type. Type 2 sources provide a crucial test to the overall performance of the pipeline. Tuning the many options available at each step of the pipeline one should be able to recover most—if not all—of these not fully resolved binary systems. On the other hand, resolved pairs pose a problem for the creation of a good PSF. For this reason, type 3 sources are ignored as their fluxes can be derived using more conventional PSF photometry techniques. Higher types are used to flag nonstellar sources (e.g., galaxies, proplyds, etc.).

Entries in the multiple-visit target dataframes get a new flag to indicate if the source is either a good, unresolved, or bad candidate for PSF subtraction (corresponding to the previously mentioned type 1, 2, or 0), a good-PSF star (also a type 1 but with user-defined limits on photometry), or a wide double (type 3). In particular, to be selected as a PSF star, a source must be bright and well isolated, but not saturated. This selection can be made by parsing a maximum number of saturated pixels in the tile, as well as a user-defined range of magnitudes and errors. Good candidates are selected within a user-defined range of magnitudes and errors. Bad candidates, i.e., sources either saturated, not detected in the filter, or too faint, as well as known doubles (type 3) are ignored. This means that any type different from 0 and 3 can be treated by the pipeline. The pipeline will still run on them and the residuals can easily be retrieved at the end by calling the right type flag.

A last flag is added to the multiple-visit target dataframes to indicate the quadrant (or cell) where the source appears on the detector. Cells have equal sizes and should be large enough to contain enough PSF stars so that different KLIPmodes can be investigated during the PSF subtraction step. This is because the number of KLIPmodes chosen to truncate the Karhunen–Loève transformation is strictly related to the number of reference stars in the PSF reference library. On the other hand, the size of the cells should be small enough to avoid spatial variations of the PSF, providing a local PSF model. The user can decide how many equal-size cells will divide the FOV.

A practical example is provided in Figure 2, where the WFC3-IR FOV has been divided into 10 × 10 cells. Good candidates (gray dots), PSF reference stars (yellow dots), and known doubles (blue dots) are shown in the plot. The particular choice for this partition of the focal plane allows one to maintain the PSF asymmetry within a nominal 1% variation, with about 50 sources per cell.

The pipeline saves an output data frame as an HDF5 file, composed of six header and average/multiple-visit target dataframes and the cross-match IDs dataframes. It will be updated in the following step as the data relative to the target and residual tiles (as output of the PSF subtraction) and photometry are derived.

5. Steps 4 and 5: Tiles

During this step the pipeline runs through every entry in the multiple-visit target data frame that is not labeled as a known double and creates a tile centered on the (x, y) coordinates of the source on the input image. The size of the tile can be defined by the user, the default being 15. In setting the dimensions of the tile, one has to consider the following factors:

1. The area must be large enough to contain the bright wings of the PSF, for sources matching our assumed range of magnitudes.
2. The area must have enough pixels to provide a meaningful noise calculation. Detections of close companions are affected by small number statistics and a correction to the estimated contrast and signal-to-noise ratio (S/N) has to be applied (Mawet et al. 2014).
3. The area must be small enough so that tiles do not overlap. At the actual state of development, the pipeline routines are able to detect only the first brightest companion in a multiple system, so if the companion is already present in the input catalog, the pipeline will skip this system (see type 3 sources in Section 4).

To create a tile for analysis, the routine first creates an initial version using one of the following options, before cleaning CRs and refining the centroid:

1. input coordinates: Uses the coordinates from the input catalog.
2. reference filter: Uses the coordinates from the filter with the smallest magnitude error, excluding detections labeled bad. This option should only be used when the exposures in different filters are taken back to back, without changing the telescope pointing, and the source is barely detected in some of the filters.
3. reference ID: Uses the average coordinates of all detections in the same filter. Again, this may be handy when the pointings between two visits are the same and the sources are faint.

The presence of CRs in the initial tile can heavily affect both the alignment process and later the identification of candidate companions in the residual tile (see Section 6). Therefore, two options are available to mask pixels suspected to be affected by a CR:

1. DQ mask: If the occurrence of a CR has been recorded in the DQ image layer of the input image, a median filter around the flagged pixel is applied. For each CR-flagged pixel in the temporary tile, the median value of the counts in a 3 by 3 pixel mask will be evaluated (ignoring any CR-flagged pixel in the mask) and attributed to the pixel to be corrected.
2. L.A. cosmic ray removal: CRs can also be identified and removed using the Astro-SCRAPPY\textsuperscript{13} Python module (McCully et al. 2018—based on the L.A.Cosmic algorithm from van Dokkum 2001). Figure 3 shows what this module performs on a typical tile: the CR is identified and removed without affecting the central resolved binary (labeled type 2).

\textsuperscript{13} Available at https://github.com/astropy/astroscreppy.
After CR cleaning, a second pass is performed to better estimate the centroid of each source. For this step the user can choose between three different strategies:

1. **maximum**: The brightest pixel within $n$ pixels (default 3 pixels) from the center of the temporary tile is selected as the new center of the tile.
2. **centroid**: The routine looks for a centroid within $n$ pixels (default 3 pixels) from the center of the temporary tile, and assigns to its coordinates the new center of the tile. If selected as default, this approach is usually overruled by the **maximum** for any known unresolved double in the catalog (type 2).
3. **no corrections**: The initial input coordinates are maintained, and no further correction is applied.

By default, the pipeline uses the **maximum** option, but any other option can be selected for all entries in the *multiple-visit target dataframe* (or the average ID can be parsed for individual targets when their behavior is different from that of the typical set). This maximizes the versatility allowing one to adopt the best strategy, given the characteristics of each source and the S/N of each detection.

When the final tile is created, after the centering step, the routine evaluates the new coordinates (if any correction has been made) moving the center of the tile to the new positions in the *input image* and updating the position-labeled columns in the *multiple-visit target dataframe* accordingly. The original subpixel position of the source is preserved during this process. Once the final positions are established, a data cube comprising the following three (four if the option to remove CRs has been enabled) different tiles is saved for each target:

1. the tile itself containing science data, labeled *Data*
2. the corresponding tile cut from the *ERR* layer of the *input image*, labeled *eData*
3. a similar tile obtained from the DQ layer of the input image, labeled DQ
4. If the option to remove CRs has been enabled, a CR-free data tile is also recorded with label CR_Clean.

It is advised that one carefully inspect each newly created tile before moving on to the next step. For this purpose, routines have been developed to facilitate visual inspection and extract useful information at a glance. These routines can be easily accessed through a notebook. A median cube is also saved for each target and filter, containing the median of all the visit tiles and, if the option has been selected, also the CR-free version of it.

6. Steps 6 and 7: PSF Subtraction and Candidate Detection

The pipeline performs PSF subtraction using a pipeline derived from pyKLIP\(^\text{14}\) (Wang et al. 2015) modified to perform reference differential imaging (RDI) of many sources at once. The pyKLIP module is a Python implementation of the KLIP algorithm, invented to achieve accurate PSF subtraction through an operation that can be represented as (for more details see Soummer et al. 2012)

\[
T(n) - \hat{I}_{\psi_0} = \epsilon A(n). \tag{1}
\]

Here \(T(n)\), the target image, is the \(n\)th input tile (either Data or CR_Clean label); \(\hat{I}_{\psi_0}\) is the best representation of \(I_{\psi_0}\), the model of an isolated source with no other astronomical signal in the search area \(S\); and \(\epsilon A(n)\) is an extra astronomical source that may \((\epsilon = 1)\) or may not \((\epsilon = 0)\) be present in the input tile \(T(n)\).

While the “true” PSF to be subtracted cannot be known, a set of reference stars can be regarded as random realizations. The Karhunen–Loéve transformation consists in using the spatial correlation between these realizations to create an orthonormal basis of eigenimages onto which the target star can be projected. The typical output of this process is a new tile having the same size as the input tile, containing the residual of the subtraction process with zero mean. Any additional astronomical signal possibly present in \(T(n)\) will be apparent in this residual’s tile. The residual \(\epsilon A(n)\) is nearly orthogonal to the main telescope’s PSF and is therefore minimally affected by the subtraction process. Depending on its brightness and on the sampling, it may appear as an extended PSF (see Figure 4) or just as a few bright pixels (see Figure 5).

For each input tile, the PSF subtraction process is repeated iterating over a different number of eigenvalues, i.e., the number of modes where to truncate the Karhunen–Loéve transformation, KLIPmode \(K_{\text{klip}}\). Different KLIPmode values can be used to optimally sample different portions of the search area \(S\), with smaller values allowing one to better retrieve signals very close to the target source than higher ones.

In essence we are carrying out RDI with the reference PSFs coming from stars in the same mosaicking program and in the same neighborhood of the detector focal plane.

The tiles containing the residual of each KLIP subtraction, as well as the ones containing the models of the isolated star, are added to the multiple-visit tiles created before.

To select a candidate detection, the routine inspects all KLIPmode layers relative to the same source (observed over different visits and filters) and compares the position of the brightest pixel in each tile. A candidate will be flagged if the following three conditions are realized:

1. the brightest pixel in the residual tile is detected with counts above a user-defined threshold (usually a multiple of the spatial standard deviation, calculated after an initial 3σ cut to eliminate outliers);
2. the brightest pixel is detected as such at the same sky position, for all filters and visits; and
3. the brightest pixel is present in at least two different KLIPmodes.

To determine spatial coincidence between different visits, the routine factors in the telescope rotation and small misalignment, allowing for a user-defined number of pixels, typically 1, of discrepancy. To choose the KLIPmodes, the routine compares the S/N for the KLIPmodes that result in a detection, choosing the one with the highest possible S/N and the lowest possible

\(^{14}\) https://pyklip.readthedocs.io/en/latest/
KLIPmode at the same time. Once the KLIPmode representative of this candidate is selected, a median tile is created for the residual and added to the median cube of each target.

7. Step 8: Photometry

The pipeline offers three different photometry methods: aperture, matched filter (MF), and PSF photometry. While aperture photometry always provides an accurate value if the stars are well isolated and there are no companions, the two other methods can provide a more accurate estimate of the flux of a star if the PSF is well sampled and modeled. This is typically the case for ACS and the visible channel of WFC3, but not for WFC3-IR due to the coarser pixel scale. We therefore illustrate each method with an application to ACS simulations, underlining the strengths and flaws of each strategy in dealing with the data produced by the pipeline.

7.1. Aperture Photometry

To perform aperture photometry the pipeline uses the photometryAp routine, which integrates some of the photutils\textsuperscript{15} features to perform aperture photometry. The two

\textsuperscript{15} photutils is a Python module affiliated with Astropy that provides tools for detecting and performing photometry of astronomical sources. https://photutils.readthedocs.io/en/stable/index.html.
frames at the top of Figure 6 show the two tiles required by this routine. The routine uses the CircularAnnulus module of photutils to evaluate the median sky background \( C_{sk} \). Specifically, it takes a \( 3\sigma \) cut median of the sky in an annulus between two radii \( r_a \) and \( r_b \) (Sky tile in Figure 6). This sky estimate is then subtracted from the data tile creating a sky-subtracted tile. Then, using the CircularAperture module of photutils, the routine sets to zero all pixels outside the circular area of radius \( r_i \) creating a final sky-subtracted aperture tile (or simply Aperture tile in Figure 6), whose total counts \( C_{ap} \) provide the source counts inside \( r_i \). To check the accuracy of the sky estimate, many curves of growth are created (Figure 6, bottom right) correcting the sky value by small amounts and performing multiaperture photometry at different radii to assess the systematic errors on the sky estimate. Indeed, if the sky is correctly estimated then the curve will remain flat over the full range of selected radii. Otherwise, the multiaperture photometry will show a parabolic shape due to the fact that the sky error grows quadratically with the number of pixels in the aperture, i.e., \( \propto r_i^2 \). If the sky has been overestimated, too much flux will be removed as the radius increases and the counts will show a parabolic decrease versus radius; the converse is true if the sky has been underestimated. So, by making different growth curves, the routine can look for the flattest curve in the sample (black curve in Figure 6, bottom right) and return a correct sky estimate.

To evaluate the photometric uncertainty we use the classic relation by Stetson (1987):

\[
\Delta C_{ap} = \sqrt{\text{var}_1 + \text{var}_2 + \text{var}_3}
\]  

(2)
where $\text{var}_1$, $\text{var}_2$, and $\text{var}_3$ are the three main sources of errors added in quadrature. The first term,

$$\text{var}_1 = N_{ap} \text{ std}(C_{sk})^2,$$  

(3)

represents the random noise floor in the aperture, including readout noise and contamination from neighboring stars. This term is given by the variance of the sky values, $C_{sk}$, multiplied by the number of pixels in the aperture, $N_{ap}$. The second term accounts for photon noise associated with the brightness of the source,

$$\text{var}_2 = C_{ap},$$  

(4)

where $C_{ap}$ represents the photoelectrons counted inside the aperture, and the third term,

$$\text{var}_3 = N_{ap}^2 \left( \frac{\text{std}(C_{sk})^2}{N_{sky}} \right),$$  

(5)

is the uncertainty in the estimate of the sky brightness, $C_{sk}$, not accounted for by the variance of source counts but still affecting the actual measure.

Once the final counts in the aperture with their uncertainties have been evaluated, they are converted into magnitude, $m$, with its relative uncertainty, $dm$, through the relations

$$m = -2.5 \log_{10} \left( \frac{C_{ap}}{\text{ExpTime}} \right) + ZP$$  

(6)

$$dm = 1.0857 \left( \frac{\Delta C_{ap}}{C_{ap}} \right)$$  

(7)

where ExpTime is the exposure time and $ZP$ is the zero-point.

7.1.1. Undersampled PSF

As already noted in Strampelli et al. (2020), due to the fact that the WFC3-IR PSF is highly undersampled, most of the flux from a faint candidate companion is contained within just a few pixels of the residual tile. Also, the number of pixels at one’s disposal is so small that it is not possible to assess very precisely the star location and the centering of the aperture is usually rather poor. Thus, to derive the total flux, one has to apply a large and rather uncertain aperture correction. To address this problem, the pipeline performs photometry of the companion using a 4 pixel aperture and uses the processed data to establish a relative aperture correction. To this purpose, for each isolated source in the input catalog the routine determines the brightest $2 \times 2$ pixel area including the brightest pixel of the original image. After probing the four possible area positions, the routine selects the one providing the highest total counts, $C_{4p}$. The known magnitude of the source is then related to the 4 pixel counts through the equation

$$m_{4p} = -2.5 \log_{10}(C_{4p}) + \Delta$$  

(8)

where $\Delta$ is a zero-point term relating the 4 pixel sum to the known source photometry. Repeating this calculation over multiple “standard” stars, the final $\Delta$ value is determined as the $3\sigma$ cut median of the full set, with an uncertainty $\sigma_\Delta$ given by the associated standard deviation.

Once the final $\Delta$ has been determined, a similar 4 pixel aperture is applied to the residual tile. Equation (8), with the newly evaluated $C_{4p}$ for the companion and the final $\Delta$ value, thus provides the magnitudes of the candidate companions. The associated uncertainties are similar to those evaluated for a generic aperture photometry (see introductory paragraphs of Section 7.1) with the added uncertainty on the aperture correction. Note that in estimating the 4 pixel aperture photometry of a companion, the sky refers to the measured background in the residual tile.

7.2. MF Photometry

For well-sampled data, the pipeline offers a second method to extract photometry, MF photometry (Turin 1960). This approach solves the problem of detecting with the highest S/N a signal of known shape in noisy data. The solution is given by the cross-correlation between a known signal template (the reference PSF in this case) and an unknown noisy signal (the target). The pipeline implements this method through the photometry$_{AP}$ routine, which takes advantage of the MF routines present in the pyKLIP package (Ruffio et al. 2017). These perform convolution of the template with an image using the scpy signal processing library of fast Fourier transformations.

Figure 7 shows the different tiles used by the photometry$_{AP}$ routine. First, the sky is evaluated using the same approach adopted by the photometry$_{AP}$ routine, adopting an annulus between $r_g$ and $r_b$ (Sky annulus tile in Figure 7). Also in this case, the sky estimate can be further refined using the growth curves as explained in Section 7.1. The sky is then subtracted from the Data tile to produce a sky-subtracted target tile (or just Target in Figure 7). This tile is convolved with the reference tile, i.e., the PSF tile obtained for example from Tiny Tim,16 and the output of this operation is stored in the MF output tile.

The counts of the star can then be recovered by dividing the brightest pixel in the MF output by the throughput (i.e., the normalization factor that will rescale the count MF output to the real flux of the star):

$$C_{MF} = \frac{\max (\text{MF output})}{\text{throughput}}$$  

(9)

$$\text{throughput} = \left| \sum_{i,j} \text{abs(PSF}_{i,j})^2 \right|$$  

(10)

where PSF represents the input template tile. To estimate the uncertainties on $C_{MF}$, the routine follows the same Equations (3) to (5) used for photometry$_{AP}$ with the difference that since no aperture is used in this photometry, a “noise-equivalent area” needs to be defined to derive the number of pixels ($N_{ap}$) contributing to the uncertainty estimate.

Following King (1983), the effect of the background, $B$, can be computed by adding it over an equivalent area whose dimensions depend on the size and shape of the PSF. The pipeline uses King’s approximate rule of thumb for the equivalent-noise area

$$\sigma_B^2 = 8\pi a^2 B$$  

(11)

where $a \sim 0.5\lambda/D$ is half the diffraction-limited angular resolution of the telescope at the effective wavelength of the filter in use. Future versions will include the more rigorous results presented by King (1983) in tabular form. Once $\sigma_B^2$ is evaluated, it can be combined with the Poisson noise associated with

16 Tiny Tim is a program that generates simulated HST PSFs. [https://www.stsci.edu/software/tinytim/].
with the brightness of the star as follows:

$$\Delta C_{\text{MF}} = \sqrt{C_{\text{MF}} + \sigma_B^2}.$$  \hfill (12)

Once $C_{\text{MF}}$ and $\Delta C_{\text{MF}}$ are evaluated, the same Equations (6) and (7) can be applied.

### 7.3. MF Photometry of Binaries

In the case of a close binary system, where aperture photometry fails due to the presence of both stars in the extraction aperture, the MF photometry technique can be easily applied to retrieve the flux of the primary and companion separately. With reference to the tiles shown in Figure 8, to retrieve the flux of the primary, the routine subtracts the residual tile from the input tile (Data) obtaining a new tile (target) where the primary is isolated and the photometryAP routine can work as explained in the previous subsection. For the companion, the routine instead evaluates and subtracts the $3\sigma$ cut median background in the residual tile producing a new tile (target2) for photometryAP.

### 7.4. PSF Photometry

The third option to perform photometry of well-sampled data is PSF photometry. The photometryPSF routine performs PSF fitting making extensive use of the BasicPSFPhotometry package from photutils.\(^\text{17}\) As the initial guess for the coordinates and flux, photometryPSF evaluates a centroid on the tile and uses the flux estimated by photometryAP. Once the flux of the star has been evaluated through PSF fitting, photometryPSF takes advantage of equations similar to Equations (11) and (12) to estimate the uncertainties on the counts. Then it converts counts and errors to magnitude units using Equations (6) and (7). Moreover, the reduced $\chi^2$ is evaluated for each performed photometry as

$$\chi^2 = \frac{1}{m - n} \sum_{i,j} \left( \frac{x_{i,j} - \mu_{i,j}}{\sigma_{i,j}} \right)^2 = \frac{1}{m - n} \sum_{i,j} \left( \frac{\text{Residual}_{i,j}}{\text{eData}_{i,j}} \right)^2$$  \hfill (13)

where $m$ is the number of pixels in the tile; $n$ is the number of free parameters for the fit, typically $n = 3$ fitting both the positions $i,j$ and the flux; and $x_{i,j} - \mu_{i,j}$ is the difference between the observable $(x_{i,j})$ and the model $(\mu_{i,j})$. A residual tile is produced by photometryPSF, together with a tile comprising the pixel-related uncertainties on the flux of the observable, corresponding to the eData tile created during step 4 of the pipeline—see Section 5 and Figure 9 for a graphical example.

### 7.5. PSF Photometry of Binaries

As in the case of MF photometry, the photometryPSF routine can be utilized to estimate the flux of both members of a binary system once it has been split into its two components through KLIP. The routine will perform steps similar to photometryAP, as shown in Figure 9, evaluating the flux for both primary and companion through PSF fitting instead of performing a convolution. As explained before, the PSF residual tiles can be utilized to evaluate the goodness of the performed fit for both the primary and the companion.

\(^{17}\)https://photutils.readthedocs.io/en/stable/index.html
7.6. Pipeline Photometry Performance

Even though the photometryAP package is included in the pipeline, it is not fully integrated yet, so we advise the user to use it with caution until a new version of the pipeline is released. In the following, we will limit our tests only to the photometryAP and photometryPSF routines.

The performance of the last two photometry suites has been tested for the HST/ACS instrument simulating isolated stars and binaries with different separations and magnitudes. Given our primary scientific interest in the Orion Nebula Cluster (ONC), we focus on systems that could be plausibly observed in the ONC. The results obtained with aperture, MF, and PSF photometry are analyzed to determine photometric accuracy and the possible presence of systematic errors.

Similar tests have been conducted also for the branch of the pipeline working with HST/WFC3-IR data. Since they have been extensively presented in Strampelli et al. (2020), in the following paragraphs we will mainly focus on HST/ACS data simulations. Given the better spatial sampling of ACS versus WFC-IR, our results may provide a more accurate assessment of the advantages and limitations of each method.

7.6.1. Performance on Isolated Stars

To test the performance of the photometryAP and photometryPSF routines, thousands of isolated stars have been generated to closely resemble the stars present in the ONC both in flux and in position on the detector.

Using Tiny Tim, we create 50 PSF data cubes in five different target filters—F435W, F555W, F658N, F775W, and F850LP—with the sources equally spaced on both chips of the ACS/WFC focal plane. Each PSF data cube is created assuming effective temperatures between 2000 and 3000 K in steps of 250 K, appropriate to sample the lower end of the temperature range of young low-mass companions and M-dwarfs.

Figure 8. Top row: Input data tile (left), residual tile after PSF subtraction (center), and sky annulus (right). Central row: Isolated sky-subtracted primary (left) and isolated residual-background-subtracted companion (right). Bottom row: PSF reference tile from Tiny Tim and MF output tile for primary and companion. The blue (red) dot marks the center (maximum) of the MF output tile.
Each PSF data cube comprises 25 different subpixel-shifted PSFs created with the standard Tiny Tim SUB = 5 option, which splits each native pixel into 25 subpixels. When the SUB option is enabled, Tiny Tim does not convolve the PSF with the charge diffusion (CD) kernel. Following Hoffmann & Anderson (2017) we apply the CD kernel to the oversampled PSF, before shifting the PSF by a finite number of subpixels, and rescale to the native resolution. This operation consists in applying the CD kernel to subsets of the PSF tile that possess the same pixel phase. Arranging the subsampled pixels in this way places neighboring pixels in an order similar to the order of the natural ACS resolution. Then, each sample of subsampled pixels with the same pixel phase is convolved with the CD kernel. This process is repeated 25 times to cover all possible pixel phases in the 5 × 5 subsampled PSF.

Having convolved by the CD kernel the subsampled PSF, a subpixel shift has been applied to center the PSF in each of the 25 subpixels and the PSF is resampled to the native ACS resolution producing 25 slightly different shifted PSFs for each temperature, chip, and position on the CCD.

To keep these simulations as close as possible to reality, the stars have been generated covering for each filter a magnitude range matching the Robberto et al. (2013) catalog of ONC sources, constrained to prevent stars from saturating or being too faint to be detected given the exposure time. The stars have then been divided into bins with widths equal to 1 mag. Once a magnitude bin is selected, a random magnitude is extracted (we will refer to this reference magnitude as Mag) and converted into temperature using a 1 Myr isochrone from the BT-Settl family of models. Then a random subpixel shift is generated for the artificial target and an on-sky source’s coordinates are randomly selected from the Robberto et al. (2013) catalog. The coordinates are chosen so that all simulated sources are equally distributed both on the sky and on the detector. The CCD and the coordinates of the catalog source, as well as the random pixel phase and the magnitude-dependent temperature, are then used to select the closest PSF in the PSF data cubes previously generated. This PSF is rescaled to the selected flux using the standard HST zero-points and the background is added to each pixel. This last step is performed using make_noisy_image from photutils, which extracts each background pixel value from a Gaussian distribution with a mean given by the 3σ cut median of the sky evaluated at the source’s real on-sky coordinates and a standard deviation given by the corresponding scatter. To conclude this process, Poisson noise is added to the final tile using apply_poisson_noise from photutils as well.

The simulated star is then passed to the photometryAP and photometryPSF routines to extract photometry. Because for each position multiple PSFs are generated with subpixel shifts, photometryPSF subtracts from the target all subpixel-shifted PSFs (normalized by the total flux in the tile), choosing the one that minimizes the residual of this operation as the reference PSF. This process has been repeated for each filter on a sample of ∼1500 artificial stars.

Figure 10 and Table 2 show, for each filter, the output of the PSF photometry and aperture photometry routines as compared to the known input. The discrepancies have been analyzed by finding the average value of the three delta magnitude distributions, i.e., Mag – MagAP versus MagAP, Mag – MagMF versus MagMF, and Mag – MagPSF versus MagPSF, to find the presence of systematic shifts as compared to the known input.

The main result is that PSF photometry (right panel in the figure) offers the best trade-off between accuracy and precision, usually providing a smaller delta with smaller uncertainties.

No system is perfect though, so even if the routines try to account for missing flux due to their use of a finite aperture (apertureAP) the limited extent of the PSF used for the fit (aperturePSF) and the imperfect estimate of the background (despite having applied for all the three systems the growth curve correction) cause an offset between the input flux and the measured value. Evaluating these deltas as we have done in our specific test can help mitigate this problem (see Table 2), providing an insight about when to correct real data or not, in cases where the input magnitudes of the sources are unknown. Comparing the results in Table 2, in our simulation we observe that photometryAP and photometryPSF yield systematic errors that overall are smaller than the uncertainties on the deltas. Therefore, trying to correct for this difference in these two cases might be counterproductive, since the correction of a small systematic delta can introduce an even larger uncertainty.

The corrections take the form

\[ m_{\text{corr},i} = m_i + \Delta i \]

where \( m_i \) and \( m_{\text{corr},i} \) are the measured and corrected magnitudes, and \( \Delta i \) is the average shift of the measures from the expected value by the PSF/aperture photometry tools. The associated uncertainties can be obtained by adding the spread of the simulations to the magnitude error as

\[ \sigma_{\text{corr},i} = \sqrt{\sigma_i^2 + \sigma^2} \]

where \( \sigma_i \) represents the spread of the measures at a given magnitude bin.

7.6.2. Performance on Binary Systems

As in the case of isolated stars, the photometryAP and photometryPSF routines have been tested in the presence of binaries to understand their overall performance. As in the previous case, 1500 pairs have been simulated for each filter and for a range of separation between 3 and 10 pixels, in steps of 1 pixel. The procedure to generate a binary is in many ways similar to the one for an isolated star: two PSFs are selected and rescaled to match the flux of the primary and companion. Then, the companion PSF is injected into the tile of the primary at the given distance, with random orientation. Once the companion is injected, both background and Poisson noise are added as explained in Section 7.6.1.

The final tile containing the binary system is then parsed to the routine that performs PSF subtraction. The residual tile is then subtracted to obtain a tile with the isolated primary while the residual tile itself contains the companion. Both tiles are then parsed to the photometryAP and photometryPSF routines and the output is compared to the original known flux of the primary and companion. Even though this is primarily a test for the photometry of the pipeline, it also yields a second relevant result: since we know a priori the position of the injected companion, one can assess the performance of the pipeline in detecting a real companion. If the companion is detected at a position that does not coincide, within 1 pixel, with the

18 https://photutils.readthedocs.io/en/stable/api/photutils.datasets.make_noisy_image.html
19 https://photutils.readthedocs.io/en/stable/api/photutils.datasets.apply_poisson_noise.html
coordinates of the injection, one possesses a false positive, and the photometry must be rejected. Analyzing the rate of success and rejection, one can understand the overall performance of the pipeline to actually find companions as a function of the configuration of the binary system (i.e., the filter, the magnitude of the primary, the difference in magnitude between companion and primary, the separation, the KLIP mode selected for the subtraction, etc.). As discussed in the next section, this analysis can be used to estimate the rate of false-positive and true-positive detections as a function of the configuration of the binary, and help strengthen any decision about detecting a companion or rejecting a candidate.

For the sources correctly identified, we perform a study similar to the one previously presented for isolated stars for both primaries and companions. But in contrast to the case of isolated stars, the analysis of binaries depends on a much wider group of variables. One needs to consider the deltas $\text{Mag}_c - \text{Mag}_s$ versus the contrast (i.e., the difference in magnitude between companion and primary), where $\text{Mag}_c$ represents the output of one of the photometric routines applied on the isolated primary star itself. Second, each distribution must be evaluated for each filter, for each separation, and for the KLIP mode utilized to truncate the Karhunen-Loève transformation. We summarize the results in Figures 11 and 12, showing the outcome of the constant fit applied on each distribution as a function of the separation, with error bars representing the relative uncertainty on each fit. Each column refers to a different filter while each row refers to a different KLIP mode.

As expected, when a close companion (separation $\leq 0\arcsec 2$) is subtracted from a binary the outcome of the photometry$_{\text{AP}}$ and photometry$_{\text{PSF}}$ routines is worse than for wider binaries (separation $>0\arcsec 2$) due to the increasing overlap between the two PSFs.

For wide binaries, as in the case of isolated stars, photometry$_{\text{PSF}}$ appears more precise compared to photometry$_{\text{AP}}$. However, both photometry$_{\text{AP}}$ and photometry$_{\text{PSF}}$ show problems in correctly evaluating the
photometry of an isolated primary when a companion is found very close to the hosting star (separation $\lesssim 0''2$) due to an uncompleted subtraction of the close-in companion, which in turn leads to an overestimate of the isolated primary. This trend is reflected in the isolated companions, where both routines tend to underestimate the flux of close-in companions, leading to an underestimate of the isolated companion. In this case, smaller KLIPmodes are able to achieve better results than bigger KLIPmodes, when dealing with close-in companions. Moreover, due to the finite dimensions of the tiles we use in the simulation, when a companion is very close to the border of the tile (separation $\gtrsim 0''8$), photometry$_{AP}$ often fails to

Figure 10. The plots show the deltas between the known $\text{Mag}_k$ and $\text{Mag}_x$, where $\text{Mag}_x$ represents the output of one of the photometric routines applied to isolated stars: aperture photometry (left) and PSF photometry (right). Each row shows the results for each filter (F435W, F555W, F658N, F775W, and F850LP) analyzed in this simulation. The constant fit is shown as a blue dashed–dotted line while the black dashed line shows the locus of points where $\text{Mag}_k = \text{Mag}_x$. Note that the $Y$-axis range may differ from one technique to the other.

| Filter   | $b_{ap}$       | $b_{psf}$    | $b_{ap}$       | $b_{psf}$    | $b_{ap}$       | $b_{psf}$    |
|----------|---------------|--------------|---------------|--------------|---------------|--------------|
| F435W    | $0.003 \pm 0.061$ | $-0.006 \pm 0.143$ | $-0.007 \pm 0.262$ | $-0.010 \pm 0.106$ | $-0.004 \pm 0.099$ |
| F555W    | $0.001 \pm 0.017$ | $0.001 \pm 0.019$  | $-0.000 \pm 0.246$ | $-0.000 \pm 0.102$ | $0.000 \pm 0.095$ |
recover the correct flux of the isolated companion, while photometry_{PSF} is able to correctly reconstruct the flux of the isolated companion.

8. Step 9: False-positive Analysis

In this section we present the infrastructure developed to characterize the detection reliability. This final step relies heavily on the simulation of isolated stars and binaries described in Section 7.6, where we have shown how to inject artificial companions and isolated sources.

Once a library of both singles and binaries has been created, it can be passed to the pipeline for analysis. It is important to note that each binary system is linked to a corresponding single source tile by the same primary star, i.e., the tiles are created in tandem: one with a single source, and one with the same source plus an injected companion with random flux and a random position \(i,j\). In this way the pipeline can measure the flux at the injected position \(i,j\) both for the isolated source (building the null hypothesis) and for the linked binary (building the test hypothesis). As shown in Strampelli et al. (2020), one can use an ROC (Fawcett 2006) curve as a tool to assess the ability of a binary classification.
system; the discrimination threshold $T$ provides an estimate of the probability of having false positives as a function of the particular configuration of the binary (i.e., the filter under examination, the magnitude of the primary, the delta magnitude between companion and primary, the KLIPmode utilized to detect the companion, and the separation between the two).

The pipeline can build ROC curves and provide the true-positive and false-positive populations representative of each candidate. These curves show how the achieved sensitivity strongly depends on the configuration of the binary: the magnitude of the primary ($m_{F130N}$), the contrast ($\Delta$mag) achieved by PSF subtraction, the distance of the companion from the primary (separation), and the KLIPmode utilized during the PSF subtraction.

To encapsulate in a single number the performance of our model to distinguish between classifiers, the pipeline will evaluate the area under the curve (AUC) of an ROC. The larger the AUC, the better the model is at distinguishing between the

Figure 13. Distributions of S/N and derived ROC curves for filter F130N; magnitude bins of the primary 11–12; $\Delta$mag 0, 4, and 6; and different distances from the center of the tile. From Strampelli et al. (2020).
true-positive population and the false-positive population. An AUC of 1 indicates a perfect classifier.

Figure 13 shows examples of true-positive (blue) and false-positive (orange) histograms for a given binary configuration, and the corresponding ROC curves. Also provided for each ROC curve is the value of the corresponding AUC. For a specific application of this methodology to WFC3-IR data, see Strampelli et al. (2020).

We need to stress that this step does not provide a direct false-positive rejection, as it only focuses on the simulated sources and knows nothing about the real ones: it does not separate single false positives from true detections in the sample, but only provides an estimate of how probable it is that a specific configuration can produce false positives in the simulations. In other words, its only scope is to associate with each configuration of a binary system the probability of an outcome of a false-positive detection, which then the user can use in order to make an informed decision if a companion is plausibly a false positive or is probably real. A series of ancillary routines has been developed to perform this analysis, as shown in the two scientific applications of the pipeline. For a practical application of this analysis, we highly encourage the reader to look at Strampelli et al. (2020), where this approach is deeply discussed in relation to HST/WFC3-IR data.
9. Test with Real Data

In Strampelli et al. (2020) we analyzed the HST/WFC3-IR data for the HST Treasury Program on the ONC (GO-10246) while in G. M. Strampelli et al. (2022, in preparation) we performed a similar analysis focused instead on the HST/ACS data obtained for the HST Treasury Program on the same cluster (GO-10246). Here in the following we will briefly summarize some of the results from those papers to show the performance of our pipeline, but we still refer the reader to the specific papers to obtain more details.

Figure 14 shows a comparison between seven detections from the WFC3-IR data and the corresponding detections in the ACS survey (either they are already present in the input catalog or they are found by PSF subtraction). Note that due to the different filters employed in the two surveys, sometimes the primary may vary, or the companion may not be visible at all. Overall we find very good agreement between the detections in the two surveys.

9.1. Completeness and Multiplicity Fraction

Figure 15 shows a comparison between the completeness achieved in the WFC3-IR survey (on the left) and in the ACS survey (on the right). As expected, the IR survey allows us to detect lower-mass companions, reaching a $q \sim 0.025$ at $\sim30\%$ completeness and separations of $\sim100$ au (or a semimajor axis (SMA) $\sim0.8$), while the ACS survey allows us only to push less deep in contrast and reach only $q \sim 0.05$ for similar ranges. On the other hand though, the ACS surveys allow us to go deeper inside the system and closer to the primary, achieving completeness $\sim30\%$ already at $\sim80$ au (or SMA $\sim0.2$), compared to the $\sim100$ au (or SMA $\sim0.25$) for WFC3-IR, and a minimum completeness of $\sim10\%$ at $\sim40$ au (or SMA $\sim0.1$), compared to the $\sim60$ au (or 0.15 SMA) for WFC3-IR.

Figure 16 in contrast shows the mass ratio distribution for our candidates as a function of the mass of the primary. Overall, in the specific case of the ONC, we are able to detect brown-dwarf-mass companions down to the limit of planetary-mass and three-planetary-mass companions, reaching a minimum separation of $\sim70$ au ($\sim0.17$ SMA) at a completeness of $\sim22\%$.

In both Strampelli et al. (2020) and G. M. Strampelli et al. (2022, in preparation) we combined the observed companion population for the WFC3-IR and ACS instruments with the corresponding ROC curves and false-positive analysis to obtain an unbiased multiplicity fraction in the ONC. The two works reported remarkably close results:

$$MF_{\text{WFC3-IR}} = 10.6\% \pm 0.6\%$$
$$MF_{\text{ACS}} = 10.2\% \pm 0.9\%.$$
10. Conclusion

A new pipeline has been developed to detect and characterize faint astronomical sources close to their host star. It relies on Karhunen–Loève truncated transformation theory to perform PSF subtraction. Using a highly adaptable series of routines, the pipeline uses standard imaging data and conventional photometry to deliver a catalog of candidate close companions, their photometry with associated uncertainties, and a robust statistical estimate of their false-positive probability. At the current stage of development Equation (28), as in the case of, e.g., hierarchical triple systems, but only the brightest candidate companion in the residual tile will be identified. Extension to the case of multiple sources will be the subject a future version.

Even though this pipeline is based on methods designed for observations of a single star at a time, we can achieve remarkable results on mosaic space-based data. In fact, in the case of the ONC, the pipeline is able to reliably detect very low-mass companions, down to the brown-dwarf-mass limit and almost to the planetary-mass limit. The pipeline is also able to detect signals as close as $\gtrsim 0''1$ (or $\sim 40$ au) at the distance of the ONC with a completeness $\gtrsim 10\%$, or $\sim 0''2$ ($\sim 80$ au) with a completeness of $\sim 30\%$. This approach can potentially be applied to a wide variety of space-based imaging surveys, from the existing HST archive, to near-future JWST mosaics, to future wide-field Roman images.

G.M.S. wants to thank Instituto de Astrofísica de Canarias for their hospitality. The authors thank Carlo Manara for the excellent insights. Support for program Nos. GO-10246 and GO-13826 was provided by NASA through a grant from STScI, which is operated by the Association of Universities for Research in Astronomy, Incorporated, under NASA contract WFC, Instrument Science Report ACS 2017-8.

Acknowledging support by Ministerio de Ciencia, Innovación y Universidades of Spain (grant AYA2017-89841-P) and by Instituto de Astrofísica de Canarias.

Software: AstroSCRAPPY (McCully et al. 2018), DOLPHOT (Dolphin 2000), PyKLIP (Wang et al. 2015), Pandas (McKinney 2010), Photutils (Bradley et al. 2020).

ORCID iDs

Giovanni M. Strampelli @ https://orcid.org/0000-0002-1652-420X

Jonathan Aguilar @ https://orcid.org/0000-0003-3184-0873

Antonio Aparicio @ https://orcid.org/0000-0002-6054-0004

Gaspard Duchêne @ https://orcid.org/0000-0002-5902-6464

Massimo Robberto @ https://orcid.org/0000-0002-9573-3199

References

Amara, A., & Quanz, S. P. 2012, MNRAS, 427, 948

Biller, B. A., Close, L., Lenzen, R., et al. 2004, Proc. SPIE, 5490, 389

Bratey, L., Sipőcz, B., Robitaille, T., et al. 2020, astropy/photutils: v1.0.0, Zenodo, doi:10.5281/zenodo.4044744

Brown, A. G. A., & Verschueren, W. 1997, A&A, 319, 811

Dolphin, A. E. 2000, PASP, 112, 1383

Fawcett, T. 2006, PaRel, 27, 861

Gaia Collaboration, Brown, A. G. A., Vallenari, A., et al. 2018, A&A, 616, A1

Hoffmann, S. L., & Anderson, J. 2017, A Study of PSF Models for ACS/ WFC, Instrument Science Report ACS 2017-8

King, I. R. 1983, PASP, 95, 163

Köhler, R., Petr-Gotzens, M. G., McCaughrean, M. J., et al. 2006, A&A, 458, 461

Kouwenhoven, M. B. N., Brown, A. G. A., & Kaper, L. 2007, A&A, 464, 581

Kraus, A. L., Ireland, M. J., Martinache, F., & Hillenbrand, L. A. 2011, ApJ, 731, 8

Kraus, A. L., Ireland, M. J., Martinache, F., & Lloyd, J. P. 2008, ApJ, 679, 762

Lafrenière, D., Marois, C., Doyon, R., Nadeau, D., & Artigau, É. 2007, ApJ, 660, 770

Luhman, K. L., McLeod, K. K., & Goldenson, N. 2005, ApJ, 623, 1141

Marois, C., Doyon, R., Racine, R., et al. 2005, JRASC, 99, 130

Masciadri, E., Mundt, R., Henning, T., Alvarez, C., & Barrado y Navascués, D. 2005, ApJ, 625, 1004

Mawet, D., Milli, J., Wahhaj, Z., et al. 2014, ApJ, 792, 97

McCully, C., Crawford, S., Kovacs, G., et al. 2018, astropy/astroscapepy: v1.0.5, Zenodo, doi:10.5281/zenodo.1482019

McKinney, W. 2010, in Proc. 9th Python in Science Conf., ed. S. van der Walt & J. Millman (Austin, TX: SciPy), 56

Pet, M. G., Coudé du Foresto, V., Beckwith, S. V. W., Richichi, A., & McCaughrean, M. J. 1998, ApJ, 500, 825

Reipurth, B., Guimarães, M. M., Connelley, M. S., & Bally, J. 2007, ApJ, 134, 2272

Ren, B., Pueyo, L., Zhu, G. B., Debes, J., & Duchêne, G. 2018, ApJ, 852, 104

Robberto, M., Soderblom, D. R., Bergeron, E., et al. 2013, ApJS, 207, 10

Ruffio, J.-B., Macintosh, B., Wang, J. J., et al. 2017, ApJ, 842, 14

Schneider, G., & Silverstone, M. D. 2003, Proc. SPIE, 4860, 1

Shatsky, N., & Tokovinin, A. 2002, A&A, 382, 92

Soummer, R., Pueyo, L., & Larkin, J. 2012, ApJL, 755, L28

Stetson, P. B. 1987, PASP, 99, 191

Stolker, T., Bonse, M. J., Quanz, S. P., et al. 2019, A&A, 621, A59

Strampelli, G. M., Aguilar, J., Pueyo, L., et al. 2020, ApJ, 896, 81

Turin, G. 1960, IRE Transactions on Information Theory, Vol. 6 (New York: IEEE), 311

van Dokkum, P. G. 2001, PASP, 113, 1420

Van Reussum, G., & Drake, E. L. Jr. 1995, Python reference manual Centrum voor Wiskunde en Informatica Amsterdam

Wang, J. J., Ruffio, J.-B., De Rosa, R. J., et al. 2015, pyKLIP: PSF Subtraction for Exoplanets and Disks, Astrophysics Source Code Library, ascl:1506.001