Draft version September 20, 2019
Typeset using LaTeX **twocolumn** style in AASTeX62

Implementing the PACO Algorithm as a Pynpoint Module.

Evert Nasedkin[1] and Polychronis Patapis[1]

[1] *ETH Zürich, Exoplanets and Habitability Group*

ABSTRACT

lorum ipsum stuff stuff stuff
blah b
blah

1. INTRODUCTION

The field of exoplanet astronomy has seen dramatic growth since the initial discovery from **?** in 19??. With the advent of high-contrast imaging techniques in ???? CITE, we have been able to observe exoplanets in regimes inaccessible through other detection methods. Through the combination of dedicated instrumentation CITE(NaCo,Sphere,GPI), adaptive optics CITE and data processing techniques, we have been able to observe around 12,13? companions as of 2019, with the hope for many more discoveries following the completion of next generation observatories such as JWST and the ELT.

This report focuses on the implementation and testing of the Patch Covariance, or PACO algorithm for point source detection. PACO looks for spatio-temporal correlations in data taking in ADI mode, and offers a statistical interpretation of the detection maps.

1.1. *ADI*

Figure 1. Outline of standard angular differential imaging processing.

2. PACO STATISTICAL MODEL

There are two primary implementations of the PACO algorithm, based. First we will consider the full implementation, followed by the assumptions made to create the second fast algorithm. We will also discuss the addition of PACO to the Pynpoint analysis package.

The Patch Covariance, or PACO algorithm works by considering the known trajectory of a companion through an image stack taken with a pupil-stabilized telescope. It is important to note that while the companion will follow an arc through the image stack, speckle noise from the telescope will remain stationary. This allows us to distinguish the weak source from the much stronger background noise. The following section will closely follow the presentation of **?**, with some changes to notation to aid with clarity.

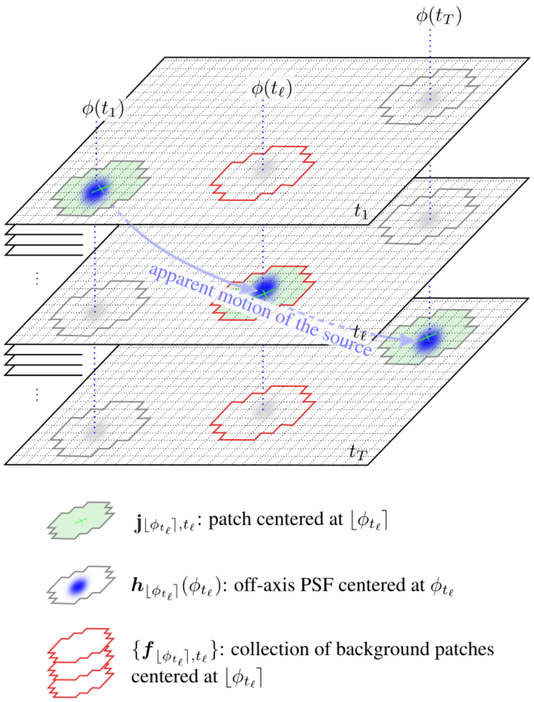

$\phi(t_T)$
$\phi(t_\ell)$
$\phi(t_1)$
t_1
apparent motion of the source
t_ℓ
t_T
$\mathbf{j}_{\lfloor\phi_{t_\ell}\rceil,t_\ell}$: patch centered at $\lfloor\phi_{t_\ell}\rceil$
$\boldsymbol{h}_{\lfloor\phi_{t_\ell}\rceil}(\phi_{t_\ell})$: off-axis PSF centered at ϕ_{t_ℓ}
$\{\boldsymbol{f}_{\lfloor\phi_{t_\ell}\rceil,t_\ell}\}$: collection of background patches centered at $\lfloor\phi_{t_\ell}\rceil$

Figure 2. Patches are selected from each temporal frame based on the hypothesis that a signal was located at an initial pixel, and follows a rotation through the parallactic angle through the frames.

Consider a stack of M temporal frames, where each frame contains N pixels. We will be considering the hypothesis, $\mathcal{H}_1$, that there is a planet located at the position of a given pixel to the background only hypothesis

Corresponding author: Evert Nasedkin
evertn@student.ethz.ch

Symbol	Description
M	Number of frames
N	Number of pixels per frame
K	Number of pixels per patch
θ	2D Pixel location
ϕ_t	Angular sub-pixel location in frame t
$[\phi_t]$	Angular location to nearest pixel
j	Intensity at a point
f	Intensity due to background at a point
α	Flux of a source at a specified location
$\mathbf{h}$	2D PSF template

Table 1. Summary of PACO notation.

$\mathcal{H}_0$. This comparison can be formalized as a likelihood ratio test.

We use the additional knowledge about the trajectory of the planet position through the image stack to determine whether there is a signal when compared to the background only. For a given initial angular position ϕ_0, this trajectory is a rotation through each temporal frame, which we write as $\mathcal{F}_t(\phi_0)$. The index t runs from 0 to M frames.

For $\mathcal{H}_1$, the observed intensity j at a given location θ will be the sum of the background intensity f, together with some intensity from the source. We will use square brackets $[\theta]$ to denote a pixel-centered location, as opposed to a true 2D coordinate θ. Also note that $[\theta]$ will be indexed per pixel as $[\theta]_n$ if necessary. For a single frame, the source is modeled as a template PSF $\mathbf{h}$ multiplied by source flux α, centered on the pixel $[\theta]$. Considering the PSF template at only at the location θ, we write the total intensity at that point as:

$$j(\theta) = \alpha h(\theta) + f(\theta) \tag{1}$$

Both the template PSF and the background vary with time, and we generalize Eqn. 1 using the 2D angular location at time t, denoted as $\phi_t = \mathcal{F}(\phi_0)$. As above, the closest on grid location will be notated with square brackets.

$$j(\theta, \phi_t) = \alpha h(\theta, \phi_t) + f(\theta, t) \tag{2}$$

We restrict α to be positive, and 0 in the absence of a source. As companions are relatively rare, we do not consider the case where two sources can overlap. If the background noise is distributed according to some PDF p_f, each observation can be treated as a random draw from this distribution, with a source of strength α overlaid. Thus the joint probability of drawing a particular collection of intensities $\{j(\theta_k, t)\}$ at a locations $[\theta]_{k=0:N}$ and times $t = 0 : M$ with a source of flux α can be written as:

$$p_j(\{j\}|\alpha, \phi_t) = p_f(\{j - \alpha h(\theta, \phi)\}) \tag{3}$$

For each possible initial angular position ϕ_0, we can now construct an unbiased maximum likelihood estimator for the flux.

$$\hat{\alpha} = \text{argmax}\left(p_f(\{j - \alpha h(\theta, \phi_0)\})\right) \tag{4}$$

With this estimator, we can now construct the hypothesis test, choosing some threshold for detection, again with sets covering the collection of temporal frames $t = 0 : M$ and pixels $[\theta]_{k=0:N}$.

$$\mathcal{H}_0 : j = \{f\} \tag{5}$$

$$\mathcal{H}_1 : j = \alpha\{h(\theta, \phi)\} + \{f\} \tag{6}$$

Thus we are comparing the null hypothesis that the collection of M observations for each of the N pixels consists of only background fluctuations according to its PDF $f(\theta, \phi)$ to the alternative hypothesis $\mathcal{H}_1$ that a source of flux α and PSF $h(\theta, \phi)$ is overlaid on top of the same background. By substituting the estimator $\hat{\alpha}$ in place of the true flux α, we can write the generalized likelihood ratio test with threshold η as:

$$\log \frac{p_f(\{j - \hat{\alpha} h(\theta, \phi)\})}{p_f(\{j\})} \gtrless \eta \tag{7}$$

Figure 3. Patch shapes and sizes

To determine the background distribution, a 2D neighbourhood centered around each θ is chosen. This *patch* of pixels should be large enough to contain the core of the off axis PSF, and additionally contains the relevant background pixels for the specified location. While in general the shape of the patch is arbitrary, a 2D discrete disk is used in this work. Each temporal slice is considered to be an independent sample, and the number of counts per pixel is large enough to model Poisson counting statistics as normal distributions. This approximation allows us to write the background distribution in closed-form. This allows us to write the background distribution for the a patch $\mathbf{j}$ with the collection of backgrounds $\{\mathbf{f}\}_{t=0:M}$ as the product of multivariate normal distributions, defined over each patch that would contain an companion, following the trajectory of ϕ_t:

$$p_f\left(\{\mathbf{f}([\phi]_t)\}_{t=0:M}\right) = \prod_{t=0}^{M} \mathcal{N}\left(\mathbf{f}([\phi]_t, t), \mathbf{m}([\phi]_t), \mathbf{C}([\phi]_t)\right) \tag{8}$$

Where $\mathcal{N}\left(\cdot, \mathbf{m}([\phi]_t), \mathbf{C}([\phi]_t)\right)$ is athe probability density of the multivariate normal distribution with mean $\mathbf{m}$ and covariance $\mathbf{C}$ centered at pixel $[\phi_t]$. To determine the mean and covariance of this distribution, we consider the collection of patches centered on the same pixel $[\theta]$ without following the trajectory to find estimators for the sample mean and covariance. The sample mean is simply:

$$\hat{\mathbf{m}}([\theta]) = \frac{1}{M}\sum_{t=0}^{M} \mathbf{j}([\theta], t) \tag{9}$$

The estimator for the covariance must be regularized to account for the case where the covariance is rank deficient, i.e. when the number of frames is smaller than the number of pixels K in a patch. A shrinkage approach was used in , based on , and this approach was used in our implementation. In short, the estimator for the covariance is written as a combination of the estimator of the sample covariance plus the diagonal of the sample covariance weighted by a shrinkage factor $\hat{\rho}$:

$$\hat{\mathbf{C}} = (1-\hat{\rho})\hat{\mathbf{S}} + \hat{\rho}\hat{\mathbf{F}} \tag{10}$$

The estimators $\hat{\mathbf{S}}$ and $\hat{\mathbf{F}}$ are chosen such that one is biased but has small variance, while the other has large variance but minimal bias. The parameter $\hat{\rho}$ is chosen to optimize this trade off, and a full derivation can be found in . Each term can be computed as:

$$\hat{\mathbf{S}}(\theta) = \frac{1}{M}\sum_{t=0}^{M} \left(\mathbf{j}(\theta,t) - \hat{\mathbf{m}}(\theta)\right)\cdot\left(\mathbf{j}(\theta,t) - \hat{\mathbf{m}}(\theta)\right)^t \tag{11}$$

$$\left[\hat{\mathbf{F}}(\theta)\right]_{ii} = \frac{1}{M}\sum_{t=0}^{M} \left[\mathbf{j}(\theta,t) - \hat{\mathbf{m}}(\theta)\right]_{ii}^2 = \left[\hat{\mathbf{S}}(\theta)\right]_{ii} \tag{12}$$

$$\hat{\rho}\left(\hat{\mathbf{S}}(\theta)\right) = \frac{\operatorname{tr}\left(\hat{\mathbf{S}}^2(\theta)\right) + \operatorname{tr}^2\left(\hat{\mathbf{S}}(\theta)\right) - 2\sum_{i=1}^{K}\left[\hat{\mathbf{S}}(\theta)\right]_{ii}^2}{(M+1)\left(\operatorname{tr}\left(\hat{\mathbf{S}}^2(\theta)\right)\right) - \sum_{i=1}^{K}\left[\hat{\mathbf{S}}(\theta)\right]_{ii}^2} \tag{13}$$

2.1. *Flux Estimation*

From the multivariate Gaussian model of the background in Eqn. 8 we find that the maximum likelihood estimator for the flux at a given location can be written as:

$$\hat{\alpha} = \frac{\sum_{t=0}^{M} b_t}{\sum_{t=0}^{M} a_t} \tag{14}$$

where

$$a_t = \mathbf{h}([\phi]_t)^t \cdot \hat{\mathbf{C}}^{-1}([\phi]_t) \cdot \mathbf{h}([\phi]_t) \tag{15}$$

and

$$b_t = \mathbf{h}([\phi]_t)^t \cdot \hat{\mathbf{C}}^{-1}([\phi]_t) \cdot \left(\mathbf{j}([\phi]_t, t) - \hat{\mathbf{m}}([\phi]_t)\right) \tag{16}$$

As the flux is required to be positive, b is constrained to be greater than or equal to 0.

$$\hat{\alpha}^{+} \equiv \frac{\max\left(\sum_{t=0}^{M} b_t, 0\right)}{\sum_{t=0}^{M} a_t} \tag{17}$$

? also shows that the variance of $\hat{\alpha}$ is

$$\operatorname{Var}(\hat{\alpha}) = 1/\sum_{t=0}^{M} a_t = 1/a \tag{18}$$

2.2. *Detection*

With an expression for the flux based on the multivariate Gaussian model of the background, we can now perform the likelihood ration test from Eqn. 7.

$$\frac{\left(\sum_{t=0}^{M} b_t\right)^2}{\sum_{t=0}^{M} a_t} \gtrless \eta \tag{19}$$

Or equivalently as a signal to noise test:

$$\frac{\sum_{t=0}^{M} b_t}{\sqrt{\sum_{t=0}^{M} a_t}} = \frac{\hat{\alpha}}{\hat{\sigma}_\alpha} \gtrless \tau \tag{20}$$

The signal to noise will be Gaussian distributed as the test statistic $\hat{\alpha}/\hat{\sigma}_\alpha$ depends linearly on the data, allowing for a determination of the false positive fraction or a p-value. The aim of PACO is to compute this SNR value for each pixel, which will produce a detection map from which the presence of a companion can be inferred.

2.3. *Unbiased Flux Estimation*

Flasseur et al. also implemented an iterative algorithm for retrieving the flux at a known location, by iterating on learning the background and subtracting the best estimate of the flux until there is background only (similar to negative psf injection). General outline of algorithm, tracing a planet through its expected position based on the ADI rotation. Statistics: null hypothesis compared to hypothesis of signal + background. Compute sample covariance, and regularize. Learn the background statistics, and compare to a template psf. Take the ratio to compute the estimator for the flux. Show that a is the variance of ahat, and that can be used to define a SNR. This allows us to output a flux map in counts and an SNR map which can be used to define a true detection.

$$p(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{21}$$

3. PACO IMPLEMENTATION

3.1. *Full PACO*

Full PACO was implemented as described in section(STATISTICS). Following the outline of Flasseur, we can summarize the algorithm as:

```
FullPACO
```

3.2. *Fast PACO*

Assume background statistics are similar, and precompute inverse covariance matrices and means for each patch position. Parallelized

```
FastPACO
```

3.3. *Resolution Scaling*

Since a source is not guaranteed to be located at the center of a pixel, the resolution at which PACO runs can be scaled to allow for sub-pixel placement of the template PSF. This improves the correlation between the template and an actual signal, but increases the run time by scaling squared.

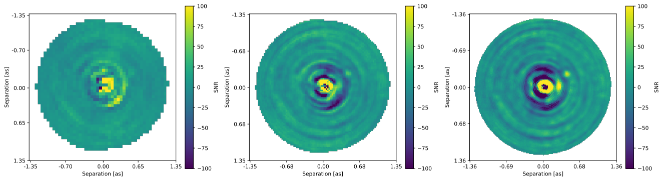


Figure 4. Typical example using β Pic b of an output from PACO for different resolution scaling. Left is 0.5 times scaling, centre is 1.0, right is 2.0. The SNR increase as the scaling increases, as the template PSF is able to be positioned more accurately over the source.

Figure 5. The iterative flux method is unbiased, and accurately retrieves the flux of an injected point source. The flux estimator is/isn't efficient, and provides greater/less precision than retrieval using negative source injection.

3.4. *Pynpoint Usage*

The PACO algorithm has been incorporated as a module into the Pynpoint analysis pipeline for exoplanet detection. Pynpoint modules are added to a pipeline, with intermediate results stored in a local database. Tutorials and examples can be found at XXXXXXX.

PACO can be run as a module following standard preprocessing of ADI data such as centering, cropping, dark frame subtraction etc. It can be added to the pipeline using the function
`pynpoint.processing.PACOModule(...)`

The module contains many user-adjustable parameters as detailed below.

`name_in`:

`image_in_tag`:

`psf_in_tag`:

`snr_out_tag`:

`flux_out_tag`:

`psf_rad`:

`scaling`:

`algorithm`:

`flux_calc`:

`threshold`:

`flux_prec`:

`verbose`:

The results (SNR maps, flux maps, point source coordinates and flux estimates) are stored in the Pynpoint database, and can be output using the standard Pynpoint IO functions.

PACO has also been added to Pynpoint's contrast curve module, where it can be selected as an option using the `algorithm` argument. Additional parameters

Runtime analysis: parallelized, but still basically unusable on large datasets.

4. RESULTS

4.1. *Test Data*

4.2. *Statistics*

5. COMPARISONS

6. DISCUSSION

7. CONCLUSIONS

ACKNOWLEDGMENTS

Sascha Quanz, Tomas Stolker, Anna Boehle, Tim ????, Natalia ???, O. Flasseur, ???? and ???? for writing the original algorithm

REFERENCES

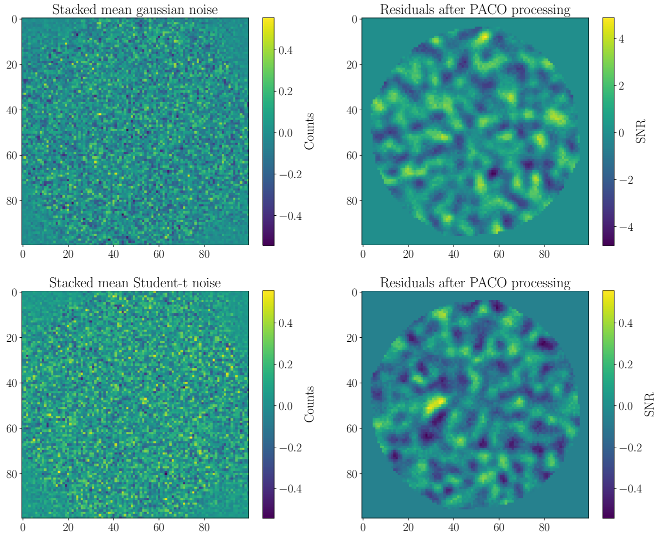
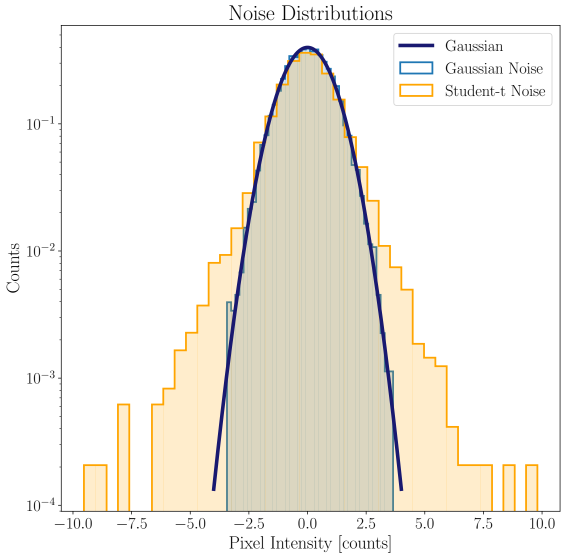


Figure 6. **Top**: On the left, random noise was drawn from a Gaussian distribution (mean 0, width 1), the shown image is the mean of the stacked and de-rotated image sequence. The right image shows the residuals from processing the image sequence with PACO. **Bottom**: The same as above, but with noise drawn from a Student-t distribution (5 degrees of freedom). The magnitude of the SNR is lower due to the wider distribution of the background noise. (see Fig. 7)



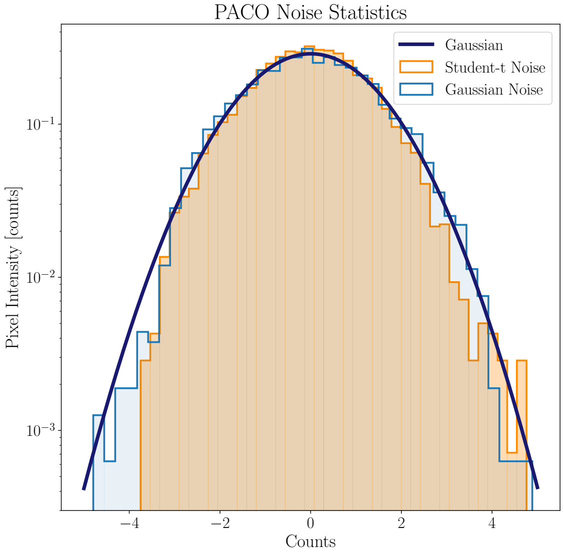


Figure 7. Testing PACO on Gaussian and Student-t distributed noise. PACO residuals for both are Gaussian distributed.

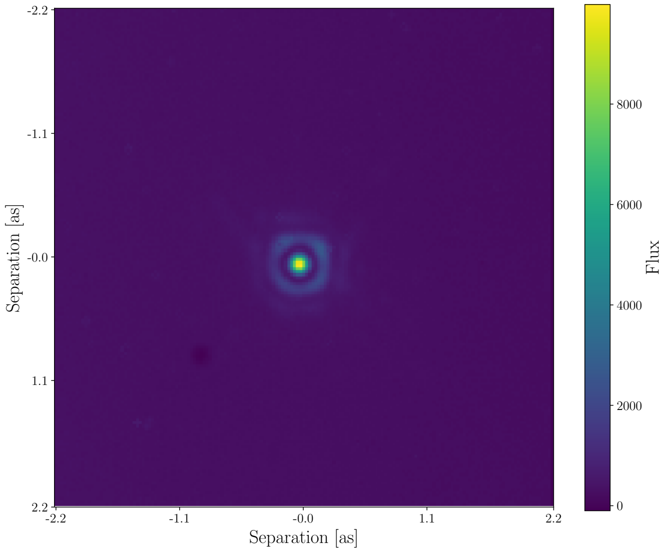


Figure 8. Example of single frame from HD131399 dataset used for testing.

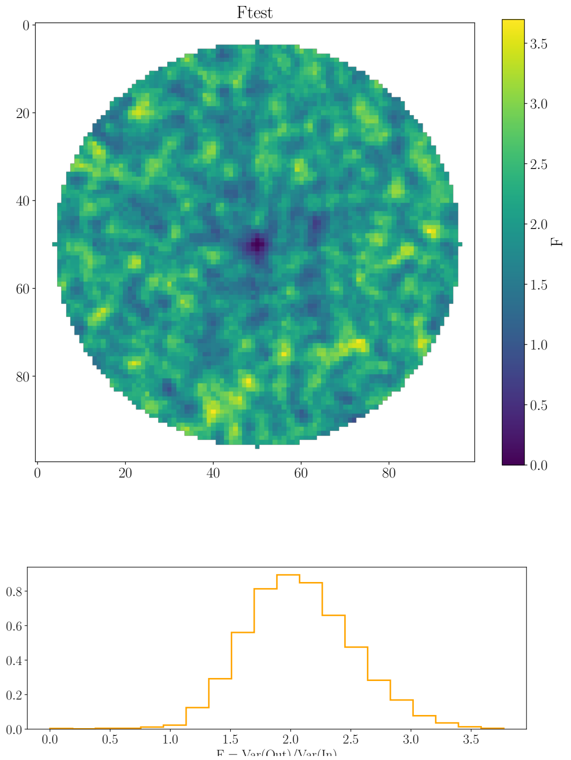


Figure 9. Here we show that the estimator for the variance of the flux accurately is a good estimator.

Figure 10. We can compute a contrast curve for a given dataset, either using standard aperture masking, or by using PACOs flux retrieval. This is compared with a contrast curve from Pynpoints PCA model on the same dataset for reference.

Figure 11. For sources injected at XXX sigma, PACO accurately retrieves YY% of the sources at the same significance.

Figure 12. We can also compare the true positive fraction against the false positive fraction using an ROC curve.