Grammar evolution and symbolic regression for astrometric centering of Hubble Space Telescope images

Sarmiento, R.
rasarmientor@hotmail.com
Universidad Internacional de la Rioja (UNIR)
26006. Logroño, Spain

de la Cruz, M.
marina.delacruz@unir.net
Universidad Internacional de la Rioja (UNIR)
26006. Logroño, Spain

Ortega, A.
alfonso.ortegadelapuente@unir.net
Universidad Internacional de la Rioja (UNIR)
26006. Logroño, Spain

Girard, T.M.
girardt2@southernct.edu
Southern Connecticut State University
New Haven, CT, USA

casetti-Dinescu, D.I.
casettid1@southernct.edu
Southern Connecticut State University
New Haven, CT, USA

Cervantes, A.
alejandro.cervantesrovira@unir.net
Universidad Internacional de la Rioja (UNIR)
26006. Logroño, Spain

Baena-Galle, R.
roberto.baena@unir.net
Universidad Internacional de la Rioja (UNIR)
26006. Logroño, Spain

ABSTRACT

Symbolic regression, in general, and genetic models, in particular, are promising approaches to mathematical modeling in astrometry where it is not always clear which is the fittest analytic expression depending on the problem under consideration. Several attempts and increasing research efforts are being made in this direction mainly from the Genetic Programming (GP) viewpoint. Our proposal is, as far as we know, the first one to apply Grammatical Evolution (GE) in this domain. GE (and further GE extensions) aim to outperform GP limitations by incorporating formal languages tools to guarantee the correctness (both syntactic and semantic) of the generated expressions. The current contribution is a first proof to check the viability of GE on astrometric real datasets. Its success in finding adequate parameters for predefined families of functions in star centering (Gaussian and Moffat PSFs) with simple and naive GE experiments supports our hypothesis on taking advantage of the expressive power of GE to tackle astrometry scenarios of interest and hence greatly improve current astrometric software thanks to specific genetic approaches.

KEYWORDS

Grammatical evolution, symbolic regression, astrometry, WFPC2, Hubble Space Telescope

ACM Reference Format:
Sarmiento, R., de la Cruz, M., Ortega, A., Girard, T.M., casetti-Dinescu, D.I., Cervantes, A., and Baena-Galle, R.. 2023. Grammar evolution and symbolic regression for astrometric centering of Hubble Space Telescope images. In *Proceedings of 46th International Conference on Software Engineering (ICSE 2024).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Grammar Evolution [17] (GE) is one of the most popular approaches to automatic evolutionary computation (AEC), which is a branch of genetic programming [12] (GP), proposed to use genetic algorithms [11] (GA) to generate programs in any programming language. Here, we explore using GE on a problem of symbolic regression (SR) to generate mathematical expressions that can describe the impulse response of an optical system, also known as the point spread function (PSF), which is usually sampled by the finite dimensions of electronic camera pixels. We have applied this approach to point-like sources images or stars from the Wide Field Planetary Camera 2 (WFPC2), installed at the Hubble Space Telescope (HST) until 2009 Figure 3. WFPC2 features 3 identical chips of 800 × 800 pixels (WF), plus one additional (PC) with the same number of pixels but of double spatial resolution. If machine learning regression is a type of problem in which the system must learn to produce a best value for each possible input, SR solves this problem by proposing a symbolic expression that generates the outputs. For understanding the principles of GE, some previous knowledge on context-free grammars (CFG) is needed.

CFG are the grammars studied at school when defining human language syntax. CFG are a formal model able to generate symbolic expressions, for example, using $+$ and $\times$ as operators, variables x and y, and any number as operands, $3 \times x + y \times x$ can be generated. These syntactic restrictions are formalized by specific rules, as natural as *an expression can be any specific number, but also a variable like x or*

ICSE 2024, April 2024, Lisbon, Portugal

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/XXXXXXX.XXXXXXX

y, or *either the addition or the multiplication of two expressions is also an expression.* These rules can be formally expressed respectively as $expr \longrightarrow x$, $expr \longrightarrow y$, $expr \longrightarrow number$, $expr \longrightarrow expr + expr$ or $expr \longrightarrow expr \times expr$.

A slightly different notation is also frequent in Computer Science in which symbols are written inside angle brackets, and instead of a right arrow, the symbol **::=** is used. In this way, for example, the rule $expr \longrightarrow expr + expr$ is written as $< expr >::=< expr > + < expr >$. This notation is known as BNF (Backus-Naur Form) and it is the one used in the listings of Figures 1 and 2.

More complex expressions can be included by means of additional rules into the grammar to incorporate any needed term (trigonometric, exponential, etc.).

GE generates expressions using an evolutionary paradigm that iteratively converges to a solution to the problem, guided by mechanisms of individual selection and diversification. The algorithm maintains a population of vectors that represent linear genotypes that correspond to a potential solution to the problem; for instance, an individual can be a vector of integers or a binary array. During execution, genotypes experiment transformations using the genetic operators and the population management techniques typically used in GA. In GE, genotypes encode an expression that is a candidate solution to the problem. In terms of evolutionary computation this expression is the phenotype that corresponds to that genotype. Translation between genotype and phenotype is performed by a mapping module driven by a CFG. The genotype is divided in codons, and each codon encodes the element that is placed in one of the nodes of the tree: either a new derivation rule from the grammar or a terminal symbol such as variable labels or numbers, which allow no further expansion. It is an iterative process that starts with the grammar's axiom (root node) as initial phenotype ($expr$ in our example). The first element of the genotype encodes the grammar rule that gets expanded in the first place. In the example, the derivation rule selected by the first codon in the genotype is: $expr \longrightarrow expr + expr$. In each iteration the next element from the genotype is used to choose a proper derivation rule to rewrite the leftmost possible position in the partially expanded phenotype. This mapping stops when the genotype is exhausted or when the current phenotype does not contain any non-terminal symbol.

Like in any GA model, a fitness function is needed to estimate how well each possible solution solves the problem. In GE this function is applied to the phenotype by means of the direct evaluation of the generated expression.

GE is considered an alternative approach to automatic evolutionary computation initiated by genetic programming (GP) aiming to improve two specific questions: to ensure in a natural and standard way the syntactic correctness of the phenotypes and to generate expressions in any syntax provided by the CFG. Further extensions have been proposed if semantic correctness is also needed (guaranteeing that all the expressions fulfil any needed arbitrary condition) [7, 16]. The current state of the art of software tools for this kind of tasks, is far from the flexibility and expressive power of GE and its semantics extensions for SR.

There is an increasing interest and research effort devoted to SR in astronomy by means of AEC genetic approaches, usually by means of GP such as Operon or PySR [6, 8] in contributions like [14, 19]. In this work we give the first step to a novel application of GE in astrometry, aiming to check that GE is a viable approach. In these experiments we are not using the full power of GE: this first naive test only evolves, in fact, the constants of single candidate expressions. The success of this test informally supports the hypothesis of explainable improving this procedure in more realistic cases. The expected improvement in current tools thanks to these genetic approaches is twofold: this domain usually requires a flexible generation of expressions and the use of their underlying grammars is almost mandatory; by the other hand, this flexibility in general implies huge search spaces in which semantic extensios to GE have shown to be specially useful.

The problem under consideration is the prediction of intra-pixel position of stars for a specific instrument: WFPC2 camera at HST. This ensures a correct measurement of the star position in the sky even when the sensor resolution is coarser than the spatial resolution required for the star position, which is especially evident in the case of WFPC2 due to the undersampling introduced by PC and WF pixels.

To solve this problem, traditional methods require finding the shape of the point spread functions (PSF) that model the dispersion of light from point light sources when reaching the detection sensor. The PSF shape is mainly affected by the optical configuration at space telescopes (Fig. 3), while it is dominated by the atmospheric turbulence in ground-based observations. Our GE approach will automatically find the PSF shape for different points of the detection field, given the provided training data.

The rest of the paper is organized as follows. In section 2 we review current methods to estimate astrometric positions of point-like sources. They all have been tested using simulations and archival data from WFPC2. In section 3 we describe our particular implementation of GE as an alternative to produce estimates for star positions within the camera pixel. Here, we also describe the data we used to test our model. In section 4 we show the results we have obtained. Finally, conclusions and future lines to continue this research are outlined in section 5.

2 STAR CENTERING

The study of stars' proper-motions allows us to go deeper into our understanding of how the local universe is evolving. In this respect, the long temporal baseline provided by WFPC2, from year 1993 until 2009, is of high importance ([5, 10]). For example, the Mikulski Archive for Space Telescope includes a rich WFPC2 database of around one hundred globular clusters in the Milky Way and other regions near the Magellanic Clouds. However, performing precision astrometry is essential for this task, taking into account that the needed proper-motion precision is of the order of tens micro-arcsec per year, and the pixel resolution of the WF chip is 0.1 arcsec/pixel (0.045 arcsec/pixel for PC). Traditionally, the problem of estimating stars' proper-motions has been faced by fitting a predefined PSF shape to every star, and using some centering algorithm to estimate its intra-pixel position [9]. Here, we train a genetic algorithm (GA) to estimate the star's position within the pixel with a precision up to the hundredth of a pixel, i.e., ∼ 1 mili-arcsec.

All classic methods used to date basically differ in the PSF model (2D Gaussians or empirical PSFs -ePSF-, which are measured from real data) and the centering method to provide star positions at milli-pixel precision [3, 13]. Among those tested in [9] we can find that the approach which makes use of a library of ePSFs within the code hst1pass [1] yields the best results, at the cost of discarding very brightsources able to saturate the detector. Recently, a novel method based on supervised Deep Learning has been proposed [2, 4]. This approach does not make any assumption of the PSF shape, but it estimates the (x,y) coordinates of the star center by measuring correlations in the pixel intensity values around the pixel where the star is placed. Nevertheless, the need of the method to be trained with known labels in a supervised manner is an important caveat.

3 METHOD AND DATA

Our GE model is tested over images of point-like sources which simulate typical observations with the PC and WF chips of the WFPC2 camera. We used a simulation code under development, which is currently being created specifically for the WFPC2 detectors. This code generates simulated images that contain more real features and observational characteristics in comparison with other simulators such as [3]. Furthermore, these simulations were fed with the ePSFs used by the library in the hst1pass code, hence, they can be considered realistic. At the current stage of this project, we are not making use of the variability of the PSF over the chip, since we assume all stars lie in the central portion of the detector. In other words, the optical system is considered shift-invariant, i.e., its output can be modelled as a discrete convolution of the input with its impulse response or PSF, hence, all stars are assumed to produce the same PSF shape.

The total dataset consists of ∼ 4,600 simulated images of individual stars from globular cluster NGC 104, taken in filter F814W, at both WF and PC detectors. We have used 6 × 6-pixel cutouts around point-like sources and all images are normalized to sum one, independently of the noise level or if the star is saturated. All stars are located over the same image pixel, hence, output positions are also normalized between 0 and 1 so the model is estimating relative shifts within the same pixel. In all cases, we have a “ground truth” of star positions and magnitudes (i.e., star brightnesses) which allows us to check the accuracy of our results.

In this first viability test two GE experiments are designed to validate the behavior of this technique. The first experiment aims to determine which cost function achieves greater precision in guiding the evolutionary process. The options include Manhattan distance, Euclidean distance, Cosine similarity, Kendall correlation coefficient, Spearman correlation, and Pearson correlation. The second experiment serves to identify the parameter values within the evolutionary process that yield superior results. Various values are tested for parameters such as initialization, population size, generations, selection parameters, crossover, and mutation.

For the first experiment, one hundred stars are randomly selected for both PC and WF detectors, and each star is tested using six different fitness functions. These functions enable the determination of the PSF similarity with the original image, thus identifying the best individuals to create the next generation in the evolutionary process. Additionally, the experiment includes testing two different analytical PSF functions: 2D elliptical Gaussian and Moffat, which are typical analytical approximations to the PSF in astronomy studies [18], to determine which yields better results in terms of process accuracy.

The 2D elliptical Gaussian function is described by Eq.1.

$$G(x,y) = S_0 + S_1 e^{-\frac{1}{2}[A(x-x_0)^2+B(y-y_0)^2+C(x-x_0)(y-y_0)]} \tag{1}$$

While the 2D elliptical Moffat function is described by Eq.2.

$$M(x,y) = S_0 + \frac{S_1}{[1+A(x-x_0)^2+B(y-y_0)^2+C(x-x_0)(y-y_0)]^\beta} \tag{2}$$

In both cases with,

$$A = \left(\frac{\cos\phi}{\alpha_x}\right)^2 + \left(\frac{\sin\phi}{\alpha_y}\right)^2 \tag{3}$$

$$B = \left(\frac{\sin\phi}{\alpha_x}\right)^2 + \left(\frac{\cos\phi}{\alpha_y}\right)^2 \tag{4}$$

$$C = 2\sin\phi\cos\phi\left(\frac{1}{\alpha_x^2} - \frac{1}{\alpha_y^2}\right) \tag{5}$$

Being α_x and α_y parameters that depend on the *full width at half maximum or FWHM* of the PSF, and *phi* the ellipticity angle. Because we were aiming to test GE as a viable approach to fully explain this process, we have currently restricted the grammar so a single Gaussian or Moffat function was used as individual. The parameters in each of the functions were encoded as genotype and optimized by the GE. For both the Gaussian and Moffat functions, the genotype encoded the parameters: S_0, S_1, x_0 and y_0 (center), and angle ϕ. It also encode α_x and α_y for both functions, meaning standard deviations along the axes for the Gaussian. The Moffat function requires and additional term for parameter β.

From a technical perspective, PonyGE2 [15] is used for running evolutionary processes described on next section.

Figure 1 shows the PonyGE2 listing with the CFG for evolving the parameters of elliptical Gaussian PSFs. Lines 2 to 6 show the returned function that is, in fact, a Python implementation of equation 1. These lines contains all the symbols further defined: *sig_1*, *sig_2*, *phi*, *S0*, *S1*, *x0* and *y0*.

The rules for *S0, S1*, *x0* and *y0* are in lines 8 to 11. For *S0* and *S1* only fractional digits 2nd to 4rd are evolved. For *x0*, *x1* the first one is also evolved.

Symbols *<v_param5>*, *<v_param5>* and *<v_param6>* are intermediate variables to later define *sig_1*, *sig_2* and *phi*, respectively in lines 28 to 31, 33 to 36 and 38 to 39. They are different possible assignments to these three symbols that depend on symbols *<num>*, *<num_0_3>*, *<num_0_4>*, *<num_5_9>* and *<num_6_9>*. The valid options for these symbols are defined in lines 41 to 45.

Lines 19 to 24 contain some needed corrections to get a proper function.

It can be seen that PonyGE2 generates a Python funcion for being used by other modules of the system. Listing in Figure 1 mixes fragments of Python code (like in lines 2 to 11, 19 to 26 and the right hand sides of rules from lines 28 to 39) with BNF syntax (like in lines 1 and 28 to 45).

Figure 2 shows the PonyGE2 listing with the CFG for evolving the parameters of elliptical Moffat PSFs. It is similar to that of Figure

1. In this case the evolved parameters are *S0*, *S1*, *x0*, *y0*, *alpha*, and *beta*; *<v_param5>* is an intermediate symbol. The rules to evolve these parameters share the same approach of Figure 1.

4 RESULTS

The first experiment included 2400 executions (100 stars x 2 chips x 6 fitness functions x 2 PSF functions). Populations of 500 individuals and 20 generations are employed in each case. The results of this experiment are compared with the values of the “ground truth” to determine accuracy and residuals.

For the GE process we selected reasonable parameters based on literature: 20 generations, population of 500, RVD initialization, tournament selection (size 10), fixed_twopoint crossover, crossover probability of 0.2, mutation per individual (5 events) and elite size of 10.

It can be observed that the results obtained in the evolutionary process using a grammar based on the Moffat function yield superior outcomes (Table 1), particularly evident in WF chip images, where the spatial resolution is poorer. The behaviors of the fitness functions in the images of the two chips are different; on one hand, the Manhattan function generates better precision for WF, whereas on the other hand, the Cosine, Euclidean, and Pearson functions perform better for PC. Among these, Pearson exhibits the lowest standard deviation, indicating more stable results.

Nevertheless, a bias effect appears in all results. Figure 4 plots the distribution of x, y-residuals (right panel) and their dependence with star brightnesses, expressed in logarithmic scales as star magnitudes (left and middle panels). For PC, Pearson exhibits a bias in dx of -19 mpix with a standard deviation of 17 mpix , and in dy of +42 mpix with a standard deviation of 26 mpix. For WF, Manhattan shows a bias in dx of +89 mpix with a standard deviation of 37 mpix, and in dy of -35 mpix with a standard deviation of 37 mpix.

The biases appear to be related to the asymmetry in the distribution of light within an image, i.e., the PSF models do not fit completely the light distribution in the image, which exhibits more complex shapes and structures than can be described with simple model distributions such as Moffat or Gaussian.

For the second experiment, we performed extensive exploration of the GE evolution parameters. This experimental case was built upon the best-case scenario from experiment number one, i.e., the 2D elliptical Moffat PSF function was fit, while Pearson Fitness function was used for PC, and Manhattan one for WF. Hence, 28 scenarios are formulated by combining evolutionary process parameters to determine their impact on accuracy and bias error. These 28 scenarios are shown in Table 2.

All parameter values from experiment one are retained, with only the specified parameter values altered to observe how they modify the outcome. PC and WF results are analyzed separately, and as shown in experiment one, the evolutionary processes exhibits distinct behaviors in the images of different chips. For PC and WF, parameters that yield the highest accuracy in the results are described in Table 2.

For every execution of an evolutionary process, a symbolic representation is generated based on the used PSF function, thus providing explainability for the results. As depicted in Figure 5, a comparison is made between a real image and the PSF generated based on the specified model. Equation 6 shows the symbolic representation for left image in Figure 5.

5 CONCLUSIONS AND FUTURE WORK

We can conclude that star-centering process, using grammar evolution (GE), is technically feasible, allowing for the use of various fitness functions and base functions such as point spread function (PSF). For both PC and WF images, the elliptical 2D Moffat PSF function yielded superior results in terms of both accuracy and result deviation, as suggested by [18]. For PC, the cost function based on the Pearson coefficient was more precise, while for WF, the Manhattan distance function provided better accuracy.

An important advantage of using GE is the potential explainability of the results, as each individual in the evolutionary process generates a model equivalent to the symbolic representation of the defining PSF. Therefore, the PSF model can be explained combining different mathematical expressions to fit the best one to the image. However, in our particular data set, using the mentioned fitness functions, the technique exhibits a bias profile seemingly related to the luminosity asymmetry in the image, i.e., the 2D elliptical Moffat is not capturing the star profiles completely. While it might be possible to attempt a correction, such a process could potentially negatively impact the computational cost as more complicate expressions should be tried.

As a future work, we plan to extend our experiments to other instruments and contexts, not necessarily for the astrometric estimation of point-like sources, in which optical responses cannot be explained with simple mathematical models, for example, shift-variant PSFs (e.g., the TESS space telescope), or extended PSFs (which need an image support of several thousands of pixels, e.g., the SDSS ground-based telescope). GE should exhibit all its power when designing a more complex grammar aiming to generate more complex expressions that, in addition, will suggest and explain explicitly the models involved in the process. Some of the problems in this domain will need models that handle semantic restrictions, like our extensions to GE.

ACKNOWLEDGEMENTS

This work was funded by the research project BBforTAI (PID2021-127641OB-I00MICINN/FEDER) and HumanCAIC (TED2021-131787B-I00); and by the research Project "ADELA: Aplicaciones de Deep Learning para Astrofisica", with reference PP-2022-13, awarded by the Call "Proyectos Propios de Investigacion UNIR 2022"; by the Call for Grants for Research Stays Abroad 2021/2022, both of Universidad Internacional de la Rioja (UNIR). TG and DC are funded by the NASA Connecticut Space grant 80NSSC20M0129.

REFERENCES

[1] J. Anderson and I.R. King. 2000. Toward High-Precision Astrometry with WFPC2. I. Deriving an Accurate Point-Spread Function. Publications of the Astronomical Society of the Pacific 112, 776 (oct 2000), 1360. https://doi.org/10.1086/316632

[2] Roberto Baena-Galle, Terrence Girard, Dana Casetti-Dinescu, and Max Martone. 2023. Astrometric Centering of WFPC2/HST images with Deep Learning. (2023), 312–317.

[3] Bertin, E. and Arnouts, S. 1996. SExtractor: Software for source extraction. Astron. Astrophys. Suppl. Ser. 117, 2 (1996), 393–404. https://doi.org/10.1051/aas:1996164

[4] Dana Casetti-Dinescu, Terrence Girard, Roberto Baena-Galle, Max Martone, and Kate Schwendemann. 2023. Star-image Centering with Deep Learning:

HST/WFPC2 Images. PASP 135, 1047, (May 2023), 054501. https://doi.org/10.1088/1538-3873/acd080

[5] Dana I. et al. Casetti-Dinescu. 2017. Proper motion of the Sextans dwarf galaxy from Subaru Suprime-Cam data. Monthly Notices of the Royal Astronomical Society 473, 3 (10 2017), 4064–4076. https://doi.org/10.1093/mnras/stx2645 arXiv:https://academic.oup.com/mnras/article-pdf/473/3/4064/21841755/stx2645.pdf

[6] Miles Cranmer. 2023. Interpretable Machine Learning for Science with PySR and SymbolicRegression.jl. arXiv:2305.01582 [astro-ph.IM]

[7] Marina de la Cruz Echeandía, Alfonso Ortega de la Puente, and Manuel Alfonseca. 2005. Attribute Grammar Evolution. In Artificial Intelligence and Knowledge Engineering Applications: A Bioinspired Approach, José Mira and José R. Álvarez (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 182–191.

[8] Junlan Dong, Jinghui Zhong, Wei-Neng Chen, and Jun Zhang. 2023. An Efficient Federated Genetic Programming Framework for Symbolic Regression. IEEE Transactions on Emerging Topics in Computational Intelligence 7, 3 (2023), 858–871. https://doi.org/10.1109/TETCI.2022.3201299

[9] Dana I. Casetti-Dinescu et al. 2021. A Comprehensive Astrometric Calibration of HST's WFPC2. I. Distortion Mapping. Publications of the Astronomical Society of the Pacific 133, 1024 (jun 2021), 064505. https://doi.org/10.1088/1538-3873/abf32c

[10] Terrence M. Girard et al. 2011. THE SOUTHERN PROPER MOTION PROGRAM. IV. THE SPM4 CATALOG. The Astronomical Journal 142, 1 (jun 2011), 15. https://doi.org/10.1088/0004-6256/142/1/15

[11] John H. Holland. 1975. Adaptation in natural and artificial systems. https://api.semanticscholar.org/CorpusID:58781161

[12] John R. Koza. 1994. Genetic programming as a means for programming computers by natural selection. Statistics and Computing 4, 2 (1994), 87–112. https://doi.org/10.1007/BF00175355

[13] James F. Lee and Wm. F. van Altena. 1983. Theoretical studies of the effects of grain noise on photographic stellar astrometry and photometry. The Astronomical Journal 88 (1983), 1683–1689. https://api.semanticscholar.org/CorpusID:121772197

[14] Pablo Lemos, Niall Jeffrey, Miles Cranmer, Shirley Ho, and Peter Battaglia. 2022. Rediscovering orbital mechanics with machine learning. arXiv:2202.02306 [astro-ph.EP]

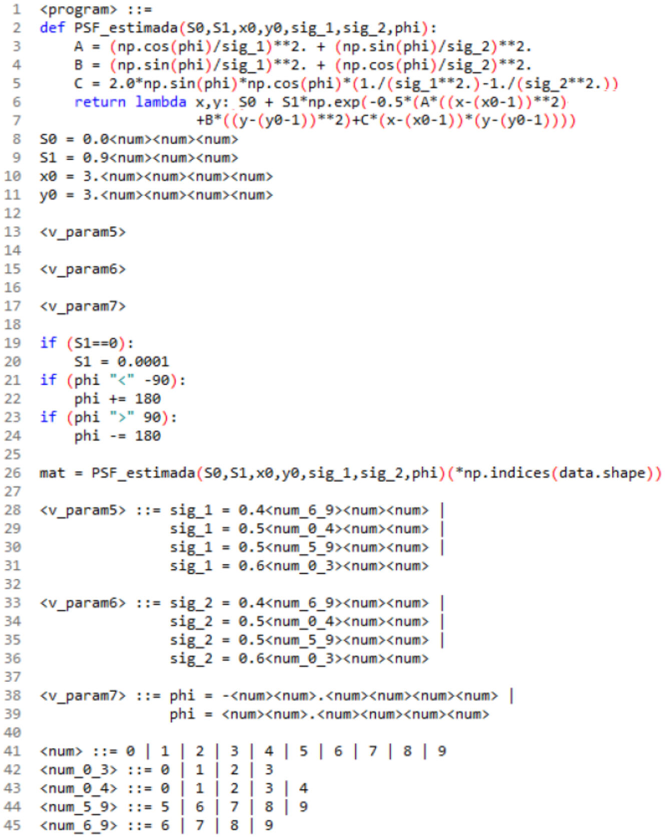
[15] Michael Fenton et al. 2017. PonyGE2: Grammatical Evolution in Python. In In Proceedings of GECCO '17 Companion. Berlin, Germany, 1194–1201. https://doi.org/10.1145/3067695.3082469

[16] Alfonso Ortega, Marina de la Cruz, and Manuel Alfonseca. 2007. Christiansen Grammar Evolution: Grammatical Evolution With Semantics. IEEE Transactions on Evolutionary Computation 11, 1 (2007), 77–90. https://doi.org/10.1109/TEVC.2006.880327

[17] Conor Ryan, Miguel Nicolau, and Michael O'Neill. 2002. Genetic Algorithms Using Grammatical Evolution. In Genetic Programming, James A. Foster, Evelyne Lutton, Julian Miller, Conor Ryan, and Andrea Tettamanzi (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 278–287.

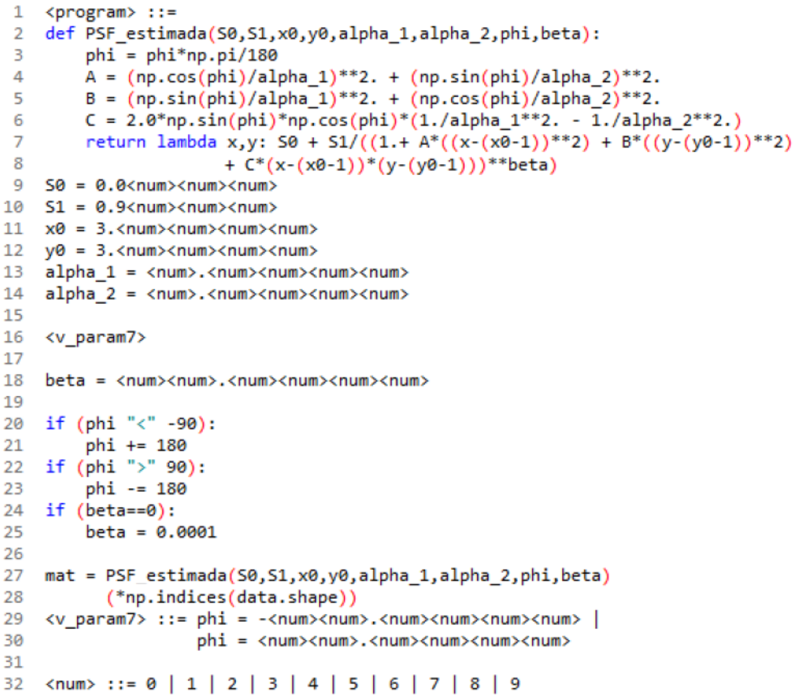
[18] I. et al. Trujillo. 2001. The effects of seeing on Sérsic profiles – II. The Moffat PSF. Monthly Notices of the Royal Astronomical Society 328, 3 (12 2001), 977–985. https://doi.org/10.1046/j.1365-8711.2001.04937.x arXiv:https://academic.oup.com/mnras/article-pdf/328/3/977/4078255/328-3-977.pdf

[19] Digvijay et al. Wadekar. 2023. The SZ flux-mass (Y–M) relation at low-halo masses: improvements with symbolic regression and strong constraints on baryonic feedback. Monthly Notices of the Royal Astronomical Society 522, 2 (04 2023), 2628–2643. https://doi.org/10.1093/mnras/stad1128 arXiv:https://academic.oup.com/mnras/article-pdf/522/2/2628/50119367/stad1128.pdf



```
<program> ::=
def PSF_estimada(S0,S1,x0,y0,sig_1,sig_2,phi):
    A = (np.cos(phi)/sig_1)**2. + (np.sin(phi)/sig_2)**2.
    B = (np.sin(phi)/sig_1)**2. + (np.cos(phi)/sig_2)**2.
    C = 2.0*np.sin(phi)*np.cos(phi)*(1./(sig_1**2.)-1./(sig_2**2.))
    return lambda x,y: S0 + S1*np.exp(-0.5*(A*((x-(x0-1))**2)
                        +B*((y-(y0-1))**2)+C*(x-(x0-1))*(y-(y0-1))))
S0 = 0.0<num><num><num>
S1 = 0.9<num><num><num>
x0 = 3.<num><num><num><num>
y0 = 3.<num><num><num><num>

<v_param5>

<v_param6>

<v_param7>

if (S1==0):
    S1 = 0.0001
if (phi "<" -90):
    phi += 180
if (phi ">" 90):
    phi -= 180

mat = PSF_estimada(S0,S1,x0,y0,sig_1,sig_2,phi)(*np.indices(data.shape))

<v_param5> ::= sig_1 = 0.4<num_6_9><num><num> |
               sig_1 = 0.5<num_0_4><num><num> |
               sig_1 = 0.5<num_5_9><num><num> |
               sig_1 = 0.6<num_0_3><num><num>

<v_param6> ::= sig_2 = 0.4<num_6_9><num><num> |
               sig_2 = 0.5<num_0_4><num><num> |
               sig_2 = 0.5<num_5_9><num><num> |
               sig_2 = 0.6<num_0_3><num><num>

<v_param7> ::= phi = -<num><num>.<num><num><num><num> |
               phi = <num><num>.<num><num><num><num>

<num> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<num_0_3> ::= 0 | 1 | 2 | 3
<num_0_4> ::= 0 | 1 | 2 | 3 | 4
<num_5_9> ::= 5 | 6 | 7 | 8 | 9
<num_6_9> ::= 6 | 7 | 8 | 9
```

Figure 1: CFG used to map genotypes into the parameters of elliptical Gaussian PSFs.



```
<program> ::=
def PSF_estimada(S0,S1,x0,y0,alpha_1,alpha_2,phi,beta):
    phi = phi*np.pi/180
    A = (np.cos(phi)/alpha_1)**2. + (np.sin(phi)/alpha_2)**2.
    B = (np.sin(phi)/alpha_1)**2. + (np.cos(phi)/alpha_2)**2.
    C = 2.0*np.sin(phi)*np.cos(phi)*(1./alpha_1**2. - 1./alpha_2**2.)
    return lambda x,y: S0 + S1/((1.+ A*((x-(x0-1))**2) + B*((y-(y0-1))**2)
                  + C*(x-(x0-1))*(y-(y0-1)))**beta)
S0 = 0.0<num><num><num>
S1 = 0.9<num><num><num>
x0 = 3.<num><num><num><num>
y0 = 3.<num><num><num><num>
alpha_1 = <num>.<num><num><num><num>
alpha_2 = <num>.<num><num><num><num>

<v_param7>

beta = <num><num>.<num><num><num><num>

if (phi "<" -90):
    phi += 180
if (phi ">" 90):
    phi -= 180
if (beta==0):
    beta = 0.0001

mat = PSF_estimada(S0,S1,x0,y0,alpha_1,alpha_2,phi,beta)
      (*np.indices(data.shape))
<v_param7> ::= phi = -<num><num>.<num><num><num><num> |
               phi = <num><num>.<num><num><num><num>

<num> ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Figure 2: CFG used to map genotypes into the parameters of elliptical Moffat PSFs.

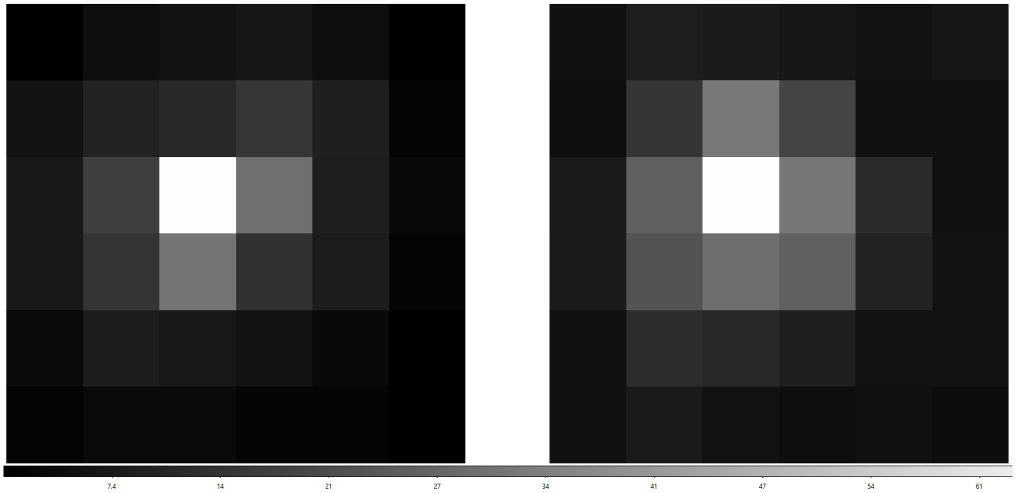


Figure 3: Point-like sources simulated for WFPC2 camera used in this work. Left: PC detector. Right: WF detector.

$$M(x,y) = 0.9221 + \frac{0.9452}{[1 + 3.1692(x - 3.2164)^2 + 3.2790(y - 3.8935)^2 + 1.3774(x - 3.2164)(y - 3.8935)]^{1.0146}} \tag{6}$$

Received 20 February 2007; revised 12 March 2009; accepted 5 June 2009

	2D Elliptical Gaussian				2D Moffat function			
	PC		WF		PC		WF	
Cost function	**Precision**	**Std dev**	**Precision**	**Std dev**	**Precision**	**Std dev**	**Precision**	**Std dev**
Cosine	52	23	139	34	50	24	123	36
Euclidean	53	23	145	35	52	24	129	41
Kendall	102	88	140	81	91	84	127	75
Manhattan	60	26	165	51	65	34	101	40
Pearson	52	24	139	34	51	21	120	32
Spearman	109	100	161	87	105	94	153	78
Total	**71**	**63**	**148**	**59**	**69**	**60**	**125**	**56**

Table 1: Summary of Experiment 1 using both PSF functions.

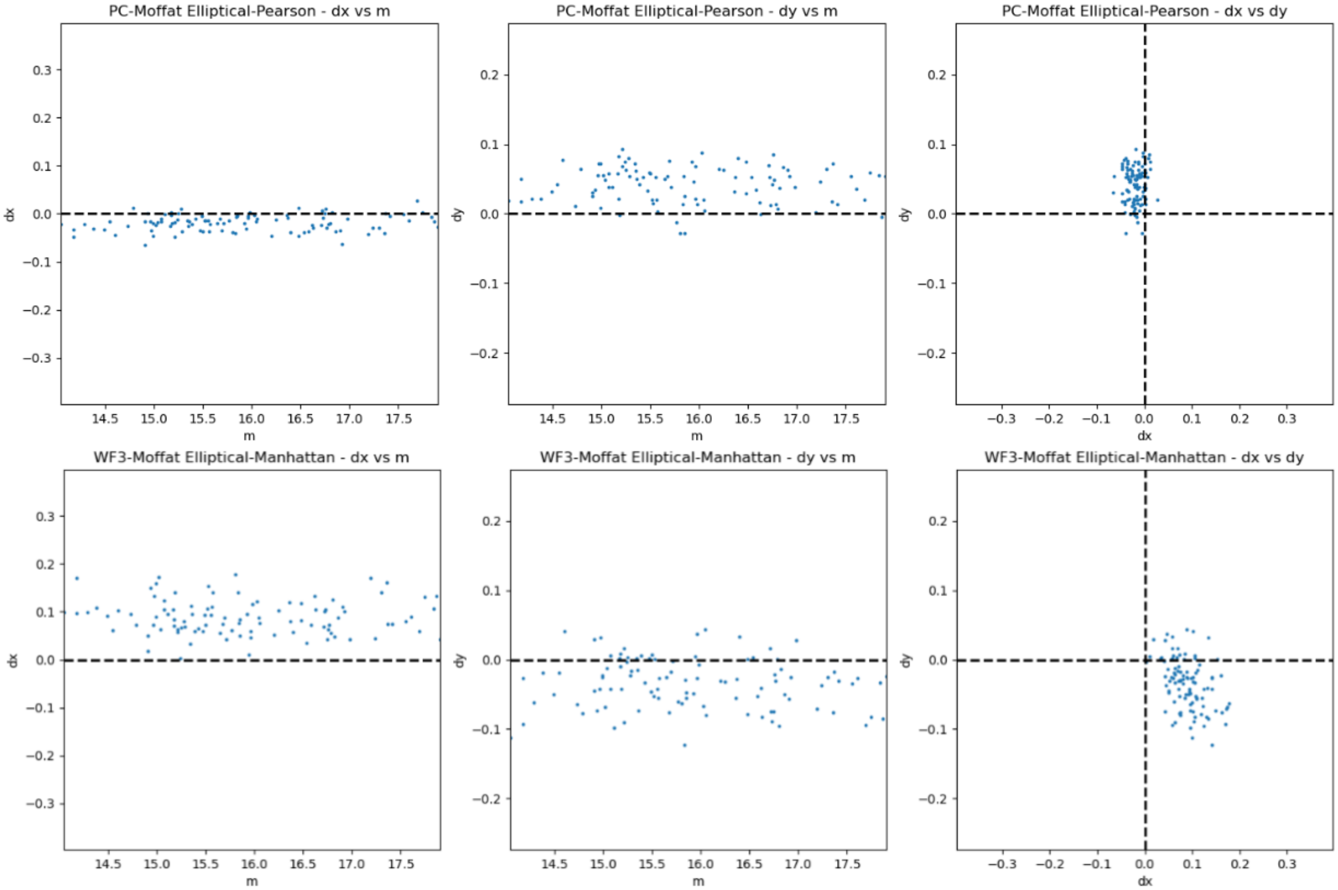


Figure 4: Bias profile for PC using a Pearson fitness function (top), and for WF using Manhattan (bottom). Left and middle columns: residuals in x- and y- w.r.t. the star magnitude. Right panel: residuals distribution w.r.t. (x,y) position.

	Scenarios	PC Best	WF Best
Generations	10, 15 and 20	20	20
Population	100, 500, 1000	1000	500
Initialization	RVD / uniform_genome	uniform_genome	RVD
Selection type	Tournament / Truncation	Tournament	Tournament
- Tournament Size	1, 5 or 10	5	10
- Selection Proportion	0.2, 0.5, or 0.8	No trunc.	No trunc.
Crossover type	(fixed or variable) x (onepoint or twopoints)	variable_onepoint	fixed_twopoint
- Crossover probability	0.2, 0.5, and 0.8	0.2	0.2
Mutation type	Per individual / per codon	Per codon	Per individual
- Mutation events	fixed (1, 5, 10) or Per codon prob (0.25, 0.50, or 0.75)	Per codon fixed	Per ind fixed (5)
Elite size	1, 5, 10 and 20	1	10

Table 2: Experiment 2: Scenarios and best parameters found.

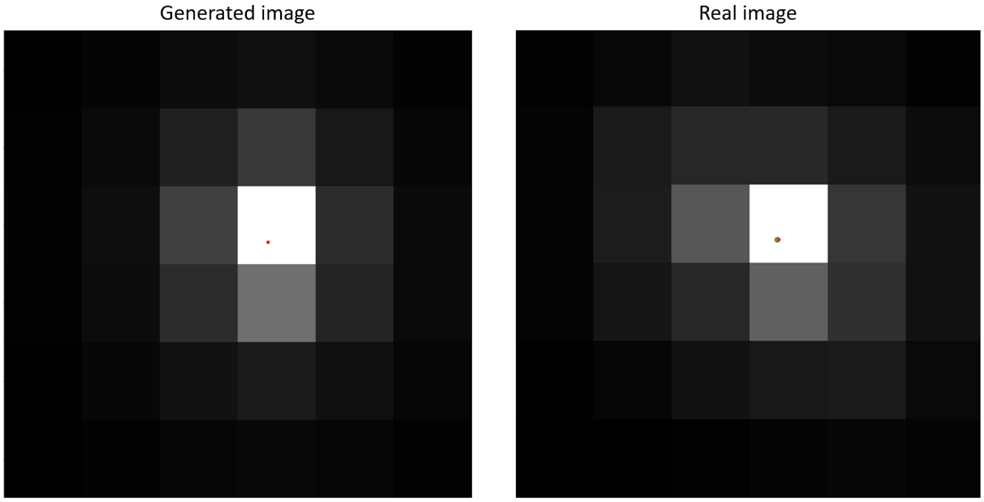


Figure 5: Model generated for a single evolutionary process, with generated image (left) and original image (right). Green dot is the real center $(x_{real}, y_{real}) = (3.2004, 3.8636)$**, red dot is the estimated one** $(x_{est}, y_{est}) = (3.2164, 3.8935)$**. Precision for this specific case is** 34 **mpix.**