STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **12**, May 2024, pp 660–671.
Published online in International Academic Press (www.IAPress.org)

Celestial Object Detection in Astronomical Images Using MSE and Jacobi Moments

Ismail Naouadir [1,*], Omar El Ogri [1,2], Jaouad El Mekkaoui [1], Mohamed Benslimane [1], Amal Hjouji [2]

[1] *LTI, Innovative Technologies Laboratory, Higher School of Technology, Sidi Mohamed Ben Abdellah University, Fez, Morocco*
[2] *CED-ST, STIC, Laboratory of Information, Signals, Automation and Cognitivism LISAC, Dhar El Mahrez Faculty of Science, Sidi Mohamed Ben Abdellah University, Fez, Morocco*

Abstract Astronomical images share a common characteristic, which is low variability. This term refers to images with minimal changes or variations in pixel values or features. In this article, we leverage the low variability characteristic of astronomical images to detect celestial objects. We compute the mean squared error for the reconstructed astronomical image, derived from its Jacobi moments. These moments are calculated multiple times, focusing on different regions within the image. The mean squared error metric serves as the score function for the simulated annealing algorithm, aiding in the identification of regions with the highest information loss. The parameters used to compute Jacobi moments to focus on that region are then interpreted as the coordinates of celestial objects. This method proves effective for preprocessing images since it provides optimal parameters for Jacobi polynomials, which will enhance their feature extraction capability. Additionally, it serves as an object detection method, as we can interpret the Jacobi moments' parameters as coordinates for the objects within the images.

Keywords Object Detection, Pattern Recognition, Jacobi Moments, Mean Squared Error, Astronomical Images.

DOI: 10.19139/soic-2310-5070-1959

1. Introduction

Object detection methods have become indispensable, revolutionizing fields such as autonomous vehicles, surveillance, medical imaging, and augmented reality. These methods are often grounded in advanced deep learning techniques like deep convolutional neural networks [1, 2, 3]. Noteworthy advancements in object detection are Faster R-CNN and YOLO [4, 5]. Object detection in astronomical images helps identify and categorize celestial objects, including stars, galaxies, and transient events like supernovae. With the ever-increasing volume of astronomical data from modern telescopes, automated object detection becomes essential for efficient analysis and systematic sky surveys [6]. The majority of object detection algorithms are two-stage detectors, employing two passes of the input image to predict the presence and location of objects. In the first pass, a set of proposals or potential object locations is generated, and the second pass refines these proposals for final predictions. While this approach is generally more accurate than single-shot object detection, it comes with a higher computational cost. Consequently, employing two two-stage algorithms is impractical for astronomical images due to their substantial dimensions and the varying scale of objects. One-stage detectors, such as YOLO, offer faster processing times but have limitations, as they need to be pretrained for specific use cases. The call for single-shot object detection without pretraining stems from the need for rapid adaptability and resource efficiency. These techniques allow

[*] Correspondence to: Ismail Naouadir (Email: ismail.naouadir@usmba.ac.ma). Higher School of Technology, Sidi Mohamed Ben Abdellah University, Fez, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

quick deployment and flexibility in handling diverse, unseen data, which is particularly beneficial in resource-limited scenarios like astronomy.

Jacobi polynomials and moments are named after the eminent mathematician Carl Gustav Jacob Jacobi [7]. They are used in pattern recognition, image analysis, and signal processing, where they excel at capturing essential features and patterns within complex data [8, 9]. These moments offer an excellent solution for feature extraction in astronomical images. Their unique advantage lies in their ability to reduce moment order and focus on specific regions of interest within an image while maintaining a satisfactory quality for most applications. In this paper, we use the mean squared error (MSE) metric as a scoring function for the simulated annealing algorithm. While maximizing the MSE metric may seem counterintuitive, it takes on a new meaning when we focus on specific regions using Jacobi polynomial parameters and look for regions with maximum information loss. In doing so, we obtain varying MSE scores depending on the content and variability of that region. Through the simulated annealing algorithm, we determine the region with maximum information loss and consequently optimal Jacobi polynomial parameters. These parameters serve as precise coordinates, effectively representing objects within low-variability images. This approach not only aids in object detection but also paves the way for a more nuanced and adaptive approach to handling diverse image datasets.

We will start by introducing orthogonal Jacobi moments along with their properties and parameters. Next, we'll present the mean squared error and its role in identifying regions of maximum information loss using Jacobi polynomials. Afterward, we will introduce the simulated annealing algorithm as an optimization tool for the mean squared error. Finally, we will employ the proposed algorithm to detect celestial objects in astronomical images.

2. Orthogonal Jacobi Moments

2.1. Jacobi Polynomials

For $\alpha, \beta \in \mathbb{R}; \alpha > -1, \beta > -1$, and a nonnegative integer n, we denote by $J_n^{(\alpha,\beta)}(x)$ the Jacobi polynomials, which comprises all the polynomials solutions to singular Sturm-Liouville problems on $[-1, 1]$. The Jacobi polynomials $J_n^{(\alpha,\beta)}(x)$, have the following Gauss hypergeometric representations:

$$J_n^{(\alpha,\beta)}(x) = \frac{(\alpha+1)_n}{n!}\,{}_2F_1(-n, 1+\alpha+\beta+n, \alpha+1; \frac{1-x}{2}) \tag{1}$$

where $(\alpha+1)_n$ is the Pochhammer's symbol given by:

$$(\alpha+1)_n = (\alpha+1)(\alpha+2)\dots(\alpha+n\ -\ 1), n \geq 1\ \ and\ \ (\alpha+1)_0 = 1 \tag{2}$$

and ${}_2F_1$ is the generalized hypergeometric which is defined as:

$${}_2F_1(a, b, c; x) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{x^k}{k!} \tag{3}$$

In this case, the series for the hypergeometric function is finite, so we get the expression of the Jacobi polynomials $J_n^{(\alpha,\beta)}(x)$ of degree n given by:

$$J_n(\alpha, \beta)(x) = \sum_{k=0}^{n} \chi_k^{(\alpha,\beta,n)} x^k \tag{4}$$

where

$$\chi_k^{(\alpha,\beta,n)} = (-1)^{n-k} \frac{\Gamma(n+\beta+1)\Gamma(n+k+\alpha+\beta+1)}{\Gamma(k+\beta+1)\Gamma(n+\alpha+\beta+1)(n-k)!k!} \tag{5}$$

Directly calculating Jacobi polynomials is highly complex, demanding significant computation time and resulting in numerical fluctuations, particularly when dealing with large orders. To address this issue, it is recommended to employ the recursive form of Jacobi polynomials [10]. These polynomials are generated using three-term recurrence relations:

$$J_{n+1}^{(\alpha,\beta)}(x) = (A_n x - B_n) J_n^{(\alpha,\beta)}(x) - C_n J_{n-1}^{(\alpha,\beta)}(x), n \geq 1 \tag{6}$$

with

$$J_0^{(\alpha,\beta)}(x) = 1 \quad and \quad J_1^{(\alpha,\beta)}(x) = (\alpha+\beta+2)x - (\beta+1) \tag{7}$$

and

$$A_n = \frac{(2n+\alpha+\beta+1)(2n+\alpha+\beta+2)}{(n+1)(n+\alpha+\beta+1)} \tag{8}$$

$$B_n = \frac{(2n+\alpha+\beta+1)(2n^2+(1+\beta)(\alpha+\beta)+2n(\alpha+\beta+1))}{(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)} \tag{9}$$

$$C_n = \frac{(2n+\alpha+\beta+2)(n+\alpha)(n+\beta)}{(n+1)(n+\alpha+\beta+1)(2n+\alpha+\beta)} \tag{10}$$

The Jacobi polynomials constitute an orthogonal system with respect to the following weight function: $\omega^{(\alpha,\beta)}(x) = (1+x)^{\alpha}(1-x)^{\beta}$ over $I = [-1, 1]$, that is,

$$\int_{-1}^{1} J_j^{(\alpha,\beta)}(x) J_k^{(\alpha,\beta)}(x) \omega^{(\alpha,\beta)}(x) dx = h_k^{(\alpha,\beta)} \delta_{jk} \tag{11}$$

where δ_{jk} is the Kronecker function and

$$h_k^{(\alpha,\beta)} = \frac{2^{\alpha+\beta+1}\Gamma(k+\alpha+1)\Gamma(k+\beta+1)}{(2k+\alpha+\beta+1)k!\Gamma(k+\alpha+\beta+1)} \tag{12}$$

The normalized Jacobi polynomials $\widetilde{J}_n^{(\alpha,\beta)}(x)$ are defined as:

$$\widetilde{J}_n^{(\alpha,\beta)}(x) = \sqrt{\frac{\omega^{(\alpha,\beta)}(x)}{h_k^{(\alpha,\beta)}}} J_n^{(\alpha,\beta)}(x) \tag{13}$$

We can deduce the orthogonality relation of the normalized Jacobi polynomials:

$$\int_{-1}^{1} \widetilde{J}_j^{(\alpha,\beta)}(x) \widetilde{J}_k^{(\alpha,\beta)}(x) dx = \delta_{jk} \tag{14}$$

2.2. Computation of Jacobi Moments

The classical Jacobi moments (CJMs) of order (n+m) of any function $f(x, y)$ defined on $[-1, 1] \times [-1, 1]$ can be computed using continuous integrals [8] as:

$$CJM_{nm} = \int_{-1}^{1}\int_{-1}^{1} f(x,y) \widetilde{J}_n^{(\alpha,\beta)}(x) \widetilde{J}_m^{(\alpha,\beta)}(y) dx dy \tag{15}$$

where $n = 0, 1, 2, ..., N-1$ and $m = 0, 1, 2, ..., M-1$

Therefore, the Classical Jacobi Moments $CJM_{nm}^{(\alpha,\beta)}$ of a digital image function $f(x, y)$ with the size $N \times M$, mapped into the region $[-1, 1] \times [-1, 1]$, can be computed using Zeroth Order Approximation (ZOA) method as:

$$CJM_{nm} = \sum_{i=0}^{N-1}\sum_{j=0}^{M-1} f(i,j) \widetilde{J}_n^{(\alpha,\beta)}(x_i) \widetilde{J}_m^{(\alpha,\beta)}(y_j) \Delta\mathrm{x}\Delta\mathrm{y} \tag{16}$$

where $\widetilde{J}_n^{(\alpha,\beta)}(x)$ is the nth order orthogonal Jacobi polynomials and $\Delta\mathrm{x} = x_i - x_{i-1}$, are sampling intervals in the 'x' and 'y' directions respectively, (x_i, y_j) is the center of (i, j) pixel. The image coordinate transform is defined as follows: $x_i = \left(i + \frac{1}{2}\right)\Delta\text{x} - 1, y_j = \left(j + \frac{1}{2}\right)\Delta\text{y} - 1$ with $\Delta\text{x} = \frac{2}{N}, \Delta\text{y} = \frac{2}{M}, i = 0, 1, 2 \ldots.., N - 1$ and $j = 0, 1, 2 \ldots.., M - 1$.

An approximation $f\left(x, y\right)$ of the original image can be obtained by:

$$f\left(i, j\right) = \sum_{n=0}^{N-1}\sum_{m=0}^{M-1} CJM_{nm}\widetilde{J}_n^{(\alpha,\beta)}\left(x_i\right)\widetilde{J}_m^{(\alpha,\beta)}\left(y_j\right) \tag{17}$$

If the Jacobi moments are limited to $N_{max} < N$ and $M_{max} < M$, then the image is constructed approximately as [11]:

$$\widetilde{f}\left(i, j\right) = \sum_{n=0}^{N_{max}-1}\sum_{m=0}^{M_{max}-1} CJM_{nm}\widetilde{J}_n^{(\alpha,\beta)}\left(x_i\right)\widetilde{J}_m^{(\alpha,\beta)}\left(y_j\right) \tag{18}$$

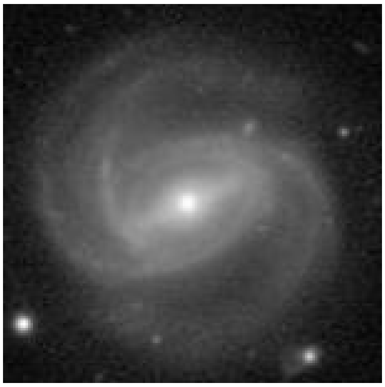
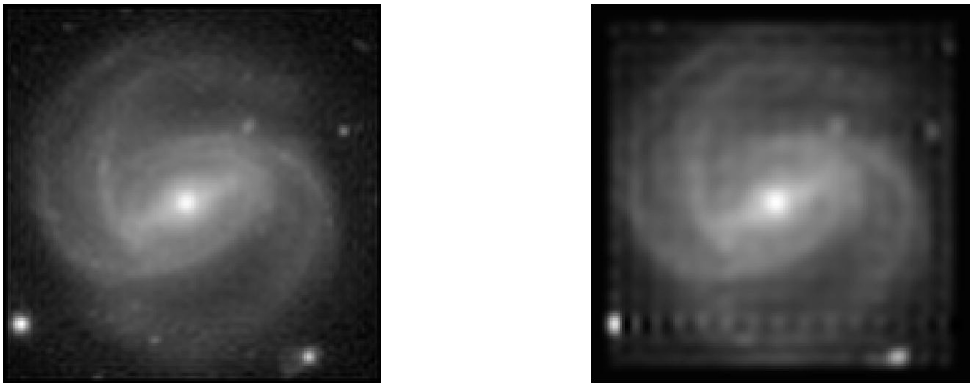


Figure 1. Image of a galaxy facing forward of size 152×152 from Sloan Digital Sky Survey.



(a) $n = 70$

(b) $n = 30$

Figure 2. Image reconstruction using Jacobi polynomials using $\alpha = \beta = 20$.

2.3. *Guidelines and Rules for Controlling the Jacobi Moments-Extracted Features*

In this section, we present experimental results to corroborate the theoretical framework developed earlier. The test image is displayed in Figure 3. Initially, we compute the Jacobi moments up to the specified order $(N_{\max}, M_{\max}) = (30, 30)$ using the parameters $(\alpha_1, \beta_1, \alpha_2, \beta_2)$ with the following equation:

$$CJM_{nm} = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} f(i,j)\, \widetilde{J}_n^{(\alpha_1,\beta_1)}(x_i)\, \widetilde{J}_m^{(\alpha_2,\beta_2)}(y_j)\, \Delta\text{x}\Delta\text{y} \tag{19}$$

and then reconstruct the image from the Jacobi moments with $(N_{max}, M_{max}) = (30,\ 30)$ and the calculated moments using the following equations.

$$\widetilde{f}(i,j) = \sum_{n=0}^{N_{max}-1} \sum_{m=0}^{M_{max}-1} CJM_{nm} \widetilde{J}_n^{(\alpha_1,\beta_1)}(x_i)\, \widetilde{J}_m^{(\alpha_2,\beta_2)}(y_j) \tag{20}$$

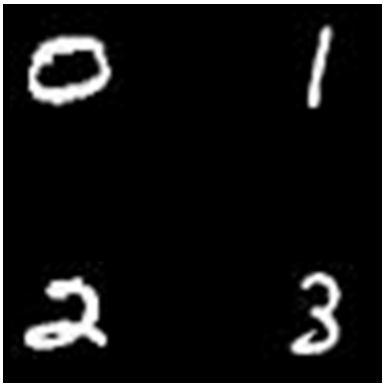


Figure 3. Image of size (278×278) with a number in each corner.

As we can see from the results in Figure 4, changing α_1 and β_1 will shift the focus between the upper and lower halves of the image depending on the larger value; for example, if α_1 is greater than β_1, the focus will be shifted to the upper half of the image, and vice versa; similarly, if α_2 is greater than β_2, the focus of Jacobi moments will be shifted to the left half of the image. By choosing these values carefully, we can target the location where the features we want to extract are, reducing the number of moments used because we don't need to extract global features, for example, in all four images in Figure 4, we used $(N_{max} \times M_{max}) = (30 \times 30)$ while the original size of the image was (278×278), resulting in an 89% reduction in image size.

3. Mean Squared Error for Object Detection

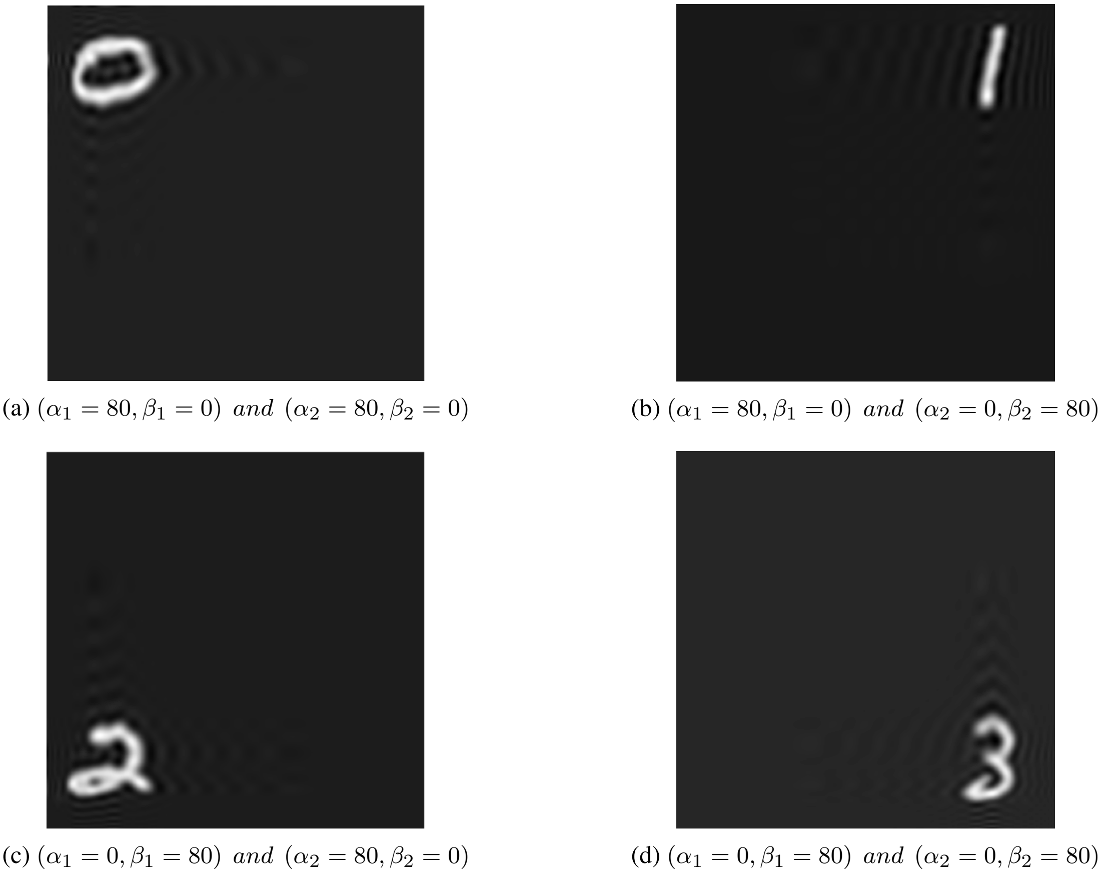
Mean Squared Error (MSE) is a widely used metric in statistics and machine learning to quantify the average squared difference between predicted and observed values [12]. It is commonly employed for assessing the performance of predictive models, such as regression models.

The formula for calculating MSE is as follows:

$$MSE = \frac{1}{n} \times \sum (y_i - \bar{y_i})^2 \tag{21}$$

where:

- MSE is the Mean Squared Error.
- n is the number of data points in the dataset.
- y_i represents the observed (actual) value for the i^{th} data point.
- $\bar{y_i}$ represents the predicted value for the i^{th} data point.



(a) $(\alpha_1 = 80, \beta_1 = 0)$ *and* $(\alpha_2 = 80, \beta_2 = 0)$

(b) $(\alpha_1 = 80, \beta_1 = 0)$ *and* $(\alpha_2 = 0, \beta_2 = 80)$

(c) $(\alpha_1 = 0, \beta_1 = 80)$ *and* $(\alpha_2 = 80, \beta_2 = 0)$

(d) $(\alpha_1 = 0, \beta_1 = 80)$ *and* $(\alpha_2 = 0, \beta_2 = 80)$

Figure 4. Image reconstruction using $(N_{max} \times M_{max}) = (30 \times 30)$ and different α and β values

MSE measures how closely a model's predictions match the actual data. A lower MSE indicates better alignment, while a higher MSE suggests larger prediction errors. A perfect match results in an MSE of 0. In image analysis [13], lower MSE values indicate greater similarity among images, signifying less variation in pixel values. Conversely, images with higher variability tend to yield higher MSE values. This concept also applies to specific regions within an image, as illustrated in Figure 5. If we concentrate Jacobi moments in the upper left corner and assess the reconstructed image using MSE, we can expect a higher value compared to focusing on other corners of the image.

After reconstructing the image twice using the parameters shown in Figure 6, once focusing on the upper left corner (Figure 6a) and the other on the upper right corner (Figure 6b), we compared the MSE scores. The first image had an MSE of 831,672,340,650, while the second had an MSE of 2,930,890,657,171. The higher MSE for the upper right corner can be attributed to the presence of the object 'star', resulting in more information loss during reconstruction. In contrast, the upper left corner contains less information, appearing as a black background. Consequently, there isn't much information to lose. This suggests that we can use high MSE scores to identify object locations in images with lower variability, such as medical and astronomical images.

A clearer perspective emerges when we set $\alpha_1 = 80$ and $\beta_1 = 0$ while varying α_2 and β_2 across all possible combinations between 20 and 80 with a step size of 5. Figure 7 vividly illustrates that when β_2 (horizontal axis) surpasses α_2 (vertical axis), it results in a higher MSE, indicating that the loss of information is more crucial in the upper right corner of the image, where the star object is located.

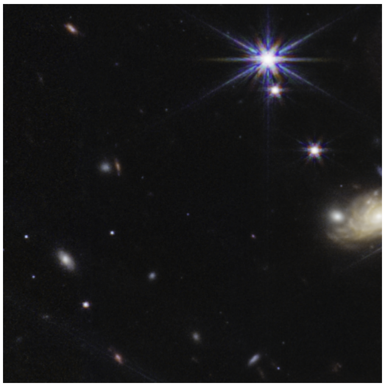
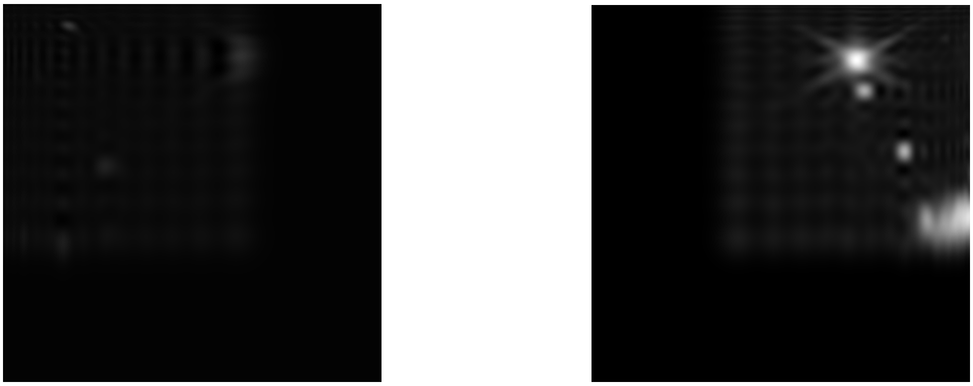


Figure 5. Lower variability image of size (500×500) with a star in the upper right corner.



(a) $(\alpha_1 = 80, \beta_1 = 0)\, and\, (\alpha_2 = 80, \beta_2 = 0)$

(b) $(\alpha_1 = 80, \beta_1 = 0)\, and\, (\alpha_2 = 0, \beta_2 = 80)$

Figure 6. Image reconstruction using Jacobi polynomials using $(N_{max} \times M_{max}) = (30 \times 30)$ and different α and β values.

4. Simulated Annealing for MSE Optimization

In problems where discovering an approximate global optimum is prioritized over pinpointing a precise local optimum within a fixed time frame, simulated annealing (SA) can be a more suitable choice compared to exact algorithms like gradient descent [14] or branch and bound [15]. SA enhances the Local Search algorithm by occasionally permitting moves to less optimal solutions. This seemingly counterintuitive approach allows the algorithm to escape local optima that may not be the global optimum.

4.1. The Basic Iteration

To apply the simulated annealing algorithm to our specific problem, several parameters need to be specified:

1. State Space: Define the state space relevant to the problem.
2. Energy Function (Score Function): Choose the energy function to maximize, which, in our case, is the Mean Squared Error (MSE).
3. Candidate Generator Procedure (Neighbor): Determine the procedure for generating candidates, which involves selecting values for $(\alpha_1, \beta_1, \alpha_2, \beta_2)$. This has been done using a random number generator between 0 and 80 with a step size of 5.
4. Acceptance Probability Function (P): Define the acceptance probability function.

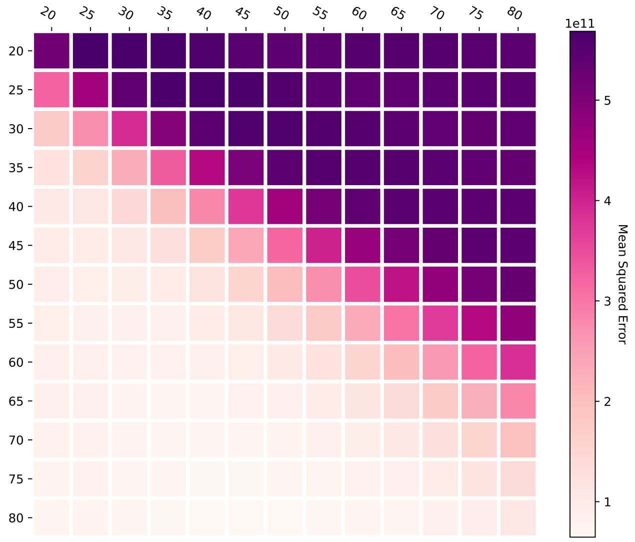


Figure 7. Heatmap of MSE scores evaluated on the image in Fig. 6 after reconstruction with Jacobi moments with $(N_{max} \times M_{max}) = (4 \times 4)$ with different (α_2, β_2) values .

5. Annealing Schedule: Specify the temperature (T) and the initial temperature. Details regarding these will be presented later in this section.

During each step, the simulated annealing heuristic evaluates neighboring states S^* of the current state S and probabilistically decides whether to transition to S^* or stay in S. These probabilities guide the system towards states with lower energy, ultimately seeking an optimal solution. Typically, this process is repeated until the system reaches a satisfactory state for the application or until a predefined computation budget is exhausted.

4.2. Convergencne

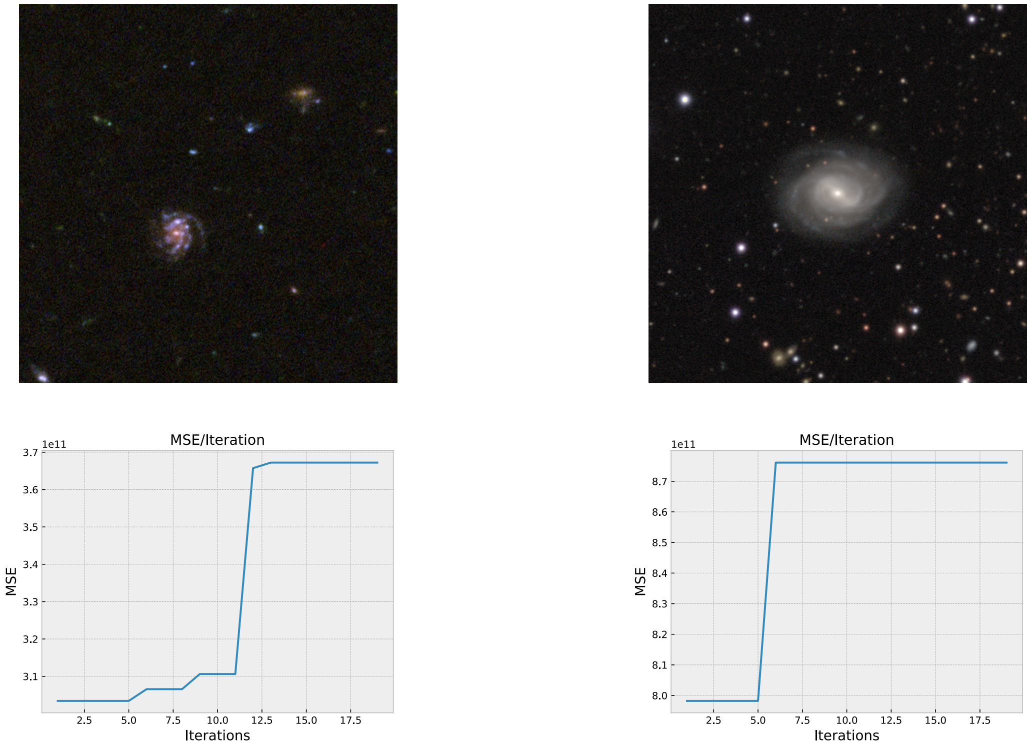
The SA algorithm can terminate based on user-defined conditions, which vary depending on the specific problem and optimization objectives. In our case, termination criteria may include reaching a maximum of 20 iterations, reaching a target temperature of zero, meeting convergence standards (such as no improvement for a predetermined number of iterations), or completing the optimization within a given one-minute time frame.

The line graphs in Figure 8 below show two instances of convergence in fewer than 20 iterations using the following parameters: Jacobi moments up to order $(N_{\max} \times M_{\max}) = (20 \times 20)$ and an initial temperature of 3. The algorithm computed the maximum MSE values for images (a) and (b), resulting in the coordinates $(\alpha_1 = 30, \beta_1 = 50)$ *and* $(\alpha_2 = 55, \beta_2 = 40)$ for image (a), as well as $(\alpha_1 = 60, \beta_1 = 60, \alpha_2 = 50, \beta_2 = 35)$ for image (b). Upon inspecting the images, we observed that these coordinates corresponded accurately to the locations of the main objects of interest.

5. Celestial Object Detection in Astronomical images

To demonstrate our method, we have chosen the first deep-field image captured by the James Webb Space Telescope (Figure 9). Since this image is larger than required and contains multiple objects, we segmented it into 81 smaller 500×500 images to make it more suitable for testing.

In Fig. 10, we demonstrate the utilization of optimal parameters generated by the SA algorithm for locating the object and calculating Jacobi moments to identify local features of the main object. The importance of parameter selection becomes evident, as using incorrect parameters can lead to the omission of the main object within the



(a) $(\alpha_1 = 30, \beta_1 = 50)\ and\ (\alpha_2 = 55, \beta_2 = 40)$

(b) $(\alpha_1 = 60, \beta_1 = 60)\ and\ (\alpha_2 = 50, \beta_2 = 35)$

Figure 8. Image reconstruction using $(N_{max} \times M_{max}) = (30 \times 30)$ and different α and β values

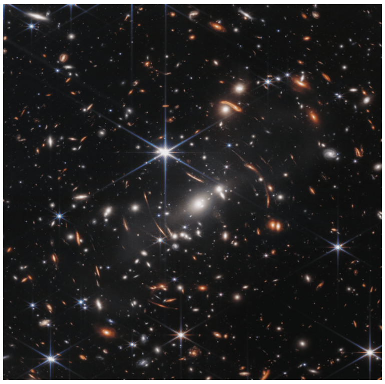
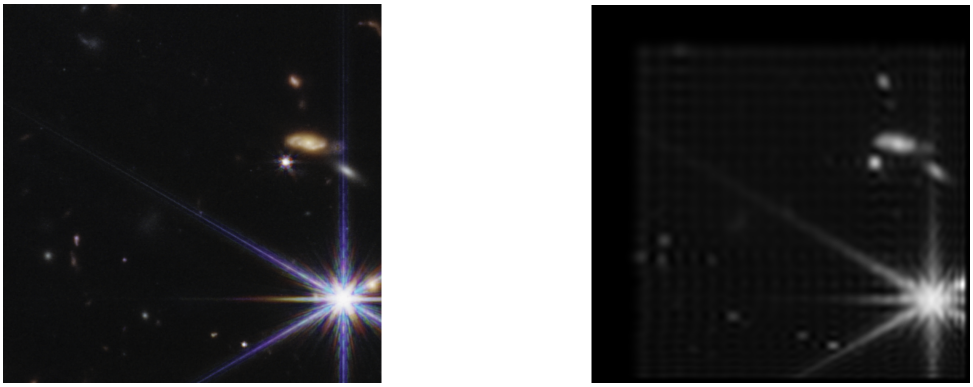


Figure 9. Webb's First Deep Field.

image, as seen in Figure 10b. In this instance, after applying reconstruction with $(N_{\max} \times M_{\max}) = (50 \times 50)$, the left regions of the image appear black due to the parameter focus on the right side of the image. The reconstruction was performed using $(N_{\max}\ \times\ M_{\max}) = (50\ \times\ 50)$ and the parameters $(\alpha_1 = 20, \beta_1 = 55)$ and $(\alpha_2 = 20, \beta_2 = 60)$ generated by the SA algorithm. Thanks to the optimal parameters, we are now able to utilize Jacobi moments of size (50×50) for our image instead of the original (500×500), which constitutes only 1% of the original size. We can calculate the size percentage compared to the original size as follows:

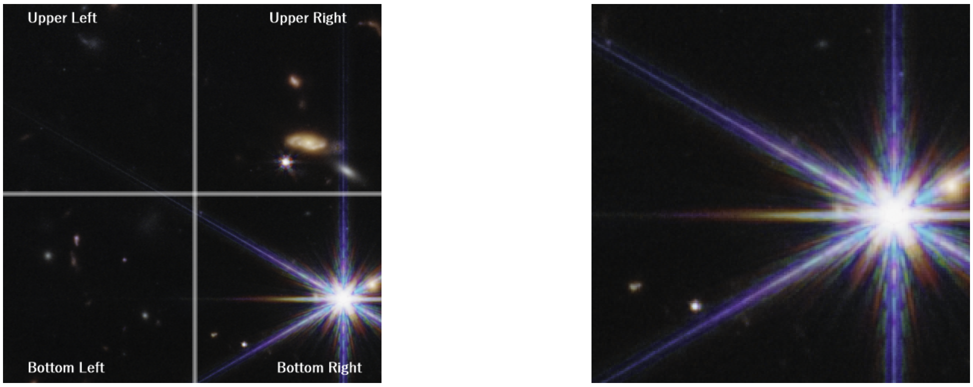
$$\frac{50 \times 50}{500 \times 500} \times\ 100\% = 0.01 \times 100\% = 1\% \tag{22}$$



(a) (b)

Figure 10. The original image (a) and the reconstructed image (b).

In addition to utilizing the parameters generated by the SA algorithm for calculating Jacobi moments, we can also use these parameters as coordinates to crop the image. For example, for the image in Figure 10a the algorithm returned $\alpha_1 \ < \ \beta_1$, which indicates that the object is situated in the bottom half of the image. Furthermore, $\alpha_2 < \beta_2$, indicates that the object is situated within the right half of the image.



(a) (b)

Figure 11. (a) Original image, (b) cropped image using the parameters generated by the SA.

After cropping the image using the provided parameters, we achieved a significant size reduction of up to 80%. For instance, in Figure 11, we followed the instructions to crop the image, resulting in a final image size of (215 × 215), which represents only 18% of the original size (500×500).

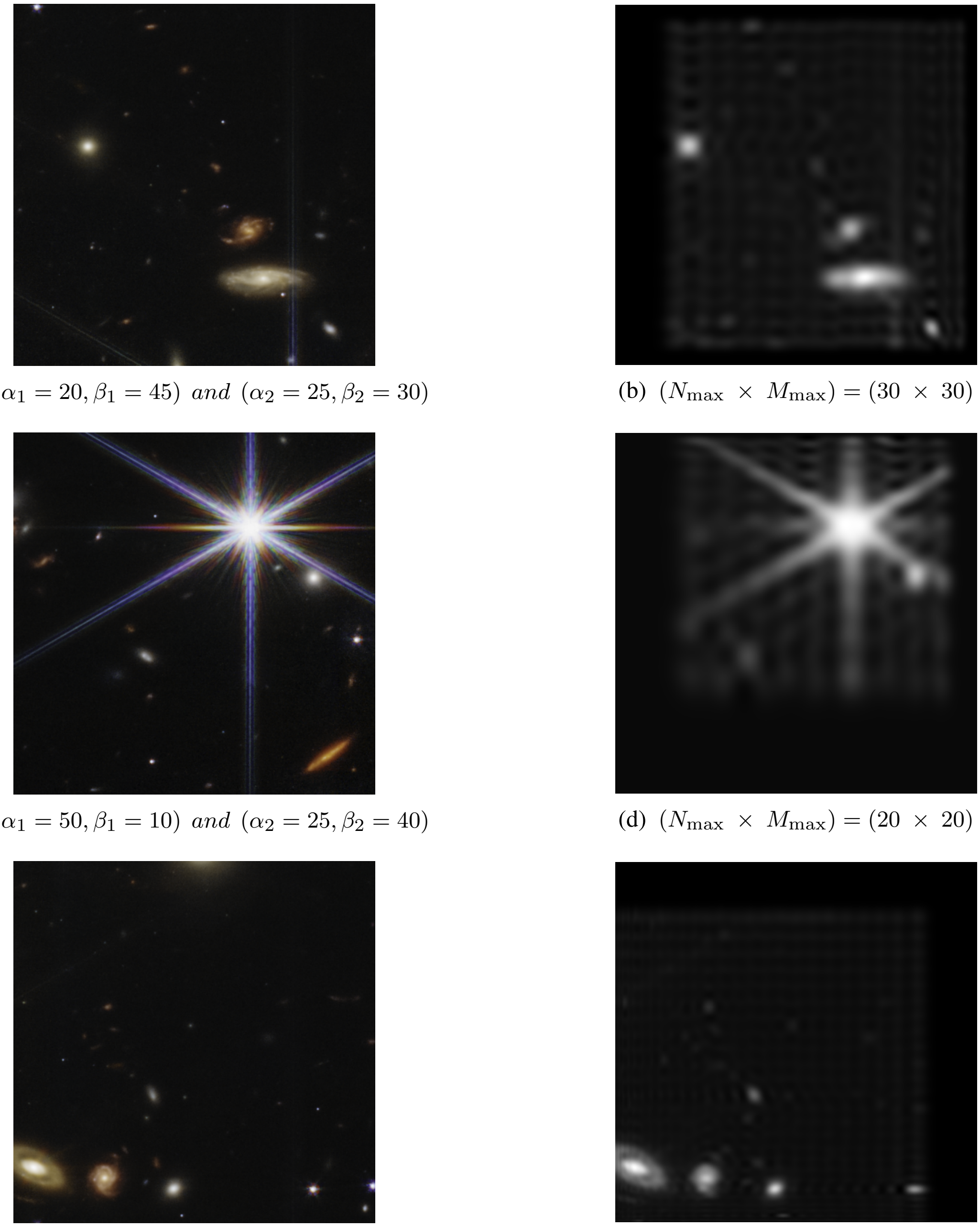
Figure 12 presents additional examples of reconstruction using the parameters generated by the SA algorithm, employing various $(N_{\max} \times M_{\max})$. Utilizing the optimal parameters generated by the SA algorithm, we can calculate the local Jacobi moments and reduce the order. Depending on the object's size, we can achieve significant size reductions, such as up to 99.84%, as demonstrated in Figure 12d. In this case, we used $(N_{\max} \times M_{\max}) =$ (20×20), which means we utilized only 0.16% of the original size (500×500).

$$\frac{20 \times 20}{500 \times 500} \times \ 100 \ \% = 0.0016 \ \times 100 \ \% \ = \ 0.16 \ \% \tag{23}$$

resulting in a remarkable 99.84% size reduction. And the object, the star, remains clear and recognizable.

8pt

Table 1 offers a concise overview of the reconstruction results achieved through the parameters generated by the Simulated Annealing algorithm.



(a) $(\alpha_1 = 20, \beta_1 = 45)$ *and* $(\alpha_2 = 25, \beta_2 = 30)$

(b) $(N_{\max} \times M_{\max}) = (30 \times 30)$

(c) $(\alpha_1 = 50, \beta_1 = 10)$ *and* $(\alpha_2 = 25, \beta_2 = 40)$

(d) $(N_{\max} \times M_{\max}) = (20 \times 20)$

(e) $(\alpha_1 = 10, \beta_1 = 60)$ *and* $(\alpha_2 = 60, \beta_2 = 5)$

(f) $(N_{\max} \times M_{\max}) = (50 \times 50)$

Figure 12. Images cropped from Webb's Deep field and the main object coordinates

Figure	Parameters (Object coordinates)	Reconstruction order	Size reduction percentage
Figure 10a	$(\alpha_1 = 10, \beta_1 = 10, \alpha_2 = 20, \beta_2 = 45)$	$(N_{\max} \times M_{\max}) = (50 \times 50)$	99.00%
Figure 12a	$(\alpha_1 = 20, \beta_1 = 45, \alpha_2 = 25, \beta_2 = 30)$	$(N_{\max} \times M_{\max}) = (30 \times 30)$	99.64%
Figure 12c	$(\alpha_1 = 50, \beta_1 = 10, \alpha_2 = 25, \beta_2 = 40)$	$(N_{\max} \times M_{\max}) = (20 \times 20)$	99.84%
Figure 12e	$(\alpha_1 = 10, \beta_1 = 60, \alpha_2 = 60, \beta_2 = 5)$	$(N_{\max} \times M_{\max}) = (50 \times 50)$	99.00%

Table 1. Size reduction percentage

6. Conclusion

We have presented a novel approach to object detection. Our method focuses on maximizing the mean squared error of the reconstructed image using the computed Jacobi moments of that image. This involves the utilization of low-order moments across various image regions, where control over the regions is achieved through Jacobi polynomials parameters. The mean squared error was then used as the score function for the simulated annealing to find the optimal parameters. The reduction in moments order leads to information loss, which is particularly high in specific image regions where the object is located, and this loss is quantified using the mean squared error. In low-variability images such as astronomical images, these regions of information loss align with the coordinates of the main object and optimal Jacobi polynomials parameters. The key advantage of this technique over others lies in its capacity to detect objects without the need for training the model. This efficiency in object detection without the requirement for extensive training or complex optimization processes makes the approach particularly advantageous for applications where real-time or computationally efficient solutions are critical.

Furthermore, by investigating alternative image quality metrics such as peak signal-to-noise ratio, our research may expand its horizons. We try to adapt and transform diverse image categories by covering a broader range of image types beyond those with lower variability. Exploration of various filters and transformation techniques may potentially make these images suitable for our method, extending its applicability to a wider range of domains and imaging scenarios.

REFERENCES

1. E. Arkin, N. Yadikar, Y. Muhtar, and K. Ubul, ‘A Survey of Object Detection Based on CNN and Transformer’, in 2021 IEEE 2nd International Conference on Pattern Recognition and Machine Learning (PRML), Jul. 2021, pp. 99–108. doi: 10.1109/PRML52754.2021.9520732.
2. F. Abdullayeva and Y. Imamverdiyev, ‘Development of Oil Production Fore-casting Method based on Deep Learning’, Statistics, Optimization & Information Computing, vol. 7, no. 4, Art. no. 4, Dec. 2019, doi: 10.19139/soic-2310-5070-651.
3. M. E. Khaili, L. Terrada, H. Ouajji, and A. Daaif, ‘Towards a Green Supply Chain Based on Smart Urban Traffic Using Deep Learning Approach’, Statistics, Optimization & Information Computing, vol. 10, no. 1, Art. no. 1, Feb. 2022, doi: 10.19139/soic-2310-5070-1203.
4. R. Girshick, ‘Fast R-CNN’, presented at the Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448. Accessed: Dec. 25, 2023. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2015/html/Girshick_Fast_R-CNN_ICCV_2015_paper.html
5. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, ‘You Only Look Once: Unified, Real-Time Object Detection’, presented at the Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788. Accessed: Dec. 25, 2023. [Online]. Available: https://www.cv-founda-tion.org/openaccess/content_cvpr_2016/html/Redmon_You_Only_Look_CVPR_2016_paper.html
6. S. C. Odewahn, ‘AUTOMATED CLASSIFICATION OF ASTRONOMICAL IMAGES’, PASP, vol. 107, no. 714, p. 770, Aug. 1995, doi: 10.1086/133622.
7. C. G. J. Jacobi and C. W. Borchardt, Vorlesungen über Dynamik. G. Reimer, 1866.
8. O. El Ogri, Omar, Hicham Karmouni, Mohamed Yamni, Mhamed Sayyouri, Hassan Qjidaa, Mustapha Maaroufi, and Badreeddine Alami, ‘Novel fractional-order Jacobi moments and invariant moments for pattern recognition applications’, Neural Comput & Applic, vol. 33, no. 20, pp. 13539–13565, Oct. 2021, doi: 10.1007/s00521-021-05977-w.
9. A. Nadira, A. Abdessamad, and B. S. Mohamed, ‘Regularized Jacobi Wavelets Kernel for Support Vector Machines’, Statistics, Optimization & Information Computing, vol. 7, no. 4, Art. no. 4, Dec. 2019, doi: 10.19139/soic-2310-5070-634.
10. B.-Y. Guo, J. Shen, and L.-L. Wang, ‘Generalized Jacobi polynomials/functions and their applications’, Applied Numerical Mathematics, vol. 59, no. 5, pp. 1011–1028, May 2009, doi: 10.1016/j.apnum.2008.04.003.
11. P.-T. Yap and R. Paramesran, ‘Jacobi moments as image features’, in 2004 IEEE Region 10 Conference TENCON 2004., Nov. 2004, pp. 594-597 Vol. 1. doi: 10.1109/TENCON.2004.1414490.
12. D. Wallach and B. Goffinet, ‘Mean squared error of prediction as a criterion for evaluating and comparing system models’, Ecological Modelling, vol. 44, no. 3, pp. 299–306, Jan. 1989, doi: 10.1016/0304-3800(89)90035-5.
13. Z. Wang and A. C. Bovik, ‘Mean squared error: Love it or leave it? A new look at Signal Fidelity Measures’, IEEE Signal Processing Magazine, vol. 26, no. 1, pp. 98–117, Jan. 2009, doi: 10.1109/MSP.2008.930649.
14. S. Ruder, ‘An overview of gradient descent optimization algorithms’. arXiv, Jun. 15, 2017. doi: 10.48550/arXiv.1609.04747.
15. E. L. Lawler and D. E. Wood, ‘Branch-and-Bound Methods: A Survey’, Operations Research, vol. 14, no. 4, pp. 699–719, Aug. 1966, doi: 10.1287/opre.14.4.699.