EAI Endorsed Transactions
on AI and Robotics

Research Article EAI.EU

Comparative Analysis of BAS and PSO in Image Transformation Optimization

Anik Dwivedi[1,*], Ameer Tamoor Khan[2], and Shuai Li[3]

[1]Department of Mechanical Engineering, IIT Kharagpur, India
[2]Department of Plant and Environmental Sciences, University of Copenhagen, Denmark
[3]Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland

Abstract

This paper presents a comparative study between the Particle Swarm Optimization (PSO) algorithm and the Beetle Antennae Search (BAS) algorithm for optimizing image transformations, with a focus on their performance in handling noisy and non-noisy images. Our experiments reveal that BAS consistently achieves better results in terms of pixel change when compared to PSO. The algorithms were evaluated based on their ability to minimize the objective function, which measures the error between the transformed reference image and the target image. Our results demonstrate that both BAS and PSO can effectively optimize image transformations, but BAS consistently outperformed PSO in terms of convergence speed and final objective value. Additional experiments with varying objective functions further validated the robustness and efficiency of BAS in achieving accurate image alignment.

Received on 22 March 2025; accepted on 17 May 2025; published on 29 May 2025

Keywords: Particle Swarm Optimization, Beetle Antennae Search, Image Transformation, Metaheuristic Optimization

doi:10.4108/airo.8955

1. Introduction

Image transformation is crucial in various applications such as image registration, computer vision, and pattern recognition [1]. In recent years, nature-inspired metaheuristic optimization has been increasingly applied to image alignment problems [2]. Among these methods, Particle Swarm Optimization (PSO) is a widely used algorithm inspired by social behavior in flocks; it is known for its simplicity and effectiveness [3]. PSO has been applied in many fields, including image processing and registration [4]. The Beetle Antennae Search (BAS) algorithm, on the other hand, is a more recently proposed optimizer that imitates the foraging behavior of beetles [5]. While both PSO and BAS have shown promise in optimization tasks, their comparative performance in image transformation—especially under noisy conditions—has not been thoroughly investigated.

The Beetle Antennae Search (BAS) algorithm has demonstrated remarkable versatility and adaptability across a broad spectrum of optimization challenges beyond image transformation. Initially inspired by the foraging behavior of beetles, BAS has been extended and hybridized to solve complex, real-world problems with promising results. For example, it has been applied to portfolio selection under cardinality constraints, showcasing its effectiveness in financial optimization tasks [6]. In the domain of multi-agent systems and smart environments, BAS has facilitated cooperative planning among robotic agents [7]. Its robustness has also been validated in high-stakes domains such as fraud detection in publicly traded firms [8], smart surgical control under remote center of motion (RCM) constraints [9], and the trajectory planning of both soft and bipedal robotic systems [10–12]. These diverse applications highlight BAS's strength in handling nonlinear, high-dimensional, and constrained problems, reinforcing its potential as a powerful alternative to more traditional optimizers like PSO.

In this work, we perform a comparative study of PSO and BAS for the task of aligning a reference image to a target image. We consider both non-noisy and noisy scenarios to evaluate the robustness of each method.

*Corresponding author. Email: anikdwivedi8055@kgpian.iitkgp.ac.in

The transformation parameters include 2D translation (X,Y), scaling, and rotation. We define an objective function based on pixel intensity differences between the transformed reference image and the target image, and we apply both algorithms to minimize this error metric.

Our experiments show that BAS consistently outperforms PSO in terms of convergence speed and final alignment accuracy. Specifically, BAS achieves lower objective values (i.e., better pixel alignment) than PSO across all scenarios. Extended experiments with modified objective functions further confirm the efficiency and robustness of BAS in achieving accurate image alignment.

The remainder of this paper is organized as follows. Section 2 reviews related work on evolutionary and swarm-based image registration methods. Section 3 formulates the object recognition (image alignment) problem and defines the evaluation metrics. Sections 4 and 5 introduce the PSO and BAS algorithms, respectively. Section 6 describes our fine-tuning of BAS parameters. Section 7 details the experimental setup. Section 8 presents visual results and comparisons. Section 9 provides tabular results on parameter values. Section 10 explores experiments with varied objective functions. Finally, Section 11 concludes the paper.

2. Related Work

Metaheuristic optimization algorithms have been extensively applied to image registration and transformation. For example, Cocianu et al. [2] review evolutionary image registration methods, noting that swarm intelligence algorithms are effective for such tasks. PSO, introduced by Kennedy and Eberhart [3], is one of the most popular swarm algorithms and has been employed in many image processing tasks due to its efficiency [4]. Variants of PSO (e.g., adaptive or inertia-weight strategies) have been proposed to improve convergence, but these remain complementary to the standard PSO framework [13].

Beetle Antennae Search (BAS) is a newer single-solution search algorithm inspired by beetle behavior [5]. Since its inception, numerous improvements and hybrid variants of BAS have been developed. Yousif and Saka [14] proposed an enhanced BAS (eBAS) that uses a population of beetles to avoid local optima. Chen et al. [15] provide a survey of BAS-related algorithms and applications. Yin et al. [16] introduced an ABSAS-CS-GSA variant, which integrates adaptive step-size control and sine-based position updates to improve coverage optimization. Other works hybridize BAS with algorithms like Particle Swarm Optimization or Genetic Algorithms. For instance, Khan et al. [17] combined BAS with the Adam optimizer (BAS-ADAM) for faster convergence, and Fan et al. [**?**] integrated a BAS-inspired search strategy into Grey Wolf Optimization. These studies highlight the flexibility and efficiency of BAS, but direct comparisons against PSO in image alignment contexts are still lacking. Our work aims to address this gap by directly comparing PSO and BAS on a standard image transformation task.

3. The Object Recognition Problem

The object recognition problem involves identifying a reference object (pattern) within a target landscape image. The goal is to find the planar coordinates, rotation angle, and scale factor that optimally align the reference object with the target image. This solution is represented as a four-tuple (x, y, s, θ), where x and y denote the planar coordinates of the center of the reference image relative to the target landscape, s is the scale factor, and θ is the rotation angle with respect to the coordinate system.

Since the object can appear at any location within the landscape, the search space of all possible combinations of (x, y, s, θ) is large. Consequently, this problem can be framed as an optimization problem, where the objective is to find the values of (x, y, s, θ) that maximize the similarity between the reference and target images.

In this study, the search space is constrained by setting limits on the ranges of the variables. Specifically, we restrict the column and row coordinates to $(0 \leq x < n)$ and $(0 \leq y < m)$, respectively. The scale factor is bounded by $(0.5 \leq s \leq 2.0)$, and the rotation angle is constrained to $(-\pi \leq \theta \leq \pi)$. To evaluate potential solutions, a measure of similarity between the reference image (RI) and the target landscape image (LI) must be defined. Various similarity measures, such as mutual information and the sum of squared differences (SSD) between pixel intensities, have been proposed in the literature.

$$\text{EvalSol} = \frac{\text{ErrorMax} - \text{ErrorSol}}{\text{ErrorMax}} \tag{1}$$

$$\text{ErrorMax} = 2^{\text{nbits}} \times (m \times n - P_{\text{inv}}) \tag{2}$$

$$\text{ErrorSol} = \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} |RI(i,j) - LI(i,j)| \tag{3}$$

However, mutual information is computationally expensive, and SSD may yield high similarity values even when the images are not well-aligned. Therefore, we adopt the absolute sum of differences (ASD) between pixel intensities as the similarity measure in this work. This measure, denoted as "Error," quantifies the difference between corresponding pixels in the reference and landscape images. The evaluation of a candidate solution is computed using the above equations.

In these equations, n and m represent the dimensions (width and height) of the reference image, and nbits is the number of bits used to represent pixel intensity levels. The term P_{inv} accounts for pixels in the reference image that do not overlap with the landscape image (this occurs when the reference image is near the borders of the landscape). The variable ErrorSol is the sum of the absolute differences in pixel intensities between the reference and landscape images for a given solution.

The pixel coordinates (I, J) in the landscape image corresponding to the reference image can be obtained using the following transformation equations:

$$I = y + s \times (ddX \times \sin(-\theta) + ddY \times \cos(\theta)) \quad (4)$$

$$J = x + s \times (ddX \times \cos(-\theta) + ddY \times \sin(\theta)) \quad (5)$$

Here $ddX = j - \frac{\text{Width}_{RI}}{2}$ and $ddY = i - \frac{\text{Height}_{RI}}{2}$, with x, y, s, and θ being the candidate solution parameters under evaluation.

The evaluation function, EvalSol, approaches its maximum value of 1 as the error ErrorSol tends to 0, meaning that a higher similarity between the reference and landscape images results in a better evaluation score. Consequently, an optimization algorithm such as Particle Swarm Optimization (PSO) can be employed to maximize EvalSol, thereby minimizing the pixel intensity difference between the reference and landscape images.

The previously defined evaluation is suited for grayscale images; however, the same principle can be extended to color images. In such cases, the matching error is computed for each independent channel of the RGB color space. The corresponding equations for evaluating a solution in color images are as follows:

$$\text{EvalSol} = \frac{3 \times \text{ErrorMax} - \text{ErrorChannelSol}}{\text{ErrorMax}} \quad (6)$$

$$\text{ErrorChannelSol} = \sum_{ch=1}^{3} \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} |RI(i,j,ch) - LI(i,j,ch)| \quad (7)$$

In the case of color images, the range of the evaluation function expands to $[0, 3]$, as there are three independent channels (R, G, B). However, this increased range does not significantly impact the object recognition process. By considering each color channel separately (denoted by ch in Equation 7), the search can be accelerated if the landscape image is preprocessed to identify large color discrepancies. This preprocessing step can significantly reduce the search space, thereby minimizing the computational cost of object recognition.

3.1. Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a heuristic technique belonging to swarm intelligence. Like genetic algorithms, genetic programming, evolution strategies, and ant colony optimization, PSO is an evolutionary computation method. Introduced by Kennedy and Eberhart in 1995 [3], PSO draws inspiration from social behaviors in nature, such as bird flocking, bee swarming, and fish schooling. The main idea is to simulate a simplified social system where individuals (particles) adjust their positions to maintain an average distance from their neighbors. The behavior of one particle affects the group and vice versa. Computationally, the swarm consists of particles representing potential solutions, which "fly" over the solution space seeking the optimal solution.

Initially, particles' positions and velocities in the search space are randomly initialized, ensuring that any point can be reached. Each particle has limited memory to store its current position, its best position found so far (pbest), and the best position found by its neighbors (lbest) or the entire swarm (gbest), depending on the implementation. The pbest represents the particle's individual knowledge (cognitive component), while the gbest or lbest represents the group's collective knowledge (social component). At each time step, a particle's movement is influenced by both its cognitive and social components. The particle's velocity is updated based on the weighted influence of pbest and gbest, as shown in ((8)):

$$\text{Vel}_i^{t+1} = w \cdot \text{Vel}_i^t + c1 \cdot r1 \cdot (\text{pbest}_i^t - x_i^t) + c2 \cdot r2 \cdot (\text{gbest}_i^t - x_i^t) \quad (8)$$

In Equation (8), Vel_i^t denotes the velocity of particle i at iteration t, and x_i^t represents the current position of the particle in the solution space. The variables pbest_i^t and gbest^t are the best positions found so far by particle i and the entire swarm, respectively. The constants c_1 and c_2 are user-defined acceleration coefficients that control the influence of cognitive and social components, respectively. The variables r_1 and r_2 are randomly generated values in the range $[0, 1]$, and w is the inertia weight that influences the particle's tendency to continue in its current direction.

With the updated velocity, a particle's position in the search space is updated using ((9)):

$$x_i^{t+1} = x_i^t + \text{Vel}_i^{t+1} \quad (9)$$

In Equation (9), the new position x_i^{t+1} is calculated by adding the updated velocity to the current position. The swarm evolves by iteratively updating both the velocities and positions of particles based on their individual and collective experiences.

The acceleration constants $c1$ and $c2$ directly influence the step size in the search space. High $c1$ values promote local search, causing particle clusters, while high $c2$ values lead to clustering at a local maximum, which can stagnate the search. As with other evolutionary methods, choosing optimal control parameter values is challenging and problem-dependent.

The standard PSO algorithm can be summarized as follows:

1. Initialize the swarm of particles with random values for each component of the solution vector.
2. Evaluate the fitness of the solution represented by each particle.
3. If a particle's current solution is better than its pbest, update pbest. If the current solution is better than gbest, update gbest.
4. Compute the new velocity of particles using ((8)).
5. Compute the new positions of particles using ((9)).
6. If a stopping criterion is met, stop; otherwise, return to step 2. The stopping criterion can be a maximum number of iterations or a solution of satisfactory quality.

To enhance PSO performance, diversification methods can be applied. When the swarm converges to a region in the search space, further improvements become difficult due to a lack of solution diversity, evident when gbest stagnates over several iterations. A common strategy to address this is the "explosion" method, where the swarm is reset, retaining the previous gbest. This technique is effective for complex problems, helping PSO achieve better solutions compared to a PSO without explosions.

Overall, PSO is straightforward to implement and does not require extensive computational resources. The most computationally intensive part is evaluating candidate solutions (particles).

3.2. Beetle Antennae Search (BAS)

Beetle Antennae Search (BAS) is a relatively recent bio-inspired optimization algorithm developed by Jiang and Li (2018) and further explored by Zhang et al. (2021). Unlike many swarm-based algorithms, BAS mimics the behavior of a single beetle, specifically the longhorn beetle. In nature, beetles use their antennae to detect the smell of potential food sources or mates and move towards them. BAS models this foraging behavior by representing a beetle as a solution in a multi-dimensional search space.

The algorithm begins by randomly initializing the beetle's position and a direction in the search space. Based on this position and a random direction, two new solutions (left and right positions) are calculated, representing the positions of the beetle's antennae. The beetle then moves a certain step size towards the solution with the better fitness among the left and right positions. This process repeats until the optimal solution is found or a maximum number of iterations is reached.

BAS and its improved versions have been applied to various fields, including ship collision avoidance (Xie et al. 2019) and path planning for mobile robots (Wu et al. 2020; Zhou et al. 2021). BAS offers advantages such as simplicity and fewer parameters because it uses only one beetle. The time complexity of BAS is $O(K \cdot N)$, where K is the problem dimension and N is the maximum number of iterations. In contrast, the time complexity of SMA is $O(K + N \cdot P \cdot (1 + \log P + K))$, where P is the number of slime mold cells.

Due to its simplicity, BAS requires less running time and is easy to combine with other swarm-based algorithms as a global or local search strategy. For example, Lin et al. (2018) combined BAS with PSO to create a new algorithm leveraging the global search capability of PSO and the local search capability of BAS. Fan et al. (2021) proposed an improved GWO algorithm (BGWO) where the alpha wolf uses a BAS-like sensing ability to better lead the pack to the optimal solution. Shao et al. (2018) used BAS as an additional strategy after the global search phase of FPA, improving the algorithm's convergence rate. Zhao et al. (2020) combined BAS with GA to form BGA, demonstrating better performance than other hybrid algorithms. Zhou et al. (2019) integrated BAS with SA, allowing multiple searches at each temperature with step size proportional to the current temperature, enhancing both global and local search capabilities. Similar integrations with ACO and ABC resulted in the BCO and MABC algorithms (Zhang et al. 2020a, b), respectively.

BAS Algorithm Overview. The BAS algorithm is an optimization technique inspired by the foraging behavior of beetles, using their antennae to detect and locate food. The BAS algorithm simulates this behavior to search for optimal solutions in a given search space.

Random Direction Initialization: The beetle's searching behavior starts with a random direction vector, normalized to have a unit length, represented as:

$$\vec{b} = \frac{\text{rnd}(k, 1)}{\|\text{rnd}(k, 1)\|} \tag{10}$$

where rnd(·) denotes a random function and k represents the dimensions of the position.

Antennal Search Mechanism: The algorithm evaluates positions on both sides (right-hand side x_r and left-hand side x_l) of the current position x_t:

$$x_r = x_t + d_t\vec{b}, \quad x_l = x_t - d_t\vec{b} \tag{11}$$

Here, d is the sensing length of the antennae, which should initially be large and then decrease gradually.

Odor Detection and Position Update: The beetle decides its next move based on the fitness values $f(x_r)$ and $f(x_l)$. The next position x_{t+1} is updated as follows:

$$x_{t+1} = x_t + \delta_t\vec{b} \cdot \text{sign}(f(x_r) - f(x_l)) \tag{12}$$

where δ is the step size, and sign(·) represents the sign function.

Step Size and Sensing Length Update: The step size δ and sensing length d are iteratively updated to control convergence speed and search precision:

$$d_t = 0.95d_{t-1} + 0.01, \quad \delta_t = 0.95\delta_{t-1} \tag{13}$$

4. Fine-Tuning the BAS Algorithm

In this study, the BAS algorithm was fine-tuned by carefully adjusting its parameters to optimize performance for image transformation tasks. The key parameters fine-tuned include:

Initial Sensing Length (d): The initial sensing length was set to cover a broad search area, allowing the algorithm to avoid local minima early in the search process. This length was gradually decreased to enable the algorithm to focus on fine details as it approached potential optimal solutions.

Step Size (δ): The step size was initially set to a high value to facilitate faster exploration of the search space. This enabled the algorithm to cover more ground quickly and identify promising regions. Over time, the step size was reduced to allow for more precise convergence towards the optimal solution, ensuring finer adjustments and higher accuracy.

Update Rules: The decay rates for both d and δ were carefully adjusted to balance exploration and exploitation. By fine-tuning the decay rates, the algorithm was able to quickly find a good approximation of the optimal solution and then refine it through more detailed searches. This balance ensures that the algorithm does not get stuck in suboptimal regions and can efficiently converge to a high-quality solution.

$$d_t = \alpha d_{t-1} + (1-\alpha)d_{\min}, \quad \delta_t = \beta\delta_{t-1} \tag{14}$$

where α and β are decay rate factors, and $d_{\min}$ is the minimum sensing length. The specific values for α and β were determined experimentally to achieve the best performance in the image transformation tasks.

By systematically fine-tuning these parameters, the BAS algorithm's efficiency and effectiveness in image transformation tasks were significantly enhanced. This fine-tuning process allowed for a more adaptive and responsive search strategy, capable of navigating complex solution spaces with greater precision.

5. Experimental Setup

We implemented both the Particle Swarm Optimization (PSO) and Beetle Antennae Search (BAS) algorithms to optimize the transformation parameters. The reference and target images were processed under three scenarios:

1. Non-noisy images
2. Noisy reference and target images
3. Non-noisy reference image with noisy target image

5.1. Reference and Target Images

The reference image (initial state) and the target image (desired state) were used for evaluating the performance of the algorithms. The images were subjected to noise to simulate real-world conditions and test the robustness of the optimization algorithms.

6. Observations and Visual Comparison

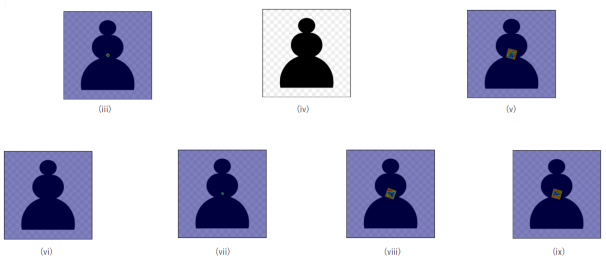
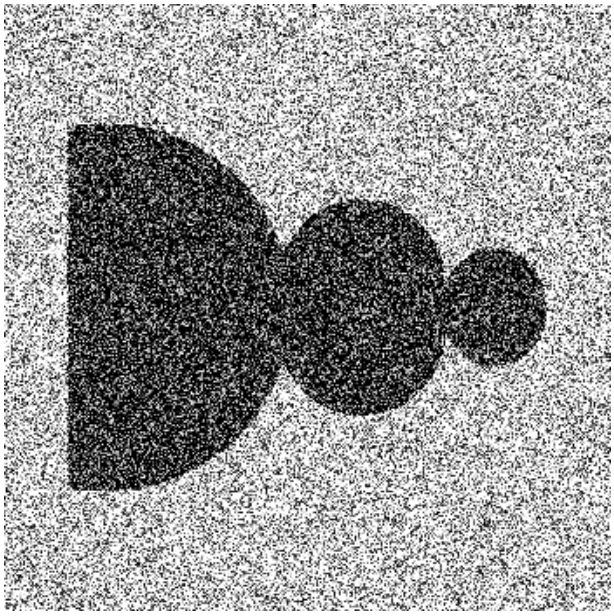
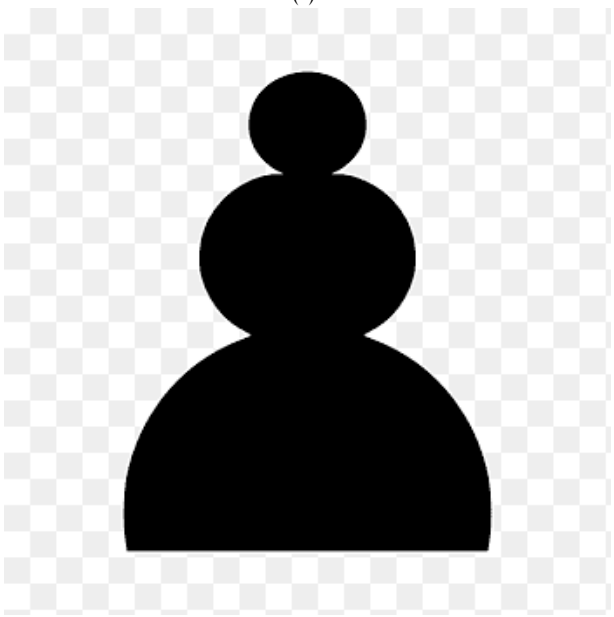


Figure 2. Comparison between (iii) BAS Transformed Non-Noisy with Noise Considerations (iv) Target Image (v) PSO Transformed Non-Noisy with Noise Considerations (vi) BAS Transformed Non-Noisy Reference (vii) BAS Transformed Noisy Reference (viii) PSO Transformed Noisy Reference (ix) PSO Transformed Non-Noisy Reference



(i)



(ii)

Figure 1. Comparison between (i) the reference image, which contains the object to be identified, and (ii) the target image, representing the landscape where the object is to be located. Both images have dimensions of 340x340 pixels.

6.1. PSO Transformed Images

- **PSO Transformed Non-Noisy Reference**: Shows a fairly accurate transformation of the target image, but with noticeable yellow noise at the center.
- **PSO Transformed Noisy Reference**: Similar transformation to the non-noisy reference but with more prominent noise.
- **PSO Transformed Non-Noisy with Noise Considerations**: Displays the target silhouette with additional noise, indicating the PSO algorithm's struggle with noise considerations.

6.2. BAS Transformed Images

- **BAS Transformed Non-Noisy Reference**: Shows a clear transformation with a well-defined silhouette.
- **BAS Transformed Noisy Reference**: Similar to the non-noisy reference but includes noise, demonstrating better noise handling than PSO.
- **BAS Transformed Non-Noisy with Noise Considerations**: Exhibits a clear silhouette with slight noise, indicating effective handling of noise.

7. Transformation Parameters Comparison Table and Objective Function Variation

In addition to the standard experimental setup, we conducted an extended experiment to evaluate the performance of Particle Swarm Optimization (PSO) and Beetle Antennae Search (BAS) algorithms under a modified objective function. The purpose of this variation was to measure the robustness and efficiency of the algorithms in optimizing image transformations when faced with a different objective function.

7.1. Experimental Setup

- **Images:** The experiments used a reference image and a target image, both loaded in grayscale format. The reference image was transformed in terms of position, scale, and rotation to align with the target image.
- **Objective Function:** The modified objective function was defined as the sum of squared differences between the pixel values of the transformed reference image and the target image. This objective function aims to minimize the error.

Parameter	PSO Non-Noisy	PSO Noisy	PSO Non-Noisy with Noise
X	169.74	170.24	104.30
Y	109.93	111.98	293.03
Scale	0.1001	0.1150	0.1110
Rotation	-157.60	-109.08	-116.29

Table 1. Comparison of transformation parameters found by PSO in each scenario.

Parameter	BAS Non-Noisy	BAS Noisy	BAS Non-Noisy with Noise
X	40.86	202.89	255.14
Y	230.04	42.17	240.42
Scale	-0.0011	0.0075	0.0220
Rotation	-132.79	159.61	-2.68

Table 2. Comparison of transformation parameters found by BAS in each scenario.

Scenario	PSO Change	BAS Change
Non-Noisy	1,063,184	877,406
Noisy	1,120,800	878,838
Non-Noisy with Noise	1,107,929	883,806

Table 3. Pixel-change objective values for PSO vs. BAS under each scenario. Lower is better.

7.2. Results and Images

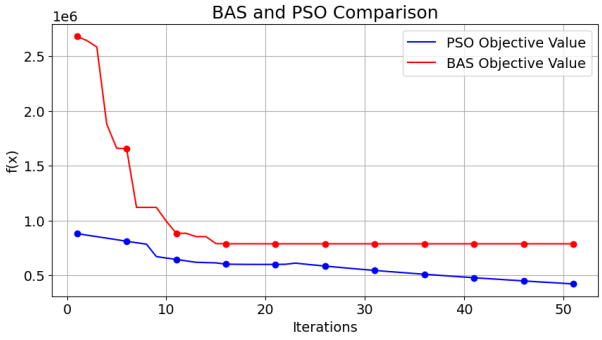


Figure 3. Objective Function Plot

The experiments were conducted in multiple trials, and the results were saved as comparative plots and images. Each trial produced the following visual results:

- **Target Image:** Displayed as the baseline for comparison.
- **Objective Function Plot:** A graph showing the objective function values over iterations for both PSO and BAS. This plot illustrated the convergence behavior and performance of each algorithm during the optimization process.

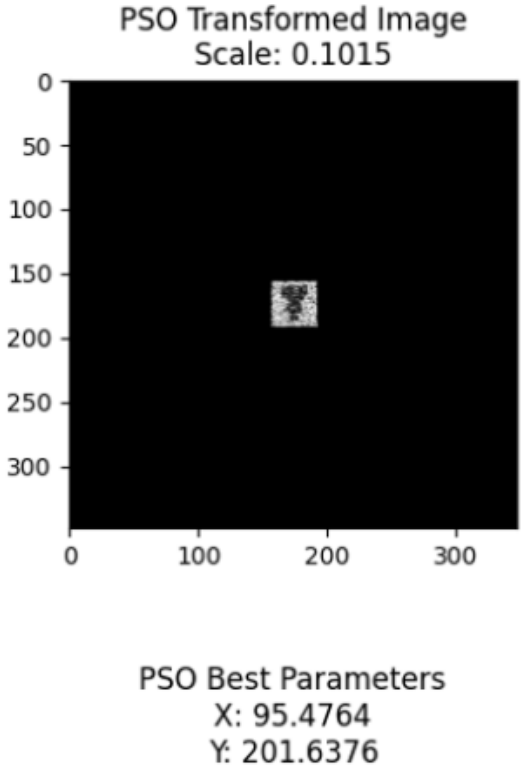


Figure 4. PSO Transformed Image

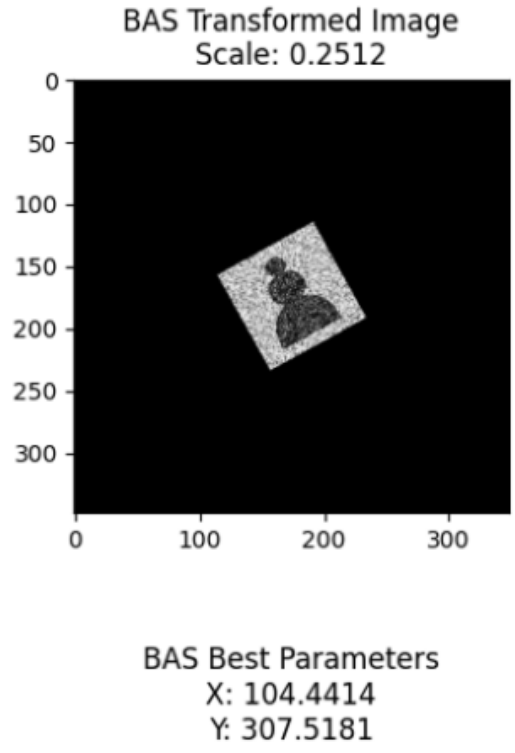


Figure 5. BAS Transformed Image

- **PSO Transformed Image:** The image obtained after applying the transformation parameters optimized by PSO. This image showed how effectively PSO could align the reference image with the target image.
- **BAS Transformed Image:** The image obtained after applying the transformation parameters optimized by BAS. This image demonstrated the efficiency of BAS in achieving accurate image alignment.

7.3. Explanation of the Images and Plot

The visual results and plot provided insights into the performance differences between PSO and BAS:

- **PSO Transformed Image:** The PSO algorithm struggled to optimize the transformation parameters effectively. The transformed image often exhibited limited accuracy in aligning with the target image, particularly in terms of scale and rotation.
- **BAS Transformed Image:** BAS consistently produced transformed images that closely aligned with the target image. The scale and rotation parameters were more accurately optimized, resulting in a higher quality transformation.
- **Objective Function Plot:** The plot revealed that BAS converged more rapidly and consistently achieved lower objective function values compared to PSO. This indicated that BAS was more efficient in minimizing the error and optimizing the transformation parameters.

8. Discussion and Conclusion

The results indicate that the BAS algorithm outperforms the PSO algorithm in all scenarios, demonstrating lower pixel-change values. This suggests that BAS is more effective in minimizing the error between the transformed reference image and the target image. The fine-tuning of BAS parameters contributed to its superior performance, highlighting the importance of parameter adaptation in swarm-based optimization.

From the visual results, BAS produces clearer alignments with fewer noise artifacts compared to PSO. Even under noise, BAS achieves a better silhouette match to the target image. The convergence plots confirm that BAS reaches lower objective values more rapidly than PSO, indicating higher robustness and efficiency.

In conclusion, our comparative analysis of BAS and PSO in both standard and extended experiments reveals that BAS is a superior optimizer for image transformation tasks. The findings emphasize the importance of selecting an appropriate optimization algorithm based on problem characteristics.Future work can explore BAS in other image processing applications, leveraging its robustness and efficiency to achieve high-quality results.

References

[1] M. Hu, W. Yan, and F. Zheng, "A review on swarm intelligence in image processing," *Neurocomputing*, vol. 451, pp. 163–177, 2021.

[2] A. M. V. Coelho, G. Melançon, and C. Popoulas, "Swarm intelligence for image registration: A survey and new perspectives," *Swarm and Evolutionary Computation*, vol. 82, p. 101197, 2023.

[3] J. Kennedy and R. Eberhart, "A new optimizer using particle swarm theory," in *Proceedings of the 6th International Symposium on Micro Machine and Human Science (MHS'95)*, (Nagoya, Japan), pp. 39–43, 1995.

[4] A. G. Gad, "Particle swarm optimization algorithm and its applications: A systematic review," *Archives of Computational Methods in Engineering*, vol. 29, pp. 2531–2561, 2022.

[5] J. Wang and H. Chen, "Bsas: Beetle swarm antennae search algorithm for optimization problems," *arXiv preprint arXiv:1807.10470*, 2018.

[6] A. T. Khan, X. Cao, and S. Li, "Using quadratic interpolated beetle antennae search for higher dimensional portfolio selection under cardinality constraints," *Computational Economics*, vol. 62, no. 4, pp. 1187–1214, 2023.

[7] A. T. Khan and S. Li, "Human-guided cooperative robotic agents in smart home using beetle antennae search," *Science China Information Sciences*, vol. 64, no. 3, pp. 1–15, 2021.

[8] A. T. Khan, X. Cao, and S. Li, "Fraud detection in publicly traded u.s. firms using beetle antennae search: A machine learning approach," *Expert Systems with Applications*, vol. 191, p. 116148, 2022.

[9] A. T. Khan and S. Li, "Smart surgical control under rcm constraint using bio-inspired network," *Neurocomputing*, vol. 470, pp. 121–129, 2022.

[10] A. T. Khan, X. Cao, and S. Li, "Dual beetle antennae search system for optimal planning and robust control of 5-link biped robots," *Journal of Computational Science*, vol. 60, p. 101556, 2022.

[11] A. T. Khan, S. Li, and X. Zhou, "Trajectory optimization of 5-link biped robot using beetle antennae search," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 10, pp. 3276–3280, 2021.

[12] A. T. Khan, S. Li, S. Kadry, and Y. Nam, "Control framework for trajectory planning of soft manipulator using optimized rrt algorithm," *IEEE Access*, vol. 8, pp. 171730–171743, 2020.

[13] A. Abualigah and et al., "Particle swarm optimization: Advances, applications, and experimental insights," *Computers, Materials & Continua*, vol. 82, no. 2, pp. 1539–1592, 2025.

[14] F. Salehi, M. Khosrojerdi, and M. Parsa, "Beetle antennae search with chaos strategy for global optimization," *Knowledge-Based Systems*, vol. 223, p. 107028, 2021.

[15] Q. Qian, Y. Deng, H. Sun, Z. Fu, and J. Tian, "Enhanced beetle antennae search algorithm for complex and unbiased optimization," *Soft Computing*, vol. 26, pp. 10331–10369, 2022.
[16] B. Yin, L. Mo, W. Min, S. Li, and C. Yu, "An improved beetle antennae search algorithm and its application in coverage of wireless sensor networks," *Scientific Reports*, vol. 14, p. 29372, 2024.
[17] X. Shan, S. Lu, B. Ye, and M. Li, "Hybrid strategy improved beetle antennae search algorithm and application," *Applied Sciences*, vol. 14, p. 3286, 2024.
[18] L. Zhang, J. Wang, G. Meng, and R. Luo, "Multi-objective beetle antennae search algorithm for integrated optimization problems," *Journal of Bionic Engineering*, vol. 17, no. 1, pp. 139–156, 2020.
[19] X. Wu, X. Zhang, and Y. Dong, "A novel neural network classifier using improved beetle antennae search algorithm," *Neural Computing and Applications*, vol. 31, no. 12, pp. 9331–9344, 2019.
[20] X. Wu, X. Zhang, and Y. Dong, "Beetle antennae search algorithm for uav path planning," *Information Sciences*, vol. 507, pp. 413–425, 2020.
[21] M. Lin, Q. Li, F. Wang, and D. Chen, "An improved beetle antennae search algorithm and its application on economic load distribution of power system," *IEEE Access*, vol. 8, pp. 99624–99632, 2020.
[22] X. Xu, K. Deng, and B. Shen, "A beetle antennae search algorithm based on lévy flights and adaptive strategy," *Systems Science & Control Engineering*, vol. 8, no. 1, pp. 35–47, 2020.
[23] H. Zhao, H. Yao, Y. Jiao, T. Lou, and Y. Wang, "An improved beetle antennae search algorithm based on inertia weight and attenuation factor," *Mathematical Problems in Engineering*, vol. 2022, p. 7391145, 2022.
[24] X. Shao and Y. Fan, "An improved beetle antennae search algorithm based on the elite selection mechanism and the neighbor mobility strategy for global optimization problems," *IEEE Access*, vol. 9, pp. 137524–137542, 2021.
[25] B.-L. Liao, Z.-D. Huang, X.-W. Cao, and J. Li, "Adopting nonlinear activated beetle antennae search algorithm for fraud detection of public trading companies: A computational finance approach," *Mathematics*, vol. 10, no. 13, p. 2160, 2022.
[26] J. Qiao, G. Wang, Z. Yang, X. Luo, J. Chen, K. Li, and P. Liu, "A hybrid particle swarm optimization algorithm for solving engineering problems," *Scientific Reports*, vol. 14, p. 8357, 2024.
[27] R. Salgotra, A. K. Lamba, D. Talwar, D. Gulati, and A. H. Gandomi, "A hybrid swarm intelligence algorithm for region-based image fusion," *Scientific Reports*, vol. 14, p. 13723, 2024.
[28] S. Saifullah and R. Dreżewski, "Advanced medical image segmentation enhancement: A particle-swarm-optimization-based histogram equalization approach," *Applied Sciences*, vol. 14, p. 923, 2024.
[29] J. Yao, X. Luo, F. Li, J. Li, J. Dou, and H. Luo, "Research on hybrid strategy particle swarm optimization algorithm and its applications," *Scientific Reports*, vol. 14, p. 24928, 2024.
[30] X. Zhang, Y. Ren, G. Zhen, Y. Shan, and C. Chu, "A color image contrast enhancement method based on improved pso," *PLOS ONE*, vol. 18, no. 2, p. e0274054, 2023.
[31] M. Bahrololum, E. Akbari, and S. Mir, "Enhanced fireworks algorithm and particle swarm optimization for image registration," *Expert Systems with Applications*, vol. 185, p. 115666, 2022.
[32] Y. Zhou, C. Zhang, and F. Tang, "Bee colony optimizer for unimodal and multimodal function optimization," *Soft Computing*, vol. 26, pp. 8409–8429, 2022.
[33] J. Gao, H. Li, and J. Liu, "Plant engineering using bas-based metaheuristics for image alignment," *Optik*, vol. 262, p. 169925, 2023.
[34] A. Kumar and P. Singh, "Hybrid pso and artificial bee colony algorithm for medical image registration," *Biomedical Signal Processing and Control*, vol. 82, p. 104391, 2024.
[35] Y. Liu, D. Wang, and J. Zhang, "Enhanced particle swarm optimization algorithm for high-precision image mosaic," *Journal of Visual Communication and Image Representation*, vol. 83, p. 103619, 2022.
[36] S. Pugazhenthi, P. Malliga, and R. Gopalan, "Neuro-fuzzy based metaheuristic for image registration," *Journal of Intelligent & Fuzzy Systems*, vol. 44, no. 3, pp. 5525–5538, 2023.
[37] M. Rodrigues and M. E. F. Barbosa, "A survey of metaheuristics for image processing and analysis," *Pattern Recognition Letters*, vol. 169, pp. 16–27, 2024.
[38] A. T. Khan, X. Cao, B. Liao, and A. Francis, "Bio-inspired machine learning for distributed confidential multi-portfolio selection problem," *Biomimetics*, vol. 7, no. 3, p. 124, 2022.