Assignment 3

Swarm Intelligence

Deadline March 10th, 2025

Handout for the *Natural Computing* lecture, February 17th, 2025

TA for this assignment: Koert Schreurs

In the lecture on swarm intelligence, you saw Craig Reynold's Boid model used for simulating swarming behavior [1], and the particle swarm optimization (PSO) method which is inspired by swarming behavior but used to solve general continuous optimization problems [2, 3]. In this assignment, you will implement and explore both algorithms.

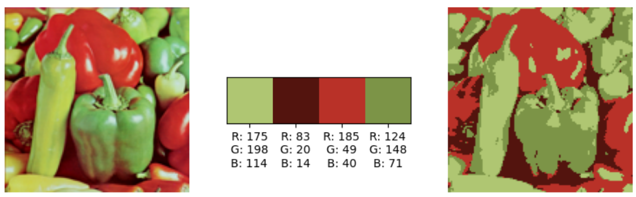
Objectives of This Exercise

1. Implement the boid simulation model of crowd motion
2. Analyze the effect of boid parameters on swarming behavior
3. Implement the particle swarm optimization for color quantization in images and compare it to another algorithm

Exercises

Exercise 3.1 Find an optimal color palette using Particle Swarm clustering

As seen in the lecture, particle swarm optimization (PSO) can be used to solve continuous optimization problems. One such problem is clustering. Pseudocode for PSO and its implementation in clustering are provided in the Appendix to this assignment. Before starting this exercise, have a look at this appendix.



1. An image can be compressed by reducing the number of distinct colors in it; this is called color quantization. Implement a function that takes an input image and a color palette, and returns a quantized image. Create a figure to visualize that your method works. (The original and quantized image shown above are provided with the assignment; please verify that you can reproduce the quantization with the provided colormap).
2. Finding the optimal color palette for an image can be seen as a clustering problem. How? What would the points and clusters represent?
3. Implement PSO clustering in order to perform color quantization on an image. Notes:
   - What would a particle represent?
   - How would you measure the quality or "fitness" of a particle?
   - If the PSO algorithm takes very long, try a smaller image (for example 128 by 128) or resize your image to a smaller size.
4. Make sure that at each iteration, your implementation stores the state of the particles and the global best particle. At several iterations of your PSO algorithm, show both the best color palette found so far, and the image quantized with this palette.
5. A popular method for clustering is K-means. Use a K-means implementation to find the best color palette to quantize the image. Compare your results with PSO, which method is better?

"Vanilla" PSO is not applied often because better and simpler alternatives are often available. A major problem with PSO is that it can be slow to converge and can easily get stuck in local optima [4]. However, research is still ongoing in variations of PSO, for example by hybridizing it with elements of evolutionary algorithms [4, 5].

Exercise 3.2 Implement the Boids model

1. Implement your own version of Craig Reynold's Boid Algorithm for swarm motion [1]. Note: **you do not need to implement the vision angles of the boids, although you are of course welcome to try if you would like to.** We will provide an example JavaScript implementation with this assignment, where we took care of some of the general setup and visualization, but you will still need to implement the update rule. If you want, you are of course welcome to implement the entire model yourself. In that case, please make sure that your model:
   - ... is able to generate text output (for example, positions of Boids over time) for use later. If you use the JavaScript implementation, see the README for instructions.
   - ... keeps Boid speeds fixed; agents should update their direction but move at a constant speed v which is a parameter of your model.
   - ... explicitly considers what should happen at the boundary of your simulation field.
   - ... uses two interaction radii: one for separation, and one for cohesion/alignment. (Ask yourself: which one should be smaller and why?)
2. First analyze your implementation visually (for example by creating a time-lapse showing Boid positions and directions over time). Using the analyses above, play with the Boid parameters (cohesion, separation, and alignment strength; exclusion and interaction radius) to verify that it is possible to reach high alignment in a run of about 300 steps (Note: make sure that the movement speed, overall Boid density, and interaction radii are in proportion to each other). Make sure to show this (visually) in your report.
3. To analyze your implementation in more depth, you can perform some quantitative analysis. For example, we suggest you look at:
   - the *order parameter* over time. The order parameter is the average normalized velocity of the Boids:
     $$O = \frac{1}{N_B} \left\| \sum_{i=1}^{N_B} \frac{\vec{v}_i}{||\vec{v}_i||} \right\| \tag{1}$$
     where the outer norm ensures that we get one non-negative value and the inner normalization ensures it lies between 0 and 1.
   - the distribution of *nearest-neighbor distances* over time, which should tell you when individuals get separated from the swarm and/or when individuals get too close to each other.

   In your simulation, what happens to the nearest-neighbor distance and alignment over time? Explain your observations (and don't forget to report which parameters you chose for this result).
4. Set up and execute small experiment to test the effect of Boid parameters on the behavior of the system. You have some freedom here, but please justify which parameters you consider and why.

Product

Write a small research report (approximately 4-5 pages in pdf) about this assignment. You can split your report in a part A and a part B for the PSO and Boid parts of the assignment.

Assessment criteria

For PSO:

- Methodology/implementation: your implementation should be able to quantize an image correctly, and it should be clear how exactly you phrased the image quantization problem in terms of PSO.
- Evaluation/analysis: you should have a sensible analysis showing how the quantization is optimized over time, and evaluating the final product, comparing it to K-means.

For Boids:

- Experimental design: how did you set up your experiment evaluating the effect of Boid parameters?
- Methodology/implementation: there should be no major bugs/artefacts in your Boid simulation. The details of your implementation should be clear from the written report without referring to the code (although you may still do that).
- Analysis: your analysis of the Boid simulation should be properly supported by both visualizations and quantitative analysis. Are they appropriately chosen and executed?
- if you upload any movies of your simulation(s) with your assignment: please ensure that you use some generic video format that can be opened from any computer. Please don't embed them in the pdf file as this can be tricky with portability. Don't rely on videos only; some (non-moving) visualization of your main results should be clearly visible in the report itself.

For both:

- Interpretation: Did you correctly interpret your results? Did you clearly answer the question/solve the problem, and are all your claims backed up by evidence in the form of visualizations and/or quantitative analyses (depending on the claim made)?
- Reporting: does your report meet basic standards of a research report? It should be comprehensible and follow a logical structure with an introduction containing the problem statement, methods, results, discussion and conclusion. You can keep the introducution/problem statement brief (i.e. you don't need to embed it in literature extensively for this assignment), but it should be clear what problem(s) you are addressing.
- Be sure to come to a final conclusion and discuss some potential limitations as well.

Feel free to double-check yourself using the Peer feedback rubric from assignment 1B: the specific choices will be different for this assignment, but many of the principles remain the same.
Hand in your report in pdf format on Brightspace; deadline: March 10th, 2025.

References

[1] C. W. Reynolds (1987) *Flocks, herds and schools: A distributed behavioral model*, In Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '87, page 25–34, New York, NY, USA, 1987. Association for Computing Machinery.

[2] Eberhart R, Kennedy J (1995). *A new optimizer using particle swarm theory.* MHS'95. Proceedings of the sixth international symposium on micro machine and human science. IEEE, pp 39–43. doi: `https://doi.org/10.1109/MHS.1995.494215`.

[3] Kennedy J, Eberhart R (1995). *Particle swarm optimization (pso).* Proceedings of IEEE international conference on neural networks, Perth, Australia, pp 1942–1948. doi: `https://doi.org/10.1109/ICNN.1995.488968`.

[4] Gad, A.G (2022). *Particle Swarm Optimization Algorithm and Its Applications: A Systematic Review.* Arch Computat Methods Eng 29, 2531–2561. doi: `https://doi.org/10.1007/s11831-021-09694-4`.

[5] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh and S. Mirjalili (2022). *Particle Swarm Optimization: A Comprehensive Survey.* IEEE Access, vol. 10, pp. 10031-10061, doi: `https://doi.org/10.1109/ACCESS.2022.3142859`.

A PSO

Algorithm 1: Particle Swarm Optimization

input: **evaluation function** f, **parameters** $\omega, \alpha_1, \alpha_2$
Initialize positions $\mathbf{x}_1, \ldots, \mathbf{x}_n$ at random, velocities $\mathbf{v}_1, \ldots, \mathbf{v}_n \leftarrow \mathbf{0}$, local best $\hat{\mathbf{x}}_i \leftarrow \mathbf{0}$, global best $\hat{\mathbf{g}} \leftarrow \mathbf{0}$
while *not happy* **do**
  foreach $i \in \{1, \ldots, n\}$ **do**
    Draw vectors $\mathbf{r}_1, \mathbf{r}_2$ componentwise from $\mathcal{U}[0, 1]$
    $\mathbf{v}_i \leftarrow \omega \mathbf{v}_i + \alpha_1 \mathbf{r}_1 (\hat{\mathbf{x}}_i - \mathbf{x}_i) + \alpha_2 \mathbf{r}_2 (\hat{\mathbf{g}} - \mathbf{x}_i)$
  end
  foreach $i \in \{1, \ldots, n\}$ **do**
    $\mathbf{x}_i \leftarrow \mathbf{x}_i + \mathbf{v}_i$
    if $f(x_i) < f(x_i)$ **then**
      $\hat{\mathbf{x}}_i \leftarrow \mathbf{x}_i$
    end
    if $f(x_i) < f(\hat{\mathbf{g}})$ **then**
      $\mathbf{g} \leftarrow \mathbf{x}_i$
    end
  end
end

B PSO Clustering

Each particle represents a list of N cluster centroids:
$\mathbf{x}_i = (\mathbf{m}_{i1}, \ldots, \mathbf{m}_{iN})$
Additionally we have a data set: $Z = \{z_1, \ldots, z_M\} \in \mathbb{R}^d$ with M points to be clustered.
Our evaluation function $f(x_i)$ becomes $f(x_i, Z)$. We evaluate the clustering of points Z and the proposed centroids x_i as follows:

Algorithm 2: PSO Clustering Evaluation function

Function $\mathtt{f}(x_i, Z)$**:**
  foreach $j \in \{1, \ldots, M\}$ **do**
    Assign $\mathbf{z}_j$ to nearest $\mathbf{m}_i$
  end
  return evaluate the clustering