2017

The Impact of Varying Sample Size of Particle Filtering on Estimation the Pose of 2D Objects: An Experimental Report

PARTICLE FILTER SEARCH

GROUP 9: LINNI QIN N9632981 JIUZHENG RONG N9814434

IFN680 | Assignment One

Table of Contents

I. Introduction

1.1 Background Statement

While applying the machine learning algorithm to detect the pose of targeted objects, edge-based method requires cheaper computation than other pose estimation approaches does, such as keypoints-based approaches (Frederic, 2017, p. 1). In the context of this report, a variant of particle filter algorithm is implemented efficiently as the searching method to estimate the pose vector of a 2-dimensional geometry shape. This algorithm will be explained in the next paragraph. In the meantime, the targeted object is the larger triangle displayed in the left *Figure 1* **imf**. Just like other patterns, it has 4-degree of freedoms for pose vector, which are x cords, y cords, theta and scale. The other known element about this triangle is the distance and location of the pixels for its three edges. They are at the deep valley in the dark blue zone as indicated in the right float *Figure 2* **imd**.

Figure 1

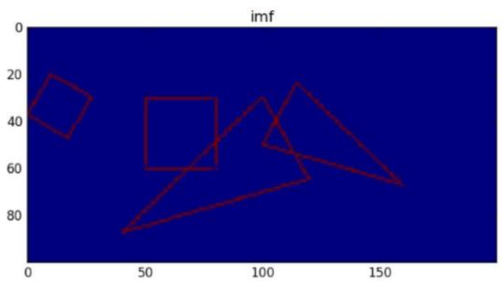
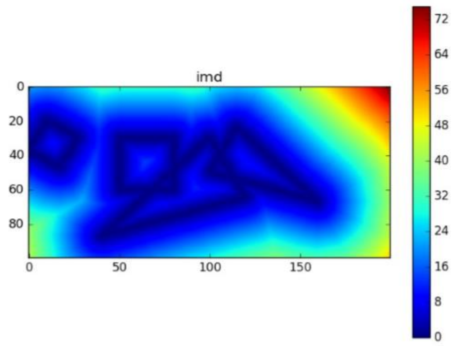


Figure 2



1.2 Brief Description of Particle Filter Algorithm

"# Particle filter search

*Initialize population **W** with random guesses of pose vectors*

***Loop** until computational budget exhausted*

> ***evaluate** the cost **C[i]** of each **W[i,:]***
>
> ***re-sample** the population according to **exp(C[i])***
>
> ***mutate** each new individual **W[i,:]** by adding some noise" (Frederic, 2017, p. 3).*

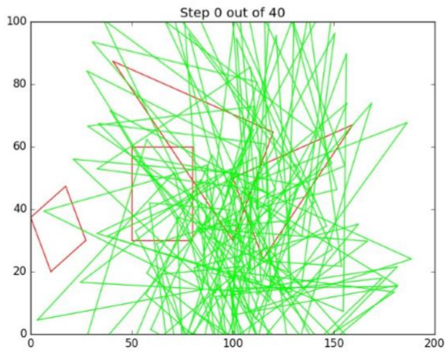


Figure 3 A random collection of W

Considering the particle filter algorithm applied in this assignment, it generates initially a collection of hypotheses (population ***W***) that are called particles as well from a random distribution. Each of them (***W[i,:]***) has four state variables as discussed above, including x cords, y cords, theta and scale. The initial particles with random poses are displayed all over the image in green colour as shown as the *Figure 3*.

Then the program will iterate until the number of generation runs out. During each loop, the distance of certain particle to the target vector (100, 30, pi/3, 40) of the larger triangle is recorded as ***C[i]*** that denotes the cost of the particle ***W[i,:]***. Following obtaining the cost, the program generates new particles as the stated the re-sample procedure. However, they are not randomly distributed over the entire map starting from this step. They are re-generated based on previous particles with respect to their cost. After the re-sample, mutation is proceeded on individuals with equal probability (1/3) with moving 1 degree in three different directions (-1, 0, +1) to avoid the convergence being stuck at a local minimum. The best solution (*self*.best_w), the improvement of the performance of best weight (*self*.best_cost) and the convergence at the last step of iteration will be the essential factors to be detected as the final aim of applying particle filtering. Those three factors look largely similar to the *Figure 3* shown below.

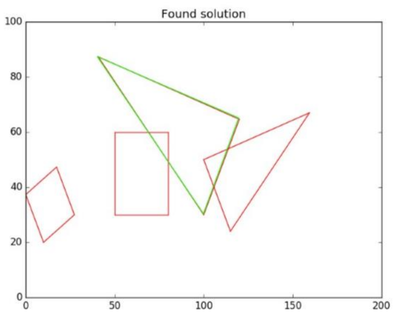


Figure 4 Best Solution

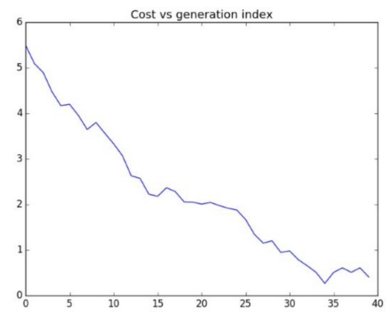


Figure 5 Best Cost

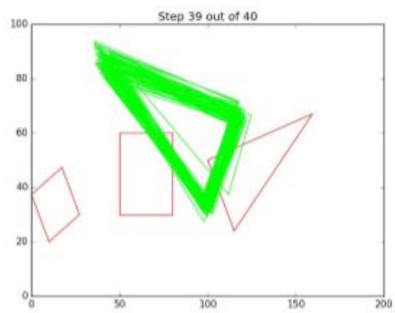


Figure 6 Best Convergence

1.3 Purpose of Report

This report is going to explore the performance of the three outcomes stated in *Figure 3* with varying finite sample sizes (*self*.**W**) by balancing the number of iteration and the size of the individual in each generation. The guidelines of experiment will be discussed in the following section two (Experimental Method) with separating into three steps. Section three (Experimental Results) will demonstrate the related outcomes. Finally, the conclusion with a rational suggestion will be offered about the optimal balance of the population size and the number of generation to maximize the effectiveness and efficiency of the program.

II. Experimental Method

2.1 Experiment Environment

The following computing environment is for deploying the particle filter search and the relevant testing experiments.

Operation Systems: Windows 7 Enterprise System x64-based

CPU: Intel® Xeon® CPU E5-2687W v4 @ 3.00GHz

Memory(RAM): 4.00 GB

Programming and diagnosing Tool: Python 3.6.0 and Spyder 3.1.2

2.2 Experiment Approach

Due to the random sampling of individuals in selection, manually setting and repeatedly trying the number of iteration and the scale of population needed for each individual step is used until the best solution, the best cost pattern and the best convergence could be found. The relevant parameters can be configured in the defined function ***test_particle_filter_search()*** as below in the enclosed Python file ***my_submission***.

Code for setting the size of individuals for each generation: ***pop_size = ##***

Code for setting the number of the iteration*: **Lw, Lc = pop.particle_filter_search(##, log=True)***

There are three main steps are used to guide the choice of the relating sizes.

2.2.1 Step one: Determining the suitable particle budget

Following the policy of varying the size of population from small to large, we started from 5 individuals and the individual number was incrementally multiplied by an integer number increasing by 1. For example, *pop_size = 5, 10, 15, 20* and so on. The generation was increased with interval of 10 until reaching the maximum 200 times of generation. The reason we did not start from less than 5 individuals is that the cost comparison requires at least two individuals. Furthermore, the minimum cost found before filtering might be the best cost ever within the small size of population. This situation will cause a blank cost image with high probability.

Given this step is for detecting the particle budget, we repeated the trial with 5 times to identify the rough development trend of the cost performance solely so that we could zoom at a small range for balancing the size. To improve the test efficiency, a loop function was added within ***test_particle_filter_search()*** for automatic execution. Then, the approximate particle budget can be analysed from the visible cost images discussed in the third section (Experimental Results).

2.2.2 Step two: Balancing the number of generation and the size of the population

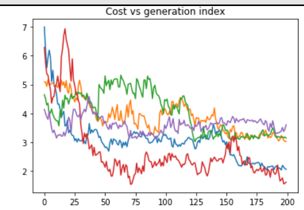
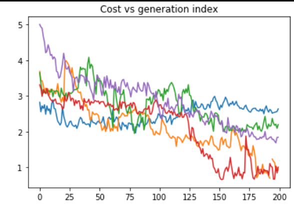
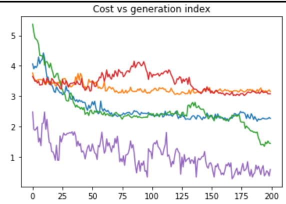
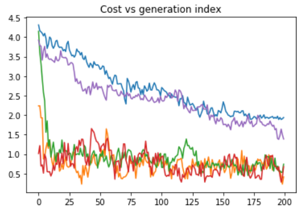
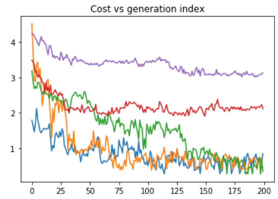
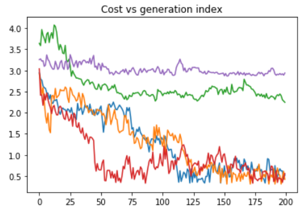
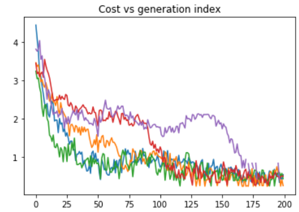
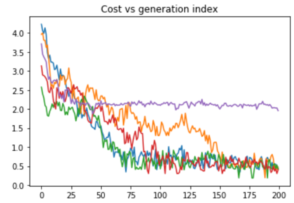
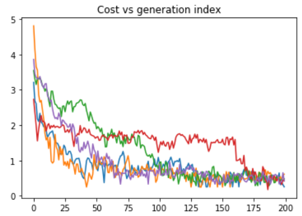
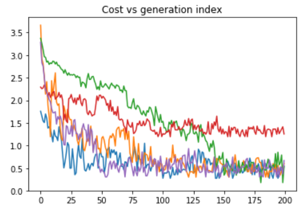
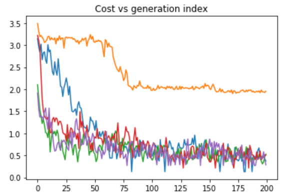
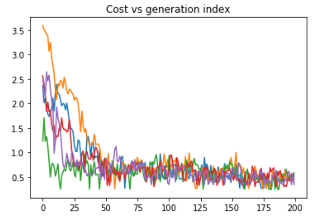
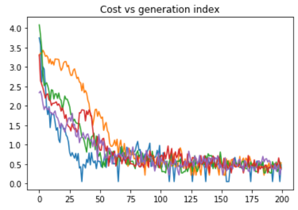
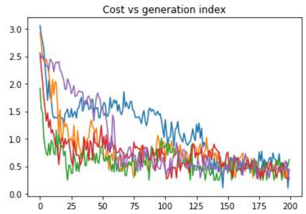
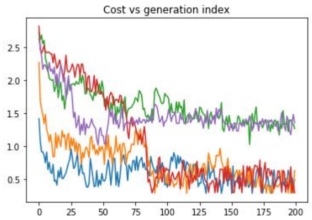
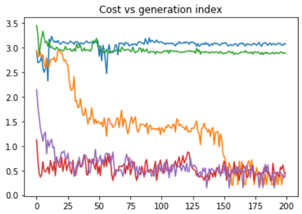
After finding the particle budget, the test scope is narrowed down. We enlarged the testing times to 30 and decreased the interval of generation to 5. In the meantime, a method to calculate the accuracy rate was used within ***test_particle_filter_search()*** to identify the best combinations. We recorded the counter number when the cost was returned with value larger than 1. Then, the rate can be calculated with dividing the total number of generation by the counter number. The relating results will be shown in a table within experimental results section.

2.2.3 Step three: Confirming the performance of the chosen combination

Consequently, the other two essential factors – best solution and best convergence – are involved into the testing so that we can prove the best combination of the size we found so far works effectively.

III. Experimental Results

3.1 Results for particle budget determination

pop_size = 5, generation max 200	***pop_size = 10, generation max 200***	***pop_size = 15, generation max 200***
		
pop_size = 20, generation max 200	***pop_size = 25, generation max 200***	***pop_size = 30, generation max 200***
		
pop_size = 35, generation max 200	***pop_size = 40, generation max 200***	***pop_size = 45, generation max 200***
		
pop_size = 50, generation max 200	***pop_size = 55, generation max 200***	***pop_size = 60, generation max 200***
		
pop_size = 65, generation max 200	***pop_size = 70, generation max 200***	***pop_size = 75, generation max 200***
		
pop_size = 80, generation max 200	***pop_size = 85, generation max 200***	
		

As shown in the above high-lighted images with orange heading, starting from 50 to 75 of population size, the cost performs overall better than others with stably descending. Until the size of individual turns more than 75, the performance turns worse again. Regarding the generation number needed for the step 3.2, we chose the step when most of the cost converges as a centre value basing on respective images. Then, the number fluctuates within the range of -10 and +10.

3.2 Accuracy rate for targeted combination

The accuracy rate is calculated relying on the cost value which is smaller than 1. The significantly high accuracy rates (larger than 90%) appear within eight groups after 30 times of replicated testing. The following experiment about testing the overall performance will indicate

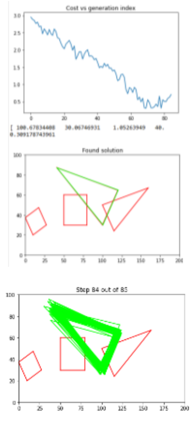
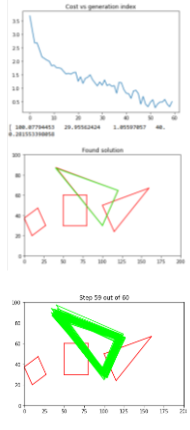
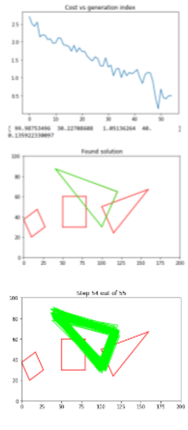
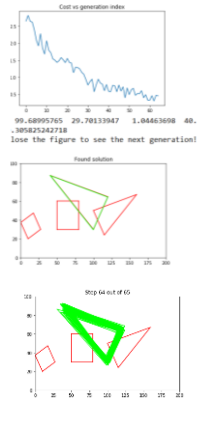
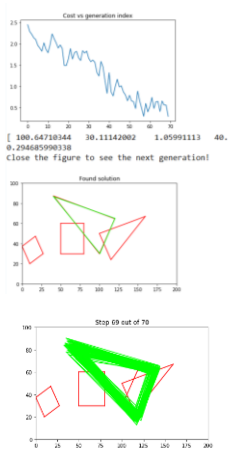
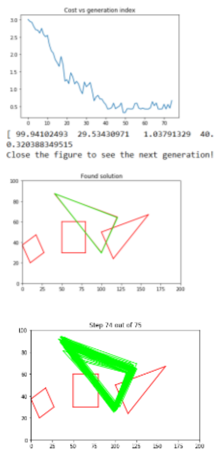
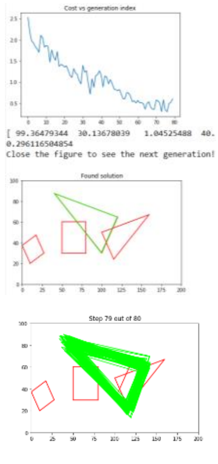
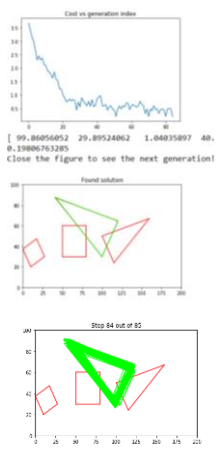
Table 1: Accuracy rate for targeted combination

pop_size * generation #	**Total Individual Size**	**Accuracy Rate**	**pop_size * generation #**	**Total Individual Size**	**Accuracy Rate**
50 * 90	4,500	80%	55 * 80	4,400	87%
50 * 95	4,750	73%	55 * 85	4,675	67%
50 * 100	5,000	87%	55 * 90	4,950	80%
50 * 105	5,250	67%	55 * 95	5,225	73%
50 * 110	5,500	83%	55 * 100	5,500	77%
pop_size * generation #	**Total Individual Size**	**Accuracy Rate**	**pop_size * generation #**	**Total Individual Size**	**Accuracy Rate**
60 * 70	4,200	80%	65 * 60	3,900	93%
60 * 75	4,500	67%	65 * 65	4,225	90%
60 * 80	5,800	80%	65 * 70	4,550	83%
60 * 85	6,100	93%	65 * 75	4,875	77%
60 * 90	6,400	90%	65 * 80	5,200	80%
pop_size * generation #	**Total Individual Size**	**Accuracy Rate**	**pop_size * generation #**	**Total Individual Size**	**Accuracy Rate**
70 * 50	3,500	83%	75* 65	4,875	97%
70 * 55	3,850	93%	75 * 70	5,250	97%
70 * 60	4,200	80%	75 * 75	5,625	100%
70 * 65	4,550	77%	75* 80	6,000	97%
70 * 70	4,900	70%	75 * 85	6,375	100%

3.3 Performance of optimal combination

As mentioned above in the third step of experiment method, all three factors for evaluating the overall performance of particle filtering are involved in this part of outcome, including the best solution, the best cost and the best convergence. Learning from the *Table 2* below, it can be able to confirm that the identified optimal combinations with applying our experimental approaches work greatly as expected.

Table 2: Performance of optimal combination

pop_size * generation # 60 * 85	pop_size * generation # 65 * 60	pop_size * generation # 70 * 55	pop_size * generation # 75 * 65
			
pop_size * generation # 75 * 70	**pop_size * generation #** 75 * 75	**pop_size * generation #** 75 * 80	**pop_size * generation #** 75 * 85
			

IV. Conclusion

In general, if a small size of total individual is chosen, there is a high risk of poor performance resulting from high loss and non-convergence when the maximum of the iteration is reached. If a large size of total individual is chosen without effectively balancing the size of population and the number of the iteration, high loss or overfitting can be caused as well. These problems can be referred to the images presented in section 3.1. One idea to overcome these problems is to use the method presented in this report to identify the optimal balance values for the size of population and the number of iteration. The optimal choice found in this experiment follows this trend: there is a small gap between the size of the population and the number of the iteration. The gap might be around the range of ±10 as proved in our experiment.

For further experiment, it needs an applicable method to calculate precisely the step of generation at which all the cost starts to converge while determining the particle budget, rather than estimating the approximate step of the generation with the way of vision. In that case, the results of the testing might be able to avoid a certain of overfitting problems.

Reference

Frederic, M., (2017). 2017_IFN680_assignment_1. Retrieved from https://blackboard.qut.edu.au/bbcswebdav/pid-6983471-dt-content-rid-9254318_1/courses/IFN680_17se2/2017_IFN680_assignment_1%281%29.pdf