Artificial Intelligence

Project2　Particle Filter

信息与电子工程学院　信息工程 [redacted]

[redacted]

May5-7, 2023

1 Introduction

This project has completed a object tracking algorithm based on particle filter. The algorithm can be divided into four steps.

First is Transition, we take out each particle and randomly sample with Gaussian distrbution. The mean is the parameters of the particle, and the variance is determined by ourselves.

Second is Weighting. We first obtain the rectangular area of the particles after transfer, extract the features in the frame, use the obtained features and the features of the particles before transfer to calculate the similarity, and get the weight of each particle according to the weight of the similarity. There are two details here. The first is feature extraction. Feature extraction can use the method based on pixel intensity or the HOG method. Both methods have been tried in this project. The second detail is similarity. We can use Euclidean distance, Manhattan distance, Chebyshev distance, Hamming distance. But this project adopts cosine method to calculate the similarity. Its expression is:

$$\text{Similarity}(\mathbf{a}, \mathbf{b}) = \cos\theta = \frac{\mathbf{a} \cdot \mathbf{b}}{||\mathbf{a}|| \cdot ||\mathbf{b}||}$$

Third is Decision, this project first adopts the simplest strategy, that is, the particle with the highest weight is considered to be the result of object tracking. Further detailed explanation of other algorithms during the section of Optimize.

Fourth is Re-sampling. In this step, Gaussian sampling is performed with the determined particle parameters to obtain a new group of particles, and then the above operation is repeated with this group of particles.

The basic code is in the file src, and the optimized code is in the file src_optimize. All experimental results are in the data folder, with the pre-optimized results located in results1 folder and the optimized results located in results2 folder.

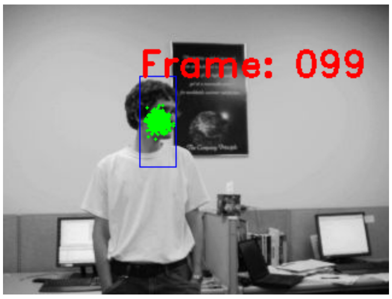
2 Result Analysis

2.1 Different Feature extraction

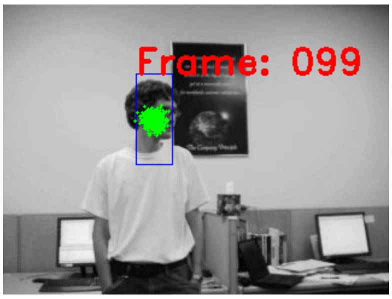
Let the number of particles be 400 and the step be 1, both intensity and HOG features can track the target object accurately, which can be seen as follows:

(a) car by intensity feature

(b) car by HOG feature



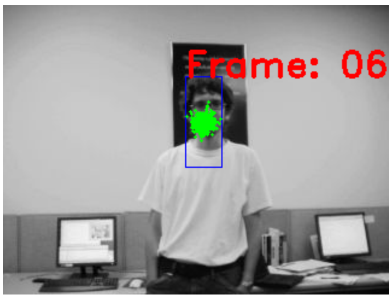
(c) David2 by intensity feature



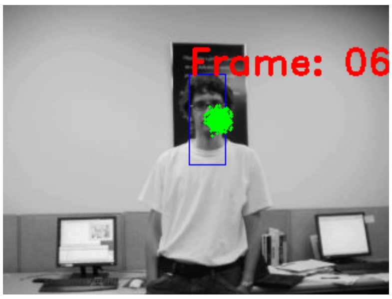
(d) David2 by HOG feature

Fig. 1: set $n\_particles = 400$, $step = 1$

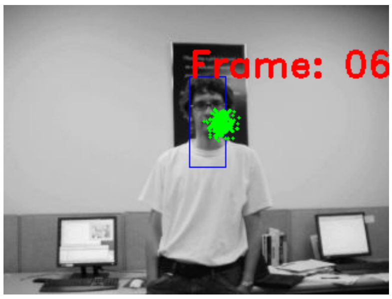
Similarly, when increase step to 2, these feature types can work right too, the images are saved in result folder. However, if we increase step more, the algorithm may not track the object accurately any longer. For example, let's set $step = 6$, while keeping $n\_particle = 400$, the result of both feature types is shown in Fig.2. It can be found that the accuracy of both cases has decreased, but the position of the rectangular frame can still track the target object.



(a) Ideal case



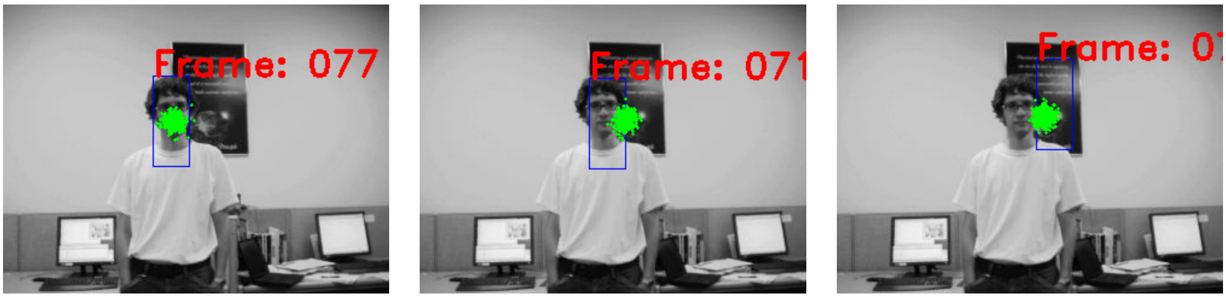
(b) Intensity feature



(c) HOG feature

Fig. 2: set $n\_particles = 400$, $step = 6$

let's set $step = 10$, while keeping $n\_particle = 400$. It can be found that both situations cannot track the target well, with HOG performing even worse. Not only is the particle distribution worse, but the position of the rectangular box is also significantly different from the target object, while the position of the intensity rectangular is still close to the target object.

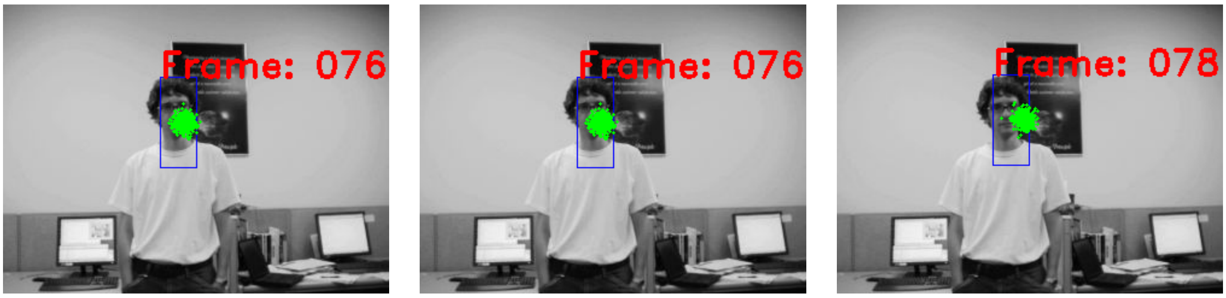


(a) Ideal case (b) Intensity feature (c) HOG feature

Fig. 3: set $n\_particles = 400$, $step = 10$

2.2 Different Step

In the previous section's discussion, it can be observed that the tracking effect gradually deteriorates with the increase of step. The following are the tracking results of the intensity feature changing with step:

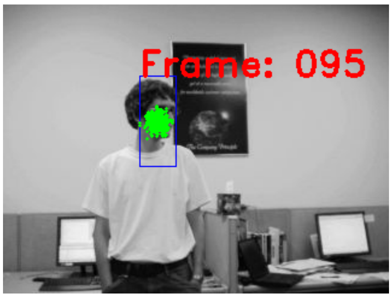


(a) $step = 3$ (b) $step = 5$ (c) $step = 7$

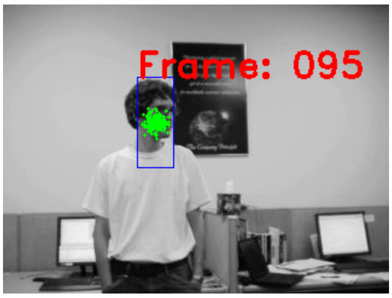
Fig. 4: set $n\_particles = 400$, Intensity feature

2.3 Different Particle number

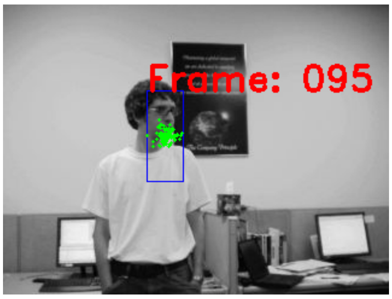
By gradually reducing the number of particles, it can be observed that the tracking effect gradually deteriorates. When the number of particles decreases to 50, it can be found that the final goal tracking effect is almost impossible to achieve. The following are the tracking results for different particle numbers when $step = 1$, Intensity feature:



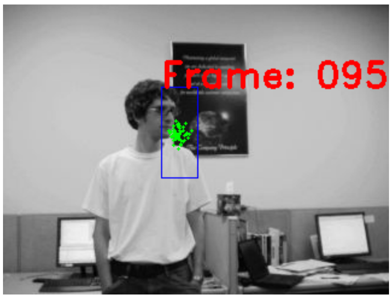
(a) $n\_particle = 400$



(b) $n\_particle = 200$



(c) $n\_particle = 100$



(d) $n\_particle = 50$

Fig. 5: set $step = 1$, Intensity feature

2.4 Cost time

Testing the running time of programs with different step sizes and particle numbers on the same dataset reveals that:

1. The running time of HOG is greater than Intensity;
2. As the step size increases, the running time decreases;
3. As the number of particles decreases, the running time decreases;

This is in line with theory, as the HOG method is more complex in calculation, with larger step sizes and fewer particles leading to a decrease in computational complexity, resulting in a decrease in runtime.

Table. 1: Time cost

Dataset	David2								
Conditions	Step=1			Step=3			Step=5		
	n=400	n=200	n=50	n=400	n=200	n=50	n=400	n=200	n=50
Intensity/s	9.871	9.321	8.539	3.615	3.394	3.309	2.365	2.027	1.978
HOG/s	16.256	12.786	9.686	5.710	4.515	3.449	4.298	2.870	2.260

3 Optimize

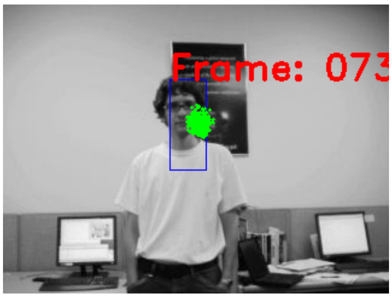
Based on the above program, we can optimize the Weighting, Decision, and Re-sampling sections. The specific optimization measures include the following parts:

1. In the weight calculation section, the features from the previous image are included as a reference, and the similarity with the current feature is calculated. The two weights obtained are summed with weights of 0.75 and 0.25, and the obtained results are used as the actual weights;

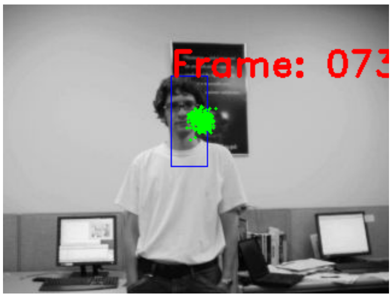
2. In the Decision stage, the reference particle number k can be adjusted, and the parameters of the first k particles with the highest weight value can be weighted and summed to obtain the "Decision particle" parameters. The "Decision particle" can be obtained through Gaussian sampling;

3. In the Re-sampling stage, the parameters of the first k particles with the highest weight value are also weighted and summed to obtain the resampling parameters;

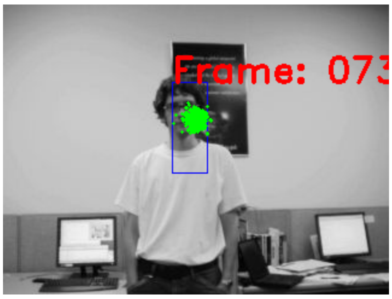
Set $step = 8$, $n\_particle = 400$, Intensity feature, the results of setting different particle numbers are as follow:



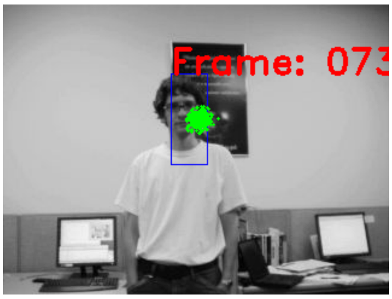
(a) k=1



(b) k=2



(c) k=5

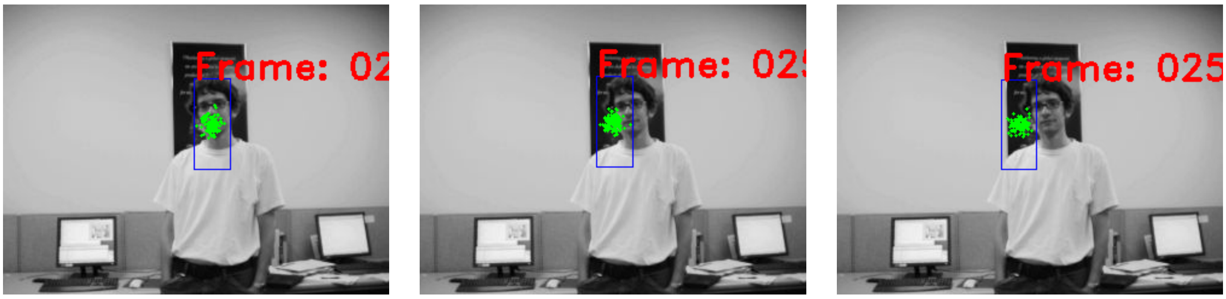


(d) k=10

Fig. 6: set $step = 8$, $n\_particle = 400$, Intensity feature

It can be found that increasing the number of reference particles can effectively enhance the performance of the algorithm in larger step sizes, but at the same time, when the number of added particles is too large, the performance will actually decrease. This is consistent with theory, as when the number of reference particles is too large, it can lead to particle interference with lower weight values, which in turn affects the accuracy of tracking.

Next, consider reducing the number of particles and observe the accuracy of the algorithm under different reference particle numbers. Set $step = 1$, $n_particle = 100$, Intensity feature, different number of reference particles performence are as follow:



(a) k=1 (b) k=2 (c) k=5

Fig. 7: set $step = 1$, $n\_particle = 100$, Intensity feature

It can be observed that adding reference particles during the Decision and Re-sampling stages does not increase the accuracy of the algorithm when the number of particles decreases. This is because in the case of a small number of particles, increasing the number of reference particles does not provide good information, but rather increases the error.

4 Conclusion

In this project, we implemented a object tracking algorithm based particle filter algorithm. Based on this algorithm, we discussed the impact of different step sizes and particle numbers on the accuracy of the algorithm. In practical problems, step size can represent object motion speed and video frame rate, and an increase in step size can lead to a decrease in the accuracy of particle filter algorithms.

We optimized the algorithm during the weight calculation, determination, and resampling stages, and ultimately achieved the similarity of referencing the previous image during the weight stage, while arbitrarily selecting the number of reference particles during the determination and resampling stages. After research, it was found that selecting an appropriate number of reference particles can effectively enhance the performance of the algorithm in larger step sizes, and increasing the number of reference particles means increasing the number of sampling particles during the sampling stage, otherwise the accuracy of the algorithm cannot be improved.