Artificial Intelligence: Project 2

—— Particle Filter Objects Tracker

3190102060 黄嘉欣

1. Basic Implementation

As has been introduced in the provided description file, I used two kinds of features, intensity and HOG, to mark the target object in the pictures. In the meantime, Gauss Distribution was applied in the transition step and resample step, and when computing the weight of each particle, cosine similarity was used. Lastly, in the decision stage, I simply considered the particle with the highest weight as the tracking result of current frame, which will be further promoted later. A comparison of the results is as follows, and the result images can be found in *result/basic_imp* folder.

a) Accuracy

According to the test results, in the easiest case, where $step = 1$ and $n\_particles = 400$, both intensity and HOG features can track the target object accurately, which can be seen as follows:

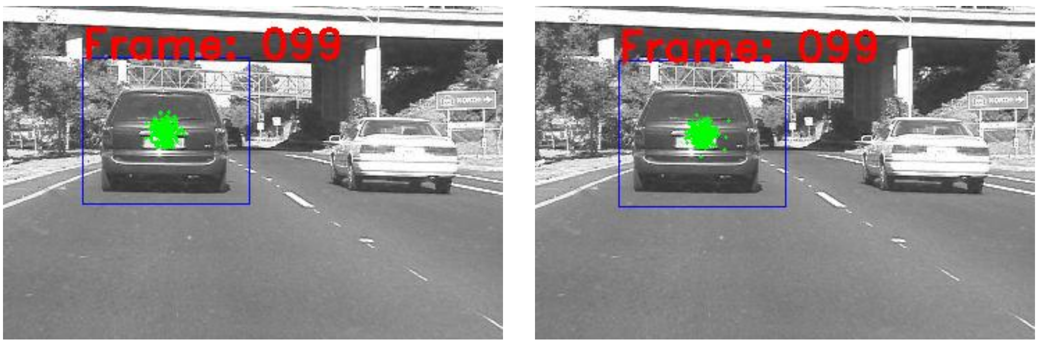

Figure 1.1 car with $step = 1$ and $n\_particles = 400$. The left one uses intensity features and the right one uses HOG features.

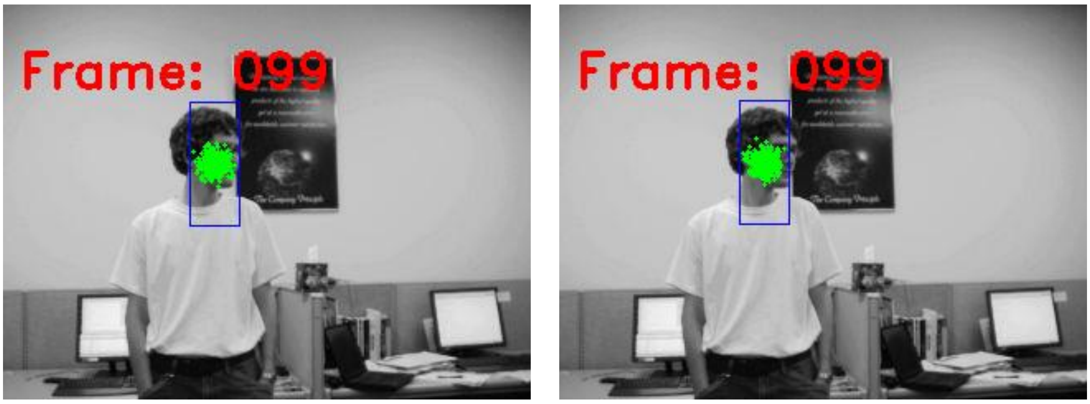

Figure 1.2 David2 with $step = 1$ and $n\_particles = 400$. The left one uses intensity features and the right one uses HOG features.

Similarly, when increase $step$ to 2, these feature types can work right too, indicated by the images in *step_2* folder in *results*. However, if we increase $step$ more, the algorithm may not track the object accurately any longer. For example, let's set $step = 8$, while keeping $n\_particles = 400$, and a randomly selected result of both feature types is shown in Figure 1.3:

Figure 1.3 David2 with $step = 8$ and $n\_particles = 400$. The left one uses intensity features and the right one uses HOG features.

Figure 1.4 David2 with $step = 1$ and $n\_particles = 400$. The left one uses intensity features and the right one uses HOG features. **Act as ground truth**.

Apparently, comparing Figure 1.3 and Figure 1.4, we can find that positions of the group of particles seem different, and the rectangle indicating the target object is different too, which is more obvious for the HOG features.

On the other hand, I kept $step = 1$ and decreased $n\_particles$ to 50 for the features, and the results are like follows:

Figure 1.5 David2 with $step = 1$ and $n\_particles = 50$. The left one uses intensity features and the right one uses HOG features.

Figure 1.6 David2 with $step = 1$ and $n\_particles = 400$. The left one uses intensity features and the right one uses HOG features. **Act as ground truth.**

In this case, the objects tracked by both feature types are incorrect apparently. From the results and comparison shown above, we may safely draw the conclusion that both intensity and HOG features can be greatly affected by the values of $step$ and $n\_particles$. If we skip too many frames or use too few particles to track the target object, the algorithm will very likely lead to inaccurate results.

b) Time cost

I have measured the time cost by these two kinds of features under different conditions, which is shown in the following table:

Table 1.1 Time cost

datasets	car		David2			
conditions	$step = 1$	$step = 2$	$step = 1$	$step = 2$	$step = 8$	$n\_particles = 50$
intensity/s	10.883315	5.748164	10.407447	5.721155	1.809734	6.998779
HOG/s	24.834727	12.015673	24.014023	11.965843	3.486705	8.534889

PS: The default condition is that $step = 1$ and $n\_particles = 400$, and it is omitted if the condition does not change in the table.

As we can see, the time we used to extract HOG features is much more than the intensity type needs, and the more frames we skip or the less particles we use, the less time we will spend finishing tracking the target object in the 100 images, which is the same as we have expected.

2. Further Implementation

Based on the work above, I used more particles to further smoothly filter the target position in both decision and resampling stage and also used features from 2 frames to compute the tracking template. In this case, I remeasured the accuracy and time cost of the both kinds of features as follows, and the images can be found in *result/further_imp* folder.

a) Accuracy

In part 1, we have seen that both feature types works well when $step = 1$. Hence, we will check the results outputted when $step = 2$, which are as follows:

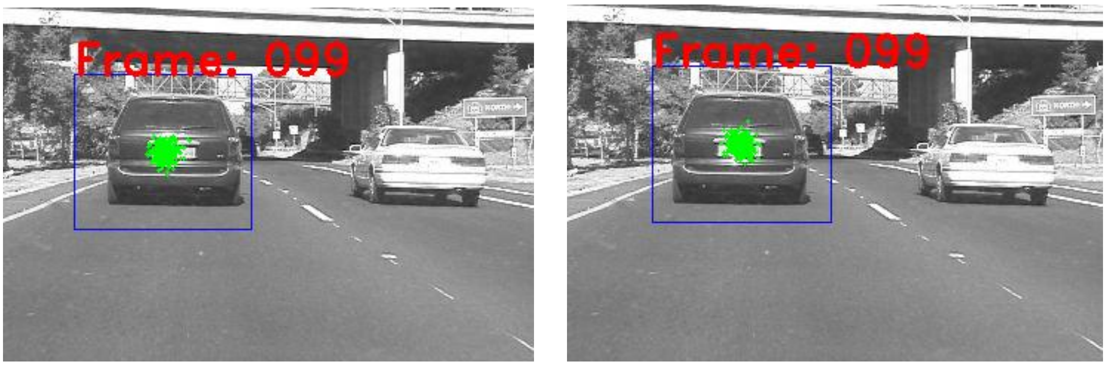

Figure 2.1 car with $step = 2$ and $n\_particles = 400$. The left one uses intensity features and the right one uses HOG features.

Figure 2.2 David2 with $step = 2$ and $n\_particles = 400$. The left one uses intensity features and the right one uses HOG features.

Although the rectangle marked in the pictures are a little different between the two feature types, the target objects — car and the person's face, have been both well found by the algorithm. Similarly, if we increase step to 8, the sampled pictures will be like:

Figure 2.3 David2 with $step = 8$ and $n\_particles = 400$. The left one uses intensity features and the right one uses HOG features.

Comparing Figure 1.3 and Figure 2.3, we can find that the quality of tracking results has improved to some extent after further implementation, especially for HOG features. Though the position of the group of particles seems weird, the blue rectangles marked the target object correctly.

However, when we sampled the images under the condition that $step = 1$ and $n\_particles = 50$, we may discover the results seem worse, especially for intensity features, as shown in Figure 1.5 and Figure 2.4. This is because we have taken the average position of the top-k weighting particles as the target position. In this case, if the top-k particles' positions vary greatly, the result will be significantly affected. On the other hand, we have also taken the template in last frame into consideration, which will lead to a slower movement of the rectangle and the errors will accumulate gradually as well.

Figure 2.4 David2 with $step = 1$ and $n\_particles = 50$. The left one uses intensity features and the right one uses HOG features.

To sum up, taking more particles and frames in to consideration can assist us to promote the quality of the tracking results to a certain degree but will also accumulate errors under some extreme circumstances. Despite the slight lack of robustness, the algorithm is able to handle enough problems, as we have tested.

b) Time cost

The time used in each case is as follows:

Table 2.1 Time cost

datasets	car		David2			
conditions	$step = 1$	$step = 2$	$step = 1$	$step = 2$	$step = 8$	$n\_particles = 50$
intensity/s	9.928545	5.298317	9.756807	5.331418	1.624814	6.784010
HOG/s	18.124487	10.847935	19.983179	10.262045	2.788790	8.015504

PS: The default condition is that $step = 1$ and $n\_particles = 400$, and it is omitted if the condition does not change in the table.

As before, when we skip more frames or use less particles to track the object, we will need less time to finish processing, which meets our expectations. On average, using HOG features may cost more time than intensity type, but it is more stable and accurate, according to the images we have shown in part 2.

3. Conclusion

In this project, we tried using the Particle Filtering Algorithm to track target objects in a number of images, and our implementations have fulfilled the expected requirements. It is reasonable that the accuracy will decrease as the value of $step$ increases or $n\_particles$ decreases. However, according to the sampled images and analysis above, the further implementation can help improve the quality of the results by taking more particles and frames into consideration, which has also deepened our understanding of the principle and process of the algorithm to some extent.