

浙江大学

**Artificial Intelligence**  
**ParticleFilter For ObjTracking**  
**Project Report**



2020~2021 春夏学期    2021    年   5   月   3   日

## Categories

CHAPTER 1: INTRODUCTION .....	3
CHAPTER 2: ALGORITHM SPECIFICATION.....	3
CHAPTER 3: TESTING RESULTS .....	4
CHAPTER 4: ANALYSIS AND COMMENTS .....	4
PROBLEMS: .....	5

# Chapter 1: Introduction

Background and Our goals:

This program implements a particle-filter algorithm to track designed target.

# Chapter 2: Algorithm Specification

Basic algorithm: particle-filter,

## 1. Transition (time update)

Each particle is moved by sampling its next position from the transition model

$$x_{t+1} = \text{sample}(P(x_{t+1}|x_t)) \quad \text{forward sampling}$$

## 2. Weighting (Measurement update)

Fix sample observation

Down weight samples based on the evidence (likelihood weighting)

$$w(x_{t+1}) = P(e_{t+1}|x_{t+1})$$

I used intensity, histogram, correlation coefficient method, Image similarity, because we only need to calculate the similarity, not necessarily extract features, the image similarity calculation is mainly used to score the similarity of the content between two images, and judge the similarity of the image content according to the score. In addition, the pixels of a grayscale image can be regarded as a two-dimensional matrix, and two two-dimensional matrices of equal size can be calculated for their correlation coefficients. The formula is as follows:

```
a = a - mean2(a);  
b = b - mean2(b);  
r = sum(sum(a.*b))/sqrt(sum(sum(a.*a))*sum(sum(b.*b)));
```

## 3. resampling

Rather than tracking weighted samples, we resample from our weighted sample distribution. We select maximum weighted particles, then perform 400 Gaussian samplings on it, replacing all particles.

## Pseudo codes:

Function particle-filtering (e, N, dbn) returns a set of samples for the next time step

Inputs: e, the new incoming evidence

N, the number of samples to be maintained

Dbn : a DBN with prior  $P(X_0)$ , transition model  $P(X_1|X_0)$ , sensor model  $P(E_1|X_1)$

Persistent (static variable): S, a vector of samples of size N, initially generated from  $P(X_0)$

Local variables: W, a vector of weights of size N

For I=1 to N do

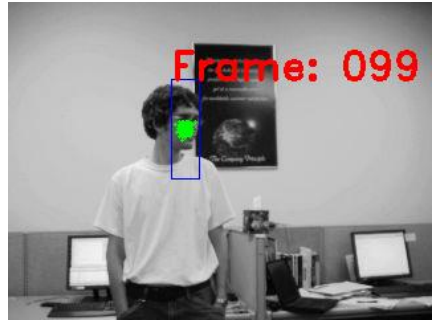
$S[i] \leftarrow$  sample from  $P(X_1|X_0 = S[i])$  //step1

$W[i] \leftarrow P(e|X_1 = S[i])$  //step 2

$S \leftarrow$  WEIGHTED-SAMPLE-WITH-REPLACEMENT(N,S,W) // step3

Return S.

## Chapter 3: Testing Results

图像	特征类型	粒子个数	结果
Car	correlation	100	可以跟踪
Car	Histogram	100	可以跟踪,虽然粒子分布的比较散
David	correlation	400	可以跟踪
			

## Chapter 4: Analysis and Comments

At first, the effect was poor and AI often lost target. Later, the car target was fine, but David is more difficult. We select maximum weighted particles, then perform 400 Gaussian samplings on it, replacing all particles. It works.

Expected improvements in the future:

1. Sometimes moving the initial position of init\_rect to the left or increasing the size a bit can have a good

effect.

2. extract\_feature feature extraction can use the histogram method to calculate the Bhattacharyya distance, it can be sift, orb or suft, orb is fast and has low accuracy, and sift has high accuracy and large calculation volume.

## Problems:

问题 1 安装出错 EnvironmentNotWritableError: The current user does not have write permissions to the target environment.

解决方法: 用管理员身份启动 anacoda prompt

问题 2 什么是匹配模板?

就是其他粒子和这个模板计算相似度.

问题 3 怎么实现极大似然估计实现权重?

相似度计算, 方法一, 欧氏距离是最常用的距离计算公式, 衡量的是多维空间中各个点之间的绝对距离, 当数据很稠密并且连续时, 这是一种很好的计算方式。

方法二: 明可夫斯基距离 (Minkowski distance)

明氏距离是欧氏距离的推广, 是对多个距离度量公式的概括性的表述.

方法三: 余弦相似度

给定两个属性向量,  $A$  和  $B$ , 其余弦相似性  $\theta$  由点积和向量长度给出,

余弦距离更多的是从方向上区分差异, 而对绝对的数值不敏感, 更多的用于使用用户对内容评分来区分兴趣的相似度和差异, 同时修正了用户间可能存在的度量标准不统一的问题 (因为余弦距离对绝对数值不敏感)。我们这里应该不能用余弦, 因为 244,255 明显和 0,11 相差很大. 我决定用欧式距离先试试.

问题 4 怎么根据每个粒子的权重, 对其重新采样, 增加高权重的粒子, 减少或剔除低权重粒子?

思考: 减少的话就排序然后最低权重删除, 那么怎么增加呢?

就复制一个最高权重的粒子. 这样看 如果一个粒子的权值很大 当权重求和加到  $q(i)$  这个粒子的时候 将在很长时间内大于 0-1 随机数  $\text{rand}(0, 1)$ , 当粒子数目较大的时候 大权值的粒子就被复制很多次了. 这个方法缺点是有错误的粒子还是会进来, 然后很相似的地方比如人脸和黑板上的白色容易出错.

问题 5 怎么保证权重之和为 1?

解决方法:

算出相似度然后加起来归一化.

问题 6 End of statement expected

解决方法: `print(x)` 代替 `print x`

问题 7: ValueError: The truth value of an array with more than one element is ambiguous. Use a.any() or a.all()

原因:

```
threshold = random.random_sample()
```

```
threshold = random.uniform(0, 1, 4)
```

```
if total >= threshold[0]:
```

这样也不行, 为啥呢?它居然不是返回一个值而是返回一个 union

Union 是当有多种可能的数据类型时使用, 比如函数有可能根据不同情况有时返回 str 或返回 list, 那么就可以写成 Union[list, str]

我加上 all 它又显示 float 不能变成 all

我把它声明为 `float`,但是它还是会出错. 已经两边都是 `float` 了.

用断点看看每个值就可以了,

因为 `j.weight` 是 `array` 然后就变成 `array`

问题 8: `pycharm` 无法最大化, `pycharm` 最小化打不开。

解决方法: 重装 `pycharm` 也不行, 换个项目就可以了。

问题 9: `cv2.error:OpenCV(4.5.1)error: (-215:Assertion failed) !ssize.empty() in function 'cv::resize'`

原因: 没有合法的图像传入,有一个粒子跑到图像边界了。

解决方法: 如果为空那就把粒子放到最左上角去。

问题 10:`res = abs(np.sqrt(sum(np.power((feature - template), 2))))`

这个出错了, 我想让他变成欧氏距离,但是它 `res` 还是一个数组.为啥我单独一个文件就不会出错?

因为单独文件是 `array`, 而工程中是 `ndarray`

解决方法: 我试试用 `np.sum` 而不是 `sum`

好像可以了

```
sum1 = np.sum(np.power((feature - template), 2))
```

```
res = abs(np.sqrt(sum1)) # nparray 的话这么写
```

问题 11: `ValueError: mean must be 1 dimensional`

原因: 在运行时出错. Python 总是会运行时变成多个维度, 加起来很容易就变成多维.

`transition_step`

第二次时就是 `array` 了, 应该是因为重采样时变成了 `array`

好像也不是, 重采样之前就是 `array` 了.

```
mean = np.array([self.cx, self.cy]) # 均值
```

```
conv = np.array([[sigmas[0], 0.0], [0.0, sigmas[1]]]) # 协方差矩阵
```

```
x, y = np.random.multivariate_normal(mean=mean, cov=conv, size=1).T
```

```
self.cx = x
```

```
self.cy = y
```

这里把 `cx` 变成数组了.

问题 12: 80 多帧锁定黑板上的白色, 怎么办?

原因: `resample` 如果用轮盘法, 还是会出现一些完全错误的粒子, 然后慢慢会变多. `Resample` 可以用权重最大的粒子

解决方法: 多帧联合. 队列, 先入队, 如果 5 帧了就 `pop` 一个然后入队, 5 帧联合,

问题 13: 特征向量的关系是啥?怎么可以多个特征向量联合?

使用权重最大的粒子作为当前帧的跟踪结果太容易偏移了, 应该多个粒子作为模板.怎么让多个粒子变成一个结果?

可以加权求坐标