浅议 Sort 算法（目标体识别）

作者：vcjmhg
原文链接：https://ld246.com/article/1594191039184
来源网站：链滴
许可协议：署名-相同方式共享 4.0 国际 (CC BY-SA 4.0)

概述

多目标追踪（Multiple Object Tracking），简单来说其主要任务就是给定一个图像序列，在识别出像中的物体后，通过一个Trace id将不同帧中的同一个物体进行表示，从而完成目标追踪的任务。当这些物体可以是任意的，例如行人车辆各种动物等。

本文讲述的是sort算法全称为Simple Online and Realtime Tracking。从名字上看我们就可以看去其它是一个比较简单的目标体识别算法，其本身并不涉及复杂的网络计算， 可以说是比较纯粹的算法。

本文的书写并不会完全论文结构来进行全文翻译，感觉这样子没有任何意义，因此本文更多的是去讲人在学习sort算法过程中的理解和遇到的相关问题。

论文地址： sort

代码地址： code

算法

sort算法的提出，大程度上是为了证明作者提出的是个观点---**追踪的质量很大程度上受到检测质量影响。**为了验证这个观点，作者设计了SORT算法。

因此，Sort算法在设计之初就十分重视检测环节，它以检测作为关键组件，传播目标状态到未来帧中将当前检测与现有目标相关联，并管理跟踪目标的生命周期。其中状态传播是通过FRrCNN网络来实，目标关联通过匈牙利算法来实现。

首先在检测环节，作者选择用FRRCNN网络作为目标检测网络，参数使用的是PASCAL VOC挑战中的认参数，对于检测的类别作者也给了一个约定，即只检测识别概率大于50%的行人检测类别，而忽略他类别。

到这里可能会有两个问题：

第一个问题：为何要选择FRRCNN网络作为目标检测网络?

选择FRRCNN网络的原因很大程度上是有该网络的特点决定，FRRCNN网络有三个特点：

1. 快，FFRRCNN网络，是一个two-stage算法，两个阶段共享参数，使得检测速度大大提升，这某程度上保证了算法的实时性。

2. 网络结构灵活，框架结构容易替换，这样作者在进行验证的时候便于将网络的框架结构进行替换，证在不同网络下架构下，检测结果。

为何只检测识别概率大于50%的目标?

一方面检测目标类别的减少可以更快的出实验效果，另一方面，cnn网络比较灵活，在实际工程应用我们可以很方便的将行人这一类别替换成其他类别。

并且作者在该部分做了一个实验验证，下图是其实验的结果图，从实验结果中我们可以看到，作者替了MDP和其所提供算法的检测，发现跟踪效果有显著的改善，这在某种程度上也验证则作者在论文开的时候提出的观点即：**追踪的质量很大程度上受到检测质量的影响。**

Tracker	Detector	Detection		Tracking	
		Recall	Precision	ID Sw	MOTA
MDP [12]	ACF	36.6	75.8	222	24.0
	FrRCNN(ZF)	46.2	67.2	245	22.6
	FrRCNN(VGG16)	**50.1**	76.0	**178**	33.5
Proposed	ACF	33.6	65.7	224	15.1
	FrRCNN(ZF)	41.3	72.4	347	24.0
	FrRCNN(VGG16)	49.5	**77.5**	274	**34.0**

估计模型部分，选择的是线性常速模型。

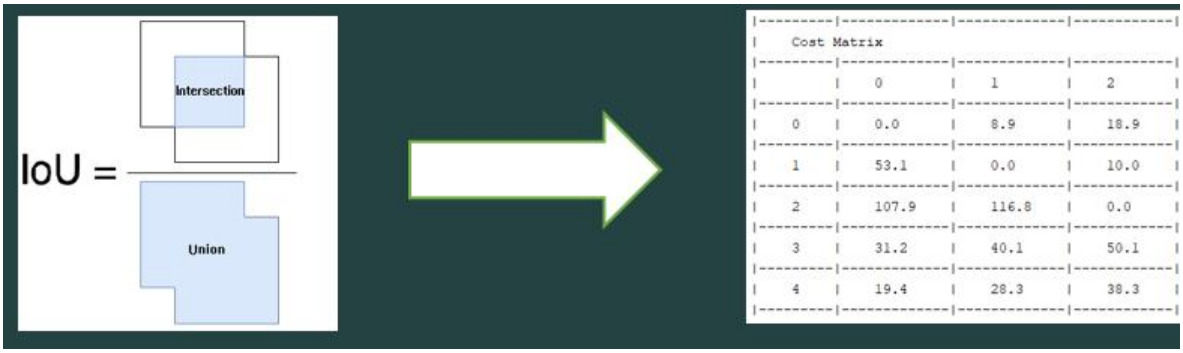
$$\mathbf{x} = [u, v, s, r, \dot{u}, \dot{v}, \dot{s}]^T$$

该模型有七个参量组成，u表示目标中心点的水平坐标，v表示目标中心的的垂直坐标，s标尺检测框尺寸，r表示检测框的纵横比。u ’ v’ s’ 分别表示下一帧检测目标中心的水平坐标垂直坐标和检测的尺寸。我们注意到该模型中没有r’ ，这是因为在目标追踪中，我们认为物体的纵横比是不变变量。

数据关联则是由四个阶段组成：

1. 预测每个目标在当前帧的位置，估计其边界形状，预测的功能通过卡尔曼滤波来实现。

2. 根据目标检测框和预测边界框的交并比计算分配矩阵。



3. 有了代价矩阵之后我们便可以通过匈牙利算法来求出一个最大匹配，从而得出相邻两帧物体的轨迹。

4. 最后它会拒绝检测目标重叠小于 IoUmin的分配。

当目标进入和离开图像时，需要相应地创建或销毁唯一标识。对于创建跟踪程序，文中认为任何重叠于IoU_min的检测都表示存在未跟踪的目标。使用速度设置为零的边界框信息初始化跟踪器。由于此无法观测到速度，因此速度分量的协方差用较大的值初始化，反映出这种不确定性。此外，新的跟踪将经历一个试用期，其中目标需要与检测相关联以积累足够的证据以防止误报的跟踪。删除时，将Tlos设置为1，如果Tlost未被检测到则删除此物体跟踪。

此处可能会有个问题--为何作者要将Tlost设置为1?

分析其原因有三部分组成：

首先，sort算法使用的是等速模型，等速模型对真实动力学的预测能力较差，即使Tlost设置较大的值踪性能提升也不明显；

其次，sort算法主要关注逐帧跟踪，目标重识别超出本工作范畴；

此外，早期删除丢失的目标有助于提高效率。如果目标重新出现，跟踪将在新标识下隐式恢复。

实验

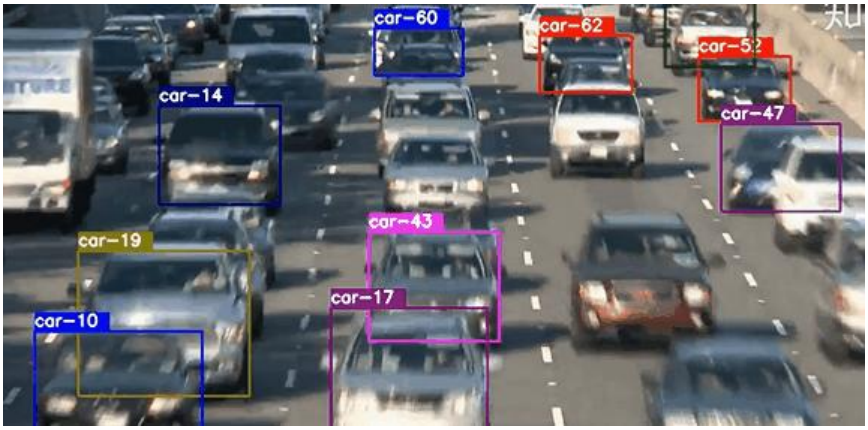
作者为了验证自己的观点，设计了一系列实验，对比sort算法和当时几种有效的多目标追踪效果，得了如下的结果图。从图中我们可以看到，SORT算法在MOTA、FAG、ML以及FP这些指标都取得了不的表现，但它也有不足之处，它ID sw指标标记大，表明在实际检测中追踪id更换频繁。

Table 2. Performance of the proposed approach on MOT benchmark sequences [6].

Method	Type	MOTA↑	MOTP↑	FAF↓	MT↑	ML↓	FP↓	FN↓	ID sw↓	Frag↓
TBD [20]	Batch	15.9	70.9	2.6%	6.4%	47.9%	14943	34777	1939	1963
ALExTRAC [5]	Batch	17.0	71.2	1.6%	3.9%	52.4%	9233	39933	1859	1872
DP_NMS [23]	Batch	14.5	70.8	2.3%	6.0%	40.8%	13171	34814	4537	3090
SMOT [1]	Batch	18.2	71.2	1.5%	2.8%	54.8%	8780	40310	1148	2132
NOMT [11]	Batch	**33.7**	71.9	**1.3%**	12.2%	44.0%	7762	32547	**442**	**823**
RMOT [4]	**Online**	18.6	69.6	2.2%	5.3%	53.3%	12473	36835	684	1282
TC_ODAL [17]	**Online**	15.1	70.5	2.2%	3.2%	55.8%	12970	38538	637	1716
TDAM [18]	**Online**	33.0	**72.8**	1.7%	**13.3%**	39.1%	10064	**30617**	464	1506
MDP [12]	**Online**	30.3	71.3	1.7%	13.0%	38.4%	9717	32422	680	1500
SORT (Proposed)	**Online**	**33.4**	72.1	**1.3%**	11.7%	**30.9%**	**7318**	32615	1001	1764

结论

从结果图上看，总体来说，sort算法当时兼顾了追踪效果和速度取得了不错的结果，作者认为实用性。



原文链接：浅议 Sort 算法（目标体识别）

但实际上我们可以看出sort算法的不足之处，首先我们可以先看一个sort检测结果gif图，gif图中的检类别设置成了汽车。从gif图中我可以看到，追踪汽车的id在频繁的更换。这种情况下实际上已经失去追踪的价值。

为了解决该问题，有人就提出了deepsort算法，deesort算法是如何解决sort算法的不足的？以及其理是怎样的？后续我会在写篇文章详细进行展开。