
第 1 章 问题描述

1.1 多目标检测与追踪问题描述

给定图像序列 I_1, I_2, \dots, I_t ，每帧图像中有 M_t 个目标，其中 t 是当前帧号，每个目标的状态为 $s_1(t)$ ，其中状态一般包括位置，速度，加速度，朝向等。

$$s_i(t) = \{x, y, z, h, w, l, v_x, v_y, v_z, \theta\} \quad (1-1)$$

当前帧的所有目标的状态就能表示成

$$S(t) = \{s_1(t), s_2(t), s_3(t), \dots, s_{M_t}(t)\} \quad (1-2)$$

而每个目标的轨迹则可以描述成

$$s_i(1:t) = \{s_i(1), s_i(2), s_i(3), \dots, s_i(t)\} \quad (1-3)$$

则所有目标的状态集合就能表示成

$$S(1:t) = \{S_1, S_2, \dots, S_t\} \quad (1-4)$$

同理，我们类似的得到观测结果的定义，记作 $o_i(t), o_i(1:t), O(1:t)$ 。

而多目标跟踪任务就是通过观测结果找出所有目标的状态，我们用后验估计来进行描述。

$$S(1:t) = \operatorname{argmax}_{S(1:t)} P(S(1:t)|O(1:t)) \quad (1-5)$$

1.2 问题的求解

多目标检测与追踪一般的求解都基于 TBD 架构，如图1-1所示。

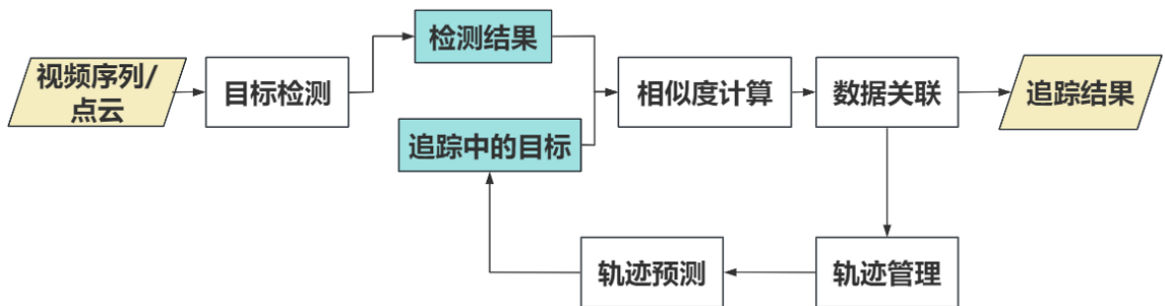


图 1-1 基于检测的追踪框架

1.3 滤波的作用

主要概括，滤波的作用主要包括两个部分，融合多个传感器的追踪结果以及对目标进行预测，最终的目的都是为了得到更好的估计值。它主要应用在图1-1中轨迹预测部分。

首先构建线性系统的状态空间描述：

$$\mathbf{x}_t = \mathbf{A}\mathbf{x}_{t-1} + \mathbf{B}\mathbf{u}_t + \boldsymbol{\epsilon}_t$$

$$\mathbf{z}_t = \mathbf{H}\mathbf{x}_t + \boldsymbol{\delta}_t$$

接着利用卡尔曼滤波器进行最优状态估计：

$$\hat{\mathbf{x}}_t^- = \mathbf{A}\hat{\mathbf{x}}_t + \mathbf{B}\mathbf{u}_t \quad (1-6)$$

$$\boldsymbol{\Sigma}_t^- = \mathbf{A}\mathbf{P}_{t-1}\mathbf{A}^T + \mathbf{Q} \quad (1-7)$$

$$\mathbf{K}_t = \frac{\boldsymbol{\Sigma}_t^- \mathbf{H}^T}{\mathbf{H}\mathbf{P}_t^- \mathbf{H}^T + \mathbf{R}} \quad (1-8)$$

$$\hat{\mathbf{x}}_t = \hat{\mathbf{x}}_{t-1} + \mathbf{K}_t(\mathbf{z}_t - \mathbf{H}\hat{\mathbf{x}}^-) \quad (1-9)$$

$$\mathbf{P}_t = (\mathbf{I} - \mathbf{K}_t\mathbf{H})\mathbf{P}_t^- \quad (1-10)$$

1.4 发展和思考

1. 发展

想要提高追踪的效果（精度，速度），可以从多个角度进行提高。大致可以包括几个方面：

仍旧基于 BDT 框架：追踪器的提升，数据融合方法的提升，数据关联的提升，滤波器的提升（包括模型改进）。

新的框架：端到端^[1]，基于点的移动的追踪^[2]。

2. 思考

首先确定融合的框架：DBT 和神经网络融合结构。接着确定数据关联方式，倾向于用特征值（点）然后改进滤波器的结构：模型的改进（长时间拟合），噪声的优化（A-KIT）最后，连接最新的检测器。

实际实验：标定，ROS 表示

第 2 章 DFR-FastMOT: Detection Failure Resistant Tracker for Fast Multi-Object Tracking Based on Sensor Fusion(ICRA2023)^[3]

2.1 解决问题

文章主要针对目标追踪中的遮挡问题做了许多工作。非学习算法往往只保存一小段的轨迹，因此难以应对长时间的遮挡情况。文章设计了一种代数的数据关联公式从而降低了需要计算复杂度，从而可以处理更长时间的轨迹。

2.2 解决算法

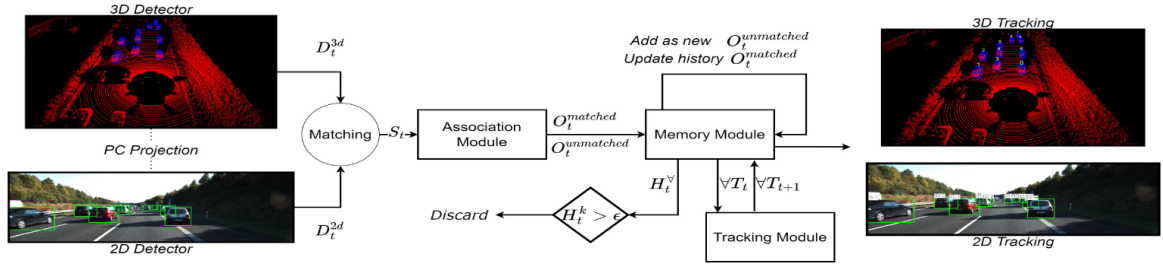


图 2-1 整体框架

2.2.1 关联矩阵的引入

为了提高计算的效率，文章为每种传感器设计了一个关联矩阵，矩阵的每个元素代表前一时刻对象（轨迹）的估计值和当前时刻检测值的关联程度。

$$M_{m \times n} = \underbrace{\begin{pmatrix} v_{0,0} & v_{0,1} & v_{0,2} & \cdots \\ v_{1,0} & v_{1,1} & v_{1,2} & \cdots \\ v_{2,0} & v_{2,1} & v_{2,2} & \cdots \\ \vdots & & & \ddots \\ v_{m,0} & v_{m,1} & v_{m,2} & \cdots \end{pmatrix}}_{n \text{ prior observed objects}} \left. \vphantom{\begin{pmatrix} v_{0,0} \\ v_{1,0} \\ v_{2,0} \\ \vdots \\ v_{m,0} \end{pmatrix}} \right\} m \text{ observed objects}$$

$v_{i,j}$ 代表关联值，公式2-1处理 2D 情况，记作 M_c 。公式2-2处理 3D 情况，记作 M_l 。公式2-3将两种情况进行统一，记作 M_f 。

$$v_{ij} = \begin{cases} v_{IoU} & : v_{IoU} \leq a_c \\ 0 & : \text{Otherwise.} \end{cases} \quad (2-1)$$

$$v_{ij} = \begin{cases} v_{dist} & : v_{dist} < a_l \\ a_l & : \text{Otherwise.} \end{cases} \quad (2-2)$$

$$\begin{cases} M_f & = \alpha_c M_c + \alpha_l (1 - M_t), \\ \alpha_c + \alpha_l & = 1, \\ \alpha_c, \alpha_l & \leq 1. \end{cases} \quad (2-3)$$

得到所有关联矩阵之后，便可以进行数据关联。文章采用了类匈牙利算法来实现该步骤。最后所有的计算复杂度为：

$$\mathcal{O}(2mn) \rightarrow \mathcal{O}(m^2n^2) \rightarrow \mathcal{O}(m^2n^2)$$

2.2.2 其它处理方法

1. 3D 距离函数的选用

为了处理遮挡的情况，文章采用了 3D 中心距离来衡量两个目标的相似程度，而不是传统的 IoU。当出现长时间的遮挡情况时，去计算 IoU 是十分困难的，而计算 3D 的距离则更容易实现。

2. KF 计算简化

在用 KF 计算目标下一时刻的状态时，需要进行大量的矩阵运算。为此，作者采用最少点数来描述目标位置：2D 两个，3D 两个。

2.3 文章结果

文章通过使用不同质量的检测器来模拟遮挡的效果。结果表明，文章显著提高了低质量检测器的追踪效果。整体而言，追踪的精度也有所提高。

Detector	Method	HOTA↑	MOTA↑	MOTP ↑	DetA↑	AssA↑	IDSW↓	IDF1↑	MT↑	ML↓	Frag↓
2D YOLOv3 [27]	EagerMOT [5]	36.5%	41.6%	76.4%	35.4%	38%	792	48.1%	105	124	855
+	DeepFusion-MOT [4]	30%	31.8%	77.4%	27.4%	33.1%	696	40.3%	28	241	819
3D PC Projection	DFR-FastMOT(Our)	39.2%	44.5%	76.3%	36.3%	42.8%	386	53.4%	113	106	907
2D RCC [28]	EagerMOT [5]	70.8%	82.2%	90.8%	79.5%	63.3%	1303	74%	413	26	213
+	DeepFusion-MOT [4]	42.6%	40.2%	90.7%	41.7%	43.7%	1203	44.9%	81	211	1063
3D PC Projection	DFR-FastMOT(Our)	81.9%	91%	90.7%	83.6%	80.3%	215	90.1%	485	12	239
2D RCC [28]	EagerMOT [5]	69.1%	67.2%	85.2%	62.6%	76.5%	112	80.9%	499	10	316
+	DeepFusion-MOT* [4]	77.5%	87.3%	86.6%	75.4%	79.9%	83	91.9%	450	14	301
3D PorintRCNN [29]	DFR-FastMOT(Our)	82.8%	90.7%	90.6%	83.1%	82.6%	177	91.4%	503	8	238
2D RCC [28]	EagerMOT [5]	78%	87.3%	87.6%	76.8%	79.5%	91	89.3%	509	7	246
+	DeepFusion-MOT [4]	77.2%	85.8%	86.8%	74.9%	79.9%	136	90.9%	426	22	280
3D PointGNN [30]	DFR-FastMOT(Our)	82.2%	90.2%	90.5%	82.5%	82%	189	90.5%	516	6	224
2D TrackRCNN [31]	EagerMOT [5]	68.6%	63.2%	85%	60.8%	77.6%	102	80.4%	508	10	262
+	DeepFusion-MOT [4]	62.2%	57.7%	86.9%	51.5%	75.3%	65	73.2%	307	129	286
3D PorintRCNN [29]	DFR-FastMOT(Our)	70%	80.1%	82.6%	67.7%	72.7%	193	85.9%	425	18	608
2D TrackRCNN [31]	EagerMOT [5]	78.2%	86%	87.5%	75.8%	80.8%	83	90.2%	522	6	184
+	DeepFusion-MOT [4]	66%	65.9%	86.2%	58.6%	74.6%	133	79 %	351	86	386
3D PointGNN [30]	DFR-FastMOT(Our)	69.7%	80%	82.6%	67.7%	72.1%	203	85.2%	429	13	561

2.4 学习总结

DFRMOT 也是采用 DBT 追踪框架，具体内容也基本和 EagerMOT 相似。其主要工作在于提出了一种提高计算效率的关联矩阵，由此多出的冗余可以用来计算更长时间的目标，间接的提高了应对遮挡的能力。

所以，在自己设计的追踪器上，就可以采用本文设计的关联矩阵提高计算效率。此外，本外还在细节上提出了许多加速方法，我们可以直接应用他的计算框架来完成更复杂的任务。

参考文献

- [1] Xin S, Zhang Z, Wang M, et al. Multi-modal 3D Human Tracking for Robots in Complex Environment with Siamese Point-Video Transformer[C]//2024 IEEE International Conference on Robotics and Automation (ICRA). 2024: 337-344.
- [2] Wu H, Li Y, Xu W, et al. Moving event detection from LiDAR point streams[J]. nature communications, 2024, 15(1): 345.
- [3] Nagy M, Khonji M, Dias J, et al. DFR-FastMOT: Detection Failure Resistant Tracker for Fast Multi-Object Tracking Based on Sensor Fusion[C]//2023 IEEE International Conference on Robotics and Automation (ICRA). 2023: 827-833.