文章编号:1001-5078(2024)09-01477-08

·图像与信号处理·

基于红外图像处理的移动目标识别跟踪算法

耿宇飞,周　宽,刘纪洲,李　红
(中电科光电科技有限公司,北京 100015)

摘　要:为解决红外热像仪在战场环境多变、背景复杂、目标运动速度过快等情况下难以对目标进行识别与跟踪的问题,本文提出了一种基于红外图像处理的移动目标识别跟踪算法。该算法主要分为移动目标的检测和跟踪两部分。首先采用高斯背景建模方法来提取运动目标,其次应用 Kalman 滤波来进行运动目标的定位与跟踪,最后通过匈牙利算法来判定目标点的归属。实验结果表明,在 MOT 基准挑战中的评价指标上,本文算法相比于 Sort 跟踪器将 MOTA 提升了 1.4 %,MT 提升了 11.9 %,ML 降低了 6.5 %,IDs 降低了 33.3 %,FN 降低了 13.6 %,提高了跟踪准确性,成功实现了对多个移动目标的跟踪。本文算法在军事领域具有广泛的应用前景。

关键词:红外图像;目标检测;识别跟踪;Kalman 滤波;匈牙利算法
中图分类号:TP391.41;TN219　　**文献标识码:**A　　**DOI:**10.3969/j.issn.1001-5078.2024.09.020

Moving target recognition and tracking algorithm based on infrared image processing

GENG Yu-fei, ZHOU Kuan, LIU Ji-zhou, LI Hong
(CETC Electro-Optics Technology Co., Ltd., Beijing 100015, China)

Abstract: To address the problem of difficulty in recognizing and tracking targets using infrared thermal imaging cameras in a complex and rapidly changing battlefield environment, a moving target recognition and tracking algorithm based on infrared image processing is proposedin this paper. The algorithm mainly consists of two parts: detection and tracking of mobile targets. Firstly, the Gaussian background modeling method is used to extract moving targets. Secondly, Kalman filtering is applied to locate and track moving targets, and finally, the Hungarian algorithm is used to determine the attribution of target points. The experimental results show that compared with the SORT tracker, the proposed algorithm improves MOTA by 1.4 %, MT by 11.9 %, reduces ML by 6.5 %, IDs by 33.3 %, and FN by 13.6 %, which improves the tracking accuracy and successfully realizes the tracking of multiple moving targets. The algorithm in this paper has broad application prospects in military fields.

Keywords: infrared images; target detection; recognition tracking; Kalman filtering; Hungarian algorithm

1　引　言

在两栖突击作战行动中,为实现对目标的精准打击,装甲战车需配备红外热成像系统以满足战车在不同地域、不同天候条件下的观测能力,用于应对复杂多变的战场环境。近年来,随着对热像仪性能指标的要求越来越高,研究人员们正在探索更加高效、准确的移动目标识别跟踪方法,以满足实际应用的需求。在目标快速移动、遮挡、目标形变、各类背景噪声的影响下,如何能够快速、实时、准确地实现目标跟踪是当前研究的难点,因此对红外图像的移

收稿日期:2023-09-07;修订日期:2023-10-16

动目标识别跟踪技术的研究具有重要意义。

目前,国内外研究学者已开展了大量关于移动目标识别跟踪算法的研究,其中李向祥提出了一种统计背景建模框架来处理目标检测问题,利用全局和局部信息来实现对运动目标的识别跟踪[1];石秋根据粗糙模糊集理论,在传统帧差法的基础上,提出了粗糙熵模型,对运动目标区域进行定位[2];刘德连提出了一种新的非线性自适应滤波器,用于提取与运动目标相关的时间轮廓上的脉冲信号[3];Lien 将 CamShift 和卡尔曼滤波相结合,解决了复杂背景下运动目标跟踪问题[4];Aparna Akula 引入了卷积神经网络(CNN)深度学习框架,用于红外图像中目标的自动识别[5]。现有的目标识别跟踪算法均存在一定程度上的弊端,大多数算法只可实现单一运动目标的定位与跟踪,无法实现多目标的定位与跟踪。在战场环境下,不可避免的将会出现两个或多个移动目标遮挡重叠,此时,如何能够准确的实现不同移动目标的精确定位与跟踪就显得尤为重要。

为了提高算法设计与实现的成功率,本文通过借鉴前人的研究方法,提出了一种新型的识别跟踪算法。该算法结合了高斯背景建模、Kalman 滤波和匈牙利算法,旨在解决红外热像仪在背景复杂、目标运动速度过快等情况下难以对目标进行识别与跟踪的问题。具体来说,该算法首先采用高斯背景建模方法对图像进行处理,建立背景模型,然后通过 Kalman 滤波对目标轨迹进行预测。接着,使用交叠率来构建关联矩阵,并通过匈牙利算法进行数据关联,从而确定每个观测目标与其对应的预测目标的匹配关系。最后,根据匹配结果,更新目标轨迹,并对新出现的目标进行初始化,成功解决了多移动目标遮挡重叠问题,为红外热像仪在实时探测过程中能够快速地识别和跟踪多个移动目标,提高识别探测效率奠定基础。

2 目标识别与跟踪算法

本文提出的移动目标识别跟踪算法分为目标检测和跟踪两部分,具体算法逻辑流程图如图 1 所示。首先,将红外热像仪采集得到的每一帧图像信号转化为灰度图,并利用高斯背景建模方法获取背景。然后采用边缘检测算法得到运动目标的轮廓,将轮廓近似为具有较少顶点的另一个形状,并绘制出矩形框,从而得到运动目标的中心点坐标(x,y)。接着就是移动目标跟踪部分的算法,在确定好跟踪目标后为每个目标分配一个 ID,同时结合匈牙利算法来解决 Kalman 滤波算法跟随预测的分配问题,并确定每一个目标以检测 ID 是否分配错误。最后,使用 Kalman 滤波算法对目标轨迹进行预测,并通过中心点与 Kalman 预测点的距离计算出目标的速度值,从而实现目标实时轨迹、速度和目标数量的记录。

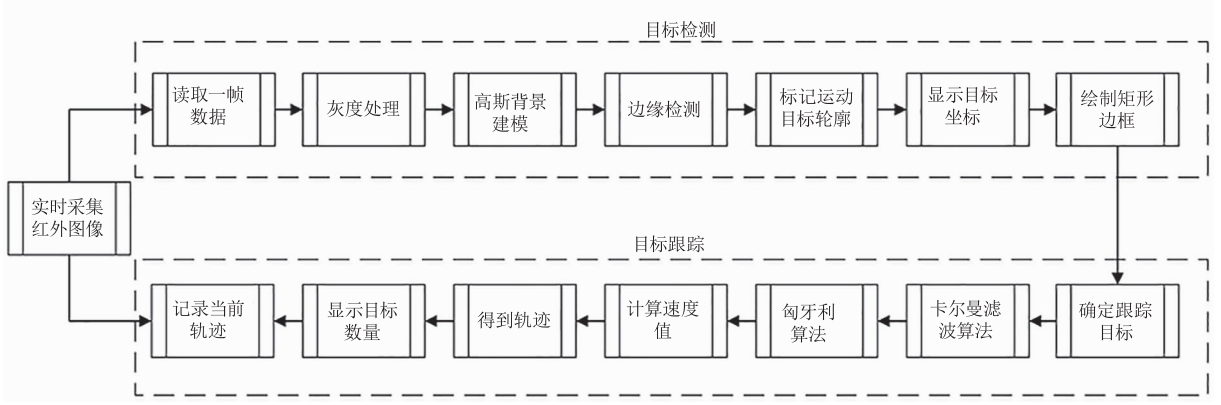


图 1 移动目标跟踪算法逻辑流程图

Fig. 1 Logic flowchart of mobile object tracking algorithm

2.1 高斯混合背景建模

高斯背景模型是一种运动目标检测过程中提取并更新背景和前景的一种方法[6]。单分布高斯背景模型认为,对一个背景图像,特定像素亮度的分布满足高斯分布,即对背景图像 B,每一个点 (x,y) 的亮度满足 $B(x,y) \sim N(u,d)$:

$$p(x) = \frac{1}{\sqrt{2\pi}d} e^{-\frac{(x-\mu)2}{2d^2}} \tag{1}$$

即每一个点 (x,y) 都包含了两个属性,均值 u 和方差 d,计算一段时间内的视频序列图像中每一个点的均值 u 和方差 d,作为背景模型。对于一幅包含前景的任意图像 G,对于图像上的每一个点(x,y)

计算，若：

$$\frac{1}{\sqrt{2\pi}d}e^{-\frac{(G(x,y)-B(x,y))2}{2d2}} > T \tag{2}$$

其中，T 为一个常数阈值，若上式成立则认为该点是背景点，否则为前景点[7]。接下来为背景的更新，每一帧图像都参与背景的更新：

$$B_t(x,y) = p^* B_{t-1}(x,y) + (1-p) * G_t(x,y) \tag{3}$$

该部分算法使用了 OpenCV 提供的函数 BackgroundSubtractorMOG2。该类公共成员函数如图 2 所示。

virtual double	getBackgroundRatio () const=0 Returns the "bckground ratio" parameter of the algorithm. More...
virtual double	getComplexityReduction Threshold () const=0 Returns the complexity reduction threshold.More...
virtual bool	getDetectShadows () const=0 Returns the shadow detection flag.More...
virtual int	getHistory () const=0 Returns the number of last frames that affect thebackground model. More...
virtual int	getNMixtures () const=0 Returns the number of gaussian components in the background model. More...
virtual double	getshadowThreshold () const=0 Returns the shadow threshold. More...
virtual int	getShadowValue () const=0 Returns the shadow value. More...
virtual double	getVarInit () const=0 Returns the initial variance of each gaussian component. More...
virtual double	getVarMax () const=0
virtual double	getVarMin () const=0
virtual double	getVarThreshold () const=0 Returns the variance threshold for the pixel-model match. More...
virtual double	getVarThresholdGen () const=0 Returns the variance threshold for the pixel-model match used for new mixture component generation.More...
virtual void	setBackgroundRatio (double ratio) =0 Sets the "background ratio" parameter of the algorithm. More...
virtual void	setComplexityReduction Threshold (double ct) =0 Sets the complexity reduction threshold. More...
virtual void	setDetectShadows (bool detectShadows) =0 Enables or disables shadow detection. More...
virtual void	setHistory (int history) =0 Sets the number of last frames that affect the background model. More...
virtual void	setNMixtures (int nmixtures) =0 Sets the number of gaussian components in the background model. More...
virtual void	setShadowThreshold (double threshold) =0 Sets the shadow threshold. More...
virtual void	setShadowValue (int value) =0 Sets the shadow value. More...
virtual void	setVarInit (double varInit) =0 Sets the initial variance of each gaussian component. More...
virtual void	setVarMax (double varMax) =0
virtual void	setVarMin (double varMin) =0
virtual void	setVarThreshold (double varThreshold) =0 Sets the variance threshold for the pixel-model match. More...
virtual void	setVarThresholdGen (double varThresholdGen) =0 Sets the variance threshold for the pixel-model match used for new mixture component generation. More...

图 2　公共成员函数

Fig. 2 Public member functions

本文所描述的函数 setVarThreshold( ) 和 createBackgroundSubtractorMOG2O( ) 都是 OpenCV 库中用于背景减除的函数。其中，setVarThreshold( ) 函数用于设定高斯混合背景模型中像素方差的阈值，从而实现对噪声的滤除。createBackgroundSubtractorMOG2O( ) 函数则是创建一种基于高斯混合模型的背景减除器，用于提取视频中的前景物体。在创建背景减除器时，需要设定多个参数。其中，history 参数是指高斯混合模型所需要考虑的帧数，历史帧数越多，背景效果越好，但计算量也会相应增加。DetectShadows 参数用于指定是否检测阴影，当该参数设定为 true 时，背景减除器会将阴影区域也视为前景物体进行处理。

2.2　Kalman 滤波算法

卡尔曼滤波（Kalman filtering）是一种最小化均方误差的线性滤波方法，它在观测方程和目标状态转移方程的基础上，使用迭代的方式对目标的状态进行预测和更新[8]。其公式如下：

$$\begin{cases} x_t = A_{t|t-1}x_{t-1} + \omega_{t-1} \\ z_t = Hx_t + v_t \end{cases} \tag{4}$$

其中，z_t 表示 t 时刻系统的观测状态；x_t 和 x_{t-1} 对应 t 时刻和 $t-1$ 时刻系统的状态。$A_{t|t-1}$ 表示状态转移矩阵，描述目标的运动方式；H 是观测矩阵，描述内部状态和观测值之间的关系。v_t 对应测量噪声，服从正态分布，其协方差为 R；ω_t 对应系统噪声，同样满足正态分布，其协方差为 Q。预测和更新方程如下：

$$\begin{cases} x^-_{t|t-1} = A_{t|t-1}x^u_{t-1} \\ P^-_{t|t-1} = A_{t|t-1}P_{t-1}A^T_{t|t-1} + Q_{t-1} \\ K_t = P^-_{t|t-1}H^t(HP^-_{t|t-1}H^tR_t)^{-1} \\ x^u_t = x^-_{t|t-1} + K_t(z_t - Hx^-_{t|t-1}) \\ P_t = P^-_{t|t-1} - K_tHP^-_{t|t-1} \end{cases} \tag{5}$$

在预测阶段，滤波器利用上一状态对当前状态进行估计。在更新阶段，滤波器利用当前状态的观测值，优化预测阶段得到的预测值，以获得更加精确的新估计值。预测阶段的估计值和更新阶段的新估计值都是该滤波器的输出结果。通过不断迭代，滤波器可以逐渐减小估计值与真实值之间的误差，从而提高估计的准确性[9]。

考虑到在卡尔曼滤波器的初始化阶段造成的随机性较大，通常会对初始的协方差矩阵进行赋值，以尽可能减小初始化误差对滤波器性能的影响[8]。将协方差矩阵赋值为一个较大的正定对称矩阵：

$$P_0 = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \tag{6}$$

其中，P_0 是初始协方差矩阵，四个元素分别对应目标

位置、速度、面积和面积变化率的方差。在初始化后的预测和更新过程中,卡尔曼滤波器会利用观测值和状态转移方程逐步调整协方差矩阵,从而逐渐减小状态估计值和真实值之间的误差,提高估计的准确性。

本文通过类 TKalmanFilter 编程实现 Kalman 滤波算法。其中包括的成员函数有:

(1)在 TkaLmanFilter(Point2f pt, float dt, float Accel_noise_mag)构造函数中,传入了目标的初始位置 pt、时间间隔 dt 和加速度噪声幅度 Accel_noise_mag,初始化了传递矩阵(matrix(A))、噪声矩阵($P'(k)$):$P'(k) = A \cdot P(k-1) \cdot At + Q$)、测量矩阵(matrix($H$))、预测状态量($x'(k)$):$x(k) = A \cdot x(k-1) + B \cdot u(k)$)、纠正状态量($x(k)$):$x(k) = x'(k) + K(k) \cdot (z(k) - H \cdot x'(k))$ 等参数;

(2)GetPrediction()函数用于计算预测状态,即利用上一个时刻的状态和传递矩阵 A 预测当前状态;

(3)Update(Point2f p, bool DataCorrect)函数用于更新状态,即利用测量值对预测状态进行校正,得到更加准确的当前状态。

通过不断迭代,Kalman 滤波器可以逐渐减小状态估计值和真实值之间的误差,从而提高估计的准确性。该算法实现效果如图 3 所示。

(a)跟踪车辆目标

(b)跟踪行人目标

图 3 高斯背景建模 + Kalman 滤波算法实现效果图

Fig. 3 Visualization of the implementation of Gaussian background modeling and Kalman filtering algorithm

2.3 匈牙利算法

为了解决 Kalman 滤波算法只能跟踪单一目标的缺陷,设计了匈牙利算法,从而实现了多目标的定位与跟踪。匈牙利算法是一种在多项式时间内求解任务分配问题的组合优化算法[10],如图 4 是一组二分图,匈牙利算法寻求其中的最大匹配。

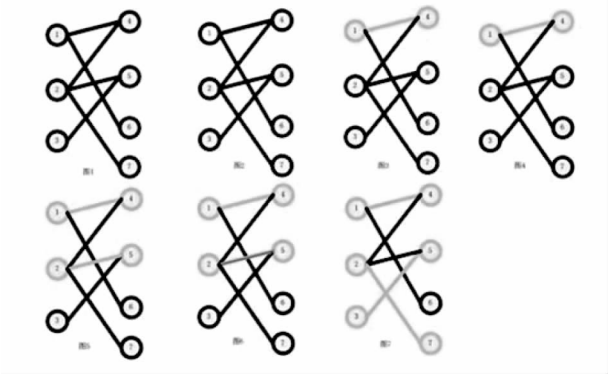


图 4 匈牙利算法示意图

Fig. 4 Schematic diagram of Hungarian algorithm

匈牙利算法的核心思想是将任务分配问题转化为图论中的匹配问题。在多目标跟踪中,将每个检测到的目标看作左部顶点,将每个观测值看作右部顶点,然后根据观测值与预测值之间的距离,计算左部顶点与右部顶点之间的边的权重。接下来,可以使用匈牙利算法来找到最大权重匹配,从而将每个目标的观测值关联起来,其基本流程如图 5 所示。

根据匈牙利算法的流程,可以看出其核心是通过寻找增广路径来找到最大匹配。具体来说,算法先将图 G 的最大匹配初始化为空,然后从一个未匹配的点 X_i 开始,在图 G 中寻找新的增广路径,如果找到了增广路径,则将路径上的边加入最大匹配中,否则说明已经找到了最大匹配,算法结束。最后输出图 G 的最大匹配,即所有已匹配的边的集合,它包含了所有两个节点之间的匹配边。

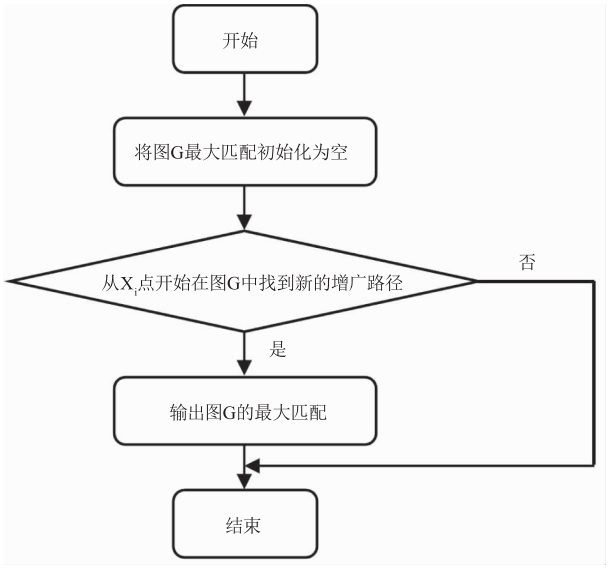


图 5 匈牙利算法基本流程图

Fig. 5 Basic flowchart of Hungarian algorithm

从 X_i 顶点开始搜索增广路径的流程如图 6 所示，在搜索增广路径的过程中，使用递归的方式来查找路径，每次递归都会从一个右部顶点开始，继续寻找下一个未匹配的左部顶点。如果找到了增广路径，就将路径上的所有点进行匹配，并返回 true，否则返回 false。

```
while（从Xi的邻接表中找到下一个关联顶点Yj）；
{
  if（顶点Yj不在增广路径上）；
  {
    将Yj加入增广路径；
    if（Yj是未覆盖点或者从与Yj相关连的顶点XK能找到增广路径）；
    {
      将Yj的关联顶点修改为Xi；
      从顶点Xi开始有增广路径，返回true；
    }
  }
  从顶点Xi开始没有增广路径，返回true；
}
```

图 6　搜索增广矩阵流程图

Fig. 6 Flowchart of searching augmenting matrix

基于上述理论，本文定义了一个 CTracker 类，通过它的成员函数 Update(vector < Point2d > & detections) 实现多目标定位与跟踪。其实现步骤如下：

(1)如果有新的目标，初始化一个 CTracker；在这个步骤中，如果有新的目标需要跟踪，就需要创建一个新的 CTracker 对象。在创建 CTracker 对象时，会使用 Kalman 滤波器对目标进行初始化，包括目标的位置、速度、加速度等状态量。初始化完成后，CTracker 对象就可以进行跟踪；

(2)ID 分配由 CTracker 对象执行。在这个步骤中，需要使用匈牙利算法来将检测到的目标与已经跟踪的目标进行匹配。具体来说，将当前的目标位置信息作为检测结果，将已经跟踪的目标位置信息作为已知的真实值，然后使用匈牙利算法来找到最优的匹配方案。匈牙利算法会将每个检测结果分配给一个已知的真实值，或者将其判定为新的目标。如果某个已知的真实值没有被分配到任何一个检测结果，就认为该目标已经消失。如果某个检测结果没有被分配到任何一个已知的真实值，就认为该检测结果对应一个新的目标。对于每个已经跟踪的目标，CTracker 对象会使用 Kalman 滤波器来更新其状态量，包括位置、速度、加速度等。如果有新的目标，就创建一个新的 CTracker 对象来进行跟踪。

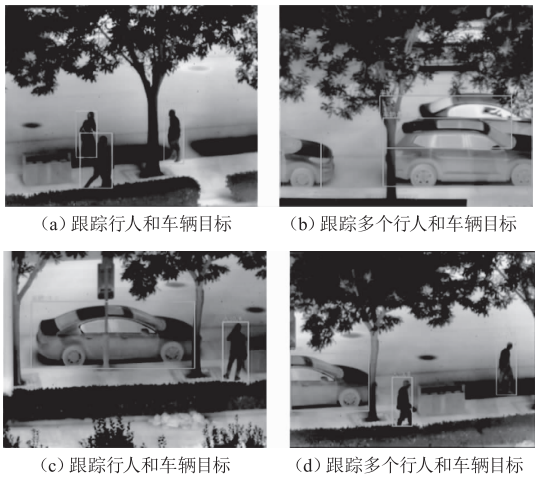
总的来说，这个算法的流程包括目标的初始化、跟踪和 ID 分配。其中，匈牙利算法是用来将检测结果与已知的真实值进行匹配的，Kalman 滤波器则是用来对目标状态进行估计和预测的。

本算法使用 OpenCV 开源计算机视觉库进行 C ++ 开发，Windows 开发是在 Visual Studio 2017 上进行的，算法的结构如图 7 所示。

图 7　算法结构

Fig. 7 Algorithm structure

其中，HungarianAlg 实现了匈牙利算法，为 Ctracker 提供了算法接口。Kalman 实现了 Kalman 滤波算法，为 Ctracker 提供了算法接口。Detector 包含图 1 中检测部分的算法，Ctracker 包含图 1 中跟踪部分的算法，Main 函数为代码的主程序。该部分算法实现效果如图 8 所示。



(a) 跟踪行人和车辆目标　(b) 跟踪多个行人和车辆目标

(c) 跟踪行人和车辆目标　(d) 跟踪多个行人和车辆目标

图 8　高斯背景建模 + Kalman 滤波 + 匈牙利算法实现效果图

Fig. 8 Implementation results of Gaussian background modeling, Kalman filtering, and Hungarian algorithm

3 多目标跟踪实验性能评估

本文通过在训练数据集上分析和验证算法的有效性,采用 MOT 基准挑战中的评价指标来评估算法性能。对于多目标跟踪问题,一般认为一个理想的评价指标应该满足下述三点要求:

(1)所有出现的目标都要能够及时找到,避免出现误检或漏检的情况;

(2)找到目标位置要尽可能和真实目标位置保持一致;

(3)保持跟踪一致性,避免跟踪目标的跳变。

3.1 MOT 基准挑战中的评价指标

3.1.1 多目标跟踪准确率(MOTA)

MOTA 是一种度量跟踪器整体准确率的指标,它通过计算所有帧上的误差距离、误检测、漏检等来衡量跟踪器的整体准确率[11]。具体公式如下:

$$MOTA = 1 - \frac{\sum_t (FN_t + FP_t + IDSW_t)}{\sum_t GT_t} \quad (7)$$

其中,FP 是误检测数;FN 是漏检数;ID 是 ID 错误变换数;GT 是真实目标数。MOTA 的取值范围是[$-\infty$, 100],越接近 100 表示跟踪器的性能越好[12]。

3.1.2 最多跟踪目标率(MT)

MT 指的是在所有帧中正确跟踪的目标数占所有真实目标数的比例[12],表示跟踪器对于每个目标的跟踪结果在目标出现的时间段内(至少 80 % 的时间)与真实轨迹的交并比(IoU)达到阈值(通常为 0.5)以上的目标数量所占的比例。

3.1.3 最多丢失目标率(ML)

ML 指的是在所有跟踪目标中,满足 Ground Truth 在小于 20 % 的时间内匹配成功的跟踪目标所占的比例。

3.1.4 ID Switch

在多目标跟踪算法中,ID Switch 是指 Ground Truth 所分配的目标 ID 发生变化的次数。具体来说,当一个跟踪器在某一帧中将一个目标的 ID 与 Ground Truth 所分配的 ID 不一致时,就会发生 ID Switch。如图 9(a)所示,当跟踪器在第二帧将目标 1 的 ID 从 1 变为 2 时,就发生了一次 ID Switch。一般情况下,ID Switch 越少,说明跟踪算法的性能越好。因为 ID Switch 过多会导致目标的不稳定性,对跟踪效果产生不利影响[13]。

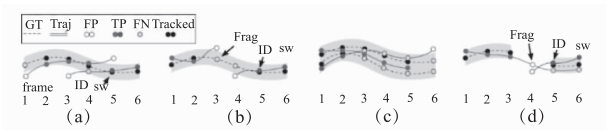


图 9 Tracker-to-target 匹配示意图

Fig. 9 Illustration of tracker-to-target matching

3.1.5 FP(False Positive)

在目标跟踪中,FP 通常是指误判为目标的非目标物体或误判为同一目标的多个轮廓。FP 的存在会导致目标跟踪算法的精度下降。

3.1.6 FN(False Negative)

FN 表示在当前帧中,未被匹配的 ground truth 点,即预测的追踪目标点没有与预测点成功匹配上,被称为 FN(也可以称为 Miss)。在目标跟踪中,FN 通常是指目标在当前帧中没有被正确识别或跟踪到的情况。

3.2 实验结果分析

本文应用工业派开发板进行算法程序设计开发,充分利用了工业派的 ARM 和 DSP 资源。在 ARM 芯片上运行 Linux 操作系统,控制数据的传输。在 DSP 上进行数学运算,实现算法的设计。

实验环境配置:由于嵌入式系统的硬件资源较少,不适合在其上进行开发。为了提高算法设计与实现的成功率,首先在 Windows 操作系统下进行算法设计,然后通过 Qt 将其移植到工业派开发板中。硬件环境配置为 CPU:ARM Cortex A73@1.6 GHz + 双核 ARM Cortex A53@1.2 GHz, GPU:ARM Mali G71@900 MHz。软件平台使用 Visual Studio 2017 + OpenCV 环境。

实验所采用的数据集为 MOT16[14],该数据集包含 14 个视频序列,其中 7 个视频序列用于训练,另外 7 个视频序列用于测试。每个视频序列都包含完整的目标注释信息,包括每个目标在每个时间步的位置、大小和类别等信息。

将本文算法与 Sort 跟踪器以及最近在 MOT16 数据集测试的其他方法进行比较,实验结果如表 1 所示。

与 Sort 跟踪器相比,本文算法在 MOT16 数据集上将 MOTA 提升了 1.4 %,MT 提升了 11.9 %,ML 降低了 6.5 %,IDs 降低了 33.3 %,FN 降低了 13.6 %。同时,在大多数指标上,本文算法也优于其他方法,例如在 MOTA 和 MT 指标上,本文算法的

表现超过了跟踪器 Sort、IOU 和 EAMTT。这表明本文算法在多目标跟踪任务中具有一定的优势,可以更准确地跟踪目标并降低跟踪误差。

表 1 本文算法与其他跟踪器的比较结果

Tab. 1 Comparison results of proposed algorithm and other trackers

追踪器	MOTA/%	MT/%	ML/%	IDs	FN
EAMTT	52.5	19	34.9	276	4040
IOU	57.1	23.6	32.9	312	4352
Sort	59.8	25.4	22.7	366	4517
本文算法	61.2	37.3	16.2	244	3914

图 10 和图 11 给出了在 MOT16 数据集上采用 Sort 方法和本文方法进行实验的跟踪效果图。通过对比两个方法的跟踪结果,可以发现本文算法相对于 Sort 方法具有更准确的跟踪结果。例如,在图 10(a)中,Sort 跟踪器没有识别到左下角的行人目标,而本文算法成功地将它们分别跟踪了下来。此外,在图 11(b)中,Sort 跟踪器对于一些复杂的场景(例如右下角行人之间的遮挡)跟踪效果不佳,而本文算法能够更好地处理这些情况,提高了跟踪的准确性,成功实现了对多个移动目标的跟踪。

(a) (b)

图 10 Sort 方法的多目标跟踪效果图

Fig. 10 Multi-object tracking results of the SORT method

(a) (b)

图 11 本文算法的多目标跟踪效果图

Fig. 11 Multi-object tracking results of proposed algorithm

4 结 论

本文提出了一种基于红外图像处理的移动目标识别跟踪算法,该算法通过结合高斯背景建模、Kalman滤波以及匈牙利算法,实现了在复杂场景中对移动目标物体的识别与跟踪,解决了多移动目标遮挡重叠问题。实验结果表明,本文算法在 MOT16 数据集上取得了较好的结果,相比于 Sort 跟踪器和其他方法,能够有效减少由于目标交错和遮挡导致的跟踪失败的情况,成功实现了对多个移动目标的跟踪与定位。此外,本文算法成功应用于热像仪系统中,也取得了较好的效果。综上,本文算法在目标识别跟踪方面具有较高的准确性和鲁棒性,能够有效地应用于多变的战场环境中。未来,可以进一步优化算法,提高其实时性和鲁棒性,从而更好地服务于军事领域。

参考文献:

[1] Li X, Zhu S, Chen L. Statistical background model-based target detection[J]. Pattern Analysis and Applications, 2016, 19: 783 – 791.

[2] Qiu S, Luo J, Yang S, et al. A moving target extraction algorithm based on the fusion of infrared and visible images[J]. Infrared Physics & Technology, 2019, 98: 285 – 291.

[3] Liu D, Li Z, Wang X, et al. Moving target detection by nonlinear adaptive filtering on temporal profiles in infrared image sequences[J]. Infrared Physics & Technology, 2015, 73: 41 – 48.

[4] Lien S F, Hsia K H, Su J P. Moving target tracking based on CamShift approach and Kalman filter[J]. Applied Mathematics & Information Sciences, 2015, 9(1): 395 – 401.

[5] Akula A, Singh A, Ghosh R, et al. Target recognition in infrared imagery using convolutional neural network[C]// Proceedings of International Conference on Computer Vision and Image Processing(CVIP), 2016: 25 – 34.

[6] Zivkovic Z. Improved adaptive Gaussian mixture model for background subtraction[C]//Proceedings of the 17th International Conference on Pattern Recognition, 2004: 28 – 31.

[7] Stauffer C, Grimson W E L. Adaptive background mixture models for realtime tracking[C]//Proceedings 1999 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. IEEE, 1999, 2: 246 – 252.

[8] ZHAO Guanghui, ZHUO Song, XU Xiaolong. Multi-object tracking algorithm based on Kalman filter[J]. Computer Science, 2018, 45(8): 253 – 257, 276. (in Chinese)
赵广辉,卓松,徐晓龙. 基于卡尔曼滤波的多目标跟踪方法[J]. 计算机科学, 2018, 45(8): 253 – 257, 276.

[9] Chen Z, Huang L. Online multi-target tracking algorithm based on Kalman filter and multiple information fusion [J]. Information/Communication, 2019, 17(3): 35 – 38. (in Chinese)
陈志鸿,黄立勤. 基于卡尔曼滤波和多种信息融合的在线多目标跟踪算法[J]. 信息通信, 2019, (3): 35 – 38.

[10] LIU Xingyu, GUO Ronghua, REN Chengcai, et al. Distributed target assignment method for UAV swarms using identity Hungarian algorithm [J]. Acta Armamentarii, 2023, 44(9): 2824 – 2835. (in Chinese)
刘兴宇,郭荣化,任成才,等. 基于身份匈牙利算法的无人机蜂群分布式目标分配方法[J]. 兵工学报, 2023, 44(9): 2824 – 2835.

[11] Bernardin K, Stiefelhagen R. Evaluating multiple object tracking performance: the clear mot metrics[J]. EURASIP Journal on Image and Video Processing, 2008, (1): 1 – 10.

[12] Leal-Taixé L, Milan A, Reid I, et al. Motchallenge 2015: Towards a benchmark for multi-target tracking[J]. arXiv preprint arXiv: 1504.01942, 2015.

[13] Wojke N, Bewley A, Paulus D. Simple online and realtime tracking with a deep association metric[C]//2017 IEEE International Conference on Image Processing, 2017: 3645 – 3649.

[14] Milan A, Leal-Taixé L, Reid I, et al. MOT16: a benchmark for multi-object tracking[J]. arXiv preprint arXiv: 1603.00831, 2016.