

# 基于自适应跟踪窗的红外小目标跟踪方法\*

汪鲁才<sup>1</sup>, 易锡年<sup>1</sup>, 陈小天<sup>2</sup>

(1. 湖南师范大学 工学院, 长沙 410081; 2. 国防科学技术大学 ATR 国防重点实验室, 长沙 410073)

**摘要:** 针对采用固定跟踪窗的传统跟踪方法容易将云层边角等干扰信息纳入跟踪窗, 从而影响目标稳定跟踪的问题, 提出了一种自适应跟踪窗的算法。在检测到目标的前提下, 通过 Kalman 滤波器动态地改变目标跟踪过程中跟踪窗的大小, 以减少跟踪过程中云层等干扰信息对跟踪的影响, 并减少算法需要处理的信息量。在真实红外跟踪平台上的实验表明, 提出的算法在干扰抑制和运行速度方面都有一定程度的提高。

**关键词:** 红外小目标; 游程编码标记; 航迹关联; Kalman 滤波; 自适应跟踪窗

**中图分类号:** TP391.41      **文献标志码:** A      **文章编号:** 1001-3695(2012)08-3185-04

doi:10.3969/j.issn.1001-3695.2012.08.103

## Algorithm for dim target tracking in infrared image based on adaptive tracking window

WANG Lu-cai<sup>1</sup>, YI Xi-nian<sup>1</sup>, CHEN Xiao-tian<sup>2</sup>

(1. College of Polytechnology, Hunan Normal University, Changsha 410081, China; 2. ATR Key Lab, National University of Defense Technology, Changsha 410073, China)

**Abstract:** In view of the problem of tracking method traditional use fixed tracking window was easy to bring the interference, such as cloud edge information into the tracking window, which impacted the robustness of target tracking, this paper presented a kind of adaptive tracking window algorithm. For the purpose of reducing the impact from the cloud edge information in tracking process and the amount of information needed to process, this algorithm dynamically changed the size of the track window by Kalman filter while target tracking on the premise of detected the target. By being tested on the real infrared tracking platform, the proposed algorithm shows a certain degree of improvement both in interference suppression and the processing speed.

**Key words:** dim target; RLE mark; track correlation; Kalman filter; adaptive tracking window

为了尽可能早地发现目标, 使防御武器有足够的反应时间, 要求目标在很远处就能被红外探测器检测到, 这时目标的图像很小, 只有一个或几个像素的面积, 在红外图像平面上表现为弱点状或不稳定斑点状, 这样的目标俗称为红外小目标。红外图像序列中检测跟踪红外小目标的问题由于其战略意义, 一直是这些年来研究的热门话题。红外小目标检测跟踪算法大致可以分为两类: a) 跟踪前检测方法, b) 检测前跟踪方法。跟踪前检测方法先在每帧图像中通过空域特性检测出目标, 然后通过时域关联实现目标的跟踪<sup>[1]</sup>, 虽然它在信噪比低时, 性能欠佳, 但是计算量小, 便于硬件实现。检测前跟踪算法是一个利用多帧图像综合时域、空域和目标的强度信息的基于时域算法<sup>[2,3]</sup>。与跟踪前检测算法相比, 检测前跟踪算法在低信噪比的条件下性能更优, 但计算更复杂。

正因为跟踪前检测算法计算简单, 易于硬件实现, 在实际应用, 特别是在一些有严格实时要求的应用场合中被广泛采用<sup>[4]</sup>。本文从抑制干扰、减少计算量、提高算法的实时性等方面出发, 提出一种基于自适应跟踪窗的红外小目标实时跟踪算法。

## 1 硬件平台及算法分析

### 1.1 硬件平台结构及参数

本文的硬件平台结构图如图 1 所示。在上位机 PC 指定搜索区域范围并发出跟踪指令后, FPGA 将红外热像仪的图像送入 DSP 进行处理; DSP 完成对红外目标的检测和跟踪即在场景中确认真实目标后, 将目标的坐标值通过 FPGA 回发给上位机 PC 用于显示目标位置, 并将目标的坐标值和图像的中心坐标值的偏差信息通过 FPGA 发送给转台, 以控制转台移动, 使目标始终锁定在图像中心位置。其中, 红外热像仪的输出图像尺寸为  $640 \times 512$ , 帧频为 25 Hz, 所以要实现实时处理, 检测跟踪算法的时间必须控制在 40 ms 以内。视场大小为  $12.5^\circ \times 10^\circ$ , 可探测距离为 10 km, 成像距离为 8 km。在 8 km 左右处的  $10 \text{ m} \times 10 \text{ m}$  大小的空中飞行目标在图像平面的投影尺寸约为  $3 \times 3$  个像素, 基本上呈现为一个较小尺寸的图像目标, 且灰度值比较高; 而对于纯净天空背景, 灰度值比较低, 如图 2(a) 所示。转台在水平范围内可以  $0 \sim 360^\circ$  无限位连续旋转, 水平旋转速度为  $0.1 \sim 45^\circ/\text{s}$ , 俯仰转动范围为  $-45 \sim +45^\circ$ , 俯仰旋转

收稿日期: 2011-11-25; 修回日期: 2011-12-29      基金项目: 湖南省科技厅科研基金资助项目

作者简介: 汪鲁才(1968-), 男, 湖北麻城人, 教授, 研究生导师, 博士, 主要研究方向为卫星图像处理、模式识别与智能系统(wlucai9776@vip.sina.com); 易锡年(1982-), 男, 湖南望城人, 硕士, 主要研究方向为图像处理; 陈小天(1980-), 男, 湖南岳阳人, 博士, 主要研究方向为图像处理与光学制导。

速度为 0.1 ~ 30°/s。PC 发出启动跟踪指令 2 s 以后,图像中的目标锁定到图像中心位置,处于稳定跟踪状态。

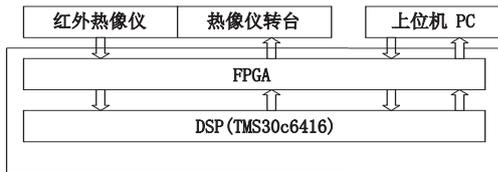


图1 硬件结构

本文算法在 DSP 上实现,整个算法的流程如图 3 所示。

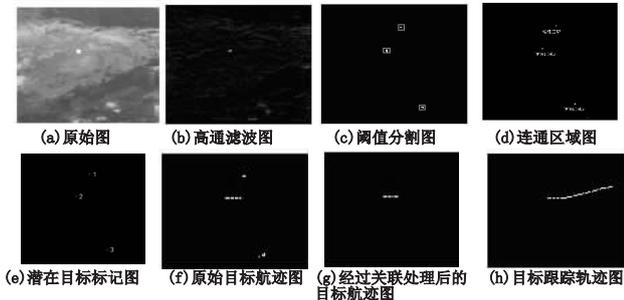


图2 算法处理结果

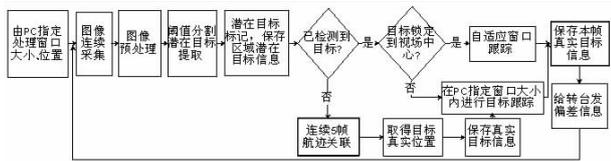


图3 算法流程

1.2 图像预处理

一幅大小为  $R \times C$  的图像可以按如下模型进行分解<sup>[5]</sup>：

$$f(i,j) = s(i,j) + n(i,j) \quad i, j \in [0, R-1] \times [0, C-1] \quad (1)$$

其中： $s(i,j)$  为目标灰度分布， $n(i,j)$  为背景灰度分布。对于远距离红外小目标而言，背景中细节成分较少，小目标在图像中与背景相比是一些突出的峰值点，其中含有大量的高频信息，且与周围背景的相关性小。而背景由一些缓慢移动的云层组成，大部分为低频信息，且噪声具有高斯特征。利用背景像素之间灰度的相关性和目标灰度与背景灰度的无关性，设计一个简单的空间高通滤波器，就可以滤除大量的背景像素，而只保留高频噪声和目标点，实现目标与背景的初步分离，如图 2(b) 所示。

1.3 阈值分割,潜在目标提取

本文采用基于似然比的目标检测方法完成潜在目标提取。根据检测理论,在图像中寻找目标的过程可以归结成为一个二元假设检验的过程,构建如下两个假设:

$$H_0: f = \hat{n}, H_1: f = s + \hat{n} \quad (2)$$

其中： $H_1$  代表包含目标的灰度分布， $H_0$  代表噪声和背景残差的灰度分布。在  $H_0$  为真的前提下， $f$  的似然函数为

$$p(f|H_0) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{f^2}{2\sigma_n^2}\right) \quad (3)$$

由于光学成像特性,目标的  $s$  的灰度同样符合高斯分布的要求<sup>[6]</sup>,在  $H_1$  为真的前提下, $f$  的似然函数为

$$p(f|H_1) = \frac{1}{\sqrt{2\pi(\sigma_s^2 + \sigma_n^2)}} \exp\left(-\frac{(f-m)^2}{2(\sigma_s^2 + \sigma_n^2)}\right) \quad (4)$$

由于检测的是小目标,目标面积仅有 1 个像素,此外目标的灰度分布应该相对均匀,可以认为  $\sigma_s^2 \ll \sigma_n^2$ ,可得出如下似然比检验公式:

$$\lambda(f) = \frac{p(f|H_1)}{p(f|H_0)} \approx \exp\left(-\frac{m^2 - 2fm}{2\sigma_n^2}\right) \quad (5)$$

根据似然比检验理论,可得到如下公式:

$$H_1: \lambda(f) > \eta, H_0: \lambda(f) < \eta \quad (6)$$

对式(6)两侧取对数,经过变换,可得到如下公式:

$$H_1: f > \ln \eta \times \frac{\sigma^2}{m} + \frac{m}{2}, H_0: f < \ln \eta \times \frac{\sigma^2}{m} + \frac{m}{2} \quad (7)$$

式(7)给出了基于像素灰度如何进行检测的公式,但由于实际运算中,目标的灰度均值  $m$  是未知的,所以以局部检测窗口的灰度均值  $u$  来代替  $\frac{m}{2}$ ,令  $k = \ln \eta \times \frac{\sigma}{m}$ ,则可得到如下检测门限:

$$Th = u + k\sigma \quad (8)$$

对于经高通滤波后的图像,可由式(8)进行阈值分割,实现目标和背景的分,  $k$  的值一般由图像的信噪比来确认。图像的信噪比定义为  $SNR = \frac{s - \mu}{\sigma}$ ,其中  $s$  为像素灰度,  $\mu$  为图像的均值,  $\sigma$  为图像的方差。假设分割门限  $Th$  已经确定,那么图像中所有灰度值大于门限  $Th$  的像素就认为是可能的目标点,其余像素为背景像素,则像素的检测概率为  $P_t = \int_{Th}^{\infty} \frac{1}{\sqrt{2\pi}\sigma} \exp\left[-\frac{(x-s-\mu)^2}{2\sigma^2}\right] dx$ ,当背景的均值较小时,  $SNR \approx \frac{s}{\sigma}$ 。令  $y = \frac{x-\mu}{\sigma}$ ,进行变量替换,则

$$P_t = \int_{\frac{Th-\mu}{\sigma}}^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}(y-SNR)^2\right] dy = 1 - \Phi\left(\frac{Th-\mu}{\sigma} - SNR\right) \quad (9)$$

可见为得到同样的检测概率,如果图像的信噪比大,分割门限  $Th$  应该取得大些,即  $k$  值应取得大些;反之, $k$  值应取得小一些。在实际应用中  $k$  的取值范围一般为 3 ~ 5。阈值分割后的结果如图 2(c) 所示。

1.4 潜在目标标记,保存区域目标信息

经过阈值分割之后,图像中还存在真实目标和部分残留的噪声点,将其统一称为潜在目标。为完成对点目标的识别,需要将检测出的潜在目标进行聚类,得到其目标特征。本文采用基于游程编码的目标标记算法对经过目标分割的图像进行聚类。目标标记逐列扫描,按照扫描先后进行排序,在扫描的过程中,将各个潜在目标的特性提取出来:

- 目标面积: FrameNowFeat[k]. candidateobject(n). area;
- 质心: FrameNowFeat[k]. candidateobject(n). centroid;
- 灰度均值: FrameNowFeat[k]. candidateobject(n). meanlight;
- 灰度峰值: FrameNowFeat[k]. candidateobject(n). peaklight.

其中:  $n = 1, 2, \dots, N(k)$ ;  $k$  为图像序列的帧数;  $N(k)$  为第  $k$  帧图像潜在目标聚类后的类别总数,即潜在目标区域的个数。经过潜在目标标记后,对于如图 2(d) 所示的阈值分割后的每一个连通区域,都有一个唯一的标记,如图 2(e) 所示。其中的信息如表 1 所示。

表 1 潜在目标信息

潜在目标	质心	面积	灰度均值	灰度峰值
FrameNowFeat[1]→ candidateobject(1)	(219,192)	3×3	199	201
FrameNowFeat[1]→ candidateobject(2)	(270,99)	3×3	183	184
FrameNowFeat[1]→ candidateobject(3)	(501,426)	3×3	212	215

### 1.5 航迹关联

在潜在目标提取阶段,可以基于目标的灰度以及面积特征对目标进行初步识别,初步排除噪声点的干扰,但实际应用中,单次检测难以满足要求,因此需要利用多次扫描图像。本文采用 5 次扫描图像后,进行目标航迹关联,去除虚假航迹,保留真实航迹。经过前面预处理的图像序列中噪声点的分布是独立的,而目标点的分布是关联的,因此采取合适的关联方法,就可以检测出目标的轨迹。图 2(f) 所示的原始目标航迹经过航迹关联后,结果如图 2(g) 所示,去除了虚假目标航迹,保留了真实目标的航迹。如表 2 所示,前阶段的 3 个潜在目标进行航迹关联后,只留下了真实目标,而其余两个干扰信息被去除。

表 2 真实目标信息

真实目标	质心	面积	灰度均值	灰度峰值
LastDetectResult	(273,103)	3 × 3	187	189

### 1.6 目标跟踪

1) 跟踪方法 在图像区域成功检测到目标后,则进入目标跟踪状态。跟踪步骤为:经过前面的阶段已经明确真实目标的正确信息,因此,跟踪阶段只需将潜在目标和前一帧的真实目标进行最近邻关联<sup>[7]</sup>即可。在目标标记后,本文统计了小目标的灰度均值和灰度峰值,分别记为  $I_u$  和  $I_p$ ,目标的形心坐标为  $(X_c, Y_c)$ 。因为在目标识别跟踪系统进行监测的短时间内,目标的热辐射变化不大,则其在探测器的成像平面上所成的目标像的灰度在短时间内也将变化不大。因此,利用目标的灰度信息可以完成初步的目标关联,即

$$\{|I_u^i(k) - I_u^i(k-1)|, |I_p^i(k) - I_p^i(k-1)|\} < \{\eta_1, \eta_2\} \quad (10)$$

其中:  $I_u^i$ 、 $I_p^i$  分别为上一帧真实目标的灰度均值和灰度峰值;  $\eta_1$ 、 $\eta_2$  分别为根据灰度均值和灰度峰值设定的阈值,本文分别设置为 20 和 10,即灰度均值差别不超过 20,灰度峰值差别不超过 10;  $i \in [1, N]$ ,  $N$  为当前帧潜在目标个数。由目标的运动速度以及目标与探测器的距离,根据空间分辨率的要求,可以算出目标在两次扫描过程中的最大像素位移量。因此,利用目标的位置信息可以完成目标的关联,即

$$\{|x_c^i(k) - x_c^i(k-1)|, |y_c^i(k) - y_c^i(k-1)|\} < \{\eta_1, \eta_2\} \quad (11)$$

其中:  $X_c^i$ 、 $Y_c^i$  为上一帧目标的真实形心;  $\eta_1$ 、 $\eta_2$  分别为目标在两次扫描过程中在  $x$  方向和  $y$  方向上的最大像素位移阈值,本文均设置为 10,也即  $x$  方向和  $y$  方向上的位移均不能超过 10;  $i \in [1, N]$ ,  $N$  为当前帧潜在目标个数。如果满足式(10)(11),则说明候选的潜在目标和真实目标相关,如果同时有多个候选目标满足两式,则取两式最小的为新的真实目标。目标跟踪过程中目标的轨迹如图 2(h) 所示。

2) 自适应跟踪窗算法 图 4 为远距离近似匀速飞行的空中目标运动过程中,通过转动转台,红外检测跟踪系统跟踪的目标形心位置变化图。从图中可以看出,通过转动转台进入稳定跟踪阶段,目标的形心基本被锁定在图像的中心坐标位置。这样在跟踪的稳定阶段,即启动跟踪大约 2 s 后,本文实验平台的热像仪的帧频为 25,即在连续跟踪 50 帧后,可以在目标形心位置设置一个根据转台和目标的运动信息自适应调整大小的跟踪窗口,只对跟踪窗口内的小部分图像信息进行处理,这样不仅能减少跟踪窗中干扰物对目标跟踪的影响,而且可以节省计算时间。

设输入图像的大小为  $640 \times 512$ ,则中心点坐标为 (320,

256),即为图像中心坐标,把它标记为 (CALIBRATIONX, CALIBRATIONY),记 memorydeltax 和 memorydeltay 分别为目标位置与图像中心位置的偏差,则有

$$\begin{aligned} \text{memorydeltax} &= \text{centerx} - \text{CALIBRATIONX} \\ \text{memorydeltay} &= \text{centery} - \text{CALIBRATIONY} \\ dx &= \text{centerx} - \text{lastdetectresult.cx} \\ dy &= \text{centery} - \text{lastdetectresult.cy} \end{aligned}$$

其中:centerx、centery 为当前帧真实目标的位置,lastdetectresult.cx、lastdetectresult.cy 为上一帧的真实目标坐标位置,则 dx、dy 为当前帧目标位置与上一帧目标位置的差值。目标运动时,将目标的真实坐标值和图像中心坐标的差值发送给转台,转台转动并锁定目标。这样,将 memorydeltax 和 memorydeltay、dx 和 dy 作为状态向量,设计一个卡尔曼滤波器,可以用来估计下一帧目标坐标位置和图像中心位置的差值。

$$X_p = \begin{bmatrix} \text{memorydeltax}[k] \\ \text{memorydeltay}[k] \\ dx[k] \\ dy[k] \end{bmatrix} = \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \text{memorydeltax}[k-1] \\ \text{memorydeltay}[k-1] \\ dx[k-1] \\ dy[k-1] \end{bmatrix} \quad (12)$$

$$\begin{bmatrix} \text{centerx}[k] - \text{CALIBRATIONX} \\ \text{centery}[k] - \text{CALIBRATIONY} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} \text{memorydeltax}[k] \\ \text{memorydeltay}[k] \\ dx[k] \\ dy[k] \end{bmatrix} \quad (13)$$

$$\begin{aligned} \text{误差协方差: } P(x(k)|x(k-1)) &= \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \\ P(x(k-1)|x(k-1)) &\times \begin{bmatrix} 1 & 0 & t & 0 \\ 0 & 1 & 0 & t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}^T + Q \quad (14) \end{aligned}$$

其中:  $Q$  为状态噪声协方差矩阵,在本文中,  $Q =$

$$\begin{bmatrix} 0.01 & & & \\ & 0.01 & & \\ & & 0.01 & \\ & & & 0.01 \end{bmatrix}, P(x(k-1)|x(k-1)) \text{ 的初值为}$$

$$P(x(1)|x(1)) = \begin{bmatrix} 100 & & & \\ & 100 & & \\ & & 100 & \\ & & & 100 \end{bmatrix} \quad (15)$$

卡尔曼增益:

$$Kg = P(x(k)|x(k-1)) \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^T \times \left( \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \times P(x(k)|x(k-1)) \times \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}^T + R \right)^{-1} \quad (16)$$

其中:  $R$  为观测噪声,  $R$  的初值  $R(1) = \begin{bmatrix} 0.2845 & 0.0045 \\ 0.0045 & 0.0045 \end{bmatrix}$ ,则  $k$  时刻的最优估计为

$$x(k|k) = (X_p + Kg \times$$

$$\left( \left[ \begin{array}{c} \text{centerx}[k] - \text{CALIBRATIONX} \\ \text{centery}[k] - \text{CALIBRATIONY} \end{array} \right] - \left[ \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{array} \right] \times X_p \right) \quad (17)$$

最后观测噪声  $R$  更新为

$$R = \begin{bmatrix} \text{centerx}[k] - \text{CALIBRATIONX} & \text{centerx}[k] - \text{centerx}[k-1] - x(k|k)[1] & & \\ & x(k|k)[3] & & \\ \text{centery}[k] - \text{CALIBRATIONY} & \text{centery}[k] - \text{centery}[k-1] - x(k|k)[4] & & \\ & x(k|k)[2] & & \end{bmatrix}$$

则下一帧目标坐标位置和图像中心位置在  $x$  方向上的估计差值为  $x(k|k)[1]$ ,  $y$  方向上的估计差值为  $x(k|k)[2]$ 。本文令下一帧中跟踪窗的宽度为  $2x(k|k)[1]$ , 高度为  $2x(k|k)[2]$ , 则  $k$  时刻跟踪窗口不再是固定的窗口大小, 而是根据卡尔曼滤波的最优估计值自适应地改变。

### 2 算法性能分析

本文将提出的自适应跟踪窗算法在硬件平台上进行测试, 图 5 为连续跟踪 1 000 帧远距离某近似匀速飞行的目标后, 目标运动时的坐标离图像中心坐标通过 Kalman 滤波估计后, 在  $x$  方向和  $y$  方向上的差值的估计误差。从图中可以看出, 随着跟踪帧数的增加, 估计误差越来越小。

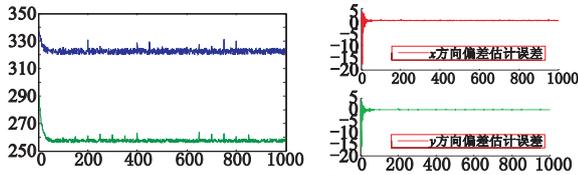


图4 跟踪过程中目标形心变化

图5 跟踪过程中估计误差曲线

图 6 为连续稳定跟踪 1 000 帧时, 跟踪窗口大小变化。从图中可以看出, 在连续跟踪 50 帧内时, 窗口大小为设定的固定大小; 当大于 50 帧, 处于稳定跟踪的阶段, 采用了自适应的窗口, 可以看出, 窗口大小在自适应的变化。从图 7 可看出, 跟踪窗口根据目标和转台的运动信息自适应不断变化, 在保证能检测到目标的前提下, 自适应地调整跟踪窗大小, 以降低外界干扰对跟踪的影响, 提高算法处理速度。

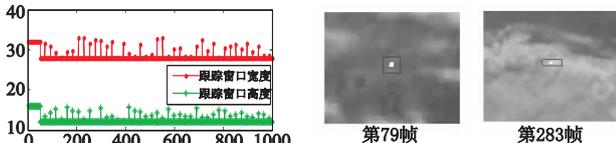


图6 跟踪过程中窗口大小变化

图7 跟踪过程中跟踪窗口变化

#### 2.1 干扰抑制方面

从图 8 可以看出, 在目标从干扰云层中穿梭时, 采用固定大小的跟踪窗口因为窗口较大, 很容易将云层边角等干扰纳入跟踪框, 如图 8(a) 的原始图经阈值分割后, 变为(b)所示图像。在图 8(b)中, 标记 3 为目标信息, 而标记 1、2 属于干扰信息, 而这样的干扰从外形上看和目标非常相似, 而且由于云层的运动和目标运动具有一定的相似性, 所以从特性上很难分清是目标还是干扰信息, 从而对目标的正确定位产生严重的影响。而采用自适应跟踪窗口后, 从图 9(a) 可以看出, 它可以在保证目标位于跟踪框内的前提下, 尽量缩小跟踪框的大小, 从而尽可能地排除云层等严重干扰信息对目标跟踪的影响。如图 9(b) 所示, 将目标信息 3 纳入跟踪框内, 将干扰信息 1、2 排除在跟踪框外, 从而提高了跟踪的精度。本文选取的跟踪框的大小为 2 倍卡尔曼滤波的最优估计值。如果跟踪框选得太小, 如等于卡尔曼滤波最优估计值, 则在阈值分割阶段, 目标可能已经占据窗口的绝大部分区域, 或者跟踪窗口中只留下目标像素, 从而利用式(8)无法正确提取出目标点。

所以在实际应用中需要设置一个窗口大小的界限, 本文设为 2 倍最优估计值是为了保证目标的正确提取。

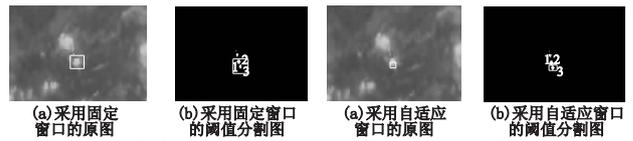


图8 采用固定窗口的原图和阈值分割图

图9 采用自适应窗口的原图和阈值分割图

#### 2.2 算法处理速度方面

由于采用了自适应的跟踪窗口, 跟踪框的大小在稳定跟踪阶段一般比固定大小的跟踪框要小, 从而缩短了跟踪过程中所需处理的信息量, 节约了处理时间, 提高了算法的实时性。在某红外目标跟踪器硬件平台上进行实验, 在连续跟踪红外空中小目标 1 000 帧的情况下, 算法的处理时间如图 10 所示。其中, 横轴坐标为帧数; 纵轴坐标为处理时间, 单位为  $s$ 。从图中可以看出, 本文算法每帧的平均处理时间为 27  $ms$  左右, 比采用固定大小跟踪框的平均处理时间大约减少了 8%。

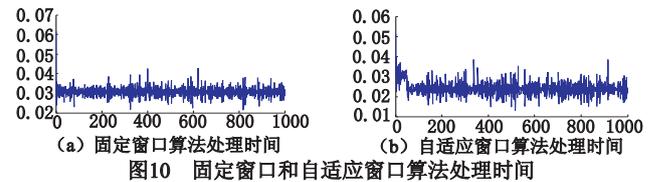


图10 固定窗口和自适应窗口算法处理时间

### 3 结束语

本文针对采用固定跟踪窗的传统空中红外小目标跟踪算法容易受云层干扰影响的问题, 提出了一种基于自适应跟踪窗的跟踪算法。通过在真实红外跟踪平台上进行连续跟踪试验, 本文方法在目标周边出现云层干扰时还能较好地准确跟踪目标, 而算法的平均处理时间为 27  $ms$ /帧左右, 比原算法的平均处理时间减少了大约 8%。

#### 参考文献:

- [1] MAEDA K, KONDO K, KONISHI Y, et al. Design of mixed-domain filters for detection of arbitrary trajectory signals by using extended complex Kalman filters[C]//Proc of the 45th Midwest Symposium Circuits and Systems. 2002:250-253.
- [2] WU Bin, YAN Hao. A novel track-before-detect algorithm for small dim infrared target[C]//Proc of International Conference on Multimedia and Signal Processing. Washington DC: IEEE Computer Society, 2011:103-106.
- [3] LEIGH A J, VIKRAM K. Performance analysis of a dynamic programming track before detect algorithm[J]. IEEE Trans on Aerospace and Electronic Systems, 2002, 38(1):228-242.
- [4] ZHOU Run-zhi, MA Liang-li, WANG Jiang-an, et al. Outlier mining based algorithm of infrared dim target detection in sea and sky background[C]//Proc of International Conference on Computational Intelligence and Software Engineering. 2009:1-4.
- [5] ZHANG hong, ZAO Bao-jun, MAO Er-ke. A real-time effective method for infrared point-target detection in spatially varying clutter [C]//Proc of International Conference on Radar. 2001:461-464.
- [6] XIONG Y. An extended track-before-detect algorithm for infrared target detection[J]. IEEE Trans on Aerospace and Electronic Systems, 1997, 33(3):1087-192.
- [7] 周武, 赵春霞, 张浩峰. 动态联合最近邻算法[J]. 电子学报, 2010, 38(2):359-365.