

运动小目标实时检测系统的设计与分析*

沈宇键 何 昕 郝志航

TP391.41

(中国科学院长春光学精密机械与物理研究所, 空间光学研究部, 吉林, 长春, 130022)

摘要 针对复杂背景条件下运动小目标的检测, 采用时域递归低通滤波和多帧相关的方法设计了一种运动小目标实时检测系统. 该系统适用于背景缓慢变化或背景处于相对静止的环境, 能够在恒虚警概率条件下, 很好地确定小目标的位置. 通过自适应阈值处理可以识别出小目标的轮廓.

关键词 时域递归低通滤波, 实时检测系统, 恒虚警概率, 自适应阈值.

运动小目标

计算机视觉

DESIGN AND ANALYSIS OF REAL-TIME DETECTION SYSTEM FOR SMALL TARGETS MOVING*

SHEN Yu-Jian HE Xin HAO Zhi-Hang

(Department of Space Optical Engineering, Changchun Institute of Optics, Fine Mechanics and Physics, Chinese Academy of Sciences Changchun, Jilin 130022, China)

Abstract Temporal recursive low-pass filtering algorithm and multi-frame correlation were presented to detect small moving targets in complicated background. A real-time implementation hardware system was designed, which fits to background varying slowly or stationary with respect to the environment. Target and its region location were determined precisely with the condition of constant false alarm probability. Contours of small targets were identified with adaptive threshold.

Key words temporal recursive low-pass filtering, real-time detection system, constant false alarm probability, adaptive threshold.

引言

运动小目标的探测与处理一直是计算机视觉与图像处理工作中的重要课题, 特别是在背景复杂, 距离较远的情况下, 很难有效地实时分离目标. 对于实时检测的要求, 通常有两种方法可以用来检测运动的小目标^[1], 一种是利用多帧相关技术估计出自适应变化的背景, 然后再将其去除, 从而获得运动的小目标. 这种方法能够比较准确的检测出小目标的轮廓, 而且在目标与背景的对比如度低, 噪声干扰大的情况下很有效. 但是这种方法要求摄像系统处于凝视状态, 也就是说背景不变或缓慢变化. 另一种方法是采用当前图像与前一帧图像相减来去除背景摄取目标. 这种方法适用于摄像视场移动的情况, 但需要将不同时刻的图像进行场景配准, 在目标与背景对比

度低、噪声干扰大的情况下, 检测小目标比较困难.

本文采用多帧相关的方法设计了一种较为简单的适用于背景缓慢变化条件下的运动小目标实时检测系统. 该系统能够在恒虚警概率条件下, 很好地识别出小目标的位置和轮廓.

1 实时检测系统算法原理

1.1 时域递归低通滤波

通常情况下, 目标运动的速度要比背景的变化快得多, 因此背景信号在时间上占据低频部分而目标则占据高频部分. 采用时域递归低通滤波可以通过逐帧累积不断更新背景, 达到预测新背景的目的. 背景预测起到了背景杂波白化和目标增强的双重作用^[2]. 利用当前输入帧再与预测的背景进行相关运算就可以提取出目标. 时域递归低通滤波原理的数

* 国防科工委“八五”期间重大项目基金资助项目
稿件收到日期 2000-04-27, 修改稿收到日期 2000-06-30

* The important project supported by the Foundation of National Defence Committee of Science and Industry in the Eighth "Five-Year" Plan of China
Received 2000-04-27, revised 2000-06-30

学表达式为

$$\hat{Y}_n = k \cdot I_n + (1 - k) \cdot \hat{Y}_{n-1} \quad (1)$$

式中 I_n 是当前输入帧； \hat{Y}_{n-1} 是不断累积的背景信号； \hat{Y}_n 则是新的背景预测，也可看作是参考帧图像； k 是一个平滑因子，取值介于 0 和 1 之间，它反映了实际输入帧的像素值影响参考帧像素值的大小的情况^[1]。

1.2 图像求差与确定阈值门限

如果背景的变化只受光照变化的影响，或者帧间采样频率足够快，那么背景的变化是缓慢的或是处于相对静止状态，这时背景是完全时间相关的，帧间相关也就转变为帧间相减，可表示为

$$D = I_n - \hat{Y}_n \quad (2)$$

D 代表帧差图像，是包含目标与噪声的图像。这一过程可以称为背景去除。为了探测目标的位置，选用一个适当的阈值 V_{th} 作为这个帧差图像的门限，即

$$S = \begin{cases} 1, & D > V_{th} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

这里 S 是小目标二值化轮廓。阈值 V_{th} 的选择与帧差图像的虚警概率有关。理想情况下，认为帧差图像的噪声是以白噪声为主，因此在恒虚警概率条件下，可按下式求出阈值门限^[2]：

$$P_{FA}(X) = 0.5 \left[1 - \operatorname{erf} \left\{ \frac{\operatorname{TNR}}{\sqrt{2}} \right\} \right] \quad (4)$$

$$\operatorname{erf} = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \quad (5)$$

$$\operatorname{TNR} = \frac{V_{th}}{\delta} \quad (6)$$

其中 $\operatorname{erf}(x)$ 是误差函数， P_{FA} 代表虚警概率， TNR 是阈值噪声比， δ 是帧差图像方差。根据给定的恒虚警概率就可以求出 TNR 的值，阈值 $V_{th} = \operatorname{TNR} \cdot \delta$ 。但是实际系统获得的帧差图像往往不一定以白噪声为主，因此以上求阈值的方法并不准确，如何对实际采集到的图像进行阈值处理将在后文具体说明。

2 实时检测系统设计

根据以上原理，设计实时检测系统，系统结构框图如图 1。整个系统由 CCD 数码相机、实时滤波运算单元、帧存储器单元、逻辑运算单元及数据通道构成。为了检测系统的工作状况，对数据进行离线计算和处理，系统中还包含高速 D/A 转换单元和微处理机控制单元。

CCD 相机选用 DALSA 公司的高速数字化面阵相机 CA-D4，相机的场频为 40Hz，数据率为 25MHz，单个像元尺寸为 $12\mu\text{m}$ 。整个系统的核心部分是实时滤波运算单元，由它完成式 (1) 的运算功能。这一单元可由 HARRIS 公司的专用图像处理芯片 HSP48212 来完成。HSP48212 是一种数字视频混和芯片，它的主要功能是在一个可编程加权因子的作用下，将两路视频源混合，其工作时钟频率可达 40MHz。HSP48212 实现的功能表达式为

$$D_{out} = 2^{-1} [DINA \cdot M + DINB \cdot (1 - M)] \quad (7)$$

式中 $DINA$ 和 $DINB$ 分别为两路输入，均为 12bit

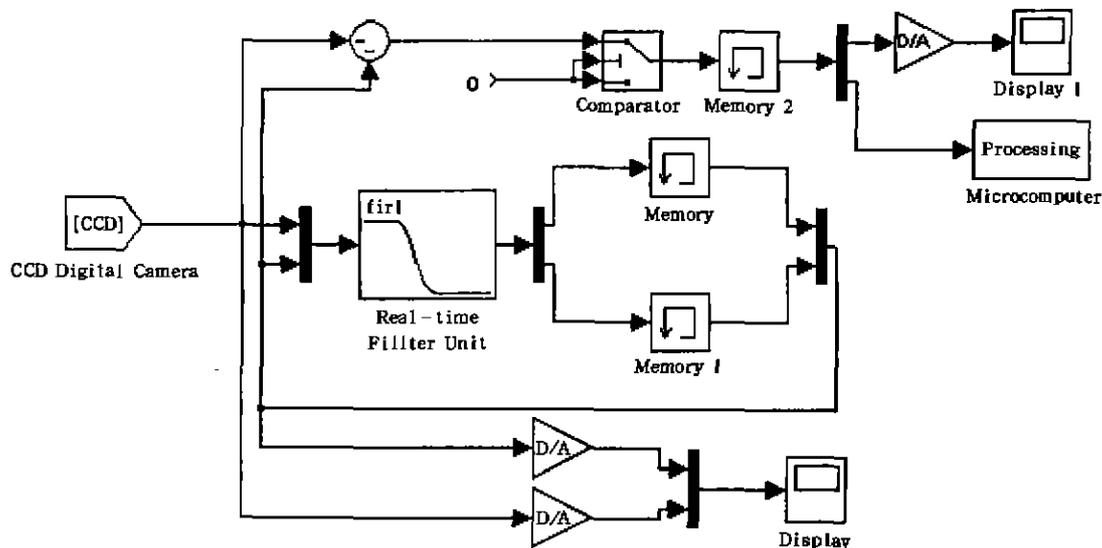


图 1 实时检测系统结构框图
Fig. 1 Block diagram of real-time detection system

数据. M 是可编程的加权因子, 取值在 $0\sim 1$ 之间, 也是 12bit 数据; 输出 D_{out} 是两路输出的加权平均, 是 13bit 数据. 比较式 (1) 和 (7), 两式很相似, M 因子代表式 (1) 中的系数 k , DINA 可表示为当前输入帧 I , DINB 则是多帧累积后的 \hat{Y}_{n-1} , D_{out} 对应新的背景预测 \hat{Y}_n . 只需对式 (7) 作除 2 运算就可实现式 (1) 的算法. 由于系统要求的数据位精度为 8 位, 因此式 (7) 中的常系数 2 可通过对输出数据右移一位来消除, 这样 D_{out} 的输出位为 $D_{out13}\sim D_{out5}$, 输出位数仍保持为 8 位. 逻辑控制单元和数据通道设计采用 CPLD (在系统可编程逻辑器件) 与 ispGDX (在系统可编程通用交叉阵列) 配合完成.

系统的工作原理是这样的: CCD 数字相机实时采集并传送外部图像数据到实时滤波运算单元, 该单元通过实时运算完成新的背景预测; 两个帧存储器分别存储实时滤波后的运算结果, 这两个帧存储器在工作时始终处于交替读写状态, 当一个帧存储器写入滤波运算后的数据时, 另一个帧存储器将上一帧的图像数据读出并反馈回实时滤波运算单元. 一帧图像数据运算完成后, 两个帧存储器的状态在逻辑控制单元的监控下完成切换, 这样就保证了累积运算后的图像数据能够逐帧反馈回实时滤波运算单元, 并与 CCD 相机实时采集的数据完成运算.

当前输入帧与反馈回实时滤波运算单元的数据除了进行滤波处理外还进行实时相减运算. 由于电路中存在噪声干扰, 相减的结果可能存在负值, 需要通过一个比较器来消除负值. 相减与比较的过程可通过 CPLD 中的宏单元设计来完成, 最终的结果暂存于目标存储器. 微处理机对目标存储器中暂存的图像进行阈值处理, 完成目标位置与轮廓的识别.

采用上述实时检测系统对运动的点光源进行检测实验. 实验结果如图 2~5. 图 2 是一幅实时采集到的 256 像素 \times 256 像素的含有运动点光源的原始图像, 背景是 windows98 的蓝天白云, 白云表示背景的起伏. 图 3 是这幅图像的三维表示, 从中可以看出背景起伏较大, 点光源目标不明显, 这种情况可近似为强背景下的运动小目标的检测. 在相邻两帧期间, 点光源在相机的焦平面移动的位移为 $224\mu\text{m}$, 移动速度为 8.96mm/s . 图 4 是递归 50 帧滤波后, 通过帧相减抑制背景得到的含点光源目标的图像, 图 5 是其三维示意图, 此时点光源目标已经清晰的显现出来, 但目标图像中也明显存在噪声和干扰, 需要进一步做阈值处理.



图 2 原始图像
Fig. 2 Original image

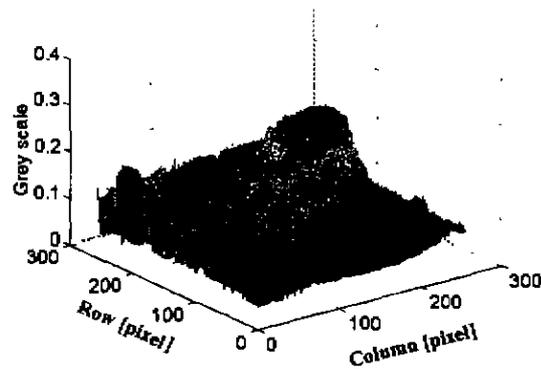


图 3 原始图像的三维表示
Fig. 3 3-D representation of original image

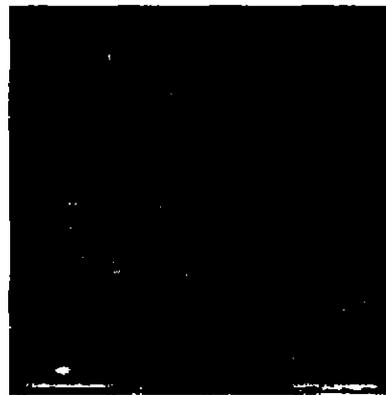


图 4 抑制背景后的帧差图像
Fig. 4 Difference image after background suppression

3 阈值处理

3.1 自适应阈值

前面提到 CCD 摄像系统的噪声不一定是以高斯分布为主的白噪声, 这样按照式 (4)~(6) 就不能准确的确定出分割目标轮廓的阈值, 需要采用其它的方法来确定阈值. 这里采用一种自适应阈值的方法来确定阈值, 它的数学表达式为^[4]

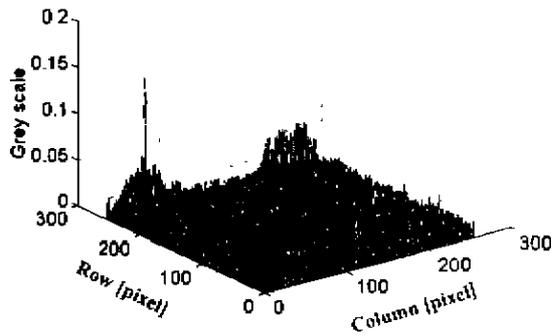


图5 帧差图像的三维表示

Fig. 5 3-D representation of difference image

$$V_{th} = E + \alpha \cdot \delta. \quad (8)$$

其中 E 是图像的均值, δ 是图像的方差, α 是一个与噪声有关的系数, 它在一次采集处理过程中对每一帧图像都保持不变. 这种自适应阈值的方法可以通过统计帧差图像的均值和方差来自适应的调整阈值以保证恒虚警率的要求.

3.2 帧差图像的阈值确定

对于实时检测系统采集到的目标图像可按式(8)来确定阈值, 但 α 系数需要先进行估算. 经验上, 在假定 10^{-5} 量级的虚警概率条件下, α 取值在 3~5 范围内. 实际上, 由于环境不同, 这个取值范围并不准确, 因此针对具体图像要先确定 α 的取值范围, α 的取值通常与目标的先验知识及获取的图像信息有关. 下面以图 4 为例, 具体说明如何确定 α 的取值 (这里所用图像的灰度值都是归一化的, 并假定最终提取的目标轮廓图像要有 10^{-5} 量级的虚警率): (1) 估算小目标占据的实际像素范围, 根据点光源在 CCD 相机成像面上表现为光学点扩散函数的成

像特点, 以及图 4 的实际等高线图 (见图 6 和 7, 利用 Matlab 语言中的 contour 函数绘出, 图 7 为图 6 中目标位置范围内的局部等高线图), 可以估算出小目标占据的像素范围大约为 28~35. (2) 确定初始化阈值. 对图 4 所示图像进行直方图统计, 统计直方图如图 8 所示. 根据直方图统计, 选择一个初始化阈值, 这个阈值能保证小目标一定能被提取出, 同时噪声和干扰则被完全抑制掉, 通过这个阈值可以确定 α 的初始值. 在直方图中, 紧密排列的聚类部分主要是背景噪声和干扰的值, 因此如果阈值选取刚好超过这一聚类, 那么噪声和干扰可以被抑制掉, 这里选择 $V_{th} = 0.055$. (3) 绘出 α 取值与小目标像素数的曲线图, 图 4 所示的帧差图像的均值和方差分别为 0.0032 和 0.0052. 将初始化阈值带入公式(8), 得到 α 的初始值为 9.9. 对 α 再进行多次试探取值, 并分别求出这些值对应的像素数, 绘出曲线如图 9. 从曲线中可以看出随着 α 值的增加, 可能属于小目标的像素数按指数阶下降. 在曲线中找出与目标像素数值为 28 时对应的 α 值, $\alpha = 9.3$. 再找出与目标像素数值为 35 时对应的 α 值, $\alpha = 8.6$. (4) 分离目标. 将 $\alpha = 9.3$ 和 $\alpha = 8.6$ 分别带入式(8)得到阈值 $V_{th} = 0.052$ 和 $V_{th} = 0.048$. 在实际测量时, 对于虚警概率的计算往往采用经验公式, 认为虚警概率是噪声或干扰超过探测阈值的概率^[5]. 分别计算 $V_{th} = 0.052$ 和 $V_{th} = 0.048$ 时图像的虚警概率为 1.5×10^{-5} 和 1.3×10^{-4} , 即经过这两个阈值处理得到的图像中的假目标数分别为 1 个和 8 个, 而 10^{-5} 量级的虚警率要求图像中的假目标数不能超过 6 个, 也就是说阈值 $V_{th} = 0.048$ 选取过低, 由于目标的特征形状表现为光学点扩散函数, 同时为了保证 10^{-5} 量级的恒虚警率要求, 这里采用阈值 $V_{th} = 0.052$ 进行图像

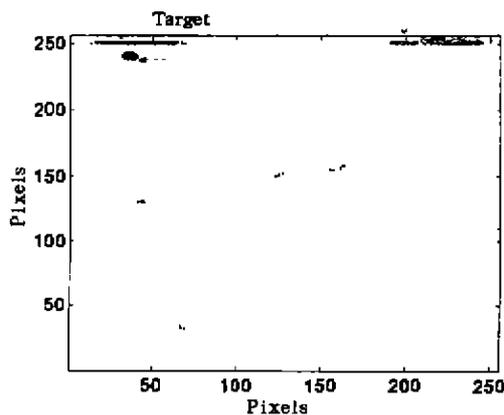


图6 帧差图像的等高线图

Fig. 6 Contour line of difference image

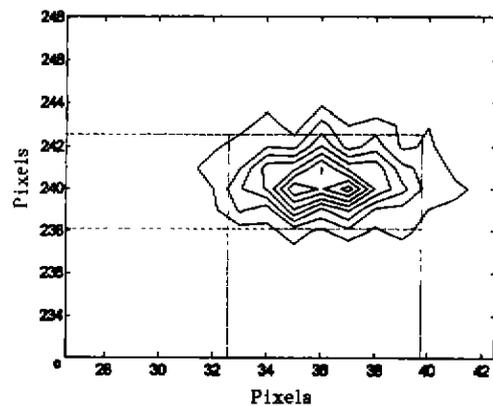


图7 帧差图像中目标的等高线图

Fig. 7 Contour line of target in difference image

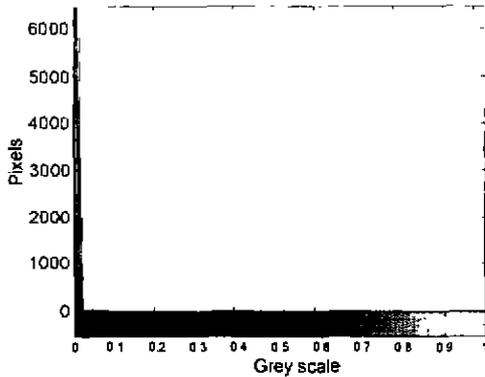


图8 含目标帧差图像的统计直方图
Fig. 8 Histogram of difference image containing targets

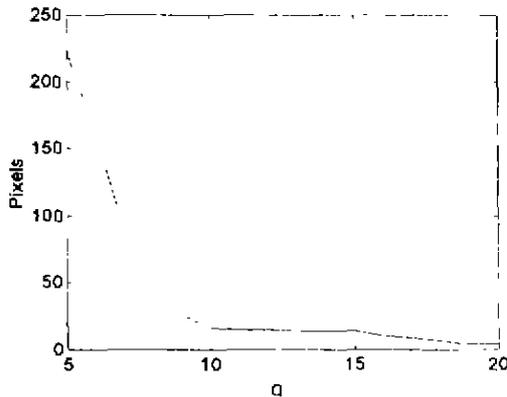


图9 α 与目标像素数的关系曲线
Fig. 9 α vs pixels of target



图10 目标的轮廓
Fig. 10 Binary mask of target

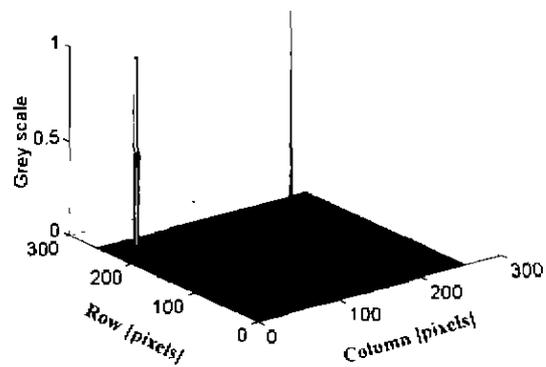


图11 目标轮廓的三维示意图
Fig. 11 3-D sketch of binary mask of target

的二值化处理,用 $V_{th}=0.052$ 分割图4得到目标轮廓图像如图10和图11所示。

4 结论

实验结果表明,文中提出的检测算法能够满足实时检测运动小目标的要求,在背景缓动或处于相对静止时,能够很好的确定出目标的位置并较精确的分割出目标轮廓。

REFERENCES

- [1] Holger Lange. Real time motion detection for target acquisition "on the move" based on a nonlinear filter using short time and medium time image difference, *SPIE Conference on Real-Time Imaging IV*, 1999, **3645**: 98—109
- [2] ZHU Hong, ZHAO Yi-Gong. Detection of weak and small moving infrared targets by adaptive prediction of background, *J. Infrared Millim. Waves* (朱红,赵亦工,基于背景自适应预测的红外弱小运动目标检测,红外与毫米波学报), 1999, **18**(4): 306—309
- [3] Shawn S Toumodge. Detection on tracking of small targets in persistence, In: *Proceedings of SPIE Signal and Data Processing of Small Targets*, 1991, **1481**: 221—228
- [4] Hannelore G Hausen, Cyrus Elyashar. Adaptive threshold adjustment and control, In: *Proceedings of SPIE, Signal and Data Processing of Small Targets*, 1989, **1096**: 44—54
- [5] Dr Paul Frank Singer, Doreen M Sasaki. A performance model for unresolved target detection using multispectral infrared data, In: *Proceedings of SPIE, Signal and Data Processing of Small Targets*, 1998, **3373**: 16—23