

文章编号: 1672-8785(2022)06-0020-07

一种基于角点检测的实时自动 对焦方法及实现

刘纪洲 官 宸 崔跃刚 王 超

(华北光电技术研究所, 北京 100015)

摘 要: 为了满足热像仪在不同场景下自动对焦的功能需求, 对现有多种自动对焦实现方案的优劣势进行了评估分析, 并选择了一种基于角点检测算法的对焦方案。通过检测图像中的角点数量来实时判断图像的清晰度。根据热像仪 FPGA 图像处理系统的性能对该算法进行了改进与优化, 并提出了多阈值角点检测实时对焦方法。根据当前场景的对焦状态, 热像仪可自主判断是否执行对焦操作, 以适应不同检测场景的需求, 最终实现其在不同场景下的实时自动对焦功能。

关键词: 实时自动对焦; 评价函数; 角点检测

中图分类号: TB4 **文献标志码:** A **DOI:** 10.3969/j.issn.1672-8785.2022.06.003

A Real-time Autofocus Method and Implementation Based on Corner Detection

LIU Ji-zhou, GONG Chen, CUI Yue-gang, WANG Chao

(North China Research Institute of Electro-Optics, Beijing 100015, China)

Abstract: In order to meet the functional requirements for autofocusing in different scenarios, the advantages and disadvantages of various existing autofocus implementation schemes are evaluated and analyzed. A focusing scheme based on corner detection algorithm is proposed. The sharpness of the image can be judged by detecting the number of corners in the image. According to the performance of the FPGA image processing system of the thermal imager, the algorithm is improved and optimized. A real-time focusing method for multi-threshold corner detection is proposed. According to the focusing state of the current scene, the thermal imager can independently determine whether to perform the focusing operation to meet the needs of different detection scenarios, and finally realize the real-time automatic focusing function of the thermal imager in different scenarios.

Key words: real-time autofocus; evaluation function; corner detection

收稿日期: 2022-02-16

作者简介: 刘纪洲(1985-), 男, 北京人, 高级工程师, 主要从事制冷型红外整机研究。

E-mail: liujizhou307@126.com

0 引言

根据光学理论, 物体成像时在像方会有一个最佳像面位置, 即在这个位置上, 图像最清晰。若偏离此位置, 图像则会模糊, 进入失焦状态^[1]。为获得清晰图像, 目前市场上的制冷型红外热像仪通常都具备自动对焦功能。当用户观察到图像明显模糊时, 可以通过上位机串口向热像仪发送自动对焦指令, 控制调焦电机驱动调焦组件前后移动, 将调焦组件推到最清晰成像的位置, 从而完成对焦过程并获得清晰的图像^[2]。

然而, 这种自动对焦方案需要用户去判断当前场景的清晰度, 再决定是否手动执行自动对焦操作。而热像仪无法自主判断对焦需求, 不适用于观测场景快速变化的使用情景(如机载探测、转台搜索、无人值守等)。为了满足热像仪在不同场景下的使用需求, 图像处理系统需自主进行场景清晰度判断并执行对焦操作, 所以开发一套基于热像仪自主判断的实时自动对焦算法很有必要。本文对比分析了现有的几种清晰度评价算子, 并以此为依据选用 Harris 角点检测算子进行了图像清晰度评价。针对实际应用中遇到的问题, 提出了一种基于区域检测的多阈值角点检测算法, 并根据 FPGA 的流水线架构特性进行了适当的优化, 以满足工程化实现的需要。

1 实现方案

1.1 原有实现方案

对于某一特定的成像系统来说, 图像的清晰度反映了系统的离焦程度, 能够直观地体现人的主观感受。当图像比较清晰时, 图像内容细节丰富, 在空域中表现为相邻像素的特征值变化较大, 在频域中表现为频谱的高频分量多^[3-5]。因此, 清晰度可以直接体现图像的模糊程度。

理想的清晰度评价函数应具有无偏性、单峰性、灵敏性、稳定性和实时性^[2]。图像清晰度评价算法从工作原理上可以分为频域函数评价、灰度函数评价和统计学函数评价

等几类^[6]。目前常用的基于灰度函数的清晰度评价算法有基于 Laplace 算子^[7]、Sobel 算子、Prewitt 算子、平均梯度算子^[8]、Brenner 算子^[9]、SMD 算子^[10]的清晰度评价算法^[11]等。这些算子利用梯度函数提取边缘信息, 在软硬件方面易于实现。

基于某小型热像仪采集了一段模糊-清晰-模糊的完整对焦过程图像。然后对用不同算子函数算得的清晰度评价进行了对比分析, 并以此为依据开展了自动对焦方案的可行性评估。

图 1 所示分别为 Laplace 算子、Sobel 算子、Prewitt 算子、平均梯度算子、Roberts 算子、Brenner 算子、SMD 算子在整个调焦过程中的清晰度评价曲线。从图中可以看出, 上述基于灰度函数的清晰度评价函数值随着调焦位置的变化具有基本相似的变化趋势。评价函数值可以反映图像的清晰度。随着评价函数值的增大, 调焦逐渐趋于清晰位置。然而, 在分辨清晰图像(图 1(b))和次清晰图像(图 1(c))时, 评价函数值几乎接近, 无法区分。

灰度函数评价方法利用固定的空间模板在图像上滑动卷积, 求出评价函数的累加和。此类方法在本质上只是累加了局部的灰度对比度, 这会导致以下几个问题: (1)无法通过某预设清晰度评价阈值来准确区别清晰图像和次清晰图像, 因为它们具有相近的评价值; (2)对于复杂场景和简单场景, 很难通过预设评价函数阈值方式判断是否应该启动对焦功能; (3)容易受到图像噪声干扰。

经过大量测试总结, 基于灰度函数的评价方法仅适用于固定场景的主动对焦过程(通过比对连续变化的评价进行对焦判断), 而无法适用于不同场景单帧图像的清晰度判断。因此, 我们期望找到一个评价标准, 在不受场景复杂度影响的前提下, 能够较好地分辨出清晰和次清晰的图像。为了解决此问题, 本文重新选择了基于图像统计信息的评价方法, 通过特定的评价模板统计图像中清晰的角点数量, 以此为依据来评价图像的清晰度。

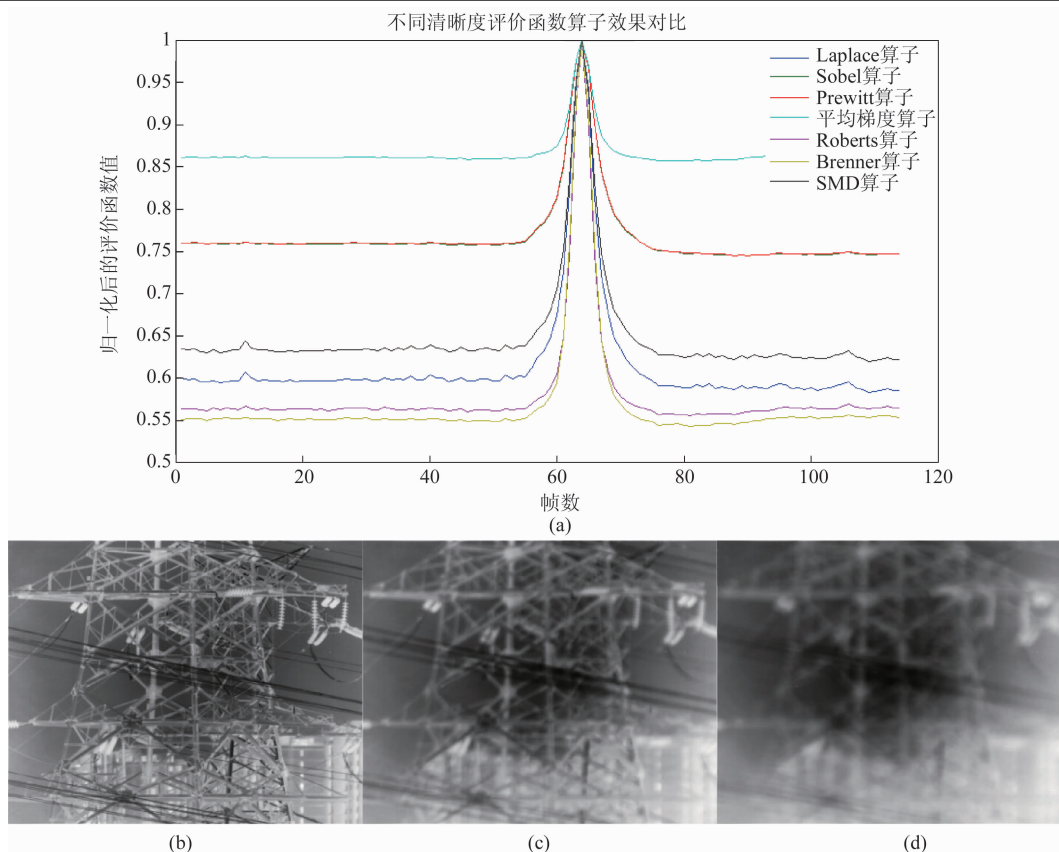


图1 (a)完整对焦过程中不同算子评价值变化的示意图；(b)第64帧清晰图像；(c)第70帧次清晰图像；(d)第86帧模糊图像

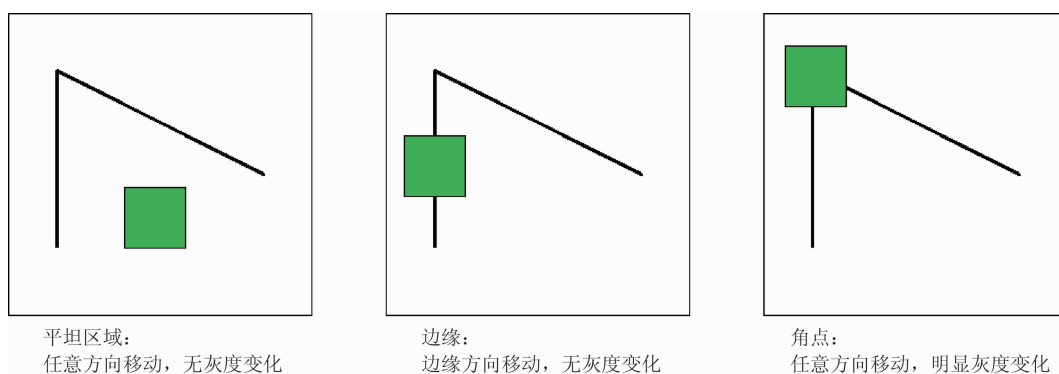


图2 角点检测原理的示意图

1.2 基于角点检测的自动对焦算法

考虑到热像仪的软硬件实现平台, 本文选用 Harris 角点检测算子进行图像清晰度评价, 并根据 FPGA 的流水线架构特性作了适当的优化。首先介绍 Harris 角点检测算法的原理。它是一种基于图像特征的角点检测方法, 其基本思想(见图 2)如下: 通过一个局部检测窗口, 沿各个方向作微小移动, 同时判断窗口内的平均能量变化; 当该能量变化超过阈值时, 就将

窗口的中心像素点提取为角点^[13]。通过考察当前图像中的角点数量和变化趋势来判断此图像是否清晰。

首先计算图像 $I(x, y)$ 在 x 和 y 方向上的梯度 I_x 和 I_y :

$$I_x = \frac{\partial I}{\partial x} = I \times [-1 \ 0 \ 1]$$

$$I_y = \frac{\partial I}{\partial y} = I \times [-1 \ 0 \ 1]^T \quad (1)$$

考量图像中心像素点在各个方向上的变化, 计算图像在三个方向上的梯度乘积并将其作为因子:

$$\begin{aligned} I_x^2 &= I_x \cdot I_x \\ I_y^2 &= I_y \cdot I_y \\ I_{xy} &= I_x \cdot I_y \end{aligned} \quad (2)$$

为了考虑局部范围内的梯度变化特性和减少图像噪声的干扰, 使用高斯函数对窗口 w 内以 (x,y) 为中心的所有像素的梯度信息进行高斯加权:

$$\begin{aligned} A &= \sum_{(x,y) \in W} g(I_x^2) = \sum_{(x,y) \in W} I_x^2 \cdot \omega(x,y) \\ B &= \sum_{(x,y) \in W} g(I_y^2) = \sum_{(x,y) \in W} I_y^2 \cdot \omega(x,y) \\ C &= \sum_{(x,y) \in W} g(I_x I_y) = \sum_{(x,y) \in W} I_x I_y \cdot \omega(x,y) \end{aligned} \quad (3)$$

以 A 、 B 、 C 为基础, 构建以像素点 (x,y) 为中心的特征值矩阵 M :

$$M = \begin{bmatrix} A & C \\ C & B \end{bmatrix} = \sum_{(x,y) \in W} \omega(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (4)$$

利用该矩阵就可求得像素点 (x,y) 处的 Harris 响应值 R :

$$R = \det(M) - k(\text{trace}(M))^2 \quad (5)$$

如图 3 所示, R 取决于 M 的特征值。当角点 R 为正且数值较大时, 平坦区域 $|R|$ 的很小, 边缘的 R 为负值。

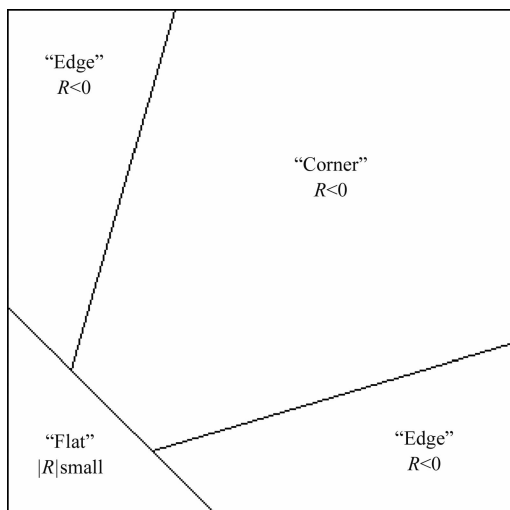


图 3 Harris 响应值 R 的数学意义

过滤大于某一阈值 t 的 R 值:

$$R = \{R: \det(M) - k(\text{trace}(M))^2 > t\} \quad (6)$$

最后在窗口内进行非最大值抑制, 其局部最大值点即为图像中的角点^[14]。

基于上面的对焦过程, 利用 Harris 角点检测算法对图像的清晰度进行评价, 得到对焦过程中角点数量的变化图(见图 4)。

从图 4 中可以看出, 角点检测方法具有更好的灵敏性。在整个对焦过程中, 当图像趋于清晰时, 角点数量明显较多; 而图像略微模糊后, 角点数量迅速减少。针对第 64 帧清晰图像和第 70 帧次清晰图像, 根据交点数量很容易加以区分。

然而, 在实际使用调试的过程中发现, 对于单一固定阈值的角点检测判断标准, 面对对比度较弱的简单场景(如室内、天空等)时, 由于图像灰度较平坦, 在各个像素点求出的角点数量都较小且小于预设阈值。这会导致一个问题, 即简单的清晰场景与复杂背景下的模糊场景会有相近的角点数量。因此, 通过预设单一角点阈值进行判断, 依然无法准确地区分单一场景清晰图像和复杂场景模糊图像。阈值较小时会造成复杂场景下图像模糊, 没有适时对焦; 若阈值较大, 面对简单场景时则会频繁进行对焦操作, 影响热像仪的使用。

2 测试与优化

为了解决这个难题, 本文提出了一种基于区域检测的多阈值角点检测算法。在图像中等间距等大小地选取 25 个子区域, 分别对其进行角点数量检测。当存在一定数量的角点时, 判断此区域存在场景内容, 并标记此区域。最后比对标记区域的总个数与设定阈值 t_1 。若标记区域个数大于 t_1 , 此场景即为复杂场景, 反之则为简单场景。

然后累加所有子区域的角点总数量 t , 用于判断当前场景是否执行自动对焦功能, 并分别将其与阈值 t_c 、 t_s 进行比较。在简单场景下, 若 $t > t_c$, 不执行对焦功能; 在若 $t < t_c$, 执行对焦功能。在复杂场景下, 若 $t > t_s$, 不执行对焦功能, 若 $t < t_s$, 执行对焦功能。

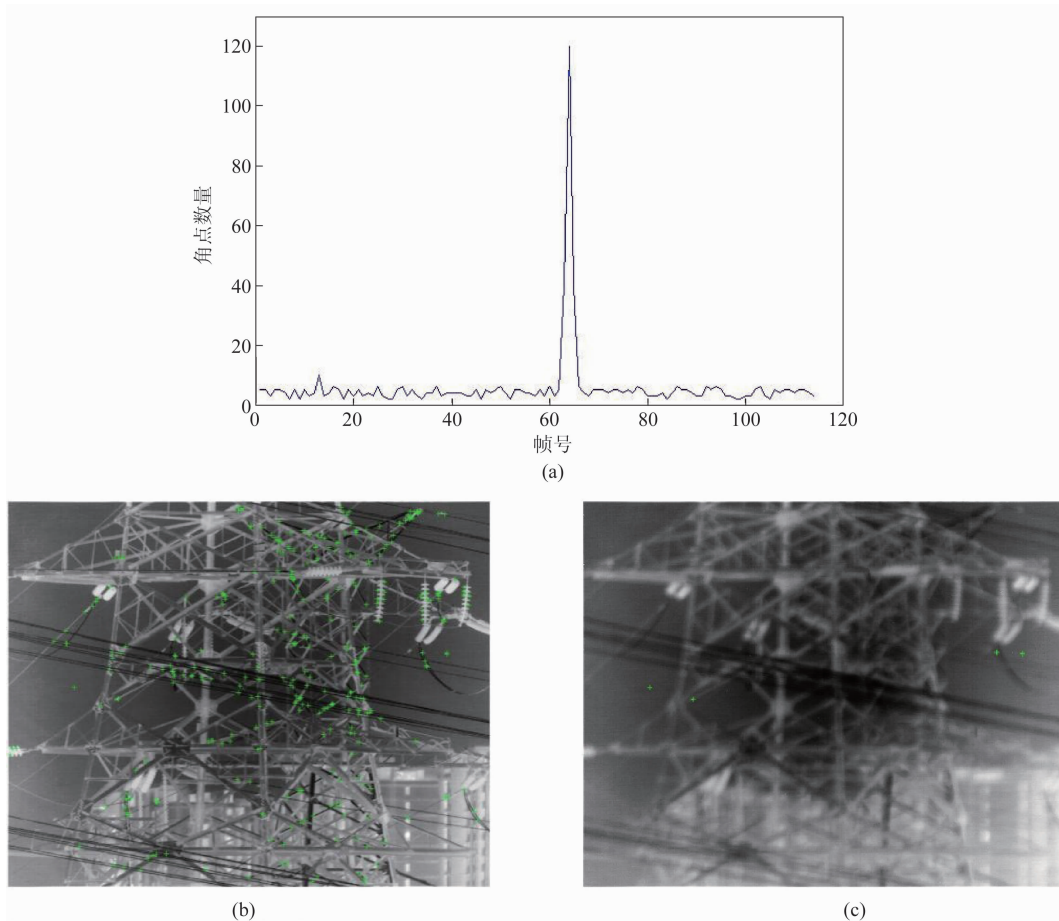


图 4 (a)完整对焦过程中的角点变化示意图；(b)第 64 帧清晰图像；(c)第 70 帧次清晰图像

图 5(a)和图 5(b)所示分别为复杂场景和简单场景。在图像中选取 25 个子区域，采集一段完整的对焦过程视频，获得对焦过程中图片的交点数量(见图 5(c)和图 5(d))。对子区域窗口内的角点数量进行判断：图 5(a)中有 20 个窗口存在场景，图 5(b)中有 9 个窗口存在场景。通过将其与阈值 t_1 进行比对，判定图 5(a)为复杂场景，图 5(b)为简单场景。然后分别统计子区域窗口下总的角点数量。通过将其与 t_c 和 t_s 进行比对，决定是否对当前场景执行对焦功能。

在实际使用中，为避免热像仪频繁进行自动对焦，设计了运动保护功能，以满足工程化应用需求：

(1)根据测试数据，热像仪在场景切换中会出现运动模糊的情况，角点数量会迅速减少且波动剧烈。通过分析角点变化情况可知图像的运动状态。当热像仪处于运动状态时，不

行对焦操作。

(2)热像仪切换场景后，当画面稳定时，若检测的角点数量大于阈值，则说明切换后场景图像是清晰的，此时依然不进行对焦操作；若切换场景后角点数量小于阈值，则说明此时图像模糊，会触发自动对焦指令。

通过这种设计可以有效防止运动过程中误触发对焦指令而导致对焦异常，并可避免对清晰场景进行重复的对焦操作。

图 6 为切换场景后热像仪通过自主执行自动对焦操作而得到的角点数量变化图。可以看出，第 80 帧时热像仪发生运动，角点数量迅速减少且剧烈变化，说明此时热像仪处于运动状态，暂不进行对焦操作。

第 140 帧时角点数量稳定下来，且稳定后角点数量减少，说明切换场景后热像仪进入了失焦状态，应触发自动对焦操作。此时，图像处理系统自主判断当前场景下应进行对焦操作，

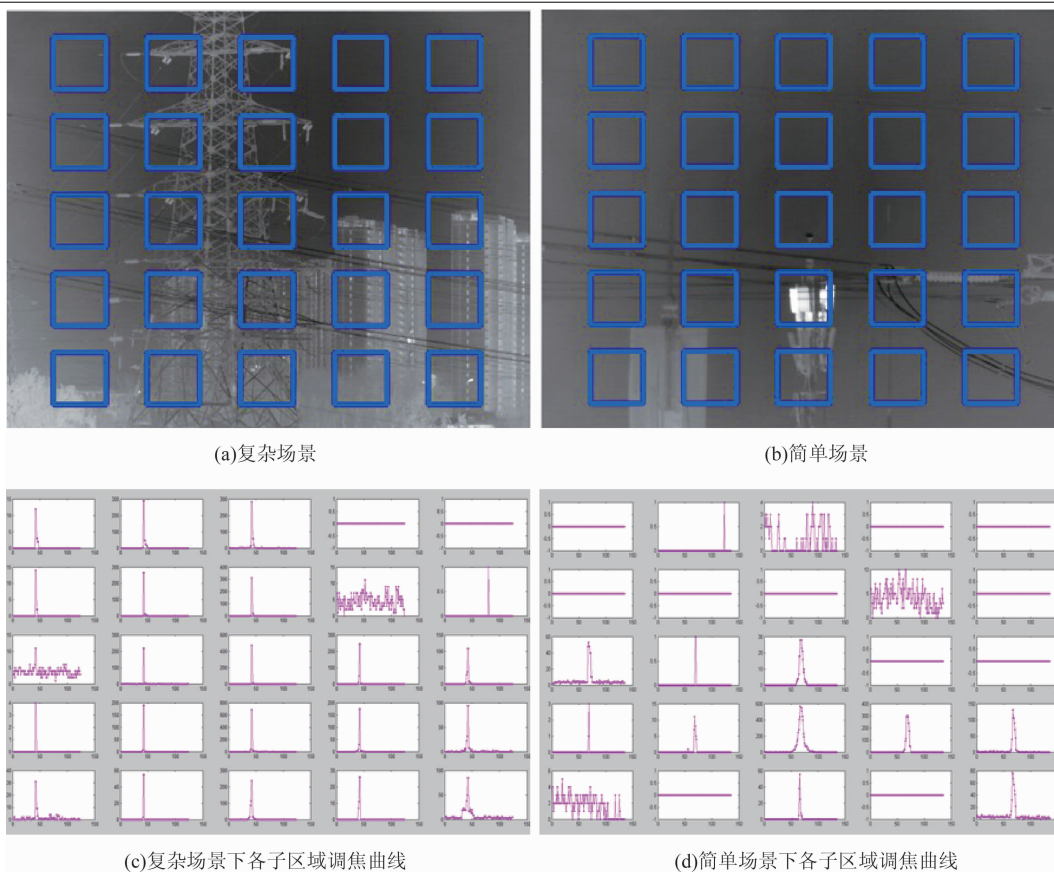


图 5 完整对焦过程中的角点变化示意图

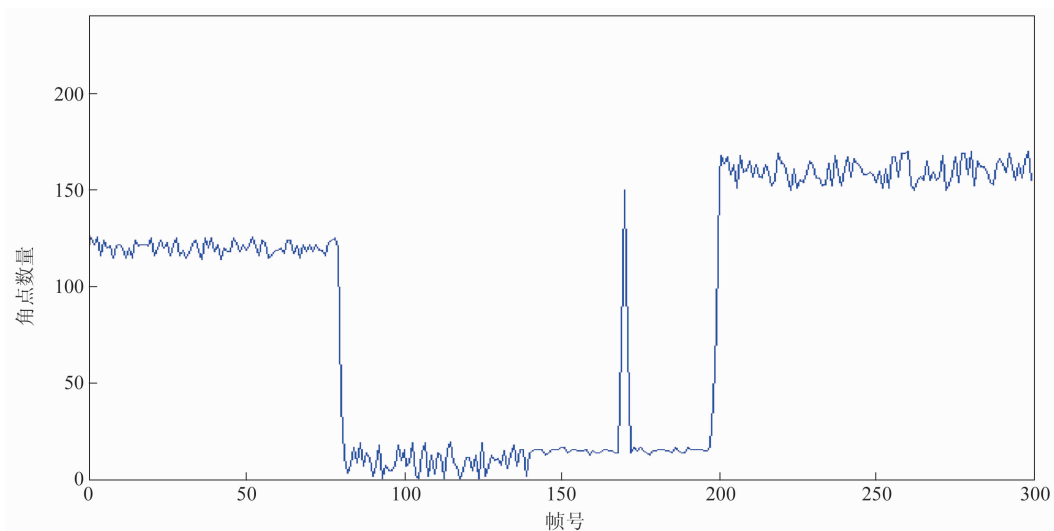


图 6 切换场景后得到的角点数量变化图

开始执行自动对焦功能, 控制调焦电机驱动调焦组件前后移动, 对清晰度评价函数进行爬坡计算。爬坡过程中, 第 170 帧时图像角点数量变多, 说明此时已是清晰位置。但由于要完成爬坡计算, 这时调焦组件依然在运动。在第

200 帧时电机移动完后, 通过爬坡算法找到了最佳成像位置, 回到了 170 帧时的对焦状态, 完成了本次自动对焦功能, 到达了最清晰位置。若热像仪或场景发生移动, 则再次重复该过程。

3 结束语

通过对现有多种自动对焦实现方案进行评估和对比,提出了一种基于 Harris 角点检测的实时自动对焦方法,并将其应用于小型化热像仪系统中。该方法能够以检测的角点数量为评价依据,正确地评价图像的清晰度。实际使用中,它在面对各种复杂的观察情景(如无云的天空、简单的室内场景、弱对比度的低温环境等)时能正确地触发自动对焦指令。同时,该算法具备运动保护功能。当检测到图像在运动时,进入对焦等待状态;只有待图像稳定后才再次执行自动对焦操作。测试结果表明,这种方法能够基本满足大部分使用情景的需要。下一步还需解决热像仪对有前后景深的场景进行成像时的误对焦问题,旨在利用场景直方图的判断来实现此类场景的准确对焦。

参考文献

- [1] 王焯茹. 基于数字图像处理的自动对焦方法研究 [D]. 杭州: 浙江大学, 2018.
- [2] 李宝. 基于 FPGA 的数字自动对焦算法的研究与实现 [D]. 哈尔滨: 哈尔滨工业大学, 2008.
- [3] 李媛媛. 数字 CMOS 自动对焦系统研究 [D]. 杭州: 浙江大学, 2007.
- [4] 田宜彬. 数字成像系统的自动对焦算法研究 [D]. 杭州: 浙江大学, 2002.
- [5] 蒋凤林. 基于数字图像处理的自动调焦算法研究 [D]. 哈尔滨: 哈尔滨理工大学, 2008.
- [6] 郑玉珍. 自动对焦中的优化爬山搜索算法 [J]. *浙江科技学院学报*, 2005, **17**(3): 4.
- [7] Krotkov E P. *Active Computer Vision by Cooperative Focus and Stereo* [M]. Berlin: Springer-Verlag, 1989.
- [8] Schlag J F, Sanderson A C, Neuman C P, et al. Implementation of Automatic Focusing Algorithms for a Computer Vision System with Camera Control [R]. Carnegie Mellon University: CMU-RI-TR-83-14, 1983.
- [9] Subbarao M, Tyan J K. Selection the Optimal Focus Measure for Auto-focusing and Depth from Focus [J]. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998, **20**(8): 864-870.
- [10] Jarvis R A. Focus Optimization Criteria for Computer Image Processing [J]. *Microscope*, 1976, **24**(2): 163-180.
- [11] 李祚林, 李晓辉, 马灵玲, 等. 面向无参考图像的清晰度评价方法研究 [J]. *遥感技术与应用*, 2011, **26**(2): 239-245.
- [12] 高一宁, 韩燮. 基于 Canny 算子和 Harris 角点提取的三维重建方法 [J]. *电子测试*, 2011, **18**(2): 6.
- [13] 王崴, 唐一平, 任娟莉, 等. 一种改进的 Harris 角点提取算法 [J]. *光学精密工程*, 2008, **16**(10): 7.