

面向嵌入式机载场景的快速图像滤波方法

李鹏亮¹ 李成文¹ 杨子怡¹ 谭琪琳¹
LI Pengliang LI Chengwen YANG Ziyi TAN Qilin

摘要

在功耗、体积及质量的约束下, 针对嵌入式机载计算设备很难满足较大半径实时滤波处理的需求, 提出一种面向嵌入式机载应用场景的快速图像滤波方法。首先, 以每个待滤波像素为中心, 以设定半径作为滤波半径, 确定待处理图像中每个待滤波像素的滤波区域; 其次, 依据每个滤波区域的梯度信息和半径, 自适应确定每个滤波区域的抽样步长; 然后, 采用所提出的抽样步长, 对每个滤波区域进行抽样得到多个抽样像素; 最后, 通过抽样像素, 对每个滤波区域进行滤波处理输出。通过基于国产 GPU 的嵌入式平台验证, 证明所提出的方法解决了现有的滤波处理方法随着滤波区域增大而导致计算耗时指数增加的问题, 大幅度提高了图像滤波处理效率, 同时达到了优秀的滤波性能表现。

关键词

嵌入式计算; 机载图像处理; 快图像滤波; 自适应步长; 像素抽样

doi: 10.3969/j.issn.1672-9528.2024.05.018

0 引言

在机载嵌入式场景下, 由于 SWaP (Size, Weight, Power and Cost) 约束, 对芯片功耗和散热有严格要求, 导致嵌入式设备计算能力相对较低^[1]。滤波算法作为基本的机载图像处理方法, 被广泛应用于机载场景的各种图像处理任务中。嵌入式机载计算平台的计算能力较弱和部署算法实时性要求非常高之间存在的矛盾, 对算法开发者的能力提出了更高的要求。

图像滤波方法被广泛应用于大量计算机视觉和图像处理任务当中, 例如引导滤波 (guided image filtering, GIF) 方法^[2]、基于迭代反投影的机载超分辨率方法^[3]和基于暗通道的机载图像去雾霾方法^[4]等。其中, 引导滤波方法借助 boxfilter 滤波实现了重复区域像素的平均化, 基于迭代反投影的机载超分辨率方法通过高斯滤波实现分辨率退化, 基于暗通道的图像去雾霾方法借助最小值滤波获取暗通道信息。面向机载嵌入式图像处理应用场景, 同样需要大量的图像滤波方法, 实现机载平台的图像去雾霾^[5]、超分辨率^[6]、基于滤波的多传感器数据融合^[7]等。

目前, CPU、GPU、FPGA 和 DSP 等作为主流的嵌入式平台信号处理器, 其在执行图像滤波任务时, 通常需要嵌套循环, 对每一个中心像素的周边 (二维、三维或多维空间中) 像素索引。目前, 嵌入式 ARM 平台 CPU 通常可以借助 OpenMP 多核并行加速^[8]、NEON 矢量加速等技术来提高运

算效率^[9]。国产 GPU 通常具有众核架构, 通过开放式计算语言 OpenCL 可实现众核并行计算^[10], 提高计算效率。由于 GPU 主频较低, 通常处理小规模并行计算任务不具备优势。DSP 芯片通常主频高, 计算核心相比 GPU 较少, 可借助多核并行以及向量加速技术实现加速任务处理^[11]。FPGA 具有众多计算核心和存储单元^[12], 片内具备众多并行计算资源, 可实现大规模并行计算。

虽然 CPU、GPU、FPGA 和 DSP 等主流计算芯片具有非常高效的计算机制, 但随着滤波区域的增大, 所需要的嵌套循环次数将呈指数增大, 进而大幅度降低了图像处理效率, 给嵌入式场景下的实时图像滤波处理需求带来巨大挑战。

面向嵌入式机载场景的快速图像滤波需求, 本文提出一种基于自适应抽样滤波的快速滤波方法, 可以解决现有的滤波处理方法随着滤波区域增大而导致计算开销指数增加的问题, 大幅提高了嵌入机载场景图像滤波效率, 并且通过国产嵌入式计算平台验证了提出方法的可靠性和高效性, 本文工作的主要创新点如下。

(1) 面向嵌入式机载滤波场景, 提出一种基于抽样像素滤波的加速机制。对于二维图像滤波, 计算开销理论上可降低为抽样步的二次方分之一。

(2) 面向嵌入式机载滤波场景, 提出一种自适应抽样步长。对平滑区域增加步长, 通过较少像素的滤波输出代替原有滤波的输出; 对于纹理区域采用较小抽样步长, 通过较多的抽样像素滤波结果替代原有滤波结果, 该机制进一步提高滤波效率。

(3) 面向嵌入式机载滤波场景, 通过国产 GPU 执行了 boxfilter 滤波方法以及 GIF 方法的加速验证, 分别实现了 1280×720 分辨率大小的图像约 301 帧/s 和 77 帧/s 的计算速度。

1 理论知识

目前, 常用滤波方法可大致分为两类: 基于模板的滤波方法, 如高斯滤波、均值滤波等; 基于非模板的滤波方法, 如中值滤波、极大/小值滤波等。这些常用滤波方法不仅在传统计算机视觉和图像处理中得到广泛应用, 在卷积神经网络 (convolutional neural network, CNN) 中也作为常用算子出现^[13], 如卷积算子属于基于模板的滤波、池化算子属于基于非模板的滤波。

1.1 基于模板的滤波方法

传统的基于模板的滤波方法通过滤波模板的值和待滤波区域的像素值来计算滤波区域, 以二维滤波为例, 计算方式可表达为:

$$f(i, j) = \frac{1}{H^2} \sum_{a=-h}^h \sum_{b=-h}^h w(a, b) x(i+a, j+b) \quad (1)$$

式中: $x(i, j)$ 表示待滤波图像的像素索引, $w(a, b)$ 为滤波系数, $w(a, b)$ 的具体数值由滤波方法类型, 如高斯滤波、boxfilter 滤波等决定。a 和 b 分别为滤波区域水平方向和垂直方向的索引, H 表示滤波区域直径, h 为滤波半径。

由公式 (1) 可知, 随着滤波区域增大, a 和 b 分别需要索引的像素范围也越来越大, 更新 $f(i, j)$ 所需要的计算开销就会变大, 导致实时性变差。

1.2 基于非模板的滤波方法

传统的非模板的滤波方法不依赖输入的引导, 仅依靠滤波区域像素值去实现滤波输出, 如中值滤波、极大/小值滤波等。对于半径为 h 的滤波区域, 以二维滤波为例, 其滤波输入表示为:

$$f_{in} = [x(i-h, j-h), x(i-h-1, j-h), \dots, 0, \dots, x(i+h, j+h-1), x(i+h, j+h)] \quad (2)$$

式中: f_{in} 中数据个数为 $(2h+1)^2$, 通常需要对 f_{in} 中所有数据排序, 根据滤波需求输出需要的结果。由公式 (2) 可知, 随着滤波半径的增加, 参与排序的数据量增大, 导致计算实时性变差。

2 快速图像滤波方法建模

机载视觉传感器工作时处于高速移动状态, 采集到的每一帧图像在执行图像处理、目标跟踪以及检测等算法时, 如果涉及的滤波算法在机载嵌入式平台运行太慢, 则会影响其他视觉算法的执行, 轻则出现严重滞后现象, 重则延误战机。

对于基于模板的滤波方法和基于非模板的滤波方法, 本文均给出了加速机制, 具体实施如下文。

2.1 基于模板的滤波方法加速

首先, 获取待滤波区域的像素值, 然后确定初始抽样步长 s, 假设待滤波像素坐标为 (i, j), 滤波半径为 h 时, 滤波区域的抽样像素可表示为:

$$f_{sin} = [\dots, x(i-s, j), x(i-s, j-s), x(i, j), \dots, x(i+s, j), x(i+s, j+s), \dots] \quad (3)$$

式中: f_{sin} 为抽样像素集, s 小于 h。由于 s 可能不能被 h 整除, 如果第 n-1 次抽样在滤波区域内, 而第 n 次抽样超出滤波区域, 则舍弃第 n 次抽样值, 仅使用前 n-1 的抽样值。由于 f_{sin} 数倍小于 f_{in} , 所以可通过大小为 n-1 的滤波模板实现快速滤波, 其数学表达式为:

$$f(i, j) = \frac{1}{(2n-1)^2} \sum_{a=-n+1}^{n-1} \sum_{b=-n+1}^{n-1} w(a, b) x(i+sa, j+sb) \quad (4)$$

式中: $w(a, b)$ 会随着抽样步长的变化而变化。

2.2 基于非模板的滤波方法加速

对于基于非模板的滤波方法, 假设待滤波像素坐标为 (i, j), 滤波半径为 h 时, 滤波区域的抽样像素集可通过公式 (3) 定义。在抽样步长确定的情况下, 非模板的快速滤波方法可表示为:

$$f = G_{nf}(f_{sin}) \quad (5)$$

式中: f 为滤波输出, $G_{nf}(\bullet)$ 为非模板的滤波算子。由于基于抽样机制, 参与 $G_{nf}(\bullet)$ 运算的数据量变小, 因此执行效率变高。

2.3 自适应抽样步长

固定步长抽样方式可视为相同尺度的滤波, 当抽样步长为 1 时, 等价于原分辨率尺度滤波; 当抽样步长大于 1 时, 等价于低分辨率较小尺度滤波。通过以上分析证明固定抽样方式的有效性。

由于图像中平滑区域像素的相似度较高, 纹理区域像素的差异较大。为了更进一步提升滤波效率, 本文提出一种可变步长的抽样方法, 如图 1 所示, 左侧 (路面) 矩形标注区域内容相似度较高, 可采用较大抽样步长, 右侧矩形 (树木+路面) 标注区域内的纹理丰富并且差异较大, 可采用较小抽样步长。由此可知, 本文提出的基于可变步长的抽样滤波方法等价于多尺度滤波方法。

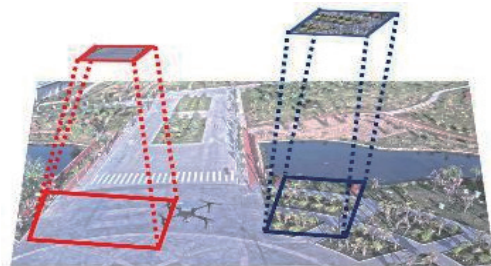


图 1 自适应抽样步长示例

计算待滤波像素 $x(i, j)$ 在水平方向和垂直方向的梯度信息 $(\nabla_h x(i, j), \nabla_v x(i, j))$ ，计算方式为：

$$\begin{cases} \nabla_h x(i, j) = x(i, j) - x(i-1, j) \\ \nabla_v x(i, j) = x(i, j) - x(i, j-1) \end{cases} \quad (6)$$

式中： $\nabla_h x(i, j)$ 和 $\nabla_v x(i, j)$ 分别表示水平方向和垂直方向的梯度值。

如果梯度大于设定的阈值，则待滤波像素 $x(i, j)$ 属于纹理区域；否则，待滤波像素 $x(i, j)$ 属于背景平滑区域（即非纹理的平缓区域）。不同区域的抽样步长确定方式为：

$$\begin{cases} s_e = \text{int}((2h+1)/5+0.5) \\ s_s = \text{int}((2h+1)/3+0.5) \end{cases} \quad (7)$$

式中： s_e 和 s_s 分别属于纹理区域和平滑区域的抽样步长， $\text{int}(\cdot)$ 表示四舍五入操作， h 为滤波半径。

由于公式（4）中的 $w(a, b)$ 和抽样步长强相关，因此，对于基于模板的滤波方法需要两个模板，分别用于纹理区域和平滑区域。由于 boxfilter 滤波对应的 $w(a, b)$ 中的每个像素的值都相同，因此，可变抽样机制不影响 boxfilter 滤波方法计算复杂度。对于基于非模板的滤波方法，无须考虑外部滤波输入，因此可变抽样机制会进一步降低计算复杂度。

在固定步长的抽样滤波模式下，对待处理图像中每个待滤波像素的滤波区域进行均匀抽样，采用每个滤波区域中的抽样像素执行滤波处理，例如传统方法对 31×31 的像素区域采用 boxfilter 滤波算法进行滤波处理时，需要在水平和垂直方向分别完成 31 次像素索引。如果将抽样步长设置为 5 或 3，仅在水平和垂直方向分别索引 6 次或 10 次，将极大提高了计算效率。由于本文提出自适应的抽样步长进行像素抽样，因此在进行索引和计算时，需要的开销理论上不大于原来的 49/961 或 121/961，且通过实验加速对比可知，本文提出的方法不仅具有很高的执行效率，而且达到了优秀的滤波性能表现。

3 实验验证与分析

本文提出的快速滤波方法在国产 NA16D3C7C1 GPU 加速卡上进行了验证，NA16D3C7C1 GPU 加速卡与飞腾 D2000 开发板通过 PCIe 总线互连，其硬件验证平台如图 2 所示。

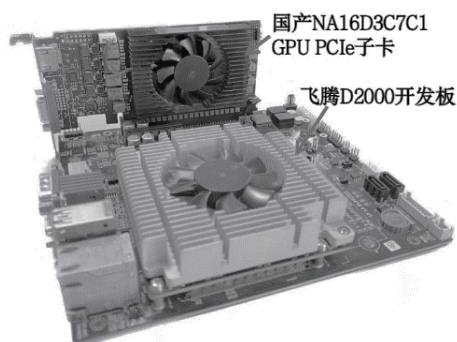


图 2 硬件验证平台

该款加速卡主频为 1.25 GHz，支持 OpenCL3.0，典型功耗为 5 ~ 15 W，加速卡内集成 1280 个 UCC（Unified Compute Core），可提供约 1.5TFLOPS 的 Float32 算力。

对于嵌入式机载场景来说，滤波方法的运行时间和滤波效果同样重要，通过传统滤波方法和本文提出的快速抽样滤波方法进行对比来证明提出方法的有效性。

本实验选择使用 guided image filter（GIF）方法^[14]作为验证方法，大量的图像处理应用使用 GIF 方法，如图像去雾霾、暗光增强、去噪及多源图像融合等处理任务^[15-16]。GIF 方法在自引导滤波方式下需要 4 次 boxfilter 运算，占据了 GIF 方法的主要计算开销。本实验将提出的快速滤波方法应用于 GIF 方法中 boxfilter 的加速，通过对比本文提出的快速滤波方法对 boxfilter 和 GIF 方法的加速计算效果来证明本文提出的方法适用机载嵌入式平台。

本实验选择四组图像分辨率大小分别为 1280×720 、 1920×1080 、 2560×1440 和 4096×3112 的图像，对 boxfilter 方法和 GIF 方法分别加速验证。

表 1 给出了嵌入式环境下国产 GPU 对 boxfilter 方法的加速时间，表 2 给出了嵌入式环境下国产 GPU 对 GIF 方法的加速时间。

表 1 嵌入式环境下国产 GPU 对 boxfilter 方法的加速时间

测试图像分辨率	boxfilter 执行时间 /ms	提出方法加速 boxfilter 执行时间 /ms	时间加速比
1280×720	56.875	3.321	17.125
1920×1080	126.731	6.859	18.476
2560×1440	224.625	11.924	18.838
4096×3112	776.431	40.129	19.337

表 2 嵌入式环境下国产 GPU 对 GIF 方法的加速时间

测试图像分辨率	GIF 方法执行时间 /ms	提出方法加速 GIF 执行时间 /ms	时间加速比
1280×720	198.956	12.975	15.333
1920×1080	444.217	27.756	16.004
2560×1440	816.393	48.4928	16.835
4096×3112	2709.422	164.247	16.496

由表 1 和表 2 可知：（1）本文提出的加速滤波方法对 boxfilter 起到约 18 倍的加速效果，对 GIF 滤波起到约 16 倍的加速效果；（2）由于 GIF 中包含 4 个 boxfilter 运算，在编写程序时可将部分 boxfilter 运算合在一个 GPU 执行函数中减少计算开销，因此表 1 和表 2 不严格满足计算时间开销大于 4 倍的关系；（3）在分辨率为 1280×720 的情况下，本文提

出的加速滤波方法加速 boxfilter 实现约 301 帧/s 的计算速度, 加速 GIF 实现约 77 帧/s 的计算速度。

图 3 给出了 GIF 方法和本文提出的加速滤波方法对 GIF 方法加速执行的结果图。图 3(a) 为输入图像, 该图像为 GIF 测试典型用例; 图 3(b) 为 GIF 方法保边平滑结果; 图 3(c) 为本文提出的快速滤波方法对 GIF 方法加速保边平滑结果。对比图 3(b) 和图 3(c) 可知, 本文提出的快速滤波方法对 GIF 方法进行了有效加速, 同时也证明了本文提出方法的有效性。



(a) 输入图像 (b) GIF 方法实验结果 (c) 本文提出的加速 GIF 方法实验结果

图 3 保边平滑实验结果对比图

4 结论

本文提出一种面向嵌入式机载场景的快速图像滤波方法, 创新性提出了滤波像素抽样机制和自适应抽样步长机制, 使嵌入式机载场景的实时性和有效性得到了保障, 并在国产 NA16D3C7C1 GPU 加速卡上得到了验证。目前面向嵌入式使用场景的国产 GPU 算力相对较低, 工艺相对落后, 随着国产 GPU 工艺和架构技术的更新, 本文提出的算法将会具备更优秀的加速效果。

参考文献:

- [1] 姜琳琳, 王纯委, 海钰琳, 等. 高效能综合处理机设计实现研究[J]. 长江信息通信, 2023, 36(3): 87-90.
- [2] OCHOTORENA C N, YAMASHITA Y. Anisotropic Guided Filtering[J]. IEEE transactions on image processing, 2020, 29: 1397-1412.
- [3] YOO J S, KIM J O. Noise-robust iterative back-projection[J]. IEEE transactions on image processing, 2020, 29: 1219-1232.
- [4] SHI S, ZHANG Y, ZHOU X, et al. A novel thin cloud removal method based on multiscale dark channel prior (MDCP)[J]. IEEE geoscience and remote sensing letters, 2022, 19: 1-5.
- [5] 北京航空航天大学. 一种基于机载嵌入式 FPGA 开发平台含雾图像去雾增强方法: CN202210939290.0[P]. 2022-12-06.
- [6] 王辰北, 张海军, 王浩然. 基于改进 EDSR 的轻量化机载图像超分辨率算法[J]. 电光与控制, 2022, 29(10): 39-43+70.
- [7] 漆昇翔, 刘强, 徐国靖, 等. 面向机载应用的多传感器图像融合技术综述[J]. 航空电子技术, 2016, 47(4): 5-11.
- [8] RODRIGUES G S, KASTENSMIDT F L, REIS R, et al. Analyzing the impact of using pthreads versus OpenMP under fault injection in ARM Cortex-A9 dual-core[C]//2016 16th European Conference on Radiation and Its Effects on Components and Systems (RADECS). Piscataway: IEEE, 2016: 1-6.
- [9] ZAMOJSKI P, ELFOULV R. Developing Standardized SIMD API Between Intel and ARM NEON[C]//2018 International Conference on Computational Science and Computational Intelligence (CSCI). Piscataway: IEEE, 2018: 1410-1415.
- [10] 肖诗洋, 王镭, 杜莹, 等. 基于异构平台的图像中值滤波的 OpenCL 加速算法[J]. 河北大学学报(自然科学版), 2024, 44(1): 92-103.
- [11] 郑康, 李晨, 陈海燕, 等. 超长指令字 DSP 标量访存单元的设计与优化[J]. 计算机工程与科学, 2023, 45(11): 1929-1940.
- [12] 徐桂芝, 郭嘉荣, 张慧港, 等. 基于 FPGA 的类脑神经元仿生电路设计与实现[J]. 电机与控制学报, 2023, 27(2): 61-68.
- [13] HU L, WANG S, LI L, et al. How functions evolve in deep convolutional neural network[C]//2018 14th IEEE International Conference on Signal Processing (ICSP). Piscataway: IEEE, 2018: 1133-1138.
- [14] HE K, SUN J, TANG X. Guided image filtering[J]. IEEE transactions on pattern analysis and machine intelligence, 2012, 35(6): 1397-1409.
- [15] ALSHANNARI R H, ALMUHSIN F H, ALBUAINAIN F A, et al. A survey on image filtering algorithms[C]//2020 3rd International Conference on Computer Applications & Information Security. Piscataway: IEEE, 2020: 1-5.
- [16] 刘丹, 朱鸿泰, 程虎, 等. 基于双引导滤波的红外和可见光图像融合算法[J]. 激光与红外, 2023, 53(11): 1778-1784.

【作者简介】

李鹏亮(1989—), 男, 陕西岐山人, 博士, 工程师, 研究方向: 嵌入式计算、类脑计算等。

(收稿日期: 2024-03-21)