

# 一种基于FPGA的红外图像细节增强加速方法

史豪斌, 胡航语

(西北工业大学 计算机学院, 陕西 西安 710072)

**摘要:**在机载、车载、舰载平台的众多座舱人机交互系统中,图像细节增强能够提升人员对红外图像的判读能力,具有重要的应用需求。运行在嵌入式计算平台上的增强算法需要具有较高的处理速度和较低的算法延迟才能满足实时交互的需求。当前红外视频传感器分辨率较低,图像细节增强算法在同构多核CPU处理平台上即可达到实时处理性能。然而,随着平台传感器分辨率的持续增加,图像处理速度和延迟难以满足。提出一种针对FPGA平台的红外图像细节增强加速方法,考虑FPGA的特定领域软硬件计算架构,采用基于局部缓存和查表等方式对传统的双边滤波图像细节算法进行加速,使得在4k分辨率输入图像下,依然能够达到实时处理性能,同时几乎不增加额外的访存延迟。算法在保证处理效果的同时,极大地提升了处理和延迟指标,满足座舱人机交互系统等嵌入式装备特定领域应用需求。

**关键词:**加速;图像细节增强;机载;FPGA

**中图分类号:**TP391.41

**文献标志码:**A

**文章编号:**1000-2758(2022)03-0524-06

座舱人机交互系统是人员与装备的人机接口,在低能见度条件下,人员通过判读座舱显示器中呈现的红外视频画面,获得舱外视觉感知信息,进一步下达操作指令。红外视频传感器能够突破人类对光谱感知的范围,获得更加丰富的环境反射和辐射信息,使得人员获得等效目视能力,扩展其低能见度条件下的任务能力。然而,红外传感器因为成像频谱、成像方式以及大气衰减等众多原因,图像画面相对可见光较为模糊,不利于人员的判读理解,因此在送显之前通常需要进行红外图像细节增强处理。

根据FAA等相关规定,座舱显示系统的显示延迟必须小于100ms。随着红外传感器分辨率的不断提升,处理的像素规模成几何倍数增加,这对以同构CPU为主要处理单元的嵌入式计算平台处理能力提出了巨大挑战。CPU在处理数据的过程中,需要读取总线后把数据在内存中进行缓存,然后通过指令和数据指令逐个进行像素处理。虽然通过Cache缓存机制可以隐藏一定的计算过程中的内存读写,然而总线到内存的数据缓存无法在计算过程中进行隐藏。

本文针对FPGA特定领域处理架构,提出基于FPGA的红外图像增强算法。该算法通过基于视频流处理模型和局部相关特征,通过缓冲线(buffer line)和缓冲窗(window line)机制,进行图像细节增强。进一步结合空域查找表(space table)和值域查找表(range table)的方式,降低滤波过程中的计算量,实现处理和传输过程中的数据延迟,达到实时处理性能。最后,基于Xilinx HLS高层次综合工具,通过C++语言实现算法,结合FPGA硬件特征,实现性能优化。通过对比本文算法在FPGA端和CPU端算法的表现,以及能够获得最为高效的并行图像增强算法,验证本文算法的低延迟、高帧率的相关处理性能。

## 1 图像细节增强

图像细节增强分为传统方法和神经网络方法,其中传统方法通过空域滤波或者频域滤波的方式增强特定区域图像细节,而神经网络方法则是通过训

练深度学习网络模型参数,探索输入图像和增强后图像的映射关系。

在传统图像细节增强领域,基于导向滤波的增强方法<sup>[1]</sup>和基于双边滤波<sup>[2]</sup>的增强方法较为流行。后续的方法大多数基于此类方法进行改进。在加速计算方法方面,基于图像上采样和下采样的方式降低像素规模,极大提升了算法的速度。但是需要对图像进行多次盒函数滤波(box filter),因此必须进行多次图像缓存,增加了计算延迟。基于双边滤波的方法,对滤波半径 $R$ 内的像素进行空域权重(space weight)和值域权重(range weight)的加权平均,以获得边缘保持滤波效果。但随着滤波半径的增长,计算量呈现几何倍数增长。为了加速基于双边滤波的方法,Yang<sup>[3]</sup>对双边滤波器进行了数值近似,使得计算复杂度下降到 $O(N)$ 。然而数值近似必然牺牲一定的精度,在对画面要求不高的情况下可以接受。Cheng等<sup>[4]</sup>通过map-reduce的并行计算思想以及定点数替代浮点数计算方法,进一步加速了迭代双边滤波的图像细节增强方法。然而,该算法基于通用多核CPU的计算架构,所有数据必须先进行缓存,然后才能参与计算,这是通用CPU体系架构的固有缺陷。本文算法基于FPGA的特定领域架构,可以在读取总线数据时不进行全局缓存,直接计算后输出,从而实现零存储延迟计算。

在基于深度学习的图像细节增强领域,研究人员分别针对低照度、水下图像颜色校正、红外图像增强等需求都提出了对应的解决方法<sup>[5-7]</sup>。Qi等<sup>[8]</sup>提出一种异构Rybak神经网络模型(HRYNN),该算法更加接近神经生理模型,具有较好的主观评价效果。为了加速计算过程,Gharbi等<sup>[9]</sup>提出的HDRnet充分融合了神经网络的特征表示能力与双边滤波器的加速计算能力,能够拟合任意参数的滤波器。然而,大量的仿射参数需要预先存储在局部缓存中,必然引起较大的访存延迟。

虽然深度神经网络类算法在大多数情况下效果相对传统算法更加优秀,但是其存在:①训练数据问题;②模型解释问题;③处理性能问题。限制了其在座舱显示交互类应用的部署。因此,目前来说,在座舱显示交互领域,还是倾向于传统的处理方法。

本文针对FPGA的软硬件处理特性,改造经典的双边滤波图像增强方法,实现特定领域红外图像细节增强算法。该算法通过建立局部缓存机制和查表机制实现FPGA端传输处理一体化的计算架构。

算法仅缓存必须的局部像素,少量延迟即可输出计算结果,不需要进行全局图像缓存,极大地加速了计算过程。经过加速优化的红外细节增强算法,能够对4k分辨率图像实现实时处理,几乎不会增加额外的缓存延迟,满足座舱显示应用需求。

## 2 红外图像细节增强加速算法

### 2.1 基于双边滤波的细节增强

单幅图像 $I$ 可以分解为低频分量 $I_L$ 与高频分量 $I_H$ 两部分,即

$$I(\mathbf{p}) = I_L(\mathbf{p}) + I_H(\mathbf{p}) \quad (1)$$

式中, $\mathbf{p} = [\text{row}, \text{col}]^T$ 为像素位置索引。任意位置的图像低频分量可以通过空域平滑滤波得到,即

$$I_L(\mathbf{p}) = f(I(\mathbf{p})) \quad (2)$$

式中, $f(\cdot)$ 为空间域滤波函数,通常表示对图像 $I$ 的邻域像素进行加权平均。在得到图像低频分量之后,高频分量直接通过公式(1)得到,即 $I_H(\mathbf{p}) = I(\mathbf{p}) - I_L(\mathbf{p})$ 。图像的细节部分体现在图像的高频分量中,通过对图像高频分量进行适当增益即可实现图像的细节增强,即

$$I_E(\mathbf{p}) = I_L(\mathbf{p}) + \gamma \cdot I_H(\mathbf{p}) \quad (3)$$

式中: $I_E$ 为细节增强后的图像; $\gamma > 1$ 为细节增强系数,通过对 $I_H$ 分量的增益实现图像细节的放大。

如果 $f(\cdot)$ 为高斯滤波器等各向同性滤波器,会使得图像边缘部分模糊化,无法保持细节,造成所谓的光晕(halo)现象。基于性能和加速潜力,本文选用双边滤波器进行图像低频分量的提取,即

$$f(I(\mathbf{p})) = 1/W_p \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) \cdot$$

$$G_{\sigma_r}(I(\mathbf{p}) - I(\mathbf{q})) \cdot I(\mathbf{q}) \quad (4)$$

式中: $S$ 为 $\mathbf{q}$ 的滤波半径覆盖的邻域像素位置; $\|\cdot\|$ 为距离范数,通常采用2次范数, $G_{\sigma_s}(\cdot)$ 和 $G_{\sigma_r}(\cdot)$ 分别为空域高斯(spatial Gaussian)和值域高斯(range Gaussian)滤波函数, $W_p$ 为归一化系数

$$W_p = \sum_{\mathbf{q} \in S} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) \cdot G_{\sigma_r}(I(\mathbf{p}) - I(\mathbf{q})) \quad (5)$$

双边滤波器过程中空间距离和像素值同时相近的像素点才能具有更强的相似性和较大的滤波权重。在图像边缘位置,即使图像空间位置相近,但是像素值差异较大,其对应的滤波权重较低,因此能够实现边缘保持滤波,避免光晕现象。

虽然双边滤波具有较好的图像低频分量提取效果,但是对每一个像素,假设滤波半径为  $R$ ,则需要进行  $(2 \times R + 1)^2$  次加权计算。当  $R > 1$  时,计算复杂度呈几何倍数增加,急剧增加了算法处理的数据量。

## 2.2 基于 FPGA 的并行加速方法

在基于 FPGA 的图像处理系统中,像素数据通常按照顺序从左上角像素到右下角像素逐个传输,通常一个时钟周期能够读取一个像素值到局部缓存

中。而图像细节增强算法具有局部性,对于处理过的像素,局部缓存的相邻像素值就可以抛弃,这就避免了对整个图像的缓存,形成典型的处理数据流。同时对单个像素的操作过程中,整个逻辑电路组成 pipeline,以保证每个时钟周期内可以输出一个像素。本文提出一种基于局部缓存的图像细节增强方法(buffer based image detail enhancement, BIDE),通过采用少量局部缓存以及相应的查找表进行加速,实现了大分辨率红外图像细节增强。

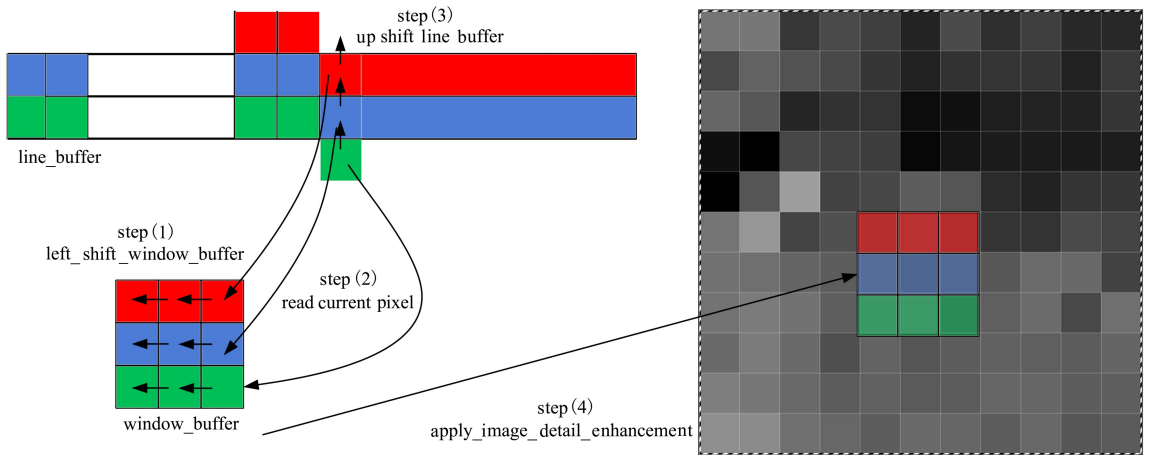


图1 图像细节增强加速算法图示

如图1所示,假设滤波半径  $R = 1$ , 输出像素只与输入  $3 \times 3$  邻域像素相关。加速过程仅需维护  $line\_buffer$  和  $window\_buffer$  2 个缓冲区。其中  $line\_buffer$  保存前 2 行输入像素值,  $window\_buffer$  则维护当前需要滤波像素的邻域像素值。通过不断更新输入像素,并舍弃不参与计算的像素值,利用最小的缓存实现增强像素的输出。具体的 BIDE 加速算法如下所示:

algorithm 1: buffer based image detail enhancement  
 input: current pixel  $I(\mathbf{p})$ ; window buffer  $\mathbf{b}_{win}$ ; line buffer  $\mathbf{b}_{line}$ ; filter radius  $R = 1$ ; scale factor  $\gamma = 2$ .  
 output: detail enhanced pixel  $I_E(\mathbf{p})$ .

```

1 for each input pixel  $I(\mathbf{p})$ ,  $\mathbf{p} = [\text{row}, \text{col}]^T$ 
2 step(1): left shift window buffer;
3  $\mathbf{b}_{win}(i, j) = \mathbf{b}_{win}(i, j-1)$ ,  $0 < i < 2R+1$ 
   and  $1 < j < 2R+1$ ;
4 step(2): read right column;
5  $\mathbf{b}_{win}(0, 0) = I(\mathbf{p})$ ;
6  $\mathbf{b}_{win}(1, 0) = \mathbf{b}_{line}(0, \text{col})$ ;
7  $\mathbf{b}_{win}(2, 0) = \mathbf{b}_{line}(1, \text{col})$ ;
    
```

```

8 step(3): up shift line buffer
9  $\mathbf{b}_{line}(2, \text{col}) = \mathbf{b}_{line}(1, \text{col})$ ;
10  $\mathbf{b}_{line}(1, \text{col}) = \mathbf{b}_{line}(0, \text{col})$ ;
11  $\mathbf{b}_{line}(0, \text{col}) = I(\mathbf{p})$ ;
12 step(4): apply image detail enhancement for
   each pixel
13  $I_E(\mathbf{p}) = f(I(\mathbf{p})) + \gamma(I(\mathbf{p}) - f(I(\mathbf{p})))$ , local
   filter range  $\mathbf{S} = \mathbf{b}_{win}$ .
    
```

在遍历像素时,可以通过 Xilinx HLS 中的 #Pragma HLS PIPELINE 指令来指导综合过程,使得一个时钟周期能够处理一个像素,降低算法延迟。算法 step(4) 中,主要计算量为局部双边滤波函数  $f(I(\mathbf{p}))$ 。此时参与局部滤波的像素值已经全部读入  $window\_buffer$  中,需要逐像素加权平均 space weight 和 range weight 后即可输出滤波值。而局部图像之间的相对位置关系已经确定,因此可以采用图 2 的 space table 提前存储空域权重。同样,图像的之间的亮度差值范围确定后,可以提前计算并存储 range table,在计算的过程中仅需查找表即可实现滤波输出,极大地降低计算量。space table 和

range table 可以通过 Xilinx HLS 中的 #pragma HLS ARRAY\_PARTITION 指令指定为寄存器变量,以增加其访问速度。

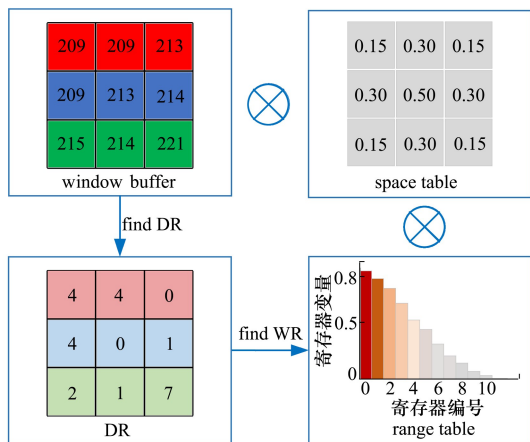


图 2 基于查找表的局部双边滤波加速

### 2.3 边界条件

如图 3 所示,在上述 BIDE 算法中,假设  $R=1$ , 则输入像素扫描到第 3 行时,才输出第 2 行像素。同样,当输入像素扫描到最后 1 行时,输出倒数第 2 行像素。此时,对于边界像素如果进行复杂的逻辑判断,则会极大地提升延迟,本文算法直接输出输入像素,对结果影响不大。同样起始列和末尾列像素的计算过程中,window buffer 跨行填充,无法保证参与计算的邻域像素值完全正确,此时也不必进行复杂逻辑判断,仅需正常 window buffer 计算,实现循环滤波,对结果影响也不大。

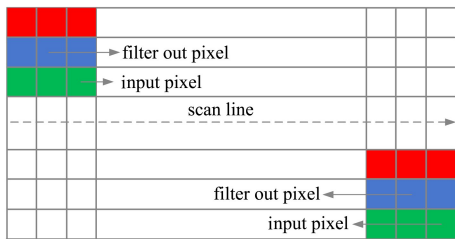


图 3 BIDE 算法的边界条件

## 3 实验对比与分析

### 3.1 实验条件

为了验证本文提出的 BIDE 算法的有效性,采用 Xilinx V7 690T FPGA 芯片进行算法部署(BIDE\_

FPGA),采用 Xilinx HLS 高层次综合环境进行综合,编程语言采用 HLS C++。为了验证算法相对嵌入式通用处理器的优势,采用 Nvidia TX2 多核 ARM 计算平台对 C++ 算法进行部署(BIDE\_CPU)。同时,在 TX2 平台实现了基于迭代双边滤波器的图像细节增强算法(PIDE)<sup>[5]</sup>,并采用单核、双核和四核版本进行算法对比,验证本文提出算法的性能。

### 3.2 实验结果

如图 4 所示,通过对红外光谱图像进行 BIDE 细节增强,能够提升整个画面的细节呈现度。在一些较暗区域显著提升了对比度,凸显了增强细节,提升人员对传感画面的辨识程度。

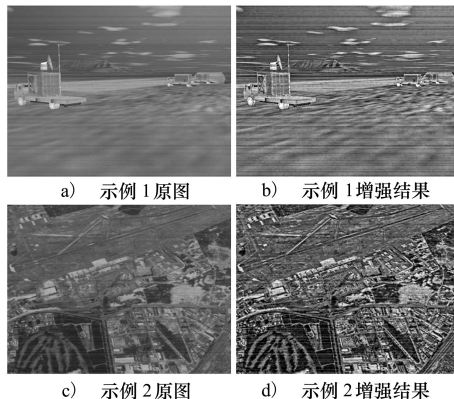


图 4 BIDE 算法图像增强结果

如表 1 所示,验证了 BIDE 算法在 FPGA 平台上综合后的性能表现。滤波半径  $R=3$  的情况下,HLS 综合工具给出相应的 Latency 和 Interval。在 4k 分辨率(4 096×2 160)下所需 Interval 仅为 9 333 362。对比 4k 分辨率的像素数量(8 847 360),近似达到一个时钟周期输出一个滤波值的极限加速性能。在 250 MHz(时钟周期 4 ns)下对应运行时间仅为 37.3 ms,帧率 26.8 frame/s。Latency 时钟周期仅比 Interval 多 3 061,对应 12.24 us,几乎可以忽略。因此,BIDE-FPGA 算法具有实时的计算性能,同时缓存延迟接近于 0。

表 1 多个分辨率下算法指标(时钟周期)

指标	分辨率			
	640×480	1 024×768	2 048×1 080	4 096×2 160
Latency	418 263	962 295	2 457 903	9 336 423
Interval	415 202	959 234	2 454 842	9 333 362

表 2 对比了不同分辨率下本文算法在 FPGA 端(BIDE-FPGA)和 CPU 端(BIDE-CPU)的性能表现,

同时对比 PIDE 算法在不同运行核数的表现。为了保持一致性,本文算法按照 250 MHz 运行频率换算为对应的图像处理时间。可以看出,BIDE 算法在 FPGA 进行部署后,算法处理时间效率提升 50 倍以上,满足实时处理需求。对比 CPU 端优化的 PIDE 算法,BIDE-FPGA 算法相对 4 核算法减少 5%左右运行时间。

表 2 不同算法的处理时间对比 ms

算法	分辨率			
	640×480	1 024×768	2 048×1 080	4 096×2 160
PIDE-1core	5.9	14.4	36.5	142.6
PIDE-2core	3.53	7.46	19.72	74.4
PIDE-4core	2.5	4.42	11.29	39.2
BIDE-CPU	56.7	139.9	403.1	1 535.5
BIDE-FPGA	1.7	3.8	9.8	37.3

本文处理算法的数据缓存延迟几乎可以忽略。对于 CPU 端算法,以嵌入式计算系统中常用的 ARINC818 视频总线速度 4.25 Gb/s,对于 4k 分辨率红外数据进行缓存,至少需要 24 ms 的缓存时间。缓存过程中无法进行计算,因此该延迟无法隐藏在计算中。本文的 BIDE 算法缓存时间(12.24 μs)对比 CPU 端算法延迟几乎可以忽略。

同样对于不同的滤波半径  $R$ ,单像素处理数据量按照滤波半径  $R$  的平方增加,处理时间在 CPU 端也急剧增加,如图 5 所示。然而在 FPGA 端,处理时间增加非常有限,如图 6 所示。这是因为在 FPGA 端,额外的计算量只是增加了部分逻辑电路,对于形成流水并行的图像处理队列,还能保持一个周期输出一个滤波像素的优异性能。

参考文献:

[1] HE K, SUN J, TANG X. Guided image filtering[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2012, 35(6): 1397-1409

[2] TOMASI C, MANDUCHI R. Bilateral filtering for gray and color images[C]//Sixth International Conference on Computer Vision, 1998: 839-846

[3] YANG Q. Recursive bilateral filtering[C]//European Conference on Computer Vision, 2012: 399-413

[4] CHENG Y, NIU W, ZHAI Z, et al. Parallel image detail enhancement for real-time applications[C]//AIAA Modeling and

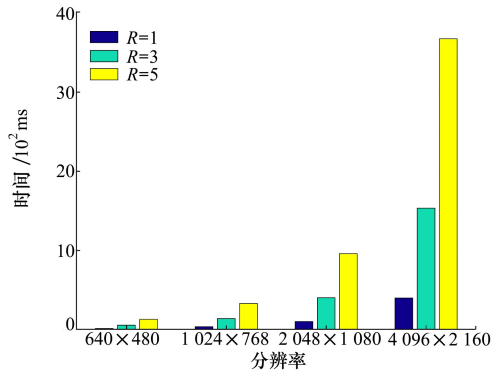


图 5 BIDE-CPU 性能表现

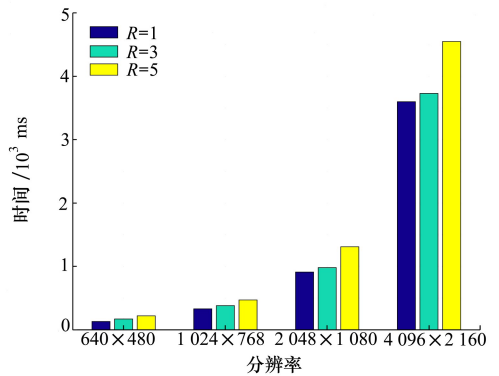


图 6 BIDE-FPGA 表现

4 结 论

本文面向座舱显示领域低延迟、高帧率的红外图像细节增强处理需求,对传统基于双边滤波器的图像细节增强算法进行加速。针对 FPGA 平台的软硬件特性,提出基于 line buffer 和 window buffer 的并行加速方案。通过 FPGA 的 pipeline 机制,结合 space table 和 range table 等方式实现红外细节增强算法的极限加速。对 4k 分辨率图像达到 37.3 ms 的处理性能,并且附加的存储延迟几乎为 0,极大地满足了座舱显控系统大分辨率红外图像细节增强的需求。

Simulation Technologies Conference, 2015: 3101

- [5] LI C, GUO J, PORIKLI F, et al. Lightnetnet: a convolutional neural network for weakly illuminated image enhancement[J]. Pattern Recognition Letters, 2018, 104: 15-22
- [6] WANG Y, ZHANG J, CAO Y, et al. A deep CNN method for underwater image enhancement[C]//2017 IEEE International Conference on Image Processing, 2017: 1382-1386
- [7] KUANG X, SUI X, LIU Y, et al. Single infrared image enhancement using a deep convolutional neural network[J]. Neurocomputing, 2019, 332: 119-128
- [8] QI Y, YANG Z, LIAN J, et al. A new heterogeneous neural network model and its application in image enhancement[J]. Neurocomputing, 2021, 440: 336-350
- [9] GHARBI M, CHEN J, BARRON J T, et al. Deep bilateral learning for real-time image enhancement[J]. ACM Transactions on Graphics, 2017, 36(4): 1-12

## Acceleration method of infrared image detail enhancement on FPGA

SHI Haobin, HU Hangyu

(School of Computer Science, Northwestern Polytechnical University, Xi'an 710072, China)

**Abstract:** Among the many cockpit human-computer interaction systems of airborne, vehicle-mounted, and shipborne platforms, the image detail enhancement technology can improve the ability of personnel to interpret infrared images, which has very important application requirements. The enhanced algorithm running on the embedded computing platform needs to have a higher processing speed and smaller time delay in meeting the needs of real-time interaction. The current infrared video sensor usually has a lower resolution, and the image detail enhancement algorithm still can achieve real-time processing performance on a homogeneous multi-core CPU processing platform. However, as the resolution of platform sensors continues to increase, image processing speed and time delay are difficult to meet application requirements. We propose a method for enhancing and accelerating infrared image details for FPGA platforms. By considering the FPGA's specific domain software and hardware computing architecture, the traditional bilateral filtering image details algorithm is accelerated using local cache and search table, so that real-time processing performance can still be achieved under the input image of the 4k resolution, while almost no additional memory access delay is added. While ensuring the processing effect, the algorithm greatly improves the processing and delay indicators to meet the application requirements of embedded equipment in specific fields such as the cockpit human-computer interaction system.

**Keywords:** acceleration; image detail Enhancement; airborne; FPGA

**引用格式:**史豪斌, 胡航语. 一种基于FPGA的红外图像细节增强加速方法[J]. 西北工业大学学报, 2022, 40(3): 524-529

SHI Haobin, HU Hangyu. Acceleration method of infrared image detail enhancement on FPGA[J]. Journal of Northwestern Polytechnical University, 2022, 40(3): 524-529 (in Chinese)