

Retinex 图像增强算法在 TS201 上的实时实现

雍 杨¹, 黄宝平¹, 王兵学², 黄自力¹

(1. 西南技术物理研究所, 四川 成都 610054; 2. 总装备部重庆军事代表局, 重庆 400042)

摘要: Retinex 图像增强算法能提升原图像的亮度、对比度和清晰度, 但其运算比较复杂, 难以满足许多工程应用所需要的实时性。通过采用 SingLeton 结构的高效浮点 FFT, 合并算法的多个循环体, 合理安排指令并行和软件流水等措施大幅提高运算速度, 并对图像进行分块处理以解决片内存储空间不够的矛盾, 在单片 TS201 DSP 平台上实时实现了 Retinex 图像增强算法。实验结果表明该算法在时钟频率 500 MHz 的条件下, 对 512×256 像素图像的处理用时 18.5 ms。

关键词: Retinex 理论; 浮点 FFT; SingLeton 结构; 汇编指令优化

中图分类号: TP391 **文献标识码:** A **DOI:** 10.3969/j.issn.1001-5078.2012.02.019

Real-time implementation of Retinex algorithm on TS201

YONG Yang¹, HUANG Bao-ping¹, WANG Bing-xue², HUANG Zi-li¹

(1. Southwest Institute of Technical Physics, Chengdu 610041, China;

2. Chongqing Military Deputy Bureau of General Armament Department, Chongqing 400042, China)

Abstract: The Retinex is an image enhancement algorithm that improves the brightness, contrast and sharpness. Many potential applications require the use of Retinex processing at video frame rates. It is difficult to achieve because the algorithm contains a large number of complex computations and data transfers. Accordingly, a fast computation algorithm executing on TS201 is proposed. The algorithm adopted series optimizations, such as SingLeton structure, parallelism of assembly instructions, loop unrolling and software pipelining. The input image is segmented in four blocks because of the limited memory of TS201. The experiment results prove that the algorithm can process a 512×256 image in 18.5 ms operating at a clock rate of 500 MHz.

Key words: Retinex theory; floating point FFT; SingLeton structure; assembly instruction optimization

1 引言

Retinex 理论是 Land^[1] 于 1977 年提出的一个关于人类视觉系统 (HSV) 如何感知和调节周围物体颜色和亮度的模型。其基本思想是将原始图像看成是由照射光图像和物体反射属性组成的, 照射光图像决定了一幅图像中像素能够达到的动态范围, 反射属性反映了图像的内在性质。因此, Retinex 理论指出, 通过分离照射光图像和反射图像, 能够消除光照不均的影响, 使分离出的反射图像有较好的视觉效果^[2]。

近年来, 越来越多的研究者认识到 Retinex 理论的科学性和合理性, 开始对它进行深入的研究。目

前已经提出多种基于该理论的算法, 比较常见的有 MaCann 算法、单尺度 Retinex 算法 (SSR)、多尺度 Retinex 算法 (MSR) 等。其中, 基于中心环绕的 Retinex 由于其处理效果好、实现方便的特点, 是目前应用最为广泛和受到关注度最大的 Retinex 理论方法。同时, 也有许多文章介绍了该理论在图像增强方面的应用, 如 Ahmed, Li 等人将其应用于彩色图像或黑白图像的增强^[3-4], 黄黎红将其用于雾天图像的

作者简介: 雍 杨 (1978 -), 女, 博士, 讲师, 博士学位, 主要从事图像处理, 目标识别与跟踪等方面的研究。E-mail: yy-min@163.com

收稿日期: 2011-06-22; 修订日期: 2011-07-25

去雾处理^[5]。诸多研究表明,Retinex 增强算法具有颜色恒常性、动态范围压缩大、色彩逼真度高、锐化等优点,是一种具有广泛应用前景的技术。但由于 Retinex 算法的运算复杂比较高,这些应用都还局限于软件仿真,并没有实时实现。本文从实际工程需求出发,对单尺度 Retinex 算法的结构进行优化,结合高效的浮点 FFT 算法,利用 TS201 指令并行规则,合理安排软件流水,在单片 TS201 DSP 平台上实现了对 512×256 大小图像的实时处理。本文还在实际工程中提炼出一些具有创新性的优化方法,如根据实际需求对图像进行分块处理以解决 DSP 片内存储空间不足的矛盾,以及用帧间平滑策略提高视频图像增强后的视觉连续性,这些优化方法具有一定的普适性,在有关图像处理的工程项目中有很好的推广价值。

2 单尺度 Retinex 算法原理简介

单尺度 Retinex 算法是 Jobson 和他的同事在 1997 年对 Edwin Land 中心环绕 Retinex (Center Surround Retinex) 的改进和实现。设原图像为 $I(x, y)$, 亮度图像为 $L(x, y)$, 反射图像为 $R(x, y)$, 根据 Retinex 理论,有:

$$I(x, y) = L(x, y) * R(x, y) \quad (1)$$

将上式两边取对数,则可抛开入射光的性质得到物体的本来面貌:

$$\log[R(x, y)] = \log[I(x, y)] - \log[L(x, y)] \quad (2)$$

在对数域中,单尺度 Retinex 可以表示为:

$$\log[R(x, y)] = \log \frac{I(x, y)}{L(x, y)} = \log[I(x, y)] - \log[I(x, y) * F(x, y)] \quad (3)$$

其中, $F(x, y)$ 为中心环绕函数。在公式(3)中,中心环绕函数与原图像的卷积可以看作是对空间中照度的计算,从原图像中减去照射光图像,就得到了物体的反射属性。 $F(x, y)$ 相当于一个高斯模板:

$$F(x, y) = Ke^{-(x^2+y^2)/\sigma^2}$$

式中, K 由归一化函数决定。

3 Retinex 在 DSP201 上的实现

美国模拟器件公司 (ADI) 在 2003 年推出的 TS201 处理器是高性能数字信号处理领域的新一代 DSP, 采用静态超标量体系结构, 片内集成了大容量的存储器, 适用于高性能、大存储量的信号处理和图像应用。其采用超级哈佛结构, 内部集成 24 Mbit 的 DRAM 存储器, 支持 32 和 40 bit 的浮点以及 8,

16, 32 和 64 bit 的定点运算。TS201 的结构经过了特殊的优化, 每周可执行四条指令, 完成 24 次 16 bit 定点运算, 12 次 32 bit 定点运算或者 6 次浮点运算, 在 600 MHz 的时钟频率下, 可以达到每秒 48 亿次乘加运算和每秒 36 亿次浮点运算的速度。

在实际应用中, 每场图像大小为 512×256 像素, 要在 DSP 上实时实现需要在 20ms 内完成对图像的处理。而 Retinex 增强算法涉及到大模板的卷积运算, 还有多次的求对数和求指数的运算, 计算复杂度高。因此, 必须对算法进行优化, 才能保证算法的实时实现。本文主要采用了以下几方面的优化措施:

(1) 将卷积运算转换为频域的乘积运算

从式(3)中看到, 求反射图像是通过原图像与高斯核函数的卷积运算实现的, 而为了获得良好的增强效果, 高斯核函数的尺寸是很大的, 如果在空间域中对每个像素点进行卷积运算会导致巨大的运算量。因此, 可以将空间域中两个函数的卷积转换到频率域中用乘积来实现:

$$I(x, y) * F(x, y) \Leftrightarrow \text{FFT}(I(x, y)) \times \text{FFT}(F(x, y)) \quad (4)$$

最后, 可以得到用 Retinex 理论求图像照射分量的计算公式:

$$R(x, y) = \log[I(x, y)] - \log[\text{IFFT}(I'(u, v) * F'(u, v))] \quad (5)$$

其中, $I'(x, y)$, $F'(x, y)$ 是原图像 $I(x, y)$ 和高斯核函数 $F(x, y)$ 的 FFT 变换结果。

(2) 浮点 FFT 的结构优化和汇编优化

根据前面的描述, FFT 是 Retinex 算法的核心环节, 它的运算速度直接影响到 Retinex 能否实时实现, 因此必须寻找一种高效的 FFT 实现方法。目前已有许多文献对 TigerSHARC 上实现 FFT 进行了研究。文献[6]中给出了 TSI01 和 TS201 采用标准结构做定点 FFT 的实现方法, 但存在 TS201 做大点数 FFT 时 DRAM 读写效率下降的问题。为了提高做大点数 FFT 时 TS201 内部 DRAM 的读写效率, 文献[7]中给出了 TS201 采用 SingLeton 结构做复数浮点 FFT 的实现方法, 采用这种结构实现的 FFT 运算其速度能比标准结构 FFT 提高两倍。文献[8]则采用 SingLeton 结构实现了大点数的定点 FFT 变换, 完成 32 K 点的 FFT 只需要 0.46 ms。因此, 本文采用了 SingLeton 结构的高效浮点 FFT 算法, 下面对这种方法的优越性做简单介绍。

图 1 示出一种输入经过比特位反转后的标准 8 点 radix-2 FFT 实现。在这一算法中,第一级和第二级蝶形运算是按照所要求的位反转方式结合在一起成为一种单一的优化循环的(因为这两级蝶形运算不要求乘法运算,只是加和减)。其余的每一级蝶形运算都是将共享相同旋转因子的蝶形运算结合成组(这样对于每一组,旋转因子只能取数一次)来进行。

ADSP-TS201 内存是针对顺序读取优化的,高速缓冲可以为非顺序读取的算法提供帮助。在标准 FFT 算法中,每一级的蝶形运算步进为双倍,这样,每一级运算的数据读取都是非顺序的,随着运算技术的增高,高速缓存越来越不可能做到对任意地址的数据快速读取,这就是大点位标准结构 FFT 在 TS201 计算上效率降低的原因。TS201 的函数库中提供了相应的 FFT 函数,在时钟频率 500 MHz 的条件下,32 K 点的实数 FFT 耗时 3.9 ms,这样的运算速度显然不能满足工程需求。如果能对一个进程的输出进行重排以确保下一个进程的读取是顺序的,运算效率应该有大的提高。

SingLeton 结构给出的按时间抽取 FFT 结构流程图,是一种输入为反序、输出为自然顺序的恒定几何结构,如图 1 所示。其特点是除了第一级运算对数据需要进行反序读取之外,其他每一级均是顺序地址读取数据。在做完蝶形运算对数据进行存储时,也是顺序存储,SingLeton 结构的关键是对每一级运算的输出进行重排以确保下一级运算的读取是顺序的, N 点 FFT 重新排序的公式是:

$$f(n) = \begin{cases} \frac{n}{2}, & \text{if } n \text{ is even} \\ \frac{n-1}{2} + \frac{N}{2}, & \text{if } n \text{ is odd} \end{cases} \quad (6)$$

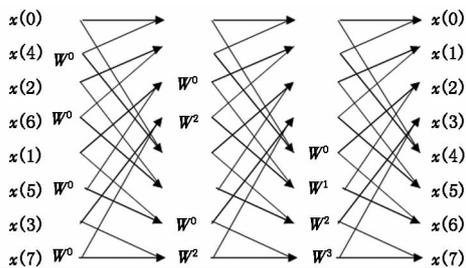


图 1 SingLeton 结构的 8 点 FFT 流程图

Fig. 1 reorganized structure of the 8-point FFT

重排操作可以避免 TS201 内部 DRAM 在随机读取时效率低的问题,其缺点是每一级运算之间不是原位操作,需要额外增加两块内存进行 ping-pong

操作。但是根据需要可以对内存进行优化,例如,当输入数据不需要保存时,可以把输入当作两个 ping-pong 缓冲器之一。

FFT 的核心是蝶型运算,一个蝶型运算包括一次复乘、一次加法和一次减法。TS201 内部有 4 条 128 bit 宽度的内部总线,通过 J 和 K 两个整型 ALU 可以实现一个周期内 4 个 32 bit 数据的读取和 4 个 32 bit 数据的存储。此外,TS201 内部有两个对称的运算单元和,这样可以在一个周期内完成两次复乘和加减法运算。据此分析,TS201 可以在一个周期内可以完成两个蝶型运算。在编写软件时,利用 TS201 指令并行规则,充分发挥了 TS201 单指令多数据(SIMD)的优势,加速了 FFT 的运算。表 1 给出了在 TS201 上用新老两种结构计算 FFT 的时钟周期对比。

表 1 新老结构 N 点复数 FFT 内核时钟周期对比
Tab. 1 core clock cycles for N -point complex FFT, new versus old structure

| 点数 | 标准结构 FFT | SingLeton 结构 FFT |
|-------|----------|------------------|
| 4096 | 197272 | 69924 |
| 8192 | 444628 | 147628 |
| 16384 | 987730 | 313292 |
| 32768 | 2133220 | 662614 |
| 65536 | 4720010 | 1397544 |

(3) 简化循环体和指令并行

在实验中发现,过多的循环次数会导致 DSP 运行效率的下降,因此,可以从两方面入手提高循环体的执行效率。一是尽量合并循环,由于 Retinex 算法中所有的运算都需要对每个像素点进行处理,因此,许多步骤可以合并到一个循环体中,如高斯模板与图像 FFT 变换结果的乘法、原图像求对数等操作就可以进行循环合并。另一方面是每个循环周期内尽量多处理些像素,利用 J 和 K 两个整型 ALU,每次循环周期内可以读取 8 个浮点数据进行运算,大大减少循环次数。

在循环体内,许多指令可以并行执行,这一级别的优化是最直接和简单的,只需按照 TS201 的并行规则将各指令并行起来即可。

(4) 图像分块处理

用 SingLeton 结构实现的 N 点 FFT,输出为 $N/2$ 点的复数结果,由于实数的 FFT 变换结果是共轭对称的,其余的 $N/2$ 个结果很容易获得。因此,对于 FFT 而言,从理论上说需要提供四个大小为 N 字节

的存储空间给输入、输出数据和两个 ping-pong 缓存空间,在输入不需要保留的前提下进行内存优化也还需要三个大小为 N 字节的存储空间。此外,IFFT 输入的数据是复数形式,因此 N 点的 IFFT 变换输入、输出和 ping-pong 缓存空间均需要 $2N$ 字节大小。这样一来,TS201 的内部存储空间就出现不够用的矛盾。为了解决这一问题,将 512×256 像素的图像分成 4 个 256×128 像素的图像块,对它们分别进行处理后再将其结果组合起来。经过实验验证,分块处理对 Retinex 算法的增强效果影响甚微。

(5) 帧间平滑

在实际应用中,Retinex 算法是对图像序列进行处理的,要把对每一帧图像的增强程度控制在一个很稳定的范围内,才能使整个图像序列的增强效果自然流畅,没有忽明忽暗的闪烁。在算法最后的线性拉伸处理中,采用的公式为:

$$I'(x,y) = \frac{I(x,y) - \min(I)}{\max(I) - \min(I)} \times 255 \quad (7)$$

其中, $\max(I)$, $\min(I)$ 是图像的最大最小值,它们对成像过程中的诸多因素非常敏感,场景稍有变化都会导致图像序列中每帧图像的最大、最小值出现比较大的波动,而它们的变化,直接影响到图像拉伸后的明暗程度,造成了图像序列在增强后会出现闪烁,影响了人眼的视觉效果。

为了解决这一问题,可以利用图像序列的多帧信息来对当前帧的拉伸处理进行平滑,具体步骤为:保留前 10 帧图像的最大最小值,并与当前帧的最大最小值进行比较,如果它们的差值在事先确定的阈值范围内,就用当前帧的最大最小值进行线性拉伸,否则用前 10 帧图像最大最小值的均值来代替。经过多帧图像极值的平滑,每帧图像的拉伸范围得到了控制,使得增强后的闪烁现象明显减弱。

此外,Retinex 增强算法需要多次对整幅图像进行求对数、求指数的运算,为了减少运算时间,可以用查找表的方式来完成这些运算。

4 实验结果

如上所述,通过算法结构优化、简化循环、指令并行三个层次的优化处理,极大地提高了 Retinex 在 TS201 上的运算速度, 512×256 像素的图像处理时间从未优化前的 78 ms 提升为 18.5 ms,实现了对图像序列的实时处理。图 2 和图 3 是是 TS201 DSP 平台对实际图像的增强效果。图 2 是对测试图像进行实时增强的结果,原图像背景中的树林灰度集中分

布在直方图的低端,缺乏层次,所以很多细节完全看不到,增强后图像中的细节得到了凸显,并且画面非常自然,具有良好的视觉效果。图 3 采用雾天城市图像作为测试图像,处理后图像清晰度有很大提高,远处楼房轮廓明显清晰,显示了较好的去雾效果。



(a) 原图像 (b) 增强结果
(a) original image (b) enhancement result

图 2 典型图像增强效果

Fig. 2 enhancement result of typical image



(a) 原图像 (b) 增强结果
(a) original image (b) enhancement result

图 3 实际采集图像增强效果

Fig. 3 enhancement result of real-time image

5 结论

本文采用高效的浮点 FFT 算法,对算法结构进行优化,充分运用汇编命令可并行性,在时钟频率 500 MHz 的条件下,把 Retinex 图像增强算法在单个 TS201 DSP 平台上的运行时间从未优化前的 78 ms 减少到 18.5 ms,保证了该算法的实时实现。Retinex 图像增强方法具有自适应好、适用面广的优点,在视频监控、自动目标识别、精确制导等领域有广泛的应用空间。因此,本论文提出的方法具有一定的通用性,可以推广应用到多个领域,有较高的工程实用价值。

参考文献:

- [1] Land E H. The Retinex theory of color vision[J]. Scientific American, 1977, 237(6): 108 - 129.
- [2] Peng Jiaqi, Liu Bingqi, Dong Wei, et al. Method of image enhancement based on multi-scale Retinex[J]. Laser & Infrared, 2008, 38(11): 1160 - 1163 (in Chinese)
彭佳琦,刘秉琦,董伟,等.基于多尺度 Retinex 的图像增强算法[J].激光与红外, 2008, 38(11): 1160 - 1163.
- [3] Ahmed A M T. Color restoration techniques for faded colors of old photos, printings and paintings[J]. IEEE International Conference on Electro/Information Technology,

- 2009;151 – 156.
- [4] Li Y, He R J, Xu G Z, et al. Retinex enhancement of infrared images[C]. 30th Annual International IEEE EMBS Conference, 2008; 2189 – 2192.
- [5] Huang Lihong. Fog-degraded image enhancement based on single-scale Retinex[J]. Journal of Applied Optics, 2010, 31(50): 728 – 733 (in Chinese)
黄黎红. 一种基于单尺度 Retinex 的雾天降质图像增强新算法[J]. 应用光学, 2010, 31(50): 728 – 733.
- [6] Lerner B. Parallel implementation of fixed point FFTs on Tiger SHARC processors [EB/OL]. [2005 – 03 – 02]. <http://www.analog.com/dsp>.
- [7] Lerner B. Writing efficient floating-point FFTs for ADSP – TS201 Tiger SHARC processor [EB/OL]. [2004 – 05 – 06]. <http://www.analog.com/dsp>.
- [8] Li Xin, Liu Feng, Long Teng. Efficient implementation of fixed-point FFT on TS201 [J]. Transactions of Beijing Institute of Technology, 2010, 30(1): 88 – 92 (in Chinese)
李欣, 刘峰, 龙腾. 定点 FFT 在 TS201 上的高效实现 [J]. 北京理工大学学报, 2010, 30(1): 88 – 92.