

文章编号: 2095-4980(2018)02-0317-06

基于 ARM NEON 的静态 YUV 图像缩小技术

陈 益, 李文钧

(杭州电子科技大学 电子信息学院, 浙江 杭州 310000)

摘 要: 在视频监控领域, 视频采集与处理的图像数据都是视频颜色编码方法(YUV)数据格式, 而传统的静态 YUV 图像处理技术存在处理速度、处理质量、应用范围不能同时得到保证等问题。文章提出的基于进阶精简指令集机器(ARM)架构处理器扩展结构(ARM NEON)的静态 YUV 图像缩小技术, 在视频监控领域中广泛应用于图像的压缩存储和样片采集。该技术使用的是 ARM 的 Ambarella S2 硬件平台和 Linux 操作系统, 采用 64 字节对齐块状处理方式来达到。相比于传统的压缩存储技术, 该技术速度要快 2 到 3 倍, 且压缩后图像清晰。仿真测试结果表明该技术具有代码效率高, 处理速度快, 输出图像质量高, 适用环境适应性强的实际应用价值。

关键词: 图像缩小; NEON 技术; 64 字节对齐; 平滑处理

中图分类号: TN919; TP317.4

文献标志码: A

doi: 10.11805/TKYDA201802.0317

Static YUV image reduction technology based on ARM NEON

CHEN Yi, LI Wenjun

(Institute of Electronic Information, Hangzhou Dianzi University, Hangzhou Zhejiang 310000, China)

Abstract: In the area of video surveillance, the image data of video collecting and processing are in Video color coding method(YUV) data format. But the classical static YUV image processing technology cannot perform well simultaneously in the aspects of processing speed, processing quality and range of application. The method proposed is a static YUV image contraction technology based on Acorn RISC Machine(ARM) architecture processor expansion structure(ARM NEON), and is widely applied in the image compressing and storing and sample collecting. Using ARM's Ambarella S2 hardware platform and Linux operating system, this technology adopts 64 byte alignment block process mode. Compared to traditional compression storage technology, the speed of the proposed technology can be fast by 2 to 3 times, and the image after compression is clear. Experimental results show that the proposed method has high value of practical application for its high code efficiency, fast processing speed, high image output quality and strong adaptation to environment.

Keywords: image contraction; NEON technology; 64 byte alignment; smooth processing

图像缩放技术在图像处理领域占有重要的地位。在视频监控领域, 视频采集和处理的图像数据主要是 YUV 格式, 所以本文图像处理的对象是静态 YUV 图像。对图像进行处理是指对图像的分辨力和像素值进行改动, 得到所需要的分辨力和像素值。这种处理技术在媒体技术发达的今天有着重要而广泛的应用, 例如在医学拍片、交通违规拍照、动画插图、工程效果图等方面都离不开图像处理技术。

图像处理技术是对图像数据的重新采集和排列。对图像数据的不同处理方式将直接影响处理的速度和处理后图像的质量。而处理速度和处理后图像的质量又是一对成反比的关系。如果要求处理速度快, 那么必然会降低输出的图像质量; 如果要求输出的图像清晰不模糊则需要对图像数据进行优化平滑处理而大大增加处理时间。传统的图像处理技术不能同时保证处理速度和处理质量。针对这 2 点, 本文提出的基于 ARM NEON 的静态 YUV 图像缩小技术能够应用于视频监控领域的图像存储、样片采集和缩小保存, 并且在提高处理速度的同时能够保证输出的图像清晰不模糊。兼顾这 2 点的图像处理技术也同时扩大了应用范围。由于 NEON 技术的特点, 本文只针对主流的 2 倍、4 倍、8 倍、16 倍缩放来说明, 但同样适用于 32 倍、64 倍、128 倍缩放。

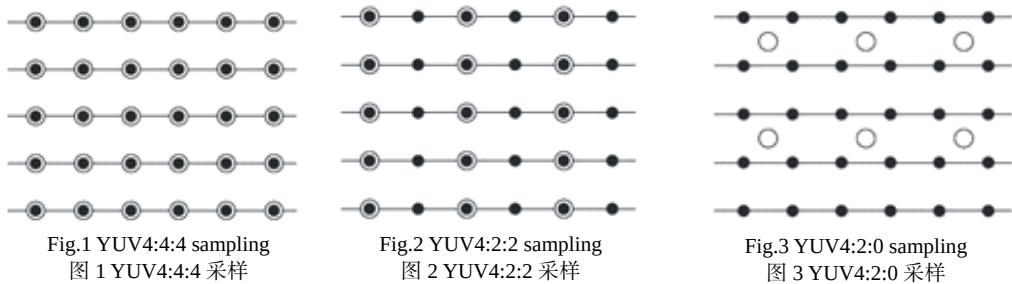
1 静态 YUV 图像缩小技术的设计与实现

本文提出的静态 YUV 图像缩小技术是基于 ARM NEON 技术和双线性插值算法进行优化。提出采用自适应对齐方式选择和块状化图像处理的方法进行静态 YUV 图像缩小处理^[1]。

从 ARMV7 开始, ARM 提供了单指令、单指令多数据流(Single Instruction Multiple Data, SIMD), 即 NEON 技术。它是一种 64 bit 和 128 bit 混合 SIMD 体系结构, 在媒体信号的处理中能大大提高程序的效率。由于其寄存器的特点, 最适合处理 8 bit 以及 8 的整数位的数据。一个像素刚好是 8 bit 也就是 1 个字节, 所以在图像处理中应用 NEON 技术, 可以有效提高图像处理速度^[2]。

1.1 YUV 数据的采集与存储

在视频监控领域中图像的存储数据有 YUV 格式和 RGB(视频颜色编码方法)格式^[3], 但因为主流的视频采集与处理最终都是针对 YUV 数据, 所以本文主要研究对象是 YUV 数据格式图像。YUV 数据中的 Y 表示亮度的数据, 就是灰度值, 如果单单只有 Y 数据的图像, 显示出来就是 1 张黑白图。而 UV 数据表示的是图像的色度, 如果是一个完整的 YUV 数据的图像, 显示出来就是 1 张彩色图像。所以静态 YUV 图像缩小技术就是对 YUV 三种数据的重新取值和存储。



主流的 YUV 数据的采集方式有 3 种形式, 分别为 YUV4:4:4 采样、YUV4:2:2 采样和 YUV4:2:0 采样。图 1 是 YUV4:4:4 采样, 就是每个像素采集 1 个 Y 数据, 和 1 组 UV 数据; 图 2 是 YUV4:2:2 采样, 就是每个像素采集 1 个 Y 数据, 每 2 个像素采集 1 组 UV 数据; 图 3 是 YUV4:2:0 采样, 就是每 1 组 2×2 的像素采集 1 个 Y 数据和 1 组 UV 数据(图中黑点表示 Y 数据, 空心圈表示 UV 数据)^[4]。

YUV 数据主流的存储方式有 YUV422 和 YUV420 2 大类, 分别采用 YUV4:2:2 采样和 YUV4:2:0 采样。其中 422 主要分为 YU16 和 YV16 2 种格式, 420 主要分为 IYUV,YV12,YU12 3 种格式^[5]。本文提出的静态 YUV 图像缩小技术对以上 5 种格式都适用。以像素分辨力为 20×2, 存储格式为 YU16 为例来说明, YU16 存储方式是先将所有的 Y 数据存储完, 再存储所有的 U 数据, 最后存储所有的 V 数据, 如表 1 所示^[6]。

表 1 YU16 格式数据存储

Table1 YU16 format data storage

Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y	Y
U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U	U
V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V	V

1.2 自适应对齐方式选择

NEON 技术加载数据的最小基本单位是 8 bit, 即 1 个字节, 每次加载都会加载满整个寄存器, 一次性可以最多加载满 4 个寄存器, 而 NEON 寄存器又分为 128 bit 宽和 64 bit 宽 2 种类型, 所以每次从内存中加载数据, 最少可获取 8 个字节, 也就是 8 个像素, 最多可获取 64 个字节, 也就是 64 个像素^[7]。而影响处理速度的关键就是每次从内存中加载像素数据的数量。

经过实验测试, 要提高程序的效率, 每次应尽可能多地加载 YUV 数据, 且当每次加载 64 个字节时达到最高效率。但也存在一个问题, 如果每次加载 64 个字节的 YUV 数据, 那么就要保证图像宽要满足 64 字节对齐, 这就大大缩小了满足条件的图像, 缩小了应用的范围。所以对于传统的方式利用 NEON 技术加速, 由于要考虑对齐情况, 并不一定会每次加载 64 个字节的数据进行图像处理, 这使快速性和实用性大打折扣。所以本文正是针对这点来进行优化, 采用的自适应对齐方式选择的方法可将图像像素要求从 64 字节对齐降低到 8 字节对齐,

但依然保持每次加载 64 个字节的 YUV 数据进行图像处理。另外经过测试,每次加载的行数并不影响图像处理的速度^[8]。

下面以分辨率为 $x \times y$ 的 YUV 图像进行 8 倍缩小为例进行详细说明,详细过程参考图 4。

首先对图像的宽分辨率是否满足 64 字节对齐进行判断。如果是,直接按 64 字节对齐进行处理;如果不是,则先剔除最后 8 行数据,然后剩下的每行由下一行前几个 YUV 数据在这行的后面补齐到 64 bit 对齐,再按 64 字节对齐进行处理。最后对剩下的 8 行数据进行特殊处理。设 $x/64$ 的商为 a ,余数为 b ,那么剩下的 8 行每行前 $64 \times a$ 个 YUV 数据仍按 64 字节对齐处理,剩下的 8 行每行最后 b 个 YUV 数据由于满足 8 字节对齐,所以 b 的值有 8,16,24,32,40,48,56 7 种情况,最后分 7 种情况来特殊处理 $8 \times b$ 个数据。

1.3 块状化图像处理的实现方法

前面说到图像处理技术离不开速度和处理后图像的质量这 2 点,上面对处理速度方面进行了优化,接下来对处理质量进行优化。

传统的图像缩放算法有最近邻插值、双线性插值和双三次插值 3 种。由于受限于 NEON 技术的特点,本文采用的是基于双线性插值算法的块状化图像处理方式。这种处理方式相对于最近邻插值法,图像缩小后的质量有大幅提高。

通过最近邻插值法,对图像进行缩小处理,最终获取到的图像像素都是原来像素中某一个,并未做任何平滑处理,所以得到的图像也会有明显的锯齿现象,清晰度比较低。而双线性插值法要取的新像素值是通过一个 2×2 的区域 4 个相邻的像素值通过加权平均求得的。具体方法如图 5 所示, A, B, C, D 为 2×2 区域的 4 个点, X 为要求的新的像素点,先由 A 和 B 求得 AB 线段的线性关系,然后再由 C 和 D 求得 CD 线段的线性关系,然后求得 E 和 F 2 点的值,再根据 E 和 F 求得 EF 线段的线性关系,最后可求得 X 点的像素值。通过这种方式缩小后得到的图像质量较高,不会有锯齿现象。

本文中静态 YUV 图像缩小倍数都是偶数倍,并且宽与高缩小相同倍数,所以根据双线性插值算法的原理,本文采取的是先取一个块状区域,然后求区域的中心位置像素值,也就是直接取相邻 2 个像素值的平均值。先对区域中同一行的 YUV 数据平滑处理成缩小相应倍数后的 YUV 数据,再对所有行的数据平滑处理成一行。由于增加加载的行数对处理速度的影响并不是很大,所以为了便于图像平滑处理,每次加载数据时加载的数据行数和缩小的倍数相对应。具体过程如图 6 所示,当图像缩小 x 倍时,每次从内存中加载 $64 \times x$ 的子方块数据进行平滑处理。

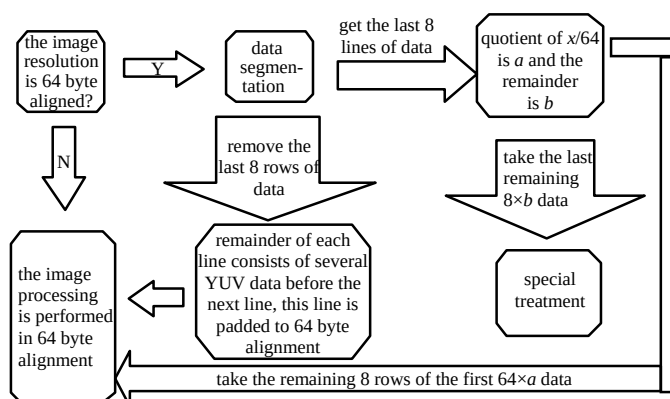


Fig.4 Flow chart of adaptive alignment selection

图 4 自适应对齐方式选择流程图

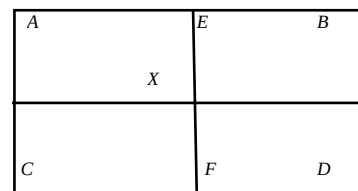


Fig.5 2×2 points on the map area

图 5 2×2 区域上各点示意图

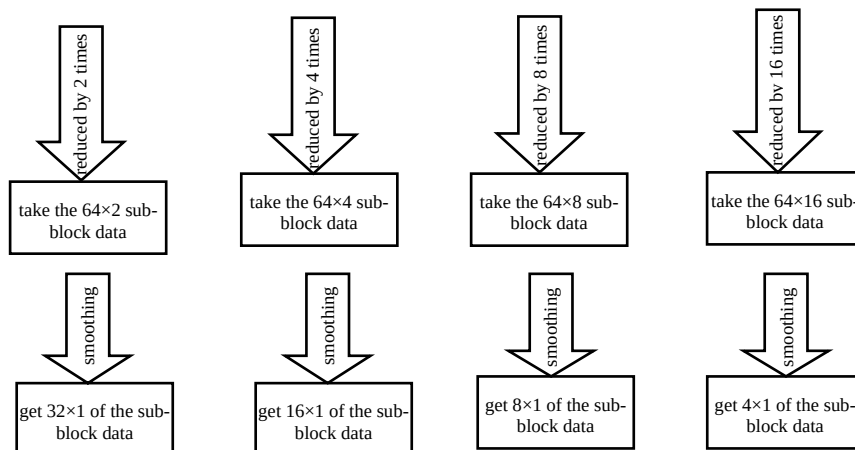


Fig.6 Schematic of blocking process

图 6 块状化处理示意图

通过这种方式得到的缩小图的像素值是经过平滑处理后的新值,能够保证一定的清晰度,不会有明显的锯齿现象。并且得益于 NEON 技术,这些处理过程都可通过相关 NEON 内嵌函数得以实现,相比于传统的方式去处理这些数据,大大缩小了处理时间。

2 实验测试与结果分析

2.1 实验平台

软件平台:实验采用 ubuntu 12.04.2 LTS 操作系统(GNU/Linux 3.5.0-23-generic x86_64),以及 ubuntu 平台的 pyuv 播放器。

硬件平台:使用海康威视的 600 万高清网络摄像机 DS-2CD4065F-AP,内置芯片为安霸高清 IPC 芯片-Ambarella S2,是一款基于 ARM V7 指令集的处理单元,其特点是能提供 NEON 技术支持。

2.2 图像处理速度测试的实验方法和结果

1) 采用 8 字节对齐和 64 字节对齐时的处理速度对比

为了体现优化的效果,所以尽量选取分辨率比较大的图像 YUV 数据。本次实验选取的是分辨率为 $3\,072 \times 3\,072$ 的图像 YUV 数据进行 8 倍缩小的处理。并且每次加载 8 行 YUV 数据进行图像处理。分 2 组实验,1 组采用本文中优化的 64 字节对齐进行图像处理,1 组采用 8 字节对齐进行图像处理。

按照 8 字节对齐进行图像缩小处理,记录 10 次实验测试的耗时情况见表 2;按照 64 字节对齐进行图像缩小处理,记录 10 次实验测试的耗时情况见表 3。

通过上面的实验数据对比可知,本文提出的自适应对齐方式采用的实时 64 字节对齐来进行图像处理,相比于传统的 8 字节对齐程序效率提高了 3.277 倍,保证了图像处理的快速性。

2) 每次加载数据时加载 8 行数据与加载 16 行数据的处理速度对比

本次实验同样选取 $3\,072 \times 3\,072$ 分辨率的图像 YUV 数据进行 8 倍缩小处理,并且每次图像处理加载 16 行的 YUV 数据,图像处理时按照 64 字节对齐进行缩小处理,记录 10 次实验测试的耗时情况见表 4。

通过表 3 的实验数据和表 2 的实验数据可以看出,每次加载数据时加载的行数对图像处理的速度影响并不是很大,对于大分辨率的图像也又快 3 ms 不到,所以每次处理时并不需要尽可能多地加载更多行。说明本文提出的块状化图像处理不影响图像处理速度。

2.3 图像质量对比测试

本次实验采用的图片样本是分辨率为 $1\,080 \times 720$ 的 YUV 图像。因为 $1\,080$ 的分辨率是 8 字节对齐,并不满足 64 字节对齐,从而可以用来测试本文提出的自适应对齐方式的处理方法,能否将图像分辨率对齐要求降低到 8 字节对齐。处理前的样本 YUV 图像用 pyuv 软件查看,如图 7 所示。

表 2 采用 8 字节对齐时图像处理的耗时情况

Table 2 Time-consuming of image processing by using 8 byte alignment

test number	time consuming/ms	average time-consuming/ms
1	201.175	204.071 1
2	207.495	
3	197.737	
4	202.301	
5	205.338	
6	204.912	
7	220.185	
8	204.344	
9	196.900	
10	200.324	

表 3 采用 64 字节对齐时图像处理的耗时情况

Table 3 Time-consuming of image processing with 64 byte alignment

test number	time consuming/ms	average time-consuming/ms
1	50.372	47.709
2	45.678	
3	46.772	
4	47.834	
5	43.921	
6	47.782	
7	54.857	
8	45.572	
9	46.001	
10	48.301	

表 4 采用 64 字节对齐每次加载 16 行数据时图像处理的耗时情况

Table 4 Time-consuming of image processing by using 64 byte alignment to load 16 lines of data each time

test number	time consuming/ms	average time-consuming/ms
1	52.422	44.694
2	41.335	
3	43.056	
4	44.903	
5	52.422	
6	44.455	
7	39.331	
8	50.137	
9	40.413	
10	40.509	



Fig.7 YUV image of sample
图 7 样本 YUV 图像

对样本 YUV 图像采用本文提出的自适应对齐方式选择和块状化图像处理的方法进行 8 倍缩小处理,缩小后的图像用 pyuv 软件查看,如图 8 所示。

对样本 YUV 图像采用传统的近邻插值法进行 8 倍缩小处理,缩小后的图像用 pyuv 软件查看,如图 9 所示。

由图 8 可以看出,缩小后的图像相比原图依然有较高的清晰度,能够成功对不是 64 字节对齐的分辨力的 YUV 图像进行缩小处理,验证了本文提出的基于 ARM NEON 的静态 YUV 图像缩小技术是可行的。通过图 8 和图 9 的对比可以看出,本文提出的自适应对齐方式选择和块状化图像处理的方法,进行图像缩小后的效果要优于传统近邻插值法的效果,验证了本文提出的基于 ARM NEON 的静态 YUV 图像缩小技术处理质量高的优点。

3 结论

通过大量实验测试验证了本文提出的自适应对齐方式选择和块状化图像处理的方法的可行性,能够在对图像分辨力要求降低到 8 字节对齐的情况下,依然能保持在处理时用 64 字节对齐进行图像处理来达到最高处理效率,并且扩大了应用范围,增强应用环境的适应性,而且保证缩小后的图像有较高的清晰度。

本文中基于 ARM NEON 的静态 YUV 图像缩小技术有 3 个明显的优点:

- 1) 自适应对齐方式选择能够保证每次从内存中加载数据时让 NEON 寄存器都处于满载状态,以此来获得最高的处理效率。
- 2) 由于不再拘束于图像分辨力要 64 字节对齐,降至 8 字节对齐,这样大大增加了满足条件的图像,扩大了应用范围。
- 3) 经过图像的块状化平滑处理,缩小后的图像相比于邻近插值法,清晰度有了明显提高,特别是当缩小倍数比较大时,效果越明显。

但也存在一些不足的地方:

- 1) 如果图像不能 64 字节对齐,要进行特殊处理时,受处理方式的影响,缩小后的图像的右边轮廓可能会出现微小的失真。
- 2) 缩小后的图像有较高的清晰度,但不能做到高保真。特别是缩小 16 倍时,由于兼顾处理效率和图像质量,会在块状处理时对行处理进行选取奇数行处理,从而影响了缩小后的图像质量。

参考文献:

- [1] 张赫龙. 基于 ARM NEON 与 Render Script 的图像处理算法研究[D]. 长春:吉林大学, 2015. (ZHANG Helong. Based on ARM NEON and render script image processing algorithms[D]. Changchun, China: Jilin University, 2015.)
- [2] CÂMARA D, GOUVÊA CPL, LÓPEZ J. Fast software polynomial multiplication on ARM processors using the NEON engine[C]// International Conference on Availability, Reliability, and Security. Springer Berlin Heidelberg: [s.n.], 2013: 137–154.
- [3] 向方明, 朱遵义, 许敬, 等. YUV 到 RGB 颜色空间转换算法研究[J]. 现代电子技术, 2012, 35(22): 65–68. (XIANG Fangming, ZHU Zunyi, XU Jing, et al. Research on YUV to RGB color space conversion algorithm[J]. Modern Electronics Technique, 2012, 35(22): 65–68.)
- [4] NOBUHARA H, HIROTA K, MARTINO FD, et al. Fuzzy relation equations for compression/decompression processes of color images in the RGB and YUV color spaces[J]. Fuzzy Optimization and Decision Making, 2005, 4(3): 235–246.
- [5] 张鹏, 陶青川, 黄凤英. 基于 YUV 色彩空间的暗原色单幅图像去雾改进算法[J]. 太赫兹科学与电子信息学报, 2016, 14(2): 287–292. (ZHANG Peng, TAO Qingchuan, HUANG Fengying. An improved de-fog algorithm for single-image of dark colors based on YUV color space[J]. Journal of Terahertz Science and Electronic Information Technology, 2016, 14(2): 287–292.)
- [6] 刘云邻, 王树东. 基于 SSE2 的 YUV 与 RGB 色彩空间转换[J]. 中国图象图形学报, 2010, 15(1): 45–49. (LIU Yunlin, WANG Shudong. Transition of YUV and RGB color space based on SSE2[J]. Journal of Image and Graphics, 2010, 15(1):



Fig.8 Reduced YUV image according to the method proposed in this paper

图 8 按本文提出的方法进行缩小后的 YUV 图像



Fig.9 Reduced YUV image by the nearest neighbor interpolation method

图 9 按近邻插值法进行缩小后的 YUV 图像

45-49.)

- [7] 陈新泽,杨斌. ARM Cortex-A9 的 NEON 技术研究及应用[J]. 单片机与嵌入式系统应用, 2013,13(10):46-49. (CHEN Xinze,YANG Bin. Research and application of NEON technology based on ARM cortex-A9[J]. Microcontroller & Embedded Systems Applications, 2013,13(10): 46-49.)
- [8] CINAR O. Profiling and NEON optimization[M]. Berkeley:Apress, 2012.

作者简介:



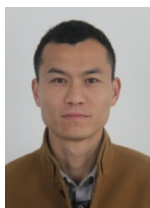
陈 益(1992-), 男, 浙江省台州市人, 在读硕士研究生, 主要研究方向为嵌入式软件开发.email:654408332@qq.com.

李文钧(1977-), 男, 杭州市人, 博士, 副教授, 主要研究方向为集成电路设计与物联网技术.

(上接第 311 页)

- [6] NEAGOE V E,ROPOT A D. Concurrent self-organizing maps for pattern classification[C]// IEEE International Conference on Cognitive Informatics. Calgary,Alberta,Canada:IEEE, 2012:304-312.
- [7] MENZE B H,JAKAB A,BAUER S,et al. The multimodal brain tumor image segmentation benchmark(BRATS)[J]. IEEE Transactions on Medical Imaging, 2015,34(10):1993-2024.
- [8] HAMAMCI A,KUCUK N,KARAMAN K,et al. Tumor-cut:segmentation of brain tumors on contrast enhanced MR images for radiosurgery applications[J]. IEEE Transactions on Medical Imaging, 2012,31(3):790-804.

作者简介:



王 磊(1982-), 男, 陕西省安康市人, 讲师, 主要从事电路与系统设计及图像处理研究工作.email:wanglei_akxy@163.com.