

AMOLED 的图像缩放及时序控制方案

尹 盛, 夏淑淳, 陈 杰

(华中科技大学 电子科学与技术系, 湖北 武汉 430074, E-mail: yinhust@hust.edu.cn)

摘 要: 为研制中的分辨率为 $640 \times \text{RGB} \times 480$ 的有源矩阵有机发光显示器 (AMOLED) 设计了图像缩放和时序控制方案。为了达到良好的显示效果且同时节省硬件资源, 水平缩放采用基于边缘的插值算法, 垂直缩放采用双线性插值算法; 此外, 利用图像缩小后产生的“行空白”时间增加写屏时间, 避免了片外存储器的使用。FPGA 验证结果表明, 该方案可以实现 64 级灰度的 VGA AMOLED 的动态视频显示。

关 键 词: 图像缩放; AMOLED; FPGA

中图分类号: TN383

文献标识码: A

DOI: 10.3788/YJYXS20112603.0334

Image Scaling and Timing Control Scheme for AMOLED

YIN Sheng, XIA Shu-chun, CHEN Jie

(Department of Electronic Science and Technology, Huazhong University of Science and Technology, Wuhan 430074, China, E-mail: yinhust@hust.edu.cn)

Abstract: A image scaling and timing control scheme based on active matrix organic light emitting display (AMOLED) with $\text{RGB} \times 640 \times 480$ resolution (VGA) is proposed in this letter. To achieve a high quality image and fewer hardware resources, the edge-based algorithm was adopted in horizontal interpolation and bilinear algorithm in vertical interpolation; Without off-chip memory, the time for writing panel was extended by ‘horizontal black time’ which stems from image scaling down. Experiments based on FPGA effectively demonstrate that the scheme can realize the dynamic video display on VGA AMOLED with 64 gray scales.

Key words: image scaling; AMOLED; FPGA

1 引 言

基于 TFT 的像素驱动单元一直是 AMOLED 面板研制的关键, 其主要制备材料有非晶硅、多晶硅和微晶硅。非晶硅 TFT 易于大面积淀积, 工艺成熟且成本低廉, 但载流子迁移率较低, 且存在阈值漂移现象; 多晶硅的载流子迁移率高, 易于集成周边驱动, 但淀积均匀性差, 不适用于大尺寸面板的制备。微晶硅 TFT 结合了非晶硅和多晶硅

的优点, 具有良好的淀积均匀性、适中的载流子迁移率和稳定的电特性^[1], 因而其在 AMOLED 中的应用成为了研究的热点。

本文中的微晶硅 TFT OLED 面板通过电压型驱动芯片进行模拟驱动。受工艺条件的限制, 该面板主要存在两个问题: 一是屏幕物理分辨率为 $\text{RGB} \times 640 \times 480$, 低于 PC 显卡支持的最小分辨率, 不能正确显示输入图像; 二是面板的微晶硅 TFT 的迁移率为 $0.6 \sim 3 \text{ cm}^2/\text{V} \cdot \text{s}$ 左右, 开启速度较

收稿日期: 2010-12-24; 修订日期: 2011-02-22

基金项目: 国家“863”计划资助项目 (No. 2008AA03A336)

作者简介: 尹盛 (1960—), 男, 山东济阳人, 博士, 副教授, 主要从事等离子体与物质的相互作用及有机电致发光器件方面的研究。

慢,直接采用源图像的行使能信号可能导致存储电容充电不完全,影响图像显示效果甚至无法显示。

针对上述问题,本文拟定了如下解决方案:先用基于边缘的插值算法对输入图像进行水平缩放,再用双线性插值算法进行垂直缩放,得到目标图像;然后,利用片内存储器及源图像与目标图像之间的“行空白”时间,修改控制电路时序,增加每一行的写屏时间。时序仿真及FPGA的验证结果表明,这种方法切实可行,且硬件开销较小。

2 图像缩放算法的选取及实现

基于边缘的图像缩放算法图像处理效果较好,但是计算电路复杂,耗时长,占用硬件资源多;基于插值的算法形成图像质量略差,但计算电路简单,耗时短,易于硬件实现^[2-4]。本文采用的基于边缘的插值算法对于图像边缘的处理效果要优于传统的插值算法^[5],但其耗费的资源仍然十分可观。因此我们将它与对硬件要求较小的双线性插值算法结合起来,在水平方向采用基于边缘的插值算法,在垂直方向采用双线性插值算法(以下简称混合插值算法),在达到了良好显示效果的同时,大幅减少了片内乘法器与存储单元的应用。

2.1 混合插值算法原理

行列采用不同算法的混合插值算法原理如图1所示。其中, $a_0 \sim a_3$ 、 $b_0 \sim b_3$ 为源像素点,它们在同一方向上等距,可将此距离视为1; p_0 、 p_1 为水平方向插值得到的像素; p 为目标像素,其灰度

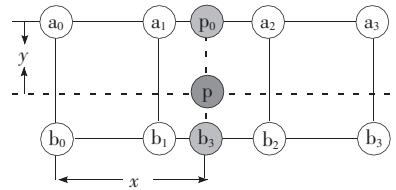


图1 混合插值算法原理图

Fig. 1 Principle diagram of mixed interpolation

值与相邻8个源像素点有关; x 为 b_0 至 p_1 的距离, $1 \leq x \leq 2$; y 为 p_0 至 p 的距离, $0 \leq y \leq 1$ 。

基于边缘的插值算法通过比较源图像数据在同一方向的二阶差商,决定像素点处于同一图像区域的可能性大小,然后用二阶牛顿插值函数计算目标像素灰度。将其作为水平缩放算法得到 p_0 的灰度值,如式(1)~(5)所示。 p_1 与 p_0 的公式相同,只需将源像素点换成 $b_0 \sim b_3$ 。显然,用基于边缘的插值算法完成水平方向的插值运算需要3次乘法和6次加法。

双线性插值算法以源像素点的灰度均匀变化为条件,以之作为垂直缩放算法可得到 p 的灰度,如式(6)所示。式(6)和式(7)是相同的,但采用式(7)来完成垂直方向的插值计算,只需1次乘法和2次加法。

$$\Delta a_0 = a_1 - a_0 \quad (1)$$

$$\Delta a_1 = a_2 - a_1 \quad (2)$$

$$\Delta^2 a_0 = a_2 - 2a_1 + a_0 \quad (3)$$

$$\Delta^2 a_1 = a_3 - 2a_2 + a_1 \quad (4)$$

$$p_0 = \begin{cases} a_0 + \Delta a_0 x + \frac{\Delta^2 a_0}{2!} x(x-1) & |\Delta^2 a_0| \leq |\Delta a_1| \\ a_1 + \Delta a_1(x-1) + \frac{\Delta^2 a_1}{2!} (x-1)(x-2) & |\Delta^2 a_0| > |\Delta^2 a_1| \end{cases} \quad (5)$$

$$p = p_0(1-y) + p_1 y \quad (6)$$

$$p = (p_1 - p_0)y + p_0 \quad (7)$$

2.2 算法的软件验证

将双线性插值算法、基于边缘的插值算法及行列采用不同算法的混合插值算法在Matlab环境中编程实现,通过对比图像边缘区域的主观视觉效果及峰值信噪比,评定混合插值算法的意义。

图2(a)是分辨率为 512×512 的peppers测试图像,黑框中的图像为 120×120 ,我们将它作为源图像;图2(b)~(d)是采用不同算法将原图

像放大为 512×512 的图像。可以看到,图2(b)中,两个pepper的交界处的轮廓较清晰;图2(d)与图2(b)的主观视觉效果类似;图2(c)的边缘处有较长的锯齿,整体图像出现了模糊现象。

峰值信噪比PSNR是评价图像缩放算法的重要指标之一^[6],其计算公式如式(8)所示。其中, S 为源图像, D 为源图像由某一算法放大后,再经相应算法缩小到源图像分辨率所得到的目标图像, n_{rows} 为源图像的行数, n_{cols} 为源图像的列数。经过计算,基于边缘的插值算法、混合插值算法、

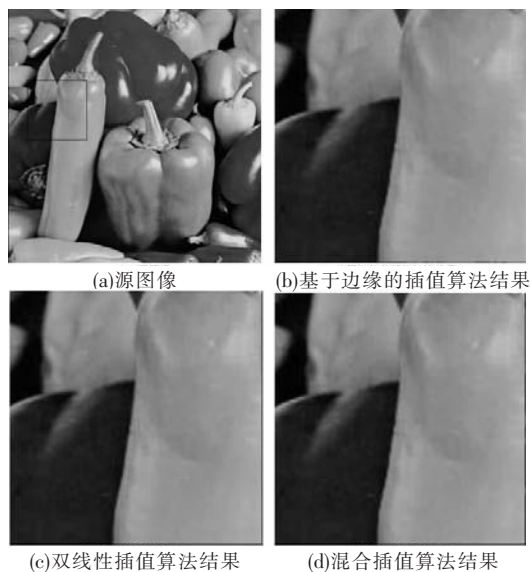


图 2 不同插值算法图像效果

Fig. 2 Visual effect of resizing image by different interpolation algorithm

双线性插值法的峰值信噪比分别为 49.585 3, 49.030 1, 47.938 5 dB。混合插值算法的 PSNR 较基于边缘的插值算法低,但是仍然高于双线性插值算法。因此,选择混合插值算法进行图像缩放处理是有意义的。

$$K_{\text{PSNR}} = 10 \lg \frac{255^2}{\frac{1}{n_{\text{rows}} \cdot n_{\text{cols}}} \sum_{i=1}^{n_{\text{rows}}} \sum_{j=1}^{n_{\text{cols}}} [D(i,j) - S(i,j)]^2} \quad (8)$$

2.3 算法的硬件实现及验证

基于算法硬件实现的实时处理要求,我们采用水平缩放和垂直缩放分开的通用缩放引擎架构,如图 3 所示。

由于缩放比例 λ 一般都是浮点数,若在 FPGA 中设计的专门的浮点运算器,不仅电路结构复杂,结果的实时性也难以保证^[7]。对此,通常的做法

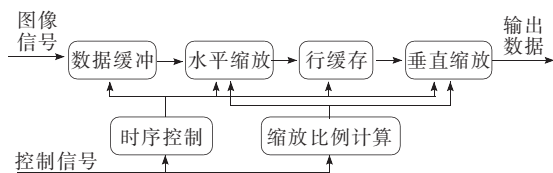


图 3 图像缩放引擎架构

Fig. 3 Architecture of image scaling engine system

是将 λ 放大一定倍数后取整,再代入插值公式中进行运算,然后将运算结果除以相应的倍数以得到最终结果。本文中取 λ 的放大倍数为 2^6 ,假设 $\lambda=1.6$,则放大后为 $\lambda \times 2^6 \approx 01100110$ (b)。为保证算法的精度,放大后的计算结果并未除以相应倍数,而是直接进入下级运算,直到行列缩放全部完成后,再取运算结果的高 6 位作为最终结果。

基于边缘的插值算法的实现需要 3 次乘法,6 次加法,而双线性插值算法只需 1 次乘法,2 次加法,对于 RGB 彩色图像,这个数目还要乘以 3。因此,以双线性插值算法作垂直缩放,可以大幅减少乘法器的使用。此外,由于基于边缘的插值算法每次需要 4 个源像素点进行运算,若以之作为垂直缩放算法,需要 4 行的缓存空间;双线性插值算法每次需 2 个源像素点进行计算,通过对行缓存的读写使能信号进行控制,以之作为垂直缩放算法,需要的缓存空间仅为 1 行,故其对 FPGA 内部 M4K 存储单元的节省量高达 75%。由此可见,与基于边缘的插值算法相比,混合插值算法的硬件开销要小得多。

图 4 是使用混合插值算法的图像缩放模块的时序仿真波形图,其中 DE 是输入数据行使能,ODCK 是输入时钟;rowen 是输出数据的行使能,wren 为输出使能。这里的源图像分辨率为 800×600 ,行缩放比为 1.25,因此每 5 行输入数据只对应着 4 行输出数据,有 1 行的空白时间。

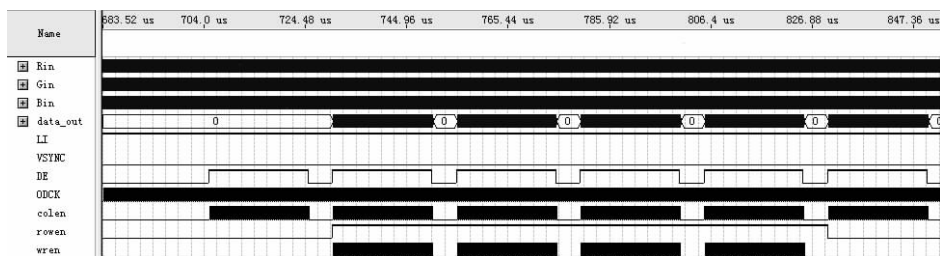


图 4 图像缩放模块仿真波形

Fig. 4 Simulation waveforms of image scaling module

3 写屏时序控制

要得到与屏幕分辨率一致的目标图像,还必须将像素数据以适当的时序输出至面板,才能进行显示。AMOLED 面板工作原理与 AMLCD 显示屏类似,只是采用了 2 管 TFT 作为像素驱动^[8]。面板外围绑定了行列驱动芯片,当行驱动芯片将每行的 TFT 打开时,列驱动芯片将输入的显示数据转换为相应的灰度电压,对显示屏存储电容进行充电,面板的每一行的充电时间便是写屏时间,亦即行时间。

由于本项目中的微晶硅 TFT 的迁移率较低,瞬态响应较慢,为了保证存储电容的充电时间,写屏时间必须大于 $32.6 \mu\text{s}$ 。而来自 PC 的图像的分辨率一般为 $800 \times 600 @ 60\text{Hz}$ 以上,由 VESA 标准可知,其行时间皆小于等于 $26.4 \mu\text{s}$,若将源图像转换为 640×480 后直接输出给面板显示,显然不能满足写屏时间的要求。

3.1 写屏时序控制方案

要增加写屏时间,最直接的方法是用两个 SRAM 存储每一帧的输出图像,利用“Ping-Pong”轮换机制实现图像数据的缓冲^[9],在从一个 SRAM 中读取上一帧数据的同时,将当前帧的数据写入另一个 SRAM。通过控制读 SRAM 的速率,可使写屏时间等于 $\frac{1}{60 \times 480} \approx 34.7 \mu\text{s}$ 。然而,对于 64 级灰度的彩色屏,每帧的数据量达 5.27Mbits ,若在 FPGA 中同时构建这样 2 个 SRAM,硬件开销过大;而若使用片外的高速异步双端口 SRAM,会增加 PCB 板的复杂度,影响到电路的稳定性。

由于本项目中只需要对图像进行缩小,缩放比例总是大于 1 的,因此源图像和目标图像的输出之间存在着“行空白”时间,可以利用它来增加输出的行时间。其原理如图 5 所示,假设行缩放比例为 1.25,DE 为源图像的行使能,周期为 T_{Hsync} ;rowen 为采用源图像时钟信号的目标图像的行使能;rde 为修改后的目标图像的行使能,其周期 T 就是面板的写屏时间。

假定源图像分辨率为 $3 \times n_{\text{cols}} \times n_{\text{rows}}$,目标图像分辨率为 $3 \times c \times r$,灰度级为 2^l ,行缩放比例为 $\eta = n_{\text{rows}}/r = m/n$,其中 m, n 为互质整数,例如 $\eta = 1.6$,则 $m=8, n=5$ 。由公式(9)可知,只要 $n_{\text{rows}} > r$,

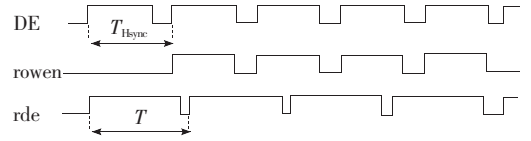


图5 时序控制信号对比图

Fig. 5 Waveforms comparison among timing control signals

便有 $T > T_{\text{Hsync}}$,写屏时间被延长了。

$$T = \frac{n_{\text{rows}}}{r} T_{\text{Hsync}} \quad (9)$$

3.2 控制电路的硬件实现和验证

要利用行空白时间来增加写屏时间,需要在片内构建双时钟异步 FIFO 或 SRAM,将输入和输出数据进行时钟域的转换^[10]。存储器的容量 s 和读写速率的关系如式(10)所示,其中 V_w, t_w 分别表示向存储器写入数据的速率和时间, V_r, t_r 分别表示从存储器读取数据的速率和时间, u 为已被使用的存储单元, $u \geq 0$ 。由于本设计要求在 m 行的时间内读出 n 行数据,因此写入速率和时间满足式(11),将其代入式(10),得到公式(13)。

$$p \leq V_w t_w - V_r t_r + u \leq s \quad (10)$$

$$\left. \begin{aligned} V_w &= \frac{3lnc}{t_{\text{Hsync}} m} \\ t_w &= T_{\text{Hsync}} m \end{aligned} \right\} \quad (11)$$

$$3lnc + u - s \leq V_r t_r \leq 3lnc + u \quad (13)$$

当 $u=0, s=3lnc$ 时,有

$$0 \leq V_r t_r \leq V_w t_w \quad (14)$$

在存储器的写入速率大于读取速率的情况下,只要 $t_r \leq t_w$,式(14)总是成立的。因此, $3lnc$ 是所需要的最大存储器容量。

选取几个常用的图像分辨率,依照公式(9)、(13),计算其缩小至 $640 \times 480 @ 60 \text{Hz}$ 后,可达到的写屏时间及需要的存储容量,如表 1 所示,这里假定灰度级为 64。可以看到,在这几个分辨率下,利用行空白时间可将写屏时间增加到 $33 \mu\text{s}$ 以上,而所需的片内存储器仅为几十 kbit,比用“Ping-Pong”结构进行帧存储所需要的容量小得多。因此这种行存储方法具有一定的通用性,可以和图像缩放算法相结合,实现微晶硅 AMOLED 面板的显示。

表 1 不同分辨率下的写屏时间和存储器容量

Table 1 Time for writing to panel and storage capacity needed in different resolutions

源图像分辨率	行缩放比例	源图像行时间/ μs	最大写屏时间/ μs	最大存储容量/kbit
800×600@60 Hz	1.25	26.400	33	45
1 024×768@60 Hz	1.6	20.667	33.067 2	56.25
1 280×960@60 Hz	2	16.667	33.334	22.5
1 400×900@60 Hz	1.875	17.878	33.344	90

4 FPGA 验证结果

将整个图像处理模块以 Verilog 硬件语言进行描述,在 Quartus II 中综合出硬件代码并仿真。综合结果显示,模块共使用了 835 个逻辑单元,16 个 9 bits 嵌入式乘法器,47 744 个存储单元。

由于微晶硅 AMOLED 面板尚在研制中,因此我们将程序下载到 FPGA 后,通过示波器观测

控制信号的波形。假设输入图像为 800×600@60 Hz,示波器的通道 1 为输出图像的行使能信号,通道 2 为输入图像的行使能信号。如图 6 所示,在 20 MHz 的输出时钟下,每一行的写屏时间约为 33 μs ,与设计要求相符合。

5 结 论

针对研制中的微晶硅 TFT AMOLED,设计了一种混合插值图像缩放算法,水平方向采用基于边缘的插值算法,垂直方向采用双线性插值算法。该算法能节省 75% 的行缓存容量,且图像处理效果较好,可解决所研制的 AMOLED 分辨率过低的问题。此外,提出了利用图像缩小产生的“行空白”时间来增加写屏时间的时序控制方案,提高了 AMOLED 面板的灰度精确性,避免了片外 SRAM 的使用。Matlab 和 FPGA 的验证结果表明,该方案可以实现分辨率为 640×480 的 AMOLED 的动态视频显示。

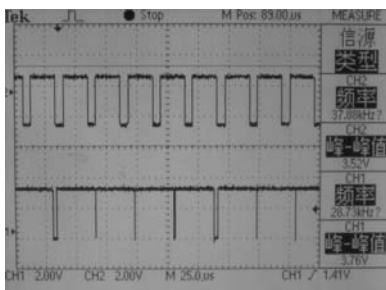


图 6 示波器实测信号波形

Fig. 6 Waveforms observed in oscilloscope

参 考 文 献:

- [1] Chan K Y, Bunte E, Stiebig H, et al. High mobility microcrystalline silicon thin-film transistors [J]. *Chin. J. Electron Devices*, 2008, 31(1): 109-114.
- [2] 林媛. 图像缩放算法研究及其 FPGA 实现[D]. 厦门: 厦门大学硕士学位论文, 2006.
- [3] 胡小龙, 冯彬. 基于 FPGA 的高分辨率实时监控图像缩放设计 [J]. *液晶与显示*, 2009, 24(6): 882-884.
- [4] 孙红进. FPGA 实现的视频图像缩放显示 [J]. *液晶与显示*, 2010, 25(1): 130-133.
- [5] 刘政林, 肖建平, 邹雪城, 等. 基于边缘的实时图像缩放算法研究 [J]. *中国图象图形学报*, 2008, 13(2): 225-229.
- [6] Han J K, Back S U. Parametric cubic convolution scaler for enlargement and reduction of image [J]. *Consumer Electronics, IEEE Transactions on Consumer Electronics*, 2000, 46(2): 247-256.
- [7] Beauchamp M J, Hauck S, Underwood K D, et al. Architectural modifications to enhance the floating-point performance of FPGAs [J]. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2008, 16(2): 177-185.
- [8] 尹盛, 刘卫忠, 刘陈, 等. 有机电致发光器件的驱动技术 [J]. *液晶与显示*, 2003, 18(2): 106-111.
- [9] 沈亮, 尹盛, 张繁, 等. 12.7 cm 彩色 AM-OLED 显示器分场驱动研究 [J]. *光电子激光*, 2006, 17(10): 1173-1176.
- [10] 乔庐峰. Verilog HDL 数字系统设计与验证 [M]. 北京: 电子工业出版社, 2009: 139-140.