

文章编号:1007-2780(2009)05-0756-06

适用于小尺寸 TFT-LCD 的图像锐化算法及其硬件实现

郭振华,魏廷存

(西北工业大学 计算机学院,陕西 西安 710072,E-mail: gocomputer@163.com)

摘 要: 针对小尺寸 TFT-LCD 驱动控制芯片的结构特点,提出了一种适用于小尺寸 TFT-LCD 的新型图像锐化算法。在传统的反锐化掩膜算法中考虑了人眼的视觉系统特性,加入了能量色散滤波器,进而提高了图像锐化效果,降低了锐化噪声,减小了灰度溢出,同时也避免了色彩失真现象。所提出的算法简单且易于硬件实现,硬件开销小,还可以通过寄存器配置锐化强度等参数,以实现显示画质的优化和微调,尤其适应于手机等小尺寸 TFT-LCD 显示终端应用。

关 键 词: 反锐化掩模算法;图像锐化;TFT-LCD

中图分类号: TN911.73 **文献标识码:** A

1 引 言

小尺寸 TFT-LCD 具有功耗低、画质高、响应速度快等优点,目前被广泛应用手机、PDA 等便携式电子产品中。随着显示屏的尺寸、分辨率以及显示颜色数的不断增加,人们对显示效果有了更高的要求,因此,各种图像显示处理技术开始应用于小尺寸 TFT-LCD 显示中。

通过采用图像锐化技术,可以使原始图像更清晰、图像细节更丰富。图像锐化的主要目的在于补偿图像轮廓、突出图像的边缘信息以使图像显得更为清晰,从而进一步提高显示质量。图像锐化的实质是增强原图像的高频分量。在大尺寸 TFT-LCD 显示设备上,由于其硬件资源丰富,具有专用的图像处理 DSP,并且对功耗的要求不是很高,所以可以使用复杂的高通滤波器,通过大量的运算得到很好的锐化效果。然而,由于小尺寸 TFT-LCD 显示系统对驱动控制芯片的尺寸和功耗都有着严格的限制,所以只能对现有的图像锐化方法进行改进和优化,研究一种适应于小尺寸 TFT-LCD 驱动控制芯片的图像锐化技术。

本文通过对现有图像锐化算法的优化和改进,提出了针对自主研发的手机用 TFT-LCD 驱动控制芯片——“龙腾 T4”的图像锐化算法。与

现有的图像锐化算法相比,该算法实现简单,可嵌入单片集成 TFT-LCD 驱动控制芯片内,同时该算法较好地克服了普通反锐化掩膜算法的锐化噪声、灰度溢出和色彩失真等缺点。

2 传统图像锐化算法及其缺陷

反锐化掩膜方法是最常用的一种图像锐化算法^[1],这种算法的实质是通过使用高通滤波器来增加输入图像的高频成分,从而达到锐化图像的效果。在传统的线性反锐化掩膜算法中,锐化的图像是通过公式(1)得到的:

$$F(m,n) = X(m,n) + \lambda \cdot Z(m,n) \quad (1)$$

其中 $F(m,n)$ 为经过图像锐化算法处理后的图像灰度值, $X(m,n)$ 为原始图像的灰度值, λ 是一个可变的因子, $Z(m,n)$ 为高通滤波后的图像灰度值。 λ 的值越大,图像锐化程度越高;反之,锐化程度越低。

尽管这种算法可以达到锐化图像的目的,但是却存在如下缺点:首先,使用高通滤波器在增强图像细节的同时在很大程度上放大了图像中的噪声;其次,对于图像中的高对比度区域,通过高通滤波后图像中包含有极大灰度值的像素点。当把滤波后产生的图像与原来的图像相加时,一些像素点的灰度值会溢出,即超出了灰度值的表示范

围,从而导致在图像的边缘处出现白色线条;另外,反锐化掩膜方法对彩色图像中的红、绿、蓝 3 种颜色分别实行锐化,会导致锐化后的图像出现色彩失真。

3 针对小尺寸 TFT-LCD 驱动芯片的锐化算法

图 1 所示为单片集成 TFT-LCD 驱动控制电路芯片的结构框图,主要由数据输入接口、数据输出接口、时序控制模块、数据处理模块以及内置 GRAM 构成。其中,数据输入接口模块主要负责接收系统传来的图像显示数据,并送给后续电路进行数据处理和存储;数据输出接口由 Gate Driver 和 Sorce Driver 组成,主要负责将存储在 GRAM 中的图像数据逐行取出,并转换成模拟信号后驱动 TFT-LCD 显示彩色图像;数据处理模块主要完成一些数据格式的调整等功能,然后将数据保存在图像存储器 GRAM 中;时序控制模块用来生成芯片内部所需时序信号,协调所有模块正常工作。

图 1 单片集成 TFT-LCD 驱动控制芯片的结构
Fig. 1 Architecture of one-chip TFT-LCD driver IC

所研究的图像锐化模块位于传统单片集成 TFT-LCD 驱动控制电路芯片的数据输入接口和数据处理模块之间。该模块主要负责将接收到的 RGB 原始数据使用锐化算法进行处理,并将处理后的数据传给后续模块。

从图 1 可以看出,图像锐化模块位于数据输入接口和数据处理模块之间。为了尽量减少加入图像锐化模块后所引起的对其它模块的修改,同时为了避免使用大量的存储器资源,该模块应相对于芯片中其它的模块更加透明,而且必须具有实时处理数据的能力。

由于在输入 RGB 数据时是将一幅图像中的数据逐行逐点串行输入的,所以为了避免图像锐化模块使用大量的存储器资源,并且保证锐化电

路具有实时处理数据的速度和方便后续模块继续处理图像数据,需要尽量简化锐化算法,即需要开发一种针对单片集成 TFT-LCD 驱动控制芯片的锐化算法。

3.1 改进的高通滤波器

为了减小在采用反锐化掩膜方法时对噪声的敏感程度,文献[2]和[3]中采用了非线性滤波器来得到图像的高频成分。这种滤波器使用图像中某一局部区域的平均亮度作为权值,成功地解决了图像锐化时产生的噪声。然而,该算法需要经过复杂的运算,对硬件的运算速度和存储量都有较高的要求。例如,上述算法采用像素点周围 8 个点的灰度均值作为滤波器权值,对 320×432 的图像进行锐化时,需要 23 040 位的存储器。可见,该方法并不适应于小尺寸 TFT-LCD 驱动控制芯片。

由于人眼对于横向图像的特征比较敏感,所以本文算法在计算高通滤波图像时只考虑了左右两侧的 4 个像素点。这样,在改进后的图像锐化模块中,仅需缓存 5 个像素点的 RGB 灰度值。高通滤波后图像的灰度值可使用公式(2)来计算:

$$Z(m,n) = X(m,n) - [X(m,n-1) + X(m,n+1) + X(m,n-2) + X(m,n+2)]/4 \tag{2}$$

其中, $Z(m,n)$ 代表高通滤波后图像的灰度值; $X(m,n-2)$, $X(m,n-1)$, $X(m,n)$, $X(m,n+1)$ 和 $X(m,n+2)$ 分别代表原图像中第 m 行的第 $n-2$, $n-1$, n , $n+1$ 和 $n+2$ 列像素点的灰度值。

经以上改进后的高通滤波算法不仅减小了对存储空间的需求,而且大幅降低了计算量。大量的实际图像验证表明,经过以上算法锐化后的图像受噪声影响很小,具有较好的图像质量,对于用于手机等的小尺寸 TFT-LCD 显示屏仍然可以获得满意的图像效果。

3.2 减少灰度溢出现象

设计中考虑了人眼视觉系统(Human Visual System:HSV)的特性,以使整幅图像得到更好的锐化效果。图 2 所示为在不同亮度情况下的人眼敏感度变化曲线,横轴代表图像中局部区域的平均亮度值,竖轴代表人眼可以感觉到的亮度变化阈值。图 2 表明,人眼对于一幅图像的不同亮度区域亮度值的变化有着不同的敏感程度,相比于高亮度和低亮度区域,人眼对于中间亮度区域灰度值的变化更加敏感^[4,5]。在图像的高、低亮度区

图 2 在不同亮度情况下人眼敏感度变化曲线
Fig. 2 Perceptivity of human eyes as a function of gray level of the background

域,人眼很难察觉到较小的灰度值变化,需要使用较大的锐化因子 λ 来得到明显的锐化效果;相反,由于人眼对图像的中等亮度区域灰度值变化较为敏感,所以在这种区域需要采用较小的锐化因子,以避免放大图像噪声。通过大量实验,本文采用公式(3)来计算锐化因子 λ :

$$\lambda(x)=\begin{cases} \left[\frac{5}{12} \left[\frac{10}{113^2} (x-113)^2 + 1 \right] \right] & \text{当 } x < 127 \text{ 时} \\ \frac{1}{219} (x-255) + 1 & \text{当 } x \geq 127 \text{ 时} \end{cases} \quad (3)$$

其中 x 为原始输入图像的灰度值。

通过采用上述计算锐化因子的方法可以使整幅图像得到很好的锐化效果。但是这样锐化后的图像会在图像中产生明显的亮边效应,这是由于像素点灰度值溢出造成的。

文献[6]中为了减少像素点的灰度值溢出,首先将一幅图像分成高、中、低 3 种对比度区域,然后使用不同的缩放因子作用于高通滤波后的图像,最后把新的高通滤波图像与原图像相加得到锐化后的图像。然而,这样做的效果并不理想,尤其对于图像的高对比度区域锐化效果不理想。这是由于这种算法虽然在不同图像区域采用了不同的缩放因子,但是这些因子却是固定不变的,不能根据图像特征自动调整。

为了更好地解决像素点灰度值溢出的问题,同时考虑到在单片集成 TFT-LCD 驱动芯片中该模块受到运算速度和存储空间的限制,需要一种简单有效的自适应缩放因子生成算法来解决这个问题,所以,本文提出了一种新的自适应对比度增强方法。在这种方法中,高通滤波器的因子是由

图像区域特性来自动得出的。

下面给出 5 种不同的缩放因子,分别针对灰度变化从平坦到急剧的 5 种图像区域。缩放因子 β 由公式(5)计算得出。

$$V = (|Z_R|, |Z_G|, |Z_B|)_{\max} \quad (4)$$

$$\beta = \begin{cases} 1 & V < 20 \\ 1/2 & 20 \leq V < 40 \\ 1/4 & 40 \leq V < 80 \\ 1/8 & 80 \leq V < 160 \\ 1/16 & V \geq 160 \\ 0 & V < T_{\text{LOW}} \text{ 或 } V > T_{\text{HIGH}} \end{cases} \quad (5)$$

这里 Z_R, Z_G, Z_B 分别代原图像经过高通滤波后的 R、G、B 灰度值。

本算法可以通过配置锐化模块中寄存器 α , T_{LOW} 和 T_{HIGH} 的值来进一步微调图像锐化效果。通过修改 α 寄存器的值,可以进一步微调图像的整体锐化效果;通过修改 T_{LOW} 和 T_{HIGH} 的值,可以调节图像局部锐化效果。

最终得到的锐化因子 $\lambda' = \alpha \cdot \beta \cdot \lambda(x)$ 。实验证明,本文算法不仅很好地解决了像素点灰度值溢出的问题,而且对于锐化噪声也有很大的抑制作用。

3.3 避免色彩失真

图 3 显示了在图像锐化时色彩失真现象的产生原因。为了避免色彩失真现象的发生,采用一个能量色散滤波器对高通滤波后的图像进行处理。整个过程如公式(6)所示:

$$Y(m,n) = \begin{bmatrix} Y_R(m,n) \\ Y_G(m,n) \\ Y_B(m,n) \end{bmatrix} = \frac{1}{9} \begin{bmatrix} 5 & 2 & 2 \\ 2 & 5 & 2 \\ 2 & 2 & 5 \end{bmatrix} \begin{bmatrix} Z'_R(m,n) \\ Z'_G(m,n) \\ Z'_B(m,n) \end{bmatrix} \quad (6)$$

其中, Z'_R, Z'_G, Z'_B 为 Z_R, Z_G, Z_B 与自适应锐化因子相乘后得到的图像像素点的 R、G、B 值, Y_R, Y_G, Y_B 为使用能量色散滤波器处理后的图像像素点的 R、G、B 值。

通过使用能量色散滤波器,可以把锐化图像边缘的子像素的能量按一定比例分配到其它子像素中,如同把一种颜色的图像边缘分配到了其它颜色的图像边缘中,使每一种颜色得到相同强度的边缘增强效果。这样,在将高通滤波图像与原始图像相加时,就不会改变原始图像的色彩平衡,从而可以避免锐化后的图像出现色彩失真的现象。将

图 3(a)和(b)对照可以看出,通过使用能量色散滤波器,减轻了锐化后的图像色彩失真现象。

图 3 色彩失真和能量色散示例图。(a)反锐化掩膜算法中的色彩失真;(b)使用能量色散的方法来解决色彩失真。

Fig. 3 Example of color distortion and energy dispersion. (a) Color distortion in unsharp masked image;(b) Using energy dispersion to remove the color distortion.

3.4 最终的图像锐化算法流程

所提出的整个图像锐化算法流程如图 4 所示,包括高通滤波器、自适应锐化因子产生器和能量色散滤波器。其中, $X(m,n)$ 表示原始输入图像, $Z(m,n)$ 表示经过高通滤波后产生的图像, $Z'(m,n)$ 表示高通滤波产生的图像与自适应锐化因子相乘后得出的图像, $F(m,n)$ 为最终锐化后的

图 4 所提出的图像锐化算法流程
Fig. 4 Flow chart of the proposed image sharpening algorithm

图像,它是由 $Z'(m,n)$ 经过能量色散滤波器处理后的图像和原始图像 $X(m,n)$ 相加得到。

4 仿真结果及硬件实现

实验得出的图像锐化效果如图 5,6 所示。图 5(a)是原始图像,(b)图是没有使用能量色散滤波

图 5 色彩失真现象。(a) 原始图像;(b)没有使用能量色散滤波器的图像锐化效果;(c)本文提出的算法的锐化效果。

Fig. 5 Example of color distortion. (a) Original image; (b) Without using energy dispersion filter; (c) Using energy dispersion filter.

图 6 使用不同的反锐化掩膜算法得到的图像锐化效果。(a),(d) 原始图像;(b),(e) 普通反锐化掩膜算法处理后的效果;(c),(f) 本文提出的算法处理后的效果。

Fig. 6 Results of applying various unsharp masking algorithms. (a),(d) Original image;(b),(e) Results of applying the ordinary unsharp masking algorithm;(c),(f) Results of applying the proposed algorithm.

器得到的图像锐化效果,图 5(c)是使用了能量色散滤波器后的图像锐化效果。通过对比可以明显的看到,图 5(b)中文字的边缘出现了较大的色彩失真现象,而图 5(c)则很好地避免了这种现象。在图 6 中,(a)和(d)为原始图像,(b)和(e)为采用常规反锐化掩膜算法处理后的图像,(c)和(f)为采用本文提出的算法处理后的图像。通过对比图 6(b)和 6(c)的高对比度区域和暗部细节可以看出,本文提出的算法可以实现比常规锐化算法更好的细节锐化效果;通过对比图 6(e)和 6(f)的高亮区域可以看出,采用常规反锐化掩膜算法得到的锐化后图像(图 6(e))中的一些高对比度区域出现了明显的噪声点,而采用本文提出的算法得到的锐化后图像(图 6(f))则很好地避免了锐化噪声现象,可以减少灰度值溢出,降低图像的失真度。

图 7 为图像锐化模块的结构框图,该锐化模块采用 Verilog HDL 编写设计代码及测试代码,使用 Modelsim5.8 进行编译仿真和功能验证,然后使用 Design Compiler 在 faraday 0.18 μm CMOS 标准单元库下对设计进行了综合。电路规模约为 5 000 门,工作速度最高可以达到 100 MHz,完全

图 7 图像锐化模块结构图

Fig. 7 Architecture of the image sharpening block

满足该模块的设计要求,适应于对功耗和芯片面积都有苛刻要求的单片集成 TFT-LCD 驱动控制芯片。

5 结 论

针对小尺寸 TFT-LCD 单片集成驱动控制芯片的特点,对传统算法进行了改进和优化,提出了一种适应于该类芯片的图像锐化算法。与现有的锐化算法相比,该算法具有锐化效果明显、噪声低、无色彩失真等特点,并且算法简单易实现,尤其适应于单片集成 TFT-LCD 驱动控制芯片中使用。

参 考 文 献:

- [1] Messing D S, Daly S. Improved display resolution of subsampled colour images using subpixel addressing [C]// *Proc. IEEE Int. Conf. Image Processing*, New York, USA:IEEE, 2002:625-628.
- [2] Mitra S K, Li H. A new class of nonlinear filters for image enhancement [C]// *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, Toronto:IEEE, 1991:2525-2528.
- [3] Ramponi G, Strobel N, Mitra S K. Nonlinear unsharp masking methods for image contrast enhancement [J]. *Electron. Imag.*, 1996, 5(3):353-366.
- [4] Polesel A, Ramponi G, Mathews V J. Image enhancement via adaptive unsharp masking [J]. *IEEE Trans. on Image Processing*, 2000, 9(3):505-510.
- [5] Chin Y J, Berger T. A software-only video codec using pixelwise conditional differential replenishment and perceptual enhancements [J]. *IEEE Trans. Circuits Syst. Video Technol.*, 1999, 9(3):438-450.
- [6] De Vries F P. Automatic, adaptive, brightness independent contrast enhancement [J]. *IEEE Trans. Signal Processing*, 1990, 21(2):169-182.

摘自：中国科协学会学术部 2002.9 通知