

look up table 方式把曲线存放在内存中, 用的时候直接查表矫正

Gamma 校正的快速算法及其 C 语言实现

曾嘉亮

(汕头职业技术学院机电工程系, 汕头 515078)

摘要: Gamma 校正是数字图像显示前必不可少的操作。若直接套用公式来实现的话, 执行效率非常低; 许多文献都提到可以使用查找表来加速这一操作, 然则语焉不详, 并未具体介绍实现方法。在对 gamma 校正原理进行深入研究的基础上, 构造出 gamma 校正查找表, 并提出了运用该表对数字图像进行快速 gamma 校正的方法。该算法特别适用于在嵌入式系统中处理视频流。

关键词: 显示系统; gamma 校正; 查找表; 快速算法

A fast algorithm of gamma correction and its implementation with C

ZENG Jia-liang

(Department of Mechanical and Electrical Engineering, College of Shantou Polytechnic,
Shantou 515078, China)

Abstract: Gamma correction is a necessary operation for a digital image before it is sent to display. But, if this operation is performed by applying the correction formula mechanically, a poor performance will be got. Many articles mentioned that using a look-up table can accelerate this operation, unfortunately, that how to do this is not spoken in detail. Basing on the understanding of the theory of gamma correction, it presents a technique for building a gamma correction look-up table, finally gives a fast algorithm for performing gamma correction by using the look-up table. For the embedded systems which are forced into processing video stream, this algorithm fits the bill.

Key words: display system; gamma correction; look-up table; fast algorithm

1 Gamma 校正的原理

理想的显示系统是线性的, 换句话说, 显示系统输出的光强度应该与输入的视频信号幅度成正比。

设显示系统的输出光强度为 L_{out} , 输入值为 L_{in} , 对于线性的显示系统而言, 存在如下公式:

$$L_{out} = L_{in} \quad (1)$$

图 1 显示了线性显示系统输出、输入信号之间的函数关系。不难看出, 线性系统的输出真实地反映出了原始图像各个像素的亮度情况。

然而, 现实中的视频显示器却是非线性的。例如, 目前在计算机领域居于主流地位的阴极射线管显示器(以下简称 CRT), 就是非线性的显示系统; 也就是说, CRT 的输出值与输入值并不成正比。

事实上, CRT 的输出与输入成指数函数的关系。如图 2 所示: 由于硬件特性指数 gamma 的存在(注意: $\gamma > 1$), CRT 的输出值与输入值之间呈现出

以 gamma 为指数的函数关系:

$$L_{out} = L_{in}^{\gamma} \quad (2)$$

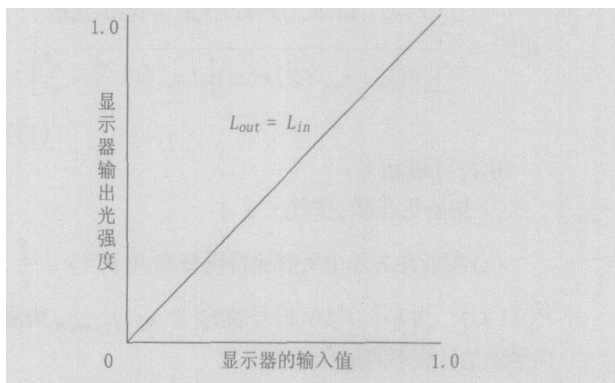


图 1 线性显示系统

收稿日期: 2005-09-26

作者简介: 曾嘉亮, 男, 电子科技大学, 主要研究方向为数字图像压缩与处理技术。

对比图1和图2,可以看到,现实中的显示器总是存在非线性,正是这种非线性使得显示器的输出较其原始图像产生了失真,失真程度由具体系统的gamma值决定。

这种失真是由硬件的特性所导致,要想校正它,一般考虑从软件方面着手。传统的gamma校正方法是先对图像数据进行预补偿,再送入CRT显示。原理如下:

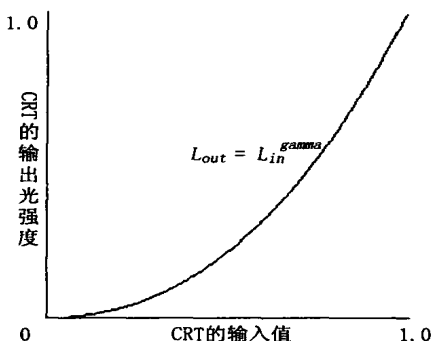


图2 非线性的CRT

假设图像数据的原始值为 L_{org} , 在将 L_{org} 送给 CRT 显示之前, 以公式(3)对之进行预补偿处理:

$$L_{in} = L_{org}^{1/\gamma} \quad (3)$$

公式(3)对原始图像数据的作用结果如图3所示。

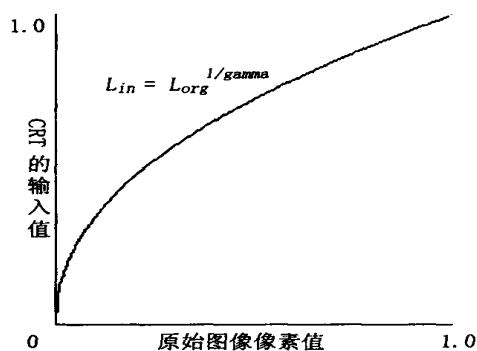


图3 gamma 预补偿曲线

然后把经过预补偿的图像数据 L_{in} 交给 CRT 显示。实际上, CRT 的输出值 L_{out} 可以通过公式(2)和(3)联立推导出来:

$$L_{out} = L_{in}^{\gamma} = (L_{org}^{1/\gamma})^{\gamma} = L_{org}$$

如图4所示,预补偿公式(3)人为地使 CRT 的输入值对比原始图像产生了 gamma 反向失真,这种反向失真正好被显示器的硬件 gamma 失真所抵消,从而使得显示器的输出值与图像的原始数据建立了线性关系,实现了对 CRT 的 gamma 校正。

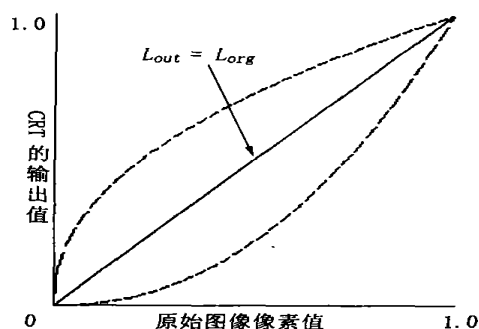


图4 校正后的 CRT 输出

以上就是对经过 CRT 的数字图像进行 gamma 校正的原理,有了上述原理及公式,就可以编写程序对数字图像进行校正。

2 Gamma 校正的计算复杂度分析

假设图像中有这样一个像素A,其值是200(注:本文使用的图像都是灰度图,并且假设像素值是0~255之间的整数,其它情况可以类推);根据上述原理对A进行校正,必须执行如下运算:

①归一化:将像素值转换为0~1之间的实数。算法如下:

$$(i + 0.5) / 256$$

这里包含1个除法和1个加法操作。对于像素A而言,其对应的归一化值为0.783203。

②预补偿:根据公式(3),求出像素归一化后的数据以 $1/\gamma$ 为指数的对应值。这一步包含一个求指数运算。若 γ 值为2.2,则 $1/\gamma$ 为0.454545,对归一化后的A值进行预补偿的结果就是 $0.783203^{0.454545} = 0.894872$ 。

③反归一化:将经过预补偿的实数值反变换为0~255之间的整数值。具体算法为:

$$f * 256 - 0.5$$

此步骤包含一个乘法和一个减法运算。续前例,将A的预补偿结果0.894872代入上式,得到A预补偿后对应的像素值为228,这个228就是最后送入显示器的数据。

综上所述,直接套用公式编制出的程序,对一幅图像进行 gamma 校正时,必须对每个像素都执行上面的5步运算,我们知道,计算机执行加、减法的效率很高,所以本算法的执行时间主要消耗在浮点数的乘、除及指数运算上。假设某图像的分辨率为 800×600 (这在目前算是比较小的图像了),对该图像进行 gamma 校正,需要执行48万个浮点数乘法、除法和指数运算。这种效率,用来处理静态图像还

勉强可以接受,但如果用在视频流传送过程中,则执行效率非常低下,远远达不到每秒校正 25 帧图像的基本要求。因此,有必要探讨一种更快的算法来取代它。

3 一种快速算法及其 C 语言实现

仔细考查上述算法,不难发现,对单个像素进行校正时,归一化、预补偿和反归一化操作是缺一不可的;但是对整幅图像校正,却可以不是单像素处理的简单重复。也就是说,并不是图像有 48 万个像素,就必须执行 48 万次乘、除及指数运算。实际上,这些运算的执行次数只取决于一个因素——像素的取值范围。

如果能够确知图像的像素取值范围,例如,0~ 255 之间的整数,则图像中任何一个像素值只能是 0 到 255 这 256 个整数中的某一个;在 gamma 值已知的情况下,0~ 255 之间的任一整数,经过“归一化、预补偿、反归一化”操作后,所对应的结果是唯一的,并且也落在 0~ 255 这个范围内。如前例,已知 gamma 值为 2.2,像素 A 的原始值是 200,就可求得经 gamma 校正后 A 对应的预补偿值为 228。

基于上述原理,我们只需为 0~ 255 之间的每个整数执行一次预补偿操作,将其对应的预补偿值存入一个预先建立的 gamma 校正查找表(LUT: Look Up Table),就可以使用该表对任何像素值在 0~ 255 之间的图像进行 gamma 校正。

下面以 C 语言源程序的方式给出 gamma 校正查找表的建立方法:

```
# include < math. h>
typedef unsigned char UINT8; //用 8 位无符号数表示 0~ 255 之间的整数
UINT8 g _ GammaLUT[ 256]; //全局数组: 包含 256 个元素的 gamma 校正查找表
//BuildTable() 函数对 0~ 255 执行如下操作:
// ①归一化、预补偿、反归一化;
// ②将结果存入 gamma 查找表。
//
//由公式(3),有 fPrecompensation= 1/gamma
void BuildTable(float fPrecompensation)
{
    int i;
    float f;
    for( i= 0; i< 256; i+ + )
    {
        f= ( i+ 0.5F)/256; //归一化
```

```
f= (float) pow(f, fPrecompensation); //预补偿(调用“求指数”函数)
```

```
g _ GammaLUT[ i] = (UINT8) ( f* 256 - 0.5F); //反归一化
    }
}
```

有了这个查找表(g _ GammaLUT),便可用于对整幅数字图像进行预补偿处理,下面给出应用该表对图像进行校正的一种简单方法:

```
void GammaCorrection (UINT8 Src[ ], //原始图像
                      int iWidth, //图像宽度
                      int iHeight, //图像高度
                      float fGamma, //系统的 gamma 值
                      UINT8 Dst[ ]) //校正后的图像
{
    int iCols, iRows;
    BuildTable( 1/fGamma); //gamma 校正查找表的初始化
    //对图像的每个像素进行查表校正
    for( iRows= 0; iRows< iHeight; iRows++ )
    {
        for( iCols= 0; iCols< iWidth; iCols++ )
        {
            Dst[ iRows* iWidth+ iCols]=
                g _ GammaLUT [ Src[ iRows* iWidth+
                                iCols] ];
        }
    }
}
```

显然,这种方法极大地减少了程序的运算量:对于像素值在 0~ 255 之间的图像,只需要进行 256 次加、减、乘、除和指数运算,就能建立 gamma 校正查找表;而对某一幅具体的图像进行校正时,若该图像分辨率为 N,则只需进行 N 次查表操作即可,非常便捷。

需要说明的是,相对于前面所提到的慢速算法而言,该快速算法在提高 gamma 校正执行效率的同时,并没有损失数值计算的精度。也就是说,快、慢两种方法的校正结果是一致的。

图 5 展示了应用上述快速算法对图像进行校正的实例(假设显示系统的 gamma 值为 2.2)。可以看出,经过 gamma 校正后,椭圆所包围的区域,图像细节清晰可见。

(下转第 108 页)

自己的热容量和比热容,因而有不同的温升(铝包钢中的铝和钢接触紧密,可瞬时传热,综合电阻为铝和钢的并联电阻,综合热容量为铝和钢热容量之和)。我们先假定一个短路电流,根据各元件的初始温度电阻进行分配,然后将短路电流持续时间分成若干等份,根据各元件的热容量求出每一时段的各自温升,再根据各元件的电阻温度系数求出温升后的电阻,反过来重新分配短路电流。得出的每种元件最终温度与其允许最高温度比较后,对原假定短路电流进行修正,直到某种元件的最终温度与其允许最高温度吻合,此时的假定短路电流就是 OPGW 光缆的允许短路电流。实际上,在每种元件温度不断升高的过程中,温差会导致热量的内部流动,若考虑传热和集肤效应等因素,最后的计算结果会更接近实际值。

提高 OPGW 光缆的热稳定性可以通过两个途径:增大光缆的允许短路电流或降低光缆的实际短路电流。提高 OPGW 光缆允许短路电流的有效办法是加大光缆截面积,这势必造成投资上的极大浪费,因此降低流过 OPGW 光缆的实际短路电流,才是提高光缆热稳定性的关键。

4 结束语

OPGW 光缆在我国的大面积推广应用已近十年。由于施工时必须停电,因此比较适合新建线路和考虑更换地线的线路。目前,由雷击导致的断股是 OPGW 光缆应用中的主要问题,普遍采用的方法是加大强雷击区 OPGW 光缆外层单线的直径。如果能从结构设计上大幅度地提高耐雷能力,再克服其带电施工的技术瓶颈,OPGW 光缆必将迎来更加广阔的明天。

参考文献:

[1] 云南省电力设计院. 电力系统光纤通信线路设计[M]. 北京: 中国电力出版社, 2003.
[2] 方洋. 关于光纤复合地线(OPGW)热稳定性的研究[J]. 东北电力学院学报, 2003, 1: 48- 50.
[3] 张建明. OPGW 光缆技术和应用问题的讨论[J]. 网络电信, 2003, 1: 26- 28.
[4] 郭致星, 鲁束. OPGW 的热稳定性分析[J]. 电力系统通信, 2004, 1: 24- 28.
[5] 郭勇. 温度对 OPGW 中的光纤余长的影响[J]. 电力系统通信, 2003, 5: 27- 29.

责任编辑: 杨 敏

(上接第 84 页)



图5 gamma 校正实例

4 结束语

本文介绍的快速算法, 特别适用于需要处理视频数据的场合, 例如在数字摄像机或数码相机等嵌入式系统中, 由前端传感器向 TFT 显示器传送视频流, 由于同一系统的 gamma 值和图像的像素取值范围是固定的, 所以只需在传送前初始化一次 gamma

查找表, 便可在传送过程中对视频流的每帧图像进行实时的校正。

参考文献:

[1] Texas Instruments. Implementing the Gamma Correction Algorithm Using the TMS320C2xx DSP[M]. 1997.
[2] Donald Heam, Pauline Baker M. 计算机图形学[M]. 2 版. 蔡士杰, 吴春, 孙正兴, 等译. 电子工业出版社, 2002.

责任编辑: 杨 敏