图像抖动技术的原理及实现

何志强　(深圳国信证券公司电脑部　518001)

摘要:用较少的颜色来表示较大的色彩空间一直是人们研究的课题,本文详细讨论了半色调技术和抖动技术,并将它们扩展到实用的真彩色空间来讨论,并给出了实现的算法。

关键词:图像处理　抖动

在计算机中我们广泛使用的是红,绿,兰(R, G, B)各 8 位的 24 位真彩色图像,但要把它在色彩空间有限的计算机屏幕上显示出来或在打印机上打印出来时就涉及到半色调(Halftoning)技术。

半色调技术广泛地应用于报刊出版等领域,它通过色点大小或密度的变化造成总体强度的变化,从而表现出较多的对比度,也就达到了用较少的色彩模拟表达较多色调的目的。

在研究 MPEG 解码的过程中,我有机会详细的研究各种图象抖动算法,这里,我们就来详细讨论半色调技术和抖动技术。

一、阀值法(Threshalding)及模式法(Patterning)

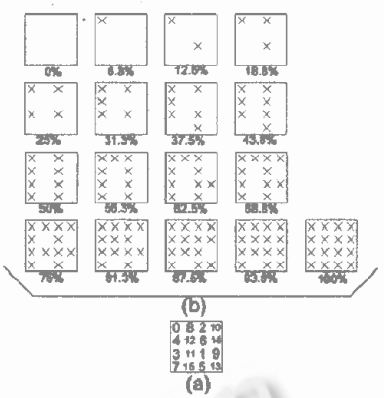
阀值法输出图象时,若象素值大于预先设定的阀值,则打开输出,对应点为亮点,反之,小于阀值时输出关闭,对应暗点。

阀值法不是对半色调的一种模拟,相反,它只是对连续色调图象输出的一种极原始的方法,它一般把图象的所有细节(即灰度)都丢失了,只剩下一个粗略的轮廓线。

与阀值法相比,模式法是种稍好点的方法,它的输出中包含一定的灰度信息. 它用输出设备上的一组象素来组合表示源图象中的一个象素. 通过控制这组象素中打开或者关闭元素的个数表现出整体亮度的变化,从而表现出源图象象素的灰度。

例如,若输出设备象素有 m 级灰度(0, 1, …, m - 1),用一个 n * n 的输出元素矩阵表示一个源图象象素,则组合结果最低亮度为 0,最高为 n * n * (m - 1),总共有 1 + n * n * (m - 1)种灰度。

我们可以做的是控制这组象素中亮点的分布以便均匀地,让人舒适地模拟出源图象灰度。这个过程可以用一个数组表示,生成数组时有一个基本原则,就是用小象组的分布递归生成大象组的亮度点分布。图 1 中的 Rglander 递归模式矩阵用 4 * 4 的黑白两色组合出 17 级灰度。



注:图(a)中 17 个矩阵对应 17 级亮度,每个矩阵中打"×"点处表示对映象点打开
图(b)是矩阵的另一种描述形式,数字表示亮点的打开次序

图 1　Rylander 4 * 4 模式矩阵

二、抖动(dithering)原理

与模式法不同,用抖动法处理图象不需扩大图象的面积,就能较好地表现出图象灰度的变化。尽管它还原出的灰度值没有模式法那么准确,但效果确实不错。

抖动法也将一个矩阵依次应用于图象(如 4 * 4 的 Bayer 矩阵),它产生 16 级灰度,但它将矩阵与图象联系起来的方式与模式法不同,不再是一输出象组对应一个源象素,而是一对一的. 也就是说,一个矩阵元素对应于

© 中国科学院软件研究所 http://www.c-s-a.org.cn

一个输出象素，也对应于一个输入象素.源图象和目的图象分辨率相同。

如果源象素强度大于与之相联系的高频矩阵元素，则与该源图象象素相对应的输出象素被打开，否则被关闭.由于抖动矩阵的值并不恒定，而是在高频"抖动"，因此对于同一象素值，由于它所对应的高频矩阵元素的不同，它被抖动(量化)的结果也在高频抖动，使人眼视觉在宏观上得到正确的亮度感。

三、带滤波的抖动算法

对于 Bayer 抖动，它的一个重大缺陷把一个固定图样施加在要处理的图象上而不管图象内容如何，当把 Bayer 抖动施加到一个人型图象时，这个图样就会显现出来，我们把这种人为的图样痕迹称为"artifact"。

一种能有效消除明显的 artifacts 的抖动方法是误差分散(error diffusion)法，它把源象素与其对应抖动矩阵象素的差值"分散"给周围的象点，则这误差在最终的抖动中就不会太显著。但这种会造成一个问题，它使得原有灰度级别悬殊的区域之间的硬边界与阀值误差一起变得柔和，分散.例如，包含某些文字的图象通常抖动得不太好。

下面讨论几种常见的带滤波的抖动算法。

最流行的是 Floyed－Steinbery 滤波器，它的误差分散方法如下：

```
      x   7
3   5   1
```

其中 X 周围的数字实际表示了 X 周围各象素分配到的 X 处象点与阀值的差的比例，我们注意到其中各数字之和为 16，则误差的 7/16 加到直接位于 X 右边的象点上，误差的 3/16 加到 X 左边下一行象点上，5/16 加到 X 下方象点，1/16 加至 X 右下象点。抖动时此过程依次重复应用于所有象点。

实际中还可以使用那些与大量象点联系从而误差分配到更多象点上的滤波器，它们通常能更好地改善原图象(比如具有文字的图象)中所有不同灰度级别区域间的边界情况，从而获得最佳的边界效应。

```
      x  8  4
2  4  8  4  2
1  2  4  2  1
```

图 2 Strcki 滤波器

```
      x  8  4
   2  4  8  4  2
```

图 3 Burkes 滤波器

图 2 中 Stucki 抖动的各数之和为 42，实现时需要用到大量的乘除操作，无法只用移位和加减法实现，图 3 中的 Burkes 滤波器是 Flyed－Steiberg 与 Strcki 滤波器间的一个折衷方案，它涉及的像点较多，各和之和为 32，可以用移位与加法操作快速实现。

还有许多其他的滤波器，有的效果很好，但因为计算量太大，无法用于小型机器中。

使用误差分散技术时仍有 artifacts 的共同起因之一是在一个扫描行上的误差过于简单地直接送到下一行使累积误差比较明显。为此以上各种滤波器都使用了"S"型扫描，即不是用以左至右的方式扫描所有行，而是从左到右和从右到左交错起来使用，要注意的是从右向左扫描行误差分配时，应使用原滤波器的一个镜象滤波器，如对 Floyed－Steinbery 滤波器为：

```
   7   x
1  5   3
```

四、向多灰度级及彩色扩展的抖动算法

以上各节所涉及的抖动技术，都是针对输出象素只有黑白二值的情况举例的，而计算机监视器不但每个象素可由红，绿，兰三原色组成，而且每个成分的亮度值都是可变的，正因为其可变才有了无限多样的计算机图象。因此，实际显示图象时出现了向多灰度级及彩色扩展的问题。

1.可变的量化－－向多灰度级的转变

将连续空间映射到有限的灰度级的思想与阀值法相同－－量化。只不过对阀值法而言，量化后的空间只有两部分－－黑与白，而这个量化值被我们称作阀值。当减小量化值时，这个连续空间将变成多个等长区间，这就是我们通常所称的量化，用一个图形象表示三者关系如图 4a，b，c 所示：

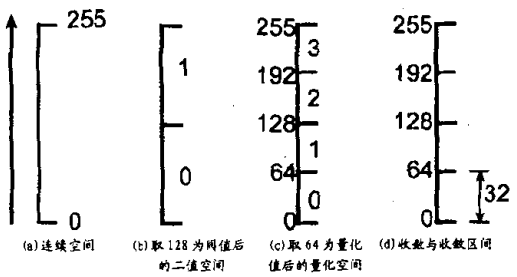
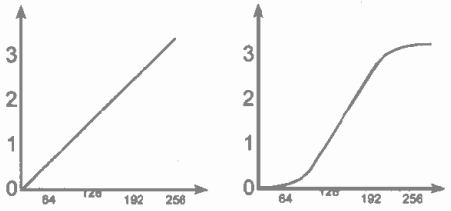


图 4 连续空间的量化

© 中国科学院软件研究所 http://www.c-s-a.org.cn

以后，当我们引用某一区间，如上图 c 中的第 0 个区间时，把它们作为 0 或作 64 引用都是不恰当的，应该作为该区间各元素的平均值来引用。我们称该区间(0～64)各元素收敛至 32，32 为收敛点，区间[0，64) 为值 32 的收敛区间，参见上图(d)。一般的，设量化步长为 s，则 n 收敛至 x = (n/s) * s + s/2，而收敛点 x 的收敛区间为半开半闭区间。[(n/s * s，(n/s) * s + s)。这些概念在稍后讨论 bayer 抖动时要用到。

经过上述量化后，各收敛点所对应收敛区间长度相等，我们称这种量化为等步长量化，如图 5(a)。实际上，图象的细节一般表现在色彩空间的中部，而最亮与最暗区间的细节较少，并且，人眼也对色彩空间中部亮度的变化比较敏感，所以，图 5(a) 中那种等步长量化手段是比较粗糙的，完全可以应用一种形如图 5(b)的不等步长量化，它中部区分得较细致，要实现它当然可以用个函数，但无论从时间还是从灵活性来考虑，还是用一个具有 256 个元素的一维数组比较好，它一一规定了每个值的收敛点。



(a) (b)

图 5 等步长量化与不等步长量化

2.彩色抖动扩展之一：位组合法

彩色显示由红，绿，兰三个色素组合，所以，从理论上说，在 MPEG 解码时，当我们得到象素的亮度及两个色度(即 L，U，V)的值后，一般先变成 R，G，B 值，再分别进行等步长或不等步长量化，然后再将量化后值映射至调色板空间输出，这个映射过程一般可用数组完成。

很冗长，是不是．为了实用时加快速度，我们使用大量的表格，把能算的尽量在初始化时算出来，存好，显示时只要查表，组合，映射显示即可。

首先，绕过了 R，G，B 的量化，可直接将 L，U，V 制成量化表，只要以后将量化值映射至调色板空间时按 L，U，V 值映射即可。

其次，这种映射用三维数组实现时，下标计算会很复杂，我们可以把这个三维数组拉平成一维数组，下标计算预先做好，即把各维的偏移量先存起来，由量化结果组合时引用。

3.彩色抖动扩展之二：检索树法

据统计，MPEG 解码时有相当时间被用于图象抖动，所以进一步开发新的加速策略实属必要。从位组合法的检索公式 map[LQuan[l] + UQuan[u] + VQuan[v]可以看到，在取得 l，u，v 值后，还要进行四次访问数组操作外加两次加法才能得到调色板索引，而这里讨论的检索树法可以省去这两加法操作，并少访问一次数组，空间消耗也是微不足道．它的访问公式是：RV[v][u][l]，l，u，v∈[0，255]．你可能惊呼"哇！这空间 256 * * 3，可是天文数字"且慢，这并不真的三维数组，而是用指针链串起来的伪三维数组，且听我细细分解。

我们知道，显示 L，U，V 时都是带量化的，输入是[0，255]，输出却会小得多，若把 7 位的色彩空间 L，U，V 分别分配 3，2，2 位，则 l，u，v 的取值范围分别为[0，7]，[0，3]，[0，3]．检索树法正是利用这种受限的输出特性来获得空间收益的。

4.实际的抖动算法

将多灰度及彩色扩展法与各种单色抖动算法相结合可产生多种实际算法．Berkeley MPEGDecoder 实现了多种抖动方法，这里选择几种有代表性的予以讨论。

(1)Grayscale 抖动。Grayscale 抖动相当简单，它抛弃了 u，v 两个色度信息，只将 L 信道的值量化至 128 个灰度级，它采用的是等步长量化，我们也可以使用不等步长量化。

(2)2 * 2 抖动。2 * 2 抖动用的就是第一节所讨论的模式法，它把象素的 L，U，V 值平均分配到 2 * 2 范围内的 4 个象素中，以增加显示面积为代价来获得更深的显示色彩空间。

对于任何要显示的灰度级 m，除以 4 取整得[m/4]平均分配到 4 个象素中，余数部分 mod(m，4)按模式 $\left[\begin{matrix}1 & 3\\ 2 & \end{matrix}\right]$分配至 4 个色素，最后这四个象素值之和为 m．L，U，V 三个 channel 的值按位组合法分配不同的的基组合在一个显示象素中，所以最终模式矩阵中每一个显示象素值都包含了彩色信息，对应一个调色板入口。

模式矩阵的定义形式是：

unsigned char pattern[Extended-V-RANGE][Extended-U-RANGE][Extended-L-range][4]；

访问模式矩阵用优化的位组合法 pattern[VQuan

© 中国科学院软件研究所 http://www.c-s-a.org.cn

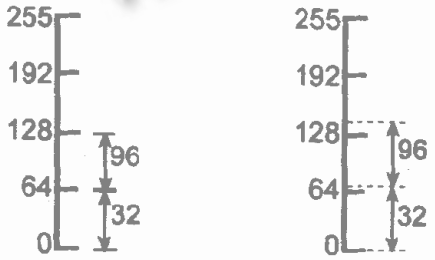
[V] + UQuan[U] + LQuan[L]]，量化矩阵 LQuan, UQuan, VQuan 内已包含了 L, U, V 各自的数组下标计算用偏移量。

因为量化矩阵的输出值空间仍为有限的，所以这里也可以用检索树法。

(3)Ordered 抖动。MPEG-Decoder 分别用位组合法及检索树法实现了基于 4 * 4 抖动矩阵的 Bayer 抖动(参见第二节)

0	8	2	10
12	4	14	6
3	11	1	9
15	7	13	5

图 6 Bayer 4 * 4 抖动矩阵



(a)对应阀值 8 (b)对应阀值 9

图 7 阀值改变时收敛点 96 的收敛区间的变化

Bayer 抖动矩阵的阀值效应在多灰度级空间中是通过改变各收敛点的收敛区间而实现的，如图 7(a)，收敛点 96 的正常收敛区间为[64, 128)，对应抖动阀值 9 的收敛区间为[68, 132)，图 7(b)比正常区间上移了((9 - 8)/16) * 64 = 4，而正常区间对应抖动阀值 8。

一般地，设量化步长为 s，则收敛点 a 的正常收敛区间为[a - s/2, a + s/2)，对于抖动矩阵值 i, i∈[0, 15]，a 的收敛区间为[a - (16 - i)/16 * s, a + i/16 * s)。这种区间平移实质上是将单值的 Bayer 抖动应用于每一段，若点 a 在段内的相对位置大于 i/16，i 为抖动阀值，则 a 收敛至 a 上方的收敛点，否则收敛至下方收敛点。

具体实现时，为加快计算过程使用了很多表格，将 L, U, V 分别各自对每一抖动阀值建表：unsigned char LDith[16][256]，UDith[16][256]，VDith[16][256]，将 L, U, V 对各抖动阀值的收敛点预先算好，并按各自的位组合偏移量存储起来，以后可以用 LDith[i][l] + LDith[i][u] + LDith[i][v]取得元组(l, u, v)在阀值 i 的组合值，并且容易将它转化为调色板入口。

用检索树实现 Bayer 抖动时，可以用 Bayer[i][v][u][l] 取得对于阀值 i，元组(l, u, v)的调色板入口，它的数据结构是这样的：

```
unsigned char Rl[16][4][4][256];
unsigned char * Ru[16][4][256];
unsigned char * * Rv[16][256];
```

它的意义与第 4.3 节类似，第一维为 16 是为了对应不同的抖动阀值 i, i∈[0, 15]. 收敛点的调整是通过 Bayer[i][v][u][l] 中上级指针所引用的下级指针的偏移量实现的，具体不再详述。

用检索树实现的 Bayer 抖动速度较快，图象质量较高，是我们解码器的首选算法。

(4) Floyed-steinbery 抖动。实现 Floyed-steinbery (FS)抖动的关键是在多灰度空间中误差如何生成。在单值 FS 抖动中，取黑白的中间值，如 128 作阀值，把象素值与阀值的差作为误差分散下去。在多灰度空间中思想与之相同，把每一收敛区间的收敛区间的收敛点作为该区间的阀值，求差，再扩散。实际上，这个误差就是象素的实际值与量化值之差。它也可以在初始化时先计算好，使用时直接查表。

实现时，为进一步消除 Artifacts 采用了"S"型扫描，即对奇数行从左至右处理，偶数行从右至左处理，采用 FS 滤波器，对偶数行用 FS 镜象滤波器处理. 用两个临时数组 current 及 next 记下当前行及下一行各象素分配到的误差，换行时将 next 数组变成 current 再行计算，相对而言，非常简单，不再详述。

参考文献

[1] MPEG, "Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s." Committee Draft of Standard ISO/IEC 11172, Nov. 1991.

[2] Ketan Patel, L.A. Rowe, Brian Smith, Source code of MPEG Video Software Encoder, Version 1.2, Univ. of Calif. at Berkeley, Oct. 1993.

(来稿时间:1999 年 2 月)

© 中国科学院软件研究所 http://www.c-s-a.org.cn