



(12) 发明专利

(10) 授权公告号 CN 1868215 B

(45) 授权公告日 2012. 10. 03

(21) 申请号 200480030008. 7

H04N 5/262(2006. 01)

(22) 申请日 2004. 10. 12

(56) 对比文件

(30) 优先权数据

60/511, 026 2003. 10. 14 US

WO 97/22204 A1, 1997. 06. 19, 说明书第 21-22 页、图 9.

(85) PCT 申请进入国家阶段日

2006. 04. 13

Cristina Gomila. SEI message for film grain encoding: syntax and results. JVT OF ISO/ IEC MPEG AND ITU-T VCEG JVT-1013r2. 2003, 第 1-9 页.

(86) PCT 申请的申请数据

PCT/US2004/033713 2004. 10. 12

审查员 夏刊

(87) PCT 申请的公布数据

W02005/039189 EN 2005. 04. 28

(73) 专利权人 汤姆森特许公司

地址 法国布洛涅

(72) 发明人 吉尔·M·博伊斯

克里斯蒂娜·戈米拉 琼·拉克

亚历山德罗斯·图拉皮斯

杰弗里·A·库珀 尹澎

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 吕晓章 李晓舒

(51) Int. Cl.

H04N 7/26(2006. 01)

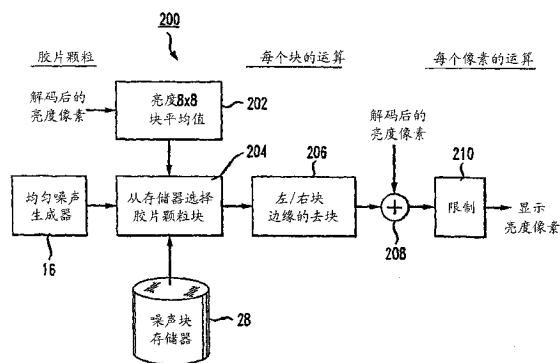
权利要求书 2 页 说明书 11 页 附图 1 页

(54) 发明名称

用于位精确的胶片颗粒模拟的技术

(57) 摘要

图像中的胶片颗粒的模拟利用包含在当传输时伴随图像的补充增强信息 (SEI) 消息中的参数。该 SEI 消息指定胶片颗粒模拟参数, 例如胶片模拟模型、混合模式和颜色空间。



1. 一种用于模拟  $M \times N$  个像素的图像块中的胶片颗粒的方法, 其中  $N$  和  $M$  是大于 0 的整数, 该方法包括以下步骤:

计算  $M \times N$  个像素的块中的像素值的平均值;

作为所述图像块的平均值和随机数的函数, 从先前建立的、包含胶片颗粒的块的池中选择  $M \times N$  个像素的胶片颗粒块; 以及

将所选择的胶片颗粒块中的每个像素与图像块中的相应像素混合。

2. 一种用于创建  $L$  个具有胶片颗粒的  $M \times N$  个像素的块的方法, 其中  $L$ 、 $N$  和  $M$  是大于 0 的整数, 用于作为图像块的平均值和随机数的函数进行的选择, 该方法包括以下步骤:

接收包括至少一个参数的胶片颗粒信息, 所述参数指定要在这  $L$  个块的每一个中出现的胶片颗粒的属性;

创建具有  $M \times N$  个随机值的  $L$  个块, 每个块具有其从先前建立的高斯随机数列表中选择  $M \times N$  个随机值;

计算具有  $M \times N$  个随机值的这  $L$  个块的每一个的离散余弦变换;

利用所接收的胶片颗粒信息中的至少一个参数来过滤由离散余弦变换产生的这  $L$  个变换后的块中每一个的  $M \times N$  个系数;

计算这  $L$  个变换后的块的每一个的过滤后的  $M \times N$  个系数的逆离散余弦变换;

按照由所接收的胶片颗粒信息中的一个参数指示的那样换算这  $L$  个变换后的块的每一个的所有像素值; 以及

将这  $L$  个胶片颗粒块的每一个存储在胶片颗粒块的池中, 用于作为图像块的平均值和随机数的函数进行的选择。

3. 根据权利要求 2 所述的方法, 还包括以下步骤: 执行离散余弦变换 (DCT) 和逆离散余弦变换 (IDCT) 的整数逼近以降低复杂度。

4. 根据权利要求 2 所述的方法, 还包括以下步骤: 换算这  $L$  个胶片颗粒块的每一个的顶部和底部边缘以隐藏块的边缘。

5. 根据权利要求 2 所述的方法, 其中, 接收胶片颗粒信息的步骤还包括以下步骤: 将包含所述至少一个参数的补充增强信息消息解码。

6. 一种用于模拟  $M \times N$  个像素的图像块中的胶片颗粒的装置, 其中  $N$  和  $M$  是大于 0 的整数, 该装置包括:

用于计算  $M \times N$  个像素的块中的像素值的平均值的部件;

用于作为图像块的平均值和随机数的函数、从先前建立的包含胶片颗粒的块的池中选择  $M \times N$  个像素的胶片颗粒块的部件; 以及

用于将所选择的胶片颗粒块中的每个像素与图像块中的相应像素混合的部件。

7. 一种用于创建具有胶片颗粒的  $M \times N$  个像素的块的装置, 其中  $N$  和  $M$  是大于 0 的整数, 用于作为图像块的平均值和随机数的函数进行的选择, 该装置包括:

用于接收包括至少一个参数的胶片颗粒信息的部件, 所述参数指定要在块中出现的胶片颗粒的属性;

用于创建从先前建立的高斯随机数列表中选择具有  $M \times N$  个随机值的块的部件;

用于计算所述具有  $M \times N$  个随机值的块的离散余弦变换的部件;

用于利用所接收的胶片颗粒信息中的至少一个参数来过滤由离散余弦变换产生的

$M \times N$  个系数的部件；

用于计算过滤后的  $M \times N$  个系数的逆离散余弦变换的部件；

用于按照由所接收的胶片颗粒信息中的一个参数所指示的那样换算具有  $M \times N$  个随机值的块中的所有像素值的部件；以及

用于将所创建的胶片颗粒块存储在胶片颗粒块的池中以便作为图像块的平均值和随机数的函数进行选择

8. 根据权利要求 7 所述的装置,还包括:用于执行离散余弦变换 (DCT) 和逆离散余弦变换 (IDCT) 的整数逼近以降低复杂度的部件。

9. 根据权利要求 7 所述的装置,还包括:用于换算所创建的胶片颗粒块的顶部和底部边缘以隐藏块的边缘的部件。

10. 根据权利要求 7 所述的装置,其中,用于接收胶片颗粒信息的部件还包括用于将包含所述至少一个参数的补充增强信息消息解码的部件。

## 用于位精确的胶片颗粒模拟的技术

### 技术领域

[0001] 本发明涉及一种用于模拟图像中的胶片颗粒 (film grain) 的技术。

### 背景技术

[0002] 运动画面胶片包括分散在乳剂中的卤化银晶体,所述乳剂被涂覆在胶片片基上的薄层中。这些晶体的曝光和显影形成由离散的微小银粒子构成的摄影图像。在彩色底片中,在生胶片的显影期间,微小的染色斑点 (blob) 出现在银的化学去除之后银晶体形成的地方。这些小染色点通常具有彩色胶片中的标志‘颗粒’。由于银晶体在原始乳剂上的随机形成,颗粒呈现为随机地分布在所得到的图像上。在均匀曝光的区域内,一些晶体在曝光之后显影而其它的则没有。

[0003] 颗粒在尺寸和形状上不同。胶片越快,所形成的银块和所产生的染色斑点越大,并且它们越趋向于以随机的图案 (pattern) 聚集在一起。术语“粒度”一般是指颗粒图案。肉眼不能分辨从 0.0002mm 变化到大约 0.002mm 的单独的颗粒。取而代之的是,眼睛分辨被称为斑点的颗粒组。观察者将这些斑点组识别为胶片颗粒。随着图像分辨率变得更大,对胶片颗粒的感觉变得更强。胶片颗粒在电影和高清晰度 (HD) 图像上变得可清楚地察觉到,而胶片颗粒在标准清晰度 (SD) 中逐渐失去重要性,并且以更小的形式变得感觉不到。

[0004] 运动画面胶片通常包含由摄影胶片的曝光和显影的物理过程或者从随后的图像编辑引起的图像相关的噪声。摄像胶片具有由摄影乳剂的物理粒度引起的特有的准随机图案或纹理。或者,相似图案的模拟可以发生在计算机生成的图像中,以便将它们与摄影胶片混合。在两种情况下,这一图像相关的噪声具有“胶片颗粒”的名称 (designation)。适度的颗粒纹理经常在运动画面中呈现出所期望的特征。在一些实例中,胶片颗粒提供帮助正确感知二维画面的视觉暗示。在单个胶片中胶片等级 (grade) 经常变化,以便提供对于时间基准、观察点等的各种线索。在运动画面工业中,对于控制颗粒纹理存在许多其它的技术和艺术需求。因此,在图像处理和传送链中保持图像的颗粒状外观已经成为运动画面工业中的要求。

[0005] 可在市场上买到的若干产品具有模拟胶片颗粒的能力,所述能力常常用于将计算机生成的对象混合到自然场景中。来自纽约 Rochester 的 EastmanKodak 公司的 Cineon<sup>®</sup>、即用来实现颗粒模拟的第一批数字胶片应用程序之一对于许多颗粒类型产生非常真实的结果。然而,由于 Cineon<sup>®</sup> 应用程序对于高颗粒大小的设置产生的可察觉到的斜条纹,因此该应用程序对于很多高速胶片没有产生良好的性能。而且,当图像受到先前的处理时,例如当图像被复制或数字处理时,Cineon<sup>®</sup> 应用程序不能以足够的保真度来模拟颗粒。

[0006] 模拟胶片颗粒的另一商用产品是来自 Visual Infinity 公司的 GrainSurgery<sup>™</sup>,其被用作 Adobe<sup>®</sup> After Effects<sup>®</sup> 的插件。Grain Surgery<sup>™</sup> 产品看起来似乎通过过滤一组随机数来生成合成颗粒。这一方法受到高计算复杂度的缺点的损害。

[0007] 因此,需要一种高效的胶片颗粒模拟技术,其减小对于存储器带宽的需要和计算

工作强度 (effort), 从而允许诸如机顶盒的成本敏感的大量设备中的胶片颗粒模拟。

### 发明内容

[0008] 简而言之, 根据本原理的优选实施例, 提供一种用于模拟  $M \times N$  个像素的图像块中的胶片颗粒的方法, 其中  $N$  和  $M$  是大于 0 的整数。该方法以首先计算该  $M \times N$  个像素的块中的像素值的平均值开始。作为所述图像块的平均值和随机数的函数, 从先前建立的、包含胶片颗粒的块的池中选择  $M \times N$  个像素的胶片颗粒块。将所选择的胶片颗粒块中的每个像素与图像块中的相应像素混合。

### 附图说明

[0009] 图 1 示出用于生成预先建立的胶片颗粒块以便在随后的胶片颗粒模拟中使用的装置的方框示意性图; 以及

[0010] 图 2 示出根据本原理的、用于使用由图 1 的装置生成的预先建立的胶片颗粒块来在逐个像素的基础上模拟胶片颗粒的装置的方框示意图。

### 具体实施方式

[0011] 本原理的方法根据与图像一起传送的胶片颗粒信息来模拟胶片颗粒, 其中所模拟的颗粒被混合到所述图像中。在实践中, 所传送的图像通常在传送之前通过诸如 H. 264 压缩方案的多种众所周知的压缩方案之一而受到压缩 (编码)。利用使用 H. 264 压缩方案压缩的所传送的图像, 胶片颗粒信息的传送通常通过补充增强信息 (SEI) 消息而发生。根据负责颁布 H. 264 标准的标准团体最近采纳的著作 (contributions), SEI 消息现在可以包括指定不同胶片颗粒属性的各种参数。

[0012] 对胶片颗粒 SEI 消息参数的限制

[0013] 本原理的方法对 H. 264 建议所允许的参数的数目及其可能值的范围施加了一些限制。

[0014] 表 1 提供这样的参数的列表, 包括对其语义的描述和本原理施加的限制。

[0015] 表 1

[0016]

胶片颗粒参数	描述和限制
model_id	此参数指定模拟模型。它将是 0, 其将胶片颗粒模拟模型标识为频率滤波。
separate_colour_description_present_flag	此参数指定在其中估计所述参数的颜色空间是否与在其中已经将视频序列 (胶片颗粒 SEI 消息已经被嵌入其中) 编码的颜色空间不同。它将是 0, 其将胶片颗粒的颜色空间标识为与编码序列相同。

blending_mode_id	此参数标识用来将所模拟的胶片颗粒和解码后的图像混合的混合模式。它将是 0, 其对应附加 (additive) 混合模式。
log2_scale_factor	此参数标识用来表示 SEI 消息中的胶片颗粒参数的对数比例因子。它将在范围 [0, 4] 内, 以确保可以使用 16 位算术来进行胶片颗粒模拟。
comp_model_present_flag[1]	此参数允许 YCbCr 颜色空间中的 Cb 颜色分量的胶片颗粒参数的传送。它将是 0, 因为不支持在色度上的胶片颗粒模拟。
comp_model_present_flag[2]	此参数允许 YCbCr 颜色空间中的 Cr 颜色分量的胶片颗粒参数的传送。它将是 0, 因为不支持在色度上的胶片颗粒模拟。
no_intensity_intervals_minus_1[0]	此参数定义已经为其估计了特定的一组参数的强度间隔的数目。它将在范围 [0, 7] 内。
intensity_interval_lower_bound[0][i+1], intensity_interval_upper_bound[0][i],	这些参数定义为其定义了不同胶片颗粒参数的亮度 (luma) 强度间隔的边界。因为不允许多代 (multigenerational) 胶片颗粒, 所以间隔 i+1 的下界必须大于间隔 i 的上界。
num_model_values_minus_1[0]	此参数指定为在其中已经模仿了胶片颗粒的每个强度间隔提供的模型值的数目。因为不允许颜色相关, 所以它将在范围 [0, 4] 内。
comp_model_value[0][i][0]	此参数表示在其中已经模仿了胶片颗粒的每个亮度强度间隔的胶片颗粒强度。它将在范围 [0, 255] 内, 以确保可以使用 16 位算术来进行胶片颗粒模拟。

[0017]

[0018] 除了前面的限制之外, 本原理还施加了: 胶片颗粒 SEI 消息在 I 画面之前, 并且只有一个胶片颗粒 SEI 消息可以在特定 I 画面之前。(等于 7 的 slice\_type 或等于 5 的 nal\_ref\_idc 在比特流中的存在指示 I 画面。)

[0019] 胶片颗粒 SEI 消息的所有其它参数对于标准规范没有限制。

[0020] 胶片颗粒模拟的位精确 (bit-accurate) 实现

[0021] 根据本原理的胶片颗粒模拟以两步的过程发生。首先,如针对图 1 更详细地描述的那样,在初始化期间生成胶片颗粒块的池。随后,如针对图 2 描述的那样,将所选择的胶片颗粒部分加到每个解码后的画面的每个亮度像素上。

[0022] 图 1 示出了根据本原理的所说明的实施例的、用于生成胶片颗粒块的池以便在胶片颗粒模拟中使用的装置 10。装置 10 通常为多达 8 个不同亮度强度间隔中的每一个生成 128 个胶片颗粒块的池。SEI 消息字段 num\_intensity\_intervals\_minus\_1[0] 指示小于亮度强度间隔数目的数目。

[0023] 装置 10 使用所指定的均匀伪随机数多项式生成器 12、并且使用所指定的、存储在查找表 14 中的 2048 个 8 位高斯分布随机数的列表来完成胶片颗粒噪声初始化。查找表 14 存储在  $[-63, 63]$  范围内的二进制补码 (2's complement) 形式的随机数。高斯随机数的列表呈现在附录中。

[0024] 根据本原理的位精确规范,胶片颗粒块的生成从最低亮度强度间隔开始。均匀随机数生成器 12 使用本原多项式模 2 算子  $x^{18}+x^5+x^2+x^1+1$  来生成存储在查找表 14 中的高斯随机数列表的索引。为了易于理解,术语  $x(i, s)$  将指示以初始种子  $s$  开始的序列  $x$  的第  $i$  码元。当接收到每个胶片颗粒 SEI 消息时,随机数种子被重置为 1。

[0025] 为了形成单独的  $8 \times 8$  的胶片颗粒块,随机块生成器 16 从高斯随机数查找表 14 中读取具有 (worth of) 8 个随机数的 8 行。来自随机数生成器 12 的随机偏移用来访问每个 8 个随机数的行。如下生成由块生成器 16 产生的块的每行:

[0026]  $index = x(i, 1)$

[0027] for  $n = 0 \dots 7, B[i \% 8][n] = Gaussian\_list[(index+n) \% 2048]$

[0028] 其中,  $i$  对于每个  $8 \times 1$  块的行递增。

[0029] 由生成器 16 读取的随机值的  $8 \times 8$  的块受到由整数离散余弦变换 (DCT) 变换块 18 执行的、通常是整数 DCT 的变换。在 DCT 变换后,根据在 SEI 消息中指定的截止频率 (cut frequency),这  $8 \times 8$  个随机值在频率滤波器 20 处受到频率滤波。在频率滤波之后,这  $8 \times 8$  个随机值受到由逆 DCT 块 22 进行的逆 DCT 变换。第一换算 (scaling) 块 24 如下换算顶部和底部块行上的像素:

[0030] for  $n = 0 \dots 7, B'[0][n] = (B[0][n]+1) \gg 1$

[0031] for  $n = 0 \dots 7, B'[7][n] = (B[7][n]+1) \gg 1$

[0032] 此过程持续到为每个亮度强度间隔生成一组 128 个胶片颗粒块为止。在由第二换算块 26 进行的随后的换算之后,胶片颗粒块被存储在胶片颗粒池 28 中。

[0033] 用来模拟胶片颗粒的块和像素操作

[0034] 图 2 图示了根据本原理说明性实施例的、用于使用胶片颗粒池 28 中存储的值在逐个像素的基础上模拟胶片颗粒的装置 200。装置 200 包括处理块 202,其用于产生每个  $8 \times 8$  的亮度像素值块的平均值以便与胶片颗粒 SEI 消息中的参数 intensity\_interval\_lower\_bound[0][i] 和 intensity\_interval\_upper\_bound[0][i] 进行比较,以确定用于当前块的正确亮度强度间隔。

[0035] 选择器块 204 使用由均匀随机数生成器 16 从模 128 多项式 (polynomial modulo

128) 生成的随机数作为块索引、从池 28 中选择第 k 胶片颗粒块。因此,使用用于针对图 1 描述的初始化过程的多项式生成均匀分布的随机数的噪声生成器 16 在图 2 的装置 200 中获得应用以选择胶片颗粒块,并且随机数种子在池创建过程之后被重置为 1。如果所得到的块索引与先前的块索引相同,则该索引的最后一位受到反转 (toggling)。这种运算可以使用逐位比较和 XOR(异或)算子 (^) 来如下进行:

[0036] previous\_index = index

[0037] index = x(k,1) % 128

[0038] index ^ = (index == previous\_index)

[0039] 在块选择之后,去块滤波器对先前选择的块最右侧的列和当前块最左侧的列上的像素进行去块。加法器 208 将去块后的胶片颗粒块加到解码后的亮度像素上。(由于需要两个水平相邻的块来执行去块,因此在 204 中选择的块和在 208 中加上的块之间存在 1 个块的延迟。)限制器 (clipper) 210 将结果限制在范围 [0,255] 内以便显示。注意:胶片颗粒噪声相加仅对亮度像素进行。截止频率的换算

[0040] 表 1 的胶片颗粒 SEI 消息中的参数假设在模拟过程中使用  $16 \times 16$  的 DCT。具体来说,由 comp\_model\_value[0][i][1] 和 comp\_model\_value[0][i][2] 提供的水平和垂直高截止频率、以及由 comp\_model\_value[0][i][3] 和 comp\_model\_value[0][i][4] 提供的水平和垂直低截止频率用来过滤  $16 \times 16$  个值的块的变换系数。

[0041] 在所说明的实施例中, $8 \times 8$  的块的使用将降低复杂度。使用基于  $16 \times 16$  的变换的截止频率参数来采用  $8 \times 8$  块的变换意味着在颗粒生成之前所有截止频率都需要换算。

[0042] 截止频率的换算如下进行:

[0043] comp\_model\_value'[0][i][j] = (comp\_model\_value[0][i][j]+1) >> 1

[0044] 其中 j 在范围 [1,4] 内。注意:该换算构成被上舍入为最接近的整数的整数除法的等价物。

[0045] 整数变换和偏差 (varaiance) 换算

[0046] 用于频率滤波的变换相当于使用下面的变换矩阵进行的对 DCT 的  $8 \times 8$  整数逼近:

$$[0047] \quad T_8 = \begin{pmatrix} 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\ 8 & 7 & 4 & 2 & -2 & -4 & -7 & -8 \\ 7 & 3 & -3 & -7 & -7 & -3 & 3 & 7 \\ 7 & -2 & -8 & -4 & 4 & 8 & 2 & -7 \\ 6 & -6 & -6 & 6 & 6 & -6 & -6 & 6 \\ 4 & -8 & 2 & 7 & -7 & -2 & 8 & -4 \\ 3 & -7 & 7 & -3 & -3 & 7 & -7 & 3 \\ 2 & -4 & 7 & -8 & 8 & -7 & 4 & -2 \end{pmatrix}$$

[0048] 可以使用 16 位算术。随机噪声的给定块的前向整数变换被定义为:

[0049]  $\tilde{B} = (((T_8 \times B + 8) \gg 4) \times T_8^T + 8) \gg 4$

[0050] 其中 11 位用于  $\tilde{B}$ 。

[0051] 逆整数变换被定义为:

[0052]  $B = (T_8^T \times \tilde{B} \times T_8 + 128) \gg 8$

[0053] 其中 8 位用于 B。

[0054] 在逆变换之后,假设块 B 在第 k 亮度强度间隔中,则它经历如下换算:

[0055]  $val = B(i, j) * comp\_model\_value[0][k][0]$

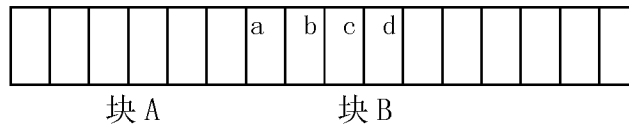
[0056]  $B'(i, j) = (((val - (val \gg 4) + 2^{\log_2\_scale\_factor - 1}) \gg \log_2\_scale\_factor) + 16) \gg 5$

[0057] 其中运算  $(val - (val \gg 4))$  补偿整数变换的换算;在 SEI 消息中传送的  $\log_2\_scale\_factor$  换算  $comp\_model\_value[0][k][0]$ ;并且 5 换算在附录中提供的高斯数。

[0058] 去块滤波器 206

[0059] 如所示出的那样,图 2 的胶片颗粒模拟装置包括用于平滑由大小较小的变换产生的分块产物 (blocking artifacts) 的去块滤波器 206。在所说明的实施例中,去块滤波器 206 采用应用于接近  $8 \times 8$  的块的左和右边缘的所有像素的 3 抽头滤波器的形式。给出属于两个相邻的  $8 \times 8$  的块的一行像素,块之间的转换位于像素 b 和 c 之间,

[0060]



[0061]

[0062] 去块的应用如下进行

[0063]  $b' = (a + (b \ll 1) + c) \gg 2$

[0064]  $c' = (b + (c \ll 1) + d) \gg 2$

[0065] 其中  $b'$  和  $c'$  分别取代原始像素 b 和 c 的值。在添加到解码后的图像之前,对每个胶片颗粒块进行左和右块边缘的去块。

[0066] 上文描述了一种用于模拟图像中的胶片颗粒的技术。

[0067] 相关申请的交叉引用

[0068] 本申请要求 2003 年 10 月 14 日提交的美国临时专利申请序列号 60/511026 在 35 U. S. C. 119(e) 下的优先权,其教导被合并于此。

[0069] 附录

[0070] 2048 个高斯分布的随机数的列表是:

[0071]  $char\ Gaussian[2048] = \{$

[0072] 0xFB, 0x05, 0x33, 0xFB, 0x14, 0xEF, 0x06, 0xID, 0x26, 0x30, 0xD5, 0x01, 0x20,

[0073] 0xD9, 0x16, 0x1B, 0xE7, 0x0A, 0x06, 0xFB, 0xF6, 0xF7, 0x10, 0xC1, 0x08,

[0074] 0xFE, 0xCC, 0x09, 0x09, 0x23, 0x17, 0xFB, 0xED, 0x15, 0xFF, 0x25, 0xDF,

[0075] 0x1A, 0xD3, 0x10, 0xE9, 0x0A, 0xFF, 0xE5, 0x18, 0x00, 0xE4, 0xEC, 0x00,

[0076] 0x3C, 0xC1, 0xCB, 0xE8, 0x04, 0x07, 0x3F, 0x3D, 0x36, 0x19, 0x3F, 0x00, 0x03,

[0077] 0x38, 0x09, 0x0E, 0x06, 0x26, 0x38, 0x28, 0xE2, 0xC1, 0x37, 0xE7, 0xF2, 0x01,

[0078] 0xE8, 0xF5, 0x1D, 0xF2, 0xDC, 0x05, 0x38, 0x21, 0x27, 0xFF, 0xC7, 0xD5,

[0079] 0xFE, 0xFE, 0x14, 0xID, 0xD8, 0x18, 0xF3, 0xF1, 0xEF, 0xCC, 0x19, 0x08, 0xF4,

[0080] 0xEF, 0xFA, 0xF9, 0xC1, 0xE5, 0xF5, 0xE5, 0xC1, 0xC8, 0x02, 0xF4, 0xDC,

[0081] 0x3F, 0x3F, 0xFF, 0x14, 0x2B, 0xE0, 0xF9, 0x1B, 0x09, 0x2D, 0xD8, 0xE0, 0xE0,

[0082] 0x11, 0xFD, 0xE5, 0x31, 0xFD, 0x2C, 0x3E, 0xF3, 0x2D, 0x00, 0x1F, 0x1D,  
[0083] 0xF9, 0xF5, 0x38, 0xF0, 0x3A, 0x06, 0x0C, 0x19, 0xF8, 0x35, 0xFD, 0x1A, 0x13,  
[0084] 0xEF, 0x08, 0xFD, 0x02, 0xD3, 0x03, 0x1F, 0x1F, 0xF9, 0x13, 0xEE, 0x09, 0x1B,  
[0085] 0x08, 0xE7, 0x13, 0x10, 0xEE, 0x3E, 0xED, 0xC5, 0x08, 0xF1, 0x00, 0x09, 0x31,  
[0086] 0x1E, 0x32, 0xFA, 0xDC, 0xF8, 0xE7, 0x31, 0x01, 0x01, 0x1D, 0x10, 0xFF, 0xFF,  
[0087] 0x04, 0xEC, 0xCC, 0xEE, 0x06, 0x3F, 0x07, 0xC1, 0xF1, 0xD5, 0xED, 0xE5,  
[0088] 0x16, 0xEC, 0x25, 0x0B, 0xF7, 0xF5, 0xDD, 0x25, 0xE6, 0x00, 0x10, 0xEA,  
[0089] 0x08, 0xD2, 0x1D, 0xE0, 0xDF, 0x1B, 0xCE, 0xF2, 0xD5, 0xEF, 0xD2, 0x21,  
[0090] 0x02, 0xDC, 0xE2, 0x2E, 0xEB, 0x06, 0xF4, 0xEE, 0xC1, 0xF8, 0x07, 0xC1,  
[0091] 0x1F, 0x11, 0x0F, 0x2E, 0x08, 0xE7, 0xE3, 0x23, 0x26, 0x28, 0x3F, 0x3F, 0x1E,  
[0092] 0x10, 0xCC, 0xD2, 0x00, 0x00, 0x25, 0xDE, 0x23, 0x3F, 0xF7, 0xC9, 0x0E,  
[0093] 0x0B, 0x07, 0x01 0x13, 0x2D, 0x02, 0x14, 0x00, 0xFE, 0x13, 0x07, 0x38,  
0xF2,  
[0094] 0xEE, 0x19, 0x15, 0x35, 0x0D, 0x3B, 0x03, 0xD9, 0x0C, 0xDE, 0xF6, 0x2E,  
[0095] 0xFB, 0x00, 0x09, 0x14, 0xE7, 0x27, 0xC1, 0xEB, 0x3F, 0x08, 0x05, 0xF6, 0x0F,  
[0096] 0xE7, 0x0D, 0xD4, 0xD3, 0xED, 0xF7, 0xFC, 0x0C, 0xC6, 0x23, 0xF4, 0xEB,  
[0097] 0x00, 0x05, 0x2A, 0xCB, 0x13, 0xF0, 0xC1, 0x17, 0x19, 0xF4, 0xF6, 0x16, 0x00,  
[0098] 0x07, 0xEF, 0xDE, 0x00, 0xDC, 0x0C, 0xFD, 0x00, 0x0E, 0xFF, 0x16, 0x10,  
[0099] 0xF0, 0x3A, 0xEA, 0x27, 0xF5, 0xF8, 0xCA, 0xFB, 0xDD, 0x2C, 0xE9, 0x0B,  
[0100] 0xD3, 0x3B, 0xEE, 0x18, 0xC1, 0x1D, 0x10, 0xD8, 0xFB, 0xF8, 0xFD, 0x16,  
[0101] 0xC1, 0xF9, 0x2C, 0x3F, 0x08, 0x31, 0xED, 0xF0, 0x12, 0x15, 0xED, 0xF1, 0xF6,  
[0102] 0x34, 0xF7, 0x09, 0x09, 0xE3, 0xFC, 0x0F, 0x00, 0xC1, 0x10, 0x3F, 0xD6, 0x25,  
[0103] 0x0B, 0xEC, 0xE8, 0xC1, 0xCB, 0xF9, 0x16, 0xDB, 0x00, 0x0E, 0xF7, 0x14,  
[0104] 0xDE, 0xED, 0x06, 0x3F, 0xFF, 0x02, 0x0A, 0xDC, 0xE3, 0xC1, 0xFF, 0xFF,  
[0105] 0xE6, 0xFE, 0xC5, 0x2E, 0x3B, 0xD8, 0xE8, 0x00, 0x09, 0xEA, 0x21, 0x26,  
[0106] 0xFA, 0xF6, 0xC1, 0x11, 0xEC, 0x1B, 0x3B, 0xFE, 0xC7, 0xF5, 0x22, 0xF9,  
[0107] 0xD3, 0x0C, 0xD7, 0xEB, 0xC1, 0x35, 0xF4, 0xEE, 0x13, 0xFD, 0xFD, 0xD7,  
[0108] 0x02, 0xD5, 0x15, 0xEF, 0x04, 0xC1, 0x13, 0x22, 0x18, 0xE1, 0x24, 0xE8, 0x36,  
[0109] 0xF3, 0xD4, 0xE9, 0xED, 0x16, 0x18, 0xFF, 0x1D, 0xEC, 0x28, 0x04, 0xC1,  
[0110] 0xFC, 0xE4, 0xE8, 0x3E, 0xE0, 0x17, 0x11, 0x3A, 0x07, 0xFB, 0xD0, 0x36,  
[0111] 0x2F, 0xF8, 0xE5, 0x22, 0x03, 0xFA, 0xFE, 0x18, 0x12, 0xEA, 0x3C, 0xF1,  
[0112] 0xDA, 0x14, 0xEA, 0x02, 0x01, 0x22, 0x08, 0xD9, 0x00, 0xD9, 0x02, 0x3F, 0x15,  
[0113] 0x0D, 0x3F, 0xC1, 0x0D, 0xE5, 0xF3, 0x1B, 0x37, 0x17, 0x35, 0x00, 0xDA,  
[0114] 0x00, 0x1A, 0xFC, 0xF5, 0xEB, 0x3D, 0x36, 0x3F, 0x32, 0x21, 0x17, 0x02, 0x00,  
[0115] 0x3D, 0xFA, 0xE5, 0xF0, 0xE8, 0x2C, 0x20, 0xCC, 0xFE, 0x2F, 0xE6, 0x1F,  
[0116] 0x16, 0x0E, 0x17, 0x09, 0xEF, 0x07, 0x14, 0x17, 0xD0, 0xF4, 0x2F, 0xDB, 0x3F,  
[0117] 0xC7, 0x3F, 0xDF, 0x00, 0xF8, 0x19, 0xD1, 0x17, 0x05, 0x11, 0xEA, 0xDB,  
[0118] 0x2C, 0xCB, 0xFC, 0xE4, 0xF2, 0xCA, 0xF4, 0x3F, 0xE2, 0xFA, 0x26, 0xEA,  
[0119] 0x08, 0x09, 0x29, 0xF5, 0x04, 0x3F, 0xDF, 0x1A, 0x01, 0x0C, 0x06, 0x37, 0x15,

[0120] 0xC8, 0xF5, 0x05, 0xF4, 0x29, 0x21, 0xFA, 0x25, 0xC3, 0xID, 0x3F, 0xFB, 0x31,  
[0121] 0xF7, 0x1F, 0xED, 0x1A, 0x04, 0x03, 0x1E, 0xE5, 0x01, 0xE4, 0x38, 0xCC,  
[0122] 0xE3, 0x01, 0xFC, 0xE9, 0x24, 0x2A, 0xE5, 0xEF, 0x06, 0x3B, 0x0D, 0x2E,  
[0123] 0xDD, 0x06, 0xCF, 0xDD, 0xF6, 0x0E, 0x23, 0xD1, 0x09, 0xE6, 0x20, 0xFA,  
[0124] 0xE1, 0xF4, 0x20, 0x24, 0xFC, 0x3F, 0x00, 0xC1, 0x33, 0xF6, 0xDC, 0xC9,  
[0125] 0xCD, 0xFD, 0x0E, 0xEC, 0xF6, 0xE3, 0xF2, 0xF4, 0x09, 0xFE, 0xE7, 0x2F,  
[0126] 0xE3, 0xD1, 0xEE, 0x11, 0x09, 0xDE, 0x3F, 0xF7, 0xC1, 0xF5, 0xC5, 0xE6,  
[0127] 0x12, 0x25, 0xC1, 0x00, 0xFB, 0xC5, 0xE6, 0xF3, 0x13, 0x22, 0x08, 0x08, 0xC7,  
[0128] 0x2C, 0x1F, 0x0C, 0x12, 0xF5, 0x18, 0xCE, 0xF1, 0xFC, 0xD1, 0xE6, 0x02,  
[0129] 0x2E, 0xF5, 0xE8, 0xFC, 0x19, 0x01, 0xDB, 0xD4, 0xFB, 0xED, 0x3F, 0xD5,  
[0130] 0xF5, 0x09, 0x0A, 0x38, 0x25, 0x19, 0xF1, 0x2E, 0xE1, 0x03, 0xFB, 0x17, 0x12,  
[0131] 0x32, 0xEB, 0xF8, 0xE6, 0xFD, 0xEE, 0xDA, 0xF1, 0xF6, 0x1F, 0x0F, 0x1F,  
[0132] 0x0A, 0xC1, 0x0F, 0x1F, 0x12, 0x33, 0xD6, 0xFC, 0x26, 0x27, 0x1D, 0xD9,  
[0133] 0xFD, 0x11, 0x04, 0x28, 0xF4, 0xFC, 0x01, 0xF8, 0x23, 0x3F, 0x29, 0xD5, 0x1B,  
[0134] 0x09, 0xC5, 0xC3, 0x12, 0x05, 0x3F, 0x1C, 0xE5, 0x38, 0x06, 0x0C, 0x10, 0xFA,  
[0135] 0xE9, 0x0A, 0xFA, 0x02, 0x1C, 0x0D, 0x0C, 0x0C, 0xFB, 0xEE, 0x12, 0xD2,  
[0136] 0x26, 0x28, 0x04, 0x19, 0x06, 0x21, 0xFA, 0x00, 0x10, 0x16, 0xDB, 0x10, 0xED,  
[0137] 0xF5, 0xE8, 0xC1, 0xF3, 0x0F, 0xFC, 0x11, 0x06, 0x23, 0x06, 0x1C, 0x05, 0xE6,  
[0138] 0xD6, 0x1A, 0xEA, 0xEF, 0x00, 0x3F, 0x05, 0xDF, 0xEA, 0x17, 0xC7, 0x01,  
[0139] 0x05, 0x1C, 0xEF, 0x3B, 0xF7, 0xE2, 0x1A, 0xE3, 0xC1, 0xE8, 0xF5, 0x01,  
[0140] 0xFE, 0x08, 0xD8, 0xFE, 0x3F, 0x0C, 0x27, 0x21, 0x1F, 0xF4, 0x06, 0xE0, 0xEE,  
[0141] 0xC1, 0xF2, 0x0A, 0xE1, 0x20, 0xE6, 0xEC, 0x36, 0xE1, 0x07, 0xF6, 0x06,  
[0142] 0x0E, 0xE1, 0x0A, 0x0D, 0x2F, 0xEA, 0xE3, 0xC6, 0xFC, 0x27, 0xE8, 0x0B,  
[0143] 0xEB, 0xF8, 0x17, 0xE9, 0xC4, 0xEF, 0xF2, 0xE6, 0xEA, 0x0E, 0x3F, 0xFA,  
[0144] 0x18, 0xFC, 0xC1, 0x25, 0xF3, 0xF5, 0x2C, 0x1D, 0x05, 0xD1, 0x28, 0xE3,  
[0145] 0x1D, 0x1E, 0xF4, 0x14, 0xD3, 0xFF, 0xF6, 0xE3, 0xEA, 0xE3, 0xF5, 0xE6,  
[0146] 0x23, 0xF2, 0x21, 0xF1, 0xF5, 0x07, 0xF8, 0xDF, 0xF4, 0xF2, 0xE2, 0x17, 0x12,  
[0147] 0x08, 0x07, 0xEE, 0xF5, 0xFB, 0x04, 0xF3, 0xF7, 0x1D, 0x16, 0xE8, 0xE9, 0xFF,  
[0148] 0xF6, 0xD8, 0x0E, 0xDF, 0xC1, 0x25, 0x32, 0x02, 0xF8, 0x30, 0x11, 0xE0, 0x14,  
[0149] 0xE7, 0x03, 0xE3, 0x0B, 0xE4, 0xF7, 0xF4, 0xC5, 0xDC, 0x2D, 0x07, 0xF9,  
[0150] 0x27, 0xF0, 0xD9, 0xC1, 0xEF, 0x14, 0x26, 0xD7, 0x00, 0x1B, 0x0B, 0xDB,  
[0151] 0x3F, 0xF8, 0xF6, 0x06, 0x0F, 0x1B, 0xC8, 0xC1, 0x2C, 0x1B, 0x1E, 0x06, 0x1B,  
[0152] 0xFA, 0xC8, 0xF9, 0x0F, 0x18, 0xDF, 0xF8, 0x2D, 0xFC, 0x00, 0x0A, 0x22,  
[0153] 0xDD, 0x31, 0xF7, 0xC8, 0x20, 0xD3, 0xFC, 0xFC, 0xDD, 0x3F, 0x19, 0xD8,  
[0154] 0xE8, 0x0C, 0x1E, 0xE2, 0xC9, 0x03, 0xEC, 0x3F, 0x2B, 0xE0, 0x35, 0xC1,  
[0155] 0xFE, 0x11, 0xF9, 0x14, 0xE8, 0x06, 0x06, 0x24, 0xCE, 0xF3, 0x26, 0x3F, 0xFD,  
[0156] 0xCE, 0x2C, 0x12, 0x3C, 0x2C, 0xC2, 0xE3, 0x06, 0xD2, 0xC7, 0x0A, 0xDF,  
[0157] 0xD5, 0xD1, 0xC5, 0x15, 0xF2, 0xF1, 0x08, 0x02, 0xE6, 0xE2, 0x0A, 0xEB,  
[0158] 0x05, 0xDA, 0xE3, 0x06, 0x0E, 0x01, 0x03, 0xDC, 0x13, 0xE3, 0xFB, 0x36,

[0159] 0xE6, 0x14, 0x21, 0xFA, 0xC1, 0xC1, 0xE8, 0x0B, 0x0E, 0x17, 0x11, 0x2D, 0x11,  
[0160] 0xF0, 0x39, 0xE7, 0xF0, 0xE7, 0x2D, 0x03, 0xD7, 0x24, 0xF4, 0xCD, 0x0C,  
[0161] 0xFB, 0x26, 0x2A, 0x02, 0x21, 0xD8, 0xFA, 0xF8, 0xF0, 0xE8, 0x09, 0x19,  
[0162] 0x0C, 0x04, 0x1F, 0xCD, 0xFA, 0x12, 0x3F, 0x38, 0x30, 0x11, 0x00, 0xF0, 0xE5,  
[0163] 0x3F, 0xC3, 0xF0, 0x1E, 0xFD, 0x3B, 0xF0, 0xC1, 0xE6, 0xEB, 0x1F, 0x01,  
[0164] 0xFE, 0xF4, 0x23, 0xE4, 0xF0, 0xEB, 0xEB, 0x10, 0xE4, 0xC1, 0x3F, 0x0C,  
[0165] 0xEF, 0xFB, 0x08, 0xD8, 0x0E, 0xE4, 0x14, 0xC1, 0xC1, 0x0A, 0xE9, 0xFB,  
[0166] 0xEF, 0xE1, 0xE7, 0xF0, 0xD8, 0x27, 0xDA, 0xDC, 0x04, 0x0D, 0xDC, 0xFC,  
[0167] 0xDB, 0xD6, 0xD6, 0xE4, 0x0C, 0x27, 0xFC, 0xD0, 0x11, 0xE0, 0x04, 0xE3,  
[0168] 0x07, 0x00, 0xEC, 0x10, 0xD5, 0xEA, 0x08, 0xFF, 0xFC, 0x1D, 0x13, 0x05,  
[0169] 0xCA, 0xED, 0x0B, 0x10, 0x08, 0xF2, 0x01, 0x19, 0xCA, 0xFE, 0x32, 0x00,  
[0170] 0x20, 0x0B, 0x00, 0x3F, 0x1E, 0x16, 0x0C, 0xF1, 0x03, 0x04, 0xFD, 0xE8, 0x31,  
[0171] 0x08, 0x15, 0x00, 0xEC, 0x10, 0xED, 0xE6, 0x05, 0xCA, 0xF7, 0x1C, 0xC1,  
[0172] 0x22, 0x0D, 0x19, 0x2E, 0x13, 0x1E, 0xE7, 0x16, 0xED, 0x06, 0x2A, 0x3C,  
[0173] 0x0D, 0x21, 0x16, 0xC9, 0xD7, 0xFF, 0x0F, 0x12, 0x09, 0xEE, 0x1D, 0x23, 0x13,  
[0174] 0xDA, 0xE9, 0x1D, 0xD9, 0x03, 0xE1, 0xEF, 0xFA, 0x1E, 0x14, 0xC1, 0x23,  
[0175] 0xFE, 0x0B, 0xE5, 0x19, 0xC1, 0x21, 0xFE, 0xEC, 0x0E, 0xE1, 0x1D, 0xFF,  
[0176] 0x00, 0xF7, 0xEA, 0xD2, 0xD8, 0xD0, 0xF9, 0xE6, 0xFB, 0xFB, 0xDA, 0x06,  
[0177] 0x00, 0x03, 0xDF, 0xC1, 0x3F, 0xF3, 0x0D, 0xFA, 0x08, 0xFA, 0xF3, 0x00, 0x04,  
[0178] 0xE9, 0xF0, 0xF9, 0x0D, 0xF1, 0xE3, 0x1D, 0x26, 0xC4, 0x0D, 0x13, 0xE5,  
[0179] 0xE1, 0xF1, 0xF6, 0xEE, 0xF1, 0xFD, 0xC1, 0xF4, 0xE2, 0x23, 0xC1, 0x38,  
[0180] 0xC1, 0x3F, 0x2B, 0xFD, 0x39, 0x36, 0x1A, 0x2B, 0xC1, 0x01, 0x07, 0x0B,  
[0181] 0x25, 0xCC, 0xE7, 0x01, 0x24, 0xD8, 0xC9, 0xDB, 0x20, 0x28, 0x0C, 0x1A,  
[0182] 0x3F, 0xEA, 0xE7, 0xCD, 0xEC, 0xE0, 0xF2, 0x27, 0xDF, 0x20, 0xF0, 0xF1,  
[0183] 0xFD, 0x3F, 0x00, 0xFA, 0xE7, 0x21, 0xF9, 0x02, 0xD2, 0x0E, 0xEF, 0xFD,  
[0184] 0xD3, 0xE4, 0xFF, 0x12, 0x15, 0x16, 0xF1, 0xDE, 0xFD, 0x12, 0x13, 0xE7, 0x15,  
[0185] 0xD8, 0x1D, 0x02, 0x3F, 0x06, 0x1C, 0x21, 0x16, 0x1D, 0xEB, 0xEB, 0x14,  
[0186] 0xF9, 0xC5, 0x0C, 0x01, 0xFB, 0x09, 0xFA, 0x19, 0x0E, 0x01, 0x1B, 0xE8,  
[0187] 0xFB, 0x00, 0x01, 0x30, 0xF7, 0x0E, 0x14, 0x06, 0x15, 0x27, 0xEA, 0x1B,  
[0188] 0xCB, 0xEB, 0xF7, 0x3F, 0x07, 0xFB, 0xF7, 0xD8, 0x29, 0xEE, 0x26, 0xCA,  
[0189] 0x07, 0x20, 0xE8, 0x15, 0x05, 0x06, 0x0D, 0x0D, 0x1E, 0x1C, 0x0F, 0x0D, 0x35,  
[0190] 0xF7, 0x1B, 0x06, 0x30, 0x02, 0xFD, 0xE2, 0xCD, 0x2F, 0x35, 0xEB, 0x1A,  
[0191] 0x0D, 0xE9, 0xFC, 0x34, 0xE6, 0x17, 0x2C, 0x33, 0xF0, 0x13, 0xEF, 0x1B, 0x19,  
[0192] 0x23, 0xD1, 0xEF, 0xD5, 0xCB, 0xF7, 0xF1, 0x04, 0xF7, 0x27, 0xF9, 0x26, 0x02,  
[0193] 0xF7, 0xCB, 0x2A, 0x0A, 0xEA, 0xED, 0xEC, 0x04, 0xF2, 0x25, 0x17, 0xDB,  
[0194] 0x1E, 0xC1, 0x3C, 0xC9, 0xE4, 0xF1, 0x14, 0x03, 0x27, 0x25, 0x21, 0x1C, 0x14,  
[0195] 0xF4, 0x0F, 0x12, 0xE9, 0xEE, 0x15, 0xDC, 0xEE, 0x1F, 0x3F, 0xDE, 0xE7,  
[0196] 0x2C, 0xF0, 0xE2, 0x1D, 0xE5, 0x15, 0x07, 0x02, 0xDF, 0x06, 0xD3, 0x1F, 0x0E,  
[0197] 0xED, 0xFF, 0x29, 0xFF, 0xED, 0xD6, 0xD6, 0x1C, 0x11, 0xDE, 0xE2, 0x0E,

[0198] 0xEE, 0xD1, 0xD9, 0x02, 0x0F, 0xFE, 0xF0, 0xD9, 0xF6, 0xFC, 0xDA, 0x16,  
[0199] 0x03, 0xD2, 0xDD, 0x20, 0x04, 0xE8, 0x3F, 0xDE, 0x0C, 0xFB, 0xED, 0xC7,  
[0200] 0x1F, 0xC1, 0xCE, 0x02, 0xF1, 0x37, 0x0B, 0xE3, 0x20, 0xCE, 0x0D, 0xEB,  
[0201] 0x0A, 0xE3, 0xF3, 0xDC, 0x01, 0xD2, 0x02, 0x3F, 0x02, 0x25, 0xD5, 0xFC,  
[0202] 0xEB, 0xCE, 0x3F, 0x00, 0x3E, 0x2D, 0xE1, 0x19, 0x1C, 0x01, 0x28, 0xC1,  
[0203] 0x3F, 0x27, 0x3F, 0xF2, 0x0E, 0x3A, 0xDB, 0xF8, 0xE4, 0x34, 0x18, 0x16, 0x0C,  
[0204] 0xDD, 0x18, 0xED, 0xCB, 0x0F, 0xF0, 0x01, 0xFB, 0x14, 0xC1, 0x19, 0xCC,  
[0205] 0xEB, 0xEE, 0x19, 0x00, 0x17, 0x2B, 0xFC, 0x26, 0x0D, 0xEC, 0xF4, 0x2D,  
[0206] 0x2B, 0xE5, 0x25, 0x05, 0x10, 0x26, 0x1D, 0x3F, 0x3F, 0xFD, 0xDC, 0x18, 0xF0,  
[0207] 0xCB, 0xEF, 0x12, 0x1C, 0x1A, 0xF8, 0xFE, 0x29, 0x1A, 0xCB, 0x1A, 0xC2,  
[0208] 0x0E, 0x0B, 0x1B, 0xEB, 0xD5, 0xF8, 0xFD, 0x17, 0x0B, 0xFC, 0x00, 0xFA,  
[0209] 0x37, 0x25, 0x0D, 0xE6, 0xEE, 0xF0, 0x13, 0x0F, 0x21, 0x13, 0x13, 0xE1, 0x12,  
[0210] 0x01, 0x0A, 0xF1, 0xE7, 0xF3, 0x1A, 0xED, 0xD5, 0x0A, 0x19, 0x39, 0x09,  
[0211] 0xD8, 0xDE, 0x00, 0xF9, 0xE9, 0xEA, 0xFF, 0x3E, 0x08, 0xFA, 0x0B, 0xD7,  
[0212] 0xD7, 0xDE, 0xF7, 0xE0, 0xC1, 0x04, 0x28, 0xE8, 0x1E, 0x03, 0xEE, 0xEA,  
[0213] 0xEB, 0x1C, 0xF3, 0x17, 0x09, 0xD6, 0x17, 0xFA, 0x14, 0xEE, 0xDB, 0xE2,  
[0214] 0x2A, 0xD9, 0xC1, 0x05, 0x19, 0x00, 0xFF, 0x06, 0x17, 0x02, 0x09, 0xD9, 0xE5,  
[0215] 0xF3, 0x20, 0xDD, 0x05, 0xCB, 0x09, 0xF8, 0x05, 0xF1, 0x1F, 0xE5, 0x12, 0x25,  
[0216] 0xF8, 0x3F, 0xDC, 0xF0, 0xF2, 0xC5, 0x34, 0x21, 0x35, 0xCD, 0xCC, 0x23,  
[0217] 0x1E, 0x01, 0x0B, 0xFF, 0x10, 0xFE, 0xF9, 0xDF, 0xF9, 0xF5, 0xE5, 0x07, 0xE1,  
[0218] 0x25, 0x1C, 0xC9, 0x00, 0x29, 0xF3, 0x0A, 0x25, 0xED, 0xF8, 0xFB, 0x20,  
[0219] 0xF8, 0xC1, 0xE5, 0xE0, 0x0F, 0x2F, 0x3A, 0x01, 0xC8, 0xFD, 0xCA, 0xE1,  
[0220] 0x30, 0x04, 0x19, 0x03, 0x25, 0xF3, 0x24, 0x38, 0xEE, 0xC9, 0x2F, 0xE7, 0x0B,  
[0221] 0xFA, 0xF7, 0x1B, 0x0A, 0x0B, 0x2D, 0x2D, 0x0B, 0xE8, 0x08, 0xDB, 0x0B,  
[0222] 0x04, 0xE8, 0xD0, 0xEE, 0x18, 0xEF, 0x11, 0xC1, 0xD6, 0x15, 0x3F, 0xF5, 0xF4,  
[0223] 0x2A, 0x29, 0xEF, 0xF0, 0xFA, 0x36, 0x33, 0xED, 0x19, 0xDF, 0x11, 0x09, 0xF5,  
[0224] 0x18, 0xF1, 0x3F, 0x14, 0x0C, 0xD2, 0xFF, 0xFF, 0x34, 0x01, 0xE4, 0xF8, 0x03,  
[0225] 0x3F, 0xF8, 0x3E, 0x21, 0x22, 0xE2, 0x0F, 0xEF, 0x1A, 0xE4, 0xF5, 0x08, 0x15,  
[0226] 0xEF, 0xF3, 0xE4, 0xDF, 0xF6, 0xFC, 0xE8, 0x21, 0x06, 0x20, 0x02, 0x17, 0x1B,  
[0227] 0x3F, 0xDB, 0x16, 0x2C, 0xE0, 0xFA, 0xDA, 0xD8, 0xD3, 0x0B, 0x0E, 0x10,  
[0228] 0xED, 0xD5, 0xF0, 0x30, 0xD3, 0x13, 0x04, 0xE1, 0xFF, 0xFB, 0x3F, 0xE8,  
[0229] 0xEE, 0xE5, 0x0B, 0xEF, 0xEF, 0xE6, 0x2C, 0xD3, 0x00, 0x18, 0x26, 0xFE,  
[0230] 0xC1, 0x08, 0x16, 0xFE, 0xDC, 0x00, 0xE4, 0xF7, 0xDC, 0x0E, 0x2E, 0x1D,  
[0231] 0x18, 0x0A, 0x08, 0x37, 0xC9, 0x10, 0xD7, 0x17, 0x17, 0xFB, 0x11, 0xD5, 0x15,  
[0232] 0x1C, 0xD0, 0x3F, 0xF8, 0x00, 0x00, 0xED, 0xC1, 0xFF, 0x00, 0x1F, 0x2E, 0x00,  
[0233] 0x12, 0xE0, 0xE2, 0xF7, 0x13, 0xC1, 0x1C, 0x18, 0xF8, 0x3F, 0x2C, 0xEB,  
[0234] 0xCA, 0xE7, 0xF8, 0x03, 0xEE, 0x22, 0x17, 0xF9, 0x35, 0x14, 0x1C, 0x03, 0x09,  
[0235] 0x03, 0x01, 0x2B, 0xD4, 0xD2, 0xF8, 0xF6, 0xF5, 0x06, 0x03, 0xFE, 0xDA,  
[0236] 0xD3, 0xFF, 0x03, 0xEF, 0xFE, 0x09, 0x01, 0xC9, 0x02, 0xDF, 0xD8, 0x3C,

[0237] 0xF7, 0xF0, 0xEE, 0xD6, 0x3F, 0x21, 0x16, 0x08, 0x17} ;

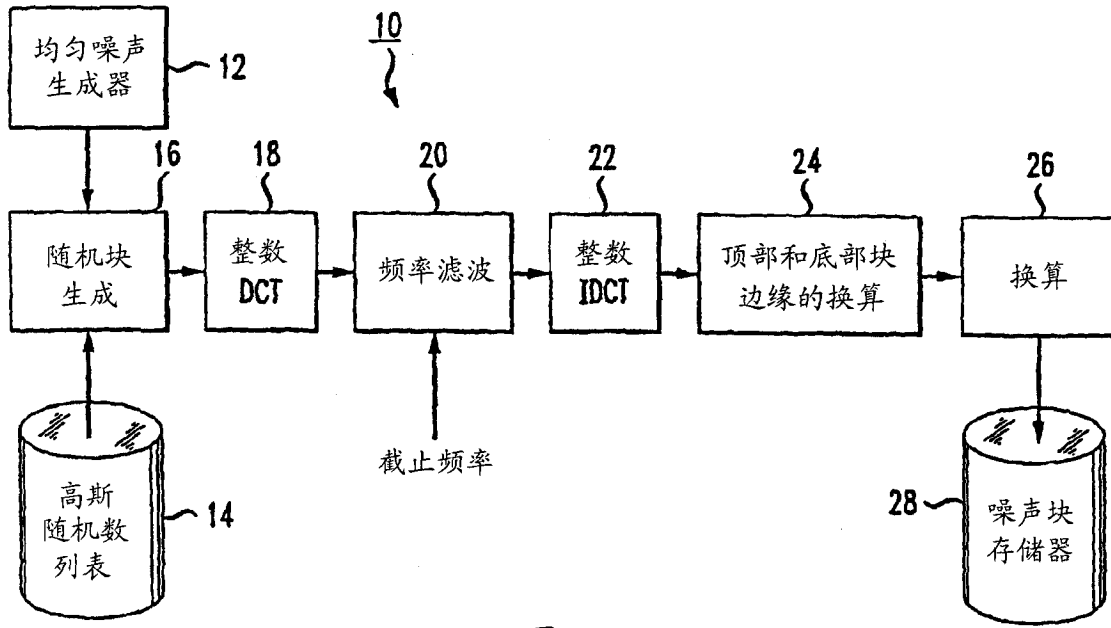


图 1

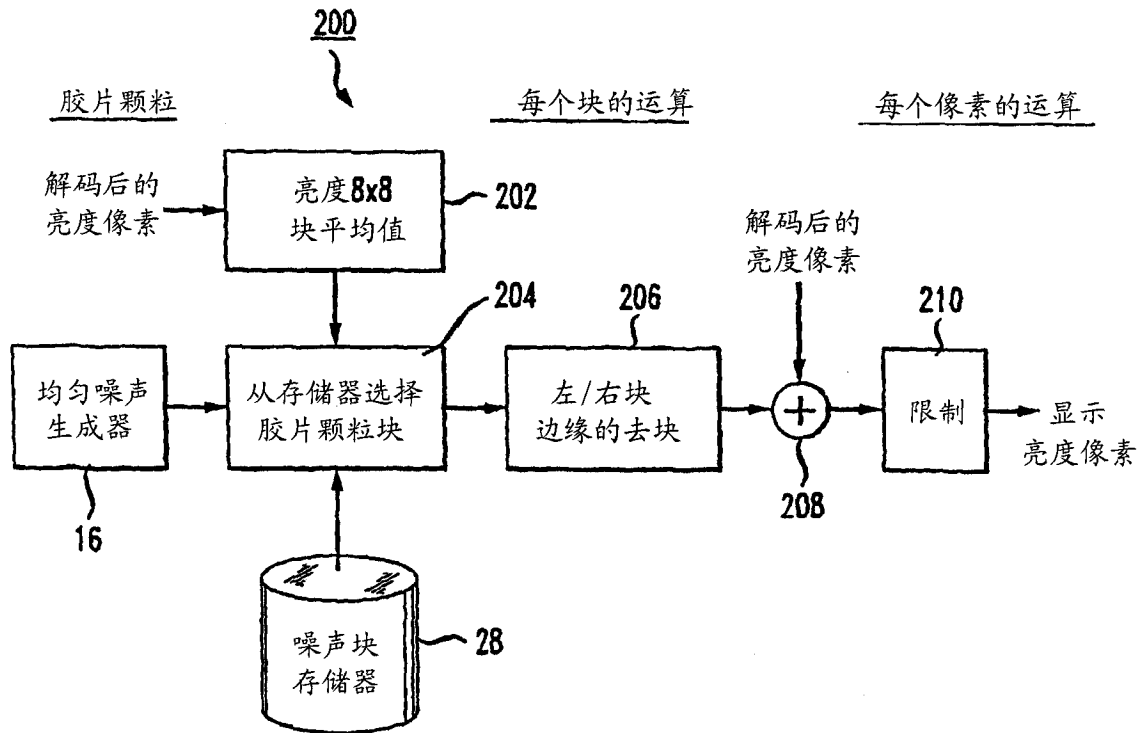


图 2