



(12) 发明专利申请

(10) 申请公布号 CN 101917616 A

(43) 申请公布日 2010. 12. 15

(21) 申请号 201010221966. X

H04N 7/34 (2006. 01)

(22) 申请日 2004. 02. 27

H04N 7/50 (2006. 01)

(62) 分案原申请数据

200480042588. 1 2004. 02. 27

(71) 申请人 TD 视觉有限公司

地址 墨西哥墨西哥城

(72) 发明人 M·R·古铁雷斯诺韦洛

(74) 专利代理机构 北京市柳沈律师事务所

11105

代理人 丁艺 沙捷

(51) Int. Cl.

H04N 7/26 (2006. 01)

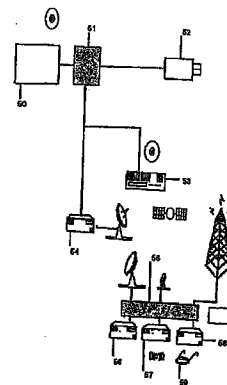
权利要求书 2 页 说明书 17 页 附图 5 页

(54) 发明名称

用于数字编码 3D 立体视频图像的方法和系统

(57) 摘要

本发明提供了一种用于数字编码 3D 立体视频图像的方法和系统。为了从数字视频流中获取三维图像,对现有的 MPEG2 编码器进行了某些修改;在编码处理的不同部分对软件和硬件进行了改变。实际上,视频数据流的结构和 video_sequence 被修改以包括在比特级识别 TDVision® 技术的图像类型所必需的标记。关于软件,将在以下处理中进行修改:video_sequence 头、识别标记修改、数据字段修改、和图像字段修改。关于硬件,在左和右信道的图像之间进行电比较,将差别作为 B 类型图像进行处理,即,执行错误校正,将结果与 TDVision® 技术识别符一起存储在临时缓冲器中。当编码信息时,DSP 被用于执行预测、比较、量化、和 DCT 函数应用处理,以形成 MPEG2 可兼容的压缩视频流。



1. 一种用于编码立体数字视频的系统,包括:
接收器,其配置成接收来自一个或多个输入视频信号的第一眼图像和第二眼图像;
处理器,其配置成计算来自所述第一眼图像和所述第二眼图像之间差别的差别图像;
以及
编码器,其配置成在一个或多个输出视频流中输出所述第一眼图像和所述差别图像,
其中所述一个或多个输出视频流包括信息头、所述第一眼图像和所述差别图像,并且其中
所述信息头包括告知解码器在所述一个或多个输出视频流中存在差别图像的数据。
2. 如权利要求 1 所述的系统,其中所述系统是电缆系统、高清晰电视系统、卫星电视系
统、光盘系统或数字电视系统中的至少一个的一部分。
3. 如权利要求 1 所述的系统,其中所述编码器进一步对所述差别图像进行错误校正。
4. 如权利要求 1 所述的系统,其中所述接收器包括照相机,所述照相机配置成拍摄立
体数字视频。
5. 如权利要求 1 所述的系统,其中所述处理器配置成计算作为 B 或 P 类型图像帧的所
述差别图像。
6. 如权利要求 1 所述的系统,其中所述编码器配置成将预定的一组比特存储在所述信
息头中,用来表示所述一个或多个输出视频流包括差别图像。
7. 如权利要求 1 所述的系统,其中所述编码器配置成将所述差别图像存储在差别图像
专用的数据结构中。
8. 如权利要求 1 所述的系统,其中所述一个或多个输出视频流包括多个视频流。
9. 如权利要求 1 所述的系统,其中所述一个或多个输出视频流是 MPEG 可兼容的视频
流。
10. 如权利要求 1 所述的系统,其中所述编码器配置成将所述信息头和第一眼帧存储
在第一视频流中,并将所述差别图像存储在预定数据结构中。
11. 一种立体视频编码系统,包括:
至少一个缓冲器,其配置成接收来自一个或多个输入视频信号的第一眼图像和第二眼
图像;
第一软件,其用于计算来自所述第一眼图像和所述第二眼图像之间差别的差别图像;
以及
第二软件,其用于将所述第一眼图像和所述差别图像存储在一个或多个输出视频流
中,其中所述一个或多个输出视频流包括信息头、所述第一眼图像和所述差别图像,并且其
中所述信息头包括告知解码器在所述一个或多个输出视频流中存在差别图像的数据。
12. 如权利要求 11 所述的系统,其中所述至少一个缓冲器是数字信号处理器的一部
分。
13. 如权利要求 11 所述的系统,其中所述系统是电缆系统、高清晰电视系统、卫星电视
系统、光盘系统或数字电视系统中的至少一个的一部分。
14. 如权利要求 11 所述的系统,进一步包括对所述差别图像进行错误校正的指令。
15. 如权利要求 11 所述的系统,其中所述第一软件计算作为 B 或 P 类型图像帧的所述
差别图像。
16. 如权利要求 11 所述的系统,其中所述第二软件将预定的一组比特存储在所述信息

头中,用来表示所述一个或多个输出视频流包括差别图像。

17. 如权利要求 11 所述的系统,其中所述第二软件将所述差别图像存储在差别图像专用的数据结构中。

18. 如权利要求 11 所述的系统,其中所述一个或多个输出视频流包括多个视频流。

19. 如权利要求 11 所述的系统,其中所述一个或多个输出视频流是 MPEG 可兼容的视频流。

20. 如权利要求 11 所述的系统,其中所述第二软件配置成将所述信息头和第一眼帧存储在所述第一视频流中,并将所述差别图像存储在预定数据结构中。

21. 一种用于编码立体数字视频的方法,包括:

接收来自一个或多个输入视频信号的第一眼图像和第一眼图像;

计算来自所述第一眼图像和所述第二眼图像之间差别的差别图像;以及

在一个或多个输出视频流中输出所述第一眼图像和所述差别图像,其中所述一个或多个输出视频流包括信息头、所述第一眼图像和所述差别图像,并且其中所述信息头包括告知解码器在所述一个或多个输出视频流中存在差别图像的数据。

22. 如权利要求 21 所述的方法,其中所述方法是由作为电缆系统、高清晰电视系统、卫星电视系统、光盘系统或数字电视系统中的至少一个的一部分的系统来执行的。

23. 如权利要求 21 所述的方法,进一步包括对所述差别图像进行错误校正。

24. 如权利要求 21 所述的方法,其中接收步骤包括用照相机拍摄立体数字视频。

25. 如权利要求 21 所述的方法,其中计算步骤包括计算作为 B 或 P 类型图像帧的所述差别图像。

26. 如权利要求 21 所述的方法,其中输出步骤包括将预定的一组比特存储在所述信息头中,用来表示所述一个或多个输出视频流包括差别图像。

27. 如权利要求 21 所述的方法,其中输出步骤包括将所述差别图像存储在差别图像专用的数据结构中。

28. 如权利要求 21 所述的方法,其中所述一个或多个输出视频流包括多个视频流。

29. 如权利要求 21 所述的方法,其中所述一个或多个输出视频流是 MPEG 可兼容的视频流。

30. 如权利要求 21 所述的方法,其中输出步骤包括将所述信息头和第一眼帧存储在所述第一视频流中,并将所述差别图像存储在预定数据结构中。

用于数字编码 3D 立体视频图像的方法和系统

[0001] 本申请是申请日为 2004 年 2 月 27 日且申请号为 200480042588.1 的发明专利申请“用于数字编码 3D 立体视频图像的方法和系统”的分案申请。

技术领域

[0002] 本发明涉及 3Dvisor™ 装置中的立体视频图像显示,特别涉及借助于数字数据压缩系统的视频图像编码方法,该数字数据压缩系统允许使用标准化的压缩技术进行三维信息存储。

背景技术

[0003] 目前,使用数据压缩技术以减小图像或者图像序列显示中的比特消耗。标准化工作由国际标准化组织的专家组执行。目前,这些方法通常称为 JPEG(联合图像专家组),和 MPEG(运动图像专家组)。

[0004] 这些技术的公共特性在于,图像块是借助于应用适用于块的变换来处理的,该变换通常通称为离散余弦变换(DCT)。形成的块被送往量化处理,随后用变长码来进行编码。

[0005] 变长码是可逆过程,其允许对已经用变长码进行编码的块进行精确重建。

[0006] 数字视频显示包括以 30 至 75Hz 频率连续显示或表示的一定数量的图像帧(30 至 96fps)。每一个图像帧仍然是由像素阵列根据特定系统的显示分辨率形成的图像。例如,VHS 系统的显示分辨率为 320 列 480 行,NTSC 系统的显示分辨率为 720 列 486 行,并且,高清晰电视系统(HDTV)的显示分辨率为 1360 列 1020 行。关于低分辨率的数字化形式,320 列乘 480 行的 VHS 格式,两小时长度的电影可等于 100GB 的数字视频信息。为了比较,传统压缩光盘的容量约为 0.6GB,磁盘的容量为 1-2GB,并且目前的压缩光盘的容量为 8GB 或者更大。

[0007] 由于存储和传输这种大信息量的限制,已经建立了几种标准的压缩处理。这些视频压缩技术使用连续图像帧之间的相似特性,称为时间空间相关性,以提供逐帧压缩,逐帧压缩是根据从帧到帧的像素表示进行的。

[0008] 我们在电影院和电视屏幕上看到的所有图像都是根据以很高的速率显示完整图像(静态图像,如照片)的原理。当它们以每秒 30 帧的速度(30fps)以快速并且连续的方式被显示出来时,由于人眼的视觉暂留(retention),我们把它们看成动画图像。

[0009] 为了将图像编码成以连续方式显示并形成视频信号,每一个图像都需要被分成行,其中每行依次被分为图像元素或像素,每一个像素具有两个相关联的值,即,亮度和色度。亮度表示每个点上的光强,而色度表示颜色,该颜色为定义的可以用三个字节来表示的颜色空间(RGB)的函数。

[0010] 图像被显示在水平-垂直网格(raster)上,从上到下,从左到右循环地显示。显示的线数和频率可以由于格式不同而改变,例如,NTSC、PAL、或者 SECAM。

[0011] 理论上,可以为每一个亮度、色度 U 和色度 V 像素赋值,但是这表示四个字节(一个用于色度,三个用于颜色),在 NTSC 的 480 行乘 720 列格式每秒约 30 帧下,结果为

4×480×720×30, 大约每秒 40MB 的存储量, 由于可用带宽的限制这是很难存储和传输的。当前, 已经能够将色度数据减小为 1:4 像素; 即, 每四个像素取一个颜色采样, 并且复制相同的信息用于缺失的三个像素, 人眼并不能感知到差别; 这些格式为, 即:

[0012] 4:4:4(在 4×4 = 16 像素组中, 四个亮度采样和四个色度采样)。

[0013] 4:2:2(在 4×2 = 8 像素组中, 四个亮度采样和两个色度采样)。

[0014] 4:1:1(在 4×1 = 4 像素组中, 四个亮度采样和一个色度采样)。

[0015] MPEG1 中的 4:2:0(在 4×2 = 8 像素组中, 八个亮度采样, 水平像素之间的两个色度采样)。

[0016] MPEG1 中的 4:2:0(在 4×2 = 8 像素组中, 八个亮度采样, 垂直像素之间的两个色度采样)。

[0017] 即使当以这种方式减少信息时, 用 NTSC 格式以 4:2:0 的质量存储一秒钟的信息所必需的数字信息量是 15MB, 或者, 对于两小时长的文件是 108GB。

[0018] 现有几种用于从二维视频序列进行三维场景重建的方法。考虑到近来技术发展以及考虑到将来的发展, MPEG4 标准试图提供时空相关的图形编码媒介, 它将是立体图像、工程应用的设计和制造中的重要工具。在重建场景的几何模型处创建虚拟空间。例如, 2003 年 12 月 9 日提交的授予 Cecile Dufour 的第 6, 661, 914 号 USP, 其中描述了新的三维重建方法, 场景连续性用简单的照相机取得, 图像的轮廓被重建, 每个场景中隐藏部分深度稍后被投影并经过提取处理。

[0019] 在图像处理的领域中, 许多人做出了有价值的贡献, 例如: 2003 年 10 月 21 日授予 Itokawa 的第 6, 636, 644 号 USP, 其涉及使用 MPEG4 的图像处理, 其中跨越图像边界延伸的图像色度值被提取出来, 由此实现了编码的更高效率, 并且能够实现图像轮廓的自然色重现。

[0020] 现有几种用于对视频信号进行编码的方法和方案, 例如, 2003 年 10 月 14 日授予 Kleihorst et al. 的第 6, 633, 676 号 USP, 该方法被应用于照相机系统中的编码器检测器, 用运动补偿 (I. B. P.) 对视频信号进行编码, 并生成高分辨率图像, 该图像是先前图像的插值结果, 总之, 确定视频信号中更感兴趣的区域, 其总共占用较少的存储空间。

[0021] 图像压缩编码主要用于以高效方式存储或传输数字图像, 一种压缩数字图像编码的方法使用 DCT, 因为这是例如 JPEG 和 MPEG 的公共标准中的一种主要技术。2002 年 2 月 5 日授予 Boon 的第 6, 345, 123 号 USP 描述了一种通过普通的 DCT 方法变换系数来进行数字图像编码的方法, 对上述系数进行量化处理, 以便于将它们变换成预先写入的量化级数, 最后, 变长编码处理被应用到量化和转换后的系数, 将它们与变长编码表进行比较。

[0022] 图像被分为多个小区域, 以进行编码, 小区域互相邻接, 从一个区域采样, 就可预测下一个图像区域的情况。在 2000 年 11 月 14 日授予 Boon 等人的第 6, 148, 109 号 USP 中使用的该预测编码方法, 其中对所生成的小区域之间的差别的图像数据进行编码并将该生成的图像数据提取出来。

[0023] 2000 年 8 月 1 日授予 Murakami et al. 的第 6, 097, 759 号 USP 描述了用于场编码图像的邻近的块编码系统。块的图案包括一个单独的多块区域和一个非交织的块; 而且, 编码系统查询奇数和偶数场运动, 以产生运动补偿预测信号, 从而提供高效率编码。

[0024] 授予 Katata 等的第 5, 978, 515、5, 963, 257、5, 815, 601 号 USP 专利, 涉及用于以这

样的方式编码图像数据的图像编码器：它们增强所选区域相对于其它区域的图像质量，而不增加用于描述该所选区域的数据量。

[0025] 1996年11月26日授予Gisle的第5,579,413号USP描述了用于将数据信号变换成量化的图像块、并将其转换成变长编码数据信号的方法，其中每个事件都用三维向量来表示。需要使用允许在较小空间中存储相同内容的数据压缩系统，专家组致力于提出压缩信息和显示图像的方法；但在实现细节与MPEG相符合时，在所有软件和硬件开发者可创建执行处理的新方式的目标下，并不提及实现的细节。当前，MPEG2是世界范围的标准，被电视和视频以及音频相关的公司广泛使用。

[0026] 音频和视频被打包到基本包(PES)中，所述音频和视频包交织在一起，以创建MPEG2数据流。每个包具有用于音频和视频在播放时间同步的时间标识(时间标签)，例如，对于每三个视频帧，结合一个音频帧。

[0027] MPEG有两种在系统的数据流中交织视频和音频的不同方法：

[0028] 传输流被用在错误概率较高的系统中，例如卫星系统，其易于受到干扰。每个包的长度是188字节，以标识头开始，这使得可以识别间隙和修复错误。不同的音频和视频程序能够同时在单一传输流上通过传输流传输；由于信息头，它们可为独立的并且可单独解码并且集成到许多程序中。

[0029] 程序流被用在错误概率较低的系统中，如在DVD播放系统中。在这种情况下，包具有可变长度和实际上比传输流中所使用的包大的尺寸。作为主要特性，程序流仅允许单一的程序内容。

[0030] MPEG2标准下的视频系统允许隔行类型的和逐行类型的视频图像的编码。

[0031] 即，逐行视频格式被存储在整帧(帧图片,fp)中，并且在隔行的视频格式中，它可以两种方法被存储，通过整帧图像(帧图片)或者通过场图像(场图片)。

[0032] 在压缩格式中，存在三种MPEG2格式的图像：

[0033] 帧内编码(I)，它们的信息被编码为图像自身内部数据的函数。

[0034] 预测编码(P)，其信息单独地取决于其它将来时间点处的数据。

[0035] 双向预测编码(B)，其信息取决于过去和将来时间点处的数据。

[0036] 按顺序，有三种应用到上面的包的压缩类型，例如，时间预测、压缩和空间压缩。

[0037] 时间上的预测压缩涉及两个时间上不同的帧，但是它们具有相互运动，该时间上的预测压缩利用帧之间的图像差别很小的事实。

[0038] 空间压缩将位于一个相同帧(帧内编码的)内的信息压缩，例如，在 100×100 像素图像中，3个字节用于颜色，1个字节用于亮度，如果需要存储该信息，则每帧需要40KB；相反，如果该图像是全白的，可以表示为一个颜色：255R,255G,255B, $X_{start} = 0, Y_{start} = 0, X_{end} = 99, Y_{end} = 99$ ，这将指示该整个区域是白色的；仅使用7KB或8KB，而没有使用40KB。这样，就实现了MPEG压缩；该处理步骤比较复杂，在本发明的范围之外。

[0039] 类型(I)图像是仅包含自身图像的，它们不涉及任何先前的或后续的图像，因此不使用时间预测压缩，而只作为它自身空间的函数。

[0040] 类型(P)图像是根据参考图像，以对它们本身进行编码的，因此它们使用时间预测压缩，也使用空间压缩。这些图像可涉及(I)类型图像或者其它(P)类型图像，但是仅使用一种图像参考图像。

[0041] (B) 类型图像需要先前的和后续的两个参考图像,以进行重建,该类型的图像具有最佳压缩指标。用于获得 (B) 类型图像的参考图像可为 (P) 或者 (I) 类型,而不能是 (B) 类型。

[0042] 编码和解码序列不同。

[0043] 为了降低信息量,完整图像被分为称为宏块的单元的整帧;每个宏块包括 16 像素 \times 16 像素,从上到下从左到右排列和命名,在屏幕上创建宏块矩阵阵列,宏块以顺序的形式在信息流中被发送,即,0,1,2,3,...,n。

[0044] 具有 (I) 类型图像的宏块仅包含自身的空间压缩;(P) 类型图像可包含 (P) 类型的宏块,以便于参考先前的图像,有包含内部编码的宏块(交织的宏块)的可能性,并不受到限制。

[0045] (B) 类型的图像也可由内部编码(交织)类型的宏块形成,其涉及先前的图像、后续的图像,或者二者均涉及。

[0046] 按顺序,宏块被分成多块,一个块是 8×8 数据或采样矩阵;由于色度格式被分为 4:4:4 格式的形式需要一个亮度采样 Y,一个色度采样 Cr,一个色度采样 Cb,因此,4:4:4 格式的每个宏块需要 12 个块,在 4:2:0 格式中,每个宏块需要 6 个块。

[0047] 一组连续宏块表示一片;一片中可以有任意数目的宏块,它们必须属于单行的,以与各宏块相同的方式,这些片从左到右从上到下命名。各片不必覆盖所有的图像,因为编码后的图像不需要对每个像素采样。

[0048] 一些 MPEG 标准需要图像必须完全符合的固定的片结构。使用适当的硬件和软件算法的组合允许 MPEG 图像压缩。

[0049] 编码后的数据是具有特定块信息的字节、宏块、场、帧、图像和 MPEG2 格式视频。

[0050] 信息必须被分成块组,并且从信息编码例如 (VLC) 获得的结果是线性比特-字节流。

[0051] 其中 VLC(变长解码器)是用较短码代替最频繁使用的符号并用较长码代替较少发生的那些符号的压缩算法。压缩后的该信息占用较少的空间,并能够较快地通过网络传输。然而,它不是容易编辑的格式,并且需要使用查阅表进行解压缩。

[0052] 反向扫描,信息必须被分成块组,并且当借助 VLC 对信息进行编码时,所得到的是线性流。块是 8×8 数据矩阵,因此有必要将线性信息转换成 8×8 的方形矩阵。根据其是逐行图像还是隔行图像,这是以向下之字形方式进行的,在两种序列类型中都是从上到下从左到右的。

[0053] 反向量化,在于简单地将每个数据值乘以一个因子。当编码的时候,块中的多数数据被量化以去除人眼所不能够感知到的信息,量化允许获得较大的 MPEG2 流的转换,并且还需要执行在解码处理中的反向处理(反向量化)。

[0054] MPEG 视频序列结构:这是 MPEG2 格式中使用的最大结构,并具有以下格式:

[0055] 视频序列 (Video_Sequence)

[0056] 序列头 (Sequence_header)

[0057] 序列扩展 (Sequence_Extension)

[0058] 用户数据 (0) 和扩展 (Extension_and_User_Data(0))

[0059] 图像组头 (Group_of_Picture_Header)

- [0060] 用户数据 (1) 和扩展 (Extension_and_User_Data(1))
- [0061] 图像头 (Picture_Header)
- [0062] 编码图像扩展 (Picture_Coding_Extension)
- [0063] 用户数据 (2) 和扩展 (Extension_and_User_Data(2))
- [0064] 图像数据 (Picture_Data)
- [0065] 片 (Slice)
- [0066] 宏块 (Macroblock)
- [0067] 动作矢量 (Motion_Vectors)
- [0068] 编码块图案 (Coded_Block_Pattern)
- [0069] 块 (Block)
- [0070] 最终序列编码 (Sequence_end_Code)
- [0071] 视频序列包括三种结构, 视频序列被应用于 MPEG1 和 MPEG2 格式, 以便于区分各版本, 必须验证在序列头后面存在序列扩展; 如果序列头后面不接有序列扩展, 则该流是 MPEG1 格式的视频流。

发明内容

[0072] 本发明的一个目的是提供一种立体 3D 图像数字编码方法和系统, 其提供用于在 **3Dvisors®** 中传输、接收和显示的编码的数据。

[0073] 本发明的另一个目的是提供一种编码方案, 其中, 视频数据流 video_sequence 的结构被修改, 并且在比特级包括识别标记。

[0074] 本发明的又一个目的是提供一种 3D 图像数字编码软件处理, 以 video_sequence、识别标记、数据字段, 和图像字段被修改的方式。

[0075] 本发明的又一个目的是提供一种 3D 图像数字编码硬件处理, 其方式是, 在左和右通道之间进行电比较, 对图像之间的差别进行错误校正, 将处理后的图像存储在具有 **TDVision®** 技术识别符的 video_sequence 中。

[0076] 本发明的又一个目的是提供一种 3D 图像数字编码硬件处理, 其方式是, DSP 的输入缓冲器的存储器被增为双倍的, 两个独立的视频信号可同时输入, 并且 DSP 能够对二个视频信号的输入缓冲器进行比较。

附图说明

[0077] 图 1 表示立体 3D 视频图像编码的硬件和软件改变;

[0078] 图 2 表示 MPEG2-4 可兼容的立体 3D 视频图像的编译处理;

[0079] 图 3 表示用于编译 MPEG2-4 可兼容的立体 3D 视频图像的软件格式;

[0080] 图 4 表示用于编译 MPEG2-4 可兼容的立体 3D 视频图像的硬件格式;

[0081] 图 5 表示本发明的编码器所属的技术分支图: 即, 立体 3D 图像处理, 其编码、解码、经由电缆的传输、卫星和 DVD、HDTV 和 **3DVisors®** 显示。

具体实施方式

[0082] 为了实现从数字视频流中获取三维图像的目的, 已经通过在编码处理的不同部分

中对硬件和软件进行改变而对现有的 MPEG2 编码器进行修改。如图 1 中所示, MPEG2-4 可兼容的 TDVision® 编码器 (1) 具有其自身的编码处理 (2), 是通过软件 (3) 和硬件 (4) 的改变而实现的。

[0083] 在图 2 中, 显示出本发明的编码器实体的编译处理, 实际上, 图像 (10) 被拍摄并被提交到动作补偿和错误检测处理 (11); 应用离散余弦变换函数来改变频率参数 (12), 随后应用量化矩阵 (13), 以执行归一化处理, 应用用于行转换处理的矩阵 (14), 在此处有执行变长编码 (15) 的可能性, 并且, 最后, 得到具有编码数据 (16) 的视频序列。为了执行该编译处理, 必须遵照格式 (30, 图 3) 或者 MPEG2 兼容的 3D 图像编译方法, 实际上, 如图 3 所示, 必须对 video_sequence (31) 在 sequence_header (32)、user_data (33)、sequence_scalable_extension (34)、picture_header (35)、picture_coding_extension (36) 和 picture_temporal_scalable_extension (37) 结构中进行修改, 从而获得适于用 TDVision® 立体照相机拍摄的立体 3D 数字图像的编译格式。

[0084] 视频数据流的结构和 video_sequence 必须被修改以包括在比特级上识别 TDVision® 技术编码图像类型所必要的标记。

[0085] 在以下编码阶段进行的修改, 即, 当以 MPEG2 (软件) 编码双图像时; 当用硬件编码图像时。

[0086] 软件:

[0087] 修改 video_sequence 头。

[0088] 识别识别标记。

[0089] 修改数据字段。

[0090] 修改图像字段。

[0091] 硬件:

[0092] 在左和右通道之间进行电比较。

[0093] 将差别作为 B 类型图像进行处理 (错误校正)。

[0094] 随后, 将其与 TDVision® 识别符一起存储。

[0095] 对附加缓冲器进行改变。

[0096] 其结果被保存并存储到辅助缓冲器中。

[0097] 实际上, DSP 缓冲器的输入存储器被增为双倍; 允许同时输入对应于来自立体 TDVision® 照相机的立体左-右现有信号的两个独立视频信号; DSP 能够比较两个视频信号的输入缓冲器。

[0098] 硬件编码处理是作为单一视频输入通道的函数, 以标准 MPEG2 方式执行的, 获得两个信号 (左和右) 并进行电比较, 得到左和右信号之间的比较的差别, 上述差别被存储到临时缓冲器中, 计算关于左信号的亮度和色度的错误校正; 应用 DCT (离散余弦变换) 函数, 并将信息存储到 B 类型块中:

[0099] a) 在 USER_DATA () (SW) 识别结构中

[0100] b) 在 PICTURE_DATA3D () 结构中

[0101] 在下一帧中继续。

[0102] 在图 4 中的框图中显示出硬件, 实际上, 获得左信号 (41) 和右信号 (42), 两个信号都存储在临时缓冲器 (43) 中, 比较左和右信号之间的差别, 计算错误差别并存储信息

(45),对正确图像进行编码(46),作为“I”、“B”或者“P”类型图像执行编码(47),并最后存储到 video_sequence 中(48)。

[0103] 复制将由 DSP 处理的存储是重要的,并且可能设置高达 8 个输出缓冲器,这允许立体图像在例如 TDVision® 的 3DVisor® 的装置上的先前的和同时的显示。

[0104] 实际上,两个信道必须在调用 Texas Instruments TMS320C62X DSP 的编程 API 的时候被初始化。

[0105] MPEG2VDEC_create(const IMPEG2VDEC_fxns*fxns, constMPEG2VDEC_Params*params)。

[0106] 其中 IMPEG2VDEC_fxns y MPEG2VDEC_Params 是为每个视频通道定义操作参数的指针结构,例如:

[0107] 3DLhandle = MPEG2VDEC_create(fxns3DLEFT, Params3DLEFT)。

[0108] 3DRhandle = MPEG2VDEC_create(fxns3DRIGHT, Params3DRIGHT)。

[0109] 从而使得两个视频通道能够被解码,并能够获得两个视频处理器,一个用于左-右立体通道。

[0110] 需要两个显示输出缓冲器,借助于软件,将定义两个缓冲器中的哪一个必须通过调用 AP 函数显示输出:

[0111] 即, MPEG2VDEC_APPLY(3DRhandle, inputR1, inputR2, inputR3, 3doutright_pb, 3doutright_fb)。

[0112] MPEG2VDEC_APPLY(3DLhandle, inputL1, inputL2, inputL3, 3doutleft_pb, 3doutleft_fb)。

[0113] 其中 3DLhandle 是指向由 DSP 的创建函数返回的句柄的指针, input1 参数是 FUNC_DECODE_FRAME 或 FUNC_START_PARA 地址, input2 是指向外部输入缓冲器地址的指针, 并且, input3 是外部输入缓冲器的大小。

[0114] 3doutleft_pb 是参数缓冲器的地址, 并且, 3doutleft_fb 是将要存储解码图像的输出缓冲器的开始部分。

[0115] 时间码和时间标签将被用于以顺序和同步的方式输出到最后的装置。

[0116] 软件和硬件处理的集成是通过被称为 DSP 的装置执行的, DSP 执行多数硬件处理。这些 DSP 通过由制造商所提供的 C 和汇编语言混合进行编程。每个 DSP 有其自己的 API, 包括位于 DSP 中并被软件调用的函数列表或过程调用。

[0117] 通过该参考信息,作出用于 MPEG2 格式可兼容的 3D 图像编码的本申请。

[0118] 实际上,在视频序列的开始,总出现序列头和序列扩展。序列扩展的重复必须与第一个相同。相反,与第一次发生相比,序列头重复变化很少,仅有定义量化矩阵的部分应该改变。因为序列重复允许对视频流的随机访问,即,解码器可在视频流的中间开始播放,这是可以实现的,仅需要搜索先前的序列头和序列扩展,以便能够解码视频流中的后续图像。这也会发生在不能从开头开始的视频流上,例如在程序已经开始时开启卫星解码器。

[0119] 即,序列头提供视频流上的较高信息级,为了明确说明,还指出对应于每一个信息级的比特数,最高有效比特位于序列扩展(Sequence_Extension)结构内,它是通过以下结构形成的:

[0120] Sequence_Header

[0121]	字段	#bits	描述
[0122]	Sequence_Header_Code	32	Sequence_Header 开始
[0123]			0x00001B3
[0124]	Horizontal_Size_Value	12	12 个较低有效比特用于宽度
[0125]	Vertical_Size_Value	12	12 个较低有效比特用于高度
[0126]		4	图像方面
[0127]			0000 禁用
[0128]	Aspect Ratio Information(纵横比		0001n/a TDVision®
[0129]	信息)		0010 4:3 TDVision®
[0130]			0011 16:9 TDVision®
[0131]			0100 2.21:1 TDVision®
[0132]			0111 将执行逻辑“与”以获
[0133]			得与 2D 系统的向后兼容性。
[0134]			0101...1111 保留
[0135]		4	0000 禁用
[0136]	Frame rate code(帧速率码)		0001 TDVision® 格式中的
[0137]			24,000/1001(23.976)
[0138]			0010 TDVision® 格式中的 24
[0139]			0011 25 “
[0140]			0100 30,000/1001(29.97) “
[0141]			0101 30 “
[0142]			0110 50 “
[0143]			0111 60,000/1001(59.94) “(将
[0144]			执行逻辑“与”以获得与 2D
[0145]			系统的向后兼容性。)
[0146]			1000 60
[0147]			1111 保留
[0148]	Bit_rate_value	18	Video_stream 比特率的 18 个
[0149]			较低有效比特 (bit_rata =
[0150]			400×bit_rate_value +
[0151]			bit_rate_extension << 18)
[0152]			最高有效比特位于
[0153]			sequence_extension 结构中。
[0154]	Marker_bit	1	总为 1(防止 start_code_失
[0155]			效)。
[0156]	Vbv_buffer_size_value	10	vbv_buffer_size 的 10 个较低
[0157]			有效比特,其确定视频缓冲
[0158]			检验器 (VBV) 的大小,VBV
[0159]			为一种结构,其用于确保数

[0160]			据流可被用于解码有限大小的缓冲数据而不超过或在缓冲器中保留过多自由空间。
[0161]			
[0162]			
[0163]	Constrained_parameters_flag	1	总为 0, 不用在 MPEG2 中。
[0164]	Load_intra_quantizer_matrix	1	指示 intra-coded 的量化矩阵是否可用。
[0165]			
[0166]	If load_intra_quantizer_matrix	8×64	如果指示量化矩阵, 则这里
[0167]	Non_intra_quantizer_matrix(64)		必须指出, 它是 8×64 的矩阵。
[0168]	Load_non_intra_quantizer_matrix	1	如果 non-intra-quantized 矩阵
[0169]			可用, 则该标记必须被激活。
[0170]	If load_non_intra_quantizer_matrix	8×64	如果先前的标记被激活, 形
[0171]	Non_intra_quantizer_matrix(64)		成量化矩阵的 8×64 数据被存
[0172]			储在这里。
[0173]	Sequence_extension		
[0174]	字段	#bits	描述
[0175]	Extension_Start_Code	32	开始, sequence_extension, 总为
[0176]			0x000001B5
[0177]	Extension_Start_code_Identifier	4	用扩展类型 0×1 标识
[0178]	Profile_and_level_indication	8	定义类 (profile) 和视频流级
[0179]	progressive_sequence	1	1 = 帧, 0 = 帧和场
[0180]	Chrome_format	2	00 保留
[0181]			01 4:2:0
[0182]			10 4:2:2
[0183]			11 4:4:4
[0184]	Horizontal_size_extension	2	sequence_header 扩展
[0185]	Vertical_size_extension	2	sequence_header 扩展
[0186]	Bit_rate_extension	12	sequence_header 扩展
[0187]	Marker_bit	1	总为 1
[0188]	Vbv_buffer_size_extension	8	sequence_header 扩展
[0189]	Low_delay	1	1 = 不具有 B 类型图像, 还可能引
[0190]			起正常播放期间的 VBV 缓冲器
[0191]			的利用不足 (称为 BIG 图像)
[0192]			0 = 可包含 B 类型图像; 但是不可
[0193]			具有 BIG 图像, 它不会引起 VBV
[0194]			缓冲器的利用不足。
[0195]	Frame_rate_extension_n	2	
[0196]	Frame_rate_extension_d	5	
[0197]	Next_start_code()		
[0198]	扩展和用户数据 (i)		

[0199] 它是一种用于存储其它结构的容器,并且不具有其自身的数据,基本上它是一系列 extension_data(1) 和 user_data() 结构,在一些情况下,结构可以完全为空。

[0200] Extension_data(i)

[0201] 该结构包含简单的结构扩展。所包含的扩展结构类型取决于 (i) 的值, (i) 的值可以是 1 或 2 的值。如果它等于“0”,则 data_extension 跟随 sequence_extension 并且 extension_data(i) 可包含以下二者:一个 sequence_display_extension 或者一个 sequence_scalable_extension。如果 i = 2,则该结构跟随 picture_coding_extension,其可包含 quant_matrix_extension()、copyright_extension()、picture_display_extension()、picture_spatial_scalable_extension(), 或者一个 picture_temporal_scalable_extension。该结构总是以 0x000001B5 开始。

[0202] User_data

[0203] user_data 结构允许应用程序的特定数据被存储在视频序列 (video_sequence) 内。MPEG2 规范没有定义该函数的格式,也没有定义用户数据的格式。结构以 user_data_start_code = 0x000001B5 开始,并包含任意数量的数据 (user_data),其持续到数据流 (stream) 中的下一个开始码为止。唯一的条件是,不能有多于 23 个连续的零,因为它会被认为是开始码。

[0204] Sequence_display_extension()

[0205] 该结构提供了不在解码处理中使用的信息,涉及编码内容的对正确显示解码视频有帮助的信息。

[0206] Sequence_display_extension()

[0207]	字段	bits#	描述
[0208]	Extension_start_code_identifier	4	必须为 2,标识开始
[0209]	Video_format	3	000 组件
[0210]			001 PAL
[0211]			010 NTSC
[0212]			011 SECAM
[0213]			100 MAC
[0214]			101 未指定
[0215]			110 保留,TDVision®
[0216]			111 保留,TDVision®
[0217]	Color_description	1	0 = 不指定颜色参数。
[0218]			1 = 包含下面的 3 个颜色参数。
[0219]	Color_rprimaries	8	0 禁用
[0220]			1 ITU-R-B T. 709
[0221]			推荐
[0222]			2 未指定的视频
[0223]			3 保留
[0224]			4 ITU-R-BT. 470-2
[0225]			系统 M

[0226]			推荐
[0227]			5 ITU-R-BT. 470-2
[0228]			推荐
[0229]			系统 B, G
[0230]			6 SMPTE 170M
[0231]			7 SMPTE 240M
[0232]			8-255 保留
[0233]	Transfer_characteristics	8	0 禁用
[0234]			1 ITU-RB T. 709
[0235]			推荐
[0236]			2 未指定的视频
[0237]			3 保留
[0238]			4 ITU-R-B T. 470-2
[0239]			系统 M
[0240]			推荐
[0241]			5 ITU-R-B T. 470-2
[0242]			系统 B, G
[0243]			推荐
[0244]			6SMPTE 170M
[0245]			7SMPTE 240M
[0246]			8 真实转移特性
[0247]			255 保留
[0248]	Matrix_coefficients	8	0 禁用
[0249]			1ITU-RBT8709
[0250]			推荐
[0251]			2 未指定的视频
[0252]			3 保留
[0253]			4FCC
[0254]			5ITU-R-BT. 470-2
[0255]			系统 B, G
[0256]			推荐
[0257]			6SMPTE 170M
[0258]			7SMPTE 240M
[0259]			8-255 保留
[0260]	Display_horizontal_style	14	未在 MPEG2 中指定
[0261]	Marker_bit	1	总为 1
[0262]	Display_vertical_size	14	未在 MPEG2 中指定
[0263]	Next_start_code		
[0264]	Sequence_scalable_extension		

[0265] 该结构必须存在于每个可扩展的视频流中,其包含基层和一个或多个增强层。存在不同类型的 MPEG2 可扩展性,主层的可扩展性的一个实例是,它包含视频内容的标准定义,而扩展层具有增加定义的附加的数据。

[0266]	Sequence_scalable_extension		
[0267]	描述	bits#	描述
[0268]	Extension_start_code_identifier	4	总为 5
[0269]	Scalable_mode	2	00 分区数据
[0270]			01 空间可扩展
[0271]			10SNR 可扩展
[0272]			11 时间可扩展
[0273]	Layer_id	4	层数 (0)
[0274]	Lower_layer_prediction_horizontal_size	14	
[0275]	Marker_bit	1	
[0276]	Lower_layer_prediction_vertical_size	14	
[0277]	Horizontal_subsampling_factor_m	5	
[0278]	Horizontal_subsampling_factor_n	5	
[0279]	Vertical_subsampling_factor_m	5	
[0280]	Vertical_subsampling_factor_n	5	
[0281]	Picture_mux_enable	1	
[0282]	Mux_to_progressive_sequence	1	
[0283]	Picture_mux_order	3	
[0284]	Picture_mux_factor	3	
[0285]	Group_of_picture_header()		
[0286]	该结构标记图像组的开始。		
[0287]	Group_of_picture_header()		
[0288]	字段	bits#	描述
[0289]	Group_start_code	32	0x000001B8
[0290]	Time_code	25	group_of_picture_header 之前第一图像的时间标签
[0291]			
[0292]			Dromp_frame_flag-1
[0293]			Time_code_hours-5
[0294]			Time_code_minutes-6
[0295]			Marker_bit-1
[0296]			Time_code_seconds-6
[0297]			Time_code_pictures-6
[0298]	Closed_gop	1	如果 1 = B 图像,不参照先前的图像
[0299]	Broken_link	1	1 = 指示不再存在的丢失的 I 类型帧
[0300]			0 = 链接不被断开
[0301]	Next_start_code()		

[0302]	Picture_header		
[0303]	字段	bits#	描述
[0304]	Picture_start_code	32	0x00000100
[0305]	Temporal_reference	10	图像显示顺序
[0306]	Picture_coding_type	3	0000 禁用
[0307]			001 内部编码 (I)
[0308]			010 预测编码 (P)
[0309]			011 (B) 双向预测编码
[0310]			100 为 MPEG1 保留
[0311]			101 保留
[0312]			110 保留
[0313]			111 保留
[0314]	Vbv_delay	16	视频缓冲器检验机制 (临时存储器)
[0315]	Full_pel_forward_vector	1	在 MPEG 1 中使用
[0316]			MPEG2 = 0
[0317]	Forward_f_code	3	在 MPEG1 中使用
[0318]			MPEG2 = 111
[0319]	Full_pel_backward_vector	1	在 MPEG 1 中使用
[0320]			MPEG2 = 0
[0321]	Backward_f_code	3	在 MPEG 1 中使用
[0322]			MPEG2 = 111
[0323]	Extra_bit_picture	1	可忽略
[0324]	Extra_information_picture	8	可忽略
[0325]	Extra_bit_picture	1	可忽略
[0326]	Extra_start_code		
[0327]	Picture_coding_extension		
[0328]	字段	Bits#	描述
[0329]	Extension_start_code	32	总为 0x000001B5
[0330]	Extension	4	总为 1000
[0331]	_start_code_identifier		
[0332]	F_code(0)(0)	4	用于解码运动矢量;当它为 I 类型图像时,该数据充满 1111。
[0333]			
[0334]	F_code(0)(1)	4	
[0335]	F_code(1)(0)	4	解码运动矢量 (B) 信息,当它是 (P) 类型图像时,它必须被设置成
[0336]			1111,因为没有向后运动。
[0337]			
[0338]	F_code(1)(1)	4	解码运动矢量信息,当它是 P 类型图像时,它必须被设置成 1111,
[0339]			因为没有向后运动。
[0340]			

[0341]	Intra_dc_precision	2	反向量化 DC 离散余弦变换系数的精度。
[0342]			
[0343]			00 8 比特精度
[0344]			01 9 比特精度
[0345]			10 10 比特精度
[0346]			11 11 比特精度
[0347]	Picture_structure	2	指示图像被分成多场还是整帧。
[0348]			00 保留 (TDVision®格式的图像)
[0349]			01 顶场
[0350]			10 底场
[0351]			11 逐帧图像
[0352]	Top_field_first	1	0 = 先解码底场
[0353]			1 = 先解码顶场
[0354]	Frame_pred_frame_dct	1	
[0355]	Concealment_motion_vectors	1	
[0356]	Q_scale_type	1	
[0357]	Intra_vic_format	1	
[0358]	Alternate_scan	1	
[0359]	Repeat_first_field	1	0 = 显示逐行帧
[0360]			1 = 显示两个相同的逐行帧
[0361]	Chrome_420_type	1	如果色度格式为 4 : 2 : 0, 则它必须
[0362]			等于 progressive_frame, 否则它必
[0363]			须等于零。
[0364]	Progressive_frame	1	0 = 隔行的
[0365]			1 = 逐行的
[0366]	Composite_display_flag	1	警告初始编码的信息
[0367]	V_axis	1	
[0368]	Field_sequence	3	
[0369]	Sub_carrier	1	
[0370]	Burst_amplitude	7	
[0371]	Sub_carrier_phase	8	
[0372]	Next_start_code()		
[0373]	Picture_temporal_scalable_extension()		
[0374]	在具有时间可扩展性的情况下, 存在两个空间分辨率流, 底层提供视频帧的较少索引版本, 而顶层可以被用于获取相同视频的较多帧版本。低质量、低成本或免费解码器可使用时间可扩展性, 而相同付费下每秒可传输更多帧。		
[0375]	Picture_temporal_scalable_extension()		
[0376]	字段	bits#	描述
[0377]	Extension	4	总为 1010

[0378]	_start_code_identifier		
[0379]	Reference_select_code	2	用于指示参考图像将被用于解码
[0380]			intra_coded 图像
[0381]			对于 0 类型图像
[0382]			00 增强最近的图像
[0383]			01 按显示顺序的较低和最近的
[0384]			帧层
[0385]			10 按显示顺序的帧下的一层
[0386]			11 禁用
[0387]			对于 B 类型图像
[0388]			00 禁用
[0389]			01 增强模式下的最近解码的图
[0390]			像
[0391]			10 增强模式下的最近解码的图
[0392]			像
[0393]			11 按显示顺序的底层中的最近
[0394]			图像
[0395]	Forward_temporal_reference	10	时间参考
[0396]	Marker_bit	1	
[0397]	Backward_temporal_reference	10	时间参考
[0398]	Next_star_code()		
[0399]	Picture_spatial_acalable_extension()		
[0400]	在图像空间可扩展的情况下,增强层包含数据,其允许基层的更好的分辨率,以便于它能够被重建。当增强层作为运动补偿的参考被用作基层的函数时,底层必须被增强和补偿,以便于获得增强层的更大分辨率。		
[0401]	Picture_spatial_scalable_extension()		
[0402]	字段	bits#	描述
[0403]	Extension_start_code_identifier	4	总是 1001
[0404]	Lower_layer_temporal_reference	10	参考较低层的时间图
[0405]			像
[0406]	Marker_bit	1	1
[0407]	Lowerlayer_horizontal_offset	15	水平补偿(补偿)
[0408]	Marker_bit	1	1
[0409]	Lower_layer_veretical_offset	15	垂直补偿(补偿)
[0410]	Spatial_temporal_weight_code_table_index	2	预测细节
[0411]	Lowerlayer_progressive_frame	1	1 = 逐行的
[0412]			0 = 隔行的
[0413]	Lower_layer_desinterlaced_field_select	1	0 = 使用顶场
[0414]			1 = 使用底场

[0415]	Next_start_code()		
[0416]	Copyright_extension()		
[0417]	Extension	4	总为 010
[0418]	_start_code_identifier		
[0419]	Copyright_flag	1	如果它等于 1 则它使用版权
[0420]			如果它是零 (0), 不需要附加的版权
[0421]			信息
[0422]	Copyright_identifier	8	1 = 初始
[0423]			0 = 复制
[0424]	Original_or_copy	1	
[0425]	Reserved	7	
[0426]	Marker_bit	1	
[0427]	Copyright_number_1	20	授予的版权号
[0428]	Marker_bit	1	
[0429]	Copyright_number_2	22	授予的版权号
[0430]	Marker_bit	1	
[0431]	Copyright_number_3	22	授予的版权号
[0432]	Next_start_code()		
[0433]	Picture_data()		
[0434]	这是简单结构, 它本身中不具有字段。		
[0435]	Slice()		
[0436]	包含相同的垂直位置处的一个或多个宏块的信息。		
[0437]	Slice_start_code	32	
[0438]	Slice_vertical_position_extension	3	
[0439]	Priority_breakpoint	7	
[0440]	Quantizer_scale_code	5	
[0441]	Intra_slice_flag	1	
[0442]	Intra_slice	1	
[0443]	Reserved_bits	7	
[0444]	Extra_bit_slice	1	
[0445]	Extra_information_slice	8	
[0446]	Extra_bit_slice	1	
[0447]	Macroblock()		
[0448]	Macroblock_modes()		
[0449]	Motion_vectors()		
[0450]	Motion_vector()		
[0451]	Coded_block_pattern()		
[0452]	Block()		
[0453]	EXTENSION_AND_USER_DATA(2)		

[0454] 该 MPEG2 可兼容的编码处理当前被用于编码图 5 的立体照相机 (52) 拍摄的 3D 数字图像,随后经过编译器 (51) 传送,随后可获得在 PC(50) 和 DVD(53) 中显示的信号;当解码器 (54) 中的信号被编码时,其可被发送到解码器 (55) 中,用于经由电缆 (56)、卫星 (57)、高清晰电视 (59) (HDTV)、或在 3DVisor® 装置 (59) 等中显示。这样,图像可显示在以下装置上:

[0455] DVD (数字多用盘)

[0456] DTV (数字电视)

[0457] HDTV (高清晰电视)

[0458] CABLE (DVB 数字视频广播)

[0459] SATELLITE (DSS 数字卫星系统);并且它是软件和硬件处理的集成。

[0460] 关于硬件,大多数处理是由被称为 DSP (数字信号处理器) 的装置执行的。即,一个 Motorola 模型和一个 Texas Instruments (TMS320C62X) 模型。

[0461] 这些 DSP 通过由所讨论的制造商提供的 C 和汇编语言的混合语言编程。每一个 DSP 具有其自己的 API,包括位于 DSP 中被软件调用的函数列表或过程调用。从该参考信息,3D 图像被编码,其可与 MPEG2 格式兼容,并可与它们自身的编码算法兼容。当信息被编码时,DSP 负责执行预测、比较、量化和 DCT 函数应用程序处理,以便形成 MPEG2 压缩的视频流。

[0462] 以上示例说明并描述了本发明的具体实施例,对于本领域的技术人员来说显而易见的是,可以进行一些修改或变化,而不会脱离本发明的范围。所有这种修改和变化意在由所附权利要求覆盖,使得所有变化和修改落在本发明的范围内。

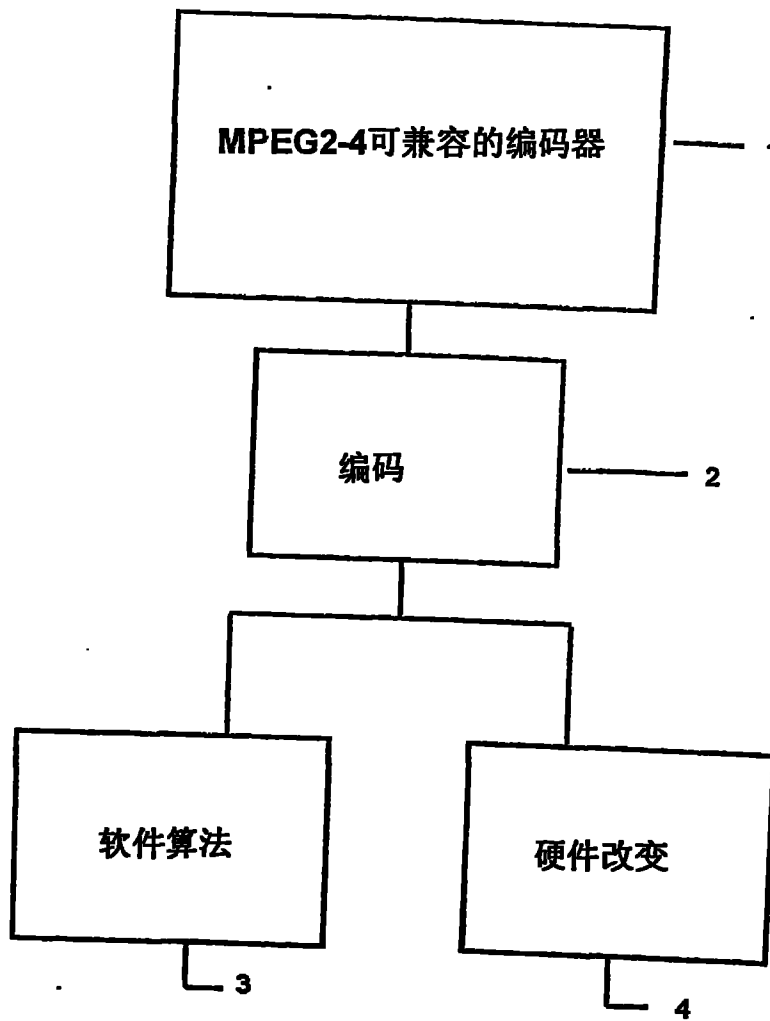


图 1

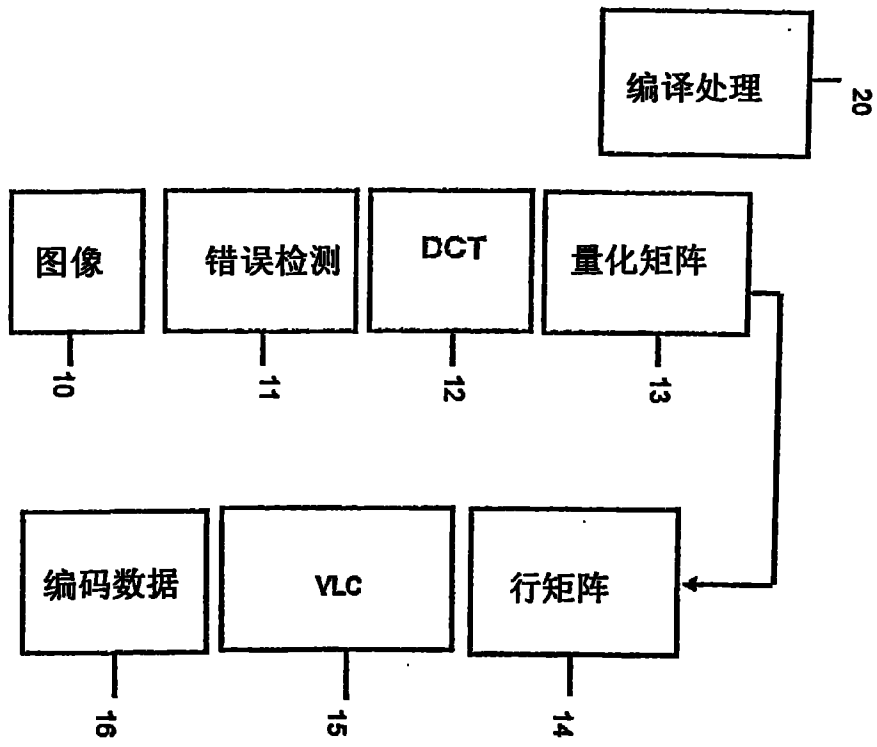


图 2

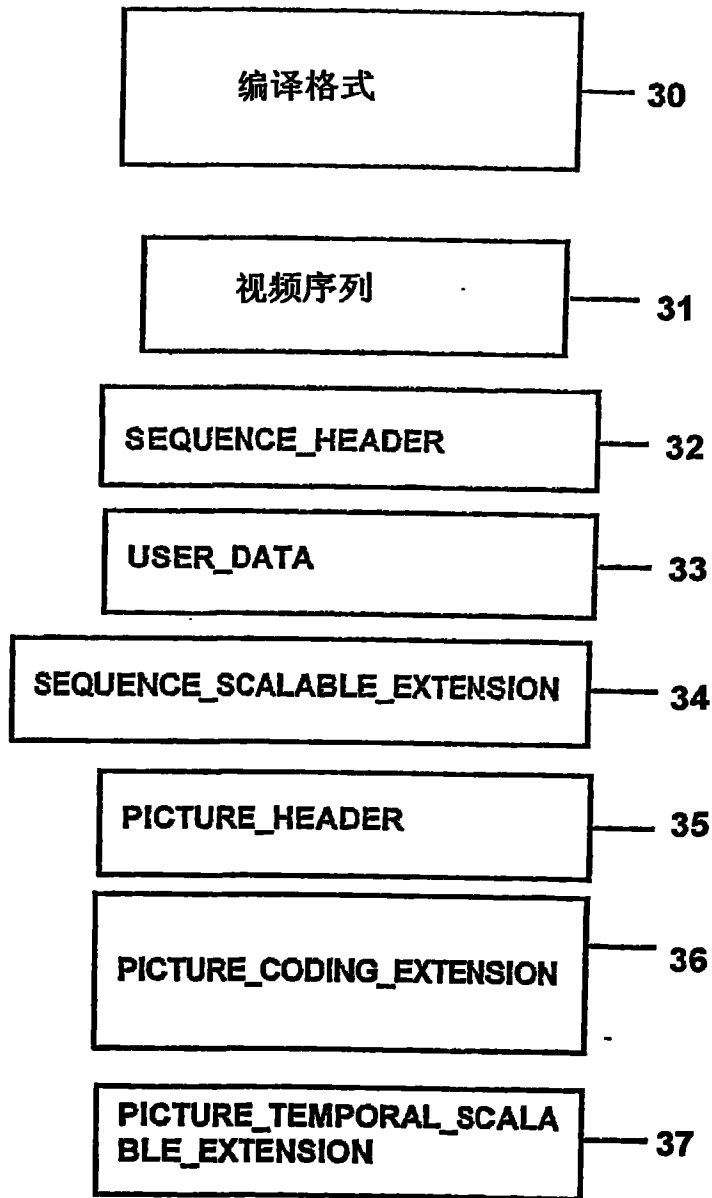


图 3

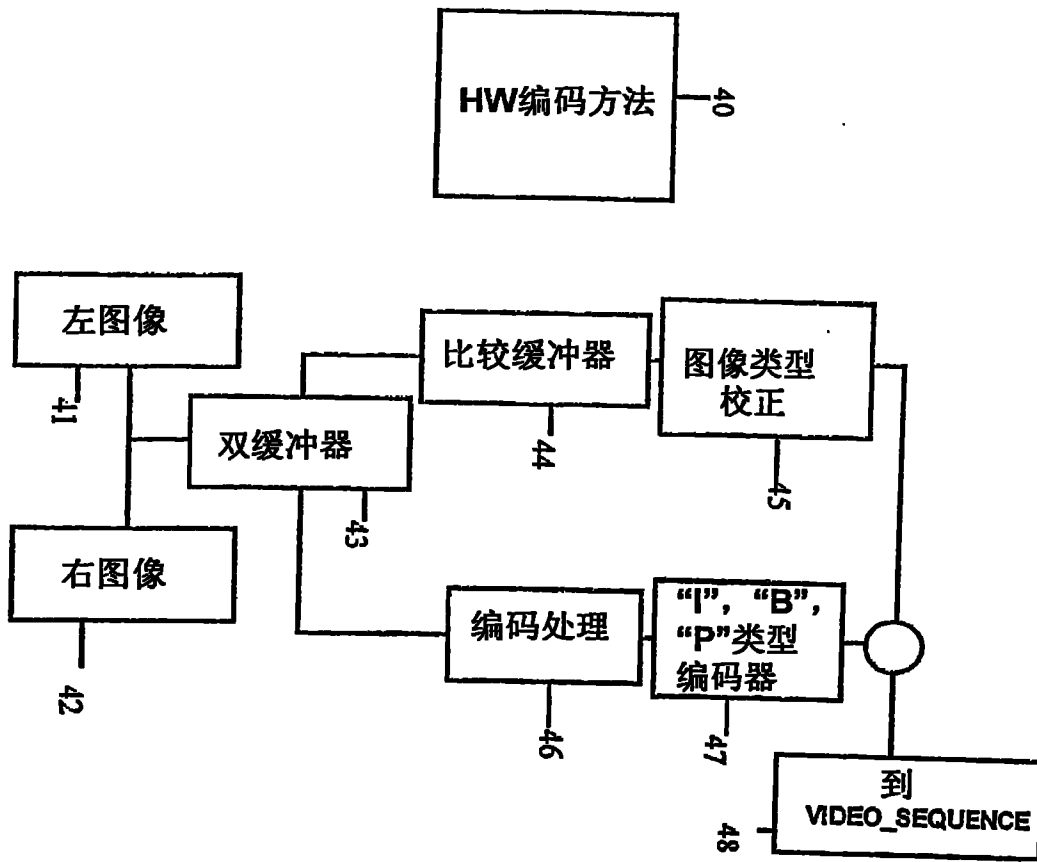


图 4

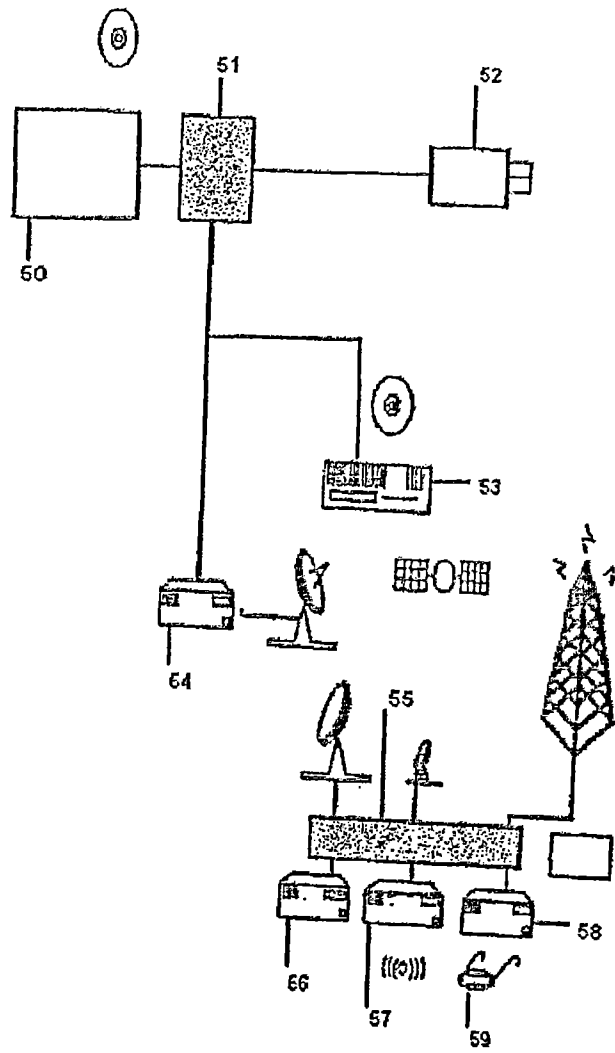


图 5