



(12) 发明专利申请

(10) 申请公布号 CN 116648917 A

(43) 申请公布日 2023. 08. 25

(21) 申请号 202180085486.1

(22) 申请日 2021.12.15

(30) 优先权数据

2020066.3 2020.12.17 GB

2105269.1 2021.04.13 GB

2105852.4 2021.04.23 GB

(85) PCT国际申请进入国家阶段日

2023.06.16

(86) PCT国际申请的申请数据

PCT/EP2021/086002 2021.12.15

(87) PCT国际申请的公布数据

W02022/129235 EN 2022.06.23

(71) 申请人 佳能株式会社

地址 日本东京都大田区下丸子3丁目30番2号

(72) 发明人 赫夫·鲁伦 弗兰克·德诺奥

弗雷德里克·梅兹

内尔·奥德拉奥果 深田昌敬

(74) 专利代理机构 北京魏启学律师事务所

11398

专利代理师 魏启学 王小香

(51) Int.Cl.

H04N 21/236 (2006.01)

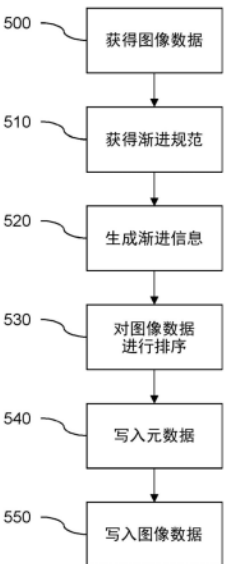
权利要求书2页 说明书51页 附图9页

(54) 发明名称

将图像数据封装在文件中以用于渐进绘制的方法和设备

(57) 摘要

本发明涉及一种用于将图像数据封装在媒体文件中的方法,所述图像数据与基于多个子图像要生成的主图像有关,其中,所述方法包括:获得所述多个子图像,各个子图像以至少一个版本被提供;生成用于描述所述主图像和所述多个子图像的描述性元数据;将所述多个子图像和所述描述性元数据封装在所述媒体文件中,其中,所述方法还包括:生成渐进信息,所述渐进信息定义用于生成所述主图像的版本的连续的渐进步阶的集合,各个渐进步阶与生成所述主图像的该版本所需的子图像版本的集合相关联;以及将所述渐进信息嵌入所述描述性元数据中。



1. 一种用于将图像数据封装在媒体文件中的方法,所述图像数据与基于多个子图像要生成的主图像有关,其中,所述方法包括:

获得所述多个子图像,各个子图像以至少一个版本被提供;

生成用于描述所述主图像和所述多个子图像的描述性元数据;

将所述多个子图像和所述描述性元数据封装在所述媒体文件中,

其中,所述方法还包括:

生成渐进信息,所述渐进信息定义用于生成所述主图像的版本的连续的渐进步阶的集合,各个渐进步阶与生成所述主图像的该版本所需的子图像版本的集合相关联;以及

将所述渐进信息嵌入所述描述性元数据中。

2. 根据权利要求1所述的方法,其中,各个子图像表示所述主图像的层的不同子集。

3. 根据权利要求1所述的方法,其中,子图像是输入图像,至少一个输入图像与该输入图像的不同版本相关联。

4. 根据权利要求2或3所述的方法,其中,所述渐进信息针对各个渐进步阶包括与该渐进步阶相关联的子图像版本数据的在所述文件中的位置。

5. 根据权利要求4所述的方法,其中,所述位置指示与所述渐进步阶相关联的、所述文件中的子图像版本数据的最后字节。

6. 根据权利要求4所述的方法,其中,所述位置包括指示与所述渐进步阶相关联的子图像版本数据的偏移和长度。

7. 根据权利要求4所述的方法,其中,所述位置包括与所述渐进步阶相关联的子图像版本数据的区间的列表。

8. 根据权利要求2或3所述的方法,其中,子图像数据被组织成一个或多个区间,各个区间由与一个子图像的版本有关的连续图像数据组成,所述渐进信息在描述区间的描述性元数据中包括指示该区间的最后字节对应于渐进步阶的信息。

9. 根据权利要求2或3所述的方法,其中,子图像版本被描述为图像项,所述渐进信息针对各个渐进步阶包括标识与该渐进步阶相关联的图像项的图像项标识符的列表。

10. 根据权利要求3所述的方法,其中,所述渐进信息针对各个渐进步阶包括在该渐进步阶中包含的输入图像版本的数量。

11. 根据权利要求10所述的方法,其中,至少一个输入图像由多个层组成,各个层与该输入图像的版本相关联,所述渐进信息还包括与该输入图像的图像项标识符相关联的层标识符。

12. 根据权利要求1所述的方法,其中,生成渐进信息包括:生成渐进精化数据结构,所述渐进精化数据结构包括用于确定所述媒体文件中的重构点的位置的数据,各个重构点指示能够使用与先前接收的子图像版本相关联的图像数据来重构所述主图像。

13. 根据权利要求12所述的方法,其中,所述渐进精化数据结构还包括多个重构点。

14. 根据权利要求1所述的方法,其中,所述渐进信息表现所述主图像的结构,使得所述主图像的质量逐渐提高。

15. 根据权利要求1所述的方法,其中,所述渐进信息与所述主图像相关联。

16. 根据权利要求1所述的方法,其中,所述主图像是实体组的一部分,并且所述渐进信息与所述实体组相关联。

17.一种用于从图像数据文件生成基于多个子图像要生成的主图像的方法,其中,所述方法包括:

从所述图像数据文件获得描述所述主图像和所述多个子图像的描述性元数据,各个子图像以至少一个版本被提供;

从所述描述性元数据获得渐进信息,所述渐进信息定义用于生成所述主图像的版本的连续的渐进步阶的集合,各个渐进步阶与生成所述主图像的该版本所需的子图像版本的集合相关联;

从所述图像数据文件获得与子图像相对应的图像数据;以及

生成与相应渐进步阶相对应的、所述主图像的至少两个版本,其中,在从所述图像数据文件获得与所述相应渐进步阶相关联的子图像版本的集合的情况下,生成所述主图像的各个版本。

18.一种用于可编程设备的计算机程序产品,所述计算机程序产品包括指令序列,所述指令序列在被加载到所述可编程设备中并且被所述可编程设备执行时实现根据权利要求1至17中任一项所述的方法。

19.一种计算机可读存储介质,其存储用于实现根据权利要求1至17中任一项所述的方法的计算机程序的指令。

20.一种计算机程序,其在执行时使得进行根据权利要求1至17中任一项所述的方法。

21.一种用于将图像数据封装在媒体文件中的装置,所述图像数据与基于多个子图像要生成的主图像有关,其中,所述装置包括处理器,所述处理器被配置为:

获得所述多个子图像,各个子图像以至少一个版本被提供;

生成用于描述所述主图像和所述多个子图像的描述性元数据;

将所述多个子图像和所述描述性元数据封装在所述媒体文件中,

其中,方法还包括:

生成渐进信息,所述渐进信息定义用于生成所述主图像的版本的连续的渐进步阶的集合,各个渐进步阶与生成所述主图像的该版本所需的子图像版本的集合相关联;以及

将所述渐进信息嵌入所述描述性元数据中。

22.一种用于从图像数据文件生成基于多个子图像要生成的主图像的装置,其中,所述装置包括处理器,所述处理器被配置为:

从所述图像数据文件获得描述所述主图像和所述多个子图像的描述性元数据,各个子图像以至少一个版本被提供;

从所述描述性元数据获得渐进信息,所述渐进信息定义用于生成所述主图像的版本的连续的渐进步阶的集合,各个渐进步阶与生成所述主图像的该版本所需的子图像版本的集合相关联;

从所述图像数据文件获取与子图像相对应的图像数据;以及

生成与相应渐进步阶相对应的、所述主图像的至少两个版本,其中,在从所述图像数据文件获得与所述相应渐进步阶相关联的子图像版本的集合的情况下,生成所述主图像的各个版本。

将图像数据封装在文件中以用于渐进绘制的方法和设备

技术领域

[0001] 本发明涉及用于将图像数据封装在文件中的方法和设备。本发明更特别地涉及允许图像的渐进绘制(progressive rendering)的封装方法。

背景技术

[0002] 可以从本地盘、网络盘或从远程服务器加载显示给用户的图像。为了加载,图像通常被封装在包括图像数据和元数据的文件内。元数据用于描述图像数据在文件中的组织、内容的类型以及用于图像数据的绘制的任何有用信息。

[0003] 通常对图像进行编码以减小存储装置上的数据的大小。可以使用许多编码标准,如JPEG、AV1或更加新的HEVC或VVC标准。

[0004] HEVC和VVC标准定义了用于对静止图像进行编码的配置文件,并且描述了用于压缩单个静止图像或连续的静止图像的特定工具。已经提出了用于这种图像数据的ISO基本媒体文件格式(ISOBMFF)的扩展,以包括在ISO/IEC 23008标准的第12部分中(名称为“HEIF”或者“High Efficiency Image File Format(高效图像文件格式)”)。

[0005] HEIF(高效图像文件格式)是由运动图像专家组(MPEG)开发的用于存储及共享图像和图像序列的标准。

[0006] MIAF(多图像应用格式)是由MPEG开发成ISO/IEC 23000标准部分22的标准。MIAF规范规定了多媒体应用格式、多图像应用格式(MIAF),其使得能够实现用于对以高效图像文件(HEIF)格式嵌入的图像进行创建、读取、解析和解码的精确互操作性点。MIAF规范完全符合HEIF格式,并且仅定义附加约束以确保更高的互操作性。。

[0007] 取决于图像的位置,从本地或远程存储系统加载图像可能需要足够长的延迟才能被用户注意到。诸如JPEG或JPEG-2000等的一些格式能够实现图像的渐进绘制,其中在渐进绘制中,图像的完全的但低质量的版本被包含在文件的开头中。利用文件的剩余部分,可能多遍对图像的该低质量版本进行精化,从而提供具有整个文件的全质量版本。渐进格式使得能够快速显示图像的全大小预览版本,而无需等待图像的完全接收。由此增加了缓慢加载的图像的显示的反应性并且改善了用户体验。

[0008] HEIF格式对于渐进绘制没有明确支持。

[0009] HEIF定义导出图像,该导出图像是通过对于存在于同一文件中的其他图像应用操作而构建的图像的表示。两种类型的导出图像特别是网格类型和叠加类型。网格是全部具有相同大小的较小图像的阵列。叠加通过在较大画布内以给定的分层顺序叠加一个或多个于一个图像来定义导出图像。

[0010] HEIF和MIAF文件格式没有提供允许网格和叠加导出图像项的渐进绘制的机制。特别地,没有描述如何从它们的单独成分渐进地构建并且作为整体而渐进地绘制网格类型或叠加类型的导出图像、或者基于多个输入图像的任何类型的导出图像。

[0011] HEIF和MIAF文件格式也没有提供允许多层图像的渐进绘制的机制。

发明内容

[0012] 已经设想本发明以应对一个或多个前述问题。本发明涉及鉴于图像的渐进绘制而将图像数据封装在文件中的方法和装置。更特别地，本发明涉及基于多个输入图像构建的图像。本发明特别适用于网格类型或叠加类型的导出图像，但不限于这些类型的图像。

[0013] 根据本发明的第一方面，提供一种用于将图像数据封装在媒体文件中的方法，所述图像数据与基于多个子图像要生成的主图像有关，其中，所述方法包括：

[0014] 获得所述多个子图像，各个子图像以至少一个版本被提供；

[0015] 生成用于描述所述主图像和所述多个子图像的描述性元数据；

[0016] 将所述多个子图像和所述描述性元数据封装在所述媒体文件中，

[0017] 其中，所述方法还包括：

[0018] 生成渐进信息，所述渐进信息定义用于生成所述主图像的版本的连续的渐进步阶的集合，各个渐进步阶与生成所述主图像的该版本所需的子图像版本的集合相关联；以及

[0019] 将所述渐进信息嵌入所述描述性元数据中。

[0020] 在实施例中，各个子图像表示所述主图像的层的不同子集。

[0021] 在实施例中，子图像是输入图像，至少一个输入图像与该输入图像的不同版本相关联。

[0022] 在实施例中，所述渐进信息针对各个渐进步阶包括与该渐进步阶相关联的子图像版本数据的在所述文件中的位置。

[0023] 在实施例中，所述位置指示与所述渐进步阶相关联的、所述文件中的子图像版本数据的最后字节。

[0024] 在实施例中，所述位置包括指示与所述渐进步阶相关联的子图像版本数据的偏移和长度。

[0025] 在实施例中，所述位置包括与所述渐进步阶相关联的子图像版本数据的区间的列表。

[0026] 在实施例中，子图像数据被组织成一个或多个区间，各个区间由与一个子图像的版本有关的连续图像数据组成，所述渐进信息在描述区间的描述性元数据中包括指示该区间的最后字节对应于渐进步阶的信息。

[0027] 在实施例中，子图像版本被描述为图像项，所述渐进信息针对各个渐进步阶包括标识与该渐进步阶相关联的图像项的图像项标识符的列表。

[0028] 在实施例中，所述渐进信息针对各个渐进步阶包括在该渐进步阶中包含的输入图像版本的数量。

[0029] 在实施例中，至少一个输入图像由多个层组成，各个层与该输入图像的版本相关联，所述渐进信息还包括与该输入图像的图像项标识符相关联的层标识符。

[0030] 在实施例中，生成渐进信息包括：生成渐进精化数据结构，所述渐进精化数据结构包括用于确定所述媒体文件中的重构点的位置的数据，各个重构点指示能够使用与先前接收的子图像版本相关联的图像数据来重构所述主图像。

[0031] 在实施例中，所述渐进精化数据结构还包括多个重构点。

[0032] 在实施例中，所述渐进信息表现所述主图像的结构，使得所述主图像的质量逐渐提高。

[0033] 在实施例中,所述渐进信息与所述主图像相关联。

[0034] 在实施例中,所述主图像是实体组的一部分,并且所述渐进信息与所述实体组相关联。

[0035] 根据本发明的另一方面,提供一种用于从图像数据文件生成基于多个子图像要生成的主图像的方法,其中,所述方法包括:

[0036] 从所述图像数据文件获得描述所述主图像和所述多个子图像的描述性元数据,各个子图像以至少一个版本被提供;

[0037] 从所述描述性元数据获得渐进信息,所述渐进信息定义用于生成所述主图像的版本的连续的渐进步阶的集合,各个渐进步阶与生成所述主图像的该版本所需的子图像版本的集合相关联;

[0038] 从所述图像数据文件获得与子图像相对应的图像数据;以及

[0039] 生成与相应渐进步阶相对应的、所述主图像的至少两个版本,其中,在从所述图像数据文件获得与所述相应渐进步阶相关联的子图像版本的集合的情况下,生成所述主图像的各个版本。

[0040] 根据本发明的另一方面,提供一种用于可编程设备的计算机程序产品,该计算机程序产品包括当被加载到该可编程设备中并且被该可编程设备执行时用于实现根据本发明的方法的指令序列。

[0041] 根据本发明的另一方面,提供一种存储用于实现根据本发明的方法的计算机程序的指令的计算机可读存储介质。

[0042] 根据本发明的另一方面,提供一种在执行时使得进行本发明的方法的计算机程序。

[0043] 根据本发明的另一方面,提供一种用于将图像数据封装在媒体文件中的装置,所述图像数据与基于多个子图像要生成的主图像有关,其中,所述装置包括处理器,所述处理器被配置为:

[0044] 获得所述多个子图像,各个子图像以至少一个版本被提供;

[0045] 生成用于描述所述主图像和所述多个子图像的描述性元数据;

[0046] 将所述多个子图像和所述描述性元数据封装在所述媒体文件中,

[0047] 其中,方法还包括:

[0048] 生成渐进信息,所述渐进信息定义用于生成所述主图像的版本的连续的渐进步阶的集合,各个渐进步阶与生成所述主图像的该版本所需的子图像版本的集合相关联;以及

[0049] 将所述渐进信息嵌入所述描述性元数据中。

[0050] 根据本发明的另一方面,提供一种用于从图像数据文件生成基于多个子图像要生成的主图像的装置,其中,所述装置包括处理器,所述处理器被配置为:

[0051] 从所述图像数据文件获得描述所述主图像和所述多个子图像的描述性元数据,各个子图像以至少一个版本被提供;

[0052] 从所述描述性元数据获得渐进信息,所述渐进信息定义用于生成所述主图像的版本的连续的渐进步阶的集合,各个渐进步阶与生成所述主图像的该版本所需的子图像版本的集合相关联;

[0053] 从所述图像数据文件获取与子图像相对应的图像数据;以及

[0054] 生成与相应渐进步阶相对应的、所述主图像的至少两个版本,其中,在从所述图像

数据文件获得与所述相应渐进步阶相关联的子图像版本的集合的情况下,生成所述主图像的各个版本。

[0055] 根据本发明的方法的至少部分可以是计算机实现的。因此,本发明可以采取完全硬件实施例、完全软件实施例(包括固件、常驻软件、微代码等)或组合软件和硬件方面的实施例的形式,这些实施例在本文中通常都可以被称为“电路”、“模块”或“系统”。此外,本发明可以采取体现在任何有形表达介质中的计算机程序产品的形式,该有形表达介质具有体现在介质中的计算机可用程序代码。

[0056] 由于本发明可以用软件实现,因此本发明可以体现为用于在任何合适的载体介质上提供给可编程设备的计算机可读代码。有形的非暂态载体介质可以包括存储介质,诸如软盘、CD-ROM、硬盘驱动器、磁带装置或固态存储器装置等。瞬态载体介质可以包括诸如电信号、电子信号、光信号、声信号、磁信号或电磁信号(例如微波或RF信号)等的信号。

附图说明

[0057] 现在将仅通过示例的方式并参考以下附图来描述本发明的实施例,在附图中:

[0058] 图1示出HEIF文件的示例,该HEIF文件包含媒体数据,诸如一个或多个静止图像以及可能的一个或多个视频或者一个或多个图像序列;

[0059] 图2示出用于实现网格项的渐进绘制的文件结构的第一示例;

[0060] 图3示出用于实现类似网格项的渐进绘制的另一文件结构;

[0061] 图4a至4c示出图像数据在图像文件中的不同排序;

[0062] 图4d示出与四个不同的渐进步阶相关地使用与图4b相同的排序的另一示例;

[0063] 图4e表示与四个不同的渐进步阶相关地使用与图4c不同的排序的另一示例;

[0064] 图4f示出在与图4d相对应的第二渐进步阶之后的网格图像项的绘制;

[0065] 图4g示出在与图4e相对应的第二渐进步阶之后的网格图像项的绘制;

[0066] 图5示出根据本发明实施例的生成HEIF文件的主要步骤;

[0067] 图6示出根据本发明实施例生成的HEIF文件的渐进绘制的主要步骤;

[0068] 图7示出根据本发明的其他实施例生成的HEIF文件的渐进绘制的主要步骤;

[0069] 图8是用于实现本发明的一个或多个实施例的计算装置的示意性框图;

[0070] 图9示出将缩略图和预览与图像项相关联以提供针对该图像项的渐进绘制的示例;

[0071] 图10示出使用项属性来描述基于图像项的编码层的渐进绘制步阶的示例;

[0072] 图11示出使用项属性来描述基于导出图像项的输入图像项的渐进绘制步阶的示例;以及

[0073] 图12示出基于导出图像项的输入图像项的编码层的该导出图像项的渐进绘制步阶的示例。

具体实施方式

[0074] HEVC或VVC标准定义了用于对静止图像进行编码的配置文件,并且描述了用于压缩单个静止图像或连续的静止图像的特定工具。已经提出了用于这种图像数据的ISO基本媒体文件格式(ISOBMFF)的扩展,以包括在ISO/IEC 23008标准的第12部分中(名称为

“HEIF”或者“High Efficiency Image File Format(高效图像文件格式)”)”。

[0075] HEIF和MIAF标准涵盖对应于不同用例的两种形式的存储:

[0076] • 图像序列的存储,图像序列可以被指示为被显示为定时序列或通过其他手段被显示,并且在图像序列中图像可以依赖于其他图像,以及

[0077] • 单个编码图像或独立编码图像集合的存储,可能具有所导出的图像。

[0078] 在第一种情况下,封装接近于ISO基本媒体文件格式中的视频轨的封装(参见文档《Information technology—Coding of audio-visual objects—Part12:ISO base media file format》,w18855,ISO/IEC 14496-12,第六版,2019年10月),并且使用类似的工具和概念,诸如“trak”框以及用于描述样本组的样本分组等。“trak”框是包含用于描述轨(即,相关样本的定时序列)的子框的文件格式框。

[0079] 框(也称为容器)是被提供以描述文件中的数据的数据的层级元数据结构。框是由唯一类型标识符(通常是四字符代码,也称为FourCC或4CC)和长度定义的面向对象的构建块。文件中的所有数据(媒体数据和描述媒体数据的元数据)包含在框中。文件内没有其他数据。文件级框是不包含在其他框中的框。

[0080] 在第二种情况下,使用ISOBMFF框、“meta”框的集合。这些框及其层次结构提供比“track-related(轨相关)”框(“trak”框层次结构)更少的描述工具,并且涉及“信息项”或“项”来代替相关样本。应当注意,词语“框”和词语“容器”可以以相同的含义使用,以指代包含描述文件中的图像数据的组织或/和属性的元数据的数据结构。

[0081] 图1示出HEIF文件101的示例,该HEIF文件101包含媒体数据,诸如一个或多个静止图像以及可能的一个或多个视频或者一个或多个图像序列。该文件包含“ftyp”框(FileTypeBox)111,其包含文件类型的标识符(通常是四字符代码集合)。该文件包含称为“meta”的框(MetaBox)102,其用于包含包括描述一个或多个静止图像的元数据结构的通用非定时元数据。该“meta”框102包含描述若干单个图像的“iinf”框(ItemInfoBox)121。各个单个图像由也表示为项1211和1212的元数据结构ItemInfoEntry描述。各个项具有唯一的16位或32位标识符item_ID。对应于这些项的媒体数据被存储在用于媒体数据的容器(例如“mdat”框104)中。媒体数据也可以被存储在“idat”或“mdat”框中、“imda”框或其他文件中。“iloc”框(ItemLocationBox)122为各个项提供“mdat”、“idat”或“imda”框104中的其相关联的媒体数据的偏移和长度。项的媒体数据可以被分段成区间(extent)。在这种情况下,“iloc”框122提供项的区间的数量和针对各区间的在“mdat”、“idat”或“imda”框104中的偏移和长度。“iref”框(ItemReferenceBox)123也可以被定义为描述一个项经由类型化参考而与其他项的关联。

[0082] 可选地,为了描述图像序列或视频的存储,HEIF文件101可以包含称为“moov”的框(MovieBox)103,其描述一个或多个图像序列或视频轨131和132。通常,轨131可以是被设计为描述时间信息不一定有意义的图像集合的图像序列(“pict”)轨,并且轨132是被设计为描述视频内容的视频(“vide”)轨。这两个轨描述一系列图像样本,图像样本是同时拍摄的像素集合,例如视频序列的帧。两个轨之间的主要区别在于,在“pict”轨中,定时信息不一定是有意义的,而对于“vide”轨,定时信息旨在约束样本的显示的定时。对应于这些样本的数据存储在用于媒体数据的容器(“mdat”框)104中。

[0083] “mdat”容器104存储与由数据部分141和142表示的项相对应的未定时编码图像以

及可选地与由数据部分143表示的样本相对应的定时编码图像(当视频轨也存在于HEIF文件中时)。

[0084] HEIF文件101提供不同的替代方案来存储多个图像。例如,可以将多个图像存储为项或存储为可以是“pict”轨或“vide”轨的样本的轨。实际选择通常由生成文件的应用或装置根据图像的类型和文件的预期用途来做出。

[0085] 当HEIF项具有对一个或多于一个其他图像项的“dimg”项参考时,HEIF项是导出图像项,其中该一个或多于一个其他图像项被输入到导出。

[0086] 具有item_type值“iovl”的项通过在较大画布内以给定的分层顺序叠加一个或多于一个输入图像来定义导出图像项。输入图像以它们在针对该导出图像项的“dimg”项参考中被分层的顺序而列出。叠加图像项的数据指定各输入图像在较大画布内的位置。

[0087] 具有item_type值“grid”的项定义导出图像项,该导出图像项的重构图像由较大画布内的给定网格顺序的一个或多于一个输入图像形成。输入图像以在针对该导出图像项的“dimg”项参考中的网格顺序而列出。网格图像项的数据指定网格中的行和列的数量以及较大画布的大小。

[0088] ISO基本媒体文件格式指定根据保持要分组的样本或项的容器对样本或项进行分组的若干替代方案。这些替代方案可以被认为是分组数据结构或分组机制(即,提供描述分组准则和/或分组属性和/或分组实体的元数据的框或数据结构)。

[0089] 由EntityToGroupBox表示的第一分组机制应用于项或轨的分组。在该机制中,词语“实体”用于指代项或轨或其他EntityToGroupBox。该机制指定实体的分组。EntityToGroupBox根据以下句法定义:

```
aligned(8) class EntityToGroupBox(grouping_type, version, flags)
extends FullBox(grouping_type, version, flags) {
    unsigned int(32) group_id;
    unsigned int(32) num_entities_in_group;
[0090]    for(i=0; i<num_entities_in_group; i++)
        unsigned int(32) entity_id;
    // the remaining data may be specified for a particular grouping_type
}
```

[0091] grouping_type用于指定组的类型。在HEIF中指定grouping_type的若干值。group_id为实体的组提供标识符。entity_id表示组成该组的实体的标识符,即,轨的track_ID、项的item_ID或实体组的其他group_id。在图1中,从EntityToGroup框1241和1242继承的实体的组被包括在由GroupsListBox的四字符代码“gpr1”标识的容器124中。

[0092] 实体分组在于关联分组类型,该分组类型对项、轨或其他实体组的集合的分组的原因进行标识。在本文档中,将分组信息称为EntityToGroup框之一中的信息,该EntityToGroup框传送用于对图像集合进行分组的信息。

[0093] grouping_type值“altr”指示映射到该分组的项和轨是彼此的替代,并且仅应绘制项和轨其中之一。播放器应当选择分组中它可以处理的并且适合应用需要的第一实体。

[0094] ISOBMFF提供描述属性并将属性与项相关联的机制。这些属性称为项属性。ItemPropertiesBox“iprp”125使得能够将任何项与项属性的有序集合相关联。ItemPropertiesBox由两部分组成:包含项属性1253的隐式索引列表的项属性容器框

“ipco”1251,以及包含一个或多个条目的项属性关联框“ipma”1252。项属性关联框中的各个条目将项与其项属性相关联。HEIF标准扩展了该机制,以使得能够将项属性与项和/或实体组相关联。注意,在描述中,为了一般性,我们通常使用项属性来指定项的属性或实体组的属性这两者。与实体组相关联的项属性应用于整个实体组,而不是单独地应用于组内的各个实体。

[0095] 相关联的句法如下:

```

aligned(8) class ItemProperty(property_type)
    extends Box(property_type)
{
}

aligned(8) class ItemFullProperty(property_type, version, flags)
    extends FullBox(property_type, version, flags)
{
}

aligned(8) class ItemPropertyContainerBox
    extends Box('ipco')
{
    properties ItemProperty() []; // boxes derived from
    // ItemProperty or ItemFullProperty, to fill box
}

aligned(8) class ItemPropertyAssociation
    extends FullBox('ipma', version, flags)
{
[0096] unsigned int(32) entry_count;
    for(i = 0; i < entry_count; i++) {
        if (version < 1)
            unsigned int(16) item_ID;
        else
            unsigned int(32) item_ID;
        unsigned int(8) association_count;
        for (i=0; i<association_count; i++) {
            bit(1) essential;
            if (flags & 1)
                unsigned int(15) property_index;
            else
                unsigned int(7) property_index;
        }
    }
}

aligned(8) class ItemPropertiesBox
    extends Box('iprp') {
    ItemPropertyContainerBox property_container;
    ItemPropertyAssociation association[];
}

```

[0097] ItemProperty和ItemFullProperty框被设计用于项属性的描述。ItemFullProperty允许定义框的句法的若干版本,并且可以包含一个或多于一个参数,参

数的存在由版本或标志参数来调节。

[0098] ItemPropertyContainerBox被设计用于将项属性集合描述为ItemProperty框或ItemFullProperty框的阵列。

[0099] ItemPropertyAssociation框被设计为描述项和/或实体组及其项属性之间的关联。提供项标识符和/或实体组标识符的列表的描述,各个标识符(item_ID)与参考ItemPropertyContainerBox中的项属性的项属性索引的列表相关联。

[0100] 本发明的目的是提供一种对由若干图像成分构建的图像项的HEIF中的渐进绘制的粒度的精细控制。这样的图像项可以是网格导出图像项(也称为网格项)、叠加导出图像项(也称为叠加项)或基于多个输入图像的任何其他类型的导出图像。这些是从若干输入图像项构建的。这样的输入图像项还可以是使用HEVC区块或层或使用VVC子图片或层进行编码的编码图像项。为了清楚起见,本说明书的其余部分是基于网格项,但也可以应用于从若干图像成分(也表示为输入图像)构建的其他图像项。

[0101] 在以下描述中,渐进绘制的目标图像被称为主图像。用于构建该主图像的成分或输入图像被称为子图像。渐进步阶(progressive step)是指该主图像的重构版本。初始渐进步阶对应于空图像,而最终渐进步阶对应于主图像。中间渐进步阶可能具有比主图像低的分辨率和/或低的质量。注意,在以下描述中,为了简单起见,有时忽略初始渐进步阶。

[0102] 可能地,渐进绘制功能可以用于除显示图像之外的其他目的。例如,用于检测图像上的人的处理可以使用对图像的低质量版本的快速第一检查来快速拒绝不包含人的图像。

[0103] 主图像的渐进绘制通过使用子图像的不同编码版本来实现。主图像的重构版本可以在于对所有输入图像进行解码并且根据主图像项的类型(例如,网格、叠加、子图片组成……)组装这些解码的子图像。子图像版本可以是子图像自身、子图像的缩略图、子图像的较低质量和/或较低分辨率的编码版本。不同的编码版本也可以被包含在多层图像的一个或多于一个层中。当以若干质量或分辨率级别对子图像进行编码时,主图像的渐进绘制可以使用这些质量或分辨率级别的子集。渐进步阶的内容是指在主图像的与该渐进步阶相对应的版本的 recon 期间使用的子图像版本的集合。渐进步阶的精细化是指作为该渐进步阶的内容的一部分而不作为先前渐进步阶的内容的一部分的子图像版本的子集。

[0104] 各渐进步阶可以与重构点相关联。重构点是文件中的与文件中的点相对应的位置,在该点处,已经运行通过了生成主图像的与渐进步阶相对应的版本所需的输入图像的所有版本。这意味着,当读取文件时,重构点是该文件中的已经读取了给定渐进步阶的所有所需图像数据的点。

[0105] 当子图像是多层图像(即,以若干质量或分辨率层进行编码)时,子图像可以由媒体数据包含所有编码层的单个项来表示。可能地,与不同层相对应的媒体数据可以被分段并在项位置框“iloc”中使用区间进行描述。例如,可以使用两个区间来定位编码为具有两个层的多层图像的子图像的媒体数据,各区间指示当通过一个图像项描述该子图像时与层相对应的数据的位置。

[0106] 多层图像也可以由与不同层或不同层的子集相对应的若干图像项来表示。这些图像项可以共享媒体数据的共同部分。例如,两个不同的图像项可以与被编码为具有两个层的多层图像的子图像相对应。第一图像项将对应于第一层,并且其媒体数据将使用指示与第一层相对应的数据的位置的区间来定位。第二图像项将与这两个层的组合相对应,并且

其媒体数据将使用分别指示与两个层相对应的数据的位置的两个区间来定位。层的描述在图像属性中也是可用的,以指示在图像项中包含哪些层。由此,表示编码为多层图像的子图像的各图像项可以被认为是子图像版本,其类似于非多层图像项的不同编码版本。

[0107] 当主图像是用区块编码的HEVC图像或用子图片编码的VVC图像时,子图像可以是与区块(例如,“hvt1”项)或子图片(例如,“vvs1”或“vvc1”项)相对应的图像项。主图像可以是HEVC项(这里也称为HEVC基本项),其从使用从各区块子图像项到基本项的类型“tbas”的项参考与主图像相关联的区块子图像来重构,该区块子图像的位置由“rloc”属性提供。主图像可以是子图片子图像重构的VVC基本项,该子图片子图像使用从基本项到该子图片子图像项的类型“subp”的项参考与主图像相关联。子图像版本可以是以给定质量或分辨率级别编码的区块或子图片。还可以将区块或子图片编码为多层图像,并且可以通过对其质量或分辨率层的子集进行解码来获得这种子图像的版本。

[0108] 图2示出用于实现网格项的渐进绘制的文件结构的第一示例(该示例应用于叠加或者HEVC基本项或者VVC基本项:叠加也通过“dimg”项参考类型参考子图像,而HEVC基本项经由“tbas”项参考类型由区块子图像参考,并且VVC基本项通过“subp”项参考类型来参考子图像)。主图像项或主要图像项是网格项202(“高质量网格”),该网格项是使用九个子图像“h1”(标记为220)至“h9”(标记为228)来构建的。这些子图像以高质量进行编码。该主图像被包括在类型“altr”200的实体组中,该实体组还包含网格项201(“低质量网格”),该网格项是使用九个子图像“l1”(标记为210)至“l9”(标记为218)来构建的。这些子图像以低质量进行编码。主图像202的渐进绘制可以通过以下方式实现:一旦所有低质量的子图像210至218可用,首先绘制网格项201,然后一旦所有高质量的子图像220至228可用,则绘制网格项202。然而,该文件结构仅提供两个精化步阶。特别地,如果主图像202大且具有良好质量,则与子图像220至228相对应的数据量可能大且需要一些时间来加载。此外,不能保证文件以实现渐进精化的方式组织子图像的低质量版本和高质量版本的数据。

[0109] 图3示出用于实现类似网格项的渐进绘制的另一文件结构。主图像项或主要图像项是由以高质量编码的九个子图像“h1”(标记为330)至“h9”(标记为338)组成的网格项300。各子图像通过“altr”实体组与低质量编码替代物相关联:九个子图像“h1”至“h9”分别与九个子图像“l1”(标记为320)至“l9”(标记为328)相关联。例如,将“h1”与“l1”相关联的“altr”实体组被标记为310。如此,取决于哪个版本可用于各子图像,可以使用各子图像的低质量版本或高质量版本来重构网格项。该文件结构可以提供多达十个不同的渐进步阶:第一步阶由所有低质量子图像组成,第二步阶用高质量子图像替换这些低质量子图像之一,以此类推,直到仅使用高质量子图像的第十步阶为止。

[0110] 然而,实际渐进步阶取决于HEIF文件内的子图像数据的排序。图4a表示几乎不可能进行渐进绘制的第一排序:在HEIF文件中交织低质量和高质量子图像。此外,对于多层子图像,基本层和增强层可能不被组织以首先具有所有基本层、随后是所有增强层。因此,直到接收到倒数第二子图像“l9”才有可能完全重构网格项。在此之后,当接收到最后子图像“h9”时,可以立即重构主图像。

[0111] 图4b表示更适于渐进绘制的第二排序,在该排序中,在HEIF文件中首先出现所有低质量子图像,随后是所有高质量子图像。利用该排序,从仅使用低质量子图像到仅使用高质量子图像的十个渐进步阶是可能的。在加载各高质量图像之后,新的渐进步阶是可能的。

可以将相同的数据排序用于多层图像以首先提供基本层且接着提供增强层。在这种情况下,主图像所使用的子图像对其他子图像项具有依赖性,依赖子图像项的数据应当放在子图像项的数据之前。

[0112] 图4c表示也适于具有十个可能的渐进步阶的渐进绘制的第三排序,尽管对于高质量子图像使用不同的排序。图4b的排序提供逐行的渐进绘制,而图4c的排序提供逐列的渐进绘制。

[0113] 虽然关于图3所描述的文件结构能够实现比关于图2所描述的结构更多的渐进步阶,但仍具有若干缺点。首先,网格项的典型使用可能使用大得多的数量的输入图像项,从而导致许多可能的渐进步阶。更大数量的渐进步阶虽然允许对图像进行更平滑的精细化,但也需要更多的处理能力,这是因为网格图像必须在各步阶被重构。其次,不能保证HEIF文件中的子图像版本的排序有效地实现网格图像的渐进绘制。此外,无论是逐行的还是逐列的,或者使用其他精细化模式,都不存在关于如何组织该渐进绘制的指示。任意精细化可能导致在观看图像时不良的用户体验。

[0114] 本发明提出通过添加与关于渐进绘制的文件的组织有关的信息来扩展HEIF。该信息(下文中称为渐进信息)可以采取在下文描述的渐进实施例中详述的若干形式。该渐进信息可以包括渐进绘制策略或渐进策略。渐进绘制策略的示例是来自图4b的上下排序或来自图4c的左右排序。该渐进信息还可以包括由文件提出的渐进步阶的描述。

[0115] 图4d表示与四个不同的渐进步阶相关地使用与图4b中相同的排序的另一示例:仅包含低质量子图像的初始渐进步阶,随后是三个渐进步阶,各渐进步阶添加一行高质量子图像。

[0116] 图4f示出在已经接收到与网格的第一行相对应的三个高质量子图像时在第二渐进步阶之后的网格图像项的绘制。在该示例中,第二渐进步阶的内容是三个高质量子图像h1至h3和六个低质量子图像14至19。该渐进步阶的精细化是三个高质量子图像h1至h3。

[0117] 图4e表示与四个不同的渐进步阶相关地使用与图4c中相同的排序的另一示例:仅包含低质量子图像的初始渐进步阶,随后是三个渐进步阶,各渐进步阶添加一列高质量子图像。

[0118] 图4g示出在已经接收到与网格的第一列相对应的三个高质量子图像时在第二渐进步阶之后的网格图像项的绘制。在该示例中,第二渐进步阶的内容是三个高质量子图像h1、h4和h7以及六个低质量子图像12、13、15、16、18和19。该渐进步阶的精细化是三个高质量子图像h1、h4和h7。

[0119] 图5示出根据本发明实施例的生成HEIF文件的主要步骤。

[0120] 在第一步骤500中,获得要包含在HEIF文件中的图像数据。在图3的示例中,该图像数据由九个低质量子图像和九个高质量子图像组成。另外,获得关于HEIF文件的结构的信息。这包括例如主图像的类型、主图像与子图像的关系以及子图像之间的关系。在图3的示例中,该信息是主图像是由九个子图像h1至h9组成的网格项,并且这些子图像h1至h9各自具有范围从11至19的关联低质量子图像。

[0121] 在第二步骤510中,获得用于组织用于主图像的渐进绘制的文件内容的渐进规范。这些渐进规范取决于所选择的渐进实施例。这些渐进规范可以直接从用户输入或从配置文件获得,或者可以通过一些处理来确定。例如,软件组件可以确定由主图像表示的内容的类

型,并且选择最适当的渐进绘制策略。如果主图像对应于风景图片,则可以选择上下渐进策略,而如果主图像对应于面部在图像中心的人,则可以选择中心到边界渐进策略。作为另一示例,当存在感兴趣的对象或区域时,渐进绘制策略可以通过首先对感兴趣的对象或区域周围的区进行精化然后朝着图像边界渐进而开始。

[0122] 在下一步骤520中,根据在步骤510获得的渐进规范来生成要包括在HEIF文件中的渐进信息。该生成取决于所选择的渐进实施例。下面详细描述这些实施例。

[0123] 在下一步骤530中,根据渐进规范对图像数据进行排序。例如,如果渐进规范指示如图4b的示例中的上下渐进绘制策略,则将如图4b所示对图像数据进行排序。

[0124] 对于该排序步骤,图像数据被组织成渐进数据块。这些渐进数据块根据渐进规范被排序。

[0125] 当主图像是网格项时,存在包含针对网格项自身的数据的渐进数据块。对于构成网格的各子图像的各版本,也存在一个渐进数据块。在图4b的示例中,存在十九个渐进数据块:网格、九个低质量图像数据和九个高质量图像数据。

[0126] 当主图像是叠加项时,存在包含针对叠加项自身的数据的渐进数据块以及针对构成叠加的各子图像的各版本的一个渐进数据块。

[0127] 这可以基于输入子图像针对任何种类的主图像类似地变换。

[0128] 当主图像是用区块编码的HEVC图像或用子图片编码的VVC图像时,可以存在针对图像自身的一个或多于一个渐进数据块,并且可以存在针对各区块或子图片的各版本的一个渐进数据块。当子图像被编码为具有若干质量层(例如,基本层和一个或若干增强层)的多层图像时,各层对应于渐进数据块。

[0129] 可能地,渐进数据块可以进一步拆分成若干子块。

[0130] 优选地,首先放置包含与主图像结构有关的数据的数据块。例如,针对网格项(或针对叠加)自身的数据优选地首先被放置。

[0131] 该排序还可以覆盖不与主图像直接相关的数据。例如,可以存在与主图像的缩略图相对应的渐进数据块。该渐进数据块优选地被排序在针对主图像的任何渐进数据块之前。

[0132] 可以存在对应于与主图像相关联的元数据的渐进数据块。优选地,这些数据块被排序在针对主图像的所有渐进数据块之后。可能地,例如当在加载主图像之前处理或绘制一些元数据信息是有用的时,可以在两个渐进步阶之间插入这些数据块。一些渐进规范可以指示将这些渐进数据块放置在何处。

[0133] 可以存在对应于与主图像相关联的区域项的渐进数据块。优选地,这些数据块被排序在针对主图像的所有渐进数据块之后。可能地,例如当在加载主图像之前处理或绘制一些区域相关信息是有用的时,可以在两个渐进步阶之间插入这些数据块。一些渐进规范可以指示将这些渐进数据块放置在何处。

[0134] 在下一步骤540中,写入HEIF文件的元数据。这些元数据包括在步骤520处生成的渐进信息。

[0135] 在最后步骤550中,所有渐进数据块以在步骤530处确定的顺序被写入文件的介质部分,例如,“mdat”框。

[0136] 可能地,可以切换步骤520和530的排序。如果渐进信息的生成取决于渐进数据块

的排序,则这可能是有用的。

[0137] 图6示出根据本发明的示例生成的HEIF文件的渐进绘制的主要步骤。这些步骤旨在加载HEIF文件时被执行。可以例如使用HTTP协议从本地存储、从网络存储加载HEIF文件、或者从网络服务器接收HEIF文件。

[0138] 在第一步骤600中,加载HEIF文件的开始。HEIF文件的该开始包括构成HEIF文件的“meta”框的所有元数据。

[0139] 在步骤610中,“meta”框被解析以获得HEIF文件的结构,包括例如项的数量、这些项的类型、它们的关系、与这些项相关联的属性等。该“meta”框可以包含提供定义叠加项或网格项结构的数据的“idat”框。该“meta”框还可以包含缩略图。

[0140] 在下一步骤620中,从步骤610的解析结果中提取渐进信息。渐进信息的精化取决于所选择的渐进实施例。作为该提取的一部分,获得包含在HEIF文件中的渐进步阶。该获得取决于所选择的渐进实施例。

[0141] 在下一步骤630中,加载存在于文件中的数据的一部分。

[0142] 在步骤640处,检查是否已加载与新的渐进步阶相对应的数据。该检查取决于所选择的渐进实施例。

[0143] 如果检查的结果是肯定的,则下一步骤是步骤650,在步骤650实现与该渐进步阶相对应的主图像的版本的新绘制。该绘制可以包括根据与所达到的渐进步阶相对应的子图像版本计算重构图像。还可以包括对重构图像的版本计算的一些后处理,诸如重新缩放该图像以用于拟合显示区,或应用由用户选择的滤波器。还可以包括将一些变换属性应用于重构图像的版本,诸如旋转、缩放、镜像或裁剪。

[0144] 注意,在一些实施例中,基于渐进信息和所达到的渐进步阶来实现进一步检查,以确定是否应当绘制重构图像的版本。例如,如果仅显示主图像的版本的一部分,则该进一步检查可以验证所达到的渐进步阶是否对重构图像的版本的显示部分提供精化。

[0145] 如果步骤640处的检查的结果是否定的以及在步骤650之后,则下一步骤是步骤660。该步骤检查文件是否已被完全接收。

[0146] 如果检查的结果是否定的,则下一步骤是等待接收文件的附加部分的步骤630。

[0147] 如果检查的结果是肯定的,则下一步骤是算法结束的步骤670。

[0148] 注意,在步骤630处,可以接收与若干精化步阶相对应的数据。在这种情况下,步骤640处的检查将是肯定的,并且在步骤650处,使用与所达到的最后渐进步阶相对应的子图像来绘制重构图像。

[0149] 注意,所有这些步骤都旨在按顺次顺序接收HEIF文件。然而,这些步骤可以容易地适于以任意顺序接收HEIF文件作为数据块。这种适配主要涉及步骤640和660。在这些步骤中,检查必须考虑到HEIF文件的各部分可以以任意顺序被接收,从而明确地验证为与渐进步阶或整个文件相对应的所有数据已经被接收并且不利用HEIF文件的顺次接收。

[0150] 现在描述将渐进信息编码在HEIF文件中的不同实施例。

[0151] 精化模式(refinement pattern)实施例

[0152] 在第一实施例(下文中称为精化模式实施例)中,包含在HEIF文件中的渐进信息基于定义了HEIF文件中使用的渐进精化策略的渐进精化模式。渐进精化模式可以是例如图4d和4f的上下渐进精化或图4e和4g的左右渐进精化。渐进精化模式指示输入图像的扫描顺

序。当与输入图像的扫描顺序中的步阶相对应的输入图像的版本集合已经被加载时定义渐进步阶,从而允许绘制主图像的版本。换言之,渐进精化模式指示针对主图像的不同空间区域的精化的排序。例如,在网格项的情况下,渐进精化模式指示针对构成网格的不同子图像的精化的排序。

[0153] 该第一实施例有利地为HEIF文件内的渐进信息提供紧凑信号通知。

[0154] 在HEIF文件内使用的渐进精化模式可以在应用级被展示,从而允许应用基于渐进精化模式在若干HEIF文件之中选择HEIF文件。例如,对于相同的主图像,第一HEIF文件可能使用上下渐进精化,而第二HEIF文件可能使用左右渐进精化。应用可以在这两个文件之间选择最适合其需要的文件。

[0155] 预定义精化变型

[0156] 在该第一实施例的第一变型(下文中被称为预定义精化模式变型)中,预定义渐进精化模式的列表。渐进信息包括预定义渐进精化模式之一的指示。

[0157] 该变型有利地允许简单用信号通知在HEIF文件中使用的渐进精化策略。

[0158] 渐进精化模式可以被表示为与其所应用的图像项相关联的项属性。该渐进精化模式项属性可以由以下结构来描述:

```
[0159] aligned(8) class ProgressivePatternProperty
      extends ItemFullProperty('ppat', version = 0, flags = 0) {
          unsigned int(8)    pattern_type;
          unsigned int(1)    reverse_flag;
          unsigned int(1)    bidirectional_flag;
          unsigned int(1)    parameter_flag;
          unsigned int(5)    reserved;
[0160] unsigned int(8)    scale;
          if (parameter_flag == 1) {
              unsigned int(32)    parameter;
          }
      }
```

[0161] 例如,可以通过“ppat”4cc标识该渐进精化模式项属性。可以使用其他4cc。

[0162] 在该结构中,pattern_type指定所使用的精化模式的类型。可以定义若干精化模式类型,包括:上下、左右、左上到右下对角线、左下到右上对角线、中心到边界、顺时针……。此外,可以定义针对重叠的特定精化模式类型,例如,前到后或者最大到最小。

[0163] 作为示例,可以针对pattern_type定义以下值:

[0164] • 当等于0时,基于行从上到下进行精化;这保证了输入图像的数据在基于行被排序。

[0165] • 当等于1时,基于列从左到右进行精化;

[0166] • 当等于2时,基于对角线从左上至右下进行精化

[0167] • 当等于3时,从中心到边界进行精化;

[0168] • 当等于4时,基于输入图像项以逆时针顺序进行精化;

[0169] • 当等于5时,基于输入图像项从前到后进行精化(即,根据叠加的分层顺序);

[0170] • 当等于6时,基于输入图像项从最大项到最小项进行精化。

[0171] `reverse_flag`指示由精化模式定义的排序被反向。例如,利用反向标志,上下精化模式变成下上精化模式。可能地,可以从`ProgressivePatternProperty`中去除该标志,并且用新的精化模式类型来替换该标志。例如,可以添加下上精化模式类型。

[0172] `bidirectional_flag`指示排序从由精化模式定义的两端开始。例如,上下精化模式变成如下的精化模式:渐进精化从顶行和底行这两者开始并且朝着中心行进行。该标志可以与`reverse_flag`组合,从而使在应用与`bidirectional_flag`相关联的重新排序之后获得的排序反向。可能地,可以从`ProgressivePatternProperty`去除该`bidirectional_flag`,并且用新的精化模式类型替换该`bidirectional_flag`。例如,可以添加上下到中心精化模式类型。

[0173] `parameter_flag`指示是否存在针对精化模式的参数值。如果该标志的值是1,则参数值存在于项属性中。该参数值的含义取决于精化模式类型。例如,顺时针精化模式可以默认从与以顺时针顺序连续的时钟上的中午相对应的位置开始。针对该精化模式的参数值可以通过指示不同的开始小时、或通过给出以度数为单位的开始角度来指示不同的开始位置。可以针对所有精化模式类型存在参数值。也可能针对所有精化模式类型而不存在参数值。参数值的存在可取决于模式类型。参数值可以是一组值或更复杂的结构。

[0174] `scale`值指示由精化模式类型定义的渐进步阶被分组在一起成单个较粗略步阶(表示为经缩放渐进步阶)。在这种情况下,应当由读取器绘制仅经缩放渐进步阶而不是各单个渐进步阶。`scale`值可以指示多少个渐进步阶被分组在一起。例如,针对左右精化模式类型的`scale`值2指示各经缩放渐进步阶的精化一次包含针对两列的精化。`scale`值也可以是分数值。在这种情况下,针对由项属性指定的精化模式的经缩放渐进步阶 n 对应于如由精化模式类型定义的渐进步阶编号 $[n \times \text{scale}]$,即,渐进步阶的索引 n 乘以`scale`值的向下取整的值。在该公式中,与子图像的低质量版本相对应的初始步阶被编号为0。例如,在`scale`值1.5的情况下,上下排序交替经缩放渐进步阶,从而添加1或2行高质量图像。可能地,可以从`ProgressivePatternProperty`中除去该`scale`值,并且用新的精化模式类型来替换该`scale`值。例如,可以存在上下 $\times 2$ 行(top-bottom by 2rows)精化模式类型。

[0175] 在变型中,`scale`值可以被指示有多少个渐进步阶被分组到各个经缩放渐进步阶中的编号列表所替换。例如,如果列表具有针对上下精化模式的值[2,1,3],则第一经缩放渐进步阶的精化包含2行高质量图像、下一个(1行)和最后一个(3行)。如果存在比列表中包含的值的总和更多的渐进步阶,则可以重复该列表以继续对渐进步阶进行分组,可以使用默认组大小对剩余的渐进步阶进行分组,或者可以单独保持剩余的渐进步阶。

[0176] 在该实施例中,在图5的步骤510中,所获得的渐进规范包括模式类型,并且可以包括指示是否使用反转排序的标志、指示是否使用双向排序的标志、用于渐进步阶的缩放值、以及与模式类型相关联的参数值。

[0177] 在图5的步骤520中,生成用于表示所有这些规范的项属性结构。该项属性结构在步骤540处被写为HEIF文件的“meta”框的一部分,并且与对应于HEIF文件中的主图像的图像项相关联。例如,项属性结构被写为项属性容器框“ipco”的一部分,并且经由项属性关联框“ipma”与对应于主图像的图像项相关联。

[0178] 在图5的步骤530中,检索由模式类型定义的排序并将其应用于渐进数据块。该排序可以由反向标志和双向标志来修改。

[0179] 在该实施例中,在图6的步骤620中,通过解析与图像项相关联的渐进精化模式属性项来提取针对该图像项的渐进信息。使用在渐进精化模式属性项中指定的模式类型以及该属性项中包含的不同标志和值,计算渐进步阶的列表。各渐进步阶由其添加到先前渐进步阶中的子图像版本的列表来指定。例如,在图4f的情况下,该模式是上下模式。存在三个渐进步阶:针对网格项的各行中的一个。并且各渐进步阶包含与网格项的三列相对应的三个子图像版本。

[0180] 在步骤640处,检查在与渐进步阶(或当scale值存在且不等于1时经缩放渐进步阶)相关联的列表中包含的所有子图像版本是否已经被加载。

[0181] 可能地,在步骤620处不计算渐进步阶。在这种情况下,步骤640的检查可以基于接收到的子图像版本的数量。例如,每当已经接收到新的预定义数量的新子图像版本时,或者当接收到的新子图像版本的数量是子图像版本总数的预定义百分比时,检查可以是肯定的。在这些检查中,可能不考虑可能低质量的子图像版本。

[0182] 检查还可以基于接收到的字节的数量。例如,每当已经接收到新的预定义数量的字节时,或者当新接收到的字节的数量是HEIF文件的总字节数量的预定义百分比时,检查可以是肯定的。可能地,在这些检查中,可能不考虑与低质量子图像版本的内容和/或HEIF文件结构的内容(诸如“meta”框)相对应的字节。

[0183] 在变型中,scale值可以由scale_fraction值替换,该scale_fraction值指示有多少个渐进步阶作为渐进步阶总数的分数被分组在一起成经缩放渐进步阶。

[0184] 在另一变型中,scale值可以由scale_number值替换,该scale_number值指示渐进步阶被分组在一起成等于这个scale_number值的多个经缩放渐进步阶。

[0185] 分析精化模式

[0186] 在第一实施例的第二变型(下文中称为分析精化模式)中,以分析方式定义渐进精化模式。在该实施例中,渐进信息包括基于等式和相关联的参数的输入图像的扫描顺序的指示。

[0187] 该变型有利地使得在HEIF文件中使用的可能的渐进精化模式中实现丰富的表达性。

[0188] 可以在渐进精化模式项属性内定义分析精化模式。该渐进精化模式项属性的结构可以是:

```

aligned(8) class ProgressivePatternProperty
extends ItemFullProperty('ppat', version = 0, flags = 0) {
    unsigned int(2)    pattern_type;
    unsigned int(1)    bidirectional_flag;
    unsigned int(5)    reserved;
    // Linear
    if (pattern_type == 0) {
[0189]        unsigned int(8) start_x;
        unsigned int(8) start_y;
        unsigned int(8) end_x;
        unsigned int(8) end_y;
        unsigned int(16) step_numerator;
        unsigned int(16) step_denominator;
    }
    // Radial
    else if (pattern_type == 1) {
        unsigned int(8) center_x;
        unsigned int(8) center_y;
        int(16)          step_numerator;
        unsigned int(16) step_denominator;
    }
    // Circular
[0190] else if (pattern_type == 2) {
        unsigned int(8) center_x;
        unsigned int(8) center_y;
        unsigned int(16) starting_angle;
        int(16)          step_numerator;
        unsigned int(16) step_denominator;
    }
}

```

[0191] 在该结构中, pattern_type指示精化模式的一般类型, 例如是线性的、径向的还是圆形的。

[0192] bidirectional_flag指示排序从由精化模式定义的两端开始。

[0193] 如果pattern_type值是0, 则精化模式是线性精化模式。线性精化模式可以是例如上下精化模式、或左右精化模式。

[0194] start_x和start_y值指示精化模式的开始点, 该点是网格项的中心, 该中心在网格中的坐标由这些值给出。

[0195] 类似地, end_x和end_y值指示精化模式的结束点, 该点是网格项的中心, 该中心在网格中的坐标由这些值给出。

[0196] 最后, step_numerator和step_denominator值指示各渐进精化步阶的大小(等于step_numerator除以step_denominator)。

[0197] 为了确定网格项的子图像的高质量版本属于哪个渐进步阶, 使用网格坐标来实现

以下计算。首先,网格项的中心被正交地投影在由开始点和结束点定义的线上。计算该投影点与开始点的距离。如果投影点属于也包含结束点的半条线,则该距离被认为是正的,并且在另一情况下是负的。然后,该距离除以渐进精化步阶的大小、向上取整并且增加一以获得该网格项所属于的渐进步阶的索引。如果所计算的索引低于一,则网格项属于第一渐进步阶。如果所计算的索引大于结束点的索引,则网格项属于最后的渐进步阶。

[0198] 网格项的渐进步阶的索引的计算可以由以下公式来概括:

$$[0199] \quad i_G = \left\lceil \frac{\frac{\overrightarrow{P_i G} \cdot \overrightarrow{P_i P_f}}{\|\overrightarrow{P_i P_f}\|}}{\frac{\text{step_numerator}}{\text{step_denominator}}} \right\rceil + 1$$

[0200] 其中, P_i 是精化模式的开始点, P_f 是精化模式的结束点,并且G是网格项的中心点。

[0201] 如果pattern_type值是1,则精化模式是径向精化模式。径向精化模式可以是例如中心边界精化模式。

[0202] center_x和center_y值指示径向精化模式的开始点作为在网格中的给定位置处的网格项的中心。

[0203] 最后,step_numerator和step_denominator值指示各渐进精化步阶的大小。step_numerator是有符号整数,从而允许大小是正的或负的。正的大小指示渐进步阶从精化模式的中心向外进行,而负的大小指示渐进步阶向内朝向精化模式的中心进行。

[0204] 可能地,step_numerator是无符号整数,并且添加标志来用信号通知渐进步阶是向外还是向内进行。

[0205] 为了确定网格项属于哪个渐进步阶,使用网格坐标来实现以下计算。计算网格项的中心与径向精化模式的开始点的中心之间的距离。该距离除以大小的值,向上取整并增加一,以获得该网格项所属于的渐进步阶的相对索引。一旦已计算出针对所有网格项的渐进步阶的所有相对索引,则通过相对索引减去最小相对索引的值来计算绝对索引。

[0206] 网格项的渐进步阶的索引的计算可以由以下公式来概括:

$$[0207] \quad ri_G = \left\lceil \frac{\|\overrightarrow{CG}\|}{\frac{\text{step_numerator}}{\text{step_denominator}}} \right\rceil + 1$$

$$[0208] \quad i_G = ri_G \pm \min_G ri_G$$

[0209] 其中,C是径向精化模式的开始点并且G是网格项的中心点。

[0210] 如果pattern_type值是2,则精化模式是圆形精化模式。圆形精化模式可以是例如顺时针精化模式。

[0211] center_x和center_y值指示圆形精化模式的参考点作为在网格中的给定位置处的网格项的中心。

[0212] starting_angle值指示圆形精化模式的开始方向,其中,值0对应于朝向图像上部的方向,并且starting_angle值以具有顺时针定向的度数表示。

[0213] 最后,step_numerator和step_denominator值指示各渐进精化步阶的大小。step_numerator是有符号整数,从而允许大小是正的或负的。正的大小指示渐进步阶围绕参考点

顺时针进行,而负的大小指示渐进步阶围绕参考点逆时针进行。

[0214] 可能地,step_numerator是无符号整数,并且添加标志来用信号通知渐进步阶是顺时针还是逆时针进行。

[0215] 为了确定网格项属于哪个渐进步阶,使用网格坐标来实现以下计算。计算开始方向与从参考点到网格项中心点的向量之间的角度。该角度除以大小的值,向上取整并增加一,以获得该网格项所属于的渐进步阶的相对索引。一旦已计算出用于针对网格项的渐进步阶的所有相对索引,则通过相对索引减去最小相对索引的值来计算绝对索引。

[0216] 网格项的渐进步阶的索引的计算可以由以下公式来概括:

$$[0217] \quad ri_G = \left\lceil \frac{(\vec{d}, \vec{RG})}{\frac{step_numerator}{step_denominator}} \right\rceil + 1$$

$$[0218] \quad i_G = ri_G - \min_G ri_G$$

[0219] 其中, $\vec{d}$ 是对应于开始方向的向量, R 是圆形精化模式的参考点并且 G 是网格项的中心点。

[0220] 可以使用其他坐标系来代替网格坐标。例如,可以使用像素坐标,可以使用链接到编码块的坐标,或者可以使用与预定义数量的像素相对应的坐标。可能地,水平单元和垂直单元可以对应于不同数量的像素。

[0221] 可能地,包含在渐进精化模式项属性中的值可以被编码在更大数量的位上。这例如在使用像素坐标时可能是有用的。例如,start_x和start_y值可以被编码在16位或32位上。

[0222] 可能地,start_x、start_y、end_x、end_y、center_x和center_y值可以被编码为有符号整数,以允许这些点落在网格自身之外。

[0223] 可能地,可以定义其他精化模式类型。例如,可以定义用于提供百叶窗效应的模式。

[0224] 可能地,一些标志可以被添加到精化模式以更精确地定义所使用的不同点的位置。例如,可以添加一些标志以指定点是在相应网格项的中间还是在网格项的拐角之一处。可能地,2位标志可以用于使得能够将各点定位在网格项的四个位置之一上:左上角、中心、左边界的中间和上边界的中间。

[0225] 多级变型

[0226] 在这些先前描述的实施例的变型中,子图像具有多于两个不同的版本。例如,各子图像可以具有低质量版本、中质量版本和高质量版本、或者甚至更多的中间版本。

[0227] 在该变型中,相同的精化模式可以被重复地用于描述不同质量级别之间的所有转变。例如,相同的精化模式可以用于定义低质量子图像和中质量子图像之间的渐进步阶,并且还用于定义中质量子图像和高质量子图像之间的渐进步阶。

[0228] 例如,在图4f中,使用上下渐进精化模式,存在与网格项的三行相对应的三个渐进步阶。对于各子图像有三个版本,这将导致这些行的顺次渐进精化中的七个渐进步阶:仅具有低质量子图像的初始渐进步阶,随后是各自针对一行添加中质量子图像的三个渐进步阶,随后是各自针对一行添加高质量子图像的三个其他渐进步阶。

[0229] 可能地,不同的精化模式可以用于描述子图像的不同版本之间的转变。例如,第一精化模式可以用于定义低质量子图像和中质量子图像之间的渐进步阶,并且第二精化模式可以用于定义中质量子图像和高量子图像之间的渐进步阶。

[0230] 为了允许若干精化模式,渐进精化模式项属性可以包括其包含的精化模式的数量和用于描述这些模式中的各个模式的循环。

[0231] 可能地,可以定义特定的多级精化模式,以允许混合子图像的不同版本。例如,在中心到边界精化模式中,可以在边界子图像的中质量版本之前的渐进步阶中使用中心子图像的高质量版本。

[0232] 在预定义精化模式变型中,可以定义新的多级精化模式。

[0233] 可能地,step_offset参数可以被添加到渐进精化模式描述,以指示在哪个渐进步阶引入针对子图像的版本的新质量级别。在图4f的示例中,利用值为2的step_offset参数,第一行的高质量子图像被包含在与第三行的中质量子图像相同的渐进步阶中:在利用中质量子图像精化第一行之后的2个渐进步阶,实现利用高质量子图像精化该第一行。

[0234] 可能地,version_split_flag参数可以优选地与step_offset参数相组合地被添加到渐进精化模式描述。如果该标志的值为真,则在精化包含与当前质量级别相对应的子图像的版本的渐进步阶之前,与新质量级别相对应的子图像的版本被包含在单独的渐进步阶的精化中。在图4f的示例中,在step_offset参数具有值2的情况下,在精化包含第二行的中质量版本的渐进步阶之后,下一渐进步阶的精化仅包含第一行的高质量版本,并且随后是精化包含第三行的中质量版本的另一渐进步阶。

[0235] 可能地,可以针对子图像的最低质量版本定义精化模式。在图4f的示例中,可以将上下精化模式应用于子图像的低质量版本,并且第一渐进步阶可以仅包含网格的第一行中的子图像的低质量版本。

[0236] 不完全级别变型

[0237] 在变型中,可能缺失子图像的一些版本。在图4f的示例中,右下子图像的高质量版本h9可能缺失,或者左下子图像的低质量版本l7可能缺失。

[0238] 可能地,这些缺失子图像恰好被忽略。如果渐进步阶的精化仅包含缺失子图像,则恰好在主图像的渐进绘制期间跳过该渐进步阶。

[0239] 可能地,通过将缺失的子图像的列表添加到渐进精化模式项属性来指示这些缺失子图像。

[0240] 可能地,如果在渐进步阶的精化中包含的缺失子图像版本的数量低于预定阈值,则在步骤640处,渐进步阶仍然视为被加载。该阈值可以被定义为子图像版本的数量。该阈值还可以被定义为在渐进步阶的精化中包含的子图像版本的总数量的百分比(例如,子图像版本的5%或10%)。可能地,渐进步阶可以视为被加载,而在该渐进步阶的精化中包含的一些子图像版本仅在自从前一渐进步阶已被加载以来的经过时间大于或等于预定义阈值时缺失。

[0241] 当使用图5的步骤生成HEIF文件时,渐进规范可以包括不被包括在所得HEIF文件中的子图像的列表。

[0242] 多级模式

[0243] 在变型中,在渐进步阶内的渐进精化顺序可以进一步由其他渐进精化模式来指

定。例如，上下渐进精化模式指示主图像的渐进精化是逐行实现的。第二渐进精化模式可以用于指示在各行内，渐进精化从行的中心开始并且朝向行的边界进行。

[0244] 可以使用例如以下结构在渐进精化模式项属性内指定该第二渐进精化模式：

```
aligned(8) class ProgressivePatternProperty
extends ItemFullProperty('ppat', version = 0, flags = 0) {
    unsigned int(8)    main_pattern_type;
    unsigned int(1)    main_reverse_flag;
    unsigned int(1)    main_bidirectional_flag;
    unsigned int(1)    main_parameter_flag;
    unsigned int(5)    main_reserved;
    unsigned int(8)    main_scale;
[0245]    if (parameter_flag == 1) {
        unsigned int(32)    main_parameter;
    }
    unsigned int(8)    secondary_pattern_type;
    unsigned int(1)    secondary_reverse_flag;
    unsigned int(1)    secondary_bidirectional_flag;
    unsigned int(1)    secondary_parameter_flag;
    unsigned int(5)    secondary_reserved;
    unsigned int(8)    secondary_scale;
    if (parameter_flag == 1) {
[0246]        unsigned int(32)    secondary_parameter;
    }
}
```

[0247] 可以针对不同的渐进步阶指定若干第二渐进精化模式。

[0248] 第二渐进精化模式可以具有与主要渐进精化模式不同的结构。

[0249] 偏移实施例

[0250] 在本发明的第二实施例(下文称为偏移实施例)中，使用HEIF文件中的位置来指示渐进步阶。

[0251] 有利地，该第二实施例提供HEIF文件的渐进步阶的明确且简单的指示。

[0252] 关于这些渐进步阶的一些信息可以在应用级被展示，从而允许应用基于其渐进步阶的数量在若干HEIF文件中选择HEIF文件。

[0253] 步阶位置变型

[0254] 在该第二实施例的第一变型中，对于各渐进步阶，在HEIF文件中存储的渐进信息中，指定作为该渐进步阶的精化中包含的最后子图像版本的一部分的、HEIF文件中的最后字节的位置。

[0255] HEIF文件内的图像项的渐进步阶的位置可以用与该图像项相关联的项属性来指示。该渐进步阶位置项属性可以通过以下结构来描述：

```
aligned(8) class ProgressiveStepLocationProperty
{
    extends ItemFullProperty('ploc', version = 0, flags = 0) {
        unsigned int(16) step_count;
        for (i=0; i < step_count; i++) {
            if (flags & 1) {
                unsigned int(32) item_index;
[0256]         }
            else {
                unsigned int(16) item_index;
            }
            unsigned int(16) extent_index;
        }
    }
}
```

[0257] 不同的大小可以用于该渐进步阶位置项属性的字段。可能地,某些标志(使用针对ItemFullProperty的flags参数或直接编码在渐进步阶位置项属性内的标志值)可以用于指定不同字段的大小。

[0258] 在该结构中,step_count指示渐进步阶位置项属性内所描述的渐进步阶的数量。

[0259] 对于各渐进步阶,item_index值指示项位置框“iloc”中的阵列内的资源的索引。extent_index值指示该资源的区间的索引。渐进步阶的位置是该指示区间的最后字节。

[0260] 可能地,extent_index是可选字段。如果该字段不存在,则渐进步阶的位置是由item_index值指示的项的最后区间的最后字节。

[0261] 可能地,渐进步阶位置项属性不包含extent_index字段。

[0262] 在该实施例中,在图5的步骤510中,渐进规范包括用于定义渐进步阶的一些信息。该信息可以包括例如包含在各渐进步阶精化中的子图像版本的列表。在该实施例中,步骤520和步骤530的排序被切换。

[0263] 首先,在步骤530中,使用渐进步阶的定义来实现渐进数据块的排序。使用该顺序,创建HEIF文件的结构并生成“iloc”框的内容。

[0264] 然后,执行步骤520,基于“iloc”框的内容和渐进步阶的定义生成渐进步阶位置项属性。该项属性在步骤540处被写为HEIF文件的“meta”框的一部分,并且与HEIF文件中的对应于主图像的图像项相关联。例如,项属性结构被写为项属性容器框“ipco”的一部分,并且经由项属性关联框“ipma”与对应于主图像的图像项相关联。

[0265] 在该实施例中,在图6的步骤620中,通过解析渐进步阶位置项属性来提取渐进信息。该渐进信息用于生成主图像的渐进步阶的列表。各渐进步阶由在渐进步阶位置项属性中指示的区间的最后字节的位置指定。

[0266] 可能地,如果extent_index字段不存在,则渐进步阶的位置是在渐进步阶位置项属性中指示的项的最后区间的最后字节。

[0267] 在步骤640处,将渐进步阶的位置与HEIF文件的最后加载的字节进行比较。如果渐进步阶的位置在最后加载的字节的位置之前,则渐进步阶已经被加载。

[0268] 如果渐进步阶根据它们的位置被排序,则仅尚未被加载的第一渐进步阶需要在步骤640处被检查。

[0269] 优选地,渐进步阶根据它们在HEIF文件内的位置顺序在渐进步阶位置项属性中被排序。如果情况并非如此,则渐进步阶可以在已经从渐进步阶位置项属性中被提取之后根据其位置而被重新排序。

[0270] 可能地,在该变型中,项可以通过其标识符而不是其在项位置框中的阵列内的索引来标识。

[0271] 步阶精化变型

[0272] 在第二变型(步阶精化变型)中,各渐进步阶由包含该渐进步阶的精化中所包含的子图像版本的数据的区间的列表来指定。

[0273] 组成了图像项的不同渐进步阶精化的区间的列表可以用与该图像项相关联的项属性来指示。该渐进步阶精化项属性可以通过以下结构来描述:

```
aligned(8) class ProgressiveStepRefinementProperty
extends ItemFullProperty('prco', version = 0, flags = 0) {
    unsigned int(16) step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(16) item_count;
        for (j=0; j < item_count; j++) {
            if (flags & 1) {
                unsigned int(32) item_index;
            }
            else {
[0274]         unsigned int(16) item_index;
            }
            unsigned int(16) extent_count;
            for (k=0; k < extent_count; k++) {
                unsigned int(16) extent_index;
            }
        }
    }
}
```

[0275] 不同的大小可以用于该渐进步阶精化项属性的字段。可能地,某些标志(其使用针对ItemFullProperty的flags参数或直接编码在渐进步阶精化项属性内的标志值)可以用于指定不同字段的大小。

[0276] 在该结构中,step_count指示渐进步阶精化项属性内所描述的渐进步阶的数量。

[0277] 对于各渐进步阶,item_count值指示在渐进步阶的精化中包括多少项。对于各项,item_index值指示项位置框“iloc”中的阵列内的资源的索引。

[0278] 对于各项,extent_count值指示在针对该项的渐进步阶的精化中包括多少区间。对于各区间,extent_index值指示该资源的区间的索引。

[0279] 使用该变型的HEIF文件的生成类似于用于“iloc”参考的生成。

[0280] 在该变型中,在图6的步骤620中,各渐进步阶由作为其精化的一部分的区间列表来指定。

[0281] 在步骤640处,检查属于渐进步阶的精化的所有区间是否已被加载。

[0282] 该变型可以与拆分成空间部分(例如,区块或子图片或甚至层)的图像项相关,其中,各空间部分具有由“iloc”框中的区间标识的其数据。

[0283] 可能地,在该变型中,项可以通过其标识符而不是其在项位置框中的阵列内的索引来标识。

[0284] 可能地,通过以下结构来描述渐进步阶精化项属性:

```
aligned(8) class ProgressiveStepRefinementProperty
extends ItemFullProperty('prco', version = 0, flags = 0) {
    unsigned int(16) step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(16) item_count;
        for (j=0; j < item_count; j++) {
            if (flags & 1) {
[0285]                unsigned int(32) item_index;
            }
            else {
                unsigned int(16) item_index;
            }
            unsigned int(16) extent_count;
        }
    }
}
```

[0286] 在该变型中,extent_count值指示在渐进步阶的精化中或在先前渐进步阶的精化中包括多少用于项的区间。在该变型中,对于项,按照解码顺序组织区间。

[0287] 可能地,通过将在渐进步阶精化项属性中针对该渐进步阶指定的区间的列表与在先前渐进步阶的精化中存在的区间相比较来计算作为渐进步阶的精化的一部分的区间。在步骤640处使用这些区间以检查是否已经加载渐进步阶。

[0288] 可能地,修改步骤640的检查以检查在渐进步阶精化项属性中针对该渐进步阶指定的所有区间是否已被加载。

[0289] 可能地,在该变型中,extent_count字段是可选的。如果该字段不存在,则项的所有区间被包括在渐进步阶的精化中或先前渐进步阶的精化中。

[0290] 在变型中,渐进步阶的规范可以使用至HEIF文件中的指针。该变型可以与步阶位置变型或与步阶精化变型一起使用。

[0291] 可能地,这些指针可以是到HEIF文件中的绝对偏移。

[0292] 例如,在步阶位置变型的情况下,对于各渐进步阶,作为包含在该渐进步阶中的子图像版本的一部分的最后字节的位置被直接指定为到HEIF文件中的偏移。

[0293] 作为另一示例,在步阶精化变型的情况下,针对各渐进步阶,在渐进步阶精化中包含的子图像版本的位置被指定为到HEIF文件中的偏移的列表,其中各偏移与长度相关联。

[0294] 可能地,可以使用与项位置框所使用的机制类似的机制来指定这些指针。可能地,可以仅使用项位置框所使用的机制的一部分。例如,这些指针可以被指定为到ItemDataBox“idat”中或到IdentifiedMediaDataBox“imda”中的偏移。

[0295] 基于“iloc”的变型

[0296] 在变型中,在项位置框的新版本中实现渐进步阶的规范。

[0297] 在该变型中,向各个区间添加标志以指示该区间的最后字节是否是在渐进步阶精化中包含的子图像版本的最后字节。以这种方式,可以根据包含在项位置框中的信息来重构渐进步阶。该标志可以命名为progressive_step_end。

[0298] 可能地,该标志可以被添加到各项而不是各区间。

[0299] 可能地,渐进步阶编号字段被添加到各区间,以指示该区间被包含在哪个渐进步阶的精化中。值0可以被保留以指示区间不被包含在任何渐进步阶的精化中。该渐进步阶编号字段可以被命名为progressive_step_number。可能地,可以使用标志来指示该字段的在存在。

[0300] 可能地,渐进步阶编号结束字段被添加到各区间,以指示该区间的最后字节是在相应渐进步阶的精化中包含的子图像版本的最后字节。可以保留值0以指示区间的最后字节不是在渐进步阶的精化中包含的子图像版本的最后字节。该渐进步阶编号字段可以被命名为progressive_step_number_end。可能地,可以使用标志来指示该字段的在存在。

[0301] 项实施例

[0302] 在本发明的第三实施例(下文称为项实施例)中,这些渐进步阶是使用它们在其精化中所包含的子图像版本的指示来指示的。

[0303] 有利地,该第三实施例提供了组成主图像的子图像与用于该主图像的渐进步阶之间的简单链接。

[0304] 有利地,该第三实施例的一些变型仅需要HEIF文件中的最小信号通知。

[0305] 在该第三实施例的第一变型(下文中称为版本列表变型)中,对于各渐进步阶,在该渐进步阶的精化中包含的子图像版本的列表是在存储在HEIF文件中的渐进信息中指定的。

[0306] 在HEIF文件内的图像项的渐进步阶精化中包含的子图像版本可以用与该图像项相关联的项属性来指示。该渐进版本列表项属性可以通过以下结构来描述:

```
aligned(8) class ProgressiveVersionListProperty
extends ItemFullProperty('pvli', version = 0, flags = 0) {
    unsigned int(16) step_count;
    for (i=0; i < step_count; i++) {
[0307]         unsigned int(16) item_count;
        for (j=0; j < item_count; j++) {
            if (flags & 1) {
                unsigned int(32) item_ID;
            }
            else {
                unsigned int(16) item_ID;
[0308]            }
        }
    }
}
```

[0309] 在该结构中,step_count指示渐进版本列表项属性内所描述的渐进步阶的数量。

[0310] 对于各渐进步阶, item_count值指示被包含在该渐进步阶的精细化中的子图像版本的数量。

[0311] 对于各子图像版本, item_ID值指示与该子图像版本相对应的图像项的标识符。

[0312] 在该实施例中, 在图5的步骤510中, 渐进规范包括用于定义渐进步阶的一些信息。该信息可以包括例如各渐进步阶的精细化中包含的子图像版本的列表。

[0313] 在图5的步骤520中, 基于渐进步阶的定义来生成渐进版本列表项属性。该项属性在步骤540处被写为HEIF文件的“meta”框的一部分, 并且与HEIF文件中的对应于主图像的图像项相关联。例如, 项属性结构被写为项属性容器框“ipco”的一部分, 并且经由项属性关联框“ipma”与对应于主图像的图像项相关联。

[0314] 在该实施例中, 在图6的步骤620中, 通过解析渐进版本列表项属性来提取渐进信息。该渐进信息用于生成主图像的渐进步阶的列表。各渐进步阶由其在精细化中包含的子图像版本的列表来指定。

[0315] 在步骤640处, 检查是否已经加载了在渐进步阶的精细化中包含的子图像版本的列表中指定的图像项的所有数据。

[0316] 可能地, 子图像可以被编码为由单个图像项描述的多层图像。子图像的不同版本被编码为该图像项的不同层。在这种情况下, 子图像的所有版本可以由单个图像项描述。渐进版本列表项属性可以被修改以指示与子图像的版本相对应的编码层。该修改的渐进版本列表项属性可以通过以下结构来描述:

```
aligned(8) class ProgressiveVersionListProperty
extends ItemFullProperty('pvli', version = 0, flags = 0) {
    unsigned int(16) step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(16) item_count;
        for (j=0; j < item_count; j++) {
            if (flags & 1) {
                unsigned int(32) item_ID;
            }
            else {
[0317]         unsigned int(16) item_ID;
            }
            unsigned int(1) has_layer;
            if (has_layer == 1) {
                unsigned int(15) layer_id;
            } else {
                unsigned int(7) reserved;
            }
        }
    }
}
```

[0318] 在该结构中, has_layer值是指子图像的版本是否与具有给定item_ID的图像项的特定编码层相对应的标志。如果该has_layer标志的值为真, 则layer_id值指示与子图像

的版本相对应的编码层的标识符。

[0319] 可能地,包含在与子图像相对应的图像项中的不同层可以由与该图像项相关联的项属性来描述。该渐进层项属性可以通过以下结构来描述:

```
aligned(8) class ProgressiveLayerProperty
extends ItemFullProperty('play', version = 0, flags = 0) {
    unsigned int(8)    layer_count;
[0320]    for (i=0; i < layer_count; i++) {
        unsigned int(16) layer_id;
    }
}
```

[0321] 在该结构中,layer_count值指示用于渐进绘制的层的数量。在图像项中存在的层的总数量可以大于所指示的层的数量。

[0322] 对于各层,layer_id值指示该层的标识符。

[0323] 渐进版本列表项属性结构可以被修改以利用该渐进层项属性的优点:

```
aligned(8) class ProgressiveVersionListProperty
extends ItemFullProperty('pvli', version = 0, flags = 0) {
    unsigned int(16) step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(16)    item_count;
        for (j=0; j < item_count; j++) {
            if (flags & 1) {
[0324]                unsigned int(32) item_ID;
            }
            else {
                unsigned int(16) item_ID;
            }
            unsigned int(8)    layer_index;
        }
    }
}
```

[0325] 在该结构中,layer_index值指示与对应于子图像版本的并且由item_ID的值指定的图像项相关联的渐进层项属性内的层的基于1的索引。值0被保留用于指示子图像版本对应于图像项的全部内容。

[0326] 可能地,渐进层项属性可以被若干个“lssel”层选择项属性替换,各“lssel”层选择项属性指示包含在图像项中的层之一。在这种情况下,layer_index值指示与对应于子图像的并且由item_ID的值指定的图像项相关联的“lssel”项属性的基于1的索引,并且指示对应于子图像版本的层。

[0327] 对于这两个变型,在图6的步骤640处,针对在渐进步阶的精细化中包含的子图像版本的列表中指定的各图像项检查是否已经加载了相应的子图像版本。如果没有针对该图像项指定的层标识符,并且已经加载了该图像项的全部数据,则已经加载了相应的子图像版本。如果存在针对该图像项指定的层标识符,并且已经加载了该层的和所有先前层的数据,

则已经加载了相应的子图像版本。

[0328] 可能地,按照渐进精化顺序分配子图像版本的项的标识符。渐进版本列表项属性可以通过以下结构来描述:

```
aligned(8) class ProgressiveVersionListProperty
[0329] extends ItemFullProperty('pvli', version = 0, flags = 0) {
    unsigned int(16) step_count;
    for (i=0; i < step_count; i++) {
        if (flags & 1) {
            unsigned int(32) item_ID;
        }
        [0330] else {
            unsigned int(16) item_ID;
        }
    }
}
```

[0331] 在该结构中,item_ID值指示与包含在渐进步阶的精化中的子图像版本相对应的最后图像项的标识符。

[0332] 项列表变型

[0333] 在该第三实施例的第二变型(项列表变型)中,对于各渐进步阶,在存储在HEIF文件中的渐进信息中,指定版本被包含在该渐进步阶的精化中的子图像的列表。

[0334] 虽然第一变型明确指示包含在渐进步阶的精化中的子图像版本,但该第二变型仅指示版本被包含在渐进步阶的精化中的子图像。这意味着绘制器必须针对关于渐进步阶所指示的各子图像确定哪些子图像版本被包含在该渐进步阶的精化中。

[0335] 版本被包含在HEIF文件内的图像项的渐进步阶精化中的子图像可以用与该图像项相关联的项属性来指示。该渐进项列表项属性可以通过以下结构来描述:

```
aligned(8) class ProgressiveItemListProperty
extends ItemFullProperty('pili', version = 0, flags = 0) {
    unsigned int(16) step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(16) item_count;
        for (j=0; j < item_count; j++) {
            if (flags & 1) {
                [0336] unsigned int(32) item_ID;
            }
            else {
                unsigned int(16) item_ID;
            }
        }
    }
}
```

[0337] 在该结构中,step_count指示渐进项列表项属性内所描述的渐进步阶的数量。

[0338] 对于各渐进步阶,item_count值指示版本被包含在该渐进步阶的精化中的子图像

的数量。

[0339] 对于各子图像, item_ID值指示版本被包含在该渐进步阶的精细化中的图像项的标识符。

[0340] 在该实施例中, 在图5的步骤510中, 渐进规范包括用于定义渐进步阶的一些信息。该信息可以包括例如在各渐进步阶的精细化中包含的子图像版本的列表。

[0341] 在图5的步骤520中, 基于渐进步阶的定义来生成渐进项列表项属性。该项属性在步骤540处被写为HEIF文件的“meta”框的一部分, 并且与HEIF文件中的对应于主图像的图像项相关联。例如, 项属性结构被写为项属性容器框“ipco”的一部分, 并且经由项属性关联框“ipma”与对应于主图像的图像项相关联。

[0342] 优选地, 对于作为主图像的成分的各子图像, 其版本的列表被明确地包括作为HEIF文件的“meta”框的一部分。如果版本基于子图像的编码层, 则渐进层项属性、或层选择项属性的集合(“lsl”)可以用于列出该子图像的版本。否则, “altr”实体组可以用于列出与该子图像的版本相对应的所有图像项。在该“altr”实体组中应首先列出子图像的最高质量版本。

[0343] 在该实施例中, 在图6的步骤620中, 通过解析渐进项列表项属性来提取渐进信息。该渐进信息用于生成主图像的渐进步阶列表。各渐进步阶由版本被包含在渐进步阶的精细化中的子图像的列表来指定。

[0344] 在步骤640处, 检查新版本是否可用于包含在与渐进步阶相关联的列表中的所有子图像。

[0345] 作为变型, 在步骤620中, 可以计算各子图像的版本列表。优选地, 从HEIF文件中的明显描述中提取该版本列表。如果子图像版本是由“altr”实体组所参考的图像项, 则子图像的所有版本的列表是被包含在该“altr”实体组中的图像项的列表。如果子图像版本是具有相关联的渐进层项属性的图像项, 则子图像的所有版本的列表是在该渐进层项属性中指示的编码层的列表。如果子图像版本是由“altr”实体组所参考的并且具有相关联的渐进层项属性的图像项, 则子图像的所有版本的列表是这两种情况的列表的组合。

[0346] 可能地, 从HEIF文件的“meta”框的其他结构推断子图像的版本列表。例如, 如果子图像版本是与缩略图相关联的图像项, 则该缩略图是子图像的版本。

[0347] 一旦已经计算各子图像的该版本列表, 这些版本就按照渐进精细化顺序从最低质量到最高质量进行排序, 并且与渐进步阶相关联。对于各子图像, 其版本被关联以用于列出该子图像的渐进步阶。

[0348] 在该变型中, 在步骤640处, 检查在渐进步阶的精细化中包含的所有子图像版本是否已经被加载。如果子图像版本与多层图像项的编码层相对应, 则检查是否已加载针对该编码层和针对所有先前编码层的数据。否则, 检查是否已加载针对图像项的数据。

[0349] 项参考变型

[0350] 该第三实施例的第三变型(下文中称为项参考变型)类似于该第二变型。代替在项属性中对版本被包含在渐进步阶的精细化中的子图像的列表进行存储, 该列表被存储在项参考框的条目中。

[0351] 项参考框条目可以具有“prli”4cc。条目的from_item_ID字段是主图像的标识符, 并且使用条目的to_item_ID字段指定子图像。该参考类型针对主图像指示在渐进步阶的精

化中包含的图像项的列表。对于各渐进步阶,存在类型“prli”的项参考框条目,并且根据渐进步阶顺序对这些条目进行排序。

[0352] 项计数变型

[0353] 在该第三实施例的第四变型(下文中称为项计数变型)中,对于各渐进步阶,在HEIF文件中的渐进信息中指定在该渐进步阶的精细化中包含的子图像版本的数量。

[0354] 在针对HEIF文件内的图像项的渐进步阶的精细化中包含的子图像版本的该数量可以用与该图像项相关联的项属性来指示。该渐进项计数项属性可以通过以下结构描述:

```
aligned(8) class ProgressiveItemCountProperty
extends ItemFullProperty('pcnt', version = 0, flags = 0) {
    unsigned int(16) step_count;
[0355]     for (i=0; i < step_count; i++) {
        unsigned int(16) item_count;
    }
}
```

[0356] 在该结构中,step_count指示渐进项计数项属性内所描述的渐进步阶的数量。

[0357] 对于各渐进步阶,item_count值指示版本被包含在该渐进步阶的精细化中的子图像的数量。

[0358] 在该实施例中,在图5的步骤510中,渐进规范包括用于定义渐进步阶的一些信息。该信息可以包括例如被包含在各渐进步阶的精细化中的子图像版本的数量。

[0359] 在图5的步骤520中,基于渐进步阶的定义来生成渐进项计数项属性。该项属性在步骤540被写为HEIF文件的“meta”框的一部分,并且与HEIF文件中的对应于主图像的图像项相关联。例如,项属性结构被写为项属性容器框“ipco”的一部分,并且经由项属性关联框“ipma”与对应于主图像的图像项相关联。

[0360] 优选地,对于构成主图像的各子图像,其版本的列表被明确地包括作为HEIF文件的“meta”框的一部分。如果版本基于多层子图像的编码层,则渐进层项属性可以用于列出该多层子图像的版本。否则,“altr”实体组可以用于列出与该子图像的版本相对应的所有图像项。在该“altr”实体组中应首先列出子图像的最高质量版本。

[0361] 在步骤640处,对从先前渐进步阶开始已经加载了新版本的子图像的数量进行计数。然后检查该数量是否大于或等于与当前渐进步阶相关联的编号。

[0362] 作为变型,在步骤620中,可以计算各子图像的版本列表。优选地,从HEIF文件中的明确描述中提取该版本列表。如果子图像版本是由“altr”实体组所参考的图像项,则子图像的所有版本的列表是在该“altr”实体组中包含的图像项的列表。如果子图像版本是具有相关联的渐进层项属性的图像项,则子图像的所有版本的列表是在该渐进层项属性中指示的编码层的列表。如果子图像版本是由“altr”实体组所参考的并且具有相关联的渐进层项属性的图像项,则子图像的所有版本的列表是这两种情况的列表的组合。

[0363] 可能地,从HEIF文件的“meta”框的其他结构推断子图像的版本列表。例如,如果子图像版本是与缩略图相关联的图像项,则该缩略图是子图像的版本。

[0364] 一旦已经计算各子图像的该版本列表,这些版本就按照渐进精细化顺序从最低质量到最高质量进行排序,并且与渐进步阶相关联。对于各子图像,其版本被关联以用于列出该子图像的渐进步阶。

[0365] 在该变型中,在步骤640处,检查在渐进步阶的精细化中包含的所有子图像版本是否已经被加载。如果子图像版本与多层图像项的编码层相对应,则检查是否已加载针对该编码层和针对所有先前编码层的数据。否则,检查是否已加载针对图像项的数据。

[0366] 在变型中,代替存储渐进步阶的精细化中所包含的子图像版本的数量,存储该渐进步阶的内容中所包含的子图像版本的数量。在该变型中,在步骤640处,对所加载的子图像版本的总数量进行计数,并且将其与同该渐进步阶相关联的编号进行比较。

[0367] 在变型中,单个item_count值被用于所有渐进步阶。在该变型中,渐进项计数项属性可以通过以下结构描述:

```
aligned(8) class ProgressiveItemCountProperty
[0368] extends ItemFullProperty('pcnt', version = 0, flags = 0) {
    unsigned int(16)    item_count;
}
```

[0369] 在另一变型中,版本被包含在渐进步阶的精细化中的子图像的数量被指定为子图像版本的总数量的百分比。可能地,该总数量没有考虑低质量子图像版本。

[0370] 项步阶实施例

[0371] 在本发明的第四实施例(下文中称为项步阶实施例)中,这些渐进步阶是使用它们在它们的精细化中所包含的子图像版本的指示来指示的,各子图像版本被指示为该子图像的渐进步阶。

[0372] 有利地,该第四实施例使用相同的机制来描述导出图像的渐进步阶和子图像的不同版本。

[0373] 在该实施例的第一变型中,对于各渐进步阶,在该渐进步阶的精细化中包含的子图像版本的列表是在存储在HEIF文件中的渐进信息中指定的。

[0374] 在针对HEIF文件内的图像项的渐进步阶的精细化中包含的子图像版本可以用与该图像项相关联的项属性来指示。该渐进导出的信息项属性可以通过以下结构描述:

```
aligned(8) class ProgressiveDerivedInformationProperty
extends ItemFullProperty('prdi', version = 0, flags = 0) {
    unsigned int(8)    step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(8) item_count;
        for (j=0; j < item_count; j++) {
            unsigned int(16) input_item_index;
[0375]         unsigned int(1)  has_steps;
            unsigned int(7)  reserved;
            if (has_steps) {
                unsigned int(8)  step_index;
            }
        }
    }
}
```

[0376] 在该结构中,step_count指示在针对相关联的导出图像项的渐进导出的信息项属性内描述的渐进步阶的数量。

[0377] 对于各渐进步阶, `item_count` 值指示该渐进步阶的精细化中所包含的子图像版本的数量。

[0378] 对于各子图像版本, `input_item_index` 值指示与该子图像版本相对应的子图像项的标识符, 作为用于构建导出图像项的输入图像的列表内的子图像的索引。输入图像的列表由来自导出项的“`dimg`”项参考提供。

[0379] 对于各子图像版本, `has_steps` 标志指示由 `input_item_index` 指示的输入图像项是否具有被描述为渐进步阶的一个或多个版本。如果输入图像项具有相关联的渐进步阶, 则 `has_steps` 标志具有值1, 否则具有值0。可能地, 如果输入图像项仅具有单个渐进步阶, 即输入图像自身, 则 `has_steps` 标志可以具有值0。优选地, 索引是基于0的, 其中值0对应于输入图像的第一渐进步阶。

[0380] 如果 `has_steps` 标志具有值1, 则 `step_index` 值是与用于构建导出项的渐进步阶的子图像的版本相对应的输入图像项的渐进步阶的基于0的索引。

[0381] 可以使用类型“`altr`”的实体组用信号通知输入图像项的渐进步阶。类型“`altr`”的实体组可以用于对输入图像项的替代版本进行分组。在这种情况下, 与输入图像项的渐进步阶相关联的索引是它在类型“`altr`”的实体组内的位置。

[0382] 优选地, 使用专用于该任务的更具体的实体组类型(例如, (用于渐进实体组)的类型“`prgr`”的实体组)来用信号通知输入图像项的渐进步阶。类型“`prgr`”的实体组将在下文中以非限制性方式使用, 以描述用于用信号通知作为项列表的项的渐进步阶的任何实体组。

[0383] 在“`prgr`”实体组内, 输入图像项版本优选地以增加质量顺序列出。优选地, 与这些输入图像项版本相对应的数据的排序与实体组内的排序匹配。在“`prgr`”实体组中包含的任何图像项可以用作在同一“`prgr`”实体组中包含的另一图像项的替换, 特别地用于实现在实体组中包含的图像项之一的渐进绘制。优选地, 在渐进绘制“`prgr`”实体组中包含的图像项期间, 在实体组内出现在该图像项之前的图像项作为临时替换被绘制。因此, “`prgr`”实体组中包含的不同图像项优选地对应于类似图像, 但具有不同的质量级别和/或分辨率。

[0384] 导出项的输入图像项可以是与该输入图像项相关联的类型“`prgr`”的实体组内的最后项。该输入图像项也可以不是该实体组内的最后项。例如, 实体组可以包括输入图像项的缩略图, 输入图像项自身作为输入图像的低动态范围图像(LDR)和高动态范围版本(HDR)。该实体组还用信号通知缩略图和LDR图像项可以用于HDR输入图像的渐进绘制。优选地, 如果输入图像项不是类型“`prgr`”的实体组内的最后项, 则在类型“`prgr`”的实体组内出现在该项之后的项不被认为是该输入图像项的版本。

[0385] 在该实施例中, 在图5的步骤510中, 渐进规范包括用于定义渐进步阶的一些信息。该信息可以包括例如各渐进步阶的精细化中包含的子图像版本的列表。

[0386] 在图5的步骤520中, 基于渐进步阶的定义生成渐进导出的信息项属性。该项属性在步骤540处被写为HEIF文件的“`meta`”框的一部分, 并且与HEIF文件中的对应于导出图像的图像项相关联。例如, 项属性结构被写为项属性容器框“`ipco`”的一部分, 并且经由项属性关联框“`ipma`”与对应于导出图像的图像项相关联。

[0387] 此外, 对于与基于渐进步阶的定义具有若干版本的导出图像项相关联的各输入图像项, 生成类型“`prgr`”的实体组以按照增加质量顺序对这些图像项版本进行分组。

[0388] 在该实施例中,在步骤610中,从“meta”框中提取的结构也包括实体组。

[0389] 在步骤620处,从导出的信息项属性和从“prgr”实体组提取渐进信息。

[0390] 在步骤640处,检查是否已经加载了在渐进步阶的精细化中包含的子图像版本的列表中指定的图像版本的所有数据。

[0391] 可能地,导出图像项自身被包含在类型“prgr”的实体组中,该实体组用于指示除了基于不同输入图像版本的图像项之外,还基于其他图像项的渐进步阶。例如,网格项可以被包括在具有缩略图和预览的类型“prgr”的实体组中,并且类型“prdi”的项属性可以与网格项相关联。在这种情况下,网格的渐进显示可以以缩略图开始,随后是预览,并且随后是在“prdi”项属性中描述的所有渐进步阶。可能地,在导出项的绘制期间,并非使用所有这些渐进步阶。

[0392] 可能地,在“prdi”项属性中描述的一个或多个初始渐进步阶可以不包括由导出项使用的所有子图像项的版本。在这种情况下,对导出项的绘制可以生成导出项的部分结果。绘制可以用空白图像或用透明图像替换没有任何版本的子图像项。例如,在“prdi”项属性中描述的第一渐进步阶可以仅包括对应于网格项的顶行的子图像项。然后,该部分结果可以是网格项的顶行。可能地,如果类型“prgr”的实体组包含导出项,则绘制器可以将另一图像项与部分结果组合,以提供导出项的完整绘制(尽管具有可变的质量)。在先前示例中,绘制器可以将网格项的顶行与网格项的缩略图组合以生成完整图像,在该完整图像中,顶行具有良好的质量并且其余的行具有低的质量。

[0393] 可能地,has_steps标志可能不存在。在这种情况下,step_index字段的值指示渐进步阶的索引加一。值0用于指示不存在与输入图像相关联的渐进步阶。可能地,值0用于指示:不存在与输入图像相关联的渐进步阶并且输入图像自身应当被用作输入图像版本,或者存在与输入图像相关联的渐进步阶并且第一渐进步阶应当被用作输入图像版本。

[0394] 在该实施例的第二变型中,渐进导出项信息项属性还可以利用以下结构指示输入图像项的层:

```
aligned(8) class ProgressiveDerivedInformationProperty
extends ItemFullProperty('prdi', version = 0, flags = 0) {
    unsigned int(8) step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(8) item_count;
        for (j=0; j < item_count; j++) {
            unsigned int(16) input_item_index;
            unsigned int(1) has_steps;
[0395]    unsigned int(1) has_layers;
            unsigned int(6) reserved;
            if (has_steps) {
                unsigned int(8) step_index;
            }
            if (has_layers) {
                unsigned int(8) layer_index;
            }
        }
    }
}
```

```

    }
[0396] }

```

[0397] 在该结构中,has_layers标志指示由input_item_index指示的输入图像项是否具有被描述为层的一个或多个版本。如果输入图像项具有被描述为层的一个或多个版本,则has_layers标志具有值1,否则具有值0。可能地,如果输入图像项仅具有对应于整个输入图像的一个层,则has_layers标志可以具有值0。优选地,索引是基于0的,其中值0与输入图像项的第一层相对应。

[0398] 如果has_layers标志具有值1,则layer_index值是用于构建导出项的渐进步阶的输入图像项的层或层集合的基于0的索引。

[0399] 输入图像项的层可以使用“play”渐进层项属性、或若干“lsei”层选择项属性、或描述组成输入图像项的层的任何类似属性来描述。可能地,可以使用具有以下结构的“liip”项属性来描述输入图像项的层:

```

class LayeredImageIndexingProperty
extends ItemFullProperty('liip', version = 0, flags) {
[0400]     field_length = ((flags & 1) + 1) * 16;
        unsigned int(field_length)[4] layer_size;
    }

```

[0401] 本文使用的“liip”4cc是用于分层图像索引属性的4cc的示例,并且可以使用与现有4cc不冲突的任何其他4cc。

[0402] “liip”项属性通过在图像项数据中给出相关联的图像项的层的大小来描述该相关联的图像项的层。

[0403] layer_size字段指示项有效载荷中的与各层相对应的字节的数量。可能地,该字段可以由在项有效载荷中、或在包含项有效载荷的框中、或在HEIF文件自身中以字节为单位的偏移值来替换。

[0404] 可能地,layer_size的编码大小可以是常数(例如,32位)。在这种情况下,不计算field_length值。

[0405] 可能地,用于指示渐进绘制的层的项属性可以具有不同的名称(例如,具有相关联的“plai”4cc的ProgressiveLayerInformationProperty)。可能地,“plai”渐进层信息项属性可以仅指示与其相关联的图像项的层中的一些层:用于图像项的渐进绘制的层。作为变型,“plai”项属性可以使用“prli”4cc。“plai”项属性在下文中以非限制性的方式使用,以描述与图像项相关联的用以用信号通知用作该图像项的渐进步阶的层的任何项属性。此外,除非特别指出,否则,“plai”项属性可以由与图像项相关联的用以用信号通知在该图像项的编码中使用的层的更一般项属性(诸如“liip”项属性)来替换。

[0406] 渐进层信息项属性可以具有以下结构:

```
class ProgressiveLayerInformationProperty
extends ItemFullProperty('plai', version = 0, flags) {
    field_length = ((flags & 1) + 1) * 16;
    unsigned int(8) step_count;
[0407]    for (i=0; i < step_count; i++) {
        unsigned int(field_length) step_size;
    }
}
```

[0408] step_count字段指示由渐进层信息属性描述的渐进步阶的数量。

[0409] step_size字段指示项有效载荷中的与各渐进步阶相对应的字节的数量。

[0410] 可能地,最后渐进步阶的大小可以被省略或者被设置为0值以用于减小渐进层信息项属性的大小。

[0411] 可能地,具有相关联的“plai”项属性的图像项也可以被包含在“prgr”实体组中。

[0412] 优选地,图像项不具有相关联的“prdi”项属性和相关联的“plai”项属性这两者,因为这些项属性对应于不同种类的图像项。

[0413] 类型“prgr”的实体组将在下文中以非限制性方式使用,以描述用于用信号通知作为项列表的项的渐进步阶的任何实体组。

[0414] 对于该变型,在图5的步骤520处,针对若干层用作渐进步阶的各输入图像项生成渐进层信息项属性“plai”。

[0415] 对于该变型,在图6的步骤640处,针对在渐进步阶的精细化中包含的子图像版本的列表中指定的各图像项检查是否已经加载了相应的子图像版本。如果不存在例如通过“plai”项属性针对该图像项指定的层标识符,并且已经加载了该图像项的全部数据,则已经加载了相应的子图像版本。如果存在针对该图像项指定的层标识符,并且已经加载了针对该层和针对所有先前层的数据,则已经加载了相应的子图像版本。

[0416] 可能地,has_layers标志可能不存在。在该情况下,layer_index字段的值指示层的索引加一。值0用于指示不存在与输入图像项相关联的层。可能地,值0用于指示:不存在与输入图像项相关联的层并且整个输入图像项应当被用作输入图像项版本,或者存在与输入图像项相关联的层并且第一层应当被用作输入图像项版本。

[0417] 可能地,layer_index字段可以由与数据的结束处的位置相对应的以字节为单位的偏移值来替换,其中该数据对应于所指示的层。

[0418] 可能地,layer_index字段可以由用于生成输入项的所指示版本的所有层的大小来替换。在后面这两个变型中,当has_layers标志不存在时,值0可以用于指示输入图像没有层并且输入图像是整个输入图像。

[0419] 在该实施例的该第二变型中,通过去除has_steps和has_layers标志,可以简化渐进导出项信息项属性,从而得到以下结构:

```

aligned(8) class ProgressiveDerivedInformationProperty
extends ItemFullProperty('prdi', version = 0, flags = 0) {
    unsigned int(8) step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(8) item_count;
        for (j=0; j < item_count; j++) {
[0420]         unsigned int(16) input_item_index;
            unsigned int(8) step_index;
            unsigned int(8) layer_index;
        }
    }
}

```

[0421] 该结构描述与导出图像项相关联的渐进绘制步阶。各渐进步阶可以被视为指定在代替由导出图像项使用的输入图像时使用的替换图像。如前所述，替换图像是输入图像的版本并且可以是空图像、输入图像的较低质量版本、或输入图像自身。

[0422] 在该结构中，各渐进绘制步阶被描述为相对于先前步阶的差。该描述列出要用于重构导出图像项的替换图像或输入图像的版本。最初，在第一渐进绘制步阶之前，输入图像的替换图像全部对应于空图像。各渐进绘制步阶添加新图像作为替换图像和/或用其他替换图像更新现有替换图像。

[0423] 该结构可以描述替换图像或输入图像的版本两个不同的情况。首先，输入图像可以通过被包含在相同的“prgr”实体组中而由与该输入图像相关联的较低质量的图像来替换。其次，输入图像可以由与组成该输入图像的一些层相对应的并且通过相关联的“plai”项属性描述的较低质量版本来替换。可能地，输入图像可以被包含在“prgr”实体组中并且具有相关联的“plai”项属性。

[0424] 使用该结构，如果输入图像不被包含在“prgr”实体组中，则step_index字段被设置为值0。如果输入图像被包含在“prgr”实体组中，则step_index值是用作该输入图像的替换的图像项的“prgr”实体组内的基于0的索引。

[0425] 使用该结构，如果输入图像不具有相关联的“plai”项属性，则layer_index字段被设置为值0。如果输入图像具有相关联的“plai”项属性，则layer_index值是用作该输入图像的替换的层集合的相关联“plai”项属性内的基于0的索引。

[0426] 使用该结构，如果输入图像既不被包含在“prgr”实体组中也不与“plai”项属性相关联，则step_index和layer_index字段这两者都具有值0。

[0427] 使用该结构，如果输入图像既被包含在“prgr”实体组中又与“plai”项属性相关联，则仅当step_index与输入图像在“prgr”实体组内的位置相对应时，才考虑由layer_index字段指示的“plai”项属性内的信息。否则，layer_index被设置为0，并且忽略“plai”项属性内的信息。

[0428] 可能地，在该结构中，input_item_index可以具有不同的名称，例如，item_index。

[0429] 可能地，在该实施例的第一变型或第二变型中，或者在其他可应用的情况下，step_index字段的含义可以扩展为支持自身导出图像项的输入图像。在该变型扩展中，输入图像可以由如在与该输入图像相关联的“prdi”项属性中所描述的该输入图像自身的渐

进重构步阶来替换。

[0430] 在该变型扩展中,如果输入图像被包含在“prgr”实体组中,则step_index值是用作该输入图像的替换的图像项的“prgr”实体组内的基于0的索引。如果输入图像具有相关联的“prdi”项属性,则step_index值是用于构建该输入图像的替换的渐进步阶的“prdi”项属性内的基于0的索引。如果输入图像既不被包含在“prgr”实体组中也不具有相关联的“prdi”项属性,则step_index值是0。

[0431] 如果输入图像既被包含在“prgr”实体组中又具有相关联的“prdi”项属性,则step_index值被用作“prgr”实体组中或“prdi”项属性中的索引。从0起直到输入图像在“prgr”实体组内的位置减1为止的值用于索引“prgr”实体组内的项。在输入图像在“prgr”实体组内的位置处开始的值用于在“prdi”项属性中对渐进绘制步阶编索引。

[0432] 可能地,可以在渐进导出项信息项属性的定义中引入新字段derived_index,以用于指示用作作为导出图像项的输入图像的替换的渐进绘制步阶。使用该新字段,如果输入图像不具有相关联的“prdi”项属性,则derived_index字段被设置为值0。如果输入图像具有相关联的“prdi”项属性,则derived_index值是用作该输入图像的替换的渐进绘制步阶的相关联“prdi”项属性内的基于0的索引。该derived_index字段与step_index字段的使用的组合可以与layer_index字段与step_index字段的使用的组合类似。针对layer_index字段可能描述的其他变型也可以应用于derived_index字段。优选地,derived_index字段和layer_index字段不与非零值一起使用,这是因为它们对应于不同种类的输入项。

[0433] 可能地,如第三变型中所描述的,step_index字段可以组合三个不同字段的使用:step_index、derived_index和layer_index。

[0434] 可能地,在该实施例的第一变型或第二变型中,或者在别处可应用的情况下,layer_index字段的含义可以扩展为支持作为具有单个输入图像(其具有若干层)的导出图像的输入图像的情况。

[0435] 在该变型扩展中,如果输入图像是导出图像项并且具有包括相关联的“plai”项属性的单个输入图像,则layer_index是用于重构输入图像的替换的层集合在相关联的“plai”项属性内的基于0的索引。

[0436] 可能地,layer_index字段的这种使用可以扩展为使用通过导出项的参考链从输入图像到图像项的参考。例如,layer_index字段可以用作与图像项相关联的“plai”项属性内的基于0的索引,该图像项是导出图像项的单个输入图像,该单个输入图像自身是作为在渐进导出项信息项属性中由input_item_index字段指示的输入图像的、导出图像项的单个输入图像。

[0437] 可能地,以类似的方式,step_index字段或derived_index字段的含义可以扩展为使用单个导出项或通过导出项的参考链从输入图像到图像项的参考。例如,derived_index字段可以用作与网格项相关联的“prdi”项属性内的基于0的索引,该网格项是导出图像项的单个输入图像,该单个输入图像是在渐进导出项信息项属性中由input_item_index字段指示的输入图像,

[0438] 可能地,layer_index字段的含义可以扩展为支持作为具有若干输入图像(其具有一个或多于一个层)的导出图像的输入图像的情况。

[0439] 在该变型扩展中,如果输入图像是导出图像项并且具有若干输入图像并且这些输

入图像至少之一具有相关联的“plai”项属性,则利用相关联的“plai”项属性,将layer_index用于所有输入图像,作为用于重构输入图像的相应替换的相应层集合的相应相关联的“plai”项属性内的基于0的索引。如果layer_index大于或等于在输入图像的相关联的“plai”项属性中描述的层集合的数量,则使用由相关联的“plai”项属性描述的最后层集合。如果输入图像不具有相关联的“plai”项属性,则当重构导出图像项时使用输入图像自身。

[0440] 可能地,在该变型扩展中,layer_index值可以不同地用于选择用于重构各输入图像的替换的层集合。例如,如果一些输入图像具有2个层集合而其他输入图像具有3个层集合,则layer_index的值0可以用于指示所有输入图像的第一层集合。值1可以用于指示具有2个层集合的输入图像的第一层集合和具有3个层集合的输入图像的第二层集合。值2可以用于指示所有输入图像的最后层集合。

[0441] 可能地,在该变型扩展中,layer_index字段的含义可以扩展为使用通过导出项的参考链的从输入图像到若干图像项的参考。可能地,将输入图像链接到不同图像项的导出项的数量可以根据图像项而变化。

[0442] 可能地,以类似的方式,step_index字段或derived_index字段的含义可以扩展为支持作为具有若干输入图像的导出图像的输入图像的情况。可能地,step_index字段或derived_index字段的含义可以扩展为使用通过导出项的参考链的从输入图像到若干图像项的参考。可能地,将输入图像链接到不同图像项的导出项的数量可以根据图像项而变化。

[0443] 可能地,layer_index、step_index和derived_index字段中的两者或多于两者可以联合用于支持作为具有若干输入图像(其具有若干层)的导出图像的输入图像的情况,该输入图像被包含在“prgr”实体组中和/或是具有相关联的“prdi”项属性的导出图像项。根据可用于若干输入图像中的各个输入图像的渐进绘制信息,相应字段用于确定输入图像的渐进步阶的重构。例如,输入图像可以是具有两个输入图像的叠加。叠加的第一输入图像可以是具有用相关联的“plai”项属性指示的两个层集合的图像项。用于该第一输入图像的绘制步阶将基于layer_index字段的值。叠加的第二输入图像可以是以缩略图被包含在“prgr”实体组内的图像项。用于该第二输入图像的绘制步阶将基于step_index字段的值。

[0444] 在该实施例的第三变型中,利用以下结构使用单个字段来指示用于输入项的版本的指示:

```
aligned(8) class ProgressiveDerivedInformationProperty
extends ItemFullProperty('prdi', version = 0, flags = 0) {
    unsigned int(8) step_count;
    for (i=0; i < step_count; i++) {
[0445]         unsigned int(8) item_count;
                for (j=0; j < item_count; j++) {
                        unsigned int(16) item_index;
                        unsigned int(8) step_index;
                }
[0446]     }
}
```

[0447] 在该变型中,step_index值是与用于构建导出项的渐进步阶的输入图像项的版本

相对应的渐进步阶的索引。渐进步阶可以通过类型“prgr”的实体组、或通过“plai”项属性、或通过这两者用信号通知。item_index与先前的变型中描述的input_item_index类似。

[0448] 与要显示的目标项相关联的渐进步阶的完整列表可以在若干阶段中构建。第一，如果目标项被包含在类型“prgr”的实体组中，则在该实体组中包括的各项被添加到列表中作为渐进步阶。优选地，实体组中在目标项之后列出的项不被包括在列表中。如果目标项不被包含在这样的实体组中，则将目标项添加到列表。

[0449] 第二，如果目标项具有相关联的“plai”项属性，则在列表中由与项属性中指示的层相对应的渐进步阶的列表替换目标项。

[0450] 例如，对于图像项，存在包含缩略图和图像项的类型“prgr”的实体组，并且存在与用信号通知两个层的图像项相关联的“plai”项属性。然后，该图像项的渐进步阶是：缩略图、图像项的第一层、和图像项的第二层。

[0451] 第三，如果目标项是导出项并且具有相关联的“prdi”项属性，则在列表中由与“prdi”项属性中指示的渐进步阶相对应的渐进步阶的列表替换目标项。

[0452] 例如，对于网格项，存在包含缩略图和网格项的类型“prgr”的实体组，并且存在与用信号通知两个步阶的网格项相关联的“prdi”项属性。然后，针对该网格项的渐进步阶是：缩略图、针对网格项的第一步阶、和针对网格项的第二步阶。

[0453] 可能地，如果添加到列表并且不同于目标项的项具有相关联的“plai”项属性，则该项可以在列表中由与项属性中指示的层相对应的渐进步阶的列表来替换。

[0454] 例如，对于图像项，存在包含预览和图像项的类型“prgr”的实体组，并且存在与用信号通知两个层的预览相关联的“plai”项属性。然后，该图像项的渐进步阶是：预览的第一层、预览的第二层、和图像项。

[0455] 可能地，如果添加到列表并且不同于目标项的项是导出项并且具有类型“prdi”的相关联的项属性，则该项可以在列表中由与“prdi”项属性中指示的渐进步阶相对应的渐进步阶的列表替换。

[0456] 可能地，如果目标项是不具有相关联的“prdi”项属性的导出图像项，并且导出图像项的至少一个输入图像项具有相关联的“prdi”项属性、具有相关联的“plai”项属性、和/或被包含在类型“prgr”的实体组中，则该目标项在列表中由与使用在“prdi”项属性中指示的输入图像项的渐进步阶的导出图像项的生成相对应的、或者与在“prdi”项属性中指示的层相对应的渐进步阶的列表来替换，以及/或者由类型“prgr”的实体组用信号通知该目标项。可以如这里所描述的那样构建输入图像项的列表渐进步阶。

[0457] 可能地，仅当导出图像项具有单个输入图像项时，才使用该获得导出图像项的渐进步阶的该机制。

[0458] 优选地，具有若干输入图像项的导出图像项具有相关联的“prdi”项属性，以不依赖于该机制。

[0459] 例如，对于作为具有两层的图像项的旋转的导出项，渐进步阶是：图像项的第一层的旋转、和图像项的第二层的旋转。

[0460] 作为另一示例，对于作为具有两个层的图像项的旋转的导出项，图像项以缩略图被包含在类型“prgr”的实体组中，导出项的渐进步阶是：缩略图的旋转、图像项的第一层的旋转、和图像项的第二层的旋转。

[0461] 可能地,可以跟随从导出项到输入图像项的若干连续参考。

[0462] 例如,对于作为自身是具有两个层的图像项的旋转的另一导出项的裁剪的导出项,渐进步阶是:图像项的第一层的旋转的裁剪、和图像项的第二层的旋转的裁剪。

[0463] 可能地,如果添加到列表并且不同于目标项的项是不具有相关联的“prdi”项属性的导出项,则相同的处理可以用于由渐进步阶的列表替换该项。

[0464] 在第四变型中,单个项属性将包含在“prgr”实体组、“prdi”项属性和“plai”项属性中的信息组合。该渐进信息项属性可以通过以下结构来描述:

```
aligned(8) class ProgressiveInformationProperty
extends ItemFullProperty('prif', version = 0, flags = 0) {
    // Number of progressive steps.
    unsigned int(8) step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(32) step_item_id;
        unsigned int(8) layer_index;
    [0465] unsigned int(8) item_count;
        for (j=0; j < item_count; j++) {
            unsigned int(16) item_index;
            unsigned int(8) step_index;
        }
    }
}
```

[0466] 在该结构中,step_count指示渐进信息项属性内所描述的渐进步阶的数量。该数量对应于与该属性相关联的图像项的重构步阶的数量。

[0467] 对于各渐进步阶,step_item_id指示与步阶相对应的图像项的标识符(即,item_ID)。

[0468] 对于各渐进步阶,layer_index指示与由step_item_id字段标识的图像项的渐进步阶相对应的层的索引。如果图像项不具有层,则layer_index的值是0。

[0469] 对于各渐进步阶,如果所标识的图像项是导出项,则item_count值指示该导出项的渐进步阶的精细化中所包含的子图像版本的数量。如果所标识的图像项不是导出项,则item_count的值是0。

[0470] item_index和step_index字段与先前变型中的“prdi”项属性中描述的字段类似。然而,在该变型中,可以直接从与由item_index字段标识的图像项相关联的“prif”项属性重构渐进步阶的完整列表。因此,step_index值可以是到由与标识的图像项相关联的“prif”项属性限定的渐进步阶的列表中的索引。

[0471] 在组合该第四实施例与第二实施例的第五变型中,对于各精细化步阶,指定与该精细化步阶相对应的最后字节的位置。

[0472] 可以使用与图像项相关联的项属性针对各渐进步阶指示这些位置。该渐进步阶结束属性可以通过以下结构来描述:

```
aligned(8) class ProgressiveInformationProperty
extends ItemFullProperty('prif', version = 0, flags) {
    unsigned int(1) layer_based;
    unsigned int(7) reserved;
[0473]    unsigned int(8) num_steps;
    for (i=0; i < num_steps; i++) {
        unsigned int(32) last_byte_position;
    }
}
```

[0474] 在该结构中,layer_based值是指示相关联项的渐进重构步阶是基于层还是基于导出图像项的输入图像项的版本的标志。如果layer_based字段的值是1,则相关联项的渐进重构步阶是基于该图像项的编码层。否则,相关联项是导出图像项,并且其渐进重构步阶是基于使用其输入图像项的不同版本。

[0475] num_step值指示相关联图像项的重构步阶的数量。

[0476] last_byte_position指示用于重构步阶的HEIF文件中的最后字节。当layer_based字段是1时,该last_byte_position指示相关联图像项的层的最后字节。当layer_based字段是0时,该last_byte_position指示作为导出图像项的相关联图像项的输入图像的版本的最后字节。

[0477] 图9示出描述使用“prgr”实体组900的图像项930的渐进绘制的示例。实体组包含图像项和缩略图910和预览920。缩略图和/或预览可以用于图像项的渐进绘制。

[0478] 图10示出描述编码有若干层1030、1040和1050的图像项1020的渐进绘制的示例。图像项1020在缩略图1010旁边被包含在“prgr”实体组1000中,从而指示缩略图是可以用于图像项的渐进绘制的图像项的替代版本。

[0479] 此外,“plai”项属性1060与图像项1020相关联,从而指示图像项的两个渐进步阶可以用于渐进绘制。第一渐进步阶使用图像项的层0(1030)和层1(1040)。第二渐进步阶附加地使用图像项的层2(1050)。

[0480] 在该示例中,主图像对应于来自图像项1020的层2(1050)的重构图像(当存在层之间的一些编码相依性时可能取决于先前层),并且层1030、1040和1050对应于图像项1020的子图像。在该示例中,各层或子图像版本为子图像自身。

[0481] 图11示出描述作为网格的导出图像项1120的渐进绘制的示例。导出图像项在缩略图1110旁边被包含在“prgr”实体组1100中,从而指示缩略图是可以用于图像项1120的渐进绘制的图像项1120的替代版本。

[0482] 在该示例中,主图像对应于导出图像项1120。

[0483] 该网格图像项1120由以 2×2 网格布置的四个输入图像项组成。这些输入图像项是C0(1132)、C1(1142)、C2(1152)、C3(1162)。各输入图像项还具有较低质量版本(分别为L0至L3)。这些较低质量版本通过被包含在“prgr”实体组中而与它们相应的输入图像项相关联。以这种方式,输入图像项C0(1132)及其较低质量版本L0(1131)被包含在“prgr”实体组1130中。以相同的方式,输入图像项C3(1162)及其低质量版本L3(1161)被包含在“prgr”实体组1160中。

[0484] 此外,“prdi”项属性1125与图像项1120相关联,从而指示图像项的三个渐进步阶

可以用于渐进绘制。第一渐进步阶使用作为输入图像项C0、C1、C2和C3的替换的较低质量版本L0、L1、L2和L3。第二渐进步阶用相应的图像项C0和C2替换L0和L2,从而得到网格的全质量左列和较低质量右列。第三渐进步阶用相应的图像项C1和C3替换L1和L3,从而得到整个网格的全质量。

[0485] 图12通过添加使用分层图像1020作为其输入图像项的导出图像项1270来扩展图10所示的示例。在该示例中,主图像对应于导出图像项1270。该导出图像项1270对其输入图像项应用旋转(如由相关联的“irot”项属性1280所指示的)。

[0486] 在该示例中,通过首先绘制被包括在与输入图像项相同的“prgr”实体组中的缩略图并且通过向其应用由“irot”项属性所指示的旋转,可以实现导出图像项1270的渐进绘制。然后,可以通过绘制如由“plai”项属性1060所指示的输入图像项的层0和层1并且通过将“irot”项属性所指示的旋转应用于这些层来实现第二绘制步阶。最后绘制步阶可以通过对图像项应用旋转来绘制完全输入图像项。

[0487] 导出图像项1270可以单独被包括在“prgr”实体组中以指示其支持渐进绘制。该指示也可以通过与其相关联的特定项属性(诸如下文描述的“prog”项属性)来实现。

[0488] 可能地,在该示例的变型中,缩略图1010可以不通过“prgr”实体组1000与分层图像项1020相关联,但是缩略图1010可以通过“prgr”实体组直接与导出图像项1270相关联。在该变型中,缩略图1010还将具有相关联的“irot”属性或者将已经包含旋转图像。在该变型中,可以通过首先绘制缩略图来实现导出图像项1270的渐进绘制,而无需应用由与导出图像项相关联的“irot”项属性1280所指示的旋转。然后,如上所述,可以通过使用输入图像项的层实现第二绘制步阶和第三绘制步阶。

[0489] 可能地,在该示例的另一变型中,分层图像项1020和导出项1270这两者均包含在将它们各自关联到不同缩略图的“prgr”实体组中。优选地,在该变型中,对于导出项1270的渐进绘制,使用通过“prgr”实体组与该导出项相关联的缩略图,并且忽略通过“prgr”实体组1000与分层图像项1020相关联的缩略图1010。可能地,如果缩略图1010可以用于代替与导出项1270相关联的缩略图,例如,在与导出项1270相关联的缩略图之前,该缩略图1010是可用的。可能地,例如,如果这两个缩略图具有不同的质量级别,则这两个缩略图可以用于导出项1270的不同的渐进绘制步阶。

[0490] 在适用的情况下,针对先前变型所描述的所有变型也可以用于该变型。

[0491] 实施例组合

[0492] 可能地,本发明的两个或多于两个实施例可以组合。在实施例的组合中,可以组合描述与不同实施例相关的渐进信息的结构。这些结构可以保持分离。

[0493] 特别地,将精化模式实施例与其他实施例之一进行组合可以是有利的。在该组合中,渐进精化模式指示在HEIF文件中使用的渐进精化策略,而偏移指示或项指示指定各渐进步阶的精化。

[0494] 例如,用于指定渐进精化模式和各渐进步阶的位置这两者的项属性可以具有以下结构:

```
aligned(8) class ProgressivePatternLocationProperty
extends ItemFullProperty('pplo', version = 0, flags = 0) {
    unsigned int(8)    pattern_type;
    unsigned int(1)    reverse_flag;
    unsigned int(1)    bidirectional_flag;
    unsigned int(1)    parameter_flag;
    unsigned int(5)    reserved;
    unsigned int(8)    scale;
    if (parameter_flag == 1) {
        unsigned int(16)    parameter;
    }
[0495] unsigned int(16)    step_count;
    for (i=0; i < step_count; i++) {
        if (flags & 1) {
            unsigned int(32)    item_index;
        }
        else {
            unsigned int(16)    item_index;
        }
        unsigned int(16)    extent_index;
    }
}
```

[0496] 可能地,本发明的实施例的一些部分可以与本发明的其他实施例组合。

[0497] 实现变型

[0498] 在通过子图像的标识符指示该子图像的变型中,代替指定各子图像的标识符,可以使用将主图像链接到子图像的项参考条目中的子图像的索引。

[0499] 可能地,代替在单个项属性中指定若干渐进步阶,各渐进步阶可以在其自身的项属性中指定。这些项属性可以根据渐进步阶顺序进行排序。

[0500] 可能地,用于描述渐进信息的不同项属性可以用于主图像的各类型。例如,可以存在用于描述链接到网格项的渐进信息的特定项属性和用于描述链接到叠加项的渐进信息的不同项属性,各项属性具有其自己的pattern_type值列表。

[0501] 可能地,渐进信息可以被包含在例如具有“prog”类型的新项(渐进项)中。该渐进项可以具有与本发明中描述的项属性之一相同的结构。该渐进项可以利用项参考(例如,利用“prog”4cc)与其应用的主图像项相关联。该项参考可以是从小渐进项到图像项或从图像项到渐进项。

[0502] 可能地,渐进项可以指定单个渐进步阶。在这种情况下,项参考可以将主图像项链接到描述该主图像的渐进步阶的渐进项。渐进项可以根据其相应的渐进步阶的顺序被排序在项参考条目中。

[0503] 可能地,渐进信息可以被包含在新框(例如,具有“prog”4cc的渐进框)中。该新框可以位于HEIF文件的“meta”框内。该渐进框可以具有与本发明中描述的项属性之一相同的结构。此外,例如通过指定主图像项的标识符,该渐进框可以包含其所应用的主图像的指示。该渐进框的实例可以如HEIF文件中存在支持渐进绘制的图像项那样多。

[0504] 可能地,渐进框可以指定单个渐进步阶。在该情况下,渐进框可以包括渐进步阶编号。

[0505] 可能地,渐进步阶描述可以包括渐进步阶编号。

[0506] 可能地,不同的大小可以用于这里描述的各种字段中的一些字段。

[0507] 可能地,可变的大小可以用于这里描述的各种字段中的一些字段。可能地,字段的大小可以基于包含该字段的框的“flags”参数、或基于该框的“version”、或基于包括在该框中的标志字段。可能地,若干字段的大小可以取决于相同的指示。

[0508] 可能地,渐进信息可以指示渐进步阶的内容,而不是渐进步阶的精细化。

[0509] 可能地,主图像的渐进步阶不被组织为列表,而是组织为有向无环图。这意味着渐进步阶可以具有两个或多于两个后续渐进步阶,以及渐进步阶可以具有两个或多于两个先前渐进步阶。这还意味着,渐进步阶自身不应当直接或间接地在其之后或之前。当HEIF文件不旨在被顺序地加载时或者当主图像的内容横跨可能按不确定的顺序加载的若干文件时,这样的组织是有利的。

[0510] 渐进步阶变型

[0511] 优选地,渐进步阶的说明顺序对应于这些渐进步阶的顺序。

[0512] 可能地,一些渐进步阶可以是空的或缺失的。

[0513] 可能地,渐进信息可以定义用于渐进加载子图像的低质量版本的渐进步阶。在图4f的示例中,使用中心到边界的渐进精细化模式,第一渐进步阶可以仅包括中心子图像的低质量版本“15”,然后第二渐进步阶精细化可以包括四个子图像的低质量版本“12”、“14”、“16”和“18”。

[0514] 可能地,在最后渐进步阶之后,一些子图像的一些高质量版本可能缺失。

[0515] 可能地,子图像的一个或多于一个版本可能缺失。

[0516] 具有多个内容的文件

[0517] 可能地,HEIF文件可以包括具有相关联的渐进信息的若干图像项。这些图像项的数据可以被交织或顺次排序。

[0518] 例如,HEIF文件可以包含具有如图4f所描述的渐进精细化模式的第一图像项和具有如图4g所描述的渐进精细化模式的第二图像项。可以如下组织这些图像项的不同子图像版本。首先,存储第一图像项的子图像的所有低质量版本,随后是第二图像项的子图像的低质量版本。然后,存储第一项的第一行的高质量版本,随后是第二项的第一列和第二列的高质量版本,等等。

[0519] 对于同一示例,可以不同地组织两个图像项的不同子图像版本:首先,存储第一图像项的所有子图像版本,随后是第二图像项的所有子图像版本。

[0520] 可能地,可以指示与支持渐进绘制的图像项有关的HEIF文件的交织或顺次组织。该指示可以由HEIF文件中的类别来实现。可以存在用于指示顺次组织的类别(例如“spro”)。可以存在用于指示交织组织的类别(例如“ipro”)。

[0521] 可能地,项属性可以针对支持渐进绘制的各图像项指示是顺次存储还是以交织的方式存储。

[0522] 可能地,不支持渐进绘制的其他图像项或其他非图像项可以被包括在包含支持渐进绘制的一个或多于一个图像项的HEIF文件中。可能地,在所有渐进图像项之后存储这些

其他图像项或非图像项。

[0523] 可能地,与图像项相关联的渐进信息指示与该项相关联的元数据与该图像项的渐进内容相关地存储在何处。例如,渐进信息可以指示元数据被存储在图像项的所有内容之后。作为另一示例,渐进信息可以指示元数据被存储在图像项的子图像的高级版本之前。

[0524] 可能地,图像项的成分自身可以由若干子图像组成。例如,叠加项可以使用网格项作为其成分之一。在这种情况下,与组成的子图像相关联的渐进信息定义该子图像的不同版本。

[0525] 实体组的属性

[0526] 可能地,描述渐进步阶的项属性可以与实体组相关联,该实体组指示在该组中声明的实体(例如项)可以用于渐进绘制(例如,“prgr”或“altr”实体组)。例如,在图10中,“plai”项属性1060可以与“prgr”实体组1000(而不是如图所示的图像1020)相关联。作为另一示例,在图11中,“prdi”项属性1125可以与“prgr”实体组1100(而不是如图所示的网格图像1120)相关联。

[0527] 可能地,描述与实体组相关联的渐进步阶的项属性用信号通知在实体组中包含的最高质量项的渐进步阶。在“prgr”实体组的情况下,这是该实体组的最后项。例如,在图11中,如果“prdi”项属性1125与“prgr”实体组1100相关联,则可以描述该“prgr”实体组中包含的最后项(即,网格图像1120)的渐进步阶。

[0528] 可能地,描述与实体组相关联的渐进步阶的项属性用信号通知其可应用的实体组中包含的任何项的渐进步阶。例如,如果项属性是如在第一实施例中描述的“ppat”项属性,则可以应用于相关联的实体组中包含的任何导出项。

[0529] 可能地,描述与实体组相关联的渐进步阶的项属性可以指示其应用于实体组的哪个项。该指示可以是项属性应用于的项的item_ID,或者在实体组所参考的实体的列表内的该项的索引(即,在实体组中的entity_ids的列表中的等于该项的item_ID的entity_id的索引),或者任何其他指示。

[0530] 可能地,描述与实体组相关联的渐进步阶的项属性可以描述该实体组中包含的若干项的渐进步阶。该属性可以具有与第四实施例的第四变型的“prif”项属性类似的结构:

```
aligned(8) class ProgressiveGroupInformationProperty
extends ItemFullProperty('prgi', version = 0, flags = 0) {
    // Number of progressive steps.
    unsigned int(8) step_count;
    for (i=0; i < step_count; i++) {
        unsigned int(8) step_item_index;
        unsigned int(8) layer_index;
        unsigned int(8) item_count;
        for (j=0; j < item_count; j++) {
            unsigned int(16) item_index;
            unsigned int(8) step_index;
        }
    }
}
```

[0531]

[0532] 在该结构中,step_count指示渐进信息项属性内所描述的渐进步阶的数量。该数

量对应于与该属性相关联的实体组(例如,“prgr”、“altr”实体组或任何其他类型的实体组)的重构步阶的数量。step_item_index针对各渐进步阶指示与该步阶相对应的实体组内的图像项的索引。该索引可以是基于0的。

[0533] 对于各渐进步阶,layer_index指示针对step_item_index字段标识的图像项的与渐进步阶相对应的层的索引。如果图像项不具有层,则layer_index的值是0。

[0534] 对于各渐进步阶,如果所标识的图像项是导出项,则item_count值指示该导出项的渐进步阶的精细化中所包含的子图像版本的数量。如果所标识的图像项不是导出项,则item_count的值是0。如果所标识的图像项是导出项,则具有相关联的“prif”项属性。

[0535] item_index和step_index字段与先前变型中针对“prdi”项属性所描述的字段类似。可以根据与由item_index字段标识的输入图像项相关联的“prif”项属性,直接重构渐进步阶的完整列表。因此,step_index值可以是到由与所标识的图像项相关联的“prif”项属性限定的渐进步阶列表中的索引。

[0536] 在该结构中,可以根据与包含输入图像的“prgr”实体组相关联的“prgi”项属性,直接重构由包含在相关联的“prgr”实体组中并且由item_index字段标识的导出项所使用的输入图像的渐进步阶的完整列表。

[0537] 在该结构中,如果实体组中包含的一些项对应于单个渐进步阶,则可以不描述这些项。例如,即使在渐进绘制期间可以使用缩略图,缩略图可能也不具有在该项属性中描述的相关联的渐进步阶。

[0538] 可能地,该项属性可以利用以下结构列出实体组中包含的各项的渐进步阶:

```
aligned(8) class ProgressiveGroupItemInformationProperty
extends ItemFullProperty('pgii', version = 0, flags = 0) {
    // Loop over the entities of the group.
    for (i=0; i < entity_count; i++) {
        // Number of progressive steps.
        unsigned int(8) step_count;
        for (j=0; j < step_count; j++) {
            unsigned int(8) layer_index;
            unsigned int(16) item_count;
            for (k=0; k < item_count; k++) {
                unsigned int(16) item_index;
                unsigned int(8) step_index;
            }
        }
    }
}
```

[0539]

[0540] 在该结构中,layer_index、item_count、item_index和step_index字段与“prgi”项属性的字段类似。entity_count值与相关联的“prgr”实体组中包含的实体的数量相对应。可能地,可以在“pgii”项属性的字段中明确地指定该值。

[0541] 在该结构中,依次针对相关联的“prgr”实体组中包含的各项描述渐进步阶。

[0542] 在该结构中,step_count字段指示正被描述的相关联的“prgr”实体组中包含的项的渐进组项信息属性内描述的渐进步阶的数量(即,与实体组中第i个实体相对应的项)。

[0543] 在该结构中,可以根据与包含输入图像的“prgr”实体组相关联的“pgii”项属性,直接重构由包含在相关联的“prgr”实体组中并且由item_index字段标识的导出项使用的输入图像的渐进步阶的完整列表。

[0544] 如果正描述的项具有与项自身的绘制相对应的单个渐进步阶,则step_count值可以被设置为0。

[0545] 可能地,与实体组相关联的项属性和与图像项相关联的项属性可以在同一HEIF文件中联合使用。例如,在图11中,“prdi”项属性1125可以与“prgr”实体组1100相关联,并且另一“prdi”项属性(未示出)可以与C0图像项1132相关联。可能地,用于描述与实体组相关联的渐进步阶的项属性可以与用于描述与图像项相关联的渐进步阶的项属性相同。例如,这两者可以都是“prdi”项属性。可能地,这些项属性可以具有不同类型或具有不同内容。例如,用于描述与实体组相关联的渐进步阶的项属性可以是“prgi”项属性,而用于描述与图像项相关联的渐进步阶的项属性可以是“prdi”项属性。

[0546] 将描述渐进步阶的项属性与实体组相关联的所有这些可能性也可以应用于不同于“prgr”实体组的实体组,诸如“altr”实体组。也可以使用描述渐进步阶的其他项属性。

[0547] 多次渐进绘制

[0548] 可能地,可以针对同一图像项描述若干渐进绘制。例如,在图9中,第二“prgr”实体组可以仅包括缩略图910和图像930。优选地,在HEIF文件内的图像数据的排序使得能够进行所描述的所有渐进绘制。例如,在图9中,如果第二“prgr”实体组包括缩略图910、第二预览和图像930,则与预览920相对应的数据和与第二预览相对应的数据这两者应在与缩略图910相对应的数据之后和与图像930相对应的数据之前存储。

[0549] 意图被绘制的图像项可以具有针对其描述的若干渐进绘制。例如,图像项可以被包含在若干“prgr”实体组中。作为另一实例,导出图像项可以与若干“prdi”项属性相关联。在该情况下,绘制器可以选择这些渐进绘制描述中的任一者(例如,“prgr”组,具有渐进步阶属性的“prgr”组,具有“prdi”的项……),以实现图像项的渐进绘制。

[0550] 可能地,要使用的渐进绘制描述的选择可以基于绘制器的特性。例如,被配置为节省CPU资源的绘制器可以选择具有最少绘制步阶的渐进绘制描述。

[0551] 可能地,要使用的渐进绘制描述的选择可以基于在HEIF文件中的该描述的位置。例如,如果图像项被包含在若干“prgr”实体组中,则绘制器可以选择第一“prgr”实体组。绘制器还可以选择最后“prgr”实体组。作为另一示例,绘制器可以选择具有最低group_id的“prgr”实体组。绘制器还可以选择具有最高group_id的“prgr”实体组。

[0552] 可能地,不同的渐进绘制描述可以包括用于帮助绘制器选择描述之一的信息。例如,渐进绘制描述可以包括与使用该渐进绘制描述时的意图的下载速度有关的信息。该信息可以被存储在针对基于项属性的渐进绘制描述的项属性(例如,“prdi”项属性或先前描述的任何其他项属性)内。该信息可以被存储在与针对基于实体组的渐进绘制描述的实体组相关联的项属性(例如,“prgr”实体组)内。

[0553] 可能地,不同的渐进绘制描述可以包括偏好相关信息。例如,渐进绘制描述可以包括指示其为优选渐进绘制描述的标志。该信息可以被存储在针对基于项属性的渐进绘制描述的项属性(例如,“prdi”项属性或先前描述的任何其他项属性)内,或者当主要项具有渐进绘制描述时被存储在文件顶层的渐进下载信息框中,可能地,具有设置为零的initial_

delay参数,以用于绘制器开始无延迟地绘制所选图像的版本。该信息可以被存储在与针对基于实体组的渐进绘制描述的实体组(例如,“prgr”实体组)相关联的项属性内。该信息可以例如使用不同的分组类型或者使用不同的版本或标志值被直接存储在针对渐进绘制描述的实体组内。例如,“prgr”实体组可以用于指示优选的渐进绘制,而“prg2”实体组可以用于指示可替代渐进绘制。作为另一示例,可以在“altr”实体组中声明若干替代“prgr”实体组。

[0554] 可能地,图像项可以对应于渐进绘制描述的主图像和另一渐进绘制描述的绘制步阶。在这种情况下,为了渐进地绘制该图像项,绘制器可以选择渐进绘制描述,其中在该渐进绘制描述中,图像项是该描述的主图像。例如,如果第一“prgr”实体组包含缩略图和LDR(低动态范围,与高动态范围相反)图像,并且第二“prgr”实体组包含另一缩略图、LDR图像和HDR图像,则当渐进地绘制LDR图像时,绘制器可以选择第一“prgr”实体组。

[0555] 可能地,绘制器可以组合与不同的渐进绘制描述相对应的渐进绘制步阶。例如,如果第一“prgr”实体组包含第一缩略图、第一预览和图像项,并且第二“prgr”实体组包含第二缩略图、第二预览和图像项,则绘制器可以通过首先显示第一缩略图、然后显示第二预览并最终显示图像项来渐进地绘制图像项。

[0556] 可能地,在选择渐进绘制描述之前,绘制器可以检查这些描述是否与HEIF文件内的图像数据的排序兼容。绘制器可以仅在渐进绘制描述与HEIF文件内的图像数据的排序兼容的情况下选择该渐进绘制描述。

[0557] 可能地,与不同图像项相对应的两个或多于两个渐进绘制描述可以包含共同的初始步阶。例如,包含一对立体图像项的HEIF文件可以包括在其自身的“prgr”实体组中的各图像项,并且对于这两个图像项使用相同的缩略图作为第一绘制步阶,这是因为这些图像项仅具有在缩略图上可能不可见的微小差异。

[0558] 可能地,不是所有的初始步阶对所有不同的图像项都是共同的。例如,与两个立体图像项的第一绘制步阶相对应的缩略图可以不同,而与第二绘制步阶相对应的预览可以相同。

[0559] 可能地,这些图像项的渐进绘制可以利用共同的初始步阶。

[0560] 可能地,当在若干渐进绘制描述中绘制作为中间步阶存在的图像项时,绘制器可以如上所述选择这些绘制描述之一。

[0561] 当使用不同的渐进绘制描述时,图5的步骤510和520被修改以获得多个渐进规范并且生成多个渐进信息。

[0562] 当使用不同的渐进绘制描述时,图6的步骤620被修改以提取多个渐进信息并且选择信息之一。

[0563] 注意,参照第四实施例的变型所描述的所有这些可能性也可以在适用的情况下应用于其他变型或其他实施例。

[0564] 实体组的渐进绘制

[0565] 可能地,渐进绘制信息可以与实体组相关联。例如,缩略图可以用作图像项的幻灯片、集合或相册的临时替换。该缩略图可以不同于幻灯片、集合或相册的第一图像的缩略图。

[0566] 可能地,“prgr”实体组可以用于指示其他项或实体组可以用作绘制实体组的临时

替换。例如,“prgr”实体组可以包含缩略图和幻灯片实体组,以指示缩略图可以用于幻灯片实体组的渐进绘制。作为另一示例,“prgr”实体组可以包含缩略图、第一幻灯片实体组和第二幻灯片实体组。第一幻灯片实体组可以包含第二幻灯片实体组中包含的图像项的预览。可以通过首先绘制缩略图、然后绘制第一幻灯片实体组并且最后绘制第二幻灯片实体组来实现第二幻灯片实体组的渐进绘制。作为第三示例,“prgr”实体组可以包含缩略图和幻灯片实体组。此外,在幻灯片实体组中包含的各图像项在其缩略图旁边也被包含在“prgr”实体组中。幻灯片实体组的渐进绘制可以通过首先绘制其缩略图然后绘制幻灯片自身来实现。例如当幻灯片中的下一图像尚未被播放器接收时,作为绘制幻灯片的一部分,幻灯片的各图像项可以通过在显示图像项自身之前首先显示其缩略图来渐进地绘制。代替冻结,用户可以通过针对该下一图像应用重构步阶来看到该下一图像渐进地出现。

[0567] 可能地,特定项属性可以用于描述与实体组相关联的渐进绘制步阶。

[0568] 注意,参照第四实施例的变型所描述的所有这些可能性也可以在适用的情况下应用于其他变型或其他实施例。

[0569] 可能地,即使图像项的全部数据已经可用于实现图像项的快速绘制,也可以使用对图像项的渐进绘制。例如,为了快速回顾幻灯片的所有图像,可以代替图像自身来绘制图像的初始渐进绘制步阶。这可以由用户用来快速检查他的幻灯片或实现幻灯片的一些快进。

[0570] 图7示出根据本发明的实施例生成的HEIF文件的渐进绘制的主要步骤。图7是图6所示主要步骤的更一般的替代。要注意的是,参照图像项描述这些步骤,但是这些步骤还可以用于轨的渐进绘制,尤其用于时间信息不一定有意义的轨(例如,“pict”轨或封装在轨内的一系列独立静止图像)。

[0571] 在第一步骤700中,选择要绘制的图像项。

[0572] 然后,在步骤710至735中,确定用于渐进精化的潜在实体。首先,将图像项自身视为渐进精化的潜在实体。

[0573] 在步骤710处,检查所选择的图像项是否被包括在类型“altr”的实体组中。如果是这种情况,则在步骤715处,在该实体组中包含的图像项被视为渐进精化的潜在实体。下一步骤是步骤730。

[0574] 否则,在步骤720处,检查所选择的图像项是否具有相关联的缩略图图像项。如果是这种情况,则在步骤725处,相关联的缩略图被认为是用于渐进精化的潜在实体。下一步骤是步骤730。

[0575] 在步骤730处,检查所选择的图像项是否被编码为具有若干层的多层图像。如果是这种情况,则在步骤735处,不同的编码层被认为是用于渐进精化的潜在实体。如果渐进层属性与所选择的图像项相关联,则仅该项属性中指示的层被认为是用于渐进精化的潜在实体。

[0576] 在步骤740处,获得下一潜在实体。例如,监视数据的加载以检查何时加载了与新项或新层相对应的数据。如果所加载的数据使得能够获得被视为潜在实体的新图像项,则该图像项是下一潜在实体。如果所加载的数据使得能够获得被视为潜在实体的新层,则该层是下一潜在实体。如果所加载的数据使得能够获得作为潜在实体的子图像的新图像项,则该潜在实体是下一潜在实体。否则,监视继续。

[0577] 然后在步骤750处,检查是否已经绘制任何实体。如果没有实体被绘制,则在步骤755处,绘制下一潜在实体。下一步骤是步骤790。

[0578] 否则,在步骤760处,检查在步骤710处是否发现“altr”组,并且如果是,则下一潜在实体在该“altr”组中是否比当前绘制的实体更早。如果是,则在步骤765处,绘制下一潜在实体。下一步骤是步骤790。

[0579] 否则,在步骤770处,检查当前绘制的实体是否是缩略图并且下一潜在实体是否是其主图像。如果是,则在步骤775处,绘制下一潜在实体。下一步骤是步骤790。

[0580] 否则,在步骤780,检查下一潜在实体是否与当前绘制的实体相同并且具有与其相关联的渐进步阶。如果这是真的,则进一步检查是否可以绘制新的渐进步阶。如果这是真的,则在步骤785,显示这个新的渐进步阶。下一步骤是步骤790。

[0581] 在步骤790处,检查是否已到达HEIF文件的结束。如果是这种情况,则不再执行步骤。否则,下一步骤是步骤740。

[0582] “altr”分组类型可以用于与图像项相关联,其他图像项可用于该图像项的渐进绘制。本发明的实施例的一些变型依赖于用于指示子图像的所有版本的“altr”实体组。其他分组类型可以用于指示哪些图像项可以用于给定图像项的渐进绘制。特定的分组类型(例如,“prog”)可以用于鉴于渐进绘制而使图像项相关联。

[0583] 可能地,渐进精化模式项属性可以与任何图像项相关联,以用于指示针对该图像项所选择的渐进精化策略。

[0584] 可能地,渐进精化项属性可以与图像项相关联,以用于指示该图像项支持渐进绘制。该渐进精化项属性的结构可以如下:

[0585] aligned(8)class ProgressiveProperty

[0586] extends ItemFullProperty('prog',version=0,flags=0){

[0587] }

[0588] 可能地,一些渐进信息可以与实体组相关联,以用于指示实体组支持渐进绘制。例如,左右渐进绘制模式可以与全景组相关联,以用于指示全景自身可以以渐进方式绘制。

[0589] 图8是用于实现本发明的一个或多个实施例的计算装置800的示意性框图。计算装置800可以是诸如微计算机、工作站或轻型便携式装置等的装置。计算装置800包括通信总线,该通信总线连接到:

[0590] -标记为CPU的中央处理单元801,诸如微处理器等;

[0591] -标记为RAM的随机存取存储器802,用于存储本发明实施例的方法的可执行代码以及适于记录变量和参数的寄存器,该变量和参数是用于实现根据本发明的实施例的方法所必需的,该随机存取存储器的存储器容量可以例如通过连接到扩展端口的可选RAM来扩展;

[0592] -标记为ROM的只读存储器803,用于存储用于实施本发明实施例的计算机程序;

[0593] -标记为NET的网络接口804,其通常连接到通信网络,通过该通信网络来发送或接收要处理的数字数据。网络接口804可以是单个网络接口,或者由不同的网络接口(例如,有线和无线接口,或者不同种类的有线或无线接口)的集合构成。在CPU 801中所运行的软件应用的控制下,将数据包写入网络接口以用于发送,或者从网络接口读取数据以用于接收;

[0594] -图形用户界面805可以用于从用户接收输入或向用户显示信息;

[0595] -标记为HD的硬盘806可以设置为大存储装置;

[0596] -I/O模块807可以用于从/向外部装置(诸如,视频源或显示器等)接收/发送数据。

[0597] 可执行代码可以存储在只读存储器803中,可以存储在硬盘806上,或者可以存储在可移动数字介质(例如,盘)上。根据变型,可以借助于通信网络经由网络接口804来接收程序的可执行代码,以使程序的可执行代码在被执行之前存储在通信装置800的存储部件其中之一(诸如,硬盘806等)中。

[0598] 中央处理单元801适于控制和指导根据本发明实施例的一个或多个程序的指令或部分软件代码的执行,这些指令存储在前述存储部件其中之一中。在接通电源之后,CPU 801能够在例如从程序ROM 803或硬盘(HD)806加载了与软件应用有关的来自主RAM存储器802的指令之后执行这些指令。这样的软件应用在由CPU 801执行时,使得进行本发明的流程图的步骤。

[0599] 本发明的算法的任何步骤可以通过由可编程计算机器(诸如PC(“个人计算机”)、DSP(“数字信号处理器”)或微控制器等)执行指令集或程序而在软件中实现;或者通过机器或专用组件(诸如FPGA(“现场可编程门阵列”)或ASIC(“专用集成电路”)等)在硬件中实现。

[0600] 尽管以上已经参考具体实施例描述了本发明,但是本发明不限于具体实施例,并且对本领域技术人员来说,在本发明的范围内的修改将是明显的。

[0601] 在参考前述说明性实施例时本领域技术人员将想到许多进一步的修改和变形建议,这些实施例仅作为示例给出并且不意在限制本发明的范围,该范围仅由所附权利要求确定。特别地,在适当的情况下,可以互换来自不同实施例的不同特征。

[0602] 上述本发明的各实施例可以单独实现或实现为多个实施例的组合。此外,来自不同实施例的特征可以在必要时组合或者来自单独实施例的元素或特征的组合在单个实施例中是有益的情况下进行组合。

[0603] 在权利要求中,“包括”一词不排除其他元素或步骤,并且不定冠词“a”或“an”不排除复数。仅仅在相互不同的从属权利要求中限定不同特征的这一事实并不指示不能有利地使用这些特征的组合。

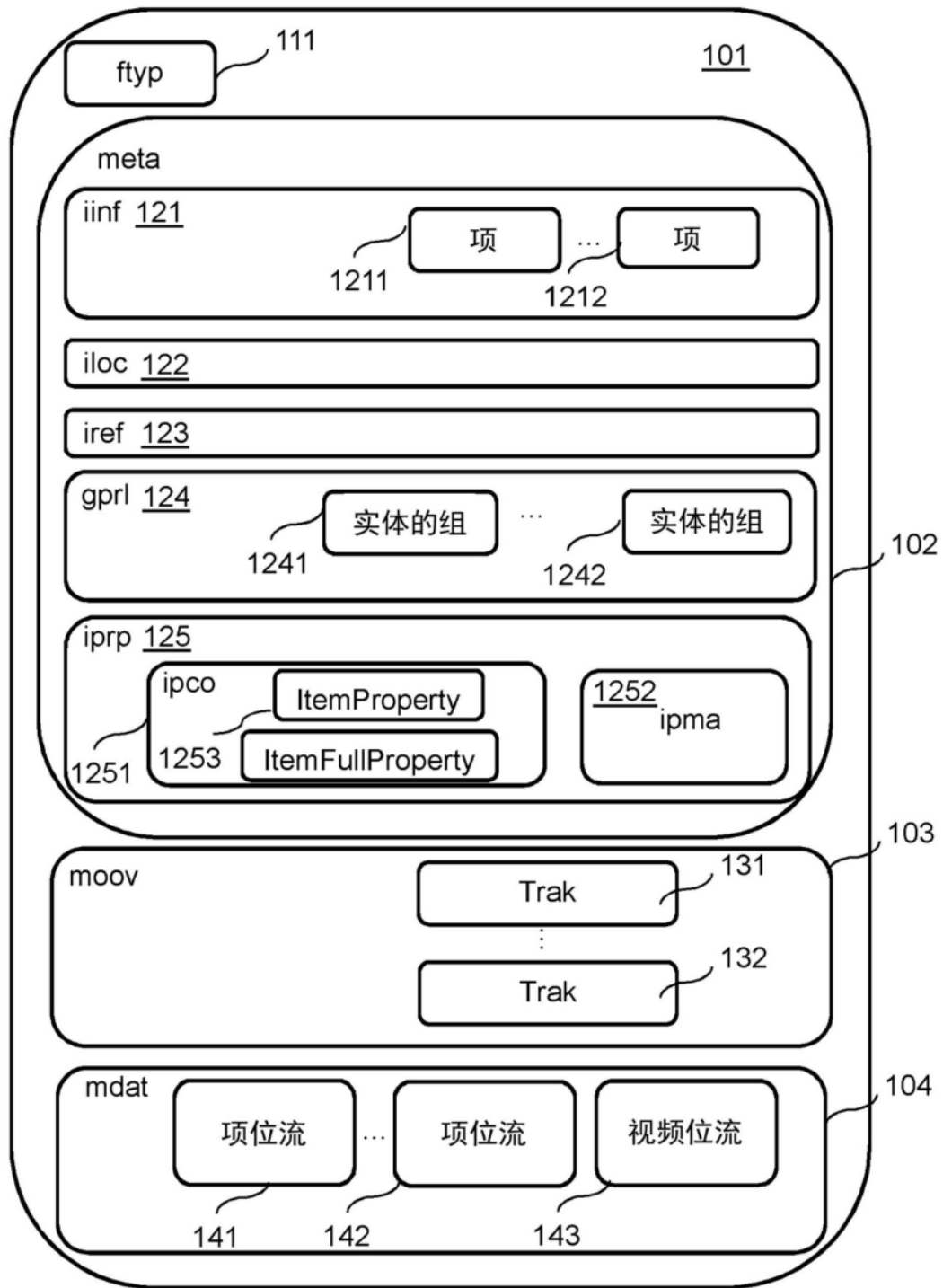


图1

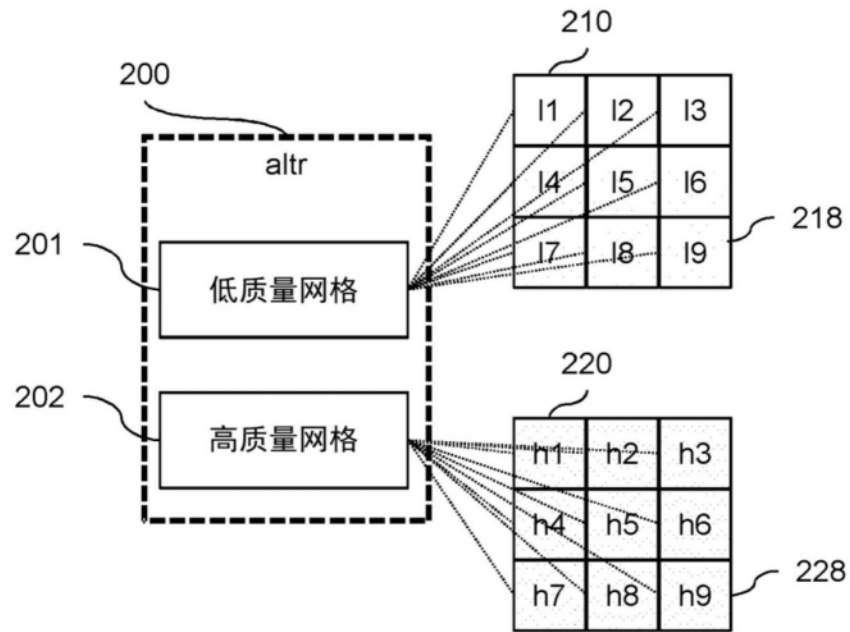


图2

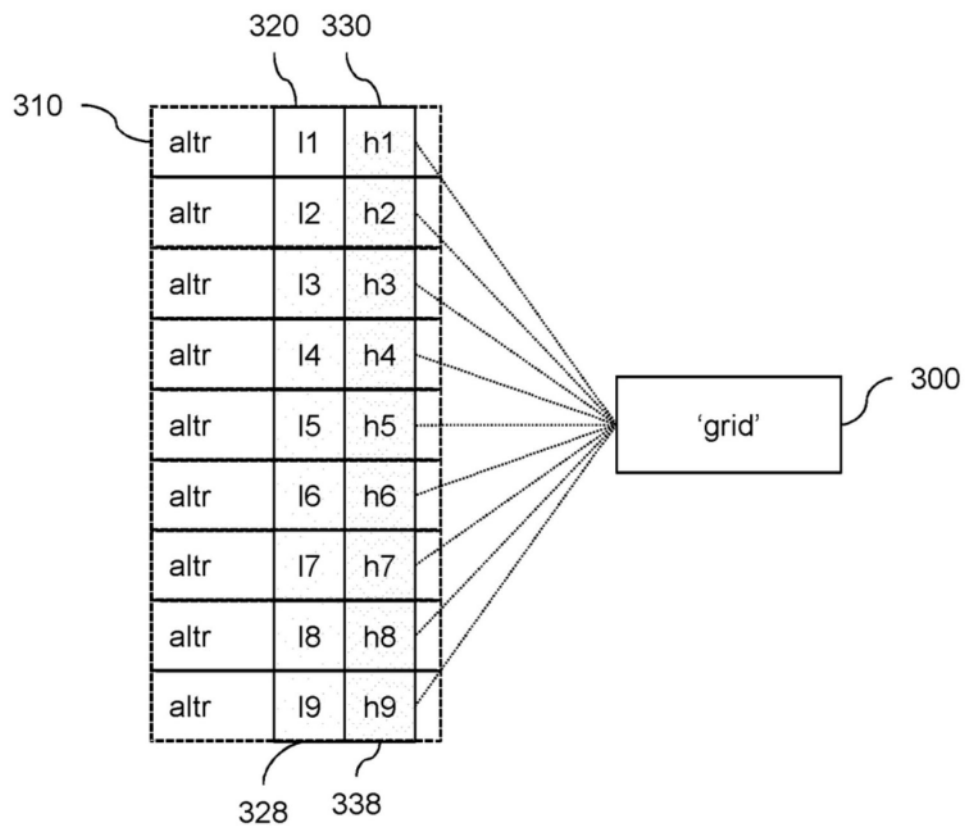


图3

...

l3	h3	l4	h4	l5	h5	l6	h6	l7	h7	l8	h8	l9	h9
----	----	----	----	----	----	----	----	----	----	----	----	----	----

图4a

...

l5	l6	l7	l8	l9	h1	h2	h3	h4	h5	h6	h7	h8	h9
----	----	----	----	----	----	----	----	----	----	----	----	----	----

图4b

...

l5	l6	l7	l8	l9	h1	h4	h7	h2	h5	h8	h3	h6	h9
----	----	----	----	----	----	----	----	----	----	----	----	----	----

图4c

...	l5	l6	l7	l8	l9	h1	h2	h3	h4	h5	h6	h7	h8	h9
-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

图4d

...	l5	l6	l7	l8	l9	h1	h4	h7	h2	h5	h8	h3	h6	h9
-----	----	----	----	----	----	----	----	----	----	----	----	----	----	----

图4e

h1	h2	h3
l4	l5	l6
l7	l8	l9

图4f

h1	l2	l3
h4	l5	l6
h7	l8	l9

图4g

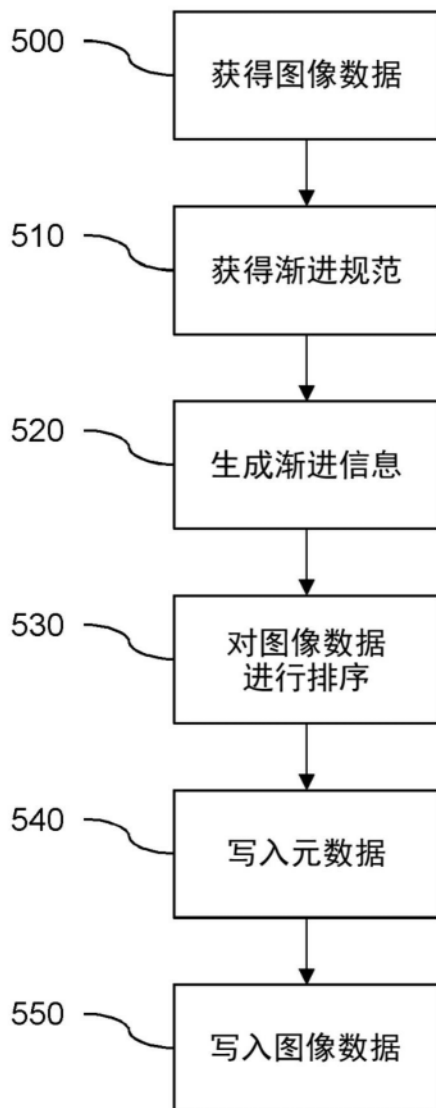


图5

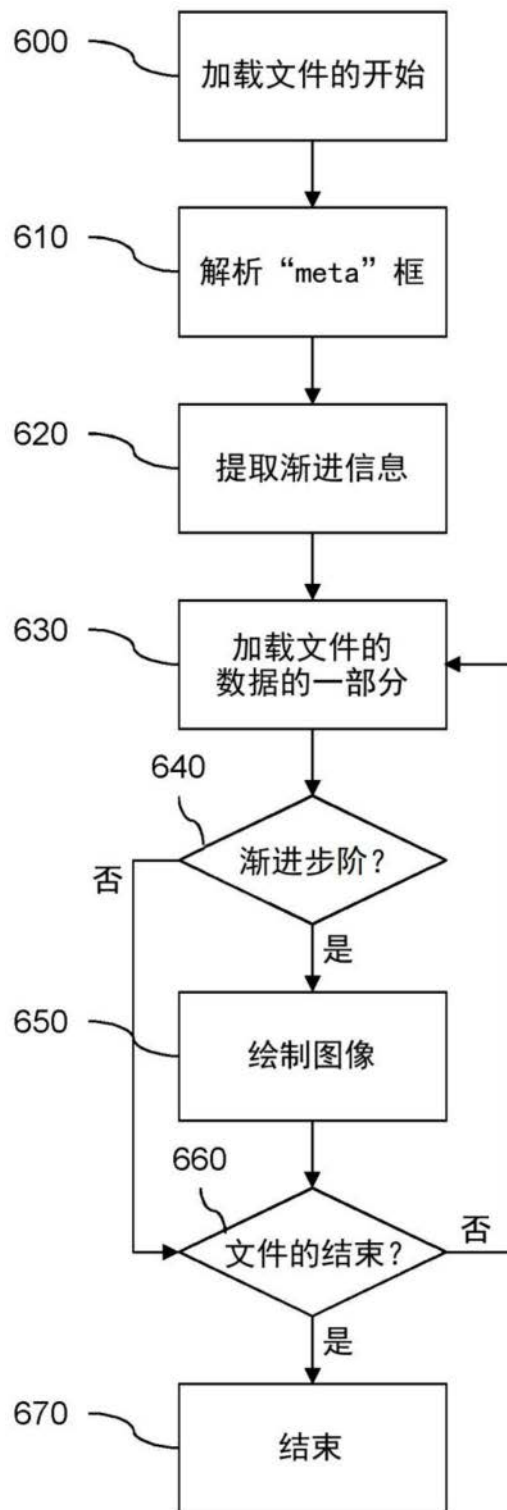


图6

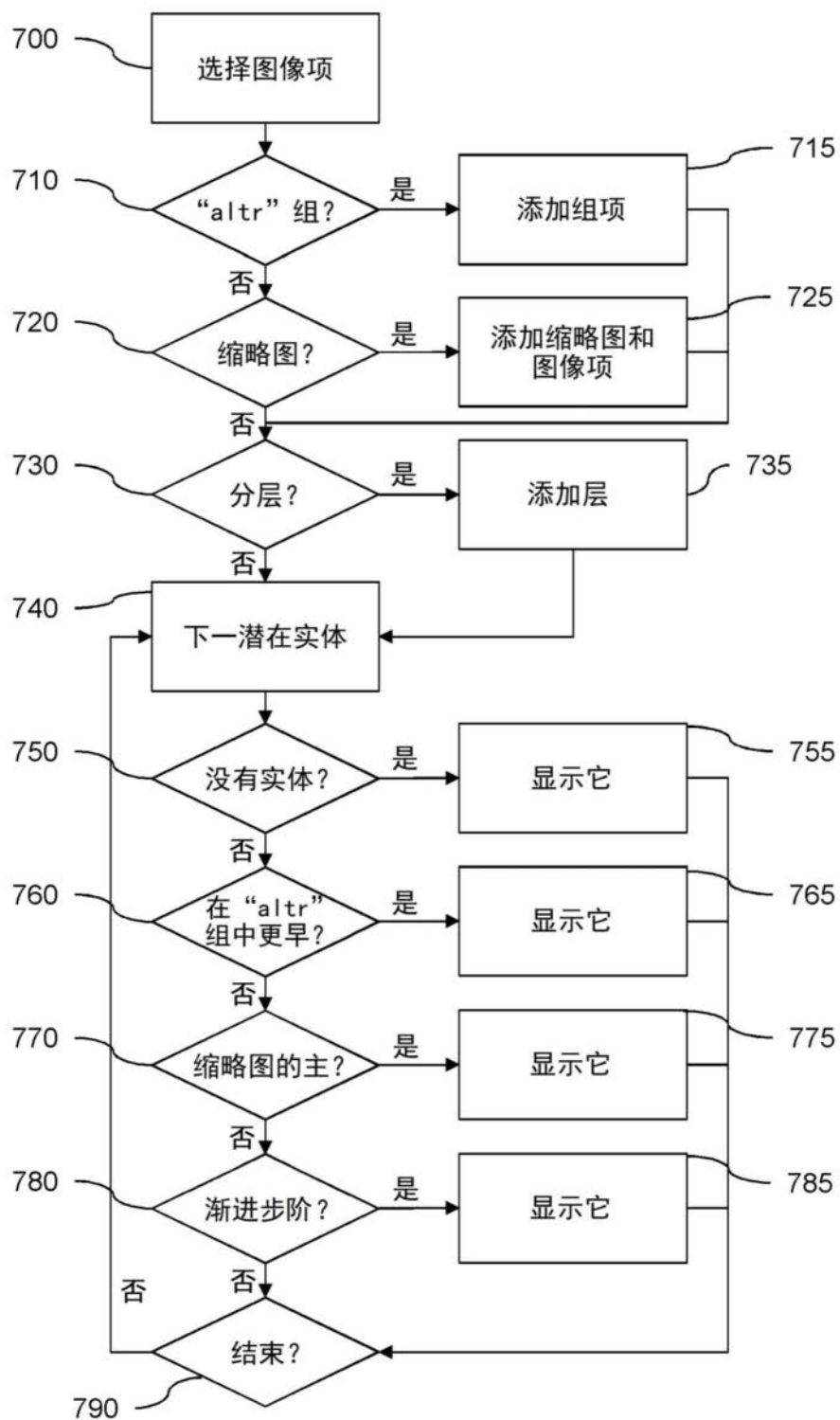


图7

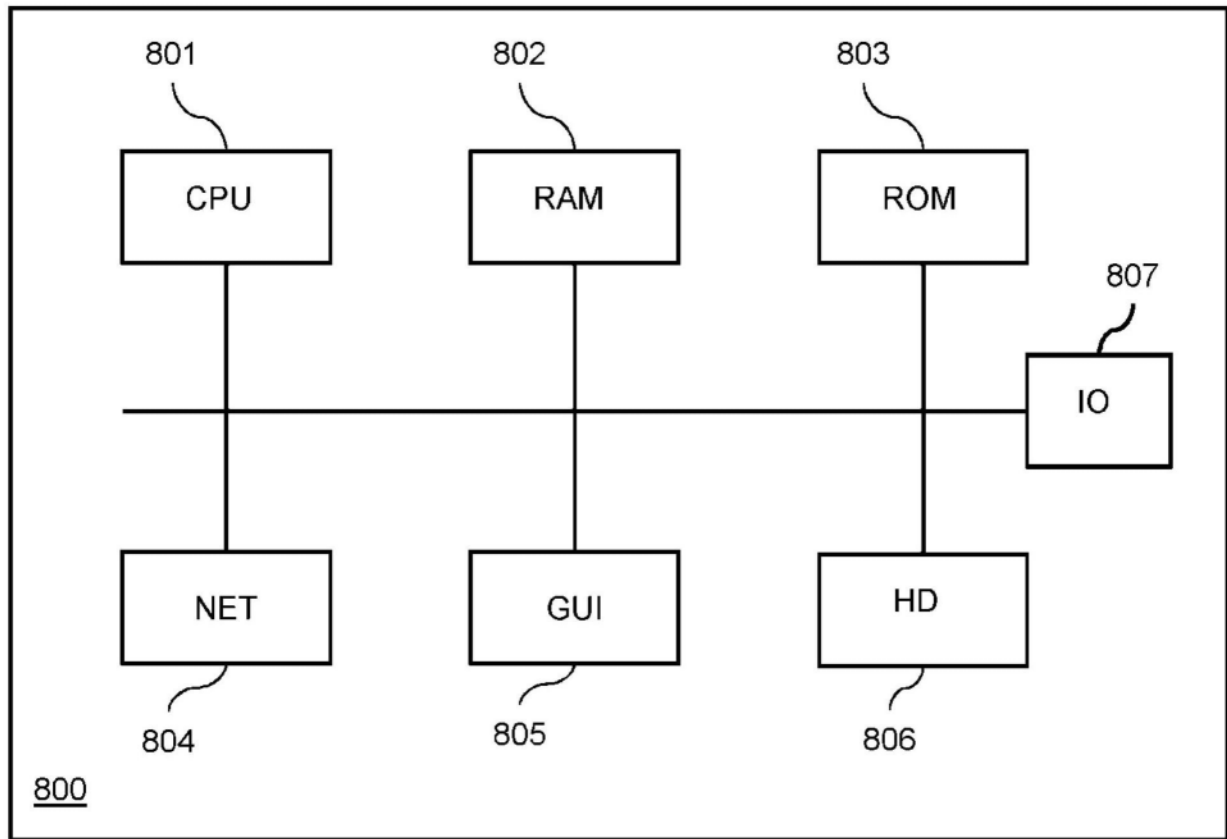


图8

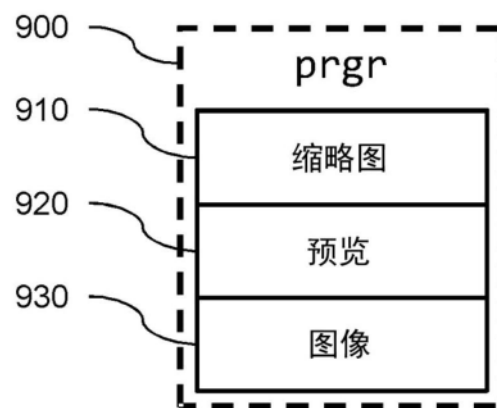


图9

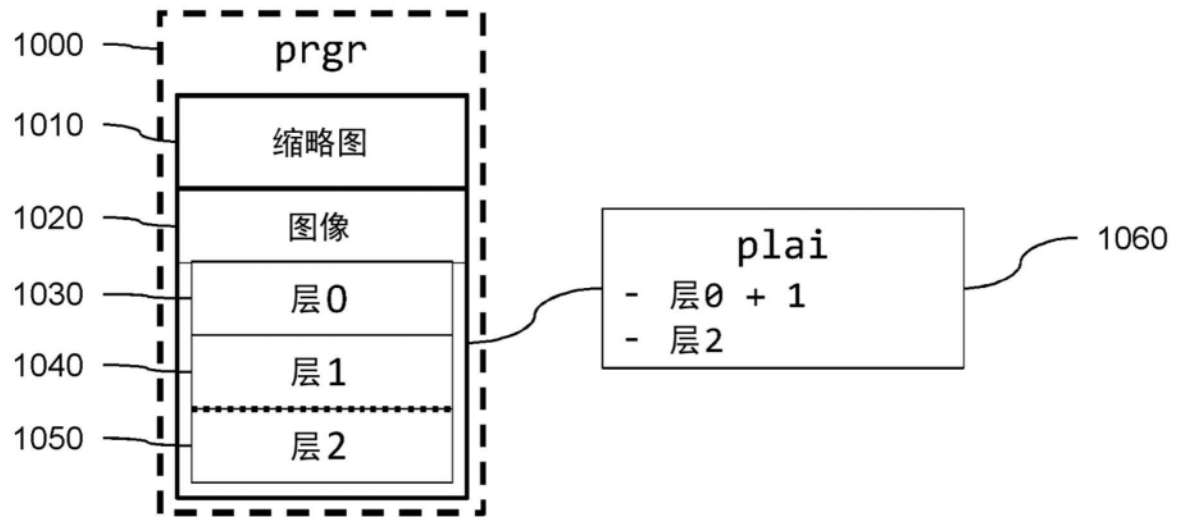


图10

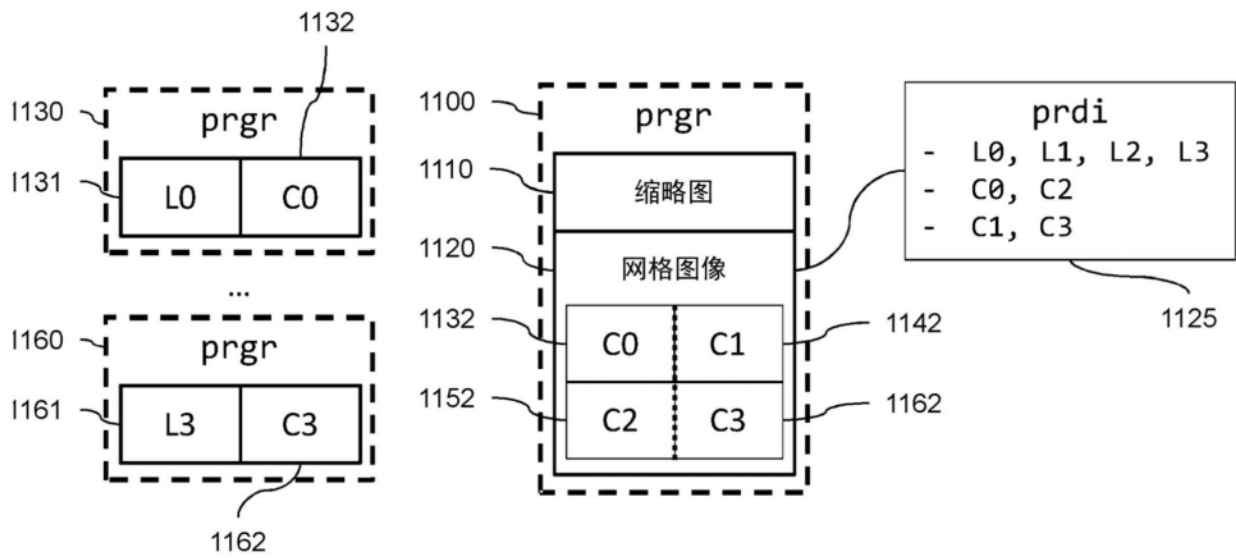


图11

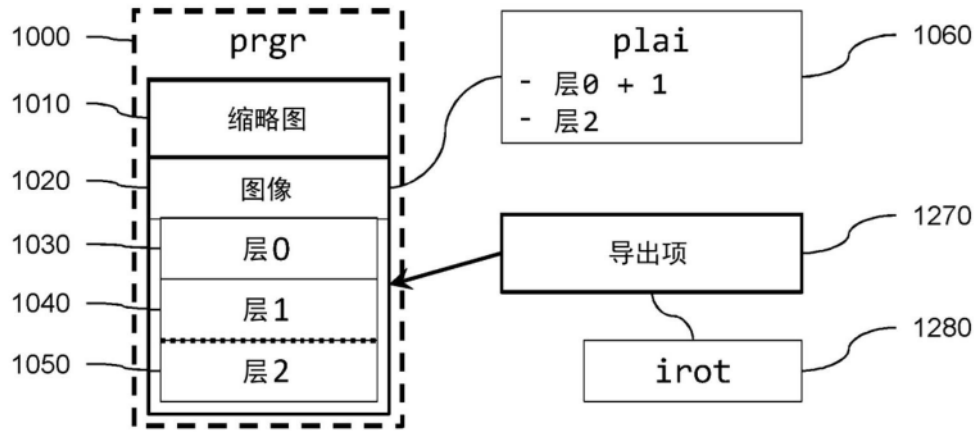


图12