

[12] 发明专利申请公开说明书

[21] 申请号 99812372.2

[43]公开日 2001年12月12日

[11]公开号 CN 1326637A

[22]申请日 1999.10.13 [21]申请号 99812372.2

[30]优先权

[32]1998.10.21 [33]SE [31]9803593-4

[86]国际申请 PCT/SE99/01847 1999.10.13

[87]国际公布 WO00/24188 英 2000.4.27

[85]进入国家阶段日期 2001.4.20

[71]申请人 艾利森电话股份有限公司

地址 瑞典斯德哥尔摩

[72]发明人 M·拉松 C·克里托普鲁斯

M·让德尔 D·桑塔克鲁茨

T·埃布拉希米

[74]专利代理机构 中国专利代理(香港)有限公司

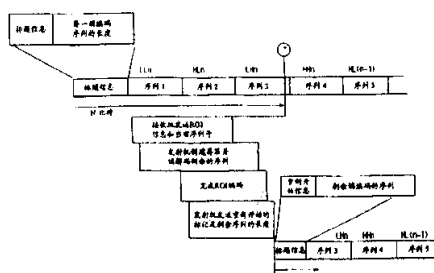
代理人 陈景峻

权利要求书 2 页 说明书 11 页 附图页数 6 页

[54]发明名称 在压缩域中的图像局部检索方法和系统

[57]摘要

用于在客户-服务器系统中检索图像的方法和系统,图像被存储在该服务器上的一组独立解码的单元(CU)中。一个客户发出一系列针对图像信息的请求。每一请求包含请求号码、有关客户随后想要看的图像信息种类的信息和有关当请求发出时客户已接收的图像信息种类的信息。该服务器不必存储任何状态信息(例如先前请求)。接收一个请求时,服务器发送一个再起动物标记、一个请求号码的确认以及对应于该请求的增加的图像信息。在此描述的方法和系统的使用将导致在该服务器中不需要对整个比特数据流进行解码。由于不需要执行该数据流的全解码,所以这将在发送一侧(服务器方)节省许多时间。





权 利 要 求 书

1.在服务器和客户之间发送被存储为若干可独立地解码的编码单元的一个图像的方法,其特征在于包括以下步骤:

- 5
- 从该客户发送一个针对图像数据的请求到该服务器,
 - 从该服务器开始发送该请求的图像数据到该客户,
 - 在发送过程中或之后发送针对该图像的一个新部分的请求,以及
 - 仅使用还未发送的编码单元把该图像的该请求的新部分从服务器发送到客户。

10 2.根据权利要求 1 的方法,其特征在于,该图像被存储在该变换域中。

3.根据权利要求 1-2 任何之一的方法,其特征在于,来自该客户的每一请求包括一个请求号码。

4.根据权利要求 1-3 任何之一的方法,其特征在于,来自该客户的每一请求包括关于该客户感兴趣的图像的信息以及关于该客户已经接入的图像的信息。

15 5.根据权利要求 1-4 任何之一的方法,其特征在于,在把请求信息发送到该客户之后,该服务器直接放弃用客户提供的全部信息。

6.根据权利要求 1-5 任何之一的方法,其特征在于,在发送请求的编码单元之前,该服务器发送一个标记码。

20 7.根据权利要求 1-6 任何之一的方法,其特征在于,在发送该新图像部分之前,该服务器执行一个代码转换。

8.一个客户-服务器系统,其中在该服务器中的图像被存为若干编码单元,特征是:

- 25
- 在服务器中用于从客户接收针对图像数据的一个请求的装置,
 - 用于从该服务器发送该请求的图像数据到该客户的装置,
 - 在该服务器中用于在发送期间或之后接收针对该图像一个新部分的请求的装置,以及
 - 在该服务器中仅使用未被发送的编码单元用于把该请求的该图像的新部分从服务器发送到客户的装置。

30 9.根据权利要求 8 的一个系统,特征在于用于把该图像存储在该变换域中的装置。

10. 根据权利要求 8-9 任何之一的系统,其特征在於, 来自该客户的每一请求包括一个请求号码。

5 11. 根据权利要求 8-10 任何之一的系统,其特征在於, 来自该客户的每一请求包括关于该客户感兴趣的图像的信息以及关于该客户已经接入的图像的信息。

12. 根据权利要求 8-11 任何之一的系统,其特征在於, 在客户提供的信息已经被处理之后, 该服务器被用于直接放弃由客户提供的全部信息。

13. 根据权利要求 8-12 任何之一的系统,其特征在於, 在发送请求的编码单元之前, 该服务器被用于发送一个标记码。

10 14. 根据权利要求 8-13 任何之一的系统,其特征在於, 在发送该新图像部分之前, 该服务器被用于执行尚未发送的该编码单元的一个代码转换。

在压缩域中的图像局部检索方法和系统

5 技术领域

本发明涉及用于在客户与服务器之间发送图像的方法和系统。本方法和系统尤其适合于对存储在服务器中的图像的局部发送。

发明背景和已有技术

10 数字图像能够存储在服务器中并在电信网络上分配。图像还可以存储在用于例如一个 CD-ROM 的物理介质上。如果是大规模的图像,则发送图像将可能要占一段时间。为了加速该发送,客户可以选择更重要的一个图像区域,并因此仅接收该区域。此方式被称为感性趣区域(ROI)编码。

在当今的客户-服务器系统中,如果客户要从服务器存取一个图像,则要在客户方的用户程序和该服务器之间建立一个连接。用户程序随即发送请求以便得到期望的图像。服务器通过检出该文件作出反应并开始发送。用户程序能够在任何

15 任何时候把一个新请求发送到该服务器,例如发送针对文件一部分的一个请求。当发送结束时,该连接关闭。这样的系统在 Fielding 等人的“超文本传送协议 HTTP/1.1 05 版”(HTTP 工作组,互联网络-草案)1998 年 9 月 11 日(工作仍在进行中)中有说明。

20 但是问题在于大部分静止图像压缩技术,例如 JPEG,产生的比特数据流是不能分割的编码单元。因此,如果选择一个感性趣区域,可独立解码单元的缺乏将迫使服务器执行一个全部解码,随后是整个比特数据流的新编码。依据该服务器软件,有时甚至必须从该存储介质再装入该图像。此方式的缺点是耗时和需要复杂的计算方案,故对于服务器的计算能力有很高要求。

25 此外,新出现的静止图像标准 JPEG2000 使用了所谓编码单元(CU)的独立熵编码。一个编码单元可以是例如在该变换域中的一个子频带(即在子波变换的情况下)、可以是一个子频带的一部分,例如确定尺寸(实例 16×16)的比特平面或数据块、或可以是针对一个子频带之内的一个区域的比特平面。

发明概要

30 本发明的一个目的是克服上面描述的问题,尤其是当客户请求一个 ROI 时,

即当请求图像的一部分时,减少在服务器中的处理和编码量。

此目的和其它目的是通过把图像作为可独立解码的单元(CU)组存储在一个服务器上而达到的。当客户仅请求图像的一确定部分时,从尚未发送的CU来的信息被随即重新编码,这将在服务器中节省许多处理时间。

5 例如,一个客户正在发出一系列针对图像信息的请求。每一请求包含请求号码、有关客户随后想要看的图像信息种类的信息和有关当请求发出时客户已接收的图像信息种类的信息。该服务器不必存储任何状态信息(例如先前请求)。接收一个请求时,服务器发送一个再起动标记、一个请求号码的确认以及对应于该请求的增加的图像信息。

10 在此描述的方法和系统的使用将导致在该服务器中不需要对整个比特数据流进行解码。由于不需要执行该数据流的全解码,所以这将在发送一侧(服务器方)节省许多时间。

附图的简要描述

现将参照附图更详细地描述本发明,其中:

15 图 1 示出在一个客户服务器处理中执行的基本步骤。

图 2 示出在根据第一实施例的客户服务器处理中执行的步骤。

图 3 示出在根据第二实施例的客户服务器处理中执行的步骤。

图 4 示出在根据第三实施例的客户服务器处理中执行的步骤。

20 图 5 示出在根据第三实施例的第一可选方案的客户服务器处理中执行的步骤。

图 6 示出根据第三实施例的第二可选方案的一个客户服务器处理中执行的步骤。

图 7 示出根据第三实施例的第三可选方案的一个客户服务器处理中执行的步骤。

25 图 8 示出在根据第四实施例的客户服务器处理中执行的步骤。

详细描述

图 1 中示出客户和服务器之间的一个基本交互作用。步骤 101 中,客户首先请求一个图像。下一步骤 103 中,服务器开始把图像发送到客户。步骤 105 中,客户随即在发送过程的一段时间中请求正在发送的图像的一部分。响应请求步
30 骤 105,服务器随即在步骤 107 开始发送该请求的部分。如在步骤 109 中表明,

客户还可以在什么时候请求图像的另外的部分。在一个较低协议等级中,比如 TCP 或 HTTP 中, 发送随时能够被中断。

在下面的描述和相应的附图中使用下列定义、句法和注释:

5 - 编码单元(CU)是可被独立解码的比特数据流的一个部分。因此,有可能解
码比特数据流的一个确定部分而不必解码整个数据流。

- TAG, 即重新同步标记是不能从该熵编码器产生的比特的一个组合。

在该客户服务器处理中的句法成分

一个基本客户服务器处理的几个详细实例在下面描述。它们对应于在该客
户服务器处理中的该句法成分的各种具体格式。

10 <Image data request>(图像数据请求)可以具有下列格式:

-<Image data request 1>=发送第一<数目>子频带

-<Image data request 2>=发送第一<数目>比特平面

-<Image data request 3>=发送该 ROI< ROI 数目><ROI 描述>的第一<数目>
子频带

15 -<Image data request 4>=发送该 ROI< ROI 数目><ROI 描述>的第一<数目>
比特平面

-<Image data request 5>=发送<CU 序列>(描述何种 CU 被请求)

<Image data acknowledge> (图像数据确认) 可以具有下列格式:

20 -<Image data acknowledge 1>=<图像数据确认标记>接收的比特/字节/CU<数
目>

-<Image data acknowledge 2>=<图像数据确认标记>接收的 CU<数目><CU
n1>...<CU nm>,其中<CU n>是指定到一个已收 CU 的数目

版本 1 与一个重新发送和数据包排序发送通信协议, 比如 TCP, 一起使用。
版本 2 与一个无接点最佳成果发送通信协议, 比如 UDP, 一起使用。

25 <Image data> (图像数据) 可以具有下列格式:

- <Image data 1> = 对应于其余子频带的第一<数目>的<CU 数据流>

- <Image data 2> = 对应于其余比特平面的第一<数目>的<CU 数据流>

- <Image data 3> = 对应于 ROI< ROI 号码>的第一<数目>的其余子频带的
<CU 数据流>

30 - <Image data 4> = 对应于 ROI< ROI 号码>的第一其余<数目>比特平面的

<CU 数据流>

- <Image data 5> = 对应于在<Image data request 5>中规定的该 CU 的<CU 数据流>

<CU stream> (CU 数据流) 的几个可能的不同格式:

5 -<CU stream 1> = <标题> <长度 CU1> <长度 CU2> ... <长度 CUN> <CU1>
<CU2> ... <CUN>

-<CU stream 2> = <标题> <长度 CU1> <CU1> <长度 CU2> <CU2>... <长度 CUN><CUN>

-<CU stream 3> = <标题> <Tag 1> <CU1> <Tag2>< CU2>... <Tag N><CUN>

10 -<CU stream 4> = <标题> <CU1> <Tag1> <CU2> <Tag 2>... <CUN> <Tag N>

如果图像标题已经由该客户接收, 则该图像标题是不需要的。

<Image transcoding header> (图像代码转换标题) 包含一种代码转换操作的描述, 即为了产生代码转换图像而对于原始图像执行的代码转换的操作的描述。客户使用此信息把 CU 从原始图像映射到代码转换图像, 以及把 CU 从代
15 码转换图像映射到原始图像。

<image posting notification> (图像传递提示) 被用于告诉客户该代码转换的图像存储在该服务器。 它给出该代码转换的图像的 URL。可选择包括一个时限, 指示该图像将仅被保存一个规定的时间。

20 图 2 示出的是本发明第一实施例。在此情况中该客户想要存取存储图像的一部分。此图像能够是确定的比特平面、子频带即预先存储的感性趣区域。除了在通信协议(例如 HTTP)中使用的操作之外, 在此情况中的该服务器不使用任何特殊的功能来响应该请求。 针对此实施例的处理在图 2 中示出。

步骤 201 中,客户发送针对存储图像的请求。该图像被作为压缩比特数据流并且以包括独立编码的编码单元(CU)的格式存储。 步骤 203 中,服务器响应和
25 开始该发送。步骤 205 中,客户决定更重要的一个图像部分。客户从图像标题得到其被存储的所在之处的信息,并且发送针对该期望部分的请求到服务器。该部分能够是一个 ROI 或图像数据,其增加整个图像的质量。步骤 207 中,服务器可以发送一个重新起始标记, 并且开始发送所需要的 CU。

30 该重新起始标记的功能还可以通过堆栈中较低的通信协议提供,例如由 TCP 或 HTTP 通信协议提供。如果能够从该较低协议等级中的信息获得, 该重新起

始标记能够被可选择地排除。

图 3 示出另一类型的客户服务器处理。在图 3 所示的处理中,在服务器侧不执行进一步的处理。在结合图 3 描述的实例中,客户开始接收一个图像,然后在
5 该发送中决定该图像的某些部分更重要,并且仅希望发送该部分图像。此图像
可以是若干比特平面、子频带或感性趣区域。应该注意,在此情况中的该图像
能够以一个感性趣区域(ROI)存储,但用户要选择另一 ROI。

在此情况中,服务器必须使用某些附加功能,以便找到该压缩图像的请求
部分。客户和服务器之间的交互作用以及在每一侧的操作如下所述,为了清楚起
见,没示出确认信息以及类似信息。针对这种信息可以使用例如 IP、TCP/UDP、
10 HTTP 或类似通信协议。

步骤 301 中,客户发送针对存储图像的请求。该图像被作为压缩比特数据流
并且以具有独立编码的编码单元(CU)的格式存储。步骤 303 中,服务器响应和开
始该发送。步骤 305 中,客户决定该图像部分更重要,并且如果选择一个感性
趣区域,则还要发送所选区域的形状以及其它需要的信息。这可以是例如该请
15 求的时间数字、CU 或接收字节的数量,图 4 中以(*)标记表示。在此时,客户
已经在该变换域中创建了一个屏蔽,例如使用在 Charilaos Christopoulos(编辑)
的“JPEG 2000 验证模化版本 1.2, ISO/IEC JTC1/SC29/WG1, N982 (1998 年 8
月 14 日)中描述的方法,选择该服务器所需要的系数,以便确定需要从该服
务器得到何种 CU。

20 步骤 307 中,服务器得到该请求。服务器使用这比特数据流中的信息,比如
TAGS 信息,以便找到期望的 CU。服务器发送重新开始标记,如果需要还发
送其余 CU 的长度。随后该服务器发送请求的 CU。

在下列情形中,客户开始接收一个图像,然后在该发送中决定该图像的某些
部分更重要,并且仅希望发送该涉及部分图像。这也被称之为感性趣区域的选
25 择。应该注意,在此情况中的该图像能够以一个感性趣区域(ROI)存储,但用
户要选择另一 ROI。在本例如中使用的是 JPEG2000 下的清晰度渐进(PBR)方
案。但是可以执行类似方法的精确度渐进(PBA)方案,如在 Charilaos
Christopoulos(编辑)的“JPEG 2000 验证模化版本 1.2, ISO/IEC JTC1/SC29/WG1
N982, (1998 年 8 月 14 日)”中描述的方案。

30 在图 4 中示出 JPEG 2000 中的清晰度渐进方案。JPEG 2000 的 PBR 模式中

的每一子频带以简化方式可以被视为一个编码单元 CU,因为整个子频带是被独立熵编码的。这是如果使用所谓的非自适应模式的情况,因为在此情况中的一个子频带与一个序列是同一回事。如果客户知道在该比特数据流中的何处找到该子频带的话,将有可能实现对任何子频带的独立熵解码。在 JPEG 2000 的最基本的模式中,这是由一个阵列支持的,该阵列存储在包含每一 CU 比特/字节长度的图像标题中。因此该客户有可能解析该比特数据流,因为其已知每一熵编码的子频带的长度。

该客户发送一个请求到服务器,并且请求存储的图像。该图像被存为一个压缩比特数据流。该服务器得到该比特数据流并且开始发送。在一个最佳实施例中,客户和服务器之间的交互作用以及在每一侧的操作如下所述,为了清楚起见,确认信息以及类似信息被省去。

-客户首先发送针对一个存储图像的请求。图像被存为一个压缩比特数据流并且是以 CU 格式存储。

- 服务器随即响应和开始该发送。

- 在发送过程中的某点,客户决定该图像一部分更重要并且发送该选择区域的形状以及其它需要的信息。这可以是例如该请求的时间数字、CU 或接收字节的数量,图 4 中以(*)标记表示。

- 服务器随即得到针对该 ROI 的请求并且对该还没发送的 CU 执行熵解码。在此情况中,是该子频带尚未发送。该熵解码将给出该量化的变换系数。服务器创建该变换域中的一个屏蔽。随即该服务器能够所选该感性趣区域所需要的系数,例如使用在 Charilaos Christopoulos(编辑)“JPEG2000 验证模化版本 1.2 ISO/IEC JTC1/SC29/WGI N982 (1998 年 8 月 14 日)”中描述的方法。因此,该服务器通过使用该屏蔽而选择哪个系数是每一个剩余子频带中所需要的。属于该感性趣区域的量化系数即同样属于熵编码的子频带。因此保持同一个 CU 结构。服务器可以发送一个重新起始标记以及将要发送的 CU 的长度,并且开始发送 CU。

- 客户从服务器得到描述已经使用的代码转换类型的响应。客户随即创建在该变换域中的期望的屏蔽,例如使用在 Charilaos Christopoulos(编辑)的“JPEG 2000 验证模化版本 1.2, ISO/IEC JTC1/SC29/WGI N982 (1998 年 8 月 14 日)”中描述的方法,从服务器选择针对该响应所需要的系数。

从客户的角度看, 结果是该感性趣区域将具有完整清晰度,而背景将具有降低的清晰度(如果使用上述利用移位的方法则将能够在后级有所改进)。清晰度降低的程度当然取决于做出感性趣区域请求的时间。

5 如果针对感性趣区域的请求选中在一个 CU 的中间, 这也是最可能的情况, 则有两个处理方式。 或者是把选中的 CU 重新发送,或者是完成该 CU 的传输然后开始重新编码。

在 JPEG2000 中的精度渐进的情况下, 能够使用与先前实例相同的概念而不做任何大的改变。 在此情况中的 CU 是一个比特平面。因此,该比特数据流是通过精确度排序的。首先发送最高的比特平面, 然后发送次高的比特平面等
10 等。

客户和服务器之间的交互作用以及在每一侧的操作如下所述,其中省去确认信息以及类似信息。

- 客户首先发送针对一个存储图像的请求。图像被存为一个压缩比特数据流, 并且是以 CU 格式存储。

15 - 服务器响应并且开始该发送。

- 客户决定该图像一部分更重要并且发送该选择区域的形状以及其它需要的信息。这可以是例如该请求的时间数字、CU 或接收字节的数量,图 4 中以(*)标记表示。 在此时, 客户已经在该变换域中创建了一个屏蔽, 例如使用在 Charilaos Christopoulos(编辑)的“JPEG 2000 验证模化版本 1.2, ISO/IEC
20 JTC1/SC29/WG1, N982 (1998 年 8 月 14 日)中描述的方法, 选择针对来自该服务器的响应所需要的系数。

- 服务器得到针对该 ROI 的请求并且对该还没发送的 CU 执行熵解码。在此情况中, 是该比特平面尚未发送。该熵解码将给出该量化变换系数的剩余部分。服务器创建该变换域中的一个屏蔽。随即该服务器能够所选该感性趣区域
25 所需要的系数,例如使用在 Charilaos Christopoulos(编辑)“JPEG2000 验证模化版本 1.2ISO/IEC JTC1/SC29/WGI N982 (1998 年 8 月 14 日)”中描述的方法。因此,该服务器通过使用该屏蔽而选择哪个系数是每一个比特平面中所需要的。属于该感性趣区域的量化系数的剩余部分即同样属于熵编码的比特平面。因此保持同一个 CU 结构。服务器现在可以发送一个重新起始标记以及将要发送的 CU
30 的长度, 并且开始发送 CU。

不象先前例子中那样具有该背景中的清晰度的降低,而是减小不属于该感性趣区域的像素的精确度。这是通过简单地跳过用于属于该背景的比特平面的其余比特平面实现的。

接收一个请求时,服务器对该原始图像做代码转换。代码转换的图像或者被发送到 5 一个输出缓冲器(下面的处理 C1 和 C2)或作为一个新图像递送(下面的处理 C3)。新的客户请求将总是参考一个递送图像,或是参考 C1 和 C2 中的原始图像(参见下面内容),或是参考在 C3 中递送的代码转换的图像(参见下面内容)。客户负责在图像格式之间进行变换,以使参考该原始图像的已收图像信息能够在客户对该代码转换的图像的复制中能够被再使用。成功的再使用 10 被在<Image data acknowledge>(图像数据确认)信息中报告给该服务器。

客户服务器处理 C1(下载速度的优化)

能够在没有任何先前图像信息的传输的条件下开始处理。但是,根据上述情况的结合图 2 或 3 的上述客户服务器处理可能已经出现。在此处理过程中,原始图像的某些 CU 可能已经转移到该客户。该客户知道此预先的活动,但是该 15 服务器不存储任何这种状态信息。

- 客户请求 1)<request number><Image data acknowledge><Image data request>
(<请求数目><图像数据确认><图像数据请求>)

- 服务器响应 1)<restart marker><request number><image transcoding header><image header><CU stream> (<再起动标记><请求数目><图像代码转换 20 标题><图像标题><CU 数据流>)(发送该代码转换的图像的标题)

如果该客户不懂该新格式,则其能够以 TCP 电平中断该数据流。可以继续新的一组客户服务器交换。它们将总是参考该原始图像,因为该代码转换的图像通常不被该服务器保持。如果根据该情况出现一个新请求,则服务器将通常随即重复该代码转换操作。该服务器可能决定保存该代码转换图像的一个缓存的 25 复制图像,但是这不能由该客户设定。

客户服务器处理 C2(针对低带宽的优化)

还可以在没有任何先前图像信息的传输的条件下开始这个处理。但是,根据上述情况的结合图 2 或 3 的上述客户服务器处理可能已经出现。在此处理过程中,原始图像的某些 CU 可以已经转移到该客户。该客户知道此预先的活动, 30 但是该服务器不存储任何这种状态信息。

- 客户请求 1) <request number><Image data acknowledge><Image data request>

- 服务器响应 1) <restart marker><request number><image transcoding header><image header> (<再起启动标记><请求数目><图像代码转换标题><图像标题>) (该服务器仅计算了该代码转换图像的标题。还没执行一个完整的代码转换)

- 客户请求 2) <request number><Image header acknowledge><Image data acknowledge><Image data request> (<请求号码><图像标题确认><图像数据确认><图像数据请求>) (该客户确认其能够操作该代码转换格式。由于该服务器不保存该旧请求的所以该请求被重复)

- 服务器响应 2) <restart marker> <request number><CU stream> (<再起启动标记><请求数目><CU 数据流>) (该服务器执行一个完整的代码转换并且把该图像文件的主体送到该输出缓冲器)

客户服务器处理 C3(由服务器传递的一个代码转换的图像)

能够在没有任何先前图像信息的传输的条件下开始处理。但是,根据上述情况的结合图 2 或 3 的上述客户服务器处理可能已经出现。在此处理过程中,原始图像的某些 CU 可以已经转移到该客户。该客户知道此预先的活动,但是该服务器不存储任何这种状态信息。

- 客户请求 1) <request number><Image data acknowledge><Image data request> (<请求数目><图像数据确认><图像数据请求>)

- 服务器响应 1) <restart marker><request number><image transcoding header><image header><image posting notification> (<再起启动标记><请求数目><图像代码转换标题><图像标题><图像传递提示>)

- 客户请求 2) (客户现在以 http 协议等级寻址该传递的代码转换图像) <request number><Image header acknowledge><Image data acknowledge><Image data Image data request> (<请求号码><图像标题确认><图像数据确认><图像数据请求>) (客户负责把该 CU 从原始图像变换到代码转换图像的格式。操作的结果被放进该 <Image data acknowledge>)

- 服务器响应 2) <restart marker><request number><CU stream> (<再起启动标记><请求数目><CU 数据流>)

可以继续新的一组客户服务器交换。根据由客户作出的判定,能够参考原始或代码转换图像。

5 应该注意,在某些情形中不需要该接收 CU 或比特/字节的数量。如果客户在请求已经发送之后继续接收图像数据即属于这种情况。服务器发送该重新起始标记以及其它可能的信息,以便通知客户从现在起该请求的比特平面、子频带或感性趣区域正在到来。

下面描述的是当数据流不包括期望的感性趣区域时在从比特数据流传输过程中交互性地选择感性趣区域附加实例。该实例示出不同比特数据流格式<CU 数据流 1>-<CU 数据流 4>。

10 结合图 4,先描述 JPEG2000 编码器的 PBR 模式。上述方案的一个改进将把 CU 的长度与该数据一起发送。因此,当服务器得到针对该 ROI 的请求时对该还没发送的 CU 执行熵解码。不是传送将要被发送的该 CU 的重新起始标记和长度以及开始该 CU 的发送,而是该服务器现在可以发送一个重新起始标记以及随后 CU 的长度。

15 图 5 示出了产生的客户服务器处理。这将导致不需要重新发送如图 4 所示的 CU 长度阵列。

也有可能在该比特数据流中使用 TAGS 或重新同步标记。因此不是具有上面结合图 4 的描述每一 CU 长度的一个阵列,而是该 CU 能够通过一个不是由该熵编码器产生的比特码型在该比特数据流中标出。以顺序方式搜索该比特数据流,以便找到不同的编码单元。客户与服务器之间的交互作用以及在每一方面的操作被改变,以便不送出一个重新起始标记以及将要被送出的该 CU 的长度,而是该服务器在能够发送在对应 CU 之前的一个重新起始标记以及一个 TAG。图 6 示出该产生的客户服务器处理。

25 应该注意,如图 7 所示,一个可选方法是使用在每一 CU 之后发送的 TAGS 或标题,而不是使用在该 CU 之前的 TAGS 或标题。因此,客户与该服务器之间的交互作用以及在每一侧的操作被改变,以便不发送该将要被发送的该 CU 的重新起始标记和长度以及开始该 CU 的发送,而是该服务器发送在该对应 CU 之后的一个重新起始标记和一个 TAG。

30 能够被使用的另一解决方案是"定标基础方法",在 JPEG2000 中用于其余子频带。这意味着仍然产生用于其余子频带的 ROI 屏蔽,但是不必须执行的 ROI

屏蔽系数的不同编码。该 ROI 屏蔽系数按照一个确定因数按比例放大。随即将不改变地继续该子频带的编码。移动值必须存储在比特数据流中,使得该客户能够下移。因此,客户与该服务器之间的交互作用以及在每一侧的操作被改变,以便不发送该将要被发送的该 CU 的重新起始标记和长度以及开始该 CU 的发送,而是该服务器发送在该对应 CU 之后的一个重新起始标记和一个 TAG。

在上面结合图 4-7 描述的方案中,服务器需要执行一个熵解码,然后执行量化系数的熵编码。如果期望真正地快速存取图像的不同部分,这不是一个好的方案。

对于此问题的解决方案是把该图像分成独立解码的数据块。这些数据块将是该 CU。客户和服务器之间的交互作用以及在每一侧的操作如图 8 所示。应该注意,该确认信息以及类似信息被省去。

- 步骤 801 中,客户发送针对存储图像的请求。该图像被存为一个压缩比特数据流。

- 步骤 803 中,服务器响应和开始该发送。

- 步骤 805 中,客户决定图像的一部分更重要并且创建变换域中的一个屏蔽,参见参考文献 1,选择需要的 CU。该期望的 CU 被送到该服务器。

- 步骤 807 中,服务器得到针对该 CU 的请求。服务器现在可以发送一个重新起始标记和针对该重新起始标记的在其对应 CU 之前的 TAG。

与在此之前的用于检索图像的客户-服务器系统比较,在此描述的方法和系统提供了若干优点。因此,服务器不需要任何存储器来存储其已经发送的部分的信息。无论什么原因希望把比特数据流的希望的一部分供给到该客户,在第一实施例中的服务器都不需要执行任何处理。客户将从该图像标题中获得请求部分存储的位置的信息。在第一和第二实施例中,服务器不需要执行任何熵解码,仅必须发送该请求的 CU。因此,极大地减少该传输时间。在该第三实施例中,服务器不必解码整个比特数据流。由于不需要执行该数据流的全解码,所以这将在发送一侧(服务器方)节省许多时间。

在此描述的方法和系统还可以被延伸与一个视频压缩算法一起使用,该算法具有在该压缩视频数据流中的可独立解码的单元。

说明书附图

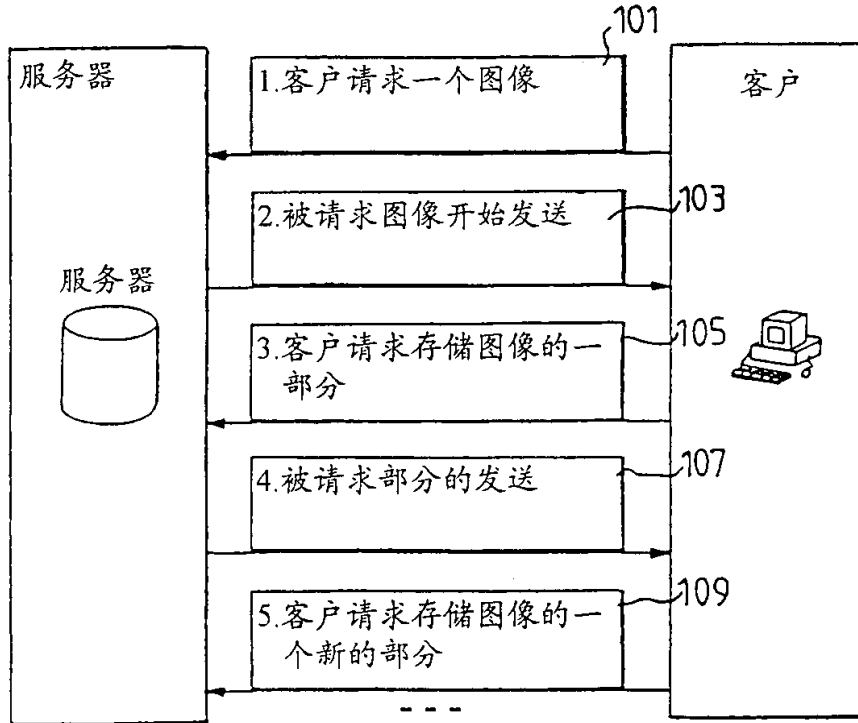


图 1

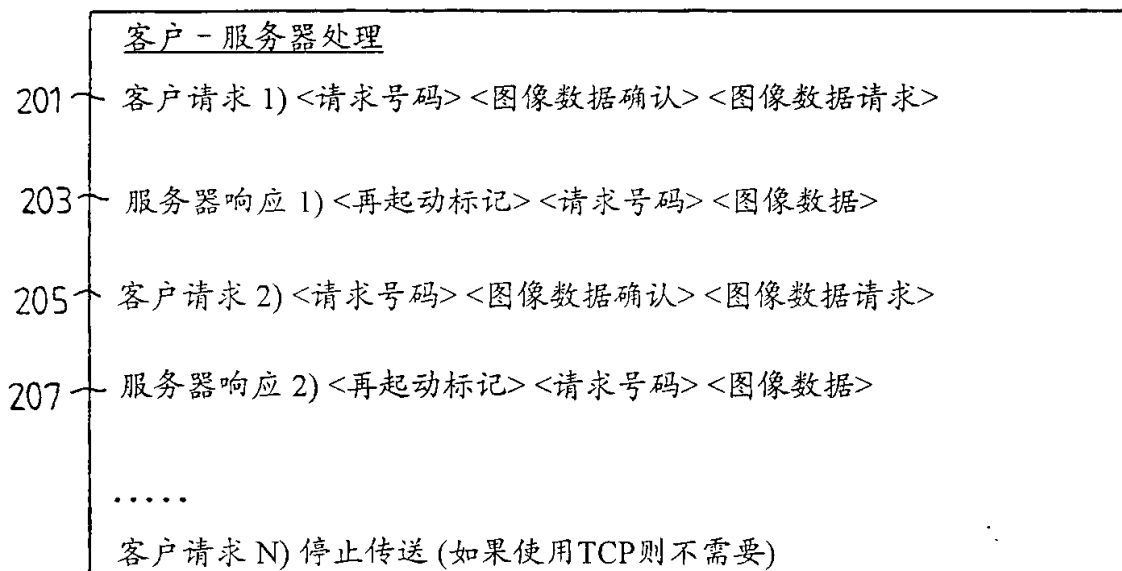


图 2

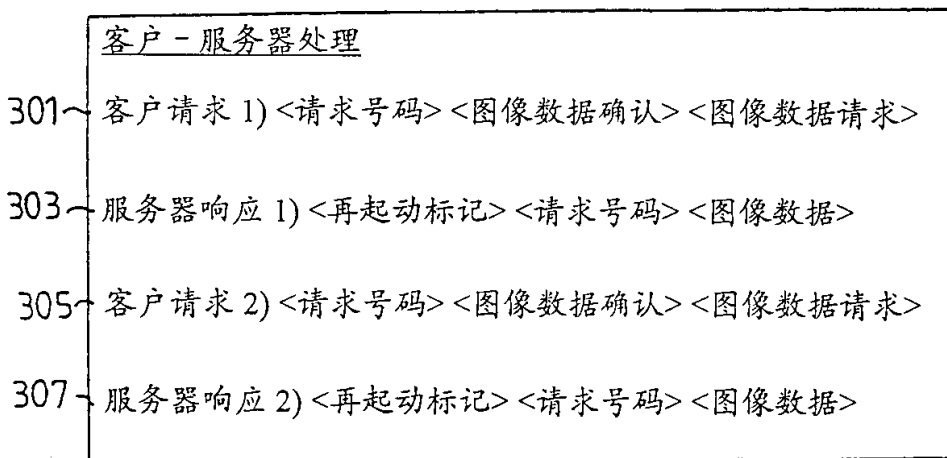


图 3

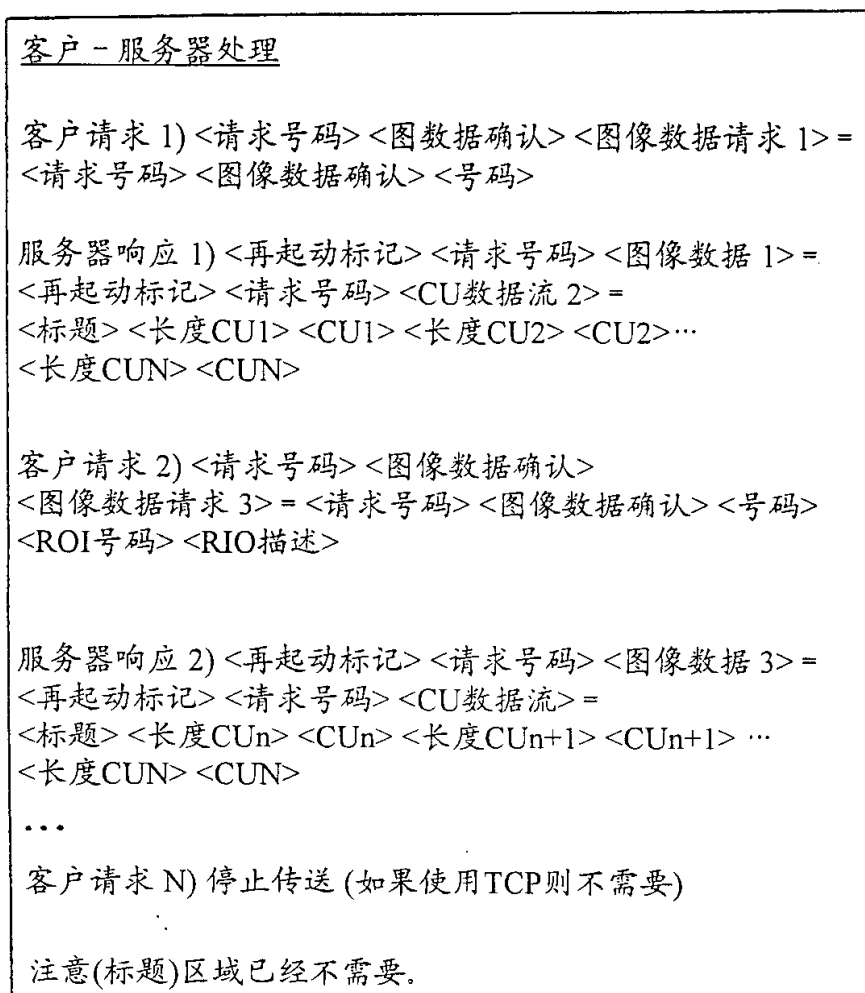


图 5

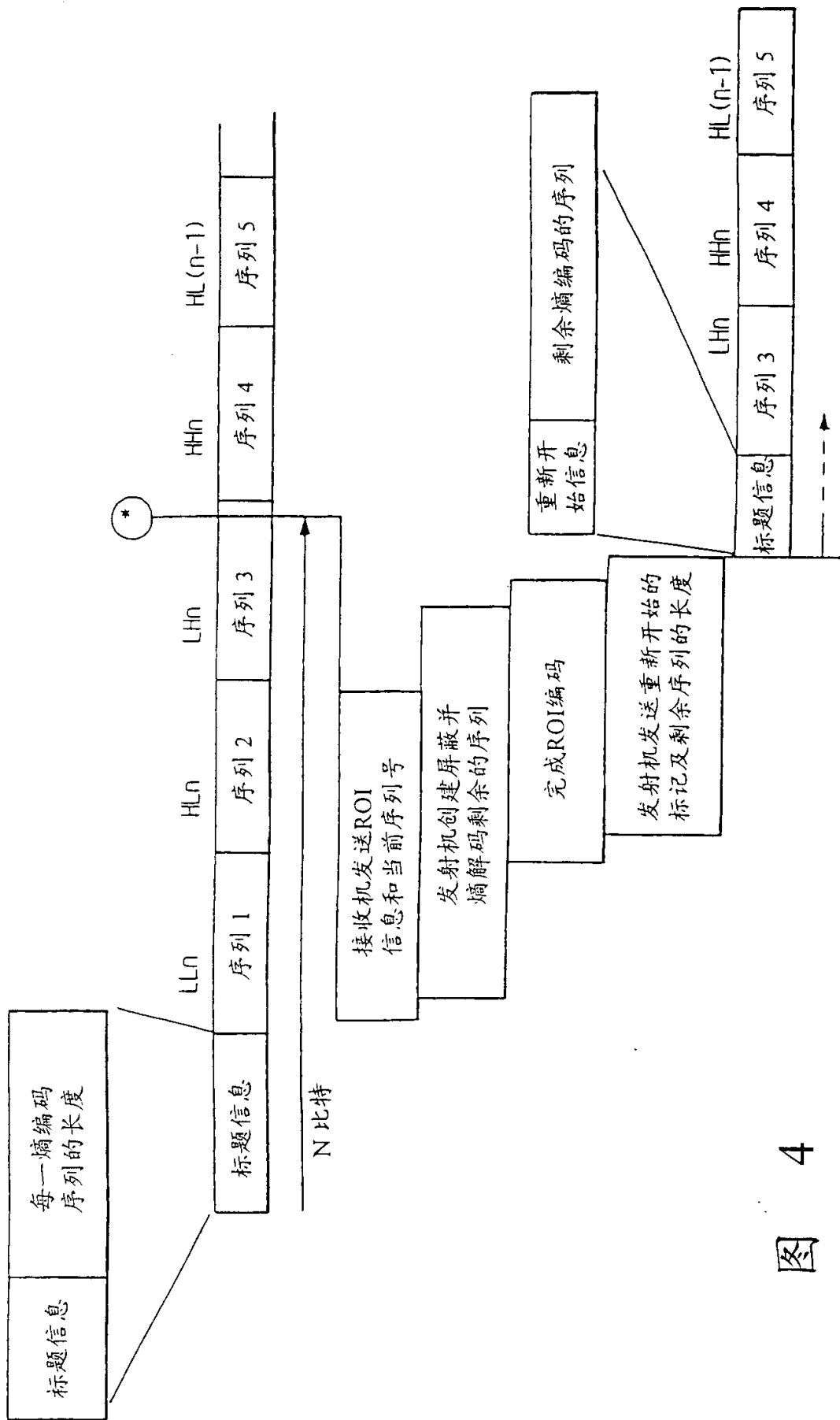
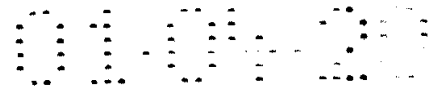


图 4



客户-服务器处理

客户请求 1) <请求号码> <图像数据确认>
 <图像数据请求 1> = <请求号码> <图像数据确认> <号码>

服务器响应 1) <再启动标记> <请求号码> <图像数据 1> =
 <再启动标记> <请求号码> <CU数据流 3> =
 <标题> <Tag 1> <CU 1> <Tag 2> <CU 2> ...
 <Tag N> <CUN>

客户请求 2) <请求号码> <图像数据确认>
 <图像数据请求3> =
 <请求号码> <图像数据确认> <号码> <ROI号码>
 <ROI描述>

服务器响应 2) <再启动标记> <请求号码> <图像数据 3> =
 <再启动标记> <请求号码> <CU数据流> =
 <标题> <Tag n> <CUn> <Tag n + 1> <CUn + 1> ...
 <Tag N> <CUN>

...

客户请求 N) 停止传送(如果使用TCP则不需要)

注意(标题)区域已经不需要。

图 6

客户 - 服务器处理

客户请求 1) <请求号码> <图像数据确认>
 <图像数据请求 1> = <请求号码> <图像数据确认> <号码>

服务器响应 1) <再启动标记> <请求号码> <图像数据 1> =
 <再启动标记> <请求号码> <CU数据流 4> = <标题> <CU 1>
 <Tag 1> <CU 2> <Tag 2> ... <CUN> <Tag N>

客户请求 2) <请求号码> <图像数据确认>
 <图像数据请求 3> = <请求号码> <图像数据确认> <号码>
 <ROI号码> <ROI描述>

服务器响应 2) <再启动标记> <请求号码> <图像数据 3> =
 <再启动标记> <请求号码> <CU数据流> =
 <标题> <CU n> <Tag n> <CU n + 1> <Tag n + 1> ...
 <CUN> <Tag N>

...

客户请求N)停止传送(如果使用TCP则不需要)

注意(标题)区域已经不需要。

图 7

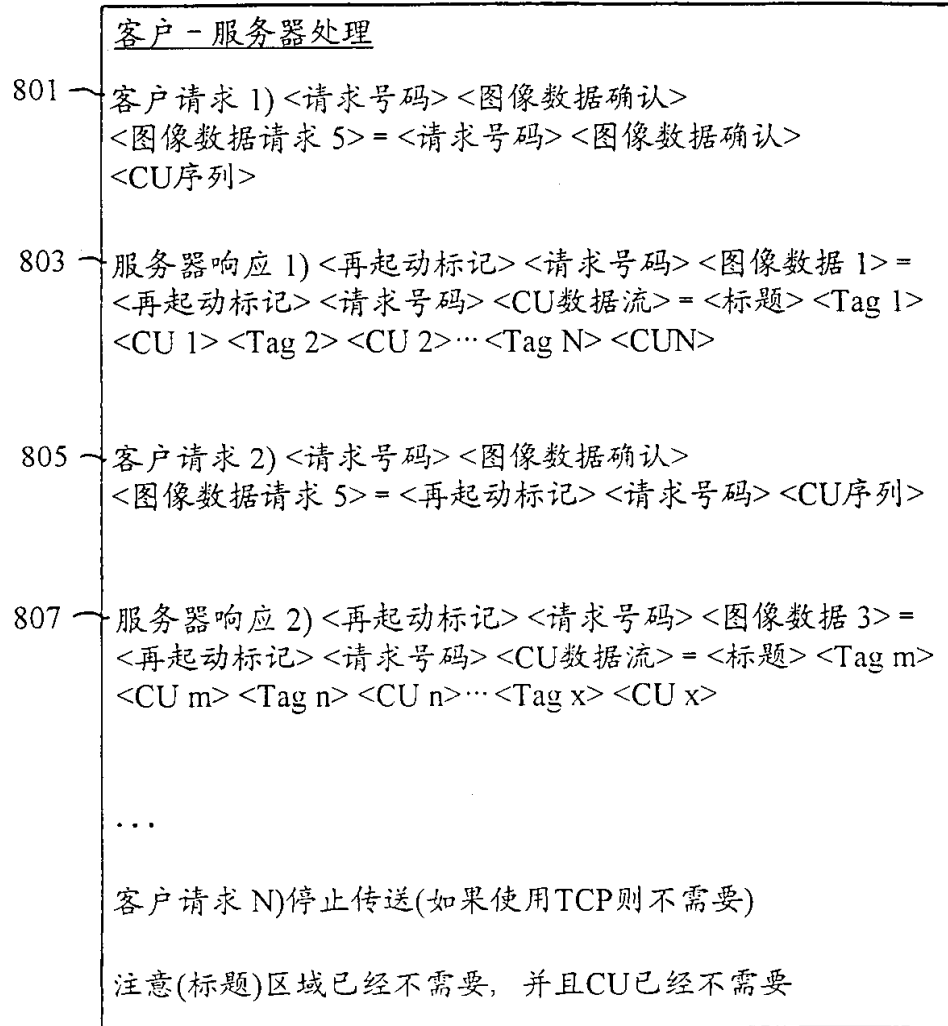


图 8