



(12) 发明专利

(10) 授权公告号 CN 101039420 B

(45) 授权公告日 2011.02.02

(21) 申请号 200710086668.2

第 1-3 段 .

(22) 申请日 2007.03.30

同上 .

(73) 专利权人 孟智平

曾焕强

蔡灿辉

地址 646006 四川省泸州市茜草坝长起厂一
生活区 26 号楼 1 号

.H. 264 /AVC 快速帧内预测模式
选择算法 . 中国图象图形学报 Vol. 12
No. 2. 2007, Vol. 12 (No. 2), 189-194.

(72) 发明人 孟智平

审查员 张立国

(51) Int. Cl.

H04N 7/26 (2006.01)

H04N 7/32 (2006.01)

H04N 7/50 (2006.01)

(56) 对比文件

CN 1812582 A, 2006.08.02, 全文 .

CN 1204102 A, 1999.01.06, 全文 .

CN 1239250 A, 1999.12.22, 说明书第 1 页第
1-3 段, 第 2 页第 2 段 - 第 5 页第 1 段, 附图 1-5.

JP 特开平 7-110858 A, 1995.04.25, 全文 .

CN 1455600 A, 2003.11.12, 全文 .

CN 18901929 A, 2006.07.12, 说明书第 2 页

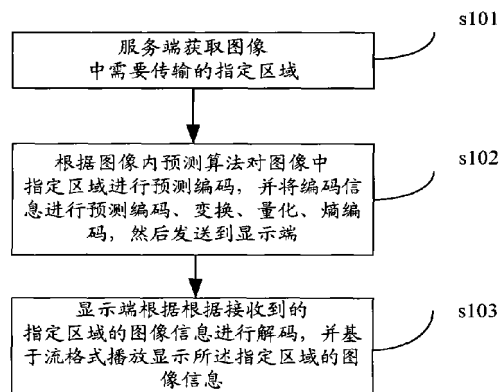
权利要求书 1 页 说明书 16 页 附图 16 页

(54) 发明名称

一种基于流格式的指定区域图像传输及显示
方法

(57) 摘要

本发明公开了一种基于流格式的图像传输方法 : 服务端通过流格式把图像中指定区域的图像信息传送到显示端 ; 所述显示端根据接收到的指定区域的图像信息进行解码, 并基于流格式播放显示所述指定区域的图像信息。本发明实施例中, 不需要传输整张图像, 而是以某一条线或某一块为基础来显示的, 即边传输边显示 ; 使用 I 条带或 I 型块来作为可以单独解码的最小单位, 然后根据相邻块或相邻条带之间的相关性来实现预测编码, 然后向显示端传输预测模式信息, 显示端根据所述预测模式信息对编码信息进行解码。



1. 一种基于流格式的指定区域图像传输方法,其特征在于包括以下步骤:
服务端通过流格式对图像中指定区域的图像信息进行编码,并传送到显示端;
所述显示端根据接收到的指定区域的图像信息进行解码,并基于流格式播放显示指定区域的图像信息;

所述编码时,先把图像转换成点阵模式,然后通过图像内预测方法对宏块或条带进行编码;

所述解码时,根据所选择图像内预测方法解码出图像信息;

所述图像内预测方法,包括以下步骤:

获取宏块或条带周边的相邻像素值;

使用带顶角像素加权和不带顶角像素加权两者中任意一个进行图像内预测;

其中,像素加权公式为:

$$X = \sum_{i=1}^n M_i Y_i$$

其中, X 像素和 Y_i 相关, M_i 代表相关系数,

相关系数表示为:

$$M = p \times q$$

其中 p 为距离系数,即在一个能量通道里面,相关系数点距离被预测位置占整个通道长度的比例;

q 为比例系数,即这条能量通道里面的分割占整个预测像素的比例值;

所述图像内预测方法对条带进行预测包括已知一边或已知两对称边的情况下,对条带的水平模式预测、水平向下模式预测、水平向上模式预测、对角线模式预测;

所述图像内预测方法对宏块进行预测包括已知一边、已知两对称边、已知三边及已知四边的情况下,对宏块的水平模式预测、水平向下模式预测、水平向上模式预测、垂直模式预测、垂直向左模式预测、垂直向右模式预测,对角线模式预测。

2. 如权利要求 1 所述基于流格式的指定区域图像传输方法,其特征在于:所述服务端通过流格式把指定区域的图像信息进行编码、传送到显示端之前还包括:所述服务端获取图像中需要传输的指定区域,所述指定区域是指在整个图像内部具体某个位置范围,该区域由一个或多个条带或宏块构成,所述条带或宏块为构成图像的基本单位,也是最小的可解码显示单位。

3. 如权利要求 2 所述基于流格式的指定区域图像传输方法,其特征在于采用点对点 P2P 协议配合宏块或条带的传输。

4. 如权利要求 1 所述基于流格式的指定区域图像传输方法,其特征在于:在需要网络传输的时候实时编码;或先编码之后存储然后再在网络上传输;其中,先编码之后采用文件分级存储形式。

一种基于流格式的指定区域图像传输及显示方法

技术领域

[0001] 本发明涉及视频传输技术领域,尤其涉及一种基于流格式的指定区域图像传输及显示方法。

[0002] 背景技术

[0003] 现有技术中,音/视频等多媒体信息的传输方式主要包括下载和流式传输两种。下载方式通常采用数据全部传输,数据全部保存,数据全部显示,然而,音/视频文件一般都较大,需要的存储容量也较大,同时由于网络带宽的限制,采用下载方式常常要花数分钟甚至数小时,所以这种处理方法延迟很大。采用流式传输方式时,声音、影像或动画等时基媒体由音视频服务器向用户计算机实时传送,用户不必等到整个文件全部下载完毕,只将开始部分内容存入内存,而只需经过几秒或十数秒的启动延时即可进行观看,数据流随时传送随时播放,只是在开始时有一些延迟,当时基媒体在客户机上播放时,文件的剩余部分将在后台从服务器内继续下载。因此,流式传输不仅使启动延时成十倍、百倍地缩短,而且不需要太大的缓存容量,避免了用户必须等待整个文件全部从网上下载才能观看的缺点。

[0004] 其中,流式传输主要指通过网络传送媒体(如视频、音频)的技术总称。其特定含义为通过互联网络将影视节目传送到个人计算机。实现流式传输有两种方法:实时流式传输(Realtime streaming)和顺序流式传输(progressivestreaming)。一般说来,如视频为实时广播,或使用流式传输媒体服务器,或应用如 RTSP(RealTimeStreamingProtocol,实时流协议)的实时协议,即为实时流式传输。如使用 HTTP(Hypertext Transfer Protocol,超文本传输协议)服务器,文件即通过顺序流发送。采用那种传输方法依赖客户的需求,当然,流式文件也支持在播放前完全下载到硬盘。

[0005] 顺序流式传输是顺序下载,在下载文件的同时用户可观看再线媒体,在给定时刻,用户只能观看已下载的那部分,而不能跳到还未下载的前头部分,顺序流式传输不象实时流式传输在传输期间根据用户连接的速度做调整。由于标准的 HTTP 服务器可发送这种形式的文件,也不需要其它特殊协议,经常被称作 HTTP 流式传输。顺序流式传输比较适合高质量的短片段,如片头、片尾和广告,由于该文件在播放前观看的部分是无损下载的,这种方法保证电影播放的最终质量,对通过调制解调器发布短片段,顺序流式传输显得很实用,顺序流式传输允许用比调制解调器更高的数据速率创建视频片段。另外,顺序流式文件是放在标准 HTTP 或 FTP(File Transfer Protocol,文件传输协议)服务器上,易于管理,基本上与防火墙无关。但是顺序流式传输不适合长片段和有随机访问要求的视频,如:讲座、演说与演示等,用户在观看前,必须经历延迟,对较慢的连接尤其如此,因此,顺序流式传输不支持现场广播,严格说来,顺序流式传输职能是一种点播技术。

[0006] 实时流与 HTTP 流式传输不同,需要专用的流媒体服务器与传输协议,实时流式传输总是实时传送,特别适合现场事件,也支持随机访问,用户可快进或后退以观看前面或后面的内容。

[0007] 实时流式传输必须配匹连接带宽,这意味着在以调制解调器速度连接时图像质量较差,而且,由于出错丢失的信息被忽略掉,网络拥挤或出现问题时,视频质量很差。如

欲保证视频质量,顺序流式传输也许更好。实时流式传输需要特定服务器,如 QuickTime Streaming Server、RealServer 与 WindowsMedia Server。这些服务器允许对媒体发送进行更多级别的控制,因而系统设置、管理比标准 HTTP 服务器更复杂。实时流式传输还需要特殊网络协议,如:MMS(Microsoft Media Server,微软媒体服务器)。这些协议在有防火墙时有时会出现问题,导致用户不能看到一些地点的实时内容。以上传输控制协议,如 RTP(Real-timeTransportProtocol,实时传输协议)/UDP(User DatagramProtocol,用户数据报协议)和 MMS/HTTP/TCP(Transmission Control Protocol,传输控制协议)等。实时流式传输关注的是流式文件本身,对传输层做到透明支持。

[0008] 现有的图像传输技术,即通过网络来下载并显示图像技术,基本上都全部下载,全部显示或从图像的头端开始一点点显示。这种技术的缺点是把图像中所有的信息都平行看待,又由于现有图像的编码的限制,不能做到对信息的重点提取、重点传输,当然就不能从图像中的某一区域可控的开始显示。另外现有的图像编码技术中,没有使用预测算法,在传输时,不能很好的节省带宽,而 H. 264 现有的帧内预测算法,必须知道上面和左边的数据作为参考来进行解码,所 H. 264 的帧内算法,也只能按扫描顺序,从左到右,从上到下解码和显示整个图像。

[0009] 现有技术中的流式传输,不论是顺序流式传输,还是实时流式传输,通常采用分帧视频传输,单独解码,整帧显示,帧内预测(需要知道上面和左面编码数据作为预测参考)、帧间预测(运动补偿)。无法实现对视频数据一起保存(或分散保存),分散传输,分散显示,灵活的在比帧更小的范围内进行视频数据传输。

[0010] 现有的巨型的图形系统,如, GIS 或城市地区地图等,往往都是采用现有的图像编码方式,如 JPEG、GIF 等格式,由于数据量很大,网络传输成为其一个应用瓶颈。再如,现有的电子书系统,传输方面也是一个巨大的瓶颈。尤其是对于没有帧结构的巨型图像,如何编码,如何传输,如何解码都是一个尚待解决的问题。

[0011] 发明内容

[0012] 本发明实施例要解决的问题是提供一种基于流格式的指定区域图像传输及显示方法,以解决现有技术中无法实现对图像数据一起保存(或分散保存),分散传输,指定区域显示的缺陷。

[0013] 为了实现上述目的,本发明的实施例提供了一种基于流格式的指定区域图像传输方法,包括以下步骤:

[0014] 服务端通过流格式对图像中指定区域的图像信息进行编码,并传送到显示端;

[0015] 所述显示端根据接收到的指定区域的图像信息进行解码,并基于流格式播放显示所述指定区域的图像信息。

[0016] 所述服务端通过流格式把指定区域的图像信息传送到显示端之前还包括:所述服务端获取图像中需要传输的指定区域,所述指定区域是指在整个图像内部具体某个位置范围,该区域由一个或多个条带或宏块构成。所述条带或宏块为构成图像的基本单位,也是最小的可解码显示单位。

[0017] 在编码时,需要先把图像转换成点阵模式,然后通过图像内预测方法对宏块或条带进行编码,所述预测算法在对像素进行预测时,使用带顶角像素加权和不带顶角像素加权两者中任意一个进行图像内预测。

[0018] 采用 P2P 协议配合宏块或条带的传输。

[0019] 在需要网络传输的时候实时编码；或先编码之后存储然后再在网络上传输；其中，先编码之后采用文件分级存储形式。

[0020] 在编码之前，需要以点阵结构存储的图片原始信息，如 YUV（一个亮度信号：Y；两个色差信号：U 分量、V 分量）模式就是一个典型的点阵模式，即每点像素对应一个亮度值两个色度值，同样 RGB（红绿蓝）模式也是点阵模式。然后再通过本发明的预测算法求出残差，再利用现有的 DCT（Discrete Cosine Transform，离散余弦变换）变换、量化、熵编码等方法进行处理。

[0021] 在解码的时候，首先经过反熵编码、反量化、IDCT（反离散余弦变换）变换等步骤，然后根据所选择预测算法解码出图像信息。

[0022] 在编码时候，判断那种预测模式最好通常采用 SAD（绝对差值和）或方差的方法。

[0023] 本发明还提供了一种基于流格式的图像内预测方法，包括以下步骤：

[0024] 获取宏块或条带周边的相邻像素值；

[0025] 使用带顶角像素加权和不带顶角像素加权两者中任意一个进行图像内预测；

[0026] 其中，像素加权公式为：
$$X = \sum_{i=1}^n M_i Y_i$$
；其中，X 像素和 Y_i 相关， M_i 代表相关系数，相关系数表示为： $M = p \times q$ ；其中 p 为距离系数，即在一个能量通道里面，相关系数点距离被预测位置占整个通道长度的比例；q 为比例系数，即这条能量通道里面的分割占整个预测像素的比例值。

[0027] 所述条带预测包括已知一边或已知两对称边的情况下，对条带的水平模式预测、水平向下模式预测、水平向上模式预测、对角线模式预测；所述宏块预测包括已知一边、已知两对称边、已知三边及已知四边的情况下，对宏块的水平模式预测、水平向下模式预测、水平向上模式预测、垂直模式预测、垂直向左模式预测、垂直向右模式预测，对角线模式预测。

[0028] 采用简化的系数代替由预测算法产生的复杂的算术运算。

[0029] 本发明还提供了一种基于流格式的指定区域图像显示方法，使用宏块或条带作为图像显示的基本单位，具体为使用宏块或条带显示指定区域的图像信息。显示指定区域的图像信息中，所述指定区域是指在整个图像内部具体某个位置范围，该区域由一个或多个条带或宏块构成，并可以用预先设置的程序来控制条带或宏块的显示速度和显示位置；所述控制显示速度对于条带模式，表现为通过快进快退的方式观看图像内容。

[0030] 与现有技术相比，本发明实施例具有以下优点：

[0031] 本发明实施例中，不需要传输整张图像，也不分帧结构，条带或宏块为构成图像的基本单位，同时也是解码显示的最小单位；然后根据本发明的图像预测算法来实现预测编码，然后本发明的图像传输算法向显示端传输预测模式和残差代码，显示端根据所述预测模式和残差代码对编码信息进行解码，获得指定区域的图像信息。本发明中，图像内预测方法和图像传输算法可以根据具体需要选择使用。

[0032] 在 H. 264 的抗误码方法中，有 FMO 特性，允许独立和分散的宏块映射到 Slice 里，然后用 Slice Group 来封装。这种方法的目的主要是为了抗误码需要，用在视频传输里面。预测方法支持帧内和帧间预测，其中帧内预测和本专利完全不同，角度也不一样。虽然提出

了分散传输的概念,但是并不能很好的对指定位置进行有针对性的传输,更不可能对指定区域进行有针对性的显示,属于完全不同的技术范畴。

[0033] 本发明还规定了一种新型可预测编码和显示的单位-条带,这种显示单位和 H. 264 里的 Slice 完全不是一个概念,而是针对显示和编码的需要设计出来的。而不是以传输和抗误码为目的的,因此实现方法也完全不同。

[0034] 本发明还专门设计了用于支持流式图像传输的预测方法,该预测方法既保证了可以及时解码的特性又提高传输的效率。在解码的时候还提供算法可以实现对播放速度和位置的控制。

附图说明

[0035] 图 1 是本发明一种基于流媒体的图像传输方法流程图;

[0036] 图 2 是本发明宏块构成示意图;

[0037] 图 3 是本发明宏块构成条带的示意图;

[0038] 图 4 是本发明纵向条带组成整帧示意图;

[0039] 图 5 是本发明纵向条带中的 I 条带示意图;

[0040] 图 6 是本发明横向条带组成整帧示意图;

[0041] 图 7 是本发明横向条带中的 I 条带;

[0042] 图 8 是本发明流式图像的服务端编码方法;

[0043] 图 9 是本发明流式图像使用方法;

[0044] 图 10 是本发明一解码实例示意图;

[0045] 图 11 是本发明宏块分级示意图;

[0046] 图 12a 是本发明平均模式预测示意图;

[0047] 图 12b 是本发明水平模式预测示意图;

[0048] 图 12c 是本发明水平向下模式预测示意图;

[0049] 图 12d 是本发明对角线 1 模式预测示意图;

[0050] 图 12e 是本发明垂直向右模式预测示意图;

[0051] 图 12f 是本发明垂直向左模式预测示意图;

[0052] 图 12g 是本发明对角线 2 模式预测示意图;

[0053] 图 12h 是本发明水平向上模式预测示意图;

[0054] 图 12i 是本发明为 DCT 变换预测模式示意图;

[0055] 图 12j 是本发明 B 块示意图;

[0056] 图 13a 是本发明已知三块的预测方法;

[0057] 图 13b 是本发明已知四块的预测方法;

[0058] 图 14a 是本发明已知四边水平模式预测示意图;

[0059] 图 14b 是本发明已知四边对角线 1 模式预测示意图;

[0060] 图 14c 本发明已知四边水平向下模式预测示意图;

[0061] 图 14d 本发明已知四边垂直向右模式预测示意图;

[0062] 图 14e 本发明已知四边垂直模式预测示意图;

[0063] 图 14f 本发明已知四边对角线 2 模式预测示意图;

- [0064] 图 14g 本发明已知四边水平向上模式预测示意图；
- [0065] 图 14h 本发明已知四边垂直向左模式预测示意图；
- [0066] 图 15a 是本发明已知三边水平模式预测示意图；
- [0067] 图 15b 是本发明已知三边对角线 1 模式预测示意图；
- [0068] 图 15c 本发明已知三边水平向下模式预测示意图；
- [0069] 图 15d 本发明已知三边垂直向右模式预测示意图；
- [0070] 图 15e 本发明已知三边垂直模式预测示意图；
- [0071] 图 15f 本发明已知三边对角线 2 模式预测示意图；
- [0072] 图 15g 本发明已知三边水平向上模式预测示意图；
- [0073] 图 15h 本发明已知三边垂直向左模式预测示意图；
- [0074] 图 16a 是本发明已知一边水平模式预测示意图；
- [0075] 图 16b 是本发明已知一边对角线 1 模式预测示意图；
- [0076] 图 16c 是本发明已知一边水平向下模式预测示意图；
- [0077] 图 16d 是本发明已知一边垂直向右模式预测示意图；
- [0078] 图 16e 是本发明已知一边垂直模式预测示意图；
- [0079] 图 16f 是本发明已知一边对角线 2 模式预测示意图；
- [0080] 图 16g 是本发明已知一边水平向上模式预测示意图；
- [0081] 图 16h 是本发明已知一边垂直向左模式预测示意图；
- [0082] 图 17 是本发明确定总公式系数示意图；
- [0083] 图 18a 是本发明已知两边条带水平模式预测示意图；
- [0084] 图 18b 是本发明已知两边条带水平向上模式预测示意图；
- [0085] 图 19 是本发明 8×8 带顶角信息的对角线 1 模式预测示意图；
- [0086] 图 20a 和图 20b 是本发明增加边缘宏块预测示意图；
- [0087] 图 21 是本发明解码的次序为 1、2、3、4、5、6、7 示意图；
- [0088] 图 22 是本发明图 21 实现流程图；
- [0089] 图 23 是本发明解码的次序为以中心往边缘扩散示意图；
- [0090] 图 24 是本发明图 23 实现流程图；
- [0091] 图 25 至图 27 是本发明圆形扩散算法示意图；
- [0092] 图 28 是本发明分散存储和分散传输的示意图；
- [0093] 图 29 是本发明以 I 条带为例来实现对显示速度的条带序列图；
- [0094] 图 30 是本发明以宏块为例来实现对显示速度的控制示意图；
- [0095] 图 31 是本发明服务端和显示端的系统结构图；
- [0096] 图 32 是本发明用协议的方式改变或确定显示位置的消息过程和改变显示速度的消息过程；
- [0097] 图 33 是本发明一种图像实时传输示意图；
- [0098] 图 34 是本发明以宏块编码方式为例二级索引文件分级存储示意图；
- [0099] 图 35 是本发明以条带为例三级索引文件分级存储示意图。
- [0100] 具体实施方式
- [0101] 本发明实施例中，一种基于流媒体的图像传输方法，如图 1 所示，具体包括以下步

骤：

[0102] S101, 服务端在编码前或编码时需要把先把图像转换成 YUV 模式或其它点阵模式, 然后获取图像中需要传输的指定区域, 所述指定区域包括条带或宏块, 在图像信息传输之前, 将图像设置成以条带或宏块为基本单元的模式, 而不是以整帧或整幅图像作为基础单元编码, 也不需要从左到右, 从上到下的固定顺序。其中, 构成图像的基本单位宏块或用于变换的基本单位块如图 2 所示, 包括 1×1 、 1×2 、 2×1 、 2×2 、 4×2 、 2×4 、 4×4 、 4×8 、 8×4 、 8×8 、 8×16 、 16×8 、 16×16 等; DCT 变换通常采用 2×2 、 4×4 、 8×8 、 16×16 。

[0103] 宏块可以由多个像素构成, 条带可以由多个宏块构成, 条带也可以直接作为最小的解码显示单位也可以作为最小的预测单位, 条带可以分为纵向和横向两种, 条带通常具有相同的结构, 比如构成条带的横向像素和纵向像素数目通常是恒定的; 如果条带由宏块构成, 则构成条带的宏块数目也是恒定的, 如图 3 所示; 纵向条带组成整幅图像的示意图如图 4 所示; 纵向条带中的 I 条带和 B 条带示意图如图 5 所示, 其中 I 条带表示可以单独解码显示的条带, 在编码的时候也不需要预测算法以及和周围条带求残差的过程, 而 B 条带是一种不能够独立解码显示的条带, 它需要在周围 I 条带已经解码的基础上才能够解码显示, 而且 B 条带在编码的时候也需要预测编码; 横向条带组成整幅图像的示意图如图 6 所示; 横向条带中的 I 条带和 B 条带示意图如图 7 所示, 横向条带和纵向条带只有方向上的区别, 编解码的方式和传输过程都是大同小异的, 只是在预测的时候把纵向条带横过来作横向预测就可以了, 当然解码的时候也采用横向的方式进行解码。如果编码中条带是由宏块构成, 即预测算法的基本单位是宏块而不是条带, 这是 I 条带也必须能够独立解码。由于条带内部在编码的时候也需要 DCT 变换, 而 DCT 变换最好为方形矩阵, 因此条带必须能被划分为多个方形的集合。在编码时, 可以通过特定的位来区别 I 条带或 B 条带。

[0104] S102, 根据图像内预测方法对指定区域或整幅图像进行预测编码, 并将编码信息进行预测编码、变换、量化、熵编码, 然后发送到显示端。实现条带或宏块间的预测编码, 服务端编码方式有两种, 一种是预编码方式, 即在传输前就把编码工作全部完成, 通常是统一编码, 统一编码通常是按一定方向一定顺序进行, 如从左到右, 从上到下等, 这个时候往往每一条带或宏块都有唯一的序列号和其对应, 或者通过某一算法公式可以找到需要的条带或宏块, 在实际存储过程中往往采用分级存储的方法, 这样可以很容易检索和定位; 另一种方式是实时编码方式, 即在需要传输时才编码, 一边编码一边传输。例如采用实时编码方式的时候, 以宏块为例, 选定一种扩散算法, 然后一层一层往外编。对于采用预测方式编码的块, 需要传输预测模式、和残差代码, 然后再变换、量化、熵编码。

[0105] 在网络传输方面: 通过 I 条带可以实现快进快退功能, 具体参考后面关于速度控制模块和位置控制模块的说明。条带或宏块还可以以 P2P (点到点) 的方式来传输文件, 达到网络的高效使用的目的。而现有的图像由于要整体显示, 所以 P2P 使用的性能不高, 但是由于本发明是基于流模式的, 所以在使用 P2P 时, 把图像信息分布存储在一些服务端上, 通过 P2P 后台控制来传输图像内容, 这样可以边传输边显示, 这样就可以快速得到图像信息, 而且不会因为某台服务器的处理能力影响服务质量。

[0106] S103, 显示端根据接收到的指定区域的图像信息及从服务端获取或预先约定的预测算法进行解码获取原始图像, 并基于流格式播放显示所述指定区域的图像信息。显示端进行反熵编码、反量化、反变换, 并根据预测模式和残差代码对接收的编码信息进行解码,

得到条带或宏块,从而获得一幅图像的一部分。结合上述原理,一种流式图像的编码方法如图 8 所示,包括:图像转换为 YUV 格式;按条带读取;判断是否为 I 条带,如果是,则进行条带内预测,如果不是,则进行条带间预测;进行变换、量化、熵编码。流式图像使用方法如图 9 所示,包括:在服务端获取图像信息;将其存储为流格式图像;进行流传输;在显示端读取成流格式图像。

[0107] 一个具体解码实例如图 10 所示,A、B 和 D 块为 1 型宏块(已对其进行编码,1 型宏块可以独立解码不需要参考其它信息),如用户关心区域由 A 区域开始向周围扩散,扩散的模式以正方形模式为例。

[0108] 能解码的先决条件至少满足以下两个条件中的一个:要么是 1 型宏块;要么上下或左右两宏块已经解码、或相邻三块已经解码、或相邻四块已经解码;

[0109] 其中,解码的顺序可以为固定顺序,例如:设置 A 位置为 (0,0),右边第一块 (0,1),左边的一块 (0,-1),上边的一块 (-1,0),下边的一块 (1,0),以这样的坐标方法标记,每一块都会对应一组整数,按照这一组整数的奇偶特性可以分为偶偶块、奇偶块、偶奇块和奇奇块四种结构,先解码中心的偶偶块,再解下层偶偶块,再解奇偶块,再解偶奇块,再解奇奇块。

[0110] 解码顺序也可以为:距离中心远近的方法:A、BBBB、CCCC、DDDD、FFFF、EEEEEEEE;即先传输最靠近的一个可解码块,然后判断四个周围块是否可以解码,(按中心最近原则,也就是说,离 A 块中心最近的四个块是否可以解码,然后判断离 A 块中心次近的四个块,然后以此类推)这样到了 B 块,发现可以直接解码,然后再从 A 块开始判断还没有解码的,判断和 A 块中心最近的未解码块能否解码,发现 C 块可以解码,则解码之。同理下去发现 F 不能解码,E 也不能,解码 D,然后再从 A 距离条件开始判断,发现如果可以解码,则解码,如果不可以,然后 F 解码,最后 E 解码。

[0111] 解码的顺序还可以为:按里圈到外圈的顺序:A(里圈)、DBDBDBDB(外圈);再扫描里圈 CCCC、EEEEEEEE(外圈);再扫描里圈 FFFF。

[0112] 为了可以从一幅图像中任何一个地方切入来接收图像部分的数据或直至接收图像的整个数据,必须要在图像内部设置可以单独完成解码的显示单元。在视频流媒体中是通过设置 I 帧的方法来完成的,即用户可以在视频播放过程中随机接入视频流,只要获得一个完整的 I 帧就可解码出完整一帧图像,然后通过帧之间的预测编码的逆过程还原出 P 帧和 B 帧的图像,同时可以利用 I 帧的引入而实现快进快退的功能。本发明借用了在视频中设置 I 帧的思想,让用户可以定位一个关心的位置来接收这部分图像数据的内容,而不需要等图像传输完毕之后再显示图像。这种编码和数据传输方法可以用于电子地图,地理信息系统(GIS),巨型图像传输和军事图像数据传输等。

[0113] 在图 11 中,A 到 I 是图像中的一部分数据,定义 A、C、G、I 为 1 型宏块,不适用预测模式,为传输的可以独立解码的最基本单元,使用 DCT 变换、量化、熵编码等方式完成数据编码。解码的时候采用反熵编码、反量化、IDCT 的方法进行解码即可。B、D、F、H 为 2 型宏块,2 型宏块根据左右或上下的 1 型宏块进行预测,然后使用 DCT、量化、熵编码,完成编码;解码时,通过反熵编码、反量化、IDCT、使用预测逆算法解码,通常只有左右或上下一侧为已解码 1 型宏块。E 为 3 型宏块,当 1、2 型宏块已经解码,则 3 型宏块周围数据都已经完成解码,这个时候可以采用预测的逆算法进行解码。在编码的时候可以用不同的位来区别 1、2、

3 型三种不同的宏块。

[0114] 图 11 中由 1 型宏块获得 3 型宏块 E 的公式为：

$$\begin{array}{l}
 \left. \begin{array}{l} A \\ C \end{array} \right\} \Rightarrow B \\
 \left. \begin{array}{l} A \\ G \end{array} \right\} \Rightarrow D \\
 \left. \begin{array}{l} G \\ I \end{array} \right\} \Rightarrow H \\
 \left. \begin{array}{l} I \\ C \end{array} \right\} \Rightarrow F
 \end{array} \Rightarrow E$$

[0116] 编码时可以在 DCT 之前,也可以在 DCT 之后,也就是说可以通过 DC/AC 系数做宏块间预测;量化过程只是为了优化编码的性能,并不一定是必须的。

[0117] 图像一般都是先存储的方式,既将整幅图像都存储后,再进行编码。但是对于单向扫描的视频则需要实时编码的方式,既文件分级存储,以宏块编码方式为例,如图 34 所示,是一个两级索引,A 点是一级存储索引,A 对应下面一个 4×4 的块的范围,这样存储的方法可以通过一级一级的细分,最终找到最关心的区域。如果点击 A 对应宏块,可以深入到下一级索引,看到更精细的部分。上面只是两级存储,根据实际需要,可以分多级的方法存储。

[0118] 如图 35 所示,是分级存储中一种三级索引的条带结构,每上一层条带对应下一层四个条带,在查找某一具体位置时,可以根据索引快速定位。

[0119] 预测算法采用加权预测的基本模式,根据像素之间的相关性进行预测,如: X 像素和 Y_i 相关,用 M_i 代表相关系数,则有如下公式: $X = \sum_{i=1}^n M_i Y_i$ 。在预测宏块中某一像素的时候通常用一个正方形区域代表一个像素,所谓相关是指在这个正方形区域内延预测方向的平行线所能穿过的相邻宏块中最靠近被预测宏块的像素。这些像素中包括带顶角和不带顶角两种情况。

[0120] 当对某一位置像素进行预测时,首先把这—个像素进行分割。分割的方法是按相邻快相关像素所在位置利用和预测方向相平行的线段把需要预测的像素分成若干块,如图 13a 所示。总有一个或两个相关像素点可以控制—定的预测像素的一个分割,并把所述一个或两个相关像素围成的平行四边形区域成为能量通道。

[0121] 得到像素加权公式为: $X = \sum_{i=1}^n M_i Y_i$;其中, X 像素和 Y_i 相关, M_i 代表相关系数,相关系数表示为: $M = p \times q$;其中 p 为距离系数,即在一个能量通道里面,相关系数点距离被预测位置占整个通道长度的比例; q 为比例系数,即这条能量通道里面的分割占整个预测像素的比例值。

[0122] 在计算 q 时,有两种思路,一种是将一个像素能量平均分布在像素所代表的小方块内, q 值不为 1;另一种是将一个像素的能量集中在像素所代表的小方块的中心点位置,这是 q 值通常为 1,但是当分割线穿过中心点时,均按 1/2 计算。

[0123] 在计算 P 时,在本发明中,均从相邻块相关像素点代表的小方块的边开始计算,即能量通道的两边为代表相关像素点的小方块的最靠近两边。当然可以以相邻像素点代表的

小方块的中心作为能量通道的两边来计算 P 值。

[0124] 其中 p 为距离系数,即在一个能量通道里面,相关系数点距离被预测位置占整个通道长度的比例。如果能量通道只有一边像素可知,则定义 p 值为 1。

[0125] q 为比例系数,即这条能量通道里面的分割占整个预测像素的比例值,为了方便计算,通常有 1/4、1/2、3/4、1/8、3/8、5/8、7/8 等。

[0126] 为了计算方便通常把相关系数的分母调成 2 的指数,如 2、4、8、16、32 等,这样在计算机处理时可以方便地使用移位操作完成计算。而且在实际计算中,为了进一步简化计算,往往根据以上计算结果,人为规定一些相关系数使计算更加简便,但是考虑的依据仍然是上面计算的结果,并取一些近似的结果加以简化,近似的方法为常用的数学方法,如取整、四舍五入等。在人为规定一些参数的时候往往考虑和相关像素距离远近,有时候不在一个能量通道里面也把相关像素引入到计算公式里面,来计算某能量通道里面的像素的预测值。

[0127] 预测算法如图 12a 至 12i,为相邻宏块中有两个对称宏块已解码时的情况,设 A、C 已解码,需要预测 B,其中,B 块为 B[1] 到 B[16],如图 12j 所示。在计算 B 块内每个点的像素的预测值时,采用加权的方法,即根据预测的方向,通过对能够影响到 B 块内该像素值的相邻块的相关像素加权得到。

[0128] 图 12a 为平均模式预测,预测公式为:

[0129]

$$B[1] = B[2] = B[3] = B[4] = B[5] = B[6] = B[7] = B[8] = B[9] = \frac{a+b+c+d+e+f+g+h}{8};$$

[0130]

$$B[10] = B[11] = B[12] = B[13] = B[14] = B[15] = B[16] = \frac{a+b+c+d+e+f+g+h}{8}; \text{或者}$$

$$[0131] \quad B[1] = B[2] = B[3] = B[4] = \frac{a+e}{2}; \quad B[5] = B[6] = B[7] = B[8] = \frac{b+f}{2};$$

$$[0132] \quad B[9] = B[10] = B[11] = B[12] = \frac{c+g}{2}; \quad B[13] = B[14] = B[15] = B[16] = \frac{d+h}{2};$$

[0133] 图 12b 为水平模式预测,影响 B[1]、B[2]、B[3]、B[4] 像素值的相邻像素点分别为 A 块中的 a 像素点和 C 块中的 e 像素点。因此 B[1]、B[2]、B[3]、B[4] 只和 a、e 两点像素有关,公式为:

$$[0134] \quad B[1] = a + \frac{(e-a)}{8} = \frac{7a+e}{8}; B[2] = a + \frac{3(e-a)}{8} = \frac{5a+3e}{8};$$

$$[0135] \quad B[3] = a + \frac{5(e-a)}{8} = \frac{3a+5e}{8}; B[4] = a + \frac{7(e-a)}{8} = \frac{a+7e}{8};$$

$$[0136] \quad B[5] = \frac{7b+f}{8}; B[6] = \frac{5b+3f}{8}; B[7] = \frac{3b+5f}{8}; B[8] = \frac{b+7f}{8};$$

$$[0137] \quad B[9] = \frac{7c+g}{8}; B[10] = \frac{5c+3g}{8}; B[11] = \frac{3c+5g}{8}; B[12] = \frac{c+7g}{8};$$

$$[0138] \quad B[13] = \frac{7d+h}{8}; B[14] = \frac{5d+3h}{8}; B[15] = \frac{3d+5h}{8}; B[16] = \frac{d+7h}{8};$$

[0139] 图 12c 为水平向下模式预测,预测公式为:

$$[0140] \quad B[1] = \frac{2a+g+f}{4}; \quad B[7] = \frac{a+2g+f}{4}; \quad B[5] = \frac{3b+2a+g+2h}{8};$$

$$B[11] = \frac{2b+a+2g+3h}{8};$$

$$[0141] \quad B[9] = B[15] = \frac{2c+b+h}{4}; \quad B[2] = B[8] = \frac{a+2f+g}{4}; \quad B[6] = \frac{3a+2b+2g+h}{8};$$

$$[0142] \quad B[12] = \frac{2a+b+3g+2h}{8}; \quad B[10] = B[16] = \frac{2b+c+h}{4}; \quad B[3] = \frac{3f+e}{4}; \quad B[4] = \frac{f+3e}{4};$$

$$[0143] \quad B[13] = \frac{c+3d}{4}; \quad B[14] = \frac{3c+d}{4};$$

[0144] 图 12d 为对角线 1 模式预测, 预测公式为:

$$[0145] \quad B[1] = \frac{7a+h}{8}; \quad B[6] = \frac{5a+3h}{8}; \quad B[11] = \frac{3a+5h}{8}; \quad B[16] = \frac{a+7h}{8};$$

$$B[5] = B[10] = B[15] = \frac{a+b}{2};$$

$$[0146] \quad B[2] = B[7] = B[12] = \frac{g+h}{2}; \quad B[3] = B[8] = \frac{f+g}{2}; \quad B[9] = B[14] = \frac{b+c}{2}; \quad B[13] = \frac{c+d}{2};$$

$$[0147] \quad B[4] = \frac{e+f}{2};$$

[0148] 图 12e 为垂直向右模式预测, 预测公式为:

$$[0149] \quad B[1] = B[10] = a; \quad B[2] = B[6] = B[11] = B[15] = \frac{a+h}{2}; \quad B[3] = B[12] = \frac{3h+g}{4};$$

$$[0150] \quad B[4] = \frac{2f+e+g}{4}; \quad B[5] = B[14] = \frac{3a+b}{4}; \quad B[7] = B[16] = h; \quad B[8] = \frac{2g+f+h}{4};$$

$$[0151] \quad B[9] = \frac{a+2b+c}{4}; \quad B[13] = \frac{b+2c+d}{4};$$

[0152] 图 12f 为垂直向左模式预测, 预测公式为:

$$[0153] \quad B[4] = B[11] = e; \quad B[3] = B[7] = B[10] = B[14] = \frac{e+d}{2}; \quad B[2] = B[9] = \frac{3d+c}{4};$$

$$B[1] = \frac{2b+a+c}{4};$$

$$[0154] \quad B[8] = B[15] = \frac{3e+f}{4}; \quad B[6] = B[13] = d; \quad B[5] = \frac{2c+b+d}{4}; \quad B[12] = \frac{e+2f+g}{4};$$

$$[0155] \quad B[16] = \frac{f+2g+h}{4};$$

[0156] 图 12g 为对角线 2 模式预测, 预测公式为:

$$[0157] \quad B[4] = \frac{7e+d}{8}; \quad B[7] = \frac{5e+3d}{8}; \quad B[10] = \frac{3e+5d}{8}; \quad B[13] = \frac{e+7d}{8};$$

$$B[8] = B[11] = B[14] = \frac{e+f}{2};$$

$$[0158] \quad B[3] = B[6] = B[9] = \frac{c+d}{2}; \quad B[2] = B[5] = \frac{b+c}{2}; \quad B[12] = B[15] = \frac{f+g}{2}; \quad B[16] = \frac{g+h}{2};$$

$$[0159] \quad B[1] = \frac{a+b}{2};$$

[0160] 图 12h 为水平向上模式预测,预测公式为:

$$[0161] \quad B[4] = \frac{2e+c+b}{4}; B[6] = \frac{e+2c+b}{4}; B[1] = \frac{3f+2e+c+2d}{8}; B[10] = \frac{2f+e+2c+3d}{8};$$

$$[0162] \quad B[12] = B[14] = \frac{2g+f+d}{4}; B[3] = B[5] = \frac{e+2b+c}{4}; B[7] = \frac{3e+2f+2c+d}{8};$$

$$[0163] \quad B[19] = \frac{2e+f+3c+2d}{8}; B[11] = B[13] = \frac{2f+g+d}{4}; B[2] = \frac{3b+a}{4}; B[1] = \frac{b+3a}{4};$$

$$[0164] \quad B[16] = \frac{g+3h}{4}; B[15] = \frac{3g+h}{4};$$

[0165] 图 12i 为 DCT 变换预测模式,先使用 DCT 变化,然后再预测,即预测是在 DCT 域内进行的。在预测时,把当前块作 DCT 变换,然后再和周围已经 DCT 变换的块进行逐像素比较,找出相似程度最大的块。然后作残差,再 DCT 编码。判断是否为最相似的块采用 SAD(绝对差和)或方差的方法。

[0166] 对于如图 12 的情况,即当前需要预测块周围有三个已知块同样采用类似的方法,如图 13a 所示;还有当前需要预测块周围四块已知,则采用同样类似的方法,如图 13b 所示。

[0167] 如图 13a 所示,由于三边相邻块的相关像素点已知,比两边多一边,但是并不是预测块中每一个像素点都需要上述三边信息,在 13a 中,用水平向下的预测模式,只有 a 到 e 第一排四个像素点和 b 到 f 第二排后两个像素点才用得相邻 B 块的相关像素点信息,其它像素点的预测和对称两相邻块已知第三边未知的情况是一样的,这里不再累述。以 13a 中 X 像素点的预测为例,延水平向下预测方向通过 X 块的平行线会穿过 a、i、j、f、g 五个相关像素点。a 到 X 的距离系数为 7/8, a 到 X 的比例系数为 1/4; i 到 X 的距离系数为 6/7, i 到 X 的比例系数为 1/2; j 到 X 的距离系数为 4/5, j 到 X 的比例系数为 1/4; f 被分为了两部分,上半部分和 j 对应,下半部分和 i 对应,上半部分到 X 的距离系数为 1/5,到 X 的比例系数为 1/4,下半部分到 X 的距离系数为 1/7,到 X 的比例系数为 1/2; g 到 X 的距离系数为 1/8,到 X 的比例系数为 1/4。

[0168] 得到 X 块的预测公式:

$$[0169] \quad X = \frac{7}{8} \times \frac{1}{4} a + \frac{6}{7} \times \frac{1}{2} i + \frac{4}{5} \times \frac{1}{4} j + \frac{1}{5} \times \frac{1}{4} f + \frac{1}{7} \times \frac{1}{2} f + \frac{1}{8} \times \frac{1}{4} g$$

[0170] 对相邻四块的预测方法和三块是一样的,只是对于一些块的预测会使用更多相关像素点的信息。

[0171] 当采用通过相邻的四块已知宏块进行预测时,水平模式预测如图 14a 所示,预测公式与已知两边情况相同;对角线 1 模式预测如图 14b 所示,预测公式为;水平向下模式预测如图 14c 所示;垂直向右模式预测如图 14d 所示;垂直模式预测如图 14e 所示;对角线 2 模式预测如图 14f 所示;水平向上模式预测如图 14g 所示;垂直向左模式预测如图 14h 所示。

[0172] 当通过相邻三块已知宏块进行预测,水平模式预测如图 15a 所示;对角线 1 模式预测如图 15b 所示;水平向下模式预测如图 15c 所示;垂直向右模式预测如图 15d 所示;垂直模式预测如图 15e 所示;对角线 2 模式预测如图 15f 所示;水平向上模式预测如图 15g 所示;垂直向左模式预测如图 15h 所示。

[0173] 当通过相邻一已知宏块进行预测时,水平模式预测如图 16a 所示;对角线 1 模式预

测如图 16b 所示 ;水平向下模式预测如图 16c 所示 ;垂直向右模式预测如图 16d 所示 ;垂直模式预测如图 16e 所示 ;对角线 2 模式预测如图 16f 所示 ;水平向上模式预测如图 16g 所示 ;垂直向左模式预测如图 16h 所示。

[0174] 下面介绍结合图 17 说明如何确定已知相邻块中各个像素点对当前需要预测像素点的影响,也就是在公式中如何确定系数。如果要预测 X 块,采用的预测方式是水平向下,则和 X 像素点所在块相邻块的相关像素点主要有, A、B、C、D 四像素点 ;判断的方法为,延 X 像素点边缘作预测方向的平行线,刚好经过的相邻块的点为相关像素点。显然,如果把和预测方向平行的 ab 直线再往上移,将不通过 X 像素点,同样,把和预测方向平行的 gh 直线再往下移,将不通过 X 像素点。这样就可以确定 ag 范围为一边的相关范围,bh 为另一边的相关范围。

[0175] 得出 ag 正好落在 A 和 B 两个相邻块的相邻像素点内,bh 正好落在 C 和 D 两个相邻块的相关相邻像素点内,因此得出在水平向下的模式下,和 X 点相关的相邻块的相关像素点有 A、B、C、D 四像素点。ab 和 ef 所构成的区域通过 X 像素点,并占了 X 像素点四分之三的面积 ;即 A 像素点和 C 像素点对 X 影响要占影响作用的四分之三 ;ef 和 gh 所构成的区域通过 X 像素点,并占了 X 像素点四分之一的面积 ;即 B 像素点和 D 像素点对 X 影响要占影响作用的四分之一。再从空间上来考虑,A 与 X 块的中点距离 :m ;C 与 X 块中点距离 :n,即 : m : n = 3 : 5,A 距离上影响作用占八分之五,C 距离上影响作用占八分之三,影响作用正好成反比,同理 B 和 D 也符合上面比例。

$$[0176] \quad \text{因此 : } X = \frac{3}{4}A \times \frac{5}{8} + \frac{1}{4}B \times \frac{5}{8} + \frac{3}{4}C \times \frac{3}{8} + \frac{1}{4}D \times \frac{3}{8} = \frac{15A + 5B + 9C + 3D}{32}$$

[0177] 在实际的应用中,通常会采用比较简单的计算公式来简化算法复杂度。

[0178] 在预测时,可以使用顶角像素,如图 19 所示,A、a、J、j 就是顶角像素,即正方形四个顶角对应的像素,8×8 带顶角信息的对角线 1 模式在计算公式 中,需要进行一些处理 ;被预测的模块,可以是 2×2、4×4、8×8、16×16 等结构块,也可以是条带,如图 18a 或 18b 所示,或其他形状。

[0179] 在预测的时候,以块的边缘作为能量集中的区域,即把一个像素看作一个小正方形,在预测的时候把正方形的和预测区域相邻的边作为能量集中的区域。然后用现有的预测方法来对未知区域进行预测的方法,是前面提到过的预测方法,公式就是前面给出的那个总公式,这种方法不需要顶角像素。下面提到预测方法不是把能量集中到和预测区域的相邻边,而是放在现有像素的中心点,即把能量集中到中心点,如图 23 所示,然后去预测未知区域的方法。这种方法需要顶角像素,如图 19 所示。详细的算法推理见图 24 所示,以对角线 1 模式为例进行 4×4 的预测,中心区域为需要预测的块。按预测方向用平行线连接个像素中心点,然后再用平行线平分这些能量区域,形成如图 26a、图 26b、图 26c 中的 a、b、c、d、e、f、g、h、i 等能量通道。同一能量通道里面的能量变化是线性的。在对角线模式下,很容易计算,如果能量通道只用一端已知,另一端未知,则假定这段能量通道内能量恒定,这样就可以预测能量通道中的预测模块内的相关像素点。如果能量通道两端已知,即两边像素值已知,则采用线性的方法计算出中间预测模块内需要预测的像素点对应的位置的值即可。如在 e 通道内,如果两端像素值已知(线段两端点已知),则需要求这条线段上其余四个点 e₁、e₂、e₃、e₄ 的值。

$$[0180] \quad e_1 = m + \frac{(n-m)}{5}; e_2 = m + \frac{2(n-m)}{5}; e_3 = m + \frac{3(n-m)}{5}; e_4 = m + \frac{4(n-m)}{5};$$

[0181] 如图 20a 和图 20b 所示, 1、2、3 型宏块的基础上, 增加一种边缘宏块, 即已经到了整个画面的边界部分, 在图 20 中用 B 表示, 这个时候, 可以采用图 15 或图 16 的预测方法来对边界宏块进行预测编码。在图 20 中, 当选择以中心区域为中心开始显示的时候, 显示的顺序是由中心往周围扩散的, 这样的方法, 最容易找到用户关心的区域。

[0182] 如图 21, 解码的次序为 1、2、3、4、5、6、7...。当然根据不同的算法有不同的次序, 这个解码次序和前面不一样, 但是基本规律是一样的, 先传 1 型宏块, 把 1 型宏块区域看作已知区域 A, 再传已知区域 A(1 型宏块周围) 的 1 型宏块, 再解码可以解码的所有 2、3 型宏块, 然后再把所解码的区域看作已知区域 A, 然后再传已知区域周围的 1 型宏块, 然后再解码可解码的 2、3 型宏块; 以此往复, 如图 22 所示。

[0183] 如图 23 所示, 以中心往边缘扩散的方法有多种, 图 21 展示了一种常见的, 正方形扩展算法, 算法描述: 传 1 型宏块, 看作小正方形区域; 各边步长加一, 如果没有搜索到可以传输解码的块; 各边步长再加一; 如果有则解码; 然后把得到的已经解码的正方形区域作为基本的正方形; 再重复上面步骤, 各边步长加一, 如图 24 所示。

[0184] 如图 25- 图 27 所示, 为能量集中到像素所代表的正方形中心的情况。在划分能量通道的时候, 把相邻的像素代表的能量点用直线连接起来, 取连接两相邻像素能量点线段的中点作平行于预测方向的直线, 这样就可以划分出不同的能量通道。在预测时, 如果一个当前预测的像素在某一个能量通道里面时, 则通过如图 26 公式计算出当前所预测的像素值。如果一个像素同时在两个能量通道里面, 即所预测的像素代表的正方形中心被分割能量通道的直线穿过时, 则将两个不同的预测值通过加权得到当前预测值, 如各取二分之一之后相加得到。如果采用能量集中到一点这种计算方法时, 像素加权公式中比例系数 q 值为 1。

[0185] 本发明的存储和传输可以是两个独立完成的过程, 例如月球表面图像数据, 先把它分块后统一编号, 如果按图 22 的顺序编号(按行从左到右编号), 并假设整体数据是保存在 8×8 的区域内。很容易发现存储是按一种的顺序, 从左到右或从右到左, 从上到下或从下到上, 或一些特有的数学排列的顺序, 这和提取数据的顺序是完全不一样的, 在提取数据时, 可能从中间的某一位置开始按一定的顺序或设定的算法提取。

[0186] 在图像存储的时候可能是采用非预测编码的原数据, 也可以是采用预测编码时候的数据。需要注意的是, 预测编码的目的主要是为了节省在网络上传输时候带宽的问题。所以, 可以在需要网络传输的时候再编码, 当然可以先编码之后存储然后再在网络上传输。

[0187] 当需要传输以区域 (28) 为中心的图像数据时, 如果是传输的时候才预测编码的话, 可规定中心为 1 型宏块进行编码, 如果是已经预测编码好之后在传输则, 找离中心 28 最近的 1 型宏块进行传输; 还是以正方形扩散为例来说明, 首先, 找 28 周围的 1 型宏块, 需要先传输 10、12、14、26、30、42、44、46 这些 1 型宏块; 然后需要做的是, 如何计算出上面对应的这些块的序号, 公式如下: 设整个图像存储在 $m \times n$ (m 为行, n 为列) 的区域的宏块内, 而且 X 为 28, 为中心也为 1 型宏块, Y 为需要再传输的 1 型宏块, 舍最大块为 MAX, 最小块编号为 1:

$$[0188] \quad Y_1 = X-2n-2; Y_2 = X-2n; Y_3 = X-2n+2; Y_4 = X-2; Y_5 = X+2; Y_6 = X+2n-2; Y_7 =$$

$X+2n ; Y_8 = X+2n+2 ;$

[0189] 最后,再继续往周围扩散的时候,把 Y_1 到 Y_8 来替换现有的 X 即可,如,以 Y_1 为例再往下计算: $Z_1 = Y_1-2n-2 ; Z_2 = Y_1-2n ; Z_3 = Y_1-2n+2 ; Z_4 = Y_1-2 ; Z_5 = Y_1+2 ; Z_6 = Y_1+2n-2 ; Z_7 = Y_1+2n ; Z_8 = Y_1+2n+2。$

[0190] 这样就可以计算出下一层需要传输的 1 型宏块。通过这样递归的方法,就可以把需要显示的区域不断无间隔地扩大。

[0191] 计算中已经计算过的,提取了就不用再提取了,如果发现小于 1 的就舍去,如果发现大于最块值块 MAX 的也舍去。对于条带式比较简单,虽然也是整体存储,但是提取的时候传输 I 条带,然后再传输 B 条带就好了,因为 I 条带之间的间隔是固定的或者 I 条带位置可以计算得到,所有很容易计算其位置,也方便提取和传输。

[0192] 主要用于巨大的图像传输,由于使用流式结构,可以提供网络上一些实时的业务,如电子商务等。对于一些只需要单向增加数据的地方,比如扫描系统,监控监测系统等,以扫描系统为例:由于已经拍摄过的影像不需要修改,而且需要对正在进入镜头的一部分数据进行存储或编码传输,如果是这样情况,实际上也只需要对正在进入镜头的数据部分进行编码。这样做可以在节省编码资源也便于实时传送。

[0193] 如图 28 所示,在存储的时候既可以统一存储在一个地方也可以分散在多个地方存储,把图像信息分散存储到源 1 到源 n 中,当 X 端发起对指定图像区域的请求时,可以同源 1 到源 n 中包含指定区域信息的源向 X 端响应,并向 X 端传输指定区域图像信息。可以采用 P2P 作为分散传输的底层协议。

[0194] 本发明还提供了一种基于流格式的图像内显示方法,其中以 I 条带来实现对显示速度的条带序列如图 29 所示,当显示端向图像服务端发起显示请求时,可以包含显示速度信息,如每个时间单位内发送的内容包含 5 个 I 条带,这样,当服务端接收到这条消息后,就会调整自身的发送速度,保证按显示端的需求的速度发送。要达到这个目的,还必须先将 I 条带或带有显示速度信息的条带均匀或非均匀的分布在这个图像中,并且有相应的序列编号。

[0195] 当需要在图像内做跳转时,即从当前显示条带位置跳到某一个设定条带位置时,只需要知道需要跳转的条带序列编号即可。如图 29 所示,需要跳转到 n 位置,则向服务端发送跳转信息和跳转的位置信息,则马上就可以看到 n 位置的图像信息。如果显示端接收的图像信息的速度比显示图像的速度快,则可以将接收的图像信息存储在显示端,在按照显示速度进行显示。

[0196] 如图 30 所示,对于宏块的显示,同样也可以使用类似条带的方法控制显示的速度和显示的位置。图 30 中,宏块显示了一个“E”的图案,可以用设定条带显示位置的方案来生成所需要显示的图案。前面说的宏块的扩散算法也是显示位置的一种特定算法,是位置显示控制的一个实例。在图 30 中,在显示端的位置控制模块中向服务端发送位置信息:【1, 2】、【1, 3】、【1, 4】、【1, 5】、【2, 2】、【3, 2】、【3, 3】、【3, 4】、【4, 2】、【5, 2】、【5, 3】、【5, 4】、【5, 5】。显示端就可以显示出预先设计出的“E”图案。

[0197] 如图 31 所示,服务端和显示端的系统结构图,包括服务端和显示端。其中,服务端包括服务控制模块、编码模块、消息模块、存储模块和网络模块,消息模块中进一步包括速度控制子模块和位置控制子模块;其中,服务控制模块用于根据用户需求对服务端进行相

关控制操作 ; 编码模块用于对需要传输的信息进行编码 ; 消息模块中的速度控制子模块用于控制图像传输的速度、位置控制子模块用于控制图像现实的位置 ; 存储模块, 用于存储相关信息, 网络模块, 用于与显示端通信。显示端包括显示模块、解码模块、消息模块、存储模块和网络模块, 消息模块中进一步包括速度控制子模块和位置控制子模块 ; 消息模块中的速度控制子模块用于控制图像显示的速度、位置控制子模块用于控制图像显示的位置 ; 存储模块用于存储显示图像的相关信息 ; 解码模块用于对接收信息进行解码 ; 显示模块用于显示图像。

[0198] 如图 32 所示, 表现出用协议的方式改变或确定显示位置的消息过程和改变显示速度的消息过程。具体包括 : 显示端通过向服务端发送显示图像的位置信息 ; 服务端向显示端反馈接受 OK ; 服务端向显示端发送确定图像内需要传送的位置, 并传送 ; 显示端向服务端发送显示的速度信息 ; 服务端向显示端反馈接受 OK ; 服务端向显示端发送改变现有发送速度信息, 并发送。本图中只画出了最重要的消息。如图 32, 当使用 TCP 作为网络的底层协议可以很好的通过消息来控制, 但是使用 UDP 时和 TCP 不同, 是一种尽力传输模式, 可能会出现显示的速率和传输的速率不匹配的情况, 比如, 显示的速率过快, 因此在非实时要求的情况下, 可以考虑先把传输到显示端的数据缓存一部分到本地, 显示时直接从本地取出数据即可, 以减小由于网络性能不稳定给显示端带来的影响。或直接采用传输和控制分离的方法, 即传输使用 UDP 控制层 TCP, 分别采用不同的端口传输。利用现有的协议族就是 RTP/RTCP 协议。

[0199] 如图 33 所示, 在实时传输过程中, 以条带传输为例, 服务端利用图像成像设备, 如扫描仪, 扫描需要传输的信息。由于扫描的光电器件长度固定, 扫描宽度随时间不断增加, 所获图像信息也越多。这种情况下, 可以选用条带的方式来进行编码和预测, 条带的长度为扫描光电器件的长度, 在图中为 a 边长度, 宽度随时间变化, 以单位时间内扫描仪扫描的宽度为条带的宽度, 途中为 b 边长度。这样当扫描仪扫描某一个位置时, 可以对这个位置所在的区域进行编码, 图中为当前编码条带所代表位置。然后将当前编码条带实时传输到显示端进行显示。如果要对当前编码条带进行预测编码, 则等到下一编码条带信息已知以后再对当前编码条带进行编码。解码时, 只有 I 型条带可以直接解码, 其它条带必须根据前后的 I 型条带的或已经解码的条带进行解码。这样就可以完成条带编码的实时传输和实时显示。在显示端, 为了观看到最新传输过来的图像信息, 通常是把图像做滚动处理, 在 33 图中, 将图像往左边不断滚动, 滚动的速度和传输的速度一致, 图像的一端会在左边最边缘处消失, 消失的速度也和传输的速度保持一致, 这样可以保证我们观看的图像尺寸不会发生变化。

[0200] 为了平衡编码效率, 解码时间和实时这三方关键因素, 适当的在条带中设置 I 型条带的位置就非常关键, 通常采用均匀的部署模式, 如每隔一个条带设置一个 I 型条带, 或每个两个条带设置一个 I 型条带等。

[0201] 本发明实施例所描述的技术可以用硬件、软件、或组合执行。如果用软件执行, 则该技术可以直接指包含程序代码的计算机可读介质, 该程序代码在对视频序列进行编码的设备中执行, 进行该实施例中的一个或多个残差编码技术。在该种情况下, 计算机可读介质可以包括 RAM(随机存储器)、SDRAM(同步动态随机存储器)、ROM(只读存储器)、NVRAM(非易失性随机存储器)、EEPROM(电可擦除只读存储器)、FLASH(闪存) 等。

[0202] 程序编码可以以计算机可读指令的形式存储在存储器中。在该情况下, 一个或多

个处理器可以执行存储在存储器中的指令,从而执行一个或多个残差编码技术。在一些情况下,处理器可以用 DSP(数字信号处理)设备执行,DSP 使用各种硬件元件来加速编码处理;在其它情况下,编码设备可以作为一个或多个微处理器,一个或多个或者多个 ASIC(专用集成电路)、FPGA(现场可编程门阵列)、或一些其它等效集成或分立逻辑电路或硬件软件组合来执行。

[0203] 以上公开的仅为本发明的几个具体实施例,但是,本发明并非局限于此,任何本领域的技术人员能思之的变化都应落入本发明的保护范围。

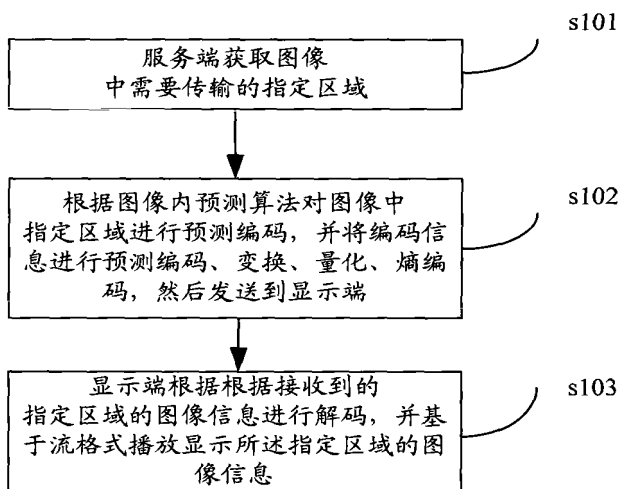


图 1

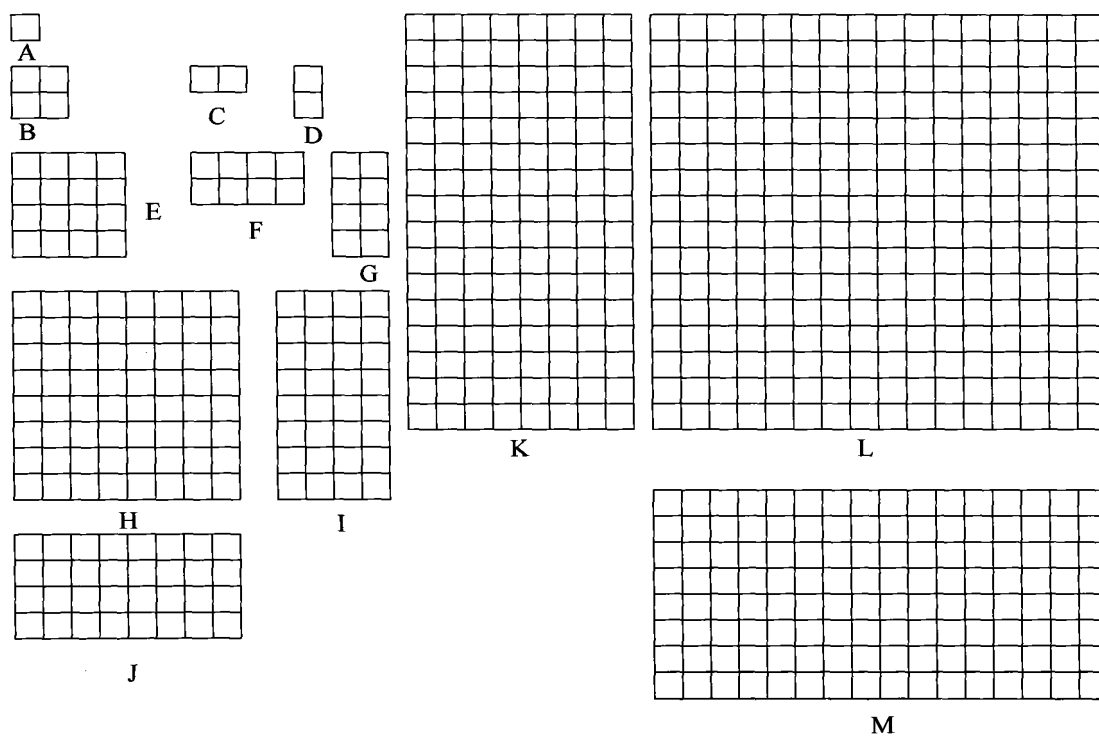


图 2



图 3

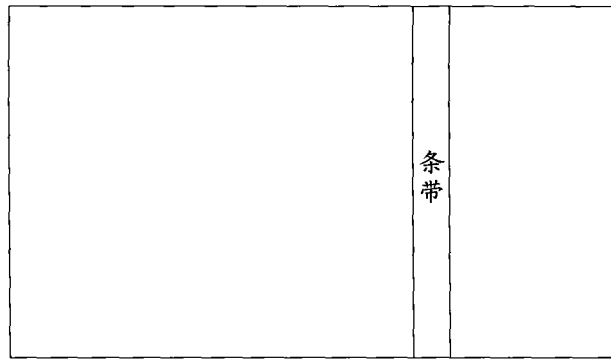


图 4

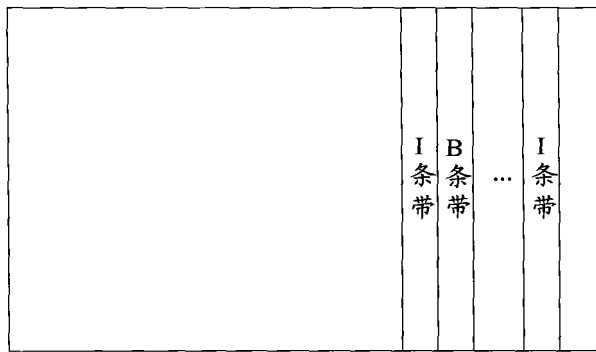


图 5

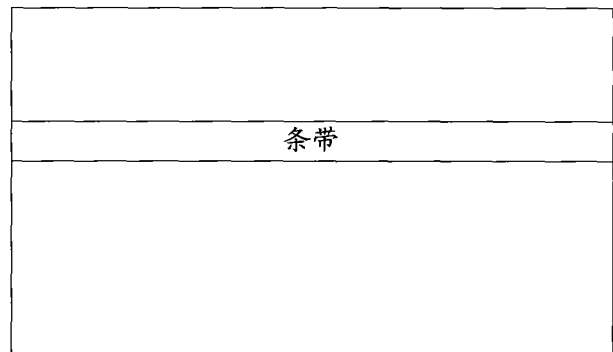


图 6

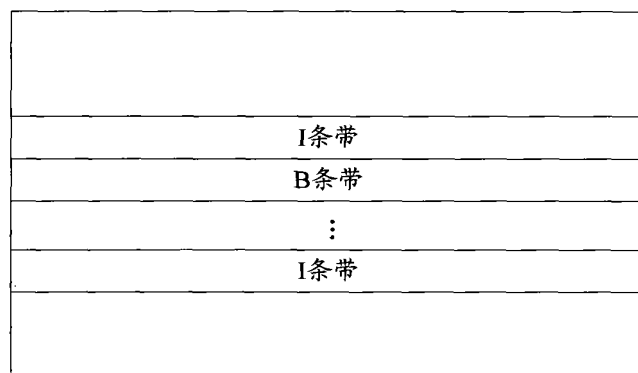


图 7

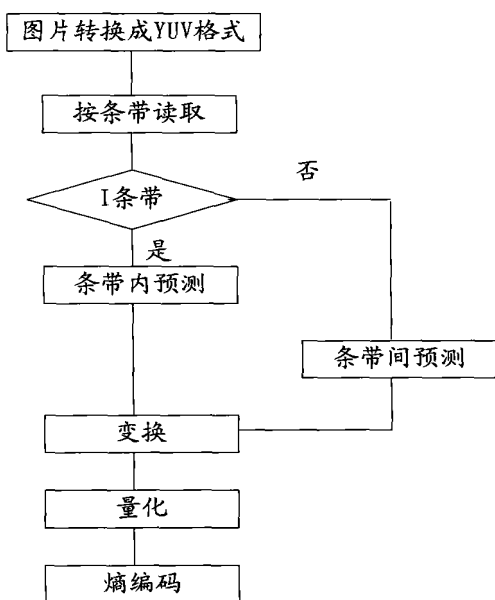


图 8

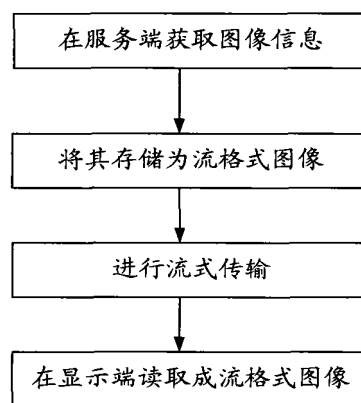


图 9

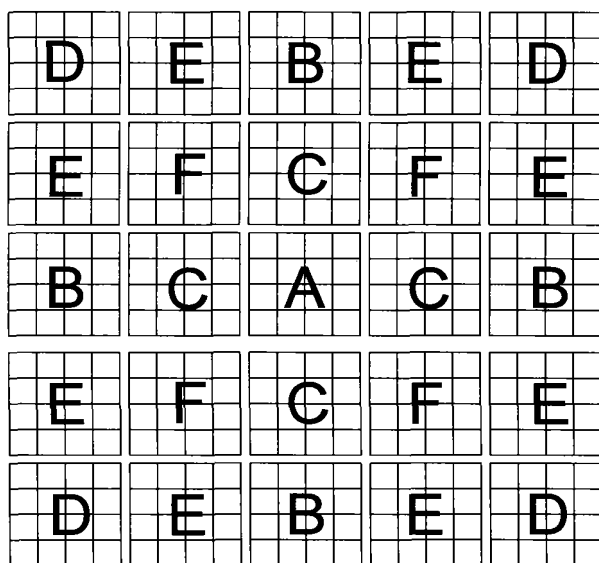


图 10

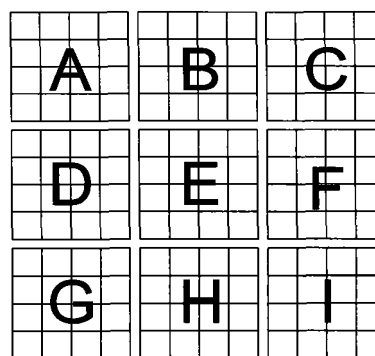


图 11

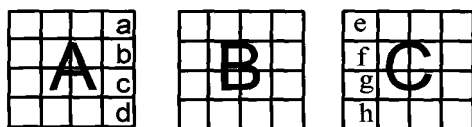


图 12a

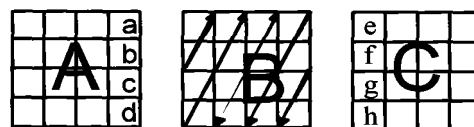


图 12f

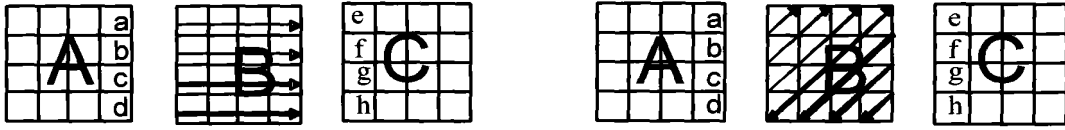


图 12b

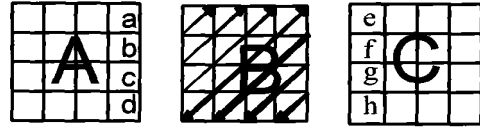


图 12g

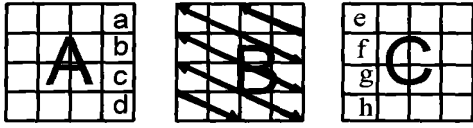


图 12c

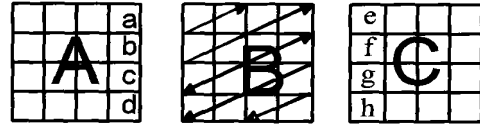


图 12h

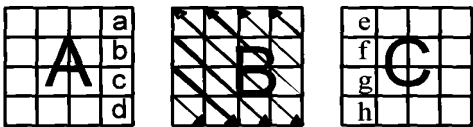


图 12d

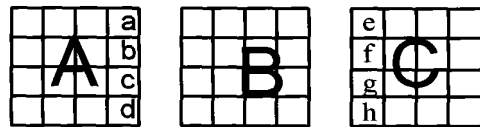


图 12i

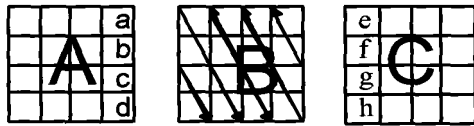


图 12e

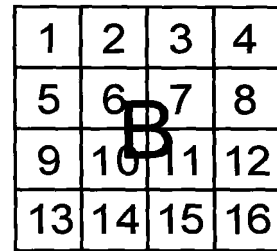


图 12j

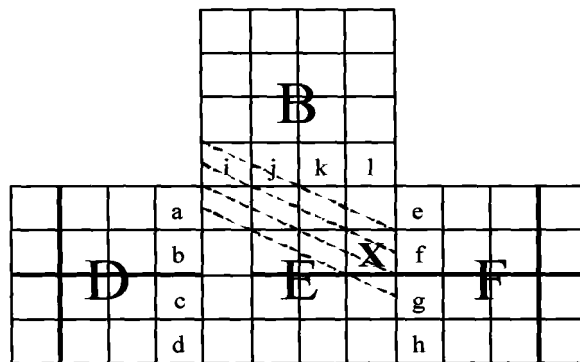


图 13a

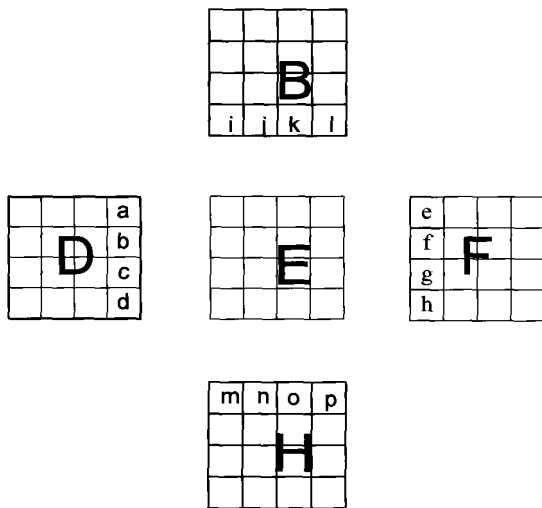


图 13b

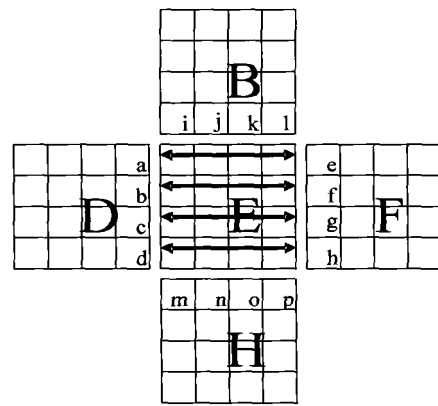


图 14a

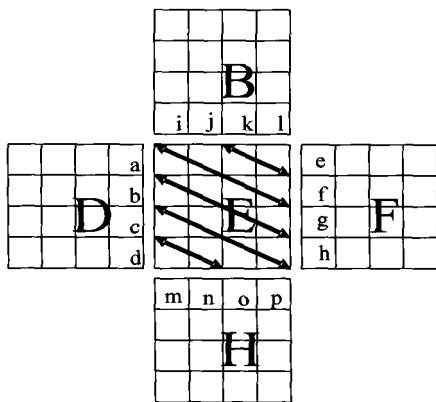


图 14c

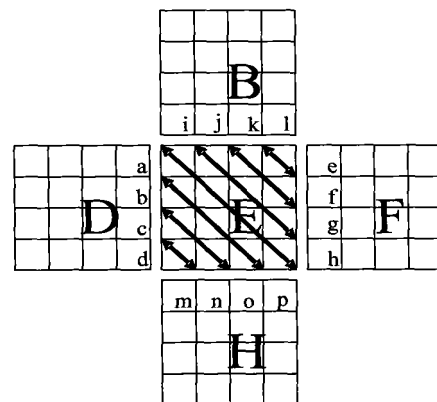


图 14b

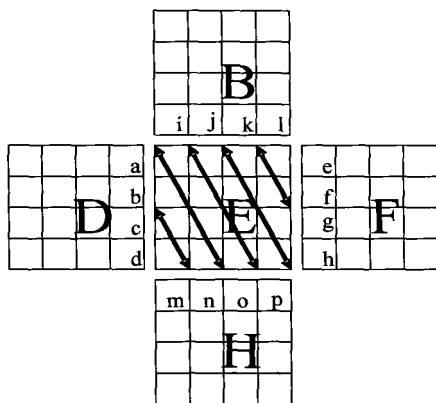


图 14d

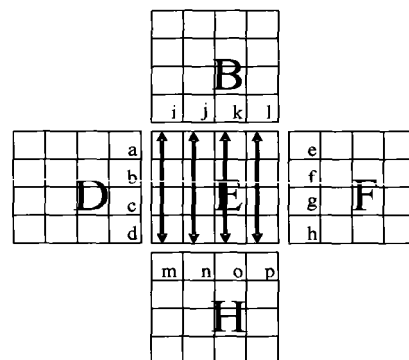


图 14e

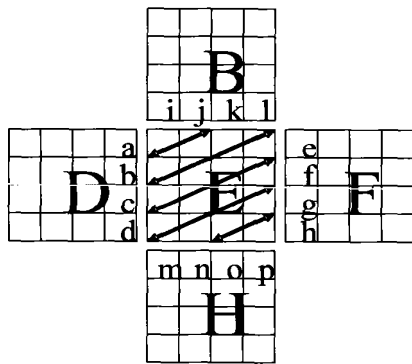


图 14g

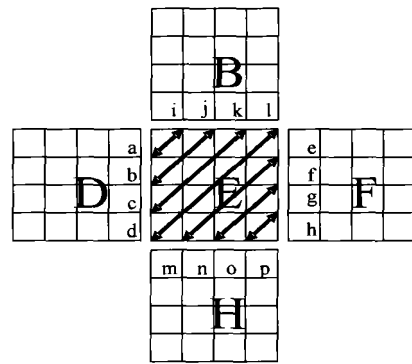


图 14f

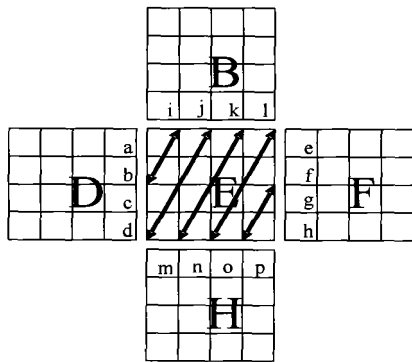


图 14h

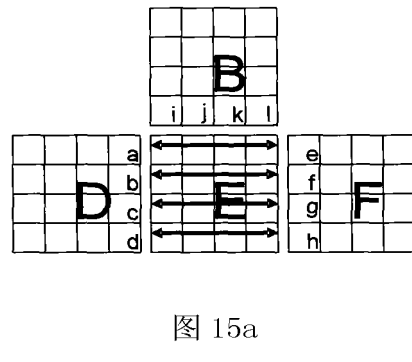


图 15a

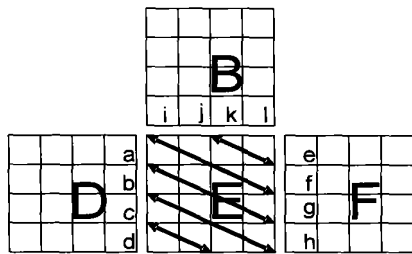


图 15c

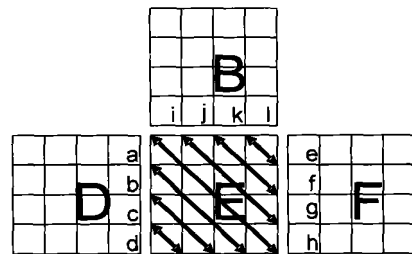


图 15b

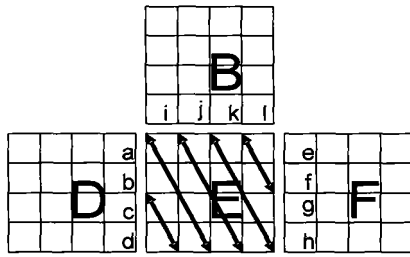


图 15d

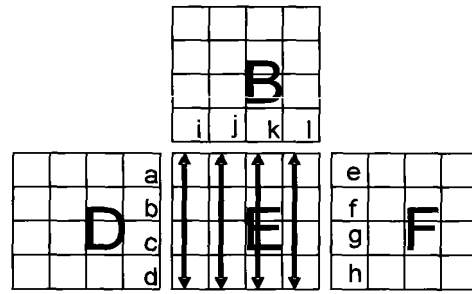


图 15e

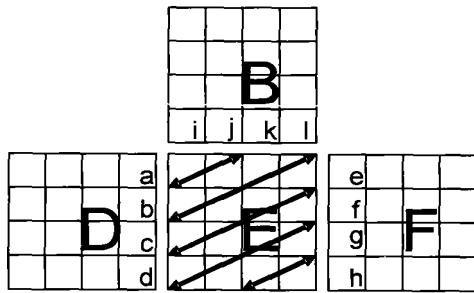


图 15g

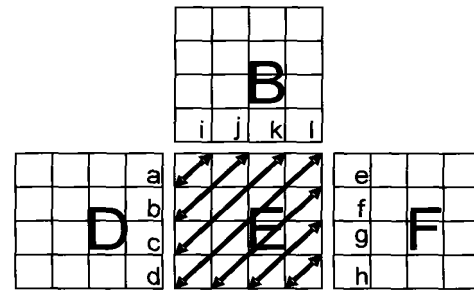


图 15f

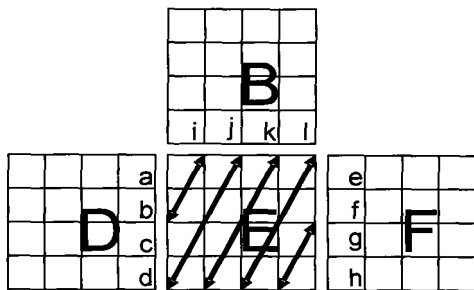


图 15h

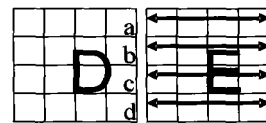


图 16a

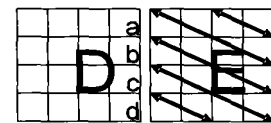


图 16c

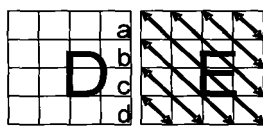


图 16b

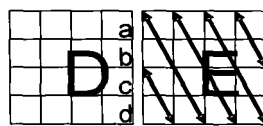


图 16d

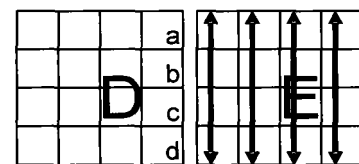


图 16e

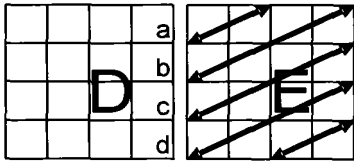


图 16g

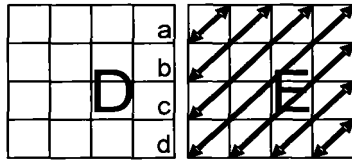


图 16f

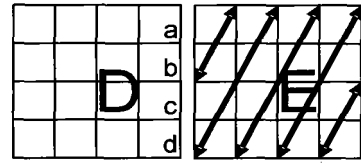


图 16h

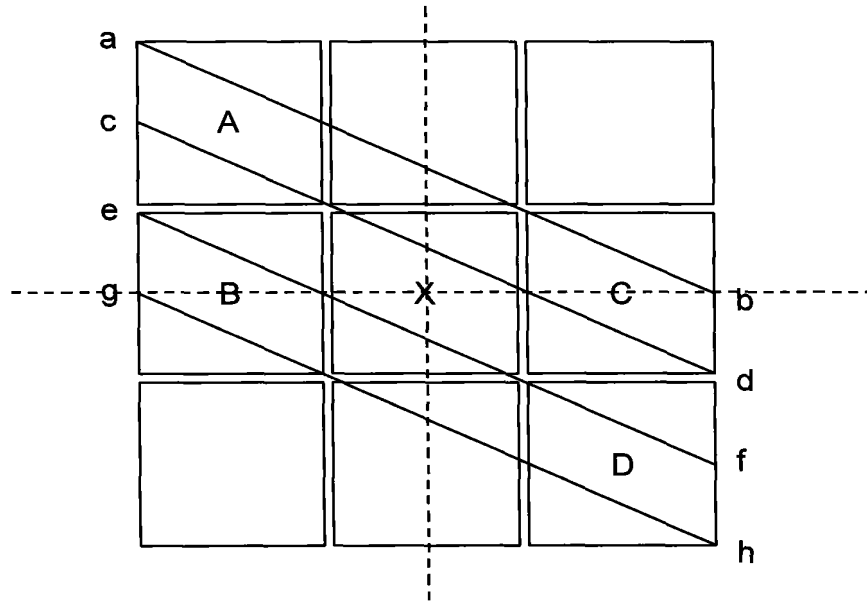


图 17

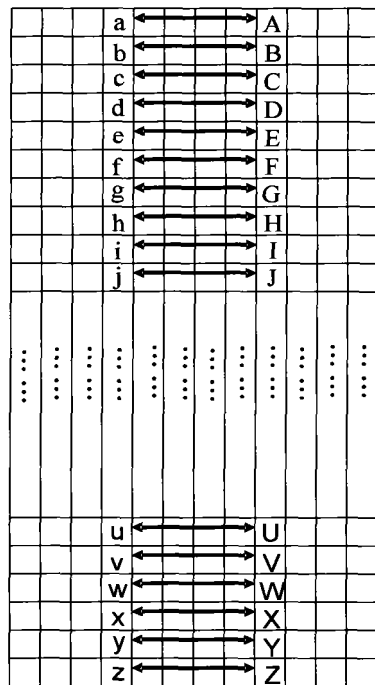


图 18a

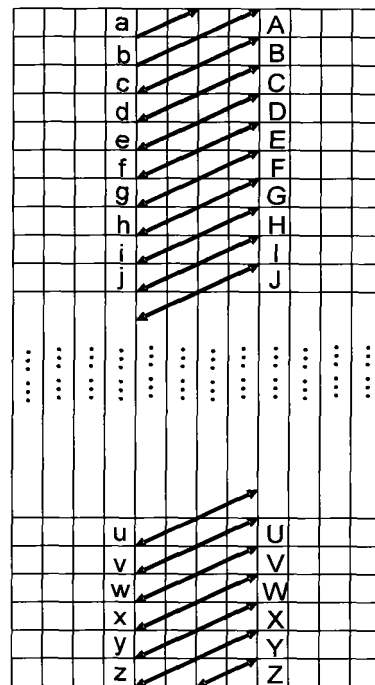


图 18b

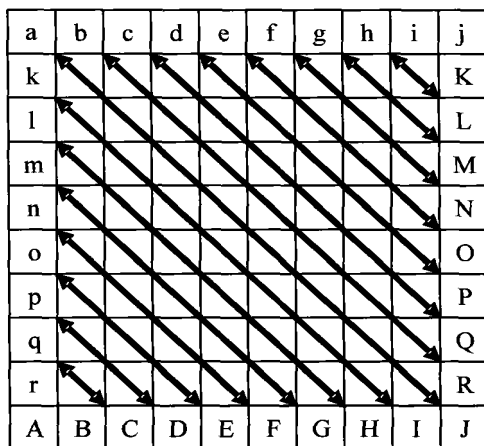


图 19

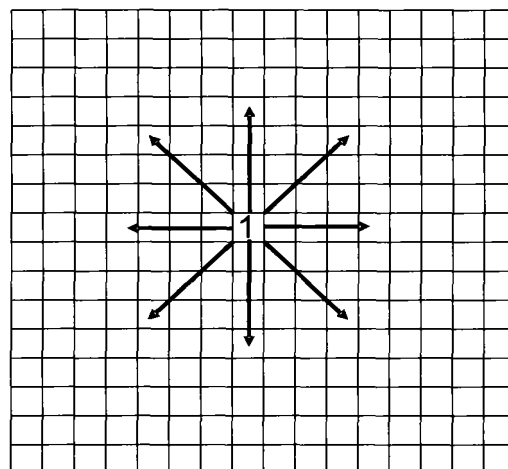


图 20a

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| B | B | B | B | B | B | B | B | B | B | B | B | B | B | B | B |
| B | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| B | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 |
| B | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| B | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 |
| B | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| B | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 |
| B | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| B | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 |
| B | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| B | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 |
| B | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| B | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 |
| B | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |
| B | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 | 3 | 2 |
| B | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 | 2 | 1 |

图 20b

| | | | | | | | | | | | | | | | |
|--|--|--|---|---|---|---|---|---|---|---|---|--|--|--|--|
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | 5 | 6 | 5 | 6 | 5 | 6 | 5 | 6 | 5 | | | | |
| | | | 6 | 7 | 6 | 7 | 6 | 7 | 6 | 7 | 6 | | | | |
| | | | 5 | 6 | 2 | 3 | 2 | 3 | 2 | 6 | 5 | | | | |
| | | | 6 | 7 | 3 | 4 | 3 | 4 | 3 | 7 | 6 | | | | |
| | | | 5 | 6 | 2 | 3 | 1 | 3 | 2 | 6 | 5 | | | | |
| | | | 6 | 7 | 3 | 4 | 3 | 4 | 3 | 7 | 6 | | | | |
| | | | 5 | 6 | 2 | 3 | 2 | 3 | 2 | 6 | 5 | | | | |
| | | | 6 | 7 | 6 | 7 | 6 | 7 | 6 | 7 | 6 | | | | |
| | | | 5 | 6 | 5 | 6 | 5 | 6 | 5 | 6 | 5 | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

图 21

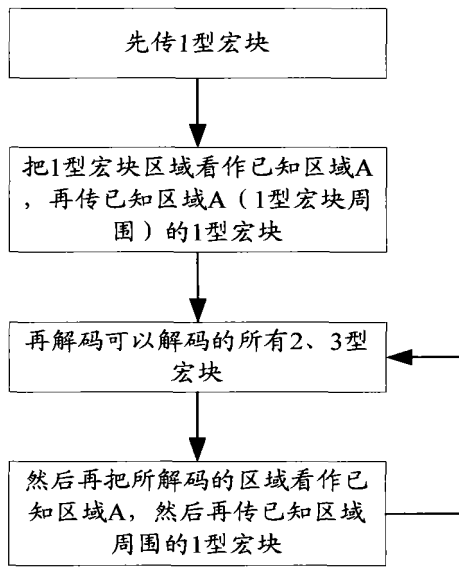


图 22

| | | | | | | | |
|----|----|----|----|----|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 |
| 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 |
| 25 | 26 | 27 | 28 | 29 | 30 | 31 | 32 |
| 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 |
| 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 |
| 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 |
| 57 | 58 | 59 | 60 | 61 | 62 | 63 | 64 |

图 23

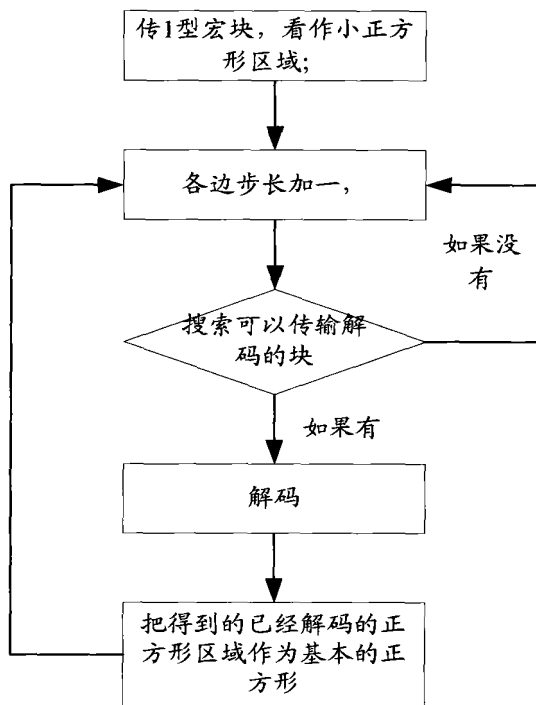


图 24

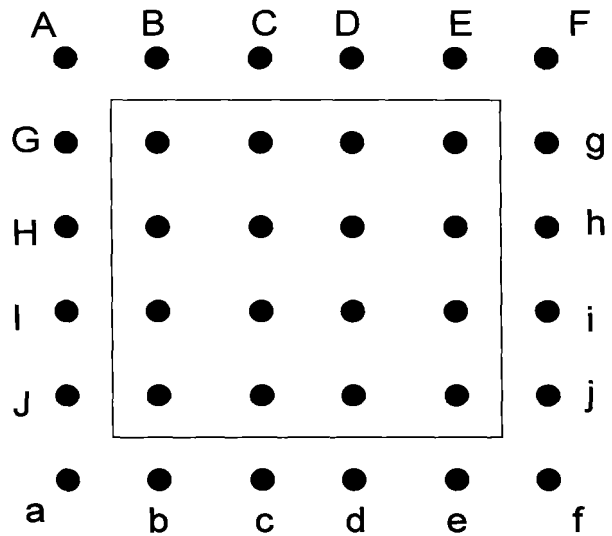


图 25

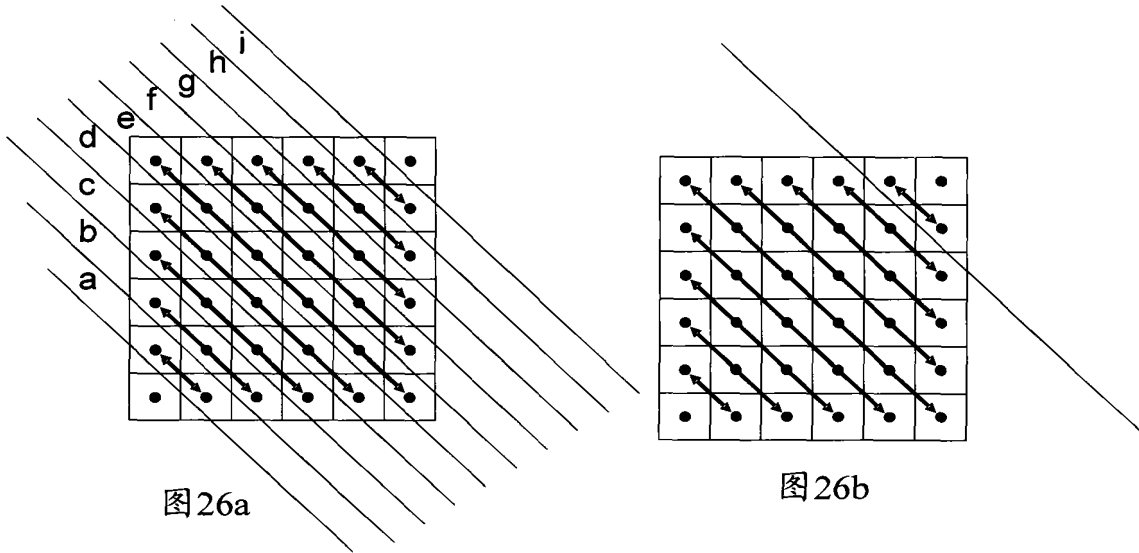


图26a

图26b

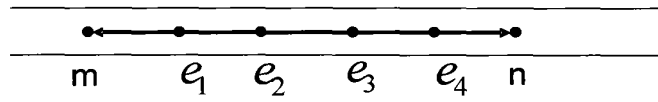


图26c

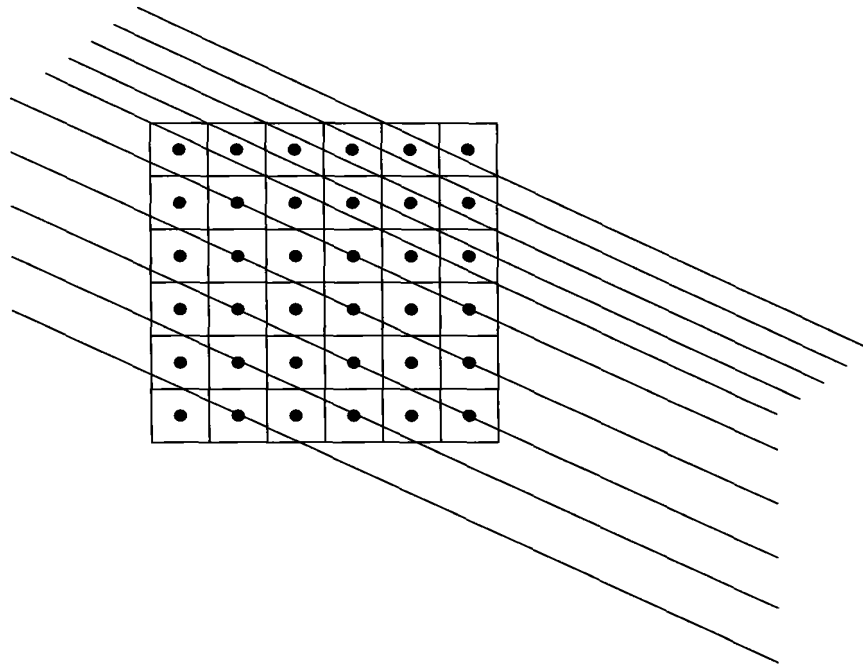


图 27

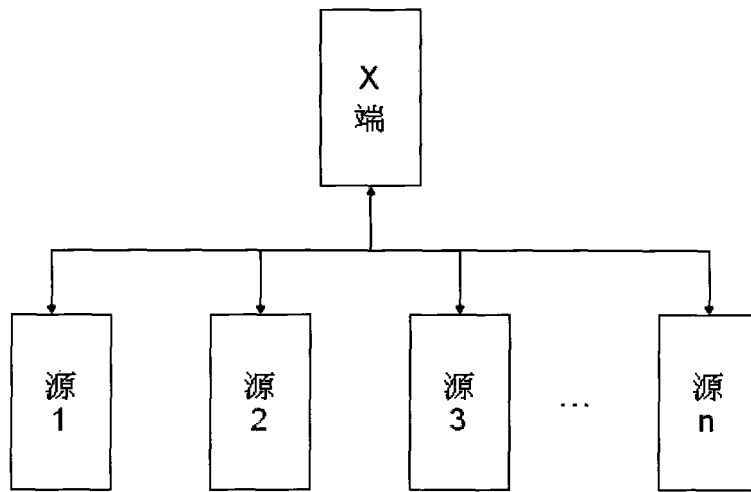


图 28

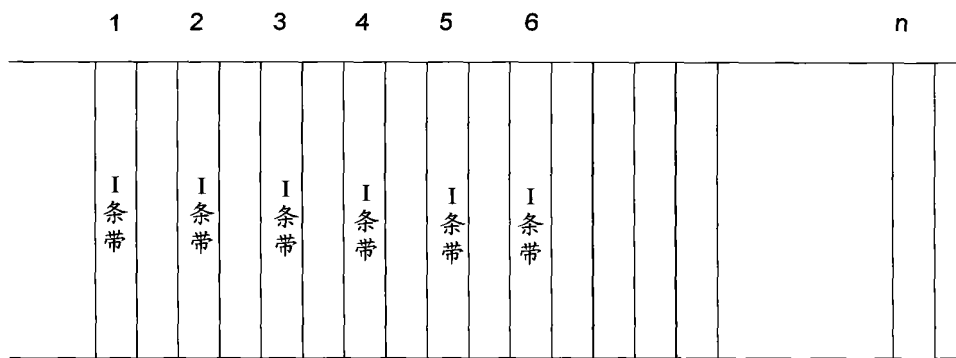


图 29

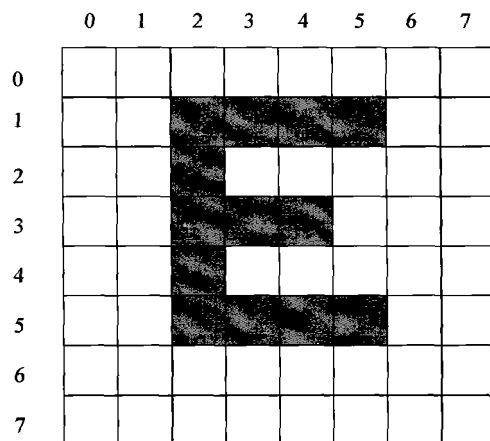


图 30

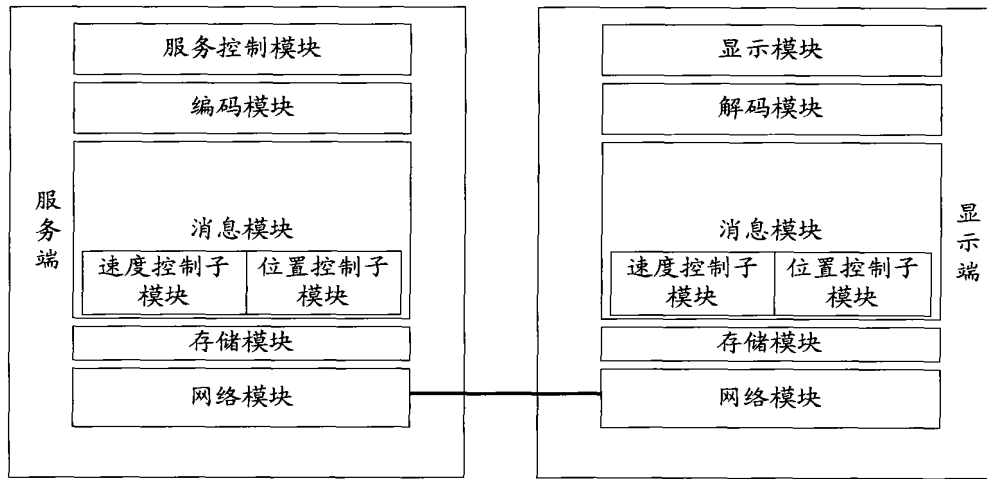


图 31

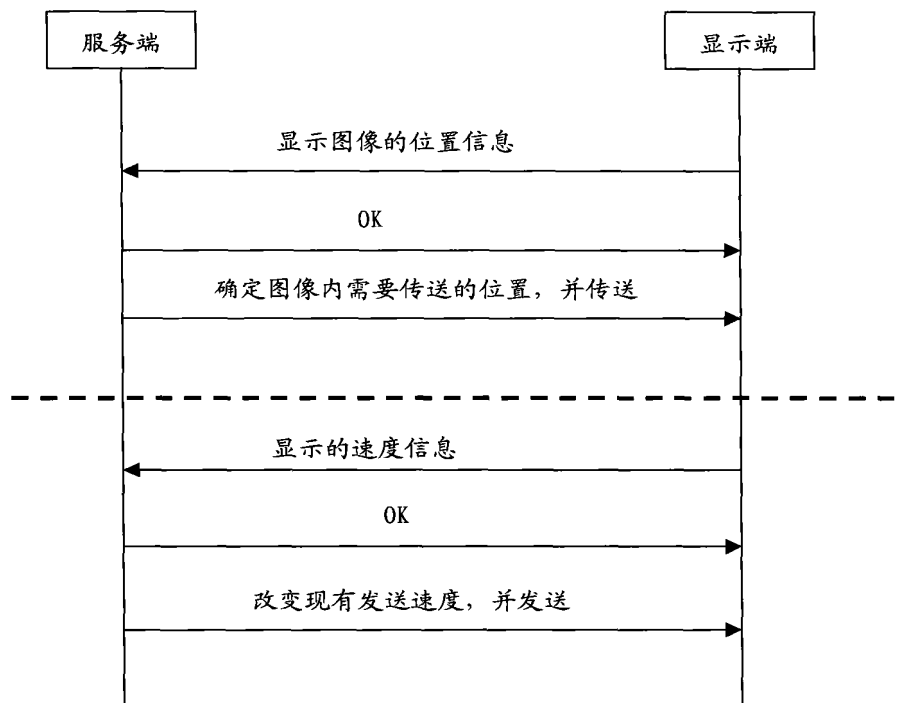


图 32

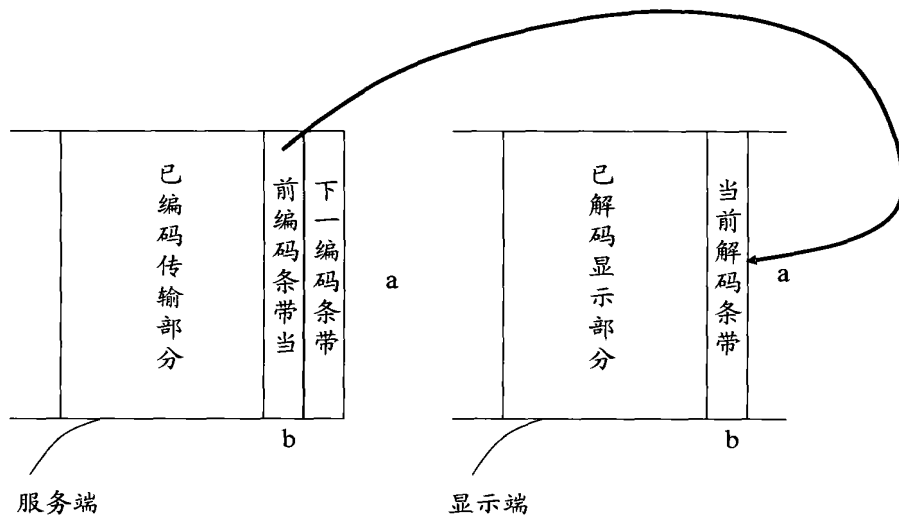


图 33

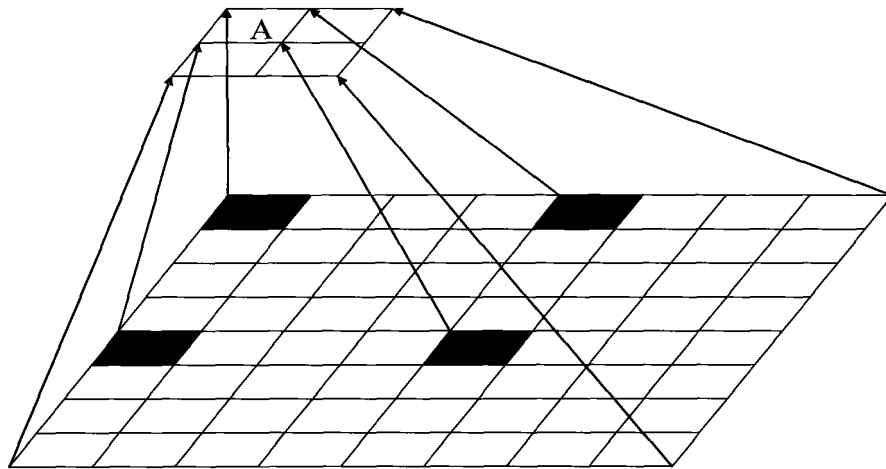


图 34

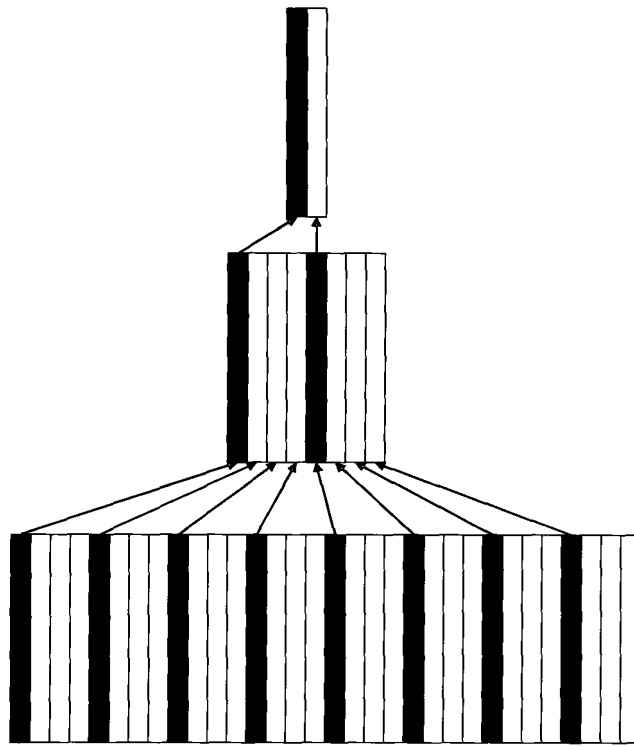


图 35