



(12) 发明专利

(10) 授权公告号 CN 105844609 B

(45) 授权公告日 2020.10.16

(21) 申请号 201511035853.X

(22) 申请日 2015.11.18

(65) 同一申请的已公布的文献号
申请公布号 CN 105844609 A

(43) 申请公布日 2016.08.10

(30) 优先权数据
14306826.0 2014.11.18 EP

(73) 专利权人 达索系统公司
地址 法国韦利济-维拉库布莱

(72) 发明人 L·佩蒂厄 A·勒纳尔
F·莱策尔特

(74) 专利代理机构 永新专利商标代理有限公司
72002
代理人 张扬 王英

(51) Int.Cl.
G06T 7/11 (2017.01)

(56) 对比文件

CN 102067173 A, 2011.05.18
US 2006227292 A1, 2006.10.12
CN 102542099 A, 2012.07.04
CN 104103061 A, 2014.10.15
CN 102270299 A, 2011.12.07
CN 103400365 A, 2013.11.20
CN 102246208 A, 2011.11.16
US 4956906 A, 1990.09.18
US 8411115 B1, 2013.04.02
US 2013177246 A1, 2013.07.11
JP 2010257315 A, 2010.11.11
US 8315479 B1, 2012.11.20
JP H09120278 A, 1997.05.06

审查员 田子茹

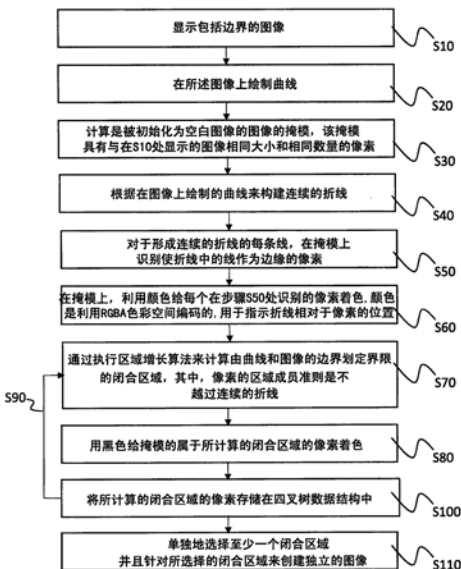
权利要求书2页 说明书11页 附图10页

(54) 发明名称

对图像进行划分

(57) 摘要

本发明涉及一种用于对图像进行划分的计算机实现的方法。该方法包括显示(S10)包含边界的图像;在所述图像上绘制(S20)曲线;计算由所述曲线和所述图像的边界划定界限的所有闭合区域,每个闭合区域独立于其它计算的闭合区域,并且是单独地能够选择的。



1. 一种用于对图像进行划分的计算机实现的方法,包括下列步骤:

- 显示 (S10) 包括边界的图像;
- 在所述图像上绘制 (S20) 曲线;
- 根据在所述图像上绘制的所述曲线来构建 (S40) 连续的折线;
- 针对形成所述连续的折线的每条线,识别 (S50) 所显示的图像的、使所述每条线作为边缘的像素,从而创建了由位于所述折线的两侧的两个像素构成的所述折线的边界;
- 通过使用区域增长算法来计算由所述折线和所述图像的所述边界划定界限的所有闭合区域以及由所述折线单独地划定的所有闭合区域,其中,像素的区域成员准则是不越过所述连续的折线,各个闭合区域彼此独立并且是单独地能够选择的。

2. 根据权利要求1所述的计算机实现的方法,其中,所述构建 (S40) 所述连续的折线的步骤包括:

- 计算链接在所述图像上绘制的所述曲线的两个不同端点的像素集,其中,所述像素集中的邻近像素具有公共的边缘;
- 构建所述连续的折线,其中,所述折线中的每条线是所计算的像素集中的所述像素中的一个像素的边缘。

3. 根据权利要求2所述的计算机实现的方法,其中,在识别 (S50) 所述显示的图像的、使所述每条线作为边缘的所述像素之后,标记每个识别的像素,所述标记指示折线相对于所述每个识别的像素的位置。

4. 根据权利要求3所述的计算机实现的方法,其中,所述标记每个识别的像素的步骤是通过利用颜色来对每个识别的像素着色 (S60) 来执行的,所述颜色是利用RGBA色彩空间来进行编码的:

- 第一RGBA值指示折线位于所述像素的右侧;
- 第二RGBA值指示折线位于所述像素的左侧;
- 第三RGBA值指示折线位于所述像素的上侧;
- 第四RGBA值指示折线位于所述像素的底侧。

5. 根据权利要求4所述的计算机实现的方法,其中,对每个像素进行着色是累积的。

6. 根据权利要求5所述的计算机实现的方法,其中,所述区域增长算法包括下列步骤:

- (i) 选择所述图像的属于第一闭合区域的第一像素;
- (ii) 通过使用选择的像素上的所述标记来识别选择的第一像素的满足所述准则的邻近像素;
- (iii) 将在步骤 (ii) 处识别的每个邻近像素添加到要被选择的并且被当做所述选择的第一像素的像素的列表中;
- (iv) 将所述选择的第一像素标记为选中的,并且将所述选择的第一像素添加到属于所述第一闭合区域的像素的列表;
- (v) 在要被选择的所述像素的列表中选择第二像素,并且针对所述第二像素执行所述步骤 (ii) 至 (iv);
- (vi) 重复所述步骤 (v),直至要被选择的所述像素的列表中的所有像素已经被选择为止。

7. 根据权利要求6所述的计算机实现的方法,还包括:

(vii) 识别所述图像的属于第二闭合区域的第三像素,所述第三像素在步骤(iv)处未被标记;

(viii) 执行所述步骤(ii)至(vi);

(ix) 重复所述步骤(vii)至(viii),直至所述图像的所有像素被标记为选中的为止。

8. 根据权利要求1至7中的一项所述的计算机实现的方法,其中,从所述构建连续的折线的步骤开始执行的步骤是在掩模上执行的,其中,所述掩模是具有与所述显示的图像相同大小和相同数量的像素的图像。

9. 根据权利要求7所述的计算机实现的方法,从所述构建连续的折线的步骤开始执行的步骤是在掩模上执行的,其中,所述掩模是具有与所述显示的图像相同大小和相同数量的像素的图像,并且其中,所述掩模被初始化为空白图像,并且被标记为选中的所述像素在被添加到属于闭合区域的像素的列表时用黑色着色。

10. 根据权利要求1至7或9中的一项所述的计算机实现的方法,其中,计算的闭合区域被存储在四叉树数据结构中。

11. 根据权利要求1至7或9中的一项所述的计算机实现的方法,在计算所有闭合区域的步骤之后,进一步包括下列步骤:

- 单独地选择至少一个闭合区域;

- 针对选择的每个闭合区域来创建独立的图像。

12. 一种计算机可读存储介质,其具有被记录在其上的计算机程序,所述计算机程序包括在被计算机执行时使所述计算机执行权利要求1至11中任一项所述的方法的指令。

13. 一种用于对图像进行划分的系统,其包括被耦合至存储器和图形用户界面的处理器,所述存储器具有被记录在其上的计算机程序,所述计算机程序包括在被所述系统执行时使所述系统执行权利要求1至11中任一项所述的方法的指令。

对图像进行划分

技术领域

[0001] 本发明涉及计算机程序和系统领域,并且更具体地,涉及一种用于对图像进行划分的方法、系统和程序。

背景技术

[0002] 被用于处理图像的大多数绘图软件提供用户用于提取图像的子部分的功能。通常地,用户在对图像进行划分时必须执行以下操作。首先,其给其正在寻找的子部分的分界线划定界限;其可以绘制自由形状或基本形状(矩形、圆形……)。然后,基于用户的绘图,软件试图将图像划分为两个不同的部分:位于绘图内部的第一部分,以及位于其外部的第二部分。

[0003] 在由用户绘制的自由形状的特定情况下,软件通过将第一点链接至最后一点来使绘图闭合。为了确信绘图在图像中限定“内部”部分和“外部”部分,这是必需的。这在图1和图2上示出。在图1中,由用户绘制的自由形状10不是闭合的,因为在自由形状的起始点和其终点之间没有连接。如图2上所示,链路20是由软件创建的,用于获得闭合的自由形状(其是分界线)。通过这种方式,用于对图像进行划分的已知的方法能够将图像分离成两个不同的区域:第一区域在分界线的内部,以及第二区域在分界线的外部。然后,在用户确认该分离后(有时其可能是自动的),只保留内部部分。该选择有时可以变得相反,以保留外部部分,但是并非总是如此。仅选择图像的子部分。为了根据该选择创建另一个图像,或将其或别的东西去除,用户必须做另一个动作。例如,为了利用其选择创建另一个图像,用户必须创建新图像,并且然后进行复制粘贴,或使其选择反转并且去除不需要的部分。

[0004] 用于对图像进行划分的已知的解决方案包括若干限制。第一限制是用户不能总是精确地限定其想要的子部分。这是因为,在不考虑绘图的形状的情况下(例如,如果其已经限定了闭合曲线),软件仅允许其使用基本形状(例如,矩形)来指定分界线,或软件将第一点链接到最后一点以创建闭合分界线。如图3和图4上所示的,这导致坏的结果。在图3中,用户已经在图像上绘制了自由形状,该自由形状是闭合的自由形状。然而,在闭合的自由形状的起点与终点之间不存在连接,以至于软件在这两个点之间创建链路,因此导致如图4上所示的第二闭合区域的创建。因此,用户的意图未被考虑。

[0005] 另一个缺陷是由软件保留的子部分总是内部的子部分。在一些程序中,用户能够做一些操作来使其反转,并且保留外部部分而不是内部部分,但是这并不总是可行的。

[0006] 进一步的缺陷是对若干子部分的选择是耗时的并且乏味的。这在以下的场景中示出,其中,用户从事图像工作,并且想要从其中提取两个不同的部分。利用现有的软件,其必须遵循这些步骤:导入原始图像,通过小心地绘制闭合形状来限定第一子部分,在新图像中复制/粘贴该部分(或使选择反转,并且去除未被选择的部分),再次导入原始图像,通过小心地绘制闭合形状来限定第二子部分,在新图像中复制/粘贴该部分(或使选择反转,并且去除未被选择的部分)。

[0007] 又一个缺陷是,如图5和图6上所示的,如果用户想要简单地将图像切成两半,其将

必须在其想要的部分的周围画圆。这并不直观、耗时，而且不是人机控制的。

[0008] 另外的缺陷是对子部分的选择可能是不可预知的，因为已知的解决方案依赖于对用户绘制的闭合区域的或基本形状的“内部”和“外部”的检测。然而，对自由形状的“内部”和“外部”的识别对于当前的算法可能是复杂的任务；尤其是，当自由形状包括若干交点时，算法可能不能够识别自由形状的哪部分是内部的和外部的。而且，对闭合区域的“内部”和“外部”的检测对于计算机系统是耗时的，因为该检测对于当前算法不简单。因此，大量的计算资源被用于可能是不可预知的结果。

[0009] 在该上下文内，仍然存在针对用于将图像划分成一个或多个能够选择的子部分的改进的方法的需求。

发明内容

[0010] 因此，提供了一种用于对图像进行划分的计算机实现的方法。所述计算机实现的方法包括下列步骤：

[0011] -显示包括边界的图像；

[0012] -在所述图像上绘制曲线；

[0013] -计算由所述曲线和所述图像的边界划定界限的所有闭合区域，每个闭合区域独立于其它计算的闭合区域，并且是单独地能够选择的。

[0014] 所述计算机实现的方法还可以包括：

[0015] -在所述图像上绘制所述曲线的步骤之后，根据在所述图像上绘制的所述曲线来构建(S40)连续的折线的步骤；针对形成所述连续的折线的每条线，识别所显示的图像的、使所述每条线作为边缘的像素的步骤；

[0016] -构建所述连续的折线的步骤包括：计算链接在所述图像上绘制的曲线的两个不同端点的像素集，其中，集合中的邻近像素具有公共的边缘；构建连续的折线，其中，所述折线中的每条线是所计算的像素集中的像素中的一个像素的边缘；

[0017] -在识别所显示的图像的像素的步骤之后，标记每个识别的像素的步骤，所述标记用于指示折线相对于所述每个识别的像素的位置；

[0018] -标记每个识别的像素的步骤是通过利用颜色来对每个识别的像素着色来执行的，所述颜色是利用RGBA色彩空间来进行编码的：第一RGBA值指示折线位于所述像素的右侧；第二RGBA值指示折线位于所述像素的左侧；第三RGBA值指示折线位于所述像素的上侧；第四RGBA值指示折线位于所述像素的底侧；

[0019] -对每个像素的着色是累积的；

[0020] -计算由所述曲线和所述图像的边界划定界限的所有闭合区域的步骤是通过使用区域增长算法来执行的，其中，像素的区域成员准则是不越过连续的折线；

[0021] -所述区域增长算法包括下列步骤：(i) 选择所述图像的属于第一闭合区域的第一像素；(ii) 通过使用所选择的像素上的标记来识别所选择的第一像素的满足所述准则的邻近像素；(iii) 将在步骤(ii)处识别的每个邻近像素添加到要被选择的并且被当做第一选择的像素的像素的列表中；(iv) 将所选择的第一像素标记为选中的，并且将所选择的第一像素添加到属于第一闭合区域的像素的列表；(v) 在要被选择的像素的列表中选择第二像素，并且针对第二像素执行步骤(ii)至(iv)；(vi) 重复步骤(v)，直至要被选择的像素的列

表的所有像素已经被选择为止；

[0022] - (vii) 识别所述图像的属于第二闭合区域的第三像素，第三像素在步骤 (iv) 处未被标记；(viii) 执行步骤 (ii) 至 (vi)；(ix) 重复步骤 (vii) 至 (viii)，直至所述图像的所有像素被标记为选中的为止；

[0023] - 从构建连续的折线的步骤开始执行的步骤是在掩模上执行的，其中，所述掩模是具有与所显示的图像相同大小和相同数量的像素的图像；

[0024] - 所述掩模被初始化为空白图像，并且被标记为选中的像素在被添加到属于闭合区域的像素的列表时用黑色来着色；

[0025] - 所计算的闭合区域被存储在四叉树数据结构中；

[0026] - 在计算所有闭合区域的步骤之后，单独地选择至少一个闭合区域的步骤；针对选择的每个闭合区域创建独立的图像的步骤。

[0027] 还提供了一种计算机程序，其包括用于执行上面的方法的指令。

[0028] 还提供了一种计算机可读存储介质，其具有被记录在其上的所述计算机程序。

[0029] 还提供了一种系统，其包括被耦合至存储器和图形用户界面的处理器，所述存储器具有被记录在其上的所述计算机程序。

附图说明

[0030] 现在将通过非限制性例子，并且参考附图来描述本发明的实施例，其中：

[0031] - 图1至图6示出了用于对图像进行划分的已知的方法的例子；

[0032] - 图7是示出了本发明的例子的流程图；

[0033] - 图8是示出了在本发明中使用的传播算法的例子的流程图；

[0034] - 图9是示出了依据根据本发明划分的图像来创建新图像的操作的流程图；

[0035] - 图10示出了由用户在图像上绘制的曲线；

[0036] - 图11至图13示出了根据本发明的根据曲线来计算折线的例子；

[0037] - 图14至图21示出了根据本发明对闭合区域的计算；

[0038] - 图22示出了用于执行本发明的系统的例子。

具体实施方式

[0039] 参照图7的流程图，提出了一种用于对图像进行划分的计算机实现的方法。对图像进行划分意味着图像被划分成可分离的至少两个部分。该方法包括显示包括边界的图像的步骤。然后，曲线被绘制在显示的图像上。然后，计算由曲线和图像的边界划定界限的所有闭合区域。每个计算的闭合区域是独立于另一计算的闭合区域的，并且是单独地能够选择的。

[0040] 本发明的方法提供了更高效的对图像的划分。实际上，闭合区域的数量（即，子部分的数量）是仅根据由用户绘制的曲线和图像的边界计算的，并且其无需曲线的起始点与终点之间的链接。有趣的是，图像的边界在计算闭合区域的数量时被使用。用这样的方式，如果用户想要将图像切成两半，则其只需要绘制一条从图像的顶部到底部的线。因为图像的边界在计算闭合区域的数量时被使用，所以存在至少一个闭合区域——当由用户绘制的曲线不包括交点时的完整的图像——至若干个闭合区域，这取决于其绘图中的闭合形状的

数量。因此,因为不存在对“内部”、“外部”的确定,所以虽然需要较少的计算资源来计算闭合区域,但是用户的意图未被更改。在描述中将讨论进一步的优点。

[0041] 该方法是计算机实现的。这意味着该方法的步骤(或基本上所有的步骤)是由至少一台计算机或任意类似的系统执行的。因此,该方法的步骤可能完全自动地或半自动地由计算机执行。在例子中,对该方法的步骤中的至少一些步骤的触发可以通过用户—计算机的交互来执行。需要的用户—计算机交互的级别可能取决于预见的自动化的级别,并且促使与对于实现用户的意愿的需求的平衡。在例子中,该级别可以是用户定义的和/或预定义的。

[0042] 例如,在图像上绘制曲线的步骤(S20)是在进行用户动作时触发的。单独地选择至少一个闭合区域的步骤(S110)还可以在进行用户动作时执行。

[0043] 该方法的计算机实现的典型的例子是利用适合于该目的的系统来执行该方法。该系统可以包括被耦合至存储器和图形用户界面(GUI)的处理器,该存储器具有被记录在其上的计算机程序,该计算机程序包括用于执行该方法的指令。该存储器还可以存储数据库。该存储器是适合于这样的存储的任何硬件,其可能包括若干物理不同的部件(例如,一个部件用于程序,以及可能的一个部件用于数据库)。

[0044] 就“数据库”来说,其意指被组织用于搜索和取回的数据(即,信息)的任意汇总。当被存储在存储器上时,数据库允许由计算机进行快速搜索和取回。数据库实际上被构造以便于结合各种数据处理操作来对数据进行存储、取回、修改和删除。数据库可以包括能够被分解成记录的文件或文件集合,所述记录中的每个记录包括一个或多个域。域是数据存储的基本单元。用户可以主要通过查询来取回数据。使用关键词和排序命令,用户可以迅速地对许多记录中的域进行搜索、重新排列、分组和选择,以根据正被使用的数据库管理系统的规则来取回或创建关于特定数据聚合的报告。

[0045] 该方法通常地操纵图像。图像可以是二维的。图像可以是三维的,例如,在计算机辅助设计(CAD)系统中的三维场景中示出的三维建模对象的表示。在本发明中,术语图像指的是数字图像,其是图像的数字表示。如本领域中已知的,数字表示涉及图像的数据是用二进制数字的集合来表示的。图像可以是,但不限于光栅图像或矢量图像。光栅图像被存储为图片元素(像素)的有限集,每个像素表示数字值。矢量图像被存储为基于数学表达式的几何图元的集合,以表示图像。

[0046] 当被显示时,光栅图像和矢量图像被表示为像素集。为了该目的,矢量图像在要被显示前被转换成光栅图像;这被称为光栅化。像素是图像中最小的个体元素。像素保存图像的一个具体点处的表示色彩系统中给定颜色的亮度的量化值。在实践中,图像是由被计算机系统执行的图像浏览器软件显示的。图像浏览器软件能够实现本发明的方法,或该方法能够用独立的计算机程序来实现。

[0047] 光栅图像或矢量图像或利用像素表示的任何其它类型的图像的像素集通常形成矩形的像素网格,其中,每个像素也是矩形的。每个像素与邻近像素连接。如本领域中已知的,像素连接是在其中二维或三维图像中的像素与其邻居关联的方式。例如,如果图像的像素是矩形,则每个像素具有四个边缘,并且与邻近像素共享一个或多个边缘。

[0048] 像素网格不限于矩形网格,并且可以是六边形网格或错缝接合的矩形网格,其中,如本领域中已知的,一个像素可以具有多达六个的邻近像素。在随后的描述中,仅为了解

释,将讨论利用矩形像素的矩形网格来表示图像的最常见的情况。

[0049] 图22示出了用于执行本发明的方法的系统的例子。该系统通常是计算机,例如,个人计算机。图22的计算机包括被连接至内部通信总线(BUS) 1000的中央处理单元(CPU) 1010、也被连接至BUS的随机存取存储器(RAM) 1070。该计算机还被提供有图形处理单元(GPU) 1110,其与被连接至BUS的视频随机存取存储器1100相关联。视频RAM 1100在本领域中还被称为帧缓冲器。大容量存储设备控制器1020管理对诸如硬盘驱动器1030的大容量存储设备设备的访问。适合于有形地体现计算机程序指令和数据的大容量存储设备包括所有形式的非易失性存储器,举例来说,包括诸如EPROM、EEPROM和闪存设备的半导体存储设备;诸如内部硬盘和可移动盘的磁盘;磁光盘;以及CD-ROM磁盘1040。前述中的任何一种可以由专门设计的ASIC(专用集成电路)补充,或被并入其中。网络适配器1050管理对网络1060的访问。该计算机还可以包括诸如光标控制设备、键盘等的触觉设备1090。光标控制设备被用在计算机中以允许用户选择性地将光标放置在显示器1080上的任何所期望的位置处。另外,光标控制设备允许用户选择各种命令和输入控制信号。光标控制设备包括用于向系统输入控制信号的多个信号生成设备。通常,光标控制设备可以是鼠标,被用来生成信号的鼠标按钮。替代地或另外地,该计算机系统可以包括感应板和/或感应屏。

[0050] 本发明可以由计算机程序来实现。计算机程序包括由计算机可执行的指令,指令包括用于使上面的系统执行所述方法的单元。程序可能可记录于任意数据存储介质上,包括系统的存储器。程序可以例如用数字电子电路、或用计算机硬件、固件、软件或用其组合来实现。程序可以被实现为装置,例如,被有形地体现在机器可读存储设备中用于由可编程处理器执行的产品。方法步骤可以由可编程处理器来执行,所述可编程处理器用于执行指令的程序以通过对输入数据进行操作并且生成输出来执行所述方法的功能。处理器因此可以是可编程的,并且被耦合以从数据存储系统、至少一个输入设备和至少一个输出设备接收数据和指令,以及向其发送数据及指令。应用程序可以用高级过程或面向对象的编程语言、或用汇编或机器语言(如果需要的话)来实现。无论如何,语言可以是编译或解释语言。程序可以是完全安装程序或更新程序。无论如何,系统上的程序应用产生用于执行所述方法的指令。

[0051] 参考回图7,在步骤S10处,图像是例如由所述系统显示的。显示图像意指像素集被表示在显示设备上,例如图22的显示器1080上。显示图像的过程是如本领域中已知的来执行的。图像包括边界。图像的边界是其像素连接被部分地履行的图像的像素集。例如,如果矩形像素具有三个或更少的邻近像素,则所述像素属于图像的边界。对图像的显示可以在进行用户动作时(例如,在用户对图像进行选择后)触发。

[0052] 接下来,在步骤S20处,曲线是例如由用户绘制在步骤S10处显示的图像上的。术语曲线意指与线段类似的对象,但是其无需是直的;线段是由两个不同的端点约束的线的一部分,并且包含其端点之间的线上的每个点。在本发明中,术语曲线与自由形状这一表述同义。或者,能够在图像上绘制预定的形状。可以使用任何预定的形状。

[0053] 曲线的绘制是由用户执行的。其如本领域中已知地进行。例如,用户可以使用诸如鼠标、触敏显示器上的触笔的触觉设备。

[0054] 在图像上绘制曲线这一表述涉及用户与图像的代表进行交互(例如,利用在其上表示图像的计算机屏幕),并且图像不被修改(例如,图像的像素不被着色)。向用户示出的

曲线的表示是暂时的,并且被重叠到图像的表示上。因此,本领域技术人员还能说曲线被绘制在图像上。

[0055] 现在参考图10,其示出了已经由用户绘制的曲线:由用户绘制的曲线的表示100被重叠到未被修改的图像上。图像的边界是用黑色纯色线表示的。有趣的是,本领域技术人员能够注意到,曲线的一些部分在图像的外部。这是可能的,因为用户在计算机屏幕上而非直接在图像上绘制曲线。或者,图像可以被放置在比图像更大的画布(未示出)上。用户可以将曲线(全部地或部分地)绘制在图像的外面;对曲线的绘制不受图像的边界的限制。

[0056] 然后,在步骤S30处,掩模是例如由系统计算的。掩模通常在用户停止绘制曲线时被计算。掩模是具有与所显示的图像相同大小和相同数量的像素的图像。例如,如果在步骤S10处显示的图像具有 $(x \times y)$ 个像素的大小,并且包括 $n = x \times y$ 个像素,那么掩模被计算,使得其具有相同的 $(x \times y)$ 个像素的大小和相同数量的 n 个像素。也就是说,掩模具有与在步骤S10处显示的图像相同的大小和分辨率。

[0057] 在掩模的一个像素与图像的一个像素之间存在一个对应,以使在掩模的一个像素上执行的操作的结果能够被延伸到图像的一个相应的像素上。

[0058] 在实践中,掩模被初始化为空白图像,并且例如,在RGBA(红、绿、蓝、透明度)色彩空间中,其像素的不透明度值对于每个像素是相同的。这允许均匀地建立掩模的每个像素的色彩通道。优选地,掩模被初始化为透明的。应当理解的是,在被执行时,掩模不被显示给用户,并且应该被视作由本方法所使用的内部结构。未被显示,用户不在掩模上直接地交互。

[0059] 然后,在图7的步骤S40处,连续的折线是例如由系统根据在图像上绘制的曲线来构建的。连续的折线包括其中每段的端点被指定的一条或多条线段(还被称为线)。因为除了被连接至一个点的两个不同的端点之外,折线的每个点被连接至两个段,所以折线是连续的。构建连续的折线意指绘制的曲线的每个点的像素位置被计算。

[0060] 在曲线的部分已经被绘制在如图10上所示的图像外部的情况下,不考虑这些部分。这在图14上示出,在图14中,根据图10的曲线已经构建了三条连续的折线140、142、144。忽略图像外部(即,不在图像上)的曲线的部分,以使系统认为已经在图像上绘制了三条独立的曲线;在图14上,针对每条独立的曲线,已经计算了一条连续的折线140、142、144。事实上,所忽略的曲线的部分(i)位于图像的外部,以及(ii)其从曲线与图像的边界之间的第一交点延伸至曲线与图像的边界之间的第二交点。也就是说,被保留的曲线的部分(被绘制在图像的內部和外部)(i)在图像的內部,以及(ii)其(a)从曲线与图像的边界之间的第一交点延伸至曲线与图像的边界之间的第二交点,或(b)从曲线与图像的边界之间的第一交点延伸至图像內部的曲线的端点。例如,在步骤S40处构建的连续的折线142在图像的边界上有其两个端点;连续的折线140和142在边界上仅有一个端点,以及第二端点位于图像內部。应当理解的是,如果曲线被完全地绘制在图像上,则保留整条曲线。

[0061] 在图像上绘制的曲线的像素位置可以如本领域中已知的来计算。在图8-9上描述的实施例中,本领域中众所周知的Bresenham算法的推导被用来链接所有这些像素位置,并且用来获得连续的像素线。

[0062] Bresenham算法计算用来链接两个分开的像素的最短路径。这在图11上示出,在图11中,灰色像素是通过使用Bresenham的算法来链接表示在图像上绘制的曲线113的两个不

同端点的位置的两个像素110、112获得的像素。本领域技术人员可以注意到,在原始的Bresenham的算法中,存在仅共享顶点的邻近像素114、116,而其它邻近像素116、118与其它像素至少共享一个边缘。

[0063] 现在参考图12,其示出了Bresenham的算法的推导:在链接表示不同端点的两个像素110、112的像素集中,每个像素与由原始Bresenham的算法识别的像素之中的像素至少共享一个公共边缘。也就是说,在邻近像素之间允许的唯一像素连接是边缘连接。在实践中,图12中示出的结果是通过识别未被原始的Bresenham的算法选择的像素120、122、124、126获得的,所述像素120、122、124、126 (i) 与仅共享顶点的两个像素共享一个公共边缘、以及(ii) 所绘制的曲线位于其上。

[0064] 现在参考图13, (利用虚线表示的) 连续的折线130可以根据像素的连续的边缘来计算,其中,每个边缘是折线的一条线。连续的边缘是在 (i) 链接曲线的两个不同端点并且利用原始的Bresenham的算法获得的像素集,和/或 (ii) 被添加用于确保在两个邻近像素之间仅存在边缘连接的新像素之中找到的。在图13中,形成连续的折线的像素的边缘被选择,以使折线链接所有像素的参考点。参考点是像素的四个顶点中的一个顶点。对于针对像素的参考点的选择是任意的;在该例子中,像素的左下顶点被选作参考点。任意顶点可以被选作参考点,同时所有像素服从于该选择。在图13中,每个像素使其自己的左下顶点作为参考点,以使折线130始于像素110的顶点111,并且止于像素112的顶点119。

[0065] 折线的计算通常是在步骤S30处计算的掩模上执行的。这避免对图像进行修改,并且因此减少计算资源的消耗。图16示出了根据图11的曲线113获得的折线160。

[0066] 构建连续的折线的步骤还可以包括构建针对图像的边界的折线。在实践中,图像形成矩形的像素网格,并且图像的边界因此是矩形。在这样的情况下,图像的四条边中的每条边是折线。构建针对图像的边的折线使用关于图像的信息:本领域技术人员计算图像的每个顶点的坐标,并且然后本领域技术人员利用线段来链接顶点。这在图15中示出,图15除了现在用虚线表示的图像的边界的四条折线151、153、155、157之外与图14相同,所述四条折线151、153、155、157分别连接顶点150-152、152-154、154-156、156-150。因为图像的边界现在被视作分界线,所以构建针对图像的边界的折线有利地便于由用户创建闭合区域。例如,如果用户想要将图像切成两半,其只需要绘制一条从图像的顶部到底部的线,并且其选择其是否想要保留图像的左边部分或右边部分或两部分。因此,如先前参考图5和图6讨论的,如果用户仅仅想要将图像切成两半,则其不再需要在图像周围画圆。

[0067] 然后,在图8的步骤S50处,识别图像中的、其至少一个边缘是所计算的连续的折线中的线中的一条线的所有像素。应当理解的是,如果连续的折线是根据绘制的曲线在掩模上构建的,则本领域技术人员识别掩模的像素。图17示出了图16的折线和所识别的像素。该步骤创建了由位于折线的两侧的两个像素构成的折线的边界,并且这个两像素边界使得对属于闭合区域的像素的识别更容易;使得对闭合区域的计算更容易,并且确保与折线接触的像素被分配到正确的闭合区域中。

[0068] 然后,在步骤S60处,在步骤S50处识别的每个像素现在被标记,用于指示在步骤S40处计算的折线相对于每个像素的位置。应当理解的是,可以进行相反的操作,即,针对每个像素,确定其相对于折线的位置。对像素做标记意指给像素添加信息:另外的信息可以利用像素数据来存储,或其可以被存储到新的数据结构中。优选地,标记是利用像素数据来存

储的;这为执行本方法的系统提供了最佳计算结果。

[0069] 折线相对于像素的位置是利用像素的参考点来推断的。如果像素的左下顶点被选作参考点,并且在图像参照中具有坐标(a,b)(通常,图像参照的原点是位于图像的左下角的像素的左下顶点),其它顶点具有坐标(a+1,b)、(a,b+1)、(a+1,b+1)。通过这种方式,每个边缘是由与位置相关联的一对坐标定义的,即,像素的右边缘、或左边缘、或上边缘、或下边缘。然后,对折线相对于像素的位置的确定就简单了:(i)如果折线中的线是像素的右边缘,那么折线在像素的右侧;(ii)如果折线中的线是像素的左边缘,那么折线在像素的左侧;(iii)如果折线中的线是像素的顶边缘,那么折线在像素的顶侧;(iv)如果折线中的线是像素的底边缘,那么折线在像素的底侧。

[0070] 在图7和图8上描述的实施例中,标记是通过使用掩模的RGBA通道对信息进行编码来执行的。也就是说,标记包括根据相对于折线的在步骤S50处识别的每个像素的位置来给其着色。在这里,首字母缩略词RGBA指的是红绿蓝透明度色彩空间,其中,每个颜色(红、绿、蓝)和不透明度(A)是由n个比特定义的,例如,一个像素的RGBA信息可以为n=8比特的每个信道提供单个32比特的无符号整数来对位置进行编码。

[0071] 因此,第一RGBA值(例如,透明绿R=0,G=255,B=0,A=0)指示折线位于像素的右侧;第二RGBA值(例如,透明白R=0,G=0,B=0,A=255)指示折线位于像素的左侧;第三RGBA值(例如,透明红R=255,G=0,B=0,A=0)指示折线位于像素的上侧;以及第四RGBA值(例如,透明蓝R=0,G=0,B=255,A=0)指示折线位于像素的底侧。本领域技术人员理解的是,使用RGBA值相当于给像素着色。因此,当执行步骤S60时,(被初始化为空白图像的)掩模现在包括着色的和透明的像素,其用于指示折线相对于这些像素的位置。

[0072] 有趣的是,对像素做标记的过程是累积的。这意味着像素可能被标记若干次,并且每个连续标记的信息可以被取回。例如,如果折线在像素的上侧和在右侧,则该像素将被标记两次,并且最后一次标记(即,两次标记的结果)将提供折线在像素的上侧和右侧的信息。在该例子中,如果指示折线位于像素的上侧的第三RGBA值被设为R=0,G=255,B=0,A=0,并且指示折线位于像素的右侧的第一RGBA值被设为R=255,G=0,B=0,A=0,那么像素的标记将被设为R=255,G=255,B=0,A=0。该像素现在用透明黄来着色。

[0073] 图17的所识别的像素被着色(此处,颜色是用被写入到每个像素中的数字来表示的):(用数字1表示的)蓝色像素指示折线位于像素的底侧,(用数字4表示的)红色像素指示折线位于像素的顶侧,(用数字3表示的)淡蓝色像素指示折线位于像素的底侧和左侧,以及(用数字2表示的)黄色像素指示折线位于像素的上侧和右侧。具有数字2和3的像素因此是具有累积标记的像素。

[0074] 然后,在步骤S70至S100处,由曲线和图像的边界划定界限的所有闭合区域被计算。所有闭合区域这一表述涉及一个或多个闭合区域被计算。闭合区域是图像上用户希望选择的被划定界限的表面,该表面是由根据由用户绘制的曲线构建的连续的折线和/或根据图像的边界构建的连续的折线来划定界限的。每个闭合区域独立于图像的一个其它闭合区域;这意味着在两个闭合区域之间无交点,也就是说,图像的像素可能仅属于一个闭合区域。一个或多个闭合区域是单独地能够选择的,也就是说,对一个闭合区域的选择可能不涉及对第二闭合区域的选择。计算闭合区域意指图像的像素集被识别,其中,集合中的每个像素具有集合中的作为邻居的至少一个像素。

[0075] 对由曲线和图像的边界划定界限的所有闭合区域的计算可以通过使用区域增长算法来执行。区域增长是图像分割方法,其中,种子点被选择,并且区域根据该选择的种子点进行生长。如果所选择的种子点的邻居遵守至少一条成员准则,则该区域可以生长,并且当邻居遵守至少一条成员准则时,其本身被分类为种子点。对于每个新的种子点,重复该过程,直至没有更多的种子点要检查为止。

[0076] 图8示出了可以与本发明一起使用的区域增长算法的例子。区域成员准则是选择的像素的邻近像素不越过折线。因此,区域成员准则目的在于保留不越过折线的邻近像素。

[0077] 在步骤S700处,掩模的像素被选择。所选择的像素属于在该方法的该步骤处未知的闭合区域。

[0078] 现在参考图18,其示出了已经利用添加折线的新线和添加两个像素184、185来稍微修改的图17的折线和所标记的像素。像素中的一个像素180已经被选择,例如,对像素的随机选择是由运行本划分方法的系统执行的。或者,用户可以选择像素中的一个像素。

[0079] 回到图8,在步骤S710处,所选择的像素中的邻近像素被检查,用于识别这些邻近像素中的哪些像素满足成员准则:在步骤S700处选择的像素180的越过先前构建的折线的邻近像素被拒绝,也就是说,其被认为不属于在步骤700处选择的像素属于的当前闭合区域。也就是说,本领域技术人员识别不越过折线的、并且因此属于所选择的像素的闭合区域的一个或多个邻近像素。应当理解的是,确定邻近像素是否越过折线是相对于所选择的像素而言的。对满足该准则的像素的识别因此包括识别拒绝的像素(越过折线的那些像素)或识别要被保留的像素(未越过折线的那些像素)。

[0080] 对成员的确认优选地是通过使用所选择的像素上的标记来进行的。如果在掩模上所选择的像素是空白的,则这意味着所选择的像素的边缘中的任何一个边缘都不是折线中的线,并且因此其邻近像素中的任何一个像素(相对于所选择的像素)都不越过折线。如果在掩模上所选择的像素被标记,那么该标记指示折线相对于该选择的像素的位置,并且共享被用作折线中的线的边缘的邻近像素被丢弃,因为其被认为是越过折线。

[0081] 然后,在步骤S720处,先前在步骤S710处识别的邻近像素被添加到将要被选择的并且以与当前选择的像素相同的方式处理的像素的列表。这确保区域增长算法的正确传播。

[0082] 返回图18,该例子中的像素连接是针对图像的每个像素的四个邻近像素的(除了图像的边界的或掩模的边界的像素)。因此,所选择的像素(图18上加阴影线的像素180)具有四个邻近像素。加阴影线的像素未被着色,这意味着像素180的边缘中的任何一个边缘都不包括先前构建的折线中的线。被识别为要被保留的像素的像素被添加到要被选择的并且以与在步骤S700处选择的像素相同的方式被处理的像素的列表:在该例子中,因为所选择的像素180的四个邻近像素满足该准则,所以其被保留并且被添加到列表。

[0083] 然后,在步骤S730处,所选择的像素被标记为选中的。这可以通过使用RGBA通道来完成,例如,该像素用黑色来着色。然后,所标记的像素被添加到属于当前计算的闭合区域的像素的列表。

[0084] 接下来,在步骤S740处,新像素是从在步骤S720处创建的列表中选择。该像素与在步骤S710处选择的像素相同的方式处理。因此,步骤S710至S740是针对该像素执行的。

[0085] 现在参考图19,现在关注新选择的像素182,其是在图18上示出的先前选择的像素

180的邻近像素。该像素182被添加到要被选择的并且以与先前选择的像素180相同的方式被处理的像素的列表。所选择的像素182用蓝色来标记,其指示(i)折线位于该像素的底侧,并且(ii)(用黄色着色,并且用数字2表示的)邻近像素190越过折线。因此,该像素190被拒绝,并且将不被添加到要被选择的像素的列表,并且因此,其将不被标记为属于与在步骤S700处选择的像素相同的闭合区域。相反,两个其它邻近像素将被添加到要被选择的像素的列表;本领域技术人员理解的是,像素180不被添加到要被选择的并且以其已经被标记为选中的方式处理的像素的列表。通过这种方式,对步骤S710至S760的重复可以收敛到一个解。

[0086] 返回图8,在步骤S760处,本领域技术人员确定要被选择的像素的列表中的所有像素实际上已经被处理。因此,当被包含在列表中的每个像素尚未被标记为选中的(例如,用黑色来着色)时,重复(S750)步骤S710至S740。

[0087] 在所述过程的该步骤处,掩模包括用黑色着色的、属于图像的第一闭合区域的像素集。这些用黑色着色的像素因此属于形成第一闭合区域的像素集。这在图20上示出,图20示出了图18的折线和所识别的像素,其中,位于折线的顶部的所有像素已经用黑色着色。有趣的是,在步骤S50处识别的并且位于折线之下的像素在所述过程的该步骤处未被触及:针对类似地计算与折线相接触的第二闭合区域所需的像素上的标记尚未被更改。

[0088] 在实践中,在闭合区域已经被计算之后,该新计算的闭合区域的像素被存储(S100)在数据结构中。针对图像的每个像素,该数据结构存储所述每个像素属于的闭合区域。如本领域中已知的,所使用的数据结构可以是四叉树数据结构。当搜索计算的闭合区域时,四叉树数据结构有利地改善性能。

[0089] 然后,在掩模上选择属于第二闭合区域的新像素(步骤S770)。该选择的像素因此未被标记为选中的,例如,未用黑色着色。应当理解的是,图像可能仅包括一个闭合区域,例如,由用户绘制的曲线不自相交或不与图像的边界交叉至少两次,以便仅图像的边界创建闭合区域。

[0090] 接着,类似地重复步骤S710至S760,直至第二闭合区域已经被计算为止,也就是说,第二闭合区域的所有像素用黑色来着色。然后,第二闭合区域的像素被存储在同一四叉树数据结构中,在其中第一计算的闭合区域的像素已经被存储。

[0091] 当在步骤S770处未对掩模的所有像素进行标记时,重复步骤S710至S760。

[0092] 接下来,在步骤S780处,本领域技术人员确定掩模的所有像素已经用黑色着色。因此,所有闭合区域已经被计算。从现在开始,并且针对掩模的每个像素,本领域技术人员在图像上识别其相应的像素,并且图像的像素是根据先前计算的闭合区域来分组的。

[0093] 应当理解的是,该步骤S780在四叉树数据结构被用于存储每个新计算的闭合区域时被暗含地执行;实际上,四叉树是利用属于新发现的闭合区域的像素的坐标来填写的。

[0094] 图21示出了根据图15上表示的连续的折线140、142、144、151、153、155、157计算的三个闭合区域220、222、224。

[0095] 现在参考图9,其示出了依据根据本发明划分的若干最初图像来创建新图像的例子。

[0096] 若干图像被显示给在所显示的图像中的一个或多个图像上绘制自由形状的用户。同样地,图像可以被并排地显示,或其可以部分地或全部地交迭。由用户绘制的曲线是在未

被修改的所显示的图像上表示的。

[0097] 然后,检测如关于步骤S70描述的已经计算的闭合形状的存在。图像的边界生成闭合区域,其能够被考虑或不被考虑。例如,在由用户绘制的曲线不自相交或不与图像的边界交叉至少两次的情况下,仅图像的边界创建一个或多个闭合区域。能够决定的是,不向用户显示这些闭合区域,或相反地向用户显示。

[0098] 然后,闭合区域被显示给用户。例如,这能够利用但不限于每个闭合区域的特定渲染、形成每个闭合区域的边界的像素的特殊渲染……来执行。在该步骤处,图像被切割。这是通过使用最终的掩模(其可以使用二叉树数据结构来存储)来执行的,所述最终的掩模针对图像的每个像素存储所述每个像素属于的闭合区域。

[0099] 然后,用户选择图像的一个或多个闭合区域。该选择如本领域中已知的来执行,例如,用户在要被选择的闭合区域上移动鼠标的光标,并且点击鼠标来告诉系统选择图像的该子部分。

[0100] 在不修改最初图像的情况下,新图像是针对那些闭合区域中的每个闭合区域来创建的。这能够利用每个选择的子部分的像素到新图像中的复制来执行。

[0101] 然后,用户能够停止该过程,或因为最初图像未被本划分方法修改,所以其可以执行进一步的选择。

[0102] 已经描述了本发明的优选实施例。将理解的是,在不背离本发明的精神和范围的情况下,可以进行各种修改。因此,其它实现方式也在所附权利要求的范围内。



图1

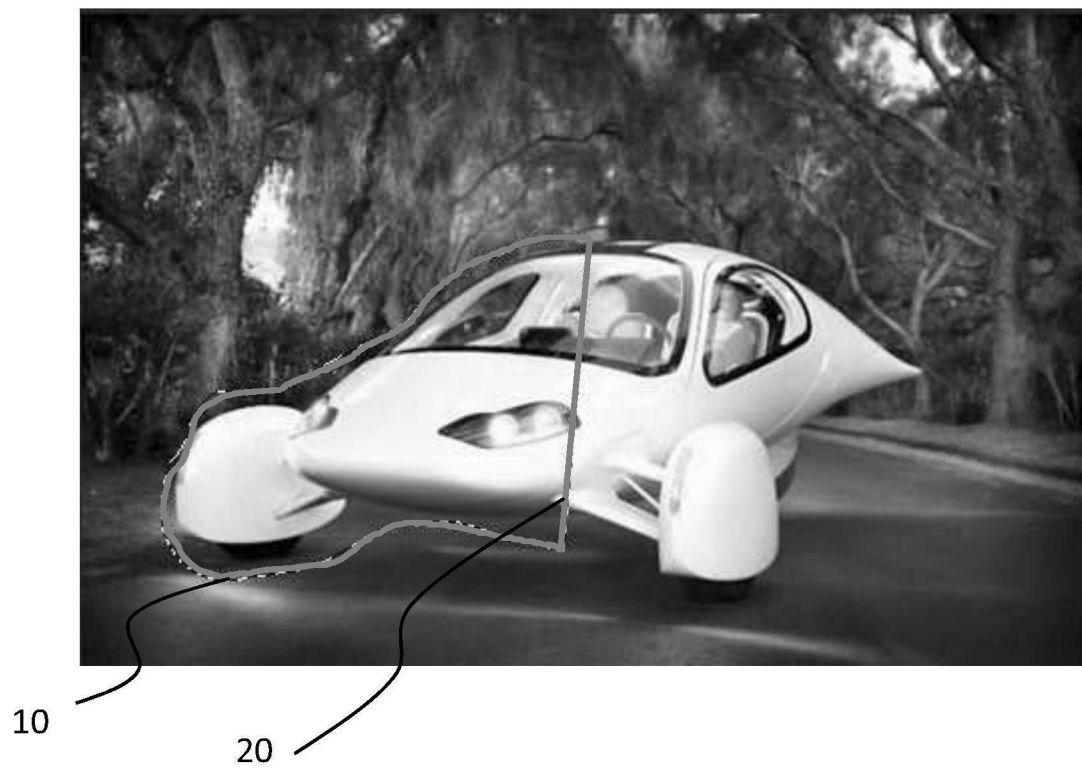


图2



图3

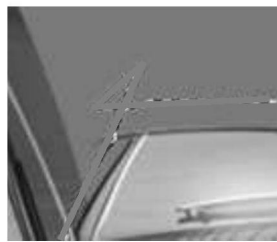


图4



图5



图6

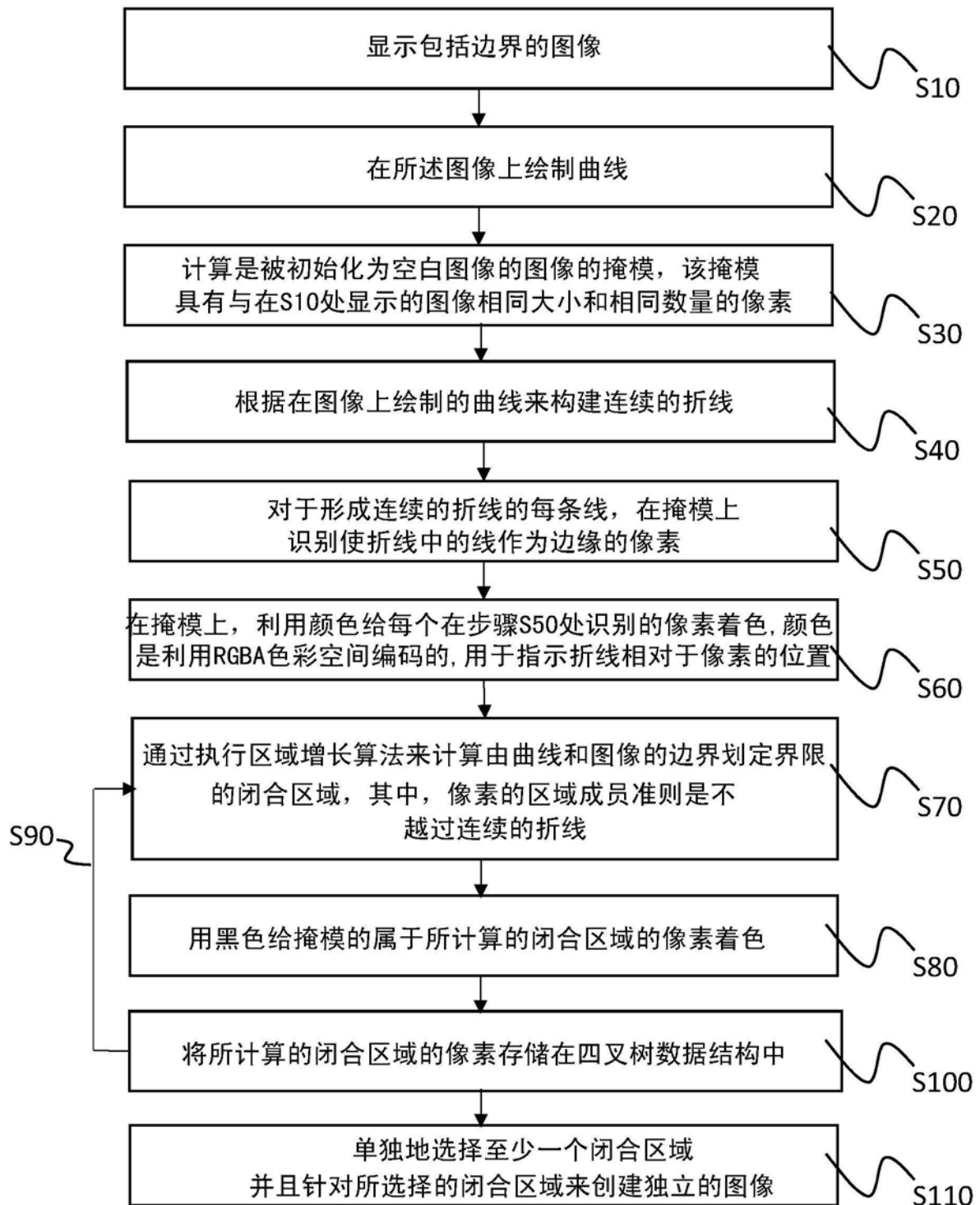


图7

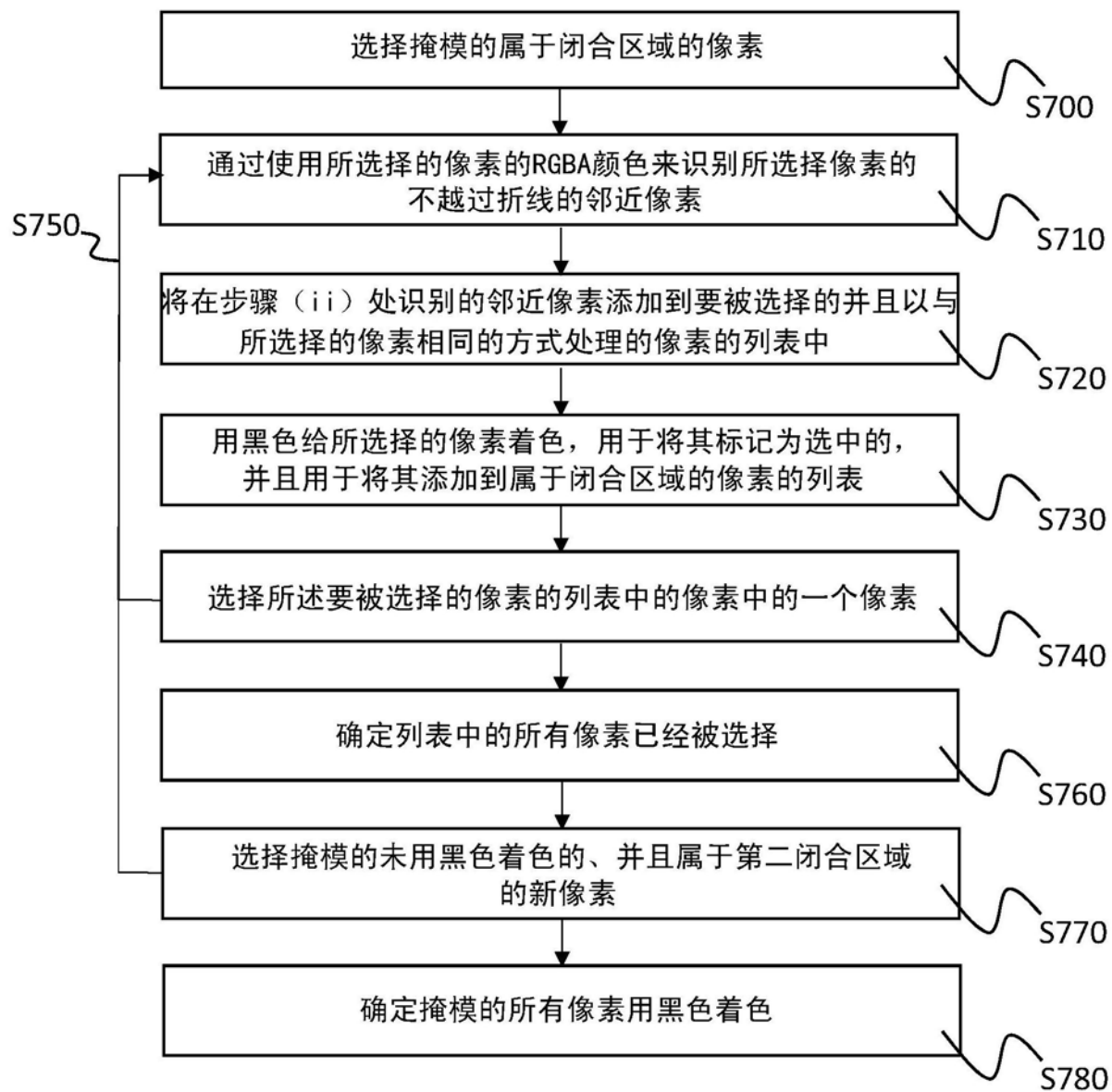


图8

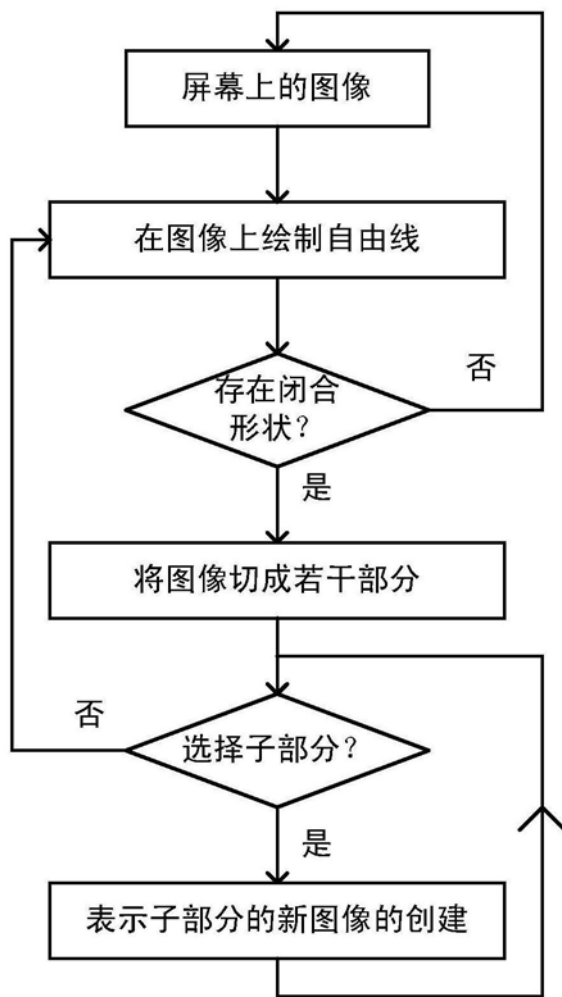


图9

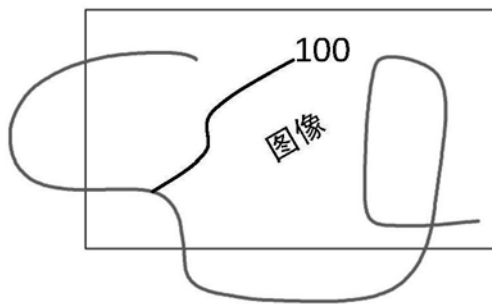


图10

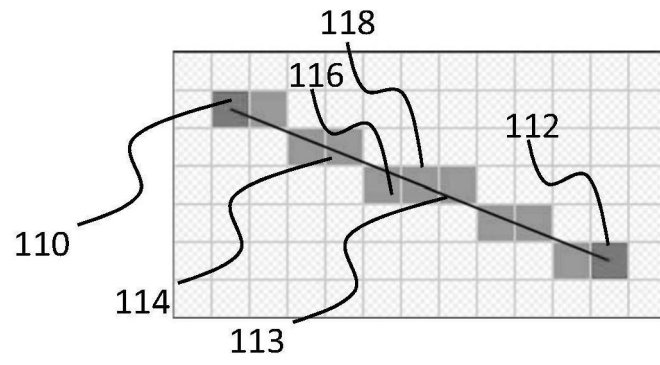


图11



图12

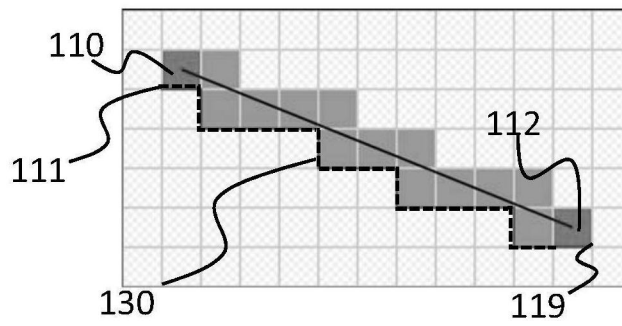


图13

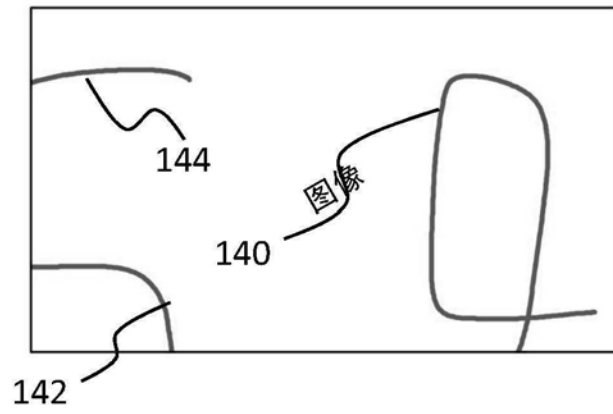


图14

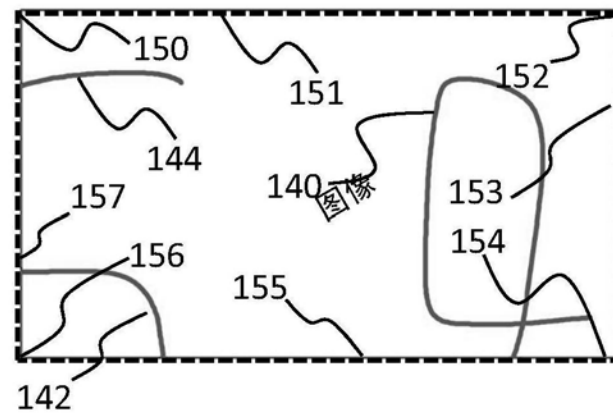


图15

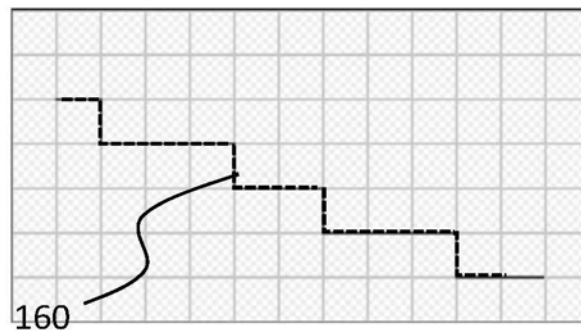


图16

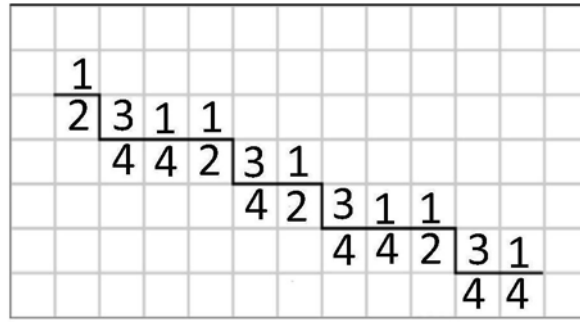


图17

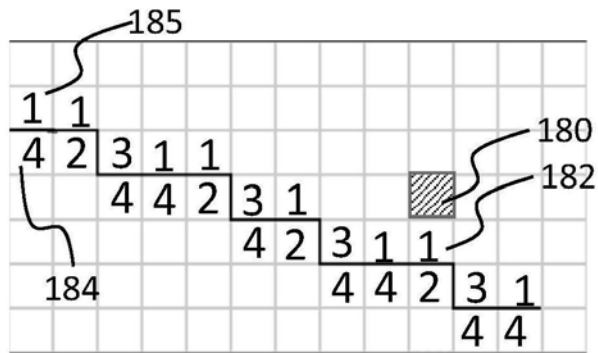


图18

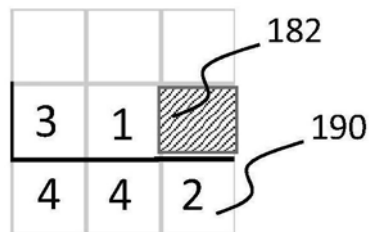


图19

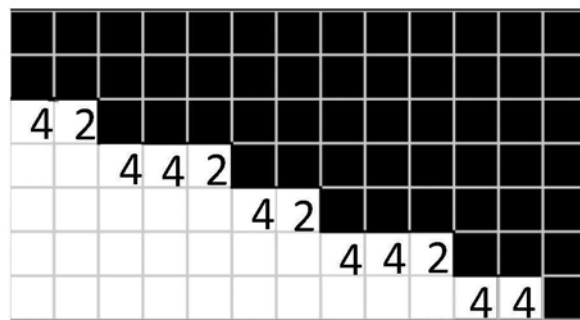


图20

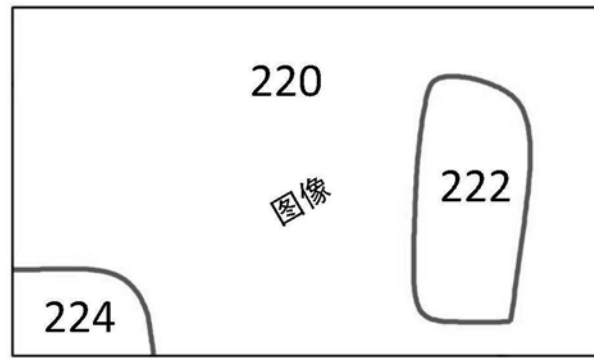


图21

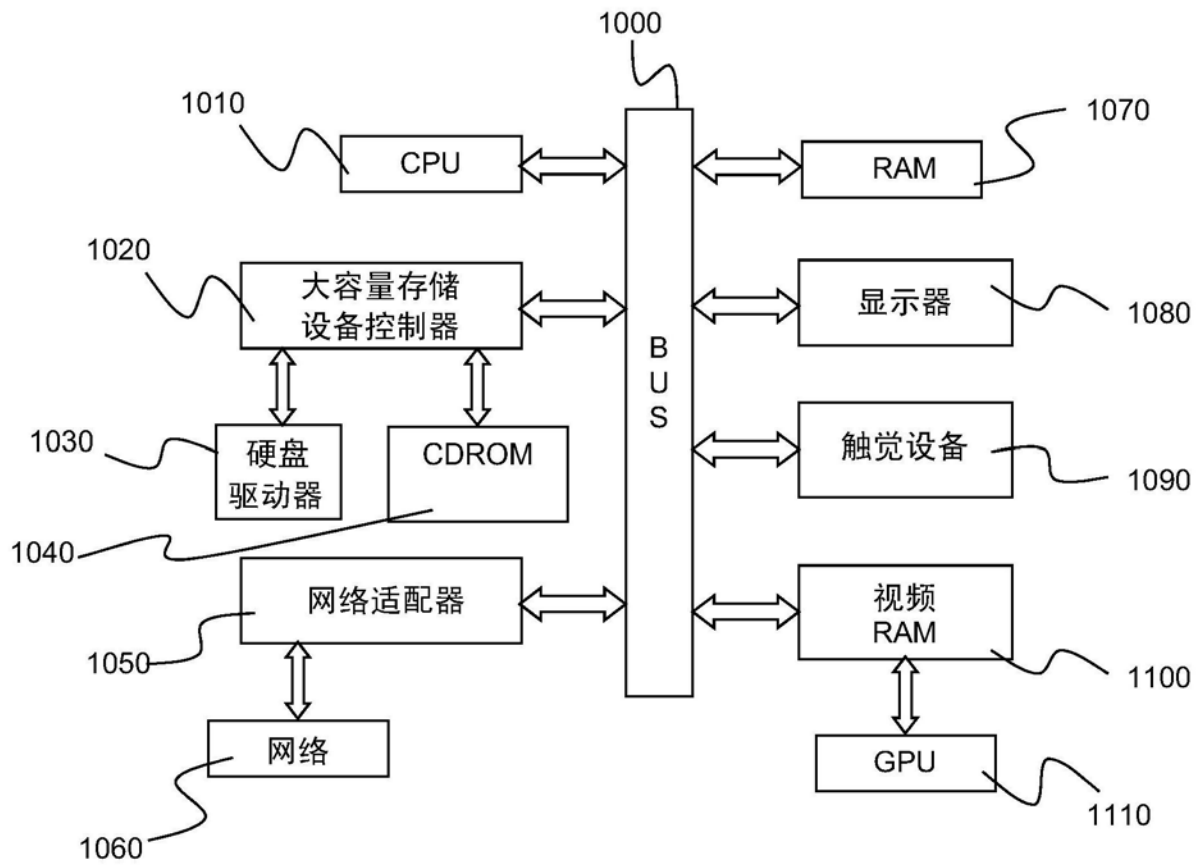


图22