

(12) 发明专利申请

(10) 申请公布号 CN 102609961 A

(43) 申请公布日 2012. 07. 25

(21) 申请号 201210012954. 5

(22) 申请日 2012. 01. 16

(71) 申请人 上海驿创信息技术有限公司

地址 200434 上海市虹口区广纪路 838 号 B
幢 616

(72) 发明人 周红全 陈金娣

(51) Int. Cl.

G06T 9/00 (2006. 01)

权利要求书 1 页 说明书 4 页 附图 6 页

(54) 发明名称

基于参考点的矢量图形复合编、解码方法

(57) 摘要

本发明提供了一套完整的矢量图形编码、解码方法。利用本发明可以对任意矢量图形进行高效、快速的编解码,并且保证了矢量图形在缩放过程中不失真。本发明在对具有一定规律的矢量图形编码时,可以极大的降低存储容量。本发明还可解决矢量图形在缩小时变得模糊的技术难题,通过参考点进行栅格调整,能够保证缩放到很小时图形清晰易辨,这对于小字号矢量字库屏幕显示,具有重要意义。本发明可以应用到各种矢量图形场合例如曲线字库、电路原理图,工程图、动漫画等。

1. 利用原始坐标系和参考坐标系（参考点）对基本矢量图形进行编码，根据编码需要路径上的点保存的坐标既可是相对于原始坐标系，也可以是相对于参考坐标系。这样路径上的点可以在原始坐标系和参考坐标系下进行各种变换，包括旋转、缩放、平移、修改等。参考坐标系主要是针对基本矢量图形的局部变换设置的，以更加自由的适应多种形状，达到节省编码空间的目的。

2. 利用复合方法将一个或多个基本矢量图形进行变换、修改后得到复合矢量图形，再利用复合方法对复合矢量图形进行多次变换、修改，从而形成更加复杂的矢量图形。

3. 基本矢量图形和复合矢量图形编码时都以组件对象来表达，利用一颗树来表示一个组件对象，这颗树的叶节点是基本矢量图形，其他节点是复合矢量图形。组件对象解码后是可以直接用来填充和渲染的矢量图形。

4. 组件对象中根节点可以对这棵树上子节点以下的其它任何节点使用附加的复合方法来进行调整控制，其目的是最大限度的共用组件对象，节省编码空间，同时又确保编码精度。

5. 利用参考坐标系（参考点）对基本矢量图形进行栅格调整，以改善矢量图形缩小时的清晰度。

6. 标记频度出现很高的组件对象事先缓存以加快解码速度。

基于参考点的矢量图形复合编、解码方法

技术领域

[0001] 本发明涉及一种矢量图形编解码规则,具体的说是一种用于矢量图形存储和还原的方法。

背景技术

[0002] 目前,矢量图一般采用直接保存轮廓点和矩阵变换的方式进行编码,这种方法原理简单,解码效率高,但是编码效率低,占用存储空间大,且不利于传输。在需要大量用到矢量图形的地方,如曲线字库、工程图、电路图、动画片等会占用大量资源,尤其不适合在系统资源有限的嵌入式设备上使用。

发明内容

[0003] 本发明的目的是为了解决上述技术问题的不足,提供一种高效的矢量图形编解码方法。本发明为解决上述技术问题的不足而采用的技术方案是:在原始坐标系基础上,引入参考坐标系,采用原始坐标系和参考坐标系同时对矢量图形轮廓点进行存储;在对复杂矢量图形编码的时候,最大限度地利用复合规则,重复使用一系列相对简单的矢量图对象,复合生成复杂矢量图形。该技术方案共包含两部分:1,基于参考点的矢量图形编码方法。2,基于参考点的矢量图形解码方法。

[0004] 1. 基于参考点的矢量图形编码方法

[0005] 在利用本发明时,按照以下方法对矢量图形进行编码。该方法包含以下内容:矢量图形、环路和路径、轮廓点、参考点、基本矢量图形、基础坐标系、参考坐标系、组件对象和复合方法。

[0006] 矢量图形:也称为面向对象的图像或绘图图像,在数学上定义为一系列由线连接的点。

[0007] 环路和路径:由曲线所构成的一段闭合或者开放的曲线段称之为路径,闭合的路径称之为环路。矢量图形可以由路径构成。

[0008] 轮廓点:形成矢量图形轮廓的,由线连接的点我们称之为轮廓点,为了形成复杂的轮廓,我们定义了轮廓点属性,例如可以把轮廓点可以分为直线点、控制点(包含样条曲线、贝塞尔曲线控制点)等。在连接这些点时我们需要根据轮廓点的属性进行不同的连接处理。对于曲线字而言,轮廓点就是环路(闭合路径)上的点。

[0009] 参考点:在本发明中,对矢量图形编码时,保存轮廓点的坐标可以选择原始坐标系或者参考坐标系。参考坐标系原点就称之为参考点。参考点也是具有属性的,该属性可以用来标识参考坐标系特点,例如我们可以定义自由移动、横向移动、竖向移动等。在解码时如果有必要,可以根据这些属性可以对图形进行栅格调整。

[0010] 基本矢量图形:基本矢量图形由一条或多条路径构成。路径是由一系列的直线点或曲线点构成的,闭合的路径可以使用奇偶填充算法或非零填充算法进行填充。开放的路径一般表达平面内具有粗细、颜色的一条曲线。因此基本矢量图形是只由轮廓点或(轮廓

点和参考点)组成的矢量图形,在本发明中它是组件对象的叶节点。

[0011] 基础坐标系:即原始坐标系,矢量图形编码、解码时所依赖的一个统一的坐标系;

[0012] 参考坐标系:根据编码需要,基础坐标系上的任一点都可以作为一个相对坐标系原点,编码时轮廓点的坐标可以用这个坐标系为基准保存,这个相对坐标系就称之为参考坐标系,因此相关的变换、修改也可以用这个坐标系作为基准。参考坐标系主要是针对基本矢量图形的局部变换设置的,在重复利用时可以使基本矢量图形更加自由的适应多种形状,达到节省编码空间的目的。

[0013] 组件对象:组件对象是包含基本矢量图形和其他组件对象的集合,每个组件对象解码后都是一个仅由路径构成的可直接用来填充和渲染的矢量图形,它可以具有形状、轮廓、大小和位置、颜色等属性。通过定义的操作(例如矩阵变换等)一个组件对象可以调用另一个或几个组件对象并形成新的组件对象。我们称这种情况为组件复合。在实际使用中,组件对象是一棵树,树的每一个节点都是一个组件对象,树的叶节点就是基本矢量图形,其他节点称之为复合矢量图形。通过组件复合我们可由简单图形形成复杂的矢量图形。在本发明中所有矢量图形都是以组件对象进行编码保存的。

[0014] 复合方法:一个组件对象调用另一个组件对象时,对该组件对象进行的一系列操作称之为复合方法,它可以包含矩阵变换、点的插入、删除和改变点的属性、改变点的位置、改变对象颜色,指定填充方法以及被调用组件对象子组件的删除等各项操作。组件对象中根节点可以对这棵树上子节点以下的其它任何节点(包括叶节点)使用附加的复合方法进行微调控制,对于矩阵变换而言,一般是把附加调整使用前置矩阵的方式来实现,这样利用基本矢量图形和复合方法,在编码时我们就可以在降低存储容量的同时生成任意复杂的高精度矢量图形。

[0015] 2. 基于参考点的矢量图形解码方法

[0016] 实际使用中,由于组件对象在计算机中都是以树的形式保存,对组件对象解码时实际上就是进行树的遍历。我们从根节点出发,到达叶节点时开始根据复合方法中的参数向父节点进行融合,最后在根节点上得到完整的矢量图形,图9中组件对象C4最终由三个基本矢量图形P1, P2, P3复合构成,其解码可以参照以下公式进行。

• P1的复合: $M(C4C5)*M(C5P1)*M(C4P1)$ [P1+C4. P1+C5. P1]

• P2的复合: $M(C4C5)*M(C5P2)*M(C4P2)$ [P2+C4. P2+C5. P2]

P3的复合: $M(C4P3)*[P3+C4. P3]$

其中M表示复合方法,P表示基本矢量图形,*表示融合。

M(C4C5)表示组件对象C4的对组件对象C5的复合方法,其余类推;

C4. P1表示组件对象C4的对组件对象P1的修改结果,其余类推;

上述P1, P2, P3复合后的结果得到的矢量图形就是C4所表达的最终矢量图形。

附图说明

[0017] 图1 组件对象(仅有轮廓点的基本矢量图形,汉字笔划“点”)

[0018] 图2 组件对象(含有参考点的基本矢量图形,汉字“一”)

[0019] 图3 组件对象(含有参考点的基本矢量图形,汉字笔划“撇”)

[0020] 图4 组件对象(含有参考点的基本矢量图形,汉字笔划“捺”)

- [0021] 图 5 组件对象 (由两个环路构成的基本矢量图形, 汉字“口”)
- [0022] 图 6 组件对象 (复合矢量图形, 汉字“口”的一种变形)
- [0023] 图 7 组件对象 (复合矢量图形, 汉字“口”的一种变形)
- [0024] 图 8 组件对象 (复合矢量图形, 汉字“口”的一种变形)
- [0025] 图 9 组件对象树状图, 曲线汉字字库的编码示意图。
- [0026] 图 10 栅格调整示意图

具体实施方式

[0027] 下面以曲线字库字形图例的方式讲述具体的概念和编码过程, 图中的点凡是以 P 表示点的都是环路上的点 (轮廓点), 轮廓点中以实心方块表示的点称之为直线点或线上点, 为空心圆的表示曲线控制点或线外点 (贝塞尔曲线或 B 样条曲线的控制点, 实现时不同的控制点类型也需要记录, 图中未做标记), 以 C 开头的点表示参考点。图 10 中的组件对象凡是以 C 表示的是复合矢量图形, 以 P 表示的是基本矢量图形。

[0028] 1. 基础坐标系和轮廓点设置。

[0029] 基础坐标系 (原始坐标系) 是每一个组件对象编码和解码时的统一坐标系。路径或环路上的所有点, 如果没有明确指定参考坐标系, 那么其保存的坐标值就是相对于基础坐标系的, 所有组件对象完整解码得到的可用于渲染的矢量图形坐标都是相对于基础坐标系的。图 1 是一个基本矢量图形, 由一个环路构成, 包含六个轮廓点, 其中 P0, P1, P3, P4 是线上点, P2, P5 是线外点; 编码时需要保存所有轮廓点相对于基础坐标系或参考坐标系的位置以及点的属性等。

[0030] 2. 参考坐标系和参考点设置。

[0031] 图 2 是一个基本矢量图形, 包含 4 个轮廓点 P0, P1, P2, P3 和 2 个参考点 C0, C1; 编码时需要存储 2 个参考点的位置、属性以及轮廓点在对应参考坐标系下的坐标位置和属性。

图 5 是一个包含 2 个环路的基本矢量图形, 其中 P0, P1, P2, P3, P4, P5, P6, P7 构成第 1 个环路 (点 P7 有环路结束标志, 图中未标明), P8, P9, P10, P11 构成第 2 个环路, 在汉字字形上表达一个“口”字。

利用参考点调整图形是本发明的关键之一, 图 1 中参考点可在字号很小时对横线进行栅格调整, 使横线变得清晰, 如果使用其他复合方法, 我们也可以让横线变成斜线或竖线。图 5 中轮廓点 P0 和 P1 是相对于参考点 C0 的, 轮廓点 P4 和 P5 是相对于参考点 C5 的, 轮廓点 P7 和 P8 是相对于参考点 C1 的, 轮廓点 P6 和 P9 是相对于参考点 C4 的, 轮廓点 P2 和 P11 是相对于参考点 C2 的, 轮廓点 P3 和 P10 是相对于参考点 C3 的。让图 5 中的参考点 C2 和 C3 向下移动, 就可以在不改变粗细的情况下, 把“口”字变成如图 6 所示形态。如果令 C0 和 C5 向上移动至 P7 或 P6 所处位置, 此时 P0 与 P7 重合, P5 与 P6 重合, 则“口”字将变成图 7 中的形状。如果要调整粗细, 只要把轮廓点相对于参考点的坐标乘以一个放大或缩小系数即可, 图 8 是由图 5 调整粗细后得到。对于图 3 (表达汉字的“撇”) 和图 4 (表达汉字的“捺”), 改变线上点对应的参考点的位置, 可以很容易实现撇、捺变长或缩短, 改变线外点对应的参考点的位置, 就可改变曲线的形状, 再加上粗细调整, 这样就可以满足不同的汉字里面地方撇、捺的需求。参考点的设置根据需要可以放在图形或图形某个局部的形心上, 也

可以与轮廓点重合,还可以放在图形之外,也可以让几个参考点共线。

图 10 是利用参考点进行栅格调整的说明,若参考点 C0, C1 处在同一条直线上,且 C1, C2 处在同一条直线上,就可以在图形缩小到很小的时候利用 C0, C1, C2 进行栅格调整,让这个笔划“横折”反走样(一种灰度填充算法)处理后看上去清晰。具体步骤如下,如果 C0, C1, C2 处在中线上,则令 C0, C1 的 Y 坐标在移动到最近的半个栅格之处,对应的轮廓点 P0, P1, P2, P3 相对于参考点 C0, C1 的相对 Y 坐标均设为半个栅格, C1, C2 的 X 坐标移动到最近的半个栅格之处,对应的轮廓点 P2, P3, P4, P5 相对于参考点 C1, C2 的相对 X 坐标均设为半个栅格。这样反走样后就就是实心的一条横线和竖线,而不会出现“发虚”的现象。如果 C0, C1, C2 与轮廓点 P1, P2, P5 重合,调整方法类似,区别在于让其 X, Y 坐标处于栅格的边界上,对应的 P0, P3, P4 需要调整的相对坐标调整为一个栅格。

[0032] 3. 组件复合。

[0033] 利用已经有的组件对象按照复合方法形成新的组件对象也是本发明的关键所在,它在形成各种复杂图形的同时可以大大提高编码效率和减小存储容量。参考点给出了多个空间自由度,使得图形大小、形状、粗细可以分别调整,配合其他复合方法(包括增加或删除点、改变轮廓点位置、矩阵变换等)就可以任意生成矢量图形并且保证了存储开销很小。组件对象中根节点可以对子节点以下的每一个节点使用附加的复合方法来进行调整控制,这也是本发明的关键点之一,图 9 中汉字“器”需要用到组件“犬”,“犬”由两个组件“大”和“、”构成,器需要分别使用附加的复合方法来调整这两个节点上的组件对象,才能使字形满足要求。

[0034] 基于本发明下面给出几个汉字在计算机中的表达形式。图 9 是曲线汉字的树状关系图,7 个汉字(“器”、“犬”、“太”、“大”、“人”、“一”、“口”)仅仅由 5 个基本矢量图形 P1, P2, P3, P4, P5 构成,其他都是通过复合方法得到,因此编码时实际存储的轮廓点是非常少的,存储的都是复合方法中的参数。大大节省了存储容量,编码时只需要调用不同的组件对象,减小了直接保存原始轮廓点的描点工作,效率得到极大提升。基于以上过程,我们可以看出,给出几个简单的基本矢量图形,通过组件复合形成可重复利用的组件对象,由于复合过程仅仅保存了一些复合参数,因此这种编码方法的结果是大大减小了存储空间和降低了编码复杂度。

[0035] 5. 解码速度的提高处理和编码时的注意事项。

[0036] 由于组件对象是一颗树,如果这棵树节点很多,比较复杂,这无疑增加了解码的空间复杂度,在一些资源有限的设备上可能会造成解码速度不够理想,可以从两个方面着手来解决这个问题。首先在编码时,找出矢量图形有共性的地方,把它做成一个组件对象,如果这个组件对象出现频率很高,则把他标记下来。其次在解码时,把标记好的组件对象和部分基本矢量图形事先缓存起来,这样既可以保证编码文件尺寸很小同时解码速度又大大提高。

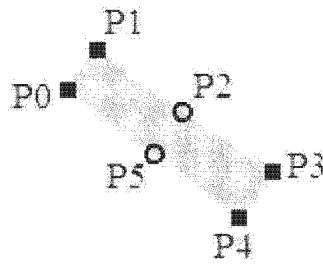


图 1

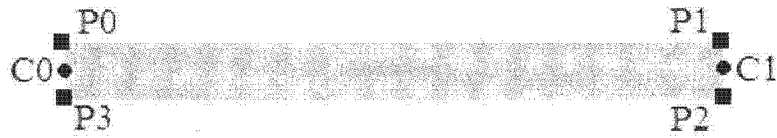


图 2

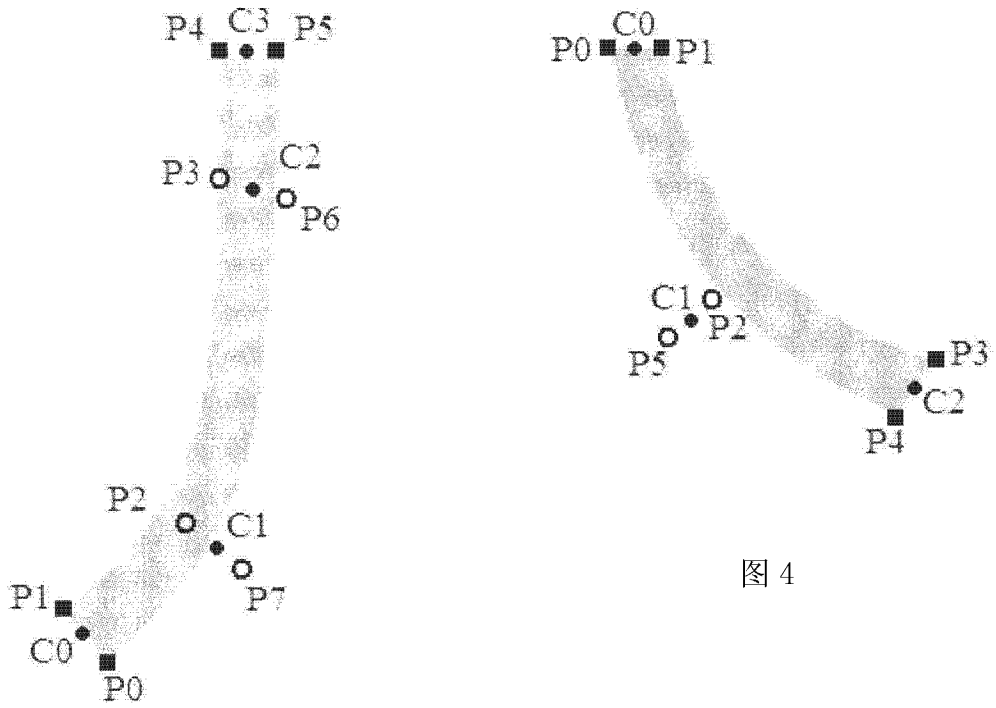


图 4

图 3

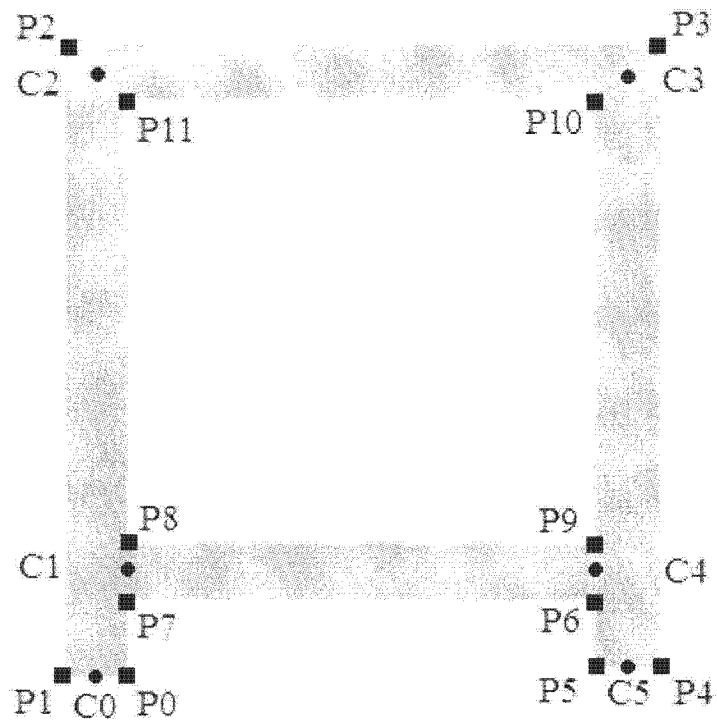


图 5

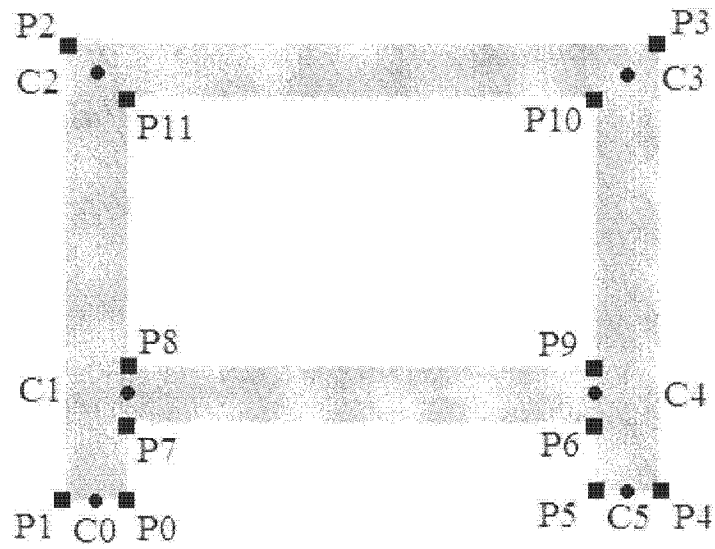


图 6

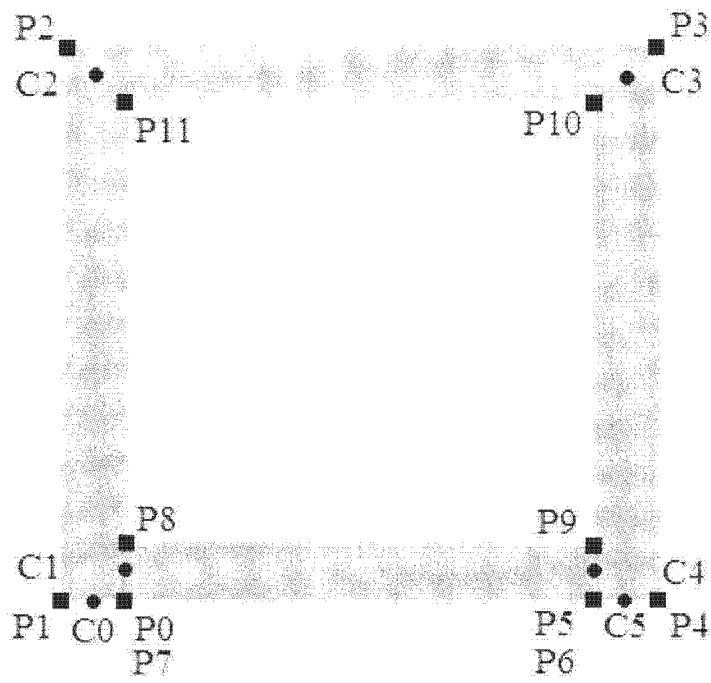


图 7

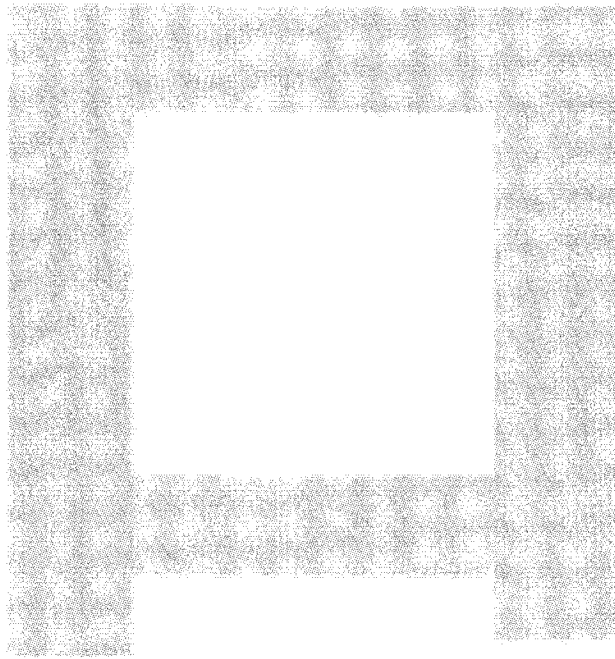


图 8

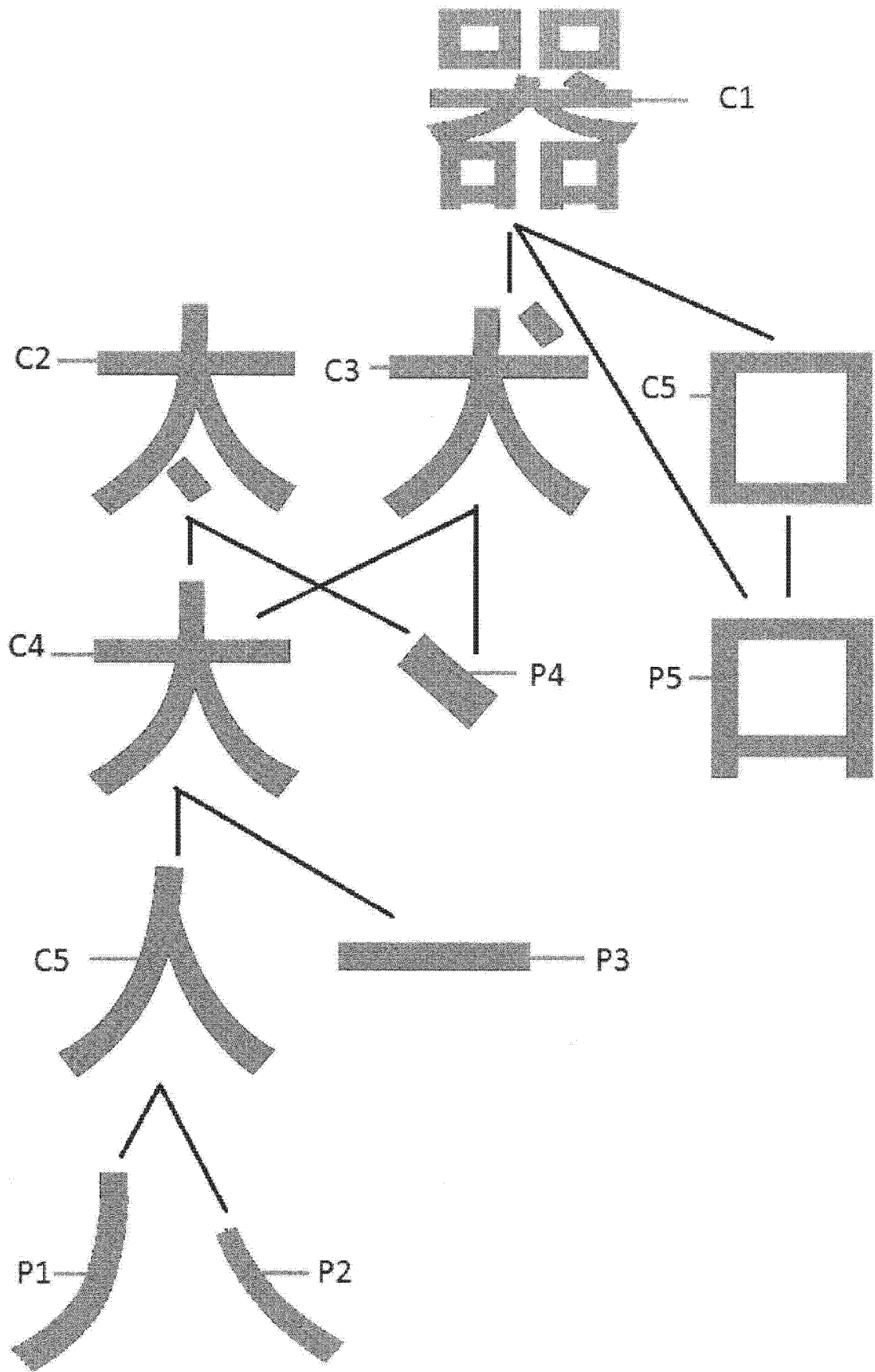


图 9

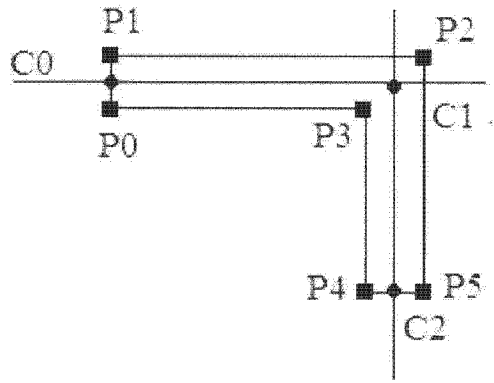


图 10