(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101334898 B
(45)授权公告日 2011.12.14

(21)申请号 200710167498.0

(22)申请日 2004.03.22

(30)优先权数据
10/396,673 2003.03.25 US

(62)分案原申请数据
200480000055.7 2004.03.22

(73)专利权人 三菱电机株式会社
地址 日本东京

(72)发明人 罗纳德·N.·佩利
莎拉·F.·弗里斯肯

(74)专利代理机构 中国国际贸易促进委员会专利商标事务所 11038
代理人 马浩

(51)Int.Cl.
G06T 11/20(2006.01)
G06T 11/00(2006.01)
G06T 5/10(2006.01)

(56)对比文件
WO 0033256 A1,2000.06.08,
EP 1244061 A2,2002.09.25,
EP 0786757 A1,1997.07.30,
US 6229521 B1,2001.05.08,

审查员 郭永强

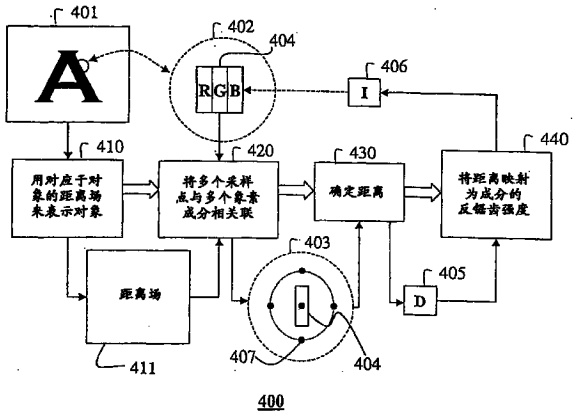
权利要求书 2 页 说明书 17 页 附图 11 页

(54)发明名称

一种确定象素成分的反锯齿强度的方法及设备

(57)摘要

根据代表一个对象的二维距离场确定一个象素的一个成分的反锯齿强度的方法及装置。所述二维距离场中的一个采样点集合与所述的象素的成分相关联。根据所述的二维距离场以及采样点集合确定一个距离，并将该距离映射为所述象素成分的反锯齿强度。



CN 101334898 B

1. 一种用于对代表一个对象的二维距离场的一个区域进行反锯齿的方法，其中所述二维距离场被分割为多个单元，每个单元包括在所述每个单元内重建所述二维距离场的方法，所述进行反锯齿的方法包括：

识别代表一个对象的二维距离场的一组单元，所述一组单元包含要被渲染的二维距离场的一个区域；

定位与所述区域相关联的一组像素；

指定用于所述一组像素中每个像素的一组成分；以及

根据所述一组单元中的距离确定所述一组像素中每个像素的每个成分的反锯齿强度，所述确定还包括：

根据所述一组单元重建所述距离；以及

把所述距离映射到所述像素的所述成分的所述反锯齿强度。

2. 根据权利要求 1 所述的方法，其中所述的二维距离场是一个二维自适应采样的距离场。

3. 根据权利要求 1 所述的方法，其中确定所述距离还包括：

定位所述像素的所述成分附近的所述一组单元内的一个单个采样点；以及

根据所述一组单元在所述单个采样点重建所述距离。

4. 根据权利要求 1 所述的方法，其中所述的映射是将所述距离与反锯齿强度相关联的一维函数。

5. 根据权利要求 1 所述的方法，其中所述的映射是一个将所述距离与反锯齿强度相关联的一维查找表。

6. 一种用于对代表一个对象的二维距离场的一个区域进行反锯齿的设备，其中所述二维距离场被分割为多个单元，每个单元包括在所述每个单元内重建所述二维距离场的方法，该设备包括：

用于识别代表一个对象的二维距离场的一组单元的装置，所述一组单元包含要被渲染的二维距离场的一个区域；

用于定位与所述区域相关联的一组像素的装置；

用于指定用于所述一组像素中每个像素的一组成分的装置；

用于根据所述一组单元中的距离确定所述一组像素中每个像素的每个成分的反锯齿强度的装置，用于确定的所述装置还包括：

用于根据所述一组单元重建所述距离的装置；

用于把所述距离映射到所述像素的所述成分的所述反锯齿强度的装置；以及

用于显示所述像素的所述成分的所述反锯齿强度的显示装置。

7. 一种根据动态化脚本把一个对象动态化为一序列帧的方法，包括：

把所述对象表示为二维距离场；

根据所述动态化脚本，为所述序列帧中的每一帧，更新所述对象的形态；以及

使用更新后的形态和一种反锯齿化渲染方法来渲染所述对象，所述反锯齿化渲染方法包括：

把表示所述对象的所述二维距离场中的一组采样点与一个像素的一个成分相关联；

根据所述二维距离场和所述一组采样点确定一个距离；

把所述距离映射为所述像素的所述成分的反锯齿强度。

8. 一种根据动态化脚本把一个对象动态化为一序列帧的方法，包括：

把所述对象表示为二维距离场；

根据所述动态化脚本，为所述序列帧中的每一帧，更新所述对象的形态；以及

使用更新后的形态和一种反锯齿化渲染方法来渲染所述对象，

其中所述二维距离场被分割为多个单元，每个单元包括在所述每个单元内重建所述二维距离场的方法，以及

其中所述反锯齿化渲染方法包括：

识别代表所述对象的所述二维距离场的一组单元，所述一组单元包含要被渲染的二维距离场的一个区域；

定位与所述区域相关联的一组像素；

指定用于所述一组像素中每个像素的一组成分；以及

根据所述一组单元中的距离确定所述一组像素中每个像素的每个成分的反锯齿强度，所述确定还包括：

根据所述一组单元重建距离；以及

把所述距离映射到所述像素的所述成分的所述反锯齿强度。

9. 一种在含有二维对象的一拐角的一单元内产生二维距离场的方法，包括：

为所述二维对象确定一组边界描述符；

根据所述一组边界描述符识别被包围在所述单元内的一个拐角点，所述拐角点把所述一组边界描述符分割为第一边界描述符子集和第二边界描述符子集，所述拐角点表示所述对象的所述拐角；

把所述单元分割为与所述拐角点最近的第一区域和与所述一组边界描述符最近的第二区域，所述分割还包括：

确定穿过所述拐角点的两条有向直线，每条有向直线垂直于所述第一和第二边界描述符子集之一，以及

确定位于这两条有向直线外侧的点处在第一区域内，而位于任何一条有向直线内侧的点则处在第二区域内；

指定一组采样距离值和一种重建方法以使用所述第一和第二区域在所述单元内重建所述距离场；以及

为所述单元在存储器中存储所述拐角点、限制第一和第二区域的所述有向直线、所述重建方法以及所述一组采样距离值，以允许通过应用所述重建方法在所述单元内重建所述距离场。

一种确定 象素成分的反锯齿强度的方法及设备

[0001] 本申请是申请号为 200480000055.7、申请日为 2004 年 3 月 22 日、发明名称为“根据代表对象的二维距离场确定一个象素成分的反锯齿强度的方法及装置”的专利申请的分案申请。

技术领域

[0002] 本发明主要涉及计算机图形学领域，并且尤其涉及由距离场表示的二维对象的渲染。

背景技术

[0003] 在计算机图形学领域，二维对象的渲染 (rendering) 是十分重要的。二维对象，如字符形状、企业标志以及包含在文档中的图例元素，都是作为静态图像或是作为组成一个动画的一系列帧进行渲染的。二维对象有多种表示方法，并且常常会出现这样一种情况，即对于特定的操作如渲染与编辑，一种表示方法要优于其他的表示方法。在这种情况下，就要进行从一种形式到另一种形式的转换。

[0004] 尽管此处我们关注数字的字型 (type)，这可能是最常见最重要的二维对象了，但是下列说明适用于所有类型的二维对象。

[0005] 我们从数字字型的一些基础背景开始。一种典型的拉丁字体族，如 Times New Roman 或 Arial，包括一组字体 (font)，比如常规、斜体、粗体或粗斜体。每种字体包括一组单独的字符形状，称为字形 (glyphs)。每种字形由它不同的设计风格来区分，比如基本的几何形状、笔画浓度、衬线、连笔形式、布局以及轮廓、薄 - 厚笔画的比例和尺寸的数值。

[0006] 有多种方法可用来表示字体，包括位图、轮廓线，比如 Type 1[Adobe Systems, Inc.1990] 和 Truetype[Apple Computer, Inc.1990]，以及过程字体，比如 Knuth 的 Metafont，其中轮廓线起主导作用。基于轮廓线的表示方法被 Mass.Cambridge 的 Bitstream Inc.、Calif.Mountain View 的 Adobe System, Inc.、Calif.Cupertino 的 AppleComputer, Inc.、Wash.Bellevue 的 Microsoft Corporation、德国 Hamburg 的 URW 以及 Mass Wilmington 的 Agfa Compugraphic 采用并推广。

[0007] 由 Hersch 所著并发表于 Cambridge University Press,1993 的“Visual and Technical Aspects of Type(字体的视觉与技术观点)”以及由 Knuth 所著并发表于 Digital Press, Bedford, MA 1979 的“TEXand METAFONT :New Directions in Typesetting(TEX 与元字形 :排版的新方向)”中，包含了大量关于字体的历史与科学的评论。

[0008] 具有特殊重要性的是两类字型尺寸 :正文字型尺寸和显示字型尺寸。正文字型的字体以相对较小的磅值渲染，如 14 磅或更小，并且被使用在一个文档的主体 (正文) 部分中，就像在本段落中一样。正文字型要求高质量的渲染，以获得清晰度和阅读舒适度。正文字型的尺寸、字样以及基线方向在单篇文档中很少改变。

[0009] 显示字型的字体以相对较大的磅值渲染，如 36 磅或更高，并且被用于标题、头

条、以及为表达情绪或引起注意的设计及广告中。与正文字型不同，显示字型注重美观，缺少空域及时域上的锯齿 (aliasing) 是重要的，而非清晰度，其中对比度比反锯齿 (antialising) 更重要。用来表示及渲染字型的框架应该能够处理好具有相互冲突的需求的这两种类型，这一点是至关重要的。

[0010] 字型可以以二值、灰度或彩色的形式被提供给一个输出装置，比如打印机或显示器。某些渲染引擎为非常小的字型尺寸使用二值渲染，以达到较好的对比度。然而，具有良好提示的灰度字体是同样清楚可辨别的。

[0011] “提示”是与每种字形存储在一起的一组规则或工序，用以规定在渲染过程中如何修改字形的轮廓线，以保持对称性、笔画磅值之类的 特性，以及一种字体中所有字形的统一外观。

[0012] 尽管人们已经尝试设计自动和半自动的提示系统，但是提示过程仍然是新字体设计以及转换现有字体以供低分辨率显示设备使用中的主要瓶颈。另外，解释提示规则的复杂性导致不能运用硬件进行字体渲染。缺少硬件支持，就不得不在软件光栅化的过程中作出折衷，比如每个象素使用较少的样值，特别是在实时地生成动态字型 (animatedtype) 时。

[0013] 灰度字体渲染通常包含某些形式的反锯齿。反锯齿 (Antialising) 是一种处理过程，它能消除二值字体中出现的锯齿边缘或阶梯效应。尽管有多种字体渲染引擎都适合使用，但是大部分都会在网格装配及提示之后以每象素 4 或 16 个采样值进行过采样，随后分别以 2×2 或 4×4 的箱式滤波器进行下采样。

[0014] 基本的滤波，如箱式滤波 (box filtering)，由渲染速度的需要来决定其必要性。但是，即使是那种方法，对于实时渲染（如动态字型所需的那样）而言常常也是太慢的，从而经过渲染的字形会带有空域或时域上的锯齿。

[0015] 印刷术的两个重要发展趋势揭示了现有技术的字体表示方法的某些固有局限性，由此提出了变革的需求。

[0016] 第一种趋势是越来越多地强调在屏幕上阅读文本，这种趋势是由计算机在办公室中的主角地位、家庭互联网浏览普及程度的提高以及 PDA 与其他手持电子设备的广泛使用共同引起的。这些显示设备通常具有每英寸 72-100 点的分辨率，这种分辨率显著低于印刷设备的分辨率。

[0017] 在对字型进行光栅化时，这种低分辨率需要特殊对待以保证阅读舒适度和清晰度，比如 Microsoft 与 Bitstream 公司在他们各自的 ClearType 与 FontFusion 技术中投入使用的资源就证明了这一点。

[0018] 第二种趋势是动态字型或动态印刷术的使用。动态字型被用来表达情感、增添趣味性以及在视觉上吸引读者的注意。动态字型的重要性由它在电视及互联网广告中的广泛使用即可见一斑。

[0019] 不幸的是，传统的基于轮廓线的字体在这两个领域内都有局限性。在低分辨率显示器上渲染字体需要仔细对待，以便平衡良好对比度以及降低空域和 / 或时域锯齿的要求，前者是清晰度所需，后者则是阅读舒适度所需。

[0020] 如上所述，基于轮廓线的字体通常会被加入提示来向渲染引擎提供指示，从而获得最佳外观。字体提示是煞费劳力且成本高昂的。例如，为日文或中文字体开发一套具有完

善提示的字体样式可能要耗费数年时间，这两种字体中可能包含了超过一万种的字形。由于提示的焦点在于改善正文字型的渲染质量，因此这种提示对于放置在任意路径上的字型以及对于动态字型来说将会是效果不佳的。

[0021] 尽管高质量的滤波可以被用来反锯齿静态文档中的灰度字型，这种文档中只含有有限数量的字体尺寸和字体样式，但是在动态字型中进行滤波往往会受到实时渲染要求的限制。

发明内容

[0022] 本发明提供了根据代表对象的二维距离场确定一个象素成分的反锯齿强度的方法及装置。所述二维距离场中的一组样点与所述的象素成分相关联。根据所述的二维距离场以及所述的样点组确定一个距离，并将该距离映射到所述象素成分的反锯齿强度。

[0023] 在本发明的一方面内容中，所述的二维距离场是一种规则采样的距离场。

[0024] 在本发明的另一方面内容中，所述的二维距离场是一种自适应采样的距离场。

[0025] 在本发明的另一方面内容中，所述的样点组由一个单个采样构成，以提供所述对象的高效高质量反锯齿。

[0026] 在本发明的另一方面内容中，基于距离的自适应过采样提供了所述对象的最优、高效、高质量反锯齿。

[0027] 在本发明的另一方面内容中，基于距离的自动提示提供了所述对象的高质量反锯齿。

[0028] 在本发明的另一方面内容中，基于距离的网格装配提供了所述对象的高质量反锯齿。

[0029] 在本发明的另一方面内容中，所述象素成分的独立可寻址性被充分利用，以提供所述对象反锯齿的有效分辨率的提高。

附图说明

[0030] 图 1A 与 1B 示出了现有技术下字形的距离场表示的框图；

[0031] 图 2A 与 2B 根据本发明的优选实施例示出了距离场表示的框图；

[0032] 图 3 根据本发明的优选实施例示出了距离场的一个双二次单元的框图；

[0033] 图 4 根据本发明示出了一种用来按图像顺序反锯齿一个对象的方法的流程图；

[0034] 图 5 示出了本发明所使用的一种线性滤波器的图；

[0035] 图 6A、6B 及 6C 示出了一个象素成分附近的采样的示意图；

[0036] 图 7 根据本发明示出了一种用来按对象顺序反锯齿一个对象的方法的流程图；

[0037] 图 8 根据本发明示出了一种用于基于距离的自动提示的方法的流程图；

[0038] 图 9 根据本发明示出了一种用来将一个笔画转换成一个距离场的方法的流程图；

[0039] 图 10 根据本发明示出了用来将一个二维对象转换成一个距离场的方法的流程图；

[0040] 图 11 根据本发明示出了用来将一个距离场转换成边界描述符的方法的流程图；

[0041] 图 12 根据本发明示出了用来动态化一个对象的方法的流程图；以及

[0042] 图 13 根据本发明示出了用来生成一个包含有一个二维对象的一角的单元中的二

维距离场的方法的流程图。

具体实施方式

[0043] 字形的距离场表示

[0044] 我们的发明将一个闭合的二维形状 S——如字形、企业标志或一个对象的任何数字化表示形式——表示为一个二维的有符号距离场 D。为了我们的说明目的，我们借助于字形。

[0045] 通俗地说，一种字形的距离场度量了从场中任意一点到字形边缘的最短距离，其中如果所述点位于所述字形之外，那么所述距离的符号就是负的，而如果所述点位于字形之内，则所述距离的符号就是正的。边缘上的点距离为零。

[0046] 正式地说，距离场是作用于所有 $p \in R^2$ 的一种映射 $D: R^2 \rightarrow R$，从而使得 $D(p) = sign(p) \cdot \min\{\|p-q\| : 对于所有位于零等值面上的点 q，即 S 的边缘\}$，$sign(p) = \{-1, 如果 p 在 S 之外; +1, 如果 p 在 S 之内\}$，并且 $\|\cdot\|$ 为欧几里得范数。

[0047] 现有技术基于覆盖的渲染方法使用单个离散的样点来表示每个象素，即使在所述的样点充分靠近轮廓线时，这种方法也可能完全弄错字形。被渲染的字形有锯齿状边缘以及漏失的部分，这两者都是空域锯齿的表现形式。如果所述的字形被动态化，那么时域锯齿就会导致闪烁的轮廓和锯齿状的边缘，它们在运动期间就像“蠕动”一样。对每个象素采用额外的样点以形成一种反锯齿的渲染，这样做可以消除这些锯齿效应，但是要达到可接受的结果需要许多样点。

[0048] 相反，即使当样点位于形状之外时，根据我们的发明的连续采样距离值也能表示出字形的相似性。实际上，单个样点值可以被用来估算有多少字形位于以各个象素为中心的滤波器覆盖范围之内。

[0049] 另外，由于距离场变化平缓，即它是 C^0 连续的，因此采样值会随字形运动而缓慢变化，从而减少了时域锯齿成分。

[0050] 距离场还拥有其他优点。由于它们是一种隐式的表示方法，因此它们享有隐函数的好处。具体地说，距离场提供了一种用于设计字体的直观界面。例如，字形的个别部分，如直笔、横笔、弧形笔画或衬线可以分别独立设计。在设计之后，所述的部分就可以利用隐式组合方法组合在一起，以构成相同字体样式的不同字形。

[0051] 距离场在动态排版术或动态字型领域还能提供许多（好处），因为距离场所提供的信息对于模拟对象间的互动是非常重要的。

[0052] 在一种优选实施例中，我们采用自适应采样的距离场，即 ADF，参见美国专利 6396492 号，“Detail-directed hierarchical distance fields（面向细节的分级距离场）”，该专利由 Frisken、Perry 及 Jones 提交，此处通过引用包含进来。

[0053] ADF 是距离场的有效数字表示法。ADF 运用面向细节的采样以减少表示距离场所需的样点数。所述的样点被存储在单元的空域分级结构如四叉树中，以便进行高效的处理。另外，ADF 还提供了用来根据采样值重建距离场的方法。

[0054] 面向细节的或自适应的采样根据场内的局部变化对距离场采样：当局部变动大时使用更多的样点，而在局部变动小时使用较少的样点。与常规采样的距离场以及三色四叉树相比，自适应采样显著降低了对存储器的需求，常规采样以均匀的速率采样整个距离场，

而三色四叉树则总是在边缘附近以最高速率采样。

[0055] 图 1A-1B 对用三色四叉树表示 Times Roman 的‘a’和‘D’所需的单元数量以及用图 2A-2B 中的双二次 ADF 表示相同精度所需的单元数量进行了比较。单元数量与存储需求直接相关。两种四叉树都具有相当于 512×512 的距离值图像的分辨率。用于表示‘a’和‘D’的三色四叉树分别具有 17393 和 20813 个单元，而相应的双二次 ADF 只有 457 和 399 个单元。双二次 ADF 通常所需的单元数比 Frisken 等人提出的现有技术的双线性表示方法少 5-20 倍，这一点在“AdaptivelySampled Distance Fields :a General Representation of Shape forComputer Graphics（自适应采样的距离场 :用于计算机图形学的通用图形表示法）”中有所说明，该文发表于 Proceedings ACM SIGGRAPH2000，pp.249-254，2000。

[0056] 双二次重建方法

[0057] Frisken 等人利用四叉树作为 ADF 空域分级结构，并根据在每个单元的四个角上采样的距离、通过双线性内插在每个单元内重建距离 及梯度。他们提出“更高阶的重建方法可以被用来进一步提高压缩率，但是数字已经表明额外努力所换得的回报开始减小”。

[0058] 然而，双线性 ADF 对于本发明所述的表现、渲染、编辑以及动态化字符字形而言还是不够用的。具体地说，它们需要过多的存储器，处理效率太低，并且在非边缘单元中的重建场的质量对动态模拟之类的操作来说是不够的。

[0059] 一种“有界表面”方法可以强制在非边缘单元中进行进一步的细分，它要求处在离所述表面（即边缘）有界距离内的非边缘单元通过一项误差判定测试，参见 Perry 等人所著的“Kizamu :A System forSculpting Digital Characters(Kizamu :一种用于塑造数字字符的系统）”，该文发表于 Proceedings ACM SIGGRAPH 2001，pp.47-56，2001。尽管上述方法减少了该有界区域内距离场的误差，但我们发现对于双线性 ADF 来说，那种方法会导致无法容忍的单元数量增加。

[0060] 为了应对那些局限性，我们用双二次重建方法替代了双线性重建方法。典型字形的双二次 ADF 所需要的单元比双线性 ADF 少 5-20 倍。当我们要求在非边缘单元中得到精确的距离场以便进行动态模拟和动态化字型这样的操作时，所需单元数就出现了更高的减少量。

[0061] 这种显著的存储量减少使得一个典型动画所需的字形能够被存入现代 CPU 的片上高速缓存中。这对处理时间有着惊人的影响，因为对系统存储器的访问基本上被排除了，这样就轻易地补偿了高阶重建方法所需的额外计算。

[0062] 图 3 根据我们的优选实施例示出了一个双二次 ADF 单元 300。双二次 ADF 中的每个单元包含 9 个距离值 301。点 (x,y)302 处的距离与梯度都可以根据下列方程 1-3 由这 9 个距离值重建出来。

[0063] 有多种双二次重建方法可供使用。我们采用二变量内插多项式，它能保证沿同样大小的相邻单元的公共边缘的 C^0 连续性。与使用双线性方法一样，不同大小的相邻单元之间距离场的连续性是利用误差判定来维持在一个规定容限内的。所述的误差判定在 ADF 生成期间控制单元的细分，参见上述 Perry 等人的著作。

[0064] 点 (x,y)302 处的距离与梯度如下确定，其中 x 和 y 都是用单元坐标表示的，即 (x,y) ∈ [0,1]×[0,1]：

[0065] 令 xv1 = x-0.5 且 xv2 = x-1

[0066] 令 yv1 = y-0.5 且 yv2 = y-1

[0067] 令 bx1 = 2xv1 • xv2、bx2 = -4x • xv2 且 bx3 = 2x • xv1

[0068] 令 by1 = 2yv1 • yv2、by2 = -4y • yv2 且 by3 = 2y • yv1

[0069] dist = by1 • (bx1 • d1+bx2 • d2+bx3 • d3)+

[0070] by2 • (bx1 • d4+bx2 • d5+bx3 • d6)+ (1)

[0071] by3 • (bx1 • d7+bx2 • d8+bx3 • d9)

[0072] gradx = -[by1 • (4x • (d1-2d2+d3)-3d1-d3+4d2)+

[0073] by2 • (4x • (d4-2d5+d6)-3d4-d6+4d5)+ (2)

[0074] by3 • (4x • (d7-2d8+d9)-3d7-d9+4d8)]

[0075] grady = -[(4y-3) • (bx1 • d1+bx2 • d2+bx3 • d3)-

[0076] (8y-4) • (bx1 • d4+bx2 • d5+bx3 • d6)+ (3)

[0077] (4y-1) • (bx1 • d7+bx2 • d8+bx3 • d9)]

[0078] 利用浮点运算重建一个距离可能需要约 35 次浮点操作 (flops)，而利用浮点运算重建一个梯度则需要约 70 flops。由于我们的重建方法不包含分支，并且字形可以完全存放在一个片上高速缓存中，因此我们就可以通过利用特殊的 CPU 指令以及现代 CPU 的深指令流水线来进一步优化这些重建方法。另外，我们还可以利用定点运算来重建距离与梯度。

[0079] 用于传输及存储的压缩

[0080] 线性四叉树

[0081] ADF 四叉树的空域分级结构是某些处理过程（如冲突检测）所需要的，但是对于其他处理 - 比如下文中将要说明的基于单元的渲染来说却不是必要的。

[0082] 要为 ADF 字形的传输及存储提供压缩，我们采用一种线性四叉树结构，该结构将我们的双四叉树 ADF 存储为一个叶单元列表。该树状结构可以根据需要由所述的叶单元恢复出来。

[0083] 线性 ADF 四叉树中的每个叶单元中包括该单元的 x 和 y 坐标，每个坐标用 2 个字节表示，该单元的级别，用 1 个字节表示，单元中 心处的距离值，用 2 个字节表示，以及 8 个离开中心的距离偏移量，各用 1 个字节表示，每个单元总共 15 个字节。

[0084] 各个距离偏移量是通过从中心距离值中减去其对应样点的距离值、用单元大小进行比例缩放以减小量化误差、再截取成 8 个比特，从而得到的。表示每个单元位置的两个字节以及表示单元级别的一个字节可以以高达 $2^{16} \times 2^{16}$ 的分辨率表示 ADF。这对于表示在显示屏分辨率下渲染的字形来说是足够了。

[0085] 字形可以用 16 比特的距离值精确表示。将 8 个距离值编码成 8 比特的距离偏移量比各用两个字节存储这些值要节省的多。尽管在理论上，这样做可能会导致大单元的距离场中出现一些误差，但我们还未发现任何视觉效果上的退化。

[0086] 一个高分辨率的字形通常需要 500-1000 个叶单元。无损熵编码可以额外得到 35-50％的压缩。因此，高分辨率 ADF 的整个字体样式可以用 300-500 Kbytes 表示出来。如果只需要正文字型或者目标分辨率比较粗糙，就像用于移动电话的情况，那么就可以使用仅需 1/4 至 1/2 单元数的低分辨率 ADF。

[0087] 这些存储量显著小于灰度位图字体，并且与带有良好提示的基于轮廓线的字体的存储量相当，所述灰度位图字体的每种字体样式的每个点尺寸需要约 0.5 Mbyes。TrueType

字体的存储量从数十 Kbytes 到数十 Mbytes 不等，取决于字形数量以及提示方法的数量。Arial 和 Times New Roman 是 Monotype Corporation 的两种具有良好提示的字体，它们分别需要 266 Kbytes 和 316 Kbytes。

[0088] 根据轮廓线实时生成

[0089] 根据我们的发明，ADF 可以根据已有的轮廓线或边界描述符（比如 Bezier 曲线）利用 Perry 等人所述的平铺生成器快速地生成，下文中将对此进行详细的说明。到字形轮廓线或边界的最小距离可以利用 Bezier 曲线高效地计算出来，参见 Sederberg 等人所著的“GeometricHermite Approximation of Surface Patch Intersection Curves（曲面交线的几何赫尔米特近似）”，该文发表于 CAGD，8(2)，PP.97-114，1991。

[0090] 在 2GHz Pentium IV 处理器上生成每个字形需要 0.04-0.08 秒。整套字体样式可以在大约 4 秒内生成。由于不需要传统的提示，生成 ADF 所需的边界描述符远远小于它们相应的有提示的对应物。

[0091] 因此，我们可以存储这些最小轮廓线，并根据需要由这些轮廓线动态地生成 ADF 字形，而不是存储 ADF。这些最小轮廓线的小尺寸对于存储空间有限的设备以及在有限带宽的网络上传输字形的应用来说是很重要的。

[0092] 图 10 示出了用于将二维对象（比如字形）转换成二维距离场的方法 1000。对象 1001 被表示成一组边界描述符（比如曲线 spline）和一个填充规则（比如一种奇偶规则或非零环绕规则）。

[0093] 所述的一组边界描述符首先经过预处理 1010。该预处理过程对边界描述符进行细分以减小它们的空间跨度。所述的边界描述符还可以被接合起来以减小边界描述符组的基数。该预处理过程使得我们可以在确定无符号距离时减少每个位置所需查询的边界描述符数量，这将在下文中说明。

[0094] 根据经过预处理的边界描述符组 1011 构建 1020 一个空域分级结构 1021，比如一棵四叉树。交点（intersections）的高速缓存 1031 被初始化 1030。交点的高速缓存 1031 中储存边界描述符与一组直线（比如距离场的水平线、垂直线、对角线等等）相交的位置，以及相交的方向。这样做能够去除决定无符号距离的符号时的冗余计算。这些相交点可以间隔存储。

[0095] 然后在一组位置上查询 1040 所述的空域分级结构 1021，以确定那些位置处的一组距离。该组距离被用来构建一个二维距离场 1041。所述的查询在每个位置上调用一个距离函数，比如 Bezier 裁剪，以测定一个无符号的距离。交点的高速缓存、所述的位置以及填充规则都被用来为该距离决定一个符号。

[0096] 借助基于组件的字体的压缩

[0097] 适用于中文、日文以及韩文字体的有效压缩可以通过使用 Font Fusion 中所用的基于组件的表示法来实现，所述的这些字体中可能包含 10000 或更多的字形。所述的表示法将字形分解成通用的笔画和部首，即多种字形共用的复杂形状，将这些笔画和部首存储到一个字体库中，然后在字体渲染引擎中重新组合它们。

[0098] 由于距离场是隐式表示法，因此可以利用混合或 CSG 操作轻易地组合 ADF，由此也可见 ADF 是非常适合用基于组件的方法进行压缩的。

[0099] 在二维距离场中表示拐角

[0100] 双线性或双二次重建方法所用的面向细节的采样允许 ADF 用较少数量的距离值来表示一个二维对象相对反锯齿的部分。然而，在拐角附近，距离场有较大的变化，这些重建方法就不能很好地近似了。为了精确地表示拐角附近的距离场，这些 ADF 要求包含拐角的单元被高度细分，从而显著提高了存储器需求。另外，ADF 的最大细分层次限制了用双线性及双二次 ADF 单元表示拐角所能达到的精度，如 Perry 等人所述，其中最大细分层次是在 ADF 生成期间施加的。

[0101] 为了应对这个问题，我们的发明提供了一种方法 1300，该方法可以在含有二维对象（如字形）的拐角的单元内生成一个二维距离场。

[0102] 方法 1300 根据二维对象确定 1310 一个边界描述符的有序集合 1311，并根据边界描述符 1311 的有序集合识别出 1320 一个单元内的拐角点 1321。然后该单元被划分 1330 成两个区域，最接近拐角的第一区域和最接近对象边界的第二区域。方法 1300 还规定了 1340 一种重建方法以及一组采样距离值 1371，用于重建所述单元内的距离场，并且该方法将拐角点 1321、界定区域的线条、重建方法以及采样距离值集合 1371 储存 1380 在一个存储器中。

[0103] 所述的重建方法根据点所在的区域来决定单元内某点处的距离。第一区域内一个查询点的距离被定为从该查询点到拐角点的距离。

[0104] 为了确定第二区域内的距离，我们将边界描述符的有序集合 1311 划分 1350 成两个子集，一个子集中包含拐角点 1321 之前的边界描述符，一个子集中包含拐角点 1321 之后的边界描述符。然后，边界描述 符的每个子集被扩充 1360 以形成一条将所述单元划分成内部与外部两个部分的延长曲线。对于每个部分来说，所述单元内的距离场可以根据采样距离值 1371 重建，所述的采样距离值是根据相应的延长曲线确定的 1370。双二次重建方法可能需要为所述的两个部分各存储 9 个距离值。

[0105] 注意，所述的两个内部部分的相交处形成了所述对象的拐角。因此，可以通过重建到第一内部部分的距离以及到第二内部部分的距离，然后再选取两个确定距离中的较小者，来重建第二区域内的距离。

[0106] 所述的两个区域可以根据通过拐角点的两条有向直线来指定，所述的两条直线各垂直于两个边界描述符子集之一。每条直线都可以由拐角点以及在该拐角点处对应的边界描述符子集的外向法线决定。当一条直线被如此定义后，我们就可以通过求出从查询点到拐角点的向量与外向法线两者的向量积来确定查询点位于该直线的哪一侧。位于两条线外侧的点处在第一区域内，而位于任何一条线内侧的点则处在第二区域内。

[0107] 字体渲染

[0108] 在当今的字体渲染引擎中，字体主要被表示为轮廓线，这些轮廓线可按需要被缩放以匹配所需的输出尺寸。尽管大多数高分辨率打印机使用两级渲染，但是现代显示设备更多地在小点阵尺寸上使用灰度渲染或是灰度与两级渲染的组合形式。

[0109] 用于光栅化灰度字形的常见方法涉及缩放及提示它们的轮廓线。经过缩放及提示的轮廓线被扫描转换为高分辨率的图像，通常比所需的分辨率高 4 或 16 倍。然后，通过施用一种滤波方法（比如箱式滤波器）来对该高分辨率图像进行下采样，以产生最终的灰度图像。

[0110] 对于正文字型来说，单独的字形可以被光栅化一次并作为一幅灰度位图存储在高

速缓存中，以便在预处理步骤中再使用。对字形的子象素布局需要可能要求对每个字形的几个版本进行光栅化。为正文字型使用高速缓存就可以在某些任务期间进行短延时（如1/2秒）的更高质量渲染，所述的任务可以是例如翻阅一份Adobe Acrobat PDF文 档。

[0111] 然而，在任意路径上渲染的字型以及动态化的字形不允许使用高速缓存，因此必须根据需要来产生。实时渲染的需要要求必须使用分辨率较低的滤波，通常为每象素四个采样点以及箱式滤波。这可能引起空域及时域的锯齿。这种锯齿可以利用系统存储器中具有提示的设备字体来消除。然而，维持实时帧率对于如何使用具有提示的设备字体提出了严格的限制，比如说，具有提示的设备字体不能被动态缩放或旋转。

[0112] Microsoft最近在ClearType上的工作带来了用于LCD彩色显示器的特殊处理方法，在这种显示器中含有重复样式的可寻址彩色子象素，即成分(component)。Platt在“Optimal Filtering for PatternedDisplays(用于样式化显示器的最优滤波)”一文中描述了一组用于各个颜色成分的感观最优滤波器，该文发表于IEEE Signal ProcessingLetters,7(7)，pp.179-180,2000。在实际应用中，所述的最优滤波器被实现为一组三个移位箱式滤波器，每个滤波器用于一种颜色。

[0113] ClearType使用现有技术的基于覆盖的反锯齿方法，来决定每个象素的各个成分的强度。与此相对，我们的基于距离场的方法使用距离场来决定每个象素的各个成分的强度，并且使用了较少的样点。下文中所述的ADF反锯齿方法可以取代箱式滤波器，来提供每象素样点数较少的更好的最优滤波器的模拟。

[0114] 反锯齿(Antialiasing)

[0115] 理解经过渲染的字体中的外表成分需要对锯齿有所认识。通常，一个象素由离散的成分组成，比如彩色打印机或显示器中的红色、绿色及蓝色成分。在一个灰度设备中，象素是单个离散成分。由于象素是离散的，因此将其呈现给输出设备本质上就是一个采样过程。采样率取决于设备的分辨率。采样率至少是源信号中最高（奈奎斯特）频率的两倍，否则采样信号就会出现锯齿。

[0116] 边缘（比如字形轮廓线）具有无限的频率分量。因此，边缘不能由采样数据准确地表示出来。边缘的采样不足会导致锯齿状图形，这 种锯齿状会在运动图像的采样边缘上蔓延。如果源信号也含有一种空域样式，比如‘m’的重复垂直线或是‘i’的单根垂直线，其频率成分对于采样率来说太高了，那么采样数据就会出现信号丢失、波纹样式以及时域闪烁。

[0117] 为了避免锯齿，输入信号必须经过预滤波以去除超过采样率所允许范围的频率成分。通常，有两种方法进行预滤波。

[0118] 第一种称为解析滤波。它在采样前对源信号的连续表示形式进行某种形式的空域平均。不幸的是，解析滤波常常是不可行的，或是因为源数据不是以连续信号的形式提供，在图像处理中这是很正常的情况，或是因为确定信号在滤波器作用范围内的解析表示太复杂了。对于计算机图形学中所有除了简单几何图形外的几何形状都会出现这种情况，当然对基于样条线的轮廓线来说也会同样出现这种情况。

[0119] 第二种方法称为离散滤波。在那种方法中，通常以高于目标速率的速率采样源信号，以获得一幅过采样图像。然后，在将图像下采样到目标速率之前，使用一个离散滤波器来减少过采样图像中的高频成分。该离散方法在计算机图形学中被称为常规过采样。

[0120] 根据处理预算、硬件考虑以及个人对输出图像中对比度与反锯齿度的偏好，可以

使用各种不同的离散滤波器。通常被用来渲染字型的箱式滤波器用一个过采样值方阵的算术平均来替代这个过采样值方阵，这种方法在信号处理界通常被认为是比较低级的。

[0121] 在另一种方法中，自适应过采样将可用于采样及滤波的资源集中用于具有较高的局部频率分量的图像区域。最优自适应采样可以根据图像中的局部变化性来确定。然而，这种技术的实用性会受到估算图像局部变化的需要的限制，这种估算过程可能是计算代价高昂的。

[0122] 由于对高频样式的常规采样不足而引起的波纹样式，是人类视觉系统尤其反感的。在一般的图像处理中，随机或跳动采样被用来解决这一问题。对于随机采样，采样点会随机地略微偏离它们的指定位置。随机采样会用高频噪声替代波纹锯齿，并且已经被证实在减少时域锯齿中特别有效。

[0123] 使用基于距离的反锯齿的渲染

[0124] 在现有技术的字体渲染中，由字形边缘引入的无限频率分量是锯齿的主要成分。与之相对，通过利用 2D 距离场来表示 2D 对象，并根据本发明采样 2D 距离场，我们就能避开这些边缘，因为所述的表示方法是 C^0 连续的。最高频率取决于字形本身的空域样式，比如‘m’的重复垂直线或是‘i’的单根垂直线。

[0125] 通过用字形的 2D 距离场表示字形，我们就能有效地对该字形使用解析预滤波器。下文中所述的我们用于渲染距离场的反锯齿方法所产生的输出与传统解析预滤波器的输出不同。

[0126] 利用距离场进行反锯齿

[0127] 图 4 示出了用来按图像顺序反锯齿一个对象 401（如一种字形）的方法 400，所述的对象被表示 410 成一个二维距离场 411。每个象素 402 中可以包括一个或多个成分 404，通常是用于‘RGB’类型输出设备的红、蓝或绿色成分。该方法可以为每个象素 402 的各个成分 404 使用一个或多个采样点。方法 400 提供自适应的基于距离的过采样、基于距离的自动提示以及基于距离的栅格装配。结果得到的反锯齿象素强度可以被作为一幅图像的一部分呈现在 CRT 以及类似 LCD 的显示器上。该方法对于渲染运动模糊特别有用。

[0128] 表示对象 401 的二维距离场 411 的采样点 407 的一个集合 403 与每个象素 402 的各个成分 404 相关联 420。根据二维距离场 411 以及采样点 403 的集合来确定 403 一个距离 (D)405。然后，距离 405 被映射 440 为象素 402 的成分 404 的反锯齿强度 (I)406。

[0129] 在优选实施例中，字形 401 由双二次 ADF 411 表示，如上文所述。这样就使得在字体渲染过程中使用基于距离的反锯齿方法效率很高。其他的表示方法，比如二维距离图、二维距离壳以及程序性距离场，也都可以被采用。

[0130] 对于一幅图像中每个象素 402 的各个成分 404，都要利用四叉树遍历方法来定位一个含有成分 404 的单元，比如叶单元，所述的遍历方法在美国专利申请 10/209302 中有所说明，该专利于 2002 年 7 月 31 日提交，题为“Method for Traversing Quatrees,Octrees, andN-Dimensional Bitrees（用于遍历四叉树、八叉树以及 N 维二叉树的方法）”，本文通过引用将该专利申请完全包含进来。尽管本技术领域内其他已知的方法也可以被用于我们的发明，但是上述方法是无需进行比较的，因此执行效率更高。成分 404 处的距离根据单元的距离值重建出来，并被映射 440 为反锯齿强度 (I)406。

[0131] 可以使用不同的映射，包括线性、高斯以及 S 形函数。最佳映射函数的选择是主观

的。在一个实施例中，我们的映射是两个函数的组合。第一个函数如上所述，第二个函数是对比度增强函数。这两个函数被组合起来以便将距离场 (D)405 映射 440 为成分 404 的反锯齿强度 (I)406。

[0132] 图 5 以距离 502 的函数形式示出了强度 501 的线性映射 500，比如 [0,1]。该映射为象素的各个成分把距离转换成反锯齿图像强度。距离在对象内部是正的，在对象外部则是负的。不同的截止值 503 和 504 会影响边缘对比度以及笔画浓度。我们为显示字型采用 (-0.75,0.75) 象素的外部 503 及内部 504 滤波器截止值，为正文字型采用 (-0.5,0.625) 象素的滤波器截止值，达到了良好的效果。

[0133] 基于距离的最优自适应过采样

[0134] 上述基于距离的反锯齿方法减少了字形边缘引起的锯齿。然而，当笔画宽度或是字形各部分之间的距离对于显示器采样率来说太小时，锯齿成分仍会出现。在这种情况下，我们就要使用下文中将要说明的基于距离的自适应过采样，以进一步减少空域及时域的锯齿。

[0135] 在优选实施例中，我们使用带有我们独创的基于距离的自适应过采样的双二次 ADF，相对现有技术的基于轮廓线的表示法以及基于覆盖的自适应过采样方法而言，该方法具有显著的优势。由于 ADF 采用面向细节的采样，因此局部变化性较强的距离场区域可由较小的叶细胞表示。由此，ADF 四叉树的结构提供了实现基于距离的最优自适应采样所需的局部变化映象，从而克服了上文中所述的现有技术自适应过采样反锯齿方法中确定局部变化性的困难。

[0136] 对于图像中每个象素 402 的各个成分 404，都要定位含有成分 404 的单元，并且要将位于该成分的滤波器工作半径 r 以内的一组 403 采样点 407 与该象素成分 404 关联起来 420。每个成分 (spc) 中的采样点 407 的数量取决于单元尺寸 (cellSize) 对 r 的相对大小。采样点 407 处的采样距离经过滤波，以确定 430 一个单个加权平均距离 405，然后该距离被映射 440 成象素 404 的成分 404 的反锯齿强度 406。

[0137] 有多种不同的滤波器及采样策略可以使用。在优选实施例中，我们使用一种普通形式的高斯滤波器，用 $W^{-1}2^{-3(d/r)^2}$ 对各个距离采样值加权，其中 d 是该采样点到象素成分的距离，W 则是用于那个成分的权重总和。类似的结果也可以用箱式滤波器、漏斗形滤波器、负瓣滤波器以及其他形式的高斯滤波器得到。

[0138] 图 6A-C 示出了我们的采样策略。采样点 407 被定在成分 601 附近的同心圆 610 上，以便高效地计算权重及权重总和。我们使用象素间距的 1.3 倍作为滤波器半径 r 602，并在 cellSize ＞ r(图 6A) 时采样 1 个 spc，在 r/2 ＜ cellSize ≤ r(图 6B) 时采样 5 个 spc，在 cellSize ≤ r/2(图 6C) 时采样 13 个 spc。

[0139] 除了同心圆之外，本发明还可以使用多种其他策略来关联采样点 407 与象素成分 404。我们的方法对具体的采样策略并不特别敏感。

[0140] 下文中所述的另一种自适应采样策略，将采样点定在滤波器半径 r 以内所包含的所有单元的中心。该策略能得到同样良好的结果。

[0141] 基于单元的反锯齿

[0142] 上述的距离场反锯齿方法可以利用基于扫描线的光栅化方法在软件中实现。或者，被划分成单元的距离场也可以一个单元接一个单元地反锯齿，即按对象顺序反锯齿。基

于单元的渲染消除了为定位含有采样点的单元而进行的树遍历，消除了用于计算单个单元内距离以及梯度的冗余设置，并且减少了单元数据的重复检索，即内存读取。

[0143] 另外，由于渲染所需的单元可以被表示为固定大小的自包含的单元的顺序模块，即单元内的点的距离及梯度可以根据单元的距离值确定，因此我们的基于单元的方法容易用硬件实现，从而使得实时渲染 成为可能。

[0144] 图 7 示出了一种用来按对象顺序渲染对象 701（例如一个字形）的方法 700，所述的对象被表示 710 为一个二维距离场 711。方法 700 提供了基于距离的自适应过采样，基于距离的自动提示以及基于距离的网格装配。最终得到的经过反锯齿的象素强度可以作为一幅图像的一部分被呈现在 CRT 以及 LCD 类的显示器上。该方法对于渲染运动模糊特别有用。当二维距离场 711 的单元被安排在一个空域分级结构中时，我们可以利用 MIP 映射 (mipmapping) 来减少所需的距离采样点数量。

[0145] 二维距离场 711 被划分成单元 712。在一个优选实施例中我们使用双二次自适应采样的距离场，其中每个单元的大小取决于二维距离场的局部变化性。每个单元都包括一种方法 (M)713，用于重建该单元内的二维距离场。一组合有待渲染的距离场区域（虚线）722 的单元 721 被识别 720 出来。

[0146] 区域 722 被用来定位 730 一组与该区域有关的象素 731。对应于象素 731 集合中的各个象素的一组成分 741 被指定 740。然后，根据单元集合中的距离为每个象素的各个成分决定反锯齿强度 751。这里，所述的距离是根据单元集合重建的。然后所述的距离就被映射为反锯齿强度，如上所述。

[0147] 在一个实施例中，我们可以通过定位单元集合中靠近象素成分的一个单个采样点、并重建该单个采样点到单元集合的距离，来确定距离。在我们的优选实施例中使用了双二次自适应采样的距离场，我们通过对小于滤波器半径的单元进行特殊处理而增强了该方法，其用于基于距离的自适应过采样。由于小单元出现在距离场中变化性较强的位置，因此可以在将距离映射为强度前，对靠近这些单元的象素距离进行预滤波。

[0148] 我们对元素的合成缓存器进行初始化，其中的每个元素对应于所述象素集合中各个象素的一个成分。单元集合中的每个单元都可以被独立地处理。在优选实施例中，每个单元包括一个加权距离以及一个 累计权重，两者都被初始化为零。在一个单元被处理时，这些加权距离以及累计权重在缓冲器元素中被递增，所述的缓冲器元素与位于所述单元内或是位于所述单元中心的滤波器半径内的象素成分相对应。

[0149] 在处理了所有的单元之后，用累计权重对每个象素的各个成分的加权距离归一化，以产生接着将被映射为反锯齿成分强度的距离。在优选实施例中，我们使用与上文所述相同的高斯加权和滤波器半径。

[0150] 至此所述的基于单元的渲染总是要处理单元集合中的每个叶单元，而不管每个单元对滤波器半径的相对大小。理论上，这样做提供了最优的基于距离的自适应过采样。而实际上，ADF 四叉树可以被用作 MIP 映射以减少单元数量。

[0151] ADF 四叉树结构使得我们可以用小的叶单元的前辈来替代小的叶单元，从而有效地裁剪了某预定单元尺寸的四叉树。只要该单元尺寸小于或等于象素间间隔的 1/4，那么在基于距离的自适应过采样结果中就不会存在视觉效果上的退化。这样做减少了渲染该区域所需的单元数。

[0152] 处理象素成分

[0153] 一个象素包括一个或多个成分。例如，典型的CRT或LCD彩色显示器上的象素包括红色、绿色以及蓝色的成分。在我们的发明中，当象素包括多个成分时，它们就会被单独地处理，如上文中所述，或是被当作一个单个成分处理。当多个成分被当作单个成分处理时，就可以根据单个成分的反锯齿强度来确定该象素的彩色或灰度值。

[0154] 将多个成分当作一个单个成分处理有两个原因。首先，它能减少渲染时间。其次，当多个成分不能被单独寻址，或是当诸单个成分的相对位置未知时，对每个成分进行个别处理就是不可能的。

[0155] 当显示设备，如LCD，具有可寻址的象素成分时，单独地处理多个成分能够提高显示设备的有效分辨率，这一点在现有技术中是已知的。我们的发明可以充分利用这类设备的这一特性，来提供质量优于现有技术的基于距离的反锯齿。

[0156] 动态化二维对象

[0157] 图12示出了方法1200的流程图，该方法根据动画脚本1202将对象1201动态化成为一系列的帧。动画脚本1202为所述帧序列中的每一帧指示出对象的条件，例如对象的位置、大小、方向以及变形。该对象被表示为一个二维距离场。根据动画脚本1202，为帧1221序列中的每一帧更新1210对象1201的形态1211。利用经过更新的形态1211以及基于距离的反锯齿渲染方法1212来渲染对象1201。

[0158] 表示对象1201的二维距离场可以从不同的对象表示法中得到，例如对象的轮廓线描述或是对象的位图描述。

[0159] 为特定对象1201进行形态1211的更新1210，可以通过对该对象进行各种不同的操作来实现，其中包括刚体变换、自由形态变形、软体冲击变形、水平级方法、颗粒模拟以及改变其渲染属性。

[0160] 在渲染1220所述的对象时，我们将表示对象的二维距离场中的一组采样点与帧1221序列中的一个帧内的一个象素成分关联起来。通过由所述的二维距离场以及采样点组来确定距离，我们就可以将距离映射成所述象素成分的反锯齿强度。

[0161] 在一个优选实施例中，我们将表示对象1201的二维距离场划分成单元，每个单元中都包括了用于重建该单元内的二维距离场的方法。为了在该例中进行渲染1220，我们要在表示对象1201的二维距离场中识别出一组单元，其含有待渲染的二维距离场区域，还要定位与该区域相关的一组象素。指定对应于该组象素中各个象素的一组成分。根据该组单元为象素的各个成分确定一个距离，并将该距离映射为该象素成分的反锯齿强度，从而为该组象素中每个象素的各个成分确定一个反锯齿强度。

[0162] 基于距离的自动提示

[0163] 标准字体表示法中的提示是一种耗时的手工过程，其中字型设计人员以及提示专家要生成一组规则，以便更好地将字形个体匹配到象素网格中。良好的提示有助于在小字型尺寸下产生间隔合适、具有良好对比度且外表均匀的字形。

[0164] 这些规则提供：具有相同对比度分配的垂线，其左边与底部边缘 具有可能实现的最锐利的对比度；字形的对角线条以及纤细圆滑的部分具有足够的对比度，以便向眼睛传递视觉结构；以及衬线，这些衬线结合在一起并提供足以被人眼捕获的强化突出，参见Hersch等人所著的“Perceptually Tuned Generation of Grayscale Fonts（灰度字体的

感观调节生成）”，该文发表于 IEEE CG&A，Nov，pp.78-89，1995。

[0165] 注意，现有技术的滤波方法会产生模糊的字符，并向不同的字符部分指定不同的对比度方案，从而违反了字型设计的重要原则。为了克服这些局限性，就要为每种字体的每个字形开发提示。现有技术的提示方法存在许多问题：它们的开发耗费劳力，而且非常复杂以至于不能用硬件实现。

[0166] 对基于轮廓线的字体而言，用提示进行渲染是一个三步骤的过程。首先，所述字形的轮廓线被缩放并与象素网格对齐。其次，轮廓线被修改以控制纵线、横线以及衬线的对比度，并提高非常纤细部分以及弧线的浓度。第三，对经过修改的轮廓线进行过采样，随后进行带有滤波的下采样。

[0167] 尽管上文中所述的我们未经提示的基于距离的反锯齿渲染方法比现有技术的字体渲染方法更为有利，但是众所周知，感观提示可以提高小字型尺寸下的阅读舒适度。

[0168] 因此，如图 8 中所示，我们充分利用距离场来提供基于距离的自动提示，以便进行小点尺寸下的字形渲染。提示中的第一步 810 是缩放距离场并将其与象素网格对齐。这一步可根据给定的或推导出的字体规格自动完成，比如盖帽高度、x 高度以及基线的位置。

[0169] 在应用了这种网格装配的形式之后，我们就可以利用距离场及其梯度场来提供感观提示。

[0170] 在一个实施例中，距离场梯度的方向被用来检测 820 对象左边与底部边缘上的象素。通过暗化 830 这些象素并亮化 840 对面边缘上的象素，我们就可以实现左边与底部边缘上较高的对比度，而无需改变明显的笔画浓度。这可以通过降低或提高相应的象素强度来完成。

[0171] 在另一个实施例中，梯度场被用来为对角线条及细弧线提供更好 的对比度。我们注意到，如果一个象素位于或靠近字形的狭窄区域，那么该象素两侧的相邻象素就会具有相反的梯度方向，即，它们的点积为负数。通过检测梯度方向的突变，我们就可以暗化 850 这些狭窄区域上的象素。

[0172] 这些只是如何利用距离场来自动提供感观提示的两个实例。所述的距离场也可以被用来提供最佳的字符间距以及均匀的笔画浓度。

[0173] 生成与编辑字体

[0174] 有两种设计字体的基本方法。第一种是手工。其中，字形靠手绘制、数字化，然后将轮廓线匹配到数字化的位图中。第二种方法则是由计算机完成。

[0175] 在后一种情况下，有三类工具可供使用。直接的视觉工具可被用于曲线处理。程序设计工具通过执行一个程序的指令来创建字形的形状。该程序或是定义一个形状的轮廓并填充它，或是定义一条由笔尖绘出的路径，该路径有多种属性，其中包括几何形状及方向。基于组件的设计工具允许设计者们创建基本的组件，如直线、弧线以及其他重复出现的形状，然后组合这些组件以产生字形。

[0176] 我们使用造型编辑器来提供基于笔画的设计。这是与 3D 雕刻相对的一项技术，后者在美国专利申请 09/810261 中有所说明，该专利申请题为“System and Method for Sculpting Digital Models（用于塑造数字模型的系统与方法）”，于 2001 年 3 月 16 日提交，此处通过引用包含进来。笔画描绘可以被交互地完成，或者也可以编写脚本令其模拟可编程的设计工具。

[0177] 也可以采用基于弧线的设计，这种设计方法利用了 Bezier 曲线处理工具，该工具与 Adobe Illustrator 中所用的工具相似。基于弧线的设计方法可以与用来将轮廓线转换成距离场以及将距离场转换为轮廓线的方法组合起来，以提供设计规范间的无缝接口。

[0178] 基于组件的设计方法在隐式距离场使用 CSG 以及混合操作。这样就使得组件可以被单独设计，并在编辑或渲染期间组合起来。

[0179] 我们还提供了一种由模拟及数字字体模板自动生成 ADF 的方 法。

[0180] 对于基于组件的设计方法，我们的字体编辑器提供了利用四叉树操作来有效反射及旋转 ADF 的能力，用以塑造字形中常见的对称性。其他的特性包括 ADF 缩放、平移以及组合多个 ADF 的操作，比如 CSG 与混合。

[0181] 对于基于笔画的设计方法，我们提供了刻图工具，该工具具有几何轮廓以模拟笔尖。模拟笔尖的方向及尺寸可以沿笔画变化，以模仿书法笔迹。

[0182] 图 9 示出了用来由一个笔画生成一个二维距离场 931 的方法 900。我们在笔绘期间采样笔的状态，该笔状态包括笔绘期间该笔的位置。该笔状态还可以包括方向与几何形状。根据笔状态采样点 901，我们可以沿笔画生成 910 笔状态的有序列表 911。然后，由笔状态的有序列表生成 920 一组边界描述符 921。最后，我们由该组边界描述符 921 生成 930 一个二维距离场 931。

[0183] 在优选实施例中，边界描述符 921 是三次 Bezier 曲线之类的曲线。

[0184] 在优选实施例中，我们运用曲线匹配程序来将 G^2 连续曲线的一个最小集合匹配到笔的路径上，其精度由用户指定。我们还利用工具尺寸与方向生成离开该条路径的偏移点的两个额外的有序列表，并将曲线匹配到这些偏移点上，以产生笔画轮廓线。轮廓曲线被放置在一个空域分级结构中，以便有效地处理。我们利用平铺生成器由该分级结构生成一个二维 ADF，参见美国专利申请 09/810983 号，该专利申请于 2001 年 3 月 16 日提交，此处通过引用包括进来。

[0185] 到轮廓线的最小距离可以利用 Bezier 裁剪来有效地计算。笔画被转换成 ADF，对于用户来说没有感观上的延时。为了进行曲线操作，我们提供了 Bezier 曲线编辑器。

[0186] 如图 11 中所示，我们还提供了将距离场转换为边界描述符 - 如 Bezier 曲线 - 的能力，以提供三种设计规范间的无缝接口。

[0187] 在优选实施例中，我们使用双二次 ADF，其中该转换过程利用 ADF 分级结构遍历叶单元，以便进行快速的相邻搜索，生成沿 ADF 零等值线上的点的有序列表，然后如下文中参照图 11 所述的那样匹配曲线以生成边界描述符。

[0188] 在现有技术中，边界描述符误差是根据点列表计算得出，与之相对，我们直接由距离场计算出边界描述符误差。我们特别注意了尖锐的棱角。我们的方法快得足以让用户在范例间无缝切换，而不会引起明显的延时。

[0189] 图 11 示出了用来将二维距离场 1101 转换成一组边界描述符 1131 的方法 1100。首先，我们选取 1110 二维距离场 1101 的等值轮廓线 1111，比如距离为零或其他偏移量的。

[0190] 其次，我们由等值轮廓线 1111 及二维距离场 1101 生成 1120 点 1121 的有序列表。在我们使用了双二次自适应采样距离场的优选实施例中，该步骤利用相邻搜索技术遍历了自适应采样距离场 1101 的相邻单元。该搜索技术充分利用了自适应采样距离场 1101 的空域分级结构，以便沿着等值轮廓线 1111 有效地定位下一个相邻的单元。

[0191] 在另一个实施例中，我们通过选取 ADF 1101 中的边界单元，以各个边界单元为种子生成一组有序点，并利用 ADF 1101 的距离场和梯度场将各个点移动到 ADF 1101 的等值线 1111 上，以此来生成 1120 点 1121 的有序列表。

[0192] 然后，我们初始化 1130 一组边界描述符 1131，以匹配点 1121 的有序列表。边界描述符 1131 的初始化 1130 是通过将点 1121 的有序列表中的相邻点结合来完成的，以此形成一组线段，这些线段构成了最初的边界描述符 1131。

[0193] 在另一个实施例中，我们通过定位拐角点、将所述的点的有序列表划分成由这些拐角点界定的片段、并确定片段边界描述符以匹配各个片段，从而初始化 130 一组边界描述符 1131。片段边界描述符的组合形成了最初的边界描述符 1131。

[0194] 拐角点可以通过测量由距离场求得的曲率来定位。在优选实施例中，距离场是双二次 ADF，其中高曲率的区域由 ADF 中的小单元表示，从而可以利用 ADF 的单元尺寸来定位拐角点。

[0195] 一旦边界描述符 1131 被初始化 1130，边界描述符 1131 就会被更新 1140。更新 1140 通过重建距离场并测量边界描述符离开等值轮廓线的平均或最大偏差，来为每个边界描述符求出一个误差。

[0196] 边界描述符 1131 被持续更新 1140，直到每个边界描述符的误差令人满意、或是过了预定时间、或是边界描述符 1131 的集合的基数最小为止。

[0197] 为了把以非数字形式－即模拟模板－或以位图之类的数字形式－即数字模板－储存的现有传统字体合并进来，我们的编辑系统提供了用来从高分辨率两级位图生成 ADF 的方法。

[0198] 模拟模板首先被扫描来产生某个分辨率下的两级数字模板，该分辨率至少要比目标 ADF 分辨率高 4 倍，比如说，4096×4096 的数字模板对与现代的显示分辨率及显示尺寸来说就足够了。然后，对位图进行准确的欧几里德距离变换，以生成表示字形的规则采样距离场。

[0199] 然后，我们利用平铺生成器由该规则采样的距离场生成一个 ADF。在 2GHzPentium IV 处理器上，每个字形的位图到 ADF 的转换需要约 10 秒钟。

[0200] 为了从现有技术的字形描述符转换到距离场，我们应用了参照图 10 说明的方法，其中在所述的距离场中字形由一组边界描述符表示。

[0201] 用于动态版式的计算底版

[0202] 我们 ADF 字形框架的距离场以及空域分级结构属性也可以被用于 2D 对象的计算机模拟，比如字形、企业标志或是任何 2D 形状。例如，上述的两种属性都可以被用在冲突检测与避免中，用于计算互相穿过的实体间的作用力，以及用于模拟软体形变。

[0203] 级设置方法利用了有符号的距离场，该方法可被用来模拟许多效应，比如熔化与流体动力学。ADF 是紧凑的隐式表示法，它可被有效地查询以计算距离值和梯度，这是上述方法所需的两种重要计算。

[0204] 与此相对，由移动或变形的轮廓线来求出距离值和梯度在用于 实时互动的软件中是不实用的，参见Hoff等人所著的“Fast andSimple 2D Geometric Proximity Queries Using Graphics Hardware(利用图形硬件进行快速简单的 2D 几何近似查询)”，该文发表于 Proc. Interactive 3D Graphics’01，2001。Hoff 等人利用图形硬件来为变形曲线实时

生成规则采样的 2D 距离场，其中所述的变形曲线由线段近似。

[0205] 距离场的隐式本质允许复杂的拓扑变化，比如表面偏移，这类变化很难用基于轮廓线的字体来模拟。另外，距离场还可以被用来提供动态化对象的非光学现实主义渲染，以添加艺术效果。

[0206] 本发明的作用

[0207] 本发明提供了一种新颖的框架，用于表示、渲染、编辑以及动态化字符字形、企业标志或是任意二维对象。在一种优选实施例中，本发明利用二维双二次 ADF 来表示二维对象。该双二次重建方法提供了存储器使用与计算负荷间的最佳平衡。

[0208] 本发明包括一种方法，用来在一个含有二维对象拐角的单元内生成一个二维距离场。相比现有技术，该方法提供了对存储器需求的显著减少，以及精度的明显改善。

[0209] 我们基于距离的反锯齿渲染方法利用每象素一个单个未经提示的距离采样点提供了比现有技术中所用的过采样方法更好的反锯齿。

[0210] 我们基于距离的方法充分利用了 ADF 的空域分级结构，来提供高效最优的基于距离的自适应过采样，从而实现了出色的空域与时域反锯齿。我们的方法还为基于距离的自动提示、为基于距离的网格装配、为联合三种常用的数字字体设计规范以及为动态字体样式产生多种特殊效果，提供了计算基础。

[0211] 尽管本文中已经通过优选实施例的方式对本发明进行了说明，但是应该明白，在本发明的指导思想及范围内还可以作出多种其他的变化与改进。因此，附属权利要求的目的就是要涵盖所有这些包含在本发明的真正指导思想与范围内的变化及改进。

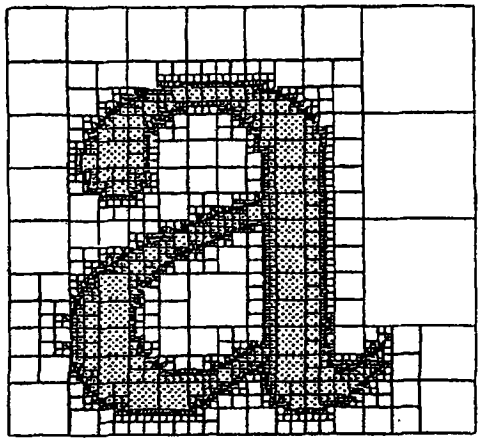


图1A

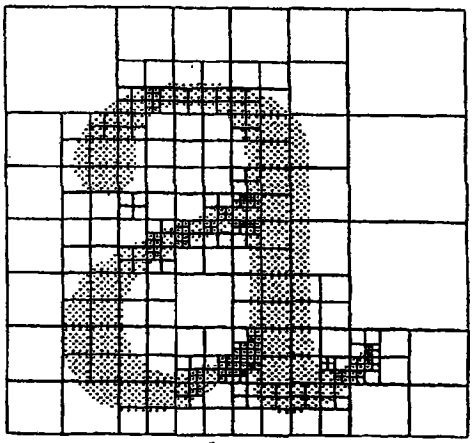


图2A

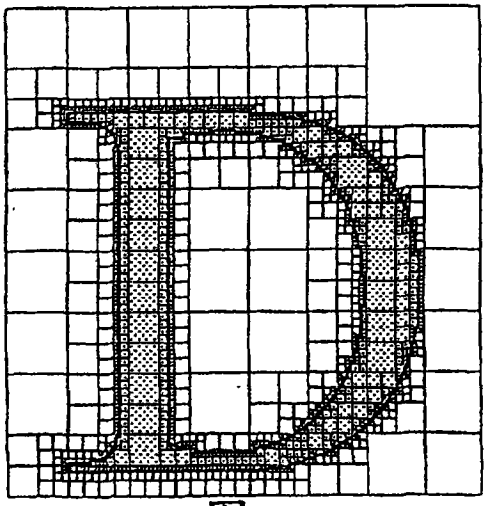


图1B

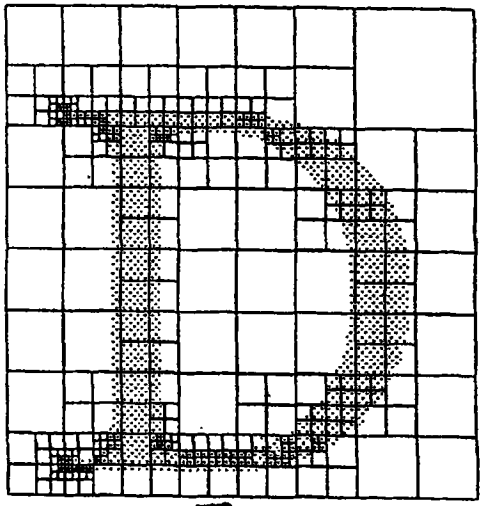


图2B

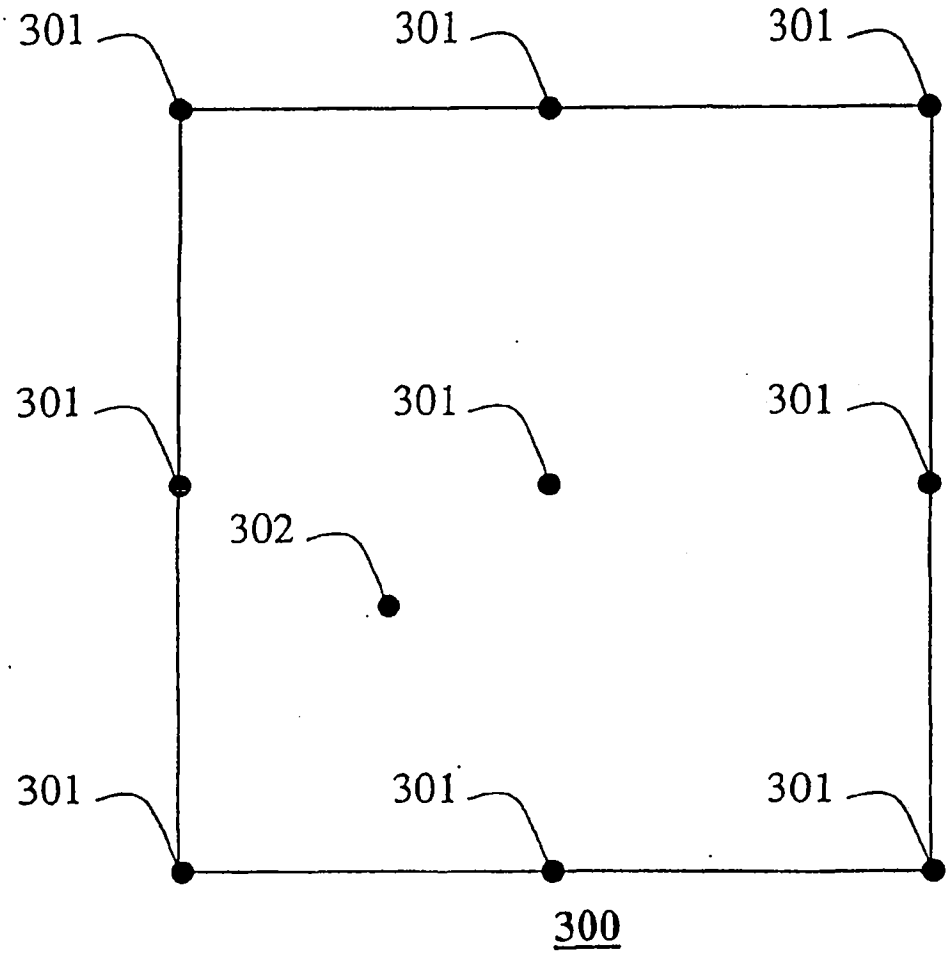


图 3

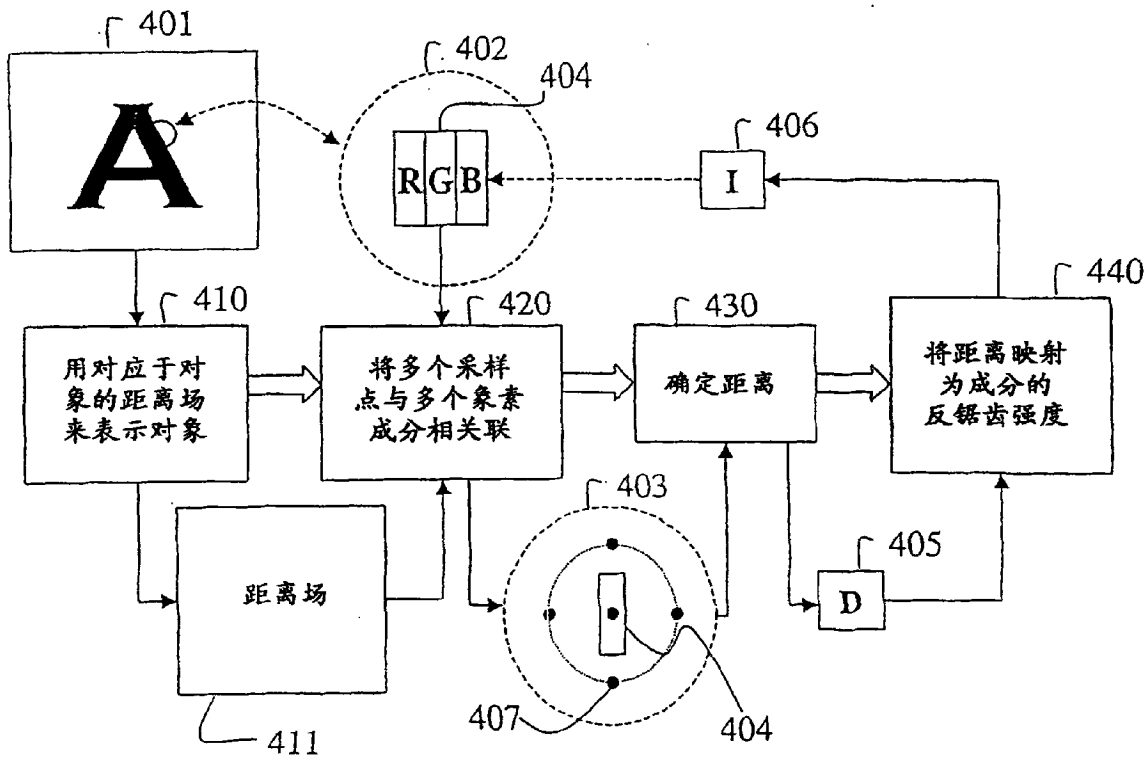
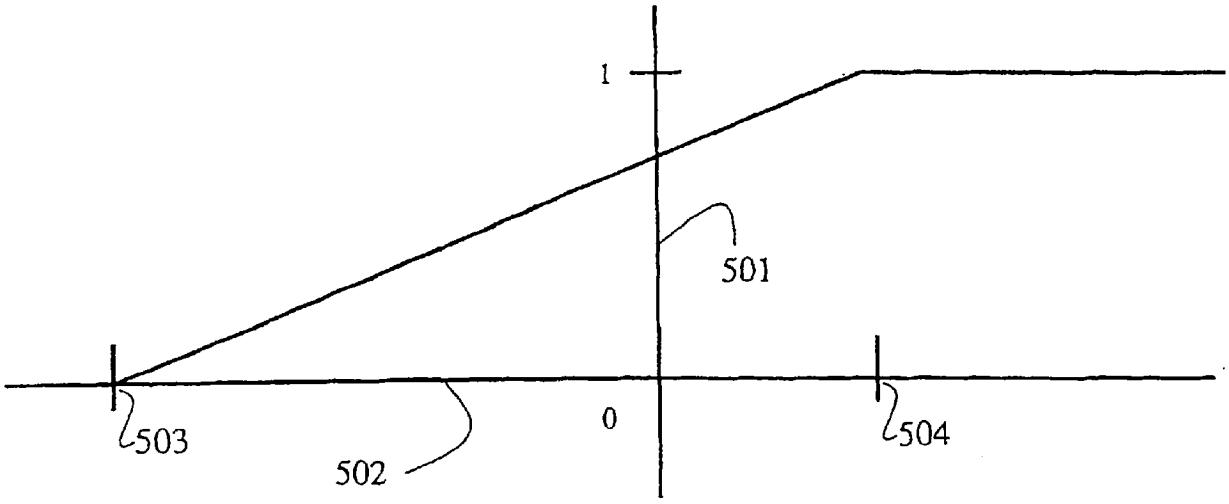

401
A
402
404
R G B
406
I
440
410
用对应于对象的距离场来表示对象
420
将多个采样点与多个象素成分相关联
430
确定距离
将距离映射为成分的反锯齿强度
403
距离场
405
D
407
404
411
400

图 4



500

图 5

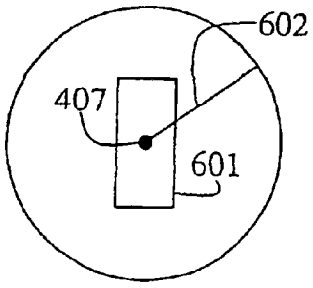


图 6A

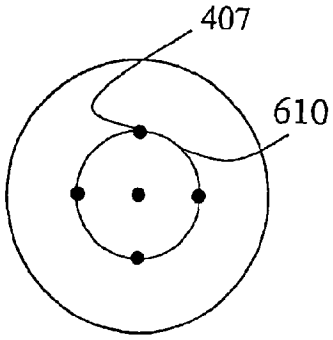


图 6B

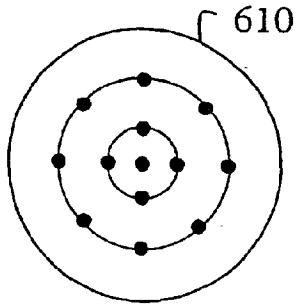


图 6C

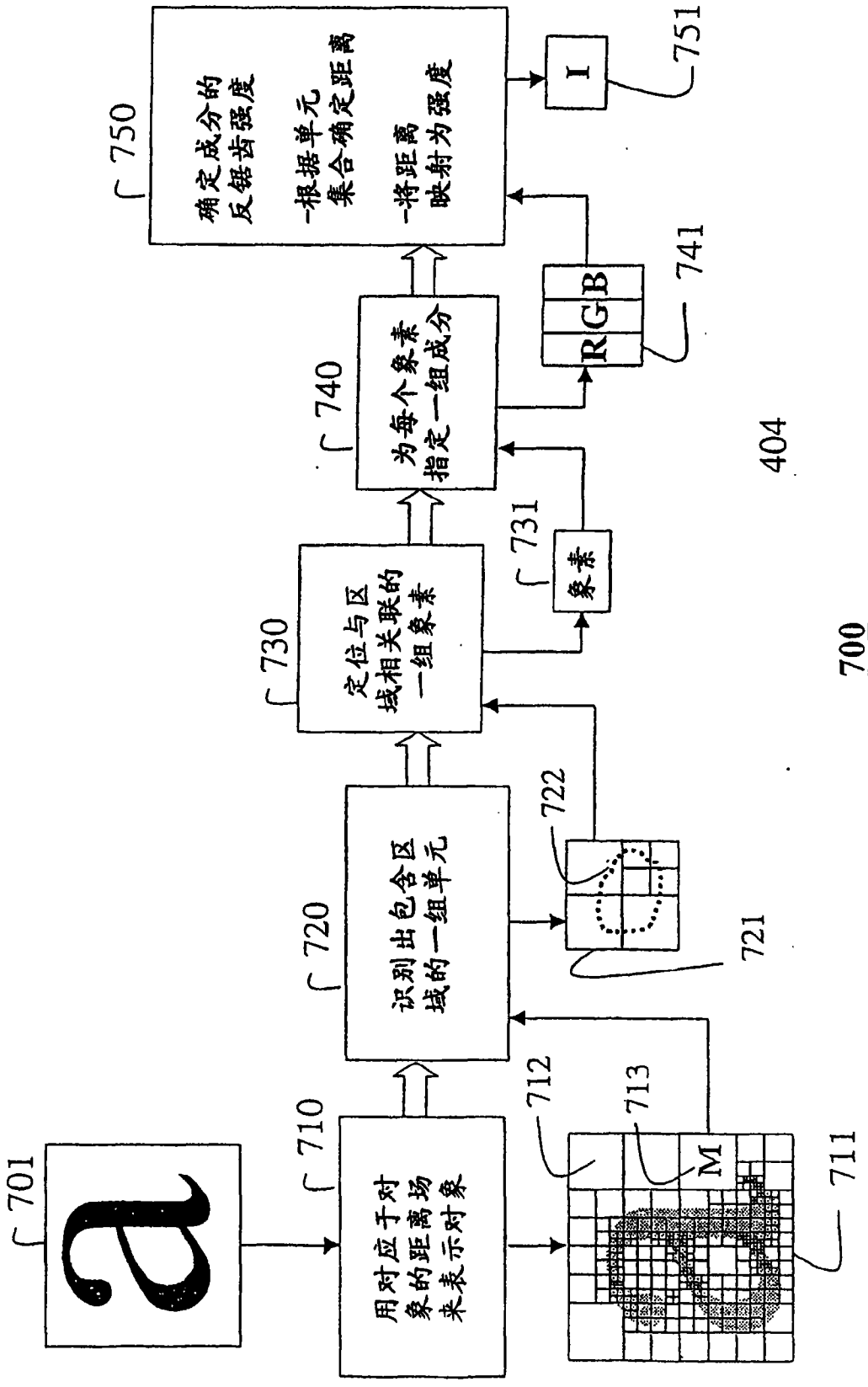
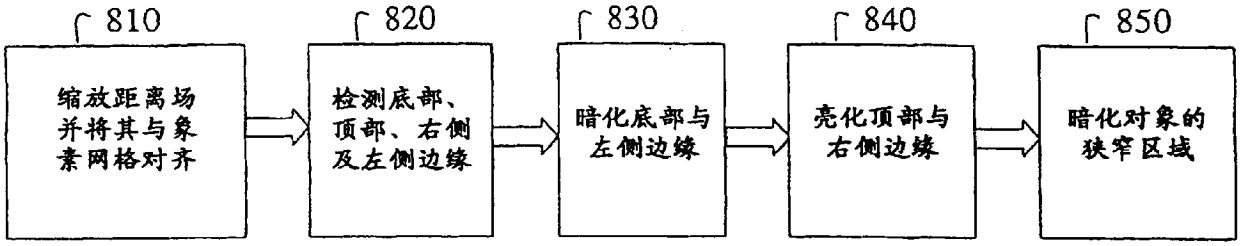
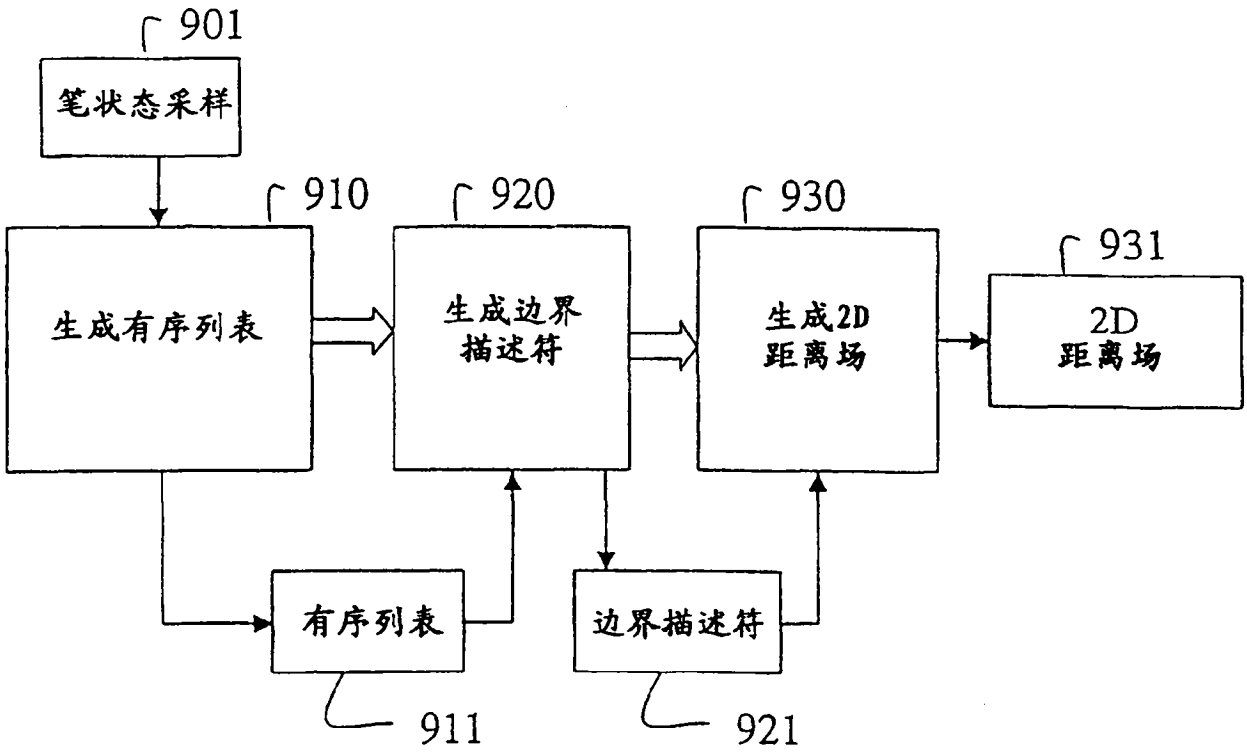

701
710
用对应于对象的距离场来表示对象
720
识别出包含区域的一组单元
730
定位与区域相关联的一组象素
740
为每个象素指定一组成分
750
确定成分的反锯齿强度
-根据单元集合确定距离
-将距离映射为强度
751
I
741
RGB
731
象素
404
700
712
713
M
711
721
722

图 7


810
缩放距离场并将其与象素网格对齐
820
检测底部、顶部、右侧及左侧边缘
830
暗化底部与左侧边缘
840
亮化顶部与右侧边缘
850
暗化对象的狭窄区域

800

图 8


901
笔状态采样
910
920
930
931
生成有序列表
生成边界描述符
生成2D距离场
2D距离场
有序列表
边界描述符
911
921

900

图 9

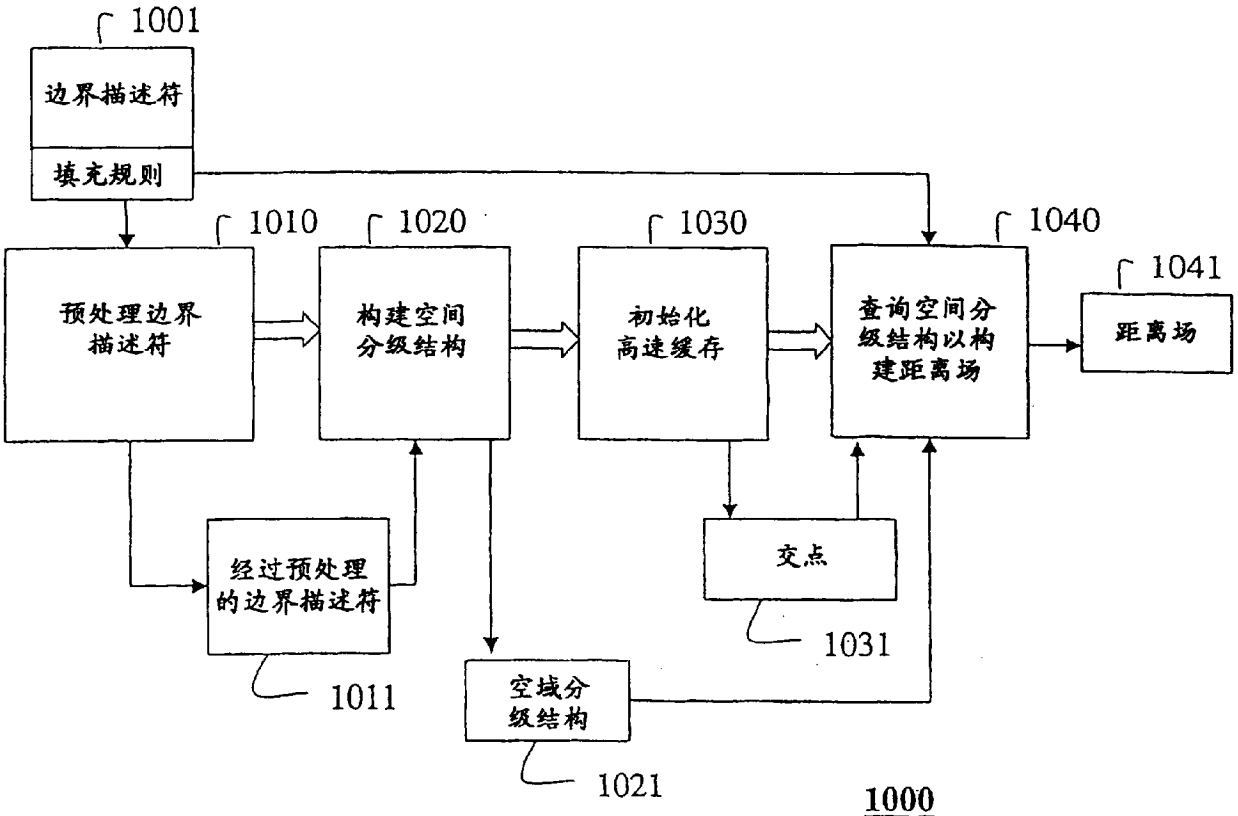

1001
边界描述符
填充规则
1010
1020
1030
1040
1041
预处理边界描述符
构建空间分级结构
初始化高速缓存
查询空间分级结构以构建距离场
距离场
经过预处理的边界描述符
交点
1031
1011
空域分级结构
1021
1000

图 10

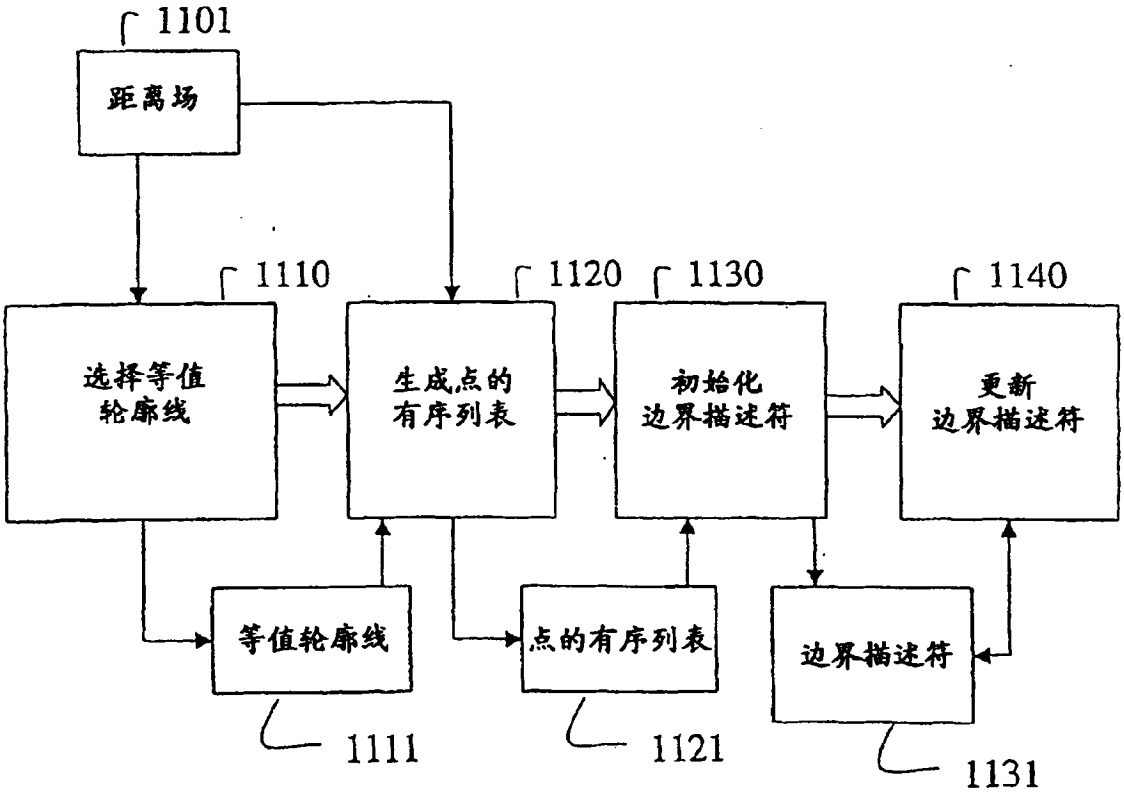
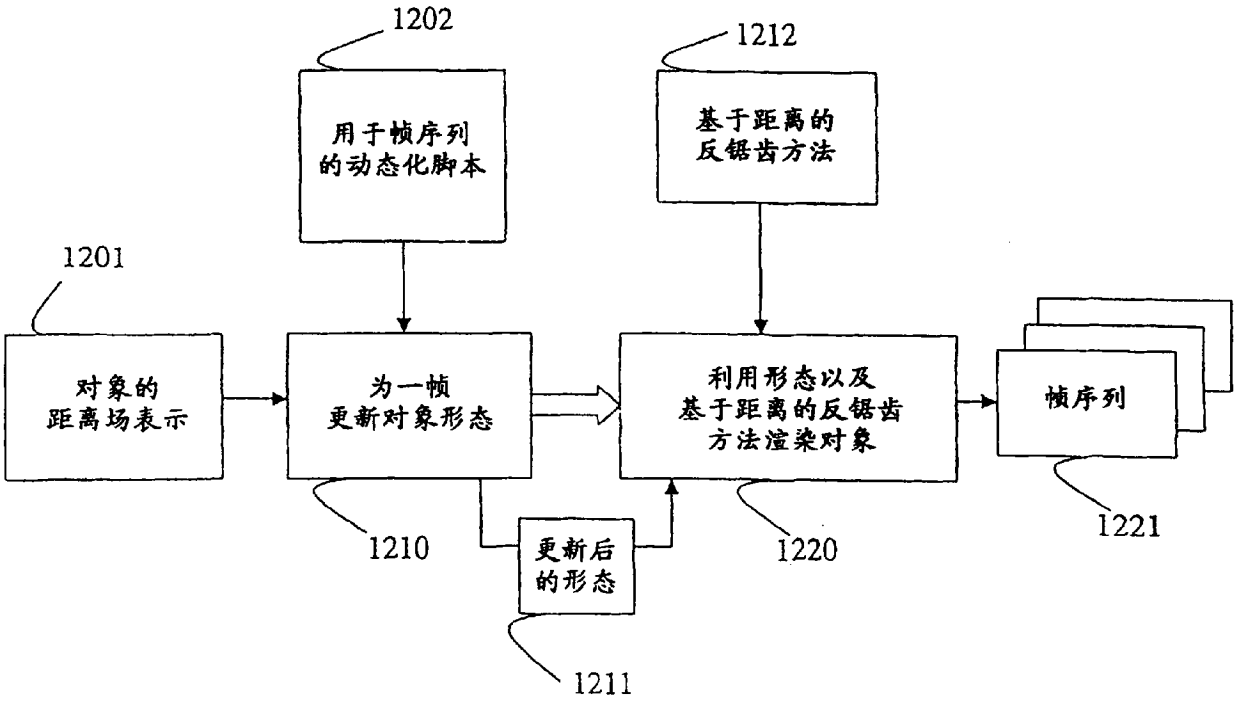
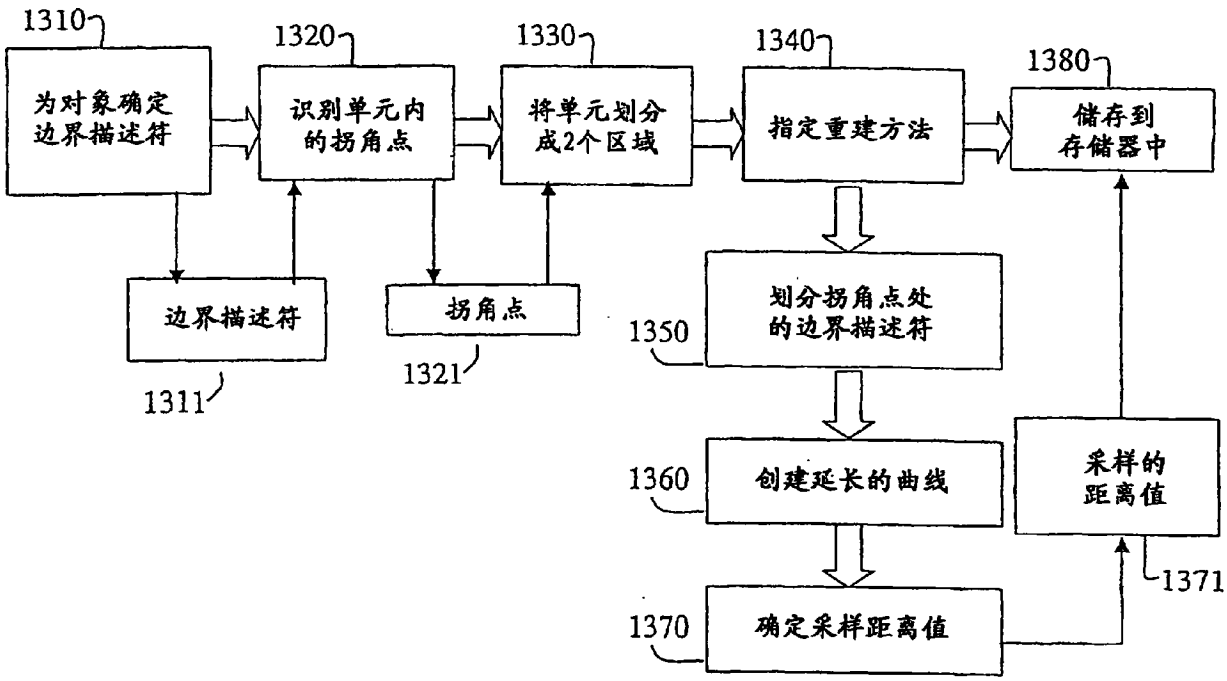

1101
距离场
1110
选择等值
轮廓线
1120
生成点的
有序列表
1130
初始化
边界描述符
1140
更新
边界描述符
等值轮廓线
1111
点的有序列表
1121
边界描述符
1131
1100

图 11


1202
用于帧序列
的动态化脚本
1212
基于距离的
反锯齿方法
1201
对象的
距离场表示
为一帧
更新对象形态
利用形态以及
基于距离的反锯齿
方法渲染对象
帧序列
1210
更新后
的形态
1220
1221
1211

1200

图 12


1310
为对象确定边界描述符
1320
识别单元内的拐角点
1330
将单元划分成2个区域
1340
指定重建方法
1380
储存到存储器中
边界描述符
1311
拐角点
1321
划分拐角点处的边界描述符
1350
创建延长的曲线
1360
采样的距离值
1371
确定采样距离值
1370

1300

图 13