

[19] 中华人民共和国国家知识产权局

[51] Int. Cl.

G06T 15/00 (2006.01)

H04N 13/00 (2006.01)

A63F 13/00 (2006.01)



# [12] 发明专利申请公布说明书

[21] 申请号 200380111047.5

[43] 公开日 2007年2月7日

[11] 公开号 CN 1910619A

[22] 申请日 2003.12.19

[21] 申请号 200380111047.5

[86] 国际申请 PCT/MX2003/000112 2003.12.19

[87] 国际公布 WO2005/059842 西 2005.6.30

[85] 进入国家阶段日期 2006.8.18

[71] 申请人 TD 视觉有限公司

地址 墨西哥墨西哥城

[72] 发明人 M·R·古铁雷斯诺韦洛

[74] 专利代理机构 北京纪凯知识产权代理有限公司

代理人 沙捷

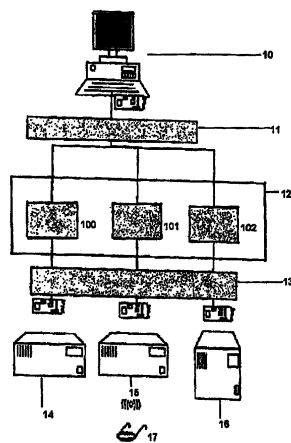
权利要求书 3 页 说明书 18 页 附图 13 页

[54] 发明名称

三维视频游戏系统

[57] 摘要

一种 3D 视频游戏系统，能够通过不同的、独立的 VGA 或视频信道利用以浸入方式共享存储器的显示装置来显示左 - 右序列。该系统具有视频游戏引擎，该引擎控制和确认图像视点，分配纹理、明暗分布、位置、移动和与参与游戏的各对象有关的方面；创建左和右后缓冲器，创建图像并呈现前缓冲器中的信息。该系统允许实时操纵与对象的图像的 xyz 坐标有关的数据的信息，增加左 - 右后缓冲器的 RAM，可以鉴别和占据对应后缓冲器，其信息发送到前缓冲器或以浸入方式共享存储器的另外的独立显示装置。



1、一种三维视频游戏系统，其能够利用处理三维图形要求代码的图形引擎来处理视频数据流，其特征在于包括：

- 对应于左眼和右眼创建一对缓冲器或物理存储线；
- 5 在存储器中创建左后缓冲器；  
鉴别其是否为 TDVison 技术；  
在存储器中创建右后缓冲器；  
设定右相机视图；  
在右后缓冲器中绘制图像，作为右相机当前矢量的函数；  
10 在右前缓冲器（屏幕）上显示图像；  
计算左对坐标；  
设定左相机视图；  
在左后缓冲器中绘制图像，作为左相机当前矢量的函数；  
通过一系列到一编程界面的图形驱动函数在 TDVison 屏幕上实时  
15 （左、右）或当前显示信息。

- 2、根据权利要求 1 所述的三维视频游戏系统，其特征在于，视频卡的后缓冲器或物理 RAM 临时且快速地绘制，而不将其输出到视频卡，通过三维旋转和平移目标，针对原点计算和分配距离、深度、和  
20 位置视点，对图像组进行处理。

- 3、根据权利要求 1 所述的三维视频游戏系统，其特征于，图形引擎控制和确认目标视点，分配纹理、明暗分布、位置、移动和与参与三维视频游戏的各对象有关的每个其它方面。

- 25 4、根据权利要求 1 所述的三维视频游戏系统，其特征在于，TDVision 视频游戏的软件实现以任何编程语言开发，所述编程语言使用游戏源代码和响应用户动作和事件的逻辑，通过诸如发送图像到表面显示的 OpenGL 和 DirecTX 的到编程界面内的一系列图形处理函数  
30 来编程。

5、根据权利要求1所述的三维视频游戏系统，其特征在于，设定左后缓冲器，鉴别其是否为 TDVision 技术，设定右后缓冲器。

6、根据权利要求1所述的三维视频游戏系统，其特征在于，在  
5 TDVision 后缓冲器上的绘制包括以下步骤：

创建左视图；

在左后缓冲器上绘制，作为相机位置的函数；

在左前缓冲器（屏幕）中显示图像；

鉴别其是否为 TDVision 格式技术；

10 计算右对的坐标；

在后缓冲器中绘制，作为左相机位置的函数；

在右前缓冲器（屏幕）中显示信息。

7、根据权利要求1所述的三维视频游戏系统，其特征在于，用于  
15 从后缓冲器到屏幕的图像显示的算法包括以下步骤：

清除后缓冲器；

获得后缓冲器的指针；

关闭后缓冲器；

重新绘制场景；

20 打开后缓冲器；

解开针对后缓冲器的指针；

鉴别其是否为 TDVision 格式；

在左屏幕中显示图像；

如果其为 TDVision 格式，则：

25 清除后缓冲器；

获得后缓冲器的指针；

关闭后缓冲器；

获得点坐标；

重新绘制场景；

30 打开后缓冲器；

解开后缓冲器的指针；

在右屏幕上显示图像。

8、根据权利要求 1 所述的三维视频游戏系统，其特征在于，数字视频的数据流是实时视频流，图形引擎包括用于在各自的监视器上实时显示左和右图像的数字视频流的控制。

9、三维视频游戏系统，其具有一个独立的附加 VGA 或视频显示装置（前缓冲器），其以浸入方式共享存储器，并能够为不同的独立信道来显示左和右序列。

10

10、根据权利要求 9 所述的三维视频游戏系统，其特征在于增加左后缓冲器的 RAM。

11、根据权利要求 9 所述的三维视频游戏系统，其特征在于增加右后缓冲器的 RAM。

15

12、根据权利要求 9 所述的三维视频游戏系统，其特征在于其具有鉴别和占用相应的后缓冲器的能力。

20

## 三维视频游戏系统

### 技术领域

- 5 本发明涉及三维电视图像的显示，更具体地涉及用于观看三维（3D）图像的硬件和软件设计，该硬件和软件设计易于集成到现有的电视、个人计算机与视频游戏系统装置。

### 背景技术

- 10 可视人机界面一直在努力改善广泛范围的应用中的图像，这些应用是：军事、生物医学研究、医用成像、基因操作、机场安全、娱乐、视频游戏、运算以及其它显示系统。

三维（3D）信息对于在要求逼真三维图像的重要任务中获得成功是关键的，三维图像为用户提供了可靠的信息。

- 15 立体视觉系统是基于人眼从两个不同视点（左和右）观察同一物体的能力。大脑合并两个图像，形成深度和体积感觉，该感觉然后被大脑转换为距离、表面和体积。

在现有技术中，为获得 3D 图像已做出许多尝试，例如，已应用下面的技术：

- 20 -红-蓝偏振  
-垂直-水平偏振  
-复合图像眼镜  
-3D 虚拟现实系统  
-体显示器  
25 -自动立体显示器

所有上述技术的表现方式互不相容，并伴随有副作用，且与目前现有技术缺乏相容性，即：

- 为进行观看，红-蓝偏振系统要求特殊的投影仪和大尺寸白屏幕；在观看几分钟后，副作用开始显现，如头痛、头昏眼花以及与利用三  
30 维效果显示的图像有关的其它症状。在电影放映系统中过去长时间地

使用该技术，但由于上述问题，该系统最终退出了市场。左眼和右眼接收的内容（一只眼接收蓝-偏振信息，另一只眼接收红-偏振信息）的很大差别引起副作用，造成视神经和大脑的过度紧张。此外，同时显示两个图像。为进行观看，该技术要求外部屏幕和使用偏振色彩眼镜。

5 如果用户不戴红-蓝眼镜，则不能看到三维效果，而只看到两个模糊的图像。

水平-垂直偏振系统合并由具有两个镜头的立体相机所拍摄的两个图像；左右图像分别具有水平和垂直偏振。在诸如迪斯尼与 IMAX3D 影院的一些新电影院中使用这些系统。该技术要求极昂贵的制作系统并限于专门和特定的观众，因而减小了市场和活动的领域。对三维(3D)内容的特殊关注在过去三年中一直在增长；例如 Tom Hanks 的作品和 Titanic，它们的 3D 内容是由 IMAX3D 技术来制作的。该技术在几分钟的显示后也给用户带来了副作用，要求外部屏幕和使用偏振眼镜；

10 如果用户不戴这些眼镜，则只能看到模糊的图像。

利用复合图像遮光眼镜技术的系统通过阻挡左右图像之一而轮流切换左右图像，这样图像在短时间中不能达到相应的眼睛。这种阻挡是与（在监视器或电视机中）图像显示同步。如果用户没戴眼镜，只看到模糊的图像，并在几分钟后副作用变得明显。BARCO 系统公司（和其它公司）目前为 Mercedes Benz<sup>®</sup>、Ford<sup>®</sup>和 Boeing<sup>®</sup>公司提供该技术，

15 通过提供一种“场所（room）”来创建利用复合（关闭器眼镜）的 3D 图像，从而在将其在生产线上组装之前制成其原型。

3D 虚拟现实系统（VR3D）是基于计算机的系统，这些系统创建可通过诸如数据手套（data glove）和位置检测器的位置接口与用户进行交互的计算机场景。这些图像是计算机生成的，并使用矢量、多边形和基于单眼深度复制的图像，以模拟由软件所计算的深度和体积，

25 但是图像是利用放置在眼前的作为显示装置的头盔来表现的；用户沉浸在计算机生成的仅存在于计算机而不存在于真实世界的场景中。该计算机生成的场景的名字是“虚拟现实”。该系统要求非常昂贵的计算机，如 SGI Onyx Computers<sup>®</sup>的 SGI Oxygen<sup>®</sup>，这超出普通用户所承受的范围。真正的游戏和仿真是利用该技术创建的，该技术通过相同的

30 VGA 或视频信道生成左-右序列，软件包括用于在屏上显示时间中以

60Hz 的频率轮流切换视频图像的特定指令。视频游戏软件或程序直接与图形卡相互作用。

存在有一种被叫作 I-O 系统的技术,其以左-右复合系统和以 80 到 100Hz 的频率轮流切换图像的方式在双目屏幕中显示复合图像,但是  
5 尽管这样,也感觉到闪烁。

仅有少数几个制造商(如 Perspectra 系统公司)制造体显示系统<sup>®</sup>。他们利用了人眼保留图像几毫秒的能力和很高速旋转的显示器;然后,根据观察角度,由于显示器的高速旋转,装置显示出打开和关闭像素色彩的相应的图像,眼睛可接收“浮动图像(floating image)”。这些系  
10 统是非常昂贵的(“球状物(sphere)”花费大约 50000 美圆)并要求特殊的和适当的软件与硬件。该技术目前用于军事应用。

自动立体显示器是具有从顶到底的半圆柱形线条的监视器,并只用于前面和后面图像;这不是真正的三维,而只是在两个透视平面中的仿真。Philips<sup>®</sup>以及 SEGA<sup>®</sup>目前正研究该三维技术,以获得技术优势。  
15 结果是不乐观的,并存在 50%的分辨率损耗。该技术与目前的技术基本构架不相容,并要求彻底更换用户的监视器。对于不是为该技术特别创建的应用,在显示时是模糊的,使这些应用完全与当前基本构架不相容,造成不方便。为观看 3D 图像,观察者需要位于约 16" (40.64  
20 厘米)的距离处,这根据监视器的大小而变化,并且观察者必须垂直地看屏幕的中心,把他/她的视线固定在实际屏幕之外的焦点上。只要视线稍有偏离或视线角度有变化,就失去三维效果。

在现有技术中,存在涉及该技术研究的多项专利,即:

授予 Timothy Van Hook 等人的 2003 年 7 月 15 日的 USP No. 6593929 与 2003 年 4 月 29 日的 USP No. 6556197,其涉及低成本的视  
25 频游戏系统,该系统模仿三维世界,并投影到二维屏幕上,图像利用游戏控制器而基于用户实时的可交替视点。

授予 Claude Comair 等人的 2003 年 7 月 8 日的 USP No. 6591019,其使用压缩和解压缩技术,用于将矩阵转换为计算机产生的 3D 图形系统,该技术包括在矩阵内的零搜索过程中将实数矩阵转换为整数矩阵。  
30 压缩后的矩阵在存储器中占有小得多的空间,3D 动画可以实时地以有效率的方式解压缩。

授予 David Reed 等人的 2003 年 4 月 1 日的 USP No. 6542971, 其提供存储器访问系统和一种方法, 该方法不使用辅助存储器, 而使用这样的系统, 即该系统具有附接到存储器的存储空间, 该存储器一次读取和写入从一个或多个外设输入的数据。

5 授予 Stephen Morein 等人的 2002 年 12 月 10 日的 USP No. 6492987, 其描述了用于处理不被表现的对象的元素的方法和装置; 通过比较一个对象的至少一个元素的几何属性与根据像素群的代表性的几何属性而开始。在表现对象的元素的过程中, 确定新的代表性的几何属性, 并利用新的值来更新。

10 授予 Vimal Parikh 等人的 2002 年 9 月 24 日的 USP No. 6456290, 其提供图形系统界面, 用于使用和学习程序的应用。该特征包括顶点的唯一表示, 该表示允许图形线条来保留顶点状态信息, 设定投影矩阵和浸入式缓冲器帧命令。

任何视频游戏是以某一计算机语言编写的软件程序。其目标是模拟不存在的世界和将游戏者或用户带入该世界, 大多数集中于在竞争和改进(难度级)环境中增强视觉和手控的灵巧、图案分析和决策, 并且以具有高艺术性内容的大场面来表现。作为游戏引擎, 大多数视频游戏被分成下面的结构: 视频游戏, 具有相关的图形和音频引擎的游戏库, 图形引擎包含 2D 源代码和 3D 源代码, 并且音频引擎包含效果  
15 和音乐代码。以被称为游戏循环的周期性方式执行提到的游戏引擎的各个块, 这些引擎和库的每一个负责不同的操作, 例如:

图形引擎: 通常显示图像

2D 源代码: 在视频游戏屏幕上出现的静止图像、“背景”和“精灵”。

25 3D 源代码: 作为独立实体并利用计算机生成的世界中的 xyz 坐标来处理的动态、实时矢量控制的图像。

音频引擎: 声音回放

效果代码: 当特殊事件发生时, 如爆炸、冲撞、跳跃等

音乐代码: 通常根据视频游戏的气氛播放的背景音乐。

30 循环方式下的所有这些块的执行允许当前位置、条件和游戏规格(metrics) 的确认; 作为该信息的结果, 影响集成视频游戏的元素。



为游戏控制台创建的游戏程序与计算机之间的不同是，最初 IBM PC 不是用于玩乐而创建的；可是，许多最好的游戏运行在 IBM PC-兼容技术下。如果我们对过去的 PC 与目前的视频游戏和处理能力进行比较，则我们可以说，这些 PC 是十分过时的，并且仅是以创建初级游戏的低级控制（汇编语言）的方式，直接利用计算机的图形卡和扬声器。形势已经改变，目前 CPU 的处理动力和图形能力以及专门为图形处理加速（GPU）而设计的卡的创建已发展到它们迄今超越了八十年代的所谓超级计算机特性的程度。

在 1996 年，引入了称为 HARDWARE ACCELERATION 的图形加速系统，该系统包括能够以高速进行数学与矩阵运算的图形处理器，因而通过专用卡的通信和位于称为 HAL（硬件提取层）的层中的编程语言来降低主 CPU 的负担，这通过坐标矩阵和诸如相加、纯量乘法和浮点矩阵比较的矩阵数学运算而允许与实时 xyz 坐标有关的数据的信息控制。

15

## 发明内容

本发明的目的是解决三维图像显示技术的不相容问题。

本发明的另一个目的是提供允许最终用户观看视频图像、计算机图形、视频游戏以及利用相同装置的仿真的多用途技术。

20 本发明的另外目的是提供一技术，即使连续使用多个小时，在观看本技术提供的三维图像之后，可以不产生副作用。

本发明的另一目的是提供一种软件中的高技术的集成，这是通过创建对应于左眼和右眼的一对缓冲器，具有附加独立的以浸入形式共享存储器的显示装置的硬件，以及数字视频图像处理器来实现的。

25 本发明的另一目的是通过由图形处理单元或 GPU 创建两个前缓冲器而在屏幕上物理地显示出图像。

本发明又一目的是获得具有高度逼真图像的深度和体积的大脑感觉，即使这些图像是由计算机图形软件创建。

30 本发明又一目的是提供 TDVision<sup>®</sup>算法，以创建高度逼真的计算机图像。

本发明的另一目的是在当前技术基础中作出改变，利用光学技术

创建新的数字成像处理，以通过设定右侧相机的视图获得真实图像感觉。

本发明的另一目的是获得数字媒体的整合，其中播放 DVD 的计算机、制作电影的膝上型电脑、互联网的视频图像发送能力以及 PC 与视频游戏控制台都可用于互联网结构。

本发明的另一目的是提供汇编语言算法、模拟和数字硬件，以获得对现有技术的 3D 装置的最佳适配性。

本发明的又一目的是提供三维视觉的计算机系统，用于通过动画、显示器与软件模仿来产生立体图像。

10

## 附图说明

图 1 示出 TDVision<sup>®</sup> 视频游戏技术示意图。

图 2 示出基于现有技术的视频游戏的主要结构。

图 3 示出用于构建空间中一定位置的对象的基本三维元素。

图 4 示出基于 OpenGL 和 DirectX API 函数技术的视频游戏的开发纲要。

图 4a 示出用于创建左和右缓冲器的算法以及另外地鉴别是否使用 TDVision 技术的框图。

图 4b 示出用于绘制右后缓冲器中的图像之后设置右相机视图作为右相机矢量的函数的子程序的框图，还鉴别该子程序是否使用 TDVision 技术格式。

图 5 示出对编辑 TDVision 技术所需要的图形适配器进行修改的运算纲要的框图，另一方面，其容许进行通信并包含编程语言，以及容许与设定的图像有关的数据的信息控制。

图 6 表示允许在 TDVision 后缓冲器中绘制信息并把它以 DirectX3D 格式呈现在屏幕上的算法的框图。

图 7 示出利用 OpenGL 格式的显示序列。

图 8 示出通过利用 OpenGL 算法的左右后缓冲器的屏幕信息显示的框图。

图 9 示出 TDVision 技术使用的视频卡中需要的改变。

30

## 具体实施方式

5 视频游戏是多个处理，其通过提供多个独立相关的包括一组编程选项的逻辑状态而开始，其中各编程选项对应于不同的图像特征。通用程序指令可由多个计算装置编辑为代码，而没有必要为各装置独立地生成对象代码。

10 诸如个人计算机、膝上型电脑、视频游戏等的计算机装置包括中央处理单元、存储系统、视频图形处理电路、音频处理电路和外围端口。典型地，中央单元对软件进行处理以参照待显示的图像生成几何数据，并将该几何数据提供给视频图形电路，该视频图形电路生成存储

10 存储在存储帧中的像素数据，其中将信息发送给显示装置，上述元素整体被称为视频游戏引擎。

15 一些游戏引擎授权给第三方，如同在 Quake III Arena 程序的情形下，其具有 QUAKE ENGINE 游戏引擎；该引擎授权给使用 quake 引擎的 VOYAGER ELITE FORCE 游戏。这样，游戏开发者可集中在游戏规格上，而没必要从零开始开发游戏引擎。最初，视频游戏仅使用被称

15 做“精灵”的二维图像，它们是游戏的主角。

大多数视频游戏和技术发展且容许与三维环境或世界中的模拟对象一起工作，把 xyz 位置属性给予各对象，由具有相同特性的其它对象包围，并且在具有 (0, 0, 0) 原点的世界内一起动作。

20 起先，与计算机世界分离的视频游戏控制台采取了第一步，把 3D 图形包含进来，作为后来 PC 中使用的硬件所采用的所述装置、技术的物理图形能力。通常被称作视频游戏应用人工智能的条件分析元素也包含进来，该元素分析该情形、位置、碰撞、游戏风险与优势，并基于该分析，针对参与视频游戏的各对象生成响应动作。

25 使用后缓冲器，后缓冲器是待显示的图像将被临时“绘制”，而不输出到视频卡的存储位置。如果这是直接在视频存储屏幕上完成，将观察到屏幕上的闪烁；因此，绘制信息，并快速在后缓冲器中处理。该后缓冲器通常位于视频卡或图形加速卡内的物理 RAM 存储器中。

视频游戏的算法内的典型序列将是：

- 30
- 1) 显示标题屏幕
  - 2) 把字符、对象、纹理和声音载入存储器

- 3) 创建临时处理的存储位置，称为双缓冲器或后缓冲器
- 4) 显示背景
- 5) 记录各元素参与游戏情况下的图像
- 6) 从存储器（双缓冲器）中清除全部元素
- 5 7) 用户输入验证和游戏者位置更新
- 8) 利用人工智能（AI）的敌人位置处理
- 9) 移动各参与的对象到其新位置
- 10) 对象碰撞验证
- 11) 动画帧增加
- 10 12) 绘制后缓冲器存储器中的对象
- 13) 把后缓冲器数据传送到屏幕
- 14) 返回步骤 5)，除非用户想结束游戏（步骤 15）
- 15) 从存储器删除全部对象
- 16) 结束游戏

15 在视频游戏控制台中最常用的装置是：CPU 或中央处理单元，其处理游戏循环、来自作为游戏键盘或操纵杆的键盘、鼠标或游戏装置的用户输入以及游戏的人工智能处理。

GPU 或图形处理单元控制多边形模仿、纹理映射、变换以及明暗分布仿真。

20 音频 DSP 或数字信号处理器控制背景音乐、声音效果和 3D 位置声音。

图形引擎是游戏部，游戏部针对视频游戏控制台或 PC，负责控制和确认视点、指定纹理（金属、皮肤等）、明暗分布、位置、移动和参与视频游戏的各对象有关的各其它方面。相对于指定的原点来处理该  
25 设定的图像，并计算距离、深度和位置视点。这在两步中进行，但由于涉及的数学运算，使得它是复杂的处理，即对象平移处理（自原点偏移）以及对象旋转处理（相对于当前位置的旋转角）。

注意，重要的是，最小图像单元（图 3）由被称为“顶点”的最小控制单元组成，其表示 xyz 空间中的一点。允许的最小几何单位是在  
30 空间最少由三点构建的三角形。利用三角形基本单位形成较大的对象，其包括数以千计的较小的三角形，作为 Mario Sunshine 字符。该表现方

式被称为“网格 (Mesh)”和纹理，色彩甚至图形显示特征可以与各网格或甚至于各三角形相关联。该信息被称为 3D 图形。十分重要的是，要提到，即使由于其由 xyz 矢量构建的性质而将它称为 3d 图形，给用户的最终显示一般是利用基于 3D 矢量的内容在平面引擎中以 2D 显示，在用户看来，这些 3D 矢量仿佛是在用户的前面，它们只表现为具有某种智能深度和明暗分布特征，但对于大脑来说，这些 3D 矢量不表现为在空间中具有体积。

最初，对于视频游戏程序来说，必须直接与图形卡进行通信，以执行加速和复合数学运算，这意味着实际上必须重写游戏以支持不同的视频卡。针对该问题，Silicon Graphics®集中于开发与硬件直接通信的软件层 (OpenGL®)，其具有一系列有用的函数和子程序，这些函数和软件层独立于硬件，与其仅在图形方面进行通信。Microsoft®也开发被称为 DirectX 3D 的类似的函数组，很像 OpenGL®但具有更完整的功能，因为其包括声音控制和网络游戏区域以及其它功能。

这些函数和子程序组被称为图形应用程序设计接口 (GRAPHICS API)。这些 API 可根据不同的编程语言来访问，如 C、C++、Visual .Net、C#和 Visual Basic 以及其它编程语言。

提到的每个虚拟现实系统目前通过相同的 VGA 或视频信道方案使用左-右序列，这些系统要求包括用于在后缓冲器中在屏幕上显示时间交替视频图像的特定指令的软件，其应用使用偏移量和类似仿真角度的已知的偏移算法。

除了 OpenGL®和 DirectX® API 提供的函数之外，一系列图形控制函数可在由 Windows®提供的被称为 WINDOWS API 的应用程序设计接口内实现。

基于这些技术的视频游戏程序的开发在图 4 中示出，其中包括由 TDVision®公司开发实现的目前应用中的视频游戏软件。图 4 示出以具有视频游戏的适当的规格的软件实现开始 (40) 的流程图的示意图，该软件用任何合适的编程语言 (如 C、C++、Visual Basic 及其它) 来开发 (41)，加入视频游戏 (42)、游戏逻辑和对象特征、声音、事件等的源代码 (43)，在 (44) 中定位事件选择器，这是通过 Windows API (45)、OpenGL (46) 或 DirectX (47) 的方式来实现的，并且最终发

送到视频显示 (48)。

所有这些都参照软件，需要注意的是 DirectX 提供许多函数，而 Microsoft®已经可以做到，即使最初一些函数要求特定的硬件，DirectX API 自身也能够通过软件来仿效硬件特性，如同硬件实际存在。

5 本发明最大化并且最优化 OpenGL®和 DirectX®技术的使用，导致具有特定特性、算法和数字处理的软件，以满足由本应用中使用的 TDVision 设定的规范。

对于硬件，Hal 和直接接口可由每个卡的驱动器来分析，为了实现 TDVision 技术，必须分析最小规范和要求，以及容许在 TDVision 的 10 3DVisors 中获得真实 3D 的技术中的任何可能变化。

对于显示或表现系统，由软件生成的和存储在 Graphic Device Context 和 Image Surface 中的信息直接传送到图形卡的末级，其把数字视频信号转换为模拟或数字信号（取决于显示监视器），然后，在屏幕上显示图像。

15 当前的显示方法是：  
具有数字计算机信号的模拟监视器，  
数字监视器，  
具有 TV 信号的模拟监视器，  
3D 虚拟现实系统。

20 输出类型取决于视频卡，该视频卡应该连接到兼容的监视器。

图 4a 示出用于临时图形处理的存储位置（左和右的后缓冲器）的创建，其中基本上它增加额外的存储位置，即，在 (400) 中设置右缓冲器，并且在 (401) 中鉴别是否存在 TDVision 技术；在肯定的情况下，其在(402)中设置左缓冲器，并且在(403)结束；当不存在 TDVision 25 技术时，处理在 (403) 结束，因为不存在要鉴别的事情。

图 4b 示出用于左相机和右相机图像的鉴别和显示的流程图；在 (410) 中设置左视图，在左后缓冲器 (411) 中作为相机位置的函数来绘制图像，该图像显示在左屏幕 (412) 中，然后，在 (413) 中鉴别图像是否具有 TDVision 格式，并且在肯定情况下计算右视图位置坐标 (414)，绘制作为左相机位置的函数的右后缓冲器中的图像 (415)， 30 然后，图像显示在右屏幕中 (416)，如果当以目前的现有技术格式提

供图像时就不须在（413）中进行鉴别，则处理在（417）结束，子程序跳到最后阶段（417），并且结束，因为没有必要计算其它坐标和显示并行信息。本应用的新颖部分指的是图 5 中示出的图形处理单元（GPU HARDWARE（GPU 硬件）），以及图形引擎（GRAPHICS ENGINE（图形引擎），SOFTWARE（软件））

对硬件的修改是：

- 针对左与右的后缓冲器，RAM 增加，
- 在显示缓冲器中实现另外独立的显示装置，但以浸入方式共享存储器，使它占有相应的后缓冲器。

在该情况下，必需的是，后缓冲器的 RAM 存储器和视频卡的前缓冲器足够大，以同时支持左右信道。这要求最小必需具有 32MB，以支持各具有 1024x768x4 色彩深度字节的深度的四个缓冲器。另外，视频输出信号是双重的（两个 VGA 端口），或它具有控制多个监视器的能力，如同 ATI RADEON 9500<sup>®</sup>卡的情况，其具有两个输出显示系统，从中选择的一个为 VGA，另一个为 S-Video 视频端口。创建具有双重输出的图形卡，只满足每个左-右信道每秒 60 帧的显示，以连接到 3DVisor，这些输出是 SVGA，S-Video，RCA 或 DVideo 类型的输出。

如在图 5 中描述的 CPU（50），存储驱动器（52），扩展存储器（52），提出具有针对 TDV 编辑的修改的计算方案，该存储器馈入音频驱动器（53）和扬声器（54），还有，输入和输出驱动器（55）反过来控制磁盘的端口（56）和与用户（57）互动的元件，如鼠标、键盘、游戏键盘和操纵杆；另一方面，图形驱动器直接与监视器（59）以及三维脸罩 3DVISORS（59b）互动。

具体来看图形硬件（HAL），需要作出变化以利用 TDVision 技术来编辑，由于图形硬件支持（502）而发送信息到图形驱动器（501）进行操作的应用程序（500）有效地需要利用 TDVision 技术来编辑的物理变化。为实现利用 OpenGL 和 DirectX 的 TDVision 技术，必需在视频游戏的软件部的一些部分中作出修改，并如同先前提到的，在一些硬件部中作出修改。

对于软件，必需在典型工作算法内添加一些特别特征，以及调用 TDVision 子程序，如在图 6 中所示。

- 载入表面信息 (600),  
载入网格信息 (601),  
创建 TDVision 后缓冲器 (602), 其中在存储器创建左后缓冲器,  
如果是 TDVision 技术, 则在存储器创建右后缓冲器。
- 5     应用初始坐标 (603),  
      应用游戏逻辑 (604),  
      确认和人工智能 (605),  
      位置计算 (606),  
      碰撞确认 (607),
- 10     绘制 TDVision 后缓冲器中的信息和在屏幕上的显示 (608), 其中  
      必须设定右相机视图, 在右后缓冲器中绘制图像作为当前右相机矢量的  
      函数, 并在右屏幕 (前缓冲器) 上显示图像。如果是 TDVision 技术,  
      则: 计算左对坐标, 设定左相机视图, 绘制作为左相机当前矢量的函  
      数的左缓冲器中的图像, 并在右屏幕 (前缓冲器) 上显示要求硬件修  
15     改的信息。

这样, 创建对应于左眼和右眼的一对缓冲器, 其中当在游戏循环  
中评估时, 得到对应于下面示出的各右相机 (当前) 和左相机 (以  
SETXYATDV 函数计算的补数 (complement)) 的可视化的矢量坐标。

- 重要的是, 要提到, 所述屏幕输出缓冲器或前缓冲器从一开始就  
20     被分配给视频显示表面 (设备上下文) 或分配给关注 (表面) 的表面,  
      但为了以 TDVision 3Dvisor 显示信息, 必需物理地呈现出两个视频输  
      出, 即呈现右输出 (正常 VGA) 和左输出 (另外的 VGA, 数字补码或  
      S-Video), 以与 TDVision 兼容。在示例中使用 DirecTX, 但可应用相  
      同的处理和概念到 OpenGL 格式。

- 25     图 7 表示有效地进行图形应用通信接口的显示线的算法 (70) 的  
      纲要, 通过利用了顶点运算 (77) 的三角法 (72) 而构建图像 (71)  
      并通过像素运算或根据命令 (73) 的图像元素 (75), 显示列表 (74)  
      和指定纹理给图像 (76) 的存储器导致显示由运算 (79) 被发送到存  
      储器帧 (70F)。Windows 软件 (700) 与 (702) 和图形语言卡 (701)  
30     通信, 这反过来包含对馈入 (703) 和 (704) 是有用的图形信息库。

图 8 表示利用 OpenGL 算法 (80) 以显示对象的左和右图像的



TDVision 技术, 它清除后缓冲器 (81), 得到针对后缓冲器的指针 (82), 关闭后缓冲器 (83), 重新绘制场景 (84), 打开后缓冲器 (85), 解开后缓冲器指针 (86), 发送图像到左显示表面; 在 (800) 它鉴别是否为 TDVision 技术并在肯定的情况下, 它清除存储器 (801) 并得到针对后缓冲器的指针 (802), 关闭后缓冲器 (803), 得到新视点的坐标 (804), 重新绘制场景 (805), 打开存储器 (806), 解开后缓冲器指针 (807), 并发送图像到右显示表面 (808)。

图 9 表示在视频卡中为编辑 TDVision 技术所必需的变化 (90), 即正常的一级左后缓冲器 (92) 之前的正常的左后缓冲器 (91), 正常的一级左后缓冲器反过来连接到监视器的 VGA 输出 (95), 并应具有另一个 VGA 输出, 这样它可接收一级的右缓冲器 (94), 一级右缓冲器反过来具有作为前一个的 TDVision 技术的后缓冲器。左和右两个后缓冲器可连接到具有双重 VGA 输入的 3DVisor (96), 以接收和显示后缓冲器 (91) 与 (93) 所发送的信息。

该软件修改使用下面的在 Direct X 中的 API 函数:

TDVision 后缓冲器创建:

```

FUNCTION CREATE BACKBUFFERTDV()
    Left buffer
    Set d3dDevice =
20   d3d.CreateDevice(D3DADAPTER_DEFAULT, _
                   D3DDEVTYPE_HAL, hWndL, _
                   D3DCREATE_SOFTWARE_VERTEXPROCESSING,
25   d3dpp)
    If GAMEISTDV then
        Right Buffer
        Set d3dDeviceRight =
30   d3d.CreateDevice(D3DADAPTER_DEFAULT, _
                   D3DDEVTYPE_HAL, hWndR, _
                   D3DCREATE_SOFTWARE_VERTEXPROCESSING,
d3dpp2)

```

```

        Endif
        END SUB
        Draw image in TDVision backbuffer:
        FUNCTION DRAWBACKBUFFERTDV()
5           DRAW LEFT SCENE
           d3dDevice.BeginScene
           d3dDevice.SetStreamSource0,    poly    1_vb,
Len(poly1.v1)
10           d3dDevice.DrawPrimitive
D3DPT_TRIANGLELIST,0,1
           d3dDevice.EndScene

15           Copy    backbuffer    to    frontbuffer,
screen
           D3dDevice.Present By Val 0,By Val 0,
0, By Val 0

20           'VERIFIES IF IT IS A TDVISION PROGRAM
BY CHECKING THE FLAG
           IF GAMEISTDV THEN
               'CALCULATE    COORDINATES    RIGHT
CAMERA
25           SETXYZTDV ()
               ' Draw right scene
           d3dDevice2.BeginScene
           d3dDevice2.Set StreamSource 0,
30 poly2_vb, Len(poly1,v1)
           d3dDevice2.DrawPrimitive
D3DPT_TRIANGLELIST,0,1
           d3dDevice2.EndScene
35

```

```

d3DDevice2.Present ByVal 0,
    ByVal 0, 0, ByVal
        END SUB.
        Modifications to xyz camera vector:
5           VecCameraSource.z = z position
           D3DXMatrixLook    AtLH    matView,
           vecCameraSource, _
           VecCameraTarget,    CreateVector
10          (0,1,0)
           D3DDevice 2.SetTransform D3DTS_VIEW,
           matView
           VecCameraSource.x = x position
           D3DXMatrixLook    AtLH    matView,
15          vecCameraSource, _
           VecCameraTarget,    CreateVector
           (0,1,0)
           D3DDevice 2.SetTransform D3DTS_VIEW,
20          matView
           VecCameraSource.y = y position
           D3DXMatrixLook    AtLH    matView,
           vecCameraSource, _
25          VecCameraTarget,    CreateVector
           (0,1,0)
           D3DDevice 2.SetTransform D3DTS_VIEW,
           matView

```

30 因此，创建对应于左眼和右眼的一对缓冲器，其中当在游戏循环中评估时，通过通常的坐标变换等式而得到对应于右相机和左相机的可视化的矢量坐标（以 SETXYZTDV 函数计算的补码）。

重要的是，要提到，所述屏幕输出缓冲器或前缓冲器从一开始就被分配给设备上下文（device context）或分配给关注的表面，但为了以

TDVision 3Dvisor 显示信息，必需物理地呈现两个视频输出，即呈现右输出（正常 VGA）和左输出（另外的 VGA，数字补码或 S-Video），与 TDVision 兼容。

5 在示例中使用 DirectX，但针对图 8 所示的 OpenGL 格式可应用相同的处理和概念。

在该情况下，必需的是，后缓冲器的 RAM 存储器和视频卡的前缓冲器足够大，以同时支持左右信道。这要求最少必需具有 32MB，以支持各具有 1024x768x4 字节的色彩深度的四个后缓冲器。如上文所述，视频输出信号必须是双重的（2 个 VGA 端口），或它具有控制多个监视器的能力，如同 ATI RADEON 9500®卡的情况，其具有两个输出显示系统，可从中选择一个 VGA，一个 S-Video 和一个 DVideo 端口。

创建具有双重输出的图形卡，只满足每个左-右信道每秒 60 帧的显示，以连接到 3DVisor，这些输出可以是 SVGA, S-Video, RCA 或 DVideo 类型的输出。

15 因此，可获得对应于左和右两个视点中的相机视点的图像，并且在不需复合和实时显示的情况下硬件将识别待显示在两个不同且独立的视频输出中的信息。目前，所有的技术使用复合和软件仿真，在本申请所提出的技术中可获得真实的信息，并且当利用 3DVisors 时，从两个不同的视点显示图像，大脑将关联图像在空间中所占有的体积，

20 而没有与当前现有技术有关的在屏幕上闪烁的副作用。

副立体相机的坐标计算方法 (SETXYZTDV()) 允许获得三维计算机图像系统来通过动画、显示及软件程序的建模而生成立体图像。该方法允许获得必须分配给两个计算机生成的虚拟可视化相机的空间坐标 (x,y,z)，通过使用任何软件程序来获得立体图像，该软件程序模拟

25 第三维并且利用对象的移动生成图像，或通过瞬间由计算机生成的对象观察到的“虚拟相机”移动而生成图像，这样的软件程序如：Autocad、Micrografix Simply 3D、3Dmax Studio、Point、Dark Basic、Maya、Marionette、Blender、Excel、Word、Paint、Power、Corel Draw、Photo paint、Photoshop 等；但所有这些程序都设计为仅显示具有一个

30 固定或移动视点的一个相机。

另外的 3D 建模和动画特性利用坐标变换等式增加到先前的程序

中，即：

$$x = x' \cos \phi - y' \sin \phi$$

$$y = x' \sin \phi + y' \cos \phi$$

针对直接链接到第一相机的第二或副相机，计算精确的位置，用  
 5 这种方法，从模拟人的立体视觉视点的不同视点获得同时发生的图像。  
 该过程利用算法实时地计算副相机的位置，以将相机放置在恰当的位置，  
 并且获得利用坐标变换等式所得到的第二相机的建模图像和表现，  
 将相机带回原点，计算副相机与对象或目标之间的角度和距离，然后，  
 将主相机、目标和副相机重新放置在所获得的位置处。然后，需要知  
 10 道七个参数，即：主相机在原始坐标系统中的第一坐标 ( $X_p, Y_p, Z_p$ )，  
 第四参数是与眼睛的平均间距 (6.5 到 7.0cm) 相等的距离，以及被相机  
 观察时的目标的位置的三个坐标。

输出参数将是观察相同目标点的副相机的坐标，即 ( $X_s, Y_s, Z_s$ )，  
 这些坐标根据下面这些步骤来获得：

- 15       - 获知主相机在原始坐标系统中的坐标 ( $X_p, Y_p, Z_p$ )，  
           获知目标的坐标 ( $x_t, y_t, z_t$ )  
           - 仅变换“x”和“z”坐标，相机的坐标和/或高度保持恒定（对  
 于观察者不存在视觉偏差）  
           将主相机的坐标带到 (0,  $y_s$ , 0) 位置。  
 20       并且平移目标  
           计算连接相机和目标的线的倾斜度  
           创建主相机和目标相结合的轴与矢量之间的角度。  
           对于角度计算中特殊考虑因素的应用，所属的象限由反正切函数  
 分类。  
 25       获得新坐标，以轴与矢量之间相同的角度从其轴旋转全部坐标系  
 统，获得新坐标系，把对象放置在“z”轴，主相机将保持在新坐标  
 系统的原点。  
           通过把它放置在人眼平均距离的位置上，获得副相机的坐标，  
           以相同的初始角旋转这些坐标，  
 30       添加“x”和“z”偏移量，该偏移量最初被减去，以把主相机放  
 置在原点，

最后，把这两个新的  $x_S y z_S$  坐标指定给副相机并保持  $y_P$  坐标，从而确定对于指定给副相机的相同值的最终坐标点  $(x_S y_P z_S)$  的高度。

可以如 Delphi、C、C++、Visual C++、Omnis 等语言来实现该过程，但结果是相同的。

5 该算法的一般化应用将使用于需要获得副相机的实时位置的任何程序中。

该算法必须以现有任何软件实现，这些软件处理二维但已开发用于立体视觉应用。

10 已例示和描述本发明的具体实施例，对于本领域技术人员来说，明显的是，在不超出本发明的范围的情况下可作出多种修改或变化。所附的权利要求旨在覆盖上述信息，以使所有的变化和变型都处于本发明的范围内。

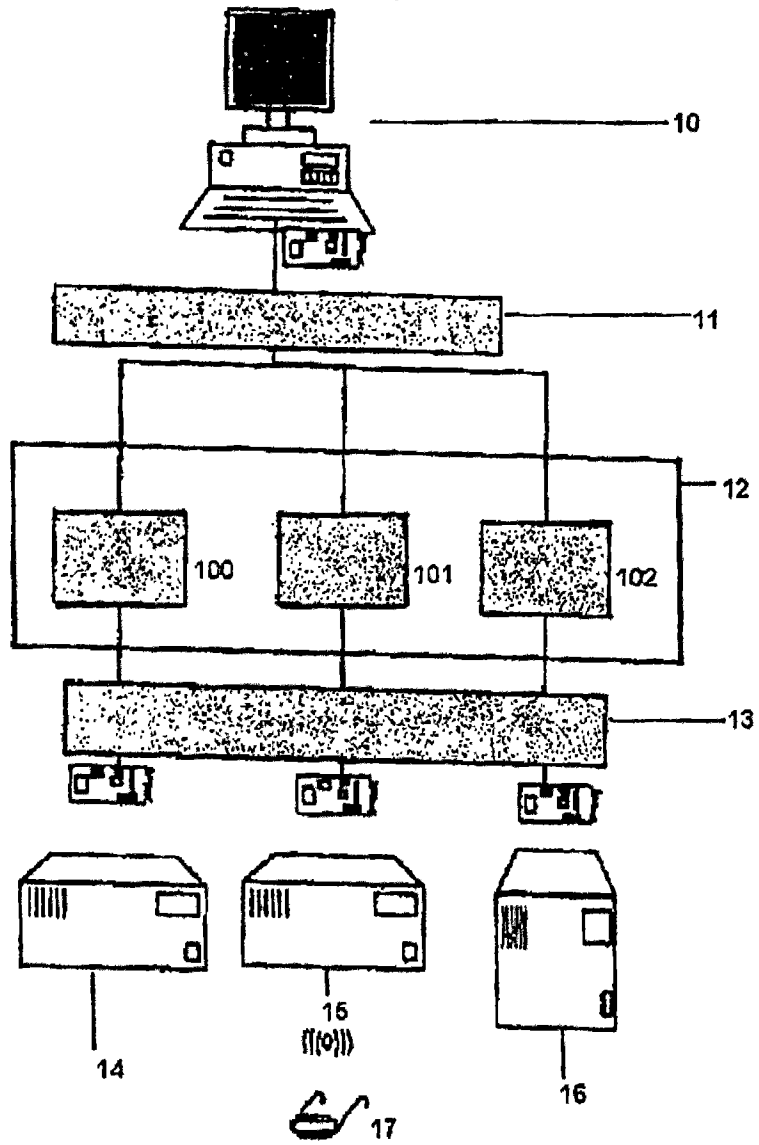


图1

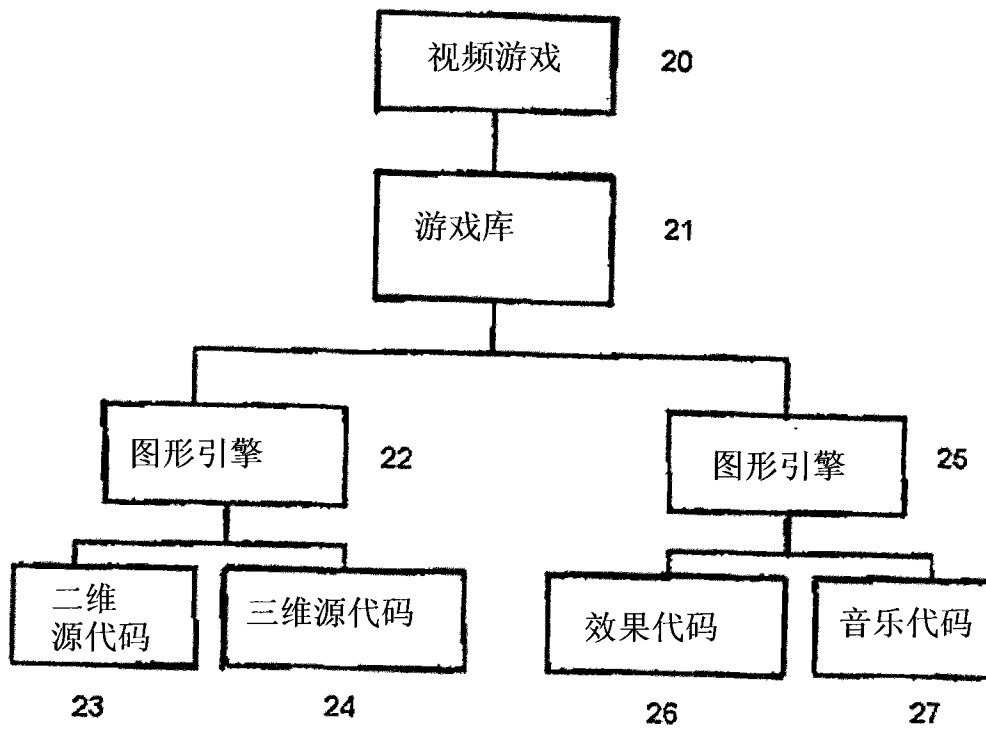


图2



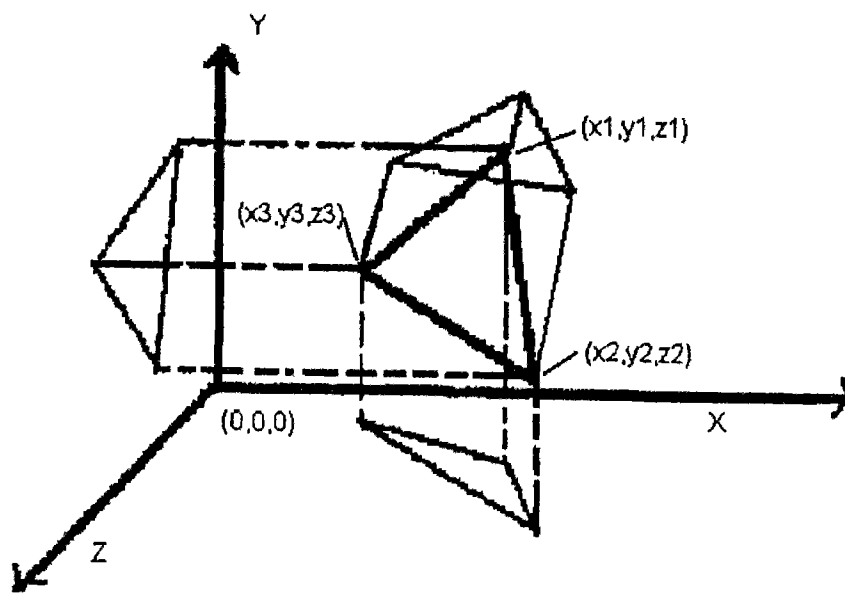


图3

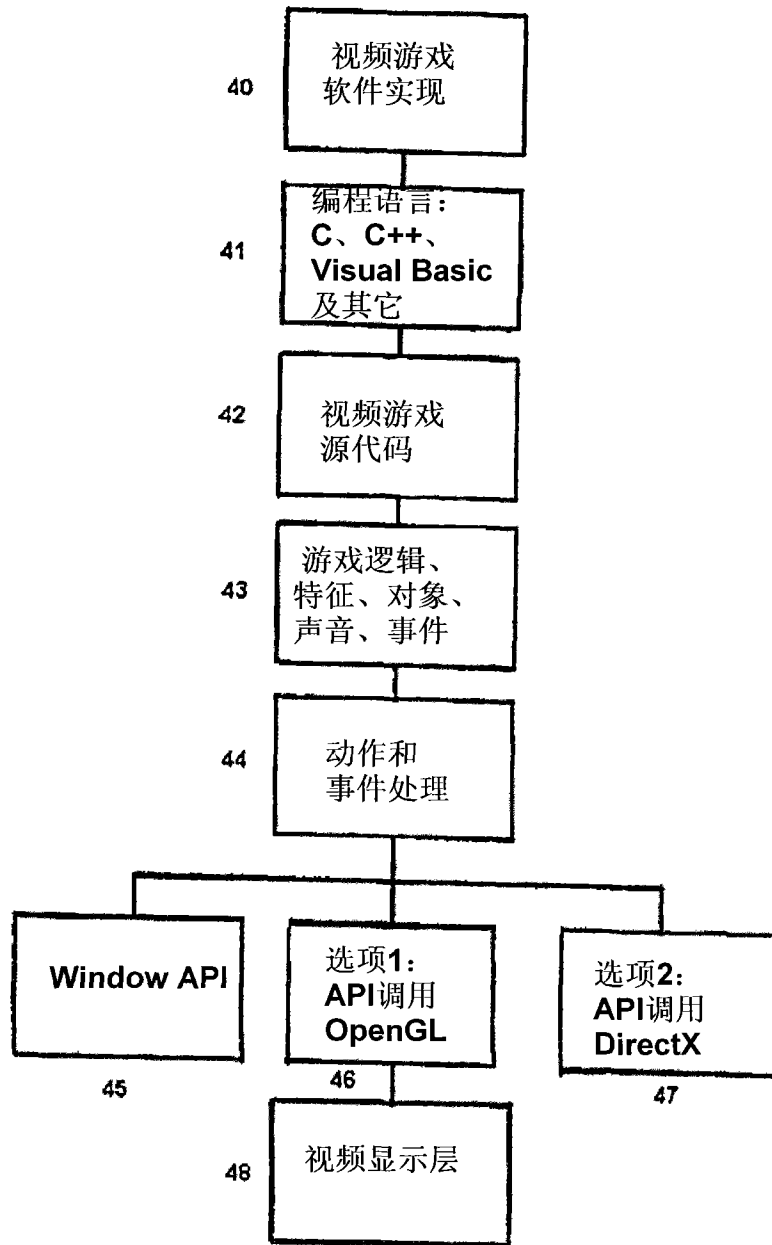


图4

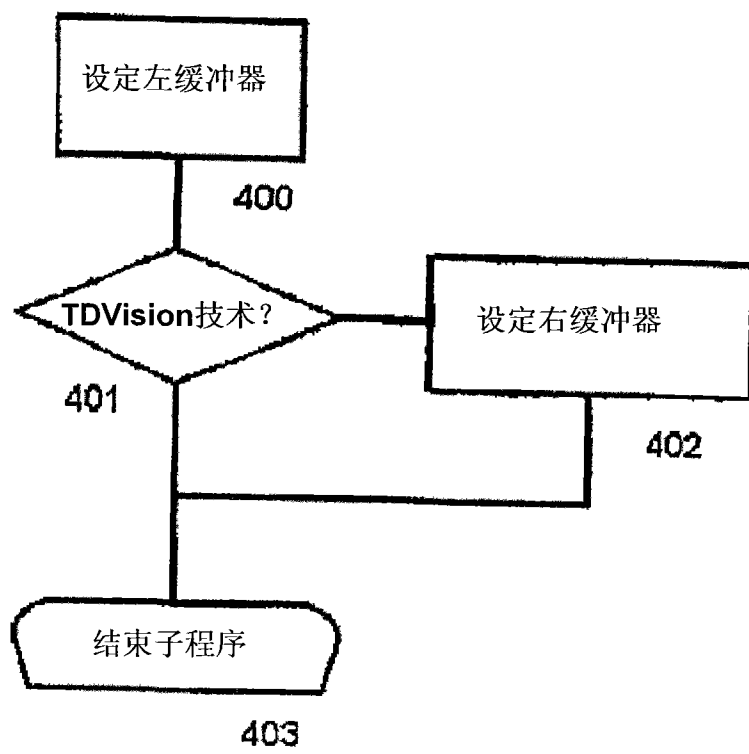


图4a

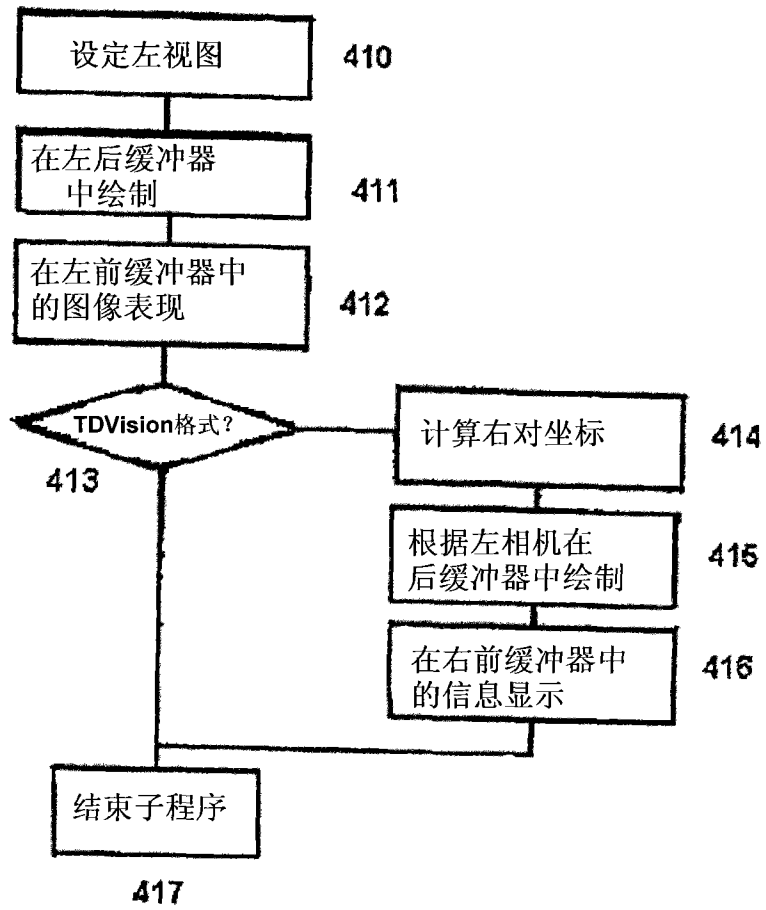


图4b

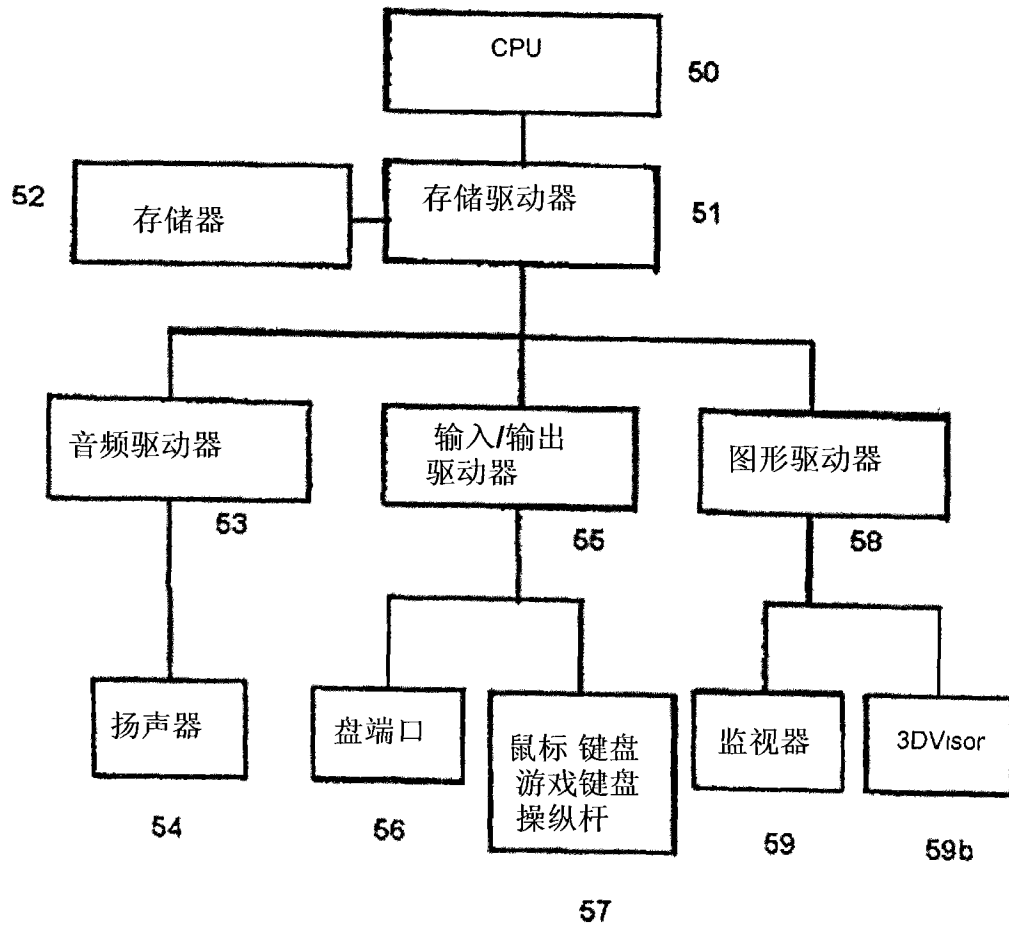


图5a

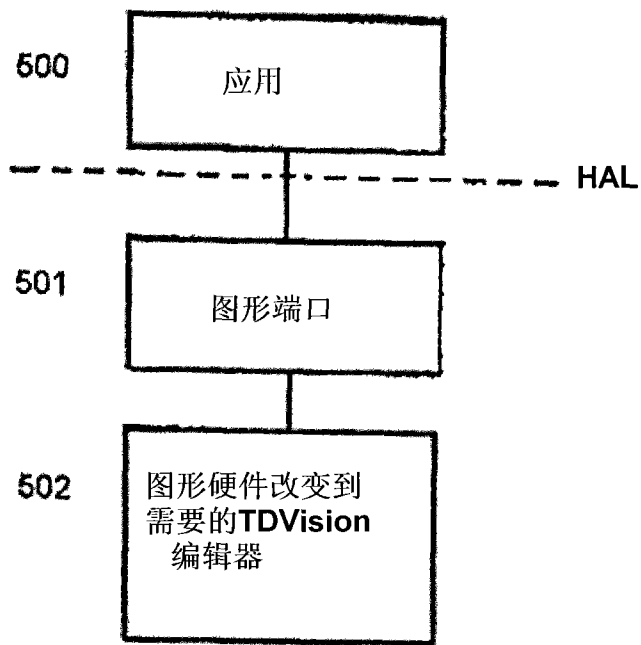


图5b

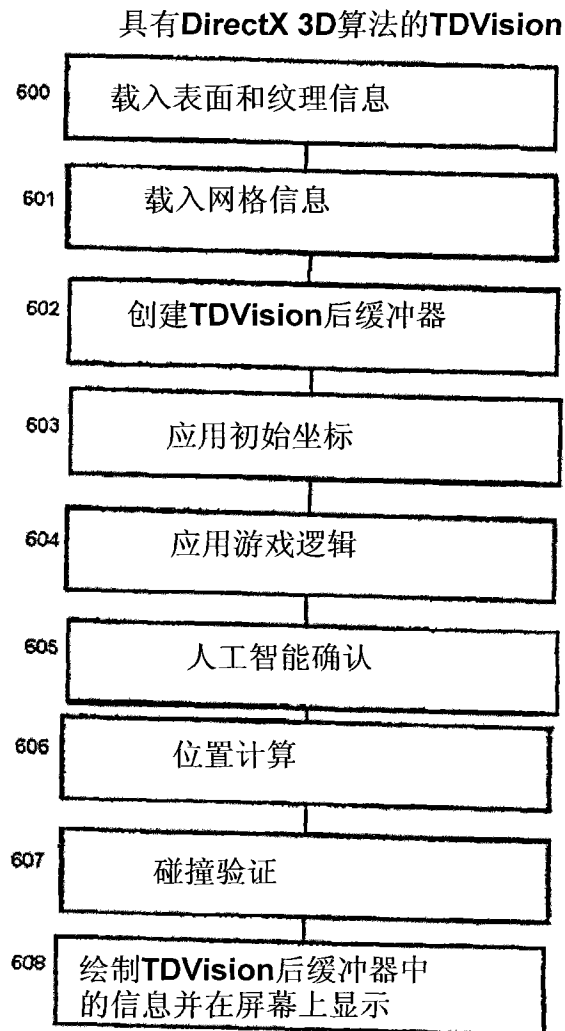


图6

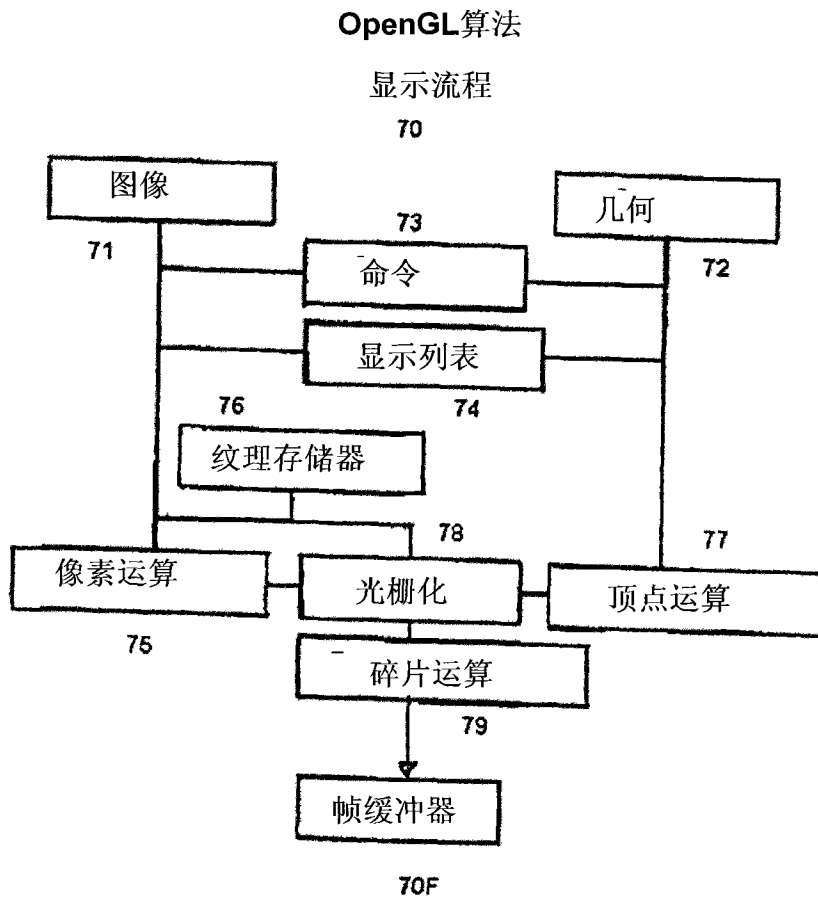


图7a



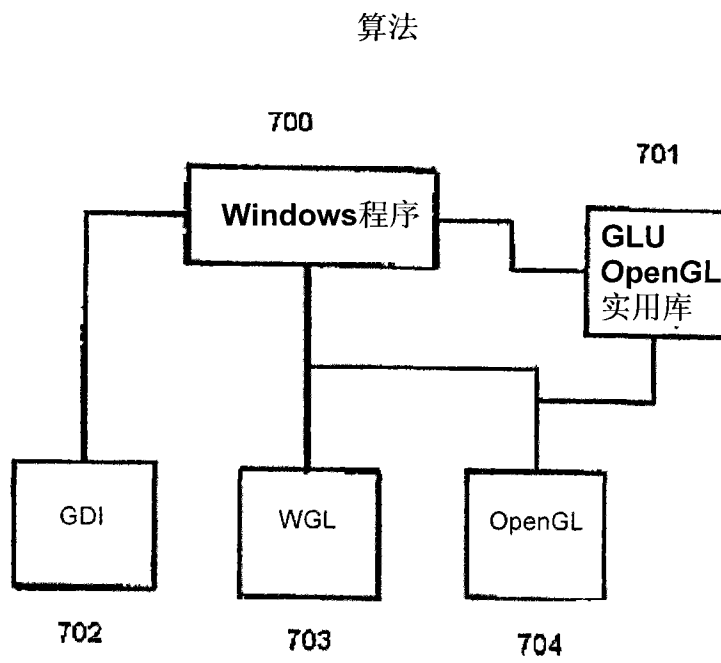


图7b

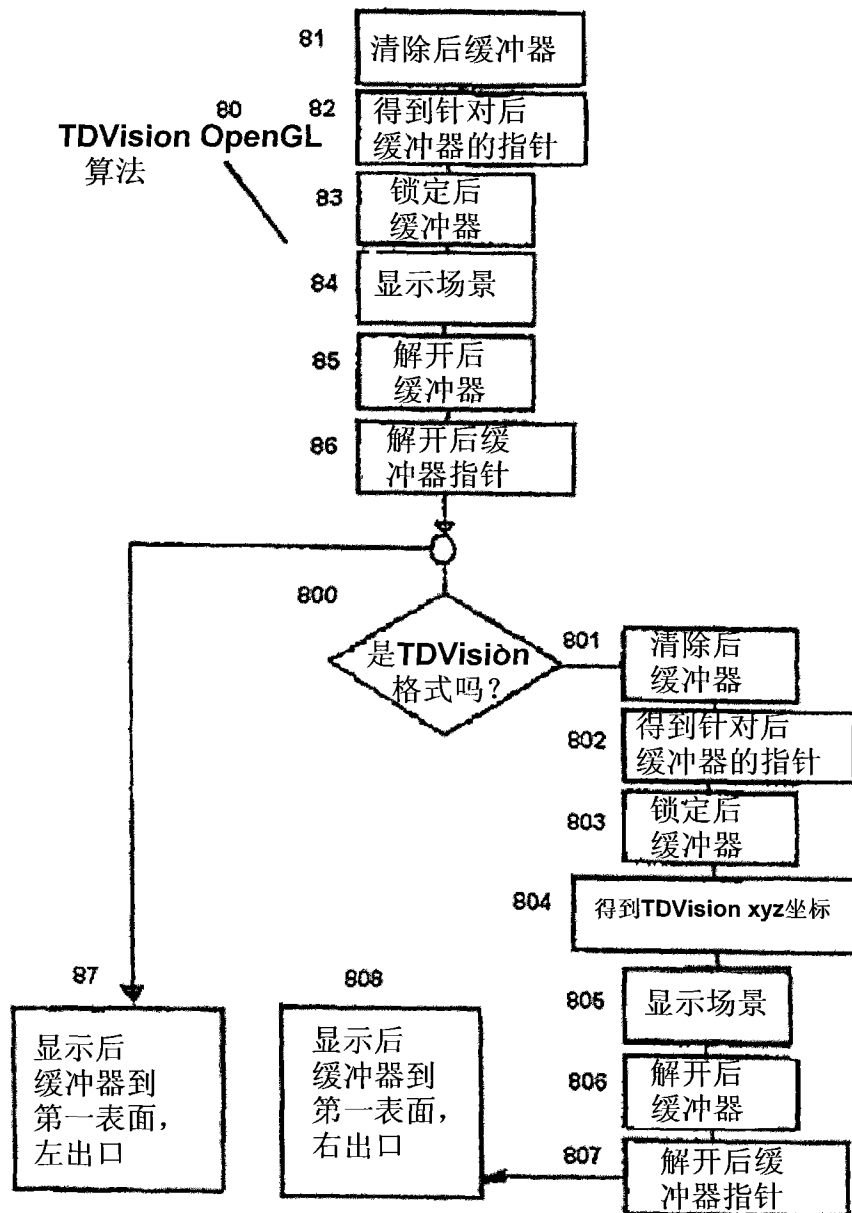


图8

针对TDVision视频卡编辑的改变 90

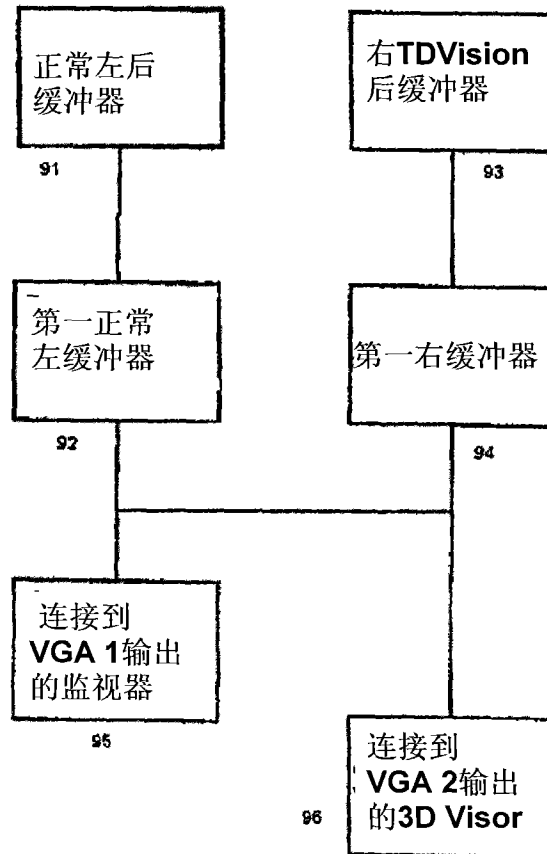


图9