[19] 中华人民共和国国家知识产权局

[51] Int. Cl.
G06T 7/60 (2006.01)
G06T 15/20 (2006.01)

[12] 发明专利说明书

专利号 ZL 200310101803.8

[45] 授权公告日 2007 年 9 月 26 日

[11] 授权公告号 CN 100339874C

[22] 申请日 2003.10.17
[21] 申请号 200310101803.8
[30] 优先权
[32] 2002.10.19 [33] US [31] 60/419，881
[73] 专利权人 威盛电子股份有限公司
地址 台湾省台北县新店市中正路 535 号 8 楼
[72] 发明人 柏瑞斯·柏克潘克 提莫·佩塔西 德瑞克·格兰丁
[56] 参考文献
US5222204A 1993.6.22
US6414683B1 2002.7.2
US6664958B1 2003.12.16
EP0319165B1 1995.10.18
US5361386A 1994.11.1

审查员 赵向阳
[74] 专利代理机构 北京申翔知识产权代理有限公司
代理人 周春发

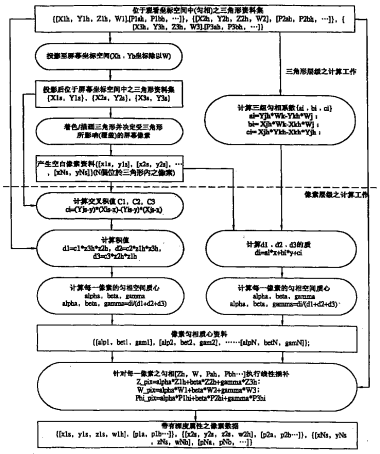
权利要求书 5 页 说明书 15 页 附图 12 页

[54] 发明名称

于匀相空间中进行三角形插补工作的方法及其可程序装置

[57] 摘要

一种用以获取一位于匀相空间中像素的属性资料的方法及其装置。本方法首先获取一三角形的顶点，随后每一顶点的世界空间坐标与属性资料被转换成为位于观看空间中的观看空间坐标与属性资料。接着，对于每一顶点，根据上述的观看空间坐标以计算一组匀相系数，并投影每一顶点的观看空间坐标至屏幕空间坐标。之后，根据上述屏幕空间坐标以决定位于屏幕空间中受上述的三角形所影响的像素。对于每一上述受三角形所影响的像素，则根据上述匀相系数以计算得到一组位于匀相空间中的质心系数，并根据该匀相质心系数以及上述的三角形位于观看空间中的属性数据以得到该像素位于匀相空间中的属性数据。



1、一种用以获取一位于匀相空间中三角形的一属性资料的方法，该方法包含：

获取该三角形的顶点，其中每一顶点是以一组位于世界坐标空间中的坐标表示且其具有一属性资料；

对于每一顶点，转换该顶点位于世界空间中的该组坐标与该属性资料至一匀相空间坐标及一组匀相空间中的属性资料，根据该匀相空间坐标以计算得一组顶点的匀相系数，并投影该匀相空间坐标至一屏幕空间坐标；

根据该屏幕空间坐标决定在屏幕空间中受该三角形所影响的像素；以及

对于每一受该三角形所影响的像素，根据该匀相系数以计算得一组位于匀相空间中的质心系数，并根据该组匀相质心系数与该顶点位于匀相空间中的该属性资料执行一线性插补动作以得到受该三角形所影响的该像素位于匀相空间中的属性数据。

2、如权利要求 1 所述的用以获取一位于匀相空间中三角形的一属性资料的方法，其中对于每一顶点而言，该顶点位于匀相空间中的该坐标与该属性资料是以$[Xih, Yih, Zih, Wi]$与$[piah]$所表示，其中，i 是一与该顶点相配合的标号且 i=1,2,3，且 Wi 为一透视修正参数。

3、如权利要求 2 所述的用以获取一位于匀相空间中三角形的一属性资料的方法，其中上述顶点的该匀相系数为$\tilde{a}_i, \tilde{b}_i, \tilde{c}_i$，且其计算方式是根据该匀相空间坐标以及以下计算式：

$$\tilde{a}_i = Yjh \cdot Wk - Ykh \cdot Wj;$$

$$\tilde{b}_i = Xjh \cdot Wk - Xkh \cdot Wj; \text{与}$$

$$\tilde{c}_i = Xjh \cdot Ykh - Xkh \cdot Yjh;$$

其中， $j = i \bmod 3+1$ 且 $k = j \bmod 3+1$，且 Wj 与 Wk 为透视修正参数。

4、如权利要求 1 所述的用以获取一位于匀相空间中三角形的一属性资料的方法，其中上述根据该屏幕空间坐标决定在屏幕空间中受该三角形所影响的像素包含着色/描画该三角形。

5、如权利要求 1 所述的用以获取一位于匀相空间中三角形的一属性资料的方法，其中上述根据该屏幕空间坐标决定在屏幕空间中受该三角形所影响的像素包含提供与受该三角形所影响的该像素相关的空白像素资料。

6、如权利要求 5 所述的用以获取一位于匀相空间中三角形的一属性资料的方法，其中对于 N 个受该三角形所影响的该像素，该空白像素资料包含 N 个屏幕空间坐标$\{[X_1s,Y_1s],[X_2s,Y_2s],...[X_Ns,Y_Ns]\}$。

7、如权利要求 4 所述的用以获取一位于匀相空间中三角形的一属性资料的方法，其中上述的着色/描画步骤提供与受该三角形所影响的该像素相关的空白像素资料。

8、如权利要求 3 所述的用以获取一位于匀相空间中三角形的一属性资料的方法，包含计算代表该匀相空间的质心系数 α、β 与 γ，其中 α、β 与 γ 的计算是利用该匀相系数$\tilde{a}_i,\tilde{b}_i,\tilde{c}_i$与计算式$d_i=(\tilde{a}_i\cdot X+\tilde{b}_i\cdot Y+\tilde{c}_i)$计算出 d1、d2 与 d3 后，再以 d1、d2 与 d3 与下列计算式：

$\alpha=\frac{d_1}{d_1+d_2+d_3}$； $\beta=\frac{d_2}{d_1+d_2+d_3}$； 与 $\gamma=\frac{d_3}{d_1+d_2+d_3}$ 所计算得出，

其中

$$\begin{cases} X=\alpha\bullet x_1+\beta\bullet x_2+\gamma\bullet x_3 \\ Y=\alpha\bullet y_1+\beta\bullet y_2+\gamma\bullet y_3 \\ \alpha+\beta+\gamma=1 \\ 0\le\alpha\le 1 \\ 0\le\beta\le 1 \\ 0\le\gamma\le 1 \end{cases}$$

，其中该三角形的三个顶点的坐标分别为(x1, y1)、(x2, y2)、(x3, y3)。

9、如权利要求 8 所述的用以获取一位于匀相空间中三角形的一属性资料的方法，其中上述的线性插补动作包含对于 Z_pix、W_pix 与 Phi_pix 的计算工作，其中，Z_pix 是为一位于匀相空间中的 Z 方向坐标；W_pix 是为一透视修正参数；Phi_pix 是为该像素位于匀相空间中的属性资料，该计算工作所使用的计算式如下：

$$Z\_pix = \alpha \cdot Z1h + \beta \cdot Z2h + \gamma \cdot Z3h;$$

$$W\_pix = \alpha \cdot W1 + \beta \cdot W2 + \gamma \cdot W3; \text{ 与}$$

$$Phi\_pix = \alpha \cdot P1hi + \beta \cdot P2hi + \gamma \cdot P3hi \quad 。$$

10、如权利要求 1 所述的用以获取一位于匀相空间中三角形的一属性资料的方法，其中上述的根据该屏幕空间坐标决定在屏幕空间中受该三角形所影响的像素的位置与深度是以位于匀相空间中的该坐标表示。

11.一种用以获取位于匀相空间中的一属性资料的一通用可程序硬件单元，该通用可程序硬件单元包含：

一像素内存单元，该像素内存单元储存三角形在匀相空间中的坐标资料及属性资料；

一三角形内存单元，该三角形内存单元储存空白像素坐标资料；

一可程序单一指令多重资料纯量单元，该可程序单一指令多重资料纯量单元包含一微码内存，该微码内存储存施行一单一指令多重资料处理算法的复数个指令；以及

双算术逻辑单元，该双算术逻辑单元是依据该复数个指令对该三角形在匀相空间中的坐标资料或该空白像素坐标资料进行三角形插补算法。

12. 如权利要求 11 所述的用以获取位于匀相空间中的一属性资料的一通用可程序硬件单元，其中上述的指令是命令该可程序单一指令多重资料纯量单元执行如下的步骤操作包含：

对于每一顶点转换的指令，根据该匀相空间坐标计算得三角形的顶点的匀相系数，进一步投影该匀相空间坐标至一屏幕空间坐标；

决定受该三角形所影响的像素的指令，是根据该屏幕空间坐标决定在屏幕空间中受该三角形所影响的像素；以及

对于每一受该三角形所影响的该像素进行处理的指令，是根据该匀相系数以计算得一组位于匀相空间中的质心系数，以根据该组匀相质心系数与该顶点位于匀相空间中的该属性资料执行一线性插补动作以得到受该三角形所影响的该像素位于匀相空间中的属性执行。

13. 如权利要求 12 所述的用以获取位于匀相空间中的一属性资料的一通用可程序硬件单元，其中对于每一顶点而言，该顶点位于匀相空间中的该坐标与该属性资料是以$\left[\tilde{X}ih,\tilde{Y}ih,\tilde{Z}ih,W\right]$与$[piah]$所表示，其中，i 是一与该顶点相配合的标号且 i=1,2,3，且 Wi 为一透视修正参数。

14. 如权利要求 13 所述的用以获取位于匀相空间中的一属性资料的一通用可程序硬件单元，其中上述的顶点的该匀相系数为$\tilde{a}_i,\tilde{b}_i,\tilde{c}_i$，且该可程序单一指令多重资料纯量单元计算$\tilde{a}_i,\tilde{b}_i,\tilde{c}_i$，其计算方式是根据该匀相空间坐标以及以下计算式：

$a_i = Yjh\cdot Wk - Ykh\cdot Wj, b_i = Xjh\cdot Wk - Xkh\cdot Wj$,以及$c_i = Xjh\cdot Ykh - Xkh\cdot Yjh$；

其中，j = i mod 3+1 且 k = j mod 3+1，且 Wj 与 Wk 为透视修正参数。

15. 如权利要求 12 所述的用以获取位于匀相空间中的一属性资料的一通用可程序硬件单元，其中上述的决定受该三角形所影响的像素的指令包含着色/描画的指令，以着色/描画该三角形。

16. 如权利要求 12 所述的用以获取位于匀相空间中的一属性资料的一通用可程序硬件单元，其中上述的决定受该三角形所影响的像素的指令包含提供受该三角形所影响的该像素相关的空白像素资

料的指令。

17. 如权利要求 16 所述的用以获取位于匀相空间中的一属性资料的一通用可程序硬件单元，其中上述的受该三角形所影响的该像素相关的空白像素资料包含 N 个屏幕空间坐标 $\{[X_1s,Y_1s],[X_2s,Y_2s],...[X_Ns,Y_Ns]\}$。

18. 如权利要求 14 所述的用以获取位于匀相空间中的一属性资料的一通用可程序硬件单元，包含计算代表该匀相空间的质心系数 α、β 与 γ，其中 α、β 与 γ 的计算是利用该匀相系数 $\tilde{a}_i,\tilde{b}_i,\tilde{c}_i$ 与计算式 $d_i=(\tilde{a}_i\cdot X+\tilde{b}_i\cdot Y+\tilde{c}_i)$ 计算出 d1、d2 与 d3 后，再以 d1、d2 与 d3 与下列计算式：

$\alpha=\frac{d_1}{d_1+d_2+d_3}$；$\beta=\frac{d_2}{d_1+d_2+d_3}$；与 $\gamma=\frac{d_3}{d_1+d_2+d_3}$ 所计算得出，其中

$$\begin{cases} X=\alpha\bullet x_1+\beta\bullet x_2+\gamma\bullet x_3 \\ Y=\alpha\bullet y_1+\beta\bullet y_2+\gamma\bullet y_3 \\ \alpha+\beta+\gamma=1 \\ 0\le\alpha\le 1 \\ 0\le\beta\le 1 \\ 0\le\gamma\le 1 \end{cases}$$

，其中该三角形的三个顶点的坐标分别为(x1, y1)、(x2, y2)、(x3, y3)。

19. 如权利要求 18 所述的用以获取位于匀相空间中的一属性资料的一通用可程序硬件单元，其中上述的线性插补动作包含执行对于 Z_pix、W_pix 与 Phi_pix 的计算工作，其中，Z_pix 为一位于匀相空间中的 Z 方向坐标；W_pix 为一透视修正参数；Phi_pix 为该像素位于匀相空间中的属性资料，该计算工作所使用的计算式如下：

$Z\_pix=\alpha\cdot Z1h+\beta\cdot Z2h+\gamma\cdot Z3h;$

$W\_pix=\alpha\cdot W1+\beta\cdot W2+\gamma\cdot W3;$ 与

$Phi\_pix=\alpha\cdot P1hi+\beta\cdot P2hi+\gamma\cdot P3hi$。

于匀相空间中进行三角形插补工作的方法及其可程序装置

技术领域

本发明是有关于一种图形处理技术，特别是有关于一种于匀相空间中进行三角形插补工作的技术。

背景技术

多边形(三角形)的插补对于硬件绘图装置而言是一种所需计算强度相当高的工作，因其是针对屏幕上每一对象(object)中的每一像素(pixel)所进行。每一三角形皆会有多个需考虑的属性(attribute)如深度，颜色，纹理(textures)等；最终产生影像的品质主要则是取决上述的插补工作的准确性。此外，整体图形框架产生的速度也是取决于此插补工作的速度。现代绘图芯片中，用以施行此种插补工作的硬件是一关键零件，且通常也是决定芯片开发预算的关键因素。

目前已有许多三角形插补技术的解决方案被施行在不同结构(architecture)种类的芯片中。大部分现行的技术是属于下列三种类型之一：a. 在屏幕的卡氏坐标(Cartesian coordinates)下所进行，并辅以透视修正(perspective correction)的固定点插补程序；b. 在屏幕质心(barycentric)坐标下所进行，并辅以透视修正的固定点插补程序；以及c. 在匀相坐标下所进行的固定点插补程序。

上述前两种类型的技术必须将所有参数(parameter)投影至屏幕空间(即除以W)中以便于针对每一像素进行上述的插补与透视修正(即除以1/W)的工作。请参阅图1与图2所示，其即是为此两种类型技术的示意图。

上述第三种类型的技术可省去上述额外的将所有参数投影至屏幕空间的步骤以及相伴而来的修正工作，且其所计算的是相同的匀相

质心(homogeneous barycentric)。请参阅图3所示，其即是为此种类型技术的示意图。上述三种类型的技术皆是针对每一像素进行计算，因此若欲处理中大型三角形则所需计算量将大幅增加；此外，上述三种类型的技术的施行皆需要可观的专属硬件配合，亦即其所使用的硬件资源无法分享供其它计算工作所使用。

具体来说，上述第一种类型的技术的缺点包含：a. 为进行上述的将所有参数投影至屏幕空间(即除以W)以使其线性化的步骤则需要额外的计算工作；b. 投影步骤、插补步骤以及修正步骤以回复(recover)真实值(true value)； c. 1/W的插补工作需要额外的参数delta建立计算，故产生精确度的问题；d.需通过由插补1/W_pix值的除法运算回复每一像素中额外的真实参数值；以及e. 需可观的专属硬件配合。

上述第二种类型的技术的缺点包含：a. 同样具有上述的将所有参数投影至屏幕空间(即除以W)以使其线性化的步骤及其所需的额外计算工作；b. 同样具有上述的用以回复真实值的投影步骤、插补步骤以及修正步骤；c. 同样具有上述的通过由插补1/W_pix值的除法运算以回复每一像素中额外的真实参数值的步骤；以及d. 需可观的专属硬件配合，且所使用的硬件无法分享以供其它计算工作使用。

上述第三种类型的技术的缺点包含：a. 像素质心(barycentric)的计算工作必须先于像素层级中完成，且遇到具有多重像素的三角形的状况则计算工作量将呈倍数增加；以及b. 可观的专属硬件。

因此，可大幅减少计算工作且不需大量专属硬件配合的适以在硬件绘图装置中施行多边形插补工作的方法和装置是目前产业所迫切需要的。

发明内容

为解决上述产业对于具有更佳效率的多边形插补工作的需求，本发明提供了一种兼具快速、精确与高效率特性的解决方案，其是于硬件绘图装置中施行多边形插补程序以找出一三角形中每一像素的个别参数的正确值(Z值、颜色、多重纹理坐标；multiple texture

coordinates)。在算法的施行上，本发明使用了一可程序单一指令多重资料纯量单元(single instruction multiple data scalar unit; SIMD scalar unit)；此可程序单一指令多重资料纯量单元也可符合 Microsoft Dx9.10与OpenGL API 对于可程序绘图机器的硬件需求而用于更专业的像素处理。本发明所提供的三角形插补算法可于上述的可程序单一指令多重资料纯量单元中执行，且其执行效率可逼近上述的专属硬件所能达到的。事实上，所有的插补操作原本就是在尽可能精确的前提下以浮点运算进行的。

为了达成上述目的，本发明提供了一种新的方法以获取匀相空间中的属性资料。此方法的步骤包含：取得一个三角形的顶点，其中，每一个顶点由一组在世界坐标空间(world coordinate space)中的坐标代表，且每一顶点皆具有其属性(attribute)；随后，转换每一个顶点在世界空间中的坐标与属性成为在观看空间(viewer space)中相对应的坐标与属性；之后，根据上述顶点的观看空间坐标计算其相对应的匀相空间系数(homogeneous coefficients)，并将上述顶点的观看空间坐标投影至屏幕空间(screen space)。本发明所提供的方法接着根据上述的三角形在上述屏幕空间中的坐标判定会受到此三角形所影响的像素；最后则是根据上述的匀相空间系数针对上述的受三角形所影响的像素计算出一组位于匀相空间中的质心(barycentric)系数，并且根据该组匀相质心系数与上述的位于观看空间中顶点的属性执行一线性插补程序以得到该受影响像素位于匀相空间中的属性。

本发明的一个优点为与先前技艺相比较，本发明使用较少的运算操作，因此与上述的在屏幕空间中依序执行三角形插补与透视修正的方式相较，本发明具有较佳的效能。

本发明的另一优点为可省去上述的为便于在屏幕空间中执行插补工作并进一步在观看空间中执行透视修正计算以回复真实值而所执行的参数投影步骤。

附图说明

上述本发明的发明特征、优点与实施方式将通过由详细叙述、权利要求与所附图标加以说明，其中：

图1所示为先前技艺中在卡氏坐标下所执行的三角形着色/描划(rasterization)与插补方法；

图2所示为先前技艺中在屏幕质心坐标下所执行的三角形着色/描划(rasterization)与插补方法；

图3所示为另一先前技艺中在匀相质心坐标下所执行的三角形插补方法；

图4A所示为一根据本发明所建构的插补方法的流程图；

图4B所示为一位于a.世界坐标空间 b.匀相坐标空间 c. 屏幕坐标空间中的三角形与受其所影响的像素；

图4C与图4D所示为根据本发明所建构的三角形建立方程式与像素处理方程式；

图5A所示为图3与图4所示方法的步骤的比较；

图5B所示为图3与图4所示运算的比较列表；

图6A所示为一利用长数据处理模式(long data processing mode)执行三角形质心插补(triangle barycentric interpolation)的可程序单元；

图6B所示为上述的可程序单元所使用的一指令集(instruction set)；以及

图7所示为一执行本发明的短数据处理模式(short data processing mode)的算术逻辑单元。

【主要代表符号】

10 可程序硬件单元;

12 单一指令多重资料纯量单位;

14 微码内存;

16, 18 算术逻辑单元;

20 倒数单元;

22 三角形着色/描画器;

24 像素内存单元;

26 顶点几何处理单元;

28 三角形内存单元;

30, 32 旁通缓存器;

120 三角形于世界坐标空间中;

120' 三角形于匀相坐标空间中;

120" 三角形于屏幕坐标空间中;

P1, P2, P3 世界坐标空间三角形的顶点;

P1h, P2h, P3h 匀相坐标空间三角形的顶点;

P1s, P2s, P3s 屏幕坐标空间三角形的顶点;

具体实施方式

在一根据本发明所建构的绘图系统中，像素(primitives)首先被分解成个别的小三角形，随后进一步处理这些小三角形以便于使原图像有较佳的呈现效果。为使上述的三角形处理过程易于了解，参阅图4A所示，其为一根据本发明所建构的三角形插补方法的步骤流程图。具体来说，此方法是用于三角形着色/描划(rasterizing)工作以及利用匀相质心坐标(barycentric coordinate)所进行的线性参数插补(linear parameter interpolation)工作。本方法使用一在两种不同层级中进行的系数计算(coefficient calculation)，其中，上述的"两种层级"分别是指三角形层级与像素层级；在此两种层级中执行的某些计算操作可同时进行，其详细执行步骤将于后叙述。在以下的叙述中，计算操作步骤将表示于椭圆形方块中，而所得结果资料集(data set)将表示于长方形方块中。

为将图4A中所示的方法作更为清楚的解说，图4B中所示的包含：a. 一位于世界坐标空间(world coordinates space)中的三角形120；b.一位于匀相坐标空间(homogeneous coordinates space)中的三角形120' ；以及 c. 一位于屏幕坐标空间(screen coordinates space)中，且带有受其所影响的像素的三角形120"。在世界坐标空间中的三角形120是以三个顶点P1、P2以及P3所定义，其中，此三顶点P1、P2以及P3的坐

标分别为[X1,Y1,Z1]、[X2,Y2,Z2]以及[X3,Y3,Z3]且此三顶点P1、P2以及P3的属性数据则是分别为p1a、p1b…；p2a、p2b…；p3a、p3b…。

因此，在上述的三角形层级中，三角形的资料集(data set)是首先在世界坐标空间中通过由三角形的三顶点坐标以及其所对应的纹理坐标(texture coordinate)加以特征化，其结果则是如下所示：

$$\{\{X1,Y1,Z1,1][p1a,p1b,...]\},\{\{X2,Y2,Z2,1][p2a,p2b,...]\},\{\{X3,Y3,Z3,1][p3a,p3b,...]\}$$

上述世界坐标空间(三维空间)的作用为在执行一几何转换操作(geometric transformation)前定义一对象(object)。上述的三角形资料集于世界空间中建立之后，接着则被转换至一观看坐标空间 (viewer coordinate space) 中并被修剪(clipped)以成为一观看体积(view volume)。基本上，最先在世界坐标空间中被指定的三角形会被当作一观看点(viewpoint)处理。在观看空间中，此观看点所在之处即是为坐标原点，且观看线(view ray)是顺着Z轴而延伸出去。在观看空间中经上述的计算处理所得的三角形资料集，亦即图4B的b部分中所示的位于观看空间(匀相空间)中的三角形120'，表示如下：

$$\begin{aligned}&\{\{X1h,Y1h,Z1h,W1][p1ah,p1bh,...]\},\{\{X2h,Y2h,Z2h,W2][p2ah,p2bh,...]\},\\&\{\{X3h,Y3h,Z3h,W3][p3ah,p3bh,...]\},\end{aligned}$$

其中，[X1h, Y1h, Z1h]、[X2h, Y2h, Z2h]与[X3h, Y3h, Z3h]是为在匀相空间中个别顶点的坐标；W1、W2与 W3为个别顶点的透视修正参数(perspective correction parameter)；p1ah, p1bh, …、p2ah, p2bh, …与 p3ah, p3bh, …则是为在匀相空间中个别顶点的属性数据。

在转换至观看坐标空间的步骤进行之后，随之进行的为一三角形建立程序(triangle setup phase)，此程序自上述的匀相空间中的三角形资料集导出三组匀相系数(a1, b1, c1)、(a2, b2,c2)与(a3,b3,c3)，例如，a1=Y2h*W3 - Y3h*W2。 推广来说则此三组匀相系数是由下列方式得出：

$$a_i = Yjh \cdot Wk - Ykh \cdot Wj;$$

$b_i = Xjh \cdot Wk - Xkh \cdot Wj$; 以及

$c_i = Xjh \cdot Ykh - Xkh \cdot Yjh$,

其中 i, j & k = 1, 2 & 3.

同时，较佳的方式是将此三角形观看空间资料集投影至屏幕坐标空间中以产生{[X1s, Y1s], [X2s, Y2s], [X3s, Y3s]}。

屏幕坐标空间需要更进一步的转换以将对象在屏幕坐标系统中的几何形式经由投影与转换动作在最后显示对象时赋予对象深度。参阅图4A所示，上述的三角形资料集接着被着色/描画(rasterized)以产生空白像素资料(blank pixel data)。上述的着色/描画动作进行时是将像素如三角形等切割成为个别的像素，并执行一Z方向缓冲动作(Z-buffering)(以便于获取深度的有关资料，也就是Z方向的资料)与其它像素层级的功能操作。因此，对于被一三角形(也即图4B的c部分中所示的120")所覆盖的N个像素，上述的着色/描画动作所产生的空白像素资料将由在屏幕空间中的N个坐标所描述，且其表示方法如下：

{[X1s, Y1s], [X2s, Y2s],… [XNs, YNs]}。

如图4B中所示，随后进行的为一像素建立程序(pixel setup phase)，也即，在匀相空间中，对于每一个被三角形所覆盖的像素P，会有一组匀相空间系数dl、 d2与d3以及一组质心系数(barycentric coefficient) α、β与γ被计算出来，并由此可得到该受三角形所影响的像素的匀相质心资料(homogeneous barycentric data)。上述的像素建立程序即是利用来自上述三角形建立程序的资料与自上述的着色/描画步骤所得到的空白像素资料所进行。

最后，一插补程序(interpolation phase)藉由上述计算的质心坐标以针对各个像素的匀相坐标以及属性资料进行线性插补动作。此程序包含对于在匀相空间中的Z方向坐标Zh、上述的透视修正参数W，以及上述的在匀相空间中的属性资料Pah, Pbh,…的线性插补动作。对于每一像素，经此程序所得的Z方向坐标Z_pix为：

$Z\_pix = \alpha \cdot Z1h + \beta \cdot Z2h + \gamma \cdot Z3h$;

经此程序所得的透视修正参数W为：

$$W\_pix = \alpha \cdot W1 + \beta \cdot W2 + \gamma \cdot W3;$$ 以及

对每一像素而言，其每一种属性资料在经此程序后所得的属性Phi为：

$$Phi\_pix = \alpha \cdot P1hi + \beta \cdot P2hi + \gamma \cdot P3hi.$$

在经过线性插补动作后，所得的带有深度属性资料的像素资料可以描述如下：

$$\{\{X1s, Y1s, Z1s, W1h][p1a, p1b, ...\}, \{X2s, Y2s, Z2s, W2h][p2a, p2b, ...\}, \{X3s, Y3s, Z3h, W3h][p3a, p3b, ...\}, ...$$

更具体地说，如图4C与图4D所示，其即为上述各步骤中所使用的根据本发明所建构的三角形建立方程式与像素处理方程式(triangle setup and pixel processing formulae)。如图所示，对于三角形中的每一像素，其坐标皆可从三角形的三顶点，利用其质心系数表示出；此种表示方式特别常用在纹理相关应用技术(texture application techniques)中。例如，Silicon Graphics公司所出的的产品OpenGL™中即定义有一质心坐标系统：给定一具有三个顶点P1 (x1, y1, z1)、P2(x2, y2, z2)以及P3 (x3, y3, z3)的三角形，一位于该三角形所在平面上的定点P (x, y, z)的位置可用上述的三个顶点加以表示，其表示方法为： P = αP1+βP2+γP3。其中，α、β与γ质心系数。因此，此定点P的坐标可以如下方式表示：

X = α x1 + β x2 + γ x3, 以及

Y = α y1 + β y2 + γ y3。

所应注意的是，此定点P是以一组特定的质心坐标所表示，且该坐标需满足α + β + γ =1的条件。

因此，以质心系数为基础，每一受三角形所影响的像素的X与Y坐标皆可被推导而得出，也即定点P可为三角形三顶点(P1, P2, P3)的线性组合。由于此质心系数可应用于像素任一属性资料的插补工作，

包含Z坐标(深度)、颜色、UV坐标…等，因此只要计算出质心系数，则上述的属性数据皆可从而计算得到。其中，α、β与γ的计算方式是如图4C的方程式(1)所表示。此外，计算出质心系数之后，此质心系数也可用于整个多边形(三角形)的插补工作；更进一步，由于利用匀相质心坐标可帮助定位每一像素点(P)，因此可藉由指定各点的位置对多边形进行加上阴影或是纹理贴图(texture mapping)的工作。

分别给予三角形的三顶点i=1, 2, 3、j= i mod 3+1与 k= j mod 3+1的编号顺序并且如方程式(2)所示令$a_i=y_i-y_k$、$b_i=x_k-x_j$且$c_i=x_j*y_k - x_k*y_j$；将上述两者代入方程式(1)中则可得如方程式(3)所示的质心系数的相互关系。

如方程式(4)所示，其是为定点P的透视修正参数w。如图所示，w是为上述的质心系数α、β与γ以及三角形三顶点个别的透视修正参数w1、w2与w3的函数。应注意的是，保存透视修正参数w的目的为保存与观看点(viewpoint)之间距离的信息。然而，在进行再计算以获取匀相 (观看点) 空间中的质心坐标之前，顶点坐标(x_i, y_i)须先利用相对应的修正参数w_i以转换为在匀相空间中的坐标($\tilde{x}_i,\tilde{y}_i$)；接着利用修正参数w以导出定点P在匀相空间中的坐标($\tilde{x},\tilde{y}$)。

随后，进行一系列的计算操作以作为计算质心坐标的准备，其包括：(1) 对于每一三角形计算三组匀相空间系数ai、bi与ci，此三组匀相空间系数可由受三角形所影响的每一像素分享使用；(2) 利用个别透视修正参数wi转换上述的匀相空间系数至匀相空间以得到$\tilde{a}_i,\tilde{b}_i,\tilde{c}_i$；以及(3) 针对每一像素计算di 。如图4D的方程式(6)所示，计算出d1、d2与d3之后，对于定点P而言，其在匀相空间中之质心系数$\tilde{\alpha},\tilde{\beta},\tilde{\gamma}$ 可接着被计算出来以完成在匀相空间中的三角形插补动作，如方程式(5)所示。藉由上述的对受三角形影响的像素中每一定点P的插补动作，上述的匀相空间中的质心系数将可进一步用以获取真实值(如此则已避免进行上述除以1/W的动作)。如此，则此三角形的对象已被具有恰当的深度属性资料的像素所仿真。

参阅图5A所示，其为图3与图4A中所示步骤的比较。左侧具阴影的方块中所示的步骤为图3中所示与图4A中不同的；右侧具阴影的方块中所示的步骤为图4A中所示与图3中不同的。不具阴影的方块中所示的步骤则是图3与图4A中皆有出现的。参阅图5B所示，其为图3与图4A中所示步骤于操作数目上的比较表。

本发明所提供的方法与图3中所示的不同处在于，本发明对于匀相空间中质心的计算分别在两种层级中进行：(a) 三角形层级；以及(b) 像素层级，此一方式在遇到具有多重像素的三角形的状况时可减少所需计算工作量。本发明所提供的方法与第三图中所示的另一不同处则为本发明所需运算操作的数目较少，因而与先前技术中依序执行在屏幕空间中的三角形插补动作与透视修正动作的方式相比，本发明可提供较佳的运算效能。此外，本发明所提供的方法可避免先前技术中依序执行在屏幕空间中的插补动作与透视修正动作以便于回复观点空间中的真实值的过程所需的参数投影动作。

本发明的一特点为提供一种相当精确而快速的三角形插补方法，其是利用相同的算述逻辑单元(ALU)执行三角形与像素的处理工作，其中，上述的处理工作内容是不断穿插有用以处理三角形与用以处理像素的指令。本发明中执行插补工作的硬件单一指令多重资料(SIMD)单元的大小可调整以增加质心插补工作执行性能。此硬件单元可随所需的精确度与工作性能而作调整，并可用于其它的绘图处理工作，据以增加整体效益。此外，此可程序单一指令多重资料纯量单元(programmable SIMD scalar unit)也可符合Microsoft Dx9,10 and OpenGL API对于可程序绘图机器的硬件需求而用于更专业的像素处理工作；本发明所提供的三角形插补方法比先前技术更适合施行在可程序绘图处理单元上，且可提供逼近专属硬件的工作效率。

本发明所使用的是一施行于一通用可程序单元(universal programmable unit)上的装置，其中，此装置并可取代个别的深度、颜色以及纹理插补器。请参阅图6A所示，其为一根据本发明的一实施例所建构的可程序硬件单元10的示意图。此通用硬件单元10包含一可

程序单一指令多重资料(SIMD)纯量单元12以执行不同类型的插补工作。尤其，此可程序单一指令多重资料纯量单元12能够以逼近专属硬件的执行效率执行本发明所提供的三角形插补算法(triangle interpolation algorithm)。事实上，所有的插补操作皆是在尽可能精确的前提下，以浮点运算的方式进行的。上述的硬件单元10可包含复数个可程序单一指令多重资料纯量单元12以达到提高性能的目的。

上述的硬件单元10是为一具有双算术逻辑单元(ALU)的硬件单元，其中纯量算术逻辑单元16与18是配合一平移(shifted)处理周期以执行工作。上述周期的平移是藉由旁通缓存器(bypass register)30与32以达成。在个别时脉中，此硬件单元10根据所需的精确度以及所接收的三角形或像素处理相关指令以执行一个三角形或是复数个像素(例如：2到4个像素)的处理工作，其中，上述的三角形处理相关指令与像素处理相关指令是不固定地交互出现。另有一倒数单元(reciprocal unit)20是用以处理相除计算(division calculation)操作(参阅图4A、图4C与图4D中所示的方法及方程式)。上述的单一指令多重数据处理算法(SIMD Processing algorithm)是在一微码内存(microcode memory)14中执行且所有指令集皆经过最佳化处理以减少指令的数目。上述的硬件单元10也包含一三角形着色/描画器(triangle rasterizer)22；其所产生的空白像素屏幕坐标资料(blank pixel screen coordinates data)则储存于一像素内存单元(pixel memory unit)24中。此外，三角形在匀相空间中的坐标资料是由一顶点几何处理单元(vertex geometry processing unit)26产生，且此资料是储存于一三角形内存单元(triangle memory unit)28中。

参阅图6B所示，其为一单一指令多重数据算术逻辑单元指令集(SIMD ALU instruction set)。为缩短程序的长度(length of the program)，可将两种分别为长浮点(long floating point; long FP)类型以及短浮点(short FP)类型的资料个别或混合处理。执行图4D中方程式(6)的计算工作时所需的指令类型其中之一是为"交叉模式长指令"(Cross Mode Long instruction)，其内容为XPRDL、D、D"、P0与P1；其是用

以产生一交叉积值(cross product)，例如，指令"XPRDL、A0、V1.yw与V2.yw是用以计算一匀相系数a1。之后，一"折合长指令"(Folded Long instruction)则被用以计算di值；此折合长指令也可产生交叉积值，其内容则是为FSUBL、D、P0"与P2"。

为详述上述指令的使用方式，以下将提出实例程序以为说明：在此程序中，指令XPRDL被重复使用以因应执行方程式(6)中 i=0、1与2(也即 a0、a1、a2、b0、b1、b2、c0、c1以及c2)的状况所需的计算工作。应注意的是此程序中标号i等于0、1与2而非如上所述的1、2与3，然而计算结果则并不受此影响。此外，指令MOV、FBL (折合参杂模式；'Folded Blend Mode')、 FWD与 FSUBL (折合长；"Folded Long')等是用以因应在执行像素质心系数(α,β,γ)的计算工作时方程式(5)所需；指令MULL (长乘法；multiply long) 以及FSUBL则是用在如图4A中所示的像素属性插补工作。

具体而言，以下所列出的是对于(i)三角形建立工作；(ii)像素处理工作；以及(iii)像素属性资料插补工作而言所需的计算及其相对应的单一指令多重数据算术逻辑单元指令集(instruction set of SIMD ALU)：

```
//Barycentric triangle setup:
//The calculations as required by Equation (6) for index i=0:
# tri_a[0] = v[1].y * v[2].w - v[2].y * v[1].w
# tri_b[0] = v [2].x * v[1].w - v[1].x * v[2].w
# tri_c[0] = v[1].x * v[2].y - v[2].x * v[1].y
//
//The calculations as required by Equation (6) for index i=1:
# tri_a[1] = v[2].y * v[0].w - v[0].y * v[2].w
# tri_b[1] = v[0].x * v[2].w - v[2].x * v[0].w
# tri_c[1] = v[2].x * v[0].y - v[0].x * v[2].y
//
//The calculations as required by Equation (6) for index i=2:
```

tri_a[2] = v[0].y * v[1].w - v[1].y * v[0].w

tri_b[2] = v[1].x * v[0].w- v[0].x * v[1].w

tri_c[2] = v[0].x * v[1].y - v[1].x * v[0].y

实现上述的计算需执行下列指令(单一指令多重数据4位模式需执行9个指令):

//

//The instructions for executing calculations as required by Equation (6) for index i=0:

XPRDL A0, V1.yw, V2.yw

XPRDL B0, -V1.xw, V2.xw

XPRDL C0, V1.xy, V2.xy

//

//The instructions for executing calculations as required by Equation (6) for index i=1:

XPRDL A1, V2.yw, V0.yw

XPRDL B1, -V2.xw, V0.xw

XPRDL C1, V2.xy, V0.xy

//

//The instructions for executing calculations as required by Equation (6) for index i=2:

XPRDL A2, V0.yw, V1.yw

XPRDL B2, -V0.xw, V1.xw

XPRDL C2, V0.xy, V1.xy

//Pixel barycentric data processing, as required by Equation (5):

//Barycentric parameters calculation for each pixel, with calculations of di for i=0,1,2:

pix_d[0] = x * tri_a[0] + y * tri_b[0] + tri_c[0]

pix_d[1] = x* tri_a[1] + y* tri_b[1]+tri_c[1]

pix_d[2] = x * tri_a[2] + y * tri_b[2] + tri_c[2]

pix_d[3] = 1 / (pix_d[0] + pix_d[1] + pix d[2])

//

//Pixel barycentric coefficients (α , β , γ)calculation based on di:

alpha = pix_d[0] * pix_d[3]

#beta = pix_d[1] * pix_d[3]

gamma = 1 - alpha - beta

实现上述计算需执行下列指令(需执行7个指令):

//

MOV pr, xy

FBLDL D0, pr, AB0, C0

FBLMP NULL, D1, pr, AB1, C1, AB2, C2

FWD RCPL

MULL ALPHA, D0, ACC

MULL BETA, D1, ACC

FSUBL GAMMA, HW_ONE, ALPHA, 0, BETA

//Pixel attribute interpolation:

//Required calculations:

pix_att = v_att0 * alpha + v_attl * beta + v_att2 * gamma

Z interpolation mode - long operands

实现上述计算需执行下列指令(单一指令多重数据8位模式需执行两个指令):

//

MULL R0, ALPHA, V_ATT0

FBLDL PIX_ATT, BETA, V_ATT1, R0, GAMMA, V_ATT2

//Other attribute mode - short parameter operands (with 3 instructions, for SIMD 16 bits (word) mode):

```
//
MULM r0, v_att0, ALPHA
MADM r0, v_att1, BETA, r0
MADM pix_att, v_att2, GAMMA, r0
//
```

最后参阅图7所示，其为一单一指令多重数据-算术逻辑单元结构(SIMD-ALU structure)，此单元是用于根据本发明的一实施例所建构的方法中的短数据处理模式。如前所述，此单元是由算术逻辑单元16(ALU0 16)与算术逻辑单元18(ALU118)两个算术逻辑单元所构成。如图所示，算术逻辑单元16与算术逻辑单元18皆利用乘法运算(multiplication)、平移(shifting)以及其它额外的操作以处理上述各系数如c1、a1h、b0l、a1l、b1l、a0h、b0h、a0l、b1h与c0。算术逻辑单元16与算术逻辑单元18在一个平移周期操作(shift-cycle operation)中，藉由旁通缓存器30与旁通缓存器32以处理4个像素，其中，每个平移周期操作包含18x4=72个位。

由以上所述，本发明所提供的方法与插补算法能够以极高的效率与精确度在浮点算术单元上运作；此外，与先前技术相比，本发明可改善在三角形尺寸大于一个像素的状况时三角形插补工作的工作效率，并且藉由将匀相质心坐标的计算工作分成在三角形层级与像素层级上执行，本发明可简化执行像素层级的插补工作所需的硬件。

上述的详细叙述仅为本发明的较佳实施例，本发明仍可能有其它形式的实施方式，因此，本发明的精神与范围不应限制于上述的实施例中。

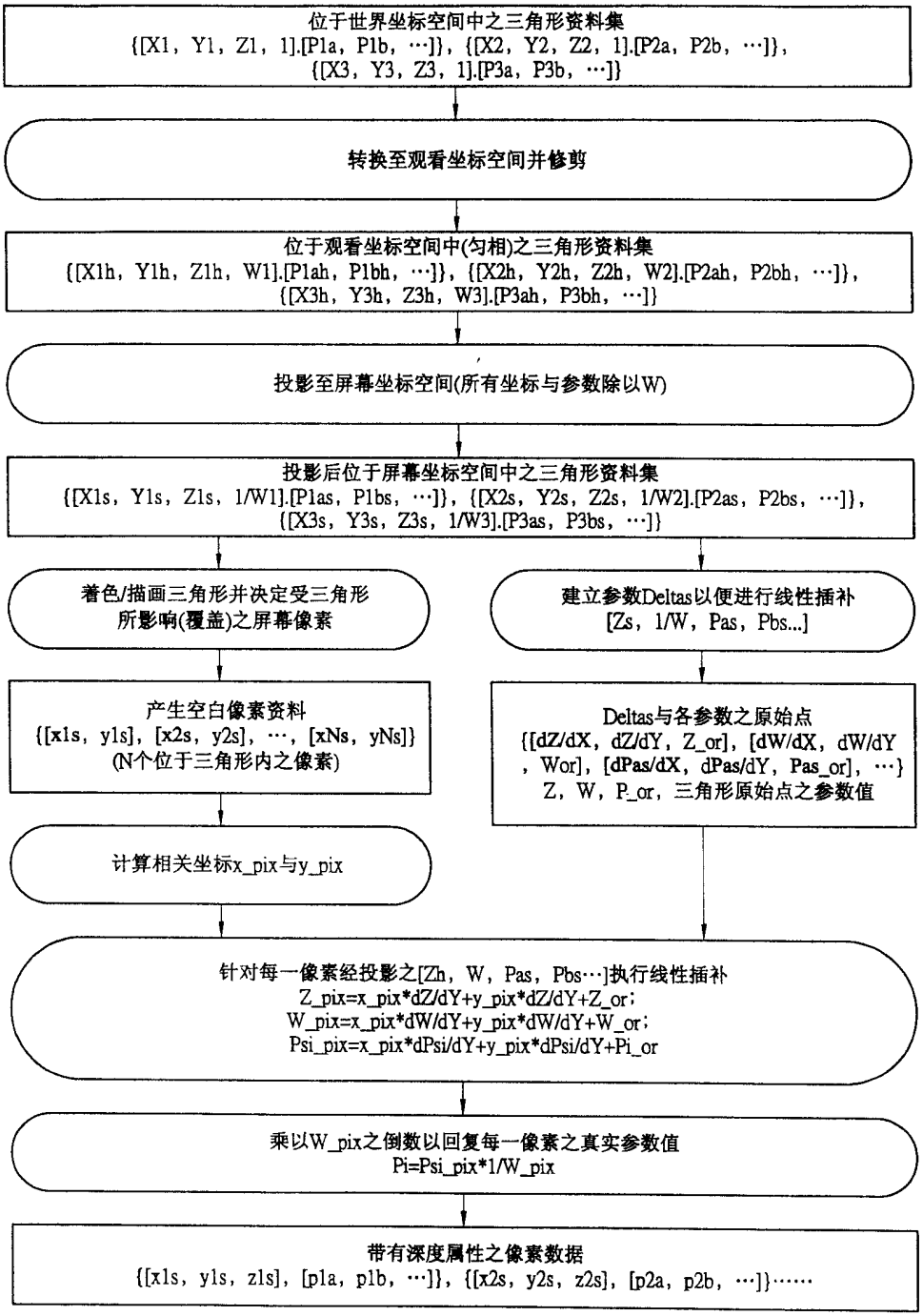

位于世界坐标空间中之三角形资料集
{[X1，Y1，Z1，1].[P1a，P1b，…]}，{[X2，Y2，Z2，1].[P2a，P2b，…]}，
{[X3，Y3，Z3，1].[P3a，P3b，…]}
转换至观看坐标空间并修剪
位于观看坐标空间中(匀相)之三角形资料集
{[X1h，Y1h，Z1h，W1].[P1ah，P1bh，…]}，{[X2h，Y2h，Z2h，W2].[P2ah，P2bh，…]}，
{[X3h，Y3h，Z3h，W3].[P3ah，P3bh，…]}
投影至屏幕坐标空间(所有坐标与参数除以W)
投影后位于屏幕坐标空间中之三角形资料集
{[X1s，Y1s，Z1s，1/W1].[P1as，P1bs，…]}，{[X2s，Y2s，Z2s，1/W2].[P2as，P2bs，…]}，
{[X3s，Y3s，Z3s，1/W3].[P3as，P3bs，…]}
着色/描画三角形并决定受三角形
所影响(覆盖)之屏幕像素
建立参数Deltas以便进行线性插补
[Zs，1/W，Pas，Pbs...]
产生空白像素资料
{[x1s，y1s]，[x2s，y2s]，…，[xNs，yNs]}
(N个位于三角形内之像素)
Deltas与各参数之原始点
{[dZ/dX，dZ/dY，Z_or]，[dW/dX，dW/dY
，Wor]，[dPas/dX，dPas/dY，Pas_or]，…}
Z，W，P_or，三角形原始点之参数值
计算相关坐标x_pix与y_pix
针对每一像素经投影之[Zh，W，Pas，Pbs…]执行线性插补
Z_pix=x_pix*dZ/dY+y_pix*dZ/dY+Z_or；
W_pix=x_pix*dW/dY+y_pix*dW/dY+W_or；
Psi_pix=x_pix*dPsi/dY+y_pix*dPsi/dY+Pi_or
乘以W_pix之倒数以回复每一像素之真实参数值
Pi=Psi_pix*1/W_pix
带有深度属性之像素数据
{[x1s，y1s，z1s]，[p1a，p1b，…]}，{[x2s，y2s，z2s]，[p2a，p2b，…]}……

图 1

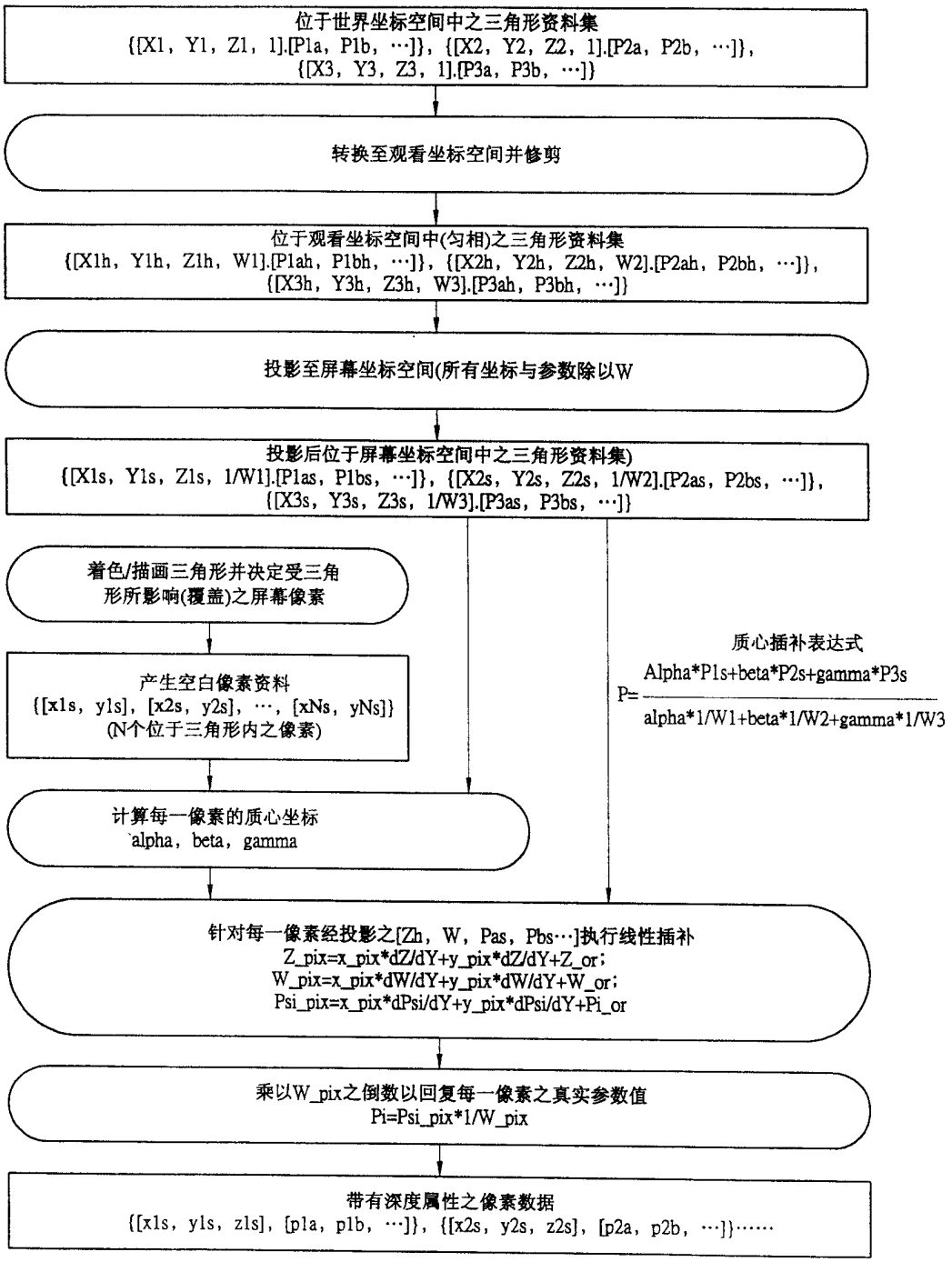

位于世界坐标空间中之三角形资料集
{[X1, Y1, Z1, 1].[P1a, P1b, …]}, {[X2, Y2, Z2, 1].[P2a, P2b, …]},
{[X3, Y3, Z3, 1].[P3a, P3b, …]}
转换至观看坐标空间并修剪
位于观看坐标空间中(匀相)之三角形资料集
{[X1h, Y1h, Z1h, W1].[P1ah, P1bh, …]}, {[X2h, Y2h, Z2h, W2].[P2ah, P2bh, …]},
{[X3h, Y3h, Z3h, W3].[P3ah, P3bh, …]}
投影至屏幕坐标空间(所有坐标与参数除以W
投影后位于屏幕坐标空间中之三角形资料集)
{[X1s, Y1s, Z1s, 1/W1].[P1as, P1bs, …]}, {[X2s, Y2s, Z2s, 1/W2].[P2as, P2bs, …]},
{[X3s, Y3s, Z3s, 1/W3].[P3as, P3bs, …]}
着色/描画三角形并决定受三角
形所影响(覆盖)之屏幕像素
产生空白像素资料
{[x1s, y1s], [x2s, y2s], …, [xNs, yNs]}
(N个位于三角形内之像素)
计算每一像素的质心坐标
`alpha, beta, gamma
质心插补表达式
P= Alpha*P1s+beta*P2s+gamma*P3s / alpha*1/W1+beta*1/W2+gamma*1/W3
针对每一像素经投影之[Zh, W, Pas, Pbs…]执行线性插补
Z_pix=x_pix*dZ/dY+y_pix*dZ/dY+Z_or;
W_pix=x_pix*dW/dY+y_pix*dW/dY+W_or;
Psi_pix=x_pix*dPsi/dY+y_pix*dPsi/dY+Pi_or
乘以W_pix之倒数以回复每一像素之真实参数值
Pi=Psi_pix*1/W_pix
带有深度属性之像素数据
{[x1s, y1s, z1s], [p1a, p1b, …]}, {[x2s, y2s, z2s], [p2a, p2b, …]}……

图 2

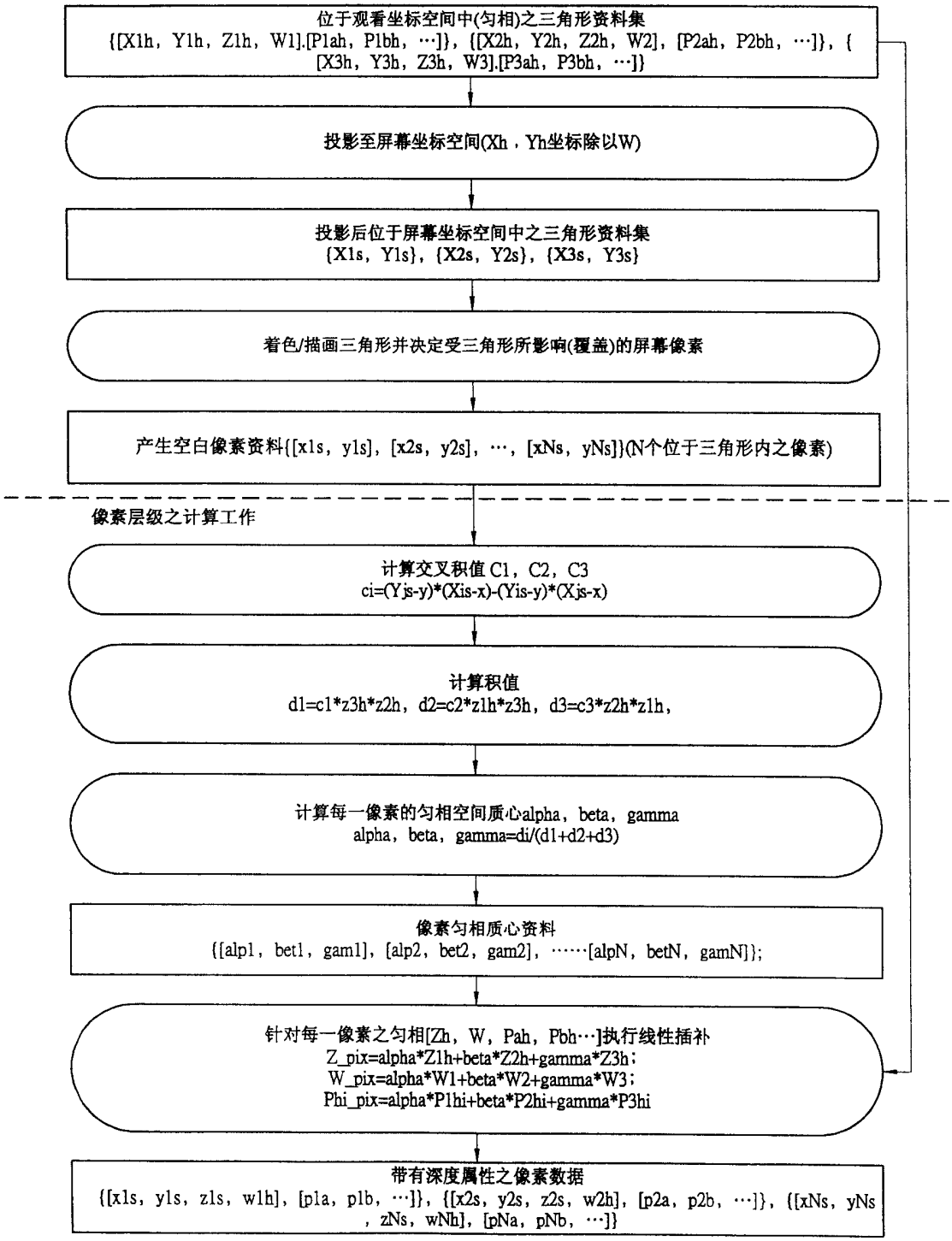

位于观看坐标空间中(匀相)之三角形资料集
{[X1h, Y1h, Z1h, W1].[P1ah, P1bh, …]}, {[X2h, Y2h, Z2h, W2], [P2ah, P2bh, …]}, {[X3h, Y3h, Z3h, W3].[P3ah, P3bh, …]}
投影至屏幕坐标空间(Xh，Yh坐标除以W)
投影后位于屏幕坐标空间中之三角形资料集
{X1s, Y1s}, {X2s, Y2s}, {X3s, Y3s}
着色/描画三角形并决定受三角形所影响(覆盖)的屏幕像素
产生空白像素资料{[x1s, y1s], [x2s, y2s], …, [xNs, yNs]}(N个位于三角形内之像素)
像素层级之计算工作
计算交叉积值 C1, C2, C3
ci=(Yjs-y)*(Xis-x)-(Yis-y)*(Xjs-x)
计算积值
d1=c1*z3h*z2h, d2=c2*z1h*z3h, d3=c3*z2h*z1h,
计算每一像素的匀相空间质心alpha, beta, gamma
alpha, beta, gamma=di/(d1+d2+d3)
像素匀相质心资料
{[alp1, bet1, gam1], [alp2, bet2, gam2], ……[alpN, betN, gamN]};
针对每一像素之匀相[Zh, W, Pah, Pbh…]执行线性插补
Z_pix=alpha*Z1h+beta*Z2h+gamma*Z3h;
W_pix=alpha*W1+beta*W2+gamma*W3;
Phi_pix=alpha*P1hi+beta*P2hi+gamma*P3hi
带有深度属性之像素数据
{[x1s, y1s, z1s, w1h], [p1a, p1b, …]}, {[x2s, y2s, z2s, w2h], [p2a, p2b, …]}, {[xNs, yNs, zNs, wNh], [pNa, pNb, …]}

图 3

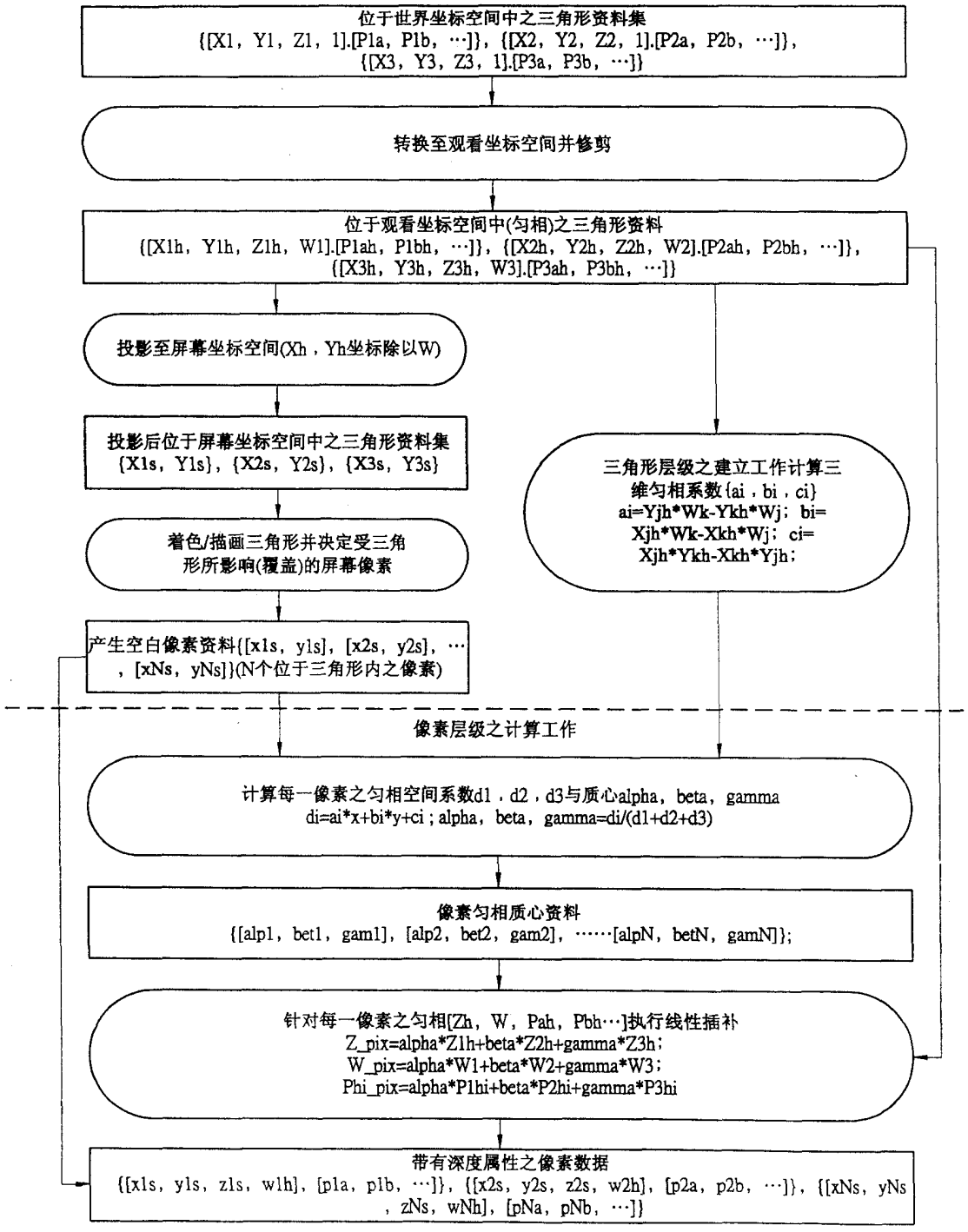

位于世界坐标空间中之三角形资料集
{[X1, Y1, Z1, 1].[P1a, P1b, …]}, {[X2, Y2, Z2, 1].[P2a, P2b, …]},
{[X3, Y3, Z3, 1].[P3a, P3b, …]}
转换至观看坐标空间并修剪
位于观看坐标空间中(匀相)之三角形资料
{[X1h, Y1h, Z1h, W1].[P1ah, P1bh, …]}, {[X2h, Y2h, Z2h, W2].[P2ah, P2bh, …]},
{[X3h, Y3h, Z3h, W3].[P3ah, P3bh, …]}
投影至屏幕坐标空间(Xh，Yh坐标除以W)
投影后位于屏幕坐标空间中之三角形资料集
{X1s, Y1s}, {X2s, Y2s}, {X3s, Y3s}
着色/描画三角形并决定受三角形所影响(覆盖)的屏幕像素
产生空白像素资料{[x1s, y1s], [x2s, y2s], …, [xNs, yNs]}(N个位于三角形内之像素)
三角形层级之建立工作计算三维匀相系数{ai，bi，ci}
ai=Yjh*Wk-Ykh*Wj; bi=Xjh*Wk-Xkh*Wj; ci=Xjh*Ykh-Xkh*Yjh;
像素层级之计算工作
计算每一像素之匀相空间系数d1，d2，d3与质心alpha, beta, gamma
di=ai*x+bi*y+ci ; alpha, beta, gamma=di/(d1+d2+d3)
像素匀相质心资料
{[alp1, bet1, gam1], [alp2, bet2, gam2], ……[alpN, betN, gamN]};
针对每一像素之匀相[Zh, W, Pah, Pbh…]执行线性插补
Z_pix=alpha*Z1h+beta*Z2h+gamma*Z3h;
W_pix=alpha*W1+beta*W2+gamma*W3;
Phi_pix=alpha*P1hi+beta*P2hi+gamma*P3hi
带有深度属性之像素数据
{[x1s, y1s, z1s, w1h], [p1a, p1b, …]}, {[x2s, y2s, z2s, w2h], [p2a, p2b, …]}, {[xNs, yNs, zNs, wNh], [pNa, pNb, …]}

图 4A

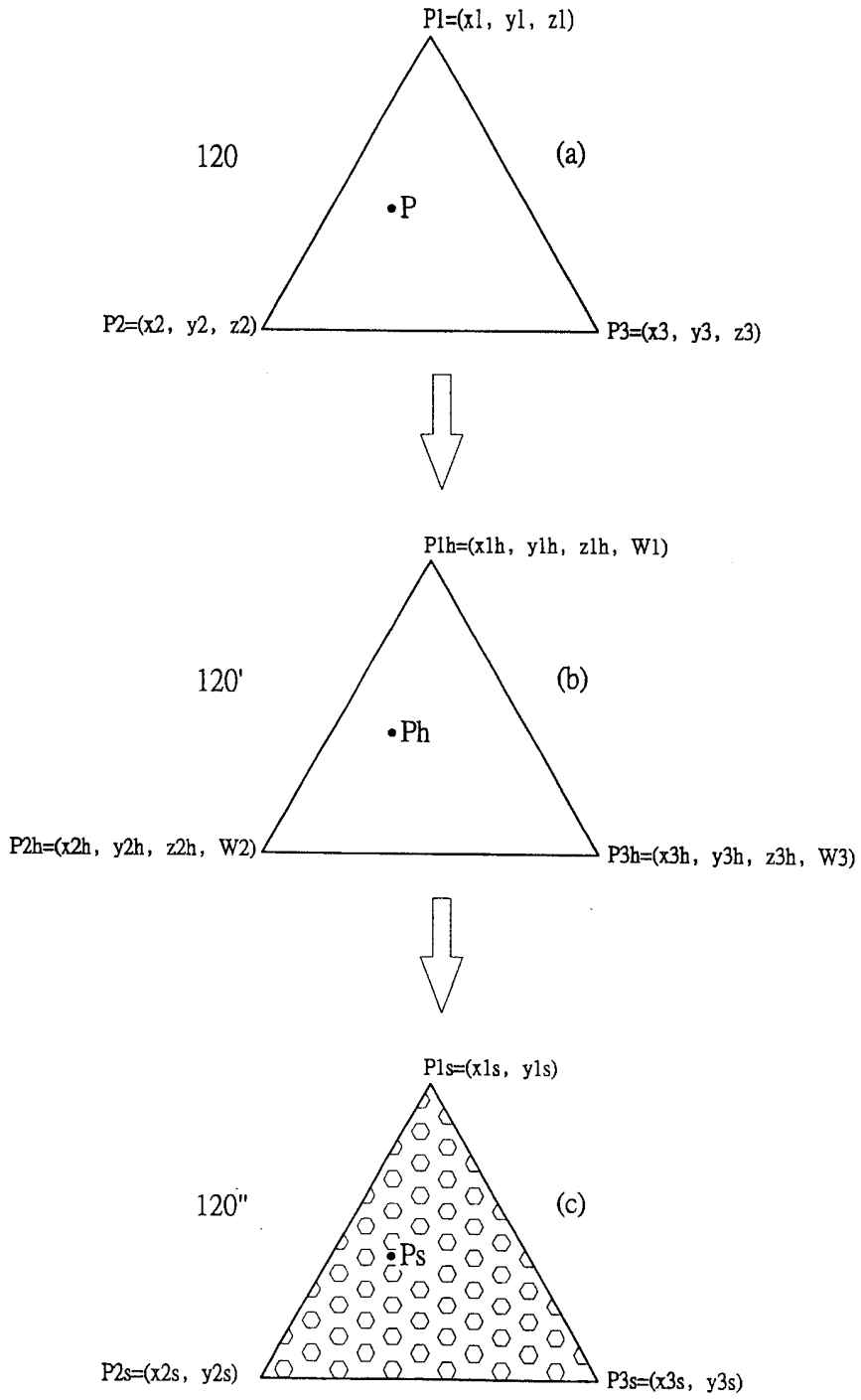

P1=(x1, y1, z1)
120
(a)
•P
P2=(x2, y2, z2)
P3=(x3, y3, z3)
P1h=(x1h, y1h, z1h, W1)
120'
(b)
•Ph
P2h=(x2h, y2h, z2h, W2)
P3h=(x3h, y3h, z3h, W3)
P1s=(x1s, y1s)
120"
(c)
•Ps
P2s=(x2s, y2s)
P3s=(x3s, y3s)

图 4B

质心建立方式

对于三角形内的每一个像素，其坐标皆可利用质心系数而以三角形三顶点之坐标表示：

$$\begin{cases} x=\alpha\bullet x_1+\beta\bullet x_2+\gamma\bullet x_3 \\ y=\alpha\bullet y_1+\beta\bullet y_2+\gamma\bullet y_3 \\ \alpha+\beta+\gamma=1 \\ 0\le\alpha\le 1 \\ 0\le\beta\le 1 \\ 0\le\gamma\le 1 \end{cases}$$

质心系数α、β与γ可用以计算多种不同的像素属性，包含 Z 方向坐标(深度)、颜色、UV 坐标 等。因此，得出质心系数之后则三角形内任一像素的任一属性皆可藉由三角形三顶点之属性值计算而得。质心系数之计算方式如下：

$$\begin{aligned} &\alpha=((x_3-x_2)\bullet y+(y_2-y_3)\bullet x+x_2\bullet y_3-x_3\bullet y_2)\bullet\delta \\ &\beta=((x_1-x_3)\bullet y+(y_3-y_1)\bullet x+x_3\bullet y_1-x_1\bullet y_3)\bullet\delta \\ &\gamma=((x_2-x_1)\bullet y+(y_1-y_2)\bullet x+x_1\bullet y_2-x_2\bullet y_1)\bullet\delta \\ &\delta=1/(x_1\bullet y_2-x_2\bullet y_1+x_2\bullet y_3-x_3\bullet y_2+x_3\bullet y_1-x_1\bullet y_3) \\ &\alpha+\beta+\gamma=1 \\ &0\le\alpha\le 1 \\ &0\le\beta\le 1 \\ &0\le\gamma\le 1 \end{aligned} \qquad (1)$$

令 i=1,2,3；j=imod3+1；k=jmod3+1(三角形三顶点之顺序编号)

且

$$\begin{cases} a_i=y_j-y_k \\ b_i=x_k-x_j \\ c_i=x_j\bullet y_k-x_k\bullet y_j \end{cases} \qquad (2)$$

则(1)式可改写成：

$$\begin{cases} \alpha=(a_1\bullet x+b_1\bullet y+c_1)\bullet\delta \\ \beta=(a_2\bullet x+b_2\bullet y+c_2)\bullet\delta \\ \gamma=1-\alpha-\beta \\ \delta=1/(c_1+c_2+c_3) \end{cases} \qquad (3)$$

图 4C

进行透视修正时，每一个像素有一相对应之 w：

$$w = \frac{1}{\frac{\alpha}{w_1} + \frac{\beta}{w_2} + \frac{\gamma}{w_3}} \tag{4}$$

接着计算在匀相空间中的质心系数，其中像素的坐标系表示如下：

$$\tilde{x}_i = x_i \bullet w_i$$
$$\tilde{y}_i = y_i \bullet w_i$$
$$\tilde{x} = x \bullet w$$
$$\tilde{y} = y \bullet w$$

经计算后得到：

$$d_i = (a_i \bullet x + b_i \bullet y + c_i) \bullet w_j \bullet w_k$$
$$\tilde{\alpha} = \frac{d_1}{d_1 + d_2 + d_3}$$
$$\tilde{\beta} = \frac{d_2}{d_1 + d_2 + d_3} \tag{5}$$
$$\tilde{\gamma} = \frac{d_3}{d_1 + d_2 + d_3}$$

又，由(2)式以及(5)式已知：

$$a_i = y_j - y_k$$
$$b_i = x_k - x_j$$
$$c_i = x_j \bullet y_k - x_k \bullet y_j$$
$$d_i = (a_i \bullet x + b_i \bullet y + c_i) \bullet w_j \bullet w_k$$

因此可简化得到：

$$d_i = ((y_j - y_k) \bullet x + (x_k - x_j) \bullet y + x_j \bullet y_k - x_k \bullet y_j) \bullet w_j \bullet w_k \Rightarrow$$
$$d_i = (y_j \bullet w_j \bullet w_k - y_k \bullet w_j \bullet w_k) \bullet x + (x_k \bullet w_j \bullet w_k - x_j \bullet w_j \bullet w_k) \bullet y + x_j \bullet y_k \bullet w_j \bullet w_k - x_k \bullet y_j \bullet w_j \bullet w_k \Rightarrow$$
$$d_i = (\tilde{y}_j \bullet w_k - \tilde{y}_k \bullet w_j) \bullet x + (\tilde{x}_k \bullet w_j - \tilde{x}_j \bullet w_k) \bullet y + \tilde{x}_j \bullet \tilde{y}_k - \tilde{x}_k \bullet \tilde{y}_j \Rightarrow$$
$$\begin{cases} \tilde{a}_i = \tilde{y}_j w_k - \tilde{y}_k w_j \\ \tilde{b}_i = \tilde{x}_k w_j - \tilde{x}_j w_k \\ \tilde{c}_i = \tilde{x}_j \tilde{y}_k - \tilde{x}_k \tilde{y}_j \\ d_i = \tilde{a}_i x + \tilde{b}_i y + \tilde{c}_i \end{cases} \tag{6}$$

每一个三角形有一组 ai、bi 与 ci；每一个像素有一组 di，以作为之后根据(5)式对质心作进一步计算时的基础。

图 4D

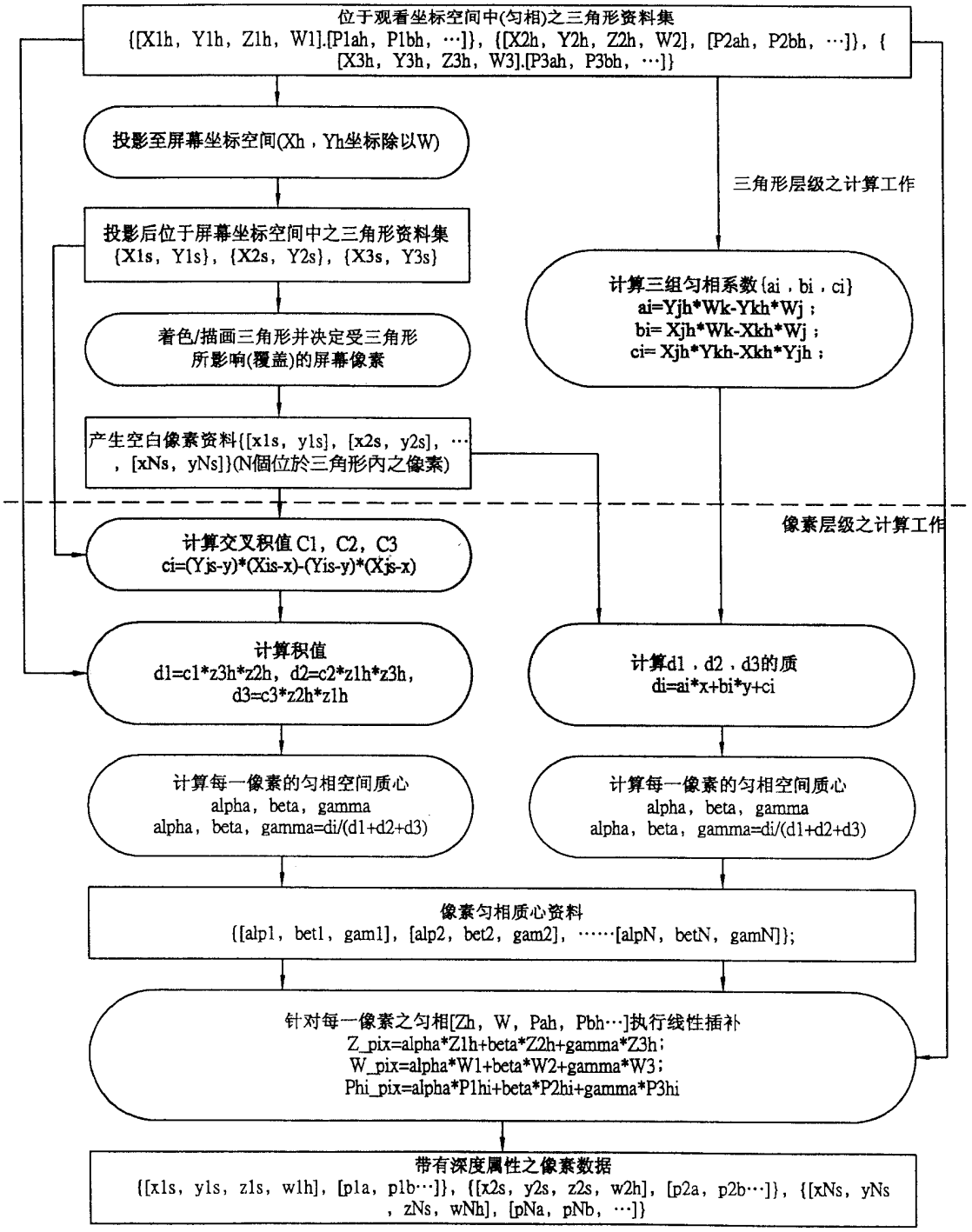

位于观看坐标空间中(匀相)之三角形资料集
{[X1h, Y1h, Z1h, W1].[P1ah, P1bh, …]}, {[X2h, Y2h, Z2h, W2], [P2ah, P2bh, …]}, {[X3h, Y3h, Z3h, W3].[P3ah, P3bh, …]}
投影至屏幕坐标空间(Xh，Yh坐标除以W)
三角形层级之计算工作
投影后位于屏幕坐标空间中之三角形资料集
{X1s, Y1s}, {X2s, Y2s}, {X3s, Y3s}
计算三组匀相系数{ai，bi，ci}
ai=Yjh*Wk-Ykh*Wj；
bi= Xjh*Wk-Xkh*Wj；
ci= Xjh*Ykh-Xkh*Yjh；
着色/描画三角形并决定受三角形所影响(覆盖)的屏幕像素
产生空白像素资料{[x1s, y1s], [x2s, y2s], …, [xNs, yNs]}(N個位於三角形內之像素)
像素层级之计算工作
计算交叉积值 C1, C2, C3
ci=(Yjs-y)*(Xis-x)-(Yis-y)*(Xjs-x)
计算积值
d1=c1*z3h*z2h, d2=c2*z1h*z3h, d3=c3*z2h*z1h
计算d1，d2，d3的质
di=ai*x+bi*y+ci
计算每一像素的匀相空间质心
alpha, beta, gamma
alpha, beta, gamma=di/(d1+d2+d3)
计算每一像素的匀相空间质心
alpha, beta, gamma
alpha, beta, gamma=di/(d1+d2+d3)
像素匀相质心资料
{[alp1, bet1, gam1], [alp2, bet2, gam2], ……[alpN, betN, gamN]};
针对每一像素之匀相[Zh, W, Pah, Pbh…]执行线性插补
Z_pix=alpha*Z1h+beta*Z2h+gamma*Z3h;
W_pix=alpha*W1+beta*W2+gamma*W3;
Phi_pix=alpha*P1hi+beta*P2hi+gamma*P3hi
带有深度属性之像素数据
{[x1s, y1s, z1s, w1h], [p1a, p1b…]}, {[x2s, y2s, z2s, w2h], [p2a, p2b…]}, {[xNs, yNs, zNs, wNh], [pNa, pNb, …]}

图 5A

所需操作步骤数目																		
	先前技术	本发明																
	三角形层级操作																	
乘法器	0	18																
加法/减法器	0	9																
	像素层级操作																	
乘法器	12	6																
加法/减法器	12	6																
三角形大小 (以像素数目表示)	1	1	2	2	4	4	8	8	16	16	32	32	64	64	128	128	256	256
乘法运算操作	12	24	24	30	48	42	96	66	192	114	384	210	768	402	1536	786	3072	1554
加法/乘法运算操作	12	15	24	21	48	33	96	57	192	105	384	201	768	393	1536	777	3072	1545
操作步骤数目总计	24	39	48	51	96	75	192	123	384	219	768	411	1536	795	3072	1563	6144	3099

图 5B

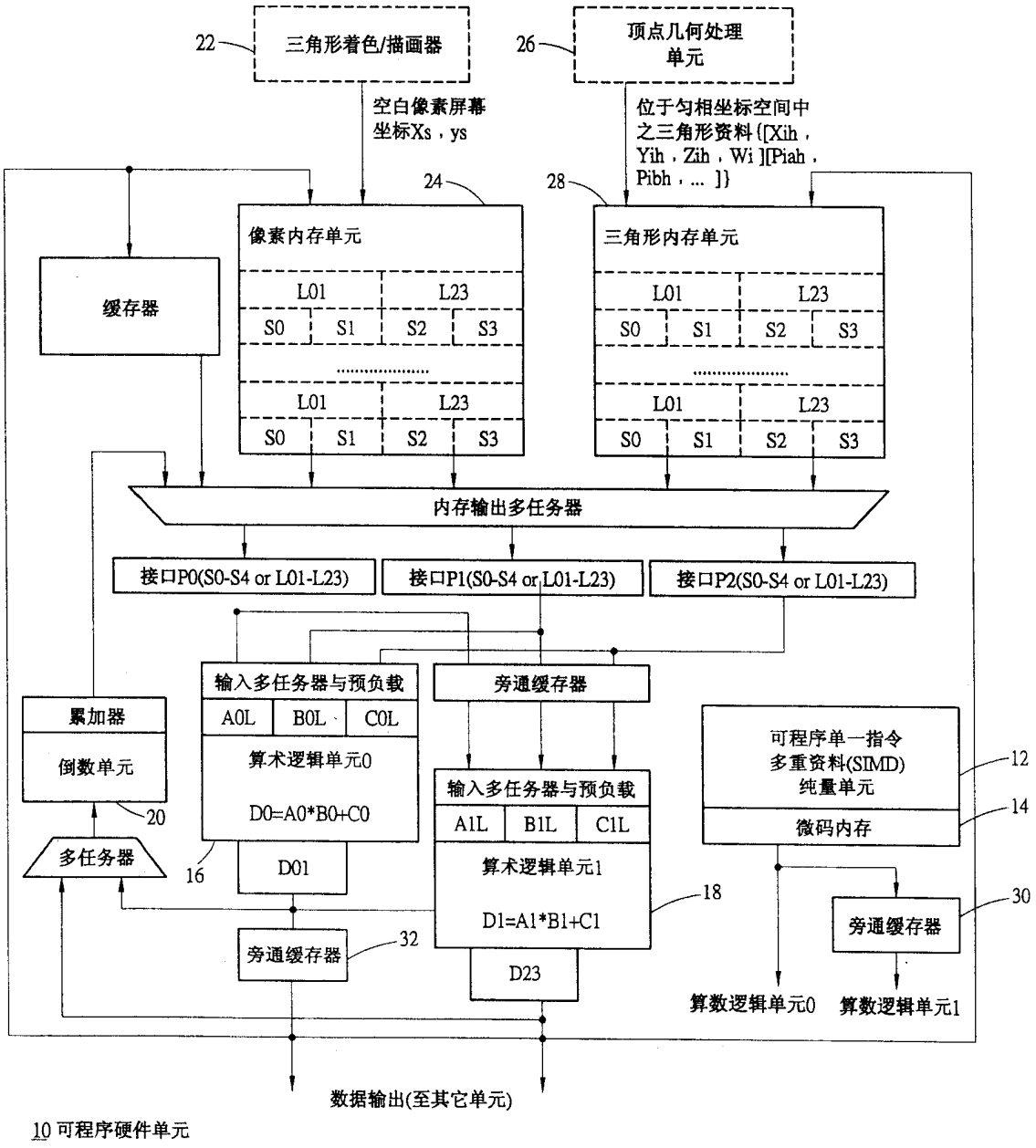

22
三角形着色/描画器
26
顶点几何处理单元
空白像素屏幕坐标Xs，ys
位于匀相坐标空间中之三角形资料{[Xih，Yih，Zih，Wi][Piah，Pibh，...]}
24
像素内存单元
28
三角形内存单元
缓存器
L01
L23
S0
S1
S2
S3
内存输出多任务器
接口P0(S0-S4 or L01-L23)
接口P1(S0-S4 or L01-L23)
接口P2(S0-S4 or L01-L23)
累加器
倒数单元
20
多任务器
输入多任务器与预负载
A0L
B0L
C0L
算术逻辑单元0
D0=A0*B0+C0
16
D01
旁通缓存器
32
输入多任务器与预负载
A1L
B1L
C1L
算术逻辑单元1
D1=A1*B1+C1
18
D23
旁通缓存器
可程序单一指令多重资料(SIMD)纯量单元
12
微码内存
14
30
旁通缓存器
算数逻辑单元0
算数逻辑单元1
数据输出(至其它单元)
10 可程序硬件单元

图 6A

	执行模式	指令 （大写字：长格式 普通字：短格式）	模式与资料之格式 三角形，像素，长短， 混合	功能与注释 （大写字：长格式 普通字：短格式）
	1 混合模式 混合之 18-36 位操作 <M>	MADE d，p0，P1c，P2	Pixel mixed	d=p0*p1c+p2 P1c-long Common operand for 2 short format channels according to SIMD factor for operand
X	混合模式	ADDM d，p0，p2	Same	d=p0+p2
X	混合模式	SUBM d，p0，p2	Same	d=p0-p2
	2 混合模式	MULM d，p0，P1c	Same	d=p0*P1c
	3 混合模式	MACM d，p0，P1c	Same	d=p0*P1c+macc
	4 混合模式 36-位操作数	MADL D，P0，P1，P2	Triangle Long Pixel Long	D=P0*P1+P2
	5 长模式	ADDL D，P0，P2	Same	D=P0+P2
	6 长模式	SUBL D，P0，P2	Same	D=P0-P2
	7 长模式	MULL D，P0，P1	Same	D=P0*P1
	8 长模式	MACL D，P0，P1	Same	D=P0*P1+MACC
	折合长模式 9 36-位操作数 <F...L>	FMADL D，P0"，P1"，P2" Folded instruction has doubled set of operands	Triangle Long	D=(P0*P1+P2)+(P0"*P1"+P2")"-operands in folding channel
	10 折合长模式	FADDL D，P0"，P2"	Same	D=(P0+P2)+(P0"+P2")
	11 折合长模式	FSUBL D，P0"，P2"	Same	D=(P0-P2)+(P0"-P2")
	12 折合长模式	FMULL D，P0"，P1"	Same	D=(P0*P1)+(P0"*P1")
	13 折合长模式	FMACL D，P0"，P1"	Same	D=(P0*P1+Macc)+(P0"*P1"+Macc")
	14 折合参杂模式 混合模式预负载	FBLMP D，D" x，y，PO P1，P2，PO"，P1"，P2"，	Same	D"=x*P"0+y*P"1+P2"D=x*P0+y*P1+P2+D"
	15 折合参杂模式 长模式	FBLDL D，P0"，P1"，P2"	Triangle Long Similar FMADL	D=(P0"*P1"+P2")+(P0*P1+'0')"-cperands in folding channel
	16 交叉模式	XPRDL D，D"，P0，P1	Same	D"=P0.23*P1.01 D=P0.01*P1.23-D Cross Product

图 6B

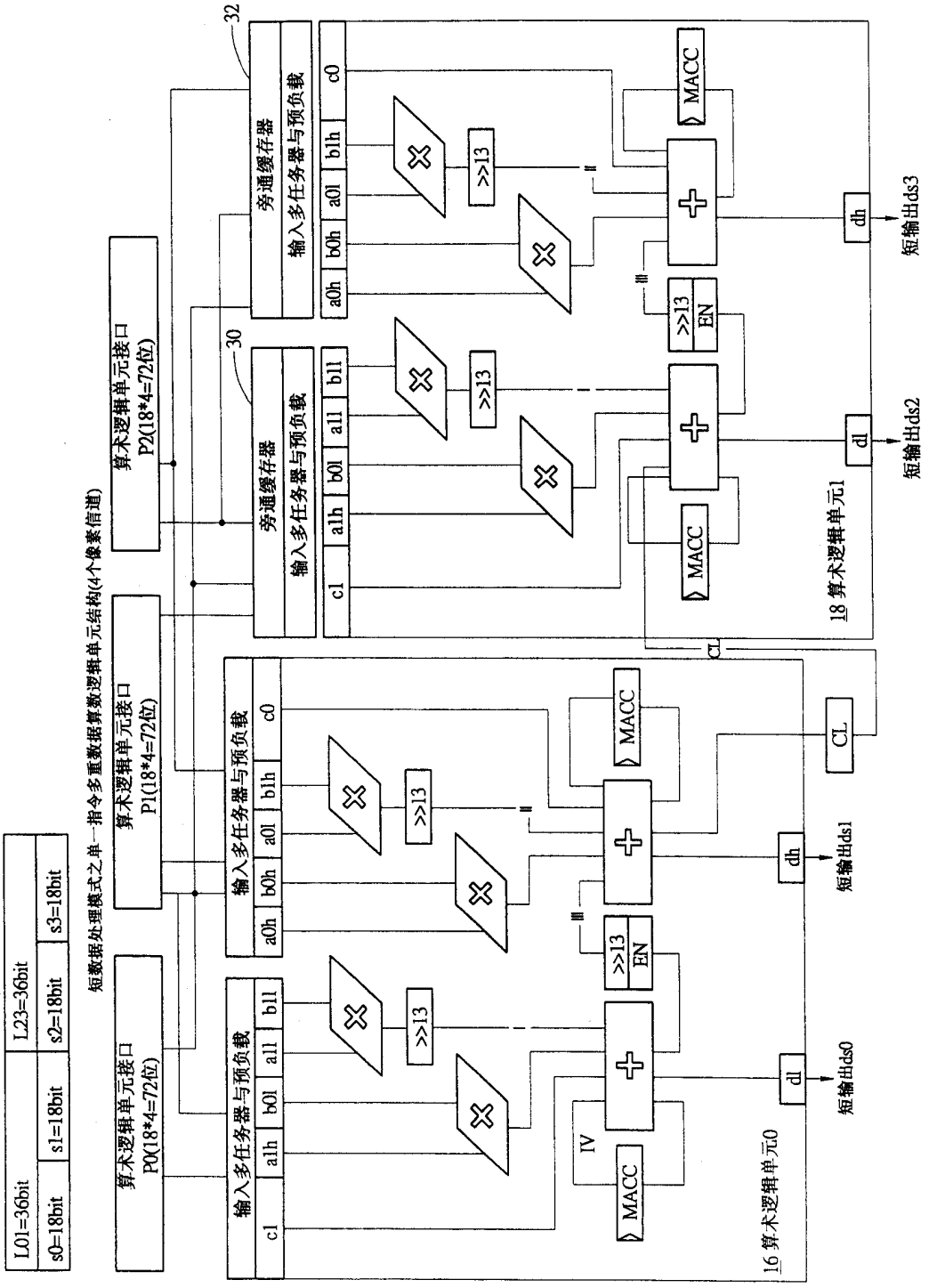

L01=36bit
L23=36bit
s0=18bit
s1=18bit
s2=18bit
s3=18bit
短数据处理模式之单一指令多重数据算数逻辑单元结构(4个像素信道)
算术逻辑单元接口 P0(18*4=72位)
算术逻辑单元接口 P1(18*4=72位)
算术逻辑单元接口 P2(18*4=72位)
旁通缓存器
输入多任务器与预负载
32
30
c1 a1h b0l a1l b1l
a0h b0h a0l b1h c0
>>13
MACC
IV
CL
dl
dh
短输出ds0
短输出ds1
短输出ds2
短输出ds3
16 算术逻辑单元0
18 算术逻辑单元1

图 7