



[12] 发明专利说明书

[21] ZL 专利号 97193508.4

[45] 授权公告日 2003 年 10 月 1 日

[11] 授权公告号 CN 1123209C

[22] 申请日 1997.11.17 [21] 申请号 97193508.4

[30] 优先权

[32] 1996.12.6 [33] EP [31] 96203399.9

[32] 1997.3.13 [33] EP [31] 97200753.8

[86] 国际申请 PCT/IB97/01451 1997.11.17

[87] 国际公布 WO98/25403 英 1998.6.11

[85] 进入国家阶段日期 1998.9.29

[71] 专利权人 皇家飞利浦电子有限公司

地址 荷兰艾恩德霍芬

[72] 发明人 E·J·波尔 M·范斯普伦特

审查员 魏 玮

[74] 专利代理机构 中国专利代理(香港)有限公司

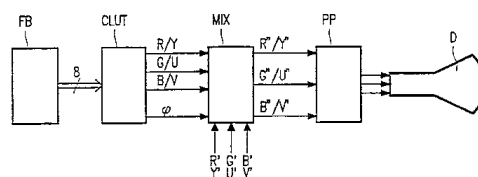
代理人 吴增勇 张志醒

权利要求书 1 页 说明书 10 页 附图 1 页

[54] 发明名称 图形信号与视频信号的混合方法和装置及含该装置的多媒体装置

[57] 摘要

在将图形信号与视频信号按照比率 $\alpha : (1 - \alpha)$ 混合的方法中, 图形信号是通过对存储有颜色值(R, G, B)的查色表(CLUT)寻址而获得的, 对于每一混合比, 所存储的颜色值的数目取决于该混合比, 以便获得最佳的查色表的填充。



1.一种将颜色值(R,G,B)和混合因子(α)存储到查色表(CLUT)内的方法,其特征在于:对于每一个混合因子(α),所存储的颜色值(R,G,B)的数目取决于所述混合因子(α)。

2.一种把图形信号(R,G,B)与视频信号(R',G',B')按照比率 $\alpha:(1-\alpha)$ 相混合的方法,其中图形信号(R,G,B)是通过对存储有颜色值(R,G,B)的查色表(CLUT)寻址而获得的,其特征在于:对于每一混合比,所存储的颜色值(R,G,B)的数目取决于所述混合比。

3.一种把图形信号(R,G,B)与视频信号(R',G',B')按照比率 $\alpha:(1-\alpha)$ 相混合的装置,其特征在于该混合装置包括:

一颜色查色表(CLUT),以如此方式存储颜色值(R, G, B),即对于每一混合比,所存储的颜色值(R, G, B)的数目取决于所述混合比;以及

通过对所述颜色查色表(CLUT)的寻址而获取图形信号(R, G, B)的装置(FB)。

4.根据权利要求3的装置,其特征在于:对于每一混合比,所存储的颜色值(R,G,B)的数目取决于因子(α)的三次方,在混合运算中,颜色值(R,G,B)与所述因子相乘。

5.根据权利要求3的装置,其特征在于:不同的绿颜色值(G)的数目等于或超过不同的红颜色值(R)的数目,并且,不同的红颜色值(R)的数目等于或超过不同的蓝颜色值(B)的数目。

6.根据权利要求5的装置,其特征在于:不同的混合比的数目等于或超过不同的绿颜色值(G)的数目。

7.一种多媒体装置,其特征在于包括:

由权利要求3所定义的图形和视频信号的混合装置;以及
与所述混合装置输出端连接的显示器(D)。

图形信号与视频信号的混合方法和装置
及含该装置的多媒体装置

5

本发明涉及将图形信号与视频信号混合的方法和装置，以及含有这种装置的多媒体装置。

10 本发明的目的之一是提供一种将图形信号与视频信号混合的方法，以获得硬件元件的最佳效能。为了这一目的，本发明的第一方面提供一种将颜色值(R, G, B)和混合因子(α)存储到查色表(CLUT)内的方法，其中，对于每一个混合因子(α)，所存储的颜色值(R, G, B)的数目取决于所述混合因子(α)。本发明的第二方面定义一种把图形信号(R, G, B)与视频信号(R', G', B')按照比率 α : (1- α)相混合的方法，其中图形信号(R, G, B)是通过对存储有颜色值(R, G, B)的查色表(CLUT)寻址而获得的，其中，对于每一混合比，所存储的颜色值(R, G, B)的数目取决于所述混合比。本发明的第三方面定义一种把图形信号(R, G, B)与视频信号(R', G', B')按照比率 α : (1- α)相混合的装置，其中图形信号(R, G, B)是通过对存储有颜色值(R, G, B)的查色表(CLUT)寻址而获得的，其中，对于每一混合比，所存储的颜色值(R, G, B)的数目取决于所述混合比。本发明的第四方面定义一种多媒体装置，它包括：由本发明的第三方面所定义的图形和
15 视频信号的混合装置；以及与所述混合装置输出端连接的显示器(D)。

25 对于每一混合比，所存储的颜色值(R, G, B)的数目取决于因子(α)的三次方，在混合运算中，颜色值(R, G, B)与所述因子相乘。

不同的绿颜色值(G)的数目等于或超过不同的红颜色值(R)的数目，并且，不同的红颜色值(R)的数目等于或超过不同的蓝颜色值(B)的数目。

不同的混合比的数目等于或超过不同的绿颜色值(G)的数目。

本发明的第一方面提供一种按照 $\alpha:(1-\alpha)$ 的比率将图形信号与视频信号混合的方法，通过对查色表寻址来获得图形信号，在所述查色表中存储有颜色值，为了获得查色表的最佳填充，根据混合比在所述查色表中存储有对于每一个混合比的多个颜色值。“颜色值”一词不仅包括红，绿，蓝的颜色值，还包括利用亮度值和色度值所表示的颜色值。

下面将参考描述的实施例说明本发明的这些以及其它方面，从这些实施例将明白本发明的这些及其他方面。

附图中，图1示出根据本发明的显示装置。

图1中，帧缓冲器FB向查色表CLUT提供8位地址信号。查色表CLUT向混合器MIX的第一输入端提供颜色值R（红），G（绿）和B（蓝），或亮度信号Y和色度信号U，V，以及混合因子 α （ α ）。混合器MIX的第二输入端以颜色信号R'，G'，B'的形式，或以亮度信号Y'和色度信号U'，V'的形式接收视频信号。混合器MIX以颜色信号R''，G''，B''的形式，或以亮度信号Y''和色度信号U''，V''的形式提供输出信号。混合器的输出信号通过后处理单元PP加到显示器D上。如果利用亮度信号Y''和色度信号U''，V''建立所述混合器的输出

则后处理单元PP将确保向RGB颜色信号的必要的转换。图中的显示装置可以是装备有图文电视功能的电视接收机；然而并不排除其它的应用，这样本发明可以广泛地应用于任何多媒体装置，在这些多媒体装置中，图像信息和视频信号在被显示之前先进行混合。包括具有预定内容（标准表）的查色表CLUT的终端（例如，电视机的顶置盒）能够接收和显示根据标准表编码的位映图，不需要传送具有位映图的标准表，也不需要在此终端内进行复杂的颜色变换。

本发明涉及包括帧缓冲器FB，查色表CLUT，和视频混合器MIX的显示系统。图像被存储在帧缓冲器内，并且能够通过视频混合系统、以几个透明度级别来显示所述图像。本发明的主要思想是如何有效地填充查色表的可用空间，以及如何将查色表CLUT的使用与图像处理

技术（如，混色和颜色空间转换）相结合。

在显示处理期间帧缓冲器 FB 所产生的每一像素资料被用来作为查色表（CLUT）的地址，在所述表中，存储有三个颜色值和一个透明度值。关于颜色成分的有用的选择是红，绿和蓝三色一组，或者 YUV 三个一组。透明度值通常由 α 来表示。混合器取出由查色表 clut 所产生的图像颜色值、并将它们与 α 相乘，取出视频颜色值、将它们与 $(1-\alpha)$ 相乘，并将两个乘积相加，得出图像和视频输入的具有由 α 确定的权重的加权平均值。这样，每一像素可以具有不同的权重。（完全透明： $\alpha=0$ ，不透明： $\alpha=1$ ）

在消费系统中，存储在帧缓冲器内的每一像素的位数和查色表 CLUT 的入口数通常非常小。一般来说，帧缓冲器 FB 存储每一像素 8 位，查色表 CLUT 的入口数大约为 256。这意味着必须认真考虑查色表内可用入口的使用。有几种填充查色表的好方法，但是这些方法只适用于产生颜色值的查色表 CLUT。这里所讨论的查色表同时产生 α 值，因此，就需要一种新的查色表填充方法。

最简单的方案是将彩色影像空间内规则的网格分配给查色表 CLUT。这意味着我们为三个颜色成分和 α 通道选择多个级别，并用这些级别的笛卡尔乘积（即，全组合）来填充查色表。例如，我们可以为每一通道指定 4 个级别，结果 $4 \times 4 \times 4 \times 4 = 256$ 个入口，精确地填充查色表 CLUT。假设我们处理 RGB 颜色分量，比较聪明的选择是（R G B α ） $4 \times 5 \times 3 \times 4 = 240$ 。这样即使查色表不能完全被填充，但实际上结果图像可能更好，这是因为绿色通道对于图像质量起着决定的作用。

当需要对准备存储在帧缓冲器内的图像进行混色时，用笛卡尔乘积来填充查色表是有用的。对图像进行混色是一种技术（或多种技术的结合），通过这种技术在保持可接受的图像质量的同时，单个图像内颜色数目将被减少。混色算法越复杂，图像质量越高，但同时也需要更强的处理能力。减少所述算法复杂性的一种方法是对每一颜色分

量分别进行混色，这种方法需要用笛卡尔乘积来填充查色表。请参考关于借助于较少的彩色调色板进行混色的最佳图象复制系统的欧洲专利 EP-A-0,639,920（代理人档案 PHN14,552）。

然而，允许所有透明度具有相同数目的彩色调色板入口不是使用查色表空间的好方法。通过观察全透明度级别（ $\alpha = 0$ ）就可以看出这一点，这时只有一个入口就够了。通常， α 值越低，获得同样颜色分辨率输出所需的输入颜色级别越少。由于 α 被用来作为每一图像颜色通道上的线性因子，所以每一图像颜色通道上的线性颜色因子的数目以及颜色级别的数目大致上应随着 α 值的增加而线性地增加。

这意味着每一透明度的颜色总数大致应随着 α 的立方增加。假设 $0 \leq a \leq A$ ，即，有 $A+1$ 种透明度。假设当 $a = 1$ 时的颜色总数是 C ，当假设所有 α 级别时，颜色总数大致为（ $A^2 * (A+1)^2 * C/4$ ）。再假设在查色表 CLUT 中颜色总数是 256，则有下列的结果：

A	=	1	2	3	4	5	6	7
C	=	256.0	28.4	7.11	2.56	1.138	.5805	.3265
$C * A^3$	=	256	227.5	192.0	163.8	142.2	125.4	112.0

下一步骤是找出对于每一透明度等级的颜色的适合的位置。假设我们对这个系统中简单的混色方法的应用感兴趣。将很方便地在查色表 CLUT 中找到每一透明度级别的笛卡尔乘积。这意味着，对于每一透明度级别，我们保存在查色表 CLUT 内的颜色数目应当是三个数的乘积，是每一颜色通道内颜色级别的数目。

作为例子，假设有 RGB 颜色通道。我们已经知道，蓝色通道需要的分辨率最低，红色通道需要的分辨率多些，绿色通道需要的分辨率更高。由于透明度级别影响所有三个颜色通道，所以在最高的 α 级别上，它需要的分辨率比起颜色通道的分辨率来是最高的。这种级别合理的分配应当是（R， G， B， a）=（5， 6， 4， 7）。应当指出，这对应于最接近上述 $A = 6$ 的数值。

从 $A=6$ 或 $A=7$ 开始，我们可以尝试找出在其它级别上用笛卡尔乘积较好地填充查色表 CLUT。需要考虑的一方面是在更低的分辨率上，级别的数目应当作某种程度的增加，因为计算结果已经进行了舍入，并且这些舍入误差对于低分辨率来说特别明显。例如，当 $a = 1$ 时，即在最低透明度级别上还可以显示一些图像，对于每一颜色通道最小级别数目是 2，结果查色表最小利用数为 8。这将导致以下的划分：

$A = 6$ （即，7 个透明度级别）

10	a	0	1	2	3	4	5	6
	$C \cdot a^3$	0	0.6	4.6	15.7	37.2	72.3	125.4
	R,G,B	1,1,1	2,2,2	2,3,2	3,3,2	3,4,3	4,5,3	5,6,4
	$R \cdot G \cdot B$	1	8	12	18	36	60	120

所使用的颜色的总数：255

15 带有每一透明度值的可变的查色表入口数的查色表的执行过程如下。

透明度值的数目：5（100% 75% 50% 25% 0%）

例子：查色表[14] = [0,127,255]，其中 14 是查色表 CLUT 的索引，0 是红颜色信号 R 的值，127 是绿颜色信号 G 的值，255 是蓝颜色信号 B 的值。

透明度 = 0% 的查色表 CLUT 入口

	$clut[0] = [0, 0, 0]$	$clut[27] = [63, 0, 0]$
	$clut[1] = [0, 0, 127]$	$clut[28] = [63, 0, 127]$
25	$clut[2] = [0, 0, 255]$	$clut[29] = [63, 0, 255]$
	$clut[3] = [0, 31, 0]$	$clut[30] = [63, 31, 0]$
	$clut[4] = [0, 31, 127]$	$clut[31] = [63, 31, 127]$
	$clut[5] = [0, 31, 255]$	$clut[32] = [63, 31, 255]$

	clut[6] = [0, 63, 0]	clut[33] = [63, 63, 0]
	clut[7] = [0, 63,127]	clut[34] = [63, 63,127]
	clut[8] = [0, 63,255]	clut[35] = [63, 63,255]
	clut[9] = [0, 95, 0]	clut[36] = [63, 95, 0]
	clut[10] = [0, 95,127]	clut[37] = [63, 95,127]
5	clut[11] = [0, 95,255]	clut[38] = [63, 95,255]
	clut[12] = [0,127, 0]	clut[39] = [63,127, 0]
	clut[13] = [0,127,127]	clut[40] = [63,127,127]
	clut[14] = [0,127,255]	clut[41] = [63,127,255]
	clut[15] = [0,159, 0]	clut[42] = [63,159, 0]
	clut[16] = [0,159,127]	clut[43] = [63,159,127]
10	clut[17] = [0,159,255]	clut[44] = [63,159,255]
	clut[18] = [0,191, 0]	clut[45] = [63,191, 0]
	clut[19] = [0,191,127]	clut[46] = [63,191,127]
	clut[20] = [0,191,255]	clut[47] = [63,191,255]
	clut[21] = [0,223, 0]	clut[48] = [63,223, 0]
	clut[22] = [0,223,127]	clut[49] = [63,223,127]
15	clut[23] = [0,223,255]	clut[50] = [63,223,255]
	clut[24] = [0,255, 0]	clut[51] = [63,255, 0]
	clut[25] = [0,255,127]	clut[52] = [63,255,127]
	clut[26] = [0,255,255]	clut[53] = [63,255,255]
	clut[54] = [127, 0, 0]	clut[88] = [191, 63,127]
	clut[55] = [127, 0,127]	clut[89] = [191, 63,255]
20	clut[56] = [127, 0,255]	clut[90] = [191, 95, 0]
	clut[57] = [127, 31, 0]	clut[91] = [191, 95,127]
	clut[58] = [127, 31,127]	clut[92] = [191, 95,255]
	clut[59] = [127, 31,255]	clut[93] = [191,127, 0]
	clut[60] = [127, 63, 0]	clut[94] = [191,127,127]
25	clut[61] = [127, 63,127]	clut[95] = [191,127,255]
	clut[62] = [127, 63,255]	clut[96] = [191,159, 0]
	clut[63] = [127, 95, 0]	clut[97] = [191,159,127]
	clut[64] = [127, 95,127]	clut[98] = [191,159,255]

	clut[65] = [127, 95,255]	clut[99] = [191,191, 0]
	clut[66] = [127,127, 0]	clut[100] = [191,191,127]
	clut[67] = [127,127,127]	clut[101] = [191,191,255]
	clut[68] = [127,127,255]	clut[102] = [191,223, 0]
	clut[69] = [127,159, 0]	clut[103] = [191,223,127]
5	clut[70] = [127,159,127]	clut[104] = [191,223,255]
	clut[71] = [127,159,255]	clut[105] = [191,255, 0]
	clut[72] = [127,191, 0]	clut[106] = [191,255,127]
	clut[73] = [127,191,127]	clut[107] = [191,255,255]
	clut[74] = [127,191,255]	clut[108] = [255, 0, 0]
	clut[75] = [127,223, 0]	clut[109] = [255, 0,127]
10	clut[76] = [127,223,127]	clut[110] = [255, 0,255]
	clut[77] = [127,223,255]	clut[111] = [255, 31, 0]
	clut[78] = [127,255, 0]	clut[112] = [255, 31,127]
	clut[79] = [127,255,127]	clut[113] = [255, 31,255]
	clut[80] = [127,255,255]	clut[114] = [255, 63, 0]
	clut[81] = [191, 0, 0]	clut[115] = [255, 63,127]
15	clut[82] = [191, 0,127]	clut[116] = [255, 63,255]
	clut[83] = [191, 0,255]	clut[117] = [255, 95, 0]
	clut[84] = [191, 31, 0]	clut[118] = [255, 95,127]
	clut[85] = [191, 31,127]	clut[119] = [255, 95,255]
	clut[86] = [191, 31,255]	clut[120] = [255,127, 0]
20	clut[87] = [191, 63, 0]	clut[121] = [255,127,127]
	clut[122] = [255,127,255]	clut[129] = [255,223, 0]
	clut[123] = [255,159, 0]	clut[130] = [255,223,127]
	clut[124] = [255,159,127]	clut[131] = [255,223,255]
	clut[125] = [255,159,255]	clut[132] = [255,255, 0]
	clut[126] = [255,191, 0]	clut[133] = [255,255,127]
25	clut[127] = [255,191,127]	clut[134] = [255,255,255]
	clut[128] = [255,191,255]	

透明度 =25%的查色表 CLUT 入口

	clut[135] = [0, 0, 0]	clut[158] = [85,170,255]
	clut[136] = [0, 0,255]	clut[159] = [85,212, 0]
	clut[137] = [0, 42, 0]	clut[160] = [85,212,255]
	clut[138] = [0, 42,255]	clut[161] = [85,255, 0]
5	clut[139] = [0, 85, 0]	clut[162] = [85,255,255]
	clut[140] = [0, 85,255]	clut[163] = [170, 0, 0]
	clut[141] = [0,127, 0]	clut[164] = [170, 0,255]
	clut[142] = [0,127,255]	clut[165] = [170, 42, 0]
	clut[143] = [0,170, 0]	clut[166] = [170, 42,255]
	clut[144] = [0,170,255]	clut[167] = [170, 85, 0]
10	clut[145] = [0,212, 0]	clut[168] = [170, 85,255]
	clut[146] = [0,212,255]	clut[169] = [170,127, 0]
	clut[147] = [0,255, 0]	clut[170] = [170,127,255]
	clut[148] = [0,255,255]	clut[171] = [170,170, 0]
	clut[149] = [85, 0, 0]	clut[172] = [170,170,255]
15	clut[150] = [85, 0,255]	clut[173] = [170,212, 0]
	clut[151] = [85, 42, 0]	clut[174] = [170,212,255]
	clut[152] = [85, 42,255]	clut[175] = [170,255, 0]
	clut[153] = [85, 85, 0]	clut[176] = [170,255,255]
	clut[154] = [85, 85,255]	clut[177] = [255, 0, 0]
	clut[155] = [85,127, 0]	clut[178] = [255, 0,255]
20	clut[156] = [85,127,255]	clut[179] = [255, 42, 0]
	clut[157] = [85,170, 0]	clut[180] = [255, 42,255]
	clut[181] = [255, 85, 0]	clut[186] = [255,170,255]
	clut[182] = [255, 85,255]	clut[187] = [255,212, 0]
	clut[183] = [255,127, 0]	clut[188] = [255,212,255]
	clut[184] = [255,127,255]	clut[189] = [255,255, 0]
25	clut[185] = [255,170, 0]	clut[190] = [255,255,255]

透明度 =50%的查色表 CLUT 入口

	clut[191] = [0, 0, 0]	clut[209] = [127,153, 0]
	clut[192] = [0, 0,255]	clut[210] = [127,153,255]
	clut[193] = [0, 51, 0]	clut[211] = [127,204, 0]
	clut[194] = [0, 51,255]	clut[212] = [127,204,255]
5	clut[195] = [0,102, 0]	clut[213] = [127,255, 0]
	clut[196] = [0,102,255]	clut[214] = [127,255,255]
	clut[197] = [0,153, 0]	clut[215] = [255, 0, 0]
	clut[198] = [0,153,255]	clut[216] = [255, 0,255]
	clut[199] = [0,204, 0]	clut[217] = [255, 51, 0]
	clut[200] = [0,204,255]	clut[218] = [255, 51,255]
10	clut[201] = [0,255, 0]	clut[219] = [255,102, 0]
	clut[202] = [0,255,255]	clut[220] = [255,102,255]
	clut[203] = [127, 0, 0]	clut[221] = [255,153, 0]
	clut[204] = [127, 0,255]	clut[222] = [255,153,255]
	clut[205] = [127, 51, 0]	clut[223] = [255,204, 0]
	clut[206] = [127, 51,255]	clut[224] = [255,204,255]
15	clut[207] = [127,102, 0]	clut[225] = [255,255, 0]
	clut[208] = [127,102,255]	clut[226] = [255,255,255]

透明度 =75%的查色表 CLUT 入口

	clut[227] = [0, 0, 0]	clut[230] = [0, 85,255]
	clut[228] = [0, 0,255]	clut[231] = [0,170, 0]
20	clut[229] = [0, 85, 0]	clut[232] = [0,170,255]
	clut[233] = [0,255, 0]	clut[238] = [255, 85,255]
	clut[234] = [0,255,255]	clut[239] = [255,170, 0]
	clut[235] = [255, 0, 0]	clut[240] = [255,170,255]
	clut[236] = [255, 0,255]	clut[241] = [255,255, 0]
25	clut[237] = [255, 85, 0]	clut[242] = [255,255,255]

透明度 =100%的查色表 CLUT 入口

clut[243] = [0,0,0]

这些 R,G,B 值无关紧要。

5 本发明的第一个方面可以总结如下。本发明涉及查色表 (CLUT) 在显示中的使用, 其中, 对于若干透明度级别之一, 将帧缓冲器的内容叠加在视频图像上。更具体地说, 描述了填充查色表 CLUT 的最佳方法。最好的方法是: 对于给定的透明度级别, 可能的颜色数目取决于混合比 α : 在 100%的透明度, 单一值足够了, 而在 0%上所需要的颜色数目最大。使用三次方的关系是有益的。颜色数目等于红色值数目 RR 乘以绿色值数目 GG 再乘以蓝色值数目 BB 的乘积, 这里最好 $GG \geq RR \geq BB$ 。

10 应当注意, 上述举例说明的实施例并不是对本发明的限制。对于本专业的技术人员来说, 他们可以设计许多其它的实施例, 而这些都是不会超出权利要求书的范围。在权利要求书中, 任何括弧内的参考符号都不应作为对权利要求书的限制。可以通过含有若干不同元件的硬件以及适当编程的计算机来实现本发明。

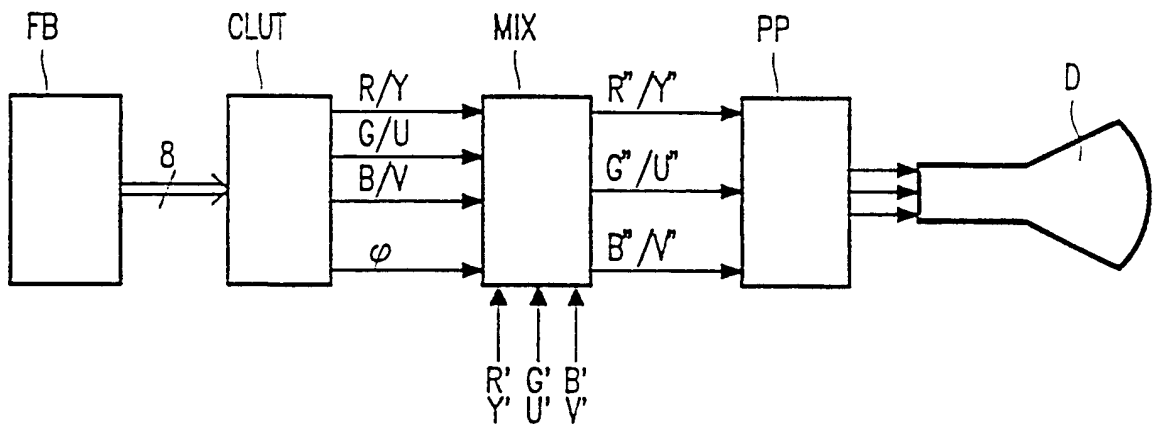


图 1