

文章编号: 1001 - 2486(2003) 06 - 0021 - 05

虚拟演播室系统中的无限蓝箱技术*

鲁敏, 匡纲要, 郁文贤

(国防科技大学电子科学与工程学院, 湖南长沙 410073)

摘要: 虚拟演播室是虚拟现实技术在电视制作领域中的一项崭新应用。无限蓝箱技术通过外部生成的键信号, 将前景视频图像中非蓝箱内的景物滤除, 使摄像机的运动可以不受实际蓝箱尺寸的限制。针对任意形状的蓝箱结构和摄像机的任意自由度运动, 提出了虚拟演播室系统中通用的无限蓝箱算法, 该算法根据蓝箱实际的透视投影变换过程, 精确提取蓝箱区域, 生成色键信号。实验结果证明, 该算法生成的色键区域精确、计算简单、通用性强, 很好地满足了虚拟演播室系统的要求。

关键词: 虚拟演播室; 无限蓝箱; 透视投影; 裁剪; 多边形填充; 色键

中图分类号: TN94 文献标识码: A

Unlimited Blue-box Technique in Virtual Studio

LU Min, KUANG Gang-yao, YU Wen-xian

(College of Electronic Science and Engineering, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: Virtual studio is a new application of virtual reality in TV program production. Unlimited blue-box technique is the way to cover the scenery out of blue-box with an external chroma-key signal. It extends the movement of the camera in virtual studio. A universal unlimited blue-box technique is proposed based on the perspective projection. It can be applied to all kinds of blue-box structures and varied camera movements. The experiment result demonstrates that this technique is accurate and robust, and is sufficient to the requirements of virtual studio.

Key words: virtual studio; unlimited blue box; perspective projection; clipping; polygon fill; chroma-key

虚拟演播室(Virtual Studio)是虚拟现实技术在电视制作领域中的一项崭新应用,是视频技术与计算机技术相结合的产物,它可以广泛应用于制作新闻、访谈、教育、专题报道、电视广告、音乐电视、天气预报等节目。虚拟演播室^[1,2,4,6]是在传统色键抠像技术的基础上,克服其背景不能移动、令节目的真实性和运动效果难以体现的局限性,充分应用了计算机三维图形和虚拟现实技术而产生的。它可以自动跟踪摄像机的位置、高度、摇移、俯仰以及镜头的聚焦和变焦等参数变化,计算机根据这些参数的变化,实时地改变虚拟三维场景,使虚拟背景的变化与摄像机的运动保持同步和一致,因此保证了前景和背景的正确透视关系,前景中的主持人看起来完全沉浸于计算机所产生的虚拟三维场景中,具有逼真的电视演播室效果。

虚拟演播室中演员的表演都在蓝箱中进行,色键(Chroma-key)合成系统用背景图像的相应部分作为背景填充蓝色部分,形成一幅天衣无缝的合成画面。但在实际演播中,由于实际场地的限制,蓝箱的天花板可能较低或对于宽角度拍摄来说太窄,当摄像机进行推拉摇移等运动时,拍摄的图像中会有蓝箱以外的景物,图像输入到视频合成系统与虚拟场景抠像合成时,这些景物也会出现在输出的视频中,影响输出的最终效果。无限蓝箱(Unlimited Blue-box)技术通过外部生成的键信号,将前景视频图像中非蓝箱内的景物滤除,满足虚拟演播的要求。一般实现方法如下:首先,对实际蓝箱进行建模,并建立统一的世界坐标系;然后根据实时获取的摄像机运动参数,实时生成前景垃圾键信号(Garbage Key);将此键信号输入到视频合成系统,与前景、背景一起合成时,遮挡前景中不需要的区域。

传统无限蓝箱技术^[5]考虑蓝箱结构为长方体,限定摄像机为固定机位,并只有平摇和俯仰两个转

* 收稿日期: 2003 - 05 - 20

作者简介: 鲁敏(1977-),男,博士生。

动自由度,生成键信号采用包围盒的方式,丢弃了一部分蓝箱区域,并且不能满足摄像机任意自由度和任意结构蓝箱模型的要求。本文针对任意形状的蓝箱结构和摄像机的任意自由度运动,提出了虚拟演播室系统中通用的无限蓝箱算法,该算法根据蓝箱实际的透视投影变换过程,精确提取蓝箱区域,生成色键信号。

1 无限蓝箱算法实现

算法基本思路是将蓝箱模型简化为有向多边形面片模型,根据透视投影变换的原理确定图像空间中蓝箱对应的多边形区域,对多边形区域进行视口裁剪,获得实际所能观察到的蓝箱区域,最后进行边界过渡的区域填充,生成最终的色键信号输出。

1.1 蓝箱区域透视投影变换

建立如图 1(a) 所示的世界坐标系,蓝箱由有向多边形构成,其多边形法向指向涂蓝的一面。因为一般蓝箱结构中蓝色区域为连续区域,为简化算法计算过程,仅取蓝箱外边界顶点表示蓝箱模型,顶点序列为面对蓝箱顺时针方向选取(此方向对下一步变换有作用):

$$V_i(X_w^i, Y_w^i, Z_w^i) \quad i = 1, \dots, N$$

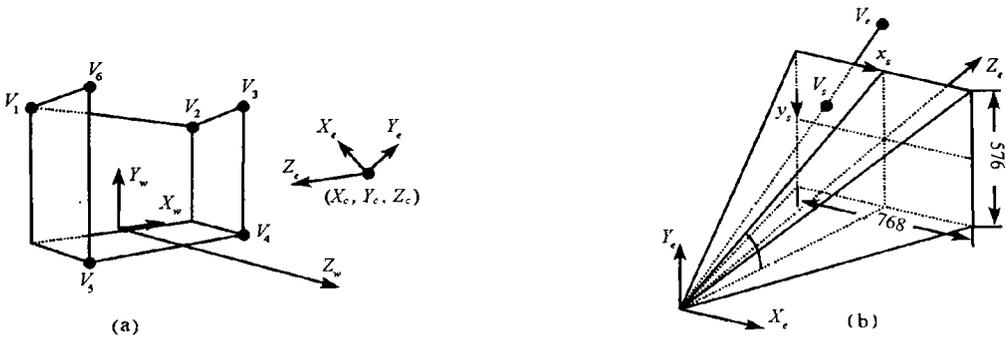


图 1 蓝箱模型及世界坐标系和观察坐标系关系

Fig. 1 Blue box model, relation about world coordinates and observer coordinates

摄像机空间位置 (X_c, Y_c, Z_c) 、方向 (φ, θ, ϕ) 以及垂直视角 γ 已知,其参数由摄像机跟踪系统获取。根据透视投影变换原理^[7]计算当前蓝箱边界顶点的世界坐标到摄像机图像坐标的映射。引入观察坐标系,首先计算蓝箱边界顶点从世界坐标系到观察坐标系的变换:

$$[X_e^i, Y_e^i, Z_e^i]^T = S_z \cdot R_z \cdot R_x \cdot R_y \cdot T \cdot [X_w^i, Y_w^i, Z_w^i]^T \quad i = 1, \dots, N$$

其中, T 为视点平移矩阵, R_y, R_x, R_z 分别是坐标系绕 Y 轴旋转 θ 角、绕 X 轴旋转 φ 角和绕 Z 轴旋转 ϕ 角的旋转矩阵, S_z 是 Z 轴反向的尺度变换矩阵。

然后计算观察坐标系到成像平面的变换,如图 1(b),设摄像机成像平面分辨率为 768×576 (PAL 制视频大小),图像坐标系原点落在图像左上角, x 轴向右, y 轴向下,根据摄像机当前垂直视角为 γ ,有:

$$\begin{cases} x_s^i = 768/2 + X_e^i \cdot \alpha \\ y_s^i = 576/2 - Y_e^i \cdot \alpha \\ \alpha = 576/[2 \cdot Z_e^i \cdot \tan(\gamma/2)] \end{cases} \quad i = 1, \dots, N$$

1.2 有效蓝箱区域裁剪算法

经过透视投影变换得到多边形顶点序列 $V_i (i = 1, \dots, N)$,注意到该序列顶点构成的多边形各边有可能相交,将多边形分割为多部分,并且某些分割区域为蓝箱背面,并不属于我们所要获得的蓝箱区域(如图 2)。

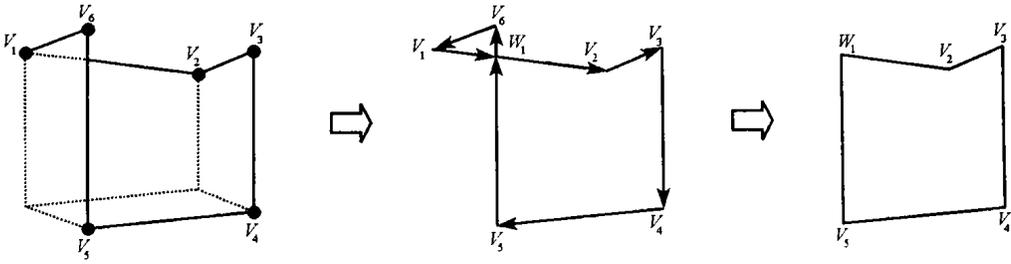


图 2 确定蓝箱实际区域
Fig. 2 Decide the region of blue-box

必须重新计算多边形蓝箱区域。对一般的蓝箱结构, 注意到一个重要性质, 蓝箱区域与非蓝箱区域构成的多边形环方向相反, 可见, 蓝箱区域多边形环方向与我们定义的蓝箱边界构成的环同向, 而非蓝箱区域为反向。于是提出了基于分割区域环方向的方法, 算法思路为: 首先计算顶点序列构成的各边之间交点, 根据交点和原顶点分割多边形为独立的简单多边形(边不交叉)区域, 最后根据独立区域简单多边形的方向取舍该区域。具体算法为:

Step1: 计算多边形非相邻边之间的交点。

多边形所有可能有交点的两条边定义如下:

$$\begin{cases} V_1 V_2 \rightarrow V_i V_{i+1} & i = 3, \dots, N - 1 \\ V_i V_{i+1} \rightarrow V_j V_{j+1} & i = 2, \dots, N - 3 \quad j = i + 2, \dots, N - 1 \quad \text{if } N \geq 5 \\ V_i V_{i+1} \rightarrow V_N V_1 & i = 2, \dots, N - 2 \end{cases}$$

设两条线段的端点分别为 P_1, P_2, Q_1, Q_2 , 则线段用向量形式表示为:

$$\begin{cases} P(t) = P_1 + (P_2 - P_1)t, & t \in [0, 1] \\ Q(s) = Q_1 + (Q_2 - Q_1)s, & s \in [0, 1] \end{cases}$$

构造方程 $P(t) = Q(s)$, 求解, 如果有

$$t \in [0, 1] \quad \text{且} \quad s \in [0, 1]$$

则存在交点, 求出交点记为 W_i ; 否则, 两线段不相交。

Step2: 根据交点分割多边形。

将所有交点按从 $V_1 \rightarrow V_2 \rightarrow \dots \rightarrow V_N$ 的遍历顺序插入顶点序列中, 交点在序列中成对出现, 然后根据每对交点分割多边形为简单多边形。例如图中多边形插入交点后形成顶点序列:

$$V_1 \rightarrow W_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5 \rightarrow W_1 \rightarrow V_6$$

根据交点分割多边形:

$$W_1 \rightarrow V_2 \rightarrow V_3 \rightarrow V_4 \rightarrow V_5, \quad W_1 \rightarrow V_6 \rightarrow V_1$$

Step3: 判断简单多边形的环方向, 决定多边形区域的取舍。

对得到的所有多边形根据顶点序列的顺序判断多边形形成的环方向, 与之前假定蓝箱边界构成环方向相反的区域舍去。

算法中, 多边形环方向判别准则如下。

设简单多边形用有序顶点序列 $V_i (i = 1, \dots, M)$ 表示, 将顶点 V_i 按 x 轴坐标大小排序, 取 x 轴坐标最小的顶点 V_n , 并根据有序顶点序列, 相应得到上一顶点 V_{n-1} 和下一顶点 V_{n+1} (如图 3)。若顶点 V_{n+1} 在由 V_{n-1}, V_n 两顶点确定的直线上方, 则多边形为顺时针方向; 若在下方, 则多边形为逆时针方向。具体计算式推导可得:

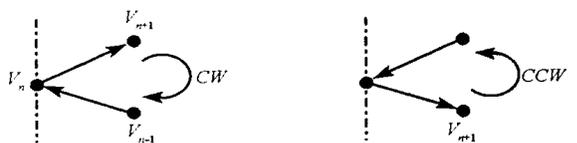


图 3 简单多边形环方向判断示意图
Fig. 3 Judge the loop direction of simple polygon

$$(x_n - x_{n-1})(y_{n+1} - y_n) - (y_n - y_{n-1})(x_{n+1} - x_n) \begin{cases} > 0 & \text{则 } CCW \\ < 0 & \text{则 } CW \end{cases}$$

其中, CW 表示顺时针方向, CCW 表示逆时针方向。

经透视变换后所得的多边形经过视口裁剪, 最终得到我们所能观察到的蓝箱区域。实际算法中采用逐次多边形裁剪算法, 即 Sutherland-Hodgman 算法^[3]。该算法的基本思想: 通过对单一边的简单裁剪, 实现对多边形的裁剪, 即原多边形和每次裁剪所生成的多边形将逐次对裁剪窗口的每一边进行裁剪。算法输出的是一个多边形的顶点序列, 所有顶点均位于视口的可见区域内。算法具有思路清晰、方法简单和速度快的特点。

1.3 有效蓝箱区域填充算法

有效蓝箱区域多边形经过区域填充可以作为垃圾键信号输出, 然而在实际应用中, 这种方式得到的键信号边缘太硬, 缺少平滑过渡, 合成后的视频有明显的硬边现象, 所以填充后的区域还要进行边界过渡处理。边界过渡处理的目的是在多边形的边界形成几个像素宽的边缘过渡带。其中多边形的边界分为两类: 第一类边界点落在视口的边界上, 边界点有一侧是不可见区域, 不需处理; 另一类为视口内的边界点, 必须模糊处理。我们提出改进的边标志区域填充算法如下:

设多边形内填充颜色为 $White$, 多边形区域外填充颜色为 $Black$, 过渡带分为 N 个等级, 由外至内对应的填充颜色分别为 $Gray[n]$ ($n = 1, \dots, N$)。改进算法分为以下四个步骤:

第一步, 对多边形的每条边进行直线扫描转换时, 对多边形边界所经过的像素给出边标志 $IsBorder = TRUE$, 并判断各线段是否落在视口的边界上, 若不是, 该边界需要过渡处理, 将该线段所经过的像素加入第一级过渡带数组 $ArrayBand[1]$ 。

第二步, 区域填充。对每条与多边形相交的扫描线, 依从左到右的顺序, 逐个访问该扫描线上的像素。使用一个布尔量 $IsInside$ 来指示当前像素的状态, 若像素在多边形内, 则 $IsInside = TRUE$; 否则 $IsInside = FALSE$ 。设定初始值 $IsInside = FALSE$, 每当当前访问像素的边标志 $IsBorder = TRUE$ 时, 填充该像素后, $IsInside$ 取反。对边标志 $IsBorder = FALSE$ 的像素, $IsInside$ 不变, 并且当 $IsInside = TRUE$ 时, 填充该像素。

第三步, 边界过渡处理。取数组 $ArrayBand[n]$ 中的像素, 对该像素进行过渡处理。该像素填充当前过渡带颜色 $Gray[n]$, 若 $n \neq N$, 分别考察上下左右四个邻域像素, 若邻域像素颜色等于 $White$, 则该邻域像素为下一级的过渡带像素, 像素加入下一级过渡带数组 $ArrayBand[n+1]$ 。

第四步, 如果 $n = N$, 算法退出; 否则, $n = n + 1$, 重复第三步。

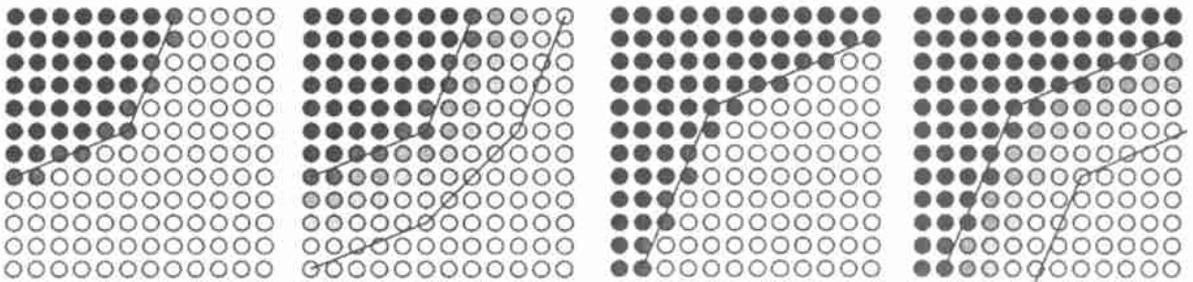


图 4 过渡带填充算法示意图
Fig. 4 Example of transition filling

图 4 中, 左边为边标志算法填充的结果, 对应的右边为改进后加入过渡带的算法填充的结果(过渡带级数 $N = 5$)。

2 实验结果

根据上述算法, 实际无限蓝箱算法流程如图 5, 注意图中虚线的指向, 因为对一部分蓝箱结构或当限定摄像机运动范围时, 蓝箱区域不存在遮挡物体, 可以无需做有效蓝箱区域的判别操作。

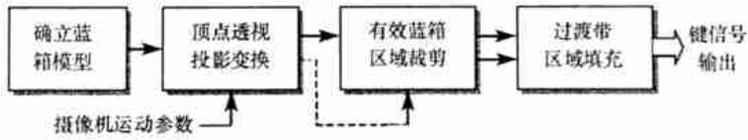


图5 实际的无限蓝箱算法流程

Fig.5 Unlimited blue-box technique flow chart

在计算机上软件编程实现算法。软件根据预先建立的蓝箱模型和实时获取的摄像机运动参数,实时生成8位键信号输出,用于虚拟演播室系统的视频合成。图6是不同的无限蓝箱键信号和视频合成结果,其中A图是摄像机拍摄的前景蓝箱图像;B图是传统无限蓝箱技术生成的键信号,与A图比较可以看出,键信号丢弃了很大部分的有用蓝箱区域,不能满足实际演播的需求;C图是未采用过渡填充算法的键信号,E图是相对应的视频合成结果,图中3个箭头标示的区域都可以看到明显的硬边现象,同样不能满足虚拟演播室系统高质量演播的需求;D图是我们提出的无限蓝箱算法输出的键信号,F图是相对应的视频合成图像,可以看出,所生成的键信号可以精确地与实际蓝箱区域匹配,将摄像机拍摄的前景无用信号全部遮蔽,并且由于过渡带的引入,合成后的图像自然协调,过渡平滑。

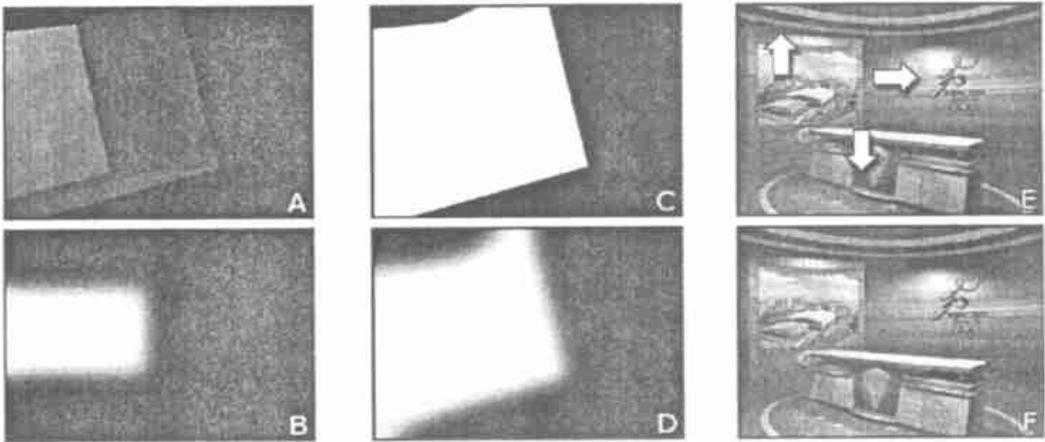


图6 虚拟演播室系统无限蓝箱技术结果比较

Fig.6 Results of unlimited blue-box technique in virtual studio

3 结论

实验结果表明,本文提出的虚拟演播室无限蓝箱算法突破了虚拟演播室系统中实际蓝箱尺寸的限制,极大地扩展了摄像机的运动范围。算法在保证精确保留图像蓝箱区域的前提下,生成的色键细腻自然,平滑过渡,满足了高质量虚拟演播室系统演播的要求;算法流程清晰、计算简单,给出了相应的快速算法实现,保证了系统的实时性要求;同时,算法通用性强,可适用于各种不同的蓝箱结构,并可适应摄像机任意自由度的运动,满足不同系统结构的虚拟演播室的系统需求。

参考文献:

- [1] Gibbs Simon, et al. Virtual Studio: An Overview[J]. IEEE Multimedia, 1998: 18- 35.
- [2] Blond L, et al. A Virtual Studio for Live Broadcasting: The Mona Lisa Project [J]. IEEE Multimedia, 1996, 3(2): 18- 29.
- [3] Sutherland I E, Hodgman G W. Reentrant Polygon Clipping[J]. CACM, 2002, 17: 32- 42.
- [4] 奇越,等. 虚拟演播室: 结构及关键技术[J]. 中国图象图形学报, 2000, 5(A)(6): 457- 460.
- [5] 庄宁,鲁敏,等. 虚拟演播室中无限蓝箱功能的实现[J]. 电视技术, 2001, 12: 55- 56, 59.
- [6] 虚拟演播室系统技术报告[R]. 国防科技大学, 2002.
- [7] 孙家广,等. 计算机图形学[M]. 北京: 清华大学出版社, 1998.