

# 基于继承性的圆形窗口多边形裁剪新算法

邢军伟 陆国栋 谭建荣

(浙江大学 CAD&CG 国家重点实验室, 杭州 310027)

**摘要** 在深入挖掘多边形的边与圆形窗口的几何特性以及二者相对位置的基础上,提出了可继承的正方形编码以及可继承的广义距离两种新技术以及裁剪新算法.可继承的正方形编码技术包括完全可继承编码和部分可继承编码,其可以快速舍弃绝大部分完全位于圆形窗口外部的多边形的边,而可继承的广义距离技术可以快速获取所有完全位于圆形窗口内部的边,并可进一步用于交点的快速求取.实验结果和算法比较表明,本算法具有较高的运算效率,稳定性良好.

**关键词** 计算机图形处理(520·6030) 圆形窗口 多边形 裁剪 继承性 编码 广义距离  
**中图法分类号**: TP391.41 **文献标识码**: A **文章编号**: 1006-8961(2003)10-1227-06

## Polygon Clipping Against Circular Window Based on the Inheriting Character

XING Jun-wei, LU Guo-dong, TAN Jian-rong

(State Key Lab. of CAD&CG at Zhejiang University, Hangzhou 310027)

**Abstract** This paper advanced two new techniques of hereditary square encoding and hereditary generalized distance based on the thorough analysis of relative positions of polygon edges with circular window, as a result, a new clipping algorithm is achieved. The hereditary square encoding technique include entirely hereditary and partially hereditary encoding, which can fast reject most of the polygon edges that totally out of the circular windows. The hereditary generalized distance technique can fast identify totally effective polygon edges and remain all polygon edges that totally inside of the circular windows. Hereditary generalized distance can also identify the relative position relations of the remained edges with the circular window. It can also be applied on fast seek the points of intersection. Experiments and the compare of algorithms show that the new algorithm has high efficiency and good stability.

**Keywords** Computer graphics, Circular window, Polygon, Clipping, Inheriting character, Coding, Universe distance

## 0 引言

裁剪是从数据集中提取信息的过程,它是计算机图形学中诸多重要问题的基础<sup>[1]</sup>.裁剪包含被裁剪元素和裁剪窗口两个要素,被裁剪元素可以分为线段、多边形、圆、一般平面图形、任意曲线和字符等;裁剪窗口,可以分为规则的矩形窗口、任意凸区域窗口、任意凹区域窗口、圆形窗口、广义的非封闭

窗口等.两个要素的不同组合和不同的处理方法形成了众多的裁剪算法,其中线段被规则矩形窗口裁剪是最基本的裁剪问题,矩形窗口的多边形裁剪已经有了许多有效的算法<sup>[1~8]</sup>,如著名的 Sutherland-Hodgman 算法和 Weiler-Atherton 算法等.

相对而言,圆形窗口多边形裁剪的研究较少.与其他裁剪问题相类似,圆形窗口多边形裁剪的关键是如何快速得到裁剪后的输出顶点集.文献[4]、[5]这两种算法在获取输出顶点集时都是通过建立

基金项目:国家自然科学基金(50106014);创新群体科学研究基金(60021201);教育部高等学校博士学科点专项科研基金(20020335093)

收稿日期:2002-08-13;改回日期:2003-04-18

多边形的边与圆形窗口相交的数学模型,求得参数  $t$  的判别式,进而通过判别式确定多边形的边与圆形窗口是否相交,当存在交点时,由参数值  $t$  求出交点,并确定交点的性质.由于圆是一种曲线,因此这种计算方法涉及非线性方程求解.当大部分的多边形的边与圆形窗口相交时,其效率较高,但当大部分多边形的边与圆形窗口不相交时,效率会受到影响.

因此,提高圆形窗口多边形裁剪效率的关键在于多边形的边与圆形窗口不相交时的处理策略.借鉴成功的矩形窗口编码技术,引入圆形窗口外切正方形编码技术,提出了旋转外切正方形编码新方法,从而无需求解一元二次方程的根即可判断出大部分完全位于圆形窗口外部的多边形的边;进一步地使用广义距离新概念,可以快速获取所有完全位于圆形窗口内部的多边形的边并能判别多边形的边与圆形窗口的相对位置关系.更有意义的是,多边形顶点的编码以及广义距离均具有可继承性,从而可以进一步提高多边形的边与圆形窗口不相交时的处理速度.

### 1 可继承的编码技术

为了便于讨论,规定:圆形窗口的圆心在坐标原点(可经坐标平移得到);多边形  $P=(P_0, P_1, \dots, P_{n-1})$  的顶点用笛卡尔坐标表示,并按顺时针方向排列.同时,将多边形的边分为如下 3 类:

- (1)完全无效边 整条边完全位于圆形窗口的外部从而不参与形成裁剪后区域的边.
- (2)完全有效边 整条边完全位于圆形窗口的内部从而整条边都参与形成裁剪后区域的边.
- (3)部分有效边 整条边的一部分位于圆形窗口内部从而部分参与形成裁剪后区域的边.

#### 1.1 完全可继承的编码

圆形裁剪窗口平行于坐标轴的切线,共有 4 条.这 4 条切线相交成一个正方形,从而把平面划分为 9 个区域,圆形窗口位于正方形所围成的区域内部.对这 9 个区域分别赋予一个编码,结果如图 1 所示.

将多边形边的两个端点按它所在的区域进行编码,通过对端点编码的位运算就可以舍去完全在外切正方形窗口上的边及其延长线同一侧的多边形的边(显然属于完全无效边).判断条件为

$$\text{code}(P_1) \& \text{code}(P_2) \neq 0$$

其中,  $P_1$  和  $P_2$  为多边形边的两个端点,  $\&$  是各位的逻辑与运算.如图 1 中多边形的边  $g$ ,  $\text{code}(P_1) \&$

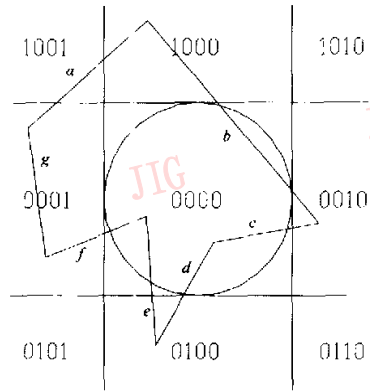


图 1 完全可继承的编码

$\text{code}(P_2) = (0001) \& (0001) = 0001 \neq 0$ , 舍弃边  $g$ .

由于多边形的边是首尾顺次相连的,因此两条相邻边的共同端点的编码只需计算一次.

**定义 1** 对于具有共同端点的多边形的两条边而言,对前序的一条边计算此端点得到的编码可直接被后序的边所利用,这种性质称为编码的可继承性.

如图 1 中的边  $g$  和  $a$ ,它们具有一个相同的端点,若对  $g$  的处理在前,则处理  $g$  时已经计算了此端点的编码,那么在接着处理边  $a$  时,只需计算  $a$  和  $b$  的共同端点的编码,即可对边  $a$  进行处理.如果多边形具有  $n$  条边,则只需计算  $n$  个编码.

#### 1.2 部分可继承的编码

当外切正方形沿圆形窗口的圆心旋转时,它与圆形窗口的相切关系显然保持不变,只是多边形的各条边与正方形的相对位置关系发生了变化,即旋转前不在正方形上的某条边及其延长线的同一侧的多边形的边,在旋转后有可能位于同一侧.若对旋转外切正方形划分平面所得的区域再次编码,就可以再次快速地舍弃多边形另一部分的边,判断条件和常规外切正方形编码中的条件相同.

从理论上讲,当正方形沿圆心旋转一周后,圆形窗口外的所有边均可以去除掉,但过多的旋转窗口显然是低效的.因此本文选择  $45^\circ$  角的一次旋转,旋转后的编码结果如图 2 所示.

此时,原来需要用一元二次方程求根法判别的多边形的边  $a$ ,现在可以通过对其端点编码进行简单的位运算而舍弃.

编码算法如下( $x$  和  $y$  是多边形的顶点坐标,  $r$  是圆形窗口的半径):

```
#define LEFT 1      #define RIGHT 2
```

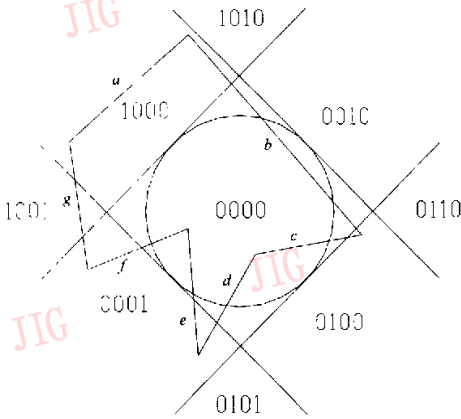


图 2 部分可继承的编码

```
#define BOTTOM 4      #define TOP 8      code=0
if(x+y)<-1.415r     code=code|LEFT
if(x+y)>1.415r     code=code|RIGHT
if(y-x)<-1.415r     code=code|BOTTOM
if(y-x)>1.415r     code=code|TOP
return code
```

此算法中用 1.415 取代理论上的  $\sqrt{2}$ , 即将窗口稍微放大, 这样显然并不影响裁剪结果.

由于旋转外切正方形编码只有在常规外切正方形编码无效的情况下才进行, 相邻的两条边并不是在任何情况下都具有这种编码, 因此这种编码是部分可继承的. 如果采用这种编码对一条边进行处理, 就首先要判别这条边的前序端点是否已经获得了这种编码, 如果已经获得编码, 只要继承这一编码即可, 无需重新计算. 在用程序实现时, 对计算过旋转外切正方形编码的端点设置一个标志即可.

### 2 可继承的广义距离技术

当编码技术失效时, 可利用可继承的广义距离技术继续对多边形的边进行处理.

#### 2.1 判断完全有效边

可继承的编码技术能有效判断大部分完全无效边, 而可继承的广义距离技术可以判断完全有效边, 还可用于确定多边形的形与圆形窗口的相对位置关系以及快速求取圆形窗口与部分有效边的交点.

定义 2 对于由二维坐标系所决定的平面上的任意一点  $P_i$  以及某一圆周而言, 定义点  $P_i$  到此圆圆心距离的平方为其广义距离, 记为  $d_i$ .

定义 3 对于具有相同端点的多边形的两条边而言, 对前序边计算此端点所得到的广义距离可直接为后序边所使用, 这种性质称为广义距离的可继承性.

显然, 与旋转外切正方形编码类似, 广义距离也是部分可继承的. 当与圆形窗口相交或位于圆形窗口内部的边占多数时, 继承性可以有效地减少运算次数. 用程序实现时, 对计算过广义距离的端点设定一个标志即可.

为了正确地形成裁剪后区域, 还必须区分圆形窗口与多边形交点的不同类型. 设  $P_i, P_j$  为平面上任意两点, 用  $P_i P_j$  表示具有从  $P_i$  到  $P_j$  方向的有向线段. 称  $P_{i-1} P_i (P_{-1} = P_{n-1})$  为顶点  $P_i$  的入边, 而  $P_i P_{i+1} (P_n = P_0)$  为顶点  $P_i$  的出边. 很明显, 对多边形的任一顶点, 均存在一条入边和一条出边. 假设多边形的顶点  $P_i$  在圆形窗口之外 (在圆形窗口边线上的顶点被视为在圆形窗口之外), 且入、出两边至少有一条边与圆形窗相交 (与圆形窗口相切的边被视为在圆形窗口外). 若其入边  $P_{i-1} P_i$  与圆形窗相交, 则称距  $P_i$  最近的交点为其入边交点, 记为  $P_{i\lambda}$ ; 若其出边  $P_i P_{i+1}$  与相交, 则称距  $P_i$  最近的交点为  $P_{i\mu}$ ; 若  $P_{i\lambda}, P_{i\mu}$  均存在, 则规定  $P_{i\lambda}$  排在  $P_{i\mu}$  之前.

多边形的边与圆形窗口的相对位置较为复杂, 根据多边形的顶点与圆形窗口的相对位置关系将其分为 4 类, 如图 3 所示.

假设多边形某条边的两个端点分别为  $P_i$  和  $P_{i+1}$ , 计算它们的广义距离  $d_i$  和  $d_{i+1}$ , 如图 3(a) 所

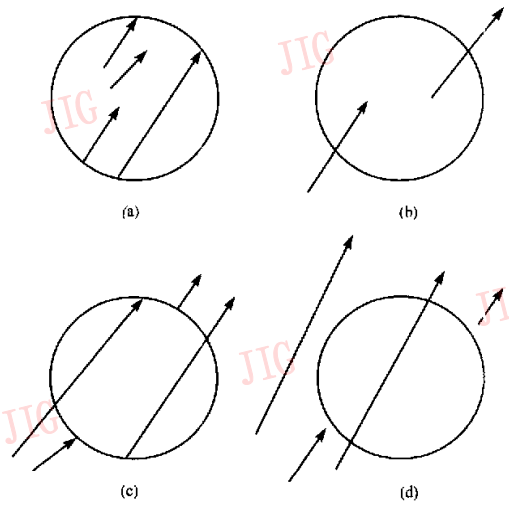


图 3 多边形的边与圆形窗口的相对位置

示,  $R=r^2$ , 如果  $\max\{d_i, d_{i+1}\} \leq R$ , 则这条边属于完全有效边, 把两个端点顺次记入输出顶点集.

### 2.2 判断剩余边、圆位置关系及求取有效交点

经过上述处理后, 剩余多边形的边与圆形窗口相交的概率将远远大于不相交的概率, 故按照一元二次方程求交点的模型进行处理.

首先求出参数  $t_1$  和  $t_2$ :

设圆的参数方程

$$\begin{aligned} x &= r \cos \theta \\ y &= r \sin \theta \end{aligned} \quad 0 \leq \theta \leq 2\pi$$

设以笛卡尔坐标表示的端点  $P_i$  和  $P_{i+1}$  所决定的多边形的边所在直线的参数方程为

$$\begin{aligned} x &= x_i + (x_{i+1} - x_i)t \\ y &= y_i + (y_{i+1} - y_i)t \end{aligned} \quad 0 \leq t \leq 1$$

联立上述方程

$$\begin{aligned} r \cos \theta &= x_i + (x_{i+1} - x_i)t \\ r \sin \theta &= y_i + (y_{i+1} - y_i)t \end{aligned}$$

消去  $\theta$ , 得  $A_i t^2 + 2B_i t + C_i = 0$ , 记

$$M_i = x_i x_{i+1} + y_i y_{i+1}, \text{ 则}$$

$$A_i = (x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2 = d_{i+1} + d_i - 2M_i$$

$$B_i = x_i(x_{i+1} - x_i) + y_i(y_{i+1} - y_i) = M_i - d_i$$

$$C_i = x_i^2 + y_i^2 - r^2 = d_i - R$$

令  $\Delta_i = \sqrt{B_i^2 - A_i C_i}$ , 从而可求得

$$t_1 = (-B_i - \Delta_i) / A_i, t_2 = (-B_i + \Delta_i) / A_i$$

然后进行如下判断:

(1) 如图 3(b) 所示, 若  $\min\{d_i, d_{i+1}\} < R$ , 而且  $\max\{d_i, d_{i+1}\} > R$ , 则这条边属于部分有效边, 计算有效交点, 把圆内端点和有效交点按顺序记入输出顶点集.

(2) 如图 3(c) 所示, 若  $\min\{d_i, d_{i+1}\} = R$ , 而且  $\max\{d_i, d_{i+1}\} > R$ , 则如果  $t_1 = 0$  且  $0 < t_2 < 1$ , 将第 1 个端点与第 2 个交点记入输出顶点集; 如果  $0 < t_1 < 1$  且  $t_2 = 1$ , 将第 1 个交点与第 2 个端点按顺序记入输出顶点集.

(3) 如图 3(d) 所示, 两个端点都在圆外. 如果  $0 < t_1 < t_2 < 1$ , 将两个交点按顺序记入输出顶点集.

常规求交法在求取参数  $t_1$  和  $t_2$  时是针对单条边来进行的, 多边形的每个顶点被两条边所共有, 故每个顶点的信息均被使用了两次, 存在重复运算. 上述新的求交方法求取参数  $t_1$  和  $t_2$  时充分地利用了已求出的广义距离的信息, 而且还间接利用了广义距离的可继承性, 减少了顶点的使用次数, 故而提高

了交点求取的效率.

### 3 算法实现与效果分析

多边形是由各条边顺次连接而成的, 多边形的裁剪不等同于各边的裁剪, 尽管前面的分析针对的是多边形的各条边. 可继承的编码和广义距离处理的直接结果是输出顶点集, 对输出顶点集继续进行进一步处理即可得到最终的裁剪结果.

如果输出顶点集为空, 需继续判断圆形窗口是完全位于多边形的外部还是内部. 显然, 前一种情况的裁剪结果为空集, 而后一种情况的裁剪结果为圆形窗口所包含的圆形区域(包括圆的边界).

如果输出顶点集不为空, 且没有圆形窗口与多边形的交点参与构成, 则多边形完全在圆形窗口内部, 裁剪结果为多边形所包围的区域(包括多边形的边界). 如果输出顶点集中有圆形窗口与多边形的交点参与构成, 则将多边形与圆形窗的交点子集相对于圆心的位置分为左右两部分, 左侧部分按  $y$  值递增排序, 右侧部分按  $y$  值递减排序, 重新组合成新的交点子集; 接着, 将输出顶点集中的顶点用直线连接闭合, 遇到出点, 输出顶点集不与后续点相连; 最后, 若新交点子集第 1 点为出点, 则自该点起, 将新的交点子集中的点, 按顺时针两两用圆形窗边界相连, 若新交点子集第 1 点为入点, 则按逆时针方向两两用圆形窗边界相连, 即可得到最终的裁剪结果.

在 P III 933 联想微机上用 C 语言实现了基于继承性的圆形窗口多边形裁剪算法. 由于文献[5]的算法中存在角划分运算, 其效率将低于文献[4]的算法, 因此本文的算法只与文献[4]的算法(简称为 DC 算法)进行比较.

取 10 条边的凸多边形和凹多边形各两个, 如图 4 的 I、II、III、IV 所示. 固定多边形 I 和 II 以及圆形窗口的圆心, 让圆形窗口的半径  $r$  以 20 为间距从 30 逐步取值到 130, 这样可以考察凸、凹多边形从与圆形窗口相离(按前面对出点、入点的定义, 相切按相离处理)到被圆形窗口包含这一过程中两种算法的效率; 多边形 III 和 IV 用来考察圆形窗口被多边形包含情况下两种算法的效率, 此处圆形窗口的半径为 30. 显然, 以上两种情况包含了圆形窗口与多边形的所有位置关系.

用本算法以及文献[4]的算法对圆形窗口下的 10 边形进行 100 万次裁剪所花时间(CPU 时钟)的

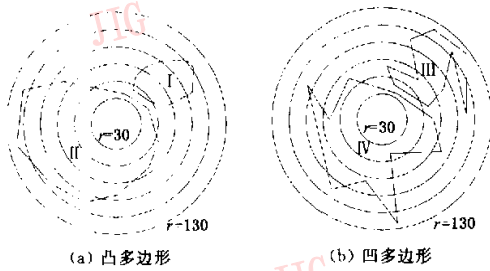


图 4 10 条边组成的凸、凹多边形

比较结果如表 1~3 所示. 时间计时从多边形的顶点集输入裁剪函数开始, 至得到输出顶点集结束.

表 1 凸多边形 I 从与圆相离到被圆包含过程的效率比较

算法	r					
	30	50	70	90	110	130
新算法	19	58	70	97	90	68
DC 算法	45	78	89	101	112	118
DC 算法/新算法	2.37	1.34	1.27	1.04	1.24	1.74

表 2 凹多边形 III 从与圆相离到被圆包含过程的效率比较

算法	r					
	30	50	70	90	110	130
新算法	21	62	68	81	114	68
DC 算法	38	56	83	92	113	116
DC 算法/新算法	1.81	0.90	1.22	1.14	0.99	1.71

表 3 多边形 II、IV 包含圆形窗口情况的效率比较

算法	多边形	
	I (凸)	IV (凹)
新算法	18	24
DC 算法	32	37
DC 算法/新算法	1.78	1.54

为便于分析比较, 将表 1 和表 2 所示结果以图 5 来表示.

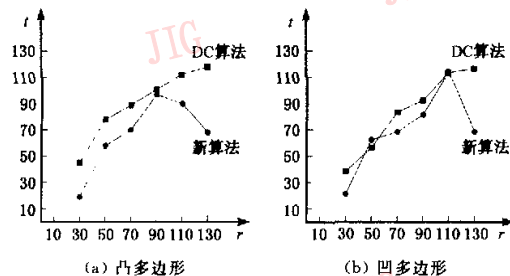


图 5 凸、凹多边形效率比较图

深入分析上面的实验结果可发现:

(1) 新算法效率总体上比 DC 算法效率高, 最大比值达到 2.37, 效率提高显著.

(2) 新算法充分体现了无交处理策略的优势, 即在多边形固定的情况下, 当圆形窗口较小时 (如  $r=30$ , 多边形完全位于圆形窗口的外部), 不相交的边可以通过编码技术直接舍弃; 当圆形窗口较大时 (如  $r=130$ , 多边形完全位于圆形窗口的内部), 不相交的边可以通过广义距离技术直接获取.

(3) DC 算法随着圆形窗口半径增大, 所耗时间有不断增加的趋势; 而新算法则在某半径达到最大值后开始不断下降, 这是很有意义的.

(4) 在某些半径处新算法时间消耗略大于 DC 算法, 这是由于此时很多边处于有交状态, 编码和广义距离增加了额外开销, 但从上一点的分析 and 图 5 可以看出, 这一半径区间范围不大.

(5) 当多边形的边数不同时, 两种算法所消耗的时间显然会有所不同, 但这不会影响两种算法效率差异的趋势. 这是因为两种算法在处理特定类型的多边形的边与圆形窗口的位置关系时所花时间不同, 而从前面的分析可以看出, 这种位置关系显然是有限的几种.

## 4 结 论

圆形窗口的多边形裁剪比矩形窗口的多边形裁剪更为复杂, 性能全面的算法较少. 本文采用的编码技术和广义距离技术及其继承性从多边形与圆形窗口的相对位置出发, 有效地处理了窗内边和窗外边这些无交边. 同时广义距离技术还可提高有效交点的求取速度. 实验结果也清楚地表明, 本文的算法总体效率较高, 稳定性良好.

显然, 编码技术和广义距离技术还可以容易地推广到其他窗口的裁剪, 从而可以在更广的范围内拓展裁剪效率提高的空间.

## 参 考 文 献

- 1 Rogers D F 著. 计算机图形学的算法基础[M]. 梁友栋, 石教英, 彭群生译. 北京: 科学出版社, 1988.
- 2 Sutherland I E, Hodgman G W. Reentrant polygon clipping[J]. Communication of ACM, 1974, 17(1): 32~42.
- 3 Weiler K J, Atherton P R. Hidden surface removal using polygon area sorting [J]. Computer Graphics, 1977, 11(2): 214~222.

- 4 杜玉建,曹德范.基于圆形窗口的简单多边形裁剪算法[J].中国图象图形学报,1998,3(11):937~940.
- 5 谢丽聪,白清源.圆形窗口的任意多边形裁剪[A].见:彭群生.计算机图形、图象和视觉的融合及应用--Chinagraph'2000会议论文集[C].杭州,2000:66~70.
- 6 唐荣锡,汪嘉业,彭群生等编著.计算机图形学教程[M].北京:科学出版社,1990.
- 7 金廷赞.计算机图形学[M].杭州:浙江大学出版社,1988.
- 8 孙家广,杨长贵.计算机图形学(第三版)[M].北京:清华大学出版社,1998.

**邢军伟** 1978年生,2003年获浙江大学硕士学位.主要研究领域为计算机图形学、管理信息系统开发等.



**陆国栋** 1963年生,2000年获浙江大学博士学位,教授,博士生导师.主要研究领域为计算机图形学、智能CAD、工程图样计算机理解等.



**谭建荣** 1954年生,1992年获浙江大学博士学位,教授,博士生导师.主要研究领域为产品信息建模、计算机图形学、计算机辅助设计等.

## 惠普图形工作站应用于动画领域

前不久惠普与美国梦工厂合作的又一力作,《辛巴达:七海传奇》动画片,在行业界引起了轰动.

《辛巴达:七海传奇》这部动画片在整个动画制作过程中,都采用了Linux系统的惠普工作站和服务器来完成的.由于采用Linux系统,整个系统具有非常良好的开放性.梦工厂在惠普的Linux硬件设备上,开发出了一套数字动画片检视工具,使得电脑上能呈现出没有压缩、全分辨率正确色彩的画面.这个工具取代了原本费用高昂功能有限的Irix-based解决方案.新的系统不但具有成本效益,并且协助工作人员有效率的完成“辛巴达”的制作工作.另外,梦工厂还开发出了数字动画回放工具.这是一种非压缩全效果彩色精确桌面回放解决方案.这个工具取代了基于Irix的高成本和性能有限的解决方案,这样梦工厂可以用它来建立更多的导演回放区域,真实再现制作情景. HP也帮助梦工厂开发了工作站上运行的ToonShooter软件,该软件允许动画制作人员数字扫描他们的手绘作品并整合在3D场景中,这样动画制作师在Linux下使用实时现场画板驱动器,感觉就和使用传统的画笔一样.

基于惠普设备提供的可靠性、扩展性,梦工厂在制作《辛巴达:七海传奇》中最大限度节省了成本.动画电影需要最精密的制图技术,这次梦工厂在惠普的Linux工作站平台上,不但突破制图过程中的限制、提供动画片绘制所需的灵活度,同时还节省了硬件支出.惠普的Linux工作站与服务器性能强大且功能多元,使得梦工厂得以在最短的时间内制作出了梦工厂有史以来人物最多的动画片.

在辛巴达剧组中的每一位画家,都使用惠普3D加速双屏幕工作站系统.双屏幕的作业环境使画家们可以同时开启多个窗口,进而加快制作的工作流程. HP Workstation xw8000则在整个制作过程中扮演了主角的角色. HP Workstation xw8000配置了Intel最新的E7505芯片组,最新的I/O技术,满足了《辛巴达:七海传奇》制作之需要.能最大限度保护用户投资成本.

由梦工厂制作发行的《辛巴达:七海传奇》是第一部完全在Linux系统平台上采用惠普工作站和服务器制作和完成的电影.在过去两年里, Linux操作系统开始进入电影业,并极大地改变了数字动画电影制作业.这一次在《辛巴达:七海传奇》扮演了关键任务的支持服务作用.惠普技术的稳定性和可扩展性使梦工厂能够综合利用“传统”的2D手绘动画和计算机生成的三维动画,创造出复杂的视觉特效.