

# 立体画的原理与生成技术<sup>\*</sup>

王宏武 吕晋育 董士海

(北京大学计算机系 北京 100871)

**摘要** 立体画是立体视觉技术与计算机技术相结合的产物。立体画的基本原理就是“墙纸效应”。立体视感是人的左右两眼视图交叠后通过大脑形成的立体虚象。同一幅立体画可以看出多个立体视感,分为交叉眼立体视感和非交叉眼立体视感。立体画分为两大类:基于模式的立体画和随机点立体画。在改进 Tyler 算法和 Thimbleby 模型的基础上,本文实现了一个 Windows 环境下的立体画生成系统——WinStereo。在 WinStereo 系统中,着重解决了“平面分层”现象,并提出了一种新的立体画——双向立体画及其生成算法。

**关键词** 立体视觉,立体画,随机点立体画,基于模式的立体画,双向立体画

## 1 立体视觉与立体画概述

三维立体画的出现并不是偶然的,它是立体视觉技术与计算机技术相结合的产物。早在 1838 年, Wheatstone 就发现了立体视觉原理。由于人的两只眼睛之间存在瞳距,所以同一景物在人的左右两眼视网膜上并非形成完全相同的图象,而是存在微小的差异——视差。Wheatstone 发现,正是由于两眼视图微小的差异,才使人具有分辨物体远近的能力。这样,便会对所看到的景物形成立体视感。根据这个原理, Wheatstone 发现了镜象立体视觉(mirror-stereoscopic)。在他之后,人们发现了“立体画对(stereo-pair)”:当两眼观看并排的两幅稍有差异的图象时,通过调节眼睛的会聚点,便会看出立体的图象来。双色滤光眼镜和偏振光眼镜的出现,使立体画对可以重叠为一幅画。当戴上滤光眼镜或偏振光眼镜看画时,重叠的画对经过眼镜的过滤,分解为左右两个视图分别进入人的左右两眼。根据这一原理,人们发明了立体电影。

但是,“立体画对”的立体效果不是很理想的,看起来很费力,而重叠画又需要专用的设备。三维立体

画却独辟蹊径,它采用计算机将三维实体融于一幅画中,利用人眼的视觉特性,达到获得立体视感的目的。由于它不需要专用设备,所以迅速流行起来,而且必将在很多领域得到应用。

## 2 立体画原理

### 2.1 墙纸效应(wallpaper effect)

人们很早就发现(1844 年, Brewster),水平重复画某一图案(例如墙纸),适当调节人眼视线的会聚点,使一只眼看到的图案与另一只眼看到的图案相隔一个周期时,便会在画平面的“后面”看到一个画平面,由于它距离物理画平面有一段距离,而使得人们感觉它象一个立体的平面,这就是所谓的“墙纸效应”。也即立体画的基本原理。

从外观上看,立体画象是一张普通的画。有的立体画布满了某种图案,而有的则被一些杂乱无章的随机点所覆盖。神奇的三维立体世界就藏于画中。如果仔细观察立体画,就会发现立体画表面的纹理是有周期性的。在每一条水平线上,总是重复某个特定的点序列。同时,重复周期有细微的变化。由 2.2 节可知,这会在每条水平线上形成凸凹不平的立体视

<sup>\*</sup> 收稿日期:1995.09.19;收到修改稿日期:1996.03.29

感。当看整幅画时,便会形成一个凸凹不平的三维轮廓构成的立体视感。简单地讲,立体画的基本原理就是“墙纸效应”。它的制作要借助于计算机,而最终的立体视感的形成则取决于人眼和人脑本身。立体画不是“实在”的立体实物,而是人脑形成的虚象。

### 2.2 立体画视觉模型

立体画就是重复某一图案所得的画。图 1 中,画平面 P 上的点  $D_1, D_2, \dots, D_n$  代表等间距重复的同一图案。 $L, R$  为人的左右两眼,从  $L, R$  两点出发,分别向  $D_1-D_n$  点做射线  $L_1-L_n$  和  $R_1-R_n$ , 代表左右两眼发出的视线束。可以看到,  $L_1$  交  $R_2$  于  $I_1$  点,  $L_2$  交  $R_3$  于  $I_2$  点, ...,  $L_{n-1}$  交  $R_n$  于  $I_{n-1}$  点, 这  $n-1$  个点  $I_1-I_{n-1}$  形成平面  $S_1$ 。由于  $I_1-I_{n-1}$  中每一个点处, 都有分别来自左眼  $L$  和右眼  $R$  的视线, 所以人脑会在  $I_1-I_{n-1}$  处形成虚象。并且, 这些虚象都具有立体视感, 它们形成立体画平面  $S_1$ 。另外,  $R_1$  交  $S_1$  于  $I_0$  点,  $L_n$  交  $S_1$  于  $I_n$  点, 这两个象点都只有一条视线穿过, 没有来自另一只眼相对应的视线与之相交, 所以没有立体视感。

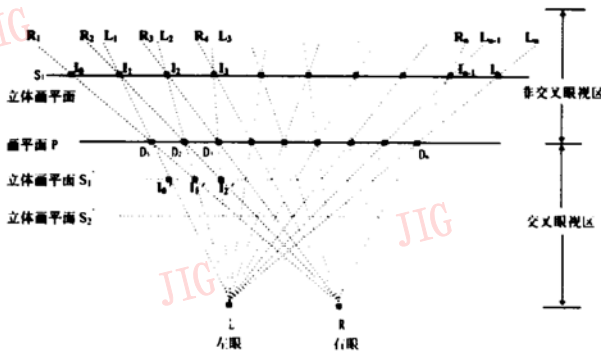


图 1  
Fig. 1

可以发现,  $L_2$  与  $R_1$  可交于  $I_1'$  点,  $L_3$  与  $R_2$  可交于  $I_2'$  点, ...,  $L_n$  与  $R_{n-1}$  可交于  $I_{n-1}'$  点, 这样, 点  $I_1'-I_{n-1}'$  在画平面 P 前面也形成一个平面  $S_1'$ 。显然  $S_1'$  也具有立体视感。由于  $I_i$  点是  $L_{i+1}$  与  $R_i$  的交点, 是左眼看  $D_{i+1}$  点与右眼看  $D_i$  点时形成的虚象, 这时两眼的视线需要交叉, 所以, 这种视感叫交叉眼立体视感。而与之对应,  $S_1$  则为非交叉立体视感。容易看出, 在  $S_1'$  之前还会形成  $S_2', S_3', \dots$ , 它们是左右眼视图相差大于一个周期时重叠而产生的视感。它们构成“交叉眼立体视区”。在  $S_1$  之后, 还有立体平面  $S_2, S_3, \dots$ , 它们构成“非交叉眼立体视区”。一般的人使用非交叉眼的方法来看立体画, 所以, 在设计立体画

时, 通常将  $S_1$  定为立体画平面。这时, 如果采用交叉法看, 会看到  $S_1'$  处的立体画, 只是其凸凹特性刚好与  $S_1$  处看到的相反; 同时, 其它画平面, 如:  $S_2, S_3, \dots, S_2', S_3', \dots$ , 处也可以看出立体效果, 但其画的内容已经更迭多次, 没有观看的意义。

图 1 中点的重复周期是相等的, 即  $D_1-D_n$  之间的间距相等。如果在重复点时周期不相等会怎么样呢? 图 2 展示了这种情况。

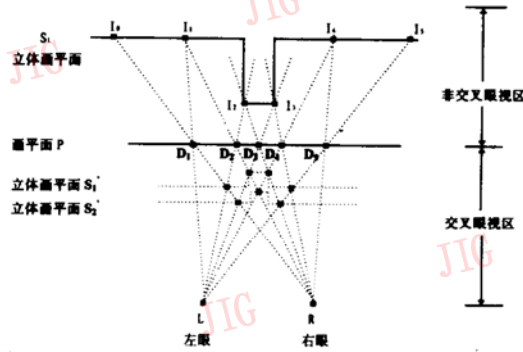


图 2  
Fig. 2

在图 2 中, 由于  $D_1-D_5$  之间的距离不相等, 致使虚象点  $I_1-I_4$  不在同一平面上, 而是  $I_2, I_3$  要比  $I_1, I_4$  高出来一些。这就产生了远近距离感, 即立体视感。精心地组织点的重复周期, 就会形成所期望的高低不平的三维实体轮廓的立体视感。

另外, 我们发现, 处于非交叉眼视区的  $S_1$  与处于交叉眼视区的  $S_1'$  具有相反的凸凹性。所以对于同一幅立体画, 采取的观察方法不同, 会形成不同的立体视感。但是在一般情况下, 非交叉眼视感比较容易获得, 而交叉眼视感需要两眼形成“对眼”。一方面, 不易看出立体效果, 另一方面, 这种看画的方法对视力也有影响。

### 2.3 立体画分类

现在市面上能见到的立体画主要有两大类: 随机点立体画和基于图案的立体画。

随机点立体画看上去杂乱无章, 没有任何图案, 三维景物藏于其中, 给人一种神秘的感觉。

基于图案的立体画看上去象一幅普通的画, 它由周期性的图案组成。这种“表面”的图案常常与“藏”于其中的立体画融为一体, 形成一个完整同一的构思, 具有很高的艺术欣赏价值。

值得一提的是, 市面上还流行另外一种立体画。它由两幅或多幅同一景物的不同视角投影图并排排

列而成,所以又称立体画对(Stereo-pair)。它是“所见即所得”的,看到的立体视感中的立体实体与画面图案的内容相同,因而不具有“画中画”的特征。使用一般的三维造型软件(如3D Studio)和图象编辑软件就可以制作它。本文讨论的立体画不包括这一类。

### 3 立体画的制作算法

#### 3.1 Tyler<sup>[1]</sup>算法

在总结前人工作的基础上,Tyler(1990)提出了一个简单而直观的立体画生成算法。Tyler算法的大致思想如下:在三维立体画面上,周期性地重复模式图案序列,以深度序列控制每点处的重复周期,形成一个有“疏”有“密”的模式图案重复序列,即立体画序列

细,刻画三维实体的能力也越强,三维效果也越好。 $T$ 为常数,代表最小的重复周期。

```
FOR y := 1 TO maxY DO
BEGIN
  FOR x := 1 TO maxX DO
  BEGIN
    index := x - (T + Z-Buffer[x,y]);
    IF index <= 0 THEN
      Image-Buffer[x,y] := Pattern-Buffer[x,y];
    ELSE
      Image-Buffer[x,y] := Image-Buffer[index,y];
  END
END
END.
```

图3  
Fig. 3

由算法可知,在任一深度为 $z(z=Z-Buffer[x,y])$ 的点 $(x,y)$ 处,可以得出 $index:=x-(T+z)$ , $(index,y)$ 表示与此点相对应的向前参考点。在三维立体画Image-Buffer中,任一点的颜色值Image-Buffer $[x,y]$ 必须与它所对应的参考点的颜色值相同。在生成立体视感时, $(x,y)$ 和 $(index,y)$ 对应于同一三维点,它们分别是这个三维点的左视图和右视图。若 $index \leq 0$ ,参考点的颜色就要从模式图案中取,即:

$Image-Buffer[x,y] := Pattern-Buffer[x,y];$

显然,三维画面Image-Buffer中点的所有颜色值来自

于Pattern-Buffer。Image-Buffer以可变的周期 $t(T \leq t < T+D; D$ 为最大深度)来重复Pattern-Buffer。周期大小 $t$ 受Z-Buffer控制:

$$t = T + Z-Buffer[x,y].$$

若Z-Buffer中的值全相同,即:

$$Z-Buffer[x,y] := d;$$

$$(1 \leq x \leq \max X, 1 \leq y \leq \max Y)$$

那么在任意点处,重复周期为恒值 $t=T+d$ 。用这样的Z-Buffer生成的立体画只能看到一个深度为 $d$ 的平面,这就是前面提到的“墙纸效应”。

若Z-Buffer中的值有大有小,那么在每一点处的重复周期 $t$ 将不同,

$$t := T + Z-Buffer[x,y];$$

从而使生成的立体画表面有“疏”有“密”。若某处越“密”则可知其越接近于平面,那么在一维视感中

一些局限性:在Tyler算法中,深度 $z$ 与重复周期 $t$ 之间的关系为一个简单的等式表达: $t=z+T$ 。这个等式不能精确表达 $z$ 与 $t$ 之间的关系,致使生成的图象在深度信息方面有所失真。

#### 3.2 Thimbleby<sup>[2]</sup>模型

为了克服Tyler方法的不足,Harold W. Thimbleby(1994)建立了一个模型。该模型不仅精确地描述了深度信息与重复周期的关系,而且提供了其它一些控制立体画生成的手段,图4给出了Thimbleby模型的示意图。其中L,R分别为人的左右眼,它们之间的距离 $E$ 为瞳距(一般取2.5英寸)。I平面为图象平面。三维实体被夹在两个平面N与F之间。为了视感舒服,画平面I被置于人眼所在平面与平面F的正中间,距离两边均为 $D$ 。深度矢量 $z$ 由F平面指向N平面(这与Tyler模型中的 $z$ 方向相反)。N与F平面之间的距离( $uD$ )可调节,一般 $u$ 取 $1/3$ 。通过调节 $u$ 的取值就可以调节目深。

对于三维实体上的任意一点P,若其深度为 $z$ ,由P向L和R连线,交I平面于 $P_L$ 点和 $P_R$ 点,它们之间的距离 $t$ 为: $t = E * (1 - uz) / (2 - uz)$ 。这个 $t$  ( $0 < t < D$ )对应于Tyler算法中的周期 $t$ 。显然,这个

表达式更精确地描述了周期  $t$  与深度  $z$  之间的关系。

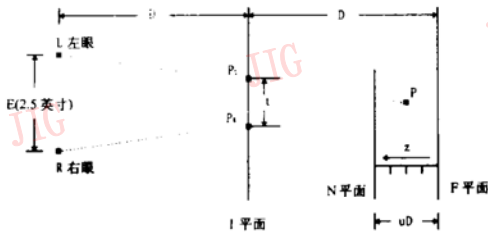


图4  
Fig. 4

### 4 WinStereo 三维立体画制作系统

#### 4.1 WinStereo 系统结构

在 Microsoft Windows 环境下,我们设计实现了一个立体画生成系统——WinStereo。WinStereo 能根据指定的模式图案和 Z 图象生成单色或彩色的立体画,或生成随机点立体画。WinStereo 的基本结构如图5。WinStereo 系统的核心是立体画发生器,它有两个输入:一是模式序列,可以来自随机序列发生器,也可以来自特定的模式图案缓冲区;二是

深度序列。深度序列定义了画“中”的三维立体景物的轮廓,可以由三维造型系统给出,或者是三维信息 Z 发生函数,或者是其它的三维信息。立体画发生器根据两个输入序列,产生出三维立体画序列,进而构成三维立体画。

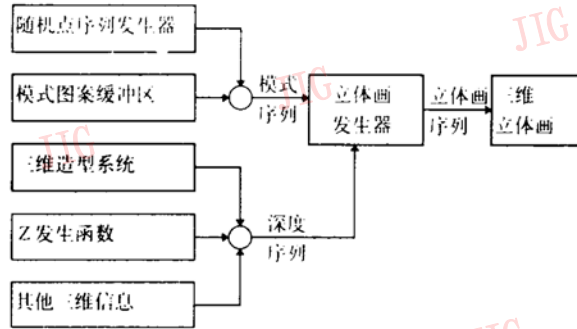


图5

Fig. 5

图6是 WinStereo 系统生成的随机点立体面(画 中为两个汉字:“长城”)。图7为 WinStereo 系统生成的一张基于模式的立体画(画中“藏”着一个半球)。

WinStereo 系统改进了 Tyler 和 Thimbleby 算法,消除了以前算法中的“平面分层”现象。并且尝试生成了一种新的立体画——双向立体画。

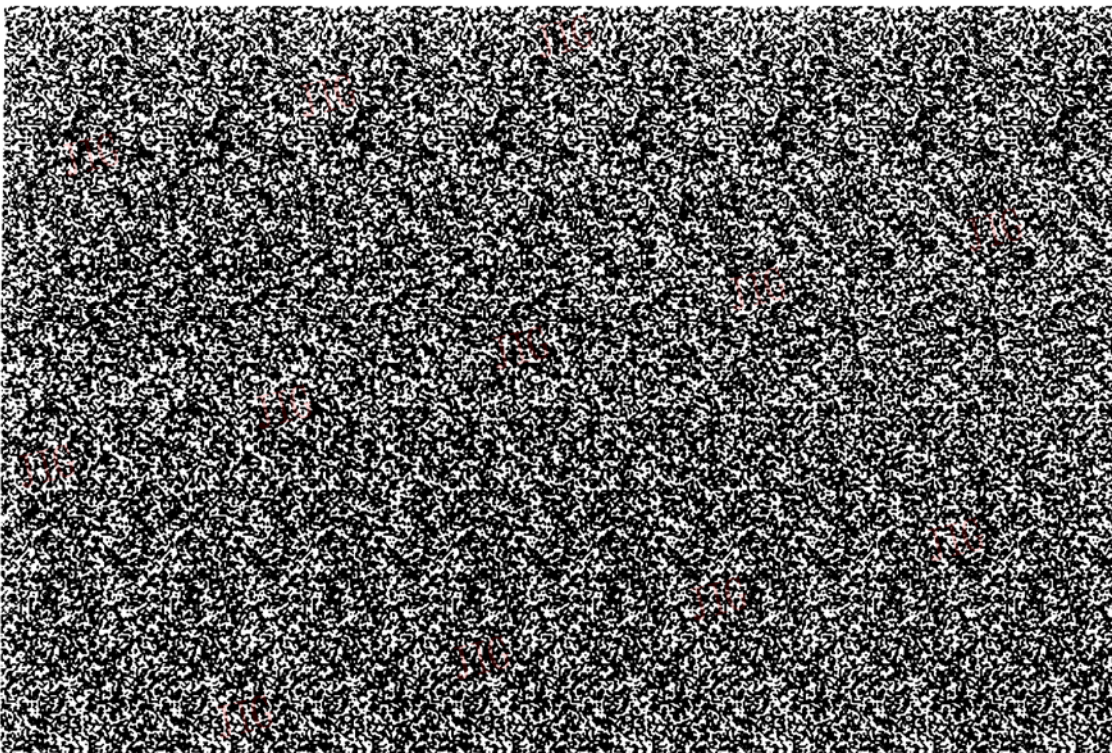


图6  
Fig. 6



图7  
Fig. 7

#### 4.2 平面分层现象的消除

在 Thimbleby 算法和以前的 Tyler 算法中,都存在“平面分层”现象,图8展示了用 Thimbleby 算法生成的一个半球的立体画。可以看出,球变成了一叠直径由大到小的圆片。这是由于图象的分辨率低引起的。

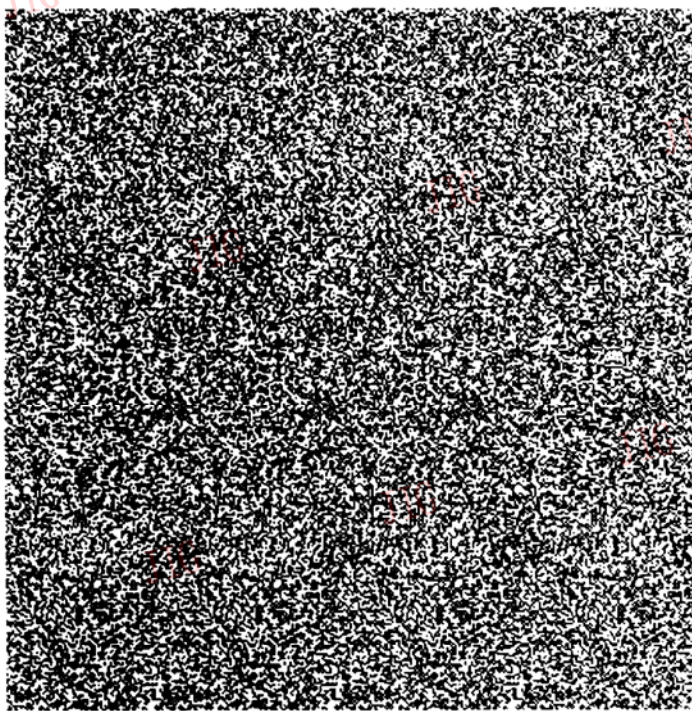


图8  
Fig. 8

解决“平面分层”现象的最直接的办法是提高立体画分辨率。但是,一般情况下,分辨率的提高受图

象尺寸的限制。在 WinStereo 系统中,我们是在不改变输出图象分辨率的前提下,通过提高计算分辨率,用在输出时加权平均的方法来消除“平面分层”现象的。

我们采用6倍分辨率来计算,输出图象 Outpix-  
Buff[x,y]由下式给出:

$$\begin{aligned} \text{OutPix-} \text{Buff}[x,y] := & (\text{Image-} \text{Buff}[x * 6,y] * 14 + \\ & (\text{Image-} \text{Buff}[x * 6 - 1,y] + \text{Image-} \text{Buff}[x * 6 + 1,y]) * 14 + \\ & (\text{Image-} \text{Buff}[x * 6 - 2,y] + \text{Image-} \text{Buff}[x * 6 + 2,y]) * 11 + \\ & (\text{Image-} \text{Buff}[x * 6 - 3,y] + \text{Image-} \text{Buff}[x * 6 + 3,y]) * 8 + \\ & (\text{Image-} \text{Buff}[x * 6 - 4,y] + \text{Image-} \text{Buff}[x * 6 + 4,y]) * 5 + \\ & (\text{Image-} \text{Buff}[x * 6 - 5,y] + \text{Image-} \text{Buff}[x * 6 + 5,y]) * 3 + \\ & (\text{Image-} \text{Buff}[x * 6 - 6,y] + \text{Image-} \text{Buff}[x * 6 + 6,y]) * 2) / \\ & 100; \end{aligned}$$

图9展示了改进后的立体画,其中的“平面分层”现象已经消除。

#### 4.3 双向立体画

一般的立体画由一幅 Z 图象生成,Z 图象控制立体画每一条水平扫描线上相关点的间距。而对垂直方向没有影响。这种立体画旋转90度后,什么也看不出来。而双向立体画则试图将两幅 Z 图象融于一幅立体画中,一幅控制立体画中水平扫描线的生成,另一幅控制垂直扫描线的生成。这种立体画不仅在水平方向能看出立体效果,当把它旋转90度或270度时,还能看出另外一幅立体景物。图10就是一幅双向立体画。

因为双向立体画中包含两幅立体景物,因此它的生成需要两幅 Z 图象。这样,图象上的每一点不仅要受水平方向的约束,还要受垂直方向的约束。很明显,高质量的双向立体画的生成需要一种双向锁定的递归算法来解决约束冲突,其时间复杂度和空间复杂度都非常大。在 Winstereo 系统中实际实现时,我们采取了一种冲突求解的近似算法。

图11给出了双向立体画生成算法的 C 语言算法。其中,pix[]为立体画矩阵,samex[][]和 samey[][]为水平方向和垂直方向的约束矩阵。如果图象上一个点的取值不受别的点约束,则约束矩阵的值就为这一点本身。即:samex[][][x]=x,samey[][][y]=y。在初始状态下,每一点都不受约束,12—19行反映了这一点。如果两个点属于同一三维点的左右视点,它们就是一对受约束点。受约束点对中的左视点的取值受其右视点的约束。即 samex[y][left]=right(第50行)。垂直方向的约束矩阵求法与水平方向相似。14—69行求解约束矩阵。70—88行为冲突求

解近似算法。可以看出，当水平约束和垂直约束相矛盾时，我们采取抓阄的算法(82--85行)。

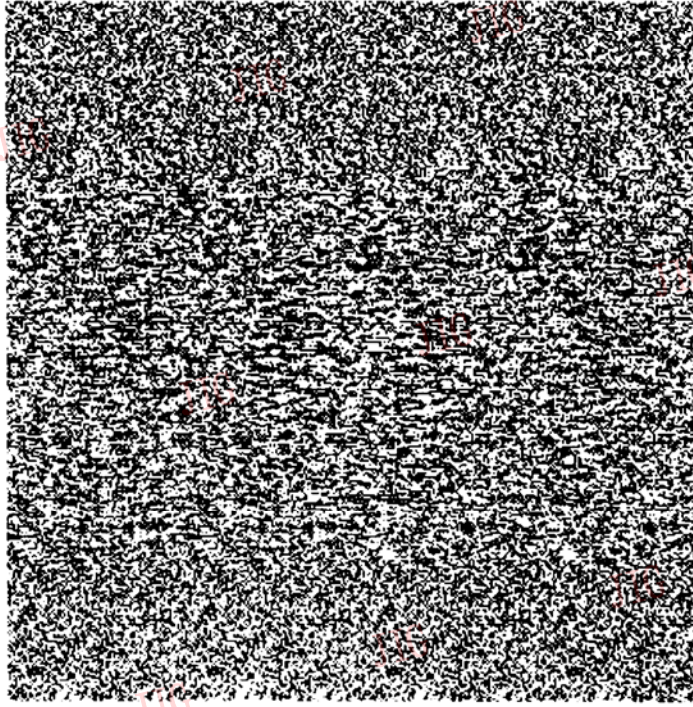


图9  
Fig. 9

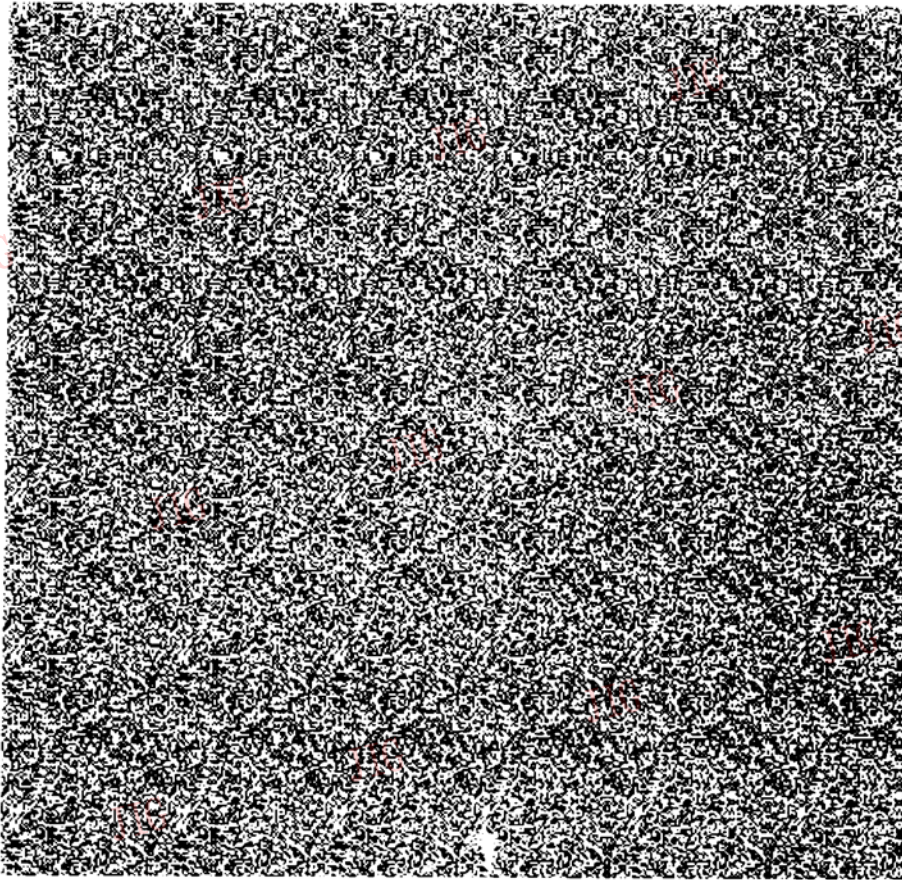


图10  
Fig. 10

```

01 #define round(x)      (int)((x)+0.5)
02 #define DPI          72          /* 分辨率 */
03 #define E             round(2.5*DPI) /* 瞳距 */
04 #define mu            (1/3.0)
05 #define separation(Z) round((1-mu*(Z))*E)/
    (2-mu*(Z)) /* 重复周期 */
06 void DrawRandomDotStereogram(void)
07 {
08     int x,y;
09     unsigned char pix[maxY][maxX];
10     int samex[maxY][maxX],samey[maxY][maxX];
11
12     for(y=0;y<maxY;Y++) /* 初始化约束矩阵 */
13     {
14         for(x=0;x<maxX;x++)
15         {
16             samex[y][x]=x;
17             samey[y][x]=y;
18         }
19     }
20
21     for(y=maxY-1;Y>=0;y--) /* 由上至下,扫描每一行 */
22     {
23         int sx,sy;
24         int left,right,up,down;
25
26         for(x=maxX-1;x>=0;x--) /* 对于任意行,自右向左检查 */
27         {
28             sx=separation(ZX(x,y));
29             sy=separation(ZY(x,y));
30
31             left=x-sx/2;
32             right=left+sx;
33             up=y-sy/2;
34

```

```

41         {
42             if(k<right
43                 left=k;
44             else
45                 left=right;
46                 right=k;
47             }
48         }
49         samex[y][left]=right;
50
51         /* 垂直检查 */
52         if(0<=up && down<maxY)
53         {
54             for(k=samey[up][x];K!=up

```

```

56     && k!=down;
57         {
58             k=samey[up][x]
59             if(k<down)
60                 up=k;
61             else
62                 {
63                     up=down;
64                     down=k;
65                 }
66             samey[up][x]=down;
67         }
68     }
69 }
70 for(x=maxX-1;x>=0;x--)
71 {
72     if(samex[y][x]==x && samey[y][x]
73     ==y)
74     {
75         pix[y][x]=rand()%256;
76     }
77     else
78     {
79         if(samex[y][x]==x && samey[y]
80         [x]!=y)
81             pix[y][x]==pix[samey[y]
82         [x]][x];
83         else if (samey[y][x]==y &&
84         samex[y][x]!=x)
85             pix[y][x]=pix[y][samex[y]
86         [x]];
87         else if (rand()%2==0) /* 抓阄算法 */
88             pix[y][x]=pix[samey[y][x]]
89         [x];
90         else
91             pix[y][x]=pix[y][samex[y][x]];
92     }
93 }

```

三维立体画技术是立体视觉领域里的一颗新星。由于它生成算法简单,不需要任何设备即可获得立体视感,从而迅速地发展起来。立体画技术不仅在艺术领域,而且将可能在科技领域(如三维动画、科学可视化)得到广泛地应用。

### 参考文献

- [1] C. W. Tyler and M. B. Clarke. The Autostereogram. *SPIE Stereoscopic Displays and Applications*, 1990, Vol. 1256, pp. 182-196.
- [2] Harold W. Thimbleby, Start Inglis and Ian H. Witten. Displaying 3D Images: Algorithms for single-Image Random-Dot Stereograms. *Computer*, 1994 Oct., pp. 38-48.



王宏武,北京大学计算机系硕士研究生。主要研究方向为三维动画、多媒体技术、超文本技术、用户界面及三维立体画。

## The Principles and Generating Techniques of Autostereograms

Wang Hongwu Lü Jinyu Dong Shihai

(Department of Computer Science and Technology Peking University, Beijing 100871)

**Abstract** Autostereogram results from the combination of computer technology and stereo vision.

### 美国政府与 CLI 签订跨世纪合约

#### CLI 被指定为 GSA 提供视频通讯设备

美国视讯公司(CLI)北京办事处转讯,加州,圣荷塞,1996年6月18日讯 美国政府今天宣布 GSA 与 CLI 签订了将连续执行到下个世纪的一系列共计58个合约,依据该合约,CLI 将为全美及遍及全球的各类美国政府机构提供全套的会议电视产品,并提供全面的安装、技术支持服务。

GSA 计划(No. GS35F1043D)包括 CLI 符合 H. 320 标准,获得工业奖项的 eclipse 集成会议电视系统;电视质量、最高30帧每秒、480线解象度高品质的 Radiance 集成会议电视系统;最多可在同一电视会议中联接48个点的 CLI 多点控制器 MCU;适应用户视频要求,获奖的 RembrandtII/VP,所有 CLI 系统符合 H. 320 I TU-TSS 工业标准、国防部 Corporation for Open Systems 的要求,CLI 的 Radiance 和 RembrandtI/VP 系统可工作于拨号 ISDN 数字线或专线,具备 CLI 的 H. CTX Plus™ 算法,可提供从336Kbps 到2.048Mbps 电视质量的视频图象,该系统被美国联邦政府的机构广泛使用。

Mike Raymond, CLI 设在 Reston, VA 的联邦市场经理说:“CLI 一向关注这一重要市场,我们非常高兴获得新的 GSA 计划,GSA 计划使我们的联邦用户可以以极具竞争力的政府价格非常方便地购买当今市场上最先进的会议电视产品和服务,对于所有参与者都非常有利。”

CLI 是世界数字图象通讯领域领先的设计制造厂家,利用其核心的压缩数字视频技术,CLI 为各国政府、商业、教育、医疗卫生等领域用户提供一系列配套的会议电视产品,并为娱乐、商业电视及远程教育供应广播级电视产品,公司总部设在加州圣荷塞,拥有遍及全美的销售服务网络,并在澳大利亚、加拿大、欧洲、日本及中国设有办事处,为全球50多个国家提供服务。

美国视讯公司北京办事处: 地址:北京市复兴门外大街6号光大大厦20层 电话:(010)68561026~30 传真:(010)68561018 邮编:100045