第 15 卷 第 3 期
1989 年 5 月

自 动 化 学 报
ACTA AUTOMATICA SINICA

Vol. 15, No. 3
May, 1989

CAD 文件显示中遮盖与透视的原理与实现

李 文 余
(中国科学院自动化研究所)

摘 要

本文提出一种实现 CAD 图形显示中多层文件或图纸的遮盖与透视的方法和理论，可以大大减少程序量和存储空间，提高各文件的转换速度.

关键词——遮盖；透视；层次空间；消盖.

一、数 学 原 理

在 n 维空间 V 中，如果存在 L 个 $n-1$ 维平行子空间，则称此空间为层次空间.

如果层次空间中每个层次子空间是具有相同 i 个有限状态，则称为等同状态层次空间.

令各层次子空间为 A, B, C, D, $\cdots$, N, 它们相应的状态为 $(A_0, A_1, \cdots, A_{i-1})$; $(B_0, B_1, \cdots, B_{i-1})$; $\cdots\cdots$; $(N_0, N_1, \cdots, N_{i-1})$.

现在按下述方法构造一个递归状态集:

$$\left.\begin{aligned} &\text{集合 } A \supseteq (A_0, A_1, \cdots, A_{i-1}), \\ &\text{集合 } B \supseteq (B_0, B_1, \cdots, B_{i-1}), \quad B_0 = A, \\ &\text{集合 } C \supseteq (C_0, C_1, \cdots, C_{i-1}), \quad C_0 = B, \\ &\qquad\cdots\cdots\cdots\cdots \end{aligned}\right\} \tag{1}$$

或者以 $G^{(1)}$ 表示集合 A, $G^{(2)}$ 表示集合 B 等，则

$$G_0^{(k)} = G^{(k-1)},$$

因此有

$$G^{(k)} \supset G^{(k-1)}.$$

以此递归状态集合所定义的等同状态层次空间称为递归状态层次空间，以后简称状态层次空间.

令 S 表示状态，则根据上面定义，各层次子空间的状态有:

$$\left.\begin{aligned} &S(A_0) = S(B_0) = \cdots = S(N_0), \\ &S(A_1) = S(B_1) = \cdots = S(N_1), \\ &\qquad\cdots\cdots\cdots \\ &S(A_{i-1}) = S(B_{i-1}) = \cdots = S(N_{i-1}). \end{aligned}\right\} \tag{2}$$

本文于 1986 年 11 月 19 日收到.

现在在空间 V 中沿与各层次子空间垂直的坐标，取此坐标与各状态层次子空间交点的状态，形成一个状态序列. 为方便起见，设层次为 4，D 层在上，A 层在下，则状态序列记为

$$D_iC_kB_lA_m, \quad i、k、l、m = 0, 1, \cdots, i-1.$$

当 $i \neq 0$ 时，此状态序列的状态为

$$S(D_iC_kB_lA_m) = i, \tag{3}$$

即状态序列的状态取决于最上层层次的状态，将状态与序列表示在一起记为

$$D_iC_kB_lA_m(i), \quad i \neq 0.$$

此时就完成上层状态对下层状态的遮盖*. 当 $i = 0$ 时为透视状态，即 A_0、B_0、$\cdots$、N_0 是处于透视态. 当该层处于透视态时，以下一层的状态表示该序列的状态，如

$$S(D_0C_kB_lA_m) = k,$$

或

$$D_0C_kB_lA_m(k).$$

进一步有

$$D_0C_0B_0A_m(m),$$

及

$$S(D_0C_0B_0A_0) = B、G. \tag{4}$$

式(4)表示全透态，B、G 称为背景.

在空间 V 中，一个点的状态序列称为遮盖元，所有遮盖元的集合即组成层次遮盖.

如果每个状态的取值是二态的，称二元遮盖；如果每个状态的取值是多值的，称多元遮盖.

定理 1. 状态序列的状态数等于 I^L

其中 I 是状态数 i 的大写，L 为层次空间个数. 因状态序列中的每一项有 I 个状态，项数为 L，总状态为 $I \times I \times \cdots \times I$，共有 L 个 I.

二、状态的转换与透视

上面提到当上层状态处于透视态时，序列的状态发生变化，设有如下序列转换：

$$D_iC_kB_lA_m(j) \to D_0C_kB_lA_m(k) \to D_0C_0B_lA_m(l) \tag{5}$$

及

$$D_iC_kB_lA_m(j) \to D_iC_0B_lA_m(j) \to D_0C_0B_lA_m(l), \tag{6}$$

两者的转换顺序不同，而终止的序列状态相同，则有：

定理 2. 状态序列的转换所获得的终点状态与转换顺序无关.

证. 因为状态层次空间各层的状态是独立的，不受其它层次影响，而且序列与状态具有对应关系，虽然过渡状态与转换顺序，即路径有关，而终止状态只决定于终点序列，与路径无关.

根据透视态的定义及序列转换可知，要完成高层状态对底层状态的透视，必须将高层状态转换为透视状态.

---

* 遮盖与逻辑覆盖不同，遮盖时递归状态表中的逻辑最小项的集合在逻辑上不覆盖.

三、状态值空间与消盖

设状态序列 $D_jC_kB_lA_m(j)$，j、k、l、$m=0,1,\cdots,i-1$。无论二元遮盖或多元遮盖，序列的状态与状态的值具有一一对应关系，所有状态的取值形成一个状态值空间。

为一般起见，假定在多元遮盖情况，令

$$j=\alpha,\quad k=\beta,\quad l=\gamma.$$

如果不改变状态而改变状态的取值，使上层状态的值等于下层状态的值，而此值对应一定的显示状态，可以取得与透视同样的效果，称之为消盖。如把 j 的值变为 β，则 $j=k=\beta$，有：

$$S_r(D_jC_kB_lA_m)=S_r(D_0C_kB_lA_m)=\beta,$$

其中 S_r 表示状态序列的取值。

对应式 (5)，先令 $j=k=\beta$，再令 $j=l=\gamma$ 及 $k=l=\gamma$；对应式 (6)，先令 $k=l=\gamma$，再令 $j=l=\gamma$。两者的动作次数不同，也就是程序量不同。

定理 3. 多层消盖时，终点状态值与顺序无关，而计算量与消盖顺序有关，自上而下消盖的计算量最大，自下而上的最小。

四、图形的透视与消盖

上面提到的透视与消盖是对一个状态序列进行的，一个图形是多个状态序列的集合。只有当整个集合的状态改变或取值被修改时，才能完成整幅图形的透视与消盖，而一幅图形的像点少则数千多则数十万个像点。以往用修改图形内容的办法[1] 要浪费许多存储空间及往返传送时间；现在由于状态序列的状态总数很少，而且有的状态与下层相同，有的已处于透视状态，当透视一层时只须改变上层状态或其取值，所以一般情况只要修改 4 到 6 个状态序列或状态值空间的 4 到 6 个数值即可实现。

五、遮盖与透视或消盖的实现

将图形帧存储器按位平面划分，它们就是三维空间中的二维平行子空间。一般表示图形遮盖有三种状态就可以实现，即透视态、文件基底态和基底文字态。为了二进制方便起见，还有一种不用的状态，称为文字出格态，这样可以用二个位平面构成一个具有四种状态的状态层次子空间。如果是八位的存储器，可以构成四层遮盖的状态层次空间。

令二进制的 00 表示透视态，01 为文件基底态，10 为文字出格态，11 为基底文字态，按表 1 构成四层递归状态集，用两位十六进制表示。

从表 1 看出，各层状态在透视态递归，A_0 为背景，已于前述。$A1$，$B1$，$C1$，$D1$ 为文件基底态；$A3$，$B3$，$C3$，$D3$ 为基底文字态。

现在主要问题是如何将状态层次空间与递归状态集结合起来。

用 P_0，P_1，$\cdots$，P_7 表示图形存储器的各个位平面，P_0 为最低位，P_7 为最高位，P_0P_1

表 1

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00	A_0	A_1	A_2	A_3	B_1				B_2				B_3				C_0
	B_0																D_0
10	C_1																
20	C_2																
30	C_3																
40	D_1																
50																	
60																	
70																	
80	D_2																
90																	
A0																	
B0																	
C0	D_3																
D0																	
E0																	
F0																	

表 2

P_7	P_6	P_5	P_4	P_3	P_2	P_1	P_0	
0	0	0	0	0	0	0	0	B. G.
0	0	0	0	0	0	0	1	*A* 层文件基底
0	0	0	0	0	0	1	1	*A* 层基底文字
0	0	0	0	0	1	\	\	*B* 层文件基底
0	0	0	0	1	1	\	\	*B* 层基底文字
—	—	—	—	—	—	—	—	(*C* 层略)
0	1	\	\	\	\	\	\	*D* 层文件基底
1	1	\	\	\	\	\	\	*D* 层基底文字

构成 *A* 层，P_2P_3 构成 *B* 层，P_4P_5 为 *C* 层，P_6P_7 为 *D* 层. 把各层的前一个平面 (P_0, P_2, P , P_6) 作文件基底,后一个平面 (P_1, P_3, P_5, P_7) 作文字图形存储，则它们就按表 2 的编码有机地结合起来.

为了实现透视变换,使用一个 256 字节的查找表. 在正常遮盖状态,它们的内容与地址呈线性关系,即内容等于地址. 当需要透视时，可将该层对应的两位置成 00. 设在只

有二层情况下，把表 1 中 $B1$ 至 $B3$ 对应的 04—0F 的高二位(二进制)置 0，就成为 00—03 的重复，也即表 2 中“/”符号处依下层的码取值．层次越高，可能置 0 的单元越多．四层时最多不超过 192 个单元，比起修改图形存储器来要少了许多倍，而且是只写不读，与图形的内容无关，不破坏图形存储器原有内容．

但是在监视器上要得到二值化的遮盖图形，还必须经过二值化转换(在黑白监示器上是二元遮盖)或彩色码转换（在彩色监示器上是多元遮盖)．因为要将各层的图形同时以遮盖形式显示在屏幕上，就不能用屏蔽与叠加的选位面方式，因那样经过数模转换后，它们以多灰级出现，不是所希望的．将递归状态表(表 1) 进行赋值，1 为白色基底，0 为黑色文字或透视或背景．这样当 $A1$、$B1$、$C1$．$D1$ 赋值“1”，$A3$、$B3$，$C3$，$D3$ 赋值“0”时，可实现遮盖．例如表 1 中“77”这个单元对应于 $D1C3B1A3$ 序列，其二进制码为 01110111，根据上述赋值后成为 11001100，这里用二位相同码代表 0 或 1．由于状态取决于高层，$D1$ 为 1 是白色基底，它遮盖住了 $C3$ 等于 0 所表示的黑色文字．为避免灰度问题，“77”这个单元是处在 $D1$ 这个大区域中，$D1$ 赋值 1，就根据上层优先的原则，将此点的值改为 11111111．这样虽实现了遮盖，却又不能实现透视，因为透视态将成为黑色遮盖．例如，当 $D1$ 为黑色赋值时变 $D3$，上述赋值序列成为 00001100，应该显示 $B1$ 的白色，可是按上述方法将变成“00000000”的黑色．

为此还需要一个变换表，如图 1．前一个称状态透视表，后一个叫赋值表．赋值表为 256 位即可，每位对应状态表中一个序列，可根据序列的透视情况予以正确赋值．这样在硬件上就要两级静态存储器的查找表，比较麻烦，与现有微机不匹配．

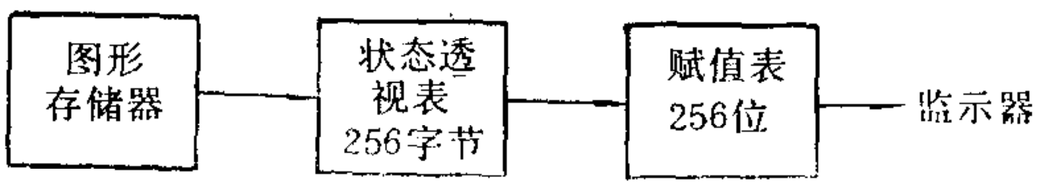

图　1

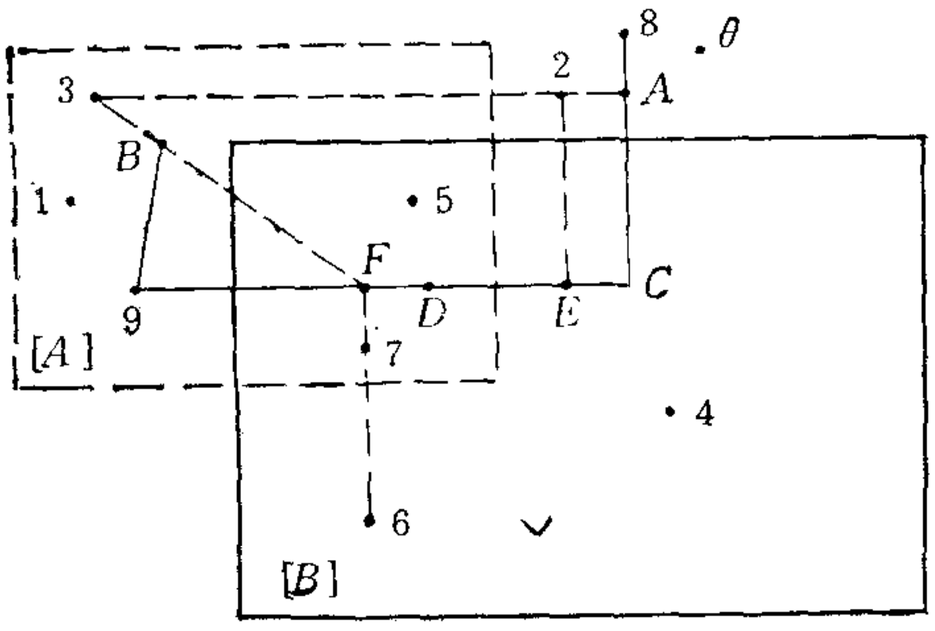

图 2　两层状态层次空间中状态分布图

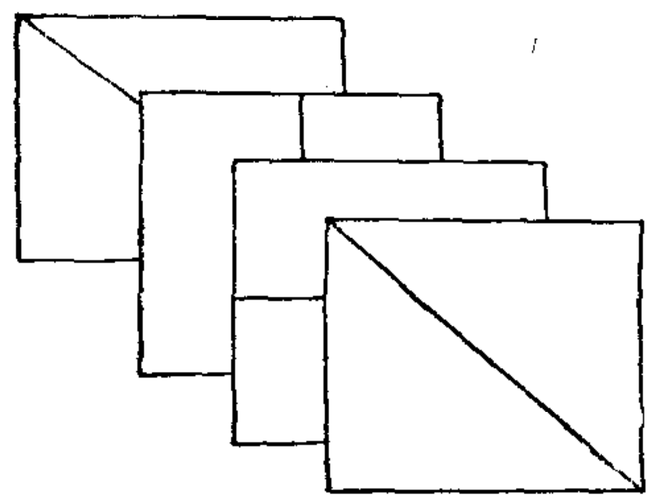

图 3　四层遮盖图形

利用上述消盖的方法可以省去透视表，直接修改代表状态值空间的二值表或彩色表中相应单元中的值(而上法图 1 中赋值表是固定不变的)即可实现．

又由于在实际情况中不存在文字出格态，查找表的实际有用地址只有 $N=I^L=3^4=81$ 个，而最大修改的单元从上述 192 个减到 54 个．再除去处于透视态的序列以及上下层相同状态的序列，因此每次平均只需修改 4—6 个单元即可．

现用一个只含两个状态层次空间的例子来说明赋值与透视变换过程，图 2 中层次 [*A*] 与 [*B*] 对应表 1 中的 *A*、*B* 集. [*A*] 层及其上的字形用虚线表示，[*B*] 层及其上的字形用实线表示. 数字 0 到 *F* 对应表 1 中的第一行中各个状态. 0 是背景，1 至 3 属 [*A*] 层与 [*B*] 层不相交，可看作 [*B*] 层处于透视态；4 至 *F* 属 [*B*] 层，4、8、*C* 三点与 [*A*] 层不相交，也可看成[*A*]层处于透明态. 2、6、*A*、*E* 是 [*A*] 层的文字出格态，它们在 [*A*] 的基底外面，并在 [*A*] 上直线的延长线上. 8、9、*A*、*B* 是 [*B*] 层的文字出格态，它们都不是正常情况. 5 点同时属于两层的文件基底，赋值为 1，呈白色；*F* 点同属于两层的基底文字态，赋值为 0，呈黑色. 这二点的赋值不随遮盖状态的变化而改变.

当 4 至 7 点取值为 1 时，[*B*] 层不透明，它遮住 [*A*] 层中与它重叠的部分. 如 *C* 至 *F* 取值为 0，则 [*B*] 层上有文字或图形. 当要实现透视时，只需把点 7 的值改为 0，再把 *D* 点的值改为 1，此时上述两层的重叠部分就变为 [*A*] 层挡住 [*B*] 层，完成透视与遮盖的转换. 所以两层时只要修改两点的数据即可完成转换，四层时转换情况较多，平均修改 4 到 6 个点也就可以了.

六、结 论

用本文所述的方法实现图形的遮盖与消盖，减少了大量数据传送时间，节约了许多存储空间，且透视转换时不必考虑图形内容，也不用记录各层遮盖边缘处的地址. 仅需一个 256 位的查找表，一般微机中的彩色表即可.

笔者在本所自行研制的图象处理器上很方便地实现了此方法，连同生成的简单图形与基底，只用了 200*H* 余字节的程序，比起纯软件方法来，除需要查找表外，唯一的使用限制就是层数与硬件有关. 一般的图象存储器为 8 至 12 位，可以显示 4 至 6 层窗口文件，完全可以应用. 图 3 是用来作试演的图形.

本文是在胡启恒所长支持下完成的，并得到戴国忠同志的协助. 对朱培基研究员、陈由迪研究员详细审阅此稿表示感谢.

参 考 文 献

[1] Macintoch 使用手册.
[2] 刘明业，计算机辅助逻辑设计理论，科学出版社，1985 年.

THE PRINCIPLE AND IMPLEMENTATION OF COVER AND TRANSPARENCE IN CAD FILE DISPLAY

LI WENYU

(*Institute of Automation, Chinese Academy of Sciences*)

ABSTRACT

This paper suggests the method and theory to implement cover and transparence of multi-layer files in CAD display, which can significantly simplify the program, decrease memory size, and increase switch speed.

Key words ——Cover; transparence; layer space; exposition.