CORP MODELLING USING FORMAL LANGUAGES

V. PREJMEREAN, V. CIOBAN and S. MOTOGNA*

Received: November 11, 1992
AMS Subject Classification : 68N20, 68Q52

REZUMAT. Modelarea corpurilor folosind limbajele formale. Definirea unui corp tridimensional se poate realiza de multe ori mai natural şi mai eficient utilizînd comenzi de deplasare a "cursorului" (vîrf de creion imaginar) în diverse direcţii. În acest fel se pot defini muchiile sau chiar suprafeţele unui corp, prin "plimbarea" cursorului pe direcţi paralele cu axele de coordonate. Pentru aceasta am definit un limbaj de comandă şi o serie de transformări elementare : translaţie, rotaţie, scalare, simetrie şi proiecţie pentru reprezentarea corpurilor descrise.

Solid corps modelling (tridimensional graphic objects) through front specification can be realised covering (describing) the edges ("wire-frame" method). This cover may be realised by moving commands of the cursor on a tridimensional frame this way: *up, down, right, left, front, back*. These moving commands compound the alphabet needed in the object outline description, which we want to model :

$$\Sigma = \{u, d, r, l, f, b\}$$

$w \in \Sigma^*$ is called command word of the cursor.

We will use for object definition an infinite frame of points obtained through intersection of the parallel plans with xOy, xOz and yOz at distance equal to 1. Two points $P(x,y,z)$ and $P'(x',y',z')$ are "neighbours" if we have the following relation :

$$|x-x'|+|y-y'|+|z-z'| = 1 \qquad (*)$$

This relation can be generalized :

$$x' = x + dx$$

$$y' = y + dy$$

* *"Babeş-Bolyai" University, Department of Computer Science, 3400 Cluj-Napoca, Romania*

$z' = z + dz$

where dx, dy, $dz \in \{-1,0,1\}$, $|dx|+|dy|+|dz| > 0$, which need an extension of the alphabet Σ by definition of other commands, obtained through composition of the existing commands.

We will use definition (*) to simplify grammars.

One command word $w \in \Sigma^*$ is, in fact, a succession of commands to the cursor and defines a "walk on the frame".

For a point $P(x,y,z)$ we define its succesors on the six directions this way :

- up : $SUC((x,y,z),u) := (x,y,z+1)$
- down : $SUC((x,y,z),d) := (x,y,z-1)$
- left : $SUC((x,y,z),l) := (x,y-1,z)$
- right : $SUC((x,y,z),r) := (x,y+1,z)$
- front : $SUC((x,y,z),f) := (x+1,y,z)$
- back : $SUC((x,y,z),b) := (x-1,y,z)$

The design specified in this way by a command word $w \in \Sigma^*$ is defined as follows :

$SUC : Z^3 \times \Sigma^* \rightarrow Z^3$

$SUC((x,y,z),\epsilon) = (x,y,z)$

and

$SUC((x,y,z),aw) = SUC(SUC(x,y,z),a),w)$

We notice that the cursor movements are executed from left to right. The cursor movement may be defined initializing the cursor in $P_0(x_0,y_0,z_0)$ and then moving through a command word $w \in \Sigma^*$: $MOV(w) = SUC((x_0,y_0,z_0),w)$.

A spatial graph unoriented described by

$w = a_1 \ldots a_n$, ($a_i \in \{u,d,l,r,f,b\}$, $i=1,n$)

is $g((x_0,y_0,z_0),w) = (V,A)$,

where $V = \{ \mathrm{MOV}(a_1 \ldots a_i) / \ i=1,n\}$

and

$A = \{(\mathrm{MOV}(a_1 \ldots a_{i-1}), \mathrm{MOV}(a_1 \ldots a_i) / \ i=1,n\}$.

We suppose that $\mathrm{MOV}(\epsilon) = (x,y,z)$.

A language formed of command words is called language of solid object description, and the grammar which generates the language is called solid forms grammar.

To represent the objects described by such grammars we will define functions for spatial transformations.

TRANSLATION :

$$\mathrm{TR}(g((x_0,y_0,z_0),w),(dx,dy,dz)) = g((x_0+dx,y_0+dy,z_0+dz),w) =$$

$$= g((x_0,y_0,z_0), S_x^{|dx|} S_y^{|dy|} S_z^{|dz|} w)$$

$$S_x = \begin{cases} f & dx>0 \\ \epsilon & dx=0 \\ b & dx<0 \end{cases}$$

$$S_y = \begin{cases} l & dy<0 \\ \epsilon & dy=0 \\ r & dy>0 \end{cases}$$

$$S_z = \begin{cases} u & dz>0 \\ \epsilon & dz=0 \\ d & dz<0 \end{cases}$$

So, if $V = (dx,dy,dz)$ is the translation vector then we may say that $TR(g(w),V) = g(S_x^{|dx|}\ S_y^{|dy|}\ S_z^{|dz|}\ w)$.

For rotations with 90°, 180°, 270° or even with 45°, 135°, 225° and 315° we can apply command word transformations depending on the axis of the rotation Ox, Oy or Oz. For example, rotation around Ox will change the word *w* so :

f, b - remain unchanged

u --> *r*

d --> *l*

l --> *u*

r --> *d*.

In the same way the substitutions applied to the productions rules of the grammar, for the other angles and axis are defined.

SCALATION :

A point $P(x,y,z)$ through scalation become $P'(x',y',z')$, where:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = f_g \begin{bmatrix} f_x & 0 & 0 \\ 0 & f_y & 0 \\ 0 & 0 & f_z \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}$$

If $f_g f_x,\ f_g f_y,\ f_g f_z \geq 1$ then the image grows and we will consider these scalar factors natural.

If $f_g \cdot f_x,\ f_g \cdot f_y,\ f_g \cdot f_z < 1$

$$f_g \cdot f_x = \frac{1}{f'_x}\ ,\quad f_g \cdot f_y = \frac{1}{f'_y}\ ,\quad f_g \cdot f_z = \frac{1}{f'_z} \text{ and}$$

$$f'_x\ ,\ f'_y\ ,\ f'_z \in \mathbb{N}$$

then the scalars applied to a command word $w = a_1 \ldots a_n$ will modify the word as follows :

a_i --> a_i^f , where

$$f = \begin{cases} f_g \cdot f_z & if\ a_i \in \{u, d\} \\ f_g \cdot f_y & if\ a_i \in \{l, r\} \\ f_g \cdot f_x & if\ a_i \in \{f, b\} \end{cases}$$

These transformations may be applied to the grammars which generate the command language.

SYMETRIES :

Reflections in planes xOy, xOz, yOz or in axis Ox, Oy, Oz are realised through such substitutions :

reflection in xOy: $u \leftarrow\rightarrow d$

xOz: $l \leftarrow\rightarrow r$

yOz: $f \leftarrow\rightarrow b$.

$SIM_{Oz}(g((x_0,y_0,z_0),w)) = g((-x_0,-y_0,z_0),w')$, where w' is obtained by substituting $l \leftarrow\rightarrow r$ and $f \leftarrow\rightarrow b$.

In the same way we construct reflections in Ox and Oy.

PROJECTIONS :

Projection parallel to Oz on xOy is defined as :

$$PR_{xOy}(g(w)) = \begin{cases} g(d^{z_0} w') & if\ z_0 > 0 \\ g(u^{z_0} w') & if\ z_0 < 0 \end{cases}$$

where w' is obtain by substituting "u" or "d" with ϵ.

We can realise images with solid forms grammars extending the alphabet Σ with simbols representing commands to choose (select) colours, where the null colour , "dark", shift the cursor without effective translation :

$\Sigma' = \Sigma \cup \{0,1,\ldots,7\}$.

In this case we consider that the initial position of the cursor is $O(0,0,0)$ and the initial colour is null. Then the word

$$(s_x^{|x_0|} s_y^{|y_0|} s_z^{|z_0|} c w)$$

draw the object $g(w)$ from the initial position (x_0,y_0,z_0) in the colour "c".

REFERENCES

1. F. J. Brandenburg, J. Dassow, *Reduction of Picture Words*, MIP 8905, Universität Passau, 1989.
2. H. A. Maurer, G. Rozenberg, E. Welzl, *Using string languages to describe picture languages*, Inf. and Control 54, 1982.
3. A. Rosenfeld - Picture Languages - *Formal Models for Picture Recognition*, Academic Press, 1979.
4. F. J. Brandenburg, M. P. Chytil, *On picture languages: Cycles and syntax-directed transformations*, MIP 9020 Universität Passau, 1990.