VOLUME : A COMPUTER PROGRAM FOR BUILDING MODELISATION AND MANIPULATION

Yves BERNARD, Ingénieur de recherche.
Albert DUPAGNE, Professeur ordinaire.

Laboratoire de Physique du Bâtiment
Université de Liège
Avenue des Tilleuls 15 - Bât. D1
4000 LIEGE (BELGIUM)

ABSTRACT

We describe a computer program VOLUME for interactive 3D modelling of building which can be used in many problems (heat losses calculations, architectural evaluation, ...) where a geometric representation is required. The program is more than a tool to input a geometry : it is also a tool for design; manipulations are very easy and many design solutions can be tried. We then discuss about developments for improving the quality of that tool in the design process.

Keywords : computer aided design, 3D modelling, computer-user interface, evaluations.

Data input and preparation is a bottleneck of many computer programs. This is especially true for programs needing data representing geometry of building (e.g. : for heat loss calculation prgorams ...). Moreover, in architecture (in fact in all disciplines dealing with design) data manipulations is very important : poor data manipulation slows or even inhibits the design process loop (analysis ⟶ design ⟶ evaluation ⟶ analysis ...).

VOLUME tries to be a response to this problem. It has been designed to easily and interactively build a 3D model of a building : the building is seen as a collection of polyhedrons (prism, pyramid, cylinder...). The elaboration of the building as a collection of simple volumes is one way of thinking which, we believe, is very understandable to the architects working at the early stage of the design.

These polyhedrons define the global 3D structure of the building : defining a building at the detail level would be impossible or very complicated if the computer (and the user) does not have a good idea of that structure; this model must be considered as a first step which, along the design process, should evolve towards a more detailed one.

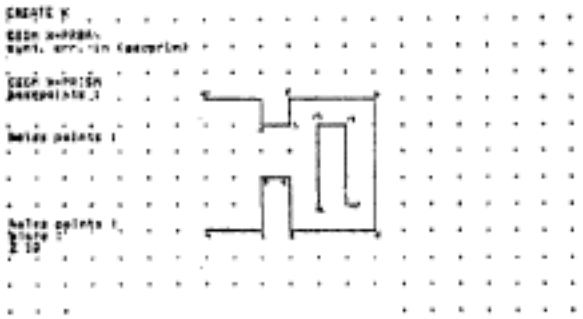
The user interface has been carefully designed :

- computer-user communication is by a simple command language :

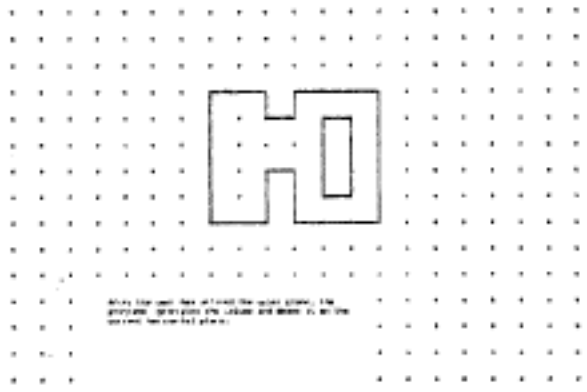
```
>?
list of commands :
CREATE     DELETE     DESTROY    DISPLAY    ESPFIL
EVAL       FILBIB     GEON       HPROJ      LIST
LOAD       LOCATION   NAME       ORIGINE
ORTHO      PERSP      POINT      RENAME
SAVE       SET        SHOW       STATUS
TWIND      UPROJ      ZOOM       END
WINDOW
To input an expression type  :  as first character ;
For help while typing a command, type ?  !
For help on the graphic syntax read the file syntax.coc
>

>SET ?
  sets values of system parameters :
    SET <param> [ = <num>]
    <param>:= ALPHA / BETA / DASHED / EFACTOR / EPS / FOCAL /GRID /
              GRIDSIZE / HIDDEN / LOCALPLOT / NOGRID / NOHIDDEN /
              FILPLOT/ NOFILPLOT/ WINDOW/ NOWINDOW/ ...
>

>PERSP ?
  perspective :
 PERSP <3point>  <3point> [<expr>]
       eyepoint  focuspoint
 PERSP will draw the perspective of the
 volumes defined by <expr>; if <expr> is not input, it
 will draw the volumes of the last <expr> input in a PERSP command
>
```

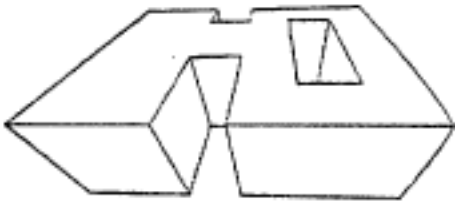
- a "graphic syntax" has been designed to input graphic items (points, lines, vectors, planes...) in a simple and flexible way;

- polyhedrons are created very simply by primitive commands creating different type of volumes with the minimum of information.



Creation of a prism X with one hole in its base. First, the external contour is input with the points 1, 2, 3,..., 11 and is finished with the terminator 0. Then the contour of the hole is input with the points 12,...,15 and finished with the terminator. The program then asks if the user wants to define an other hole. As the answer is no, the user types 0. The upper plane Z=10 is then defined.



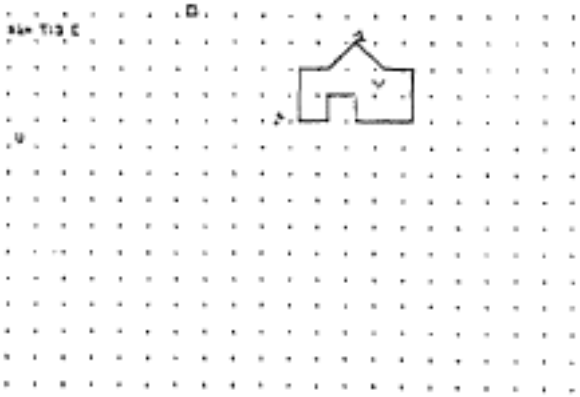


- geometric transformations (translation, scaling, symmetry, rotation) can be applied on sets of volumes : the user writes expressions (like algebraic expressions) built with graphic and set operators (union, intersection, difference) applied to set of volumes. The result of an expression is always a set of volumes and so the definition is recursive. This resulting set can be stored in a "variable".

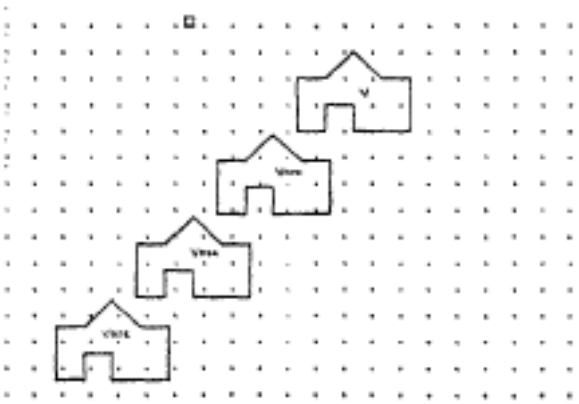
  exemple :

  - the expression SA=V+W+U defines a variable of name SA which is a set or collection of 3 volumes : the volumes V, W and U.

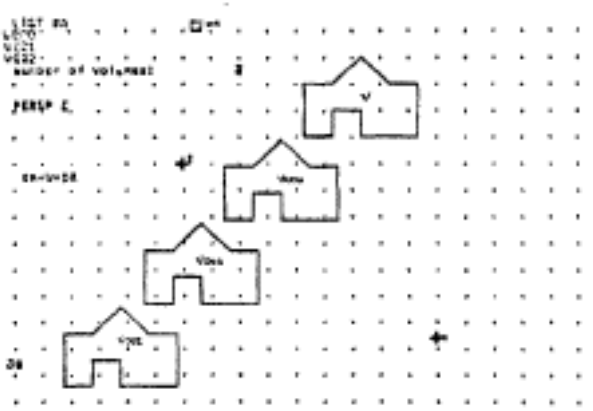
- the expression SB=I:2 seg SA will produce a set of 6 new volumes which are copies of the volumes in SA but translated 2 times with the segment seg (seg means that we have to define a graphic segment, and many ways exist for that)
- exemples in figures.



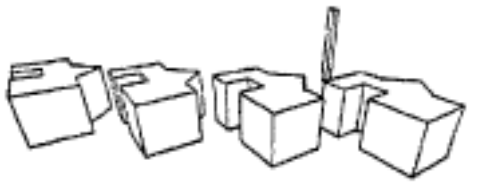
Translation with repeater.
The volume V will give 3 new volumes which are copies of V, translated successively by the segment 1→2.
The expression I: 3 (seg) V gives a set of 3 new volumes and this result is put into the variable SA.



Result. The program gives names to the 3 new volumes : VOL0, VOL1, VOL2.



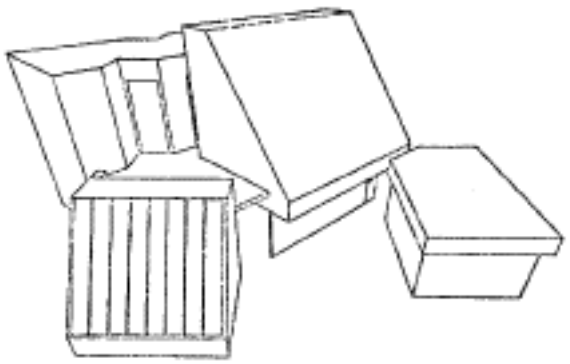
5. Content of the variable $A is asked with the LIST command.
A perspective is asked for the volumes in the expression $A+V+OR.
The eye point is the point 1 (2 * 30) and the focuspoint is
point 2 (2 * 0).

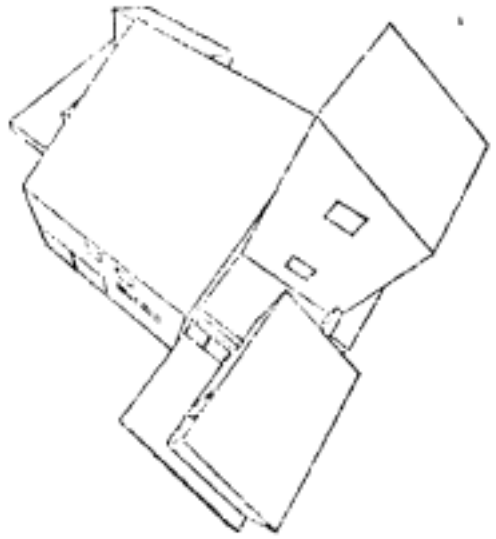


the concept of sets of volumes (or "variables") will be used to define and easily modify other attributes of volumes (constr. data, human data, ...);

openings (doors and windows) of any shape can be created in faces of volumes with the same flexibility and facility than for volumes in space;

at any time, the user can ask for perspectives or other types of projections with or without hidden lines removed.

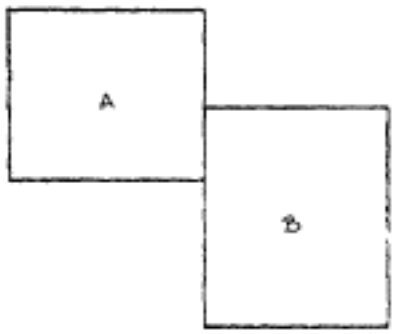


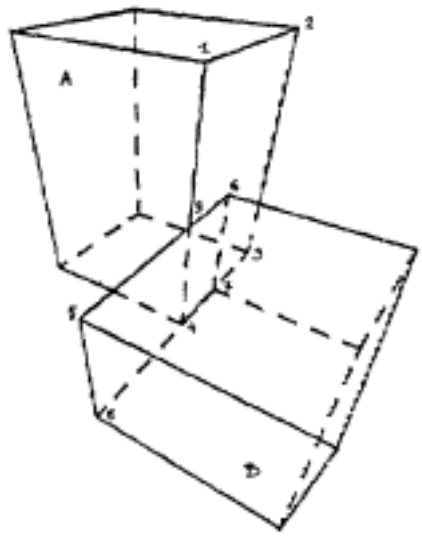


- evaluation functions on different subjects will be written; for now on, area calculation (floor, wall, roof, window areas) and volume calculation are implemented. In fact, the data base built in the computer memory is accessible to any user module which could be called to perform peculiar evaluations. Somebody with some experience in computer programming could easily add new commands and functions;

- files describing the volumes can be produced (e.g. for the ESP program).

Once the user is satisfied with a design, with one collection of volumes representing the building, the program is able to find adjacency or contact between faces of volumes and to decompose the 2 connected faces in simple faces : faces which belong only to one volume or to the intersection of 2 volumes faces.





So the data base built after this adjacency analysis and face decomposition process represents the building geometry and the geometrical relationships between the volumes : the building is now a collection of spaces (volumes) limited by faces (walls) and a collection of faces, each face being the separation between 2 volumes or one volume and the external space. This information allows more elaborate evaluations (area calculation, circulation, ...) which can help the user in judging his design.

This data base is a first step for the elaboration of a more complex geometric model : routines will be added to manipulate (add, delete, move) the faces or walls. So as the design goes on, the model will become more precise and detailed.

From this data base, files can be produced for various types of evaluation programs. Fors instance, many

heat losses calculation programs require a set of walls (each wall being described by some geometric parameters and thermal data and by the adjacent spaces) and a set of spaces with their thermal (desired temperature, occupancy, ...)parameters. This kind of files is easy to produce and routines can be written to produce many other types of files.

DEVELOPEMENTS

VOLUME claims to be a real design tool for architects : not only a program producing geometric data files for peculiar representation or evaluation of completed designs.

It is intended to be used throughout the architect's work and particularly in the first stages of the design process where major decisions, influencing the final performances, are taken very often on irrational basis.

Using VOLUME would be a process of gradual enrichment for the design through which the architect would organize his ideas, clarify his concepts, discover and explore his own way of thinking.

Therefore a real interaction between man and machine working at very high speed is obviously essential to bring efficient aid to the architect involved in the creative process of designing.

The computer is not solely a support for the architect's ideas (like a sheet of paper for the traditional architect) but is also a "thinking machine", able to perform analysis on the proposed design and to communicate judgements to the user.

For the purpose, developements have to be undertaken in the following directions.

1. In architecture, a form is more than the sum of its elementary parts. VOLUME, at its present state of development, doesn't allow to take care of the polyhedrons' organization,of the super structure giving its architectural signification to the whole.

Therefore VOLUME has to be expanded in two ways

- New attributes must be attached to elementary polyhedrons. These attributes (axis, center, plane of symmetry,roof height and slope, global floor area...) are closer to architectural thinking. They add constraints to the each polyhedrons and particularize them.
  Attention must be paid to alteration effects induced from working operators, either they are recognized and accepted or refused.
  New operators dealing with this attributes have also to be added.

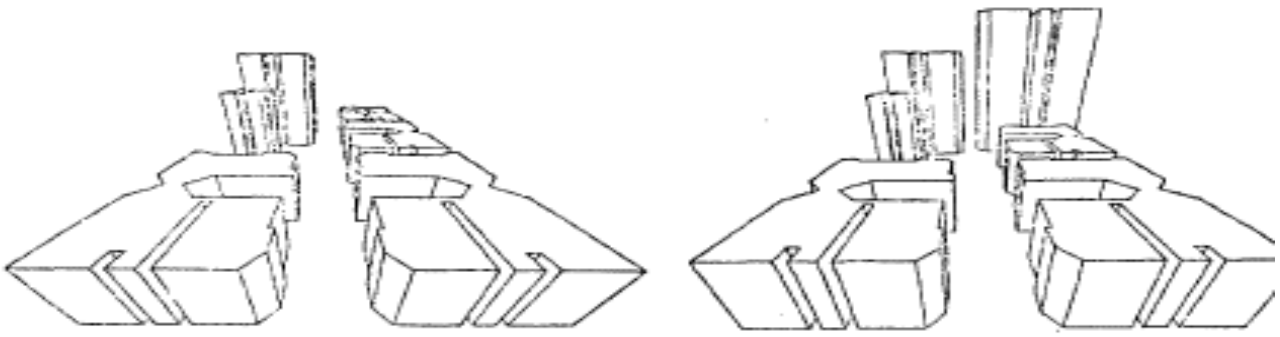
- Ways should be found to express the relationships defined between volumes and the specific structure of their layout (alinement, axis, grid ...)

These two ways expanding VOLUME should allow to define some specifications on elementary volumes avoiding the complete definition of its final shape or geometry. By this way, the design project would be defined at a very high level (and at a very early stage of the design)without being hampered by a precise geometry. The user is allowed to express his first idea of what the volumes and their organization have to be even when they are poorly defined and if the project is judged interesting then he can go on in the geometry definition process.

2. The program now deals mainly with geometric data. Routines have to be written to easily input and structure non geometric data (cost data, occupancy data, light data, confort data...) so that the user can roughly describe the building, its general frame. Default or suggested parameter values should always be proposed according to the type of building, (houses, offices,...); this frame and its parameters can in fact represent a part of the current architectural knowledge.

3. More evaluation functions should be implemented (thermal costs, privacy, circulation, check agains regulations rules... and the presentations of results (drawings, colors) sould suggest ways to solutions (e.g. showing walls with important heat losses). These functions results should be checked against the default or input values describing the general frame defined previously. (e.g. : suppose that in the general frame specifications we said an office building with 200 employees ;we then define the geometry. If the computer finds a floor area of 200 sq.m2 for that geometry, it has to warn the user). These evaluations functions constitute the tools with which the user can make a judgement on his project.

4. A site description module should be added. Then theprogram could deal with urban problems (site analysis, solar mask analysis, circulation and view analysis,...). In fact, with VOLUME it is very easy to define urban groupings.



IMPLEMENTATION

The program is written in PASCAL and FORTRAN. The FORTRAN part is due to historical reason : the hidden line removal routine is a modified version of the BIBLE program from ABACUS group and is written in FORTRAN (the number of source lines is about 4000 lines in each language).

The hidden line removal routine should be replaced with an hidden surface elimination algorithm allowing color rendering of perspectives. Routines will be added to deal with color terminals (AED 512 and 767). The FORTRAN part, if we have time, should be slowly coded into PASCAL.

For now, the polygon routines (polygon intersections and differences) needed in the adjacent faces decomposition problem have been written and are inserted in the program.

The program runs on a VAX 11/780 with tektronix series 4010 terminal or visual 500 (raster-scan) terminal. It can handle complex scenes of one or more hundreds of volumes and perspectives with hidden lines elimination are computed in time of the order of a few minutes cpu.

AKNOWLEDGEMENTS

This program has been developped under a research contract sponsored by the "Services de la Programmation de la Politique Scientifique" of the Belgian Government.