SOLID MODELING AND COMPUTER GRAPHICS RENDERING OF THE COLISEUM IN ROME

A. Paoluzzi*, D. Del Bufalo*, A. Nistri* and M. Rosina**
* Department of Informatica e Sistemistica, University of Rome "La Sapienza"
Via Buonarroti, 12, 00185 Roma - ph. +39-6-7312367
** OMI - Ottico Meccanica Italiana, Agusta Group
Via della Vasca Navale, 79, 00100 Roma - ph. +39-6-55421
Italy

Commission V

Abstract

The computer technology allows nowadays to substitute the traditional plotting of object's edges from a stereoscopic model with the rendering of a topologically and geometrically "complete" computer description of surfaces and solids. This paper therefore describes a set of work items from a large project of survey of the Coliseum in Rome, recently started. In particular, are here discussed the relational data structure designed to accomodate the wire-frame model of the monument and the main problems and solutions foreseen to draw up a solid model of a monument of complex structure and of very large size.

1. INTRODUCTION

A large project, described in [BIR88], of photogrammetric acquisition of the ancient and world-wide famous Coliseum in Rome, started recently. Inside this project, the goal of our group is the derivation of a solid model of the building. The aim of this paper is therefore to describe some guidelines that will be followed in the construction of a "complete" solid model of the Amphitheatre. This model will be subsequently used to accomplish a number of complex tasks: to test archeological hypotesis; to perform visual simulation of the ancient shape of the monument; to perform various kinds of structural computations and simulations, in order also to prevent futures injuries.

An overall view of the project is given in [BIR88]. In that work can be found a sound description of the design of the topographic network and of the technical issues of photogrammetric rendering. In this paper we discuss some computer science aspects of the project. In particular, after having given some historical notes about the monument, we estimate the size of the photogrammetric rendering and show the design of the data structures prepared to accomodate the acquisition data. Then we give some basic concepts on solid modeling and discuss the main problems to be tackled in modeling and rendering a so large building. Various extensions to our solid modeler MINERVA are outlined in the second part of the paper, concerning fast processing and rendering of very large models.

It is our opinion that solid modeling, closely coupled with photogrammetric acquisition techniques, is mature enough today to be extensively used in survey and rendering of monuments. Our challenge is to demonstrate this in the next future.

2. ARCHITECTONICAL AND HISTORICAL NOTES

2.1 Hystorical notes

The Coliseum was built in the area previously occupied by the lake of Nero's "Domus Aurea", right in front of the Temple of Venere and Roma. Emperor Vespasian started at 72 a.C. the building, but he died before it was completed. Only in 80 a.c. emperor Titus celebrated the big opening, killing more than 5,000 animals.
The history of the Flavian Anphitheatre is rich of happenings, destructions (eartquakes: 429, 442, 1231, 1255, 1349 a.C.), fires, restorations. The last "beasts show" was made under the rule of Theodoric in 523 a.c., after that the Coliseum was left to ruin for many centuries. In 1084 Rome was in a big fire and most of its monuments were destroyed: at that time the Coliseum became property of Frangipane Family and it was used as a fortess until the 14 th. century.
Pope Martin 5th in 1420's confered the anphitheatre to a Convent with the right to use it as a quarry for marble and building material. Since then and during all the Renaissance the Coliseum was the richest quarry around to be used to build the new Rome of the Popes.
The first studies and measurements on the monument were made by the architect Domenico Fontana, who designed, for Pope Sixtus 5th, a restoration that was never done. Another study and metric drawings were made by Serlio in 1544, who was the first one to draw all the different elements and orders of the monument.

In the seventeenth century was commissioned to Bernini a design for the restoration of the Coliseum as a Christian Temple. But his project was never realized, like another one by Carlo Fontana, who designed the most accurate plan of the Anphitheatre, for the first time using different colours, that showed the still up structures existing in the year 1708.

In 1798 C. Lucangeli executed the first systematic excavations inside the Arena. He discovered most of the underground rooms and built a wooden model of the monument which is still visitable inside the Coliseum. After Lucangeli, the best survey and drawings were made by the architect L.J. Duc who was one of the envoies from the Ecole des Beaux-Arts of Paris in 1830.

In the end of the past Century many studies and scientifical surveys were done by Rodolfo Lanciani, who left us one of the most precise topographic plan of the archeological sites of Rome. After him, at the beginning of this century, more studies were conducted by Prof. Lugli, but the Coliseum was never surveyed with precise instrumentation in our technological era.

2.2 The monument's structure

The Coliseum has the shape of an ellipse; its major axis measures 191 meters, the minor 158, its maximum hight is 49 mt. The two axis divide the monument into four identical quadrants, each one of them including 20 pairs of pilasters by three superimposed orders: the first order at the bottom is Tuscanic style, the second order is Ionic, the third one is Corinthian; the last ring at the top is attic decorated with Corinthians strips. A square window is opened between every intercolumnate. The external skin is therefore composed by three vertical orders plus the attic, repeted 80 times around the ellipse.
Each of the 80 arches is an entrance to the Anphitheatre, being they numbered progressively with Roman numbers (of course!) carved on the marble ontop of the arch, where they can still be seen. The spectators had thickets "tesserae" with the number of their correspondent entrance and seat. Such entrances are divided in two stairwais made in form of a rampant circular vault which brings through 160 outlets into the arena seats.
The number of seats avaliable in the Coliseum is hard to compute today, but the most realistic evaluations go between 50,000 and 75,000 seats. Despite this big numbers of spectators the Anphitheatre could be evacuated within just 30 minutes.

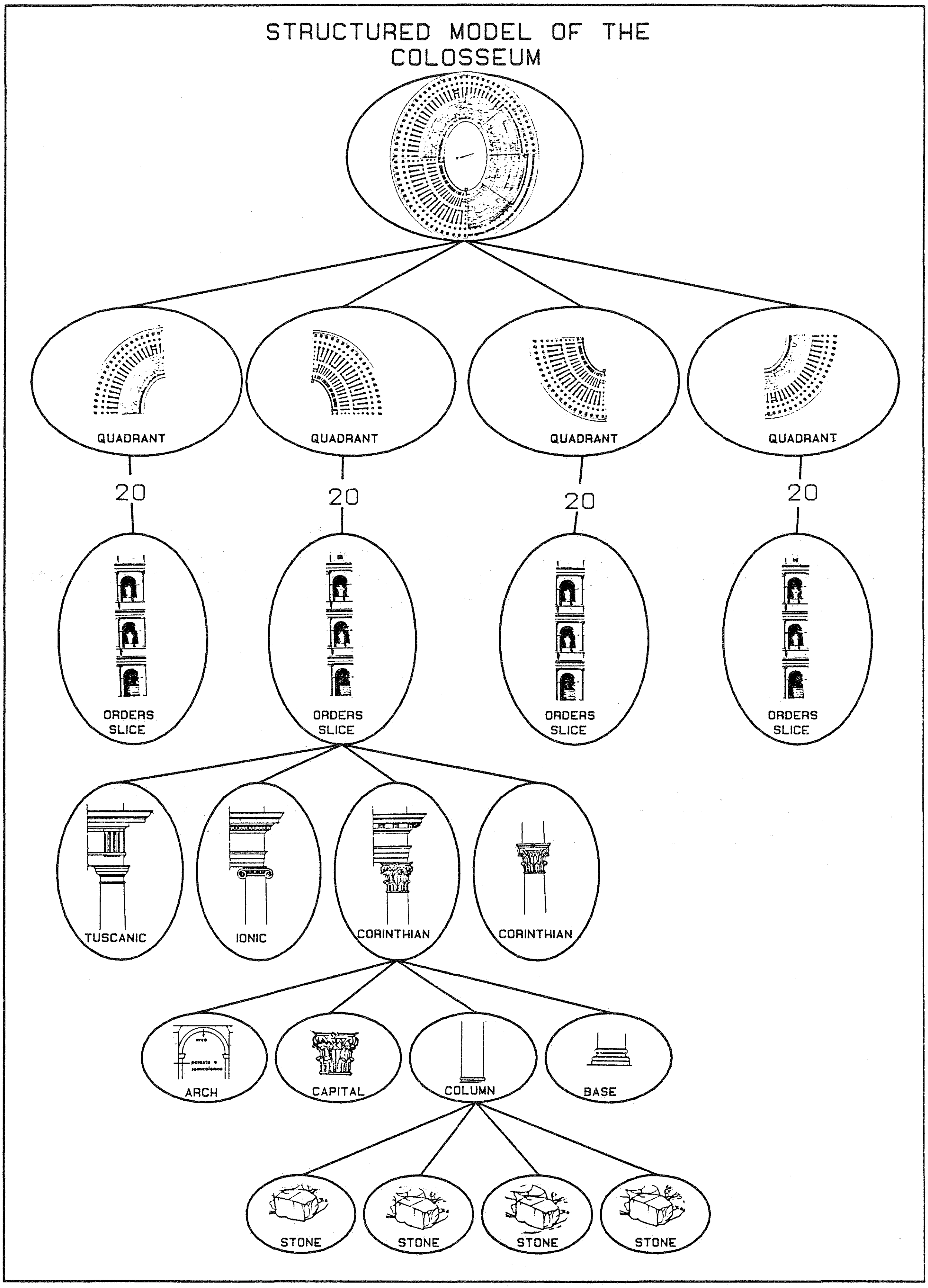

fig.2.1: The Colosseum structure

3. PHOTOGRAMMETRIC ACQUISITION DATA

3.1 Acquisition characters

In this section we will summarize some notes from [BIR88] about photogrammetric acquisition characters. The internal facings survey has similar caracters to the external one; for the aerial plan survey will be adopted a medium photographic scale of 1:660. According to explorations, a centimetric precision of the survey is expected [BIR88].

Ellipse:	larger semi-axis:	$a \cong 95$ mt.
	smaller semi-axis:	$b \cong 80$ mt.
	eccentricity:	$e \cong 0.2909$
	ellipse lenght:	$L \cong 550.8$ mt.
External facing E_1	height:	$h_1 \cong 50$ mt.
	lenght:	$l_1 \cong 239$ mt.
	surface:	$se_1 \cong 11{,}950$ mq.
External facing E_2	height:	$h_2 \cong 22$ mt.
	lenght:	$l_2 \cong 311.8$ mt.
	surface:	$se_2 \cong 6{,}860$ mq.
Internal facing	surface:	$si \cong (se_1 + se_2) * 0.7 = 13{,}170$ mq.
	facings survey:	100 * 200 points/mq.
Planimetry	surface:	$sp \cong c * a * b \cong 23{,}880$ mq.
	plan survey:	40 * 70 points/mq.

fig. 3.1: Monument's dimensions

External facings survey:

- camera focal lenght: mm. 110;
- photograms original size: 4 x 5 inches;
- camera - objet distance: 30 mt.;
- number of strips: 2 around the monument; a "low" strip taken on a tripod and a "high" one taken with ladder truck-elevator (about 20 mt. above ground level);
- number of photograms for each strip: 60;
- covered surface: a quarter of ellipse (2 x 15 photograms);
- medium photographic scale: 1:300.

Photogrammetric acquisition methodology can be schematized as an input of *polyline sequences*, oriented to the real-time generation of a sequential file containing a list of points 3D, separated by control-codes PEN_UP / PEN_DOWN, corrisponding to the instructions BEGIN_POLYLINE / END_POLYLINE. The control code PEN_UP / PEN_DOWN, given by footswitch or handswitch during photogrammetric acquisition, will be coupled with another numeric code: such second code will represent the layer, type, color, width, and others logical and graphical attributes of the surveyed polyline.

3.2 An estimate of acquisition data size

A quite approximate evaluation of the number of 3D points returned by photogrammetric acquisition, can be obtained by estimating the dimensions of the monument (see fig. 3.1.). The surface density of recorded points is therefore derived as follows.
Firstly, it is convenient to subtract the arch surfaces from overall calculation of surveyble facings surface, as they are empty areas. Each one of 194 archs surface measures about 26 square meters, so we have the working surface of external facing = $se_1+se_2-194*26$ = mq.13,810, and the working surface of internal facing = 13,810 x 0.7 = mq. 9.670. Hence we have the following estimates: number of facings points = 2,348,000 ÷ 4,696,000; number of plan points

= 955,000 ÷ 1,671,000. Therefore we estimate that the photogrammetric acquisition of the Colosseum will return a quantity of about 5 million points.

Photogrammetric survey precision is characterized by square medium errors of about 1 ÷ 2 cm. on x, y and z co-ordinates; the extension of the external facing, developed in plan, does not exceed mt. 500 x 50. If we express the coordinates in centimeters (integers), then the abscissa cannot assume more than 50,000 different values. For this reason will be convenient to use 16-bit integers (2 bytes for each co-ordinate) that can assume $2^{16} = 65{,}536$ different values. The expected memory occupation for points will then be 5,000,000 * 2 * 3 = 30 Megabytes.

3.3. Design of the acquisition data structure.

Data from the photogrammetric acquisition will be managed by using a logical data structure performing the following requirements:
- it must be "filled" in real time with photogrammetric acquisition data; in alternative it must be created, from the same photogrammetric data, in linear time with the input dimension;
- it must be founded, as much as possible, on the relational model, but without compromising the storage as well as the retrieval efficiency;
- metric informations, i.e. coordinates of surveyed points, should be easily separable from topological and graphical ones (point / vectors adiacencies, line size / width / color etc.), to achieve a more efficient data management.

With reference to the polyline sequence structure, suggested in the previous section, we can divide polyline points into 2 classes as follows:
nodal points: The extreme points of a *polyline*. A nodal point can be either the initial or the final point of one or more polylines. On a nodal point can either incide a single vector ("suspended" or "dead-end" point) or 3 or more vectors ("crossing-nodal" point);
intermediate points: The internal points of a polyline: they can be included between 2 nodal points, or between a nodal point and an intermediate point, or else between two intermediate points. An intermediate point belongs to a single polyline. Over an intermediate point incide always 2 vectors belonging to the same polyline. A *vector* can be regarded as a particular case of a polyline without intermediate points. A *simple quoted point* can be regarded as a vector with coincident extremes.

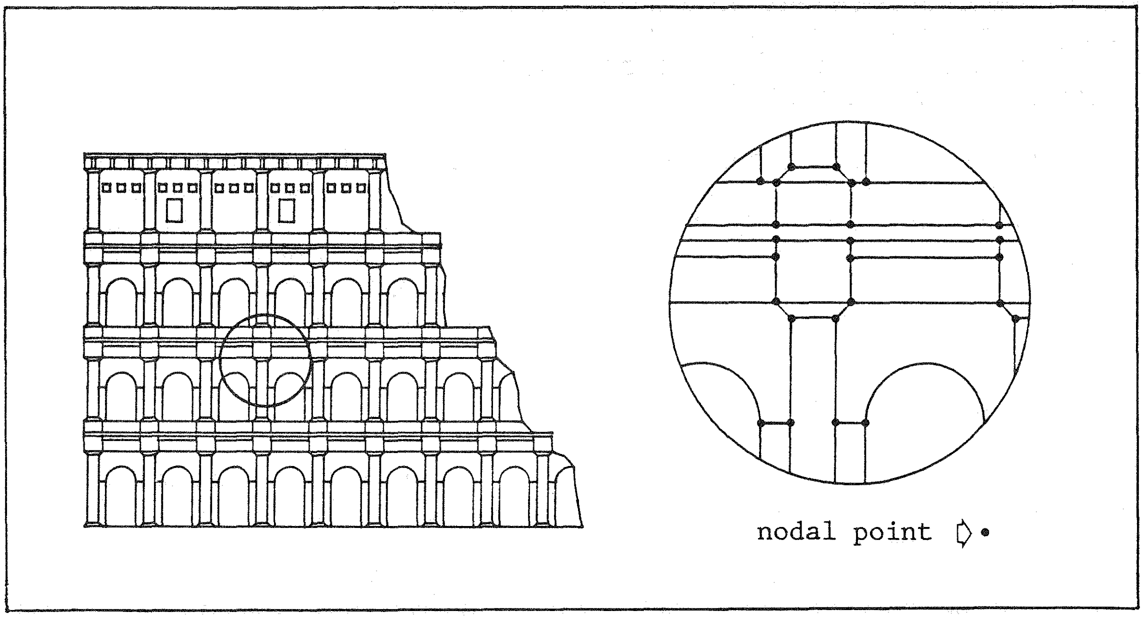

fig 3.2: Nodal and intermediate points.

In order to satisfy the above requirements, the relational database design of the storage structures for the data describing the monument from the photogrammetric acquisition will be founded on the following tables:

I. *Nodal points Table* (NODAL_POINTS)
with 4 fields:
ID: unique access key of the nodal point;
XN,YN,ZN: triple of fields containing nodal points coordinates;

II. *Polylines Table* (POLYLINES)

with 12 fields:
IDP : unique access key of the polyline;
ID1,ID2: identifiers of polyline extreme points: they represent the access keys to NODAL_POINTS and constitutes "join" attributes of POLYLINES table with the ID field of NODAL_POINTS. If the polyline in question is a simple quoted point, or a "single-loop" polygon, then ID1 will be equal to ID2;
INTACC: address to a random-access file containing all the intermediate points of each polyline. If the polyline is a simple quoted point or a vector, then INTACC will be equal to zero. Record No. INTACC of such file will contain the second point of the polyline, while record No. INTACC+NUMINT-1 will contain the last but one. First and second (nodal) points are located by ID1 and ID2 on NODAL_POINTS table;
NUMINT: number of the intermediate points of the polyline. If the polyline is either a simple quoted point or a vector then NUMINT will be equal to zero;
LTYPE: polyline line type (continuous, dots, etc.);
LWIDTH: polyline width;
LCOLOR: polyline colour;
SPLINE: boolean field to activate (ON) or not (OFF) an algorithm for polyline spline drawing;
LAYER: code representing the layer of the polyline;
SURTYPE: survey to which the polyline belongs (EXT = external facing survey; INT = internal facing survey; AER = aerial photogrammetric survey);
STERID: identifier of the stereoscopic model from which the polyline has been surveyed.

III. *Intermediate point File* (INTERM.FIL)

Random access file containing all the intermediate points of the polylines.

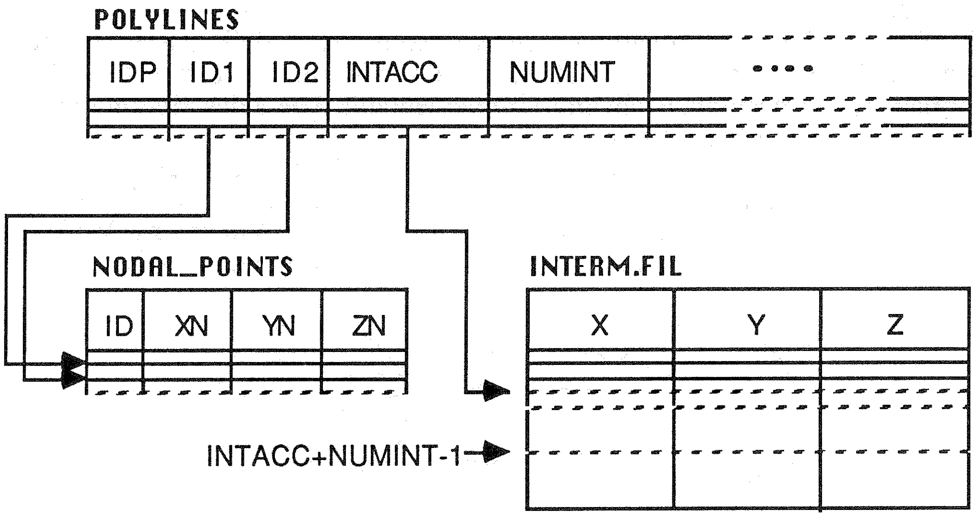

fig. 3.3: Relational structure for acquisition data

The presented data structure easily solves the collision problem, i.e. the case for which two or more polylines incide in a same point. In this case, in fact, the polylines in question will address the same element on the NODAL_POINTS table [NIST88].

INTERM.FIL random access file corrupts the theoretic "normalization" of a relational database design. In fact INTERM.FIL is not a relational table (in proper sense), but simply a random access file, where a record ordering is defined. Such ordering is necessary to represent the topological adjacency existing between contiguous points of a polygonal line. Such choice aims to represent efficiently the *polyline* graphical primitive, which is a dynamical structure of variable size, defining an ordering on its points. A polyline structure can be implemented neither in anyone of the three Normal Forms, nor (all the more reason) in Boyce-Codd Normal Form [CODD81]. The remaining two tables are in BCNF, because the only existing functional dependences determine the implication of the key of the table, over all its other fields [ULLM82].

A relational structure gives an efficient support to the spatial search for a polyline in the database. From this point of view it is very useful to add to the POLYLINES table a group of 6 fields <Xmin,Ymin,Zmin, Xmax,Ymax,Zmax> representing the overall dimension of the polyline (corresponding to a 3D containment box).The search of the polyline nearest to a point, supplied as the input of a query, can be carried out by exploring only POLYLINES table, and addressing INTERM.FIL when the point given as input is contained in its corresponding containment box. Calling N the average surveyed point number, and M the average number of points belonging to each polyline, the search time will be reduced from k * N to k*N/M, where k is a convenient constant. In order to obtain a still faster answer to any geometrical query, data will be stored as a "grid file". As we will see in a following section, such approach lies in partitioning the 3-space into a number of convenient cuboids parallel to the coordinate frame, indexed and selected by three integers. Such space partition allows to efficiently solve computational problems of very large dimensions.

4. SOLID MODELLING

4.1 Generalities on solid modelling

Capability to create and manufacture solid representations is essential to fully understand the geometry and structure of physical objects. Geometric Modelers are software systems for the definition and the manipulation of three-dimensional objects. The user of a geometric modeler is generally able to define curves, surfaces and solid objects, via graphical interfaces or special-purpose languages, and, in order to get easily new ones, to trasform and combine previously defined objects.

Solid description of objects is prevailing in recent geometric modeling systems; the trend is towards systems able to support any kind of graphical rendering and (conceptually) any function (of transformation, composition, evaluation) implementable over a description topologically and geometrically "complete" of the object.

A *representation scheme* is defined [REQ80] as a mapping from a set of geometrical models to a set of symbolic representations of objects. Representations are informationally "complete" when they correspond to only one object in the domain of the scheme. Geometric Modeling Systems are generally complete but no unique, because various representations of a given model may arise.

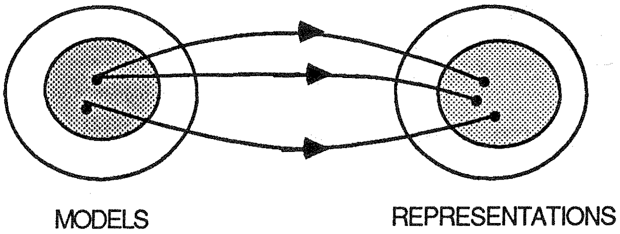

fig. 4.1: Representation scheme

We are planning to use in the Coliseum project the solid modeler MINERVA, currently under development at the Department of "Informatica e Sistemistica" of the University of Rome[1][PAOL88b]. This modeler allows for the use of a large set of primitive solids, bounded by a set of curved surfaces represented by parametric equations. In particular the modeler is able to represent the set of models composed by all the disconneted, non convex, unbounded and non manifolds polyhedrons. Minerva models may be "compressed", allowing for the hierarchical definition of complex structures.

The modeler uses, primarily, a boundary representation scheme. Boundary representation schemes are mappings from a set of geometric models to a set of topological and geometrical relationships among their vertices, edges and faces. Minerva boundary representation scheme is called "winged-triangle" [PAO88], as it is based on a "well-made" triangulation of the object surfaces, which yields a relational representation having tuples of a constant lenght. This representation allows to deal with multi-shell polyhedrons, whathever may be the topological genus of their boundary surfaces. The winged-triangle representation is a space-optimal linear approximation of solids with curved boundary. Each triangle of the approximated surface is described by the three triangle adjacencies and the three incident vertices (see fig. 4.2).

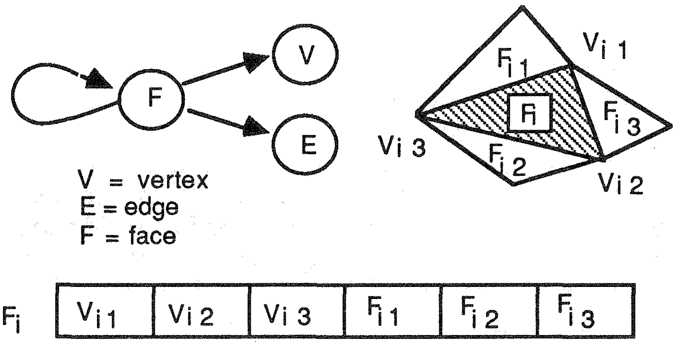

fig. 4.2: winged-triangle representation

4.2 Model decomposition

Minerva models are organised into "structures", which can be related to each other hierarchically. Each named part of the model is called "structure" (according to the ISO DIS [PHIGS87]); each structure is a collection of pointers to surfaces or polyhedrons. In a hierarchical description, only low-level components are extensively described, while complex objects are defined by the set of references to their component parts (see figure 4.3).

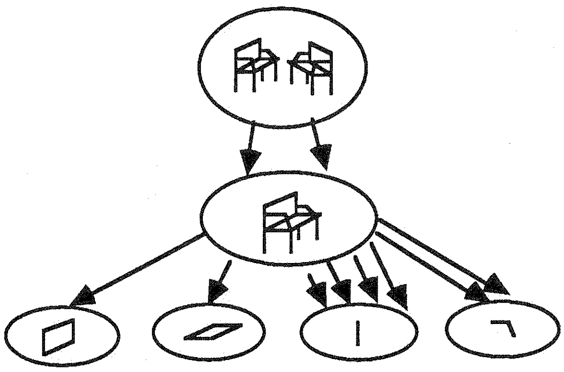

fig. 4.3: Hierarchical description

---

[1] The name Minerva, ancient italic god, is derived from the symbol of the University of Rome "La Sapienza".

MINERVA hierarchical descriptions may be of arbitrary depht and include data sharing among structures. Recursion is not allowed; therefore the structures associated to a complex object are representable as a direct aciclic multigraph. A transformation matrix (performing translation, rotation or scaling) is always associated to each arc of this multigraph model. Such matrices will affect (directly and indirectly) all referenced data.
Each surface or polyhedron is stored using a local coordinate frame, and it is transformed in the coordinate frame of the calling structures when the whole multigraph is traversed for visualization. An effective generation of all elementary data is therefore performed only at structure traversal time. Such a sort of data base is expecially suitable to store, without redundance, the geometrical models that can be decomposed in a lot of similar parts. Each part needs to be defined only once, and it is instantiated by the application of the linear transformations (i.e. rotations or scalings) in the hierarchy.

5. ALGORITHMS FOR MODEL RENDERING

5.1 Model generation and surface fitting

The Coliseum structure is very simple and highly repetitive. The whole monument is composed by four symmetric sectors, each sector beeing in turn made of twenty slices of four orders (see fig. 2.1). Each slice is different from the others of the same sector, having a different internal organization as well as a variable geometrical envelope. As we have seen, our goals is the generation and rendering of a solid model of the Coliseum, using a general purpose solid modeler and starting from data derived from the photogrammetric restitution. Depending on simmetry, only twenty slices need to be actually described as solid models. All the curve surfaces bounding each monument slice will be generated by using a parametric approach. A specific parametric equation for each surface will be extimated by extracting a suitable set of points from the optical model of the monument.
Various methods of curve and surface fitting are currently under study. Our first hypothesis concerns the use of bicubic B-splines over a suitable grid of points from the monument.
A parameter extimation of low-degree algebraic surface patches (defined in implicit form) matching, under some tolerance, the previously described parametric surfaces will complete the model building phase of our project. To derive implicit algebraic surfaces defining the monument boundaries is highly desiderable, as they probably better fit the ancient shape of the monument.

5.2 Grid file

We have estimated in section 3 the exspected quantity of data to be derived from the photogrammetric rendering of the Coliseum. Both the data from the photogrammetric rendering and the data describing the geometric model of the monument will be stored as a *grid file* in order to gain easy access to the data and fast answers to any geometric query.

In such approach all geometrical primitives (polylines and triangles) are intersected with a 3-dimensional grid of orthogonal planes, partitioning the 3-space. Each class of such partition, coincident with a cuboid parallel to the coordinate frame, is indexed by three integers; the indexes are used as data base keys to gain direct access to all the geometrical data contained in a given element of the space partition. To set up the space partition for the Coliseum project we shall probably use, according to [BIR88], cylindric coordinates or, better, "elliptical cylindric" coordinates <s,h,q>, where s is the curvilinear elliptical abscissa, h is the height and q is the depth (the "quote").

The grid file approch requires some additional storing cost for the geometrical data base. Conversely, it allows to tackle and to solve computational problems of very large dimensions, as the solid rendering of the Coliseum, involving some million points and many thousands of surface patches.

5.3 Boolean operations

The main operations performed by a solid modeler are the union, intersection, difference and complementation set operations. In particular, solid modelers using a boundary representation scheme usually perform such operation with an algorithm based on the intersection of boundaries of the considered objects. Such intersection requires (in the worst case) a time proportional to the square of the input size, i.e. to the number of surfaces describing the boundary of the objects.
Consequently, the MINERVA representation of solids will be extended to mantain a mixed "boundary and decompositive" one, in order to be efficient in solving intersection and rendering problems of very large size. As a matter of fact, if the boundary surfaces are spatially partitioned and stored in a grid file, great improvements in computing time can be obtained, both in performing boolean operations and in solving high level graphical rendering problems, as we shall see in the following.

First of all, a grid file approach allows for a great improvement of algorithms for the intersection of object's boundaries. In particular, in performing the union of two boundary representations of solid models, both spatially decomposed by using the same grid file scheme, will be sufficent to intersect only the pairs of boundary surfaces contained in corresponding grid sectors, i.e. in sectors bounded by the same grid planes and then having the same data base access key. Notice, in fact, that the intersection of other surface patches is not possible.
The following is an example of the computational advantages of such approach. Let we have to perform the union of two models bounded by N polygons, requiring in general N^2 intersections. If the polygons are spatially partitioned and stored in a grid file of n^3 cells, then it will suffice to perform $n^3 (N/n^3)^2 = N^2/n^3$ intersections. If $N = 10^6$ and $n = 10^2$ then we have 10^{12} versus 10^6 intersections.
Other strategies for improving the computational performance of basic intersection algorithms used by the geometric modeler MINERVA are currently under study. For example, improved algorithms will be used for intersection of surface patches belonging to the same grid file cell. In particular, both a presort in the three directions of space, and the use of a binary tree based on the BSP (Binary Space Partition) of Fuchs [FUCHS83] seem to give a time performance of O(n logn) for basic intersection algorithms.

5.4 Clipping

Each picture of the monument's solid model will be generated by using the camera reference model (Core System [ANSI81]), based on a three dimensional clipping to the "view volume". This volume may be either a cuboid or a truncated piramid, depending on the desired projection. In particular, orthogonal or oblique cuboids achieve ortographic and assonometric projections, while a truncated orthogonal piramid realizes a perspective projection.
The view volume is the space portion to be projected and presented in order to obtain a given picture: all the data lying outside this volume must be therefore clipped. The grid file approach is very useful to perform a fast pre-clipping, as the data contained in grid file cells having a void intersection with the view volume can be excluded from any further consideration.
The internal ordering of grid file cells can also be used to select very efficiently the subset of cells having some probability of giving a non void intersection with the view volume. Such selection will be performed by transorming the view volume in elliptical cylindrical coordinates <s,h,q> and considering the minimal containment box parallel to the axes in this coordinate space.

5.5 Hidden Surface Removal

The computing time of realistic graphic rendering is strongly dependent on the time needed to removing the hidded surfaces in the scene. The grid file technique is very useful in accelerating the solution of large scale HSR probles, as it reduces the *Depth-Sort* over the whole set of boundary surface elements, to an ordered set of DS problems over the subsets of elements contained in each grid file cell. Such an ordering of HSR sub-problems depends very simply from the observer's position in the space, and usually does not change when the observer moves slightly.

The natural (lattice) ordering of the grid file cells can be also used to remove many surfaces from the depth-sorted display list. In particular, all the surfaces can be removed which are contained in cells whose projection is completely covered by that of cells closer to the observer. This may happen only when the projection of nearest cells is complete, i.e. is in turn completely covered by that of its internal (opaque) surfaces ("holes" are the problem!)

In order to speed-up both the intersection problems and the computation of depth ordering inside each grid file cell, the surface elements will be spatially ordered by using the BSP binary tree structure proposed by Fuchs [FUCHS83]. Such binary tree allows to compute in linear time the depth ordering of each grid subproblem, when the position of the observer changes. It is the authors opinion that the composition of the grid file and the BSP techniques should make possible to realize computer animated views of the monument in a realistic time and using a general purpose hardware.

4.6 Shading and texture mapping

Our main decision related to computer graphics rendering concerns the choice of do not use any kind of ray tracing algorithm, as both of impracticable complexity, given the size of the Coliseum, and also as not suitable to render such kind of object, massive and opaque by definition!

To achieve the maximum efficiency in producing high quality pictures on a general purpose hardware we plan to use the Gouraud shading algorithm [ROGE85], aiming to perform a single scalar interpolation over the elementary surfaces (triangles) of the monument boundary. In particular we will use a "hue, intensity, saturation" colour model, and interpolate the ligth intensity variable with the Gouraud algorithm, while the hue and saturation variables will be locally defined by the texture mapping technique outlined in the following. In other words we use a divisionist techique to produce colour tones: the object materials will determine a discrete distribution of the intrinsic colour (hue) and the quantity of white (saturation); shades will then be obtained with a continous interpolation of light intensity, corresponding to put some black into the colour.

In order to obtain a close rendering of monument's surfaces and materials we will make large use of texture mapping techniques, based on projecting photographic pictures of monument's materials ("travertino", marble, etc.) over the displayed portion of the model surfaces. These techniques may be used both to render the actual aspect of surfaces in our age, and the original aspect in ancient romans era.

Considering the high quantity of elementary data to be processed to produce a single picture, we will use a front-to-back approach to the picture rasterization, by using the Weiler-Atherton general clipping algorithm [ROGE85]. In this case only the really visible portion of each displayed surface need to be shaded and rasterized, saving a great amount of computer time.

REFERENCES:

[ANSI81] American National Standard Institute, ANSI X3H32 Proposal for Structure, Document X3H32/81-09R1, April 1981.

[BIR88] G. Birardi, R. Carlucci, E. Ferrara, U. Giannoni and G. Maruffi, The Photogrammetric Survey of the Coliseum in Rome, paper presented to the 16th ISPRS, Kyoto, 1988.

[CODD81] C.J. Date Codd, *An introduction to Database systems* IBM Editorial Board - S.Jose, California 1981.

[FUCHS83] H. Fuchs, G.D. Abram, E.D.Grant. "Near Real-Time Shaded Display of Rigid Objects". *ACM Computer Graphics*, 17, 3, 1983.

[NIST88] A. Nistri, *Strutture di Dati e Formati di Trasporto per la Cartografia Numerica*. Master Thesis, Department of Computer Science, University "La Sapienza", Rome, 1988.

[PAOL88] A. Paoluzzi, A. Santarelli and M. Ramella, *A Boolean Algebra Over Linear Polyedrons*, submitted for publication, Roma, 1988.

[PAOL88b] A. Paoluzzi, "Il Modellatore Geometrico MINERVA", *Pixel*, Milano, 1988 (in corso di stampa).

[PHIGS87] International Standardization Organization, "Programmer's Hierarchical Interactive Graphics System", ISO DIS, 1987.

[REQ80] A.A.G. Requicha. "Representations for Rigid Solids: Theory, Methods and Systems". *ACM Computing Surveys*,12, 4, 1980.

[ULLM82] J.D. Ullman, *Principles of Database Systems*, second edition, Computer Science Press, 1982.

[ROGE85] D.F. Rogers, Procedural Elements for Computer Graphics, McGraw-Hill, NY, 1985.