Interference Detection in Architectural Databases *

José Miguel Salles Dias[1], Manuel Noronha Gamito[1]
Jorge Silva[2], José Fonseca[2], Miguel Vilar[3]
Yuhua Luo[4]
[1]ADETTI, Edifício Iscte, Av. das Forças Armadas, 1600 Lisbon, Portugal
[2]Oficina de Arquitectura and [3]Betar, Lisboa, Portugal
[4]UIB, University of Balearic Islands, Math & Comp. Science Dept, Palma, Spain
e-mail: *Miguel.Dias@adetti.iscte.pt, http://www.adetti.iscte.pt*

Abstract

In this paper we address the problem of geometric interference in architectural databases, that arises when different architecture and structural engineering teams integrate their designs into a common 3D scene. By analysing in detail the current work practices, we have modelled a design verification matrix for interference cheking, between architectural (wall, window and door openning, etc) and structural design elements (column, beam, roof slab, etc). In our novel algorithm, we've implemented a Binary Space Partitioning (BSP) scheme for all the scene elements, along with several boolean operators that apply to the BSP tree. Every interference can be expressed as a boolean equation that must hold true. Each pair of architectural and structural elements, from the verification matrix, is substituted into the relevant equations and, if a false result is obtained, an inconsistency is visualy signalled and can be discussed at a distance, using our own multi-user 3D virtual environment.

Keywords: architecture, structural engineering, interference detection

1. Introduction

The life cycle of a building can be decomposed into five separate stages: (i) planning, (ii) design, (iii) construction, (iv) use (including operation and maintenance) and (v) demolishing [2]. The first two phases deal with the production of architecture. They can be further refined into four distinct moments prior to the construction, where information is manipulated in digital or paper form. These are:

1. **Preliminary Study**: where the client's requirements are analysed and the outline proposal is produced.
2. **Conceptual Design or Base Program**: where a general conception of the project is defined, the technical team is formed, questions are raised regarding to the different aspects of the project and alternatives proposed.
3. **Overall Design**: where scheme designs are produced in 2D for several specialities (architectural, structural, water supply, sewage, gas, energy etc). The cost is estimated. Plans are defined and scheduled.
4. **Detailed Design**: where scheme designs are detailed in 2D. Cost estimation is revised; materials are selected and schedule of construction work is established.

Some of our authors have been working in the architectural design field over twenty years. According to their experience, the errors or inconsistency associated with the information processing can be classified in two cases. The first case is the omission of information – undefined geometry, which causes indeterminacy. The second case is the contradiction of the geometric information when two or more incompatible elements occupy the same space at the same time.
These errors are due to two technological reasons, namely insufficient functionality in the available CAD tools, and the methodological mistakes, associated with the current business process practices. These frequent inconsistency errors are mostly detected only in the building phase of the project, where the solutions to correct them are extremely costly. It may end up with total unsatisfaction for the client or the city authority. These errors could have been detected in the earlier architectural project stages, to minimise the presence of such problems during construction work.

* This work is sponsored by the European project ESPRIT 26287 M3D "Multi-site cooperative 3D design system for architecture".

Published by the Eurographics Association ISSN 1017-4656

An analytical study [3] has led to some conclusions about the reasons for the occurrence of such errors. In fact, the architect, who is responsible for the conceptual development, produces an initial geometric proposal, which is then converted into CAD data files. This information constitutes what is referred to as a *drawing basis*, which will be supplied to all specialities (structures, water and sewage, energy, air conditioning, etc), so that they are able to incorporate it into their specialised projects. After this stage, a *design verification* process takes place to produce a coherent and integrated geometric database. The verification process is currently performed manually, because of the lack of such functionality in the existing CAD tools.

In this paper we report some solutions achieved in improving the traditional CAD tools in the architecture domain. It is aiming at the *design verification* process. Our approach uses a standard neutral data format for the exchange of geometric and topologic information (VRML [1]) with the third party CAD tools. Specific algorithms are used to detect interference and design related errors, between architectural and structural engineering projects. A multi-user virtual environment [4] is provided for on-line discussion and remote visualisation of architectural databases. We report specifically the result achieved with the interference detection algorithms. Issues such as the proprietary multi-user virtual environment system (M3D system), new work practices or common databases, can be found, respectively, in references [3,4] and [9]. The interference detection stage is implemented with the binary space partitioning (BSP) scheme for the architectural or structural elements.

2. State of the Art

There is little published information about CAD tools for interference detection and design verification in the architectural domain. There exists, some tools for the automotive and aeronautic industry such as the ADV system [7]. This system performs collision detection and minimum distance checks between several distinct layers of geometry, which are produced by different design teams in the prototype study of a car or an aeroplane model. The computational technology for ADV is based on voxel spaces. The 3D space is decomposed into small cubic cells. Each cell has a list of the polygons that intersect with it. The polygon-polygon intersection tests can be made efficient, given that only small polygon neighbourhoods are considered.

	Structural Project					
	Column	Beam	Floor Slab	Foundation Piles	Foundation Beams	Roof Slab
Architectural Project						
Indoor Wall	X	X				
Outdoor Wall	X	X				
Roof	X					
Chimney	X		X			X
Door openings	X	X	X		X	
Window openings	X	X	X			
Indoor floor pavement	X					
Building base				X	X	
Staircase/elevator shaft	X	X				X

Table 1: *Object types in the design verification matrix for interference checking.*

3 User Requirements for Design Verification in Architectural Projects

Several interviews were carried out with the architect, designers and structural engineers in our study to define the required functionality for the conceptual design, overall design and detailed design processes. During these interviews, we studied the current practice in detail and used a real-life architectural project as a test bed. Our *design verification* algorithm allows the detection of interference between the elements of the architectural and structural projects, specified by the following table (the X indicates where interference is algorithmically studied):

A specific project rule is stated for each pair of elements that need to be checked for interference. We present two examples as the following:

- The concrete columns must be enclosed by the indoor and/or outdoor walls.
- Stairs must be designed according to ergonomic rules, so that there is sufficient height relative to the roof slab to accommodate human usage.

When space interference in the above table is detected, the algorithm produces a virtual camera position to view the zone on the VRML scene. The geometrical items that cause the problem are identified with a specific coded colour. The user will be allowed to access different virtual camera positions from a database of the detected interference cases. A designer may apply this study first to identify the problems easily. These problems may be further presented to the consideration of the experts. Later, a group can use our multi-user virtual environment system [5], which provides synchronous joint editing, to discuss and solve the interference problems. As a reaction to the detected designed related problems, the multi-user virtual environment system offers the possibility to edit the specialised project elements involved in the interference in 3D.

4. Technology

The interference detection component is implemented in the context of a Constructive Solid Geometry (CSG) environment. During the pre-processing step, every geometric structure is converted into a Binary Space Partitioning (BSP) tree. The evaluation of boolean operations with BSP trees is very simple to implement and also very efficient [8]. Three boolean operations are implemented: union, intersection and difference. These operations are defined in accordance with the *regularised set theory* of [6]. At this point, every inconsistency check, listed in Table 1, can be expressed as a logic equation that must hold true. Every pair of architectural and structural elements is substituted into the relevant equations and, if a false result is obtained, an inconsistency is signalled.

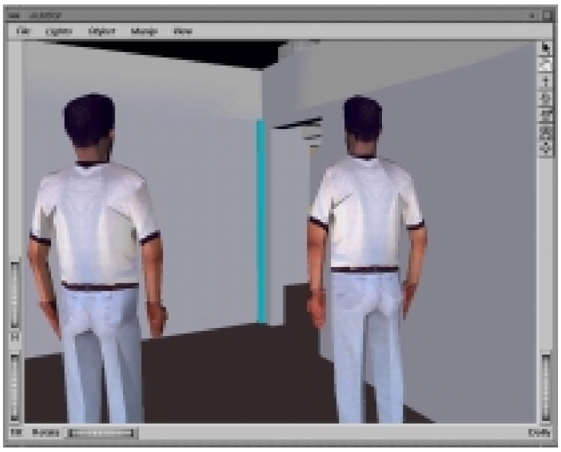

Figure 2: *The M3D Editor. A column (in cyan) not absorbed by walls (in light grey). This is a typical interference problem, analysed in a joint M3D multi-user virtual environment.*

5. Examples

As an example of how to deal with inconsistent items, we can consider an undesirable partial exclusion of a concrete column from an indoor wall. In an ideal situation, the concrete column must be absorbed by the wall. This corresponds to the boolean equation:

$$Column - Wall = \varnothing$$

where the symbol $\varnothing$ represents the empty set. The portions of the column outside the wall will be the result of the difference operation. Therefore, an inconsistency will exist if this operation does not yield $\varnothing$. Our interference algorithm identifies the geometric zone with a symbolic colour (false colour) and creates a virtual camera viewpoint, that is "directed" to the troublesome location. In the context of the M3D multi-user virtual environment, one can:

- remove the concrete column or wall.
- modify the column's section and/or height.
- modify the wall's section and/or height.
- eliminate one or both elements.

Figure 2 shows an offending column, visualised in our multi-user environment. Figure 3 shows the result of the boolean difference operator.

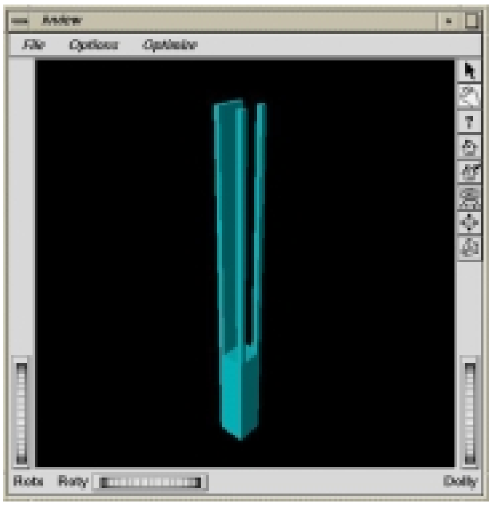

Figure. 3: *The column with the wall subtracted. The thin vertical parallelepipeds correspond to the volume of the column outside the wall. The lower portion of the column is bellow the wall and is not relevant for the purpose of this interference test.*

Another example is the window or door openings interference. Every door or window has an associated envelope that should not intersect with the rest of the elements. This will guaranty easy access and good lighting conditions. We begin with the definition of an envelope function *SafeVolume*() that, given a door or window, will return a volume that should remain free of other geometries. The interference test then becomes:

$$SafeVolume(Door) \cap Architecture = \varnothing, \text{ and}$$
$$SafeVolume(Door) \cap Structure = \varnothing$$

where *Architecture* and *Structure* are the complete geometric sets of the architectural and structural designs, respectively.

5. Conclusions

We have presented a novel technological framework for the computational verification of design rules associated with the integration of architectural and structural projects. In our original technique, we first identify design rule heuristics by analysing current work practices. This enables us to define the problem according to a design verification matrix. Then we associate every relevant element of this matrix with a logic equation that must hold true if no interference exists. The equation formalism is based on the *regularised set theory* of [6] and applies to the geometric database, which is organised as a collection of BSP trees. Finally, we allow the resulting VRML geometry to be loaded into a multi-user virtual environment (M3D), for synchronous group discussions over IP networks. We are currently conducting further research to apply this method for the interference detection between a larger set of specialities (architecture, structures, water, sewage and air conditioning), where different design rule heuristics will be necessary.

Acknowledgments

We would like to thank Ricardo Galli, David Sánchez, Renato Marques, Sérgio Alves, Marco Moita, António Almeida for their collaboration in this work, namely, in the M3D system specification and prototype development.

References

1. Ames A. L., Nadeau D. R., Moreland J. L. The VRML 2.0 Sourcebook. John Wiley, New York (1996).
2. “Product Data Representation and Exchange, Part 230 - Building Structural Frame: Steelwork”, ISO TC184/SC4/WG3 N551, Working Draft12 October 1996.
3. M. Dias (editor), R. Marques, F. Estevens , J. Silva, J. M. Fonseca, J. Arias, A. González, “*Architecture Process model”*, ESPRIT Project No. 26287 (Multi-site cooperative 3D design system for architecture) Deliverable 1.1.2, , Sept. 1998.
4. Y. Luo, D. Sánchez, S. Alves, M. Dias, R. Marques, A. Almeida, J. Silva, J. Manuel , , B. Tummers (EDC), Ed. R.Galli, “*M3D technical specifications*”, ESPRIT Project No. 26287 M3D, Deliverable 1.2, April1999.
5. Y. Luo, R. Galli, A. Almeida, M. Dias, A Prototype System for Cooperative Architecture Design, Proceedings of IEEE Symposium on Information Visualization, July 1999, London.
6. Requicha, A.A.G., “*Representations for Rigid Solids: Theory, Methods and Systems*”, Computing Surveys, 12(4), pp. 437-464, December 1980.
7. Tecoplan AG, Digital Mockup Technology, http://www.tecoplan.de/products
8. Thibault, W.C., Naylor, B.F., “*Set Operations on Polyhedra Using Binary Space Partitioning Trees*”, ACM Computer Graphics, 21(4), pp. 153-162, July 1987.
9. B. Tummers, R Galli, Y. Luo, M. Dias, A. Almeida, B. Pons , J. Silva, *Definition of database access, data structure and data exchange tools*, ESPRIT Project No. 26287 M3D, Deliverable 1.3, April. 1999.