Techniques of Superimposition

Peter Szalapaj and Andrew Kane

Abstract: This paper addresses the issues of 2-D and 3-D image manipulation in the context of a Computational Design Formulation System. The central feature of such a system is the ability to bring together two or more design objects in the same reference space for the purpose of analysis. Studies of traditional design methods has revealed the effectiveness of this technique of superimposition. This paper describes ways in which superimposition can be achieved, and, in particular, focuses on a range of domain-independent knowledge-based graphical operators that enable the decomposition of complex design forms into simpler aspects (secondary models) that can then be superimposed and/or analysed from a design-theoretic point of view. Examples of domain-independent knowledge-base graphical operators include object selection, planar bisection, 2-D closure (the grouping of lines into regions), aggregation (the decomposition of 2-D regions into aggregations of lines), spatial bisection, 3-D closure (the grouping of 2-D regions into volumes), 3-D aggregation (the decomposition of volumes into aggregations of 2-D regions). The representation of these operators is dependent upon the notion of a *parameterisable volume*, thus avoiding the need for translations between multiple representations of graphical objects by providing a common representation form for all objects. Secondary models can therefore subsequently be manipulated either through subtractive procedures (e.g. carving voids from solids), or by additive ones (e.g. assembling given design elements), or by other means such as transformation or distortion. The same techniques of superimposition can also be used to support the visualisation of design forms in two ways: by the juxtaposition of plans and sections with the 3-D form; by the multiple superimposition of alternative design representations e.g. structural schematic, parti schematic, volumetric schematic and architectural model.

Keywords: Design Formulation, Superimposition, Primary Model, Secondary Model, Parameterisable Volume

1 INTRODUCTION

This paper focuses on a range of techniques that are necessary to support the kinds of activities that occur in the context of a Design Formulation System (DFS), first introduced in (Szalapaj & Kane, 1993). The essence of such a system is that it should support the evaluation of design proposals in a highly interactive and intuitive way. Design proposals can be evaluated by means of superimposing them over either existing proposals produced at an earlier stage in the design process, or over known design precedents. Bringing together alternative design proposals in this way within the same design environment encourages rapid design development, and the re-use of previous design proposals in effective ways.

Our more specific interest in this paper is in exploring how superimposition can be used as a system to:

- combine 2-D and 3-D understandings during design
- overlay new design ideas over previous ones *comparatively*.

During traditional design activity, whilst using manual graphics techniques, we separate 2-D from 3-D understandings of the scheme under development. Designers have to manipulate relationships via 2-D layouts/descriptions because those layouts would otherwise be hidden, or buried within a 3-D image. Therefore designers always associate 2-D understandings with externalised 3-D snapshots, such as with perspective views, for example. To be able to understand complex 2-D relationships within 3-D forms, it would be preferable to superimpose interaction with 2-D layouts onto 3-D schematics. In this case, the axonometric is the key projection to use, since planar sections are always orthogonal, unlike in isometric projections where they are not. Furthermore, superimposition is the key during design to compare new ideas against either current or old proposals.

2 THE BASIC DFS ENVIRONMENT

To investigate the aforementioned ideas, we are currently specifying a prototype user-definable design environment which enables designers to describe a wide range of 2-D and 3-D graphical schemes. The DFS that we are proposing goes beyond the few 3-D sketch design systems that are currently on the market such as 3-D World (Caplin, 96), Alias Upfront, and Alias Wavefront. The emphasis of all such systems seems to be upon speeding up already well-understood visualisation and presentation techniques, such as object creation and conventional transformation, rendering and texture mapping, and animation. We are assuming that the DFS will include basic functions for object creation, and visualisation functions based increasingly upon the emerging standard control mechanisms such as QuickTime VR™ and QuickDraw 3D. The latter is particularly useful for supporting interactive Constructive Solid Geometry (CSG) operations. However, we do not take the view that sophisticated presentation facilities are necessary in the early stages of design development.

We believe that the development of superimposition techniques is the most useful way to support the reflection-in-action that takes place in early stage design development. Superimposition is also a step towards supporting multiple representations of design forms, whether these are generated by individual designers, or by several members of a design team.

What we are presently investigating are the general functions that are necessary to support user interventions within a design scheme, the language of the user-interface that controls and guides the progression from initial design proposal, to evaluation and development of particular parts and features, through to a new and enhanced scheme. We are concerned with the construction of the interaction tools: the creation of evaluation contexts, the decomposition of 3-D models into 2-D views, the reconstruction of 3-D models from edited and transformed 2-D views. In general, we are aiming to support reflection-in-action in terms of rapid movement between 3-D and 2-D design development.

3 DEVELOPMENT OF SUPERIMPOSITION TECHNIQUES

In the graphical environment of the DFS, users sketch design ideas in the form of sketch objects. The central premise of the design formulation component of our system is that design intentions should be considered as projections/reflections of design ideas. To support user-formulation of design intentions, therefore, which in turn require the subsequent manipulation of sketch objects, we are currently implementing a range of operators that allow the superimposition of intention objects over the initial sketch objects. These effectively appear on a superimposed layer over the sketching layer(s) but intention objects also have the property of being associated with objects on the underlying layer in terms of their extent. For example, in section 5.2, we will introduce the notion of planar bisection which can be applied to any underlying region containing sketched objects. This can be used to distinguish between regions of a sketch which require analysis in terms of architectonic principles in which identifiable regions are important, for example, in the notions of balance and symmetry (**figure 1**) (Clark and Pause, 85).

The design intention of assessing the subjective quality of balance, for example, involves making a partition of space, and any such partition can then be stored in the DFS for extraction at some future date. In this sense, we are developing a reflective system in which the design history is important, and can be reflected back to the designer by means of the highlighting of sketched objects to which the intention operators refer.

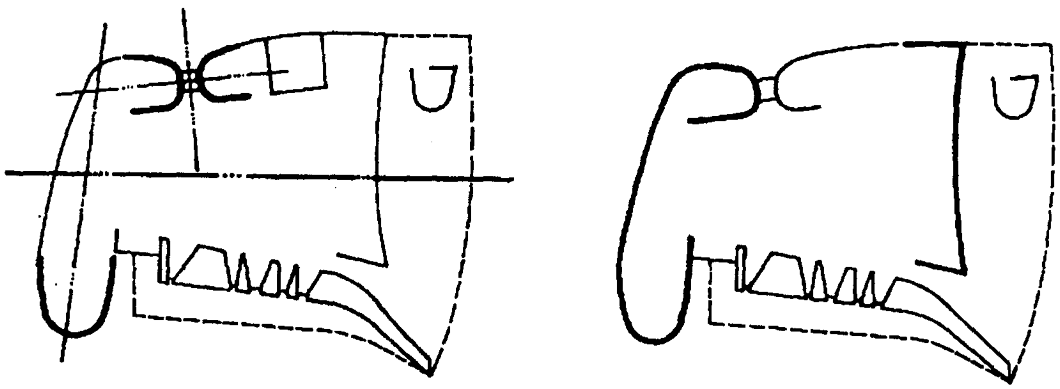


Figure 1: Application of Planar Bisection to Symmetry and Balance in Corbusier's Notre Dame du Haut Chapel at Ronchamp

4 SUPERIMPOSITION EXAMPLE

Before going in to the details of specific graphical operators, we will first give an illustration of the kind interaction that we envisage taking place. In **figure 2**, a part of a 3-D design proposal has been isolated from the rest of the scheme for the purposes of more detailed design development. The user interacts with projected section planes, and changes are reflected back into the isolated zone of the 3-D scheme. This zone is formed by the intersection of the section planes and the 3-D model. All manipulation takes place within a single 3-D environment; there are no separate plan or section 2-D views in use.

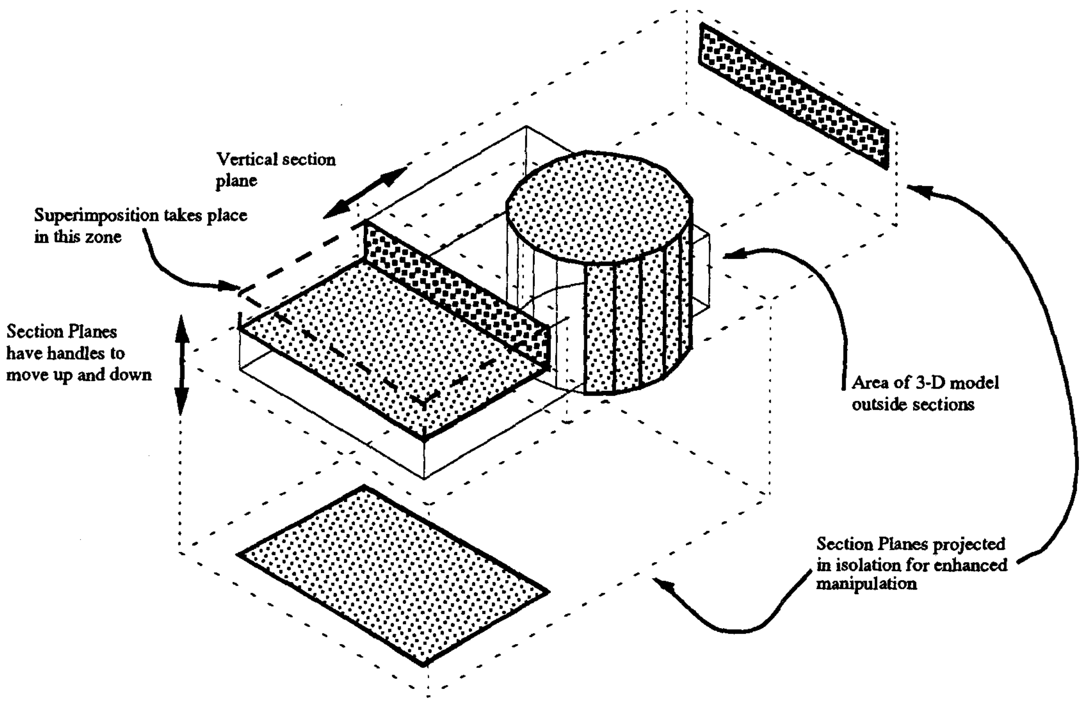


Figure 2: Example of Superimposition

5 DOMAIN-INDEPENDENT KNOWLEDGE-BASE GRAPHICAL OPERATORS

From the description given in the previous section, it is evident that a range of graphical operators are necessary to support the kinds of interactions that we envisage.

5.1 Object Selection Using Parameterisable Volumes

An important operator currently under implementation is one which supports volumetric extent. This operator is fully parameterisable in 3-D (**figure 3**), and therefore can also be collapsed down into either 2-D (spatial) or 1-D (planar) form. This operator is both an object, in the sense that it has a graphical representation, and a selection mechanism, in that it can be used to identify parts of design objects.

It is obvious that this operator can become particularly useful in supporting analysis of the important architectonic property of proportion. In its linear form, this operator can be applied to the analysis of systems of proportion used in well-known design precedents.

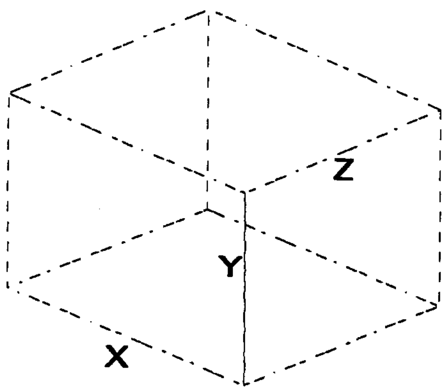


Figure 3: Parameterisable Volume

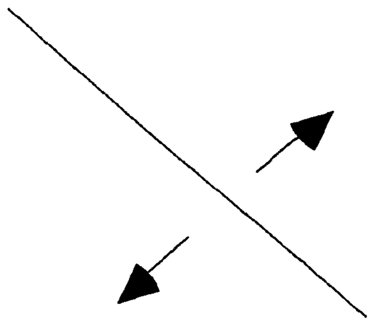


Figure 4: Planar Bisection

5.2 Planar Bisection

In a planar bisection (**figure 4**), the user indicates that there is effectively a left hand side and a right hand side of a 2-D drawing or part thereof, which can then be used for a form of comparative analysis (Kane and Szalapaj, 94).

5.3 2-D Closure (the grouping of lines into regions)

2-D Closure is effectively the grouping of lines in a 2-D sketch to form a closed region which can then be used in a further interaction sequence. These regions can then be associated with the projections of 3-D objects.

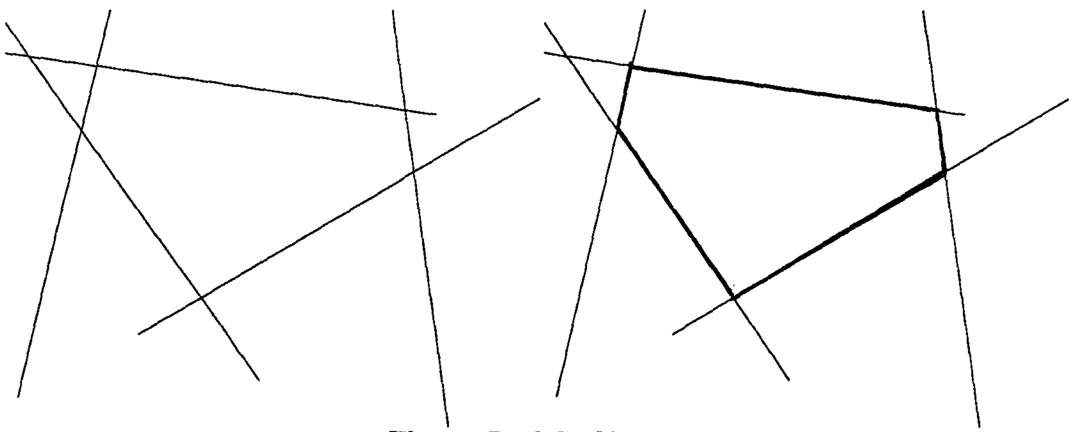


Figure 5 : 2-D Closure

5.4 Aggregation (the decomposition of 2-D regions into aggregations of lines)

Aggregation is the converse operation to 2-D Closure, ands initiated upon projections of 3-D models. The implication of this operator is that furtheediting will take place which may well break the topologies of the initial objects to form new connectivities between lines.

5.5 Spatial Bisection

This is the 2-D form of the parameterisable volume, and is used to cut sections at any angle through 3-D models. Just as in planar bisection, there is a directionality associated with the spatial bisection.

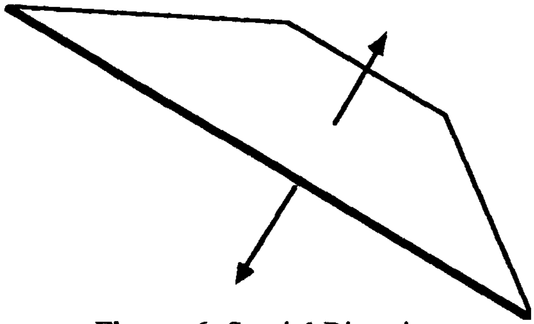


Figure 6: Spatial Bisection

5.6 3-D Closure (the grouping of 2-D regions into volumes)

3-D Closure is a more complex graphical operator than the 2-D equivalent. It requires not only the grouping of lines into regions, but also the specification of occlusion relationships i.e. certain regions are in front of or behind either other regions, or other lines. From the expression of such relationships, a 3-D model can be reconstructed. Such an operator is necessary when, having already taken a section through a 3-D model, the user carries out graphical edits which affect the topology of the initial model, and thus the occlusion relationships. The implication of an operator such as 3-D closure is that the DFS is actively supporting a transition from 2-D to 3-D.

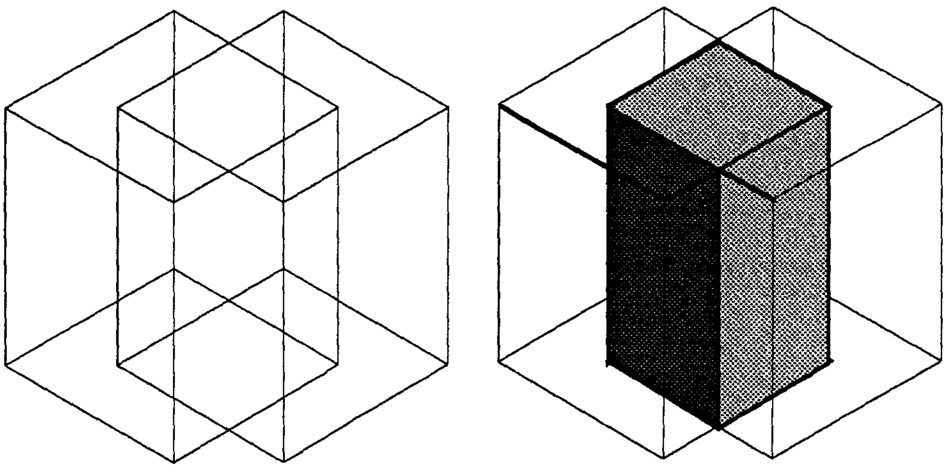


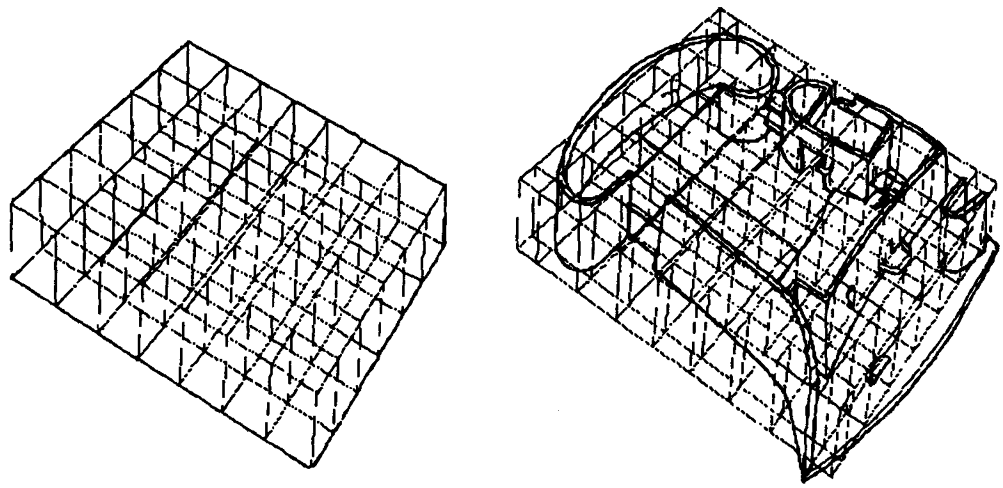
Figure 7: 3-D Closure

5.7 3-D Aggregation (the decomposition of volumes into aggregations of 2-D regions)

3-D aggregation is the converse operation to 3-D Closure. Again, just as in aggregation, this operator is used in contexts in which users transform design objects from one co-ordinate system into another.

6 MULTIPLE REPRESENTATION THROUGH SUPERIMPOSITION

Multiple representations of alternative design concept models have to date been difficult to develop (Logan, 1989). All such models are abstractions of the real world that omit details in order to focus on the aspects of reality important to a given design task. Models developed for different design tasks therefore omit different kinds of details. Building a model that is adequate for more than one task requires a clear articulation of the demands that each task places upon the model. We have earlier given examples of what such tasks might be in the context of early stage design in architecture. They might include evaluation of formal relationships such as symmetry, balance, and proportion, for example. **Figure 8** illustrates one such abstraction achievable with superimposition techniques. In Corbusier's Chapel at Ronchamp, its curved walls are tensioned against an orthogonal grid, which in turn itself becomes a model for the organisation of design elements (Baker, 1989).



a) 3-D Cartesian Grid Representation b)Superimposition of Grid and Ronchamp

Figure 8.

Determining the modelling needs of an individual task is an iterative process, involving both top-down and bottom-up activity. Each iteration requires insight into conceptual structures that are important to the given task (top-down) in conjunction with validation and testing against the needs of applications that perform the task (bottom-up). The result of each iteration is a working model, documented wherever possible by the reasons that individual conceptual structures meet the demands of the task. This recording of iterative design intentions is also an important feature of the knowledge base component of the DFS, and is analogous to the storage of successive boolean operations within CSG systems. The project design history can then be reflected back to the designer by means of a sequence of successive superimposition operations.

7 CONCLUSION

Our concern has been with the expression of design possibilities in a DFS system, primarily through the generation of schematic design proposals for the realisation of early stage design schemes. The DFS framework with superimposition facilitates user reflection and co-operation. We have proposed a general set of techniques and a collection of graphical operators that can be used by designers from many backgrounds and different design disciplines. The domain-independent graphical operators described in this paper form part of the knowledge base of the DFS. The highly interactive interface provided by such a system, frees users from the tedium and detail typically required in the drafting process, allowing them to sketch new ideas and to make modifications to existing designs in intuitive ways. We want to encourage designers to work within a medium that is radically different from following prescriptive CAD procedures with precise dimensioning. The ability to move swiftly between design models in two and three dimensions, supports designers' creativity, and allows fast exploration of many design solutions. This user-definable design environment, therefore, rather than simply being a drafting accessory which is used after designs have been developed (much as conventional CAD systems are used), will instead be a working tool for the creation of designs right at the early stages.

8 REFERENCES

Baker, G.H. (1989); *Design Strategies in Architecture,*Van Nostrand Reinhold.

Caplin, S. (1996); Review of 3D World, The Mac Magazine, February.

Clark R.H., and Pause M. (1985); *Precedents in Architecture,* Van Nostrand Reinhold.

Kane, A. and Szalapaj, P.J. (1994); *Intuitive Analysis as Mediator between Concept and Representation,* 2nd Conference on Design and Decision Support Systems, Vaals.

Logan, B.S. (1989); *Conceptualising Design Knowledge,* Design Studies, vol.10, no.3.

Szalapaj, P.J. and Kane, A. (1993); "*Computationally Assisted Design Formulation*", Advanced Technologies, M.R. Beheshti & K. Zreik (eds.), Elsevier.

Szalapaj, P.J. and Kane, A. (1994); Putting CAD in Perspective: Towards the Computational Representation of Visual Metaphors in Conceptual Design, 2nd Conference on Design and Decision Support Systems, Vaals, The Netherlands.

Szalapaj, P.J. and Bosvieux Coilliot, G. (1995); Customising Cooperative Design, Europia '95, Lyon, France, December.