Sunlighting Design: an Inverse Approach of Simulation for CAD Tools

Daniel Siret

Sunlighting Design : an Inverse Approach of Simulation for CAD Tools

Daniel Siret
Laboratoire CERMA, URA CNRS 1581
École d'Architecture de Nantes
Rue Massenet, 44300 Nantes, France
E-Mail : siret@cerma-nantes.fr

Abstract

Existing simulation tools are mainly used for appraising environmental properties of architectural and urban constructions. In most cases, these tools do not suit design practice as they work on completed schemes. A designer who wants to emphasize an environmental point of view needs a new range of CAD tools that work in an inverse way : from the environmental properties to the built forms.

Our main issue for this paper is to present an inverse approach application in the field of sunlighting simulation and design. We have developed an experimental CAD tool that models objects from declarative sunlighting properties. Our system works on proposals such as ' this shape must be sunlit (or sunless) for the end of afternoon in winter ', and it suggests solutions — shadings and openings — that check on these proposals. This methodology opens new research avenues toward a real aid for environmental design.

1. Introduction

How can an architect emphasize an environmental aesthetic design [1] ? How can he actually design environmental properties ? Many simulation tools provide an analytical way to evaluate and qualify architectural and urban constructions from an environmental point of view : sunny in winter, not so noisy, wind-protected most of the time, and so on. These tools generally need a full description of constructions. In most architectural and urban design processes, they cannot be used before the main schemes are completed. Most of the time, it's then too late or very costly to go back on the first designer's choices.

In contrast to this deductive process, we suggest that designers who want to emphasize an environmental point of view need a new searching [2] range of simulation tools. Such tools would not work in an analytical way, but in what we call an inverse and synthetic way to suit design practice. In that inverse way, the question to answer would not be : what kind of properties can I observe in this environment ? … but rather : what kind of environment(s) must I build to get these properties ?

We would like to show here that this theoretical challenge can yet be overcome for some common environmental properties like sunlighting. A usual sunlighting simulation system would model geometric sunlit or sunless shapes, for a given time, in a given architectural or urban construction. On the contrary, the inverse approach we advocate consists of finding under which conditions a given sunlit or sunless shape can exist, for a given time, in a given construction.

The system we have developed provides a new inverse simulation tool for making sunlighting an actual formgiver [3]. It works on proposals with reference to intuitive time intervals such as ' early morning in summer, the end of the afternoon in winter ', and so on. It computes a complex geometrical construction of sunlighting phenomena — an artefact connecting time and space. This complex volume provides a visualization of the sunlighting constraint and it enables the designer to model the different solutions (shadings or openings) that check on the given proposal.

In this paper, we firstly focus on the general techniques for reversing simulation and we expose the specificities of our approach. We describe our system as an illustration for this issue : first, the formalization of a sunlighting proposal, with attention to the representation of intuitive time periods ; then, the new geometrical method we have established to compute the sunlighting volume ; finally, the logic of sunlighting constraints we use to model shadings and openings. We illustrate these features with two examples of design processes that our system is suited to. To conclude, we discuss the limits of our system to date and we describe some of the new tools that should be developed to overcome them.

2. Reversing simulation

Direct environmental simulation methodology consists of computing the states of a phenomenon, for a given construction (geometry informed by some material features) and a set of hypotheses. In that way, direct sunlighting simulation consists of computing geometrical sunless

Santo H.P. (Ed.), *Advances in Computer-Aided-Design*, Proceedings of CADEX'96, Hagenberg, Autriche
IEEE COMPUTER SOCIETY PRESS, Los Alamitos California (USA), Sept. 1996, pp. 32-40

or sunlit shapes for a given construction, with time and location hypotheses.

Many authors have pointed out the drawbacks of such methodology in design practice. Two main arguments can be outlined. The first one concerns the trial-error process one must operate using successive direct simulations. The designer must come up with a solution, simulate it, evaluate the results, and then, modify it in a way he hopes will give a better solution — as well summerized by G. Stevens [4]. This process is not only boring and inefficient. Most of the time, it constrains the designer to complete his project before any simulation can be performed. This is the second and most significant argument that discourages the use of direct simulation in design process. As it generally needs a full description of constructions (topology, geometry, materials), it cannot be run before any relevant proposal is designed — even when it would be usefull *while* designing this proposal. As a result, direct simulation tools are mainly used as evaluative or corrective tools on already-designed projects.

Then, the idea of reversing the direct simulation process looks appealing. Instead of waiting for a fortunate good solution by testing hypothetical ones, one should better fix the goals to reach firstly and then, try to achieve them in some constructive way. A first step in that direction has been made with the generative tools. Following L. Khemlani : These are tools which are able to generate alternative solutions for various limited aspects of the design that satisfy some given well-definable specification. A generative process is essentially one of searching through all possible solutions to a given problem to find those that meet specific goals [5]. Such generative tools use what is known as the generate and test process : repeatedly generating solutions (by exhaustive or random enumeration) and testing them with a direct simulation process to find the best (figure 1). Khemlani used this method for designing windows in an energy conscious way. Various feasible windows configurations are first generated within important constraints in order to reduce the combinatorial explosion and to enable the simulation ; then, they are tested and sorted with a simple simulation model. The best one(s) are proposed to the designer. Genetic algorithms can be used as a refined and more efficient form of this method.

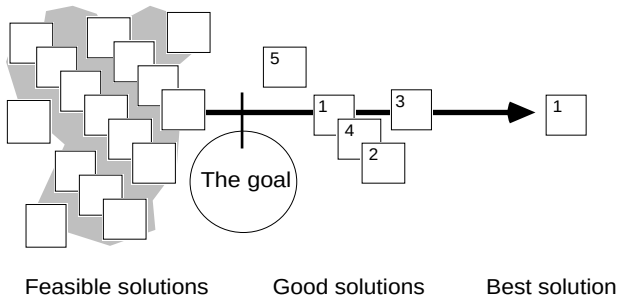

Figure 1. Generate and test process

However, generative process assumes that one can a priori define the set of feasible solutions to a given problem. This looks presumptuous for architectural design where proposals are considered as they provide an original and innovative solution to a problem in a given context. Then, instead of generating a priori solutions for testing them, it has been proposed to dynamically appraise design in progress. The challenge is to evaluate solutions that are partially completed and to make recommendations for their improvement, according to an explicit or implicit goal (figure 2). This can be performed using the well-known expert or knowledge-based systems described in the last decades literature [6]. Such ‘ intelligent ’ systems may be able to diagnose plans at the preliminary stages of design, by logically reasonning on rules and facts as an expert would do in his proper domain. Various systems have been built in the field of environmental design. They concern energy conscious design [7] or daylighting ; they utilise standard, or even, fuzzy logic to interpret and to evaluate the designer’s proposals. Nevertheless, many problems still remain with such tools. One concerns the relevance of the knowledge they rest on — which usually comes from building regulations or standard solutions, familiar to the designers. Another problem appears when the rules need a simulation result as a fact to make a decision : the system might be unable to perform any relevant simulation process at the current stage of design. Furthermore, even if some expert systems can act as generative tools, most of them cannot ensure they explore the whole set of feasible solutions. They just provide a guidance to an hypothetical better solution, whatever the context that has led the designer to the current one.

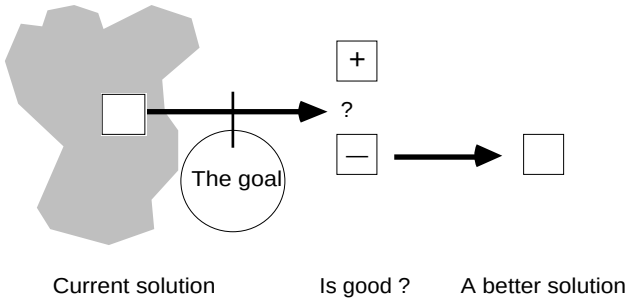

Figure 2. Expert guidance process

A third class of reversing simulation systems are those which directly find the best solution to a given problem (figure 3). These are based on optimization methods. As suggested by Radford and Gero, optimization can actually be thought of as the reverse of simulation since the desired future state (that is, the optimal future performance) is specified and the values of the variables which will produce that state are sought [8]. Many technical well-defined problems can be solved using such optimization methods, like the inverse problem of artificial lighting in a scene [9]. However, in architectural design, proposals result

from subtle compromises between various criteria — most of them (aesthetic, historical, philosophical, etc.) being non-evaluable or non-comparable, in terms of profit, to the others. Then, while generative tools provide several good solutions from which a designer may choose his proper one (the best or not), optimization tools go straight towards the best and unique solution. Moreover, they remain black boxes. Unlike expert tools, they cannot advise in any way a designer who may need a guidance.

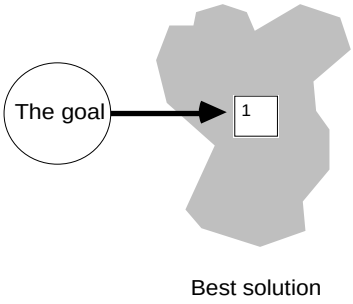

Figure 3. Optimization process

Another process for reversing simulation is the one we call inverse simulation and that we advocate in this paper. Instead of computing the supposed best or unique solution as optimization techniques do, inverse simulation aims at determining a *model* of the hypothetical good solutions, according to the specified goal (figure 4). The good solutions are obviously those among feasible solutions that satisfy the goal. A model of the good solutions is any formalization that makes up a set of sufficient conditions a solution may satisfy to be considered as a good one. This general definition allows various representation of such a model : geometrical, symbolical, physical, etc. Anyway, the model should be able to embody the whole space of solutions instead of defining each of the solutions in this space, as generative tools do. Then, inverse simulation would unite the advantages of both optimization and generative processes. Firstly, it ensures one will obtain a good solution ; secondly, it enables to explore various good solutions (possibly using an expert guidance) and to choose one, according to different non-evaluable criteria.

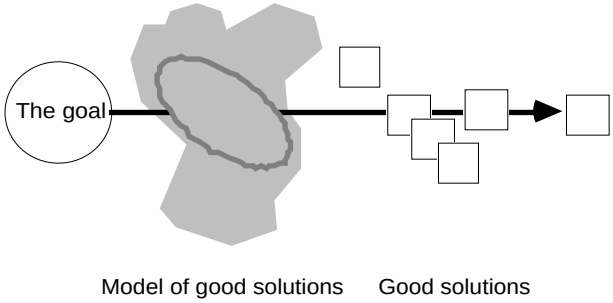

Figure 4. Inverse simulation process

Considering this schematic proposition, the two main challenges that inverse simulation raises concern one, the implementation of an inverse model of the phenomenon to simulate, and the other, the means to explore the various solutions this model may generate if applied. Following is a proposition in order to overcome these challenges in the field of sunlighting design.

3. Inverse simulation of sunlighting

Direct sunlighting simulation consists of computing geometrical sunless or sunlit shapes for a given construction, with time and location hypotheses. On the contrary, inverse simulation starts from the result. It is based on a sunlit or a sunless geometrical shape and a given time period — that is to say, a sunlighting proposal as we decribe it below. It consists of computing the conditions under which the construction can check on this sunlighting proposal, i.e : how can this construction generate the given sunspot for the given time period and the given context. If necessary, the construction may be transformed with shadings or openings ; these are the solutions to the sunlighting inverse problem.

The method we propose consists of computing the model of the whole space of solutions associated to a sunlighting proposal as a geometrical artefact connecting time and space. All solutions (shadings or openings) that achieve the given sunlighting proposal can be generated using this volume with simple logical functions.

3.1. Sunlighting proposals

As it starts from the result, inverse sunlighting simulation just needs the designer to describe the sunlighting property that has to be achieved. Such a property can be proposed at any stage of the design process : to define the massing, to arrange objects together, to make windows, to draw sun visors, etc. In any case, a sunlighting property can be formalized as a set of three parameters :

- the geometrical spot that receives sunlighting,
- the sunlighting qualification,
- the time period during which the spot must check on the qualification.

We shall call P (for Polygon) the geometrical spot, S (for Sun) the qualification and T the Time period. These parameters form the sunlighting proposal (P, S, T) that can be written : *P must be S during T*.

Following our method, the shape P can be any convex polygon. In our system, P is a rectangle specially drawn on a plane surface (of any orientation) to compose the sunlighting shape. Thus, the rectangle P represents the area, with implicit fuzzy boundaries, for which the architect wishes to design the sunlighting property.

At the present stage of the development, our system just accepts the binary qualification : *sunlit* or *sunless*. We will discuss this restriction in section 5.

Any intervals of hours and days can be used to specify the time period T. However, human aspects of architectural or urban design invite us to provide an intuitive representation of time, that refers to common life situations like : the end of the afternoon in spring, early morning in May, midday in summer, and so on.

beginning *the middle of* *the end of*	X	*the day* *the morning* *the afternoon*	=	*beginning of the day* *middle of afternoon* *the end of morning* etc.	
			+	*midday* time t	
beginning *the middle of* *the end of*	X	*January* *February* *(...)* *December*	=	*beginning January* *the middle of May* *(...)* *the end of December*	
			+	*equinoxes* *solstices*	
			+	*Summer* *Autumn* *Winter* *Spring*	

Figure 5. The lived time periods

We suggest the concept of *lived time* to define these fuzzy and intuitive intervals of time. As oppose to the absolute and linear time, the lived time represents all situations comprising common solar periods during the year and day. By this we mean, without listing them, all the time intervals that have a semantic representation (at least in French) in the two dimensions of solar movement. The tables in figure 5 show what can be described using simple temporal semantic constructions.

We have established a system that make these lived time intervals correspond to the positions of the sun. This system meshes the solar trajectories with regular test days and proportional test instants (see figure 6, the view of sun-path on sky vault). Resulting panes and groups of panes represent a lived time period with an acceptable accuracy for design practice.

3.2. The sunlighting volume

Inverse simulation of sunlighting mainly raises geometrical problems. The method we propose consists of reconstructing the complex sunlighting volume defined by a convex polygon and a time period. This volume embodies the exact set of points that may shade the polygon for one instant within the given period. Any sunlighting proposal (P, S, T) generates a sunlighting volume that we denote $\prod(P, T)$. The diagrams on figures 6 and 7 illustrate the method we use to build it.

First, we compute the simple volume π defined for any point p on the plane of P and for the period T (see figure 6). The volume π embodies the shading points for the point p during the time period T. It has its vertex on p and its edges follow the solar directions $p\text{-}t_i$ defined by the time interval T.

Geometrically speaking, the complex volume $\prod(P, T)$ results from the union of all the simple volumes π based on each point of the polygon P. As we consider P convex, the result squares with the *displacement* of π round the perimeter of the polygon P (figure 7). This displacement produces the complex sunlighting volume $\prod(P, T)$.

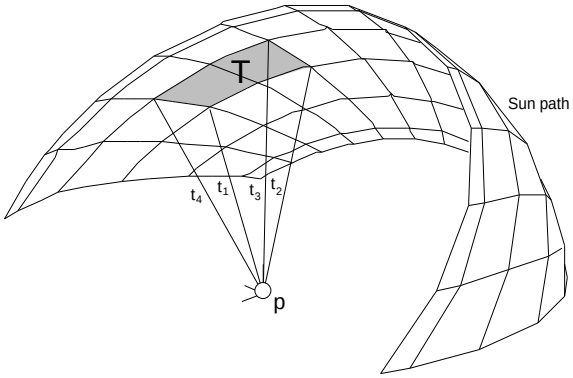

Figure 6. A simple sunlighting volume π for a lived time period T

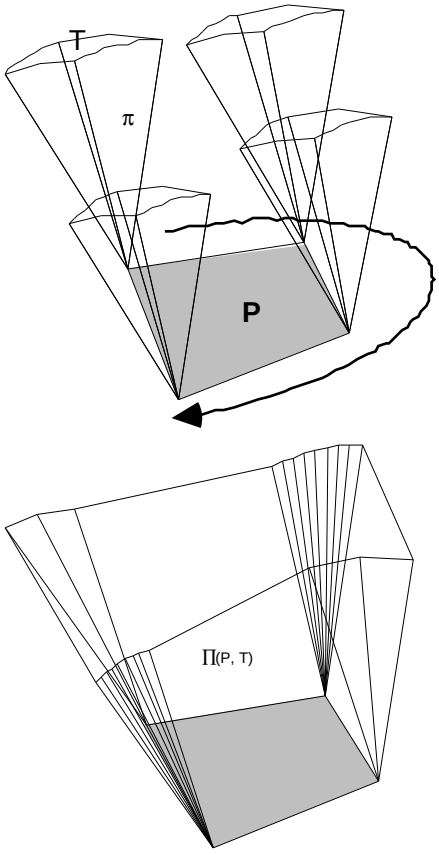

Figure 7. Displacement of π round the perimeter of P and resulting volume $\prod$

This method can be implemented using classic boolean functions on polygons — union, intersection, difference. However, some problems may appear if the volume π is concave. Then, when displacing round the convex polygon P, π can intersect its own trail. On figure 8, π pictures the ground plan of the lived time period *morning in summer*. It is shown on each vertex of the polygon ABCD. When moving along the edge AB, π intersects its own trail in M. The new direction AM must be computed.

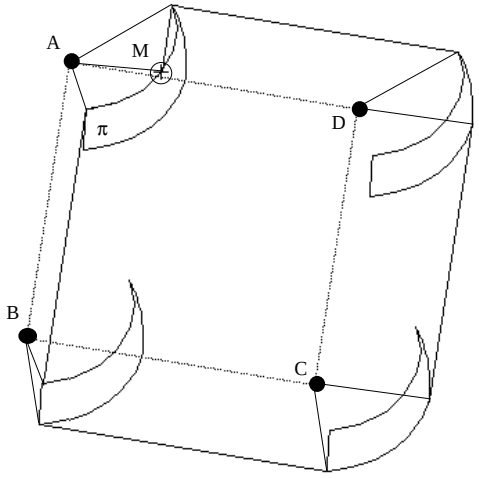

Figure 8. Intersection between π and its own trail during the displacement round ABCD.

3.3. Solutions generation

We have shown how to formalize an intuitive sunlighting proposal (P, S, T) and how to compute the complex sunlighting volume $\prod$(P, T) associated with it. Our aim is now to make the proposal (P, S, T) become true, that is to generate the openings or shadings that enable the polygon P to be S (sunlit or sunless) during the time period T.

An opening is any hole that lets the sunbeams reach the spot P during T. A shading is any object that makes a shadow on P during T. There always exists an opening or a set of openings — a shading or a set of shadings — that check on a given sunlighting proposal.

Thus, a sufficient condition for P to be sunless during T is that there exists at least one object intersecting the complex sunlighting volume $\prod$(P, T). Reciprocally, a necessary condition for P to be sunlit during T is that no object in the scene intersects, even partially, the volume $\prod$.

Given these simple rules, there are two ways of achieving a sunlighting proposal when necessary : first, to create the openings that $\prod$(P, T) generates in all the scene for type (P, Sunlit, T) proposals ; second, to define at least one shading object that intersects $\prod$(P, T) for type (P, Sunless, T) proposals. Examples in the following section illustrate both methods.

4. Illustration

We have implemented this geometrical inverse simulation method of sunlighting in an experimental CAD tool. This tool may be used in any actual design practice by uninitiated designers. It must be as intuitive as possible and it has to provide real-time solutions to suit cognitive process. It has been written in C++ language on a Macintosh machine.

This section provides an overview of its main facilities illustrated with two demonstrative examples.

4.1. Overview

Our system is interfaced with a classical CAD software that enables to model any 3D complex geometrical scene. It provides interactive tools to define sunlighting proposals in such scenes : a mouse-pencil to draw the sunspot P and menus to choose the sunlighting qualification S and the lived time period T. Given these constraints, it computes the volume $\prod$(P, T). Then, according to the qualification S, it makes openings in the scene or it generates new shadings to achieve a sunless spot.

Openings that check on a *sunlit* proposal are computed as exact geometrical intersections between the sunlighting volume associated with the property and all the objects of the scene. If an object cannot be drilled, the proposal cannot be achieved and the process fails.

Otherwise, all the intersections are created. These openings must be then designed to get an architectural or urban relevance in the project context : windows, bays, places, and so on. Facing the ‘ raw holes ’ computed by the system, a designer is free to give his own architectural interpretation. Of course, the initial spot P is slightly altered as exact openings are transformed into realistic windows and bays.

Most of the time, the number of shadings that achieve a *sunless* proposal is infinite. Our system uses geometrical planes to compute them. It first suggests a standard plane that intersects the sunlighting volume and then, it enables the designer to modify this plane using graphical tools (translations and rotations). The shading is automatically transformed as and when the plane is. Thus, any kind of shading that makes sense in the design context can be easily modeled : vertical shadings at different scales such as walls, trees, façades ; horizontal shadings such as porches, roofs, terraces … blinds, sun visors, etc.

Shadings are first outlined as ‘ raw ’ geometrical shapes that is, they correspond to the exact intersection between the sunlighting volume and the current shading plane. When outlined, they have to be designed to get an architectural or urban relevance in the project context.

In addition, the system provides classical direct simulation facilities to compute the real sunlighting on any plane of the geometrical scene. These direct tools are useful to define existing sunlighting states before composing hypothetical ones.

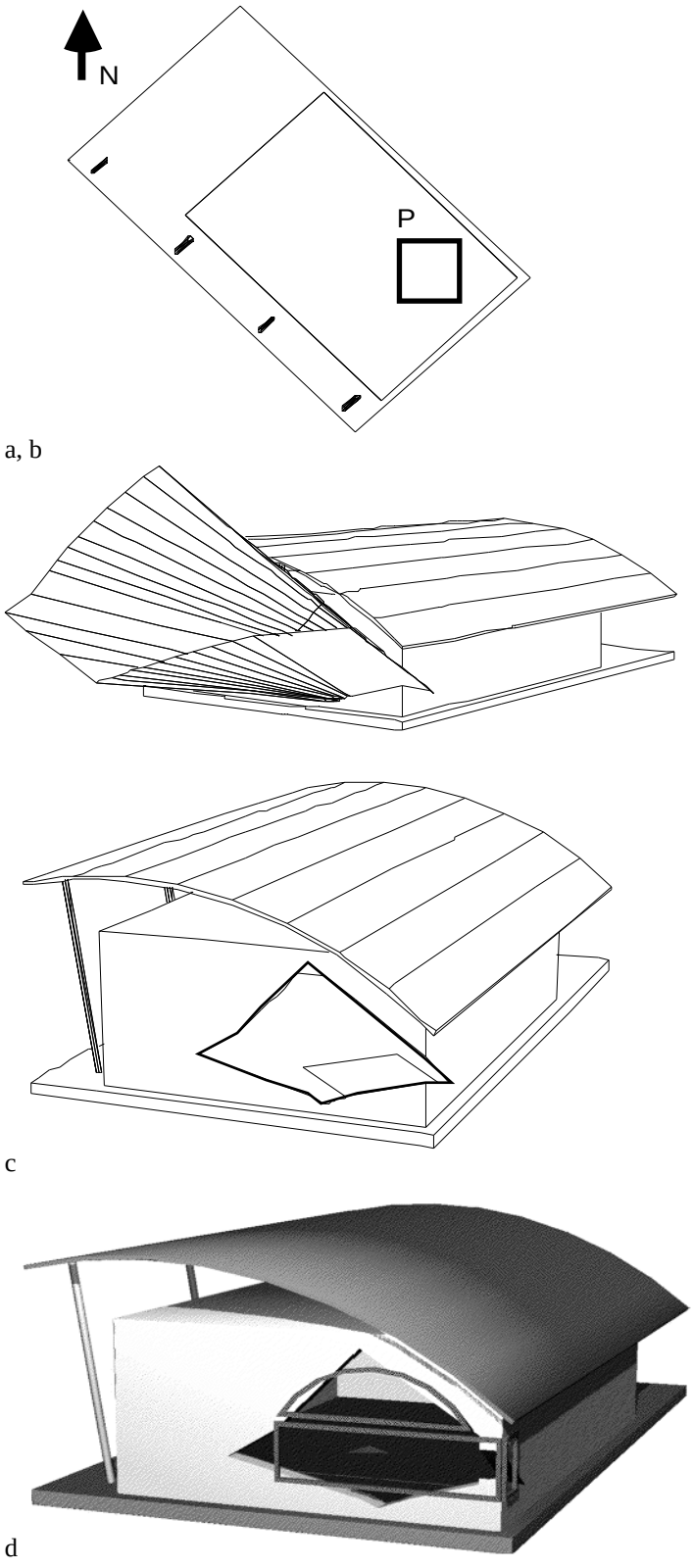

Figure 9. P, sunlit, the middle of the morning in winter

4.2. Demonstrative examples

Graphics on this page and the next one illustrate both methods (figures 9 and 10). The project concerns a small building in France (latitude 47° N) which volumetry has been first outlined following the program. Yet, two openings had to be made.

For the first one (figure 9 a, b, c, d on this page), the architect decided to enable the area figured as the rectangle P to be sunlit during the time period T the middle of the morning in winter. Given the proposal (P, sunlit, T), the system computed the sunlighting volume shown on (b) and then, created the exact opening drawn on (c). The architect followed his own interpretation to design the window figured over the hole in graphic (d). As this window recovers the largest part of the exact opening, the resulting sunlighting spot for the time period T closely matches the rectangle P.

The form of the second opening (figure 10 a, b, c, d on the next page) has been decided under the shape of the rectangle P on the South-West façade. For such a large bay, the problem is to avoid summer-time overheating. Thus, the proposal (P, sunless, the beginning of afternoon in summer) has been first composed and the system computed the associated sunlighting volume shown on graphic (b). The process the designer used to explore alternative shading planes that achieve the proposal cannot be figured here. The result is an incline plane intersecting the sunlighting volume through the exact shading polygon shown on (c). Following his interpretation, the architect finally turned this polygon into an architectural sun-visor and designed the required opening as shown on (d).

5. Discussion and perspectives

The system we present provides a basis for an inverse approach in sunlighting design. It computes a geometrical model figuring the whole space of solutions that achieve a given sunlighting proposal. In addition, it enables a designer to intuitively explore various solutions with the help of graphical tools. However, the single geometrical method has its own limits that we present and discuss now. In order to overcome these limits and to use sunlighting as a better formgiver, new tools have to be developed. We can outline the details of such tools in three different ways : firstly, the management of numerous proposals during the design process ; then, the achievement of fuzzy proposals that better fit the first designer's intents ; finally, the generation of realistic solutions according to the design context.

5.1. Management of numerous proposals

Many sunlighting intents may be proposed during the different stages of an actual design process. Our system deals with these sunlighting proposals by turns. A

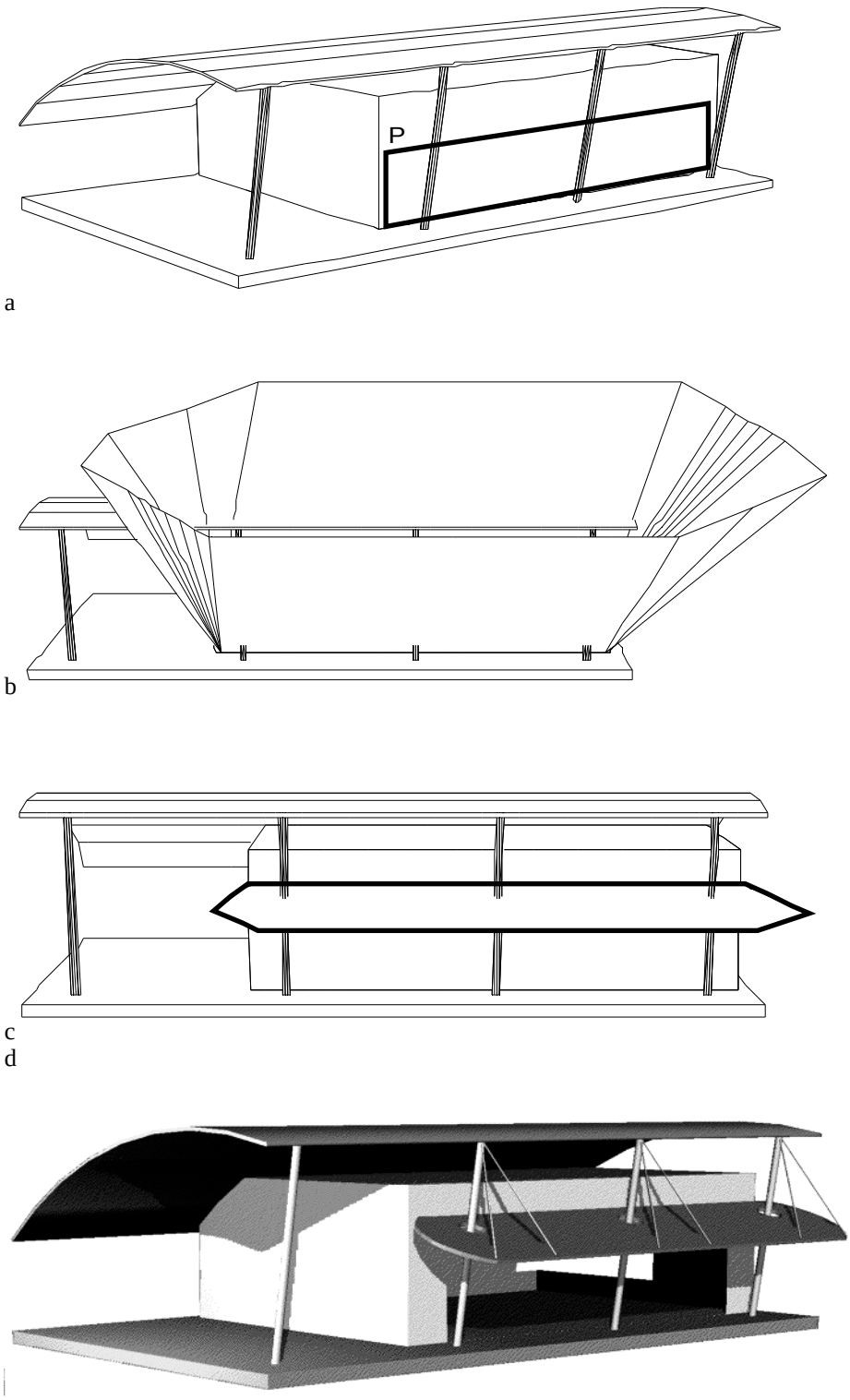

Figure 10. P, sunless, the beginning of afternoon in summer

more suitable method would be to manage all proposals together, that is to detect and to solve incompatible or redundant sunlighting effects during the design process. It is obvious that the achievement of a given sunlighting proposal indirectly affects sunlighting states of other numerous objects in the scene. How can a designer control together the shading he is designing and all other sunless shapes this shading may bring about elsewhere, for other time periods ? In particular, how can he be sure the proposal he tries to check on will not compromise another one he has achieved before — and maybe forgotten ?

A solution to this problem can be outlined as a consequence of our logic of sunlighting proposals. Let's consider two proposals a designer would like to achieve together — such as : P_1 sunlit the end of the morning in winter *and* P_2 sunless midday in summer, P_1 *or* P_2 sunless afternoon in July, and so on. The shadings or openings that realize such double constraints must check on some logical rules related to the sets of solutions associated to the proposals : to be part of both or not to be part of one of them. Such logical rules can be formalized in a geometrical way. If we compute the boolean intersection and differences between the sunlighting volumes, we generate various complex volumes, each of them being associated to a single logical state according to the proposals : one will embody the solutions for P_1 *and* P_2 sunless, another one for P_1 sunless *and* P_2 sunlit, etc. In other words, the logical operations between proposals can be performed using the equivalent boolean operations between their associated volumes.

Figure 11 on the next page illustrates the spatial cutting that two proposals P_1 and P_2 create. The parts A B C are defined as : A is the difference P_2 - P_1, B is the difference P_1 - P_2 and C is the intersection between P_1 and P_2. The part D is also defined as the intersection between P_2 and all the objects of the scene.

Given these volumes, one can easily draw the following logical rules : P_1 sunlit implies that B *and* C remain free, P_2 is *not* sunlit since the part D is not void, P_1 sunless has a solution in part B *or* in part C, D is a partial solution for P_2 sunless, and so on. A system reasonning on these rules could be associated to our inverse simulation tool. It would be able to infer knowledge between numerous proposals and to provide an expert guidance to manage them together. In our example, such a system would build relevant advices like : ‘ you would rather design a shading in part C for both P_1 and P_2 sunless / if you design a shading in part C for P_1 sunless, you will never get P_2 sunlit ’, etc.

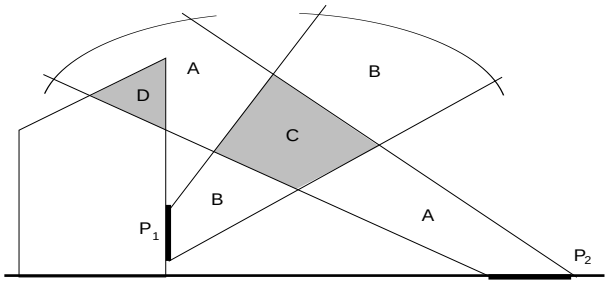

Figure 11. Spatial cutting for two sunlighting properties

5.2. Achievement of fuzzy proposals

However, this constraint management may be too rigid and severe for many sunlighting proposals. It might transform design space in a complex no-building network that would obstruct any travel … To avoid such deadlock, we will have to overcome the second limit of our system which concerns the binary qualifications sunless and sunlit. A designer would prefer to deal with fuzzy sunlighting qualifications such as : enough sunlit, not too sunless, and so on. That means he may accept that proposals are roughly, rather than perfectly, achieved.

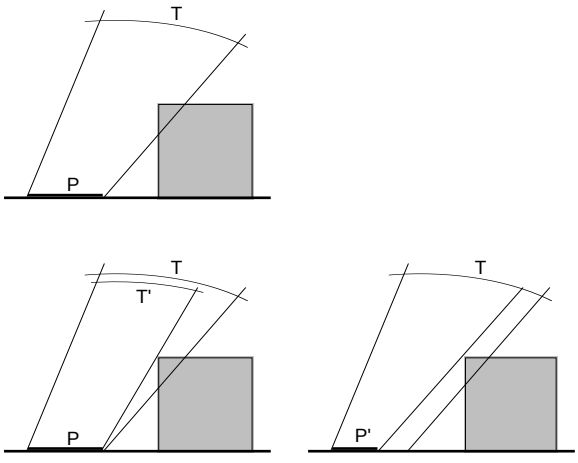

Figure 12. Half-satisfied proposal

The achievement of such fuzzy proposals could be made by combining direct and inverse approaches of simulation. Direct simulation provides the truth value for any sunlighting proposal. Considering this value, a system managing the inverse simulation engine could decide to set off, or not, the generation of solutions. Figure 12 is a demonstrative illustration of this process. The proposal (P, sunlit, T) has been composed. An evaluation of this proposal is firstly made using direct simulation tools. It produces various results : (P, sunlit, T) is not achieved but (P’, sunlit, T) is, and P’ closely matches P (figure 12, bottom right). From a temporal point of view, (P, sunlit, T’) is achieved and T’ closely matches T (figure 12, bottom left). Following these results, the system will infer that the proposal is ‘ enough true ’ or, in other words, that P is *enough sunlit* during T. Then, the property will be considered as half-satisfied and the gray cube will not be senselessly truncated, as it would have been using the hard way.

Obviously, any transformation of the architectural scene during the design process will constrain the system to update its first evaluation and to act in such a way that the given proposal still remains true.

5.3. Generation of realistic solutions

A third limit of our system concerns the shadings and openings generation. At the present stage of development, the system just computes exact geometrical objects that check on a given proposal. It is the designer's responsability to transform these objects into realistic shadings and openings that make sense according to the architectural or urban context. Won't it be better to get directly a set of relevant objects that achieve the proposal ?

The answer is actually ambiguous. On the one hand, the raw shapes our system computes do not impose any a priori sense to the solutions. From that point of view, they do act as a positive catalyst for the designer’s creativity. The raw hole bored in a façade, for instance, provides an interesting support for the design of an original window. In the same way, the interactive exploration of various shadings compels the designer to interpret the raw shapes as feasible architectural objects. Following this point of view, our system offers a stimulating aid in order to design openings and shadings that check on an architect’s intents concerning sunlighting.

On the other hand, this approach looks ineffective since the architect has to transform the objects generated to give them a real constructive and aesthetic relevance in the design context. In many cases, the resulting opening or shading will be related to some conventional and well-known architectural vocabulary. In these cases, the designer would obviously prefer to obtain directly the desired object rather than modelling it by himself using a raw template.

Following this point of view, our system should be able to generate realistic solutions as well as geometrical ones. This could be performed by associating a knowledge base of standard architectural shadings and openings to the inverse simulation tool. Such an expert system would ex-

plore various realistic solutions that fit the geometrical sunlighting shapes first generated. As a result, it would enable a designer to outline objects that not only achieve a sunlighting proposal but that also check on some architectural, aesthetic or constructive properties.

An interesting example can be found in the landscape design where vegetable shadings are used to realize environmental properties as well as visual ones. Such shadings look hard to model using the single geometrical tools. Furthermore, a designer working on these objects would like to obtain a full idea of their features as and when they are generated. Therefore, the help of an expert system seems more than just useful. For a given sunlighting proposal, such a system may propose various kind of shading trees, according to their height, their volume, their leaves in winter or summer, and so on (figure 13).

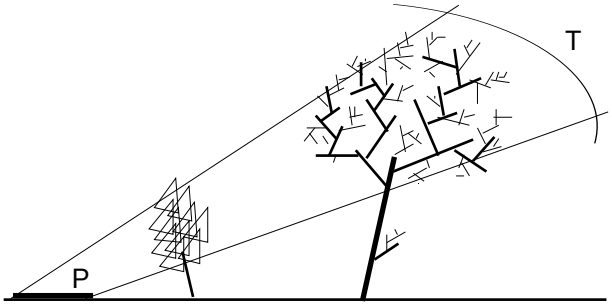

Figure 13. Two realistic shading trees for a same sunlighting proposal.

6. Conclusion

For several years, many resarch works have tried to make CAD tools suit design practice. In the field of environmental design, the single evaluative approach is now questionned as it occurs at the latter stages of decision. Such direct simulation tools just appraise or diagnose definite plans. They cannot take initial fuzzy intents into account and, finally, they look hard to use as a real design aid.

From that point of view, the so called inverse approach looks more efficient. Inverse simulation methodology provides a coherent frame to define intuitive models of environmental properties and to use them as formgivers in early stages of design — when key decisions are taken. As it starts from the result to get the architectural object, it accompanies the cognitive design process that first fixes the goal and then finds the method to reach it.

However, inverse simulation process should not rest on the single generative or optimization techniques. It should be able to produce a model of the whole set of solutions that satisfy the goal and to let the designer free to explore those solutions in a suitable way.

Our purpose was here to illustrate this general issue in the field of sunlighting design. The system we have developed provides a suitable sunlighting CAD tool. It enables a designer :

- to define, modify and visualize sunlighting constraints as geometrical volumes,
- to use sunlighting volumes just like any other architectural object, creating openings and shadings in a real environmental conscious way.

Many interesting tools should now be developed, like realistic architectural properties on shadings and openings, or intelligent logical reasoning on numerous sunlighting proposals. Concurrently, an equivalent approach applied to other environmental fields such as daylighting or sound simulation would offer many opportunities for new CAD research.

References

[1] P. Manning. Environmental Aesthetic Design, Identifying and achieving desired environmental effects, particulary image and atmosphere. *Building and Environment*, 26 (4), 1991.

[2] R.F. Woodbury. Searching for designs : Paradigm and practice. *Building and Environment*, 26 (1), 1991.

[3] W.M.C. Lam. *Sunlighting as formgiver for architecture.* Van Nostrand Reinhold Company, New-York, 1986.

[4] G. Stevens. *The reasonning Architect, Mathematics and Science in Design*. McGraw-Hill Publishing Compagny, New-York, 1990.

[5] L. Khemlani. GENWIN : A generative computer tool for window design in energy conscious architecture. *Building and Environment*, 30 (1), 1995.

[6] R. Oxman, J.S. Gero. Using an expert system for design diagnosis and design synthesis. *Expert Systems*, 4 (1), 1987.

[7] L. Adolphe. *L'aide à la décision technique dans la conception architecturale : application à l'énergétique du bâtiment.* Ph. D Thesis, Ecole des Mines de Paris, 1991.

[8] A.D. Radford, J.S. Gero. On the design of windows. *Environment and Planning B*, 6 (1), 1979.

[9] C. Schoeneman, J. Dorsey, B. Smits, J. Arvo, D. Greenberg. Painting with Light. *Computer Graphics Proc*., Anaheim California, 1993.