AN INTERACTIVE GRAPHIC INTERFACE FOR DAYLIGHTING DESIGN WITH SUPERLITE

Murray Milne and Upadi Yuliatmo,

UCLA Graduate School of Architecture and Urban Planning,

Marc Schiler,

USC School of Architecture

Los Angeles, California, USA

ABSTRACT
Fast accurate micro-computer simulations of the thermal, lighting, and energy performance of buildings offers the promise of informing architects' design decision-making. Yet this promise has only been partially realized, probably because of the mismatch between the way humans and computers communicate information. The full potential of microcomputer design tools depends on finding more effective ways for architects and computers to exchange information in a graphic or visual mode.

This paper describes a new daylighting design tool built around SUPERLITE, a powerful but cumbersome microcomputer program currently used primarily by the research community. The problem for architects is that SUPERLITE accepts data only from a formatted numerical input batch file (apparently a holdover from the days of punched card input), and only in terms of variables that are quite foreign to the way architects think about their buildings. Another problem is that SUPERLITE produces huge amounts of numerical output, which means that architects cannot instantly see the overall general trends or subtle differences that are essential for design decision-making, especially at the early phases of a building's design. Thus, although it is a potentially valuable piece of software, the design community considers SUPERLITE unusable because of the complexity of the input and output.

This new user-friendly design tool makes SUPERLITE easily accessible for architects and lighting designers, while also saving time for researchers.

INTRODUCTION

SUPERLITE was written originally at the Lawrence Berkeley Laboratory of the University of California (LBL) as a main-frame daylighting research program. Recently LBL released a public domain micro-computer version. The current joint USC/UCLA project was funded by the California Institute of Energy Efficiency (CIEE) to develope the new user-friendly interactive graphic 'front end' for the PC version of SUPERLITE. A new more user-friendly 'back end' is also now being proposed. Both the research and the design community should find this new tool useful.

DESIGN TOOLS

What exactly is a design tool? I would define it as a piece of software that is easy and natural for architects to use, and that shows them something useful about building performance.

A design tool may incorporate a complex building performance model or it may simply act as an interface to it. Similarly it may incorporate its own building description database or else it may access a CAD system.

A powerful design tool makes it possible to do something that is profoundly different from what architects could do before. They can bring their design to life, run it forward in time, and watch how it behaves - not just what it looks like, but how bright or dark it looks, how warm or cool it feels, what it sounds like, and how well it accommodates human activities. This opens a new world of opportunities that for future design tools that we are just beginning to explore.

BUILDING PERFORMANCE MODELS

Architects of course might elect to communicate directly with their building simulation routines, but the availability of an easy-to-use design tool that is more efficient and friendly is a much more attractive option, and in fact today very few architects have ever touched an energy simulation, daylighting model, or cost analysis program. Because although some architects have learned how to operate them, to be honest they require a great deal of training in arcane manipulation and otherwise unproductive effort. The amount of practice time required to maintain competency or to reach high proficiency is beyond the resources of the most architects.

A couple of distinctions are worth pointing out: phase, focus, and speed. In each phase of the design process some kinds of energy models are appropriate while others are not. For instance, at the very beginning of the design process a model like DOE-2 is useless; it requires the detailed input of every single building element, it takes a long time to execute, and produces reams of tabular data. Instead what is needed is a model that works with a minimal "fuzzy" description of the building, executes fast enough to keep the architect's mind from wondering, and presents results in a way that can be instantly grasped by right-brain thinkers.

DAYLIGHT IN BUILDINGS

Electric lights use about a third of all energy in buildings, and buildings in turn use about a third of all energy in the United States. Thus, the energy used for lighting is one of the largest single components of total energy consumption in the country. Anything that reduces lighting loads often pays off doubly; it saves the cost of lighting energy plus it also saves the cost of the air-conditioning required to get rid of the heat that the lights create. Taking advantage of daylight is an obvious solution.

This means architects and lighting designers need to know how to design buildings to capture and distribute just the right amount of glare-free daylight. That is why making SUPERLITE easily accessible to the design community is so necessary.

In Europe, interest in SUPERLITE and in daylight modeling in general is burgeoning. It is hoped that the development of this new user-friendly graphic interface will be seen as a contribution to the latest research currently underway in a number of Common Market countries (for example IEA Task XII, Subtask A.2).

HISTORY OF RESEARCH ON THIS PROBLEM

SUPERLITE uses a 'first principles' radiance approach to define by successive approximation the distribution of daylight (both beam and diffuse) throughout the interior of a room. This highly-regarded piece of research software has effectively become the international standard of comparison. Unfortunately, the data input is a holdover from the old punched card era, requiring a cryptic undecipherable string of numbers (Fig 1). The output is hardly better, being a long series of tabular printouts (Fig 2). This means that for all practical purposes SUPERLITE is unusable by the design community.

Originally, SUPERLITE was never intended for this purpose; it was written by Lawrence Berkeley Labs as a piece of research software. Today, however, with micro-computers exceeding the power of the main-frame computers of ten years ago, it is perfectly reasonable to use SUPERLITE to design actual buildings.

At UCLA over the past three years we have been developing and testing new user friendly micro-computer input

```
 2  2  4  1  1
 2  0  0  5  1  1  0
 10.0   5.0   2.7   3.0   0.0   3.0   5.0   90.  -90.  90.0 180.0  0.07  9 15
-1 -1  0  0  0  0  0 -1
 0   1.0  0.90 0.100  0
 10.0  10.0   2.7   3.0   0.0   3.0   5.0   90.  -90.  90.0 180.0  0.07  9 15
-1 -1  0  0  0  0  0 -1
 0   1.0  0.90 0.100  0
  0.0  12.0   0.0  15.0   0.0  15.0   2.7   90.   0.0   0.0   0.0   0.6  6  3
 0  0 -1  0  0  0  0  0
  0.0   0.0   0.0  12.0   0.0  12.0   2.7   90.   90.   0.0   0.0   0.6  5  3
 0  0  0 -1  0  0  0  0
 15.0   0.0   0.0  15.0   0.0  15.0   2.7   90.  180.   0.0   0.0   0.6  6  3
 0  0  0  0 -1  0  0  0
 15.0  12.0   0.0  12.0   0.0  12.0   2.7   90.  -90.   0.0   0.0   0.6  5  3
 0  0  0  0  0 -1  0  0
  0.0   0.0   0.0  15.0   0.0  15.0  12.0   90.   0.0   90.   90.   0.6  6  5
 0  0  0  0  0  0 -1  0
 15.0  12.0   2.7  12.0   0.0  12.0  15.0   90.  -90.  90.0 180.0   0.6  5  6
 2  1  2
-1 -1  0  0  0  0  0 -1
  0.0   0.0   0.8  15.0   0.0  15.0  12.0   90.   0.0   90.   90. 10  5
 0  0  0  0  0  0 -1  0
  38.  122.  8   0.0  2   0.2
 3  3  1 21   12.   12.   1.0
  3.0   0.1
```

Fig 1: Typical SUPERLITE data input file (.IN) shows why architects are not eager to use this program as a design aid. The second line, for example, shows that the space has two clear windows on one surface and five surfaces with no openings.

```
*************************************************************************
* DATA FOR WORKING SURFACE NODES; I=SURFACE, K=NODE-NO.                 *
*     X,Y,Z=COORDINATES,                                                *
*     S=ILLUMINANCE FROM EXTERNAL SOURCES (FOOT-CANDLE)                 *
*     R=INTERNAL REFLECTED COMPONENT (FOOT-CANDLE)                      *
*     I=TOTAL ILLUMINANCE (FOOT-CANDLE)  D=DAYLIGHT FACTOR (PERCENT)    *
*************************************************************************

      X = DISTANCE FROM FRONT WALL (PARALLEL TO FRONT WALL)
      Y = DISTANCE FROM ROOM CENTER LINE (PERPENDICULAR TO FRONT WALL)
          POSITIVE VALUES (LEFT OF CENTER LINE)
          NEGATIVE VALUES (RIGHT OF CENTER LINE)
      Z = HEIGHT OF WORK PLANE
```

K I		1	2	3	4	5	6	7
10	X	-1.18	-1.18	-1.18	-1.18	-1.18	-1.18	-1.18
	Y	4.50	3.00	1.50	.00	-1.50	-3.00	-4.50
	Z	32.30	32.30	32.30	32.30	32.30	32.30	32.30
	S	121.29	173.59	79.81	31.17	50.84	140.64	126.72
	R	57.57	52.81	54.07	53.47	54.95	54.36	59.27
	I	178.86	226.40	133.88	84.64	105.80	195.00	185.99
	D	6.64	8.40	4.97	3.14	3.93	7.24	6.90

K I		8	9	10	11	12	13	14
10	X	-3.54	-3.54	-3.54	-3.54	-3.54	-3.54	-3.54
	Y	4.50	3.00	1.50	.00	-1.50	-3.00	-4.50
	Z	32.30	32.30	32.30	32.30	32.30	32.30	32.30
	S	79.37	95.99	91.15	79.28	81.01	84.26	73.96
	R	55.65	55.60	54.68	54.48	55.31	56.58	56.12
	I	135.02	151.58	145.82	133.77	136.32	140.84	130.08
	D	5.01	5.63	5.41	4.97	5.06	5.23	4.83

Fig 2: Typical SUPERLITE output report of illumination on the workplane shows why it is difficult for architects to instantly grasp overall trends, or to detect subtle differences.

and output interfaces that will make it possible for architects, lighting designers, and even researchers to easily use SUPERLITE. For instance, one of our first projects demonstrated an input strategy combining plain-English interactive data input with a constantly updating three-dimensional picture of the building. A second project demonstrated our ability to read SUPERLITE output files and display the results graphically. A third project translated geometric data from an AutoCad drawing into the required SUPERLITE input format.

BACKGROUND

The design tool package described here an extension of our currently funded joint USC-UCLA Project that is now nearing completion of an input interface for SUPERLITE.

The USC team's responsibilities were to develope SUPER-IN, a new CAD-like graphic data input module for SUPERLITE, plus the overall project management. The UCLA team's responsibilities were to develop SUPERTEXT, a stand-alone translator that reads the geometric data files generated by SUPER-IN, then can add the required additional user-supplied data (like latitude, date, weather conditions), and finally writes out an 'official' SUPERLITE input data file.

STRUCTURE OF THIS PROJECT

The objective of this expanded project is to add a graphics output module plus a shell integrating all the separate pieces into a single micro-computer design-tool to give both researchers and designers a fully functional user-friendly interface to SUPERLITE. It incorporates all the software developed by the joint USC/UCLA project.

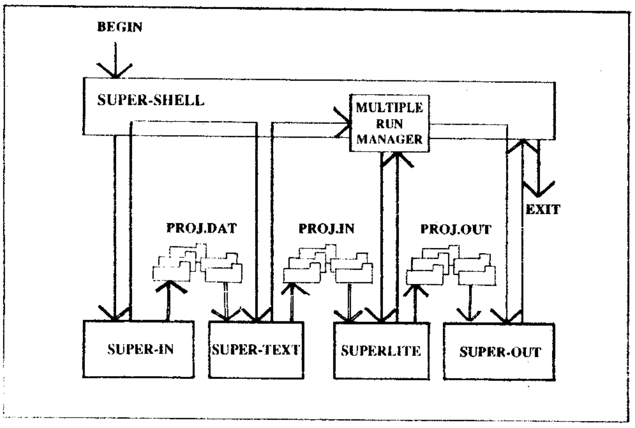

Fig 3: The complete SUPERLITE Design Tool consists of the shell that manages the whole process, and first calls SUPER-IN a CAD-like input module that writes (.DAT) files that are read in by SUPER-TEXT that translates them to the (.IN) files required by SUPERLITE which produces the (.OUT) files that are read by SUPER-OUT to produce the graphic picture of daylighting performance.

The structure of this expanded new design tool consists of a shell around SUPERLITE and four other separate stand-alone programs, each of which writes a data file that serves as input to the next program in sequence (Fig 3).

SUPER-SHELL: The shell is what the user invokes to start the process, but in fact it is invisible to the user. Its function is to manage the process by:

> Calling in order each of the four separate stand-alone modules,
>
> Testing to be sure that each module has written the correct file to serve as input to the next module in sequence,
>
> Managing multiple SUPERLITE runs to show, for example, how a space performs over two or more hours, or throughout a sequence of months.

SUPER-IN: This CAD-like interface allows the architect to graphically communicate three-dimensional data about the building to the computer. Using the mouse, the architect simply draws in the building following the traditional conventions of plan, section, elevation, and axonometric. Intuitive physical movements result in an extremely rapid and fluid tool for "making" a building. It is anticipated that SUPER-IN can also serve as a generalized graphic input tool for many different high-level building simulation programs. To describe a space the architect follows a logical sequence of steps:

> The floorplan is input much like a CAD system. A right click automatically closes any plane figures (Fig 4).
>
> The wall height and materials (i.e. surface reflectivity) are input in the sidebar. The program provides default material properties depending on what type of element is being input. These properties are displayed (and may be modified) in the sidebar.

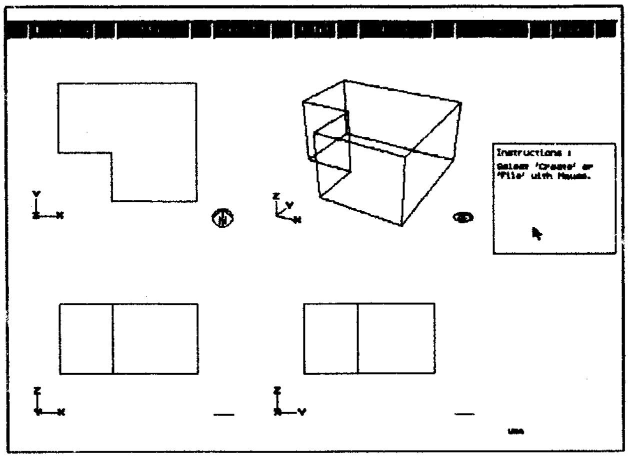

Fig 4: With the new SUPER-IN module, the mouse is used to draw in a floorplan, much like any CAD system. The wall height, materials, and surface properties are may be modified on the sidebar.

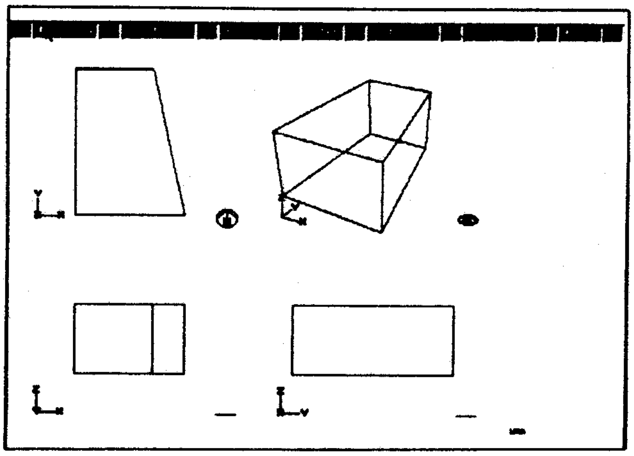

Fig 5: Once the floorplan is enclosed, the walls are extruded vertically and the object is stored as a connected and enclosed series of planes.

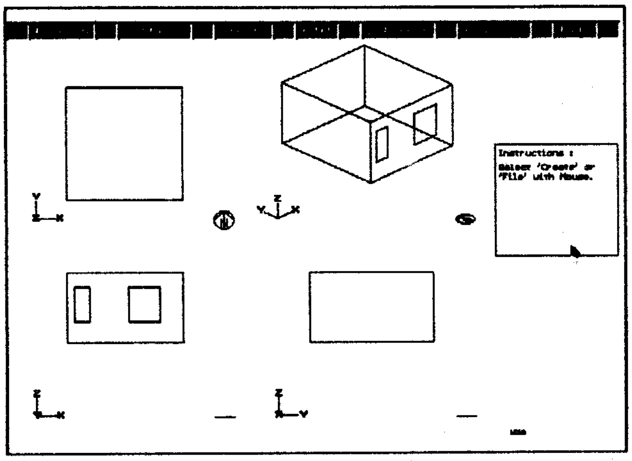

Fig 6: The wall surface is selected, and the windows are drawn onto it.

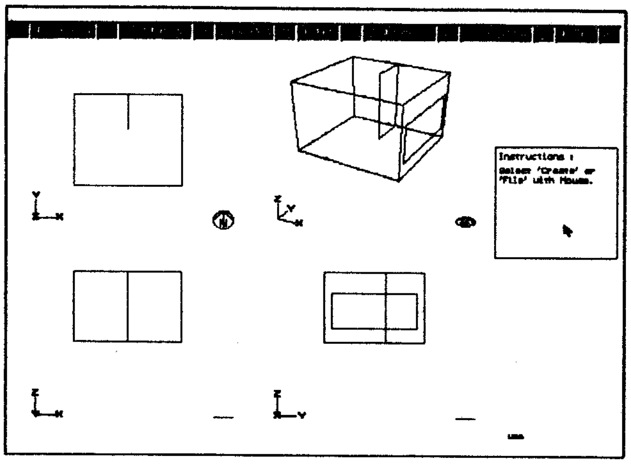

Fig 7: Planes can be attached to any surface, such as this freestanding partition in the room, or external sunshades (fins, overhangs, sills, or lightshelves), or even distant exterior obstructions.

When all the walls have been extruded vertically from the floorplan, the space is closed and the connected series of planes are stored as an object (Fig 5).

Windows and other openings can be drawn on any surface with the mouse (fig 6).

Free-standing planes like internal partitions, garden walls, adjacent buildings, can be attached to any surface. This also includes planes that serve as shading and sun control devices that protrude from walls like overhangs, fins, sills, and light shelves (Fig 7).

SUPER-TEXT: The primary function of this program is to translate building design data into the correct form for SUPERLITE. The main technical difficulty is to develop a transformation algorithm that converts the standard rectilinear coordinate data from SUPER-IN into the rather unusual trapezoidal coordinate system required by SUPERLITE. It also included a simple stand-alone program that reads and edits SUPERLITE input files and interprets the data in plain English. This program can also query the architect for the additional information not provided by the prior module.

Read in a file produced by SUPER-IN and translate it into the 'official' (.IN) format required as input to SUPERLITE (see Fig 1).

If not already provided it can ask the user for the required additional non-geometric information such as date, time, sky conditions, and latitude etc., and information on how many how multiple SUPERLITE runs are desired, for example:

-for all daylit hours of the selected design day
-for all twelve months at the reference hour
-for a given hour under various sky conditions
-comparisons of alternative building designs

Offer the option of looking at the SUPERLITE (.IN) input file in plain English, and using the editing function to make any desired changes (Caution: this option is intended more for the researcher rather than the typical design tool user, because there is no guarantee that the file as modified will be acceptable to SUPERLITE).

SUPERLITE: Here is the heart of the design tool. It is the standard unaltered version of SUPERLITE. Once it is invoked by the shell it will read the current project input data (.IN) file and will produce one or more standard output (.OUT) files depending on the number of runs requested. The operation of this program is invisible to the user, except for cryptic messages that occasionally flash across the screen (a single design run takes from three minutes to as much as 10 minutes on a 286 machine for each design hour).

SUPER-OUT: This program automatically reads the output data files that describe the distribution of daylight throughout the space produced, and it translates them into two-dimensional and three-dimensional graphic plots overlaid on drawings of the actual room. It can plot these data for most of the complex spaces SUPERLITE can calculate, including

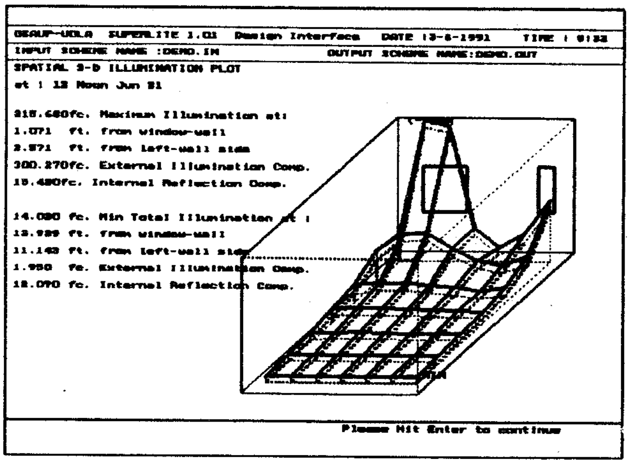

Fig 8: Spatial 3-D Illumination Plot shows graphically the total illumination as a result of daylight through the two windows plus the internal reflected component.

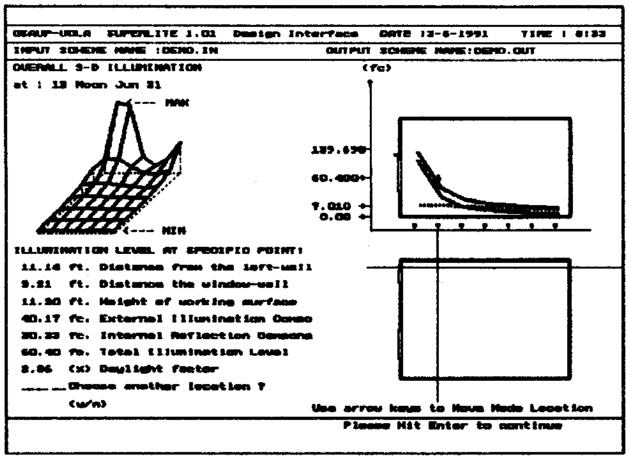

Fig 9: Illumination Level at a Specific Point in the room; the arrow keys move the cross hairs on the floorplan to select the point.

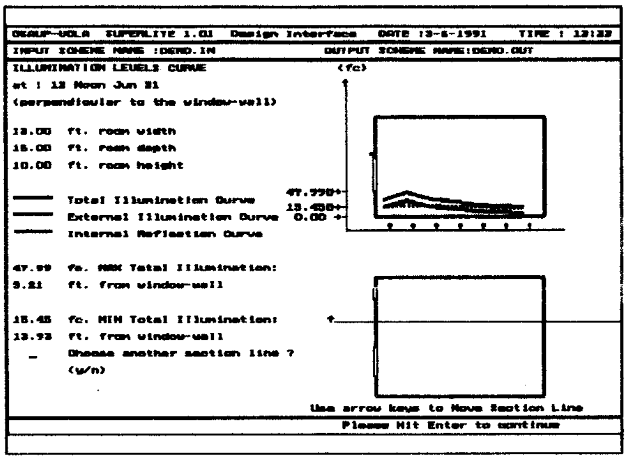

Fig 10: Illumination Level Curve plotted at the selected section plane; the cut-plane is moved by using the arrow keys

L-shaped rooms, non-orthogonal floor plans, internal partitions, and skylights.

Reads SUPERLITE output (.OUT) files, (accepts either simple or complex output)

Displays output data in various graphic forms:

> Spatial 3-D Plot of Illumination on Workplane (for a simple rectangular room see Fig 8)
>
> Illumination Level at any selected Point in the room (use arrow keys to select point; see Fig 9)
>
> Illumination Level Curves plotted on any Section (move cut-plane with arrow keys; see Fig 10)
>
> Multiple Illumination Level plots summarizing sequential runs (for example, showing all ten hours of the design day; Fig 11)
>
> Annual Performance Plots in 3-D for every hour of year (i.e. showing daylight of the minimum point in the room; Fig 12)
>
> Annual 3-D Plot of Total Light including Electric plus Daylight (assuming SUPERLITE soon offers this: Fig 13)
>
> Annual Plot in 3-D of any component of the Total Light (for example isolating electric light alone; Fig 14)
>
> Prints out Reports of all of the above,(especially the results of multiple SUPERLITE runs)
>
> > Tables of Illumination Levels for any Month (showing all components of the Total)
> >
> > Tables of Annual Performance (showing any components for each hour of each month)
> >
> > In addition, all of the standard SUPERLITE output tables can be plotted out (see Fig 2)

To illustrate our general approach, these feasibility studies have demonstrated our ability to handle all of the options in even the most complex room shapes that SUPERLITE can compute. These graphics show we can automatically plot SUPERLITE output data files for all kinds of complex room designs, for example:

> L-shaped rooms (Fig 15)
> Skylights (Fig 16)
> Internal partitions (Fig 17)
> Non-parallel floor plans (Fig 18)

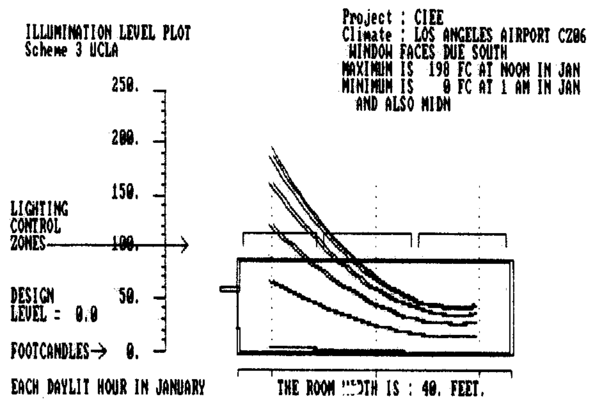

Fig 11: Illumination Level Plot summarizes sequential runs; here showing illumination curves for every hour of the design day plotted on an interior elevation of the room.

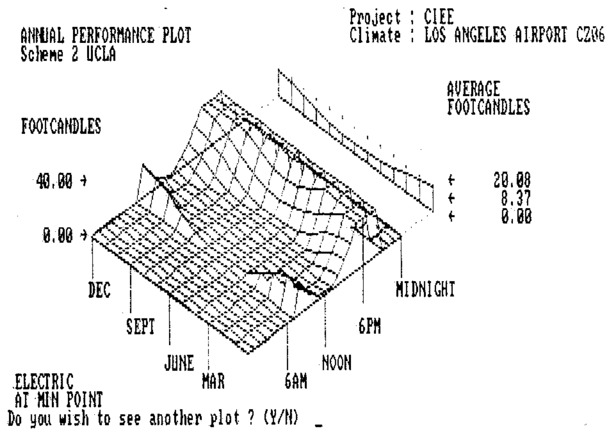

Fig 14: Annual 3-D Plot of just the Electric Light Component of Total Illumination (assuming the next version of SUPERLITE offers the option of photocell controlled artificial lighting).

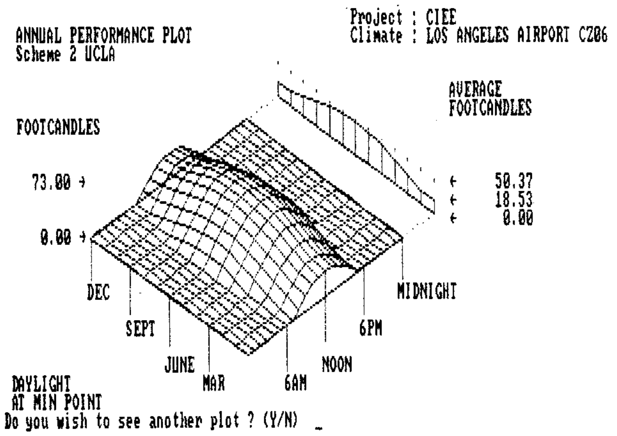

Fig 12: Annual Performance Plots in 3-D shows, in this example daylight for every hour of each month of the year, reaching a peak of 73 footcandles at the minimum point in the room.

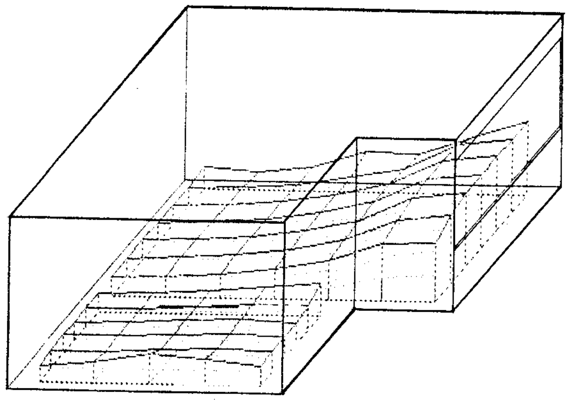

Fig 15: Spatial 3-D Illumination Level Plot for an L-shaped room generated by proof-of-concept software written for the proposed new design tool.

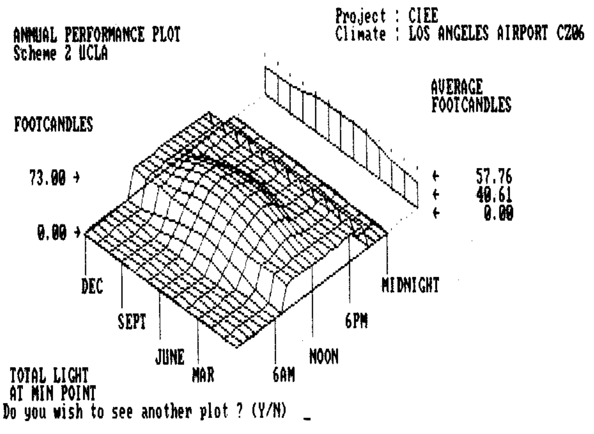

Fig 13: If photocell controlled artificial lighting is added to SUPERLITE as planned, the Annual Performance Plot would look something like this.

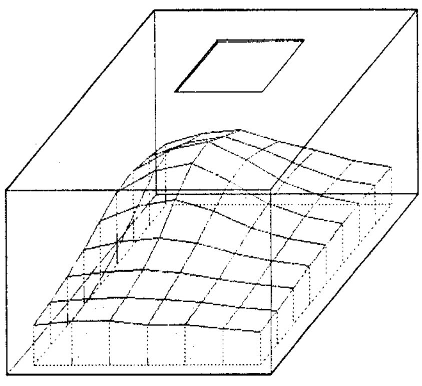

Fig 16: Spatial 3-D Illumination Level Plot of a room with one a skylight, but in fact this new interface will be able to accommodate any number of windows on any wall or roof.

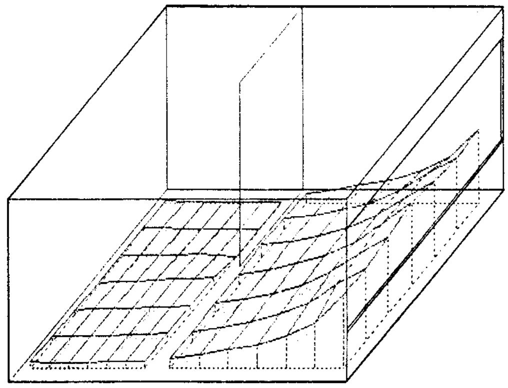

Fig 17: Spatial 3-D Illumination Level Plot of a room with an interior partition; freestanding planes can be attached normal to any surface to model such things as suncontrols or exterior obstructions.

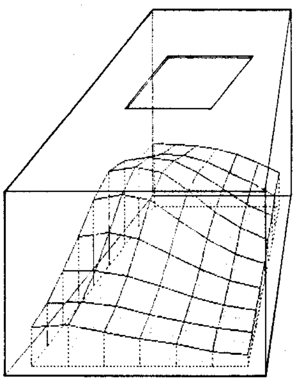

Fig 18: Even rooms with non-parallel walls can be modeled, in this case showing illumination from a skylight.

SOFTWARE DEVELOPMENT STRATEGY AND DISTRIBUTION

For the proposed new SUPERLITE 'back-end', the approach was first to demonstrate 'proof-of-concept' by writing software that quickly got up and running. These programs were essentially 'mock-ups' that are later re-written into professional quality 'bullet proof' code and expanded to accommodate all the cases that SUPERLITE can handle. More importantly, when a 'user-friendly' interface is fully developed additional support functions must be provided for such things as error trapping (to detect inappropriate input data), help functions (to explain new terms), and on-line documentation (to automatically print out the Users Manual).

The Software developed as part of this project will be some type of "shareware", which means that anyone is free to make copies for their own private use and to share them with others provided that copies are not sold for profit and that the authorship or ownership of this work is not misrepresented. In past projects we found that a good way to achieve this was to copyright all written material in the name of the Regents of the University of California, then to grant a free license to copy. Because the Users Manual is automatically printed out from the disk, simply copying someone's disk is a fast, effective, and cost-free distribution strategy.

BIBLIOGRAPHY

Chen, Jia-Yih, A Graphic Input System for Daylighting Design: Linking AutoCad to SUPERLITE, Master of Architecture Thesis, Graduate School of Architecture and Urban Planning, UCLA, 1990

Jareeboonsompote, Kanjana and Upadi Yuliatmo, SUPERLITE Design Interface, 403C Projects in CAD, unpublished, Graduate School of Architecture and Urban Planning, UCLA, 1989

Jareeboonsompote, Kanjana SUNLITE Input Design, Masters of Architecture Thesis, Graduate School of Architecture and Urban Planning, UCLA, 1990

Schiler, Marc, Eric James, and Murray Milne, Interactive Graphic Input for Daylighting and Energy Simulation, CAAD Futures '91, Zurich, July, 1991

Yuliatmo, Upadi, SUPERLITE Design Interface, Master of Architecture Thesis, Graduate School of Architecture and Urban Planning, UCLA, 1990

PERSONNEL

Murray Milne: For over fifteen years at UCLA Professor Milne has been developing "user friendly" interactive design tools for architects. His programs have won awards from the California Energy Commission, from the U.S. Department of Energy, and two awards from Progressive Architecture Magazine. As a lead-in to this project, his team at UCLA developed an interactive graphic input for the simplified version of SUPERLITE.

Upadi Yuliatmo: Currently on leave from Parahyangan University in Indonesia, Professor Yuliatmo is now in the midst of his PhD work at UCLA. As Research Assistant on the first phase of this project, he was responsible for writing the SUPER-TEXT program. Prior to the start of this contract, as part of his Masters Thesis, he wrote much of the proof-of-concept software that produces the graphics for the SUPER-OUT program, to be developed in phase two of this project.

Marc Schiler: In his recent sabbatical, Professor Schiler worked at EMPA in Switzerland on an IEA Task XI project which used SUPERLITE in conjunction with DOE-2.1C to analyze different daylighting strategies in different climates. As the Principal Investigator on the current CIEE-funded project to develop the new SUPERLITE Input Interface, he and Eric James at USC, wrote SUPER-IN the three-dimensional graphic data input module. He will be involved in the proposed second phase of this project to insure that the new SUPER-SHELL and SUPER-OUT will be compatible with SUPER-IN. He also has proposed to expand the SUPER-IN program to accommodate the input of artificial lighting systems.