Light Pen
Sketching Light in 3D

Thomas JUNG, Mark D GROSS, Ellen Yi-Luen DO
Design Machine Group, University of Washington, Seattle, USA

Abstract: We describe a lighting design system driven by sketching on 3D virtual models. Conventional lighting design tools simulate the lighting effects of design decisions such as window locations, surface treatments, and fixture placement. Light Pen takes the inverse approach by allowing the designer to indicate desired illumination on a 3D model. This serves as input to a knowledge-based lighting design system, which recommends what lights to use and where to place them, based on the designer's expressed intentions and on the geometry of the space.

1   MOTIVATIONS

Architectural lighting design is an important area of expertise that has implications for health, safety, and sustainable energy use. Until recently, lighting design has been viewed as a specialty area in building design but over the last twenty years — and especially since the California energy crisis of 2001 — lighting design has gained prominence as, increasingly, codes and guidelines for public buildings push for more economical, ergonomic, and efficient lighting systems. Much has been accomplished by adopting more efficient lamps (such as compact fluorescents) and switching systems and by recognizing the potential for using daylighting in place of electric lighting. In addition, lighting design itself can help: by placing illumination where it is needed rather than simply illuminating an entire space at work-light levels. Here is where lighting design expertise comes into play, and hence, intelligent support for lighting in computer aided design.

Computer support for architectural lighting design has been improving steadily since the early days of computer graphics (Greenberg 1974; Blinn 1976; Newell 1977), growing out of energy simulation programs such as DOE-2 and photo-realistic rendering techniques that enable a designer to propose and then simulate design configurations. Work on the simulation of lighting effects using, for example, ray tracing and radiosity techniques, has progressed to the point where lighting design applications (e.g., Lightscape, Lumen Micro) are used in professional lighting design alongside traditional physical simulations. Although simulation applications

help lighting designers understand the consequences of proposed configurations, they also require the lighting designer to specify a complete configuration of lamps and identify the colours, surface treatments, etc. of the architectural geometry. When we asked a professional lighting designer about using these simulations she indicated that although they are useful, "lighting designers really paint with light. What we'd really like to do is paint on the surfaces of the architectural model; then we work backwards, figure out where to put lamps to get the effect we want." (Erwine 2002)

Light Pen is a working prototype that lets designers sketch with light. It uses sketching in 3D as input to a knowledge-based lighting design decision-support system. The designer specifies where illumination is desired by sketching directly on surfaces in a 3D model, and Light Pen selects and places the light sources and then visually indicates their effects. However, it doesn't attempt to accurately represent the space lit.

The Light Pen system also illustrates a more general idea: using sketching and diagramming to interact with knowledge-based design systems. Earlier work explored this idea in two dimensions: posing queries to visual databases and setting scenarios for simulations (Gross and Do 2000); here we extrapolate this work to sketching on a three dimensional model. Light Pen therefore comprises two components: Space Pen, a system we built earlier for annotating computer graphics models on the Web (Jung 2001) and Lux, a simple knowledge-based expert system for lighting design decisions that we built just to illustrate this idea.

In the following sections we first describe a motivating scenario in lighting design and review related work in this area. We then outline the system and its different components: Space Pen to view and draw in 3D, the Geometry Analyser module and the Lux knowledge-based system. We conclude with a discussion and identify directions for future work.
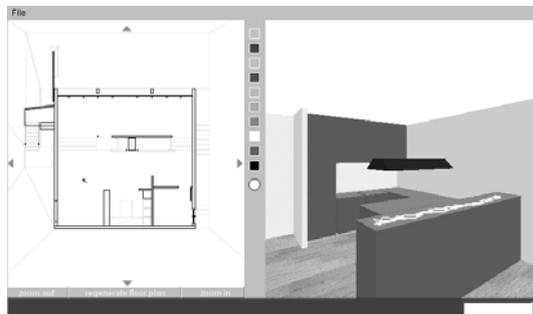
2 SCENARIO

Figure 1 The designer sketches light on the kitchen counter (right). The left frame indicates the designer's point of view.

Light Pen

Consider an architect designing a house. The architect designs and models the building using a conventional 3D modeller like Form•Z or ArchiCad and saves it in VRML format. The architect then imports the model into Light Pen to select and position lighting fixtures. After walking around the model, the architect, deciding that the kitchen counter, as a working surface, should be illuminated independently from the rest of the kitchen, picks up the light pen and sketches on the counter top (Figure 1).

The program registers the nearby surfaces and the size and shape of the light sketch mark and proposes two solutions for illuminating the area. The first is a series of three spotlights directed toward the work surface, aligned along the counter (Figure 2, left). The second solution is a pendant light close to the work surface illuminating the centre of the counter (Figure 2, right). The architect chooses the first solution and the Light Pen system adds the lighting fixture to the model and indicates its illumination pattern.
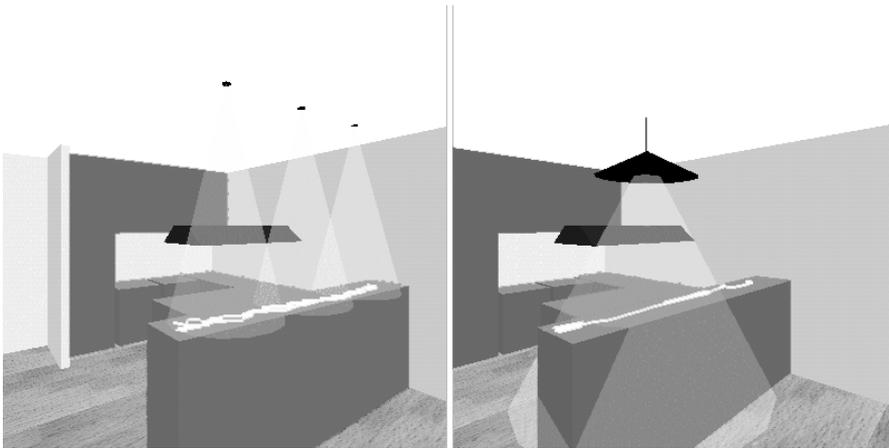

Figure 2 Light Pen proposes a series of spotlights (left) or a pendant light (right) above the working surface.

The architect would then request energy consumption and pricing information about the fixtures that Light Pen has selected, from a manufacturer-supplied database. Other solutions can be found for illuminating the entire kitchen area or a painting on the wall. Each time the system selects appropriate lighting fixtures according to the geometry of the surfaces that the light is painted on.

Our first prototype of Light Pen can already choose and display a few lighting fixtures. Multiple options and energy or pricing information are still to be implemented.

3   RELATED WORK

Since the early 1970s computer graphics research has continually improved techniques to accurately render lighting effects. Today products such as Radiance and Lightscape achieve excellent simulation of both daylighting and artificial lighting. However, rendering is still processor-intensive and time-consuming. Accurate renderings still cannot be generated in real time; therefore they can only be produced for still images or movies.

Light Pen operates in a real time 3D environment. The current version of Light Pen provides diagrammatic illumination rendering to illustrate the characteristics of a lighting solution rather than high fidelity rendering effects. Eventually, however, as in LightSketch (Glaser 2003), Light Pen could connect with high fidelity simulators to render a more accurate representation of the space with proposed alternative lighting solutions. LightSketch (Glaser 2003) is a 2D interface that enables a user to sketch light symbols (electric lighting and windows) and render the effects using a third party renderer such as RADIANCE.

In Light Pen's reverse lighting calculation the designer specifies the position of the illumination, rather than the position of the light sources. Others have also followed this approach. The "Painting with Light" system (Schoeneman et al. 1993) calculates the intensity and colour of a light source (at a given position) needed to produce a desired illuminating effect that the user paints on the surfaces of a 3D model. It then produces a precise representation of the space with the calculated light features. However, that system is designed only to calculate intensities and colours for lighting with fixed position light sources. Light Pen, in contrast, uses a simple knowledge based system to select and locate appropriate light fixtures in a building.

Poulin et al. (Poulin et al. 1997) sketch shadows and highlights on a 3D model to find the position of the most appropriate light source. However, their system has only one kind of light (a basic point light) and the position of the light source is mainly determined by the size and location of the shadow. It has no knowledge based system or fixture database. In the same vein the system of Ayatsuka et al. (Ayatsuka et al. 1996) positions objects (spheres) in 3-D given the location of shadows produced by three fixed light sources. Light Pen uses sketched highlights, the surrounding architectural geometry and a preset light database to propose the types and positions of architectural lighting.

Light Pen is a graphical interface to a knowledge-based system for lighting design. Knowledge based expert assistants for the engineering aspects of architectural design were the topic of considerable research in the 1980s. For example: HI-RISE supported the selection and dimensioning of structural systems (Maher 1985), and SOLAREXPERT supported solar design tasks (Rosenman 1987). These systems, however, were not integrated into a 3D sketching environment. In lighting design, products such as EcoLumens (www.tatainfotech.co.in) and LightCalc (www.lightcalc.com) propose alternative lighting of a whole building. You specify the size of your (rectangular) room, tell the system what kind of room it is (bedroom, hall, office, etc.) and the system displays a solution. You can then adjust by

specifying energy consumption or cost, and the system will propose a new lighting fixture. Solutions for placing the lights are limited to a few simple options.

4 SYSTEM OVERVIEW

Light Pen is a Java applet bringing together Space Pen in Java3D and Lux, which is implemented in Java. The virtual model is in VRML format and imported in Java3D through the VRMLLoader class; a Java library for VRML objects developed by the x3d task workgroup. Once converted to Java3D, the system must prepare the model to be drawn on by activating the picking capabilities of all its surfaces.

Figure 3 shows the internal structure of the Light Pen system, with its three main components: Space Pen, to import the 3D VRML model and draw on it, the Geometry Analyser module to query about the model's surfaces, and Lux, the knowledge based system that will choose and position light fixtures.
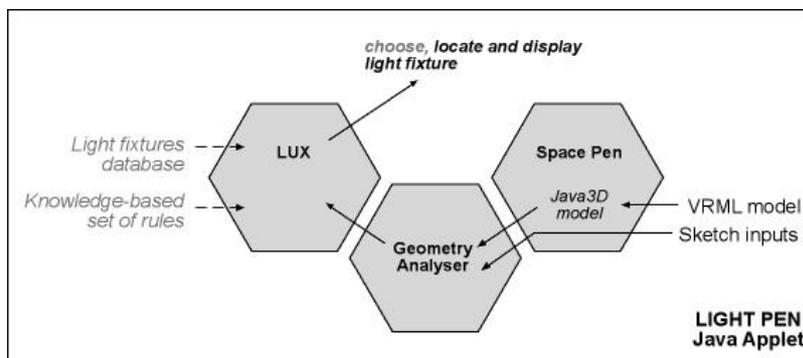

Figure 3 The Light Pen components, including the Geometry Analyser module.

Let us begin by explaining how Light Pen understands a "light object" (the sketch mark that the designer makes to indicate where light is desired) and the surfaces on which that light object is drawn. When a user marks on the model with the Light Pen, the system saves the geometry of the mark (coordinates, normals, size, and ink) and registers all the surfaces on which the mark has been made. Light Pen then analyses the registered surfaces in relation to each other and to surrounding surfaces. One might think that only seldom does a drawing touch more than one surface. However, if the user wants to indicate that an entire room is to be illuminated, it is likely that several surfaces of the room will be painted with the light pen. Also, when the model is transferred into VRML it is usually triangulated, so what appears as a single surface in the 3D environment is likely to be composed of several small triangulated surfaces. However, in most cases the registered surfaces for each light object can be reduced to a small set of coplanar ones.

Lux is a knowledge-based system for choosing and placing a lighting fixture. It demonstrates in principle how aspects of lighting design expertise can be represented in a set of condition-action rules. However, it is a simplification of a real lighting designer's expertise. The Lux expert system takes into account both the geometry of the light object and different aspects of the registered surfaces to determine which light fixture to deploy and how to deploy it.

4.1 Space Pen

Light Pen is built on top on a previous system we built called Space Pen. Space Pen is a collaborative tool that supports sketching on 3D web models. Architects and their clients communicate by sketching annotations on a 3D virtual model posted on the Space Pen web site. Clients and other team members can view their project in 3D before it is built and leave comments as post-it notes or drawings directly within the 3D environment. Later, designers can review those comments and modify the project accordingly.

Space Pen is above all an annotation system. The user can draw objects and symbols on any surface of the model. Sketched annotations are saved in VRML format for future reviewers to see. Post-it comments are indicated in the model as yellow boxes and a visitor can participate in a threaded discussion by clicking on a comment and typing in the reply text field.

Space Pen serves also as a navigation browser in a 3D environment. It automatically computes and displays a floor plan of the space and indicates the user's position and orientation. Light Pen uses Space Pen's environment and capabilities to navigate in the 3D virtual model.

4.2 The Geometry Analyser Module

Light Pen analyses some characteristics of the light sketch the designer draws (its shape, size, location, etc.) as well as the configuration of the surrounding surfaces. The goal is to identify surfaces composing the model, not only by their coordinates and appearance but also by their functions and common names. A horizontal surface placed at a height higher than 0.7 meters (and lower than 1.2 meters) from the floor is likely to be a desk or some kind of work surface. By identifying the type of object the user is drawing on, the Light Pen system can make better suggestions about what light fixture to choose.

Among the surface characteristics evaluated by the Geometry Analyser Module are height, size, orientation and co-planarity of the surface, existence of a ceiling nearby, distance to the ceiling, and. The Geometry Analyser returns values that Lux subsequently uses to infer the appropriate lighting solution. Each surface characteristic is calculated every time the designer draws a new light mark.

4.3   Knowledge Based System: Lux

Lux is a simple knowledge-based system for lighting design decisions. We built Lux only to illustrate how one might link the Space Pen to an expert decision-support system. Lux takes into account the geometry of a 3D model to make decisions about artificial lighting. The first implementation of Lux includes a database of light fixtures to choose from within the applet. In the next version, the light fixture database would be a separate text file containing a list of available lights with their characteristics such as type, intensity or colour.

Each time the designer makes a light mark on a surface, Light Pen analyses its geometry and its relation to surrounding surfaces. It labels each surface with a type, such as work surface, ceiling, wall or floor. Lighting designers seek to provide a lighting level appropriate to the activity that will take place in the space. The IESNA Lighting Handbook (IESNA 2000) identifies three different types of lighting: Ambient, Task and Accent. Ambient lights concern a whole space, task oriented lighting is used to highlight a certain work area, whereas accent lighting draws attention to a certain object such as a painting or sculpture. According to the surface type and the light sketch, Lux determines what type of lighting is needed (ambient, task or accent) and then proposes an appropriate lighting fixture to illuminate the chosen surface.

The configuration of the space and the user's intentions of where to illuminate it are two essential inputs to the lighting design's decision-making process. Lux uses a set of decision rules to first determine which lighting fixture to use and then its position. The first rule concerns the co-planarity of the registered surfaces. If the surfaces are not coplanar, the system infers that it is a larger space that needs to be lit. It must then find an ambient lighting system capable of lighting an entire space. It can probably eliminate direct spotlights from the list of possible fixtures, as spotlights are more often used for task or accent lighting.

To better understand the rest of the knowledge system let's consider the scenario above where the user wants to illuminate a kitchen counter. Figure 4 (below) shows the decision tree that drives the current Lux implementation. The arrows along the tree show the decision process for our scenario of lighting the kitchen counter.

As before, the system first determines that the surfaces on which the user has drawn (the counter) are coplanar. Then Light Pen checks the orientation of the surface. (A vertical surface won't require the same illumination as a horizontal one.) The counter is horizontal. Light Pen then checks whether there is a ceiling (or any horizontal surface) above the counter. If there is one (and in our scenario there is), it checks if the surface is in a small and enclosed space by calculating how far the ceiling is from the surface. The kitchen counter doesn't qualify so Light Pen then checks the height of the registered surfaces relative to the surface on which the user is standing (which it assumes to be the floor). In our case the illumination sketch was made on a kitchen counter 1 meter high from the floor, which registers it as a work surface. In Light Pen's knowledge base, illuminating a work surface calls for a task light, so a task lighting fixture will be needed. The last check concerns once again the distance to the ceiling. Light Pen makes sure that the distance between the

counter and the ceiling is not longer than the maximum distance where the light fixture would be effective. Because that distance is within the range of what, at least, one lamp could illuminate, Light Pen could suggest mounting either a spotlight or a fluorescent light under the cabinet above the counter surface.
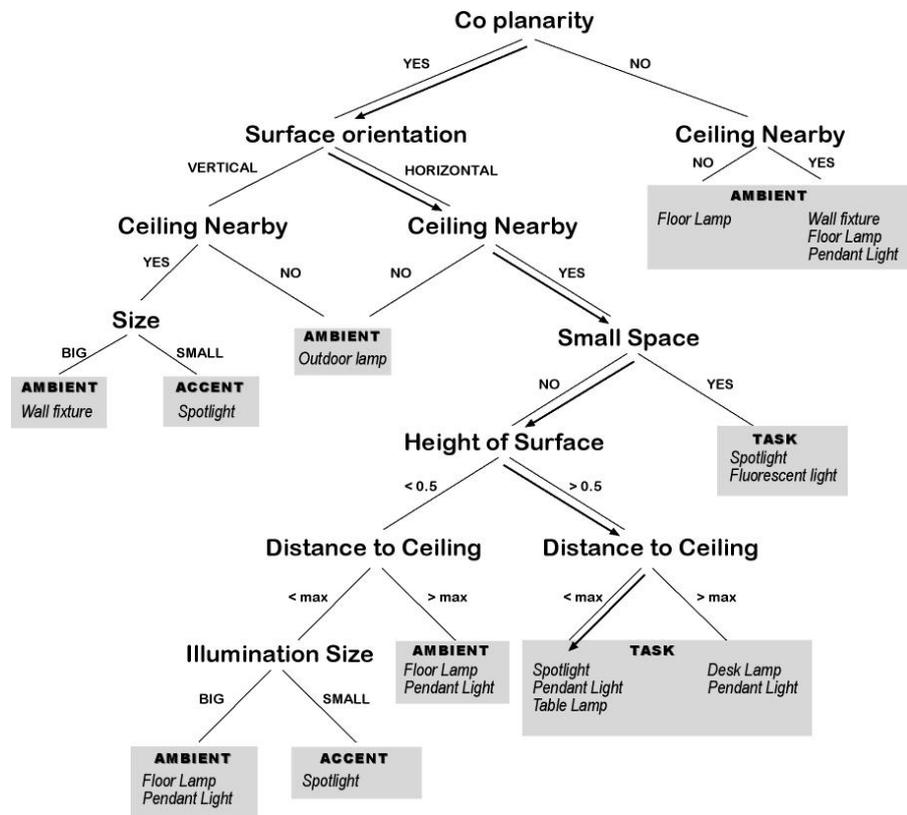

Figure 4 The decision tree used in the first implementation of Lux.

The Lux decision tree shows that each type of light fixture may be chosen by a basic set of rules on the surrounding geometry. Table 1 shows the set of conditions that drives Lux to choose a pendant light over another type of light fixture. Currently two different sets of rules both lead to a pendant light. One is for a large illuminated area on the floor when the ceiling above is higher than the maximum throw of other light fixtures. The other set of rules respond to the need for a large illuminated area over non-coplanar surfaces with a ceiling above.

Once Light Pen has decided which fixture to use, it must decide where to place it. This depends, of course, on the fixture itself, but also on some aspects of the geometry. For most cases (spotlights, pendant lights fluorescent lights), Light Pen finds the centre of the illumination sketch, projects it up toward the ceiling, and places the light source there.

Table 1 The decision rules for choosing a pendant light in Lux

RULES	EXPLANATIONS

(LIGHT? PENDANT-LIGHT)
[
    (((DETERMINE-SKETCH-COPLANARITY?)YES)
    ((DETERMINE-SURFACE-ORIENTATION?)HORIZONTAL)
    ((FIND-CEILING-NEARBY?)YES)
    ((DISTANCE-TO-CEILING?)(>min)(>max))
    ((HEIGHT-OF-SURFACE?)0)
    ((SKETCH-SIZE)BIG))
or
    (((DETERMINE-SKETCH-COPLANARITY?)NO)
    ((FIND-CEILING-NEARBY?)YES)
    ((SKETCH-SIZE)BIG))
]
	Rules to choose a PENDANT LIGHT:
	Option 1:
	    Surfaces have to be coplanar
	    The surface it's been drawn on should be horizontal
	    There has to be a ceiling nearby
	    The distance to the ceiling can be high
	    The target illumination must be on the floor
	    The size of the illumination sketch is big
	Option 2:
	    The surfaces are not coplanar
	    But there is a ceiling nearby
	    The illumination sketch is big

Light Pen combines a sketching interface (Space Pen), a knowledge-based system (Lux) and a database of light fixtures. Although the implementation is at a preliminary stage and the rules of the knowledge system reflect only a small part of the decision process of a lighting designer, Light Pen demonstrates how an expert system can be linked to a graphical user interface, all in a java applet, running over the Web.

5   DISCUSSION AND FUTURE WORK

Light Pen may develop in several directions. We could further develop the knowledge-based system to more accurately reflect the decision making of a lighting designer. Other elements must then be taken into account: the material of the surrounding surfaces (reflective surfaces, absorbing surfaces, etc.), the location of openings (windows, glass doors) and the building type. Buildings can be categorized by their purpose, size, location, each of which influences lighting design: The illumination of a hospital in Seattle has different requirements from a classroom in Casablanca or a private home in Moscow.

We are also considering how to take into account the effects of daylighting on a 3D model. We would like our system to identify skylights, windows and the sun path in the model in order to make more informed decisions about the light fixtures and their locations. Dan Glaser's project (Glaser 2000) analyses daylight inputs in a building and displays results in a two-dimensional representation of the space.

Finally we could improve the Light Pen capabilities by being able to not only specify the location of the illumination but also the intensity or/and colour of the light. In that matter, we would not only sketch with light, but paint with light as Schoeneman and Dorsey (Schoeneman et al. 1993) proposed. Linked to our knowledge base the system would be able to choose different fixtures and place them automatically.

REFERENCES

Ayatsuka, Y., S. Matsuoka and J. Rekimoto. 1996. Penumbrae for 3D Interactions. *Proceedings of the ACM Conference on User Interface Software and Tools (UIST)*, 165-166. ACM Press.

Blinn, J.F. and M.E. Newell. 1976. Texture and Reflection in Computer Generated Images. *Communications of the ACM*, vol. 19, Issue 10, 542-547.

Erwine, B. April 2002. *Personal Communication*.

Glaser, D. and S. Ubbelohde. 2001. Visualisation for Energy Efficient Building Design. *Extended abstracts of the IEEE Symposium on Information Visualisation.*

Glaser, D., J. Voung, L. Xiao, B. Tai, S. Ubbelohde, J. Canny and E.Y.L. Do. 2003, LightSketch: A Sketch-modelling Program for Lighting Analysis. *Proceedings of CAADFutures 2003: Digital Design: Research and Practice,* eds. M-L. Chiu, J-Y. Tsou, T. Kvan, M. Morozumi and T-S. Jeng, 495-505. Dordrecht, The Netherlands: Kluwer Academic Publishers.

Greenberg, Donald P. 1974. Computer Graphics in Architecture. *Scientific American*, vol. 230, no. 5, 98-106.

Gross M.D. and E.Y.L. Do. 2000. Drawing on the Back of an Envelope: A Framework for Interacting with Application Programs by Freehand Drawing. *Computers and Graphics*, Vol. 24 (6), 835-849.

Jung T., M.D. Gross and E.Y.L. Do. 2001. Space Pen: Annotation and Sketching on 3D Models. *Proceedings of CAADFutures 2001*, eds. B. de Vries, J. van Leeuwen, and H. Achten. 257-270. Eindhoven: Kluwer Academic Publishers.

Maher, M.L. 1985. HI-RISE and Beyond: Directions for Expert Systems in Design. *CAD Journal*, 17(9), 420-427.

Newell, M.E. and J.F. Blinn. 1977. The Progression of Realism in Computer Generated Images. *Proceedings of the Annual ACM/CSC-ER Conference*, 444-448.

Poulin, P., K. Ratib and M. Jacques. 1997. Sketching Shadows and Highlights to Position Lights. *Proceedings of IEEE Computer Graphics International Conference*, 56 -63.

Rosenman, M.A., J.S. Gero, R.D. Coyne, et al. 1987. *SOLAREXPERT: A Prototype Expert System for Passive Solar Energy Design in Housing,* vol. 2, 361-370. Canberra: Australia- New Zealand: Solar Energy Society.

Schoeneman, C., J. Dorsey, B. Smits, J. Arvo and D. Greenberg. 1993. Painting with Light. *Proceedings of the 20th annual ACM Conference on Computer graphics and Interactive Techniques (SIGGRAPH)*, 143-146.

IESNA. 2000. *Lighting Handbook - 9thEdition*. Illuminating Engineering Society of North America.

Light Pen

Ecolumen: www.tatainfotech.co.in

LightCalc: www.lightcalc.com

337