LiQuID: Lighting Quality for Design

James Peng, Ben Liao, Daniel Glaser, John Canny
University of California, Berkeley

Ellen Yi-Luen Do
University of Washington, Seattle

Abstract

In this paper, we present LiQuID, a tool for seeing lighting quality in design. Photographs are useful vehicles for both describing and making assessments of architectural lighting systems. A significant barrier to using photographs during the design process relates to the sheer volume of renderings that needs to be analyzed. Although there have been efforts to produce novel visualization systems to manage large sets of photographs, this research aims to reduce the complexity by classifying data into representative prototypes. A hypothetical case study is discussed.

LiQuID: Lighting Quality for Design

James Peng, Ben Liao, Daniel Glaser, John Canny
University of California, Berkeley

Ellen Yi-Luen Do
University of Washington, Seattle

1 Motivation and Background

In the past decade, there have been significant advancements in CAAD for both describing and assessing architectural lighting. The former is about deriving numbers and rendering images to describe a lighting system. The latter deals with issues between lighting systems and occupants, energy consumption, and thermal interactions[1]. Assessment and description are not mutually exclusive activities. Lighting designers calculate daylight factors during the design process to "see light" across multiple sky positions during the design process (Erwine 1999). Moeck and Selkowitz (Moeck and Selkowitz 1995) argue the value of using photorealistic images during the analysis phase of lighting design. Performance indices may not be quantifiable upfront since, for example, precise metrics of glare are still unknown[2]. Hence, lighting design and analysis is both about imagery and numbers[3].

Generative design characterizes the approach of model building and testing under a variety of simulation conditions. Daylight varies by viewpoint, time of day, day in year, and sky condition (Figure 1). A major limitation of generative design tools is that there is "too much to describe," and analysis can be short circuited before the right data is looked at. Lighting is a complex, multivariate problem, and just generating performance graphs or images may not guarantee a comprehensive view of the data. Although certain design programs (Moeck and Selkowitz 1995), (Glaser, Voung et al. 2003), and (Papamichael, Lai et al. 2002), offer innovative ways of looking at design, they require a generate/test cycle to assess lighting performance. Due to the large parameter space, though, it is difficult to generate an effective number of test cases. For example, in most of these programs, the user would have to click a mouse button thousands of times to examine a year of data! This problem is partially overcome with more condensed or sophisticated visual representations (e.g. (Cheng and Pat-Yak Lee 2001; Glaser and Ubbelohde 2002), but they do not afford in depth analysis of lighting performance.

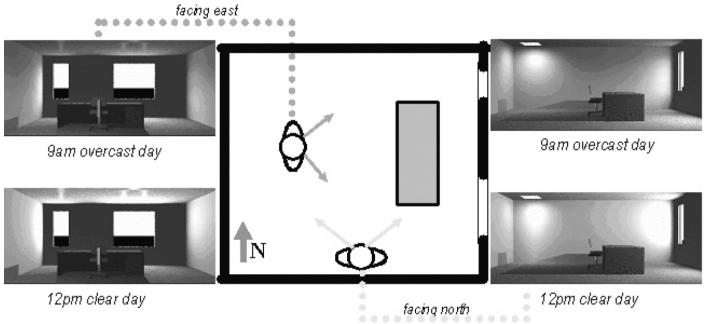

Figure 1. Lighting quality varies by viewpoint, time, and sky conditions.

We propose using clustering as a primary method for improving the process of lighting quality analysis in a generative design environment. It is based upon the theory that similar data, whether numerical or visual, can be grouped together to reduce the problem size (Aldenderfer and Blashfield 1984) (Marks, Andalman et al. 1997). Clustering does not hide important details about lighting quality that are lost with reductive performance analysis methods. Instead, it allows an architect to see the major trends in their data for focusing their description/analysis cycle. This paper presents the design tool LiQuID (Lighting Quality In Design) which aids in the classification and visualization of lighting quality data.

2 Generating Parameter Spaces in LiQuID

LiQuID develops a comprehensive parameter space for lighting quality analysis. It starts with the room layout and deconstructs it into both abstract geometric planes and perspectives. LiQuID takes the approach that building occupant's visual needs are not limited to a single perspective, but instead will vary according to a number of factors. When the viewpoint changes, so does the geometrical region of interest. Figure 2 illustrates the four-stage process that LiQuID uses for judging lighting quality.

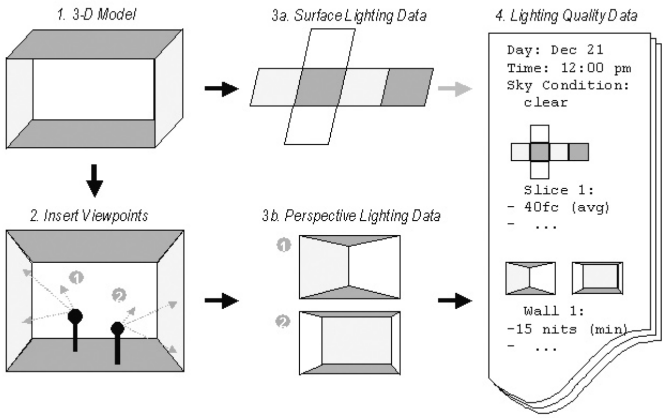

Figure 2. LiQuID generates lighting quality data in four stages. It starts with a 3D model with both electric and daylight sources (stage 1). Viewpoints are inserted at stage 2. At stage 3, both surface and perspective lighting data are generated for a range of sun angles and sky conditions and tabulated in stage 4. Note the gray arrow denotes future work.

2.1 Analysis of Geometry Files

LiQuID relies on building geometry and simulation data from the program LightSketch (Glaser, Voung et al. 2003) to perform its analysis. LightSketch is a sketch-based interface for designing architectural models with both natural and daylight features. After the user has completed the design of a space in LightSketch, the geometry of room is stored into RADIANCE (Ward 1994) compatible files. These geometry files contain information about the dimensions, layout, and locations of the various features within the building design. The size and orientation of wall surfaces are described as adjoining collections of three-dimensional coordinates. The locations and size of windows and skylights are similarly instantiated, while the geometry files regarding the luminaires also contain additional information about the luminous intensity and cutoff angle.

By examining the general layout of a building design, architects and lighting designers are able to employ certain heuristics to judge the performance and visual quality of their design. This is not a definitive measure of lighting quality, as it does not take into account individualized lighting conditions and specific concerns about the function of spaces. Nevertheless, industry standard handbooks (e.g. (Illuminating Engineering Society et al. 2000; Benya et al. 2001) provide a straightforward methodology to critique and improve upon designs. Directly from the geometry files, LiQuID can calculate the ratio of wall size and window size and note the walls with window-to-wall-size ratios of less than the recommended 20% (Illuminating Engineering Society et al. 2000). Examining the effects of luminaires upon actual viewers within a room is a complex issue, but can be partially addressed through simpler calculations. For example, LiQuID can calculate the location and range of sight from the viewer's perspective in comparison to the location and angle of a luminaire to identify direct glare.

2.2 Specifying and Locating Viewpoints

To generate luminance information, viewpoint locations and directions must be identified. This can be done either manually or automatically. Entering manual viewpoints allow for design-specific information such as identifying a place in the room that a person will likely be sitting at. Viewpoints can also automatically be generated by using the simple heuristic of looking at multiple directions at the center of the room. The emphasis on multiple views allows for a more descriptive analysis than a single view determination of lighting quality.

2.3 Surface and Perspective Data

The simulated lighting patterns within a building design are represented as a grid of luminance or illuminance values, called a slice. Each slice can represent a vertical or horizontal plane within the room, or it can represent a perspective from a viewpoint within the room. Slices derive their values from arrays of numbers in input text files or from input image files. Slices were chosen to be approximately 200x100 for images, and 20x20 for illuminance data. They are typically generated from a 12x6 sampling of a sky database, taking approximately 4 minutes with a 1.3Ghz Pentium III processor.

2.4 Lighting Quality Data

The lighting quality data that LiQuID creates allows for both formulaic and photo analysis. The quantitative analyses are still relatively basic compared with the clustering for photo analysis described in the next section. This includes determining window to wall ratios, checking sight angles for direct glare with electric light fixtures, and providing luminous intensity ranges. They are described in more detail in Section 4.

3 Clustering in LiQuID

After sketching a design and performing multiple lighting simulations across different variables, images are passed into LiQuID for analysis. A simulation is defined as the calculation of luminance and illuminance values for the design at a single specified period of time. For each simulation, images are created for all room surfaces and viewpoints within the design. Images representing one simulation are compiled into data structures called DayStructs and then fed into a clustering process. The clustering process groups DayStructs according to their similarity across a lighting quality variable, such as pattern, glare level, or uniformity of lighting. Currently LiQuID follows a data-driven approach, clustering the DayStructs according to similar lighting patterns in their image files and then performing analysis on the cluster results.

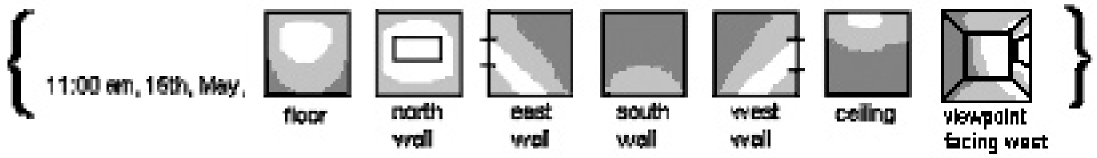

Figure 3. An example DayStruct: It is composed of a time/date header and followed by 2d planes known as Slices

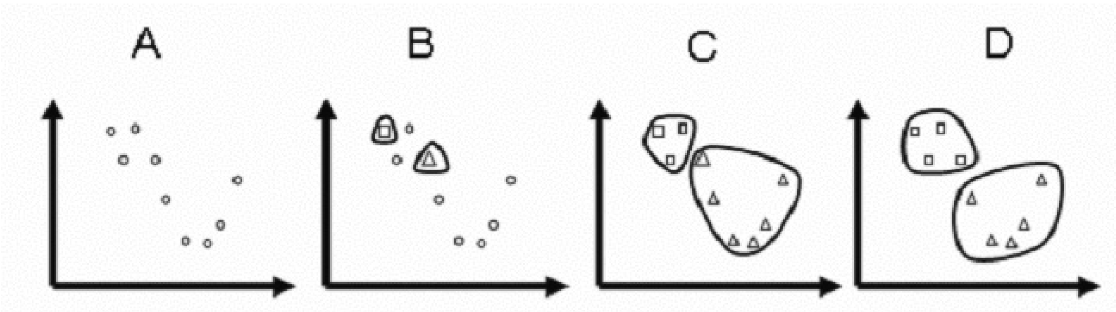

Figure 4. A) Initial Data B) Selection C) Assignment D) Reassignment

3.1 Data Structures

LiQuID stores lighting quality data in daystructs composed of multiple slices. A slice represents one viewpoint or one surface of the model for a single simulation. For example, it could represent the lighting luminance values of looking at the north side of a room March 2nd at 2 pm. A daystruct contains multiple viewpoints, luminance or illuminance values that may be pertinent for analysis of the 3D model for that period of time. It is essentially a collection of Slices with a time/date header Figure 3.

3.2 Clustering Algorithm

One of the problems that designers face is the large amount of lighting simulation data that would have to be sorted through and analyzed to accurately understand the lighting quality of their designs. Taking simulation samples randomly throughout the year to try to get a sample representation or a classification of a design's lighting quality is often tedious as well as specious, as a sample selection can be biased or not comprehensive. Therefore, the simulator program that LiQuID utilizes must uniformly and thoroughly simulate the model through different lighting conditions throughout the specified time of day, month and year, eliminating the possibility of selection bias.

To deal with a large amount of lighting simulations requires an algorithm that could take data and classify it into smaller representative units while still retaining most of the relevant information. Clustering algorithms, in general, take sets of data and partition them into groups that are "similar", as well as finding representative units for each partition. Aldenderfer and Blashfield (Aldenderfer and Blashfield 1984) have defined a clustering algorithm as a multivariate statistical procedure that starts with a data set containing information about a sample of entities and attempts to reorganize these entities into relatively homogeneous groups . For example, if a DayStruct set contained 33 simulations with extremely "high glare", 24 simulations with "uniform light", and 1 simulation which was "very dark", then a clustering algorithm would group the 33 high glare DayStructs into a "high glare" cluster, the 24 uniform lighting DayStructs into a "uniform lighting" cluster, and the 1 dark DayStruct into its very own set. Now the old set of 58 (33+24+1=58) DayStructs can instead be represented by 3 DayStructs and 3 numbers indicating the frequency that each DayStruct occurs in the set. Similarly, LiQuID clusters its simulation data and classifies them such that large quantities of similar DayStructs are represented by fewer DayStructs.

3.3 K-means Algorithm

Clustering algorithms mainly fall into two categories: hierarchical and non-hierarchical. The first type of clustering, mostly composed of algorithms known as agglomerative clustering, takes data and combines similar elements from the bottom up. In LiQuID's case, it would continuously make passes through the DayStruct set and in each pass, combine the two "most similar" DayStructs, forming them into a new cluster and thus reducing the set's size by 1. The process continues until there is only the desired number of clusters left. Agglomerative algorithms may take a large amount of time, especially in finding the "most similar" clusters to combine in each pass. Non-hierarchical clustering methods instead attempt to determine the representative DayStruct of the resulting clusters beforehand and build from there. Instead of comparing every DayStruct against each other to determine the two most similar DayStructs, each DayStruct is only compared against representative DayStructs for each pre-selected cluster. Non-hierarchical methods are often arbitrary in selecting the beginning clusters and can make unwise initial selections (like selecting extremely similar DayStructs to form two different clusters). This situation could result in bad cluster formations, but on average, non-hierarchical methods are fairly quick and relatively accurate (He 1999). Currently, LiQuID implements a non-hierarchical clustering method called K-means. The K-means algorithm is a non-deterministic clustering method. In general, the K-means algorithm initially chooses k number of DayStructs to form clusters and then slowly refines each cluster to optimize its results. One tradeoff for K-means though is that the more clusters that DayStructs are partitioned into, the more accurate and detailed the results are, but only at the cost of slower processing time. Therefore at the risk of over-allocating memory resources, more detail and more clusters may want to be generated for more detailed results.

Figure 4 illustrates the principles of his K-means algorithm. The figure shows K-means' four main steps ("A" through "D"). "A" describes the initial set of data. The pictures, stored as DayStructs, are represented by x-y coordinates on a 2d plane. This is a common technique in multidimensional scaling to manage complex data such as images. In step "B", the Selection step is to choose a specified number (K) of DayStructs as centroids. Centroids are defined as the representative DayStruct for each cluster. In the Assignment step "C", all other DayStructs are assigned to their most similar centroid. Thus the points are clustered into K different groups. During Reassignment, step "D", for each cluster, the centroids are recomputed to more accurately represent logical groups. It should be noted that steps "C" and "D" have a repetitive aspect to achieve optimal clustering (which is described in more detail in (MacQueen 1967)).

3.4 Divide and Conquer

Due to a potentially enormous size of the data sets (thousands of DayStructs of multiple images), resources have to be carefully allocated. LiQuID implements its own divide and conquer K-means clustering algorithm to avoid simultaneous management of all DayStructs. It starts by dividing the set of DayStruct into smaller subsets and clustering each subset using the K-means algorithm. Afterwards, the divide and conquer clustering algorithm is performed on the centroids of all the subsets' clusters, eventually resulting in the desired K number of clusters with each their respective centroids. This efficient use of resources allows LiQuID to manage large and complicated datasets.

3.5 Determining K

Using the K-means clustering algorithm, LiQuID runs into the problem of deciding upon the appropriate number of the K clusters. If LiQuID selects too small of a K, DayStructs that are fundamentally different may be combined into the same cluster and thus be classified the same, losing important and distinguishing lighting information. Conversely, too many clusters may defeat the purpose of clustering in the first place—namely to categorize similar data together. Currently, this problem is resolved in an ad-hoc way—with the user responsible for adding or subtracting clusters.

4 LiQuID Architecture

LiQuID is a program responsible for coordinating parameter selection with lighting quality analysis (Figure 5). By interacting with a modeler, it sets a broad range of visual and environmental parameters for daylight simulation. After the simulations are run, it serves as a processor for quantitative and visual analysis. Hence, LiQuID is positioned to capitalize on the extensive work on modeling, simulation, and analysis for lighting design.

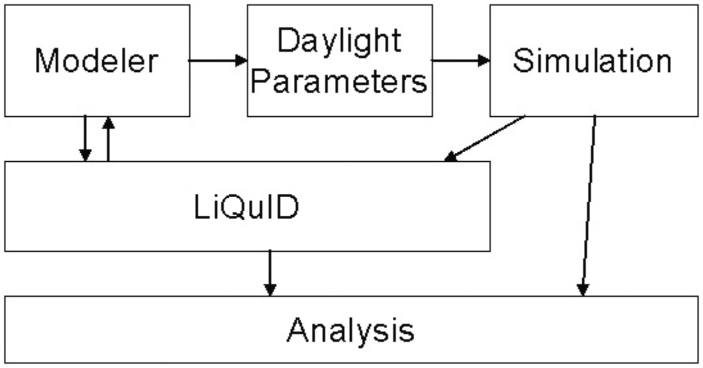

Figure 5 LiQuiD is positioned as a middleware application to coordinate the modeler with analysis components.

4.1 Modeling, Setting Parameters, and Simulation with LightSketch

LiQuID works in conjunction with the existing research program LightSketch (Glaser, Voung et al. 2003). In terms of Figure 5, Lightsketch performs the role of both modeler, parameter setter, and simulation driver. As an architectural modeler, LightSketch allows a user to quickly sketch a layout of a building and arrange the placement of windows, skylights, and various forms of luminaires. Unlike other more complex CAD programs, LightSketch only requires the understanding of a small set of architectural symbols. While the user draws and connects strokes upon the canvas, LightSketch automatically straightens out lines, which are interpreted as walls. Sketching a short line over an existing wall creates a window, while drawing small circles and boxes creates various types of luminaires and skylights in those locations. Simple to learn and use, LightSketch is intended as application for designing a building without requiring much time or technical expertise with specialized architectural software.

Aside from making 3D lighting model, LightSketch is capable of setting daylight parameters and running simulations. Multiple variables including time, date, weather, sky condition, and location can be set inside of LightSketch. The program can then drive Radiance simulations and output both luminance and illuminance data. Figure 6 illustrates a model drawn in LightSketch and illuminance data produced from it. The model is square with an east-facing window with two skylights. The illuminance data is generated though varying two time axes during clear sky conditions.

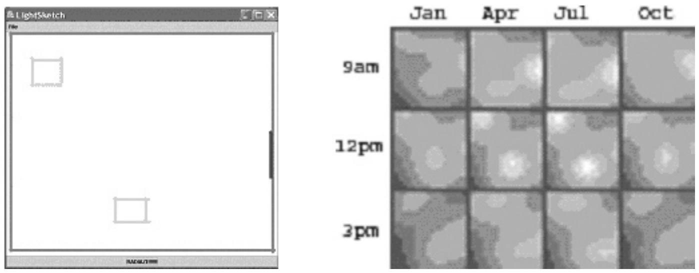

Figure 6. LightSketch showing a square room with an east window, and two skylights (left). Performance results (right), at workplane height for 4 times in the year at 3 times of day.

4.2 LiQuID

LiQuID uses a simple set of panels to aid with daylight parameter selection and analysis preferences (Figure 7). LiQuID uses this information to drive the simulation across the user specified ranges and for creating cluster sizes.

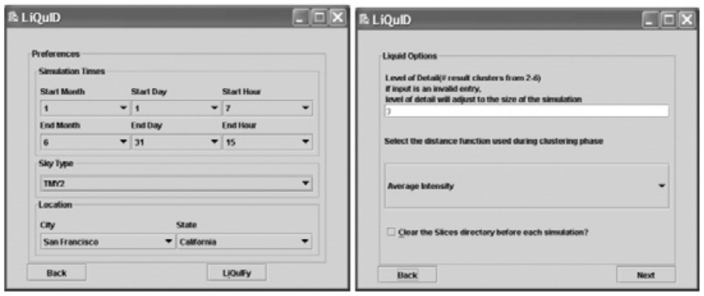

Figure 7. Portions of the LiQuID driver panel. (left) For selection of temporal, sky, and location parameters, and (right) features of the clustering utility

4.3 LiQuID Cluster Viewer and Inspector Analysis Modules

Figure 7 is a screenshot of the liquid driver panel. By modifying the level of detail requested, one can adjust the number of clusters to create. As discussed in section 3.5, this is currently done with a user input box to select the number of clusters. This flexibility is important since the data will vary in complexity (e.g. just analyzing a simple model under clear sky conditions will produce markedly different results than a complex one with measured sky conditions) and will need an appropriate number of categories to describe them. Also, the user is allowed to use two types of similarity metrics—by intensity, or a simplified glare index. Using a DG approach, the "perceptually similar" metric has been realized as comparing intensities across an image and as having similar glare effects.

The Cluster Viewer displays the results of LiQuID's clustering process. The viewpoints for each centroid (representative) is displayed along with each cluster's size (Figure 8). In this case, the 12 data points sampled were divided into three clusters of size 3, 2, and 7 daystructs respectively. The frequency of daystructs for each cluster is indicative of how often it occurs in the interval under analysis. In this case, Cluster 2 is the most frequent since 7 of 12 daystructs are most similar to its prototype. It tells us, for example, that the majority of time, the northwest skylight provides ample light across the walls it is adjacent to. Another module, the LiQuID Inpector, allows the user to see where each daystruct originates from.

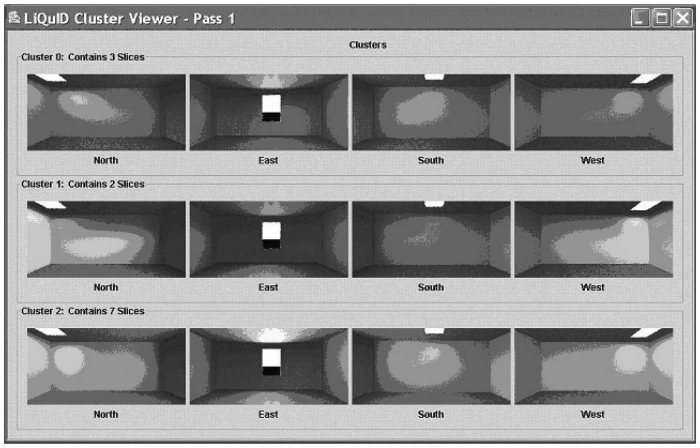

Figure 8. LiQuID's Cluster Viewer. It shows clusters (from top to bottom) and individual viewpoints for each cluster (left to right).

5 Use Scenario

Imagine that an architect, Jane, wishes to complete the fenestration scheme for an office with two exterior walls facing south and east respectively. Initially, Jane uses the program LightSketch to maximize the glazing on both exterior walls to provide for as much natural light as possible. She starts LiQuID and asks to see three clusters for lighting between 9am to 3pm. The images make clear that during this interval, under a range of sky conditions, there is far too much light. The numerical data confirms that she exceeded the recommended window to wall ratio which leads to unacceptable luminance distributions. Hence, she realizes that completely opening up the two walls would lead to a visually uncomfortable environment for most of the working day.

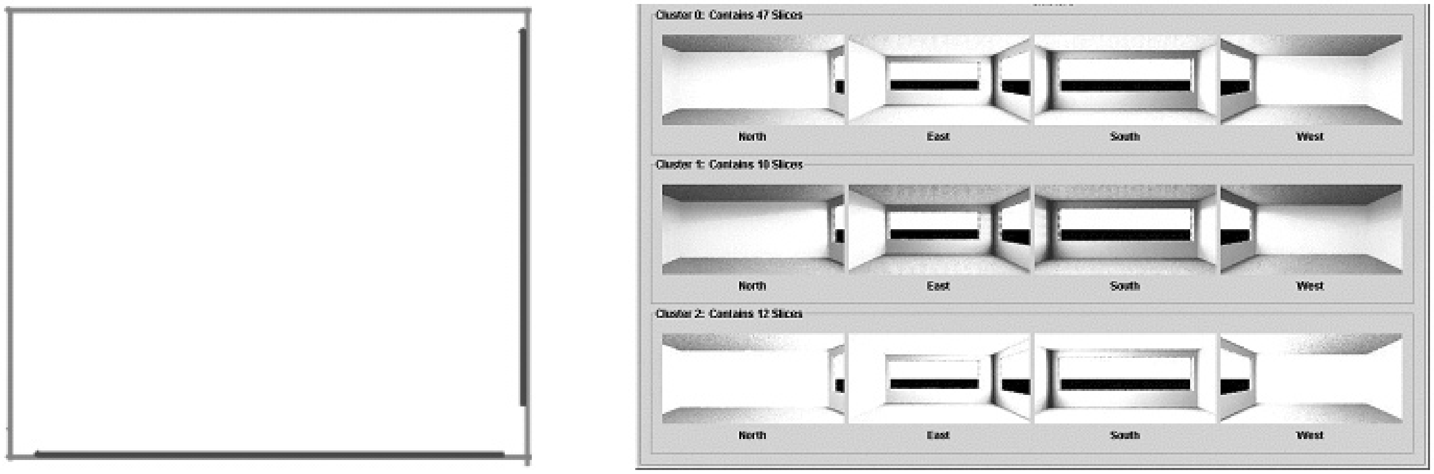

Figure 9. A sketch showing two very large windows (left) and three typical lighting conditions (right). Clusters “0” and “2” have excessive levels of glare, while the infrequent occurrence “1” is manageable.

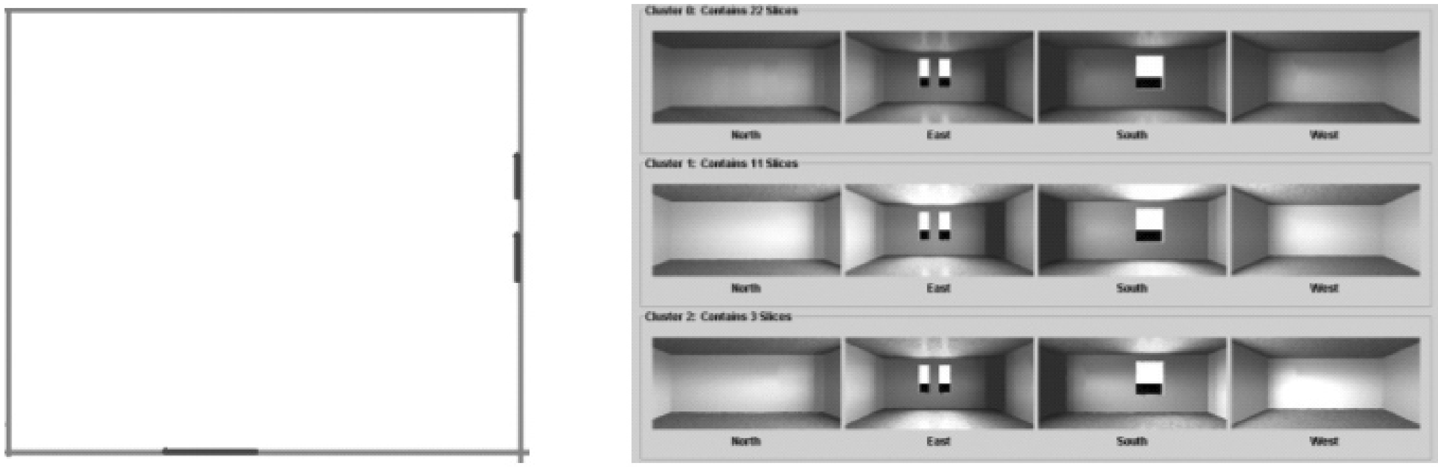

Figure 10. Using less glazing leads to a more acceptable lighting scheme. Frequently, the room appears a little dark (cluster “0”), but other times with patches of light are brighter.

Jane makes another sketch, this time using three smaller windows—two on the south side, while one on the east. She did not want to continue with her first design since she thought it would not be worth planning a shading control for the already costly large windows. She runs LiQuID on her new sketch to inspect a larger range of times (6am to 6pm) since she is more confident of her approach and is willing to wait for the additional calculations (which amount to less than a minute). Figure 10 shows the more acceptable results in LiQuID’s cluster viewer. It is important to note that the relatively low light levels in Cluster 0 occur near twilight, while the high levels in Cluster 2 appear only early mornings with clear skies. The other times have generally acceptable visual quality.

Although her design does not seem to have any significant problems, Jane notices dark spots around the southeastern, and to a lesser extent, northwestern corners. Although they could be useful places for furniture, she decides to explore the possibilities of using skylight to brighten these areas. She decides to place one on its northwest and another on the southeast corners. LiQuID confirms that the apertures she draws removes the darker areas while providing more visual interest to the model without allowing for too much light (Figure 11).

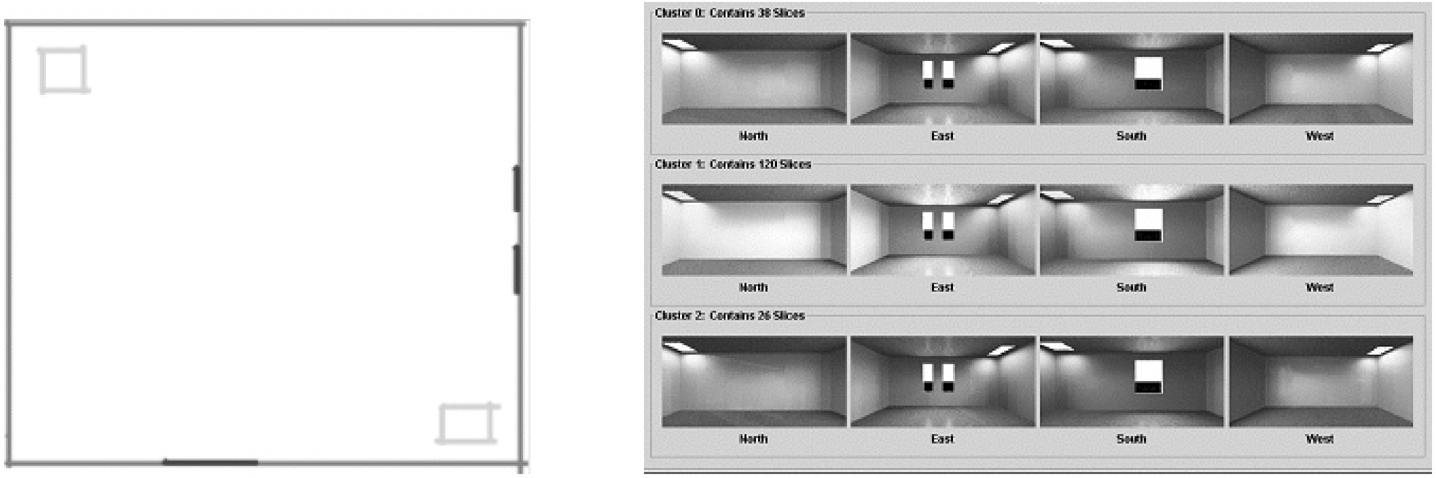

Figure 11. Adding skylights both brightens dimly lit corners and provides for more regions of visual interest.

Although it is arguable that this is a fully worked out and tested design (e.g. there is no electric lighting scheme for night occupancy, the skylights do not have diffusers, the dimensioning is still not refined, etc.), it shows how an architect could use LiQuID and LightSketch to quickly create a building design and judge its lighting quality. General trends in the data were reflected in the clusters, which were computed automatically and without a complicated interface.

6 Conclusions and Future Work

By analyzing numerous lighting simulations across multiple variables, LiQuID extracts useful lighting information from an architectural model. LiQuID aims to relieve architects and lighting designers from the tedious work required to analyze countless potential lighting situations. Also, it aims to remove some of the oversimplifications and generalizations about daylight that are utilized when evaluating lighting systems. Therefore by doing careful analysis of a comprehensive amount of simulations, LiQuID aims to provide reliable, accurate and legible feedback to building designers to encourage them to incorporate visual and environmental concerns of daylighting into their designs.

The integration of architectural modeling, lighting simulation, computer-aided classification, and lighting quality analysis is inherently a complicated issue. We are planning on improving the modeling capabilities to include more sophisticated sky descriptions and more variables for energy efficiency. More sophisticated clustering algorithms will also be employed to both improve the automation and also develop the classification system. In its current implementation, LiQuID uses a data driven process for grouping individual images for lighting simulation. Future versions will also consider classifying ranges of data to account for similarities among larger temporal units such as mornings, entire days, and weeks. The classification system can also be augmented by directly incorporating normative lighting characteristics. Lastly, links to easier to use output programs are being developed.

References

Aldenderfer, M. S. and R. K. Blashfield (1984). Cluster Analysis, Sage Publications.

Benya, J., L. Heschong, T. McGowan, N. Miller and F. Rubinstein, Eds. (2001). Advanced Lighting Guidelines. White Salmon, WA, New Buildings Institute.

Cheng, N. and E. Pat-Yak Lee (2001). Depicting Daylighting: Types of Multiple Image Display. Proceedings of the Association for Computer Aided Design in Architecture (ACADIA ‘01), Buffalo, New York.

Erwine, B. (1999). Daylight Models. Seattle, WA, Lighting Design Lab.

Glaser, D. and S. Ubbelohde (2002). “Techniques for Managing Planar Daylight Data.” Building and Environment 37(8-9): 825-831.

Glaser, D., J. Voung, Y. Xiao, B. Thai, S. Ubbelohde, J. Canny and E. Do (2003). LightSketch: A sketch-modelling program for lighting analysis. CAAD Futures 2003, Tainan, Taiwan, Kluwer.

He, Q. (1999). Review of Clustering Algorithms as Applied in IR. Urbana-Champaign, Graduate School of Library and Information Science, University of Illinois: 33.

Illuminating Engineering Society, M. S. Rea and Illuminating Engineering Society of North America (2000). The IESNA lighting handbook : reference & application. New York, N.Y., Illuminating Engineering Society of North America.

MacQueen, J. (1967). Some Methods for Classification and Analysis of Multivariate Observations. Fifth Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, CA, University of California Press.

Marks, J., B. Andalman, P. A. Beardsley, W. Freeman, S. Gibson, J. Hodgins and T. Kang (1997). Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation. SIGGRAPH 97, Los Angeles, CA.

Moeck, M. and S. Selkowitz (1995). A Computer-Based Daylight Systems Design Tool. ACADIA, Seattle, WA.

Papamichael, K., J. Lai, D. Fuller and T. Tarik (2002). A Web-based Virtual Lighting Simulator. Proceedings of the Association for Computer Aided Design in Architecture (ACADIA ‘02), Pomona, CA.

Veitch, J. (2001). “Psychological processes influencing lighting quality.” Journal of the Illuminating Engineering Society 30(1): 124-140.

Ward, G. (1994). “The radiance lighting simulation system.” Computer Graphics 28(7): 459-72.

(Endnotes)

[1] See, for example, (Illuminating Engineering Society, Rea et al. 2000; Benya, Heschong et al. 2001; Veitch 2001) for more in depth discussion of these issues.

[2] Using images for lighting quality assessment has also been advocated by (Eissa, Mahdavi et al. 2001) for electric sources.

[3] The approach of using both photographs and quantitative results has also been advocated for teaching daylight (Hanna 1996).