On Classifying Daylight for Design

Daniel Glaser and James Peng

On Classifying Daylight for Design

Daniel Glaser and James Peng

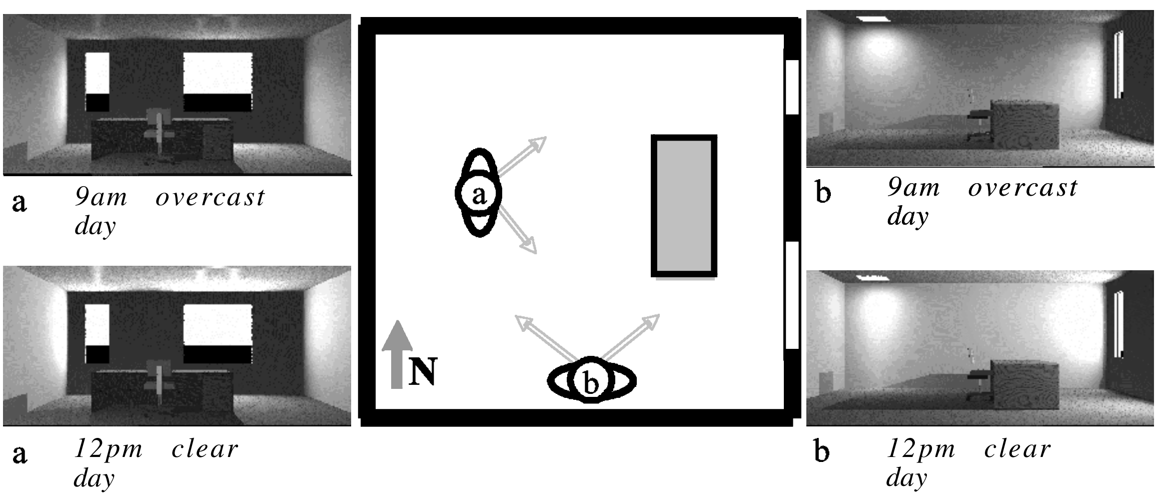
In this paper, we present LiQuID, a tool for clustering lighting simulation data. Photographs are useful vehicles for both describing and making assessments of architectural lighting systems. A significant barrier to using photographs during the design process relates to the sheer volume of renderings 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.

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[A]. *Assessment* and *description* are not mutually exclusive activities. Lighting desigenrs calculate daylight factors during the design process to "see light" across multiple sky positions during the design process [4]. Moeck and Selkowitz [5] argue the necessity 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[B]. Hence, lighting design and analysis is both about imagery and numbers[C].

Current CAAD tools for lighting design support description and assessment in a variety of ways. The most basic way of describing a model is through generative design programs such as [5], [6], and [7]. These programs allow for the user to easily create models and generate analysis upon demand. A limitation of this approach 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 ad-hoc generation of performance graphs or images may not guarantee a comprehensive view of the data. Daylight varies by viewpoint, time of day, day in year, and sky condition (Figure 1). Hence, in most generative design programs, the user would likely have to click a mouse button thousands of times to examine a

▼ Figure 1. Lighting quality varies by location, view, time, and sky condition. Hence, there is a large parameter space for generating images of a model.



[A]See, for example, [1-3] for more in depth discussion of these issues.

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

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

years worth of data for just at a single point in the room! This problem is partially overcome with more condensed or sophisticated visual representations (e.g. [8, 9]) but they do not necessarily make salient trends in the data. Furthermore, visualizing large datasets also has computationally demanding pre-processing requirements.

In addition to generative design programs for lighting, there are also tools that focus primarily on its analysis. In these cases, a parameter space is created for a comprehensive assessment. This guarantees that the data will be thoroughly examined numerically, but at the cost of potentially not asking the right questions. For example, in [10] an optimization approach places luminaires to achieve minimal energy consumption. Although the performance variable for energy is optimized, interactions with the user are limited to specifying abstract quantitative metrics[D]. This is true for other programs for energy analysis [11] and electric lighting design [12].

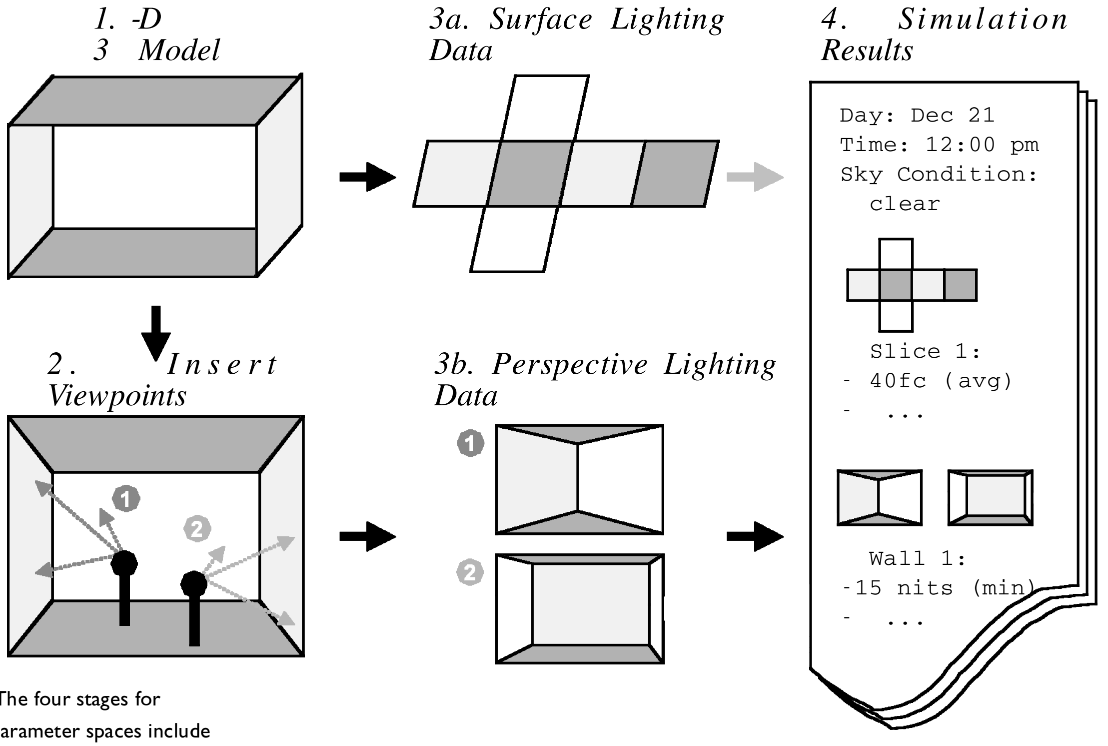
Hybrid approaches attempt to provide the benefits of both description and analysis. In GESTALT [13], the user can create a model and see numerical lighting performance (such as daylight factor). Inversely, the user can specify a performance constraint and have the model regenerate itself according to a locking and priority schedule. This requires, though, the users to specify formulaic performance criteria and design constraints. A "Design Gallery" [14] approach advocates showing users pictures of representative possible solutions a set of constraints can create. By nature of pruning data through similarity, it can also be used as an analysis tool, which is developed in this paper.

We propose clustering as a primary method for improving the design and analysis phase of daylight design. It is based upon the theory that similar data, whether numerical or visual, can be grouped together to reduce the problem size [15] [14]. Clustering does not hide potentially important details about lighting quality that can be lost with reductive performance analysis methods. Instead, it allows an architect to see the major trends in their data. It is accomplished through generative representative images for daylight models varying by solar time and cloud cover.

2. Generating Parameter Spaces in LiQuID

LiQuID develops a comprehensive temporal and sky 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.

[D]It should be noted that the proposed system did not account for daylight, an energy efficient light source, in its optimization calculations.



▲ Figure 2. The four stages for generating parameter spaces include geometric and viewpoint specification (stages 1 through 2) and simulation across a large parameter space (stages 3 and 4). Gray arrow denotes future work.

2.1. Analysis of Geometry Files

LiQuID relies on building geometry and simulation data from the program LightSketch [6]. LightSketch is a sketch-based interface for designing architectural models with both daylight and electric light features. After the user has completed the design of a space in LightSketch, the geometry of room is stored into RADIANCE [16] 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.

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 marking a place in the room that a person will likely be sitting at. Viewpoints can also automatically be generated by using the heuristic of looking at multiple directions from 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 pixels for images, and 20x20 arrays 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. Larger images can be created after the initial processing to see high-resolution results.

2.4. Simulation Results

The lighting quality data that LiQuID creates allows for comprehensive analysis in the next section.

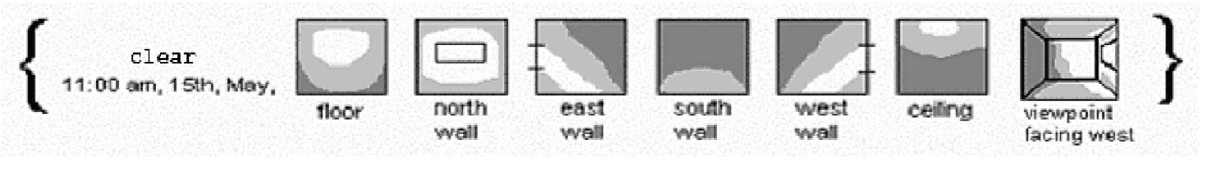
3. Classification in LiQuID

LiQuID's clustering process groups DayStructs according to their similarity in luminance values. To deal with a large amount of lighting simulations smaller representative units can be created while still retaining most of the relevant information of the original data. 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. It uses a data-driven approach, where images are reorganized into relatively homogeneous groups. The number of groups can vary with tradeoffs in number of sets and descriptive power.

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, under a clear sky condition. 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).

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

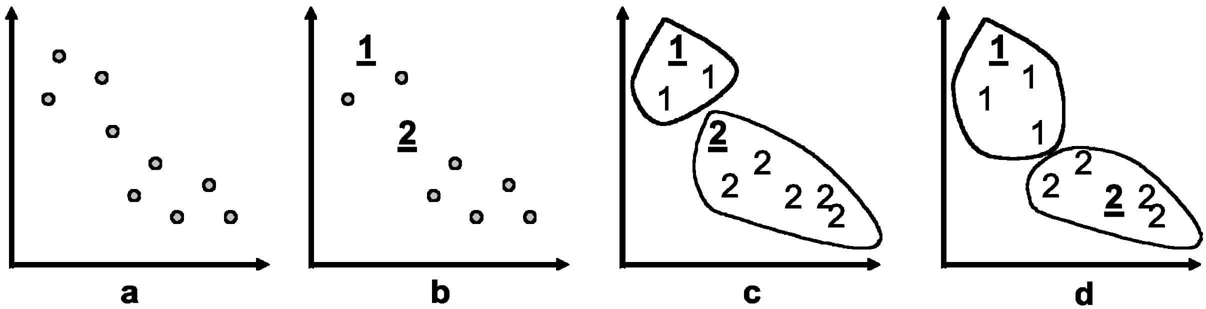


3.2. K-means Clustering Algorithm

LiQuID uses a k-means [15] clustering 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 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 one. 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 [17]. 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"). The pictures, stored as DayStructs, are represented by x-y coordinates on a 2D plane (step "a"). 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 [18]).



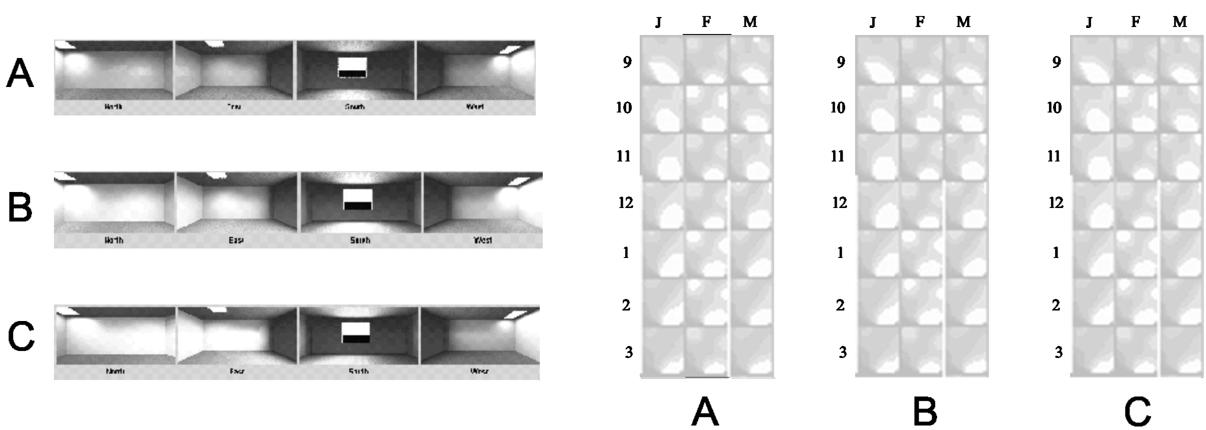
▲ Figure 4. k-means algorithm showing a) initial data b) selection c) assignment, and d) reassignment

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.

Figure 5 illustrates how LiQuID classifies the daylight for a model with a single southern window and two skylights (NE and NW corners) with a TMY-2 reference set. For the first three months (January through March), three of six distinct lighting patterns are shown in a cluster viewer (Figure 5 left). Cluster "A" shows a condition occurring in February where the northeast skylight has little impact. The lighting conditions illustrated by both clusters "B" and "C" show how the northeastern skylight illuminates the northern wall. The differences between clusters "B" and

"C" are both related to quantity and distribution. "B", on average has more light than "C" and also shows where the northwest skylight outperforms the northeast. It should be noted that LiQuID only uses the illuminance perspective data for classification and also the somewhat irregular clustering pattern is due to the nature of TMY-2 data.

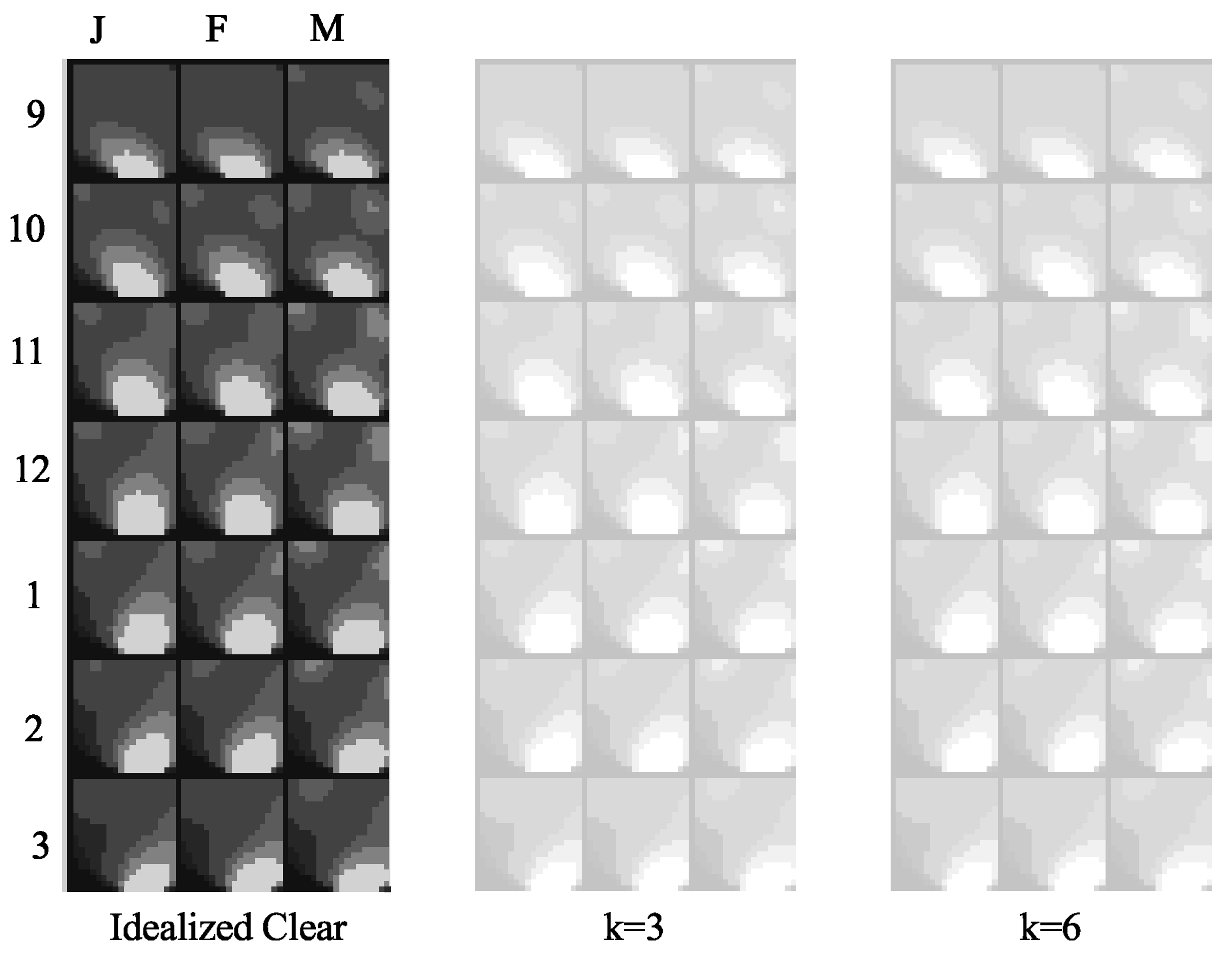
▼ Figure 5. LiQuID classifying measured daylight data into 6 categories. Three categories "A" "B" and "C" highlight different conditions that are possible with the current fenestration system.



3.3. Determining K

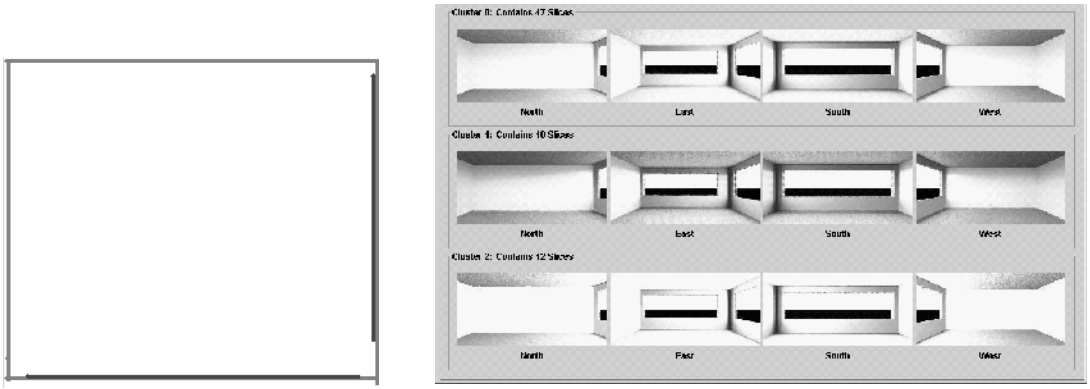
Using the K-means clustering algorithm, LiQuID runs into the problem of deciding upon the appropriate number of the K clusters. If LiQuID selects too small of a K, DayStructs that are fundamentally different may be combined into the same cluster and thus be classified the same, losing important and distinguishing lighting information. Conversely, too many clusters may defeat the purpose of clustering in the first place-namely to categorize similar data together. Figure 6 shows LiQuID using both 3 and 6 for K in the case of data generated from CIE clear skies. Since CIE clear generates relatively uniform data, increasing K from 3 to 6 has marginal benefits-notably that the additional cluster shown has only 1 daystruct and the other clusters are virtually unchanged. More complex data sets, though, require additional clusters to see meaningful patterns. For example, variable measured data as illustrated in Figure 5 in Section 3.2 show such a case where 6 clusters were effectively used.

▼ Figure 6. LiQuID clustering CIE Clear sky data. Note that there is little difference in the categorization scheme for k=3 or k=6 for this simple data set.



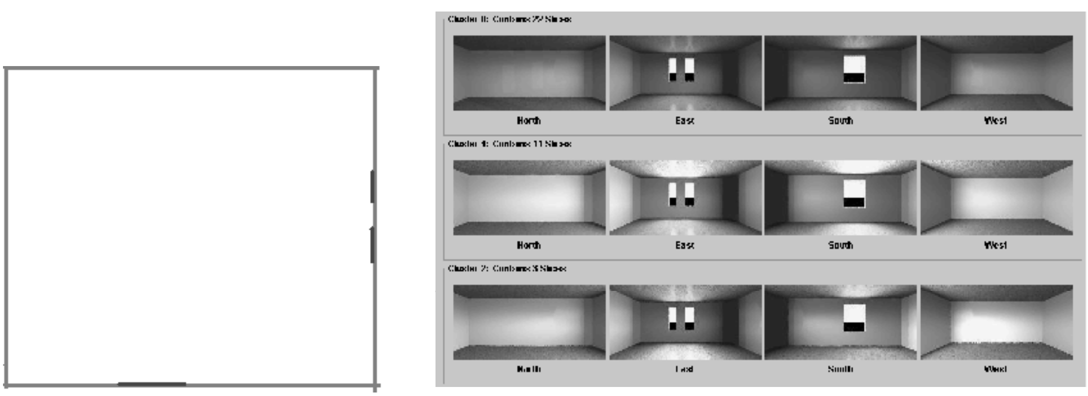
4. Use Scenario

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 (Figure 7, left). She starts LiQuID and asks to see three clusters between 9am to 3pm. LiQuID's cluster viewer illustrates three categories of light in rows organized by cluster number, frequency, and centroid images (Figure 7, right). The images make clear that during this interval, under a range of sky conditions, there is far too much light. Hence, she realizes that completely opening up the two walls would lead to a visually uncomfortable environment for most of the working day.



▼ Figure 7. A sketch drawn in LightSketch 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.

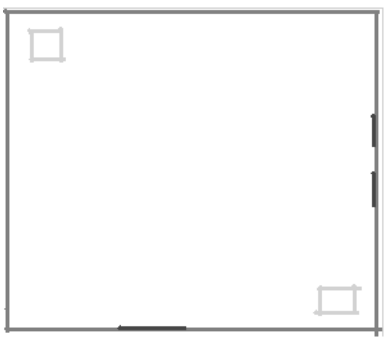
Jane makes another sketch, this time using three smaller windows-two on the south side, while one on the east. She did not want to continue with her first design since she thought it would not be worth planning a shading control for the already costly large windows. She runs LiQuID on her new sketch 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).

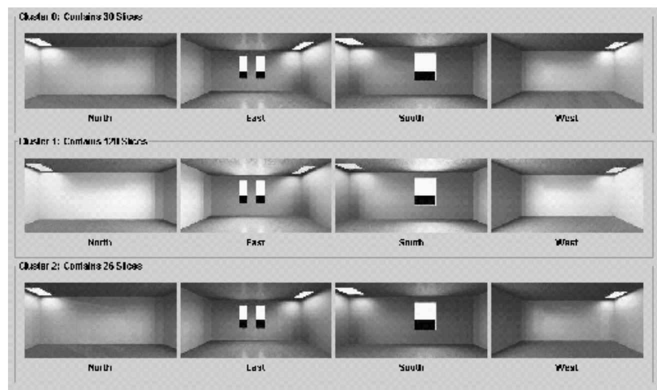


▼ Figure 8. Using less glazing leads to a more acceptable lighting scheme. Frequently, the room appears a little dark (cluster "0"), but other times with patches of light are brighter.

Figure 8 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 9).





▲ Figure 9. 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.

5. 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.

We are planning on improving a number of areas in LiQuID. The first is with expanding the sky models improving the sampling intervals and durations. Illuminance data could also be considered in the classification scheme. Clustering algorithms will also be refined to both improve the automation and also to address normative lighting characteristics. Visualization of clusters is also a future area of development. Lastly, similarities among larger temporal units are also under consideration.

Acknowledgements

We gratefully acknowledge support for this research from a California Energy Studies grant awarded by the University of California Energy Institute.

References

1. Benya, J., L. Heschong, T. McGowan, N. Miller, and F. Rubinstein, eds. *Advanced Lighting Guidelines.* ed. J. Roberts. New Buildings Institute, White Salmon, WA , 2003.
2. Illuminating Engineering Society, M.S. Rea, and Illuminating Engineering Society of North America, *The IESNA lighting handbook : reference & application.* 9th ed., Illuminating Engineering Society of North America. New York, N.Y., 1 v., 2000 , (various paging).
3. Veitch, J., Psychological processes influencing lighting quality, *Journal of the Illuminating Engineering Society*, 2001. Vol. 30, No. 1, pp. 124-140.
4. Erwine, B., Daylight Models, Lighting Design Lab: Seattle, WA., 1999.
5. Moeck, M. and S. Selkowitz. A Computer-Based Daylight Systems Design Tool, in: *Proceedings of the Association for Computer Aided Design in Architecture (ACADIA'95)*, 1995, Seattle, WA., pp. 261-279.
6. Glaser, D., J. Voung, Y. Xiao, B. Thai, S. Ubbelohde, J. Canny, and E. Do. LightSketch: A sketch-modelling program for lighting analysis, in: *CAAD Futures 2003*. 2003, Kluwer, Tainan, Taiwan, pp. 371-382.
7. Papamichael, K., J. Lai, D. Fuller, and T. Tarik. A Web-based Virtual Lighting Simulator, in: *Proceedings of the Association for Computer Aided Design in Architecture (ACADIA '02)*, 2002, Pomona, CA., pp. 269-277
8. Glaser, D. and S. Ubbelohde, Techniques for Managing Planar Daylight Data, in: *Building and Environment*, 2002, Vol. 37, No. 8-9, pp. 825-831.
9. Cheng, N. and E. Pat-Yak Lee. Depicting Daylighting: Types of Multiple Image Display, in: *Proceedings of the Association for Computer Aided Design in Architecture (ACADIA '01)*, 2001, Buffalo, New York, pp. 282-291.
10. Costa, A.C., A.A. Sousa, and F.N. Ferreira. Optimisation and lighting design, in: *WSCG'99 The 7-th International Conference in Central Europe on Computer Graphics, Visualization and Interactive Digital Media'97*, 1999. Univ. of West Bohemia Press, Plzen, Czech Replublic, 29-36.
11. Buhl, R.B., S.D. Curtis, J.J. Gates, M. Hirsch, S.P. Lokmanhekim, A.H. Jaeger, F.C. Rosenfeld, B.D. Winkelmann, M.A. Hunn, H.D. Roschke, and G.S. Ross. DOE-2: a New State-of-the-art Computer Program for Energy Utilization Analysis of Buildings, in: *Second International CIB Symposium on Energy Conservation in the Built Environment*, Copenhagen, Denmark, 1979.
12. Jung, T., M. Gross, and E. Do. Light Pen, Sketching Light in 3D, in: *CAAD Futures 2003*, 2003, Klewer, Tainan, Taiwan, 327-328.

13. Mahdavi, A. and L. Berberidou-Kallivoka, GESTALT: A Prototypical Realization of an Open Daylighting Simulation Environment, in: *Journal of the Illuminating Engineering Society*, 1994, Vol. 23, No. 2, pp. 62-71.
14. Marks, J., B. Andalman, P.A. Beardsley, W. Freeman, S. Gibson, J. Hodgins, and T. Kang. Design Galleries: A General Approach to Setting Parameters for Computer Graphics and Animation, in: *SIGGRAPH 97*, 1997, Los Angeles, CA., pp. 389-400.
15. Aldenderfer, M.S. and R.K. Blashfield, *Cluster Analysis*, Sage Publications, London, 1984.
16. Ward, G., The radiance lighting simulation system, in: *Computer Graphics*, 1994. Vol. 28, No. 7, pp. 459-72.
17. He, Q., *Review of Clustering Algorithms as Applied in IR*, Graduate School of Library and Information Science, University of Illinois: Urbana-Champaign, 1999, p. 33.
18. MacQueen, J. Some Methods for Classification and Analysis of Multivariate Observations, in: *Fifth Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, Berkeley, CA, 1967, pp. 281-297.

Daniel Glaser[1] and James Peng[2], [1]Interdisciplinary Doctoral Program, [2]Department of Electrical Engineering and Computer Sciences, University of California at Berkeley, Computer Science Division, 387 Soda Hall #1776, Berkeley, CA 94720-1776, USA

dcg@cs.berkeley.edu