

Integrating geometry and light: daylight solutions through performance-based algorithms

Michael D. Gibson
Ball State University, Muncie, Indiana

ABSTRACT:

Designing spaces for daylight is a complex problem for architects, balancing geometry with the location of daylight sources. Conventional design practice approaches this balance one-dimensionally: common procedures, rules of thumb, and building codes lead designers to default to regularity when designing windows and skylights.

The problem of daylight can be restated, starting first from the basic performance goal of distributed, uniform light. In traditional vernacular architecture, it is common to observe intentional coincidences among windows and interior surfaces, illustrating that openings and interior geometry can be integrated to distribute light in a way that is also experientially dynamic: integration also understood by great architects of the past and present.

Parametric design – a method of working where pieces of a simulated model can be manipulated ad infinitum – provides a new way of studying the relationship between light and geometry in the producing desirable, uniform, lighting conditions. Taking parametric design a step further, it is possible to tie together parametric models and computer-based simulations to produce an algorithm that ‘finds’ optimal configurations between openings and interior geometry. Such an algorithm reveals two possibilities. The first is that designers can systematically determine the best relationship among openings and interior space. Secondly, the success of these algorithms offers objective proof that, in comparison to the default of regularized patterns of openings, a more organic (i.e. less artificially ordered) relationship between openings and interior indeed is better for producing uniform daylight.

Two parametric algorithms will be discussed in the paper: an optimization algorithm, leading to a given problem to a single solution, and an evolutionary algorithm, using the random generation of individual solutions to reach better fitting results. The workings of the algorithms as well as the interpretation of the results in the context of design for daylight are discussed.

CONFERENCE THEME: Approaches: Digital and the Real World

KEYWORDS: Computing, Parametric, Optimization, Algorithm, Daylight,

I. INTRODUCTION

Designing spaces for natural light is a complex problem for architects, involving the balance of surface geometry with the size, proportion, and location of daylight sources to produce uniform, balanced light (Fig. 1). Yet the one-dimensional procedures, narrow-sighted rules of thumb, and building codes informing conventional practice lead architects to favor centered, repeating, symmetrical, and other uniform solutions for openings. Driven by intuition and a trial and error process, designers default to regularized solutions that overlook the problems and opportunities that can stem from a more precise understanding of interior daylight and its experiential dimension.

The hypothesis of the paper challenges conventional daylight design on two levels: summarizing some important problems with the conventions, while also posing a method for developing optimized solutions to daylight that can be applied without the conventions, using computer-based parametric modeling, simulation, and algorithms. The importance of deriving a solution from three-dimensional geometry is that the results are based upon the actual, three-dimensional relationships between the sky, the interior surfaces, and the plane of analysis. Geometry in this case is actually critical, in contrast with the flattened derivations of buildings and rooms as in traditional daylight calculation methods, or the one-dimensional glazing percentages often used during schematic design.



Fig. 1: In the lobby of the Canadian War Museum, designed by Moriヤマ + Teshima Architects, light from above is reflected by canted walls, resulting in complex interplay between direct and reflected light. (Source: Author)

2. CURRENT DAYLIGHT DESIGN PRACTICES

2.1. PREDICTING IDEAL DAYLIGHT CONDITIONS

Ideal daylight conditions may be defined in part by adequate illuminance, or light volume, for a given task. Secondly, ideal lighting conditions must also consider luminance, or the amount of light reflecting from surfaces; spaces with a large amount of luminance or a large degree of variation in luminance within the visual field can result in the undesirable condition of glare (Rea 2000).

Estimating the performance of daylighting schemes involves the estimation of the amount of light volume in the space contributed by the sky and other external sources. Using daylight factor analysis, a 'daylight factor' for a given position in an interior space can be computed using a uniformly luminous sky that is roughly equivalent to an overcast sky. The value for daylight factor in a given location is a percentage of interior illuminance relative to exterior illuminance, taking into account contributions from interior and exterior reflected light (also referred to as reflectance) as well as light contributed by the sky (also referred to as sky factor).

Methodologies used to calculate daylight factor across a given space have been in place throughout the 20th century. Prior to powerful computers, designers and engineers used primarily two methods to predict daylight performance: using scaled, two-dimensional drawings to draft geometrical relationship between the exterior sky and interior surfaces, and later, using scale models with small light meters. It is likely that in many cases, the tedium of these methods discouraged complex solutions in favor of solutions with more neutrality – i.e. windows and skylights centered in rooms. Model-based solutions were useful for addressing reflected light on interior surfaces, but the use of reflected light in geometry specific daylight strategies – using light shelves, reflective coatings, etc. – was not a widespread practice.

2.2. THE WINDOW PLACEMENT PROBLEM

The architectural theorist Bruno Zevi addresses the issue of window placement by asserting that light should come from "[a]nywhere, as long as it is not in the center of a wall, dividing the room into three sections, an illuminated one between two areas of darkness. Let us give each window new

meaning as a specific light carrier in function of the interior space (Zevi 1978).” Zevi clearly connects the placement of the window with its effect on three-dimensional space – something that could not be easily provided by methods of the past that ignore three-dimension light and surface interaction. A computer simulation can quickly demonstrate the unfortunate shadows that Zevi predicts from a room-centered window (Fig. 2) and the better results produced from an elongated window that washes adjacent ceiling and wall surfaces with light (Fig. 3).

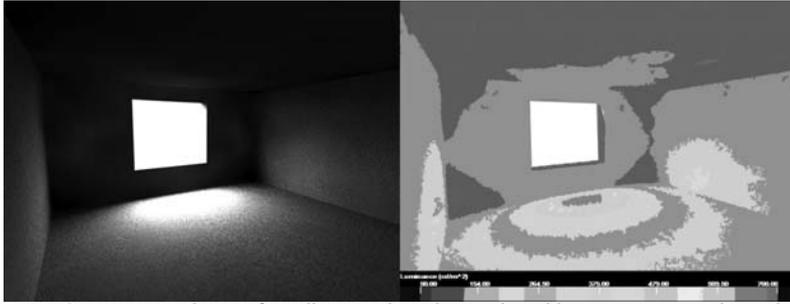


Fig 2: Computer simulation of a wall-centered window, with problematic contrast and poor light distribution; note luminance ranges in the field of view between approx.50 and 700...a serious contrast problem. (3DStudio Max radiosity rendering engine with IES diffuse skylight – typical for subsequent visible light renderings and luminance studies) (Source: Author)

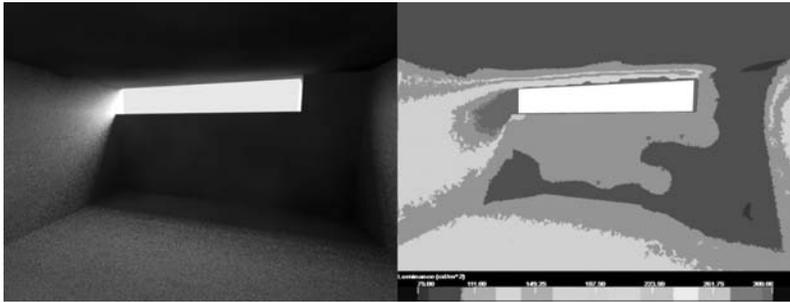


Fig 3: Better results are observed from a window whose light reflects on adjacent wall and ceiling surfaces. Discussed later: given the results of a particular simulation, how does one determine which variables to explore? Should position or proportion be changed first in this example? (Source: Author)

Computer simulation, as shown in Figures 2 and 3, provides performance feedback that is geometry-specific and as a result a powerful design tool. Yet recently and even newly built buildings show that architects still ‘yield’ to inane façade ordering systems and the convenience of symmetry (Fig. 4). Why do architects still produce these ‘yield solutions’, even if this placement compromises the distribution of light? A few historical anecdotes may explain. In the western tradition of architecture and specifically in urban buildings, the window was an important component of a façade and its position was dictated more by the exterior composition, the masonry construction methods prevailing in urban areas, and avoidance in interfering with electric illumination (of all things).

The arrival of the curtain wall and fully-glazed buildings in the twentieth century further distorted the notion of daylighting. Fully-glazed buildings pose two serious environmental control issues: the huge amount of heat loss and heat gain through the envelope, and a large volume of light at the perimeter of the building contrasted against a relatively dark interior. Decades of dealing with these two issues leave a set of codes and practices that have further reduced daylight design to one-dimensional solutions.



Fig 4: Common 'yield' window placement, with openings carelessly arranged with preference to exterior appearance. (Source: Author)

For example, current lighting standards (ASHRAE/IESNA Standard 90.1.2007 is the standard, and is also referenced by LEED) provides prescriptive limits on glazed areas for walls at 50% of exterior wall surface area, and for roofs at 5% of exterior roof surface area for the purposed of conserving energy. Regulating glazing in this manner makes sense for controlling thermal performance of the envelope, but says very little about the role of glazing as a daylight source. Such thinking suggests a building designed by a spreadsheet. The same lighting standard deals with lighting distribution, including both artificial and natural lighting, in terms of lighting power density. Lighting power density is expressed in bizarre units of Watts per square foot, a deference to the calculations for heating, ventilation, and cooling calculations that are impacted by the waste heat of artificial lighting. Without entering into a lengthy technical discussion, it may be summed up that the ASHRAE/IESNA standards dictate that artificial lighting is addressed first and daylight introduced afterwards for the sake of energy conservation. It may also be argued that buildings should be designed for artificial light because at night and on particularly gloomy days, artificial light is a necessity. (ASHRAE)

Yet these standards mentioned above, along with other prescribed strategies such as those related to ceiling height and floor proportioning, provide little inspiration for the design of day light features and their relationship to space. What is inevitable in this thinking is that windows and openings are reduced to uniform and static features, just like the grid-born artificial lighting that the windows supplement.

2.3. INTEGRATING GEOMETRY AND LIGHT

Well-known contemporary architects have demonstrated the importance of interior geometry to daylight. Concepts of daylight show up frequently in the work of Frank Lloyd Wright, and in his essays in *The Future of Architecture*, Wright the notion of daylight, light screens, and the relationship between light and surface are the underpinnings of three of the nine 'motives' (Wright 1953). Bruno Zevi wrote of this light-surface integration in Wright's prairie houses, where "every detail and moulding is conceived to receive, grasp, transform and transmit light (see Fig. 5). (Zevi 1991)

Corbusier, who clearly worked in a separate design and theory camp than Wright, also understood

this close association between light and geometry, writing in *Towards a New Architecture* “[t]o erect well-lit walls is to establish the architectural elements of the interior” (Le Corbusier 1927). In the work of many recent master architects, including Eero Saarinen, Alvar Aalto, Steven Holl, Louis Kahn, Renzo Piano, and many others too numerous to mention, this connection between daylight and interior surface is a defining feature – although other seductive qualities in their work may eclipse the thoughtful positioning of openings relative to surfaces (Fig. 1).

On the other hand, coordinated daylight and interior surface can be recognized even in traditional vernacular dwellings (as opposed to the façade-dominated urban buildings mentioned earlier), where interior features such as walls, window jambs, floors, and ceilings are assembled to reflect and distribute light (Fig. 6). From vernacular to modern design, thoughtful daylighting is more than just an environment adequate for tasks, but an environment beneficial to its inhabitants. Clearly the simple problem of positioning a window in a room calls for more sophisticated strategy of design that considers the interaction between light and surface.



Fig. 5: Frank Lloyd Wright’s prairie style interiors often exhibited windows at the corners of the rooms or positioned against walls, in order to use interior surfaces to reflect light. (Wescott House designed by Frank Lloyd Wright) (Source: Author)



Fig. 6: Vernacular dwellings typically positioned windows carefully to bring light to task areas and close to walls where it could be reflected (Shaker common residence, Canterbury, NH) (Source: Author)

3. SIMULATING DAYLIGHT: NEW COMPUTING METHODS

3.1. ADDRESSING PERFORMANCE WITH COMPUTER SIMULATION AND PARAMETRIC COMPUTING

Fine-tuning surface and light relationships in the past involved cumbersome methods, and has not been encouraged by standards and codes. Yet today's computer-based analysis gives designers the capability of creating and refining daylighting schemes that integrate surface and light.

As discussed early, computer simulations can produce analyses for daylight factor (illuminance as a percentage of exterior light) and luminance (measurements of reflected light and presence of glare); moreover, these models can simulate and subsequently test specific three-dimensional features in a daylighting scheme, including features like light shelves and clerestory windows. In doing so, we can develop solutions that, recalling the words of Zevi once again, "give each window new meaning as a specific light carrier in function of the interior space." (Zevi 1978)

To illustrate a method-based application of computer-based lighting analysis in design, a single simulation is not enough; instead we must run many solutions. Connecting performance with design implies that a designer seeks to optimize (i.e. make as high-performing as possible) a given design. To do this, designers must not only test their design, but apply a method for subsequently improving their design, in order to test it once again. Computer-based lighting analysis is particularly useful here, because testing a given iteration of a design is relatively painless, perhaps consuming a matter of minutes or even less to produce data. To apply a systematic approach, a designer may proceed using simple heuristic methods, perhaps using explicitly determined variations of a design to determine which is best in moving towards a desired criteria.

Computing's ability to provide a multitude of simulations in a short period of time brings up two issues. One is that manually-calculated lighting analysis was limited to a small number of iterations that can merely approach optimum; with a very large number of iterations, a truly optimum or very near optimum solution is reasonable. Secondly, the notion of a model that is constantly adjusted in response to performance outcomes is suggestive of an emerging area of computing called parametric modelling. In a parametric model, the model geometry is actually mutable and tied together by equations or object-based relationships, rather than fixed points in space. Parametric models can be easily changed without being rebuilt, and can in fact be driven by performance data so that the modification of the model is more or less automated.

3.2. ALGORITHMS AND EVOLUTIONARY COMPUTING

Consider the problem of a window in a room and assume that it does not belong in the center, but somewhere on a single wall. The room and the window areas are given in this example, but the window location and proportions (length and width) are flexible. Using trial-and-error methods and computer simulation, one can move through a quick succession of computer models yet problems emerge. When something goes well in the simulation, how is it determined that window location or proportion contributed to the success? Are the two variables in the problem interfering with one another somehow? Proportion and position are two variables that, in parallel, have an organic effect on daylight that is challenging to pin down with intuition and manual manipulation of the design (see Fig. 3).

Yet the data provides a link between the two variables if it is 'read' in a way that isolates patterns indicating the effect of window position and proportioning. While we can look at the graphic outputs from our analysis, we can also use statistics and data processing to look for these patterns and use them to inform changes made to the model. The idea that data can be used to drive design decisions is somewhat of a sacrilege in architecture, but has been common practice in engineering, even before modern computers.

Humans can certainly interpret data, but we can also connect the data to our parametric model to complete a sort of feedback loop where data directly triggers changes in the model. As a result of this directness, non-rational interference (aesthetics, whimsy, etc.) is suppressed and the path through several iterations should, theoretically, lead to an optimized solution, where the feedback

loop would actually slow down or stop as its criteria become satisfied. In the computing world, such a computing device is known as an 'evolutionary algorithm' (falling into the category of evolutionary computing) because its solutions are evolved towards optimization. The connotation of 'evolve' in this case is interesting; it suggests a sort of hands-off approach to problem solving, in which a solution is developed from the algorithm without intervention – at times producing unexpected results. The latter is termed 'emergence,' a phenomenon where the solution, unpredicted at the beginning, emerges as the algorithm unfolds.

4. DEVELOPING DAYLIGHT SOLUTIONS USING EVOLUTIONARY COMPUTING

4.1. APPLYING EVOLUTIONARY PROGRAMMING

Before further discussion of evolutionary computing and daylight, a method for evaluating and processing the simulation data must be recognized. Recall that uniform, glare-minimized (low contrast) daylight conditions are considered ideal. The IESNA Lighting Handbook notes that an ideally lit environment has uniform light distribution in the visual field (i.e. free of glare) and is covered by light sources of similar intensity (Rea 2000); from this definition, we may extrapolate that designs optimized for daylight factor should exhibit a consistency in daylight factor data throughout the analysis area.

Previous research by other parties has identified statistical standard deviation as a useful metric in assessing lighting consistency and thus minimization of glare (Demers 2007). By assessing the standard deviation of daylight values, the degree of variance across the analysis area is quantified and the degree of uniformity in light distribution is represented. Light distribution, in this case, is now a statistical value: a useful complement to the daylight factor contours typically produced by lighting simulations. Now each of the iterations within the modeling algorithm can be objectively compared using standard deviation, while daylight contours provide an additional visual aide.

Returning to the issue of evolutionary computing, the experiments carried out and presented in this paper fit into two classes of evolutionary problem solving: the first is an optimization algorithm, applied to a window and skylight coordination problem, and the second is an evolutionary algorithm, applied to a distribution problem involving multiple apertures on walls and the ceiling. Each approach is different, with different approaches tying together data and parametric models in a feedback-based algorithm. In the endnotes of this paper, a few technical points are discussed that address limitations of the algorithms.

4.2. OPTIMIZATION ALGORITHM: SKYLIGHT AND WINDOW

The optimization algorithm discussed in this research was developed to optimize the position and proportion of a skylight in a room containing a single window; the window's position was off-axis and in particular, resulted in an initial condition where one area of the room had a high concentration of daylight in contrast to the rest of the room. An optimization algorithm is a problem solving algorithm that iteratively cycles through a set of input and output operations that shifts in the model incrementally, rather than all at once, towards a solution (Papalambros 1988). The problem at hand involving the window and the skylight is particularly suited to this sort of algorithm because it has a single variable, and the final solution is difficult to solve intuitively. Fixing the position of the window also creates a light distribution pattern that can only be balanced with a single solution, rather than many possible solutions.

In short, the algorithm operates by handing off simulation data to a spreadsheet, which produces a series of outputs that are passed back into the parametric model to drive changes to the openings parameters for position and proportion. The resultant model is used to generate a new solution, and so on. This process continues until a terminal condition is noted; in the case of this experiment, a terminal condition would occur when light uniformity (standard deviation) is fully optimized (no further improvements conceivable). Typical of an optimization problem, the parametric model, the programming it contains, and the criteria used to drive the model all must be carefully defined in order for the algorithm to work properly (Eiben 2003).

As the algorithm reaches a terminal point, optimization is manifest not as a 'perfect' solution but a cycling in the general area of a solution (Papalambros 1988), sometimes even oscillating through a series of repeating solutions, all of which are close to optimized. An indicator that an optimization has been successful is that it demonstrates 'convergence,' where two independent optimization runs from two arbitrary starting points result in reasonably similar results (Papalambros 1988). Diagrams overlaying the path of solutions towards optimization are shown in Fig. 7; in these diagrams, two different starting points are shown to converge at a similar solution. It may be noted that while the statistical outputs in the algorithm became essentially the same between these two solutions, a slight difference in position between the two solutions resulted. This difference can be attributed to the relatively low number of data points in the analysis grid that, when processed, satisfy the optimization criteria in the same way.

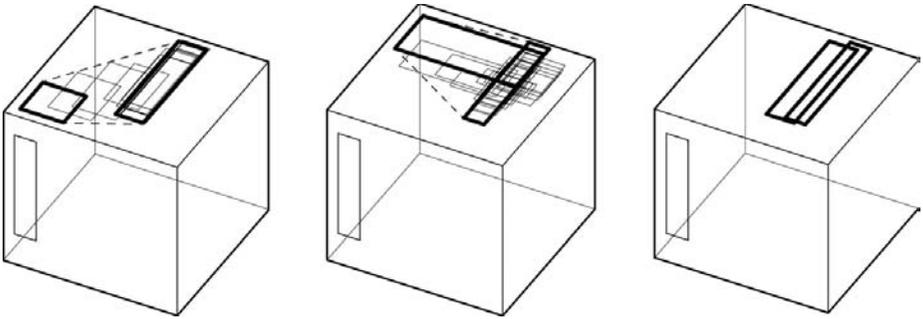


Fig. 7: Diagrams of the optimization algorithm, overlaying solutions that lead to an optimization; note that these two examples originated from different starting points yet converged at a similar solution, demonstrating convergence. The diagram on the far right overlays the two final solutions to show their similarity. (Source: Author)

Development of the optimization algorithm involved the application of three different pieces of software.

The parametric model itself was created using Rhino CADD software equipped with the plug-in Grasshopper, which enables the Rhino model to be 'parameterized' and also allows for a script (discussed later in this section) to be integrated into the model.

A program called Ecotect, a product of Autodesk, was used to calculate daylight factor for each model; the data generated in Ecotect is typically used only within the software to create a graphic display of the values as contours, graphs, etc. and may also be used to extract simple averages and extremes from the data.

The second stage of the algorithm involves processing the data from daylight analysis into outputs useful once again in the parametric model. This processing was carried out using a Microsoft Excel spreadsheet, which carries out a number of calculations. The first is to calculate the standard deviation for the data set to provide an objective assessment of the entire set of data points. The second set of calculations is more complex, and involved developing outputs that are useful in driving the parametric computer model.

In order to drive *positioning* of the skylight, the data was divided into quadrants, and each quadrant was averaged to produce a variable (Fig. 8). A higher average value in a particular quadrant could, as a result, drive the skylight to move away from that quadrant a particular amount proportional to the discrepancy in average value.

A second division of the data was made in order to drive *proportioning* of the skylight. In each direction, the middle third of the data was averaged and compared to the total average of the data; if this middle third returned a higher value than the adjacent thirds, the proportion of the skylight opening was adjusted in the opposing direction, making it 'stubbier' in that orientation. Overall,

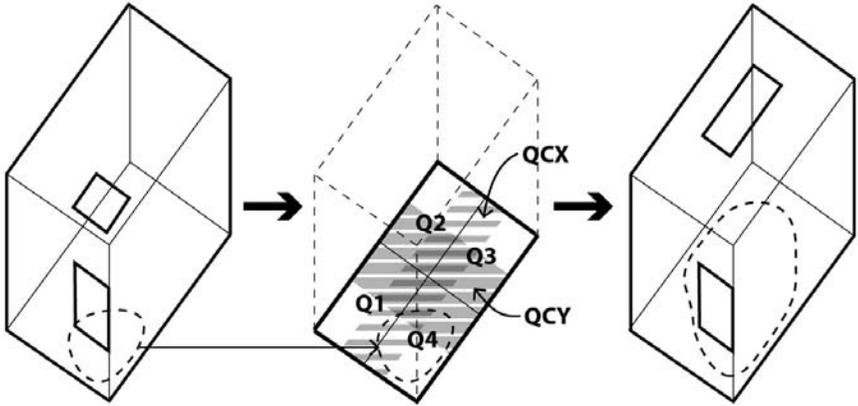


Fig. 8: Diagram of the initial optimization model, the quadrants used to evaluate the data, and the final outcome. (Source: Author)

the processing of data in the spreadsheet yields seven different averages that are returned to the parametric computer model.

The seven average outputs from the spreadsheet, upon return to the parametric model, are processed in a piece of Visual Basic computing code. The code contains a series of if-then conditional statements that function as decision-makers that are then returned into the flow of the parametric model (Fig. 9).

In summary, with every pass of the model, the model is sent to Ecotect, the resulting data moves to the spreadsheet, the spreadsheet outputs go back into the model, and a new model iteration is generated. Following this process over several iterations yields an optimized position for the skylight relative to the window (see Figs. 10-12). Additionally, the algorithm was tested using various room dimensions and proportions, some especially difficult to guess where exactly the ideal skylight position would occur. In each case, the algorithm proved successful in reaching an optimized solution (Fig. 13).

```

dblAVEQ234 = (dblQ2 + dblQ3 + dblQ4) / 3
dblAVEQ134 = (dblQ1 + dblQ3 + dblQ4) / 3
dblAVEQ124 = (dblQ1 + dblQ2 + dblQ4) / 3
dblAVEQ123 = (dblQ1 + dblQ2 + dblQ3) / 3
dblAVEQ1234 = (dblQ1 + dblQ2 + dblQ3 + dblQ4) / 4
If dblQ1 > dblAVEQ234 Then dblMVDISTQ4 = ((dblQ1 / dblAVEQ234)) * dblMVFCTR
If dblQ2 > dblAVEQ134 Then dblMVDISTQ3 = ((dblQ2 / dblAVEQ134)) * dblMVFCTR
If dblQ3 > dblAVEQ124 Then dblMVDISTQ2 = ((dblQ3 / dblAVEQ124)) * dblMVFCTR
If dblQ4 > dblAVEQ123 Then dblMVDISTQ1 = ((dblQ4 / dblAVEQ123)) * dblMVFCTR
If dblQCY > (dblAVEQ1234 * dblGRWTHRSHD) Then dblGRWY = dblGRWSCL
If dblQCY < (dblAVEQ1234 * dblGRWTHRSHD) Then dblGRWY = ( 1 / dblGRWSCL)
If dblQCY > (dblAVEQ1234 * dblGRWTHRSHD) Then dblGRWX = dblGRWSCL
If dblQCY < (dblAVEQ1234 * dblGRWTHRSHD) Then dblGRWX = ( 1 / dblGRWSCL)
If dblAVEQ1234 > dblAVEDFMAX Then dblGRWY = ( 1 / dblGRWSCL)
If dblAVEQ1234 > dblAVEDFMAX Then dblGRWX = ( 1 / dblGRWSCL)
If dblAVEQ1234 < dblAVEDFMAX And dblQCY > dblQCY Then dblGRWY = (dblGRWSCL * (dblQCY / dblQCY))
If dblAVEQ1234 < dblAVEDFMAX And dblQCY > dblQCY Then dblGRWX = 1
If dblAVEQ1234 < dblAVEDFMAX And dblQCY > dblQCY Then dblGRWX = (dblGRWSCL * (dblQCY / dblQCY))
If dblAVEQ1234 < dblAVEDFMAX And dblQCY > dblQCY Then dblGRWY = 1

```

Fig. 9: VBasic computer code that was embedded in the parametric model and used to connect analysis outcomes with model revisions. (Source: Author)

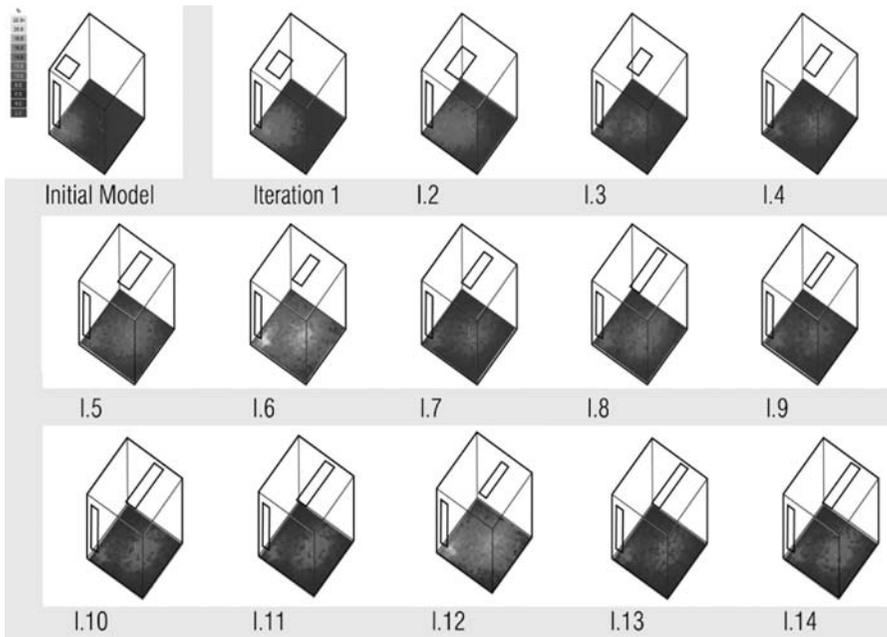


Fig. 10: Fourteen computer simulated iterations of an optimization, starting from the initial model, and showing the distribution of daylight factor values at floor level. (Source: Author)

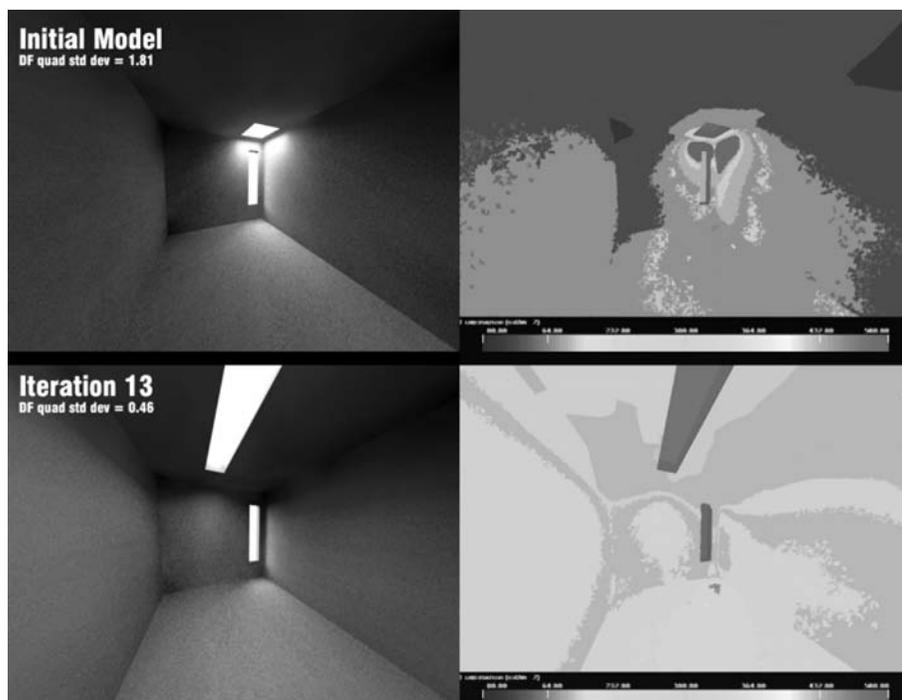


Fig. 11: Computer simulations comparing the initial model and iteration 13; note the increase uniformity of the light reaching the floor and the increase in light on the walls. (Source: Author)

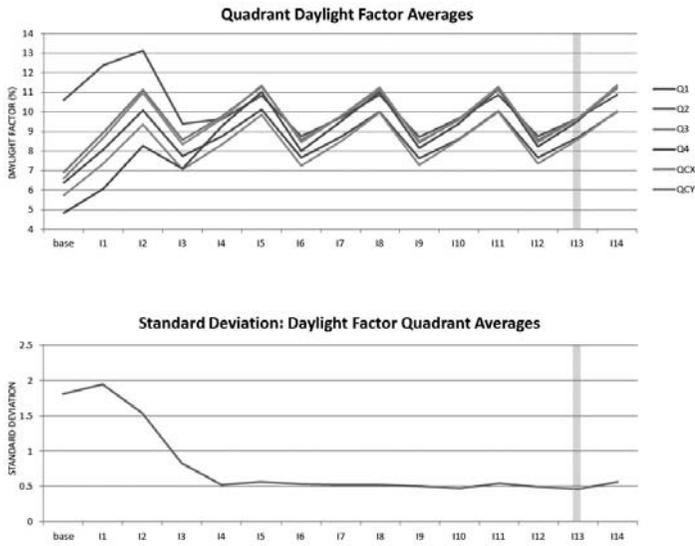


Fig. 12: Graphs showing the progress of the data during the optimization. Quickly daylight values become more uniform and cycle through high and lower average daylight factors as the parametric model reaches a prescribed limit. Iteration 13 shows the lowest standard deviation of the set, indicated it is the most uniformly lit solution of the 14. (Source: Author)

4.3. EVOLUTIONARY ALGORITHM: MULTIPLE OPENINGS

Optimization algorithms will always produce the same results given the same starting point, since outputs and performance will always lead to the same results. Evolutionary algorithms are very different, progressing in an unprogrammed manner using a process similar to that of evolution and natural selection. Based on a stochastic (in other words purely random) process these tools have the power to solve problems especially difficult for the human mind to solve.

In an evolutionary algorithm, the computing begins with an initialization of a population from randomly varying individuals who are then evaluated for fitness against a criterion (or multiple criteria), and then from these selected individuals a new population is initialized after the seeding individual's characteristics are mutated (Sivanandam 2008). After a few generations pass, surviving individuals will begin converging on characteristics that result in high fitness, or in other words, optimization around the chosen criteria for fitness.

In the research presented in this paper, an evolutionary algorithm was applied to the problem of providing uniform light from six openings (four windows and two skylights) arranged on two adjacent walls and the ceiling of a given room. The remaining two walls were left vacant of openings. Using multiple openings in this case presented a more appropriate problem for an evolutionary algorithm to solve: a single variable problem (i.e. involving one window) could be solved by human expertise or a simple linear optimization, but not a multiple variable problem (i.e. with multiple windows). An evolutionary algorithm is useful because it can assess multiple variables simultaneously in order to advance the algorithm (Sivanandam 2008), a strategy as humanly impossible as guessing a multiple digit password.

To create the evolutionary algorithm, once again a Rhino's Grasshopper plug-in was used to develop a pair of parametric models, each containing a bit of code that would randomize parameters for the openings using Visual Basic 'randomize' and 'rnd' functions in conjunction to generate a truly stochastic number within a given range.

Simply described, one model was used to initialize the first population of solutions; this model randomly scattered six openings on the assigned faces, and randomly proportioned the openings

while maintaining a defined opening area. A second model was used for generating subsequent generations from a seed solution. In this second model, the room and its original openings were the starting point, and the model randomly mutated the position and proportion of all six openings according to a given constraint.

Establishing fitness (in other words, setting criteria for the best solutions in the set) involved calculating overall standard deviation and overall average daylight factor value from the data set for each solution. The results were graphed in an X-Y graph to identify solutions whose data points, a combination of average daylight factor and standard deviation, appeared distinct from the other solutions. To generate a graph in which both standard deviation and average value numerically ascended in their axes, a mathematical inverse of standard deviation was used.

Tracking the effectiveness of the evolutionary algorithm is somewhat murky, since fitness and progress of the algorithm may be based on comparison with the original population, with the previous generation of individuals, or with the current generation of individuals. This issue is compounded when a generation produces two individuals that are highly fit for one criterion but are not successful in the other criterion. Clearly more work is needed in understanding the best application of fitness to these sorts of algorithms.

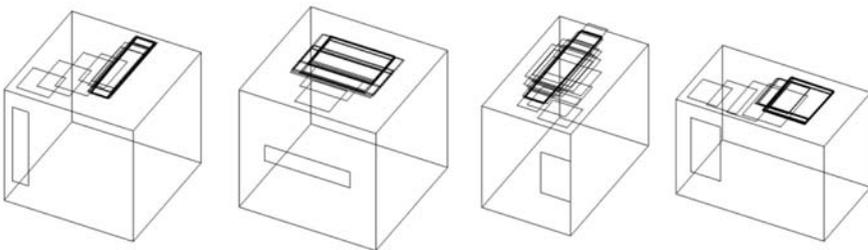


Fig. 13: Various room and window configurations tested by the optimization algorithm, and the resultant solutions. (Source: Author)

It is important to note that the evolutionary algorithm described here is not a full-blown genetic algorithm. Experts in the field of evolutionary computing are mixed on the subject of whether all evolutionary problem solving should do so, but some level of recombination or simulated reproduction (Eiben 2003): for example, combining traits from different individuals to form the next generation of solutions. More computing-intensive genetic algorithms will even use a stochastic process in the selection of individuals in order to eliminate external determinism from the process (Sivanandam 2008). True genetic algorithms, consequently, require a great deal of computing and remain beyond the scope of research presented in this paper, but intend to be explored in future research.

The results of the evolutionary algorithm experiment, though limited to only two generations due to constraints of time and programming capacity (discussed further in the endnotes), exhibit some success in increasing the performance of solutions both from generation to generation, and in comparison with the original yield solution (Figs. 14-17). On one level, the algorithm is successful in using a randomly based process to improve daylight performance in at least in one of the successive generations (the set of 13.x solutions) following the initiation set, providing evidence that the algorithm can work. Solutions in this set also performed better than the yield solution with its arbitrarily symmetrical, wall-centered openings as well. On the other hand, two of the three successive solutions that were run in parallel (21.x and 22.x sets) did not show much improvement from the seed solutions, although they still outperformed the yield case. This is a reminder that an evolutionary algorithm can certainly regress as it moves towards optimization; perhaps subsequent generations, either from the 13.x set or from the initiation set, would have performed better. It may also be possible that a better way exists to establish fitness for these solutions, something that may emerge in future research. In sum, the evolutionary algorithm verifies the hypothesis that an evolutionary

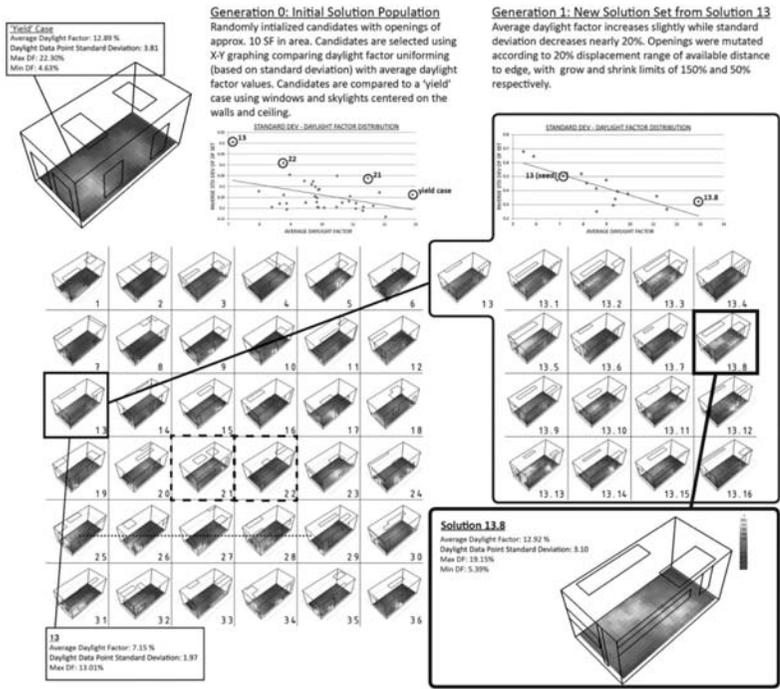


Fig. 14: The initialization population of the evolutionary algorithm, with one of the first branches shown as an example. Note the X-Y graphs synthesizing comparisons of standard deviation and average daylight factor. (Source: Author)

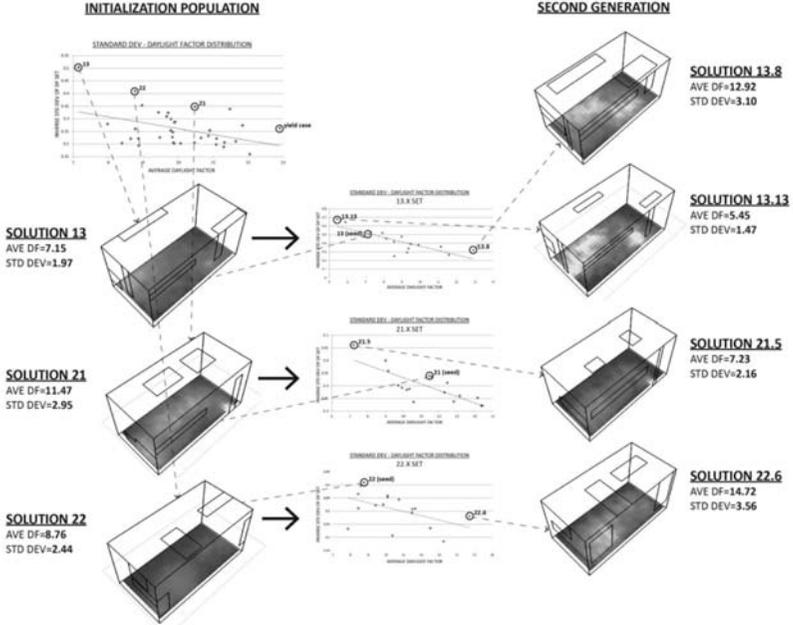


Fig. 15: Selected solutions from the evolutionary algorithm from initialization set and subsequent generation. (Source: Author)

Evolutionary Algorithm Performance Summary of Selected Solutions

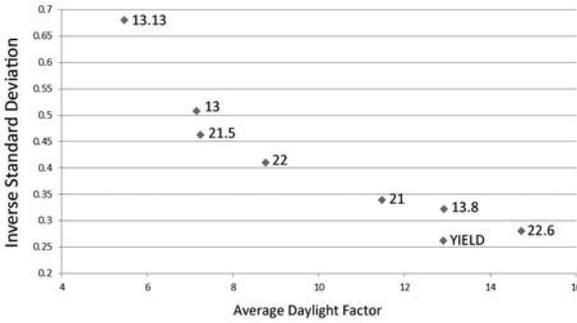


Fig. 16: X-Y graph synthesizing inverse standard deviation values and average daylight factor values for each selected solution. Note each solution outperforms the yield solution. (Source: Author)

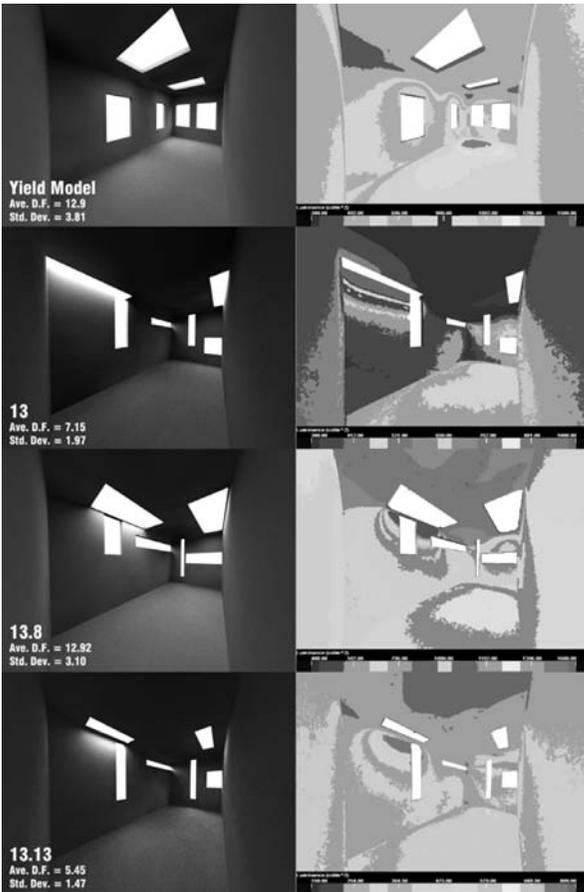


Fig. 17: Rendering and luminance studies comparing three of the selected solutions resultant from the algorithm. Note the narrowing of luminance levels into acceptable ranges and the emergence of known daylighting strategies (corner-positioned openings, high windows, etc.) in these randomly generated examples. (Source: Author)

process can be used to optimize a daylight scheme; however the potential of the algorithm as a tool at this stage is limited, and further refinement of the algorithm, the selection criteria, and the data processing is required before the real possibilities of the algorithm can be characterized.

5. CONCLUSIONS

Returning to the hypothesis of the paper, the experiments do support the assertion that the common practice of centering windows in walls results in inferior daylight performance; the optimized solutions developed by both algorithms instead positioned windows and skylights in proximity to the edges of the room. Irrespective of how much reflected light played into the analysis, it seems that placement of light sources at the perimeter of a space improves uniformity of daylight, perhaps by preventing the dark periphery that would normally occur when openings are concentrated along the middle axis of spaces. At any rate, daylight performance seems to be best when daylight is coordinated with walls. In achieving this performance-critical coordination, the algorithms presented in this paper appear to be useful strategies to serve this purpose and, in addition, strategies that are becoming more available to today's designers.

Reyner Banham called for an extension of engineering optimization beyond "acute industrial need" towards "human delight" (Banham 1984). What is of interest to me personally, as both a researcher and a designer, is that these algorithms also present evidence that daylight design is closely tied to interior space, and can be somewhat informal, providing some relief against the institutions of symmetry and stasis that so frequently informs the placement of windows and the skylights. In contrast to the indifferent 'yield' approach of centering windows in walls, the tools used in this research offer an organized approach that, with the support of simulation, leads to a higher-performing and more dynamic architectural space.

ACKNOWLEDGEMENTS

The author would like to thank the Department of Architecture at Ball State University for providing the research time necessary to continue with this project, and the Institute for Digital Fabrication of the Center for Media Design at Ball State University, along with Josh Vermillion, for support and advice with regards to computing and parametric design.

The author also thanks Masters in Architecture student Chris Alexander, who as a graduate assistant helped collect research data for this paper, particularly the optimization algorithm discussed in this paper during its development to several individual test cases, carrying out analysis, compiling data, and capturing graphics. Lastly the author would like to thank his family for their support at all levels.

REFERENCES

- Banham, Reyner. 1984. *The Architecture of the Well-Tempered Environment* (second edition). The University of Chicago Press; Chicago.
- Demers, Claude MH. 2007 "A Classification of Daylighting Qualities based on Contrast and Brightness Analysis." Conference Proceedings of the American Solar Energy Society (ASEE), SOLAR 2007: 7-12 July, Cleveland, Ohio.
- Eiben, A.E. and J.E. Smith. 2003. *Introduction to Evolutionary Computing*. Springer; Berlin.
- Energy Standard for Buildings Except Low-Rise Residential Buildings (ANSI/ASHRAE/IESNA standard 90.1-2007). 2007. Atlanta, GA : American Society of Heating, Refrigerating and Air-Conditioning Engineers.
- Le Corbusier. 1927. *Towards a New Architecture*. Payson & Clarke, Ltd. New York.
- Papalambros, Panos Y. and Douglass J. Wilde. 1988. *Principles of Optimal Design: Modeling and Computation*. Cambridge U.P.: Cambridge.
- Rea, Mark Stanley, ed. 2000. *IESNA Lighting Handbook*, 9th ed. New York, NY : Illuminating Engineering Society of North America.
- Sivanandam, S.N. and S.N. Deepa. 2008. *Introduction to Genetic Algorithms*. Springer: Berlin.
- Wright, Frank Lloyd. 1953. *The Future of Architecture*. Horizon Press: New York.

Zevi, Bruno. 1978. *The Modern Language of Architecture*. University of Washington Press; Seattle.

Zevi, Bruno. 1991. "Light as Architectural Form." *World Architecture* #14.

ENDNOTES

1. At this early stage of research, the algorithms have some technical shortcomings that are important to consider. The first of these shortcomings is the limitation in programming resources: ideally, these algorithms would be entirely automated, requiring no user input as they operate. Additionally, the algorithms have yet to be automated between programs, and instead data is passed off manually. The result is a great deal of tedium and an effective limit on the number of iterations which results in a rather coarse optimization. A second technical shortcoming involves Ecotect, where analysis is limited by the number of data points and the accuracy assigned to the analysis; limitations to each of the latter make the effects of internal and external reflectance also somewhat coarse. This issue has been addressed by rendering solutions using physics-based rendering that can show the effect of reflectance on light distribution.
2. The narrow objective of daylight performance in the research must also be considered, in that the experiments focus solely on daylight factor data as the criteria guiding the algorithms, and direct solar admittance, construction imperatives, available products and materials, etc. are issues that have no effect on the optimizations as they would in a architectural project bound by real world factors.
3. Yet another important observation from this research is that the relationship between daylight intensity and uniformity is one that at times is disruptive: high or low daylight values in a room tend to increase or decrease statistical variance respectively, impacting standard deviation and uniformity. These tendencies complicate the interpretation of results, since low daylight factors may be undesirable for a certain task, even though low values result in high uniformity and reduced glare.
4. Lastly, in terms of representing real world performance, the simulations and optimizations in this paper offer a useful prediction of daylight performance – but they because they are based on daylight factor analysis and the uniform sky conditions used in that analysis, the models do not represent performance under direct sunlight or mixed sun and reflected cloud light conditions. Because of the varying conditions of daylight over the year and depending upon location, optimizations of the sort proposed in this paper would be irrelevant, since introducing the huge range of daylighting conditions into the optimization would make it unresolvable. Like any simulation based on daylight factor analysis, we must accept that for a reasonable range of conditions (especially cloudy and mostly cloudy conditions) the conditions of the analysis will represent a part of real life.