Gaming Engine as a Tool for Designing Smart, Interactive, Light-Sculpting Systems

Constantina Varsami[1], Alexandros Tsamis[1], Timothy Logan[2]

[1] Rensselaer Polytechnic Institute, Troy, NY, USA
varsac2@rpi.edu; tsamis@gmail.com
[2] HKS, Inc, Dallas, TX, USA
tlogan@hksinc.com

Abstract. Even though interactive (Offermans et.al., 2013), adaptive (Viani et.al., 2017), and self-optimizable (Sun et.al., 2020) lighting systems are becoming readily available, designing system automations, and evaluating their impact on user experience significantly challenges designers. In this paper we demonstrate the use of a gaming engine as a platform for designing, simulating, and evaluating autonomous smart lighting behaviors. We establish the Human - Lighting System Interaction Framework, a computational framework for developing a Light Sculpting Engine and for designing occupant-system interactions. Our results include a. a method for combining in real-time lighting IES profiles into a single 'combined' profile - b. algorithms that optimize in real-time, lighting configurations - c. direct glare elimination algorithms, and d. system energy use optimization algorithms. Overall, the evolution from designing static building components to designing interactive systems necessitates the reconsideration of methods and tools that allow user experience and system performance to be tuned by design.

Keywords: User Experience, Human-Building Interaction, Smart Lighting, Lighting Simulation, Gaming Engine

1 Introduction

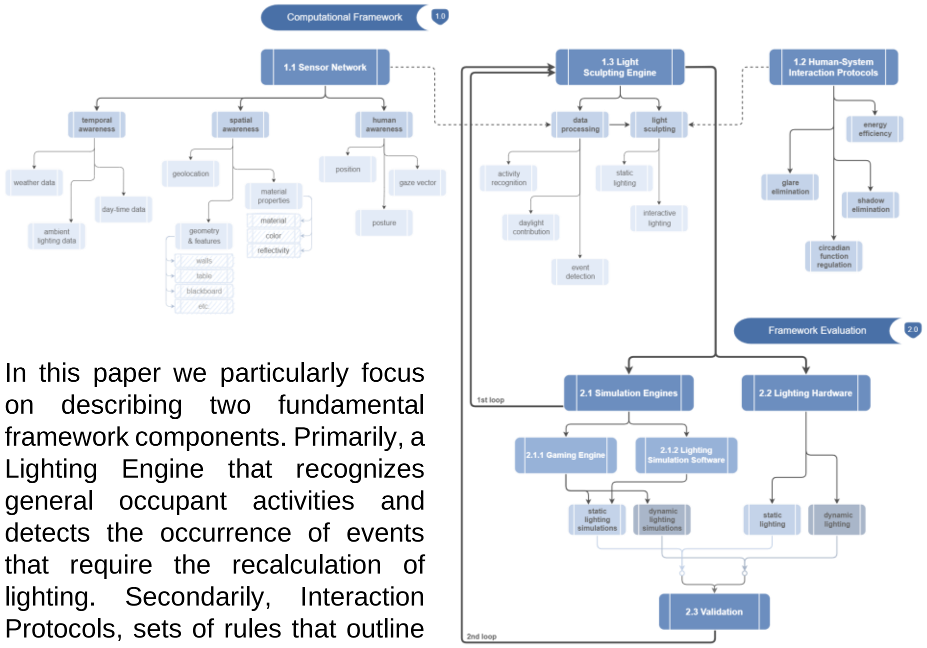
Gaming Engines are increasingly being perceived as useful design platforms by virtue of their ability to actively engage and immerse users. (Akanmu et.al., 2018) The architectural field is also progressively embracing the integration of gaming engines as tools that allow designers to interrogate concepts of immersion, interaction, and human collaboration. (Hoon et.al., 2002) At the same time, recent technological advances have significantly improved the visual realism afforded by gaming engines which has transformed them into powerful tools for simulating and evaluating the aesthetics and performance of highly complex building systems.

Intelligent lighting systems are an integral component to smart building design. Simulating them in a gaming engine allows for more than visualizing their static effect on the built environment; it enables designers to dynamically test in real-time lighting automations and controls as well as their impact on user experience. Even though studies on smart lighting systems have primarily focused on energy savings, improving user experience is another field that is gaining attention. (Sun et.al., 2020) As a result, balancing the need to accommodate occupants with the need to ensure the system's energy efficiency is central for modern lighting system design. (Aldrich et. al., 2010)

Modern LED based lighting systems are composed of digitally programmable parameters such as intensity and color temperature (Offermans et.al., 2013) which allow designers to regulate the quality and distribution of lighting. By dynamically controlling each individually addressable lighting element, visual and non-visual effects on user experience can be customized. (Linhart, 2010) The higher the number of LEDs the higher the resolution of the outputted lighting configurations and the more personalized the impact imparted to humans. Acknowledging the visual and physiological effects each lighting configuration has on humans (Bellia et.al., 2011) (Hatori et.al., 2017) is key to deliberately configuring lighting to improve occupant comfort, well-being, and performance (van Bommel et.al., 2004) (van Bommel, 2006).

2 Human - Lighting System Interaction Framework

Against this backdrop, we propose the Human - Lighting System Interaction (HLSI) Framework, a computational framework for designing smart lighting systems that acknowledge user-system interactions and regulate lighting to accommodate user and system needs (Figure 1). The HLSI framework is comprised of three components: the Sensor Network, the Interaction Protocols, and the Light Sculpting Engine. The Sensor Network provides system awareness on three levels: temporal, spatial, and human. The Interaction Protocols allow for user-system interactions to take place and the Light Sculpting Engine generates optimal lighting configurations that interface user-system interactions. The HLSI framework is evaluated by virtually simulating the generated lighting configurations and physically visualizing them in the testbed where the lighting hardware are installed. The effectiveness of the framework is collectively validated by comparing virtual and physical lighting configurations. Ultimate goal of the HLSI framework is to assist with tuning user experience and lighting system performance by design.



In this paper we particularly focus on describing two fundamental framework components. Primarily, a Lighting Engine that recognizes general occupant activities and detects the occurrence of events that require the recalculation of lighting. Secondarily, Interaction Protocols, sets of rules that outline interactive behaviors between the occupants and the system; protocols that regulate the system's efficiency and eliminate glare on an occupant level.

Figure 1.
[1.0] HLSI Framework
[2.0] HLSI Framework Evaluation

2.1 **Light Sculpting Engine**

"Light Sculpting" constitutes a strategy based on which the right amount of illumination is automatically delivered where and when needed (Department of Energy, 2020). Such an occupancy-centric lighting approach is highly contingent upon utilizing digitally programmable LED based lighting fixtures that allow designers to dynamically adjust the spectral content and directionality of each individual lighting element and employ the system's full resolution to deliver "sculpted illumination".

Digitally programmable lighting fixtures coupled with dynamic lighting controls collectively contribute to the development of light sculpting systems. Light sculpting engines control such systems by computing each time the optimal, for the circumstances, lighting configuration. Hereafter, we outline a method for developing a light sculpting engine by addressing the two fundamental tasks of setup and optimization.

2.1.1 **Optimally Sculpt Lighting Configurations**

To optimize lighting configurations based on occupancy requirements, physical and digital spaces are set up and calibrated to ensure that the Lighting Engine accounts for the geometrical specifications of the environment, the manufacturer specifications of the lighting system, and the relationship between programmatical functions and lighting requirements. This process happens only once for each space where the lighting system is installed and allows for future system operability. The setup process is completed in three stages: space setup, hardware setup, and activities setup.

***Space Setup*.** In the physical space, features whose illumination is to be controlled are identified and virtually discretized; that is, their surfaces are reduced to a discrete collection of points. Lighting scenarios are designed by referencing all sample points and assigning specific illuminance levels to each one, therefore, by allocating levels of illumination in space (Figure 2).

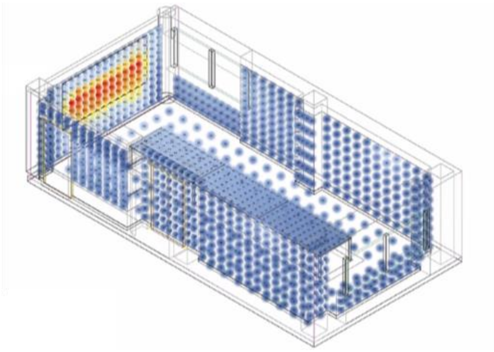


Figure 2. Space Setup

***Hardware Setup*.** The lighting hardware are project specific. They are configured by the manufacturer based on the illumination requirements of the environment in which they will be installed. During hardware setup, the structure of each fixture is encoded so that virtual lighting elements in the Gaming Engine correspond to respective physical ones; both physical and virtual elements can be simultaneously controlled. Furthermore, during hardware setup the amount of illumination each lighting element individually contributes to each sample point in space is recorded in the Contribution Matrix, a space and hardware specific input that is fundamental for optimizing lighting.

***Activities Setup*.** Given the programmatical function of each space prospective occupant activities and their lighting requirements need to be registered. To that end, lighting scenarios dictating the general illumination levels of all sample points in space are designed per registered activity. These scenarios afford general activity lighting and are later adjusted to customize lighting per occupant.

Optimization. Having completed all three stages of setup, optimally sculpted lighting configurations can be computed in real-time. To optimize lighting configurations we individually control the intensities of all light directionalities and assign desired illumination values to all sample points. Optimizing lighting is a matter of identifying the multipliers (values from 0 to 1) that will adjust (activate, dim, deactivate) the intensity of each light directionality and deliver the right amount of illumination where and when needed.

To optimize lighting we use the Bounded-Variable Least-Squares (BVLS) algorithm which solves linear least-squares problems with predefined upper and lower variable bounds (Stark et.al., 1995). The solver computes multipliers (x's) for all light directionalities so that each feature point in space gets the targeted amount of lighting (y's) generating a uniquely sculpted lighting configuration. To estimate the optimal x's the solver minimizes the error (e), the difference between the targeted and the computed y's ($y_{tp} – y_p$) (Stark et.al., 1995):

$$e = \arg \min_{l \leq x \leq u} \|\mathrm{Ax} - \mathrm{y}\|_2 \qquad (1)$$

where l, x, u ∈ Rn, y ∈ Rp, and A is a p by n matrix. Bounds l=0 (lower) and u=1 (upper); thus, 0 ≤ x ≤ 1

Circumvent the Gaming Engine's Lighting Limitations. To represent lights with visual accuracy Gaming Engines support the application of IES Profiles on lights, profiles that deliver information about the distribution of photometric data and represent emission patterns based on manufacturer specifications. (Sisson et.al., 2018) The IES file format was created so that photometric data (total luminous flux, luminous intensity) (Kelechava, 2020) are transferred electronically. When imported in Gaming Engines, IES files are used as masks that modify default light patterns to achieve physical accuracy. (Dravid, 2015)

Modern LED fixtures are comprised of hundreds of lighting elements. To accurately visualize them in a Gaming Engine, IES profiles have to be applied to each individual element. However, modeling as many lights and applying IES profiles to each one is computationally expensive. To circumvent software performance limitations (we use UNITY (version 2021.1.0f1) which is limited to rendering up to 24 real-time lights per tile) we developed an algorithm that computationally combines IES profiles into a 'combined' fixture profile which is applied to a single virtual light affording the same visual result. The algorithm sums elementwise the light distribution data of the various IES files into a single 'combined' profile. By computationally combining and updating IES profiles, fixture emission patterns are updated near real-time, qualifying the Gaming Engine as a design tool for simulating and evaluating modern lighting systems.

2.1.2 Software Communication Architecture

The Architecture that interconnects the Lighting Engine with the rest of the software and hardware elements (Figure 3) is composed of, firstly, sensing devices that capture occupant movement and photometric measurements from the built environment and transmit them to the Data Processing Component (DPC) of the Lighting Engine. The DPC is responsible for denoising the collected data, applying activity recognition algorithms, and generating events (notifications) that inform the Light Sculpting Component (LSC) about required illumination updates. The results (optimally sculpted profiles/configurations) are transmitted to both Simulation Engine **and** Lighting Hardware where the optimized lighting configurations are virtually and physically represented, validating the effectiveness of the Engine.

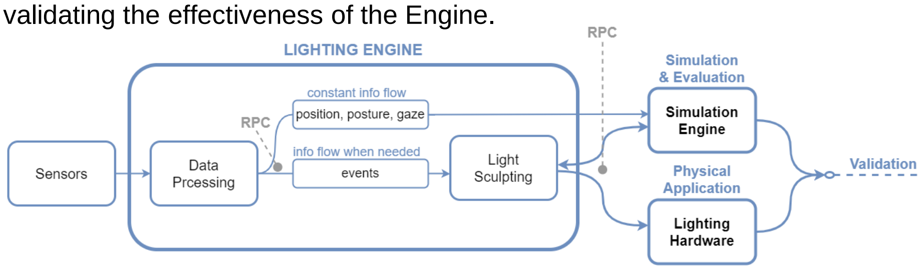


Figure 3. Software Communication Architecture

The employed communication protocol, that allows for the exchange of real-time data between the Lighting Engine, the Simulation Engine, and the Lighting Hardware, is an Open Network Computing (ONC) Remote Procedure Call (RPC) protocol. This protocol allows for an active client (Simulation Engine & Lighting Hardware) to make a call to a server (Lighting Engine) which, in turn, sends back a reply. (Thurlow, 2009) Transport protocols underneath RPC define how messages are passed. We use the TCP/IP Transmission Control Protocol as it provides a reliable connection between applications. (IBM Corporation, 2015) Most importantly, the preceding protocol allows the Lighting Engine to similarly control both simulation software and lighting hardware.

2.2 Human – Lighting System Interaction Protocols

To address occupant-system interactions, the Engine needs advanced levels of human awareness, however, by setup it can only recognize the general activity. To improve the Engine's awareness and increase the resolution of its capacity to interact, it has to be coupled with Interaction Protocols, sets of rules that outline interactive behaviors amongst occupants and the lighting system. We have developed two Interaction Protocols that regulate the system's energy use and eliminate direct glare based on occupant activity. The resulting configurations integrate preliminary scenarios (static, activity-based lighting) with energy or glare preferences that personalize configurations on an occupant level (dynamic, human aware lighting).

Regulate Energy Use. To regulate the system's efficiency we have designed three energy modes: eco, eco^{+}, eco^{++}. The first affords energy savings by delivering activity-based lighting. The second mode builds upon the first but also uses an energy savings algorithm to lower imperceptibly the lighting levels away from occupants. The third mode uses the same algorithm but lowers the lighting levels away from occupants much more perceptibly, affording increased energy savings. We recompute this algorithm whenever significant occupant movement occurs so that lighting updates accordingly.

The energy savings algorithm operates as follows. Around the center of each occupant's body two spheres (radii R1, R2) are formed (Figure 3). Values R1 and R2 are parameters that can be adjusted in advance by the designer of the system. For each sample point (p) in the room the *targeted lux value* of the sample point (y_{tp}) is calculated. It is a value between the *general activity lux value* (y_{gp}) and the *energy efficient activity lux value* (y_{ef}) and it is computed as a function of the point's distance (r) to the closest occupant. Lux values (y_{gp}, y_{ef}) are retrieved from a chart that contains predetermined lux values for each energy mode and activity. Specifically:

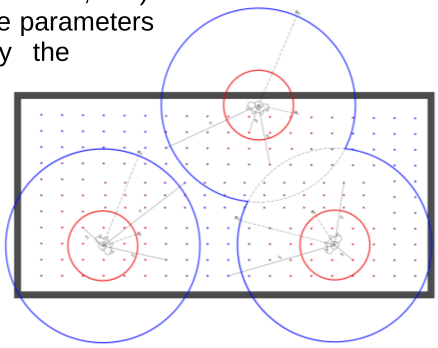


Figure 4. Determination of Targeted Activity Lux Values per Sample Point

if $r < R1$, then $y_{tp} = y_{gp}$ | if $r > R2$, then $y_{tp} = y_{ef}$ | if $R1 \leq r \leq R2$, then y_{tp} is a. or b.

a. Linear Behavior of y_{tp} as a function of y_{gp} and r:

$$\mathrm{ytp} = \frac{(\mathrm{R2}-\mathrm{r})\times(\mathrm{ygp}-\mathrm{yef})}{(\mathrm{R2}-\mathrm{R1})} + \mathrm{yef} \qquad (2)$$

b. Cosinusoidal Behavior of y_{tp} as a function of y_{gp} and r:

$$ytp = \left\{\cos\left[\pi - \frac{(\mathrm{R2}-\mathrm{r})\times\pi}{(\mathrm{R2}-\mathrm{R1})}\right]\times\frac{(\mathrm{ygp}-\mathrm{yef})}{2}\right\} + \frac{(\mathrm{ygp}+\mathrm{yef})}{2} \qquad (3)$$

Glare Elimination. To eliminate direct glare (We define as direct glare the glare that occurs when looking directly at the source of light. It does not include any light bouncing on surfaces before reaching the eye.) we limit the amount by which directionalities with glare potential contribute to achieving the targeted illumination. The glare potential is computed by interrelating each light vector with the gaze vectors of each occupant. If a directionality has glare potential, we dim or completely deactivate it to mitigate or respectively eliminate its effect. The directionalities that do not have glare potential we can fully utilize. We recalculate the glare elimination algorithm and update lighting whenever occupants significantly move or look around to account for their updated positions and gaze directions.

More specifically, a glare multiplier (g) that ranges within the domain [0,1] is assigned to each directionality retaining, mitigating, or eliminating its contribution. The glare multiplier is calculated as a function of the non-reflex or straight angle (θ) formed between each occupant's gaze vector and each of

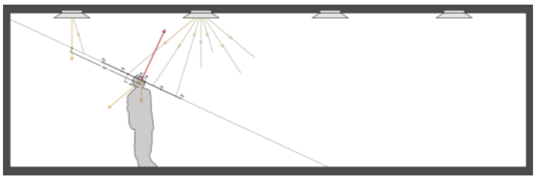


Figure 5. Gaze Vector (red) | Light Vectors (yellow) | Angles (θ) | Distance (δx)

the troffer's light vectors, and the distance (δx) from the human visual system (eyes) to the point where each light vector intersects with the plane whose normal is the occupant's normalized gaze vector (Figure 5). To simultaneously consider the contribution of θ and δx we primarily define and calculate g_θ, a multiplier that indicates the effect of glare solely based on θ, which is then used to calculate g as a function of g_θ and δx.

The glare elimination algorithm operates as follows. For each occupant the magnitude of g_θ is computed for all light directionalities by means of comparing θ against two threshold values θ1 and θ2 which are parameters adjusted in advance by the designer of the system. For $0 \le \theta1 < \theta2 \le 180$:

if $0 \le \theta < \theta1$, then $g_\theta = 1$ | if $\theta2 < \theta \le 180$, then $g_\theta = 0$ | if $\theta1 \le \theta \le \theta2$, then g_θ is a. or b.

a. Linear Behavior of g_θ as a function of θ:

$$g\theta = \frac{(\theta2 - \theta)\times(g\theta\max - g\theta\min)}{(\theta2 - \theta1)} + g\theta\min \qquad (4)$$

b. Cosinusoidal Behavior of g_θ as a function of θ:

$$g\theta = \left\{\cos\left[\pi - \frac{(\theta2 - \theta)\times\pi}{(\theta2 - \theta1)}\right]\times\frac{(g\theta\max - g\theta\min)}{2}\right\} + \frac{(g\theta\max + g\theta\min)}{2} \qquad (5)$$

For the light vectors that indicate glare potential based on θ ($g_\theta \neq 1$) a secondary calculation has been established to determine whether distance δx allows for glare. For each occupant, light fixture, and directionality with glare potential based on θ the intersection point of the light vector and the plane whose normal is the occupant's normalized gaze vector is identified. If the intersection point is located above the light fixture the light vector does not cause glare (g = 1). Otherwise, distance δx is computed and compared to thresholds x1 and x2, parameters adjusted in advance by the designer of the system.

if $0 \le \delta x < x1$, then $g = g_\theta$ | if $\delta x \ge x2$, then $g = 1$ | if $x1 \le \delta x \le x2$, then g is a. or b.

a. Linear Behavior of g as a function of gθ and δx:

$$g = \frac{(\delta x - x1)\times(g\max - g\theta)}{(x2 - x1)} + g\theta \qquad (6)$$

b. Cosinusoidal Behavior of g as a function of g_θ and δx:

$$g = \left\{\cos\left[\pi - \frac{(\delta x - x1)\times\pi}{(x2 - x1)}\right]\times\frac{(g\max - g\theta)}{2}\right\} + \frac{(g\max + g\theta)}{2} \qquad (7)$$

3 Results

General result of our study constitutes the HLSI Framework which uses a Gaming Engine as a design tool for developing smart, interactive, light-sculpting systems. The HLSI Framework and the resulting algorithms (for overcoming software performance limitations, regulating system energy use, and eliminating direct glare) will be applied and discussed in the context of a case study.

Particularly, the HLSI framework was applied to a virtual conference room where eight smart lighting fixtures were installed. Each fixture is composed of hundreds of digitally controllable LEDs that collectively afford 53 distinct directionalities (color mapped in Figure 6) which can be individually controlled (activated, dimmed, deactivated) allowing for light to be sculpted.

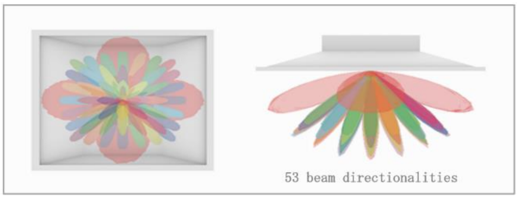


Figure 6. Case Study Hardware Specifications

To virtually represent each light fixture with physical accuracy and without exceeding the Gaming Engine's performance limitations we employed the algorithm we developed for merging multiple IES Profiles into a single 'combined' profile. Given that light intensity cannot be accurately represented and therefore evaluated in the Gaming Engine, the algorithm was assessed based on the projected light emission patterns (Figure 7). The results prove that the single combined profile (Figure 7, right) properly captures the light pattern of the uncombined ones (Figure 7, left), validating the capacity of the developed algorithm to overcome performance limitations without compromising on accuracy.

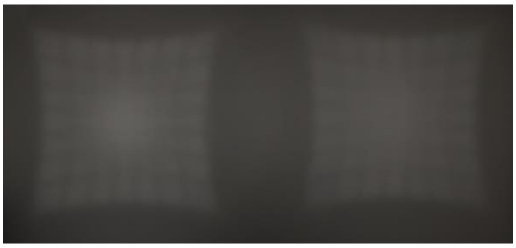


Figure 7. Light Emission Patterns projected on Horizontal Plane; Combined IES Profile (right) and Singular IES Profiles (left)

Having completed the setup, the BVLS solver was employed to compute static, activity-based lighting configurations. The implemented lighting scenarios center on delivering Uniform Lighting (uniformly 300 lux), Highlighting the Conference Table (table 300 lux, rest 50 lux), and Highlighting the Blackboard (blackboard 300 lux, rest 50 lux). By simulating these lighting configurations we visually evaluate whether the distribution of light matches the intended distribution per designed scenario. According to the simulations (Figure 8) the targeted lighting scenarios are successfully computed and the Lighting Engine's capacity to compute static, activity-based lighting is validated.

(a) Uniform | uniform 300 lux lighting (benchmark) (b) Table | table at 300 lux rest at 50 lux | 65% energy savings (c) Blackboard | blackboard at 300 lux rest at 50 lux | | 81% energy savings

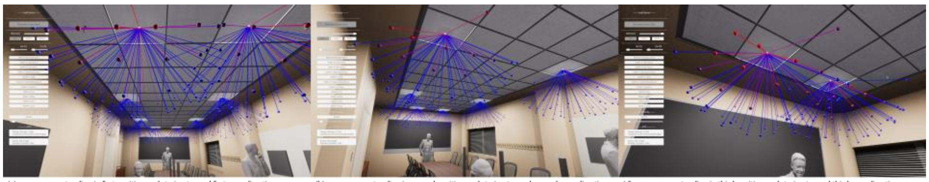
Figure 8. Static, Activity-Based Lighting Configurations
(left to right) Light Uniformly, Light the Conference Table, Light the Blackboard

To evaluate the interaction protocols we implemented primarily the energy savings algorithm for each energy mode (eco, eco+, eco++) and for each lighting scenario. Uniform lighting in eco mode constitutes the benchmark for our calculations. The results are strongly dependent on the values of the parameters R1 and R2; in Figure 9 we used R1=0.5, R2=1.0. Based on these values, the energy savings range from 45% to 76% for the Uniform scenario and reach 93% when Highlighting the Blackboard. The employed algorithm couples moderate energy savings with imperceptible lighting changes (eco+) and increased energy savings with perceptible lighting changes (eco++). This lighting strategy is limited in that system performance and user experience constitute conflicting design parameters.

(a) Uniform | eco mode | 0% energy savings (benchmark) (a) Uniform | eco^{+} mode | 45% energy savings (a) Uniform | eco^{++} mode | 76% energy savings

Figure 9. Dynamic, Human Aware Lighting Configurations | Regulate Energy Use

By implementing the direct glare elimination algorithm upon the benchmark lighting scenario (Uniform, eco mode) we computed lighting configurations for varying occupant positions and gaze directions and calculated the associated energy savings. The results are strongly dependent on the values of the parameters θ1 and θ2 as well as x1 and x2; in Figure 10 we used θ1=120°, θ2=165°, x1=1m, x2=2m. Based on these values the energy savings range from 0.16% to 1.54%. This means that eliminating glare improves user experience without affecting negatively nor significantly system performance.



(a) camera man standing in first position and staring toward first gaze direction (b) camera man standing in second position and staring toward second gaze direction (d) camera man standing in third position and staring toward third gaze direction

Figure 10. Dynamic, Human Aware Lighting Configurations | Red: eliminated directionalities | Purple: dimmed directionalities | Blue: fully utilizable directionalities

4 Conclusions/Discussion

To address the increasing complexity associated with smart lighting systems, in this paper we introduced a framework that uses a gaming engine as a tool for designing, simulating, and evaluating the behavior of interactive light-sculpting systems. We demonstrated the application of this framework in a case study and explored the associated affordances and limitations. Explicitly, we verified the capability to deliver static, activity-based lighting as well as to personalize lighting configurations on an occupant level. We determined that saving energy and imperceptibly changing lighting are competing design drivers as well as that eliminating direct glare improves user experience without significantly affecting the efficiency of the system.

The evolution from designing static building components to designing dynamic systems that interact with occupants necessitates the reconsideration of the employed design methods and tools. In the field of lighting design, this calls for the implementation of a system-oriented design approach where strategies and tools allow designers to simulate and evaluate system controls and behaviors as they change in response to users. Gaming Engines, as opposed to static, time-invariant modes of representation, allow designers to negotiate between competing parameters in real-time. This fact qualifies them as platforms where smart, interactive, light-sculpting systems can be trained and tuned by design.

Acknowledgements. This work was produced as a collaboration between the Center for Lighting Enabled Systems and Applications (LESA), and the Center for Architecture Science and Ecology (CASE), both at RPI, Lumileds LLC, and HKS, Inc. in the context of a DOE funded project (Office of Energy Efficiency and Renewable Energy). The authors would like to sincerely acknowledge the contribution of all project collaborators and the support of the DOE.

References

Akanmu, A. A., Ojelade, A., & Bulbul, T. (2018). Gaming Approach to Designing for Maintainability: A light Fixture Example. ISARC 2018 - 35th International Symposium on Automation and Robotics in Construction and International AEC/FM Hackathon: The Future of Building Things, July. https://doi.org/10.22260/isarc2018/0154.

Aldrich, M., Zhao, N., & Paradiso, J. (2010). Energy efficient control of polychromatic solid state lighting using a sensor network. Tenth International Conference on Solid State Lighting, 7784, 778408. https://doi.org/10.1117/12.860755

Bellia, L., Bisegna, F., & Spada, G. (2011). Lighting in indoor environments: Visual and non-visual effects of light sources with different spectral power distributions. Building and Environment, 46(10), 1984–1992. https://doi.org/10.1016/j.buildenv.2011.04.007

Department of Energy. Office of Energy Efficiency and Renewable Energy. (2020). Spatially Adaptive Tunable Lighting Control System with Expanded Wellness and

Energy Saving Benefits. Available: https://www.energy.gov/eere/ssl/downloads/spatially-adaptive-tunable-lighting-control-system-expanded-wellness-and-energy. Last accessed 04 June 2022.

Dravid, A. (2011). Understanding IES Lights. Available: http://www.cgarena.com/freestuff/tutorials/max/ieslights/. Last accessed 2 March 2021.

Hatori, M., Gronfier, C., Van Gelder, R. N., Bernstein, P. S., Carreras, J., Panda, S., Marks, F., Sliney, D., Hunt, C. E., Hirota, T., Furukawa, T., & Tsubota, K. (2017). Global rise of potential health hazards caused by blue light-induced circadian disruption in modern aging societies. Npj Aging and Mechanisms of Disease, 3(1), 5–7. https://doi.org/10.1038/s41514-017-0010-2

Hoon, M., Jabi, W., & Goldman, G. (2003). Immersion, interaction, and collaboration in architectural design using gaming engine. Proceedings of the 8th CAADRIA Conference, 721–738. http://orca.cf.ac.uk/27295/.

IBM Corporation. (2015). TCP/IP TCP, UDP, and IP protocols. Available: https://www.ibm.com/docs/en/zos/2.2.0?topic=internets-tcpip-tcp-udp-ip-protocols. Last accessed 14th June 2022.

Kelechava, B. (2020). IES Standard File Format for Photometric Data, IES LM-63-19. Available: https://blog.ansi.org/2020/02/standard-file-photometric-data-ies-lm-63-19/#gref. Last accessed 11th June 2022.

Linhart, F. (2010). Energetic, Visual and Non-Visual Aspects of Office Lighting. 4634, 295.

Offermans, S., Van Essen, H., & Eggen, B. (2013). Exploring a hybrid control approach for enhanced user experience of interactive lighting. HCI 2013 - 27th International British Computer Society Human Computer Interaction Conference: The Internet of Things, 1–9. https://doi.org/10.14236/ewic/hci2013.16.

Sisson, D, Pedersen, L. (2018). IES Profiles. Available: https://renderman.pixar.com/ies-profiles. Last accessed 2 March 2021.

Stark, P., & Parker, R. (1995). Bounded-variable least-squares: an algorithm and applications. Computational Statistics, 394, 1–13. http://www.stat.berkeley.edu/users/stark/Preprints/bvls.pdf.

Sun, B., Zhang, Q., & Cao, S. (2020). Development and implementation of a self-optimizable smart lighting system based on learning context in classroom. International Journal of Environmental Research and Public Health, 17(4). https://doi.org/10.3390/ijerph17041217.

Thurlow, R. (2009). RPC: Remote procedure call protocol specification version 2. RFC 5531, May.

van Bommel, W. J. M. (2006). Non-visual biological effect of lighting and the practical meaning for lighting for work. Applied Ergonomics, 37, 461–466.

van Bommel, W. J. M., Wout & Beld, G. J. (2004). Lighting for work: A review of visual and biological effects. Lighting Research and Technology, 36(4), 255–269. https://doi.org/10.1191/1365782804li122oa.