Human-Building Collaboration: Toward Lighting Enabled Collaborative System Design

Constantina Varsami\textsuperscript{1}, Daniel Rosenberg\textsuperscript{2}, Diego Pinochet\textsuperscript{3,4}, Alexandros Tsamis\textsuperscript{1},
\textsuperscript{1}Rensselaer Polytechnic Institute, Troy, NY, USA
\textsuperscript{2}Carnegie Mellon University, Pittsburgh, PA, USA
\textsuperscript{3}Massachusetts Institute of Technology, Cambridge, MA, USA
\textsuperscript{4}Adolfo Ibañez University, Santiago, Chile

Abstract

The profession of lighting design is evolving given that contemporary lighting systems increasingly rely on cutting-edge computational technologies, sensors, and IOT systems. This trend requires that designers incorporate ideas of automation, dynamic controls, and user-system interaction into their design logic. However, real-time lighting simulations are constrained by inherent limitations arising from the reductive assumptions inevitably introduced in simulated lighting environments. This raises the question of how designers can account for the discrepancies between simulated and real lighting environments, and how collaboration between humans and autonomous lighting systems can bridge this gap. To address this question, a protocol for designing collaborative interactions between humans and systems is proposed. This protocol builds on the Human-Lighting System Interaction Framework (Varsami et al., 2022) and demonstrates how the integration of human and system intelligence fine-tunes lighting qualities in a given space. The paper shows how interactive lighting systems can locally customize lighting based on user preferences in real-time and how global lighting configurations can be adjusted over time. Specifically, the paper demonstrates: a) Human-system collaboration assumptions and goals as well as how the protocol can be integrated into digitally programmable lighting systems. b) Implementations of collaboration that reveal how system autonomy, performance, and user experience are improved over short and long-term timeframes. c) How lighting design can be enhanced qualitatively, beyond what simulation-driven design optimization methodologies can afford. The associated affordances and limitations are discussed with respect to existing lighting simulation design frameworks.

Highlights

- the system delivers the requested adjustments; past user adjustments are considered in future system solutions
- the process of updating system solutions continues until the request range becomes consistent, solutions converge and, if possible, requests are minimized
- collaboration enhances user wellbeing as well as lighting system autonomy and performance over time

Introduction

Contemporary lighting systems automatically detect occupant motion as well as changes in ambient lighting (Helvar, 2021). They also integrate color preferences personalizing lighting solutions (Debiasi, 2020) to improve user experience. Designing dynamic lighting systems has inevitably introduced a new set of challenges amongst which the need for designing immersive, interactive environments for performing dynamic, real-time lighting simulations (May et al., 2020).

On the other hand, lighting simulations are intrinsically limited by the reductive assumptions inevitably introduced in simulated computational environments. According to Turkle (2009), in order to simulate responsibly users have “to do” and “to doubt” simulations, accepting that they are embedded with a level of uncertainty. In this context, Reinhart and Breton (2009) evaluated two widely used lighting simulation software (Autodesk 3ds Max Design 2009 and Daysim 3.0) against illuminance measurements released by the International Commission on Illuminance (CIE, 2006). This work shows error rates ranging from 28% to 31%, with the majority of simulations falling within a plus/minus 25% error range. In some cases error rates as high as 73% and 110% were recorded.

In this paper, the concept of human-building collaboration is adopted in the context of intelligent lighting system design. Collaboration is devised as an advanced method for designing interactive systems that attend to lighting requirements while operating within the prescribed bounds of simulation affordances and limitations.

In this paper, the concept of human-building collaboration is adopted in the context of intelligent lighting system design. Collaboration is devised as an advanced method for designing interactive systems that attend to lighting requirements while operating within the prescribed bounds of simulation affordances and limitations. Collaboration collectively employs human and artificial intelligence to configure optimal lighting configurations. Specifically, the paper demonstrates (a) how collaboration can be integrated into the automations and controls of any digitally programmable lighting system, (b) how system solutions can be adjusted to user preferences in real-time, (c) how user preferences can
inform the system over time, (d) how collaboration can deliver lighting configurations that transcend simulation limitations, and (e) metrics for evaluating collaborative human-system interactions. This work extends the Human-Lighting System Interaction (HLSI) Framework (Varsami et al., 2022), a computational framework for sculpting light and delivering activity-tailored illumination where and when needed (Figure 1).

Human-Building Collaboration

By integrating computational models in the control units of systems that are installed in the built environment artificial cognition is embedded in spaces where humans operate daily, fostering bidirectional exchanges and advanced interactions between the two modes of intelligence. Relational interactions and collaborative opportunities between humans and intelligent building systems have been discussed by Rosenberg and Tsamis (2019). Their human-building collaboration model (Figure 2) re-frames the relationship between humans and buildings from a simple action-response to a process that fully engages humans and embedded systems in action-detection-response-sensing, allowing them to bidirectionally exchange feedback and share their intelligence. In this paper, the human-building collaboration model is adopted as a guide for designing collaborative human-system interactions.

Human – Lighting Collaboration

The overarching goal behind establishing collaborative interactions is to design lighting configurations that exceed the ones users, designers, and systems could individually attain. Co-designing lighting configurations assumes that users can communicate to the system their desire to make a lighting adjustment, for example, with a gesture or through an interface. User feedback informs the system whether lighting needs to be locally increased or decreased and by how much. The location of the user indicates the area around which lighting needs to be adjusted. Since the protocol is implemented computationally, humans are modelled based on the coordinates of their head and their lighting preference (e.g., 350 lux). From the z coordinate the system deduces whether the user is sitting or standing and from the lux value the need for an adjustment and its magnitude (deviation from active solution). In a physical context, location is sensed and preference is communicated. All requests are associated with the location (x, y, z) from which they originate rather than being particular to the person who requested them. Thus, locally customizing lighting literally means customizing it for a particular location rather than for a particular person in space.

Similar to any type of interaction, users and system can have their individual goals while working toward achieving the common goal that brought them together in the first place. For example, while co-designing lighting configurations users want their lighting preference to be instantly met whereas system objectives include locally customizing and globally optimizing lighting configurations in real time. Over time, systems also aim to anticipate local patterns of preferences so that future system responses automatically cater more closely to local lighting needs.

This paper focuses on designing the behaviors that enable the system to simultaneously attend to general lighting requirements and individual occupant needs in both short and long-term time frames. By employing the protocol of collaboration (a) lighting is globally optimized in real-time, (b) user requests are addressed locally in real-time, (c) the system learns to anticipate future requests over time, (d) user requests may be minimized over time, and (e) system autonomy and performance are improved over time.

Figure 1: HLSI Framework (updated and redrawn from Varsami et al., 2022)

Figure 2: The Human-Building Collaboration Model; Rosenberg & Tsamis, 2019 (redrawn by author)
Collaboration Protocol – real-time optimization

To globally optimize lighting configurations, the geometrical specifications of the built environment and the technical characteristics of the embedded lighting system need to be computationally encoded. These mean features whose illumination is to be dynamically controlled (e.g., table, north wall, blackboard, etc.) are discretized; that is, they are reduced to a discrete collection of points (sample points). Lighting scenarios are designed by referencing all sample points and assigning specific illuminance levels (lux values) to each point. A vector of target lux values (TLVs) contains information on the lux levels of all sample points; thus, it defines the desired illumination throughout the space. Encoding digitally programmable lighting systems entails reducing each element (pixel) to its ID, direction of emission, and intensity. Optimizing lighting is a matter of estimating the intensity multipliers for all pixels. Multipliers range from 0 (no light) to 1 (max light) and they adjust (activate, dim, deactivate) each individual pixel to collectively deliver illumination that most closely approximates the targeted vector where and when needed.

The optimization process employs a Bound-Variable Least-Squares (BVLS) solver every time a change is sensed in the general activity of the room (e.g., activity shifts from “highlight the conference table” to “highlight the blackboard”) and new multipliers are estimated in real-time. An extended discussion on lighting optimization can be found in Varsami (2022).

Collaboration Protocol – real-time customization

To customize global lighting where adjustments have been requested, two threshold values (r1 and r2, where r1 < r2) are initially defined by the designer to determine the magnitude of the effect each request will have on the surrounding space. In essence, r1 and r2 are the radiiuses of two spheres that are centered around the body of each occupant who requests a change, defining the sphere of implication of his preference. To initiate the customization process, the proximity of the location at which a request was received with the highlighted feature (H0) of the active lighting scene (e.g., table) is primarily compared against threshold r1. This means that if any one of the sample points that describe the highlighted feature falls within the inner sphere it is assumed that the request is affecting only the highlighted feature. Otherwise, the request is affecting all but the highlighted feature. Any time the system receives an adjustment request the active vector of TLVs is re-computed in real-time. Specifically, each sample point (p) is associated with a new TLV (yp) that is determined as a function of the current lux value of the point (yp), the distance (dp) of the point to the nearest occupant, and the lux preference of that occupant (yo) as follows:

1. if the sample point falls within the inner sphere of the nearest occupant and, at the same time, belongs to the highlighted feature (H0) of the active lighting scene the nearest occupant’s lux preference is imposed on it
   if (dp ≤ r1 & p ∈ H0) | yp = yo

   • if it falls in between the two spheres and also belongs to the highlighted feature a value that ranges between the current (yp) and the preferred (yo) lux value is calculated as a function of the point’s distance to the nearest occupant. Here we demonstrate (a) linear and (b) cosinusoidal behaviors for that function; other behaviors can alternatively be integrated as well

   a. ynp = (1 − \[(d_p − r_1)/(2r_2 − r_1)\])yo + \[(d_p − r_1)/(r_2 − r_1)\]yp
   b. ynp = cos(\[(d_p − r_1)r_2/(r_2 − r_1)\]π)yo + \[(d_p − r_1)/(r_2 − r_1)\]yp

2. on any other occasion (point falls within the outer sphere of the nearest occupant but does not belong to the highlighted feature or point falls outside the outer sphere) the sample point will retain its lux value

   if (dp ≤ r2 & p ∈ H0) or if (dp > r2) | yp = yp

The resulting TLV vector is fed into the optimization solver which estimates new intensity multipliers in real-time, updating the distribution of light and locally customizing global configurations.

Collaboration Protocol – “co-design” of lighting scenes over time

Local adjustments are stored over time separately for each lighting scene (“highlight the table”, “highlight the blackboard”, etc.) and the system develops the capacity to proactively adjust lux values generating new TLV vectors based on user input. One of the objectives for this long-term behavior is that re-designed global lighting configurations converge over time to solutions that minimize user requests. Re-designed solutions primarily replace the original configurations the designer embedded in the system expecting - to the best of human knowledge, as aided by design and simulation tools - that all lighting requirements would be addressed. After that point, each re-designed solution modifies the latest configuration with which a scene was initialized.

For any specific scene such as “highlight the table” the designed behavior operates as follows:

• the first time the scene is activated the original vector of TLVs (TLVorig) the designer embedded is deployed as the active vector (TLVact)

   TLVact = TLVorig

• while the same activity is taking place, anytime occupants request for a local adjustment the algorithm that customizes lighting is evoked and the resulting vector of TLVs that accounts for requested occupant preferences (TLVact) replaces the active one ‘baking’ local adjustments into the original lighting configuration
TLV\textsubscript{s} = TLV\textsubscript{o}

- when the system senses a change in the global activity and before the scene changes, the latest vector of TLVs that contains information about all the adjustments that were requested across the space during this iteration (TLV\textsubscript{o}) is stored in a matrix. This matrix is initialized as a vector containing solely the original vector of TLVs for the specified scene
- after each iteration, the resulting TLV\textsubscript{s} vector is added as a new column to the matrix along with an assigned weight that indicates the impact of the configuration. All observations recorded to date are then averaged (TLV\textsubscript{avg}) according to their weights
- each value of the TLV\textsubscript{avg} vector represents the weighted average of lux delivered at a particular point in space. For a specific sample point (p) the TLV\textsubscript{avg} lux value that corresponds to that point (y\textsubscript{avg}) represents the weighted average of the lux values that were delivered at point p by the end of each iteration

\[
y\textsubscript{avg} = \frac{\sum_{i=1}^{n} w_i y_{pi}}{\sum_{i=1}^{n} w_i}
\]

where n is the number of times the scene has been re-activated to date, w\textsubscript{i} is the weight assigned to the configuration that resulted from the i\textsuperscript{th} iteration, and y\textsubscript{pi} is the the lux value at point p in that configuration
- the same equation analytically written takes the form:

\[
y\textsubscript{avg} = \frac{w_{og} y_{og} + w_{it1} y_{it1} + \ldots + w_{itn} y_{itn}}{w_{og} + w_{it1} + \ldots + w_{itn}}
\]

where y\textsubscript{og} is the target lux value initially assigned to point p by the designer and w\textsubscript{og} the weight assigned to the designer’s lighting configuration (e.g., w\textsubscript{og} = 50)
- on the occasion where no weights are assigned, all weights default to a value of 1. In such a case, the more often a specific lux value is requested at a particular location the more prominent it becomes since it occurs with increased frequency amongst other observations
- the next time the scene is activated the system passes the TLV\textsubscript{avg} vector as input to the optimization solver, instead of passing the original vector of TLVs (TLV\textsubscript{og}), shifting the scene’s outset to a solution that more closely approximates local occupant preferences in the given space

\[
TLV\textsubscript{s} = TLV\textsubscript{avg}
\]

- whenever the lighting scene is reactivated a new target lux vector replaces the previous lighting solution

There are three metrics that indicate whether the iterative process of collaboration may come to an end. The first one examines whether the range of the received requests has stabilized. The second checks if the absolute difference between each new configuration and the previous one converges to zero. The final metric examines whether requests are progressively minimized, in which case recomputations terminate after requests are reduced to zero. When requests cannot be minimized, the interaction concludes with the stabilization of the request range and the convergence of the absolute difference between the two latest configurations. A converged configuration satisfies all local preferences as closely as possible. The capacity of the system to converge to a solution that fully satisfies lighting requirements across the space depends on the disparity of the received requests. This is discussed in more detail in the following section.

Human-system collaboration is conducted periodically during the operational life of the system. The frequency at which the protocol is employed depends on the received requests. Particularly, the system keeps evaluating the request range even after convergence has been reached checking if the metric remains constant. On the occasion where the request range starts fluctuating, exceeding a threshold value of 50lux, the system reactivates the interaction protocol and resumes updating lighting configurations until convergence is reached again, in accordance with the established metrics. The indicated threshold value is a parameter that can be experimentally optimized in a physical testbed.

### Case Studies

Since this study extends the HLSI framework, the designed protocol is demonstrated in the same virtual conference room where the rest of the interaction protocols were implemented. In that context, eight smart lighting fixtures are installed. Each fixture is composed of digitally programmable LEDs that collectively afford 53 distinct light directionalities. Thus, to sculpt light, 424 intensity multipliers are estimated by the optimization solver for each configuration.

To employ the designed protocol in the conference room two distinct scenarios are implemented, each one incrementally increasing the complexity of the situation the system is asked to address. Both scenarios are initialized with the “highlight the conference table” as the active lighting scene. It is assumed that all adjustments are requested at four particular locations (Figure 3) and that all configurations are equally weighted.

**Figure 3: Locations Receiving Adjustment Requests**

Human figures indicate the locations where adjustments are requested. When people are sitting around the table the customization algorithm will only modify lux on the table. Inversely, the occupied spot across the table (Figure 3, lower right) signifies that a request is received at a place other than the originally highlighted feature. This implies that the physical space might have been updated after the lighting system was deployed; for example, a desk might have been added. Even though the lighting system remains unaware of this feature it will address the request
as best as possible, by employing the customization algorithm for all points but the ones that belong to the table.

**First Scenario – same requests at each location**

The first scenario assumes that the magnitude of any request depends on its location. This demonstrates the case where occupants recognize the same issues with the original lighting solution and are all comfortable around the same lighting levels. It may also refer to a private space (e.g., office) where the same people are occupying the same locations.

Over time, as collaboration takes place, each averaged target lux vector is compared to the respective vector of the previous iteration (Figure 4). Specifically, for each sample point the absolute difference between the current weighted average and the weighted average of the previous iteration is computed. The absolute differences of all sample points are averaged and plotted (Figure 4, blue line). The largest absolute difference is additionally plotted per iteration (Figure 4, red line). According to the resulting graph it takes 77 iterations for both metrics to converge.

![Figure 4: Convergence of the Absolute Difference of the Latest Lighting Configurations](image)

The number of received requests per iteration depends on the occupancy load of the conference room and may vary per iteration. To convey the number of locations at which the solution diverges from the expected illuminance, the number of requests per iteration is translated into a percentage (Figure 5). When lighting configurations are converging toward a solution that effectively addresses all local needs this percentage is progressively minimized and is eventually reduced to zero.

![Figure 5: Requests per Iteration, Smaller Request Range](image)

The rate at which requests are minimized depends on their magnitude. For instance, in this particular scenario where the table is originally lit at 300 lux and the requests range from 250 to 350 lux it takes 77 iterations to reduce the requests to zero. If we were to repeat the experiment with a bigger request range that ranges from 100 to 650 lux, it would take 640 iterations for all requests to be minimized (Figure 6). The minimization rate is also dependent on the assigned weights ($w_{\text{loc}}$). To demonstrate a general implementation of the algorithm, in both scenarios weights default to a value of $w_{\text{loc}}=1$.

![Figure 6: Requests per Iteration, Greater Request Range](image)

The interaction that takes place between the users and the system over time is demonstrated in Figure 7. Specifically, the afforded illuminance around each of the four locations (average lux within the inner sphere) (Figure 7, green line) as well as the preferred illuminance at each spot (Figure 7, blue line) are illustrated over time. Since illuminance requests are constant the lighting solution is progressively modified until all preferences are met. Meeting a preference does not entail explicitly delivering the requested lux value. Instead, it means to adjust the lux level to be within a designated illuminance perception range ($p_i$). The perception range is a parameter that indicates the threshold above and below the preferred level of illuminance in which a user cannot visually distinguish differences in lighting. For the purposes of this demonstration $p_i = 25$ lux. The value of this parameter is denotative and will be optimized once the physical testbed is set up and real-world experiments can be carried out. The vertical red lines in Figure 7 illustrate the point at which the system enters the designated perception range at each location.

![Figure 7: Human-System Interaction Over Time](image)
0 corresponds to the global lighting scene “highlight the conference table” from which collaboration is initiated. In this instance the table is uniformly targeted with 300 lux while the rest of the space with 50 lux. Iteration 1 demonstrates the first customized configuration being averaged with the original one. Given that original and customized configurations are equally weighted as well as the fact that user requests are fluctuating near the original configuration, the difference between iteration 0 and 1 is too small to be visible. Nonetheless, 10 iterations later the customized configuration already weighs 10 times more than the original one and local preferences are more visible. On the other end of the spectrum, the difference after a number of iterations becomes again imperceptible. For instance, diagrams associated with iterations 50 to 100 are visually identical. This observation along with the fact that convergence is numerically reached at iteration 77 reveal two things. Primarily, they visually validate the fact that letting the system recompute lighting configurations past convergence is unnecessary (iterations 78 to 100). They also indicate that collaboration may be concluded even before the solution numerically converges (possibly by iteration 50), given that further changes remain imperceptible. The number of iterations at which this becomes feasible is highly dependent on the threshold level at which illuminance differences may be perceived by humans in a physical space. Therefore, the identification of a more accurate point of termination for the process of collaboration requires real-world experiments.

Second Scenario – different requests at each location
The second scenario conversely assumes that the magnitude of requests at each location varies. This demonstrates the case where occupants with diverse lighting preferences occupy the space (e.g., public space, library). Even so, both metrics converge; this time, approximately after 200 iterations (Figure 10).

Figure 8: Visualization of Select TLV Vectors

Figure 9: Visualization of Select TLV Vectors

Figure 10: Convergence of the Absolute Difference of the Latest Lighting Configurations
With regard to the number of received requests per iteration, Figure 12 illustrates that in this scenario requests cannot be reduced to zero. This means lighting configurations cannot progressively converge to a solution that simultaneously addresses all local needs. This result is expected given that occupant preferences at each location change over time and the converged solution of Figure 10 will not be able to explicitly address them as previously. Instead, every new iteration will be initiated from an average configuration which occupants can later modify as they see fit. Given that user requests cannot be minimized, the system’s primary objective becomes to identify whether the range of requests (maximum-minimum) changes. If it does, the system will keep averaging configurations until the range of requests is stabilized and the absolute difference between the latest solutions converges.

![Figure 12: Requests per Iteration](image)

Figure 13 demonstrates the interaction that takes place between humans and the system over time at each one of the four examined locations. System afforded lux around each location is highlighted in green whereas user preference is highlighted in blue. This figure also captures how the system progressively averages lighting around the fluctuating requests.

![Figure 13: Human-System Interaction Over Time](image)

Select lighting configurations are once again color coded and visualized in the simulated conference room (Figure 13). Iteration 0 corresponds to the original “highlight the conference table” scene from which collaboration is initiated. Iteration 1 demonstrates the weighted average of the original and the first customized configuration. Over 50 iterations later the solution is starting to become stable and 100 iterations later the averaging process has pretty much reached a stable point. Updates beyond iteration 100 remain visually imperceptible. Comparing this result to the iteration number where numerical convergence is approximately reached (iteration 200) brings again to the forefront the importance of collectively accounting for both quantitative and qualitative metrics. As suggested in this implementation, this may be achieved by combining quantifiable metrics with visual convergence thresholds (e.g., illuminance perception range).

**Future Research**

Establishing explicit visual thresholds is a task that requires real-world experiments and, therefore, constitutes the subject of future research. Nonetheless, combining quantitative and qualitative metrics speaks to the importance of combining system intelligence with human intuition and reinforces the importance of...
integrating collaborative implementations in smart building systems. Moreover, concluding collaboration before numerical convergence is reached reduces the activity of the system and, thus, the total amount of energy that is consumed to light the room, without compromising the visual quality of the space and the wellbeing of the occupants. Thus, conducting a future study to compute the energy savings that could result from this implementation seems critical for understanding how the integration of qualitative metrics impacts intelligent building system performance.

Conclusions
This paper extends the Human – Lighting System Interaction Framework by introducing a protocol for designing collaborative interactions between users and intelligent lighting systems. To that end, the assumptions entailed in establishing closed loop communication between the users and the system as well as the assumptions embedded in computationally encoding the physical space and the lighting system are addressed. The paper demonstrates how the developed interaction protocol optimizes global and customizes local lighting configurations in real-time as well as how original lighting scenes are re-designed over time with user participation. Two scenarios of collaborative interactions are implemented to demonstrate the protocol’s capacity to address situations of varying complexity. Evaluation metrics for assessing collaboration over time are proposed and the resulting lighting configurations are mapped on the virtual space to visually illustrate modifications of lighting configurations over time.

Overall, human-system collaboration is a protocol that integrates user participation in lighting design. Users inform the system about lighting requirements that could not have been recognized by the system. The system delivers, in turn, lighting configurations the designer and users could not have composed by themselves. Therefore, the resulting configurations exceed the ones humans or artificially intelligent lighting systems could individually compose. Since collaboration makes human insight available to the system, another level of awareness is introduced which reinforces the accuracy of the solution and collectively improves system autonomy and performance over time.

Transitioning from designing static, activity-based lighting configurations to designing dynamic behaviors that cater to occupant needs in real-time, converge to global lighting solutions over time, and transcend simulation limitations necessitates advanced forms of human-system interaction to be established. Collaboration integrates participatory design practices in lighting design and allows users to experientially evaluate and qualitatively improve lighting as a collective.

In the general context of the built environment, collaboration aims to advance the way in which humans and embedded building systems interact. The end goal is to combine the strengths of humans (e.g., insight) with the affordances of AI systems (e.g., memory, processing power) to push the boundaries of human-system intelligence and accelerate innovation in the field of smart building design. Fundamentally, by adopting collaborative design practices and fostering collaborative interactions between humans and artificially intelligent systems, designers move past the practice of designing behaviors that solely resolve quantifiable matters; instead, they simultaneously engage with addressing qualitative pursuits of increased complexity.

Acknowledgement
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
Turkle, Sherry. (2009). Simulation and Its Discontents. 10.7551/mitpress/8200.001.0001