Computational Modeling and Optimization Applied to Controlled Environment Agriculture Lighting Systems

Honors Thesis
Presented to the College of Agriculture and Life Sciences, Plant Science,
of Cornell University
in Partial Fulfillment of the Requirements for the
Research Honors Program

By:
Daniel Santander
May, 2020

Under the Guidance of:
Dr. Neil Mattson, School of Integrative Plant Science, Cornell University
Dr. Kale Harbick, Application Technology Research, US Department of Agriculture

Table of Contents

Abstract

Supplemental lighting is integral to year-round production of greenhouse crops; however, the location of lights within the greenhouse and its effects on lighting uniformity in the growing space is often not considered. This research was conducted to assist Controlled Environment Agriculture (CEA) producers and researchers in identifying the optimum lighting layout for improved lighting uniformity. The methodology outlines the development of an algorithm for modelling supplemental lighting, based on standardized goniophotometric data, and optimizing the location of lighting fixtures within the CEA environment. This resulted in the production of a software package in the Python Programming Language that could model and optimize lighting uniformity for unique CEA environments based on their physical dimensions and specified lighting fixture. Through the implementation of this novel software, the lighting uniformity for hypothetical CEA environments with a small number of supplemental lighting fixtures were optimized.

Introduction

Supplemental lighting is a fundamental aspect that enables year-round production in controlled environment agriculture facilities; such as vertical farms and greenhouses at northern latitudes (Dorais & Gosselin, 2002). In a standard facility, supplemental lighting is frequently installed at arbitrary heights and configurations. This may provide adequate light for plants, but it does not create a uniform lighting environment. A non-uniform lighting environment can cause variable yields and plant quality; as well as, contribute to the creation of "hotspots", or areas of over-exposure, which in some cases can lead to plant stress or physiological disorders, for example, tipburn of head lettuce (Frantz et al., 2004). Ferentinos (2005) suggested that increasing lighting uniformity improves consistency of yield, as well as, that increasing uniformity lowers greenhouse energy requirements.

Manually determining the ideal location for lights (the method most frequently used by researchers and small greenhouse operations) is a tedious process, that involves the placement of lights, use of a quantum sensor to map light levels at plant canopy heights, and reconfiguration to reach an optimum layout. A less exhaustive alternative would be to model light distribution based on standardized light output files composed of angular distribution of luminous intensity. By testing different configuration of lighting systems using computer models, a local optimum for uniform illuminance may be achieved across the growing area of a greenhouse. This computer model begins with the collection of goniophotometric data obtained from any lighting fixture. Goniophotometers measure the variation of luminous intensity across various angles at a fixed distance from a luminaire. By using goniophotometric data, 3-Dimensional models for the illuminance of these lighting systems can be developed. The advantage to creating a computer model and optimization algorithm that processed goniophotometric data, is that a computer can then perform the complex and computationally intensive calculations required for optimization. Ideally, this would allow for optimized lighting set-ups for any lighting fixture, any crop, and any greenhouse to be developed automatically in the computer program. The objective of this research was to apply an original methodology to computer modelling of greenhouse supplemental light distribution, and to develop a software for the spatial optimization of supplemental lighting fixtures to improve light exposure, or illuminance, on the growing area of the plants.

Methods

In this study, two supplemental lighting sources were considered: a high-pressure sodium (HPS fixture) Gavita Pro 6/750e Flex3 US DE and a light emitting diode fixture Philips GreenPower toplighting DR/B - Low Blue. The fixtures were previously chosen by the Cornell Controlled Environment Agriculture (CEA) group based on preliminary work where the Philips LED was found to be about 40% more energy efficient than the Gavita HPS fixture. The goniophotometric data were initially recorded by the product testing company, Intertek, for these two lighting systems. Goniophotometric data are displayed in a standardized file developed by the Illuminating Engineering Society, known as IES files (ANSI/IESNA Standard File Format, 2002).The first step was to parse the files and collect the goniophotometric data points in a data matrix.

The software to parse these files was developed for this project specifically to handle IES files written according to ANSI/IESNA LM-63-2002 data systems (ANSI/IESNA Standard File Format, 2002). The first step of the parsing process began with the conversion of the file extension of the IES file from '_____.ies' to a text file '_____.txt' using UNIX in the terminal. The text file could then be directly used in the Python programming language by creating a 2-Dimensional list where each line of the text file was converted to an individual list. The file was then scanned to return relevant information; namely, the quantity and values of the vertical, θ, and horizontal, ϕ, angles measured in the goniophotometric data. These numbers defined the dimensions of the goniophotometric data, which was then transferred into a data-frame from the Python programming language's module PANDAS, which is a data structure for statistical computing (McKinney, 2010). Each value in the data-frame represented the luminous intensity in candela at a particular vertical & horizontal angle at a fixed radius from the light source.

Next, interpolation of the goniophotometric data was used to define the luminous intensity of points that fall between measured data points. This allowed for any point along the surface of the goniophotometric sphere to be approximated based on the data in the IES file. A method for interpolation based on an Inverse-Distance Weighting formula (Robeson, 1997) was applied to interpolate candela values for points that fall between the data points of the goniophotometric data. First the great-circle distance was calculated (Equation 1), this value was then weighted using an inverse weighting function (Equation 2). Finally the value at the point of interest was calculated as a weighted sum of values of the nearby data points (Equation 3). The procedure and variables are as follows:

Spatial Interpolation Variables	
ϕ	Latitudes
λ	Longitudes
n_j	Number of Control (Data) Points
i	Unsampled location
j	Control (Data) Point
z	Candela value

Equation 1. $\boldsymbol{d_{ij} = R\cos^{-1}[\sin\phi_i \sin\phi_j + \cos\phi_i \cos\phi_j \cos(\lambda_i - \lambda_j)]}$

Equation 2. $\boldsymbol{w_{ij} = d_{ij}^{-1}}$

Equation 3. $\boldsymbol{\hat{z}_j = \sum_{i=1}^{n_j} w_{ij} z_i \,/\, \sum_{i=1}^{n_j} w_{ij}}$

Next it was desirable to consider points that extend away from the surface of the sphere. To make use of the vertical and horizontal angles of the goniophotometric data, a method to interconvert between Cartesian-coordinate systems, 3-Dimensional system defined by x, y, and z coordinates, and spherical coordinates was implemented. This allowed for a point of interest to be defined by both its x-, y-, & z- coordinates in a large environment, as well as, by its relationship to a light fixture through θ (vertical angle), ϕ (horizontal angle), and r (distance to light fixture). The relationship between cartesian and spherical coordinates is outlined below (Figure 1.), and the algorithm for their interconversion is described.

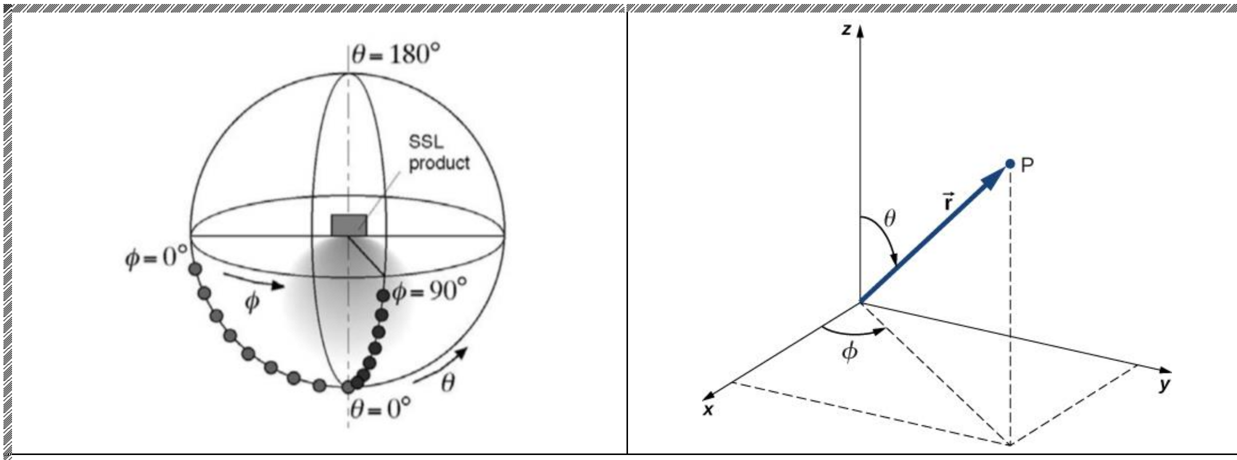

Figure 1. Relationship between angles reported in goniophotometric readings that utilize a spherical coordinate system and those of a xyz-Cartesian coordinate system. (ANSI/IESNA Standard File Format, 2002)

Provided with a θ, ϕ, and r the x, y, and z-coordinates may be calculated through the following algorithm. The angular values from the IES file were first converted from steradians to angular degrees, by multiplying θ and ϕ by 180π. The x-,y-, and z-coordinate was then calculated using equations 4, 5, and 6 respectively. An original software to follow this algorithm was developed for this project.

Equation 4. $$\boldsymbol{x = r * \sin\theta * \cos\phi}$$

Equation 5. $$\boldsymbol{y = r * \sin\theta * \sin\phi}$$

Equation 6. $$\boldsymbol{z = r * \cos\theta}$$

With a method to find the position of any point in relation to the light and the luminous intensity along the surface of the goniophotometric sphere, a method was developed to find the illuminance, or luminous flux density, of a given area, as it varies with distance from the light source. Goniophotometric readings report the value of luminous intensity in candela (lumens per steradian), though the value of interest was the illuminance provided by a point source. A candela value is a measure of luminous intensity, which is a measure of the luminous flux per unit solid angle which is photometrically weighted for human vision (Choudhury, 2016). Illuminance is a measure of the luminous flux density incident on a surface in (lux = lumens/m^2). A software to convert from luminous intensity, I, to illuminance, E, was developed based on equation 7. Where d represents the distance to the light source and θ, the angle to the light source.

Equation 7. $$\boldsymbol{E = \frac{I}{d^2} * \cos\theta}$$

Ultimately, this method altered the units of the data from candela ($\frac{lumen}{steradian}$), to lux ($\frac{lumen}{m^2}$). Additionally this allowed for the lux to be calculated for any point, including those outside of the surface of the sphere. This left one final step in developing the computer model specific for controlled environment agriculture, conversion to Photosynthetic Photon Flux Density (PPFD).

Lux is a photometric unit, it is weighted to the sensitivity of the human eye through the photopic luminosity function (Choudhury, 2016); whereas, plants utilize quanta of energy within the wavelength of 400nm to 700nm (Choudhury, 2016). After finding the photometric illuminance it was necessary to convert to the quantum measurement of

PPFD. An original software was developed to convert between these measurements specifically based on the following algorithm.

After finding the luminous flux density (lux) the units from photometric light measurements (lux) were converted to quantum units ($\frac{\mu moles}{m^2 * s}$). Photometric measurements are produced through the measurement of visible light in units that are weighted according to the sensitivity of the human eye. The weighting of the units to the sensitivity of the human eye is produced through the photopic vision luminosity function. Provided with the luminosity function ($V(\lambda), dimensionless$) and the spectral radiant flux ($\Phi_{e,\lambda}(\lambda)$ $units\ of$ ($\frac{W}{nm}$)), the photometric luminance flux for a spectrum of light in lumens was determined by Equation 8.

Equation 8:

$$\boldsymbol{\Phi_V = 683.002\ \frac{lumens}{W} * \int_{\lambda=0}^{\lambda=\infty} V(\lambda)\Phi_{e,\lambda}(\lambda)d\lambda}$$

By solving for the amount of Photosynthetically Active Radiation (PAR) in the same spectrum of light a conversion factor was established between photometric lumens and Photosynthetic Photon Flux. Photosynthetically Active Radiation is electromagnetic radiation within the wavelengths of 400 to 700 nm. This value is reported as the Photosynthetic Photon Flux Density, with the units of $\frac{\mu moles\ of\ photons}{meters^2 * second}$.

In order to find the photosynthetic photon flux of a light source incident on a point, the spectral contribution of photons at each wavelength must be considered. This may be derived beginning with the Einstein-Planck Relation (Eq. 9).

Equation 9. $$\boldsymbol{E = h\,\nu}$$

Where E is the value of a quantum of energy known as a photon (Joules), h is Planck's constant (6.62607004 * 10-34 $\frac{meters^2 \times kg}{second}$), and v is the frequency of oscillation (second-1).

Equation 10. $$\boldsymbol{\lambda \nu = c}$$

To this the equation for the speed of light (Equation 10) may be incorporated to result in Equation 11, which directly relates the energy of a photon to wavelength, to return the Photon Energy. In equation 10 and 11, λ has the units of nanometers and c has the value of $2.99792 * 10^8 \frac{meters}{second}$. Photon Energy, E$\lambda$, has the units of $\frac{J}{photon}$.

Equation 11.

$$\boldsymbol{E_\lambda = \frac{h * c}{10^{-9} * \lambda}}$$

After finding the Photon Energy for each wavelength within PAR, photon flux was found by dividing the spectral radiant flux ($\Phi_{e,\lambda}(\lambda), units\ of\ (\frac{W}{nm})$) by the photon energy, Eλ, ($\frac{J}{photon}$). By summing the photon flux at each wavelength within PAR, the flux of photosynthetic photons ($\frac{photons}{s}$)for the whole spectrum was determined (Equation 12).

Equation 12.

$$\boldsymbol{Photon\ Flux = \sum_{\lambda=400}^{\lambda=700} \frac{\Phi_{e,\lambda}(\lambda)}{E_\lambda}}$$

By adapting the photon flux to represent micromoles (Eq. 13) the final number for the Photosynthetic Photon Flux with units of $\frac{\mu moles\ of\ photons}{second}$ was determined.

Equation 13. $$\boldsymbol{PPF = \frac{photon\ flux}{6.022 * 10^{17}}}$$

Finally, a conversion factor between PPFD and photometric lumens may be established by dividing the Photosynthetic Photon Flux Density(Eq. 13) by the photometric luminous flux (Eq. 8). This provides a conversion factor with the units of $(\frac{\mu moles}{lumens * s})$. Provided with the lux $(\frac{lumens}{m^2})$ from the goniophotometric data, the irradiance was multiplied by the conversion factor to find the Photosynthetic Photon Flux Density in units of $(\frac{\mu moles\ of\ photons}{meters^2 * second})$. This conversion is unique for every lighting system and must be originally calculated based on the spectral distribution of the light. With this step completed it was now possible to use the model to find the PPFD at any point within an isolated lighting environment, specific to the light measured in the goniophotometric data.

Next a 3-Dimensional simulation of a greenhouse was created. Provided with inputs of lighting position and type, length and width of greenhouse, and the resolution of the calculation, the model created a surface of calculation points on the "greenhouse" floor. The greater the resolution, the more points present within the calculation bed. The PPFD was calculated at each of these points, which provided data for the overall illuminance on the floor of the simulated greenhouse. After defining an ideal range for PPFD, a metric was used to determine uniformity of light across the greenhouse floor which was the percentage of calculation points within the ideal range within the define area. Increased density of calculation points allows for a more accurate calculation of the uniformity of the light, but it requires the PPFD to be calculated at more points; thereby, increasing the number of calculations and decreasing the performance speed.

The final output of interest of the computer model is uniformity. This is calculated as the percent of calculation points within an optimal PPFD range, defined by a minimum and maximum value of ($\frac{\mu mols}{m^2 \cdot sec}$) specific for a crop.

The ultimate goal of developing the lighting computer model was to be able to conceive and test different lighting positions, in search of the most 'ideal' physical configuration of lights. In terms of computation, this is a challenging task to complete because it is a NP-hard problem. That is a nondeterministic polynomial time hard, meaning the algorithm to solve this problem can produce different results on different runs and the computation time increases in polynomial time as the number of computations increases. Simply stated, this puts this type of optimization problem in the most difficult groups of problems to solve. In order to reduce the number of computational steps, it was necessary to restrict the possible location of lights based on preliminary binding constraints, such as the height, width, and length of the greenhouse as well as the minimum and maximum Photosynthetic Photon Flux Density values. Additionally, a resolution for the computation is defined by the user. This resolution is reported as a percentage and affects the potential locations a light can be placed as well as the density of calculation points on the calculation bed. It is important to note that percent resolution is an abstract metric that mainly affects computation time; it is not the percent of the dimensions of the greenhouse. For instance, a resolution of 10% on a greenhouse with *length, width, and height = 10 m* will result in the program considering locations at 1 m intervals in the x, y, and z direction; whereas, a program of resolution 20% will consider locations at 0.5 m intervals. This example demonstrates that an increasing percent resolution results in denser calculation points.

Having been provided with a height of the greenhouse and a range of acceptable PPFD values the potential heights which a light can be hung are then restricted, because hanging a light too low will result in too high of a PPFD value and hanging at a height too high will result in an insufficient PPFD value. Defining an acceptable range of heights was the first step in the optimization. This range is further restricted by the resolution so that only certain positions within the range will be considered in the optimization.

In order to minimize computation time further an original system of data caching was developed to minimize the number of redundant calculations. This data cache was developed in the Python programming language and it stored the accompanying illuminance at all of the calculation points, produced by an individual light at every possible location. Because the optimization code will test every possible combination of lighting positions, there will be some combinations where a single light is in the same position in two different, unique lighting combinations. In these instances it is faster to simply reference the location of the illuminance data produced by the light at the redundant point, than it is to recompute the effects of the light at the same position. This system of data caching reduced the number of computational steps significantly. The data points that are stored within the cache are chosen with discretion, as preliminary data analysis provides a range of x, y, and z positions for the light that produce desirable results.

After designing a data cache system, the challenge of improving lighting uniformity was reduced to a simple optimization problem, which takes the following form:

Decision Variable:

$$x_i = \begin{cases} 1, Light, 'x', is\ present\ at\ location, 'i' \\ 0, Light, 'x', is\ not\ present\ at\ location\ 'i' \end{cases}$$

**Let i represent a 3-variable cartesian (x,y,z) coordinate*

Objective Function

Maximize Uniformity:

$$U = \sum_{n=0}^{n=N} u\ (x_i)$$

**Where N = total number of lighting systems*

Binding Constraints:

Physical Constraints:

$$0 \leq x_{i(x)} \leq greenhouse\ width$$
$$0 \leq x_{i(y)} \leq greenhouse\ length$$
$$0 \leq x_{i(z)} \leq greenhouse\ height$$

Hyper-illumination:

$when\ light\ is\ at\ position\ x_i,$

$$Intensity(ControlPoint_i)\ \leq maxPPFD$$

Hypo-illumination:

$when\ light\ is\ at\ position\ x_i,$

$$Intensity(ControlPoint_i) \geq minPPFD$$

An original algorithm was developed to solve this combinatorial optimization problem in the Python programming language. The optimization code tests all possible combinations of lights constrained by the binding constraints. The combination of lights that produces the greatest uniformity is decided to be the optimum based on the maximum and minimum PPFD value for the crop, type and number of lighting system, and the dimensions of the greenhouse.

Finally, to ensure that the number of lights being used is ideal, the optimization code is called recursively to consider the instances of: N + 1 and N – 1, where N = number of lights. This code examines whether adding or removing a light can increase the overall uniformity of the lighting system. If adding or subtracting a light produces a more optimal uniformity, the recursive optimization code is recalled to now test whether adding or subtracting another light will improve the uniformity. This recursive aspect to the optimization ensures that not just the location of lights is optimum, but also the number of lights used is optimized.

Results

The primary results of this research is the construction of a functional software to model the effects of supplemental lighting in greenhouses based on goniophotometric data. The starting goniophotometric data may be visualized in Figure 2, where each blue dot represents a data collection location for which there is a known luminous intensity. Each of these data points is equidistant, $r = 1$, from the light source which is represented by the orange dot.

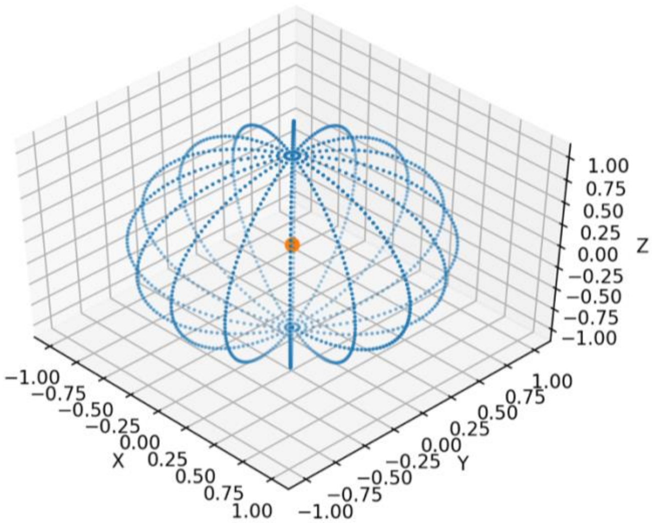

Figure 2. *3-Dimensional Representation of Goniophotometric Data Points. Each data point is equidistant from the light source, meaning the distant from the light source for each point is r = 1.*

Based on the methodology described these data points went through a computational pipeline to be able to define the illuminance of any point in a greenhouse. Beginning with interpolation between goniophotometric data points (Figure. 3).

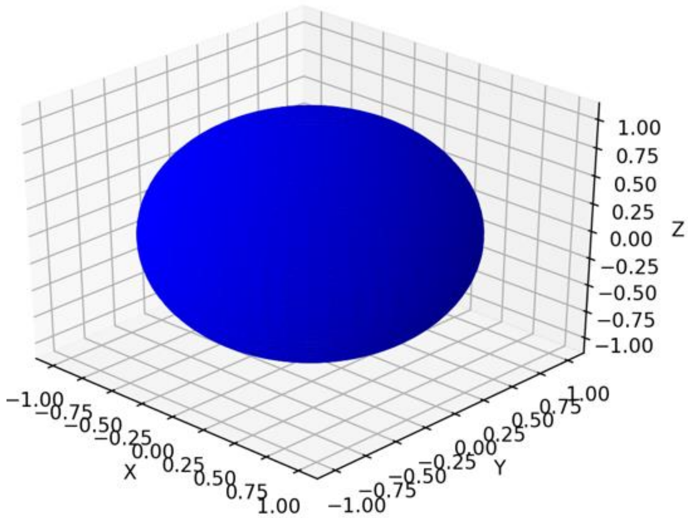

Figure 3. *Representation of Known Luminous Intensity after Interpolation. Every equidistant point of distance, r = 1, may be approximated through the interpolation algorithm.*

After interpolation, luminous intensity was converted to illuminance which allows for the illuminance of any point within an environment to be calculated. With the application of this software to controlled environment agriculture in mind, the points of interest for finding illuminance were defined by a bed of calculation points. This bed of calculation points aimed to represent the growing area of plants in a greenhouse. The quality of the output and the speed of the software is highly affected by the density of the calculation points. The user defines the desired resolution of computation, which in turn affects the density of calculation points, which can be visualized in Figure 4.

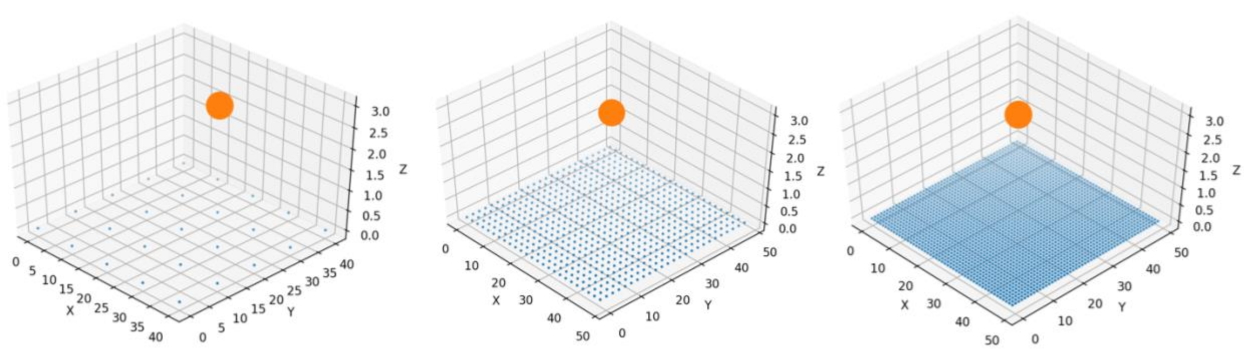

Figure 4. *Visualization of how increasing resolution, increases the density of calculation points (Blue dots) on the bed of the greenhouse environment. X, Y, Z – coordinates represent the position of a point along the width, length, and height of a greenhouse respectively. The graph on the left has a resolution of i=10, the middle i = 2, and the right i = 1, this denotes the density of calculation points / width or length of the greenhouse.*

The computer model of lighting effects was used to create heatmaps of illuminance in the greenhouse. The optimization code considers the effects of the physical location of supplemental lighting on the uniformity of illuminance across the calculation points. As an example, the effects of the height of a supplement light may be visualized in Figure 5. The uniformity varies based on the model of lighting fixtures and their position, as well as the lighting requirements for the crop growing in the greenhouse.

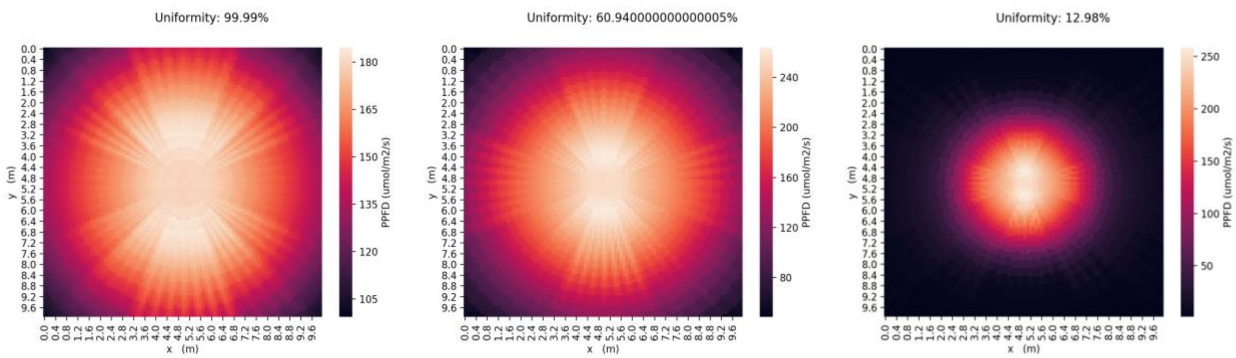

Figure 5*. Visualization of the effects of z-position of light on the total uniformity of the greenhouse bed. In (a. left) the position of the light is [5m, 5m, 10m], by lowering the light to [5m, 5m, 5m] in (b. middle) the uniformity decreases as a hotspot develops directly under the light. Finally uniformity drops as the light is lowered to [5m, 5m, 2m] as the hotspot increases in illuminance and the corners of the greenhouse lack adequate illuminance.*

Ultimately the effects of the lights' cartesian coordinates (position along the width, length, and height of a greeenhouse) affects the uniformity of light; as well as, the interaction of lights in creating areas of over exposure. As a sample output the light emitting diode fixture Philips GreenPower toplighting DR/B - Low Blue was optimized using the software. The software considered a 40m long by 10 m wide greenhouse that included 6 fixtures, the optimum illuminance range was broad ranging from 100 $(\frac{\mu mols}{m^2 \cdot sec})$ to 500 $(\frac{\mu mols}{m^2 \cdot sec})$. The difference between a conventional lighting configuration, where lights are placed in rows above growing areas, is compared to an optimized configuration in Figure 6.

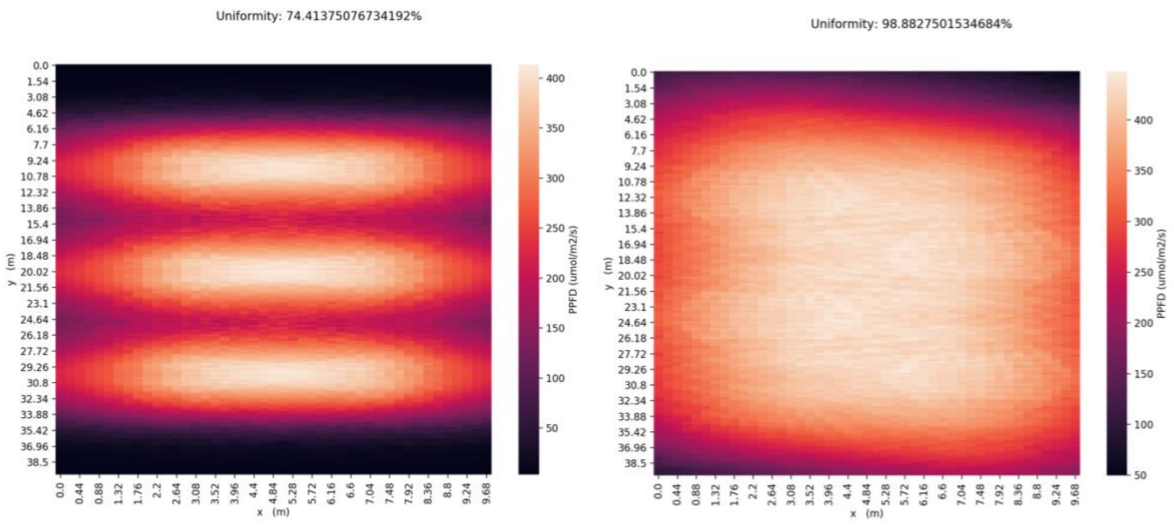

Figure 6*. Comparison of Conventional Lighting Configuration versus an Optimized Lighting Configuration. It is worth noting that optimization improved the uniformity of illuminance across 20% more of the growing area than a traditional lighting plan. This is visible along the edge of the growing area in the figure.*

Discussion & Conclusions

There has been a significant amount of research on how to optimize the uniformity of lights in a greenhouse as well as in human-occupied spaces (i.e. offices, houses, etc.) (Ferentinos, 2005)(Both et. al, 2002). The methodology outlined in this paper is unique in the development of a computer model rooted in photo-physical equations; whereas, previous research mapped lighting uniformity using quantum sensors (Both et. al, 2002). Previous research has investigated sets of optimum layouts, rather than all potential configurations of lights. This is both a benefit and a drawback to this software. While it is possible to produce high quality results that consider every possible location a light can be placed in a greenhouse, this greatly affects the computation time. This increased computation time, makes it challenging to produce results for large greenhouses with a large number of supplemental lights.

The objective of this research was to develop a methodology to standardize lighting in a greenhouse to minimize the effects of non-uniform light exposure; such as, inconsistent yield, biased lighting treatments in research, and the frivolous use of energy related to over-or-under lighting portions of the growing area. The software was developed in the Python programming language and it returns the optimum locations for supplementary lights in a greenhouse, based on the type of light, number of lights, size of the greenhouse, and the lighting requirements for a particular crop. In order to be applied to commercial or academic controlled environment agriculture this software can be further developed or integrated into a computational pipeline.

Future proposed work with the model is to validate the predicted illuminance from the model, based on experimentally collected light-maps for the same lights. This will help to adjust the accuracy of the results produced in the computer model. Additionally, creating a repository of IES files for different lighting systems will increase the applicability of the software. Finally, incorporating new methods for optimization will improve the performance of the software for large greenhouses with a large number of lighting fixtures. An obvious way to improve the dimensionality reduction in these circumstances is to incorporate machine learning. This could be easily incorporated using the module Python scikit-learn of the Python programming language, which is compatible with the software produced in this research.

Literature Cited

Both, A.J., L.D. Albright, and D.E. Ciolkosz. 2002. Evaluation of Light Uniformity underneath Supplemental Lighting Systems [Electronic Resource]. *Acta Horticulturae*, no. 580: 183–90. doi: 10.17660/ActaHortic.2002.580.23

Choudhury, Asim Kumar Roy. 2014. "Characteristics of Light Sources." *Principles of Colour and Appearance Measurement.* Woodhead Publishing. 1 - 52

Ciolkosz, D.E., A.J. Both, and L.D. Albright. 2001. Selection And Placement Of Greenhouse Luminaires For Uniformity. *Applied Engineering in Agriculture* 17: 875-882.

Dorais, M. and A. Gosselin. 2002. Physiological Response of Greenhouse Vegetable Crops to Supplemental Lighting. *Acta Horticulturae* 580: 59-67.

Ferentinos, K., and L. Albright. 2005. Optimal design of plant lighting system by genetic algorithms. *Engineering Applications of Artificial Intelligence* 18: 473–484.

Frantz, Jonathan M., et al. 2004. Exploring the Limits of Crop Productivity: Beyond the Limits of Tipburn in Lettuce. *Journal of the American Society for Horticultural Science* 129: no. 3 331-338.

Hunter, John D.. 2007. MatPlotLib: A 2D Graphics Environment. *Computing in Science & Engineering* 9: 90 - 95

McKinney, Wes. 2010. Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, 51-56

Robeson, S.M.. 1997. Spherical Methods for Spatial Interpolation: Review and Evaluation. *Cartography and Geographic Information Systems* 24: 3–20.

Waskom, Michael, Olga Botvinnik, Drew O'Kane, Paul Hobson, Saulius Lukauskas, David C Gemperline, … Adel Qalieh. September 3, 2017. mwaskom/seaborn: v0.8.1 Zenodo.

< http://doi.org/10.5281/zenodo.883859 >

2002. ANSI/IESNA Standard File Format for the Electronic Transfer of Photometric Data and Related Information. *Illuminating Engineering Society*

2008. Electrical and Photometric Measurements of Solid-State Lighting Products. *Illuminating Engineering Society.*