Energy efficiency optimization algorithm for roadway illumination using ARM7TDMI architecture

Rafael B. de Oliveira[1,*], Fausto B. Líbano[1,**]

[1] *Faculdade de Tecnologia Senai Porto Alegre, LEEQEE (Energy Efficiency and Quality Energy Laboratory), Porto Alegre, Brazil*
** Rafael Oliveira. Tel: +55 51 98898610, E-mail: rafael.oliveira@senairs.org.br*
*** Fausto Bastos Libano. Tel: +55 51 91867363, E-mail: fausto.libano@senairs.org.br*

Abstract: This paper presents an algorithm developed in C language that aims to help roadway illumination designers to create an illumination system that meets the standard's limits with minimum electrical energy consuming. As a secondary function, the program allows the user to get approximate luminance and illuminance values for a specific system without field measuring. The algorithm was created to fit into a hardware prototype based in ARM7TDMI architecture, with no need for complicated and heavy software running in PC machines. The main system variables regarded by the optimization function are pole height (H) and pole spacing (S), putting the luminaries as far from each other as possible, in order to use the minimum power per km. Through calculation of the system's luminance, the algorithm starts with S and H in their maximum values, decreasing every loop, subtracting the results from the standard limit (NBR5101-Brazil, CIE 118, EN 13201 or other standard loaded into the program) seeking for zero. Once the zero is found, the H and S values are put on a LCD or USB port, as algorithm results.

Keywords: *Roadway illumination algorithm, ARM application, Illumination energy efficiency*

Nomenclature

L	*roadway average luminance*	cd/m^2
E	*roadway average illuminance*	*lux*
H	*pole height*	*m*
S	*pole spacing*	*m*
I	*luminous intensity*	*cd*
Ir	*relative luminous intensity*	*cd/klm*
φ	*luminary luminous flux*	*lm*
ψ	*Luminary azimuth angle*	*º*
θ	*Vertical luminary angle*	*º*
β	*Horizontal observer angle*	*º*
P_a	*active power*	*W*
W	*roadway wideness*	*m*
U_o	*overall uniformity*	
n	*lanes number*	
P_r	*distance relative power*	*kW/km*

1. Introduction

One of the main problems in developing countries regarding electrical energy waste is the roadway illumination design, as the majority is over or under dimensioned, using old technology luminaries and with pole height and spacing in such values that the standard's limits are rarely achieve, which increase car accidents and criminality rate and decreasing the system's efficiency [1,2].

In order to develop a new roadway illumination system or evaluate an existing one, simulation software (Dialux, Lumisoft, Calculux) are used to calculate the main parameters required by standards such as NBR 5101, CIE 115, CIE 180, EN 13201 and others, running in PC platforms.

Another way to evaluate an existing illumination system is by field measuring, which implies in marking the grid on the ground level and taking an illuminance or luminance measure for each grid point, demanding time and a roadway free of traffic [2-4].

Therefore, facing these problems, the hypotheses of a small, low cost, device which could run an algorithm to simulated the main parameters needed to evaluate a roadway illumination system (in case of an existing system) or calculate how far away the poles could be spaced in order to consume less power (in case of a new system), was tested. It is important for the algorithm to be autonomous, that is, no computer aid.

2. Methodology

The main result variable for the algorithm implemented is P_r which is directly connected with energy consuming, therefore, it has to be as low as possible and still allows lighting parameters to meet the chosen standard's limits. For that to happen, the space between poles (S) is loaded with 50m and decreased gradually until the required value is reached. The same is done to the luminary height (H), as presented in Fig. 1.

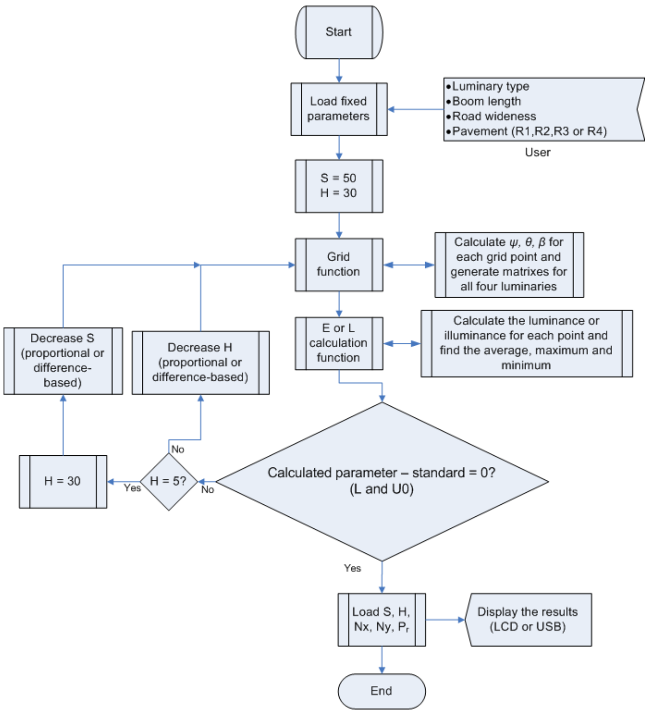

Fig.1 Block diagram of the optimization algorithm.

2.1. Hardware

The first thing to choose in the hardware design is the microcontroller where the algorithm will be running. The ARM7TDMI architecture, more precisely the Analog Device's microcontroller ADUC7026, was chosen by its processing capability of 32bits, low clock frequency (32.768kHz) with high internal speed (41MHz), the I/O pin quantity (ADCs, DACs, GPIOs) and the flash memory space of 62kB and the long multiplication and thumb mode support, allowing the process to run much faster than 16bits architecture or even standard ARM7 devices.

The hardware must have a display capable of reporting to the user of the algorithm variable results, such as: E, L, Emin, Emax, Lmin, Lmax, U_o, H, S, number of grid columns, number of grid rows and the active power consumed by kilometer (P_r).

2.2. Virtual grid creation

To calculate the system main parameters, a grid must be created on the roadway, between two luminaries with interference of, at least, one luminary after and one later, with rows spacing 1m maximum from each other and columns spacing 5m maximum [2,3], as shown in Fig. 2, where S_x is the space between grid points on a line parallel to the curb line and S_y is the space between grid points on a line orthogonal to the curb line. It was stated the symbol N_x for the total number of rows (x axis) and N_y for the total number of columns (y axis).

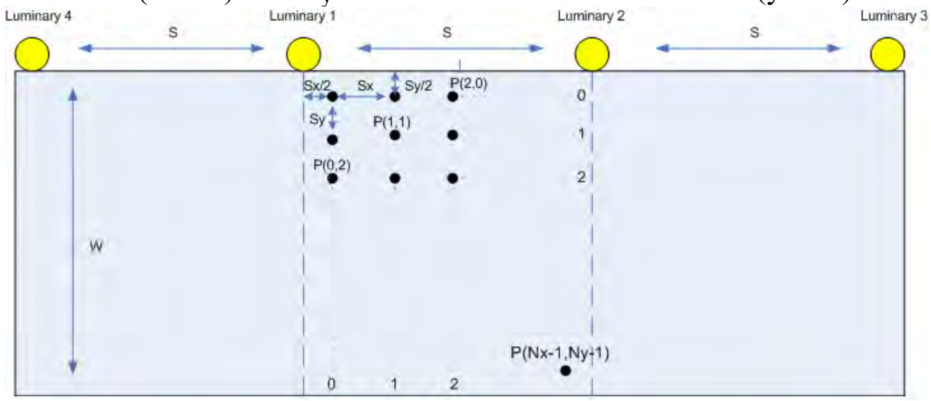

Fig.2 Creation of the calculation grid.

In order to keep the grid centralized on the space between two luminaries, the first point must be located at $S_x/2$ from the x axis and $S_y/2$ from the y axis. S_x and S_y are set by the user from the beginning and N_x and N_y are calculated dividing space between luminaries (S) by S_x and road wideness (W) by S_y.

After finding the rounded values of N_x and N_y, S_x and S_y have to be recalculated.

Each calculation point has two coordinates (x,y) that correspond with its place in relation to the grid. The real distance for each point in relation to the system's origin (Luminary 1) is calculated by Eq. °(1).

$$P(x,y) = P_0\left(\frac{S_X}{2} + x * S_X, \frac{S_Y}{2} + y * S_Y\right) \tag{1}$$

where P_0 is the coordinates of P in relation to the system's origin.

E.g. a point located at P(2,3) with a S of 35m and a W of 9m has an N_x equal to 8 and a N_y equal to 9 (using initial S_x = 5 and S_y = 1), therefore the real values of S_x and Sy are 4.375m (S/N_x) and 1m (W/N_y), respectively, and its coordinates in relation to origin, calculating from Eq. °1, are represented by P_0(10.938m,3.5m). These coordinates are used to calculate the system's main angles.

On this first part, the algorithm creates eight 10x10 matrixes based always on the same grid: one matrix for azimuth angle (ψ) and one matrix for inclination angle (θ) for each luminary in relation to every grid point. Later, the observer matrixes will be created as well, for the observer angle (β). All angles used to execute the main calculations (L and U_o) are presented in Fig. 3.

The γ is the angle between the roadway horizontal plane and the observer's eye, used to find the reduced luminance coefficient in the r-tables [2,4-6].

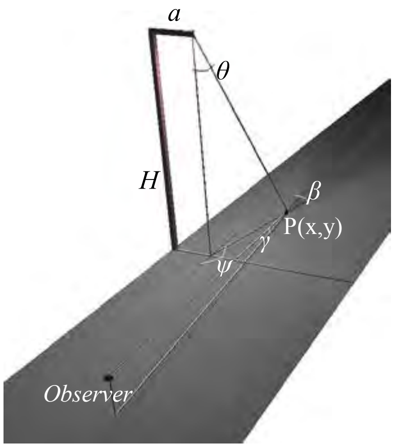

Fig.3 Main angles used to calculate E and L.

The angles represented in Fig. 3 can be calculated using straight trigonometry or algebra and are very important for the calculus, as the luminous intensity tables and r-tables are based on them. For study sake, both methods were used on this work, as the following description.

2.2.1. Azimuth angle (ψ)

The azimuth is the angle between the luminary plane and the calculation point on a horizontal plane (road plane) and is used together with the vertical luminary angle (θ) to find the luminous intensity module in the calculation point direction [2,6].

In this work, ψ was calculated based on the triangle formed by the luminary position and the calculation point position on the road plane, as presented in Fig. 4, where *a* is the boom length, X_0 is the real coordinate (in relation to the origin) of the calculation point on the X axis and Y_0 is the real coordinate of the calculation point on the Y axis.

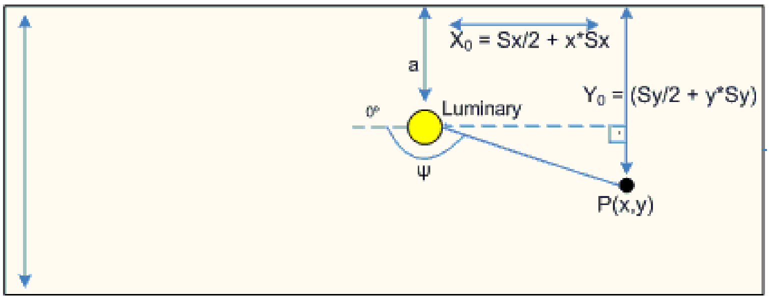

Fig.4 Triangle formed by the luminary position and the calculation point position.

The angle must be calculated in relation to each luminary (four luminaries, as previously stated) in two cases: for $Y_0 < a$ and for $Y_0 \geq a$.

For the first case, ψ can be calculated for the four luminaries using Eq. °(2-5).

$$\Psi_{L1} = 180 + \tan^{-1}\left(\frac{|Y_0 - a|}{X_0}\right) \tag{2}$$

$$\Psi_{L2} = 360 - \tan^{-1}\left(\frac{|Y_0 - a|}{S - X_0}\right) \tag{3}$$

$$\Psi_{L3} = 360 - \tan^{-1}\left(\frac{|Y_0 - a|}{(S - X_0) + S}\right) \tag{4}$$

$$\Psi_{L4} = 180 + \tan^{-1}\left(\frac{|Y_0 - a|}{X_0 + S}\right) \tag{5}$$

In the second case, ψ can be calculated for the four luminaries using Eq. °(6-9).

$$\Psi_{L1} = 180 - \tan^{-1}\left(\frac{|Y_0 - a|}{X_0}\right) \tag{6}$$

$$\Psi_{L2} = \tan^{-1}\left(\frac{|Y_0 - a|}{S - X_0}\right) \tag{7}$$

$$\Psi_{L3} = \tan^{-1}\left(\frac{|Y_0 - a|}{(S - X_0) + S}\right) \tag{8}$$

$$\Psi_{L4} = 180 - \tan^{-1}\left(\frac{|Y_0 - a|}{X_0 + S}\right) \tag{9}$$

The azimuth matrixes are generated using Eq. °(2-9) for each grid point.

2.2.2. Vertical luminary angle (θ)

The vertical luminary angle is the angle between the luminary and the calculation point on the vertical plane, as it is shown in Fig. 3, forming another rectangle whose base is the hypotenuse of the triangle presented in Fig. 4. The angle can be calculated using Eq. °(10).

$$\theta = \tan^{-1}\left(\frac{\sqrt{X_0^{\,2} + (Y_0 - a)^2}}{H}\right) \tag{10}$$

where H is the pole height.

2.2.3. Observer angle (β)

The observer has two angles, as it can be seen in Fig. 3: the angle in relation of the road plane (γ) and the angle in relation to the luminary-point vector (β).

The first is fixed in 1° to 1.5° interval [1-6] and the second is calculated using algebra, assuming two vectors $\vec{AP}$ and $\vec{LP}$, as presented in Fig. 5:

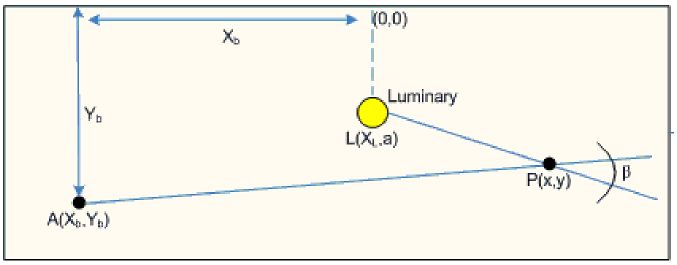

Fig.5 Angle β formed by vector $\vec{AP}$ *and* $\vec{LP}$.

Both vectors are defined by two points and the angle between them is β, which is calculated by the arccosine of the vectors scalar product over the multiplication of their modules [7], as represented in Eq. °(11).

$$\beta = \cos^{-1}\left(\frac{\vec{AP}\cdot\vec{LP}}{\left|\vec{AP}\right| * \left|\vec{LP}\right|}\right) \tag{11}$$

After this part, all angle matrixes are ready and stored in the microcontroller's memory and the parameters calculation can now start.

2.3. Illuminance calculation (E)

The illuminance is calculated using the traditional cosine equation published in several studies [2,6,8,9] represented by Eq. °(12).

$$E = \frac{I_r(\psi,\theta) * \frac{\varphi}{1000} * \cos^3\theta}{H^2} \tag{12}$$

where $I_r(\psi,\theta)$ is the relative luminous intensity taken from the luminary *I table* and φ is the luminary luminous flux.

2.4. Luminance calculation (L)

For the luminance calculation, the *r_tables* where used to find the approximate result. The equation used is represented in Eq. °(13) [2,6].

$$L = \frac{\frac{r(\beta,\theta)}{10000} * I_r(\psi,\theta) * \frac{\varphi}{1000}}{H^2} \tag{13}$$

where $r(\beta,\theta)$ is the reduced luminance coefficient taken from the *r_table*.

2.5. Optimization function

The calculation functions were designed to calculate E, L and U_o starting from six variables: *a*, W, luminary type, pavement type (R1, R2, R3 or R4), S and H. The first four variables are defined by the user, leaving only two variables to be calculated through optimization: H and S. With S been the most important as it is directly connected to energy consuming.

The optimization algorithm start placing the luminaries 50m (S) away from each other and 30m (H) high and begin to decrease S and H, calculating L (E was not used in the optimization function) and U_o every iteration, subtracting the result from the standard value (loaded into the memory) until it reaches zero, when the optimum values of S and H, together with other secondary results, are displayed on a LCD or sent through USB to a computer.

The optimization function was implemented in two ways: proportional and difference-based.

2.5.1. Proportional form

In the proportional form, both S and H are decremented in one unit each iteration making the process very simple but taking a long time to converge, as the decrement doesn't depend on the difference from the parameter being calculated (L and U_o).

Process is done using two loops: an outer loop for S decrementing and an inner loop for H decrementing. Then, for each meter taken from S, all H range is tested.

2.5.2. Difference-based form

This form was called this way for the variable decrease is based on the difference between the last parameter (k-1) calculated and the standard's limit, following Eq. °(14) and Eq. °(15). Therefore, the farther the parameter is from the standard, the bigger the decrease will be, resulting on a faster algorithm conversion.

$$H_k = H_{k-1} - [K_1 * (L_{k-1} - L_{st})] \tag{14}$$

$$S_k = S_{k-1} - [K_2 * (L_{k-1} - L_{st})] \tag{15}$$

3. Results

All algorithm results were compared with simulations on Dialux software which was taken as reliable CAD lighting software, used in several international projects.

The graphic of the algorithm conversion are presented in Fig. 6 and Fig. 7, using a = 1m, W = 8m, n = 2 lanes, SRC 612 Philips sodium-vapor luminary with P_a = 443W and road pavement R3.

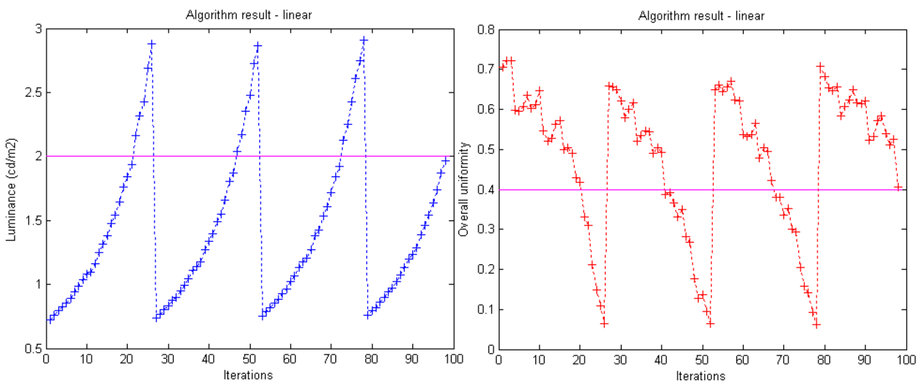

Fig.6 Algorithm graphic conversion for proportional optimization.

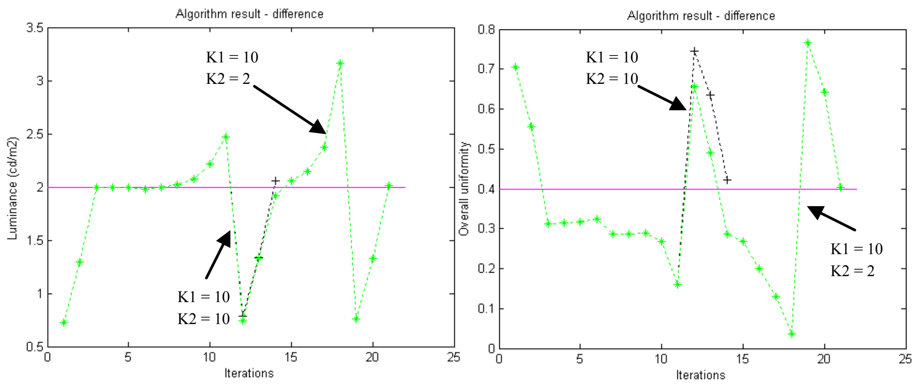

Fig.7 Algorithm graphic conversion for difference-based optimization.

On Fig. 6, the graph shows that the algorithm takes about 96 iterations to convert to the desired pattern, using proportional optimization and on Fig. 7 it takes about 22 iterations using difference-based optimization with K1 = 10 and K2 = 2 (green) and 12 iterations with K1 = 10 and K2 = 10 (black). The continuous lines indicate the EN 13201-1 standard recommendation (L = 2 cd/m^2 and U_o = 0.4) for ME1 class.

The final results, comparing with Dialux, are presented on Table 1.

Table 1. Final algorithm results.

Parameters	Proportional optimization	Difference-based optimization K1 = 2, K2 = 10	Difference-based optimization K1 = 10, K2 = 10	Dialux (validation)
L (cd/m^2)	1.969	2.009	2.058	2.0
U_o	0.406	0.404	0.422	0.4
S (m)	47	46.72	45.28	47
H (m)	11	10.9	11.25	11
P_r (W/km)	9.4k	9.5k	9.8k	9.4k

4. Conclusions

The final results confirmed that the optimization algorithm is able to calculate a distance between luminaries which meets the standard limit with minimum power consuming and a pole height to guarantee the uniformity, being sufficiently light to be run on a simple hardware (formed by the microcontroller ARM7TDMI ADUC7026, some keys to enter data and a LCD) with no need for an external memory or any other device, becoming an easy-to-use tool to new or existing roadway lighting designs, even though the algorithm secondary function, simulation, was not presented in this work due to space restrictions.

References

[1] CIE Pub. Nº 180, Road Transport Lighting for Developing Countries, 2007, pp. 2 – 31.

[2] IESNA, The IESNA Lighting Handbook Reference and Application, IES, 9th edition, 2000, pp. 757 – 790.

[3] ABNT, NBR 5101 Iluminação Pública, 1992, pp.5 – 9.

[4] W. J. Frith, M. J. Jacket, Road Lighting for Safety – a forward-looking, safe system-based review, Procedings of Australasian Road Safety Research, Policing and Education Conference, 2009, pp. 408-418.

[5] Ö. Güler, S. Onaygil, Evaluation of Visibility Level Formula in Road Lighting with Field Measurements, Istanbul Technical University, 2002.

[6] Luce e Illuminazione, EN 13201 – 3: Road Lighting Calculation for Performance, UNI, 2004, pp. 5-39.

[7] A. Steinbruch, P. Winterle, Algebra Linear, Pearson-Makron Books, 2nd edition, 2005, pp. 8-12.

[8] G. J. C. da Costa, Iluminação Econômica, EDIPUCRS, 1st edition, 1998, pp. 299-326.

[9] Luce e Illuminazione, EN 13201 – 2: Road Lighting Performance Requirements, UNI, 2004, pp. 7-14.