Reflector profile optimisation using Radiance

GIULIO ANTONUTTO

KRZYSZTOF WANDACHOWICZ

ARUP

POZNAŃ UNIVERSITY OF TECHNOLOGY
The idea

Krzysztof WANDACHOWICZ

Giulio ANTONUTTO

5th International Radiance Scientific Workshop
13-14 September 2006, De Montfort University, Leicester, UK
Room 0.09, Queens Building (#26 on the Campus map)

Organised by John Mardaljevic and Greg Ward
Hosted by the Institute of Energy and Sustainable Development

Design optimisation with Radiance
Giulio Antonutto Andrew Mc Neil Kristina Shea

Calculation of Luminaires Using Radiance
Krzysztof Wandachowicz
Poznan University of Technology,
Institute of Industrial Electrical Engineering,
Division of Lighting Engineering,
Poznan, Poland
http://lumen.iee.put.poznan.pl/kw

Reflector profile optimisation using Radiance
Giulio ANTONUTTO Krzysztof WANDACHOWICZ
Calculation of luminaires using Radiance

**Different examples:** diffuse reflector and globe, specular reflector, semi-specular reflector.

**Results compared with:** the analytical solutions, laboratory measurements.

**Example of specular, paraboloid reflector** – compared with the analytical solution

<table>
<thead>
<tr>
<th>Light source diameter (d_1=1\text{cm})</th>
<th>Light source diameter (d_2=4\text{cm})</th>
</tr>
</thead>
<tbody>
<tr>
<td>Luminous flux [(\text{lm})]</td>
<td>Luminous flux [(\text{lm})]</td>
</tr>
<tr>
<td>Analytical solution</td>
<td>929</td>
</tr>
<tr>
<td>Radiance</td>
<td>950</td>
</tr>
<tr>
<td>Difference [%]</td>
<td>2</td>
</tr>
<tr>
<td>Luminous flux [(\text{lm})]</td>
<td>929</td>
</tr>
<tr>
<td>Analytical solution</td>
<td>917</td>
</tr>
<tr>
<td>Radiance</td>
<td>9600</td>
</tr>
<tr>
<td>Difference [%]</td>
<td>-1</td>
</tr>
<tr>
<td>Luminous intensity at the axis [(\text{cd})]</td>
<td>145 121</td>
</tr>
<tr>
<td>Analytical solution</td>
<td>9 072</td>
</tr>
<tr>
<td>Radiance</td>
<td>9 600</td>
</tr>
<tr>
<td>Difference [%]</td>
<td>6</td>
</tr>
</tbody>
</table>

Luminous intensity changing in polar angles for light source diameter \(d\) and different distances \(r\) from light centre of the reflector.
Optimisation of luminaires vs calculation

**Luminaire optical design**
- determining the optimal luminous intensity distribution or optimal luminance distribution.

**Constraints can include:**
- the size of luminaire,
- the properties of material used,
- the overall cost of the solution proposed,
- the minimum efficiency that can be accepted, etc.

**Optimisation of a luminaire:** the goal is to meet a given luminous intensity distribution by adjusting:
- the geometry of the reflector,
- optical properties of the reflector material,
- position, power and geometry of lamps, material used are set,

Variables describe the geometry of the reflector

---

**Diagram:***

- **calculation of luminaires**
  - description of optical elements (reflector, diffuser)
  - model of light source
  - properties of optical materials

- **optimisation of luminaires**
  - luminous intensity distribution
  - luminance distribution
Optimisation of luminaires

MATLAB – optimisation tool
- determining the optimum reflector shape to produce the luminous intensity distribution required,
- generation and evaluation of a succession of designs,
- each variation is derived from the previous by adjusting the design variables,
- variables are determining reflector shape,
- script creates Radiance geometry,
- an algorithm decides how and which variable needs adjusting and iterates the process until an optimal solution is found,
- stopping criteria are implemented.

RADIANCE – calculation tool
- calculation of illuminance at the distance „r” from the light centre of the reflector (rtrace -I),
- calculation of luminous intensity using inverse square law,
- polygon - surface for reflector and light source,
- mirror - material for reflector,
- light - material for light source,
- light source is small sphere consists of triangles,
Implementation of the computational optimisation workflow

Start

- Start point $X_0$
- iterates $X_i$ until min is found

Matlab Optimization solver

- create the luminaire model ($X_i$)

Unix shell ray tracing (Radiance)

- transform the luminaire model into Radiance format ($oconv$) model ($X_i$)

End

- write results, history $X_i, C_i, Y_i$

- stopping criteria test

- read $C_i$
- calculate the objective function ($C_i \rightarrow Y_i$)

- simulation: ray tracing ($itrace$)
- calculation: luminous intensity ($C_i$)
- model($X_i$)= $C_i$

- save $C_i$
Modelling the reflector geometry

Rotationally symmetrical surface. Reflector shape profile revolving around Z axis.

**Pp, Pk** – fixed points, determine the maximum reflector dimensions.

**P1, P2, P3** – points that have fixed X coordinates while Z coordinates can be adjusted within a range. Surface of revolution will be concave.

Variables are recorded together as a single vector with three dimensions:  
\[ \mathbf{X_i} = [P1 \ P2 \ P3] \]

The vector \( \mathbf{X_i} \) is used by the optimisation algorithm to adjust the reflector shape.

The reflector profile is **interpolated** using a **piecewise cubic Hermite polynomial** strategy and it is constrained to points Pp, P1, P2, P3, Pk.
Hermite polynomial interpolation

MATLAB **spline** and **pchip** function.

**Spline** piecewise interpolation method is class $C^2$ (second derivative is continuous). A function generated with this approach has smooth shape but doesn’t preserve monotonicity.

**Pchip** monotone interpolation can be accomplished using cubic Hermite splines with the slopes modified. In this case the function is class $C^1$ (only first derivative is continuous). A discontinuous second derivative implies discontinuous curvature, but this method guarantees shape preservation and local monotonicity.

The monotone interpolation method available is the Fritsch–Carlson which allows calculating slopes and to preserve shape (it is used in the MATLAB function **pchip**).
Comparison of the results of the reflector geometry interpolation using the **spline** and **pchip** function.

In the middle area of the curve between points P1 and P3 the **spline** function is not monotonic.

**spline** – piecewise cubic interpolation

**pchip** – monotone Fritsch-Carlson piecewise Hermite polynomial interpolation
Implementation of the computational optimisation workflow

**Animation**

R_I_PARmax05-debug-15.mov

4 frames per second

1 frame = 1 iteration

reflector profile

luminous intensity distribution calculated for above reflector profile (using Radiance)
Computational optimisation strategies (algorithms)

**MATLAB optimisation algorithms** (find minimum):

**Pattern search** - direct search method for solving optimization problem that doesn’t require any information about the gradient of the objective function. The algorithm search a set of points, called mesh, around the current point $X_i$ (the current reflector configuration). To avoid being trapped in local minima the strategy that selects random starting points $X_0$ is used.

**Simulated annealing** is a method for solving unconstrained and bound-constrained optimization problems. The method models the physical process of heating a material and then slowly lowering the temperature to decrease defects.

**Genetic algorithm** is a method for solving both constrained and unconstrained optimization problems that is based on natural selection. The algorithm selects individuals at random from the current population to be parents and uses them to produce the children for the next generation. The main difference from other algorithms is that the genetic algorithm generates a population of points at each iteration instead of generating single point.
Case study A – optimisation of luminous intensity for a given direction

**Variables:** Z coordinate of points P1, P2 and P3

**Constrains:** \((-0.9375 x + 0.04) < P1,P2,P3 < 0.04\)

**Objective function** \(F(X)\): maximum luminous intensity at \(\gamma=0^0\)
(at the axis)

\[ F(X) = -I_{\gamma=0} \]

As the **analytical solution** to the problem is known, it is possible to compare the different approaches.

In fact, the maximum luminous intensity, which can be only obtained with a **parabolic reflector**, can be calculated using the following equation:

\[
I_{\gamma=0} = I_{\text{light\_source}} + I_{\text{reflector}} = \frac{\Phi}{4\pi} + \frac{\pi D^2}{4} L_g \rho = \frac{1000 + \pi \cdot 0.3^2}{4\pi} 1065294 \cdot 0.9 = 67850 \ [cd]
\]

where: \(\gamma\) - polar angle, \(\Phi\) - luminous flux (1000 lm),
\(D\) - diameter of the reflector (0.3 m),
\(L_g\) – luminance of light source (1065294 cd/m²),
\(\rho\) - reflectance of reflector’s surface (0.9)
Case study A – Pattern search

10 starting points have been used, of which 4 got nearly the same best solution.

**Calculated reflector profile is almost identical to parabolic curve**, therefore it can be said that **algorithm have found optimum solution**.

Luminous intensity distribution in the reflector’s axis. Final results for each of the 10 series.

<table>
<thead>
<tr>
<th>No. of series</th>
<th>Luminous intensity $I_{\gamma=0}$ [cd]</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11 000</td>
</tr>
<tr>
<td>2</td>
<td>21 000</td>
</tr>
<tr>
<td>3</td>
<td>31 000</td>
</tr>
<tr>
<td>4</td>
<td>41 000</td>
</tr>
<tr>
<td>5</td>
<td>51 000</td>
</tr>
<tr>
<td>6</td>
<td>61 000</td>
</tr>
<tr>
<td>7</td>
<td>71 000</td>
</tr>
<tr>
<td>8</td>
<td>81 000</td>
</tr>
<tr>
<td>9</td>
<td>91 000</td>
</tr>
<tr>
<td>10</td>
<td>10 000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Theoretical best solution</th>
<th>Optimization algorithm (series no. 3)</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>67 850 cd</td>
<td>69 550 cd</td>
<td>2.5 %</td>
</tr>
</tbody>
</table>

Reflector profile and parabolic curve
The effect of the re-annealing technique is clear: the temperature raises after a certain number of new points have been accepted, and starts the search again at the higher temperature. This technique avoids local minima.

Changes in luminous intensity in the reflector’s axis in consecutive iterations.

<table>
<thead>
<tr>
<th>Luminous intensity in the reflector’s axis $I_{γ=0}$</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical best solution 67 850 cd</td>
<td>Optimization algorithm 68 539 cd</td>
</tr>
</tbody>
</table>
Case study A – Genetic algorithm

The plot shows a vertical line at each generation. This line represents the range of the objective function values within a generation. Reducing the amount of mutation decrease the diversity of subsequent generations. Diversity is important to the genetic algorithm because it enables the algorithm to search a larger region of the space (no of population is 20).

The objective value changing in subsequent generations. The lower objective value, the higher luminous intensity in the reflector’s axis.

<table>
<thead>
<tr>
<th>Luminous intensity in the reflector’s axis $I_{\gamma=0}$</th>
<th>Difference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Theoretical best solution</td>
<td>Optimization algorithm</td>
</tr>
<tr>
<td>67 850 cd</td>
<td>68 359 cd</td>
</tr>
</tbody>
</table>
Simulated annealing and genetic algorithm methods return slightly different results each time when are run. This is because they both use random number generators.

Patternsearch also returns random solution as the starting points $X_0$ are defined randomly.
Case study B – optimization of luminous intensity meeting a set distribution

Variables: Z coordinate of points P1, P2 and P3

Constrains: (-0.9375 x + 0.04) < P1,P2,P3 < 0.04

Objective function $F(X)$ with penalty:
1. maximum luminous intensity at $\gamma=0^0$ (at the axis),
2. half-peak divergence value is $\delta=5^0$, the ratio $R=0.5$

$$F(X) = -\frac{I_{\gamma=0}}{m} + f_k$$

$m$ - coefficient reducing luminous intensity to the level of the penalty function $f_k$.

$$f_k = \begin{cases} 
100 \left( \frac{I_{\gamma=5}}{I_{\gamma=0}} - 0.5 \right)^2 - \text{tol} & \text{if } \left| \frac{I_{\gamma=5}}{I_{\gamma=0}} - 0.5 \right| - \text{tol} > 0 \\
0 & \text{otherwise}
\end{cases}$$

If tolerance $\text{tol}$ is set at 0.05, this means that all results with the ratio $R$ within the range from 0.45 to 0.55 are accepted without penalty ($f_k = 0$).

This example has not analytical solution which can be easily evaluated
Case study B – Pattern search

Figure shows luminous intensity in the reflector’s axis (Ig0) and ratio R as a final result for each of the 10 series.

Luminous intensity must be as much as possible, ratio R should be within the range 0.45÷0.55.

Only one series (no. 4) is close to the optimum.
Case study B – Simulated annealing

Figure shows a selected portion of the simulated annealing optimization (one re-annealing step).

Even if there are higher values of luminous intensity, they are not accepted because of the constrain at 5° (ratio $R = 0.45\div0.55$). The final solution is reached when the objective function is minimized.
Case study B – Genetic algorithm

High differences in objective function are a result of the penalty which increase the objective function value when ratio R is outside the range 0.45÷0.55.

The objective value changing in subsequent generations. The lower objective value, the higher luminous intensity in the reflector’s axis.
The shorter calculations were performed by genetic algorithm and this method has also the best ratio of success percentage.
Figure presents the luminous intensity curve of a reflectors which profiles were calculated using genetic algorithm method for:

(a) simply objective function (case study A),
(b) and objective with penalty (case study B).
Conclusions

Computational design optimisation can successfully be used to optimise simple reflector geometry to meet luminous intensity criteria.

The methodology used is general and can be easily applied to more complex scenarios.

In regards to the case studies shown, of the method used, simulated annealing and genetic algorithm seem to be the most promising techniques. They both show a good ratio of iterations per second and a high percentage of solution within the 5% best.

Pattern search shows a more cleaner convergence but it is often confused by local maxima/ minima.