11th Annual
International Radiance Workshop

Reflector calculation, validation and optimisation using Radiance

Krzysztof WANDACHOWICZ

POZNAN UNIVERSITY OF TECHNOLOGY

The idea

Radiance Workshop 2004
Calculation of Luminaires Using Radiance

Radiance Workshop 2006
***Design optimisation with Radiance* by Giulio Antonutto**

Radiance Workshop 2010
***Reflector profile optimisation using Radiance* by KW and GA**

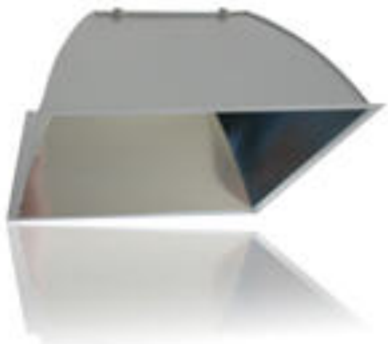
Radiance Workshop 2012
Reflector calculation, validation and optimisation using Radiance

What's new in 2012 ?
Validation:
- reflectors from ACL, Fortimo LED DLM 2000 module,
- luminous intensity curve measurement using goniophotometer,
- 3 D scanning, building reflector model, calculation luminous intensity curve using Radiance,

Optimisation:
- Genetic Algorithm, constrained and unconstrained problems, some examples of GA options,
- new objective function.

Validation

Reflectors from ACL for Fortimo LED DLM module

Measurement

Luminance distribution on the surface of the LED module.

Luminous intensity curve using goniophotometer.

Simulation:

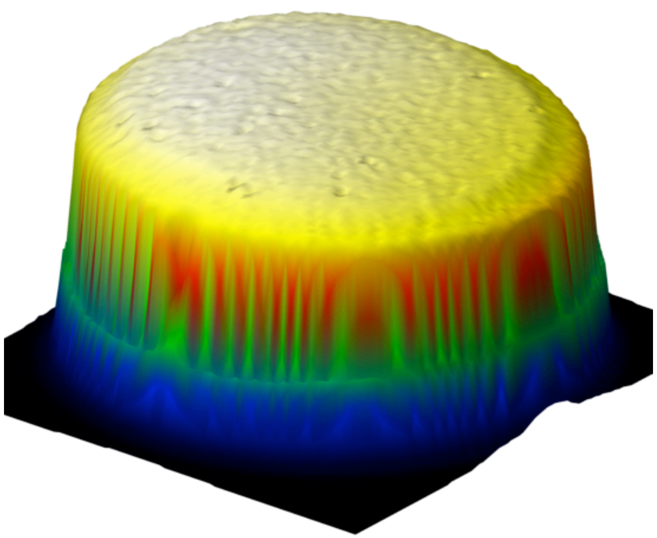
3 D scanning of reflector surface.

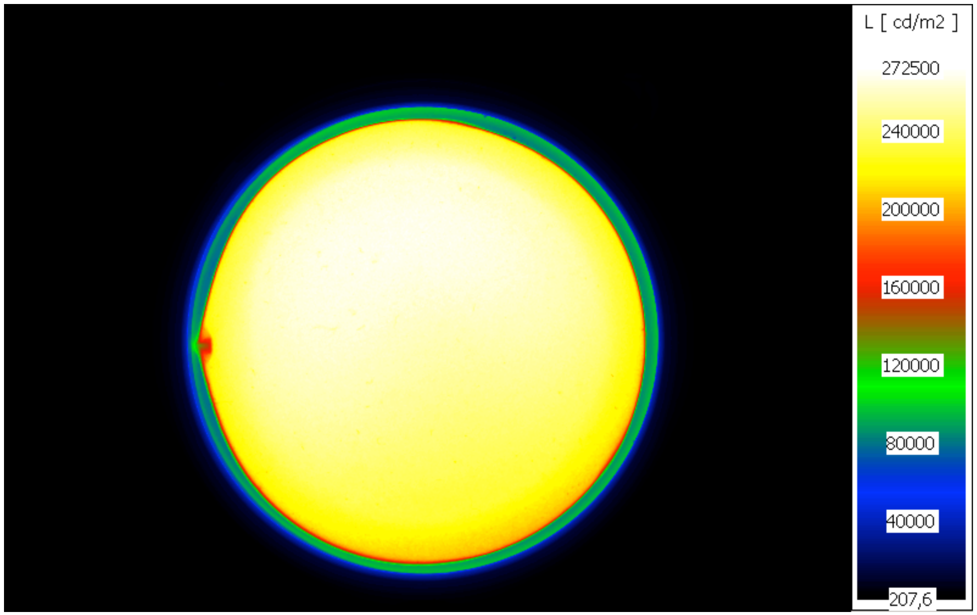
Building reflector and LED module model.

Calculation of luminous intensity curve using Radiance.

Validation

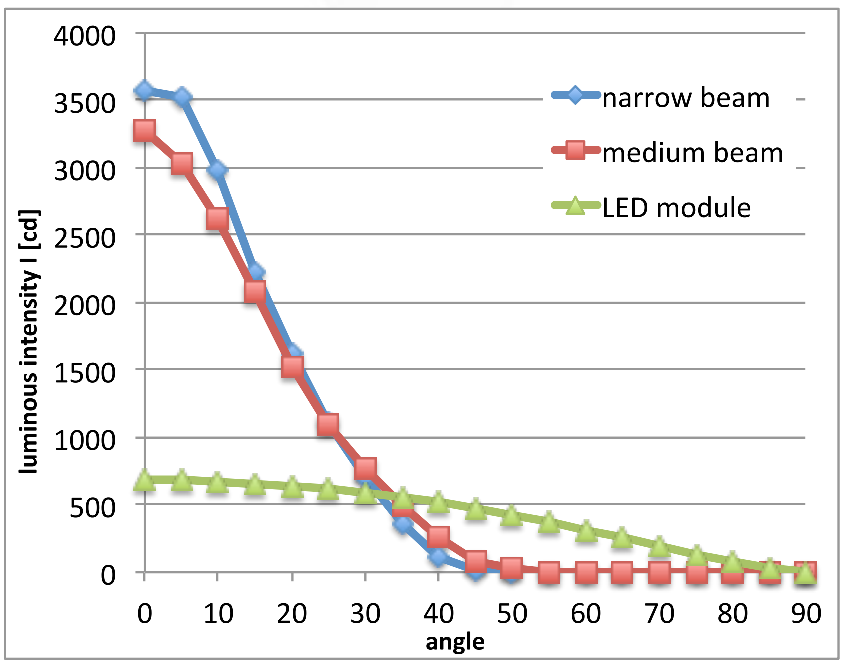
Measurement
Luminance distribution on the surface of the LED module using luminance measuring camera.

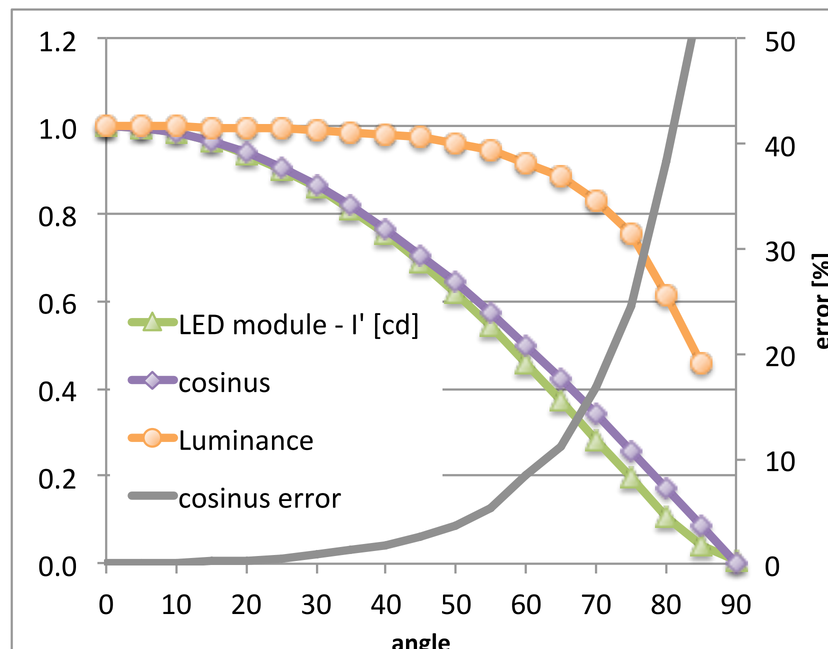




Measurement

Luminous intensity curve using goniophotometer.





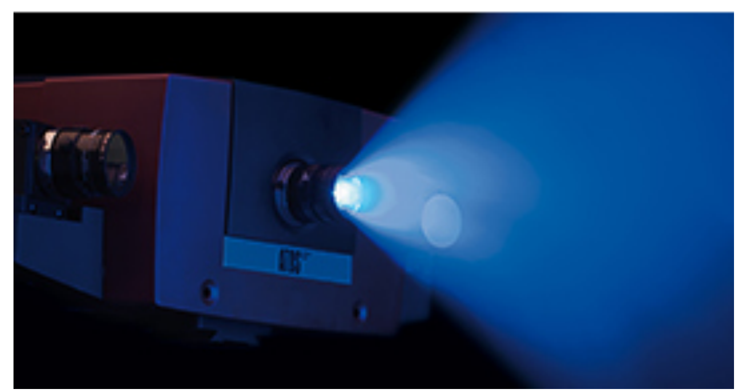
Validation

Simulation
3D scanning

Two round reflectors, specular finished.
3 D scanning of reflector surface.

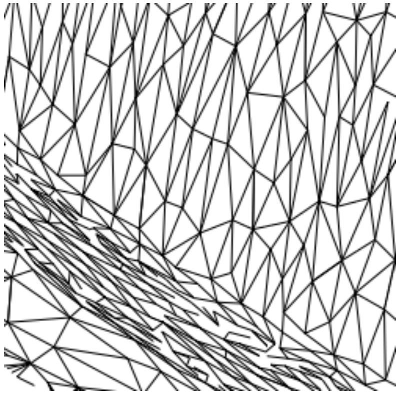
Building reflector model: STL – DXF – (dxf2rad) - RAD

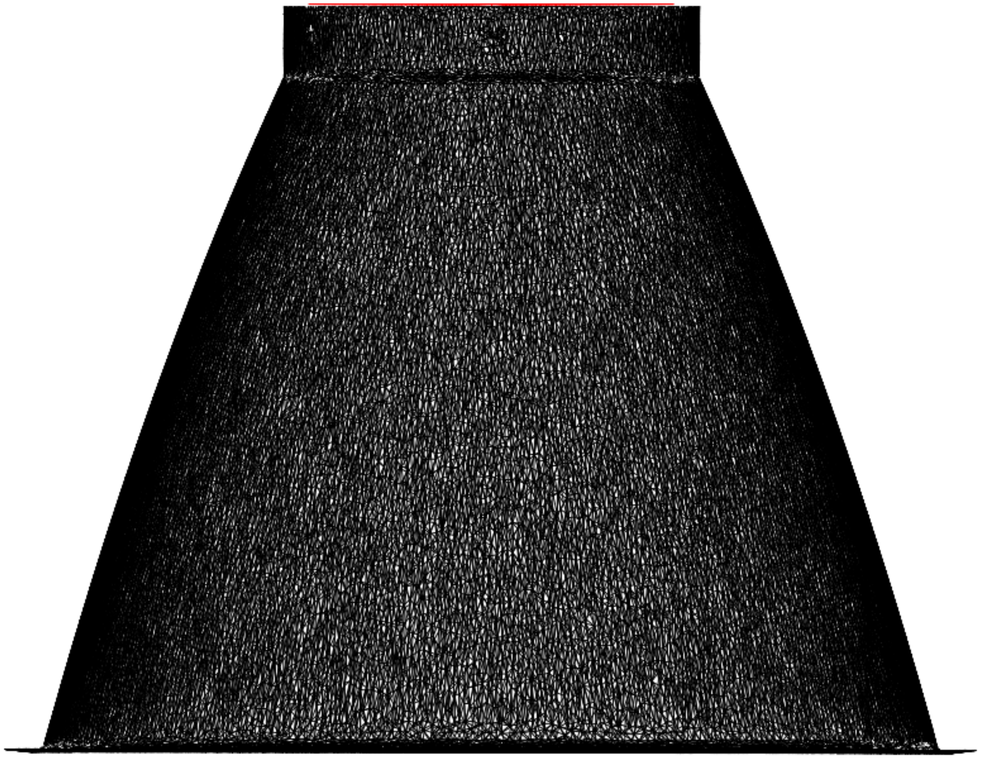
62517 and 51746 triangles per model

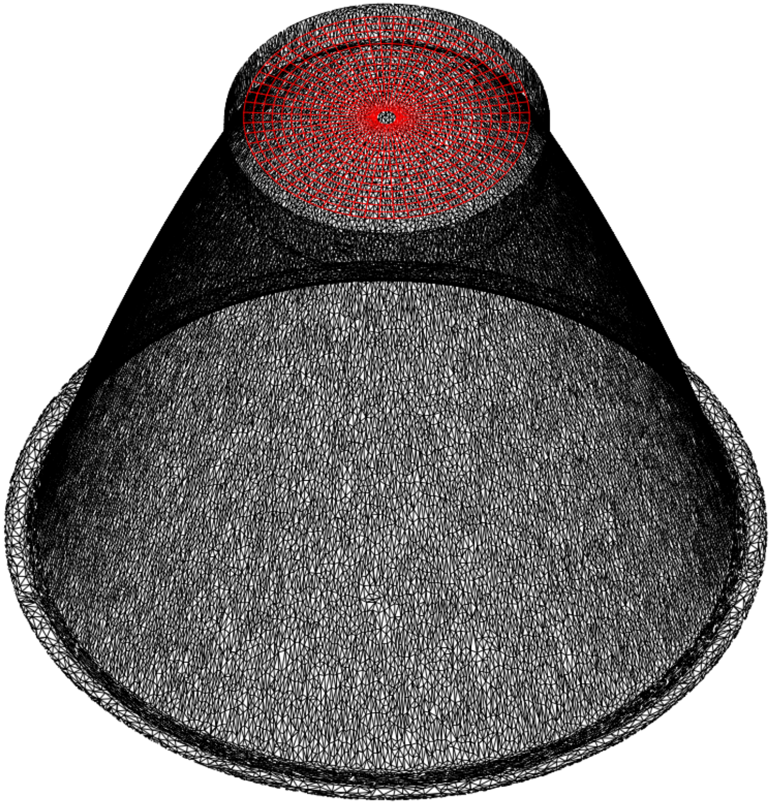


Validation

Simulation
Reflector and LED module model.

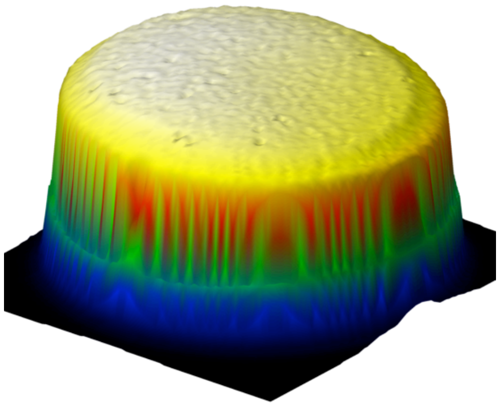


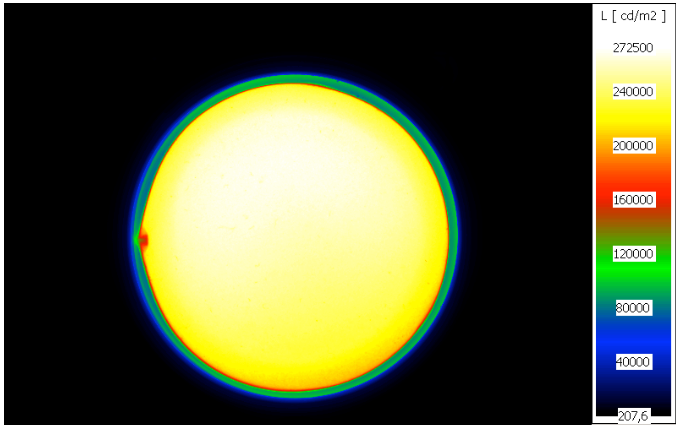




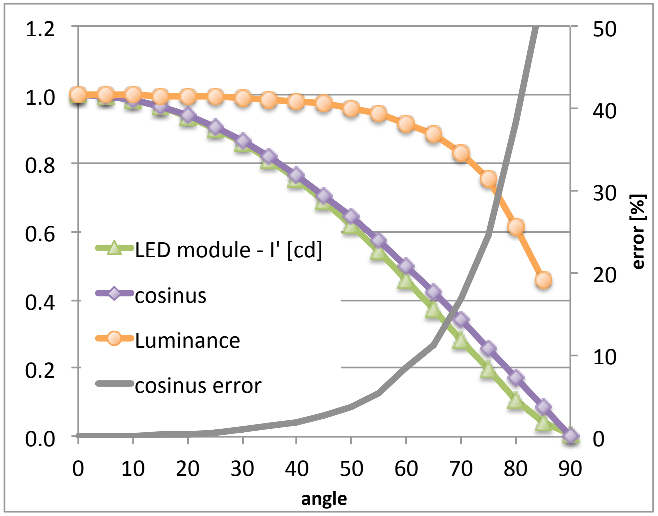
Validation

Simulation
Building LED module model.



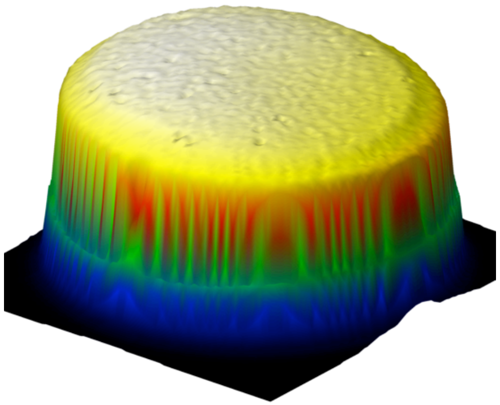


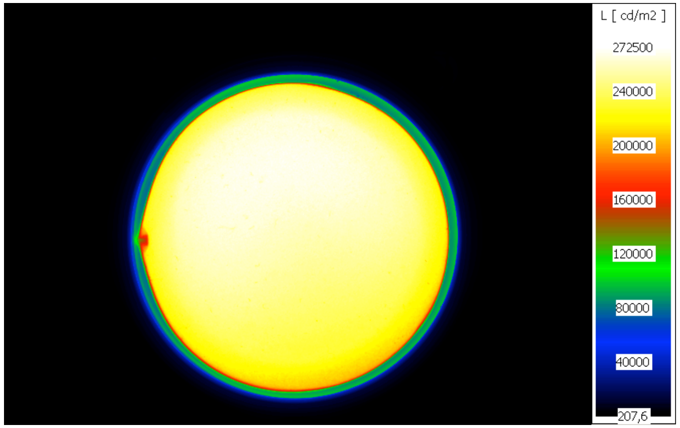
Luminance distribution is not symmetrical.
Light source surface is not Lambertian.

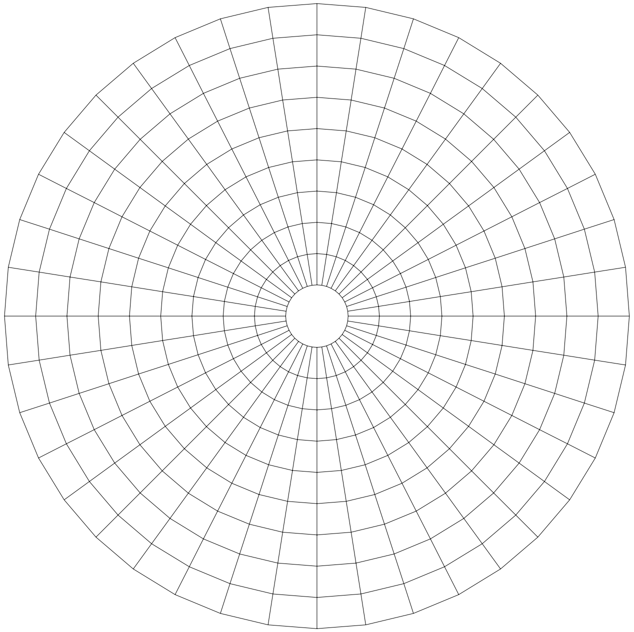


Validation

Simulation
Building LED module model.







<u>Assumptions</u>:
Luminance distribution is symmetrical but not uniform.
Light source circle is divided into ten rings. Every ring can have different luminance value.

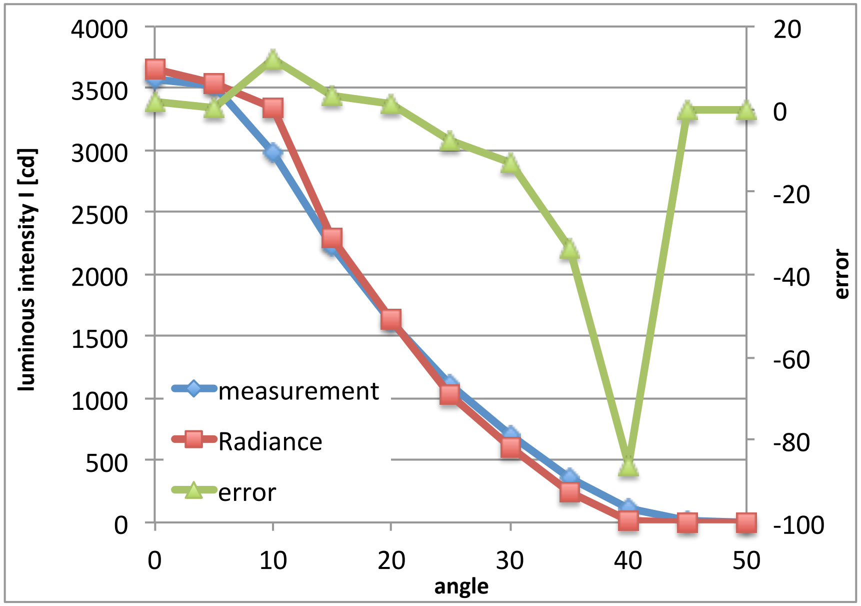
Light source surface is nearly Lambertian.
Tests performed for non-Lambertian surface got nearly the same results.

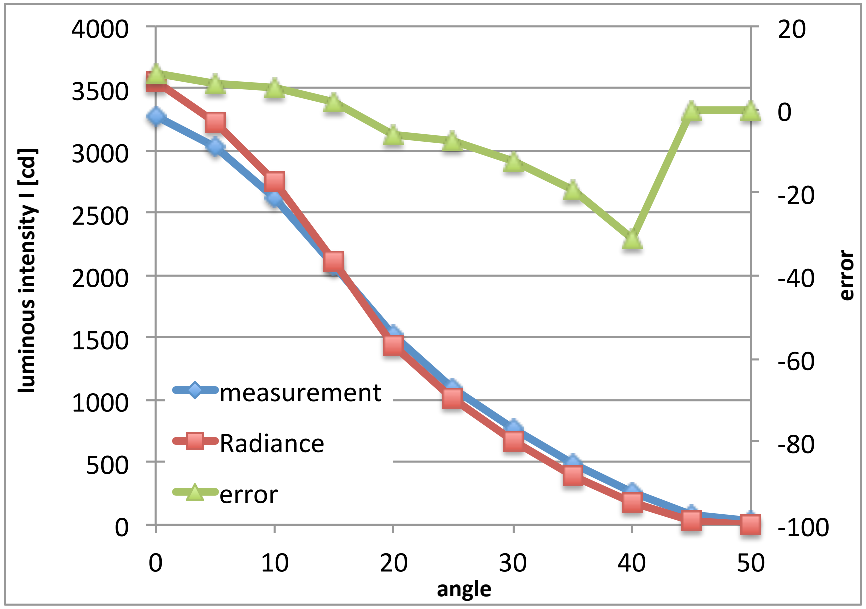
Validation

Simulation
Calculation of luminous intensity curve using Radiance.

rtrace -ov -I @param.opt name.oct < points-polar.txt | **rcalc** -e "$1=$1*179*distance^2" > I_candela.txt

materials: **mirror, light**



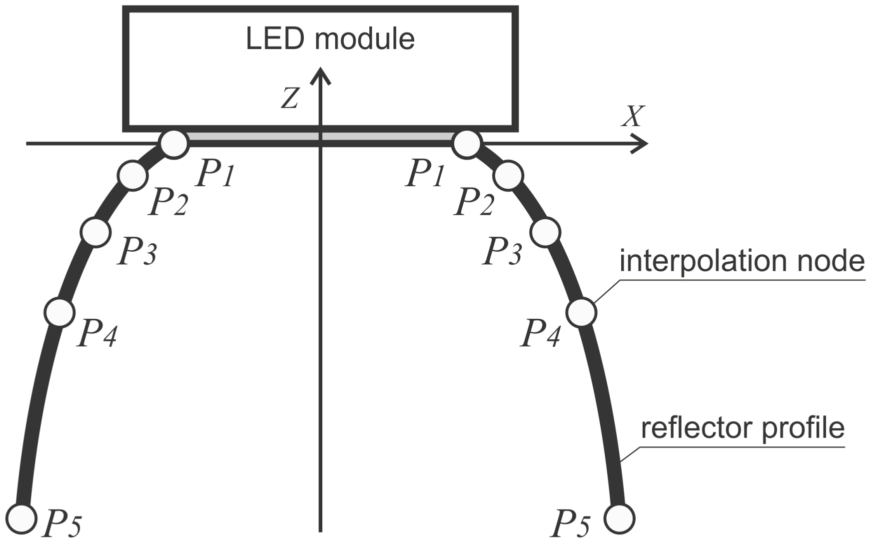


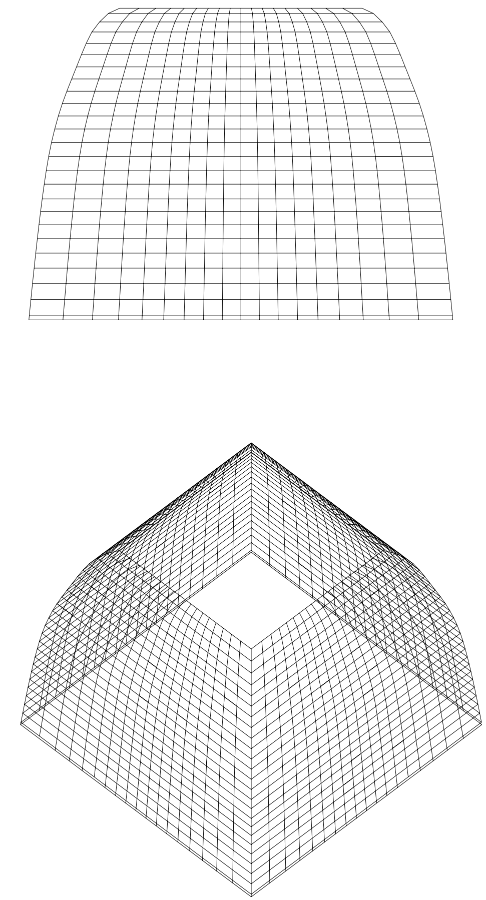
Narrow beam reflector
Calculated luminous flux:
measurement: 1722.3 lm
Radiance: 1647.6 lm
error: 4.3 %

Medium beam reflector
Calculated luminous flux:
measurement: 1764.3 lm
Radiance: 1643.3 lm
error: 6.9 %

Optimisation

Reflector model

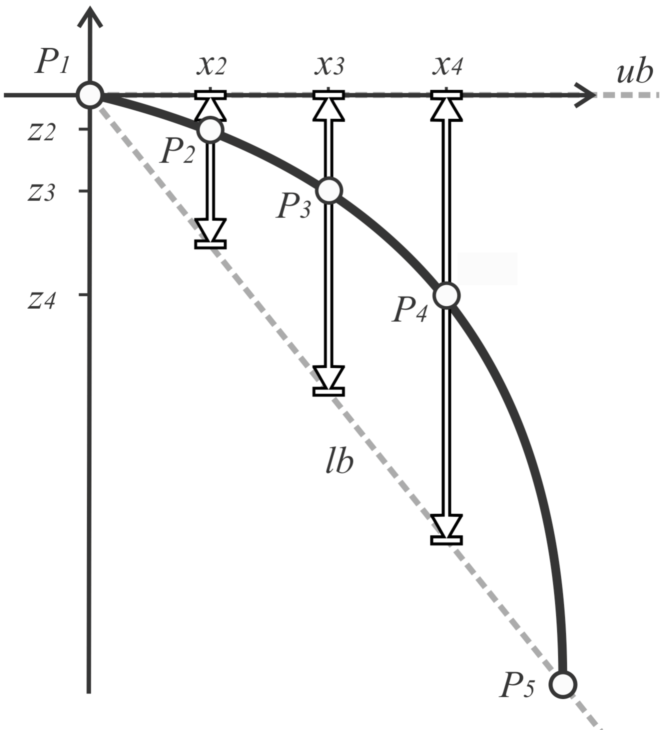


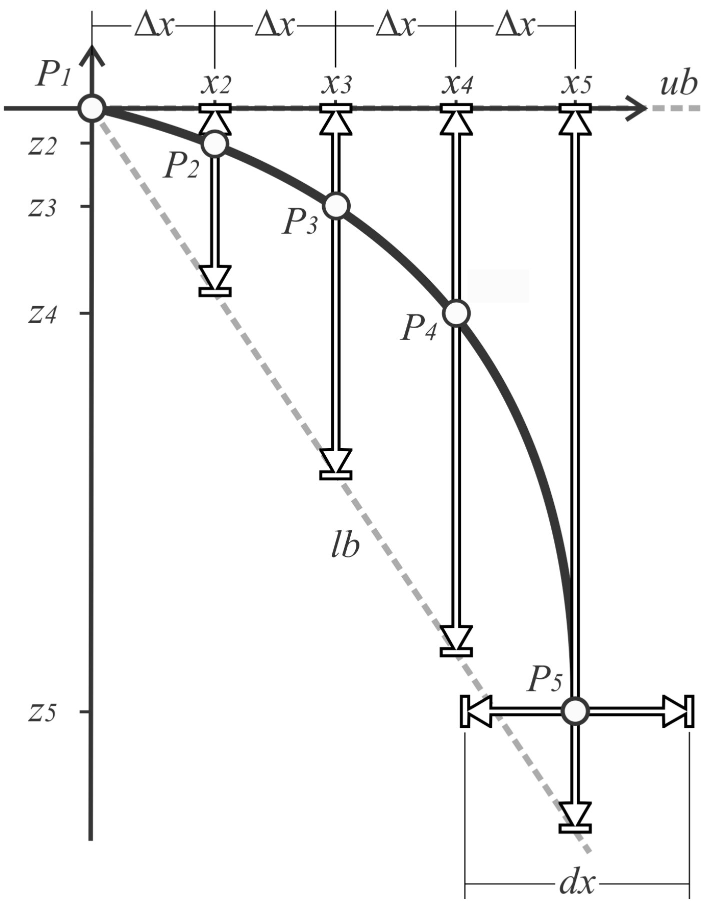


The reflector profile is described by the Hermite interpolating polynomial

Optimisation

Reflector model





Model I
Three points (P2, P3, P4) have fixed X coordinates and can change Z coordinates

Model II
Four points (P2, P3, P4, P5) can change their positions both in the axis Z and the axis X.

Optimisation

Genetic algorithm

Matlab function **ga**

Three operations at each stage of creation of a new generation from the current population:
- **selection**, select individuals (parents), that contribute to the population at the next generation,
- **crossover**, combine two parents to form children for the next generation,
- **mutation**, apply random changes to individual parents to form children.

Classic genetic algorithm operates on a binary representation of individuals.
In the floating-point representation, two points that are close to each other in the representation space will also be close to each other in the problem space (and vice versa).
This is usually not possible in binary representation.

Population type. **Double vector**.
Scaling function. **Rank** (fitscalingrank).
Reproduction. **Elite count = 2**.
Selection. **Stochastic uniform**.
Mutation. **Gaussian**.
Crossover. **Heuristic** returns a child that lies on the line containing the two parents, a small distance away from the parent with the better fitness value in the direction away from the parent with the worse fitness value.

Optimisation

How it works?

Matlab_otimisation_script.m (starts and controls optimisation, calls objective function)

Objective_function.m
(calls generate and calculate functions, reads results from disk, calculates objective function)

Generate_Radiance_geometry_function.m
(generates reflector model, save RAD file)

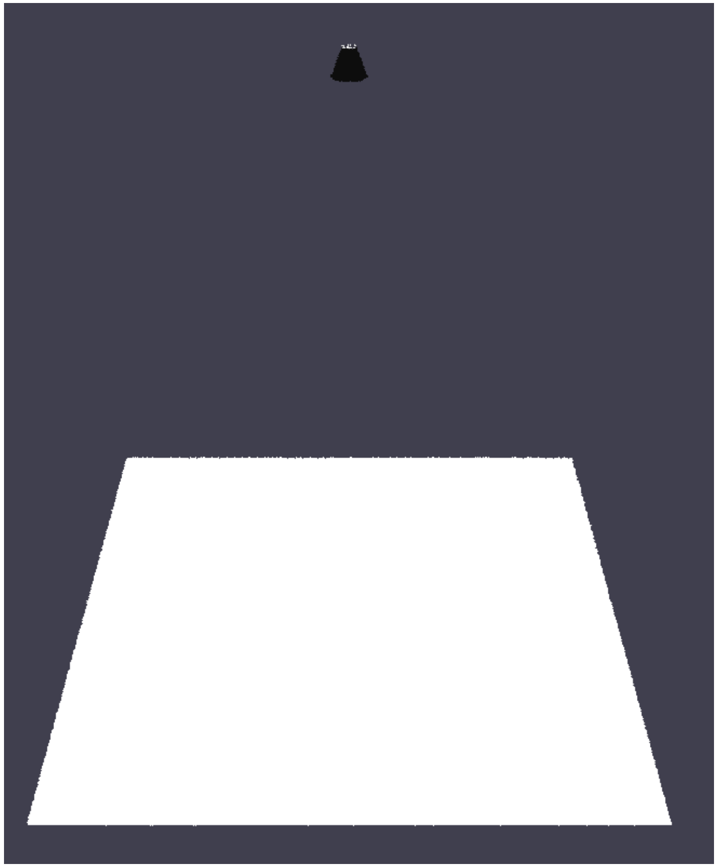
Calculate_function.m
(creates and runs Unix execute file with Radiance commands: **oconv**, **rtrace**, **rcalc** …)

RUN - execute Unix file
(calculates and saves illuminance and luminous intensity using Radiance)

Optimisation

Objective function

The optimisation problem is to finding the reflector shape which produce the maximum value of average illuminance E_{av} with the uniformity ratio of R=0.7 on the illuminated surface.

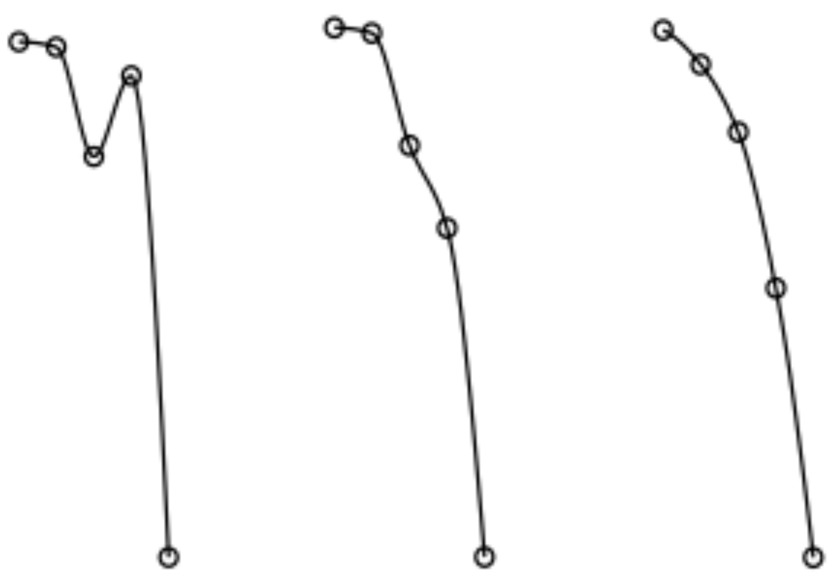


$$F(C) = -E_{av} + f_k$$

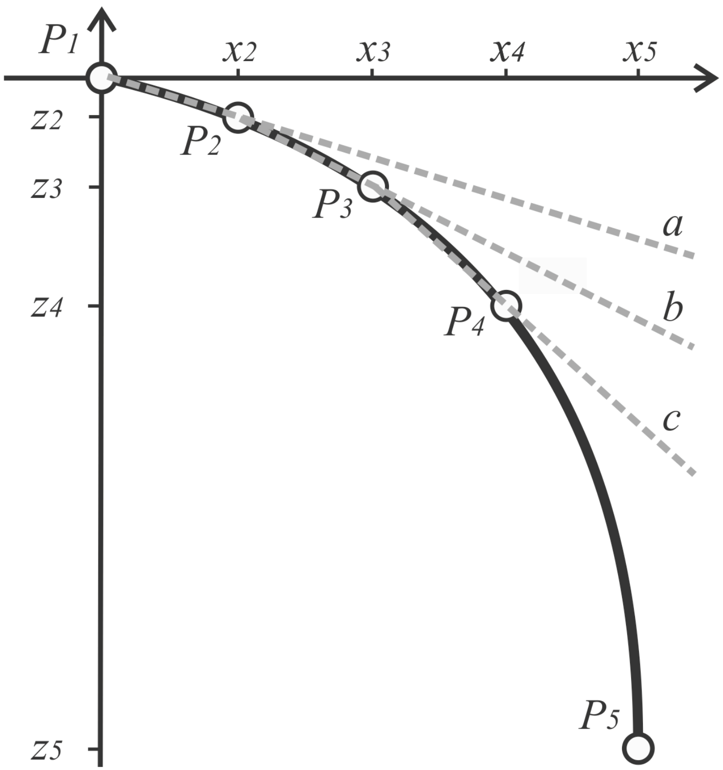
$$f_k = \begin{cases} \left(50\left(R - \frac{E_{min}}{E_{av}}\right)\right)^2 & if \left(R - \frac{E_{min}}{E_{av}}\right) > 0 \\ 0 \end{cases}$$

Optimisation

Constrained and unconstrained problem



Reflector profile curves generated in the unconstrained optimisation process in the 3rd, 10th and 50th iteration



To preserve monotony of the reflector profile curve, constraints are introduced:
- the point P3 should be below the straight line which crosses the points P1 and P2;
- … and so on for next points

Optimisation

Results

Model I

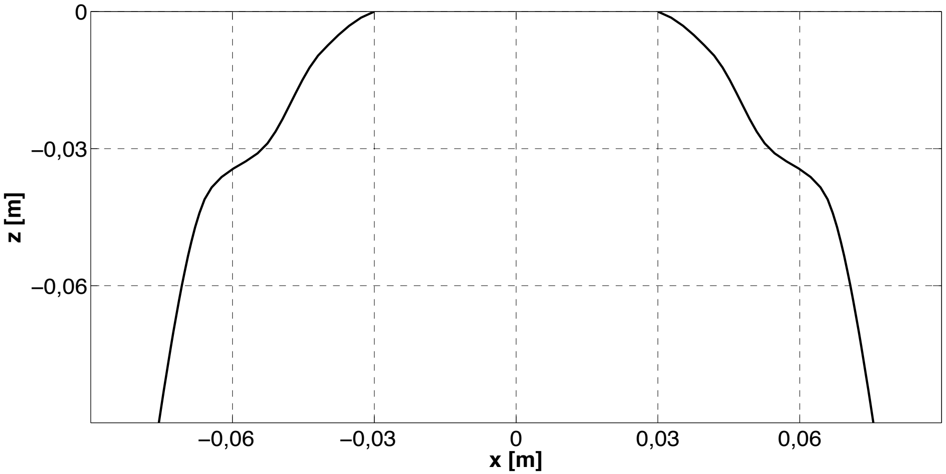
No	Name	Unconstrained		Constrained	
		Average	**Best**	**Average**	**Best**
1	*F(C)*	-91,9	**-98,6**	-91,9	**-99,7**
2	E_{av} [lx]	92,5	99,1	92,2	100,7
3	E_{min} / E_{av}	0,68	0,67	0,70	0,68
4	No of iteration	440	-	441	-
5	CPU time [s]	539	-	595	-

Model II

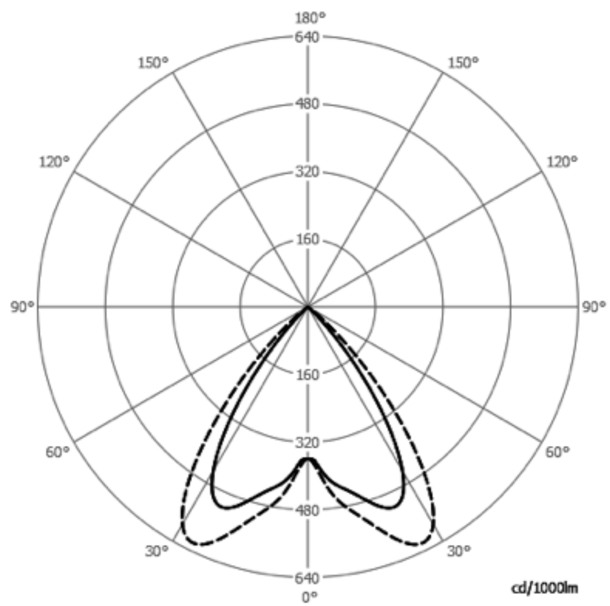
No	Name	Unconstrained		Constrained	
		Average	**Best**	**Average**	**Best**
1	*F(C)*	-102,3	**-113,6**	-95,6	**-106,9**
2	E_{av} [lx]	102,8	116,4	98,7	108,4
3	E_{min} / E_{av}	0,72	0,67	0,71	0,60
4	No of iteration	454	-	446	-
5	CPU time [s]	679	-	734	-

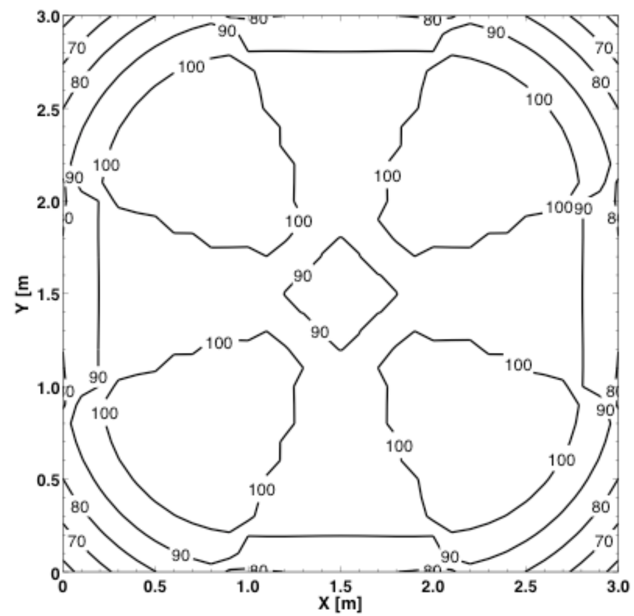
Optimisation

Results



The figure of a tri-curve reflector profile calculated with the algorithm without constraints





Conclusions

Matlab + Radiance = lighting optimisation

I recommend to use floating-point representation instead of binary

Be careful when using constraints

Cooperate with Radiance users

T H A N K Y O U