Technische
Universiteit
Eindhoven

Instituut
Wiskundige Dienstverlening
Eindhoven

Rapport IWDE 97 - 07

Lighting by mulltisources: Mathematics

J. Spoelstra

december 1997

Den Dolech 2
Postbus 513
5600 MB Eindhoven

Instituut
Wiskundige Dienstverlening
Eindhoven

Report IWDE 97-07

Lighting by multisources: Mathematics

J Spoelstra

wskjs@puknet.puk.ac.za

December 18, 1997

Contents

Introduction

In Chapter 1 we consider various approaches to the problem of achieving the light distribution needed for a given application when using numerous small light sources. In Chapter 2 we focus on one approach, and discuss ways of building up a database containing all the data files needed to approach the problem mathematically, and to program it so as to give the required information quickly and efficiently.

Chapter 3 considers ways to use these data files in a program, and to form the combined effect of numerous sources.

In Chapter 4 we start off by defining a function that measures the deviation between an obtained result for a setup and a target. We then give the ideas behind programs to optimize this function over all possible setups. With this achieved we will have a set-up of the sources that will approximate the required distribution as well as possible. Due to some physical limitations — which we discuss — the target light intensity distribution could not be duplicated closely enough. Except for this limitation, the target could be approximated just as closely using only a limited number of optical elements.

Following this, in chapter 5, we regard the illumination of the road as pricipal focus of the investigation rather than just approximating a target distribution. By means of this the criteria demanded of a road lighting system could be met.

By the use of other optical setups, as discussed in chapter 6, the requirements could be exceeded as long as the beams satisfied certain requirements, and not achieved at all if the beams were, for example, too narrow.

Chapter 1

Lighting by multisources: Mathematics

1.1 Approaching the problem

For a block consisting of numerous sources of a predetermined set, we have to *determine* the following

- how to orient and place the block;
- what type of beam and its orientation, to use for each source. A beam consists of a certain deviation and spreading.

This setting must be determined so as to achieve one of the following *objectives*:

- To approximate a prescribed light distribution as given in, for example, an IES-table, as closely as possible.
[In this approach the expertise in road lighting design, of experts in the field, is used directly and the treatment is independent of the application.]
- To achieve the standards set by CIE recommendations.
[In this approach the focus is directly on a specific application, *e.g.*, a road, floodlighting, etc.]

The first approach has one advantage in that the computing time should be shorter, because the function to be optimized can be determined more quickly. A second advantage is that it breaks the job down into two smaller disjoint units (A, obtaining an ideal light intensity distribution, and B, approximating it), wherein existing expertise can be used for stage A. Such a modular approach is almost always the most effective. The disadvantage is that the design at stage B can be at best as good as the prescribed one, *i.e.*, the IES-table given for stage A, which may not be optimal — a better block may be possible than the prescribed IES-table using the multisources.

The second approach has the advantage that it focuses directly on the task which is to be done. It has the disadvantage that it is more rigid and factors like robustness in design may be in the background (robustness in this sense means the following: If the parameters such as lamp post spacing, lamp post height, road width and road type is changed slightly — say less than 10% — the design should still be feasible). The computing time may increase considerably, as the number of multiplications increase substantially. This is due to the fact that intensities have to be divided by the square of the distances, and multiplied by cosines, and additionally, for every point on the road, the corresponding (C, γ)-values have to computed, and reflective values have to be looked up in tables.

1.2 Basic idea in approaching the problem

We define a function which "measures" the difference between the *desired output* and the *obtained output for a given arrangement(=setting of the block and beam type and orientation for each of the sources.* The arrangement of sources is then adjusted by a mathematical optimization procedure to make this difference as small as possible. In an ideal situation the desired output would be obtained, making the difference *zero*. In practice this can practically never be achieved.

The best output may be obtained with a large selection of beam types. This may be too varied for a production line. In such a case a (limited) number of "clusterings" should be identified — that is, a number of values for the prisms and spreaders such that every optimal choice lies reasonably close to one of the choices from this limited number.

Employing just this limited choice of beam types, the whole problem can be redone, with the orientation and placement of the block to be determined, as well as the orientation of each of the prisms. This will give an output which will (hopefully) still be sufficiently close to the desired output.

1.3 First approach: A given distribution

In this approach the IES-table of an existing luminaire is used as the target which is to be approached. Alternatively, an IES-table can be constructed by experts in the field, which should be excellent for the purpose of illuminating the road. For ease and economy of use in the further discussions, call this the *target-IES.*

Similiarly, for economy in style, call a choice of placement and orientation of the block, of beam types and their orientation, an *aiming.*

For each aiming the combined effect of all the individual sources will produce a new combined IES-table of intensities. Call this the *output-IES.*

The difference between the output-IES and the target-IES can be measured in different ways. One of the most effective means mathematically is to compute the difference between corresponding entries in the two tables, square each such entry, and numerically integrate the function represented by this. Call this integral over half of the unit sphere Q.

1.3.1 Detail: the IES-tables of the source's, I

For notation, we use the following convention for axes and angles fixed in space. These are the angles that will be used in the *target-IES*:

- The z-axis is vertical and fixed to the vertical post on which the luminaire is mounted.
- The x-axis is horizontal, parallel to the length of the road.
- The y-axis is horizontal, orthogonally across the road.
- The C-angle of a point P is measured in a horizontal plane from the xOz-plane to the plane zOP (through the point and the z-axis), with positive direction from the positive x- to the positive y-axis.
- The γ-angle is measured in the vertical plane formed by $C = \textit{constant}$, from the vertically downwards direction up to the line from O to the point P.

We also need the axes and a notation convention for the IES-table fixed to each source and the block itself.

- The Z-axis is orthogonal to the base of the block.

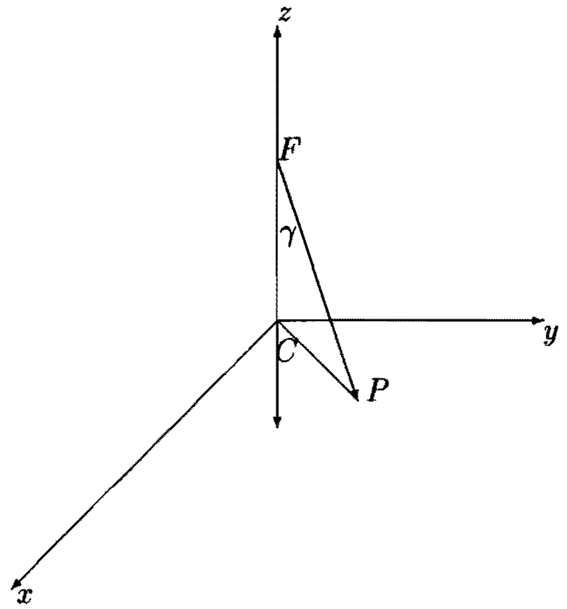


Figure 1.1: *Spatial axes and angles*

- The X-axis is along the longest edge of the block.
- The Y-axis is along the short edge of the block
- The $\hat{C}$-angle of intensities at a point P relative to the block is measured in a plane from the XOZ-plane to the plane ZOP (through the point and the Z-axis), with positive direction from the positive X- to the positive Y-axis.
- The $\hat{\gamma}$-angle is measured in the vertical plane formed by C = *constant*, from the negative Z-direction to the line from O to the point P.

Call *the number of sources utilised* n_s

The global orientation IES-table due to each individual source [*i.e.*, for the (C, γ)'s fixed in space], numbered from $i = 1$ up to $i = n_s$ depends on 5 factors:

1. the deflection angle of the prism, say δ_i [this is not entirely equivalent to some angular beam deflection, but is simply a means of characterizing one aspect of the type of beam],

2. the spreading of the beam, say σ_i, measured by some index from zero to some value, say n_f,

3. the angle between the longest edge of the block, the X-axis, and the direction in which the prism is orientated, say α_i,

4. the angle, say β, at which the whole block is tilted up from the horizontal. If its "rest" position is horizontal, with the longest edge aligned with the length of the road (the x-axis) and with the z-axis vertically upwards, then this angle is the rotation of the base of the block about the y-axis.

5. the rotation of the block about the lamp post, ϕ.

We will denote these individual IES-tables by

$$I(\delta_i, \sigma_i, \alpha_i, \beta, \phi)$$

and such a table consists of entries of the intensities for a range of $(\hat{C}, \hat{\gamma})$-values.

1.3.2 Detail: The target-IES, R

The target-IES has to be generated (or simulated) by the individual components. Call this table R (for *required*). Therefore, the effect of the individual sources each has to be computed — by means such as interpolation in table values — to obtain the effect in the positions corresponding to the target-IES.

This target has to be duplicated as far as possible by the combined effect of two blocks of n_s sources, mirror-symmetrically placed relative to yOz-axis, but effectively at the same position (the effect of the offset of blocks from the base being relatively negligable).

1.3.3 Detail: The objective Q

For the objective, we have to sum over the effects of all the individual beams. However, it is more economical to do it in the following order:

1. sum over the individual sources *relative to the block* for one of the two mirror-image-blocks; then

2. apply the tilt and rotation of the two blocks and compute the effect, and then

3. sum the contributions of the two blocks.

If we denote the process of applying the tilt and rotation by the symbol $Transform_{\beta,\phi}$, and the combined IES-table of all the source's by T, we can write the above as

$$T = \sum_{left,right} Transform_{\beta,\phi} \left[\sum_i I(\delta_i, \sigma, \alpha_i) \right] .$$

This constructs T as an IES-table of intensities (*i.e.*, a matrix) the table containing the intensities for different C and γ values.

With this notation we can define the objective function Q:

$$Q = \sum_C \sum_\gamma [R(C,\gamma) - T(C,\gamma)]^2 \, w_{C,\gamma}$$

where $w_{C,\gamma}$ is a weight function such that a good aproximation to the integral is obtained.

1.3.4 Detail: Minimizing the objective

There are numerous standard routines available for the minimization part of the procedure. The highly effective ones require information concerning the derivatives of the objective function relative to the variables. Such derivatives are not available here, and computing them numerically uses such a large number of function evaluations, that one can just as well use a slower method, not requiring derivatives.

An extremely robust and effective method not requiring derivatives is the method of *Nelder & Mead* (popularized by the *Amoeba*-routine from the *Numerical Recipes*-books). This method, albeit slow, are less sensitive — both to non-smoothness in the objective and to the objective being barely influenced by some parameters — than other comparable methods, and we therefore used it in this case.

In the first stages of solving the problem, when starting from scratch, a wild guess for the aimings can be made. We then optimize over a subclass of the parameters. The following phase is to use the settings obtained from the results of the first stage as starting values for the next stage. By repeating this procedure over different subclasses a viable set of optical elements are found. This set is used in the later runs of the program as starting values. Again, initially we only optimize over their placements, until a rough approximation to the desired objective is obtained. Only at this stage all other adjustments are considered.

1.4 Using the requirements of the application

We mention some ideas that may be applied to construct an ideal IES-table that can be used as target.

An IES table to use as target was formed by considering standard luminaires combinced with the experience of designers. A tool with which to construct this would be an advantage.

Such a tool can be similar to that fashion as used in the first and second approaches, by minimization.

In the first instance an objective function must be defined to be minimized. This should represent all factors demanded by the design or the standards organisation involved. [*This is the information as given in notes written by T van Hees and described fully in Chapter 5.*]

The vital question here is *what should the variables be over which we have to optimize?* To use all the C and γ-values would give a problem that would usually be too big to handle, and may give such a complicated and unrealistic IES-table, that this solution may be useless.

A more realistic approach will problably be to take a very sparse IES-table — that is, one that contains only a few C-values and a few γ-values, and use these as unconstrained variables over which we have to optimize. For C-values and γ-values greater than certain given values, we can assume the intensity to be zero.

The intensity for intermediate values of C and γ can be obtained by bilinear interpolation over neighbouring values. This interpolation would have the benefit that the "smoothing" effect of the interpolation procedure would approximate the design limitations in constructing a real luminaire with the required properties.

The output will then be a sparse IES-table, from which a more complete one can be obtained by the same interpolation procedure. This more complete one can be used as target for obtaining a simulation.

1.5 Second approach: Given CIE norms

In this approach the objective function is defined by directly implementing the requirements of the international standards organizations.

For each source-setting the *output-IES* can be obtained, the illumination on the road and the luminance in the direction of observers on the road can be determined.

The average illumination in the direction of observers must be sufficiently high, the quotient of the lowest luminance to highest luminance should not be too high, and the glare — defined in terms of the difference between the light from the luminaire and the light from the road — must be low enough.

All these factors can be combined in one objective function as is discussed fully in chapter 5.

Chapter 2

Intermediate prism- and spreader-values

In this chapter we consider a specific case of sources for which IES-tables for only a few prisms were available. In later investigations, such tables were constructed by experts in the field, and these procedures could be largely bypassed. However, the techniques of interpolation and treating the tilting of a luminaire as used here is applicable to all.

We will consider the first approach mentioned in the previous chapter here. The basic computational way of treating the problem remains the same for both the first and second approaches, so that this entails no loss in generality.

For some types of sources, IES-tables may be available for a large number of prism values. If the prism values are close enaough, tables for intermediate values may be obtained by interpolation between these tables.

In this chapter we will start by discussing ways of obtaining tables of light intensity distributions for any intermediate prism, if only the tables for measured values of 0, 10, 20, 30, 40 and 50 degrees are available. We discuss the background to such a way in the first section, and the transformation to coordinates that are needed in the second section. This is followed by an algorithm in the third section, and programs implementing it in the fourth.

We also consider any spreader of type between those for which measured data are available, *i.e.*, those with spreader we can designate as type 0 [ONTSP], and with spreader we can designate as type 1 [P232K*]. We also need tables for larger spreaders. This is done in Section 5 followed by an algorithm in Section 6 and programs in Section 7.

Lastly, in section 8, we give the method of constructing a large number of files of such tables to have avaliable for future use. Having these available speed up computations considerably.

For other optical systems, this large number may be constructed by ray-tracing routines directly.

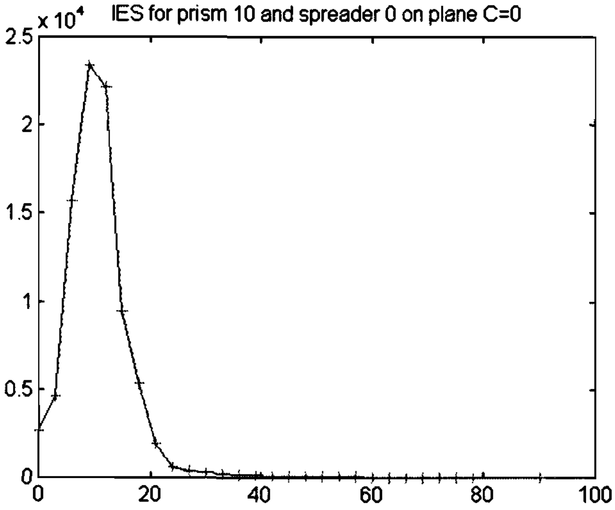
2.1 Intermediate prisms: Background

A critical aspect in the first stage of the optimization (before the "toolbox" of optical elements is fixed, *i.e.*, before we know definitely which types of prisms and spreaders will be used) is that values of the intensity for all intermediate values of the deflection angles of the prisms and of the spreaders should be available.

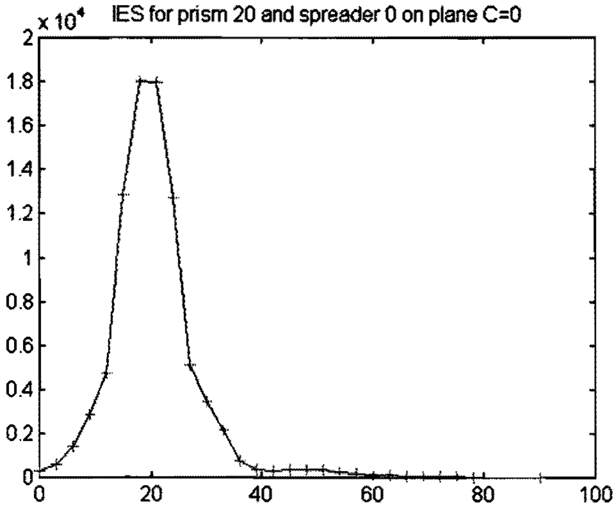
One way of doing this is by using the IES-tables directly. This method, of treating all input information of the individual luminaires by means of IES-tables, is feasible and reasonably fast.

However, we still need IES-tables for intermediate values of the deflection angle and the spreader. We will now discuss a method of obtaining these.

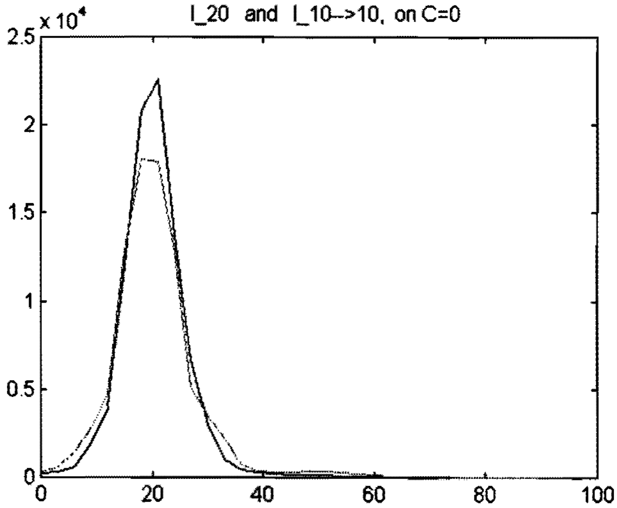
Consider the IES-tables for two prisms with 10 degrees between their deflection angles, say, for example, I_{10} and I_{20}. The following two graphs give the values in the IES-tables in the plane $C = 0°$, with the values sampled at 3 degree intervals along the γ-axis.



Along any ray out from the centre, close to the main plane of aiming, *i.e.*, a ray given by C = *constant*, with the constant not too far from 0, shift the intensities for the lower deflected prism, I_{10}, up by 10 degrees. This means that the new value at $\gamma = 10$ is taken to be the old value at $\gamma = 0$, according to the method as will be described in the following paragraph. A graph is then obtained with more or less the same form as that for I_{20}.



The following graph gives both I_{20} and the graph of I_{10} tilted up by 10°.

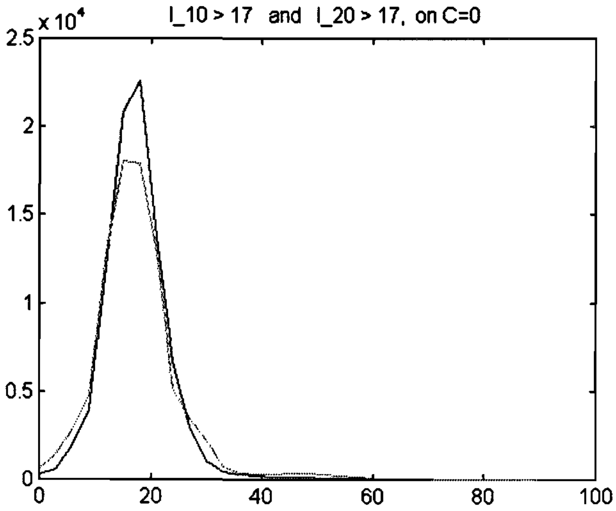


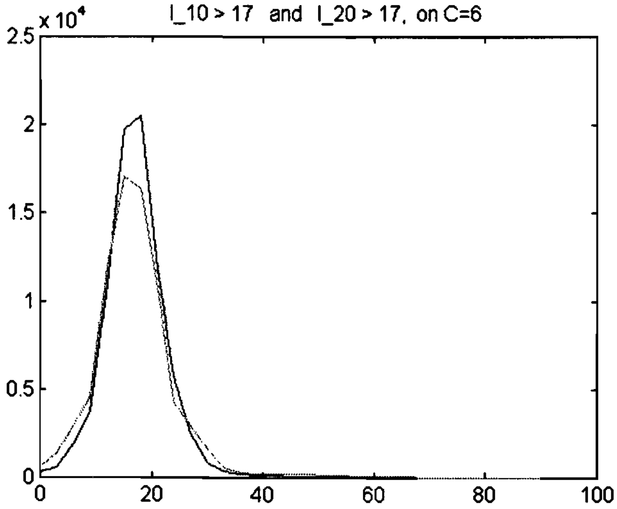
Because these graphs have approximately the same form, it is accurate to interpolate linearly between them to obtain the form for intermediate values.

Therefore, the following procedure can be used to obtain an IES-table for an intermediate prism value: Take as an example the case where we need the values in the spatial IES-table for a prism of 17 degrees. Tilt a source with a 10 degree prism up by 7 degrees, to obtain a new IES-table (relative to axes fixed in space).

Call this $I_{10\to17}$. Tilt a source with a 20 degree prism down by 3 degrees, to obtain a new IES-table, say $I_{20\to17}$.

The following graphs shows $I_{10\to17}$ and $I_{20\to17}$ along the planes $C = 0°$ and $C = 6°$





Interpolating between the values from the two shifted tables that we now have, *i.e.*, setting

$$I_{17} = \frac{3}{10} I_{10\to17} + \frac{7}{10} I_{20\to17}$$

should give very realistic values for a source with a prism deflecting 17 degrees. This statement about the accuracy can be made due to the fact that there is but a small difference between the values obtained from these two tables, and linear interpolation is therefore viable.

However, this entails obtaining (C, γ)-values for a reference frame fixed in space, from the given one fixed to the source after tilting it up (or down) by a small angle θ. The problem therefore boils down to this:

If a source (with given prism) is tilted up by θ, relative to the γ-axis, along the $C = 0°$ plane, and a direction with coordinates (C, γ) (fixed in space) is considered, what are the corresponding $(\hat{C}, \hat{\gamma})$-values (fixed to the source)?

We will consider this in the next section, and following that give an algorithm for the whole procedure. In the next chapter we will consider intermediate and extrapolated values for the spreading.

2.2 Transforming co-ordinates after a tilt

Consider the following figure, where the source has been tilted up by an angle θ. By this we mean that the direction represented by the plane $C = 0°$, with angle $\gamma = 0°$, relative to the luminaire, has now moved to a point A as given in the figure. Consider a point P in a direction given by co-ordinates (C, γ) as in the figure. Select the points to be at unit distance from F, so that the position vectors are as follows:

$$\begin{aligned} \mathbf{FA} &= (\sin\theta, 0, -\cos\theta) \\ \mathbf{FP} &= (\sin\gamma\cos C, \sin\gamma\sin C, -\cos\gamma) \end{aligned}$$

- For the angle $\hat{\gamma}$ relative to the source we have

$$\begin{aligned} \cos\hat{\gamma} &= \frac{\mathbf{FA}\cdot\mathbf{FP}}{FA.FP} \\ &= \mathbf{FA}\cdot\mathbf{FP} \\ &= \sin\theta\sin\gamma\cos C + \cos\theta\cos\gamma. \end{aligned}$$

The function `ATAN2(x,y)` usually computes faster than the `ACOS` or `ASIN`-functions. It may therefore help to have an expression for $\sin\hat{\gamma}$. To obtain this, consider the magnitude of the outer product between the two vectors,

$$\begin{aligned} \sin\hat{\gamma} &= \frac{\|\mathbf{FA}\times\mathbf{FP}\|}{FA.FP} \\ &= \|\mathbf{FA}\times\mathbf{FP}\| \end{aligned}$$

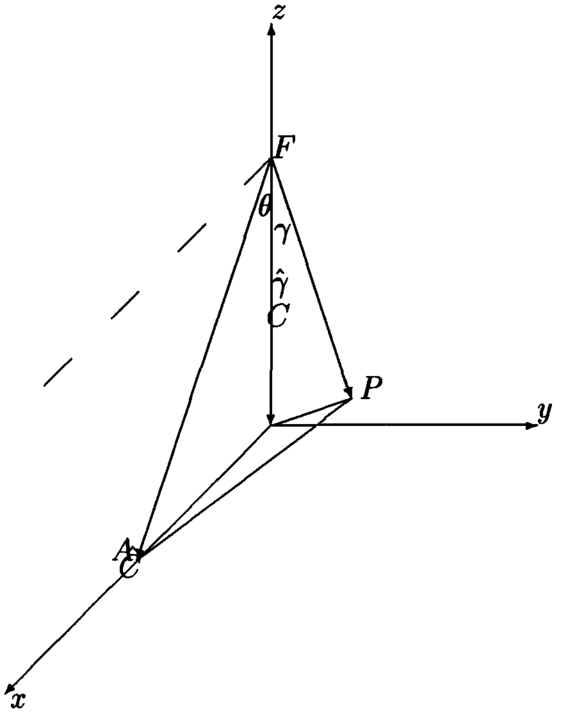


Figure 2.1: *Transforming from relative to fixed (C,γ)-values*

$$
\begin{aligned}
&= \left\| \begin{array}{ccc} \mathbf{i} & \mathbf{j} & \mathbf{k} \\ \sin\theta & 0 & -\cos\theta \\ \sin\gamma\cos C & \sin\gamma\sin C & -\cos\gamma \end{array} \right\| \\
&= \|(\cos\theta\sin\gamma\sin C, \\
&\qquad \sin\theta\cos\gamma - \cos\theta\sin\gamma\cos C, \\
&\qquad \sin\theta\sin\gamma\sin C)\|
\end{aligned}
$$

This can be shown to be equal to

$$\sqrt{\sin^2\gamma\sin^2 C + (\sin\theta\cos\gamma + \cos\theta\sin\gamma\cos C)^2}.$$

or

$$\sin\gamma = \sqrt{1 - (\sin\theta\sin\gamma\cos C + \cos\theta\cos\gamma)^2}$$

Therefore we have

$$\tan\hat{\gamma} = \frac{\sin\theta\sin\gamma\cos C + \cos\theta\cos\gamma.}{\sqrt{\sin^2\gamma\sin^2 C + (\sin\theta\cos\gamma + \cos\theta\sin\gamma\cos C)^2}}.$$

or

$$\tan\hat{\gamma} = \frac{\sin\theta\sin\gamma\cos C + \cos\theta\cos\gamma.}{\sqrt{1 - (\sin\theta\sin\gamma\cos C + \cos\theta\cos\gamma)^2}}.$$

- For $\hat{C}$ we have that it is the angle between the AFx-plane and the AFP-plane. To compute an angle between the planes, we need vectors normal to the planes, and we can compute the angles between these vectors.

  A unit normal on the AFP-plane is precisely that given above by the outer product between **FA** and **FP**, divided by its magnitude, call it **n**:

$$\begin{aligned}
\mathbf{n} &= \frac{\mathbf{FA}\times\mathbf{FP}}{\|\mathbf{FA}\times\mathbf{FP}\|} \\
&= \frac{(\cos\theta\sin\gamma\sin C, \sin\theta\cos\gamma - \cos\theta\sin\gamma\cos C, \sin\theta\sin\gamma\sin C)}{\sqrt{1-(\sin\theta\sin\gamma\cos C + \cos\theta\cos\gamma)^2}} \\
&= \frac{(\cos\theta\sin\gamma\sin C, \sin\theta\cos\gamma - \cos\theta\sin\gamma\cos C, \sin\theta\sin\gamma\sin C)}{\sqrt{1-\cos^2\hat{\gamma}}}
\end{aligned}$$

A vector orthogonal to the AFx-plane is given by **j** or by −**j**. By considering standard cases, it is evident that −**j** is the correct choice.

Taking the inner product with this unit normal to the FAx-plane gives

$$\cos\hat{C} = \frac{\cos\theta\sin\gamma\cos C - \sin\theta\cos\gamma}{\sqrt{1-\cos^2\hat{\gamma}}}$$

Using the formula $\sin^2 x = 1 - \cos^2 x$ we also obtain

$$\sin\hat{C} = \frac{\sin\gamma\sin C}{\sqrt{1-\cos^2\hat{\gamma}}}$$

and therefore

$$\tan\hat{C} = \frac{\sin\gamma\sin C}{\cos\theta\sin\gamma\cos C - \sin\theta\cos\gamma}$$

Notes

1. We are only going to consider angles θ less than 10°. For these angles, if $0° \leq \gamma \leq 90°$, then $0° \leq \hat{\gamma} \leq 100°$, which are in the range of values automatically generated by the ACOS-function. No corrective steps need to be taken here.

2. Consider a direction with $C = 0°$ and $\gamma > \theta$ In this case we must have $\hat{C} = 0°$ (and not 180°), and $\hat{\gamma} = \gamma - \theta$. In the expression above for $\tan C$ we have the enumerator zero, and the denominator positive, which gives $C = 0°$ and not 180°.

3. Consider a direction with $C = 0°$ and $\gamma < \theta$ In this case we must have $\hat{C} = 180°$ (and not 0 degrees), and $\hat{\gamma} = \theta - \gamma$. In the expression above for $\tan C$ we have the enumerator zero, and the denominator equal to $\sin(\gamma - \theta)$, which is negative. This means $C = 180°$ and not 0°.

4. Consider a direction with $C = 90°$ and γ any value less than 90 degrees. In this case we must have $\hat{C} > 90°$. In the expression above for $\tan C$ we have the enumerator equal to $\sin \gamma$, which is positive, and the denominator equal to $-\sin\theta\cos\gamma$, which is negative, giving the correct quadrant.

5. The matrices of cosines and sines, and even all the necessary combinations of the two can be computed once and saved, and the cosine and sine of θ need be determine only once for each complete IES-table, so that computing time can be considerably cut down.

2.3 An algorithm

Denote an IES-tables for a prism with deflection angle δ by I_δ .

```
Input: a prism-angle of $\delta$.

Wanted: an IES-table for a luminaire with the input prism.

1. Determine $\delta_k$ and $\delta_g$, divisible by 10, such that $\delta_k < \delta < \delta_g$.

2. Set $\delta_{op} = \delta - \delta_k$ and $\delta_{af} = \delta_g - \delta$.

3. Set $\theta = \delta_{op}$.
```

4. For all values of C and γ in the IES-table, determine
   (a) $\hat{\gamma} = \arccos(\sin\theta \sin\gamma \cos C + \cos\theta \cos\gamma)$
   (b) $\hat{C} = \arctan\left(\frac{\sin\gamma \sin C}{\cos\theta \sin\gamma \cos C - \sin\theta \cos\gamma}\right)$
   (c) $I_{op}^{temp}(C, \gamma) = I_{\delta_k}(\hat{C}, \hat{\gamma})$
5. Set $\theta = -\delta_{af}$.
6. For all values of C and γ in the IES-table, determine
   (a) $\hat{\gamma} = \arccos(\sin\theta \sin\gamma \cos C + \cos\theta \cos\gamma)$
   (b) $\hat{C} = \arctan\left(\frac{\sin\gamma \sin C}{\cos\theta \sin\gamma \cos C - \sin\theta \cos\gamma}\right)$
   (c) $I_{af}^{temp}(C, \gamma) = I_{\delta_g}(\hat{C}, \hat{\gamma})$
7. For all (C, γ) set

$$I_\delta = (\delta_{af} * I_{op}^{temp} + \delta_{op} * I_{af}^{temp})/10$$

2.4 A program for the algorithm

The algorithm given in the previous section can be programmed in MATLAB in the way as given in the following subsections.

In the first instance we give the main calling program with the prism angle as input and an IES-table a output. This program uses three subroutines, one to read the IES-files, one to compute the transformed angles (steps 4(a) and (b) and 6(a) and (b) in the algorithm) and the last one to compute the IES-tables from the known tables, by interpolating in the known tables (steps 4(c) and 6(c) in the algorithm).

Steps 1, 2, 3, 5, and 7 is done in the main calling program.

2.4.1 Main routine

```
FILENAME: e:\matlab\leds\iesdelta.m
----------------------------------

function [C_list,gam_list,idel,flux]=iesdelta(prishoek,spreiding);
% IESDELTA determines the IES-table for a source with a prism
% as given in the variable with which it is called, measured
% in degrees.
% The form of the program is:
% [C_list, gam_list, idel, flux]    = iesdelta(prishoek,spreiding);

if     (  ( abs(prishoek   ) < 0.05 ) ...
       | ( abs(prishoek-10) < 0.05 ) ...
       | ( abs(prishoek-20) < 0.05 ) ...
       | ( abs(prishoek-30) < 0.05 ) ...
       | ( abs(prishoek-40) < 0.05 ) ...
       | ( abs(prishoek-50) < 0.05 ) )
    del_on = round(prishoek);
    [C_list,gam_list,idel,flux] = leesies(spreiding,del_on,6,3);
else
    del_on = floor(prishoek/10)*10;
    del_op = prishoek - del_on;
    [C_list,gam_list, I_t,flux_op] = leesies(spreiding,del_on,6,3);
    [C_hat,gam_hat]   = intpris( del_op, C_list, gam_list);
    I_op        = illoutve( C_list, gam_list, I_t, C_hat, gam_hat);

    del_bo = del_on + 10;
    del_af = del_bo - prishoek;
    [C_list,gam_list, I_t,flux_af] = leesies(spreiding,del_bo,6,3);
    [C_hat,gam_hat]   = intpris(-del_af, C_list, gam_list);
    I_af        = illoutve( C_list, gam_list, I_t, C_hat, gam_hat);

    idel = (del_op *I_af    + del_af *I_op)/10;
    flux = (del_op *flux_af + del_af *flux_op)/10;

end
return
---------------------------------
```

2.4.2 Function: reading the IES-files

For the specific application discussed in this chapter, the following program can be used. However, as other type of optics became the main focus of the project, as discussed in the introduction of this chapter, this was superceded by versions for more general use.

```
FILEIAME: e:\matlab\leds\leesies.m
---------------------------------

function [C_list,gam_list,ies,flux] = leesies(sprei,prism,m,d)
% LEESIES reads a file named "ledS_PO.ies",
% with S equal to 0 or 1 and with P a multiple of 10, less
% than or equal to 50.
% Only every M'th C-value-row is kept in the table
% Only every d'th gamma-value is kept in each row

 if ((prism == 0|10|20|30|40|50) & (sprei == 0|1) )
```

```
  stringie=sprintf('led%1d_%1d0.ies',round(sprei),round(prism/10));
  fid = fopen(stringie,'rt');
   line = 'abcd';
   for i=1:12
      kar = '';
      while (abs(kar) ~= 10)
        kar = fscanf(fid,'%c',1);
        line = [ line(2:4) kar];
        if (line == 'FLUX' )
           flux = fscanf(fid,'%f',1);
        end;
      end;
   end

   a = fscanf(fid,'%d',10);
    scale = a(10);
    n_gam = a(4);
    n_C   = a(5);
   a = fscanf(fid,'%d',3);
   a = fscanf(fid,'%d',n_gam);
   gam_list = a(1:d:n_gam)';
   a = fscanf(fid,'%d',n_C);
   C_list = a(1:m:n_C);
   for i = 1:round(n_C/m)
      a = fscanf(fid,'%d',n_gam);
      ies(i , :) = a(1:d:n_gam)';    %Lees elke d-de gamma-waarde
      for j=1:(m-1)
         a = fscanf(fid,'%d',n_gam);
      end
   end
 fclose(fid);
  else
    'FOUT FOUT FOUT in formaat van _sprei_ of _prism_'
  end
return
---------------------------------
```

2.4.3 Function: spatial → source angles

```
FILENAME: e:\matlab\leds\intpris.m
---------------------------------

function [C_hat,gam_hat] = intpris(delta,C_list,gam_list)
% INTPRIS determines the (C,gam)-coordinates that corresponds
% to an IES-table IT, tilted upwards in the plane C=0 through
% an angle DELTA (in degrees), positive if tilted upwards.
% All data is given in degrees.
  r_d = pi/180;
        % DIVIDE radians by this to get DEGREES
  delta = delta*r_d;
  sind  = sin(delta);
  cosd  = cos(delta);
        % work with half the data to save time,and mirror it.
  n = floor(length(C_list)/2)+1;
  C      = C_list(1:n) * r_d ;
  gamma = gam_list    * r_d ;
  sing = sin(gamma);
  cosg = cos(gamma);
  cosC = cos(C);
  cosCsing = cosC*sing;
  sinCsing = sin(C)*sing;

  gam_hat= acos( (cosd*ones(size(C)))*cosg + sind*cosCsing)/r_d;

  C_hat  = atan2(              sinCsing ,                 ...
                cosd*cosCsing - (sind*ones(size(C)))*cosg)/r_d;

  gam_hat= [      gam_hat
            flipud(gam_hat( 2:(n-1), : ))];
  C_hat  = [         C_hat
           -flipud(  C_hat( 2:(n-1), : ))];
       % Add 360 degrees where C_hat is negative
  C_hat = C_hat + 359.99999*(C_hat<0) ;

return
----------------------------------
```

2.4.4 Interpolating in IES-tables

This can be done by employing the `interp2.m` program in MATLAB, after padding the input matrices and vectors so as to eliminate cases of values being out of bounds. As a functions like this is not available in FORTRAN, the following can be used.

```
FILENAME: e:\matlab\leds\illoutve.m

-----------------------------------
function lig = illoutve(C_list,gamma_list,i_tab,c,gam);
% ILLOUTVE computes the intensities for an IES-table for
% (C,gamma)-values between those in the given table, I_TAB.
% C_list is the list of C-values as used in  I_TAB
% gamma_list is the similar list of gamma-values
% I_TAB is the input IES-table for a measured source
% C is an input matrix of C-values
% GAM is an input-matrix of gamma-values
 n = length(C_list);
 C_list(n+1)  = 360;
 i_tab(n+1,:) = i_tab(1,:);
 m = length(gamma_list);
 for i = 1:n
  for j = 1:m
   if ( gam(i,j) >= max(gamma_list) )
      lig(i,j) = 0.0;
   else
      a        = find(C_list>c(i,j));
      i_ndex   = a(1)-1;
      i_ndexp1 = i_ndex+1;
      a        = find(gamma_list>gam(i,j));
      j_ndex   = a(1)-1;
      j_ndexp1 = j_ndex+1;
      t = (c(i,j)           - C_list(i_ndex) )/  ...
          (C_list(i_ndexp1) - C_list(i_ndex));
      e_t = 1.0-t;
      u = (gam(i,j)             - gamma_list(j_ndex))/  ...
          (gamma_list(j_ndexp1) - gamma_list(j_ndex));
      lig(i,j) =(1.0-u)*( e_t*i_tab(i_ndex   , j_ndex  ) ...
                         + t *i_tab(i_ndexp1 , j_ndex  ))...
                +(    u)*( e_t*i_tab(i_ndex   , j_ndexp1) ...
                         + t *i_tab(i_ndexp1 , j_ndexp1));
   end;  % if-else
  end    % for j
 end     % for i
return
-------------------------------------
```

2.5 Other spreader-values

As mentioned in the previous sections, we need IES-tables for other values of the spreaders than those that have been measured up to now. However, in most cases it has been found that as good results can be obtained using just the basic types. The question still remains how to determine which type to use where. In this project the complications of integer programming were avoided by initially interpolating, then rounding to the nearest integer, and reoptimizing using this. Doing this repeatedly gave very good results.

At this stage we have available the case of a spread of virtually zero, *i.e.*, by a simple transparant covering. Call this type 0, and denote an IES-table for this case by I_0. We also have a spreading that we can class as type 1, denoted by I_1. For a spreader between these two, designated by σ, it is quite simple to interpolate directly in the IES-tables:

$$I_\sigma(C,\gamma) = (1-\sigma)I_0(C,\gamma) + \sigma I_1(C,\gamma)$$

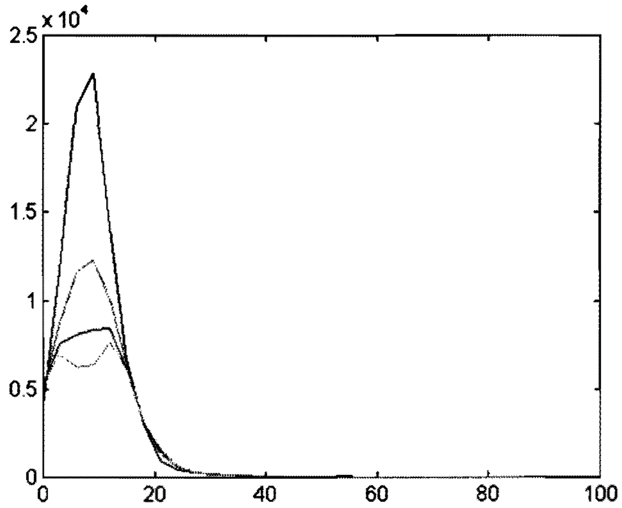
However, when we try to use the same formula for values larger than 1, we may have problems: If the value for I_0 at some point is double that of I_1 at the same point, and we take σ=2, this extrapolation gives a value of 0.

The algorithm as it stands above, is therefore only feasible for intermediate values of σ or for values slightly larger than 1.

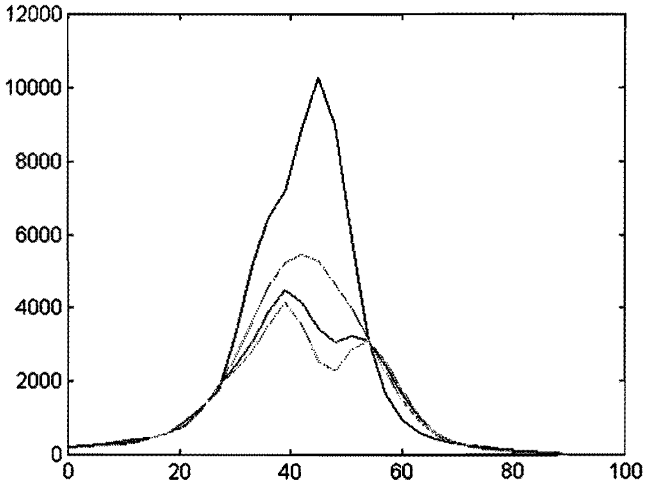
We can compensate for this to some extent in the following way: If the maximum of I_0 and the maximum of I_1 is first adjusted to give the same value at their maxima, in the second instance we extrapolate, and lastly normalize the values, it should give better results.

We can extrapolate between the total flux of the first and the total flux of the second, to get a value for the total flux of the required beam.

The following figure shows the values for a prism of 8 degrees and the values for the spreading 0 (the graph with the highest maximum in the figure) and the spreading 1, both as measured. The result of the extrapolation is shown in the two graphs with the smaller values.



That the process is not totally viable is shown in the following graph, the same type as in the previous figure, but now with a prism of 44 degrees.



Values with a spreading up to 1.5 or maybe even a bit closer to 2 may therefore still be usable using the above strategy, but for values of the spreading close to 2 and higher more measurements are needed to devise an accurate method. *We only used values up to 1.5 when employing the above method.*

2.6 Algorithm: IES-tables for given spreaders

Denote an IES-tables for a prism with deflection angle δ and a spreading of σ by $I_{\delta,\sigma}$.

Input: a prism-angle of δ and a spreading of σ.

Wanted: an IES-table for a luminaire with the given prism and given spreading.

1. Determine an IES-table, $I_{\delta,0}$,
   and a flux, $F_{\delta,0}$ for a source with spreading 0
   and prism δ according to the algorithm of the previous chapter.

2. Determine an IES-table, $I_{\delta,1}$,
   and a flux, $F_{\delta,0}$, for a source with spreading 1
   and prism δ according to the algorithm of the previous chapter.

3. Divide both IES-tables by their maxima.

4. Interpolate/Extrapolate the tables according to

$$I_{\delta,\sigma} = (1-\sigma)I_{\delta,0} + \sigma I_{\delta,1}$$

5. Interpolate/Extrapolate the fluxes according to

$$F_{\delta,\sigma} = (1-\sigma)F_{\delta,0} + \sigma F_{\delta,1}$$

6. Normalize the IES-table $I_{\delta,\sigma}$ so that the total flux, computed from these values, is 1000.

2.7 Program: IES-tables for other spreaders

```
FILENAME: e:\matlab\leds\idelsig.m
----------------------------------


function [C_list, gam_list, flux, ids] = idelsig(prisma,spreiding)
% IDELSIG  in the form
% [C_list, gam_list, flux, ids] = idelsig(prisma,spreiding)
% computes the IES-table for a source with a
% prism with deflection angle PRISMA and spreading
% given by SPREIDING, on a scale from 0 to 3.

if     ( abs(spreiding    ) < 0.005 )
    [C_list, gam_list, ids, flux] = iesdelta(prisma,0);
elseif ( abs(spreiding - 1) < 0.005 )
    [C_list, gam_list, ids, flux] = iesdelta(prisma,1);
elseif (     spreiding      < 0     )
    [C_list, gam_list, ids, flux] = iesdelta(prisma,1);
    deler = 100*abs(spreiding-0.01);
    idel = I_t/deler;
    flux = flux/deler;
elseif (     spreiding      > 3     )
    [C_list, gam_list, ids, flux] = iesdelta(prisma,1);
    deler = 1000*abs(spreiding - 3 + 0.01);
    idel = I_t/deler;
    flux = flux/deler;
else
    [C_list, gam_list, I_00, flux_0] = iesdelta(prisma,0);
    I_0 = I_00/max(max(I_00));

    [C_list, gam_list, I_11, flux_1] = iesdelta(prisma,1);
    I_1 = I_11/max(max(I_11));

    flux  = (1-spreiding)*flux_0 + spreiding*flux_1;

    ids = (1-spreiding)*I_0 + spreiding*I_1;

    lig = fluxies(C_list, gam_list,ids);

    ids = ids*1000/lig ;

end;
return;
----------------------------------
```

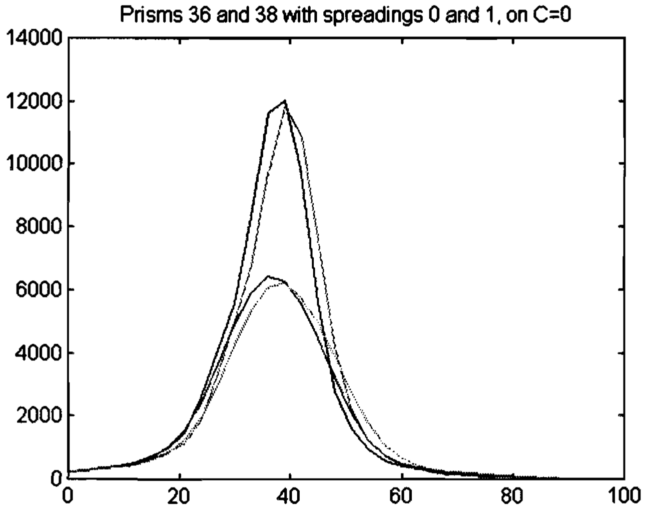
In the above program we use a program `fluxies` to compute the total flux. This is the surface integral of the function that is represented by an IES-table. The surface-integral is computed in two ways and the average taken over the two values: once by evaluating the conversion-factor for the inclined surface as part of the function, once by integrating it exactly over each patch, *i.e.*, each subsection of the surface of the unit ball, and using the average of the IES-values at the four corners of the patch as the value.

```
FILENAME: e:\matlab\leds\fluxies.m
---------------------------------

function lig = fluxies(C_list, gamma_list,i_tab);
% FLUXIES computes the surface integral for an IES-table
% by a combined formula( the  (n+1)st C taken as 360)
%
 n   = length(C_list);
 np1 = n+1;
 gamma_list = gamma_list/180*pi;
 deltaC       = (C_list(2) - C_list(1))*pi/180;
 i_tab(np1,:) = i_tab(1,:);
 m   = length(gamma_list);
 mm1 = m-1 ;
 deltag       = gamma_list(2:m) - gamma_list(1:mm1);
 lig1 = (sum( ...
  sum( (i_tab(1:n,1:mm1) + i_tab(2:np1,1:mm1))) .* ...
  ( sin(gamma_list(1:mm1)).*deltag) ) ...
   + ...
       sum( ...
  sum( (i_tab(1:n, 2:m ) + i_tab(2:np1, 2:m ))) .* ...
  ( sin(gamma_list(2:m  )).*deltag) )  )/4*deltaC;

 mat =  i_tab(1:n,1:mm1) + i_tab(2:np1,1:mm1) ...
      + i_tab(1:n, 2:m ) + i_tab(2:np1, 2:m ) ;
 deel2 = cos(gamma_list(1:mm1)) - cos(gamma_list(2:m));

 lig2 = sum( sum(mat) .* deel2 ) /4*deltaC;

 lig = (lig1+lig2)/2;
return;
---------------------------------
```

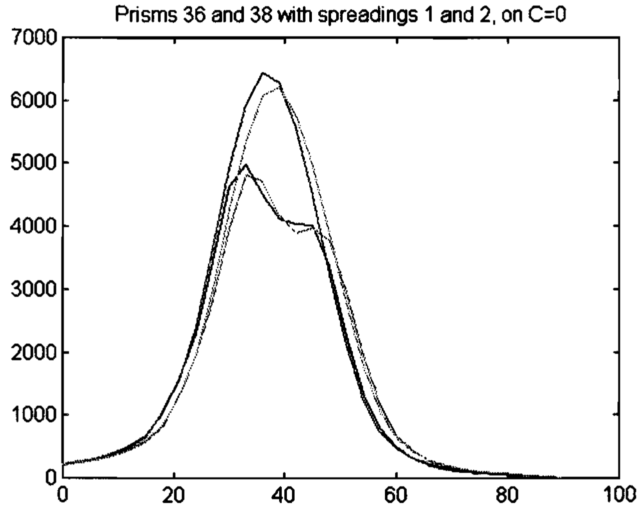
2.8 Constructing a set of data-files

The methods discussed above, in particular the transformations between sets of angles, is just too slow to use effectively in an optimization-program. However, if tables were available for prisms for which the spacing between two succesive prisms is smalle enough (say approximately 2 degrees) a direct interpolation for values between those for these tables should give more than sufficiently accurate values. Therefore we need tables for spacings of about 2 degrees between prisms, and for types 0, 1, 2 and 3 for spreaders.

If we want an IES-table for a value of, for example 38.7 degrees for the prism and 0.6 for the spreader, we would interpolate between values of 38 and 40 for the prisms, and for each between spreadings 0 and 1. In the following figure we show the values along the plane $C = 0$ for a source with prism 36, spreading 0; prism 36, spreading 1; prism 38, spreading 0 and prism 38 spreading 1.



The following figure shows the same type of information as the previous one, but now for the cases of the spreading equal to 1 (for prisms 36 and 38) and the spreading equal to 2 (for prisms 36 and 38).



The following program achieves this, and writes all the data to files containing the flux, the number of C-values and γ-values, the list of C-values and γ-values used, and the matrix of intensities.

```
FILENAME: e:\matlab\leds\leeskryf.m
-----------------------------------

for sprei=(0:1:3)
   for prism=0:2:50
     [C_list,gam_list,flux,ids] = idelsig(prism,sprei);
     sukses = skryfies(C_list,gam_list,ids,flux,prism,sprei);
   end
end
-----------------------------------
```

The above program uses a routine to write to the relevant files:

```
FILENAME: e:matlab\leds\skryfies.m
----------------------------------

function  sukses = skryfies(C_list,gam_list,ies,flux,prism,sprei)
% SKRYFIES writes the IES-info to a file named "ledS_PO.m",
% with S equal to 0, 1, 2 or 3 and with
% P = 00 02 04 06 08 10 12 . . . 46 48 50

 if (sprei == 0|1|2|3)
   if ( (prism >= 0)&( prism <= 9) )
      stringie = sprintf('led%1d_0%1d.m',round(sprei),round(prism))
   elseif ( (prism >= 10)&( prism <= 50) )
      stringie = sprintf('led%1d_%2d.m',round(sprei),round(prism))
   else
        ' FOUT in PRISM-waardes '
   end;

   fid = fopen(stringie,'wt');
   fprintf( fid, '%6.4f\n',flux);
   fprintf( fid, '%4d    %4d\n', length(C_list) , length(gam_list) );
   fprintf( fid, '%6d ', C_list);
   fprintf( fid, '\n');
   fprintf( fid, '%6d ', gam_list);
   fprintf( fid, '\n');
   for i = 1:length(C_list)
      for j=1:length(gam_list)
         fprintf( fid, '%6d ', round(ies(i,j)) );
      end;
      fprintf( fid, '\n');
   end;
   fclose(fid);

 else
    'FOUT FOUT FOUT in formaat van _sprei_ of _prism_'
 end
 sukses=flux;
return
----------------------------------
```

Chapter 3

Adding the effects of numerous sources

Consider the following installation: a number, say n_s, of sources, S_i, with $i = 1, 2, \dots, n_s$, are placed on a flat block. All sources are oriented in such a way that their beams are orthogonal to the block. Each source is fitted with a prism, deflection angle δ_i, and a spreader, spreading σ_i. The direction in which the prism deflects it, *i.e.*, the angle that the projection of the main plane of the beam on the block forms with a given line on the block, the x-axis, is called α_i.

The total effect of all these sources has to be determined. We will first of all start by considering all other effects except for the tilting and rotation of the block itself, and discuss this in the last sections of this chapter.

3.1 Intermediate prisms and spreaders: fast

All 150 data files for prism-values with increments of 2 degrees from 0 to 50 (giving 26 types), and spreader values of 0, 1, 2 and 3 (giving 4 types, and thus, in total, 104 files) can be read in at once. As all basic file types are mirror-symmetric, with the part from 0 to −180 degrees a copy of that from 0 to 180 degrees, only one line more than half of the information need to be stored. In FORTRAN 90 the tables are stored as an array, dimensioned (1:25, 1:4, 1:37, 1:31), and thus containing 114 700 numbers. In MATLAB we store it as one giant matrix. For each type of prism, 0, 2, …, 50, we store the information in rows 1:37, 38:74, ... (25*37+1):(26*37). The whole series, with spreader type 0 is stored in the first 31 columns. Spreader type 1 in the next 31 columns, etc.

If narrower beams are used, the structure can be adapted to use intervals of 5 degrees for the horizontal angle, and every 3 degrees for the vertical angle.

3.1.1 A program for reading the data into memory

All data matrices with IES information are declared global in the program, to be able to share the information with other routines.

```
FILENAME: c:\matlab\leds\ton\getsors.m
-----------------------------------
disp(['Starting to read the data files named ' fullname1]);

SORS_IES = zeros(25*37,31*numberoffilesets);
SORS_FLX = zeros(25,numberoffilesets);
indexset = [1:31] ;
tic
leestyp =['leesmike'
          'leespash'];

for jj=1:numberoffilesets
  eval(['fullname = fullname' int2str(jj)])
  for i = 0:2:8
       disp([num2str(i) setstr(176)]);
       indekse = round(i/2)*37 + [1:37];
       stringie = [fullname '0' num2str(i) '.ies'];
       eval(['[ies, flux] = ' leestyp(pash+1,:) '(stringie);'])
       SORS_IES(indekse, (jj-1)*31+indexset) = round(ies);
       SORS_FLX(round(i/2)+1, jj) = flux;
  end;
  for i = 10:2:deflctoflastfile
       disp([num2str(i) setstr(176)]);
       indekse = round(i/2)*37 + [1:37];
       stringie = [fullname num2str(i) '.ies'];
       eval(['[ies, flux] = ' leestyp(pash+1,:) '(stringie);'])
       SORS_IES(indekse,(jj-1)*31+indexset) = round(ies);
       SORS_FLX(round(i/2)+1,jj) = flux;
  end;
end

disp(['.....  It took ' num2str(toc) ' seconds for all']);

C_list   = [0:5:355]';
gam_list = [0:3:90];
-----------------------------------
```

For listing of the subroutines used in the above, see the programs `leesmike.m`, `leespash.m` and `leesies.m`, in REPORT IWDE 97-06: Section 14, p36.

The last statement in this program calls a function for doing the rotation of the prism relative to the block. In the following section we discuss a way of doing this, and give a listing of a program performing this function.

3.2 Rotating a prism on a source and adding

We consider the total effect relative to angles C and γ fixed to the block (before tilting and rotating they are the same as the spatial angles). With these angles the effect of a source with IES-table $I_{\delta,\sigma}$, placement α, at a point with block-angles (C, γ), is the same as that at a point $(C - \alpha, \gamma)$ relative to the source:

$$I_{\delta,\sigma,\alpha}(C,\gamma) = I_{\delta,\sigma}(C-\alpha,\gamma).$$

We therefore only need to be able to do interpolation between the C-planes to accomodate for these placements.

With this available, the total intensity-table for the combined sources now becomes

$$I_{total} = \sum_{i=1}^{N} I_{\delta_i,\sigma_i,\alpha_i}$$

An algorithm for the C-interpolation is the following

Input: `An IES-table` I `and an angle` α

Wanted: `An IES-table` I_α `formed by rotating` I `through` α

```
1. Set $C_w = C - \alpha$.
2. Determine $C_k$ and $C_g$ from the list of $C$-values used in the
   table such that $C_k < C_w < C_g$
3. For all $C$ and all $\gamma$ used in the table, set
```

$$I_\alpha(C,\gamma) = \frac{(C_g - C_w)I(C_k,\gamma) + (C_w - C_k)I(C_g,\gamma)}{(C_g - C_k)}$$

3.2.1 A program for the α-rotation

```
FILENAME: e:\matlab\leds\frotalf.m

-----------------------------------
function i_nuut = frotalf( i_tab, alpha);
% FROTALF frotalf(  i_tab, alpha) rotates a
% given source, i_tab, through "alpha"
% I_TAB is the input IES-table.
 global C_list;
 while alpha >= 360
    alpha=alpha-360;
 end
 while alpha < 0
    alpha = alpha+360;
```

```
end
C_t = C_list;
n = length(C_t);
C_t(n+1)  = 360;
a         = find( C_t > alpha );
i_ndex    = a(1)-1;
C_k       = C_t(i_ndex  );
i_ndexp1 = i_ndex+1;
C_g       = C_t(i_ndexp1);
ind_on = [ ((n - i_ndex   + 2):n ) (1 : (n - i_ndex   + 1) ) ];
ind_op = [ ((n - i_ndexp1 + 2):n ) (1 : (n - i_ndexp1 + 1) ) ];
t      = (  alpha - C_k )/(   C_g  - C_k );
e_t    = 1.0-t;
i_nuut =   e_t * i_tab(ind_on , : ) + t * i_tab(ind_op , : );
----------------------------------
```

3.2.2 A program for adding the sources

The program `make_hrb.m` reads in the data and adds the intensities of all contributing sources to obtain one IES-table representing the combined effect of all.

When the combined effects of all the sources on a block has been added, and a table for the combined effect been computed, we need to consider the effect of a tilt of the block up from the horizontal plane, followed by a rotation about a vertical axis.

One way of achieving this, is to use Part 4 of the algorithm of Section 2.3 for the tilt, followed by the algorithm of Section 3.2.1 for the rotation.

This is done by the program `total_rb.m` which tilts and rotates the block.

Mostly the installation consists out of two blocks, each containing numerous sources. When one is tilted up, the other is tilted up, and when the first is rotated into the road (counterclockwise as seen from above) from the plane $C = 0°$, the other is rotated into the road (clockwise) from the plane $C = 180°$.

Because the whole setup is thus mirrored, the values have to be reflected in the matrix as well, in particular the values where $C = 90°$ remain the same, the values of the original at $C = \theta$ go to $C = 180^{c}irc - \theta$.

Adding the mirror image is also done in the program `total_rb.m`, depending on whether a global variable `symmetrical` is set to 1 or not.

Listings of the above is given Section 11 of REPORT IWDE 97-06: Section 14, p36.

In the next chapter we will discuss how a function can be created from the above building blocks that would measure the distance between the intensity distribution obtained and the one that is desired.

Chapter 4

Measuring the difference and optimizing

In this chapter we start off by defining a measure for the difference between the target IES-table and the IES-table obtained for a given setup.

In Section 2 we discuss an optimization method, and the way in which initial values for the optimization process can be obtained and refined. The result of this procedure is discussed in Section 3 and some other options in Section 4.

4.1 Defining a measure

Three measures for the difference between a model and desired output are in common use.

The first is the L_1-measure, defined as the sum of the absolute values of the differences. This measure is insensitive to a few points which may be totally different. This is an advantage in cases where the data may contain some errors.

A second measure is the L_∞-measure (also called the Chebycheff measure), defined as the largest absolute difference. This is totally dominated by the largest difference. If the largest difference can be made relatively small, the fact that there may be just as large deviations over the rest of the area do not affect the measure. As we want not only the largest deviation to be small, but also the all-over fluctuations to be small, this was rejected.

The third, the L_2-measure, defined as the sum of the squares of the differences, falls between the two methods mentioned above.

We employed a weighed L_2-measure. We took the square of the differences of the two IES-tables (considered as matrices). We then multiplied this table, representing a function of (C, γ) by $\sin(\gamma + \frac{1}{2}\delta\gamma)$ (with $\delta\gamma$ representing the difference between two values in the table), and added all the obtained values. This gives a fairly good approximation to the surface integral of the function over half of the unit sphere:

$$\begin{aligned} d &= I_{obtained} - I_{target} \\ D &= (d_+) + (d_-)^2 \\ f &= \sum_i \sum_j D_{i,j} \sin \gamma_j \end{aligned}$$

where i ranges over all the C-values and j over all the γ-values, and the + and −-subscripts denote the positive and negative values of the matrices obtained.

In the function `foldback` all values of the variables are "folded back" at 0 and at a specified value, for example 45. By this we mean that the value at, *e.g.* −1.2 is taken to be the value at 1.2, and the value at 52.5 is taken to be $45 + 45 - 52.5 = 37.5$. This has the effect that no constrained minimization technique need be used, and that variable choices with less than desired efficiency are avoided.

The total flux of all the sources is normalised to agree with the flux of the source. This done for two reasons. In the first instance we want to approximate the light *distribution* as closely as possible. In the second instance we do not as yet have accurate values for the fluxes of the sources, and no value for the flux of the target distribution available.

The last part of the program draws graphs of the results obtained every so often.

As explained in Chapter 5, the objective was later changed to consider the parameters on the road at later stages, instead of considering mainly the target.

The basic function to be optimized, `func.m` is listed fully in REPORT IWDE 97-06, Section 11.

4.2 Optimizing

The objective function can be optimised over the tilt, or the rotation, or the prism selections, or the prism rotations, or the spreading, or any combination of the above. This is achieved by defining the variable `strategy` as discussed in REPORT: IWDE 97-06, section 9.

However, each new stage in this strategy begins with the values obtained from the previous stage and allow minimization over a slightly different combination of factors.

These steps, or at least some of them, should be repetitions of previous steps, to make sure that we are at the minimum.

The values of the other variables — *i.e.*, of the prisms and rotations of the prisms for the sources over which we are not optimizing — is obtained from the global varable `setup`. This would usually contain the best settings obtained at any stage for all variables.

The optimization method of Nelder & Mead is used. As mentioned in the first chapter, this method is very robust in the sense of abnormal termination occurring very seldom. Because of the simplicity in use and programming, it is easy to imbed controls and tests and various substrategies in the method itself.

A complete listing of the whole optimization routine to minimize the function `func` over a number of variables is given in Section 15 of REPORT IWDE 97-06.

The last step when exiting the program, namely `yres=func(p(iLaag,:))`, serves the purpose of defining the variable `setup` to contain the best setup achieved up to now. This can then serve as start value for a following run.

In order to use this program, all the data that the subroutines in `func.m` use, need to be read in and declared global. This is done by the program `mrbinit.m`.

The program `optimis.m` is then run. This defines the stage of the procedure, initialises the variables `setup`, `tilt` and `rot`, which is used by `func`, to the starting values. The minimizing routine is called repeatedly to minimize over the different sets as defined in the programs mentioned above. After every run the best setup is kept and a random shakeup, over the variables over which the program is to minimize, is done. The Nelder & Mead method is not always allowed to run its full course: the maximum number of steps allowed is artificially kept lower than usual, so that a restart can be made. The method always starts with one point more than the number over which we are minimizing. The first is one that was kept, the one that gave the present minimum. The other each contains one coordinate which represent a (large) random adjustment to one of the coordinates of the minimum point.

By repeating the above procedure quite a number of times, a strategy resembling simulated annealing is combined with the analytic prowess of the Nelder & Mead method.

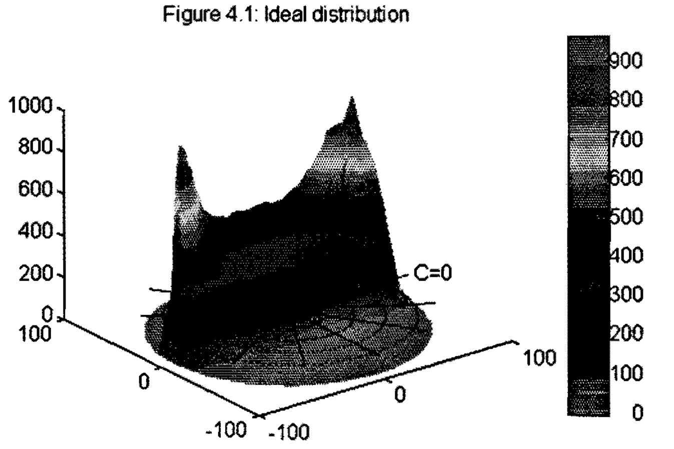


Figure 4.1: The target distribution

4.3 Results

From the first results it became clear what the problems are, what the physical limiatations are, and in what direction the programs and the source types could be improved. We discuss these results here.

As the different stages of the program was run, it became clear that the main problem in obtaining the target distribution, was in obtaining a sharp enough transit between the point of highest intensity and the points with $\gamma > 80°$, where the intensity drops to 0. A sharper transit could be obtained with less spreading of the sources. Therefore, all the spreadings were readjusted to 0, and the program restarted.

In the first figure, Figure 4.1, the target distribution is shown.

The distribution in the following figure (4.2) was the best approximation to the target distribution obtained.

The difference between the target distribution and the best is shown in Figure 4.3.

From the above graphs it is apparent that the sharp transition in the target distribution from maximum to zero in the neighbourhood of $C = 24$, $\gamma = 78$ *could not be obtained* with the first set of multiple sources. By allowing the sources illuminating the area far from the maximum to have spreadings of value 1, thereby possibly being able to focus the remainder more precisely on the maximum (as they are then not needed to help make the illumination uniform over other areas), a slight improvement could be obtained, less than 0.3%.

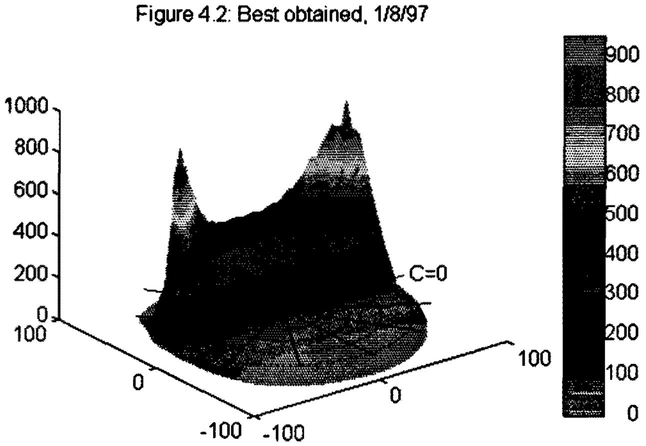


Figure 4.2: A good approximation to the target by multisources

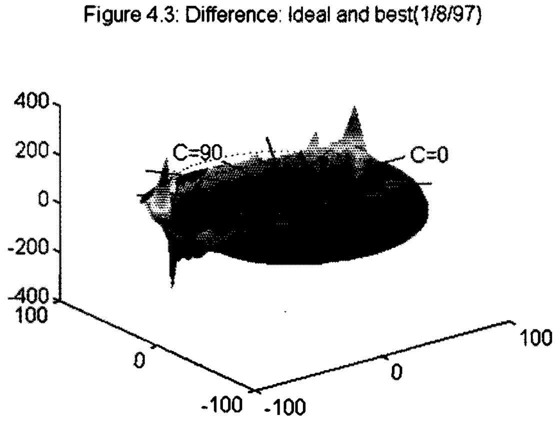


Figure 4.3: Difference between target and 'best' approximation

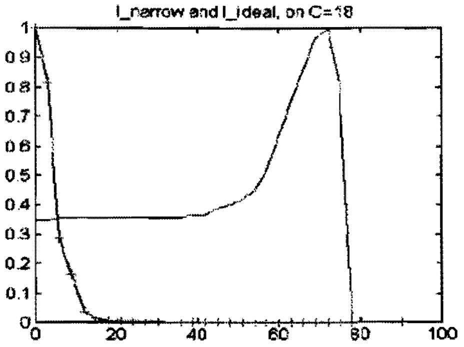


Figure 4.4: The slope of the target and the narrowest beam

The reason why no better result *can* be obtained with the present optical system becomes clear if the slope of the target in the neigbourhood of 78 degrees is compared with the slope of the narrowest beam (on the plane $C = 18$ degrees, as in figure 4.4:

It is clear that some kind of shielding or narrowing of the beam is needed to obtain the sharp cutoff that is demanded in the target beam.

That this is the main factor limiting a better approximation of the target distribution is shown in the graphs in figure 4.5. The best obtained with 72 sources is compared with the target on different C-planes. It is immediately clear that any effort to improve the illumination in the region of the maximum will be offset by the increase of light just beyond the maximum. This difference influences the objective function, the measure of fit more than all other discrepancies of the fit.

Due to the above we started using the L_1-measure where we have too much light, and the L_2-measure where we have too little. This gives additional emphasis to the under illuminated regions. In the end, as the focus shifted to the effect on the road, rather than the target distribution, this was not implemented in the final programs. There the target is basically just used for obtaining a preliminary aiming.

This produced a light distribution — allowing all sources to have different prisms — as illustrated in the following figure. We show the value of the target as a dotted line and that of the best obtained as a solid line, on the circles $\gamma = constant$, that is, for the whole set of values of C from 0° to 360°.

Grouping the prisms was done by the procedure as given in Section 12 of REPORT IWDE 97-06. This groups the variables into the specified number of

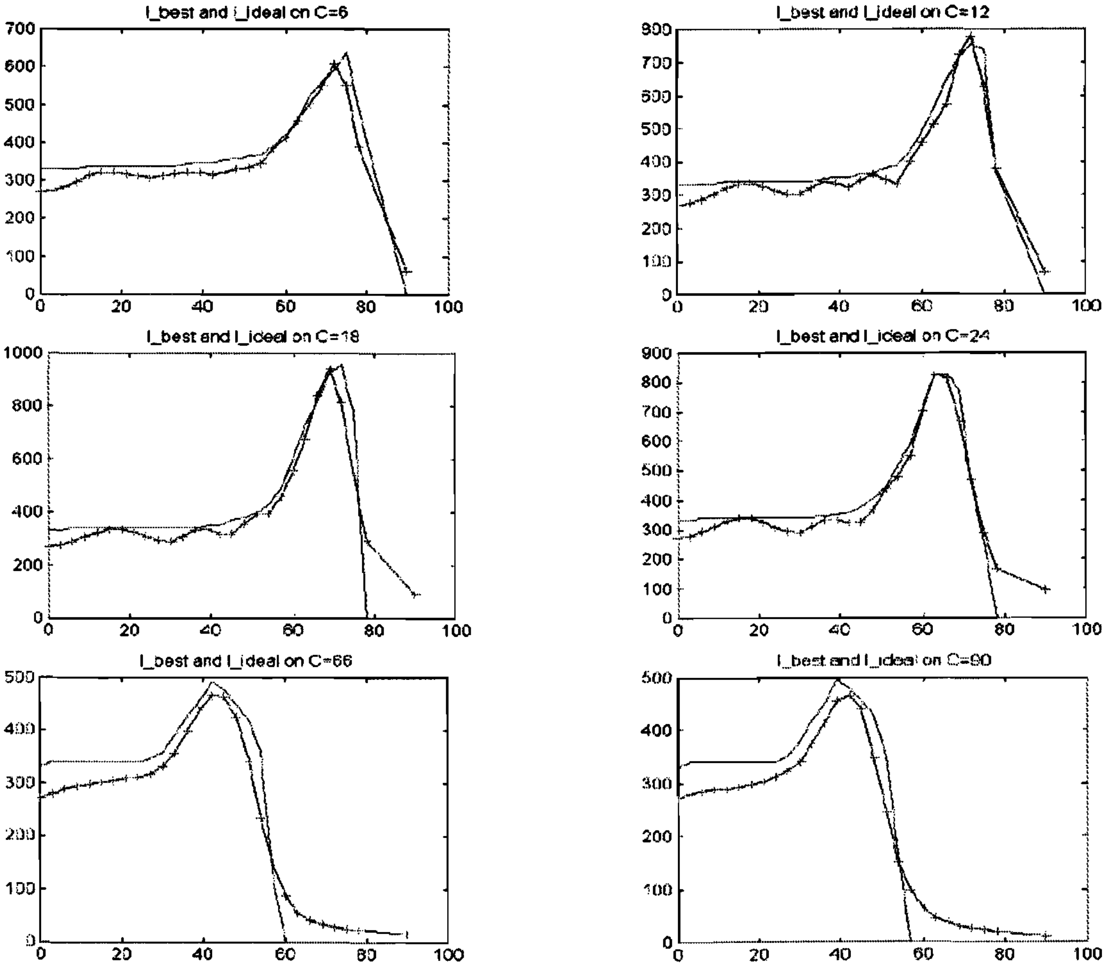


Figure 4.5: Differences between the slope of target and multisources

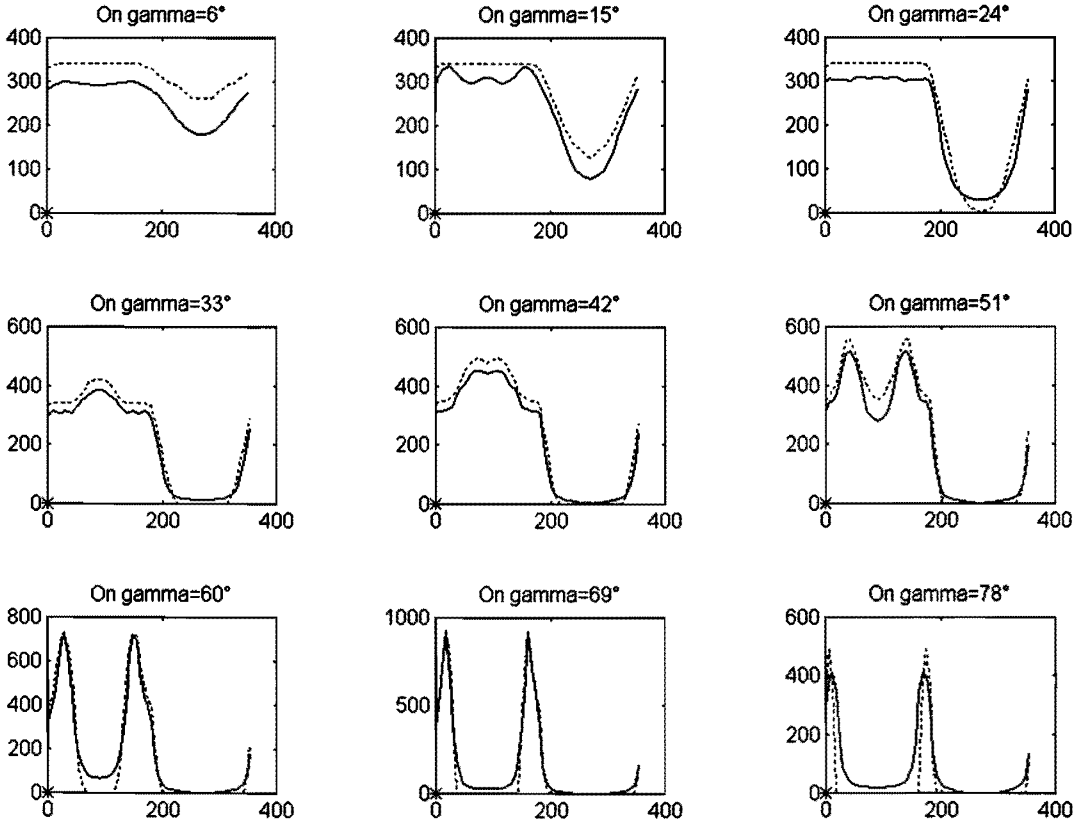


Figure 4.6: A better approximation by screening some light

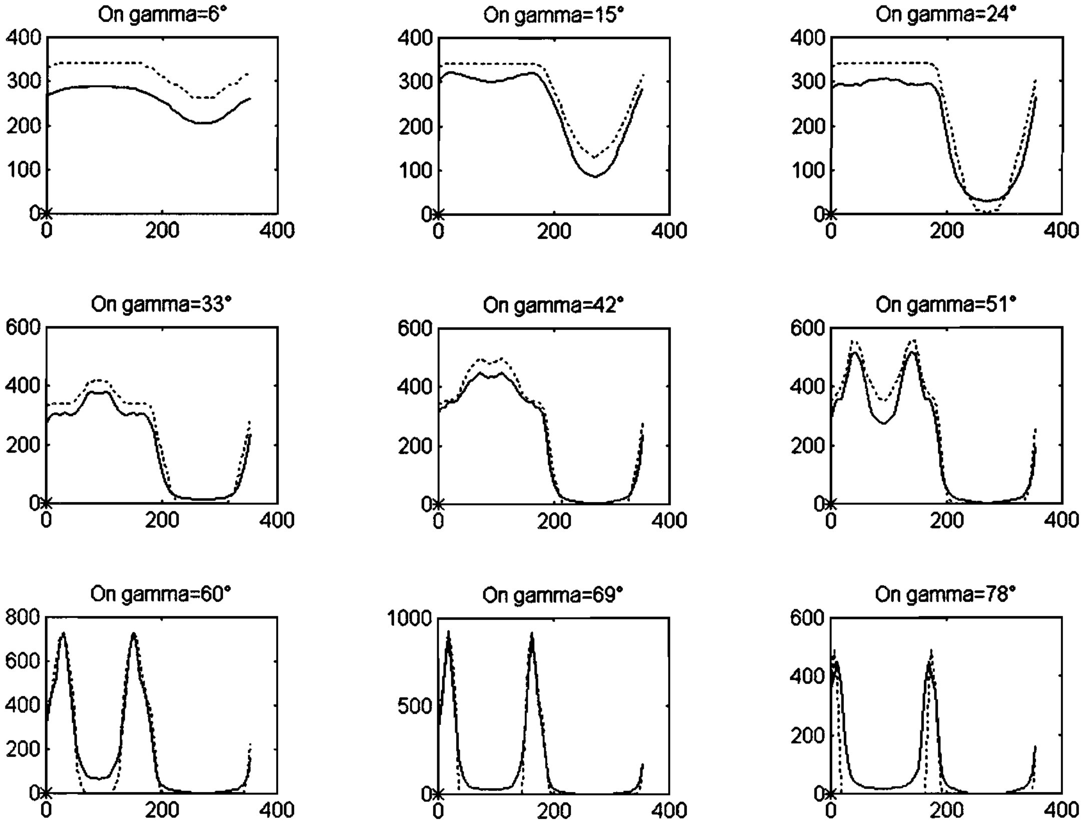


Figure 4.7: A good approximation to the target: grouped setup

intervals, and takes the average of the group in each interval. It makes sure that enough points would always remain to allow the full variation of the number of groups.

Grouping to use only 6 prisms, and the re-optimizing over the choice of these prisms and over the placements, worsened the measure of the fit by ±13%. We considered this to be somewhat too much, and to concentrate on using 7 prisms.

Grouping to use 7 prisms, and re-optimizing over this choice and the placements, worsened the measure of fit by a very small percentage.

Figure 4.7 illustrates the deviation between the best achieved now and the target light intensity distribution.

4.4 The design with 36 sources per block

The same type of strategy was run through with blocks containing 36 leds. The results are shown in the following figures. Using 36 sources mean that we can in effect use 72 sources, grouped into 36 twinnned pairs, each twin having the exact same optical structure. This will have the effect that variations in the quality of the sources will be compensated for by pairing the weakest source with the best; the second weakest with the second best, etc.

In figure 4.8 the differences on different circles $\gamma = constant$ are shown, while in the second we show the differences on different planes $C = constant$.

In the next chapter we consider the case of considering the effect of the road directly, and defining an objective based on that, rather than concentrating on the "ideal" distribution used above: Maybe we can achieve the desired effect on the road without approximation the give IES table accurately.

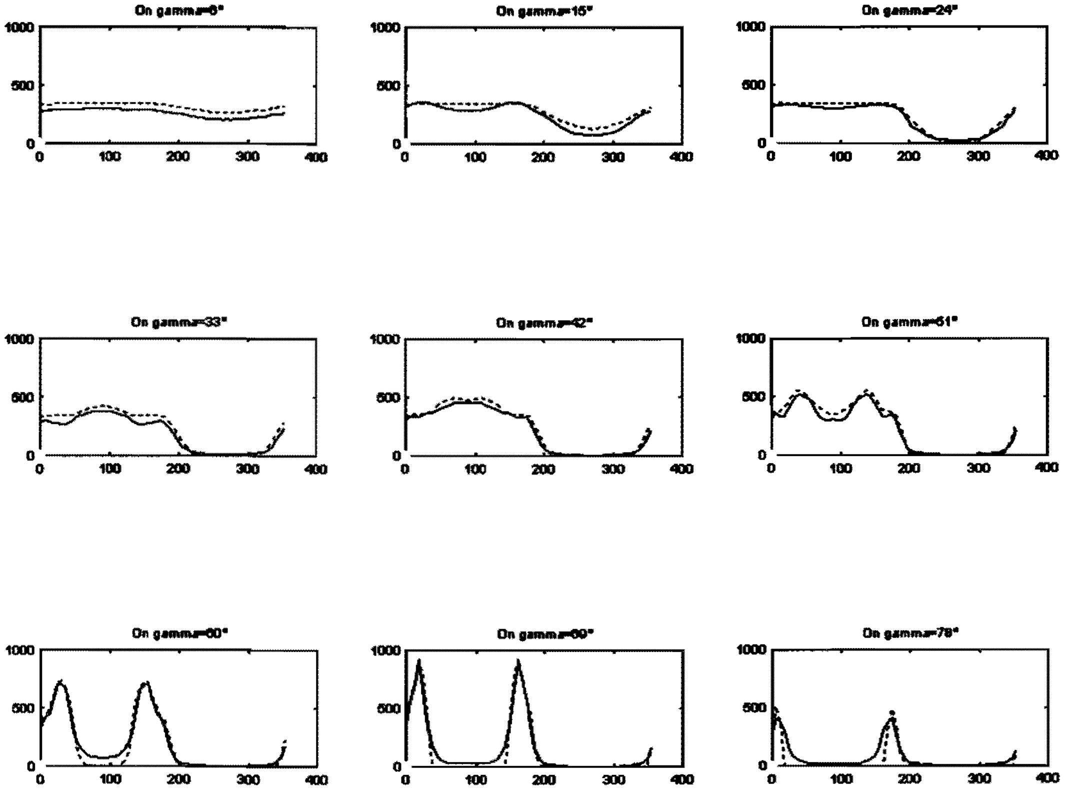


Figure 4.8: Best (36 sources) and target, on γ-circles

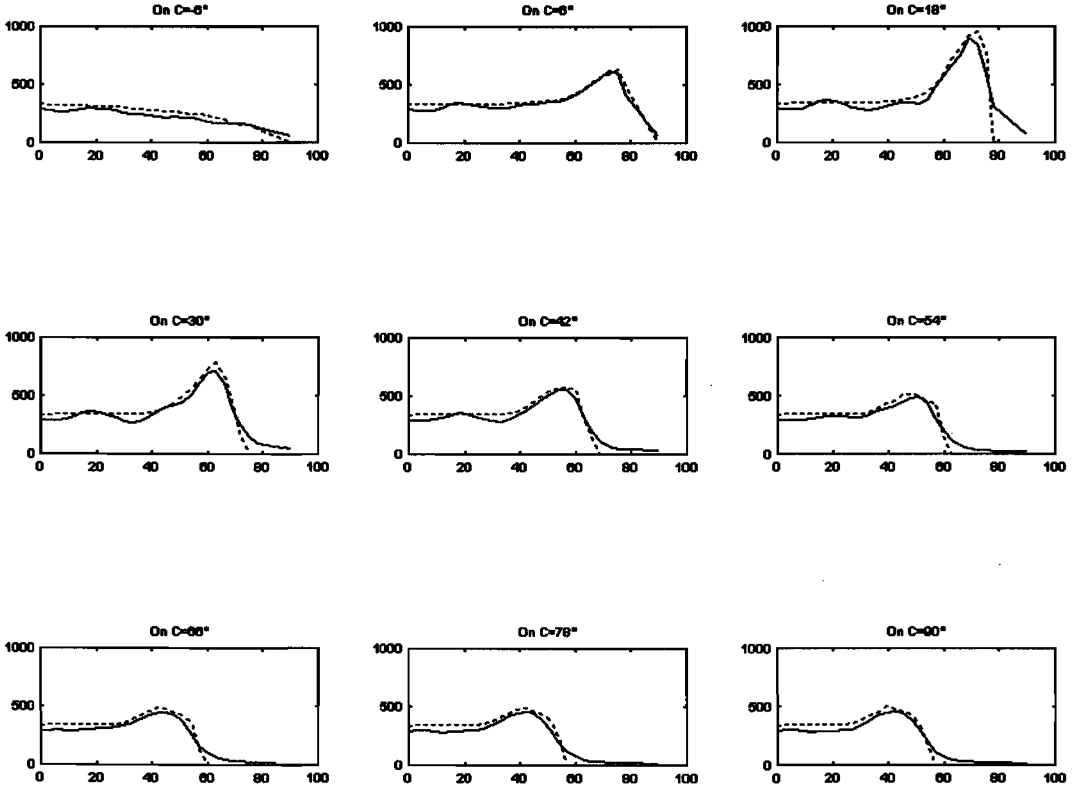


Figure 4.9: Best (36 sources) and target, on C-planes

Chapter 5

The effect on the road and CIE norms

In this chapter we consider a number of different quantities that can be computed to help determine the quality of a road lighting system. We discuss the values they must have to satisfy CIE norms and results obtained.

In section 1 we discuss the basic frame of reference used, as well as the notation used in this context and the parameters that are given for a particular road.

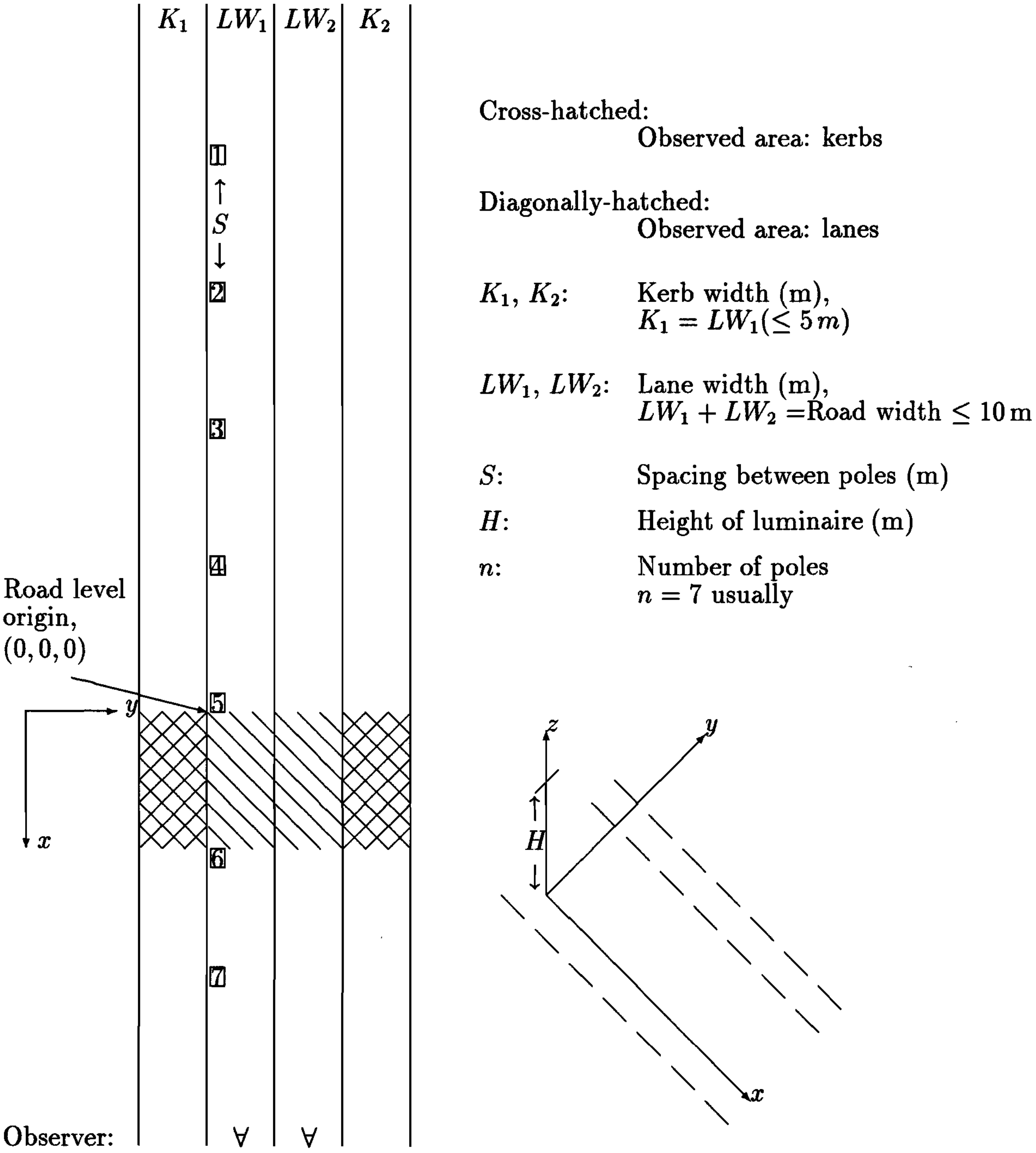
In section 2 we consider ways of calculating the different measures of the quality of the road lighting system. Subsequently we discuss a program for these quantities and in section 4 we define a figure of merit based on these quantities and present results obtained.

5.1 The geometry of the road

Consider figure 5.1 showing part of a roadway. A number of poles with luminaires at height H are placed at spacings S along the road, which has width $LW_1 + LW_2$. For the roadway we distinguish four parts; the *left kerb* (width K_1), *left lane*, *right lane* and *right kerb* (width K_2).

The whole area between the two poles is to be considered, and an observer is at a distance of $(57.3 + 171.9)/2$ from the centre of the *observed area.* As the luminaires on as many as 7 poles may have an influence on the observed area, we have to take the effect of at least seven luminaires along the road into account.

A mesh is placed over the observed area, thus forming grid points. The spots formed by the luminaires at a height of 10 m are not too small. Nevertheless,



Cross-hatched:
Observed area: kerbs

Diagonally-hatched:
Observed area: lanes

K_1, K_2:	Kerb width (m), $K_1 = LW_1 (\leq 5m)$
LW_1, LW_2:	Lane width (m), $LW_1 + LW_2$ =Road width ≤ 10 m
S:	Spacing between poles (m)
H:	Height of luminaire (m)
n:	Number of poles $n = 7$ usually

Figure 5.1: *Road geometry*

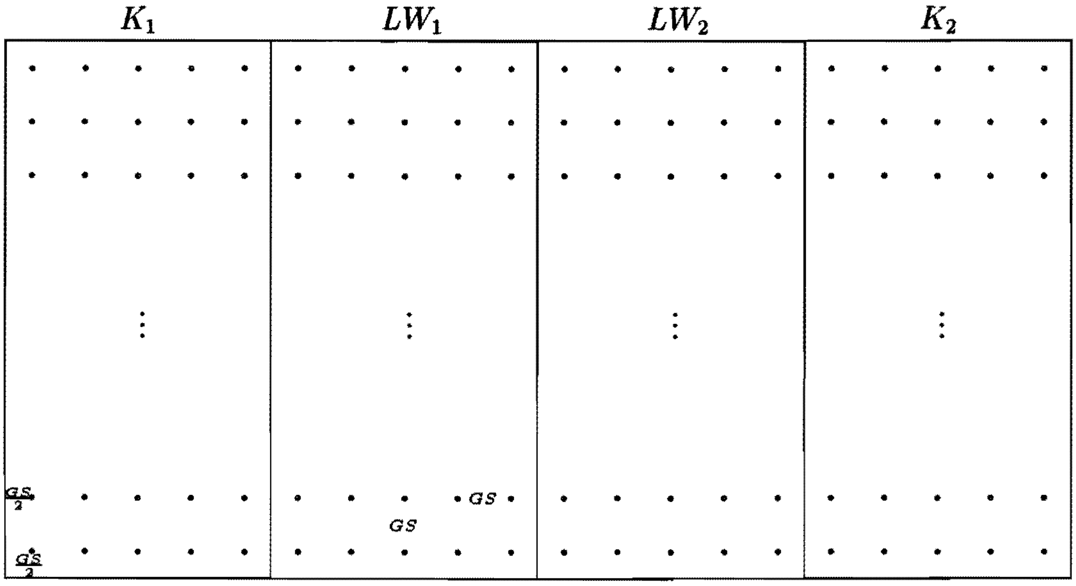


Figure 5.2: *Grid points*

as the intensities already change over a line with length of about 2 m, we should demand that the spots not be more than 2 m distant from each other, which would imply a grid spacing of not more than 1.5 m. This will ensure that we do not obtain good results at the grid points, but have bad spots in between them. The grid spacing should therefore be one fifth of the lane width, or 1.5 m, whichever is the smallest. Thus $GS = LW/5$ or smaller, with the first grid point in the grid array a distance $GS/2$ from each border of the observed region.

We make provision for two road types, **C2** and **R3**. Tables containing their reflecting properties are available, RTAB2.RTB and RTAB3.RTB, respectively.

We have to consider different *observer positions* for different uses. The observer is above the road and looks down and along the road, in the negative x-direction, slightly downwards. The specific positions of the observer for different cases will be given where we discuss these cases.

5.2 Relevant values to be calculated

After going through the procedures of interpolation, rotation, adding individual sources, tilting and rotating the block and adding the mirror image block, we are left with an IES-table for the luminaire. In this table the intensities are given in 'cd' relative to the $\gamma = 0$-axis vertically downwards, and the $C = 0$-plane in the direction of the x-axis, along one side of the road.

5.2.1 Illumination at all grid points

We compute the illumination, E, as well as the horizontal illumination, E_h, for all grid points. It is measured in 'lux' ($= lm/m^2$).

To compute this for a grid point P, we need the intensity in the direction of P. To calculate this from the IES-table, we need the (C, γ)-angles corresponding to the point P relative to the axis as given above.

The computation of these angles for grid points are straightforward: denote the luminaire by F, the grid point by P and the point at the base of the pole by O. Then,

$$\cos\gamma = \frac{\mathbf{FP} \cdot \mathbf{FO}}{FP.FO}$$

where **FP** is the vector from the luminaire to the grid point, **FO** the vector from the luminaire to the base of the pole, and FP and FO their magnitudes. Similarly,

$$\cos C = \frac{\mathbf{OP} \cdot \mathbf{i}}{OP}$$

with **i** the unit vector in the direction of the positive x-axis, or, equivalently,

$$\tan C = \frac{OP_y}{OP_x}.$$

With the C's and the γ's known for all grid points, the intensities can be looked up in the IES-tables (*i.e.*, by bilinear interpolation in these tables). Denote the intensity in such a direction by I. The distance between the grid point and the luminaire is FP in the notation given above, and the angle between the line PF and the vertical is the same as the angle γ for the point. Therefore

$$E = \frac{I}{FP^2}$$

and

$$E_h = E\cos\gamma = \frac{I}{FP^2}\cos\gamma$$

Once the horizontal illumination is known, we can compute the *surround ratios.* The surround ratios measure the illumination on the kerbs relative to that on the road lanes,

$$
\begin{aligned}
SR_1 &= \frac{E_{h_{K_1}} \times K_1 \times S}{E_{h_{LW_1}} \times LW_1 \times S} \\
SR_2 &= \frac{E_{h_{K_2}} \times K_2 \times S}{E_{h_{LW_2}} \times LW2_1 \times S}
\end{aligned}
$$

where the notation $E_{h_{K_1}}$ is the average horizontal illumination over the kerb area with width K_1 and of length S, and similarly $E_{h_{LW_1}}$ for the road area with width LW_1, etc.

For the calculation of *efficacy*, denoted by η_{road} and η_{total}, we have

$$
\begin{aligned}
\eta_{road} &= \frac{E_{h_R} \times (LW_1 + LW_2) \times S}{\Phi_{sources}} \\
\eta_{total} &= \frac{E_{h_T} \times (K_1 + LW_1 + LW_2 + K_2) \times S}{\Phi_{sources}}
\end{aligned}
$$

where the denominator is the flux of the luminaires ($= \int I\,dA$), measured in *lumen*, and the enumerator is illumination times area (measured in the same unit as the above, (lm/m^2).m.m). The half of the area between two masts is considered, and the average of the horizontal illumination taken over this, is multiplied with the relevant area and divided with half of the flux of the block [which may includes a mirror image]. This means that the ratio obtained is the same for any stretch of road, divided by the flux of all blocks having effect over that whole area.

These coefficients measure the amount of light falling on the concerned region as percentage of the output of light from the luminaires.

WANTED: *We want η_{road} and η_{total} to be as high as possible, the latter one as close to 1 as can be obtained.*

5.2.2 The average luminance at all grid points

We denote this by L_B, for an observer B, measured in 'cd/m^2'. We use the following observers for the computation of average luminance:

- z-coordinate: 1.5 m.

- x-coordinate: $114.6 + S/2$, in meters, as the observer field must be between 0.5° and 1.5° down, as shown in Figure 5.3.

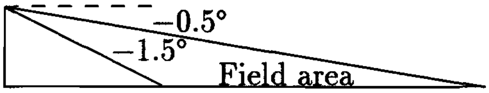


Figure 5.3: *Observer field*

- y-coordinate: One observer at $0.5LW_1$ and another at $LW_1 + 0.5LW_2$.
- Observed field: The whole road area of size $(LW_1 + LW_2) \times S$, for each of the two observers.

To compute the luminance from a grid point P in the direction of an observer B, we use the formula [LiDAC. 1993: Philips Lighting Manual, p.284]:

$$L = q\,E = \frac{r(\gamma,\beta)I}{h^2}$$

where:

- L is the desired luminance in the direction of B (cd/m^2).
- r is the reduced luminance coefficient for the road surface for the angle γ relative to the luminaire and grid point, and the angle β relative to the observer-gridpoint-luminaire combination. These angles are dicussed after this list, and the values for r are obtained from two-dimensional tables (as needs be, after interpolation) (cd/m^2/lux).
- h is the mounting height of the luminaire above the road (m).
- I is the luminous intensity of the luminaire in the direction of the grid point (cd).

The angle γ referred to above is the same as the angle γ used up to now. Furthermore, if we denote the point on the road under the observer by B_r, we have that the angle β is the angle between

OP: the elongation of the line from the point on the road under the luminaire, to the grid point, and

B_rP: the line from the point on the road underneath the observer, to the grid point.

We therefore have

$$\cos\beta = \frac{\mathbf{OP}\cdot\mathbf{B_rP}}{OP.B_rP}$$

With $\tan\gamma$ and β known, the value of r can be looked up and bilinearly interpolated for in the `*.RTB`-table. With this available, we can compute L for observers as specified, for all grid points on the road.

For each of these observers, we can then compute the mean over all the road grid points:

$$L_{ave,B} = \frac{\sum L_{grid\,points,B}}{N_{grid\,points}}$$

In cases where the distribution may not be smooth, we can do this for observer positions right across the road, in steps of 1 m.

WANTED: *We want L_{ave} for both observers to be a prescribed value, e.g., 2 cd/m^2.*

5.2.3 Overall uniformity

For the overall uniformity we consider two observers at the following positions:

- z-coordinate: 1.5 m
- x-coordinate: As in the previous case.
- y-coordinate: Observer 1: $0.5(LW_1)$
  Observer 2: $LW_1 + 0.5LW_2$
  We compute averages and minima over the field $(LW_1 + LW_2) \times S$;

The *overall uniformity* is defined as

$$U_{overall,B} = \frac{L_{min\ over\ road,B}}{L_{ave,B}}$$

where the average and the minimum is computed over all grid points in the field as given above, for each observer, B.

WANTED: We want $U_{overall}$ for both observers as close to 1 as possible, in virtually all cases ≥ 0.4.

5.2.4 Lengthwise uniformity

The *lengthwise uniformity* is defined as

$$U_{length,B} = \frac{L_{min\ along\ line}}{L_{max\ along\ line}}$$

calculated for each observer, B, along a line running from him parallel to the road.

We use the following observer positions:

- z-coordinate: 1.5 m
- x-coordinate: As above.
- y-coordinate: An observer on every grid line on LW_1 and LW_2 (at $0.5GS$, $1.5GS, \ldots, 9.5GS$.

  If the distribution is very smooth we can use only two, one at $0.5LW_1$ for field LW_1 and one at $LW_1 + 0.5LW_2$ for field LW_2. This was not used in the programs.

WANTED: We want U_{length} for every one of the observers as close to 1 as possible, usually ≥ 0.5. However, we deemed it best to demand a value higher than 0.6 in all cases, to ensure that no spottiness — a big potential complication for multisources — arise from use of the multisources.

5.2.5 The relative threshold increment (TI)

The observer here has the following coordinates:

- z-coordinate: 1.5 m
- x-coordinate: This must be such that the elevation angle of the first luminaire from a point on the edge of the road next to the observer, must be 20 degrees.
- y-coordinate: $0.5LW_1$ and $LW_1 + 0.5LW_2$.

Both observers look downwards at points straight ahead of them on the road, at angles of 1° down. Let θ be the angle between the line from the observer to the point on the road, and the line to the luminaire,

$$\theta = \arccos \frac{\mathbf{BP} \cdot \mathbf{BF}}{BP.BF}$$

Let d_i be the distance BF_i from the observer to the luminaire, F_i, and let I_i be the intensity in the direction of the observer, due to luminaire number i. Then we define

$$TI = 65 \frac{L_{veiling}}{(L_{ave})^{0.8}},$$

the answer being a percentage, where

$$L_{veiling} = \sum_{i=1}^{n} \frac{10 \times \cos\theta_i \times I_i}{d_i^2 \theta_i^2}$$

with θ_i the angle to the ith luminaire, measured in degrees.

This must be computed in absolute units, not scaled according to case, while all other measures can be calculated relative.

WANTED: We want TI for both observers as small possible, usually $\leq 10\%$.

5.3 A program for the norms

The two programs `roadinit.m` and `roadmeas.m` as listed in REPORT IWDE 97-06, Section 13, computes all these values. The first program precomputes as many needed variables as possible, and stores them in global variables, so that the second program can simply access them.

The program `roadmeas.m` also writes results to file, and draw graphs of the illumination every so often, and at the end of an optimization stage.

5.4 Results

The objective function was redefined to implement the above measures. It was buit up of the following components, where values obtained for the ideal distribution were used as base, together with the CIE-norms. Say the vector, $\mathbf{v}$, has as components the measurements discussed above:

$$\mathbf{v} = (\Phi_{need}, \eta_{road}, \eta_{total}, \min(U_{overall,1}, U_{overall,2}), \min_{a,b,\ldots,j}(U_{lengthwise}), SR, -TI/100),$$

Measure	Ideal	Best: 36
η_{road}	68%	68%
η_{total}	99%	89%
$L_{ave,1}$	1.26	1.47
$L_{ave,2}$	1.15	1.32
$U_{overall,1}$	0.43	0.40
$U_{overall,2}$	0.59	0.55
$\min_{a,b,\ldots,j}(U_{lengthwise})$	0.56	0.70

Table 5.1: Results for two blocks, 10:10:60

where the quantity Φ_{need} is the flux needed to achieve the desired minimum average luminance, W_{L_a}.

This needed flux have to be divided by, *e.g.*, 10, in the vector **c** following, in order for it to be of the same order of the other quantities, else it may totally dominate the minimization, with it being minimized and aother quantities virtually neglected.

Let the following vector contain the values of the measurements desired, all minima, except for the first and the last. The last is a maximum allowed, and therefore the values are changed in sign. The values of the threshold increment are also divided by 100 to make them of the same order as the others.

$$\mathbf{c} = (0.0, W_{\eta_r}, W_{\eta_t}, W_{U_o}, W_{U_l}, W_{SR}, -W_{TI}/100),$$

where W_a signifies the wanted value for quantity a.

Finally, let the following vector contain values that can (hopefully) realistically be aimed for:

$$\mathbf{d} = (0.0, 0.96, 0.80, 0.42, 0.72, 0.42, 0.0)$$

Define

$$\mathbf{e} = [10(\mathbf{c} - \mathbf{v})_+]^2 + [(\mathbf{d} - \mathbf{v})_+]^2$$

where the +-subscript denotes the positive part of the vector, *i.e.* that we replace negative components by zero.

Originally we also included a term for the integral of the squared difference between the obtained IES-table and the ideal. It was however found that once the ideal has been reasonably closely approximated, an improvement of the measurements can be obtained by moving somewhat away from the ideal IES, and this was eliminated.

The table shows some results for a 10 m road with 5 m kerbs and 60 m between poles. The same flux was used in all cases, showing that better luminance and uniformities could be obtained. with the multisources. The threshold increments presented a problem, as the screening on the edges of the target could not be duplicated by them, as discussed in chapter 4.

The first column shows the name of the criterion, and the second column the value of that criterion for the "ideal" light intensity distribution. This is the IES table that was constructed to satisfy criteria for a road lighting system [T van Hees: `INR16D.IES`].

The third column, *Best: 36*, presents the best setup obtained with the restriction placed on the system that no prism should be used with deflection angle larger than 45°. Originally the best setup was obtained with more than $\frac{1}{4}$ of the prisms having a deflection angle of 50°, and as these are known to be ineffective this restriction was imposed.

In the following chapter we discuss other sets of optical data, and full results for such cases.

Chapter 6

Other optical systems and results

In this chapter we briefly discuss results that were obtained using other optical subsystems, and draw some conclusions. In the first section we regard a wide road, with wide kerbs, and with a large spacing between the masts. In section 2 we consider other road geometries, and in the last section we consider the question of what road geometry will be the most effective.

6.1 Results: 10 m:10 m:60 m

We considered a 10 m wide road with 10 m high luminaires, spaced 60 m apart. On this we desire an average luminance in the direction of each observer to be higher than 1 cd/m.

The requirements *qua* uniformity on the road are that the overall uniformity, for both observers, be greater than 0.4 and that the lengthwise uniformity be greater than 0.5. Due to the risk of spottiness for multisources, we consistently demanded a lengthwise uniformity of greater than 0.6.

For some demands by standards organizations the surround ratios have to be greater than 50%, for others no requirement is set. We required a surround ratio larger than 30% for most situations.

The threshold increment (TI), for roads where a requirement is set, is usually either 10% or 15%. It soon became apparant that we would not be able to meet this. There is a physical limitation causing this. If the observer is at a position where the elevation angle of the luminaire is precisely 20°, and a ray of light just misses his eye, this ray falls onto the road at 5.49 times the height of the luminaire from the foot of the luminaire. If the light is screened or aimed so that no light falls on his eye directly, this means that no light from the luminaire will fall further than 5.5 times the height from the luminaire. The only light falling on the road will be from the luminaire behind the observer. But this is totally ineffective from the aspect of reflectance, and does not contribute significantly to the luminance in the direction of the observer. The region from 5.5 times the pole height up to the next pole must not have a lower illuminance than 0.6 the maximum. But this is impossible without light falling onto this from in front — and this is screened by the threshold increment requirements!

In the light of the above comments we deemed it sufficient if the threshold increment for these extreme spacings could be kept lower than 25%.

Results

Type	**Input needed**	**Objectives achieved?**	U_o	U_ℓ	TI	SR
	$\left(\frac{kLumen}{cd.m^{-2}}\right)$		(*percentage*	*points*)		
			$\geq 40\%$	$\geq 60\%$	$\leq 25\%$	$\geq 30\%$
g2g200	11.76	Yes	0	−2	+1	−1
g1g202	11.95	Yes	+1	−1	+1	−1
t1g202	12.14	Yes	0	−1	0	−1
g2g202	13.09	Yes	0	−1	+1	−1
t1g203	12.88	Almost	−1	−3	+2	0
t1g200	10.67	NO/Almost	−1	−5	+2	0
df30s7	11.78	NO/Almost	+1	−6	+2	−1
g2g203	14.26	NO	−2	−5	+3	+3
g1g203	14.10	NO	−3	−5	+3	−1
df30s4	11.59	NO	−3	−12	+7	−1
def30_	10.59	NO	−7	−13	+6	−1
g1g200	10.78	NO!	−17	−27	+10	+3
g1g1-set	9.61	Yes	+2	0	+1	0
g1g2-set	9.75	Yes	+2	0	+1	0
t1g1-set	10.33	Yes	+2	0	+2	0
t1g2-set	10.81	Yes	+2	0	+1	−1
g2g2-set	11.27	Yes	+1	0	+2	−1

Three types of input beams into the deflection optics were considered. A narrow gaussian shaped beam, designated *g1*, a wider one designated *g2* and an inbetween, measured one designated *t1*.

Two type of deflection optics were coupled to these, one designed for the g1-beam, called *g1*, and one for the g2-beam, called *g2*.

Further spreading of the beams were obtained by curving the optical structures. Four types of spreading structures were used, 00, for no spreading, 01 for a spreading up to 8°, 02 for no spreading up to 16° and 03 for a spreading to 24°.

Therefore, a code *g1g201* would signify an input narrow gaussian beam gl, with optics of type g2, with spread such that the beams has an 8° spread.

Most of the pure systems, with just a single type of optical structure were used to obtain results. Some were good, some on the edge of being acceptable, and some totally unusable.

For all combinations of beams and optical system, a system using combined spread were used. By this we mean that some luminaires would use a narrow beam, some a wide beam, etc, The results here were consistently good.

A last type of structure was one where the optical elements were on the inside of the surface of the deflection structure. This caused bilobed beams, with one deflected beam giving most of the light, the other spread out to the other side. Two type are presented: *df30s7* and *def30_*, the first one with a slightly wider input beam.

In the table the deviation from the required values are indicated by the symbols $+x$ or $-x$. For the threshold increment a postive value means that the required value has not been achieved, for the other quantities a positive value means it has been bettered.

The first column gives the input flux into the deflection optics, for the entire (double) block needed to obtain the luminance in the required directions.

6.2 Results for smaller roads

Scaling the problem down by 70% causes half of the flux to be needed, as shown in the following table. A road of 7 m width, with 7 m masts and a pole spacing of 5 times the pole height were considered.

Due to the fact that these were inside the limits required to achieve the required threshold increment, the demand was imposed that the TI be less than 10% and the surround ratios greater than 50%.

Results, 7 m:7 m:35 m

Type	**Input**	**Objectives**	U_o	U_ℓ	TI	SR
	needed	**achieved?**	(*percentage points*)			
	($\frac{kLumen}{cd.m^{-2}}$)		$\geq 40\%$	$\geq 60\%$	$\leq 10\%$	$\geq 50\%$
glg202	5.77	Yes	+6	0	+3	0
df30s7	5.42	Almost	+5	−3	+7	+4

Still imposing stricter conditions w.r.t. threshold increment, imposing a 0.7 requirement for lenghtwise unifomity, relaxing the surround ratios somewhat, but making the pole spacing even less than 5 times pole height, resulted in the following:

Results, stricter conditions

Type	**Road**	**Input**	**Goal**	U_o	U_ℓ	TI	SR
		needed	**?**	(*percentage points*)			
		($\frac{kLumen}{cd.m^{-2}}$)		$\geq 40\%$	$\geq 70\%$	$\leq 10\%$	$\geq 30\%$
tlg2-set	7:7:30	5.20	Yes	+14	0	+3	0
tlg2-set	7:7:21	3.30	Yes	+9	0	0	0

From this it is apparant that the stricter conditions offset the gains in effectiveness for the 1:1:4.5 spacing, but that the 1:1:3 spacing represents an gain in effectiveness, taken in combination with the larger uniformity — which requires more light at the slightly darker reaches.

6.3 Optimal road geometry

A question is what pole height and pole spacing would be the most effective over a long road. A partial answer is given by the last table of the previous section and by the following table. In this table we made a few experiments with various geometries. The objectives were kept the same as in the first table of the first section of this chapter.

One of the best of the above types was taken, *g2g200* and the same criteria as in the first case were taken ($TI < 25\%$, $SR > 30\%$, $U_o > 0.4$, $U_\ell > 0.6$).

Results: Various

Height	**Design**	**Input needed** $(\frac{kLumen}{cd.m^{-2}})$	**Objectives?**	**Input needed per 60 m road section**
10 m	10:10:60	11.76	Yes ≈	11.76
10 m	10:10:50	8.26	Yes	9.91
9 m	10:9:45	7.28	Yes !	9.71
8 m	10:8:40	6.60	Yes !	9.90
7 m	10:8:35	> 5.90	Yes ≈	> 10.10

It is apparant that a considerable gain in effectiveness is obtained with a 1:1:5 spacing, and a slight additional gain with a slightly lower pole. However, for effectiveness and robustness in design it is advisable to design for the pole height equal to the road width.

In the following table a typical output is given. Similar results for various situations are in the possesion of LumiLeds BV. [Spoelstra (1997)b]

```
 ------- Luminaire data -----------------   5 December 1997
 Optical type (name of datafiles):
c:\leddata\t1g2\t1g200 This corresponds to third factor = 0
c:\leddata\t1g2\t1g202 This corresponds to third factor = 1
c:\leddata\t1g2\t1g203 This corresponds to third factor = 2
 Tilt of the luminaire:      49.9 deg
 Rotation of the luminaire:  32.4 deg
 Deflection, placement and third factor:
       44.1        134.8          2.0
       41.1         -9.8          1.0
       33.0       -136.2          0.0
       40.5       -179.9          2.0
       31.3        -34.2          0.0
       34.1         -5.0          1.0
       31.9        -35.1          0.0
       39.6         -4.9          1.0
       22.7        -68.6          1.0
       39.4        150.0          2.0
       27.9        -26.4          1.0
       36.0        -55.1          0.0
       32.6        -72.8          1.0
       36.3        123.0          2.0
       28.4        112.2          1.0
       18.2        -39.2          1.0
       25.1        -41.5          0.0
       12.8        153.0          2.0
        8.2        -12.9          1.0
       10.5         96.5          1.0
```

```
      25.8       151.1          2.0
      18.3       124.5          1.0
      25.7      -108.6          1.0
      13.5       -71.6          1.0
      18.5      -129.4          2.0
      15.3        88.2          1.0
      12.8       -41.9          1.0
      16.1       169.9          2.0
      14.0       -84.6          2.0
      16.0       -36.6          0.0
      23.6      -176.4          2.0
      21.7      -168.7          2.0
       0.8      -177.2          1.0
       4.0       134.1          1.0
      11.2         6.9          1.0
       6.6         6.8          1.0

-------- Road      ---------------------
Left  kerb    3.5m Left lane     3.5m  Right lane    3.5m  Right kerb    3.5m

-------- Poles     ---------------------
     -120.0      0.0      7.0
      -90.0      0.0      7.0
      -60.0      0.0      7.0
      -30.0      0.0      7.0
        0.0      0.0      7.0
       30.0      0.0      7.0
       60.0      0.0      7.0

-------- Efficiencies ------------------
Output lumens needed per pole          4326.99
Input lumens needed per pole           5200.57
kilo-Lumens needed per cd/m^2             5.20
Total efficiency                          90.7 %
Road  efficiency                          69.9 %
Left  surround ratio                      29.7 %
Right surround ratio                      29.7 %

-------- Luminances    -----------------
Average luminance for  first observer       1.03 cd/m^2
Average luminance for second observer       1.00 cd/m^2
Overall uniformity for  first observer      0.54
Overall uniformity for second observer      0.61
Line uniformities  0.70 0.70 0.70 0.70 0.70 0.70 0.70 0.71 0.71 0.76
Threshold increment for  first observer    12.85 %
Threshold increment for second observer    12.86 %
Wanted values: SR >=  30.00  U_o >=   0.43,  U_l >=   0.70,  TI <=   10.0 %
```

Chapter 7

References

- Hees, A van. 1997a. Road measurements. Personal communication.
- LiDAC. 1993. Philips Lighting Manual. Philips.
- Spoelstra, J. 1997a. The LumiLeds computer program. Report IWDE-06. TUE.
- Spoelstra, J. 1997b. Results. Optimal road block setups and values of the road measurements for such setups. Unpublished, in the possesion of LumiLeds BV.