Université de Strasbourg

Master CSMI

Improving upon the 3-Ellipsoid model for adjusting r-table measurements

Author:
Carl SCHMITTBIEL

Project supervisor:
Valérie MUZET

Course lecturers:
Christophe PRUD'HOMME
Joubine AGHILI

August 21, 2023

Contents

List of Figures

Special thanks :
Thank you to Valérie Muzet for introducing me
to this research subject and guiding me in my research,
to all of the team at CEREMA Strasbourg
lead by Pierre Charbonnier,
and to Camille Gilbert for the design
of the ELLIFANT Logo.

Carl Schmittbiel

I Introduction

In our ever-evolving society, the importance of mobility cannot be overstated. As our lives become increasingly interconnected and fast-paced, safe and efficient road infrastructures have become essential. One key aspect of road safety is the visibility of road surfaces and markings. These vital components ensure that drivers can navigate their way safely, particularly during challenging conditions or when visibility is limited. As a result, the reflective properties of road surfaces play a critical role in road safety.

During nighttime, road visibility is influenced by two essential factors: the illumination from luminaires and the reflection of light from the road surfaces. While luminaires are responsible for providing light on the road, the reflective properties of the road surfaces themselves significantly impact visibility. The interaction between the amount of light emitted by the luminaires and the light reflected by the road surfaces directly affects how clearly drivers can see the road at night. Consequently, comprehending and optimizing the reflective properties of road surfaces become integral aspects of road infrastructure design.

However, measuring the reflective properties of road surfaces is a complex task. It often requires samples of road surfaces, which involves extraction of road samples (called cores) in the road to obtain accurate measurements in laboratories. Alternatively, researchers may need to use expensive equipment specifically designed for direct measurements on the road surface. Additionally, road surfaces undergo changes over time due to the effects of vehicle traffic and weather conditions. As a result, collecting data on the reflective properties of road surfaces poses further challenges, often necessitating several years of observation and data gathering.

To address these challenges and streamline the process of data collection, there is a growing need for robust mathematical models that accurately describe the relationship between road lighting and the reflective properties of road surfaces. By developing such models, researchers can minimize the effort required for extensive data collection and facilitate a more efficient analysis of road lighting conditions.

Thus this project aims to refine existing mathematical models based on ellipsoids by incorporating real-world lighting reflection data. Ultimately, through the development and application of these mathematical models, this project endeavors to enhance our understanding of road lighting and the critical role played by the reflective properties of road surfaces. By doing so, it strives to contribute to the ongoing efforts to create safer, more efficient, and sustainable road infrastructures that meet the needs of our society.

CEREMA

This project was requested by the Cerema, which is a french public institution created in 2014. It stands for "Center for Studies and Expertise on Risks, the Environment, Mobility and Urban Planning". It has over 2 400 agents in 26 different locations. The six activity domains of Cerema are :

- territorial engineering

- building
- mobility
- **transport infrastructure**
- risk and environment
- coast and sea

ENDSUM research team

The work done in this project is part of the ENDSUM research team. ENDSUM stands for "Evaluation Non Destructive des Structures et des matériaux", which means "Non-destructive evaluation of structures and materials". The principal research topics of the team are:

- **Photometry, and optical methods**
- Defect detection
- Image processing, computer vision
- 3D reconstruction
- Artificial intelligence.

REFLECTIVITY Research Project

The REFLECTIVITY project, which corresponds to *Road surfacEs Function for Lighting Evaluation, road marking ContrasT, urban heat Island to ensure VIsibility and sustainabiliTY*, is a project lead by ENDSUM Strasbourg that involves different actors: three research teams from the Cerema Carnot Institute Clim'Adapt called ENDSUM, EL and STI. an industrial company, two university laboratories and two local authorities.

The motivator for this project is to demonstrate the importance of the knowledge of the r-table of the road surfaces to answer the climatic challenge and the development of autonomous mobility.

It proposes the creation of a shared database containing data to describe both structural characteristics of road surfaces and their optical properties. Further detail about this research project can be found at :

The reflectivity project

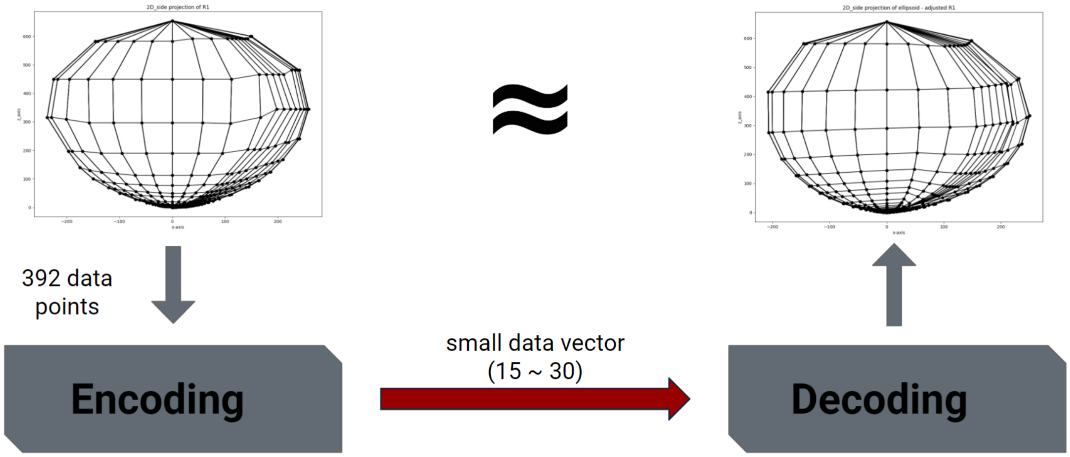

Figure 1: Our aim for this project is to get a good coding/decoding algorithm

I.1 Objectives

The objective of this work is to prototype a model that will allow to sum up r-table data in only a few parameters, and to use these parameters to generate a new r-table that will have the same properties as the original one, ideally with a rooted mean square error as low as possible. The process is illustrated in the figure 1.

The project will contain:

- A full code base for different fitting algorithms
- Complete documentation for the application

II Road Photometry

II.1 General principles

The CIE is the international commission on illumination, refered to by it's original french name (Commission Internationale de l'Eclairage). It is responsible for defining the standards in the field of illumination.

The surface of a pavement is described according to its reflection properties following CIE144:2001 [2] specifications. The most characteristic parameter is the luminance coefficient q, given as:

$$q(\alpha, \beta, \epsilon) = \frac{L(\alpha, \beta, \epsilon)}{E_h(\beta, \epsilon)} \tag{1}$$

It is the ratio between the observed luminance L in cd.m-² that the observer sees, and the illuminance E_h in lux incident on the surface.

The standardised viewing height is 1.5 m and the angle of observation α is constant at 1 degree, corresponding to an observation distance of 86 m. The lighting standards use the area of the road between 60 m and 160 m ahead of the driver, because it is considered an important area for the detection of obstacles. Since the 1980s, for practical reasons the luminance coefficient was replaced by the reduced luminance coefficient r in $cd/m-2/lx^{-1}$, which is derived from q:

$$r(\beta, \epsilon) = 10^4 \cdot q(\beta, \epsilon) \cdot cos^3(\epsilon) \tag{2}$$

Reflectance values are measured for many directions and collected in a table called an r table, where the luminance coefficient r is given for a combination of fixed lighting angles β and ϵ **??**. Delta is set to zero by convention because of a hypothesis of isotropy of the road surface.

The r-values are easier to handle and behave more intuitively than q-values, which is why they are used widely in the field of road photometry. This project exclusively uses r-tables.

These values of beta and epsilon used in r-tables are defined as follows:

$$\beta = 0, 2, 5, 10, 15, 20, 25, 30, 35, 40, 45, 60, 75, 90, 105, 120, 135, 150, 165, 180 \tag{3}$$

$$tan(\epsilon) = 0, 0.25, 0.5, 0.75, 1, 1.25, 1.5, 1.75, 2, 2.5, 3, \ldots, 10.5, 11, 11.5, 12 \tag{4}$$

Defining epsilon with it's tangent poses no problems here, the angles stay well in the range of $[0, \pi/2]$ for the angle ϵ.

Very often, for historic reasons, only 392 on these values are actually measured. As they are seemed uninteresting for the purposes of road photometry, the remaining 188 values are left as 0 in most r-tables. Researchers in the fields however now have the means to measure these values.

The reflection indicatrix

In order to visualize the r-tables data, they can be plotted in a 3D space. The standard for doing this is to use a standard coordinates transformation from spherical coordinates to cartesian coordinates. In this representation, r is the radius, β is the azimuth angle and ϵ is the polar angle.

So the transformation is given by the following equations:

$$\begin{cases} x = r \cdot cos(\beta) \cdot sin(\epsilon) \\ y = r \cdot sin(\beta) \cdot sin(\epsilon) \\ z = r \cdot cos(\epsilon) \end{cases} \tag{5}$$

A scatter plot of the r-values in this coordinate system is a reflection indicatrix (also called a photometric solid). Usually, lines are drawn between the points of the same β and ϵ to make the visualization easier. Notice on the figures **??**, the uppermost point

beta / tan gamma	0	2	5	10	15	20	25	30	35	40	45	60	75	90	105	120	135	150	165	180
0	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655
0,25	619	619	619	619	610	610	610	610	610	610	610	610	610	601	601	601	601	601	601	601
0,5	539	539	539	539	539	539	521	521	521	521	521	503	503	503	503	503	503	503	503	503
0,75	431	431	431	431	431	431	431	431	431	431	395	386	371	371	371	371	371	386	395	395
1	341	341	341	341	323	323	305	296	287	287	278	269	269	269	269	269	269	278	278	278
1,25	269	269	269	260	251	242	224	207	198	189	189	180	180	180	180	180	189	198	207	224
1,5	224	224	224	215	198	180	171	162	153	148	144	144	139	139	139	144	148	153	162	180
1,75	189	189	189	171	153	139	130	121	117	112	108	103	99	99	103	108	112	121	130	139
2	162	162	157	135	117	108	99	94	90	85	85	83	84	84	86	90	94	99	103	111
2,5	121	121	117	95	79	66	60	57	54	52	51	50	51	52	54	58	61	65	69	75
3	94	94	86	66	49	41	38	36	34	33	32	31	31	33	35	38	40	43	47	51
3,5	81	80	66	46	33	28	25	23	22	22	21	21	22	22	24	27	29	31	34	38
4	71	69	55	32	23	20	18	16	15	14	14	14	15	17	19	20	22	23	25	27
4,5	63	59	43	24	17	14	13	12	12	11	11	11	12	13	14	14	16	17	19	21
5	57	52	36	19	14	12	10	9,0	9,0	8,8	8,7	8,7	9,0	10	11	13	14	15	16	16
5,5	51	47	31	15	11	9,0	8,1	7,8	7,7	7,7										
6	47	42	25	12	8,5	7,2	6,5	6,3	6,2											
6,5	43	38	22	10	6,7	5,8	5,2	5,0												
7	40	34	18	8,1	5,6	4,8	4,4	4,2												
7,5	37	31	15	6,9	4,7	4,0	3,8													
8	35	28	14	5,7	4,0	3,6	3,2													
8,5	33	25	12	4,8	3,6	3,1	2,9													
9	31	23	10	4,1	3,2	2,8														
9,5	30	22	9,0	3,7	2,8	2,5														
10	29	20	8,2	3,2	2,4	2,2														
10,5	28	18	7,3	3,0	2,2	1,9														
11	27	16	6,6	2,7	1,9	1,7														
11,5	26	15	6,1	2,4	1,7															
12	25	14	5,6	2,2	1,6															

Figure 2: Example of an r-table.

corresponds to epsilon = 0 and with r-values typically being in the same range on this line, we can see they line up on a single point in space.

Since this is a 3D plot, it is not possible to visualize it directly on a 2D screen. So a projection is used to visualize the data. The projection can be chosen to be along any axis, but most commonly the side projection and the top-down projection are used:

- The side projection is the projection along the y axis.
- The top-down projection is the projection along the z axis.

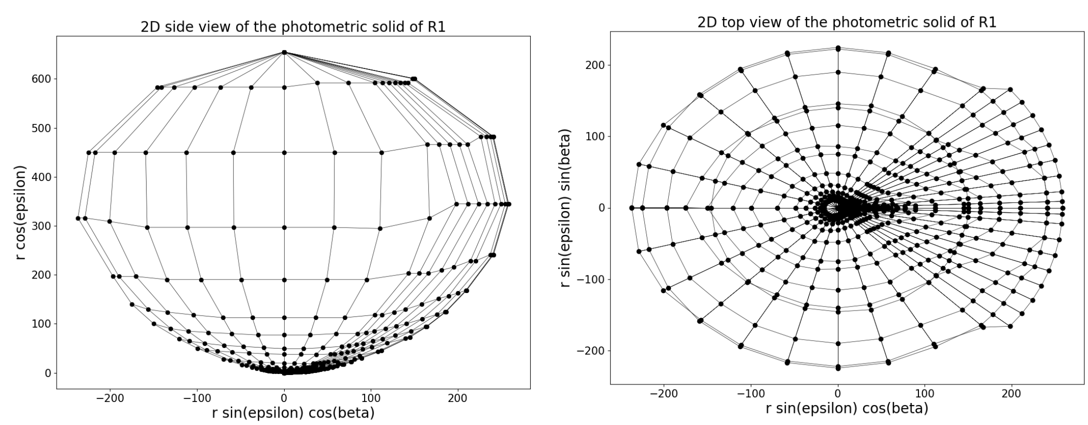

Figure 3: Both of the common 2D projections of the photometric solid

Note that for the top-down projection, the data is plotted twice. This is standard practice that allows us to see a full solid of data, as the data is only defined for $\beta \in [0, 180]$. We simply asssume that the data are the same for $\beta \in [180, 360]$ using a symetry.

II.2 The data parameters of an r-table

In road photometry, different parameters are used to describe the data of an r-table.

- The first parameter is S_1, which is defined as

$$S_1 = \frac{r(0,2)}{r(0,0)}. \tag{6}$$

This parameter is used to describe the specular component of the surface (mirror effect).

- The second parameter is Q_0, the average luminance coefficient that represents the degree of lightness of the measured surface. It is defined as

$$Q_0 = \int_0^{\Omega_0} q(\beta, \epsilon) d\Omega. \tag{7}$$

Where Ω is the solid angle of the photometric solid. This parameter determines the total amount of light reflected by the surface directed towards the observer.

One parameter that is very relevant for this project is the $\Delta_{r-table}$ parameter that was proposed by Florian Greffier *et al* [6] to measure the distance between two r-tables. It is defined as the RMSE (rooted mean squared error) divided by the mean r-value over both tables:

$$\Delta_{r-table}(r-table_i, r-table_j) = \frac{\sqrt{\sum_k \sum_l (r^i_{k,l} - r^j_{k,l})}}{\frac{\frac{\sum_k \sum_l r^i_{k,l}}{580} + \frac{\sum_k \sum_l r^j_{k,l}}{580}}{2}} \tag{8}$$

$\Delta_{r-table}$ or Delta-r will serve as our measurement of error between an r-table and it's adjusted version by a particular model.

Other parameters are also sometimes used and will be calculated, but they are meant for further research that will not be used in this project. These include Q_d computation according to CIE144 methodology [2] as well as a different way to compute Q_0 according to Boucher et al. [4].

beta / tan gamma	0	2	5	10	15	20	25	30	35	40	45	60	75	90	105	120	135	150	165	180
0	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655	655
0,25	619	619	619	619	610	610	610	610	610	610	610	610	610	601	601	601	601	601	601	601
0,5	539	539	539	539	539	539	521	521	521	521	521	503	503	503	503	503	503	503	503	503
0,75	431	431	431	431	431	431	431	431	431	431	395	386	371	371	371	371	371	386	395	395
1	341	341	341	341	323	323	305	296	287	287	278	269	269	269	269	269	269	278	278	278
1,25	269	269	269	260	251	242	224	207	198	189	189	180	180	180	180	180	189	198	207	224
1,5	224	224	224	215	198	180	171	162	153	148	144	144	139	139	139	144	148	153	162	180
1,75	189	189	189	171	153	139	130	121	117	112	108	103	99	99	103	108	112	121	130	139
2	162	162	157	135	117	108	99	94	90	85	85	83	84	84	86	90	94	99	103	111
2,5	121	121	117	95	79	66	60	57	54	52	51	50	51	52	54	58	61	65	69	75
3	94	94	86	66	49	41	38	36	34	33	32	31	31	33	35	38	40	43	47	51
3,5	81	80	66	46	33	28	25	23	22	22	21	21	22	22	24	27	29	31	34	38
4	71	69	55	32	23	20	18	16	15	14	14	14	15	17	19	20	22	23	25	27
4,5	63	59	43	24	17	14	13	12	12	11	11	11	12	13	14	14	16	17	19	21
5	57	52	36	19	14	12	10	9,0	9,0	8,8	8,7	8,7	9,0	10	11	13	14	15	16	16
5,5	51	47	31	15	11	9,0	8,1	7,8	7,7	7,7										
6	47	42	25	12	8,5	7,2	6,5	6,3	6,2											
6,5	43	38	22	10	6,7	5,8	5,2	5,0												
7	40	34	18	8,1	5,6	4,8	4,4	4,2												
7,5	37	31	15	6,9	4,7	4,0	3,8													
8	35	28	14	5,7	4,0	3,6	3,2													
8,5	33	25	12	4,8	3,6	3,1	2,9													
9	31	23	10	4,1	3,2	2,8														
9,5	30	22	9,0	3,7	2,8	2,5														
10	29	20	8,2	3,2	2,4	2,2														
10,5	28	18	7,3	3,0	2,2	1,9														
11	27	16	6,6	2,7	1,9	1,7														
11,5	26	15	6,1	2,4	1,7															
12	25	14	5,6	2,2	1,6															

Figure 4: An r-table with the three groups highlighted.

II.3 The proposed 3-ellipsoid model

An ellipsoid fitting algorithm was proposed by Ana Ogando-Martinez et al in their 2020 publication [9]. The proposed algorithm is a 3-ellipsoid model, which is a model that approximates r-tables in their representation as a photometric solid. It was observed that photometric solids can be very well approximated by ellipsoids, the 3D equivalent of ellipses, and using the parameters of these ellipsoids, the r-table can be approximated with a high degree of accuracy, using much less parameters than the original r-table.

The publication proposes to first divide the r-table into three different groups of points and to fit an ellipsoid to each group. The first group includes all the points where $0° \leq \beta < 15°$, the second group includes all the points where $15° \leq \beta < 60°$, and the third group includes all the points where $60° \leq \beta \leq 180°$.

A 3d ellipsoid's equation can be defined using nine parameters like this:

$$ax^2 + by^2 + cz^2 + 2fyz + 2gzx + 2hxy + 2px + 2qy + 2sz + d = 0 \tag{9}$$

Due to the photometric solid always passing through the origin, the ellipsoids are also constrained to pass through the origin, prohibiting any translation component. As photometric solids are inherently symmetric with respect to the y=0 plane, the ellipsoids exhibit the same symmetry. Hence, they cannot have a translation component in the y-direction either.

Additionally, rotations along the X and Z axes are also disallowed, as the photometric solid's symmetry with respect to the y=0 plane precludes such rotations. These simplifications reduce the number of parameters to only 5, resulting in a much simpler equation. Consequently, this leads to a more efficient model due to a smaller search space.

$$a'(x^2 + y^2 - 2z^2) + b'(x^2 + z^2 - 2y^2) + 2gxz + 2px + 2sz = x^2 + y^2 + z^2 \tag{10}$$

Using a least squares fitting algorithm the parameters of the ellipsoids are determined, and all the data needed to reconstruct the r-table is stored in the parameters of the ellipsoids, so $3x5 = 15$ parameters in total, which gives us a total compression rate of 97.4% and the resulting ellipsoids are shown in figure.

To rebuild the r-table, half lines are drawn from the origin is all β , ϵ directions, and the intersection of these half lines with the ellipsoid covering their group is calculated. The r-value of this intersection is the r-value of the r-table at the corresponding β and ϵ.

To improve the alignment of the data with real-world measurements, some adjustments are necessary after the fitting process. Firstly, the columns in the table, located on both sides of an ellipsoid separation, are averaged with their adjacent neighbors. Secondly, the first row is set to its average value since real data commonly exhibits this pattern. The assumption of isotropy ensures that the first row always holds the same value, representing the normal to the surface, which remains constant regardless of the azimuth angle β.

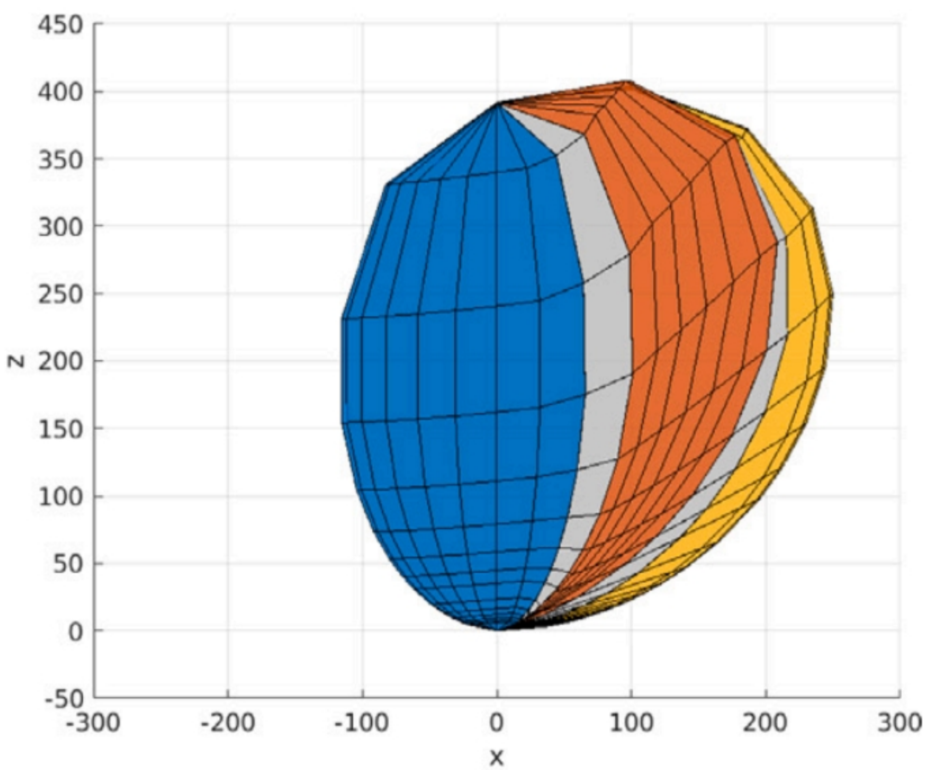

Figure 5: The r-table reconstructed from the 3 ellipsoids.

This model was implemented in a preliminary project to lay the ground work for this one, and yielded encouraging results motivating this one. The figure 6 shows the original reflection indicatrix in black compared to the resulting points of the ellipsoids in colors.

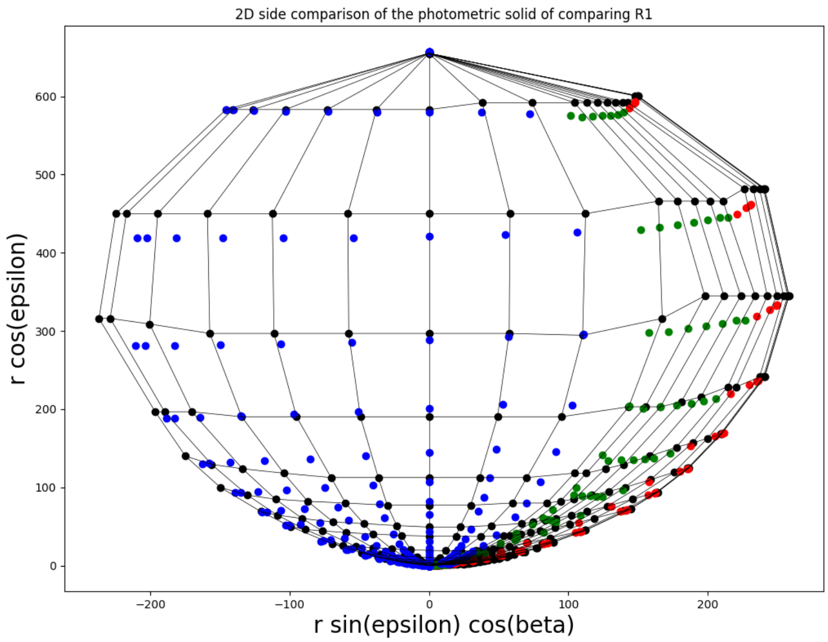

Figure 6: Comparing plot of R4

Of course this model is not perfect, and it has its limitations. Many other approaches to compressing r-tables have been proposed, such as principal component analisys (PCA) like in Boucher et al [3] or Fourier analysis, but this model is the one that was chosen for this project, is it very intuitive for r-tables.

III Tools and methods used

III.1 The databases used for this project

Our aim was to first implement and then improve upon the 3-ellipsoid model and testing it on large datasets of r-tables. A number of databases were provided for this purpose, which are described in this section.

They are excel files composed of several excel sheets, with one r-table in each sheet. The size of the table 29 lines and 20 columns and its name is the name of the excel sheet.

The 14 standard r-tables of the CIE

The CIE has defined 14 standard r-tables in CIE144:2001 [2], which are the r-tables that are used in all lighting design softwares. These typical r-tables are divided in categories depending on their specularity

- 4 types of dry road surfaces (R1, R2, R3, R4) defined in the seventies and still mostly used around the world and in France.
- 2 tables (C1 and C2) defined in 2001.
- 4 additional tables define standards for nordic countries (N1 to N4).
- 4 tables (W1 to W4) to describe wet road surfaces.

The 491 r-tables measured by Kai Sørensen

These measurements were made with a laboratory goniophotometer in the 1970s by Kai Sørensen, a danish researcher. They correspond to 287 different road samples [12]. They were all measured on their dry state. 204 of them were also measured on a defined wet state according to CIE047 protocol[1].

In the database we got, many typos were left in the dataset when it was digitized. The dataset was cleaned up manually by the authors of this project, and the cleaned up dataset was used for the rest of the project (an example in appendix).

The 287 LCPC r-tables

This dataset is a collection of dry surface samples taken from the same road over a period of 3 years. The samples were taken, right when the road was built, and then after 12, 24 and 36 months. Each time the samples were taken, they were always 3 of them, taken always within the tyre tread. A total of 287 samples were taken, and each sample was measured with the Clermont Ferrand Cerema goniophotometer, giving us the dataset [5].

The 40 r-tables in the METAS database

This dataset is a collection of dry surface samples aged of two years or more measured by the Swiss METAS on their gonioreflectometer. They were collected during the Empir SURFACE project and were included in the project dataset [7].

III.2 Programming

The project was written in python 3.11 as per Cerema's request and the following librairies were used along with it:

- argparse, for parsing command line arguments
- numpy, for storing and manipulating data, linear algebra, and trigonometric functions
- matplotlib, to plot the data in 2D and 3D
- pandas, to read and write excel files
- multiprocessing, to parallelize computations

The name "ELLIFANT", given by the author, is given to the program, which stands for "Ellipsoid Fitting and Adjusting Numeric Tool." It is a command line tool designed to accomplish a range of tasks related to ellipsoid fitting of r-tables. Excel files, being the most prevalent format for r-table databases, are processed by this tool. Its functionalities include plotting the data in both 2D and 3D, measuring different usefull data parameters for fitting ellipsoids to the data, adjusting the ellipsoids, and generating new r-tables based on the ellipsoids. The code can be found here for download, along with a few databases already in the correct format and the documentation.

Notably, ELLIFANT computes several data parameters over the tables when adjusting ellipsoids, here is an exhaustive list of all the ones computed on each table, original and the adjusted tables.

- S1, the specular factor
- Q0, the average luminance coefficient computed using [2] ponderation methodology
- Q0, integration calculated using trapezes methodology [4]
- Qd, also an average luminance coefficient but using a different standard [2]
- the smoothness of the r-table, defined as the average slope between two points

- the entropy of the r-table, requested for further research

The next three are distance computations between two tables, used for error with models. They are always computed between two tables, the original and the adjusted one.

- the RMSE the rooted mean squared error between the tables.
- $Delta_{r-table}$ proposed by Florian Greffier [6]
- $Delta_{Q0S1}$, a distance estimator proposed by florian Greffier [6] based on the Q0 and S1 values of the two tables

III.3 Improvements to the 3-ellipsoid model

Despite the 3-ellipsoid model serving as a viable starting point, a more in-depth exploration of the data and the model during my M1 project [10] reveals its limitations in accurately fitting the data .

The rigidity of the model and suboptimal data partitioning along the beta angle pose challenges. In the Ogando-Martinez 3-ellipsoid model, the authors used a beta vector partitioning scheme of $[0, 10], [15, 45], [60, 180]$, based on their observation of inflection points 7 in the data at angles 10, 15, 45, and 60 (see figure 5).

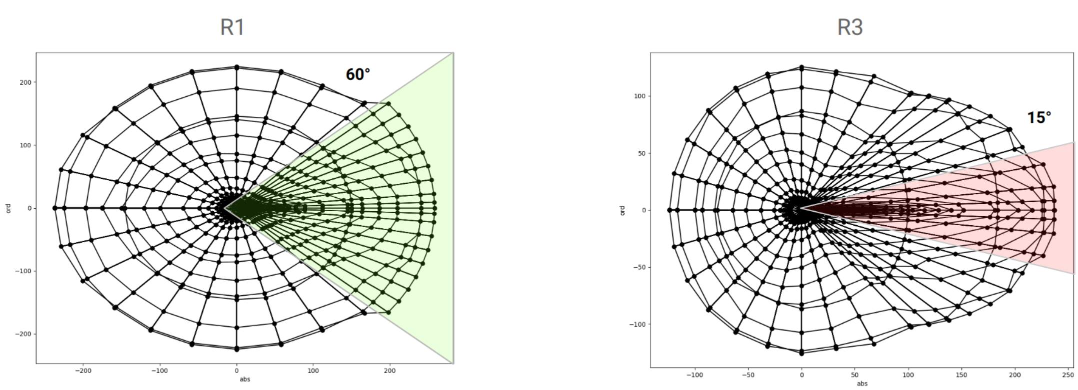

Figure 7: The angles used for the Ogando-Martinez model

However, the paper does not explain how they arrived at this partitioning scheme. To explore alternative approaches, a more flexible partitioning format of can be considered with $[0, \alpha_1], [\alpha_1, \alpha_2], [\alpha_2, \alpha_3], ..., [\alpha_n, 180]$. This approach allows for better partitioning to fit the data more accurately with 3 ellipsoids or, alternatively, using more ellipsoids to achieve a better data fit at the cost of a less compact model and lower data saving rates.

Initially, an exhaustive search was conducted to find the best partitions among the 20 beta vector values for 3, 4, and 5 ellipses. While this provided valuable insights and ideal partitions for our database, the method became impractical as the project progressed, leading us to seek a systematic and automated approach to optimize this problem in larger search spaces.

III.4 Genetic algorithms

III.4.1 General principles

The problem of finding the optimal partitioning scheme for the beta vector can be formulated as a minimization problem. The objective is to minimize the error between the data and the model's reconstruction by adjusting the partitioning scheme. This optimization challenge can effectively be tackled using a genetic algorithm [8].

A genetic algorithm initiates with a population of potential solutions, representing different partitioning schemes in this context. Each individual in the population undergoes evaluation using a fitness function, which quantifies how well the corresponding partitioning scheme performs in fitting the data. The genetic algorithm then discards individuals with lower fitness values, keeping only the fittest individuals.

The next steps involve applying mutations and crossovers to generate new individuals from the surviving ones. Mutations introduce small changes to the partitioning schemes, while crossovers combine features from two or more individuals to create new ones. These processes simulate the natural selection and evolution observed in biological systems.

The algorithm iteratively repeats the evaluation, selection, and generation of new individuals through mutation and crossover. The population grows back to its original size, and the process continues for a specified number of generations or until a stopping criterion is met. Common stopping criteria include reaching a certain number of generations, no observed improvement after a defined number of generations, or achieving a threshold fitness value. Be careful, this last one does not guaranty the termination of the algorithm.

One significant advantage of genetic algorithms is their ability to solve optimization problems without requiring prior knowledge about the problem or data. Since the beta vector is relatively small, with only 20 values, the search space is manageable. Consequently, for validation purposes, an exhaustive search is currently feasible alongside the genetic algorithm to validate its results.

Employing a genetic algorithm in this context allows for automated exploration of the partitioning space, helping to identify optimal solutions and providing a promising method to fine-tune the 3-ellipsoid model for better data fitting.

III.4.2 The beta genetic algorithm

In general we denote a $[0, \alpha_1], [\alpha_1, \alpha_2], [\alpha_2, \alpha_3], ..., [\alpha_n, 180]$ partitioning scheme as a vector $\beta = [0, \alpha_1, \alpha_2, ..., 180]$. So our standard partitioning scheme from the Ogando-Martinez publication is written as $\beta = [0, 15, 60, 180]$.

Since the first and last values of the beta vector are always 0 and 180, to separate the data into n ellipsoids, we have $\binom{20}{n-1}$ possible partitioning schemes. For 3 ellipsoids, we have $\binom{20}{2} = 190$ possible partitioning schemes, and for 4 ellipsoids, we have $\binom{20}{3} = 1140$ possible partitioning schemes. Those search space sizes allowed for exhaustive search of the search space to validate the results of the genetic algorithm.

Here, the genetic algorithm is used to find the optimal partitioning scheme and the ellipsoids that best fit the data. The algorithm generates partitioning schemes of the β

vector by generating vectors of increasing values of beta angles that start at 0 and end at 180. All those partitioning schemes are then tested, by taking the average of their $\frac{dr}{r}$ values, over a database with all our dry road surface r-tables. This value serves as the fitness of the partitioning scheme.

The algorithm then orders all schemes by their fitness in increasing order, and keeps the best half of the schemes, meaning those with the lowest fitness, since it's an error we want to minimize. Each of the winning schemes produces one offspring, by randomly shifting a random amount of its values inside the range of the values of the parent. Cross-over mutations are not done here, as the nature of the partition does not allow for it to give an advantage to offspring. This works just as well as demonstrated in [11]. That is, the offspring of two partitions mixed by say, taking every other stop from another parent would not have a similar fitness to it's parents or even in that range.

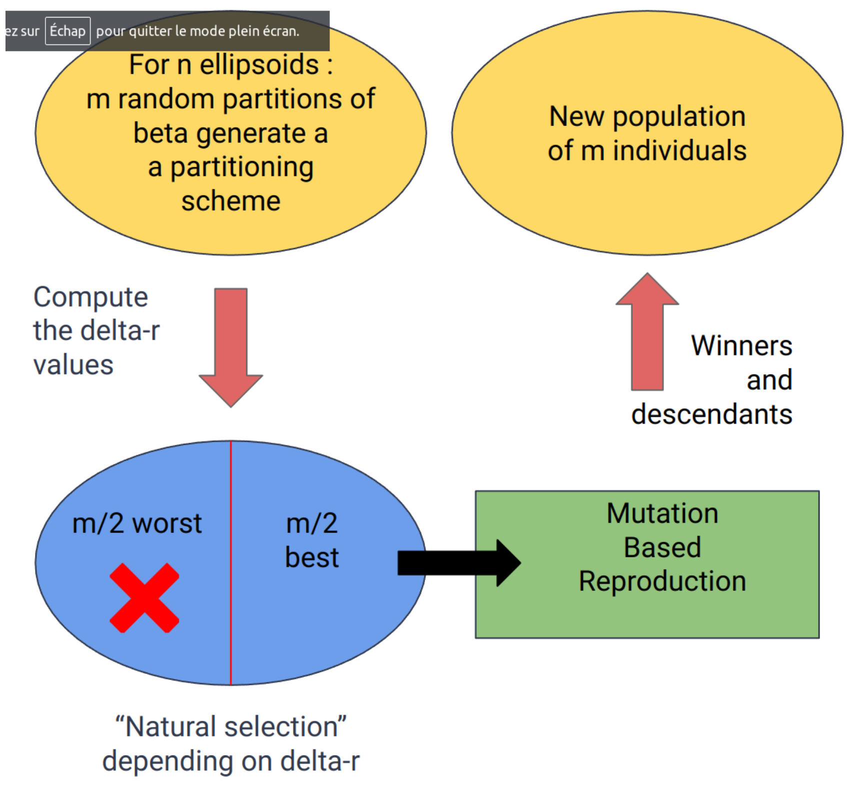

Figure 8: The genetic algorithm explained graphically

The algorithm then repeats the process of ordering the schemes by their fitness, and keeping the best half of the schemes. This process is repeated until one scheme has won a specific number of generation without being beaten.

Once the optimal partitioning scheme has been found it is printed out the console, and saved to a text file if requested.

III.4.3 The partitioning genetic algorithm

After having found the promising results from the first genetic algorithm, it was decided to investigate further. To figure out if the partitioning of the beta vector was the best idea to separate an r-table in n parts to fit it with n ellipsoids, a genetic algorithm was implemented. It would find if partitions of the 580 value-long r-table would fit better.

But finding good partitions of a 580 value-long vector is much harder than finding good partitions of a 20 value-long vector. The search space is much larger since any of the 580 values can be assigned to any ellipsoid, we get n^{580} different possible partitioning schemes for n-ellipsoids partitions.

The individuals are 2-dimensional, and the algorithm will take much longer to run. So this means significant changes had to be made. Instead of generating all the possible completely random partitions, the algorithm generates random partitions using the following method:

1. For n ellipsoid fitting, generate n random points in the r-table
2. Start n parallel four-way flood fill algorithms from those points
3. Once they have all finished, the r-table is partitioned into n parts

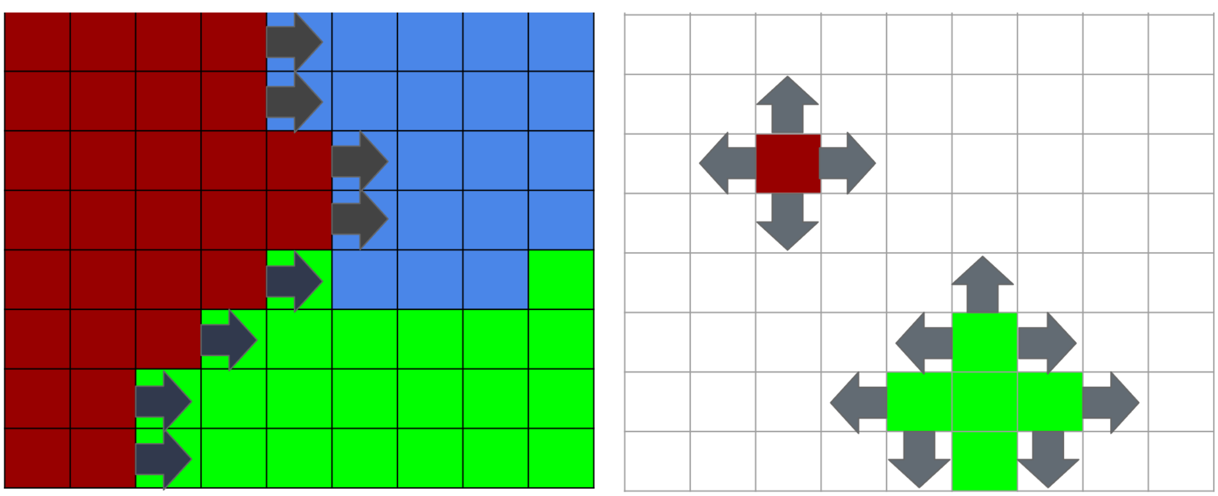

Figure 9: Representations of genetic propagation with two four-way floodfills creating a 2 part-partition of space (on the left) and one mutation of a partition (on the right).

This algorithm enables the generation of partitions for a large search space, effectively reducing it to only partitions where the components are connected and reasonably sized.

The competing four-way floodfills (illustrated in figure 9 left) requires a bit of care, all starting points must be distinct and the points of the $20x29$ matrix must be updated in a random order as to not give an edge to parts over others.

What follows is quite similar to what was done in the previous genetic algorithm, the partitions are ordered by fitness in increasing order, the half with the biggest fitnesses and therefor biggest mean errors get discarded .

Initially, finding a reliable method to mutate the partitions posed a challenge. The complexity arose from dealing with a 2D partition on a 2D surface, eliminating the option to treat the table as a vector by flattening it.

Dividing each part into smaller pieces and assigning them to new regions proved computationally expensive since we needed to test for connectivity while maintaining the same number of parts after any mutation, and generate random boundaries that make sense every single time without weird glitches.

Anything involving crossovers was just as unfeasable as with the beta genetic algorithm. No easy to implement or even think of function seems to take in two partitions and returns one with a similar delta-r value.

To address the risk of mutation destroying one of the parts and generating a partition with fewer than n parts, a simple approach was adopted. This approach guarantees maintaining the number of parts and allows for a customizable mutation rate.

1. For i=0 to the mutation rate
2. Select a random part of the partitioning scheme
3. If one part has less than 30 elements always select it instead
4. Enlarge the part by one line or column in that direction
5. increment i

This method is represented on the right of the figure 9. While the mutation algorithm is not flawless, it proves effective in generating satisfactory partitions. Following the mutation step, the algorithm evaluates these partitions similarly to the previous approach. It retains the best half of the population and continues generating new partitions in each successive generation. This iterative process improves the quality of the partitions over time, even with the limitations of the mutation algorithm.

III.5 Other methodology implemented

III.6 Allowing for a constant term in the equation of the ellipses

Wet r-tables often present a large specular lobe on the right side, corresponding to the water being highly reflective, see figure 10.

The specular lobe in wet road surface r-tables poses a challenge when adjusting ellipsoids constrained to pass through the origin without a constant term. Ana Ogando-Martinez's paper proposes this constraint, but it proves to be less suitable for handling this type of data. A more flexible approach, using freer ellipsoids unconstrained by passing through

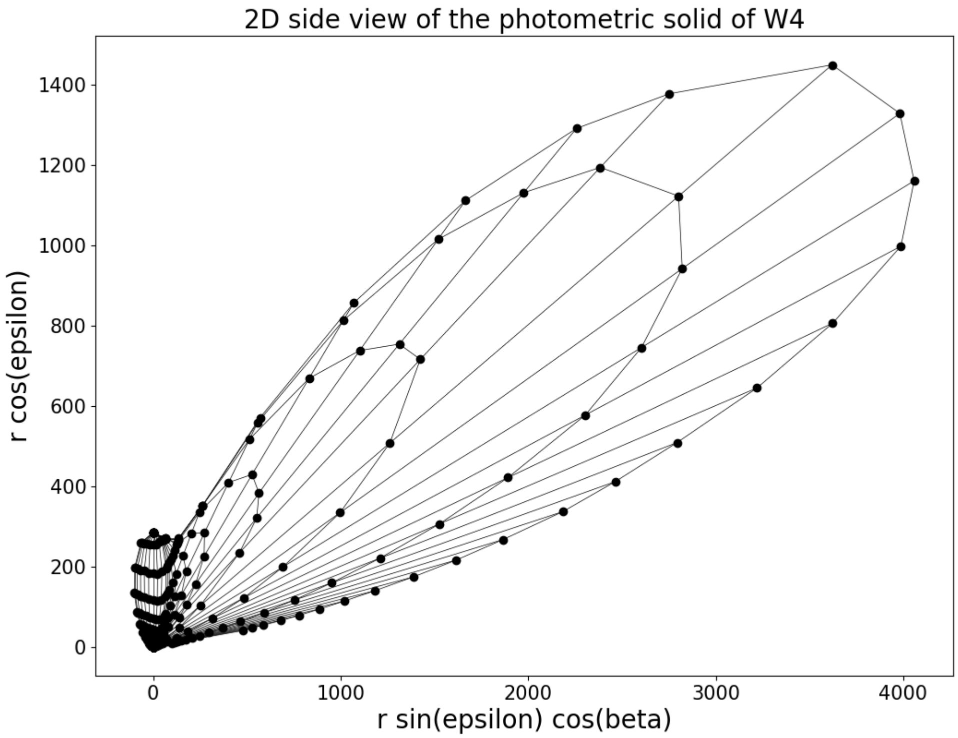

Figure 10: An r-table for a wet road surface

the origin, would likely offer a better fit to the data. However, implementing this approach presented complexities.

In the reconstruction of the r-table using the half-right method from the origin, a significant advantage is that only one solution to the intersection with the ellipsoid is guaranteed. Nevertheless, working with ellipsoids in this context introduces mathematical intricacies, especially as the solution of the quadratic equation is not longer as simple as described in the paper and has 2 solutions of which we have to determine which one is the point on the ellipsoid.

What yielded the best results was using the point closest to the original one from the r-table, but this completely defeats the point of what we are trying to do, since using the data from the original table to rebuild it is quite silly. So in the end the model just uses the point furthest from the origin.This works quite well, because the specular behaviour of r-tables makes the likelihood of the furthest point point being also the best choice quite high, and points are handled by another ellipsoid before this approximation really has any effect on the result.

III.6.1 Adequacy with CIE specific requirements

The original model by the spanish research team does not discuss whether to set the zero values from the original table into the adjusted table manually or not. This was not done for this project during the optimisation search. However it is now added as an option that was used to produce some of the results.

IV Results

IV.1 Original Ogando-Martinez model

The compression rate of the models discussed here is 580:5n with n the number of ellipsoids, so for 3 ellipsoids we get a data saving rate of $\frac{580-15}{580} = 0.974$ or 97.4%. As we mainly focus on improving the error, this value will not change except when the number of ellipsoids changes.

All optimization algorithms ran without transfering standard CIE zeros over to resulting tables.

The model proposed by Ana Ogando-Martinez *et al* was applied to 631 dry road surface r-tables and the distribution of the error measurements is presented in figure 11.

Note: The tables 1 to 286 are taken from the LCPC database, then from the METAS database (index 286 to 327), then from the Sorensen database (index 328), and finally from the CIE dry standard tables starting with index 621.

This gives us an idea of acceptable range for delta-r values, since this error measurement has no unit. Over 80% of the dry r-tables have a delta-r value in the $[0.2, 0.3]$ range.

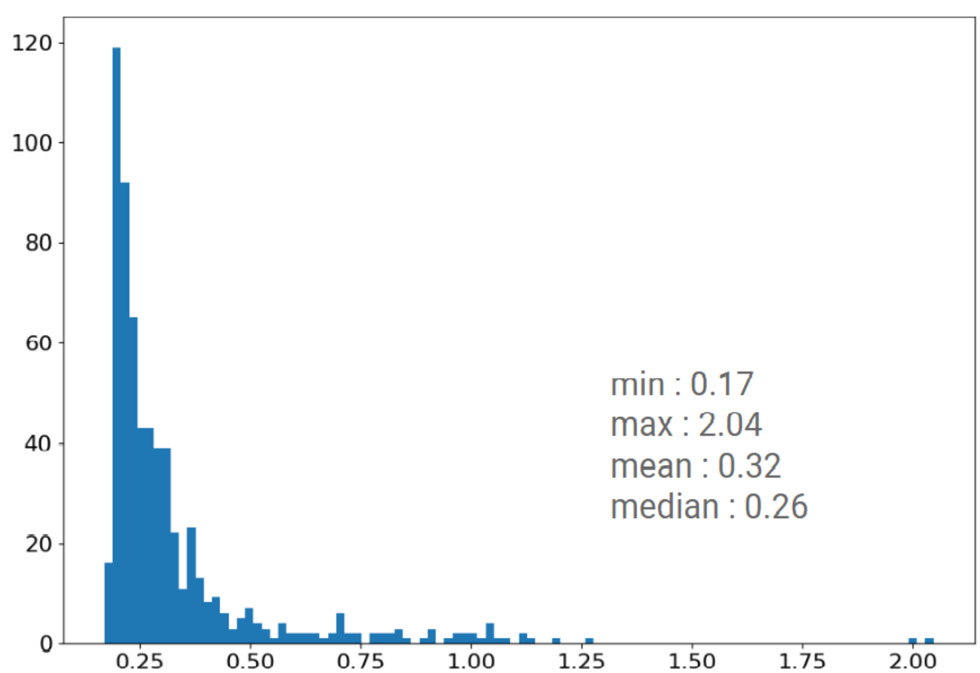

Figure 11: The distribution of delta-r values on dry roads for the Ogando-martinez model, without zero-padding

We also tested the algorithm for the 244 wet tables: 204 from the Sorensen database and the four CIE wet tables. The histograph of the figure 12 presents the obtained results for the wet tables. Wet tables present a mean of 1.8 and a median value of 1.6. Anything above 1 makes a terrible fit for most Wet tables. This demonstrates how not all real data from r-tables can be well fit by ellipsoids.

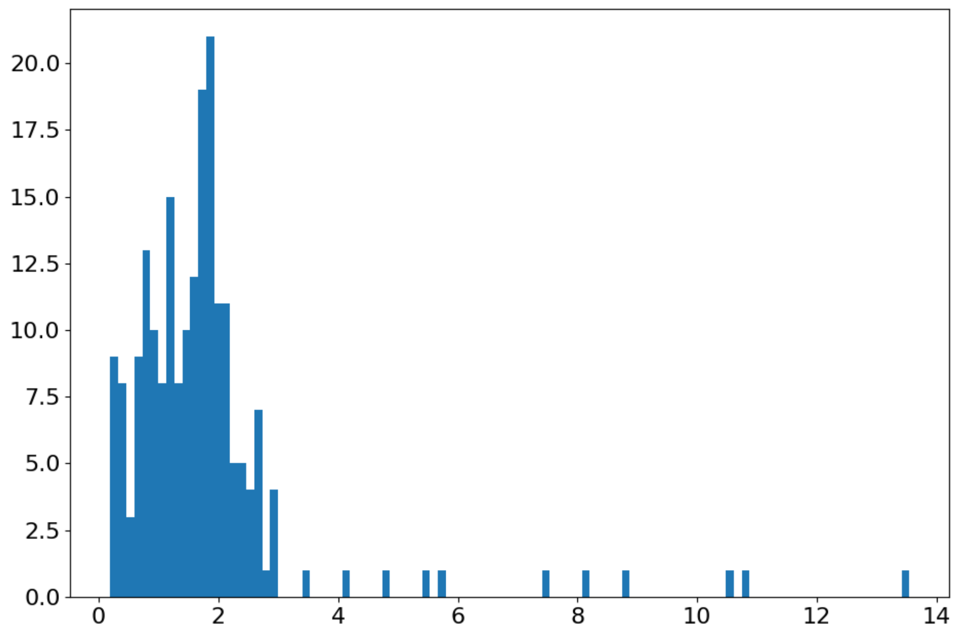

Figure 12: The distribution of delta-r values on wet roads for the Ogando-martinez model, without zero-padding

IV.2 Best partitions along beta for n ellipsoids

To validate the approach of the partitioning algorithm, an exhaustive search through all possible partitions was conducted. For 3 to 5 ellipsoids, the exhaustive search gave the same exact output as the genetic algorithm, confirming that the procedure is suited for the job. With genetic algorithm, the best output was found while searching 11% of the search space, so the genetic algorithm generally converges within only a few generations towards the best solution.

The best partition schemes for beta found for any number of ellipses are presented in the following table, note that these change just the partitioning scheme, still without zero padding.

As we can see, adding ellipsoids to the computation does very little to improve the model, as the biggest improvement in these tests came about when improving the 3-ellipsoid stops vector. This result motivated the decision to explore better partitions of the r-tables to get better fit of the data.

Number of ellipsoids	Stops vector	Mean delta-r	Mean delta-r with CIE zeros
3 (Original mode)	0,15,60,180	0.326	0.317
3	0,15,90,180	0.305	0.304
4	0,15,45,90,180	0.302	0.300
5	0,15,45,60,90,180	0.300	0.299
6	0,15,30,40,60,90,180	0.299	0.299

IV.3 Partitioning results

The partitioning algorithm aims to look for the best partitions of tables to get the lowest mean delta-r coefficient over several r-tables. This model was trained over 631 different dry-road r-tables.

It ran for 3, 4 and 5 ellipses on 1000 individuals per generation, with 100 consecutive generations of 0 improvement as a stopping condition, and still no zero-padding, the results are shown on figure 13.

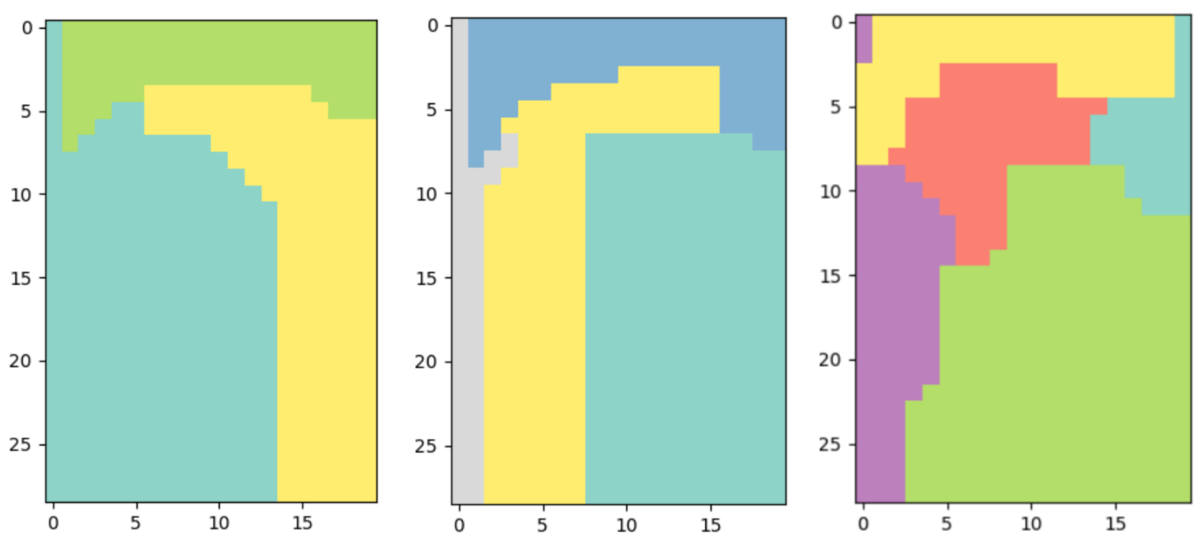

Figure 13: The best partition model for 3, 4 and 5 ellipsoids respectively

Table 1: Comparing the delta-r errors for different models

Model	Mean delta-r
Ogando-Martinez (original)	0,326
Best beta partition (3 ellipsoids)	0,305
Best r-table partition (3 ellipsoids)	0,246
Best r-table partition (4 ellipsoids)	0,236
Best r-table partition (5 ellipsoids)	0.202

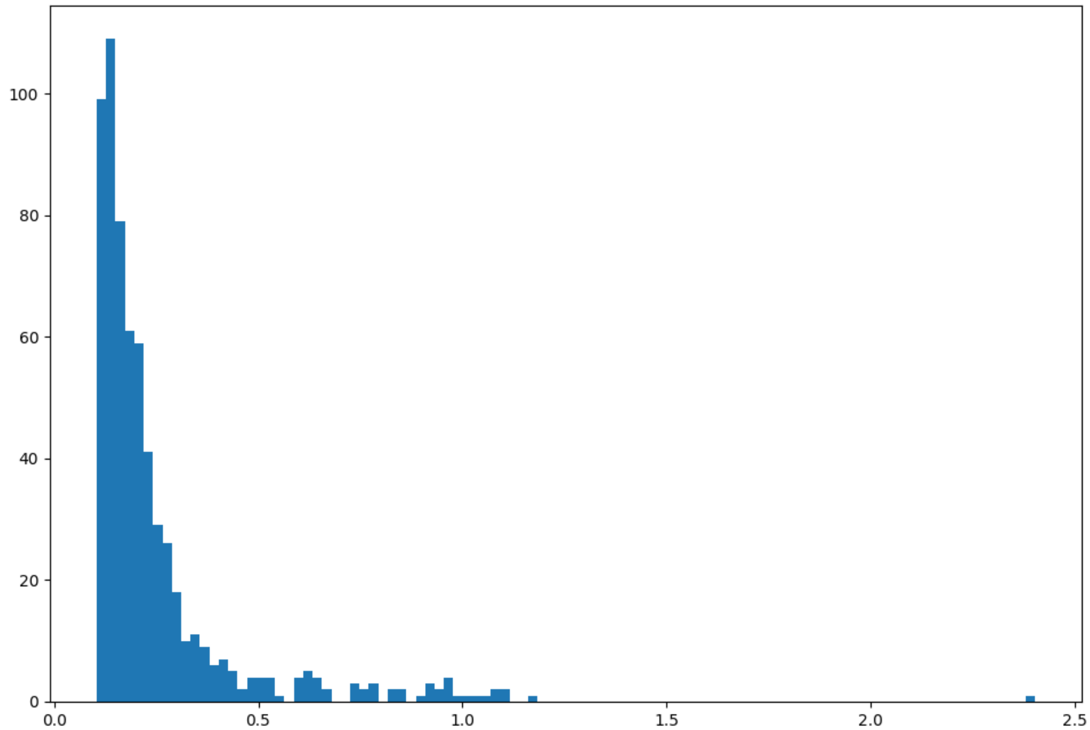

Figure 14: The delta-r distribution for 3 ellipsoid partition

Clearly, since those are results from a genetic algorithm, some post treatment of the solutions would be needed to refine the search around the solution, but as the errors clearly shows we have much better results for the improved model.

Summing up our models errors over dry road surface r-tables, we get the results presented on the table 1.

IV.4 Allowing for a constant term in the equation of the ellipses

When allowing for a constant term in the equations, much of the math gets more complicated, as the equation for the solution of the intersection of the ellipsoid may give two outputs since the half right doesn't start inside the ellipsoid. However once implemented we get the following results:

- For dry r-tables, it seems to perform just a little worse (results on table 2).

- For wet r-tables the data also is kind of peculiar (results on table 3), as the overall effect of this change is even worse than without. However, some r-table see their error drastically reduced, the vector using 130 as a stop is the best 3-ellipsoid beta partitioning for 3 ellipsoids.

Because of it's bad performance, this methodology was not kept for further research and none of the following uses it.

Table 2: Comparing results for dry r-tables with and without a constant term

Number of ellipses	Stops vector	Mean delta-r	Mean delta-r (free constant)
3	0,15,60,180	0.326	0.329
3	0,15,90,180	0.305	0.305
4	0,15,45,90,180	0.302	0.303
5	0,15,45,60,90,180	0.300	0.302
6	0,15,30,40,60,90,180	0.299	0.304

Table 3: Comparing results for wet r-tables with and without a constant term

Number of ellipses	Stops vector	Mean delta-r	Mean delta-r (free constant)
3	0,15,60,180	1.781	2,012
3	0,15,130,180	1.672	1.806

IV.5 Comparison of the different modelisation techniques

Given that r-tables are relatively small datasets with only 580 values, using image compression techniques can lead to border distortions and amplification effects, making implementation challenging. The ellipsoid model, as an alternative approach, demonstrates adaptability and effectiveness in handling such datasets.

All following results are presented with zero padding, transfering standard CIE zeros over to resulting tables. We compare the following models:

- 3 ellipsoid original Ogando-Martinez technique
- 3 ellispoid best beta partition
- 3 ellipsoid best partitioning scheme
- 5 ellispoid best beta partition
- 5 ellipsoid best partitioning scheme

It is important to note that a significant improvement in the compression rate should not be expected since it is inherently linked to the equation of ellipsoids, which requires a certain number of parameters. However, the error can be substantially reduced by employing better partitioning schemes compared to those proposed in the original 2020 publication.

To achieve the best data saving rate of 97.4% with 3-ellipsoid models, the 3-ellipsoid partitioning scheme at figure 13left should be utilized, as it results in a considerable reduction in mean delta-r compared to the original model (approximately 21% less error).

Alternatively, if one is willing to accept a lower data saving rate of down to 95.7% for a greater error advantage, the 5-ellipsoid optimal partitioning scheme of figure 13right becomes a preferred option. This choice leads to an even more substantial improvement,

with an average error reduction of 35% compared to the original model.
For each model, results are presented for all available dry r-tables and deviations from the initial table are calculated and presented on the figure 15 for S1 and Q0, figure 16 for smoothness and entropy and figure 17.

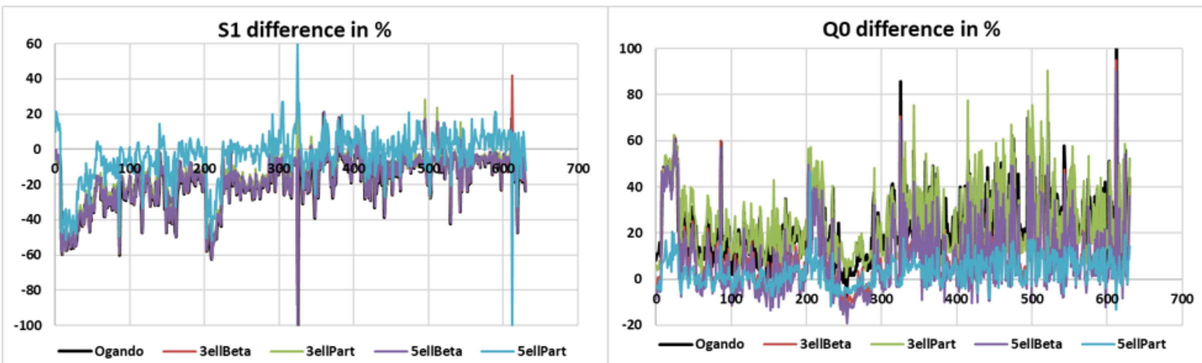

Figure 15: Computation of difference in percent with all the original dry tables and the different applied models. S1 is on the left and Q0 on the right.

As shown in figure 15, the original Ogando model strongly underestimates S1 specularity, as do the Beta 3 and 5 ellipsoid models and the 3-partition model. However, with a 5-ellipsoid partition, this effect is decreased. The mean luminance coefficient Q0 is generally overestimated by the models. Here again, the best results are obtained for the 5-ellipsoid partition (evt put average stats).

Note : The standard calculation for Q0 and S1 don't change at all regardless of zero-padding, since none of them use any of the 188 values concerned with this standard.

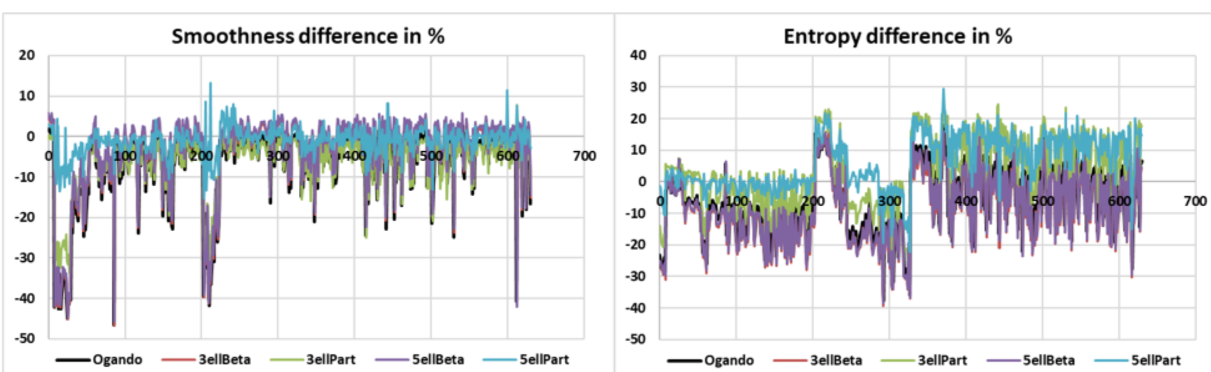

Figure 16: Computation of difference in percent with all the original dry tables and the different applied models. The smoothness is on the left and the entropy on the right.

In terms of smoothness, the best results are obtained for the 5-ellipsoid partition. However, for entropy, it's true for the LCPC and METAS databases, but not for Sorensen's or the standard tables. Finally, the RMSE and delta-r error are quite close between the

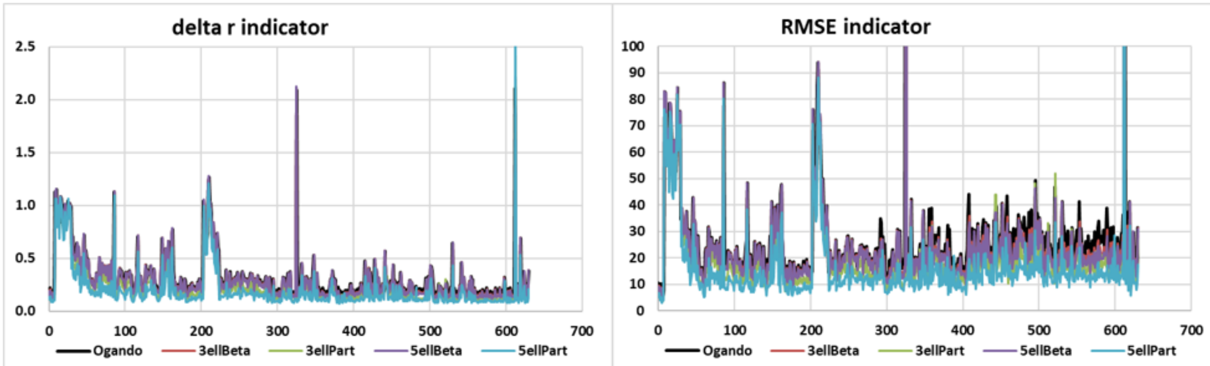

Figure 17: Computation of difference in percent with all the original dry tables and the different applied models. The deltaR is on the left and the RMSE on the right.

original model and better beta-partitionings, but significantly better when using optimal partitioning schemes.

A comparison of the original Ogando Martinez model and the better 5 ellipsoid partition model is done on the figure 18 for R1 standard table and on the figure 19 for R4 standard table.

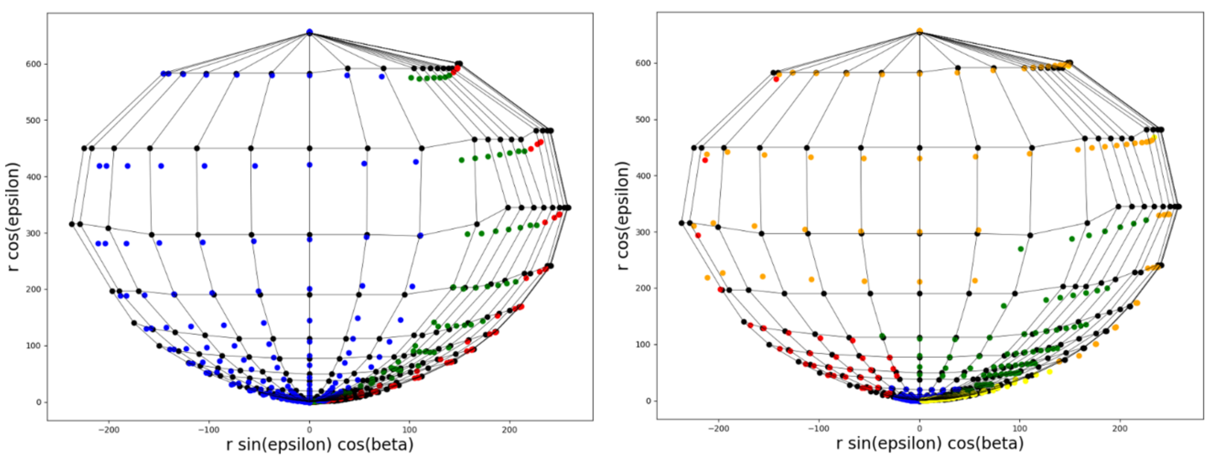

Figure 18: In both representations, the original R1 table is in black and the fitting results are represented in colors. Ogando-Martinez fitting is on the left picture and the Optimal 5 ellipsoid partition fitting is on the right picture

The adjusting of R4 by different models clearly shows what these better partitioning schemes improve in practice. The column at $\beta = 0$ getting to be fit with it's own ellipsoid corresponds to the right outer arch of the reflectance indicatrix showing a lower value at small ϵ angles. Also the plateau for smaller values is better fit through the optimal partitioning scheme, since the Ogando-Martinez model tends to just show a flat constant plateau at the bottom, where one should observe a slight gradient, see figure 19.

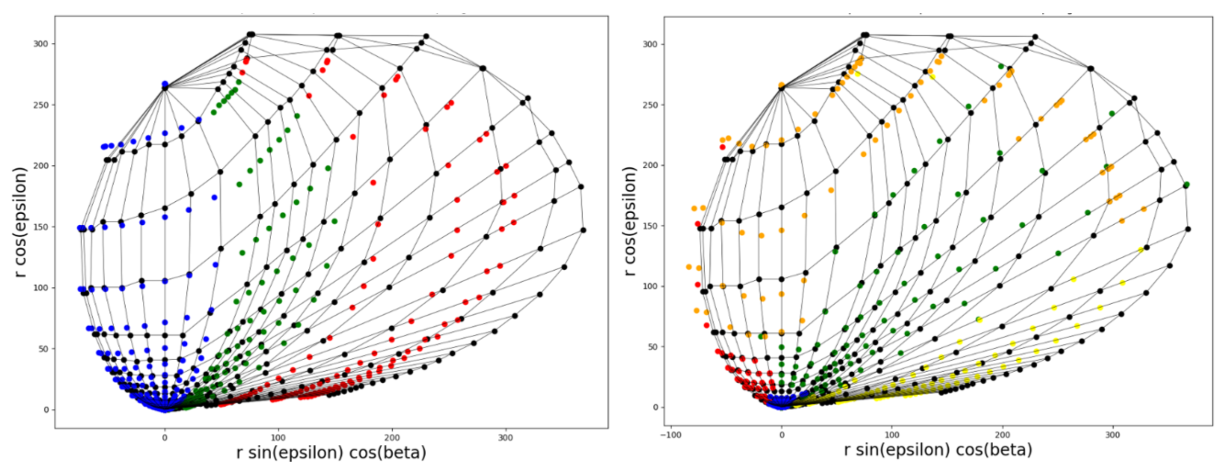

Figure 19: In both representations, the original R4 table is in black and the fitting results are represented in colors. Ogando-Martinez fitting is on the left picture and the Optimal 5 ellipsoid partition fitting is on the right picture

V Conclusion

The ellipsoid model demonstrates the successful implementation of an intuitive concept for data approximation. This project enhanced the original model by employing finer and more precise partitioning schemes suitable for large volumes of dry r-tables.

Overall, the optimal choice among the models is the generalized partitioning approach using 5 ellipsoids. This model notably outperforms the original Ogando-Martinez proposal, yielding an average reduction of 35% in mean delta-r error. It also gives better results for clarity and specularity coefficients (Q0 and S1). The only trade-off is a slightly lower compression rate due to the increased number of ellipsoids used.

However, the application of any ellipsoid model to wet r-tables is not recommended, as all models produce important delta-r errors (more than one), indicating unsatisfactory fits. Although optimal partitioning schemes mitigate this error considerably, they do not produce acceptable results.

Despite the bad outcomes observed with the introduction of free constant terms for ellipsoids, notable error reduction was witnessed in specific wet r-tables. This phenomenon warrants deeper exploration.

To achieve better results for the dry tables, the optimal partitions determined by the genetic algorithm should be refined by more complex post-processing techniques around the solution.

For future research, the impact of adding zeros needs to be studied in detail. Although it is unlikely that significant changes will be made to optimal partitioning schemes with the addition of a zero, this remains to be verified.

It is also essential to investigate through lighting calculations how tables adapted to ellipsoids influence measurements of mean luminance and luminance uniformity. This study will highlight the impact of the approximations made using ellipsoids. The photometric characteristics of road surfaces can be spatially heterogeneous. We will also investigate the robustness of the ellipsoid model to this heterogeneity.

Moreover, it is recommended that the models be thoroughly tested on fully measured r-tables without the influence of the standard CIE 188 zeros.

Finally, these refined models should be employed to analyze the evolution of road surface reflectivity over time. This long-term assessment will provide a better understanding of the durability and behavior of road surfaces under various environmental and usage conditions.In summary, the ellipsoid model, enhanced by improved partitioning schemes, offers an efficient approach to accurate data approximation and error reduction in r-table fits Its adaptability makes it a valuable tool for managing small datasets with remarkable success.

Bibliography

[1] CIE 047. CIE 047:1979 Road lighting for wet conditions. Technical report, CIE Vienna, 1979.

[2] CIE 2001. CIE 144:2001 Road Surface and Road Marking Reflection Characteristics. Technical report, CIE Vienna, 2001.

[3] Vincent Boucher and Florian Greffier. Space of basis functions to retrieve road surface reflection tables from luminance maps. *Lighting Research & Technology*, page 147715352210942, July 2022.

[4] Vincent Boucher, Valérie Muzet, and Paola Iacomussi. Mathematical considerations for road reflection properties. page 10, Ljubjana, Slovenia, September 2023. CIE.

[5] Eric Dumont. *Photométrie des chaussées et éclairage public*. Études et recherches des Laboratoires des ponts et chaussées (ERLPC). Série Chaussées, CR45, March 2007.

[6] Greffier Florian, Muzet Valérie, Boucher Vincent, Fournela Fabrice, Lebouc Laure, and Liandrat Sébastien. Influence of pavement heterogeneity and observation angle on lighting design: study with new metrics. *MDPI*, 2021.

[7] Henrik Gidlund, Mikael Lindgren, Valerie Muzet, Giuseppe Rossi, and Paola Iacomussi. Road Surface Photometric Characterisation and Its Impact on Energy Savings. *Coatings*, 9(5):286, May 2019. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.

[8] Ahmad Hassanat, Khalid Almohammadi, Esra'a Alkafaween, Eman Abunawas, Awni Hammouri, and V. B. Surya Prasath. Choosing mutation and crossover ratios for genetic algorithms—a review with a new dynamic approach. *Information*, (12), 2019.

[9] Ana Ogando-Martínez, Francisco Troncoso-Pastoriza, Enrique Granada-Álvarez, and Pablo Eguía-Oller. Ellipsoid-based approximation method for the estimation of the actual reduced luminance coefficients of road surfaces for accurate lighting simulations. *Sustainable Cities and Society*, 63:102502, 2020.

[10] Carl Schmittbiel and Valérie Muzet. Project reflectivity cerema, reimplementing the 3-ellipsoid model in python. Technical report, CEREMA, 2023.

[11] Rich Caruana Shumeet Baluja. Removing the genetics from the standard genetic algorithm. 1995.

[12] Kai Sørensen. Description and Classification of Light Reflection Properties of Road Surfaces, Report n°7, 1974.

VI Appendix

VI.1 An r-table from the Kai Sorensen database with an obvious typo

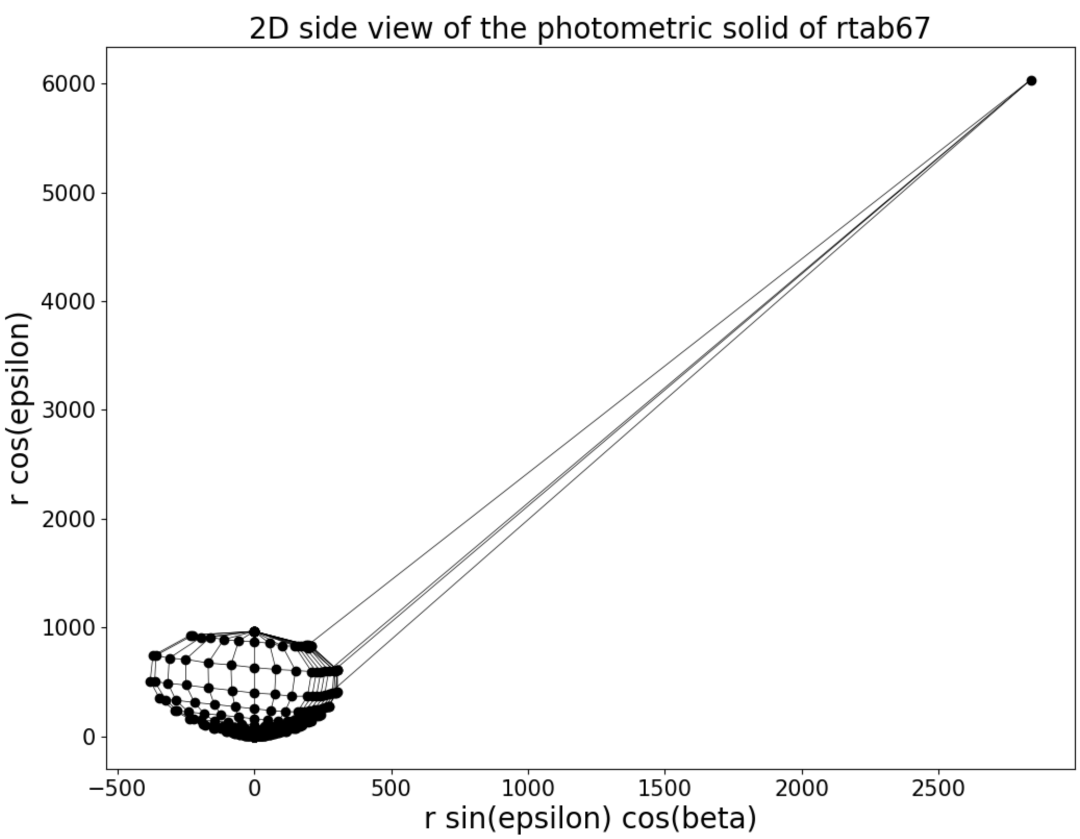

Figure 20: An r-table with a typo

ELLIFANT - An application for all your r-table ellipsoid fitting needs

ELLIpsoid Fitting and Adjusting Numeric Tool, or ELLIFANT, is a python application that allows users to adjust ellipsoids to their r-table measurements, and find the best fitting partitions for their data.

For this guide, basic understanding of what an r-table is and how it is used is assumed. For more information on r-tables, please refer to (fill in later).

We also assume good knowledge of the 2020 paper by Ana Ogando-Martinez *et al.* , which describes the ellipsoid fitting algorithm that serves as a basis for this application.

Installation

To install ELLIFANT, simply clone the repository and put it in a directory of your choice. You will need to have Python 3.7 or higher installed on your machine, and use pip or conda to install the following packages:

- numpy
- pandas
- matplotlib
- argparse
- multiprocessing

Further instructions on how to install these packages can be found on their respective websites:

- Python
- pip

Pip should be all you need to install the 5 packages, just type `pip install <package_name>` in your terminal once you have installed Python and pip.

Usage

ELLIFANT is a command line application, so you will need to use your terminal to run it.

Use `cd` to navigate to the directory where you have installed ELLIFANT, and down into the src folder.

To run ELLIFANT, simply type `python3 ellifant.py` in your terminal. For windows machines, you may need to type `py ellifant.py` instead.

But most likely, this will return you an error message, as you need to specify what data you want it to run on and what you want it to do with it, adding the -help flag will show you the different options you have, but here is a quick rundown of the arguments you can use.

You can put all command line arguments in any order you like, some values will return errors, but the order of the arguments themselves does not matter.

You can use either the short or long and comprehensive version of the arguments, for example, -ea and --ellipsoidAdjusting are interchangeable.

Once you are happy with a result from a computation, make sure to save it, as the results are not saved by default, and will be overwritten if you run another computation. This is to avoid having to delete old results manually, and acumulating a lot of useless data because of a loose command line call.

Functions

The app has 2 main functions, fitting ellipses and searching for the best fitting partitions, either along the beta vector or over the whole r-table. (see the report for more information on the algorithm)

- -ea | --ellipsoidAdjusting : Adjusts the ellipsoids to the r-table measurements according to the stops vector or a partition
- -bga | --betaGeneticAlgorithm : Searches for the best fitting partition of beta for the provided data using a genetic algorithm
- -pga | --partitionGeneticAlgorithm : Searches for the best fitting partition of the whole r-table for the provided data using a genetic algorithm

You must specify one and only one of these functions per command line call.

Boolean arguments (no value needed, typing them will set them to True)

Do not forget the -sd argument if you want to save your results !

- -sd | --saveData : Saves the results of your computations in different formats depending on what you requested, the files will be saved in the results folder of the application (or the folder you specified with the -sf argument).
- -si | --saveImages : Saves the images you want according to the plotTypes and styles you requested, the images will be saved in the images folder of the application.
- -sh | --showImages : Shows the matplotlib plots as they are computed, this is useful if you want to see the plots but do not want to save them, this is compatible with the saveImages argument.
- -fc | -- freeConstant : This is a boolean to determine weather or not you want to use the free constant in the ellipsoid fitting algorithm, it is false if you don't specify it.
- -ct | --coloredOriginalTable : This is a boolean to determine weather or not you want to color the original r-table according to the partition you use, it is false if you don't specify it. This is useful if you want to see how the partition you are using affects the r-table.

- -cz | --conserveZeros : If you want to keep the historical zeros of r-tables in the results of your computations, you can use this argument, it is false if you don't specify it. This of course brings down the error of the ellipsoid fitting.
- -v | --verbose : makes the application print more information about what it is doing, this is useful if you want to see the progress of the computations, but it can be a bit overwhelming if you are running a lot of files.

More command line arguments

- -fp | --folderPath : Path to the folder containing the r-tables you want to run ELLIFANT on, the path should be relative to the data folder of the application, ELLIFANT comes with a few databases to toy with, but you can add your own next to them.
- -p | --patron : Patron of the .xlsx folders you want to run ELLIFANT on, for example, `-p *.xlsx` is the default value, since the app only parses excel files, but you can specify a more specific patron, for example, `-p *-rtable.xlsx` will only run ELLIFANT on files that end with -rtable.xlsx
- -s | --sheets : If you only want to parse a few sheets of the files, you cans specify them here, for example, `-s R1,R2,R3` will only parse the sheets R1, R2 and R3 of the files.
- -i | --ignore : This is the opposite of the sheets argument, it allows you to ignore certain sheets, for example, `-i R1,R2,R3` will parse all sheets except R1, R2 and R3. Sheets named `Feuil1` are ignored by default, as they are usually used as info about a database.
- -sf | --saveFolder : This is the folder where the results of your computations will be saved, it is the results folder of the application by default, but you can specify a different folder if you want to.
- -sif | --saveImagesFolder : This is the folder where the images of your computations will be saved, it is the images folder of the application by default, but you can specify a different folder if you want to.
- -sdn | --saveDataName : This is the name of the file where the results of your computations will be saved, it defaults to `results`.
- -pa | --partition : This describes how you want to partition your data, it defaults to `0,15,60,180` just like in the paper, but you can specify a different partition if you want to, for example, `-pa 0,10,90,180` is known to be the optimal partition for r-tables on dry road surfaces.
- -g | --genetic : This is a vector of three values that controls the genetic algorithm, the first value is the number of ellipses you want to fit to the r-table, the second value is the number of generations one individual has to win in a row to be considered the best (the genetic algorithm always terminate at 10,000 generations), and the third value is the number of individuals you want to have in your population, it defaults to `3,5,30` as to not use up too many ressources, but to get satisfying results, you should use at least `n,20,100` for any n.
- -ps | --plotStyle : This is the style of plot you want to save or show with matplotlib, it defaults to a 2d side view plot, but you can specify a different style if you want to, for example, `-ps 2D_top` will plot the photometric solids from the top, and `-ps 3D` will plot the photometric solids in 3D. Additionally, adding `q` before the style will plot the photometric solids for the corresponding q table instead of the r table, for example, `-ps q2D_side` will plot the standard 2D side view plot for the q table.

- -pt | plotTypes : This controls what you want to plot: `-pt o` will plot the original r-table, `-pt r` will plot the resulting r-table after the fitting of ellipsoids, and `-pt c` will plot a comparison between the original and resulting r-tables. You can combine these arguments, for example, `-pt or` will work just as you would expect it to.

Examples

- To see your favourite r-table in 3D, just type `python3 ellifant.py -ea -fp <path_to_your_file> -p <your_excel_file> -s <your sheet> -pt o -ps 3D -sh` in your terminal.
- To see the results of the ellipsoid adjusting on your r-table in 2D from the top, just type : `python3 ellifant.py -ea -fp <path_to_your_file> -p <your_excel_file> -s <your sheet> -pt r -ps 2D_top -sh` in your terminal.
- To look for the best partition to sum up the r-tables from your database, just type : `python3 ellifant.py -pga -fp <path_to_your_directory> -g 3,5,30 -sd -si` in your terminal.