Approximating a full Bidirectional Reflectance Distribution Function from a slice

Creating a BRDF trough solids of revolution

Gino Tramontina[1]

Supervisor(s): Ricardo Marroquim[1]**, Yang Chen**[1]

[1]EEMCS, Delft University of Technology, The Netherlands

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
June 25, 2023

Name of the student: Gino Tramontina
Final project course: CSE3000 Research Project
Thesis committee: Ricardo Marroquim, Yang Chen, Daniël Pelsmaeker

Abstract

Bidirectional Reflectance Distribution Functions, BRDFs, describe the reflectance of light on a material, and are widely used in computer graphics to render materials. Acquiring a full measured BRDF can be costly and time consuming, so this research aims to answer the question "How can we approximate a full BRDF from a single slice (in-plane BRDF)?". Outlined in this paper is an algorithm that uses solids of revolution to approximate a full BRDF from a single slice. The algorithm finds sub-curves of the slice, creates solids of revolution for each, normalizes the data, and merges the solids while removing overlapping data. The resulting solid, described by a list of points using a Cartesian coordinate system, represents the full, three-dimensional BRDF.

1 Introduction

In computer graphics a common task is to render multiple objects comprised of different materials. To model the appearance of a material a common method is to determine the amount of light that reaches an observer when light hits the material. A function that is frequently used for this purpose is a Bidirectional Reflectance Distribution Function (BRDF). A BRDF provides the fraction of light reflected in a certain direction based on the direction of incoming light for a specific material [2].

In order to determine a BRDF a couple of methods can be used. Traditionally a BRDF is either determined by using an analytical model, such as Blinn-Phong or Oren-Nayar [11], or by measuring the reflectance of a material for each combination of incoming and outgoing directions of light [9][10]. While this second method results in accurate representations of the material, it is quite time consuming.

A slice of a BRDF, also called an in-plane BRDF, is a two-dimensional slice of a BRDF. In a full, three-dimensional BRDF the direction is in a three-dimensional space and the direction can be represented by two angles: the elevation ϕ, and the azimuth (θ), which represents the rotation around the normal. For an in-plane BRDF θ is fixed, and as such the only angle that can vary is ϕ. Figure 1 depicts a point in both a three-dimensional and an in-plane BRDF with all the variables shown.

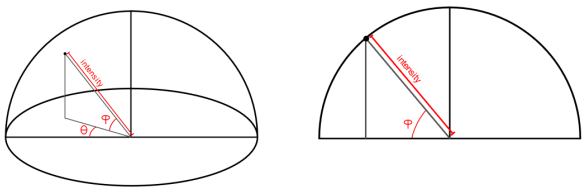

Figure 1: A three dimensional and in-plane BRDF with one point.
Left: A full, three dimensional BRDF with one point.
Right: An in-plane BRDF with one point.

While using measured BRDFs can be enticing due to the accuracy of the data, the financial costs and time requirements can be prohibitive. To combat these concerns a less costly or faster method of acquiring a measured BRDF can be quite beneficial. One way to achieve this is to approximate a full BRDF from a single slice of the data. This paper will therefore focus on the question: "How can we approximate a full, 3D BRDF for a material from one slice (in-plane BRDF)?"

One option of approximating a BRDF from a single slice is trough the use of solids of revolution. Solids of revolution are the solid figures created when rotating a curve around an axis [13]. Solids of revolution can thus be used to create three-dimensional shapes from two-dimensional curves. This application of solids of revolution can be utilized to a full, three-dimensional BRDF by creating a solid of revolution of the two-dimensional, in-plane BRDF.

To approximate a BRDF through the generation of solids of revolution one needs to first be able to find the sub-curves that comprise an in-plane BRDF. This is needed in order to create a solid of revolution for each curve. These solids can then be added together to create an approximation of the full BRDF that is represented by the slice given.

There are a couple of issues that need to be addressed, however. One issue is the fact that this method can leave areas of the BRDF empty when no sub-curve spans the entire width of the in-plane BRDF. Another issue arises when sub-curves are not purely parabolic functions, thus requiring both a more complex method of finding sub-curves, as well as requiring the determining of non-vertical axes of revolution in order to create a solid that most accurately reflects the BRDF provided.

These issues, as well as the full research question, lead to the following sub-questions:

- How to create a simple BRDF as a solid of revolution of a simple curve?
- How can solids of revolution be combined to create a more complex BRDF?
- How can complete coverage of the BRDFs domain be ensured?
- How can an in-plane BRDF be divided into sub-curves that can be used to create the full BRDF?
- What method can be used to find non-vertical axes of rotation for more complex sub-curves?

The first segment of this paper is the introduction. Here background information will be discussed, as well as the relevancy of the research detailed in this paper, the core research question, as well as the sub-questions that comprise this main question.

The second segment of this paper details the methodology used in conducting this research. This section will include the general plan that was used to conduct this research, explain the steps taken during the research to reach the result gathered by this research, and discuss the actual course of the research as it was conducted.

The third section of this paper describes the results of the research.

The final part of this paper are a set of discussions. This discussion will contain the limitations of the research that was conducted, a discussion regarding possible improvements that can be made, a list of future work that can follow from this research, and how this research improved and expanded on existing research. This section will also include a concise summary of all the results, direct answers to the research question and it's sub-questions, and discus the ethical implications of the method in which the research was conducted and the results it gathered.

2 Methodology

To answer the research questions there are a number of steps to take in order. Each step answers a sub-question, with the core research question being answered at the end. The exact steps are as follows:

1. Create a simple, three-dimensional plot of a solid of revolution made from a simple curve, such as a quadratic or sinusoidal function.
2. Combine multiple solids of revolution into a single solid while both removing redundant/ unnecessary data and ensuring that the resulting solid is still representative of the input.
3. Find the sub-curves that comprise an in-plane BRDF while ensuring coverage of the entire domain of the BRDF by the resulting solids of revolution.
4. Find non-vertical axes of rotation for complex curves for which vertical axes of rotation would not result in representative BRDFs
5. Extract sub-curves from an in-plane BRDF that has multiple data-points with differing y-coordinates and equal x-coordinates.

2.1 Simple solids of revolution

The first step in this research is to determine a method for creating a solid of revolution from a simple curve. Examples of such curves are quadratic functions or sinusoidal waves. These types of curves are simple to create into solids of revolution due to their inherent symmetry.

The first issue that needs to be addressed is how data is represented. This algorithm requires a list of two-dimensional points using Cartesian coordinates, sorted in the order of which they would appear when following the trajectory of the curve the points represent. An example of this ordering is shown in figure 2.

For each point the values for θ,ϕ , and the intensity are then calculated. These values will later be used for comparing points with differing coordinates that represent a vector in the same direction. For these values the point is interpreted as a vector from 0 to the coordinates of the points. θ is measured around the z-axis, with the x-axis in the positive direction representing zero and the angle increasing with a counter-clockwise rotation, as can be seen in the unit circle

cite unit circle

. ϕ is the angle between the vector and the x,y-plane, which can hold values between 0 and 90, these values represent the

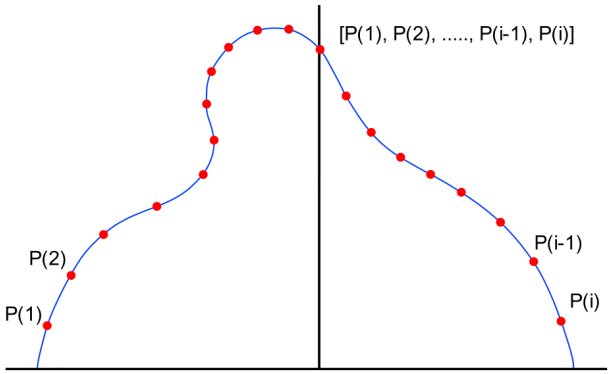

Figure 2: Points (Red) representing a curve (Blue) along with their ordering

vector being parallel and orthogonal to the surface respectively. The intensity is the length of the vector, measured by the distance from the point to 0.

using all the given points, the algorithm then finds the axis of rotation for the curve. For simple curves a measure of symmetry is assumed in a way that the distribution of points in the first half of the curve is similar or equal to the distribution in the second half of the curve. Assuming this the axis is determined by taking the vertical line through the highest point of the curve, or trough the mean of the two highest points if there are two points that are equally high.

The last step for this part of the algorithm is simply rotating each point around the axis, creating a ring of points for each point on the curve. These rings are combined into a single set of points to create the solid of revolution for the curve.

2.2 Combining solids

The next step is to find a way to combine multiple solids. This is needed due to the limitation that a full, complex BRDF cannot be created through the revolution of a complete in-plane BRDF ¡include figure of incorrect BRDF¿. This can be mitigated by creating multiple solids of revolution, and then combing them to create a single, complex shape.

Simply combing multiple solids is the not whole problem here, however. By simply adding two solids that overlap you might end up with data in the final BRDF that should not be present, but is created through the rotation of one of the sub-curves. Because of this issue this step is focused on combining solids while removing data that should not be present in the final BRDF.

To address this issue the algorithm does not remove points based on overlapping x and y coordinates, as two points can have differing coordinates while still representing a vector in the same direction. To accomplish this the algorithm compares the angles of two points and removes the point with the lower intensity, as this points would not be visible in the resulting BRDF due to the other point representing a higher intensity of light already being reflected in its direction.

Comparing all points with each other is not desirable, however. Doing this is a brute-force method that results in an O(n^2) worst-case time complexity, with n as the number of point in the solid. To alleviate this problem the algorithm

takes two steps:

1. Sorting the data.
2. Normalizing the data.

The algorithm sorts the data so it can compare points to only points that are close to it. Simply sorting the points does not fully alleviate the issue, however. All point can still have slightly differing angles, which result in needing to define which points are ”close” to each other in order to only compare points that would be considered as overlapping. Depending on the definition this can, in the worst case, still result in comparing all points to each other, if they are close enough to each other.

To address this, the algorithm normalized the data based on the direction of the vectors represented by the points. The algorithm does this by adjusting a point so it points in a direction that is distributed evenly based on θ and ϕ.

By first normalizing and then sorting the data the algorithm will compare points only if they have equal directions. Additionally, the algorithm only has to go trough all the points once, as it can linearly compare all points with equal directions, only keeping the point with the highest intensity. Doing this for all direction lowers the number of comparisons made significantly, lowering time complexity of the comparisons to O(n), meaning this entire process has an O(nlogn) worst-case time complexity, as sorting the data is done in O(nlogn) time complexity [1].

2.3 Finding sub-curves

A crucial part of this algorithm is creating a three-dimensional BRDF from a single in-plane BRDF. This could be achieved by splitting the curve represented by the slice into multiple sub-curves. These sub-curves can then be used to create solids of revolution, which can be combined into a single, full BRDF.

To accomplish the division of the slice into multiple sub-curves the algorithm needs to assume a number of aspects of the curve present in the slice. The assumptions are as follows:

- The slice is a ”smooth” curve.
- Sub-curves begin when the curve changes direction.

The algorithm assumes that the curve represented by the slice is a smooth curve, which means that the curve does not have a large number of small bumps, which would give the curve a wave-like quality. When this algorithm is given a curve that is not smooth, it determines every bump as a sub-curve, resulting in a BRDF where a lot of data had to be filled in, due to the combined solids of revolution not covering a large part of the domain. An example of these types of curves and its sub-curves is shown in figure 3.

Additionally the algorithm assumes that the curve can be split into sub-curves based on the acceleration of the curve. The acceleration of a curve at a given point is the change direction the curve travels, found trough calculating the second derivative of a mathematical curve. The algorithm splits a sub curve on areas where acceleration changes from negative (trending towards a descending angle), to positive (trending towards an ascending angle), and back to negative. The split of between the sub-curves is at the median point of the area

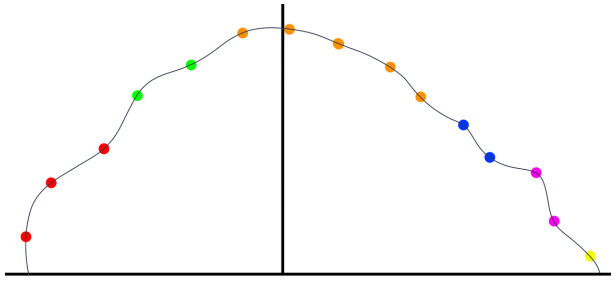

Figure 3: A bumpy curve, depicted by a grey line.
Points of the same color belong to the same sub-curve, each color represents a separate sub-curve

where the acceleration is positive. This method essentially divides sub-curves based on positions where the curves seems to make a turn. The middle of such a turn is where the sub-curves would be divided. An example can be seen in figure ....

Another issue the algorithm has to account for in this step is filling the entire domain. When multiple sub-curves are turned into solids of revolution, the combined solid does not necessarily cover the entire domain of the BRDF. This lack of data can result unexpected behaviour when the BRDF is used in rendering purposes that require this data. To solve this issue the algorithm makes a simple decision: any direction in the domain that is not filled is filled with a vector with intensity 0, which means no light is reflected in that direction. The algorithm specifically assigns an intensity of 0 as it assumes that any direction not covered by the approximated BRDF would not be covered by the full BRDF that the slice originally represents.

2.4 Non-vertical axes of rotation

The previous steps combined allow for the creation of full BRDFs from a large number of in-plane BRDFs. There are still sub-curves that do not result in representative solids of revolution when using a vertical axis of rotation, however. In order to create more correct BRDFs from these curves the next step is to find a method to determine non-vertical axes of rotation for sub-curves, and rotating sub-curves around these axes.

To find the non-vertical axis of rotation the algorithm uses Principal component Analysis (PCA) [5]. When PCA is used on a sub-curve it will result in a set of principal components, which are unit vectors in the direction of a line that best fit the data, with each principal component orthogonal to all other principal components. The first principal component, which represents the line trough the curve that best fits the data, can then be used as the axis or rotation for the sub-curve, as no other line will fit better to the data.

To rotate a sub-curve around a non-vertical axis the algorithm uses Rodrigues’ rotation formula, which rotates a vector v around an axis, represented by a unit vector k, with by an angle of θ around the axis in a counter-clockwise direction [12][1].

[1] Original text in French, translation: [4]

To use this formula with axes that do not pass trough 0, the algorithm shifts the data and the rotation axis along the x-axis until the rotational axis passes trough 0, then applies Rodrigues' formula, and finally shifts the resulting data back along the x-axis. This is needed as the data would otherwise not be in the right position relative to k, and thus result in an incorrect solid of revolution.

3 Results

Presented here are a number of BRDFs created using the algorithm. The first BRDF is a simple diffuse BRDF made from a half-circle, seen in figure 4 The second BRDF is a simple specular lobe for a BRDF, seen in figure 5 The third BRDF is a combination of a specular and a diffuse element into a single BRDF, seen in figure 6. The fourth BRDF is a combination of specular lobes to simulate front- and back-scattering, seen in figure 7. The last BRDF is a full BRDF with a front- and back-scatter lobe, seen in figure 8.

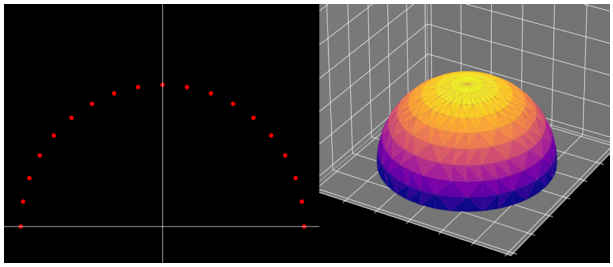

Figure 4: A simple BRDF created from a simple curve
left: the input data; right: the resulting BRDF

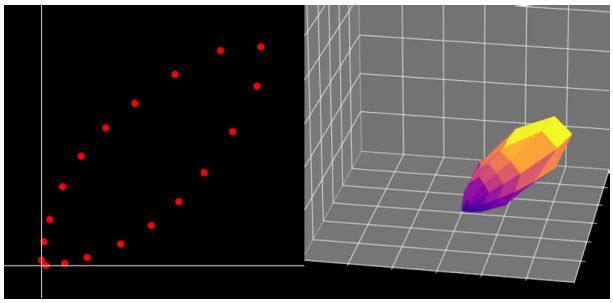

Figure 5: A simple BRDF created from a simple curve
left: the input data; right: the resulting BRDF

4 Conclusion

This research describes an algorithm that can create a full, three-dimensional BRDF from an in-plane BRDF. The algorithm takes a list of two-dimensional points to represent the in-plane BRDF. It then divides these points into sub-curves, representing rotatable segments of the in-plane BRDF. From these sub-curves it creates solids of revolution, which it combines into a single solid. This data is then treated so it represents the full, three-dimensional BRDF while ensuring coverage of the domain and removing overlapping data.

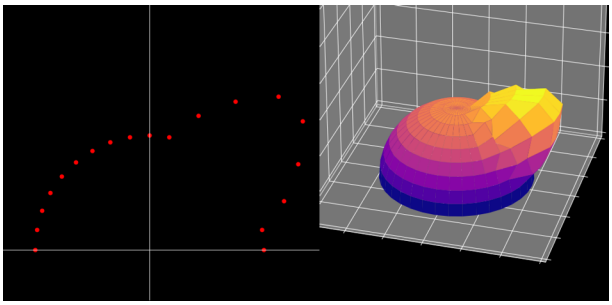

Figure 6: A simple BRDF created from a simple curve
left: the input data; right: the resulting BRDF

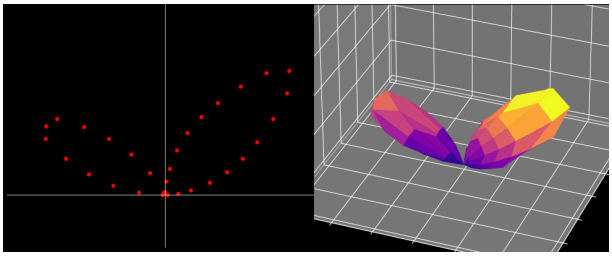

Figure 7: A complex BRDF created from a simple curve
left: the input data; right: the resulting BRDF

4.1 limitations

Due to the nature of this research, as well as the time restrictions placed upon it, there are limitations to this research. The first limitation is that it only handles measured BRDFs, as trying to approximate analytical BRDFs from a slice is too complex for this project. Another limitation is that this research only covers Isotropic materials, as anisotropic materials do not have BRDFs that can be approximated using solids of revolution. A third limitation is that the algorithm does not approximate or interpolate data in directions for which there no data in the solids of revolution, instead defaulting the intensity to 0. Another limitation is the code implementing the algorithm[2] is incomplete, as both the Principal Component Analysis and the dividing of sub-curves are not implemented.

4.2 Possible improvements

Because of the limitations on this research, as well as unforeseen circumstances, there are still improvements that could be made to this research. One notable improvement that can be made is to how the algorithm splits an in-plane BRDF into sub-curves. The current method is heavily flawed and is a heavy limiting factor of the types of BRDFs that can be modeled by the algorithm. A second improvement that could be made is interpolating the data to get a more accurate plot after normalizing the angles of each points instead of slightly rotating the vector represented by a point. Another possible improvement is adding interpolation or approximation of data in directions that have no data, as discussed in the limitations of this research.

[2] A partial implementation, not including PCA or division into sub-curves can be found at: https://github.com/Dead-gino/BRDF-rotation

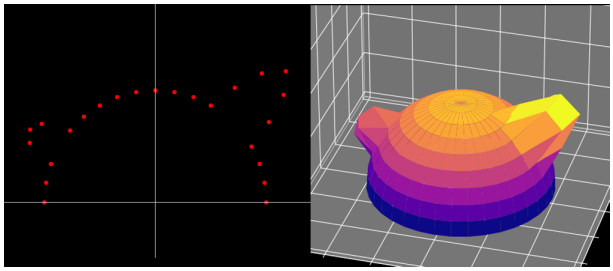

Figure 8: A complex BRDF created from a simple curve
left: the input data; right: the resulting BRDF

4.3 Future work

The algorithm detailed in this research can handle a large amount of possible slices of BRDFs, but it is not exhaustive. Possible extensions of this research could include extending the algorithm to handle in-plane BRDFs that do not have a smooth curve as described in section 3.3. Other additions to the algorithm could be addressing the possible improvements described above.

Besides extending the algorithm, another possibility of expanding upon this research is creating an algorithm for approximating anisotropic BRDFs.

4.4 Improving on existing work

Due to the heavily generative nature of this research there was not much research which to directly build upon. Most research used was either regarding smaller sub-problems encountered throughout the research process, or used to defined background information in this research paper, most notably the introduction. Besides this there is not a lot of research to improve upon.

4.5 Responsible research

While ethics are an important part of research, this research has little room in terms of ethics or the violation thereof. The algorithm created does not handle data that pertains to any person or group, nor does it contain any bias towards certain outcomes, due to the mathematical nature of the research.

All the sources used during this research have been credited as fully and correctly as possible, with any information regarding a source being stored for later reference. Sources have been provided to the full extent of what could be gathered from the sources themselves. No sources used during this research have knowingly or willingly been omitted, altered, or miscredited.

The entirety of the algorithm developed in this research has been detailed throughout this paper. As such one should not have any issues reproducing the algorithm themselves, given enough time and resources. Points where other implementations of the algorithm might differ should have no impact of functionality, and instead only be in how information is stored, how intermediate results are represented, or how data is visualized.

References

[1] Nicolas Auger, Cyril Nicaud, and Carine Pivoteau. Merge Strategies: from Merge Sort to TimSort. working paper or preprint, December 2015.

[2] B. Duvenhage, K. Bouatouch, and D. G. Kourie. Numerical verification of bidirectional reflectance distribution functions for physical plausibility. In *Proceedings of the South African Institute for Computer Scientists and Information Technologists Conference*, SAICSIT '13, page 200–208, New York, NY, USA, 2013. Association for Computing Machinery.

[3] Python Software Foundation. Python 3.9.13. http://www.python.org, May 2022.

[4] Richard Friedberg. Rodrigues, olinde: "des lois géométriques qui régissent les déplacements d'un système solide...", translation and commentary, 2022.

[5] Karl Pearson F.R.S. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[6] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[7] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[8] JetBrains. Pycharm. https://www.jetbrains.com/pycharm/download/other.html, Mar 2023.

[9] Jason Davis Lawrence. *Acquisition and Representation of Material Appearance for Editing and Rendering*. PhD thesis, Princeton University, 2006.

[10] Stephen R. Marschner, Stephen H. Westin, Eric P. F. Lafortune, Kenneth E. Torrance, and Donald P. Greenberg. Image-Based BRDF Measurement Including Human Skin. In Dani Lischinski and Greg Ward Larson, editors, *Eurographics Workshop on Rendering*. The Eurographics Association, 1999.

[11] Rosa Ana Montes Soldado and Carlos Ureña Almagro. An overview of brdf models, 3 2012.

[12] Olinde Rodriques. Des lois géométriques qui régissent les déplacements d'un système solide dans l'espace, et de la variation des coordonnées provenant de ces déplacements considérés indépendamment des causes qui peuvent les produire. *Journal de Mathematiques Pures et Appliquees*, 5:380–440, 1840.

[13] Wikipedia. Solid of revolution — Wikipedia, the free encyclopedia. https://en.wikipedia.org/wiki/Solid_of_revolution, 2023. [Online; accessed 11-May-2023].

A Tools used

During the development of the algorithm described in this paper, a program was coded to test the functionality of the algorithm. The programming language used to implement the algorithm is Python 3.9.13 [3]. In addition to the base functionality present in Python the libraries Matplotlib 3.7.1 [7] and Numpy 1.24.3 [6] are used by the implementation. Matplotlib is used to visualize the data into three-dimensional plots, whereas Numpy is used for mathematical functions not present in the base Python functionality. To create the code for the implementation itself the IDE Pycharm 2022.3.3 [8] was used.