Commonality Analysis for Family of Lighting Models

Sasha Soraine

October 3, 2019

1 Revision History

Date	Version	Notes
October 1, 2019	1.0	Original draft

2 Reference Material

This section records information for easy reference.

2.1 Table of Units

Throughout this document SI (Système International d'Unités) is employed as the unit system. In addition to the basic units, several derived units are used as described below. For each unit, the symbol is given followed by a description of the unit and the SI name.

symbol	unit	SI
rad	angle	radian

2.2 Table of Symbols

The table that follows summarizes the symbols used in this document along with their units. The choice of symbols was made to be consistent with the physics and calculus notation. The symbols are listed in alphabetical order.

symbol	unit	description
$\angle\theta_i$	rad	Angle of incidence between incident ray and surface normal.
$\angle\theta_r$	rad	Angle of reflection between reflected ray and surface normal.
θ_1	rad	Angle of incidence between incident ray and surface normal in material 1.
θ_2	rad	Angle of refraction between refracted ray and surface normal in material 2.
n_1	—	Refractive index of first material.
n_2	—	Refractive index of second material.
n_i	—	Refractive index of i-th material.
$\hat{I}$	—	Vector form of incident ray.
$\hat{R}$	—	Vector form of reflected ray.
$\hat{N}$	—	Vector form of surface normal.

2.3 Abbreviations and Acronyms

symbol	description
A	Assumption
AS	Scope Time Assumption
AB	Build Time Assumption
AR	Run Time Assumption
DD	Data Definition
GD	General Definition
GS	Goal Statement
IM	Instance Model
LC	Likely Change
PS	Physical System Description
R	Requirement
SRS	Software Requirements Specification
CA	Commonality Analysis
LM	Lighting Model
Family of Lighting Models	[put your famram name here —TPLT]
T	Theoretical Model

Contents

3 Introduction

An important part of computer graphics is modeling the lighting of objects in a virtual environment. Understanding and modeling how light interacts with objects of different materials is necessary to provide realistic lighting to virtual environments. Having an open source lighting model library that is reliable and robust would allow for new computer graphics programmers to more efficiently create scenes.

The following section provides an overview of the Commonality Analysis for a family of lighting models. This section explains the purpose of the document, scope of the family, characteristics of the intended reader, and organization of the document.

3.1 Purpose of Document

This document describes a family of lighting models to be used for computer graphics. This document is intended to be used as a reference to be provide the necessary information to verify the family of models, and implement the different family members.

This document captures the problem domain, theoretical models used to address the problem, the commonalities and variabilities between members of the family, and the requirements common to those members. It serves as a starting point to the design and implementation of a library of lighting models, and will be referenced in the creation of a verification and validation plan.

3.2 Scope of the Family

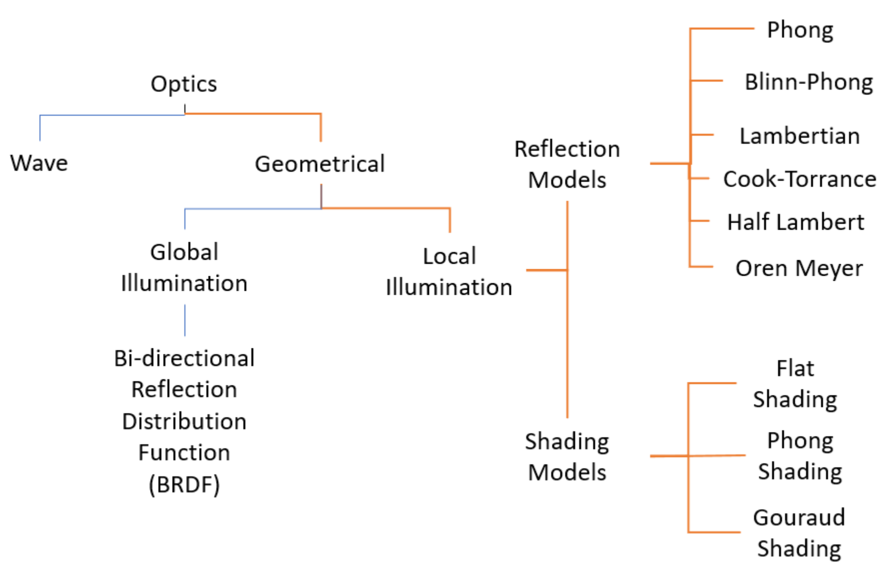


Figure 1: Problem Domain for Lighting Models of Computer Graphics.

The problem domain for lighting models in computer graphics is broad (Fig. 1). To simplify this problem we invoke assumptions AS1, AS2, AS3. This allows us to focus on the parts of the problem connected by the orange line.

The scope of the family includes geometrical optics simulation of light reflection for 3D material objects in a local illumination context.

3.3 Characteristics of Intended Reader

The intended readers of this document should have understanding of Grade 12 Physics (particularly Optics) and a undergraduate Level 2 understanding of Linear Algebra and Matrix operations.

3.4 Organization of Document

This document is organized in accordance with the CA template for scientific computing software provided by Dr. Smith for CAS 741. These templates are based on work by Smith (2006).

4 General System Description

This section identifies the interfaces between the system and its environment, describes the potential user characteristics and lists the potential system constraints.

4.1 Potential System Contexts

Figure 2 shows the high level system context. The circle represents the system user. The box is the library of lighting models system. Arrows are used to show the flow of data between the system and its environment.

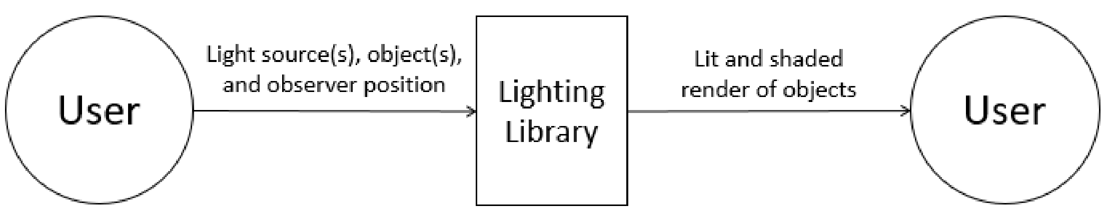


Figure 2: High level system context for Family of Lighting Models

- User Responsibilities:
  - Ensure that problem they are looking to solve matches the assumptions made for this family.

  - Provide information on the light source(s), object(s) in scene, and observer position.
  - Declare shading model to use from preset options.
  - Declare lighting model to use from present options.
  - Ensure application programming interface use complies with the user guide.

- Family of Lighting Models Responsibilities:

  - Calculate the reflections of all light rays coming from the light source(s).
  - Determine which light rays (reflected or from source) reach the observer.
  - Render a lit environment based on selected shading and lighting model.
  - Update the calculations and render in response to changes in the input data.
  - Detect data type mismatch, such as a string of characters instead of a floating point number.

4.2 Potential User Characteristics

The end user of Family of Lighting Models should have an Computer Science/Software Engineering Undergraduate Level 3 understanding of Computer Graphics (such as through completing the SFWR ENG/CS 3GC3) and moderate experience with programming.

4.3 Potential System Constraints

There are no system constraints.

5 Commonalities

This section presents a high-level view of the problem. It captures terminology and definitions relevant to the problem, theoretical models that are common to all members of the family, goal statements of the family, data definitions that will be used to solve these problems, as well as commonalities in inputs, calculations, and outputs.

This section also provides a background overview which explains the motivation for this work.

5.1 Background Overview

5.2 Terminology and Definitions

This subsection provides a list of terms that are used in the subsequent sections and their meaning, with the purpose of reducing ambiguity and making it easier to correctly understand the requirements:

- Geometrical Optics: The study of lights as rays.
- Reflection:
- Specular reflection:
- Diffuse reflection:
- Refraction:
- Lighting model:
- Global illumination:
- Emissive illumination:
- Ambient light: Light with no identifiable source or direction.
- Directional light: Light defined only by direction; light travels infinitely in that single direction.
- Point light: Light defined only by a point; light travels uniformly in every direction from that point, with intensity decreasing with square of distance from source.
- Spotlight: Light defined by a point and direction; light travels infinitely in a single direction from the defined source point.
- Shading model:
- OpenGL:

5.3 Data Definitions

This section collects and defines all the data needed to build the instance models. The dimension of each quantity is also given. [Modify the examples below for your problem, and add additional definitions as appropriate. —TPLT]

Number	DD1
Label	**Heat flux out of coil**
Symbol	q_C
SI Units	$\mathrm{W\,m^{-2}}$
Equation	$q_C(t) = h_C(T_C - T_W(t))$, over area A_C
Description	T_C is the temperature of the coil (°C). T_W is the temperature of the water (°C). The heat flux out of the coil, q_C ($\mathrm{W\,m^{-2}}$), is found by assuming that Newton's Law of Cooling applies (A**??**). This law (GD**??**) is used on the surface of the coil, which has area A_C ($\mathrm{m^2}$) and heat transfer coefficient h_C ($\mathrm{W\,m^{-2}\,°C^{-1}}$). This equation assumes that the temperature of the coil is constant over time (A**??**) and that it does not vary along the length of the coil (A**??**).
Sources	Citation here
Ref. By	IM4

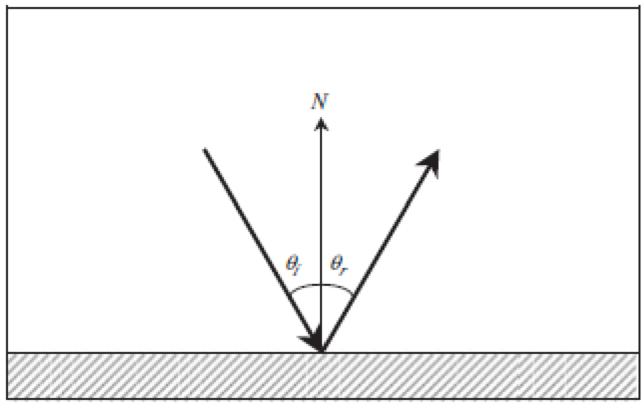
5.4 Goal Statements

Given some light source(s), some object(s) and their respective material properties, and an observer the goal statements are:

GS1: Render a fully lit and shaded scene of the objects based on the observer location.

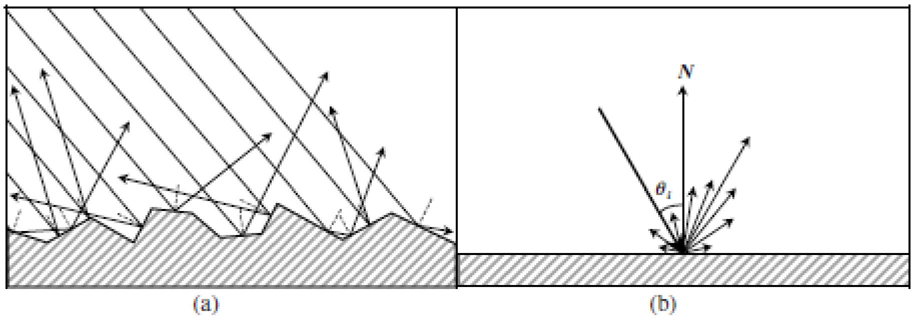
5.5 Theoretical Models

This section focuses on the general equations and laws that Family of Lighting Models is based on.

Number	T1
Label	**Law of Reflection**
Equation	$\angle\theta_i = \angle\theta_r$ or $\hat{I} \bullet \hat{N} = \hat{R} \bullet \hat{N}$
Description	When a ray of light reflects off a surface, the angle of incident is equal to the angle of reflection. This can also be written as the dot product of the incident ray and the normal is equal to the dot product of the reflected ray  and the normal.
Source	
Ref. By	GD??

5.5.1 General Definitions

This section collects the laws and equations that will be used in building the instance models.

Number	GD1
Label	**Diffuse Reflection**
SI Units	-
Equation	$\angle\theta_i = \angle\theta_r$
Description	Law of Reflection applied to rough surfaces.  (a) The surface micro-facets. (b) Diffuse reflection.
Source	Citation here
Ref. By	DD1, DD??

6 Variabilities

The follow section outlines variabilities between family members. Section 6.1 covers the assumptions made to simplify/realise the problem. Section 6.2 captures variabilities in the implementation.

Before tackling those sections, we summarize here variabilities in the problem.

6.1 Assumptions

This section outlines the various assumptions made in defining this problem. These assumptions are divided based on their binding time.

- **Scope Time Bindings**:

  AS1: Light will be defined by Geometrical Optics principles (ray-based).

  AS2: Lighting will be handled at the local illumination level.

  AS3: Objects will not reflect light at each other.

- **Build Time Bindings**:

AB1: Positions of light source(s), object(s), and the observer will be defined on a 3D Cartesian Coordinate System.

AB2: Objects will be represented by a geometric mesh of triangles.

AB3: All objects are opaque; light will not experience refraction with any object.

- **Run-Time Bindings**:

AR1: Position of light source will be defined by a vector $[x, y, z]$ representing the Cartesian Coordinates of its centre.

AR2: Type of light source will be selected from the following list:

1. Point light,
2. Spotlight,
3. Directional light, or
4. Ambient light.

AR3: Colour of light from light source will be defined by a 3-tuple (r, g, b).

AR4: Position of light source will be defined by a vector $[x, y, z]$ representing the Cartesian Coordinates of its centre.

AR5: Type of object will be selected from the following list:

1. Sphere,
2. Cube,
3. Torus, or
4. Teapot.

AR6: Object material colour is defined by a 3-tuple (r, g, b).

AR7: Type of shading model will be selected from the following list:

1. Flat,
2. Gouraud, or
3. Phong

Other assumptions about the system include:

A1: The virtual environment will be described by a 3D Caretsian Coordinate System using right-hand rules.

A2: Light source(s) information (position, type, colour) will be passed in via file.

A3: Object(s) information (position, mesh, colour, material properties) will be passed in via file.

6.2 Calculation

-

6.2.1 Instance Models

This section transforms the problem defined in Section 3.2 into one which is expressed in mathematical terms. It uses concrete symbols defined in Section 5.3 to replace the abstract symbols in the models identified in Sections 5.5 and 5.5.1.

Number	IM1
Label	**Lambertian (Diffuse) Reflection Model**
Input	
Output	
Description	T_W is the water temperature (°C). T_P is the PCM temperature (°C). T_C is the coil temperature (°C). $\tau_W = \frac{m_W C_W}{h_C A_C}$ is a constant (s). $\eta = \frac{h_P A_P}{h_C A_C}$ is a constant (dimensionless). The above equation applies as long as the water is in liquid form, $0 < T_W < 100^o$C, where 0^oC and 100^oC are the melting and boiling points of water, respectively (A**??**, A**??**).
Sources	Citation here
Ref. By	IM**??**

Number	IM2
Label	**Half Lambert (Diffuse) Reflection Model**
Input	
Output	
Description	T_W is the water temperature (°C). T_P is the PCM temperature (°C). T_C is the coil temperature (°C). $\tau_W = \frac{m_W C_W}{h_C A_C}$ is a constant (s). $\eta = \frac{h_P A_P}{h_C A_C}$ is a constant (dimensionless). The above equation applies as long as the water is in liquid form, $0 < T_W < 100^o$C, where 0^oC and 100^oC are the melting and boiling points of water, respectively (A**??**, A**??**).
Sources	Citation here
Ref. By	IM**??**

Number	IM3
Label	**Phong Reflection Model**
Input	
Output	
Description	T_W is the water temperature (°C). T_P is the PCM temperature (°C). T_C is the coil temperature (°C). $\tau_W = \frac{m_W C_W}{h_C A_C}$ is a constant (s). $\eta = \frac{h_P A_P}{h_C A_C}$ is a constant (dimensionless). The above equation applies as long as the water is in liquid form, $0 < T_W < 100^o$C, where 0^oC and 100^oC are the melting and boiling points of water, respectively (A**??**, A**??**).
Sources	Citation here
Ref. By	IM**??**

Number	IM4
Label	**Blinn-Phong Reflection Model**
Input	
Output	
Description	T_W is the water temperature (°C). T_P is the PCM temperature (°C). T_C is the coil temperature (°C). $\tau_W = \frac{m_W C_W}{h_C A_C}$ is a constant (s). $\eta = \frac{h_P A_P}{h_C A_C}$ is a constant (dimensionless). The above equation applies as long as the water is in liquid form, $0 < T_W < 100^o$C, where 0^oC and 100^oC are the melting and boiling points of water, respectively (A**??**, A**??**).
Sources	Citation here
Ref. By	IM**??**

6.3 Output

7 Requirements

This section provides the functional requirements, the business tasks that the software is expected to complete, and the nonfunctional requirements, the qualities that the software is expected to exhibit.

7.1 Family of Functional Requirements

[Since the CA will often be applied to a library, the functionality will not be a single use case. Therefore, this section should summarize the family of potential requirements. A good way to provide an overview of the functional requirements would be to provide multiple use cases on how the library will be employed. —TPLT]

R1: [Requirements for the inputs that are supplied by the user. This information has to be explicit. —TPLT]

R2: [It isn't always required, but often echoing the inputs as part of the output is a good idea. —TPLT]

R3: [Calculation related requirements. —TPLT]

R4: [Verification related requirements. —TPLT]

R5: [Output related requirements. —TPLT]

7.2 Nonfunctional Requirements

[To allow the Non-Functional Requirements (NFRs) to vary between family members, try to parameterize them. The value of the parameter is than a variability. —TPLT]

[An important variability between family members it the relative importance of the NFRs. Smith (2006) shows how pairwise comparisons can be used to rank the importance of NFRs. —TPLT]

[List your nonfunctional requirements. You may consider using a fit criterion to make them verifiable. —TPLT]

8 Likely Changes

LC1: Refractive materials incorporated into family.

9 Traceability Matrices and Graphs

[You will have to add tables. —TPLT]

References

W. Spencer Smith. Systematic development of requirements documentation for general purpose scientific computing software. In *Proceedings of the 14th IEEE International Requirements Engineering Conference, RE 2006*, pages 209–218, Minneapolis / St. Paul, Minnesota, 2006. URL http://www.ifi.unizh.ch/req/events/RE06/.

W. Spencer Smith, John McCutchan, and Jacques Carette. Commonality analysis for a family of material models. Technical Report CAS-17-01-SS, McMaster University, Department of Computing and Software, 2017.

10 Appendix

[Your report may require an appendix. For instance, this is a good point to show the values of the symbolic parameters introduced in the report. —TPLT]

10.1 Symbolic Parameters

[The definition of the requirements will likely call for SYMBOLIC_CONSTANTS. Their values are defined in this section for easy maintenance. —TPLT]
[Advice on using the template:

- Assumptions have to be invoked somewhere
- "Referenced by" implies that there is an explicit reference
- Think of traceability matrix, list of assumption invocations and list of reference by fields as automatically generatable
- If you say the format of the output (plot, table etc), then your requirement could be more abstract
- For families the notion of binding time should be introduced
- Think of families as a library, not as a single program

—TPLT]