Layered-SQT

M. Grady Saunders
mgs8033@rit.edu

1. Introduction

This document describes `layered-sqt`, a command-line tool for simulating the Bidirectional Scattering Distribution Function (BSDF) which emerges from a layered assembly. In this context, a *layered assembly* is a theoretical construction consisting of N layers separated by $N + 1$ participating media for $N \geq 1$. A layer is an infinite plane which is offset along (and normal to) the z-axis, and which is associated with a constituent BSDF that describes how light is scattered upon intersection. A medium is thought to occupy the space between adjacent layers or, in boundary cases, the spaces above and below the top and bottom layers respectively. To ensure that the emergent BSDF is well-defined, `layered-sqt` requires that participating media be homogeneous, i.e., that scattering properties be independent of spatial location. Furthermore, for simplicity/tractability, `layered-sqt` does not account for wavelength-dependence.

1.1. Basics of LSQT format

The structure of a layered assembly is easy enough to convey in plain-text with rudimentary syntax, so this is the format `layered-sqt` accepts as input. By convention, we refer to this format as "LSQT format", and we suffix associated filenames with the extension `.lsqt` (though this suffix is not strictly required for the program to run).

An LSQT file is therefore a line-by-line plain-text description of a layered assembly from top to bottom. So, the first line describes the top medium, the second line describes the top layer beneath the top medium, the third line describes the medium beneath the top layer, and so on until the bottom medium. That being the case, odd-numbered lines describe media and even-numbered lines describe layers. For media as well as layers, the general syntax is

```
Name key1=val1 key2=val2
```

where `Name` is described by keyword arguments `key1` and `key2`—importantly, this syntax is whitespace-delimited, so keyword arguments of the form `key=val` must not contain whitespace. It is also worth mentioning that keyword arguments may appear in any order.

1.2. Basics of program usage

As `layered-sqt` is a command-line tool, it executes on the command-line, whereby it scans command-line arguments for (optional) flags and a (required) filename appearing somewhere as a positional argument, i.e., an argument not consumed by a flag. It then parses the input file, simulates the emergent BSDF, and writes the results to a RAW-format file ready for conversion to SQT-format via `raw2sqt`. For instance,

```
$ ./layered-sqt example.lsqt -p 50000 \
  -o example.raw
```

simulates a layered assembly as described in the file `example.lsqt` with 50,000 paths, and writes the emergent BSDF in plain-text RAW-format to `example.raw`. To convert the BSDF to SQT-format, which is necessary for use in DIRSIG, run

```
$ ./raw2sqt example.raw
```

which will write a new file `example.sqt`.

Two of the most common flags appear above, namely `-p` (or `--path-count`) to specify the number of paths used in the simulation and `-o` (or `--output`) to specify the output filename. To see a list of all acceptable flags with brief descriptions and default values, pass `-h` (or `--help`), or simply run `layered-sqt` with no filename. As an aside, `layered-sqt` verifies parameters specified in command-line flags as well as keyword arguments specified in the input LSQT file. In the event that something has an unreasonable value, the program issues an error and fails in a controlled manner.

Lastly, `layered-sqt` recognizes the single dash filename "`-`" as standard input. So, it is possible to pipe the (presumably LSQT format) output of a script into `layered-sqt` directly, if this happens to be convenient. As a trivial example,

```
$ cat example.lsqt | \
  ./layered-sqt -p 50000 -o example.raw -
```

is equivalent to just passing `example.lsqt`.

2. Participating media

A medium is characterized by its absolute index of refraction $\eta \geq 1$, its absorption coefficient $\mu_a \geq 0$, its scattering coefficient $\mu_s \geq 0$, and a Henyey-Greenstein phase function

$$p(\omega_o \to \omega_i) = \frac{1}{4\pi} \frac{1 - g^2}{(1 + g^2 + 2g\omega_o \cdot \omega_i)^{3/2}}$$

with shape parameter $g \in (-1, 1)$. It may be helpful to note that g is the mean cosine of the scattered direction with respect to the incident direction, such that p becomes forward scattering as $g \to 1$, backward scattering as $g \to -1$, and uniformly/isotropically scattering as $g \to 0$. Of course, the Henyey-Greenstein phase is neither the only nor the canonical phase function. However, it is intuitive and often good enough to reproduce phenenomenology. So, at least for the time being, it is hard-coded into `layered-sqt`.

2.1. LSQT format

To specify a medium, simply type the name `Medium` followed by (optional) keywords arguments `eta`, `mua`, `mus`, and `g` for η, μ_a, μ_s, and g respectively. Keyword arguments for a medium have default values `eta=1`, `mua=0`, `mus=0`, `g=0`, such that any may be omitted for brevity. So, for example,

```
Medium eta=1.2 mus=0.7 g=-0.2
```

specifies a medium with refractive index 1.2 which is non-absorbing and somewhat back-scattering. As the defaults may suggest, `layered-sqt` considers that vacuuum is just a medium with refractive index 1 which neither absorbs nor scatters. So,

```
Medium
```

specifies vacuum. Note—the implementation requires that top and bottom media be non-absorbing and non-scattering, such that $\mu_a = \mu_s = 0$, and issues an error if this requirement is not met.

3. Layers

A layer is characterized by its z-height and a local BSDF model. As different BSDF models require different parameters, there is a different "type" of layer for each BSDF model implemented in `layered-sqt`.

3.1. LSQT format (common to all layers)

Every layer accepts z-height as a parameter, so every line describing a layer must begin with the name `Layer` followed by the (required) keyword argument `z`. This is followed in turn by the name of the desired BSDF, followed by any keyword arguments for the BSDF. For example,

```
Layer z=1 NullBsdf
```

specifies a layer at z-height 1 with a null BSDF. As explained in the next sub-section, a null BSDF is a special case which accepts no additional keyword arguments.

3.2. Null BSDF

It is occasionally desirable to separate media *without* scattering at a layer. This is possible in `layered-sqt` by assigning a null BSDF (with name `NullBsdf`), which may be interpreted as a 100% transmissive directional delta function aligned to the direction a ray is already traveling. This is useful, for example, to model a layer of dust on top of a surface. To do so, we might specify the following LSQT file

```
Medium
Layer z=1 NullBsdf
Medium g=0.2 mus=1
Layer z=0 LambertianBsdf fR=1 fT=0
Medium
```

which describes a layer of forward-scattering "dust" on top of a 100% reflective Lambertian surface.

3.3. Lambertian BSDF

The Lambertian BSDF models a uniformly scattering surface which may reflect, transmit, or both. That is, the (cosine-weighted) scattering function is

$$f(\omega_o \to \omega_i) = \frac{|\cos\theta_i|}{\pi} \begin{cases} f_R & (\omega_o, \omega_i) \text{ in same hemisphere} \\ f_T & \text{otherwise} \end{cases}$$

where f_R and f_T specify the amount of incident energy reflected and transmitted respectively. Both f_R and f_T should be non-negative numbers such that $f_R + f_T \leq 1$ for this to be physically plausible. In the event that $f_R + f_T = 1$, this is perfectly energy-conserving.

3.3.1. LSQT format

To specify the Lambertian BSDF in LSQT format, use the name `LambertianBsdf` followed by keyword arguments `fR` and `fT` for f_R and f_T respectively. The implementation requires that the values of `fR` and `fT` be physically plausible, i.e., non-negative numbers whose sum does not exceed 1. For example,

```
Layer z=2.2 LambertianBsdf fR=0.8 fT=0.1
```

specifies a Lambertian surface at z-height 2.2 which is 80% reflective, 10% transmissive, and 10% absorptive.

3.4. Microsurface Lambertian BRDF

A microsurface is thought to be an infinitesimally thin cloud of microfacets, where each facet is thought to scatter light according to another, simpler BSDF. The cloud is characterized geometrically by 1) a distribution of the slopes of the facets and 2) a distribution of the heights of the facets, which are assumed to uncorrelated, such that the facets are discontinuous. The distribution of slopes is parameterized by its so-called roughness α (more generally, anisotropic roughness α_x, α_y). As $\alpha \to \infty$, the distribution of slopes widens and the emergent BSDF appears rougher. As $\alpha \to 0$, the distribution of slopes collapses, recovering the initial, simpler BSDF in the limiting case $\alpha = 0$.

As given in [3], the microsurface Lambertian BRDF is a multiple scattering (stochastic) microfacet model wherein facets are assumed to be Lambertian reflectors. This is particularly useful for representing rough diffuse surfaces. The implementation in `layered-sqt` is parameterized by the microsurface roughness $\alpha > 0$, the underlying Lambertian BRDF coefficient $f_R \in [0, 1]$, and the number of local stochastic process iterations $n_{\text{iter}} \geq 1$. For rougher microsurfaces (say, $\alpha > 0.4$), it may be more efficient to increase n_{iter} rather than the global path count.

3.4.1. LSQT format

To specify the microsurface Lambertian BRDF in LSQT format, use the name

```
MicrosurfaceLambertianBsdf
```

followed by (optional) keyword arguments `alpha`, `fR`, and `iter_count` for α, f_R, and n_{iter} respectively. By default, `alpha=0.5`, `fR=1`, and `iter_count` is chosen from `alpha`. Furthermore, multiple scattering may be disabled by the keyword argument

```
use_multiple_scattering=false
```

in which case only the single scattering term is computed. Note that, if multiple-scattering interactions are ignored, then setting `fR=1` does not guarantee that the BRDF is perfectly energy conserving. For small roughness values (say, $\alpha < 0.05$), the energy carried by multiple scattering interactions is insignificant, and so this may not be a concern. For large roughness values however, multiple scattering is important to prevent energy loss.

3.5. Microsurface dielectric BSDF

References

[1] Adam A. Goodenough and Scott D. Brown. "DIRSIG5: Next-Generation Remote Sensing Data and Image Simulation Framework". In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing* 10.11 (Nov. 2017), pp. 4818–4833.

[2] E. Heitz. "Understanding the masking-shadowing function in microfacet-based BRDFs". In: *Journal of Computer Graphics Techniques (JCGT)* 3.2 (June 2014), pp. 48–107. ISSN: 2331-7418.

[3] E. Heitz et al. "Multiple-scattering microfacet BSDFs with the Smith model". In: *ACM Transactions on Graphics* 35.4 (July 2016), pp. 1–58. ISSN: 0730-0301.