Calculating and Applying BSDFs in Radiance

Greg Ward
Anyhere Software
Goals

- Support WINDOW 6 BSDF output (XML)
- Create WINDOW 6 BSDFs from geometric models using freeware (i.e., Radiance)
- Render accurate images with direct shading using mkillum
- Perform efficient annual simulations
- Support higher-resolution BSDF data
Status

- WINDOW 3 BSDF format still subject to revision
- Support for reading and creating BSDFs is currently limited to transmittance data
- mkillum has been tested and refined to produce reasonable accuracy for diffusing systems
- New genBSDF tool for creating BSDF files from geometric (Radiance or MGF) descriptions
- Annual simulation for the daring with rtcontrib
What Is a BSDF?

“BSDF” stands for “Bidirectional Scattering Distribution Function” (BRDF + BTDF)

Describes how light scatters off a surface

Incl. wavelength, a 5-dimensional scalar function

unitless ratio of outgoing radiance over incoming irradiance (1/steradian)
WINDOW 6 BSDF

- Breaks BSDF into integrated spectra:
  - Visible (380-780 nm), Solar (300-2500 nm),
    NIR (780-2500 nm), FIR (5000-40000 nm)

- Breaks scattering into four components:
  - Front Transmission, Back Transmission,
    Front Reflection, Back Reflection
Klems/Window 6 system
BSDF Coordinates (2)
<?xml version="1.0" encoding="UTF-8"?>
  <Material>
    <Name> Name </Name>
    <Manufacturer> Manufacturer </Manufacturer>
    <Thickness unit="Meter"> 0.01 </Thickness>
    <DeviceType> DeviceType </DeviceType>
    <ThermalConductivity> 1 </ThermalConductivity>
    <EmissivityFront> 0.9 </EmissivityFront>
    <EmissivityBack> 0.9 </EmissivityBack>
    <TIR> 0 </TIR>
    <Comments> Comments </Comments>
  </Material>
  <DataDefinition>
    ...
  </DataDefinition>
  <WavelengthData>
    <Wavelength unit="Integral"> NIR </Wavelength>
    <SourceSpectrum> CIE Illuminant D65 1nm.ssp </SourceSpectrum>
    <DetectorSpectrum> ASTM E308 1931 Y.dsp </DetectorSpectrum>
    <WavelengthDataBlock>
      <WavelengthDataDirection> Transmission Front </WavelengthDataDirection>
      <ColumnAngleBasis> LBNL/Klems Full </ColumnAngleBasis>
      <RowAngleBasis> LBNL/Klems Full </RowAngleBasis>
      <ScatteringDataType> BTDF </ScatteringDataType>
      <ScatteringData>
        2.443881, 0.047337, 0.041435, 0.038990, 0.041435,
        0.047337, 0.048413, 0.046964, 0.048413, 0.047337,
        0.040883, 0.035154, 0.031478, 0.030108, 0.031363,
        0.035154, 0.040605, 0.047337, 0.048086, 0.044691,
        0.042586, 0.042007, 0.042537, 0.044691, 0.047921,
        0.047337, 0.038892, 0.031273, 0.025227, 0.021345,
        0.020007, 0.021345, 0.025227, 0.031273, 0.038892,
        ...
    </ScatteringData>
  </WavelengthDataBlock>
</Layer>
Getting BSDF Data

- WINDOW 6 will have access to a database of fenestration systems
- Systems may be combined in “layers” up to a point
- Some BSDFs are measured, most are simulated
  - BSDF measurements go into simulations, too
- New genBSDF program in Radiance 4.1
Sample BTDF Data (1)

Light Shelf
Sample BTDF Data (2)

Light Shelf

Visualization by Andrew McNeil
BSDFs in mkillum

- Since 3.9 (2008), mkillum has supported BTDFs
- fully debugged by 4.0 (Spring 2010)

- Works best when:
  - BTDF is somewhat diffuse, or
  - model includes geometry for direct component
Example Results

No fenestration geometry
Averages Agree

No fenestration geometry
Specular Challenge

- 145 hemisphere directions cover 10-30° regions
- The sun covers about 0.5°
- Breaking hemisphere further leads to drastic increases in computational load
- 80000 divisions needed for sun-sized patches
- that’s an 80000x80000 BTDF (6.4e9 values)
- ...and still no direct solar pattern on interior
BSDF with Geometry

- Incorporate model of fenestration device in XML
- Use MGF (Materials and Geometry Format) developed for IESNA luminaire descriptions
  - simple, public <http://radsite.lbl.gov/mgf>
  - embeddable in XML
  - standard C parser makes support easy
Sample MGF

<Geometry format="MGF" unit="Meter">
# Y-axis points "up", Z-axis into room, right-handed coordinates
m WhitePlastic =
  rd .7
  rs .02 0
  sides 2
o VenetianBlinds
xf -rx -60 -a 67 -t 0 .03 0
  o Slat
  v v1 =
    p -2 0 0
  v v2 =
    p 2 0 0
  v v3 =
    p 2 0 .04
  v v4 =
    p -2 0 .04
f v1 v2 v3 v4
o
xf
o
</Geometry>
New genBSDF script

- Perl script calls rtcontrib to compute BTDF
  - forward transmittance only at this time
- Simple operation:
  - genBSDF fenes1.rad > fenes1.xml
- Default is to embed MGF geometry
  - options for number of samples, processes, etc.
BSDFs in mkillum

Use illum  BSDF  Thickness (flag to use MGF)

#@mkillum l- c=a d=fenes.xml t=0 u=+Z

void polygon window
0
0
12

0 4 1
0 6.25 1
0 6.25 2.5
0 4 2.5

Up orientation
When To Use “l+” with BTDF Data

- When the window geometry is unknown
- When there is no direct beam component
- Redirected components usually use BTDF
Annual Simulation

- Using mkillum with BTDFs is fairly quick, but...
- Re-rendering a scene 2000+ times for each hour?
- We need something faster...
- Can we use daylight coefficients with BTDF data?
Three Phase Method

- **Phase I:**
  Use rtcontrib to get daylight coefficients relating sky patches to incident directions

- **Phase II:**
  Use rtcontrib to relate exiting portal directions to desired measurement locations (e.g., image)

- **Phase III** (time-step calculation):
  sky * incident * BTDF * exiting
Our Matrix Equation

\[ i = \text{VTDs} \]

where:

- \( i \) is the desired result vector (radiances, irradiances, etc.)
- \( V \) is the "View" matrix defining the lighting connection between results and exiting directions for a window group
- \( T \) is the "Transmission" matrix defining the BTDF of the window group
- \( D \) is the "Daylight" matrix defining the coefficients between incoming directions for the window group and sky patches
- \( s \) is a vector of sky patch luminances for a particular time and date

In a more explicit form, this would be:

\[
\begin{bmatrix}
\text{sens}_1 \\
\vdots \\
\text{sens}_M
\end{bmatrix}
= \begin{bmatrix}
\text{sens}_1\text{dir}_1 & \ldots & \text{sens}_1\text{dir}_N \\
\vdots & \ddots & \vdots \\
\text{sens}_M\text{dir}_1 & \ldots & \text{sens}_M\text{dir}_N
\end{bmatrix}
\begin{bmatrix}
\text{edir}_1\text{dir}_1 & \ldots & \text{edir}_1\text{dir}_N \\
\vdots & \ddots & \vdots \\
\text{edir}_N\text{dir}_1 & \ldots & \text{edir}_N\text{dir}_N
\end{bmatrix}
\begin{bmatrix}
\text{i dir}_1\text{dc}_1 & \ldots & \text{i dir}_1\text{dc}_K \\
\vdots & \ddots & \vdots \\
\text{i dir}_K\text{dc}_1 & \ldots & \text{i dir}_K\text{dc}_K
\end{bmatrix}
\begin{bmatrix}
\text{sky}_1 \\
\vdots \\
\text{sky}_K
\end{bmatrix}
\]
Phase I: Compute $D$

- Apply `rtcontrib` to relate sky patches to incident directions on window exterior.

- Need separate calculation for each orientation and major geometric feature.

- New `genklemsamp` utility generates samples over a given window group.

- Written in Perl.
Example Space
Phase II: Compute $V$

- Use `rtcontrib` to relate sensor locations to exiting directions on window interiors
- A single run can cover all window groups
- `klems_int.cal` file maps to BTDF coord.
Outgoing Directions for One Window Group
Phase III: Time Step

Use `genskyvec` to create sky patch vector $s$

Use `dctimestep` to multiply it all together

$$i = \text{VTDS}$$
Phase III Example

gensky 9 21 12:00 -a 37.71 -o 122.21 -m 120 | genskyvec > sky.dat
pcomb ’!dctimestep comp/back_SouthGroup%03d.hdr blinds1.xml SouthGroup.dmx sky.dat’ \
’!dctimestep comp/back_WestGroup%03d.hdr blinds2.xml WestGroup.dmx sky.dat’ \
’!dctimestep comp/back_NorthGroup%03d.hdr blinds2.xml NorthGroup.dmx sky.dat’ \
’!dctimestep comp/back_EastGroup_%03d.hdr blinds1.xml EastGroup.dmx sky.dat’ \
> back_9-21_1200.hdr
rm sky.dat
Phase III Example

Generate sky vector for noon at the Autumn equinox

gensky 9 21 12:00 -a 37.71 -o 122.21 -m 120 | genskyvec > sky.dat
pcomb '!dctimestep comp/back_SouthGroup%03d.hdr blinds1.xml SouthGroup.dmx sky.dat' \
   '!dctimestep comp/back_WestGroup%03d.hdr blinds2.xml WestGroup.dmx sky.dat' \
   '!dctimestep comp/back_NorthGroup%03d.hdr blinds2.xml NorthGroup.dmx sky.dat' \
   '!dctimestep comp/back_EastGroup_%03d.hdr blinds1.xml EastGroup.dmx sky.dat' \
> back_9-21_1200.hdr
rm sky.dat

Each call to dctimestep computes contributions of one window group

Time to run the above is less than 8 seconds on my laptop
Future Work

- Support for BRDF (reflection) data
- Further validation on appropriate use
- Variable-resolution & native BSDF types
- Documentation and tutorials
- What about a user interface?
  - always my last question, always unanswered