Materials in Radiance

trying an overview of the state in 2025

Peter Apian-Bennewitz

pab Consulting for Optics & Engineering
info@pab.eu

28th *Radiance* workshop, Lausanne

A Preface: Users & Applications

In general: tools and software: different users, different priorities:

- Academics:
  career, papers, curiosity

A Preface: Users & Applications

In general: tools and software: different users, different priorities:

- Academics:
  career, papers, curiosity
- Institutes:
  continuous struggle to fund the group's employees and fight for floor space

A Preface: Users & Applications

In general: tools and software: different users, different priorities:

- Academics:
  career, papers, curiosity
- Institutes:
  continuous struggle to fund the group's employees and fight for floor space
- Consultants:
  competitive selling point,
  foreseeable project costs,
  reliability of results & liability to client

A Preface: Users & Applications

In general: tools and software: different users, different priorities:

- Academics:
  career, papers, curiosity
- Institutes:
  continuous struggle to fund the group's employees and fight for floor space
- Consultants:
  competitive selling point,
  foreseeable project costs,
  reliability of results & liability to client

Inter-Understanding is a little low, fruitful collaboration often difficult.

Contents

brought to you exclusively in

in selected cinemas

Motivation for Material Models

From the very simple to the complex:

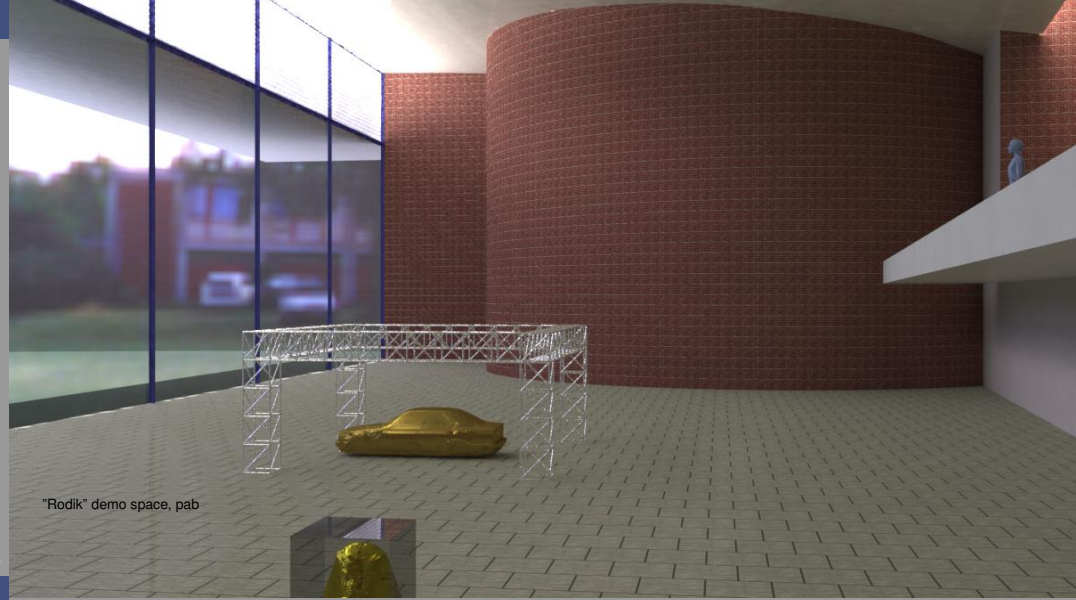
- simply differentiate the surfaces in a model

simulation, Rodic demo space, pab

Motivation for Material Models

From the very simple to the complex:

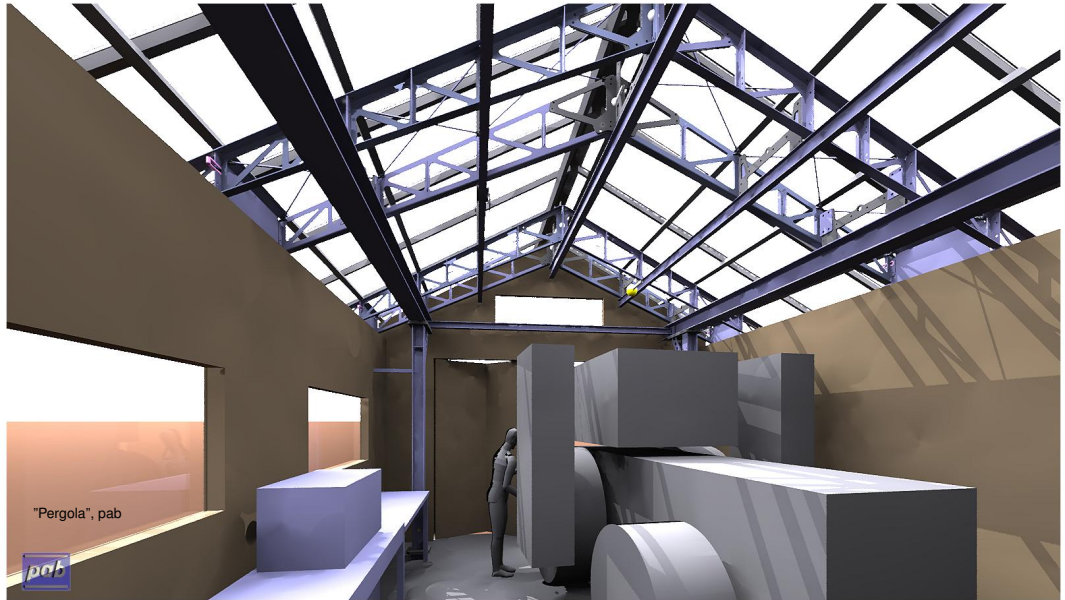
- simply differentiate the surfaces in a model



Motivation for Material Models

From the very simple to the complex:

- simply differentiate the surfaces in a model



Motivation for Material Models

From the very simple to the complex:

- simply differentiate the surfaces in a model
- reproduce the ”looks” of a real material

Southward Collection, Jan 2015, pab

Motivation for Material Models

From the very simple to the complex:

- simply differentiate the surfaces in a model
- reproduce the ”looks” of a real material

Motivation for Material Models

From the very simple to the complex:

- simply differentiate the surfaces in a model
- reproduce the ”looks” of a real material

Motivation for Material Models

From the very simple to the complex:

- simply differentiate the surfaces in a model
- reproduce the ”looks” of a real material

Motivation for Material Models

From the very simple to the complex:

- simply differentiate the surfaces in a model
- reproduce the ”looks” of a real material
- glare analysis

Motivation for Material Models

From the very simple to the complex:

- simply differentiate the surfaces in a model
- reproduce the ”looks” of a real material
- glare analysis

Motivation for Material Models

From the very simple to the complex:

- simply differentiate the surfaces in a model
- reproduce the ”looks” of a real material
- glare analysis
- indoor light levels: shading analysis with [opaque|glass] surfaces

MarinaBaySands, Singapore, May 2024, pab

Motivation for Material Models

From the very simple to the complex:

- simply differentiate the surfaces in a model
- reproduce the ”looks” of a real material
- glare analysis
- indoor light levels: shading analysis with [opaque|glass] surfaces

MarinaBaySands, Singapore, May 2024, pab

Motivation for Material Models

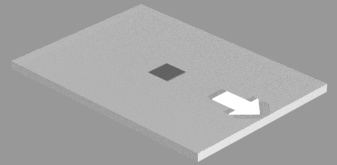
From the very simple to the complex:

- simply differentiate the surfaces in a model
- reproduce the ”looks” of a real material
- glare analysis
- indoor light levels: shading analysis with [opaque|glass] surfaces
- model light transport: light shelves, light pipes
  ⇝ related topics: ambient calculation , Photon Map extension

The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

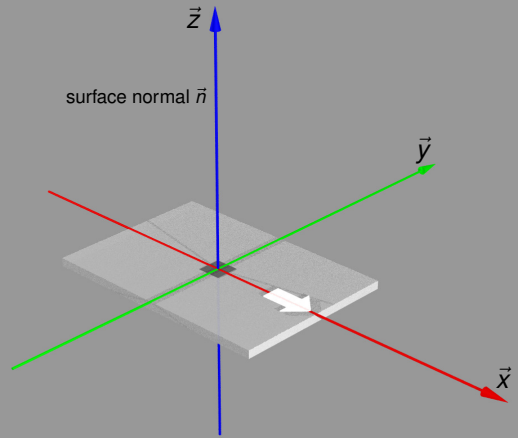
- at a surface element on a material:



The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

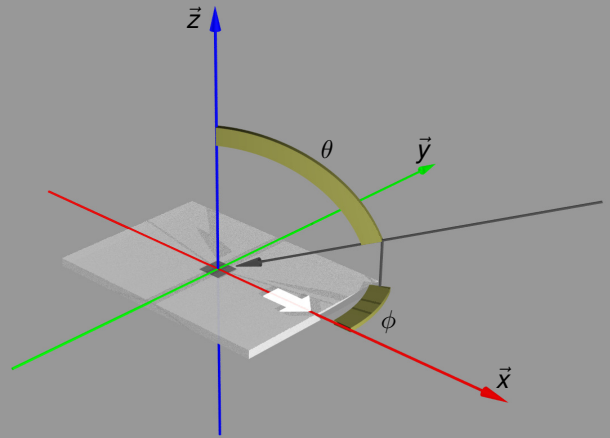
- at a surface element on a material:
- attach coordinate system



The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

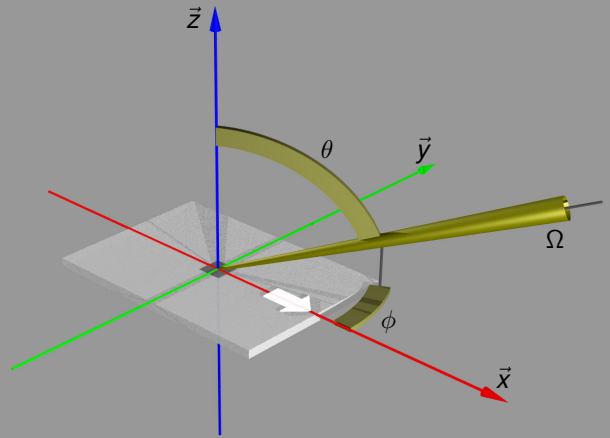
- at a surface element on a material:
- attach coordinate system
- let $\vec{x}$ denote a 3D direction, e.g. with spherical coordinates (θ, ϕ)



The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

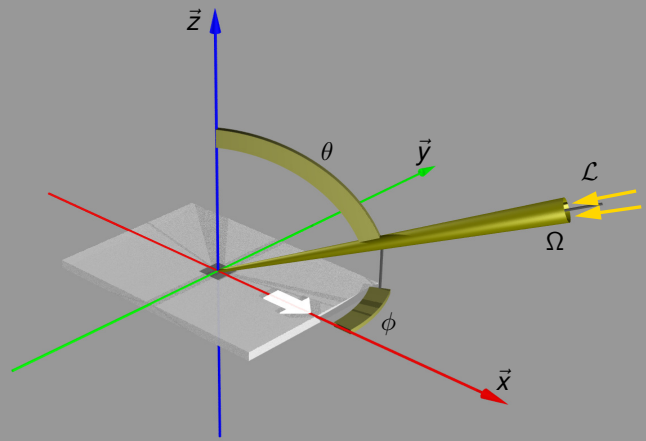
- at a surface element on a material:
- attach coordinate system
- let $\vec{x}$ denote a 3D direction, e.g. with spherical coordinates (θ, ϕ)
- let Ω denote a solid angle



The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

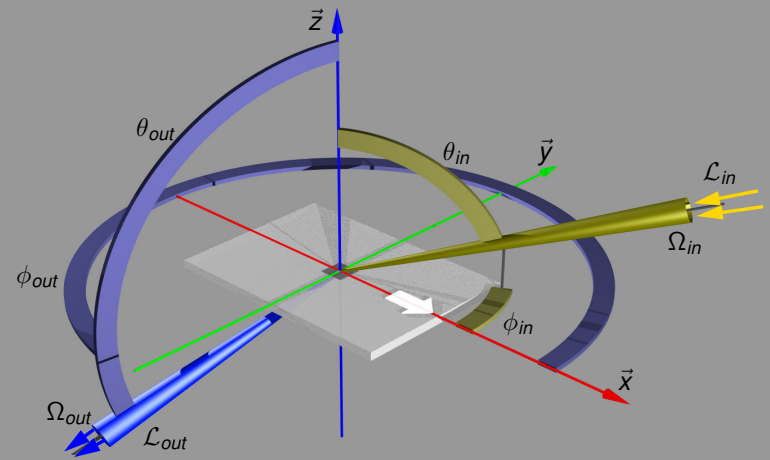
- at a surface element on a material:
- attach coordinate system
- let $\vec{x}$ denote a 3D direction, e.g. with spherical coordinates (θ, ϕ)
- let Ω denote a solid angle
- let $\mathcal{L}(\vec{x})$ denote *Radiance*, $[\frac{Watt}{sr\ m^2}]$



The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

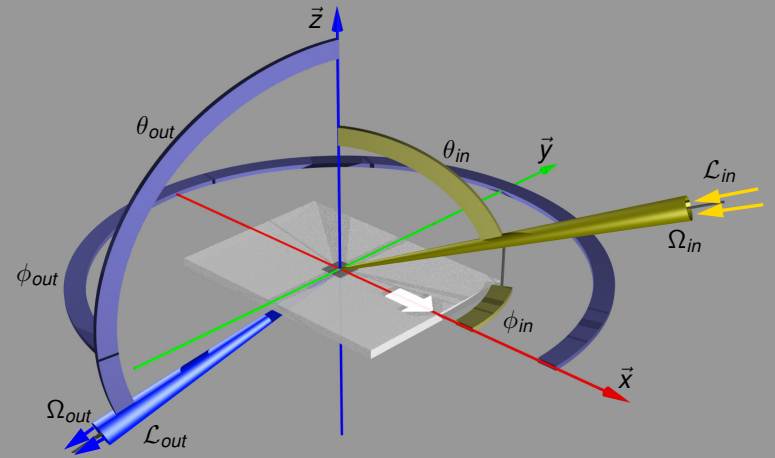
- at a surface element on a material:
- attach coordinate system
- let $\vec{x}$ denote a 3D direction, e.g. with spherical coordinates (θ, ϕ)
- let Ω denote a solid angle
- let $\mathcal{L}(\vec{x})$ denote *Radiance*, $[\frac{Watt}{sr\ m^2}]$
- $\mathcal{L}_{in}$ incident to a surface element, $\mathcal{L}_{out}$ outgoing from a surface element



The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

- at a surface element on a material:
- attach coordinate system
- let $\vec{x}$ denote a 3D direction, e.g. with spherical coordinates (θ, ϕ)
- let Ω denote a solid angle
- let $\mathcal{L}(\vec{x})$ denote *Radiance*, $[\frac{Watt}{sr\ m^2}]$
- $\mathcal{L}_{in}$ incident to a surface element, $\mathcal{L}_{out}$ outgoing from a surface element
- let $\int_{\vec{x}_{in}}^{\Omega_{in}=2\pi} ... f(\vec{x}) ... d\Omega$ describe an integral of a function f over the hemisphere

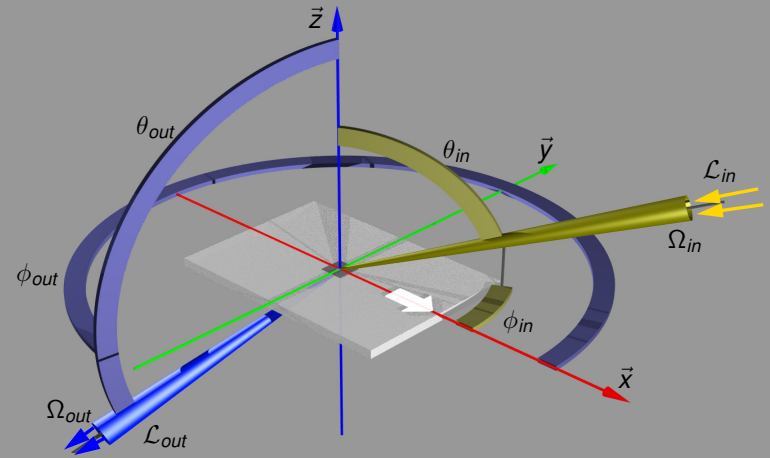


The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

- at a surface element on a material:
- attach coordinate system
- let $\vec{x}$ denote a 3D direction, e.g. with spherical coordinates (θ, ϕ)
- let Ω denote a solid angle
- let $\mathcal{L}(\vec{x})$ denote *Radiance*, $[\frac{Watt}{sr\ m^2}]$
- $\mathcal{L}_{in}$ incident to a surface element, $\mathcal{L}_{out}$ outgoing from a surface element
- let $\int_{\vec{x}_{in}}^{\Omega_{in}=2\pi} \ldots f(\vec{x}) \ldots d\Omega$ describe an integral of a function f over the hemisphere

then



The BSDF mathematics (BSDF , BRDF, BRTF, ...)

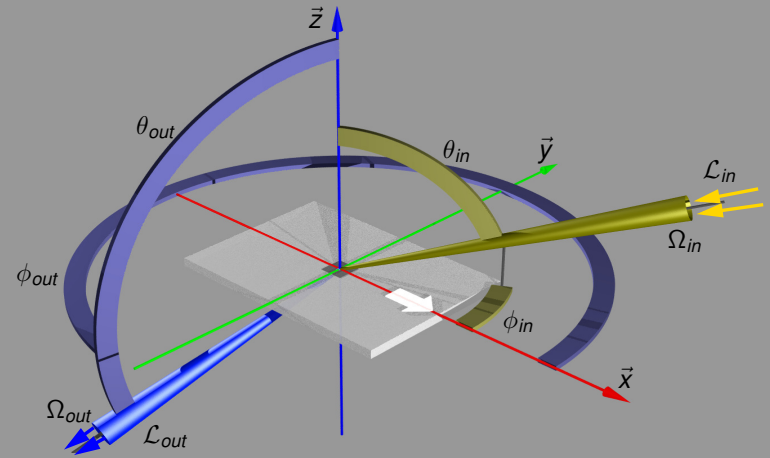
Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

- at a surface element on a material:
- attach coordinate system
- let $\vec{x}$ denote a 3D direction, e.g. with spherical coordinates (θ, ϕ)
- let Ω denote a solid angle
- let $\mathcal{L}(\vec{x})$ denote *Radiance*, $[\frac{Watt}{sr\ m^2}]$
- $\mathcal{L}_{in}$ incident to a surface element, $\mathcal{L}_{out}$ outgoing from a surface element
- let $\int_{\vec{x}_{in}}^{\Omega_{in}=2\pi} \ldots f(\vec{x}) \ldots d\Omega$ describe an integral of a function f over the hemisphere

then

- BSDF def:

$$\mathcal{L}_{out}(\vec{x}_{out}) = \int_{\vec{x}_{in}}^{\Omega_{in}=2\pi} BSDF(\vec{x}_{in}, \vec{x}_{out}) \ \cos(\theta_{out}) \ \mathcal{L}_{in}(\vec{x}_{in}) \ d\Omega_{in}$$



The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

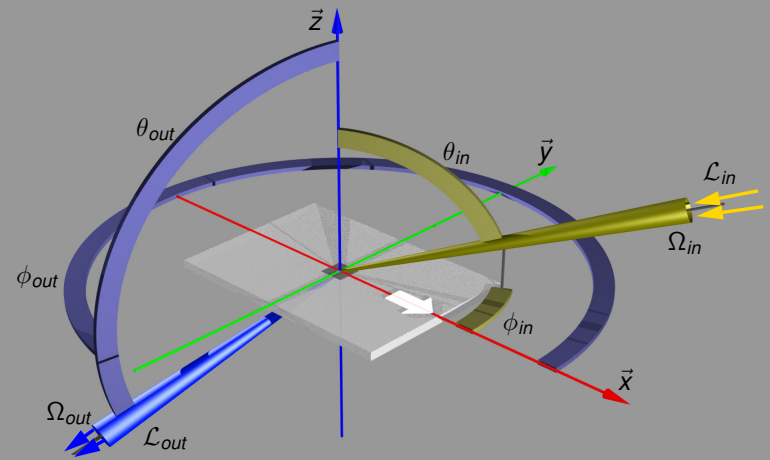
- at a surface element on a material:
- attach coordinate system
- let $\vec{x}$ denote a 3D direction, e.g. with spherical coordinates (θ, ϕ)
- let Ω denote a solid angle
- let $\mathcal{L}(\vec{x})$ denote *Radiance*, $[\frac{Watt}{sr\ m^2}]$
- $\mathcal{L}_{in}$ incident to a surface element, $\mathcal{L}_{out}$ outgoing from a surface element
- let $\int_{\vec{x}_{in}}^{\Omega_{in}=2\pi} \ldots f(\vec{x}) \ldots d\Omega$ describe an integral of a function f over the hemisphere

then

- BSDF def:

$$\mathcal{L}_{out}(\vec{x}_{out}) = \int_{\vec{x}_{in}}^{\Omega_{in}=2\pi} BSDF(\vec{x}_{in}, \vec{x}_{out}) \ \cos(\theta_{out}) \ \mathcal{L}_{in}(\vec{x}_{in}) \ d\Omega_{in}$$

- depends on 4 scalar variables: $BSDF(\vec{x}_{in}, \vec{x}_{out}) = BSDF(\theta_{in}, \phi_{in}, \theta_{out}, \phi_{out})$



The BSDF mathematics (BSDF , BRDF, BRTF, ...)

Bidirectional Scatter Distribution Function more maths: pab talk at RADIANCE Workshop 2010

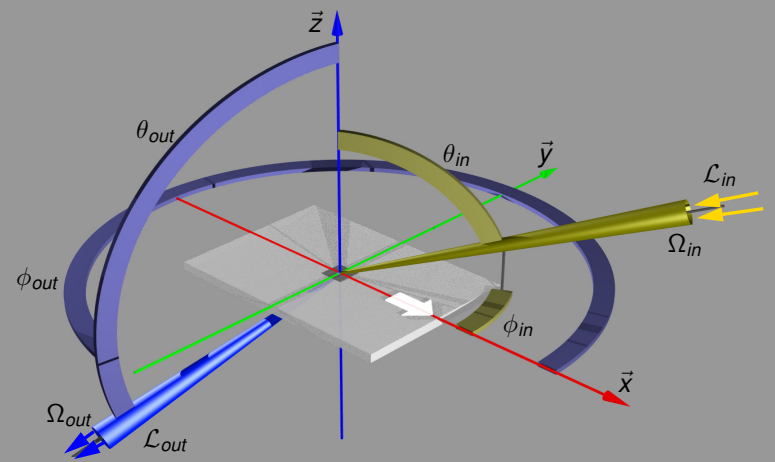
- at a surface element on a material:
- attach coordinate system
- let $\vec{x}$ denote a 3D direction, e.g. with spherical coordinates (θ, ϕ)
- let Ω denote a solid angle
- let $\mathcal{L}(\vec{x})$ denote *Radiance*, $[\frac{Watt}{sr\ m^2}]$
- $\mathcal{L}_{in}$ incident to a surface element, $\mathcal{L}_{out}$ outgoing from a surface element
- let $\int_{\vec{x}_{in}}^{\Omega_{in}=2\pi} \ldots f(\vec{x}) \ldots d\Omega$ describe an integral of a function f over the hemisphere

then

- BSDF def:

$$\mathcal{L}_{out}(\vec{x}_{out}) = \int_{\vec{x}_{in}}^{\Omega_{in}=2\pi} BSDF(\vec{x}_{in}, \vec{x}_{out}) \ \cos(\theta_{out}) \ \mathcal{L}_{in}(\vec{x}_{in}) \ d\Omega_{in}$$

- depends on 4 scalar variables: $BSDF(\vec{x}_{in}, \vec{x}_{out}) = BSDF(\theta_{in}, \phi_{in}, \theta_{out}, \phi_{out})$
- $BSDF \in [0 : \infty]$, yet integral enforced by $\rho_{dh} < 1$, "diffuse" material: $BSDF = const$



types of BSDF models in Computer Graphics

- classical BSDF model:
  built-in functions with parameters

types of BSDF models in Computer Graphics

- classical BSDF model:
  built-in functions with parameters

- user defined ”function files”,
  evaluated at runtime at each ray-material intersection,
  similar to newer concept of a ”shader language”

types of BSDF models in Computer Graphics

- classical BSDF model:
  built-in functions with parameters

- user defined ”function files”,
  evaluated at runtime at each ray-material intersection,
  similar to newer concept of a ”shader language”

- newer concepts: ”data driven”
  actually: ”automatic” models with a set set of built-in functions
  or no functions at all

types of BSDF models in Computer Graphics

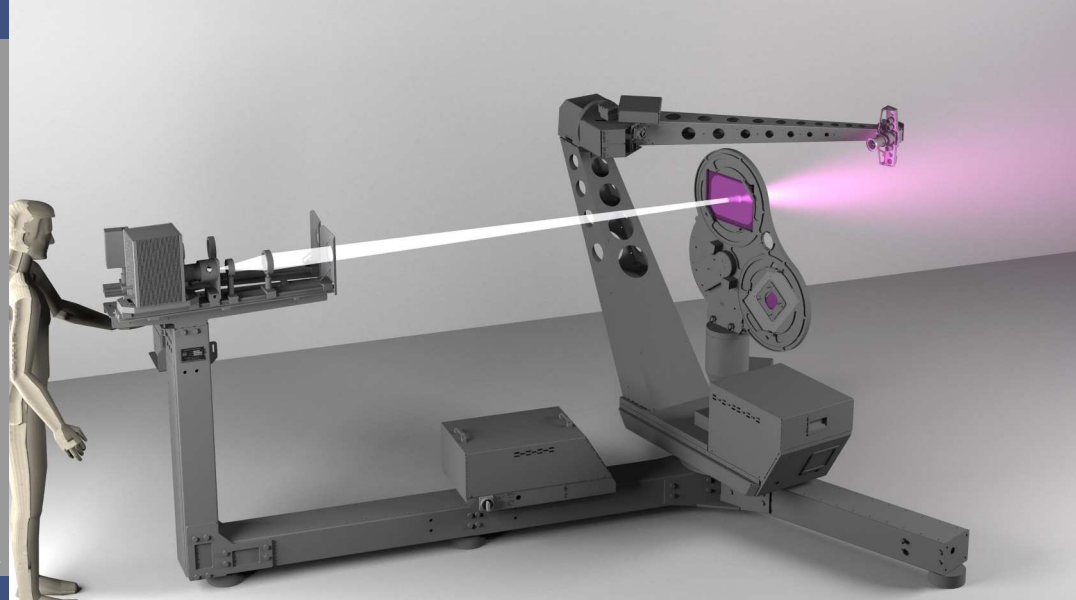
- classical BSDF model:
  built-in functions with parameters
- user defined ”function files”,
  evaluated at runtime at each ray-material intersection,
  similar to newer concept of a ”shader language”
- newer concepts: ”data driven”
  actually: ”automatic” models with a set set of built-in functions
  or no functions at all
- BSDF can (and should) be measured "in-vitro" in a lab

types of BSDF models in Computer Graphics

- classical BSDF model:
  built-in functions with parameters
- user defined ”function files”,
  evaluated at runtime at each ray-material intersection,
  similar to newer concept of a ”shader language”
- newer concepts: ”data driven”
  actually: ”automatic” models with a set set of built-in functions
  or no functions at all
- BSDF can (and should) be measured "in-vitro" in a lab
- BSDF can be visualised, analysed, compared and understood on its own

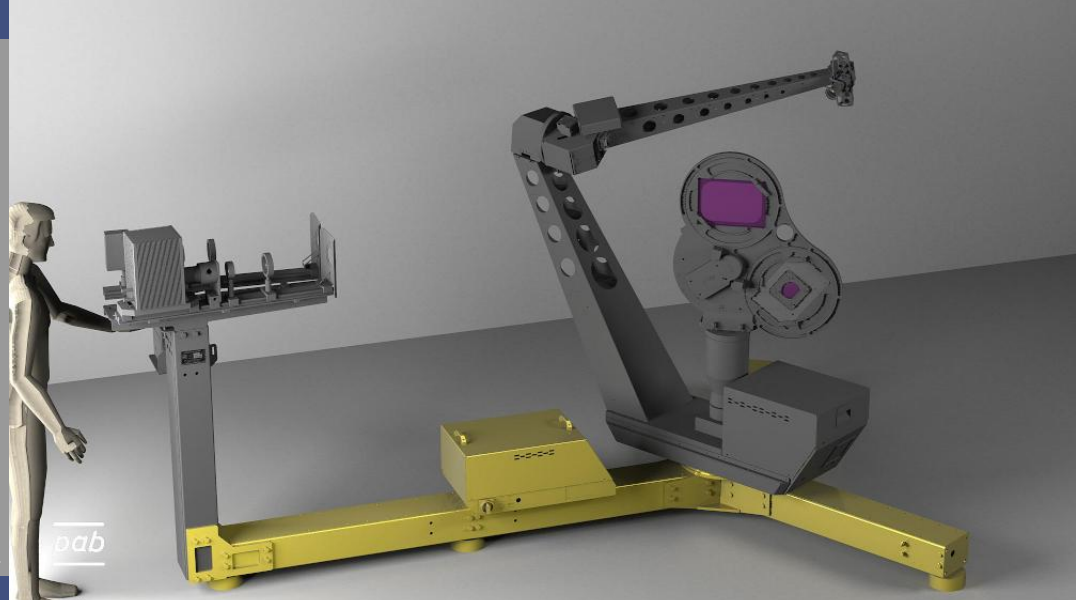
Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2



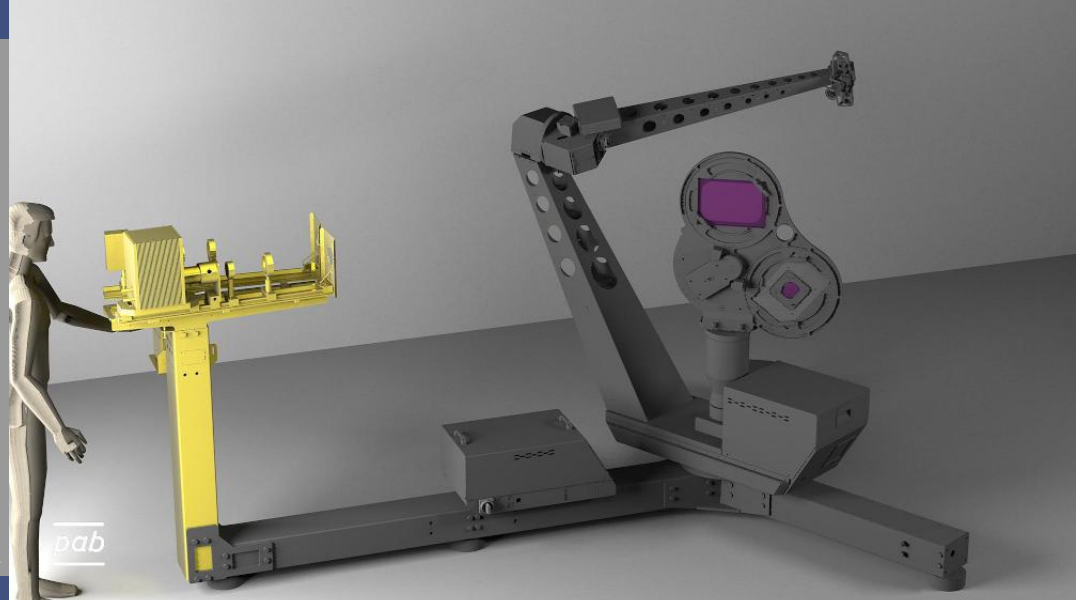
Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2



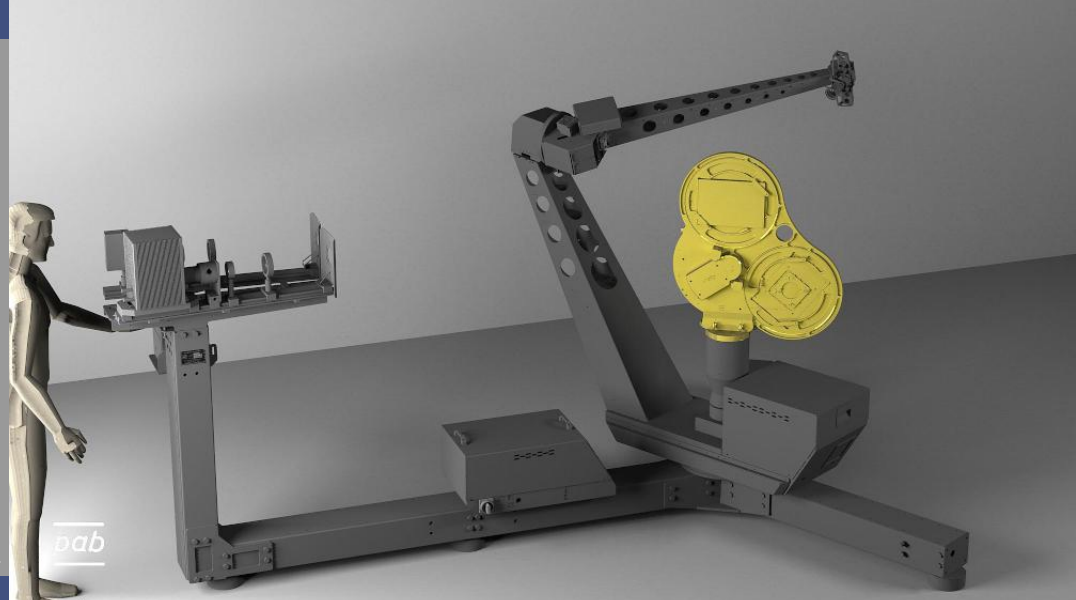
Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2



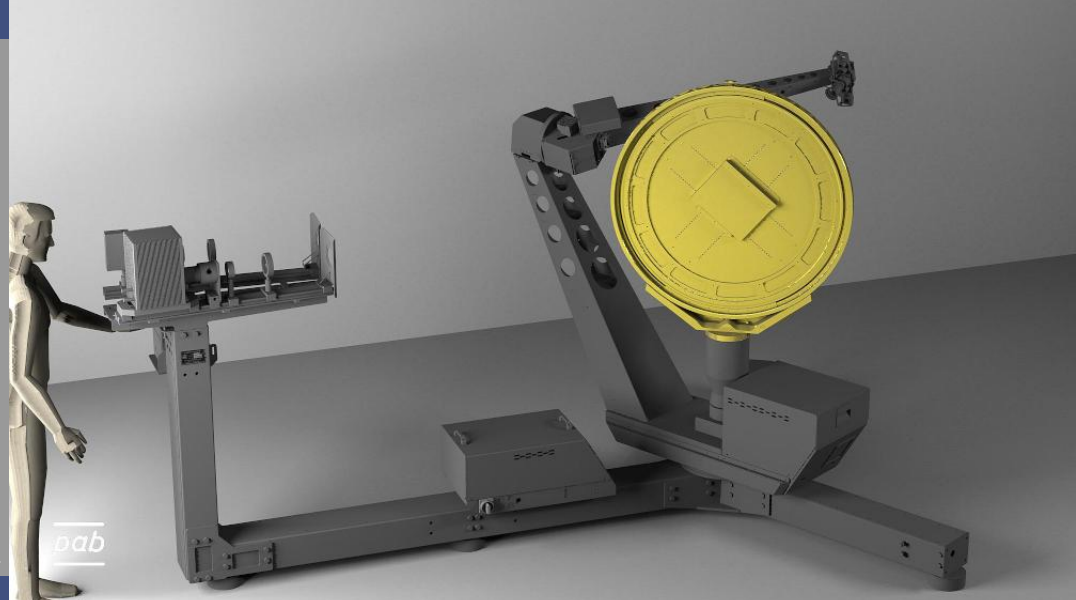
Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2



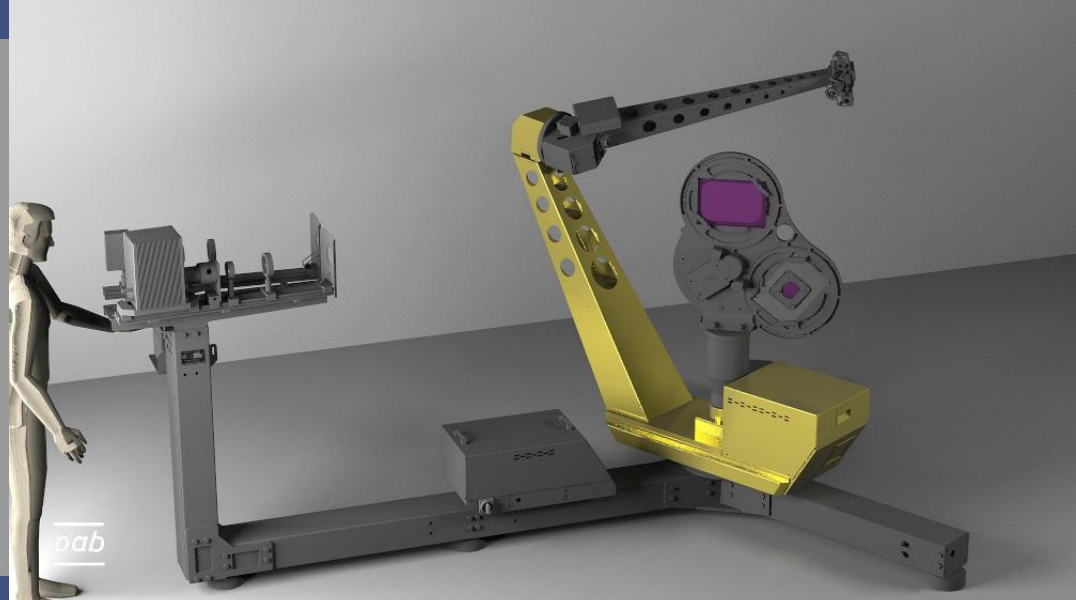
Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2



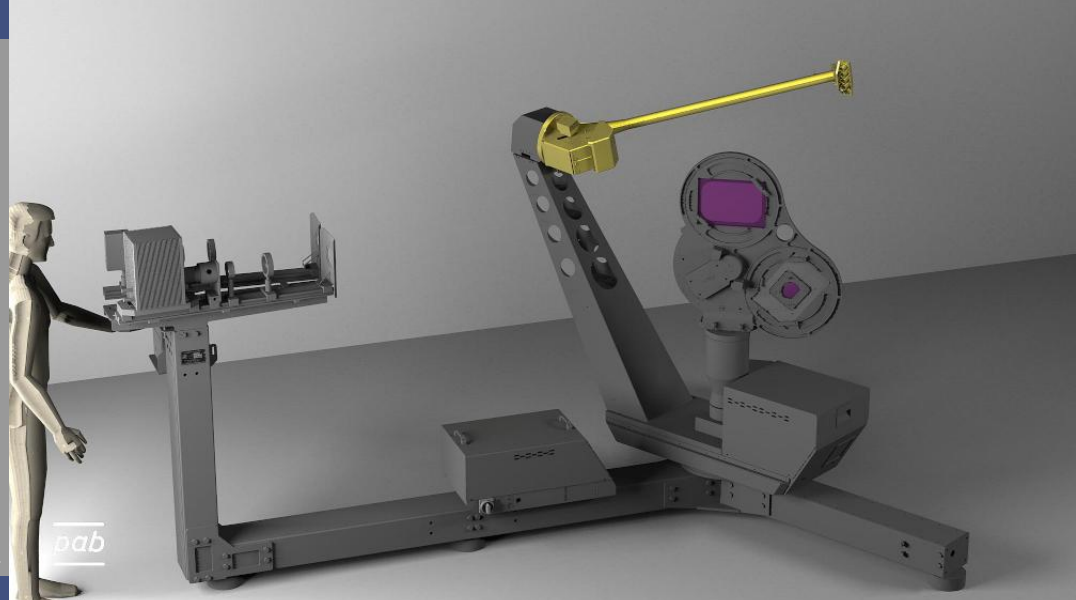
Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2



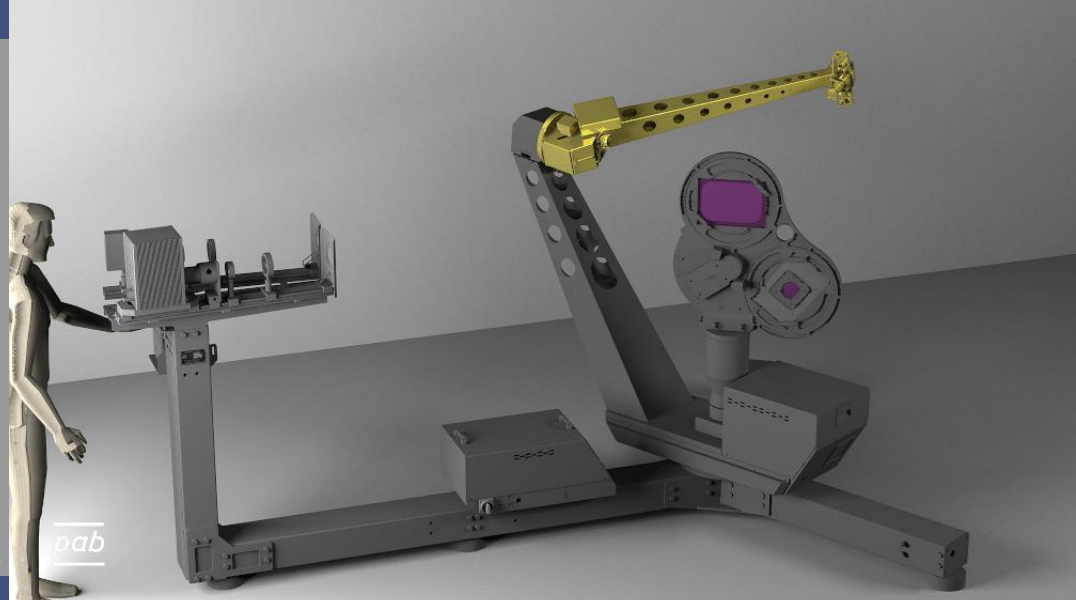
Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2



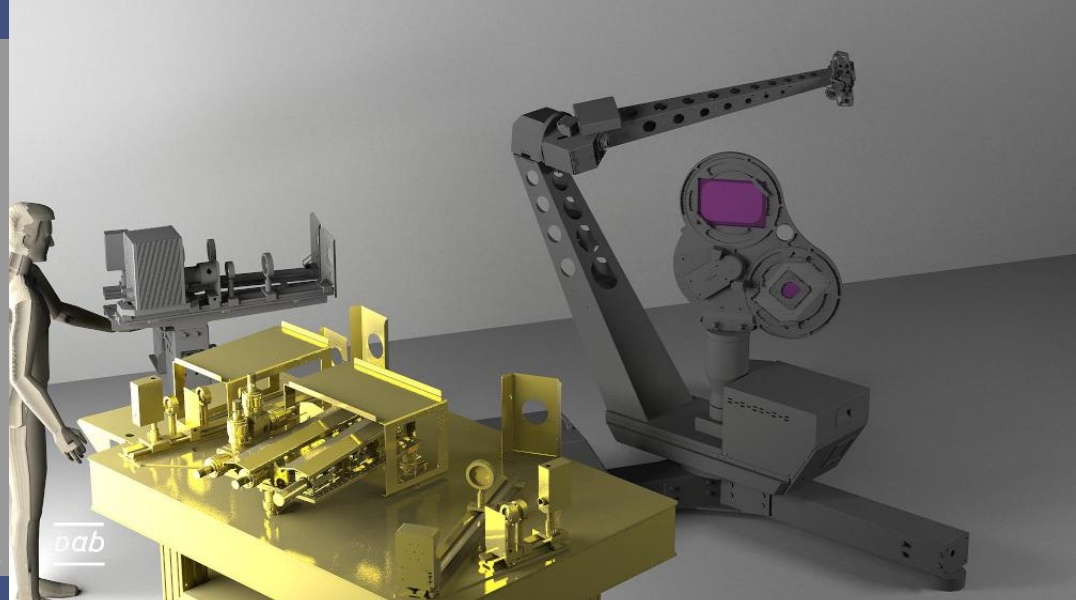
Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2



Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2



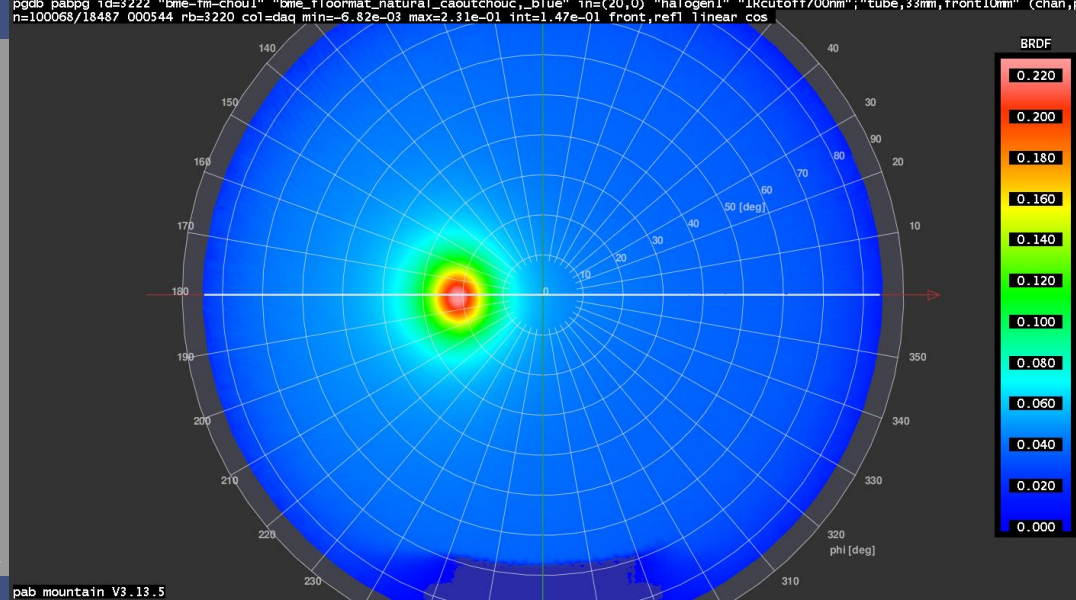
Tool-time: Measuring the BSDF

- scanning gonio-photometers, e.g. pab PG2
- image based gonio-photometers

Tool-time: Visualising the BSDF

plotting hemispherical BSDF for one incident (θ_{in}, ϕ_{in}) with **mountain**

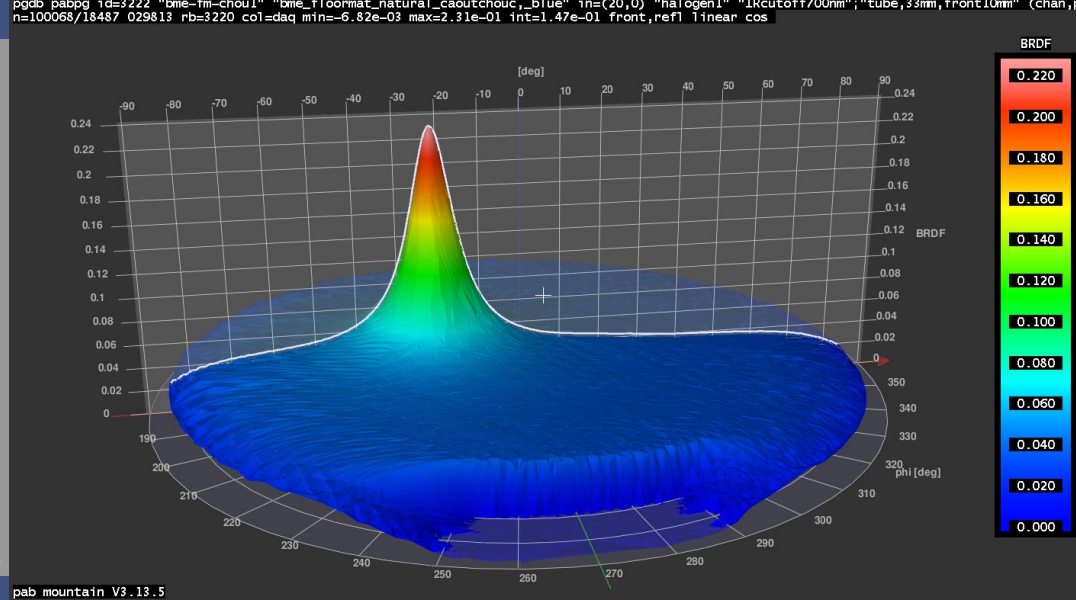
- natural rubber floor-tile, orthogonal view



Tool-time: Visualising the BSDF

plotting hemispherical BSDF for one incident (θ_{in}, ϕ_{in}) with **mountain**

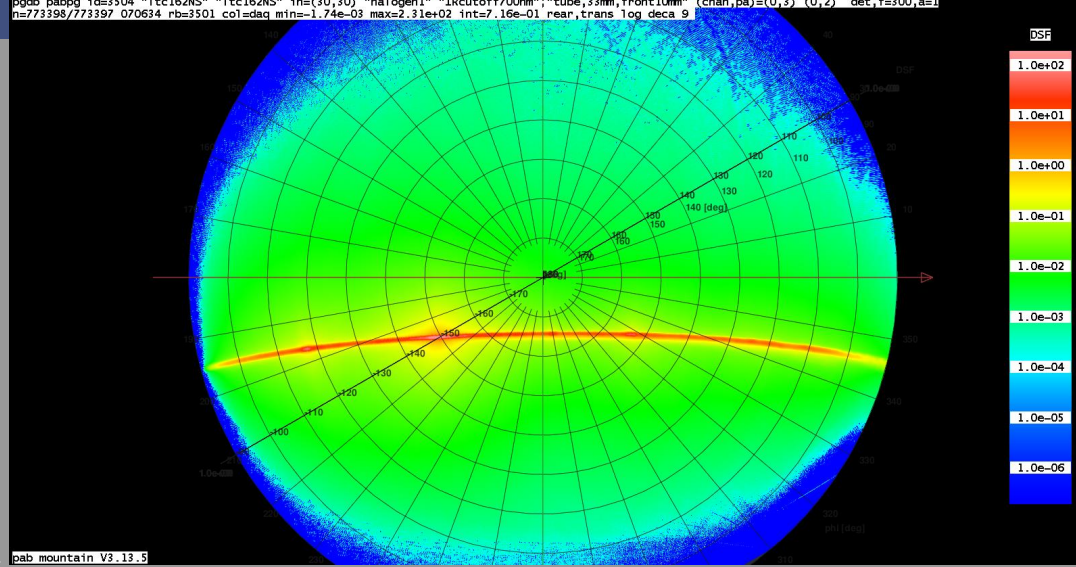
- natural rubber floor-tile, oblique view



Tool-time: Visualising the BSDF

plotting hemispherical BSDF for one incident (θ_{in}, ϕ_{in}) with **mountain**

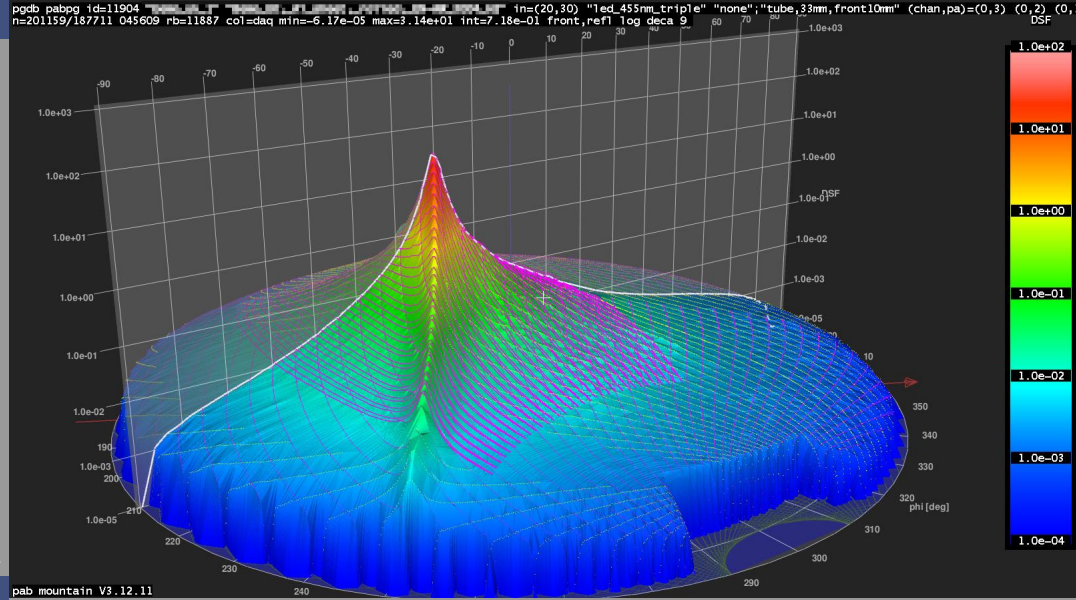
- natural rubber floor-tile,
- clear twin-wall plastic



Tool-time: Visualising the BSDF

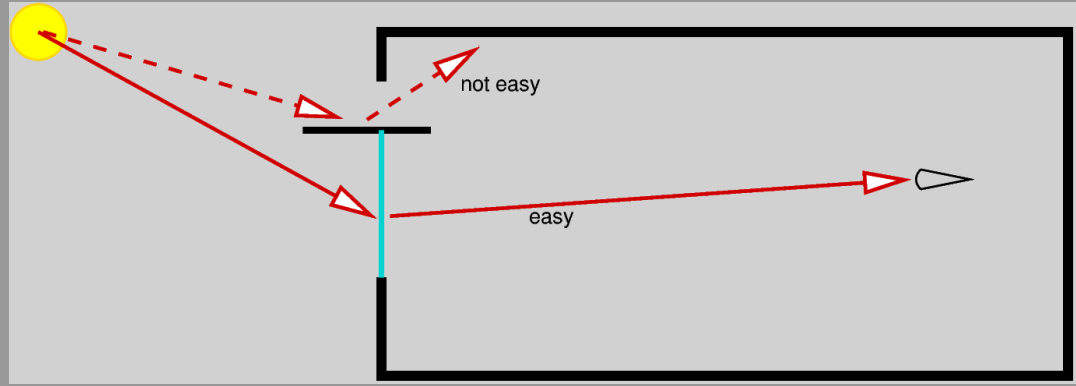
plotting hemispherical BSDF for one incident (θ_{in}, ϕ_{in}) with **mountain**

- natural rubber floor-tile,
- clear twin-wall plastic
- aluminium roof material



Results of your material modelling may be affected by how RADIANCE works:

- objects are either emitting-sources or passive surfaces.
  Advantage during rendering: list of known sources to check for.

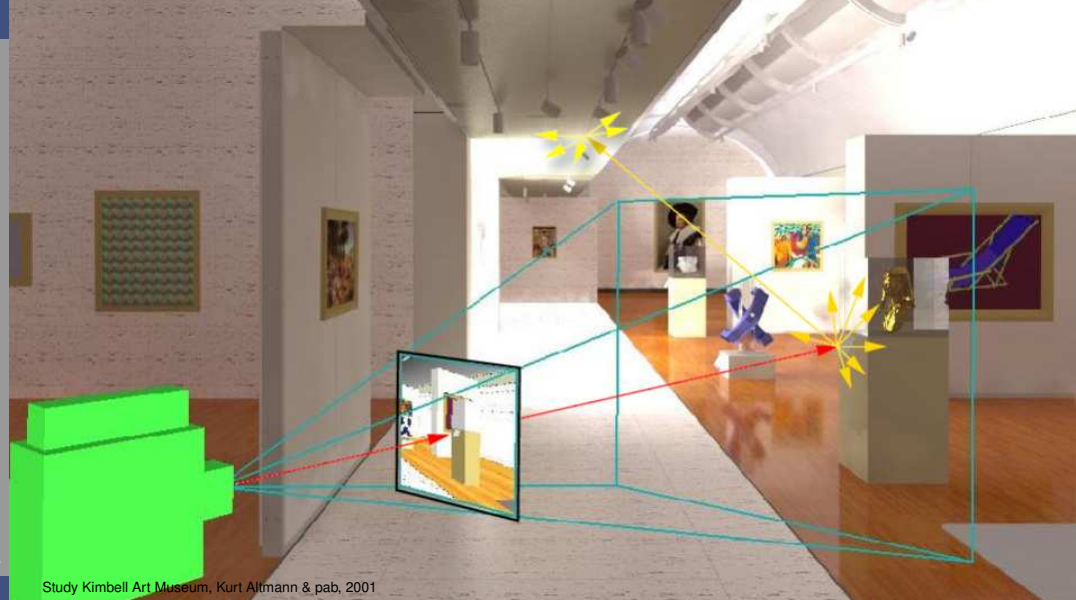


RADIANCE Rendering and Materials

Results of your material modelling may be affected by how RADIANCE works:

- objects are either emitting-sources or passive surfaces.
  Advantage during rendering: list of known sources to check for.

- is basically a backward-raytracer: from eye to sources
  disadvantage: hard to find ray-paths with near-specular surfaces in-between



Study Kimbell Art Museum, Kurt Altmann & pab, 2001

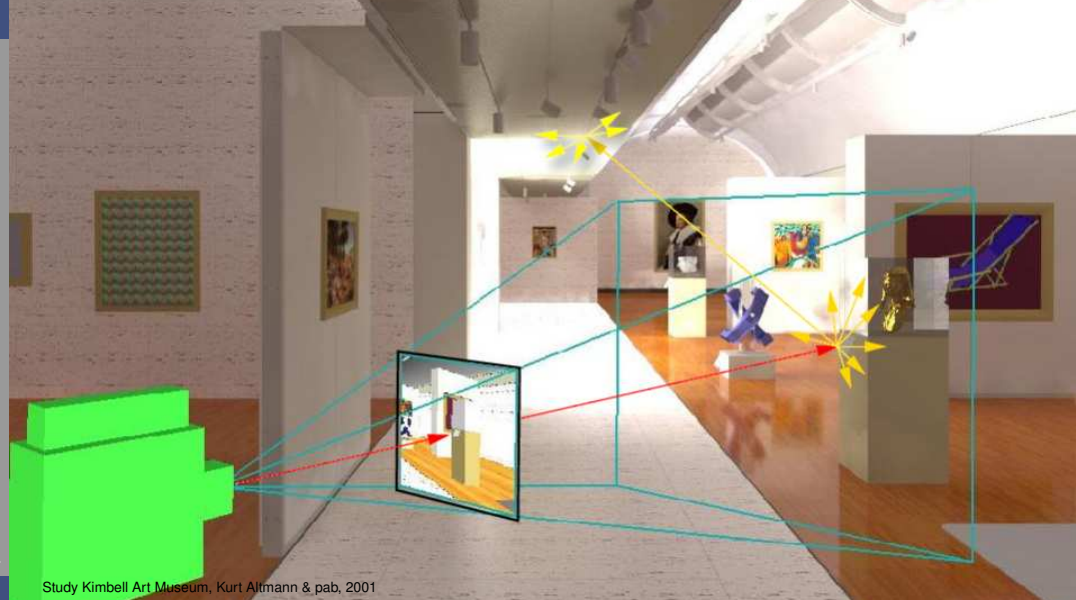
RADIANCE Rendering and Materials

Results of your material modelling may be affected by how RADIANCE works:

- objects are either emitting-sources or passive surfaces.
  Advantage during rendering: list of known sources to check for.

- is basically a backward-raytracer: from eye to sources
  disadvantage: hard to find ray-paths with near-specular surfaces in-between

⇝ remember to set **rpict** parameters well, and/or use **pmap** (PhotonMap) extension



Study Kimbell Art Museum, Kurt Altmann & pab, 2001

Material Types in RADIANCE

- light sources

- **light**
- **spotlight**
- **glow**

Material Types in RADIANCE

- light sources
- passive materials with parameters

- classical, Ward, Gaussian lobe + constant:
  plastic, **metal**, **trans**

  asymmetric versions:
  plastic2, **metal2**, **trans2**

- classical transparent non-scattering:
  glass, **dielectric**, **interface**

- newer:
  ashik2

Material Types in RADIANCE

- light sources
- passive materials with parameters
- passive materials with function file or data

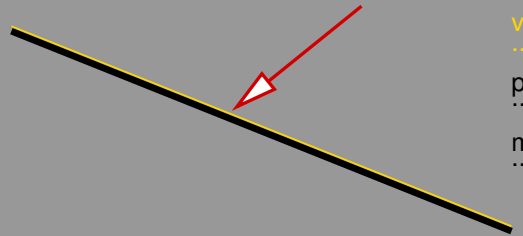
- legacy:
  plasfunc, **metfunc**, **transfunc**, **brtdfunc**: function file
  plasdata, **metdata**, **transdata**: data, but linear interpolation only
- newer:
  BSDF , XML based, ”tensor-tree”, lobe-based interpolation

  aBSDF with extra, limited, peak-handling

Material Types in RADIANCE

- light sources
- passive materials with parameters
- passive materials with function file or data
- ”patterns”: modifications of a material parameter (e.g. colour)
  e.g.: to map a picture onto a mesh or a polygon

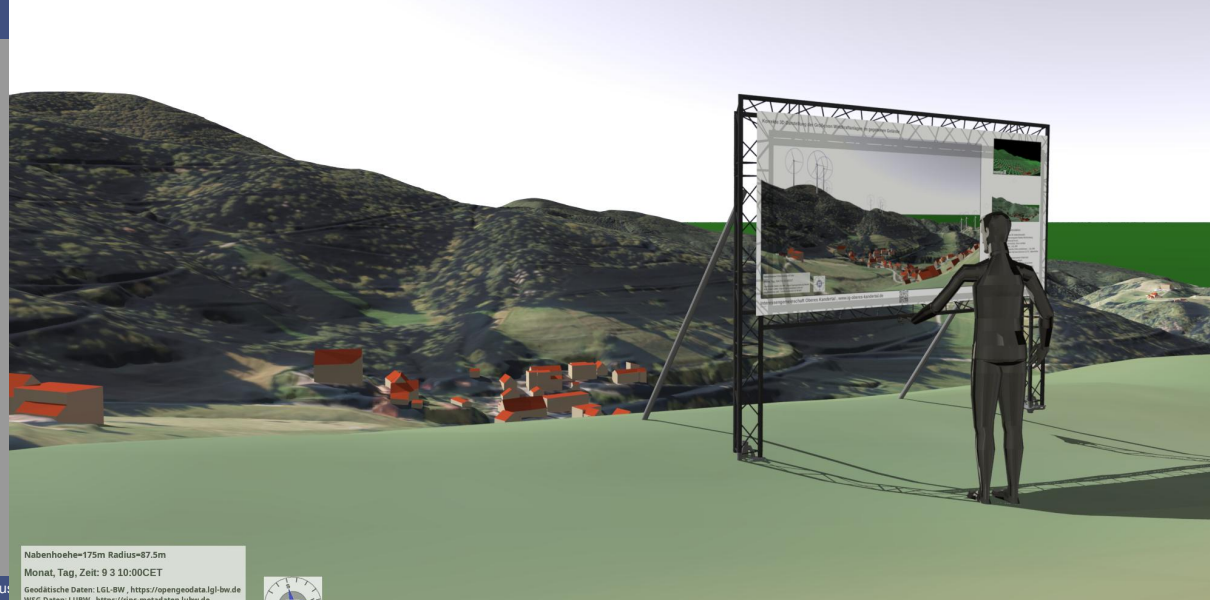
- materials with user supplied functions
  - **brightfunc**, **colorfunc**, **transfunc**, **mixfunc**
- material models using data-files with linear interpolation
  - **colordata**, **brightdata**, **colorpict**
  - **colortext**, **brighttext**
  - **plasdata**, **metdata**, **transdata**



```
void colorpict p1
...
p1 plastic mymat
...
mymat polygon p1
...
```

Material Types in RADIANCE

- light sources
- passive materials with parameters
- passive materials with function file or data
- ”patterns”: modifications of a material parameter (e.g. colour)
  e.g.: to map a picture onto a mesh or a polygon



Material Types in RADIANCE

- light sources
- passive materials with parameters
- passive materials with function file or data
- ”patterns”: modifications of a material parameter (e.g. colour)
  e.g.: to map a picture onto a mesh or a polygon

Lörrach/Weil am Rhein • Dienstag, 1. Juli 2025
https://www.badische-zeitung.de/exkursion-in-malsburg-marzell-will-die-auswirkungen-von-windkraft-auf-umwelt-zeigen

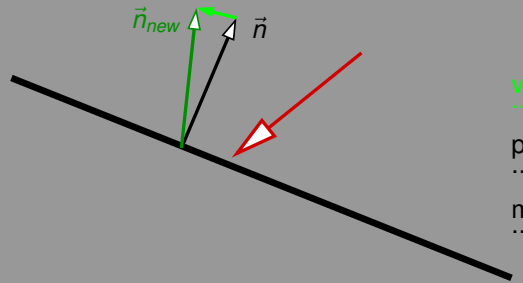
Bei der Quellen-Exkursion mit Alt- Wassermeister Wehrle FOTO: KARLHEINZ BEYERLE

Erhellende Einblicke in den Untergrund

Material Types in RADIANCE

- light sources
- passive materials with parameters
- passive materials with function file or data
- ”patterns”: modifications of a material parameter (e.g. colour)
  e.g.: to map a picture onto a mesh or a polygon
- ”textures”: modifications of the surface normal

- **texfunc**, **texdata**




```
void texfunc p1
...

p1 plastic mymat
...

mymat polygon p1
...
```

Material Types in RADIANCE

- light sources
- passive materials with parameters
- passive materials with function file or data
- ”patterns”: modifications of a material parameter (e.g. colour)
  e.g.: to map a picture onto a mesh or a polygon
- ”textures”: modifications of the surface normal

und image: teWhare, Lake Tarawera, NZ, pab, January 2015

Material Types in RADIANCE

- light sources
- passive materials with parameters
- passive materials with function file or data
- ”patterns”: modifications of a material parameter (e.g. colour)
  e.g.: to map a picture onto a mesh or a polygon
- ”textures”: modifications of the surface normal
- materials with special ”side effects” during rendering

antimatter : subtract a volume (entry-level CSG)

illum : envelop a complex geometry (e.g. light-redirecting glazing)

mirror : secondary light sources (mirror images of sources)

mist : participating media, volume scattering

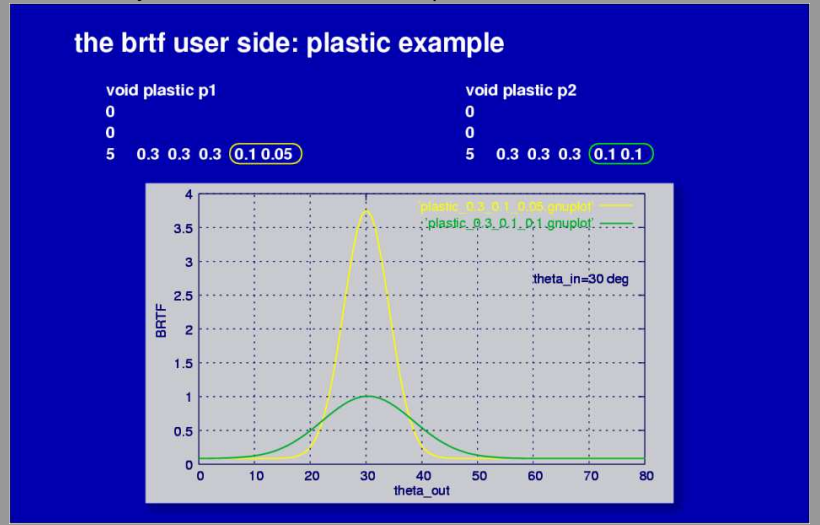
prism : legacy special for prismatic glazing

Parametric Materials: **plastic**, **metal**

The all-time favourite: **plastic**

- function type: Gaussian + constant

slide from my talk at RADIANCE Workshop 2004:

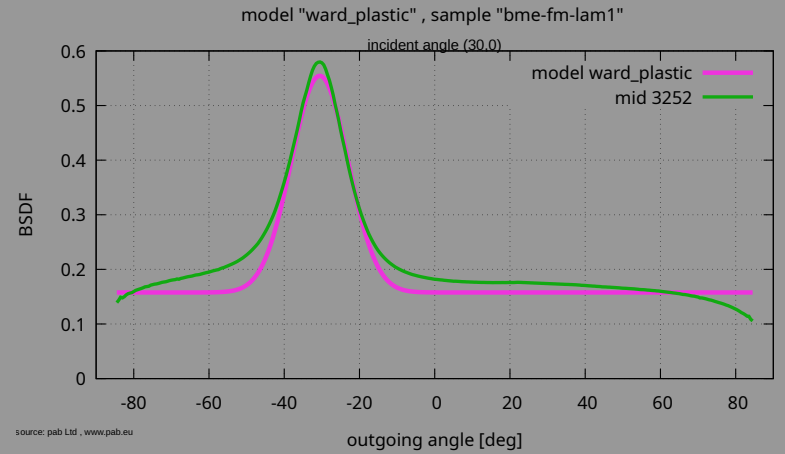


Parametric Materials: **plastic**, **metal**

The all-time favourite: **plastic**

- function type: Gaussian + constant

data source, 2011 : http://bme.pab.eu:



RGB= 0.51 specularity=0.031 roughness=0.092

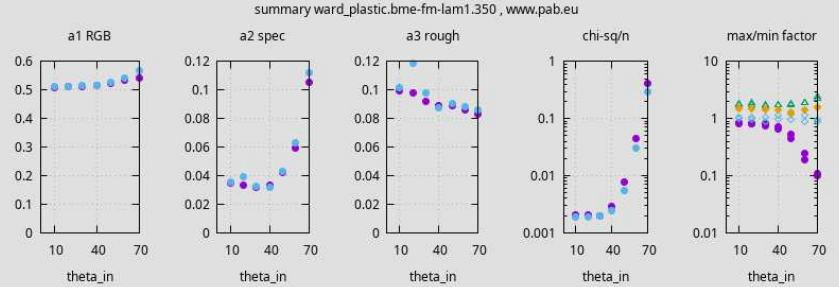
Parametric Materials: **plastic**, **metal**

The all-time favourite: **plastic**

- function type: Gaussian + constant
- however: fitted parameter may depend on θ_{in}

ward_plastic* fit of *bme-fm-lam1

description	laminate,_light_wood_(WZ)
label	bme-fm-lam1
BSDF model	ward_plastic



Parametric Materials: **plastic**, **metal**

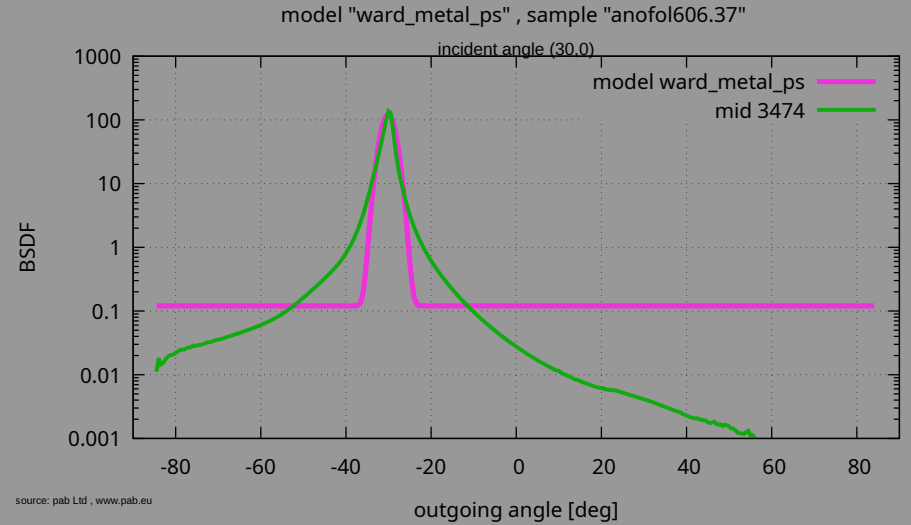
The all-time favourite: **plastic**

- function type: Gaussian + constant
- however: fitted parameter may depend on θ_{in}
- more: BSDF PG2 data: `http://bme.pab.eu` , ... well, actually since July 2011

Parametric Materials: **plastic**, **metal**

The all-time favourite: **plastic**

- function type: Gaussian + constant
- however: fitted parameter may depend on θ_{in}
- more: BSDF PG2 data: `http://bme.pab.eu` , ... well, actually since July 2011
- summary from measurements:
  Good fit for mildly ”diffuse” samples, worse for anything ”shiny”



RGB= 0.81 specularity=0.53 roughness=0.019

Parametric Materials: **plastic**, **metal**

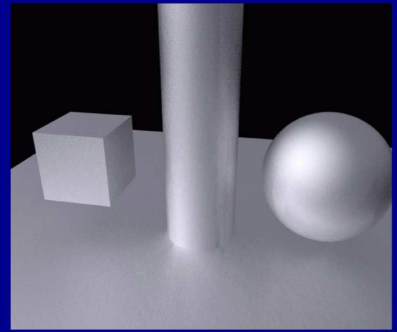
The all-time favourite: **plastic**

- function type: Gaussian + constant
- however: fitted parameter may depend on θ_{in}
- more: BSDF PG2 data: `http://bme.pab.eu` , ... well, actually since July 2011
- summary from measurements:
  Good fit for mildly ”diffuse” samples, worse for anything ”shiny”

not a new fact. from 1st RADIANCE Workshop in 2002:

Example: Sandblasted Aluminium Surface

- metal applied to objects

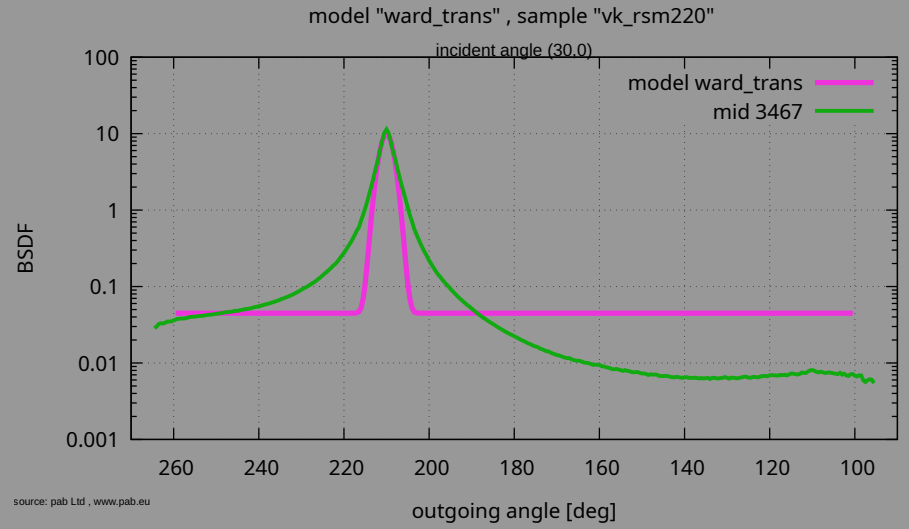


however: fitted parameters out of range

Parametric Materials: **trans**

Translucent materials, the first iteration : **trans**

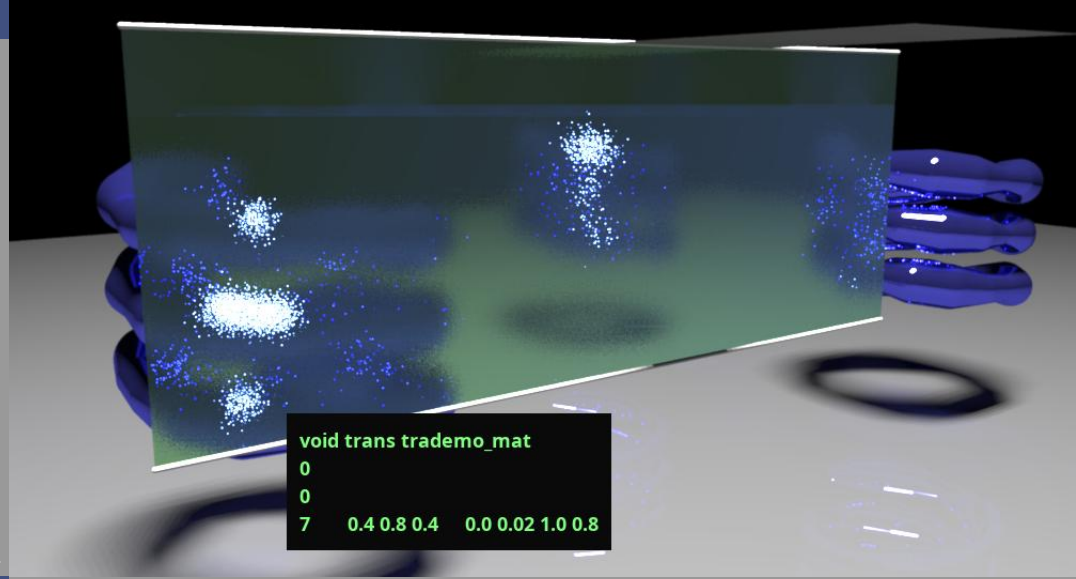
- function type: Gaussian + constant



Parametric Materials: **trans**

Translucent materials, the first iteration : **trans**

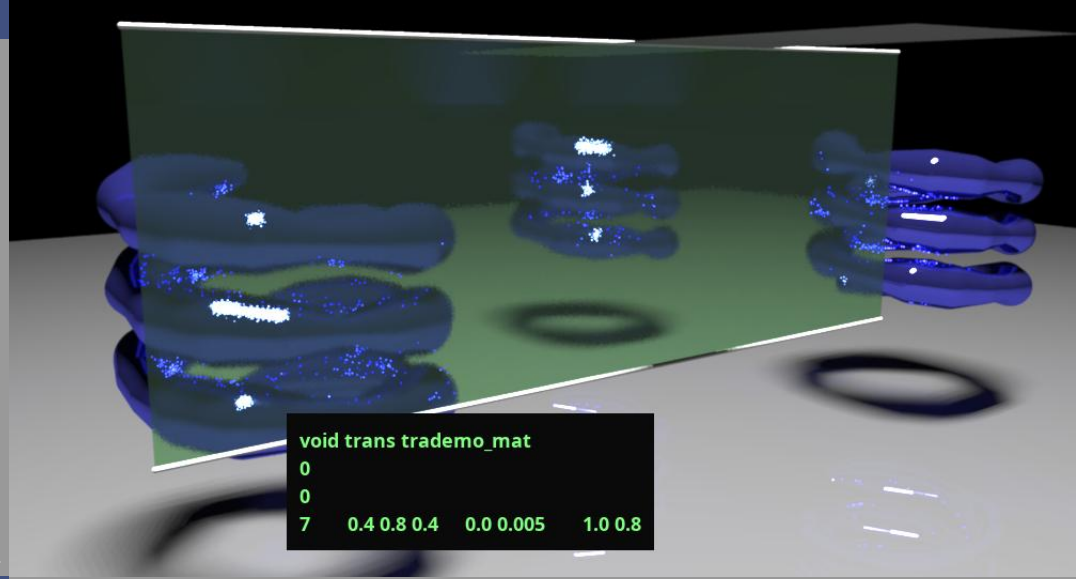
- function type: Gaussian + constant
- parameter examples



Parametric Materials: **trans**

Translucent materials, the first iteration : **trans**

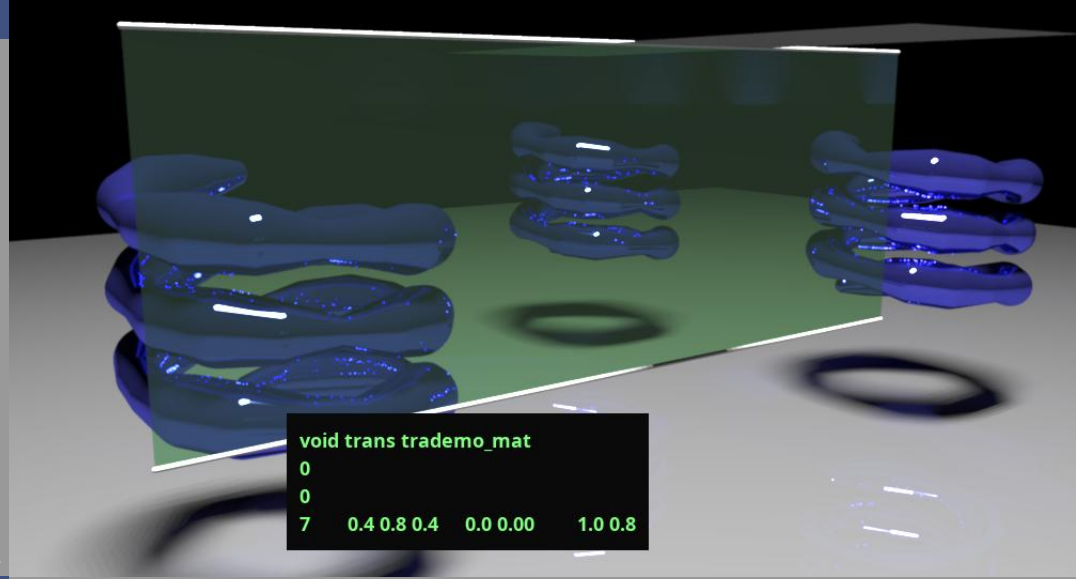
- function type: Gaussian + constant
- parameter examples



Parametric Materials: **trans**

Translucent materials, the first iteration : **trans**

- function type: Gaussian + constant
- parameter examples



Parametric Materials: **trans**

Translucent materials, the first iteration : **trans**

- function type: Gaussian + constant
- parameter examples

Parametric Materials: **trans**

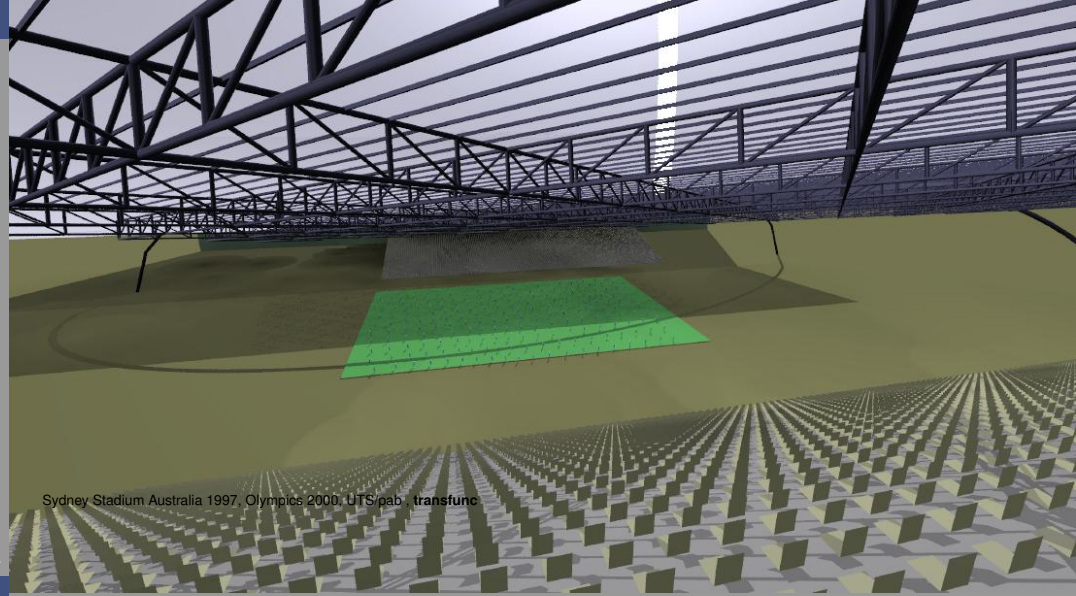
Translucent materials, the first iteration : **trans**

- function type: Gaussian + constant
- parameter examples
- conclusion: easy parameters selection, yet rarely fits real translucent materials

User Functions for Materials: **plasfunc**, **transfunc**

Nearly a catch-all for a user designing a ”good” function

- simple **transfunc** example: twin-wall clear plastic



Sydney Stadium Australia 1997, Olympics 2000, UTS/pab, **transfunc**

User Functions for Materials: **plasfunc**, **transfunc**

Nearly a catch-all for a user designing a ”good” function

- simple **transfunc** example: twin-wall clear plastic

user function defined in ASCII textfile file, excerpt from Sydney2000 cellbrtf.cal file:

```
{ angle between extrusion vector and 'a' direction in [deg] }

alpha(ax,ay,az) = dAcos( sklp( ax,ay,az, Ex,Ey,Ez ) );

{ forward peak scattering }

forward(lx,ly,lz)= if( dAcos( sklp(lx,ly,lz,Dx,Dy,Dz) ) - 2*conewidth, minvalue, maxvalue );

{ combine both }

temp(lx,ly,lz,ff)= if( abs(alpha(lx,ly,lz)-alpha(Dx,Dy,Dz)) - conewidth , 0.9*ff , maxvalue );

{ speed up by calling 'forward' once only }

cellbrtf(lx,ly,lz,Omega)= temp(lx,ly,lz, forward(lx,ly,lz) );
```

User Functions for Materials: **plasfunc**, **transfunc**

Nearly a catch-all for a user designing a ”good” function

- simple **transfunc** example: twin-wall clear plastic
- very flexible and powerful
  e.g. by defining and using other coordinate systems

user function defined in ASCII textfile file, excerpt from Sydney2000 cellbrtf.cal file:

```
{ angle between extrusion vector and 'a' direction in [deg] }

alpha(ax,ay,az) = dAcos( sklp( ax,ay,az, Ex,Ey,Ez ) );

{ forward peak scattering }

forward(lx,ly,lz)= if( dAcos( sklp(lx,ly,lz,Dx,Dy,Dz) ) - 2*conewidth, minvalue, maxvalue );

{ combine both }

temp(lx,ly,lz,ff)= if( abs(alpha(lx,ly,lz)-alpha(Dx,Dy,Dz)) - conewidth , 0.9*ff , maxvalue );

{ speed up by calling 'forward' once only }

cellbrtf(lx,ly,lz,Omega)= temp(lx,ly,lz, forward(lx,ly,lz) );
```

User Functions for Materials: **plasfunc**, **transfunc**

Nearly a catch-all for a user designing a ”good” function

- simple **transfunc** example: twin-wall clear plastic
- very flexible and powerful
  e.g. by defining and using other coordinate systems

current (2025) drawbacks:

- not supported by all rendering paths
  but further processing possible (e.g. ⇝ **bsdf2tree** XML)

user function defined in ASCII textfile file, excerpt from Sydney2000 cellbrtf.cal file:

```
{ angle between extrusion vector and 'a' direction in [deg] }

alpha(ax,ay,az) = dAcos( sklp( ax,ay,az, Ex,Ey,Ez ) );

{ forward peak scattering }

forward(lx,ly,lz)= if( dAcos( sklp(lx,ly,lz,Dx,Dy,Dz) ) - 2*conewidth, minvalue, maxvalue );

{ combine both }

temp(lx,ly,lz,ff)= if( abs(alpha(lx,ly,lz)-alpha(Dx,Dy,Dz)) - conewidth , 0.9*ff , maxvalue );

{ speed up by calling 'forward' once only }

cellbrtf(lx,ly,lz,Omega)= temp(lx,ly,lz, forward(lx,ly,lz) );
```

User Functions for Materials: **plasfunc**, **transfunc**

Nearly a catch-all for a user designing a ”good” function

- simple **transfunc** example: twin-wall clear plastic
- very flexible and powerful
  e.g. by defining and using other coordinate systems

current (2025) drawbacks:

- not supported by all rendering paths
  but further processing possible (e.g. ⇝ **bsdf2tree** XML)
- function files tend to be ”write once”,
  some syntax (e.g. `if(a,b,c)`) makes them hard to read

user function defined in ASCII textfile file, excerpt from Sydney2000 cellbrtf.cal file:

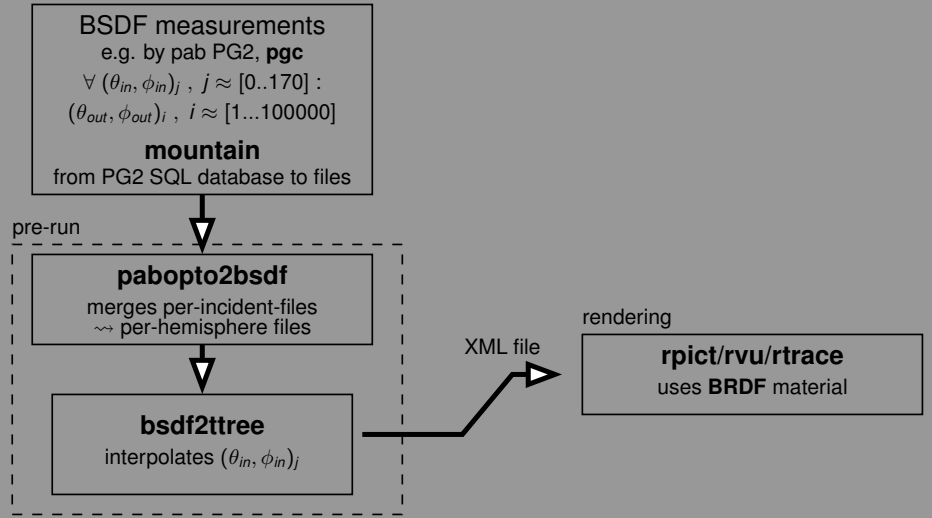
```
{ angle between extrusion vector and 'a' direction in [deg] }

alpha(ax,ay,az) = dAcos( sklp( ax,ay,az, Ex,Ey,Ez ) );

{ forward peak scattering }

forward(lx,ly,lz)= if( dAcos( sklp(lx,ly,lz,Dx,Dy,Dz) ) - 2*conewidth, minvalue, maxvalue );

{ combine both }

temp(lx,ly,lz,ff)= if( abs(alpha(lx,ly,lz)-alpha(Dx,Dy,Dz)) - conewidth , 0.9*ff , maxvalue );

{ speed up by calling 'forward' once only }

cellbrtf(lx,ly,lz,Omega)= temp(lx,ly,lz, forward(lx,ly,lz) );
```

Data-Driven: **BSDF**

Nearly a catch-all with measured BSDF data: XML based **BSDF**

- very useful if measured BSDF data is available

RADIANCE **BSDF** material data flow:

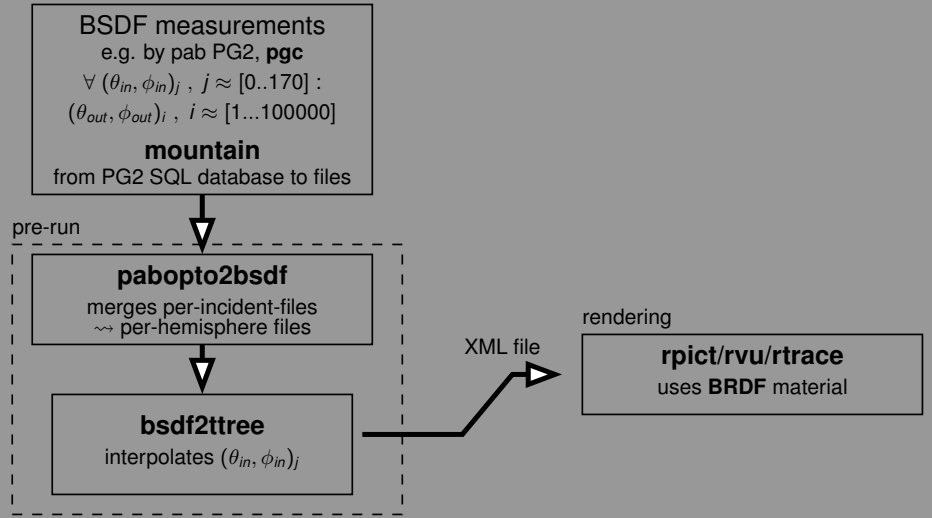


Data-Driven: **BSDF**

Nearly a catch-all with measured BSDF data: XML based **BSDF**

- very useful if measured BSDF data is available
- no custom functions, no creative thinking-per-sample-type required

RADIANCE **BSDF** material data flow:

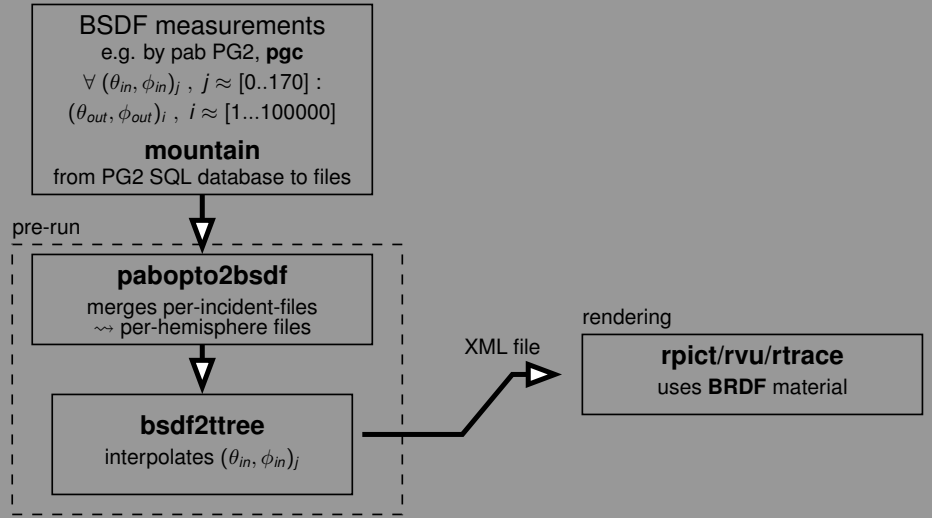


Data-Driven: **BSDF**

Nearly a catch-all with measured BSDF data: XML based **BSDF**

- very useful if measured BSDF data is available
- no custom functions, no creative thinking-per-sample-type required
- solves interpolation between incident angles

RADIANCE **BSDF** material data flow:

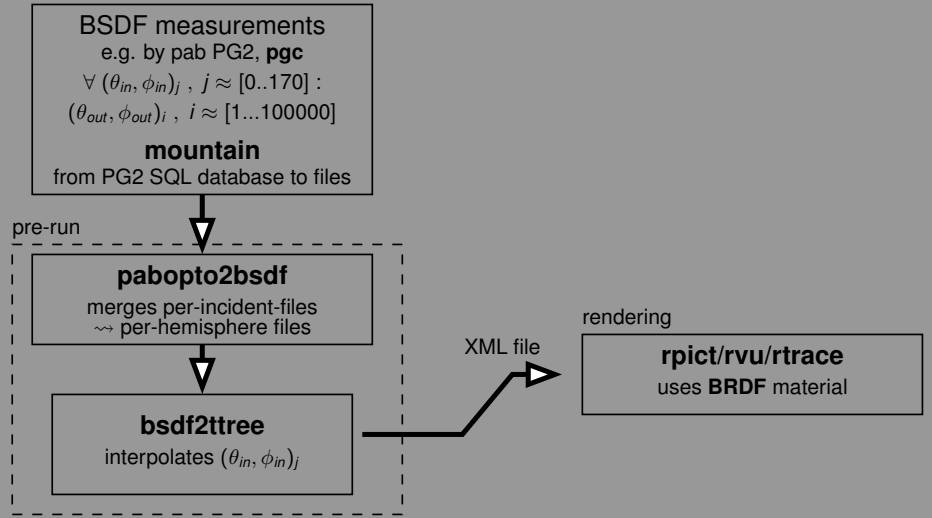


Data-Driven: **BSDF**

Nearly a catch-all with measured BSDF data: XML based **BSDF**

- very useful if measured BSDF data is available
- no custom functions, no creative thinking-per-sample-type required
- solves interpolation between incident angles
- alternative input: function files

RADIANCE **BSDF** material data flow:



Data-Driven: **BSDF**

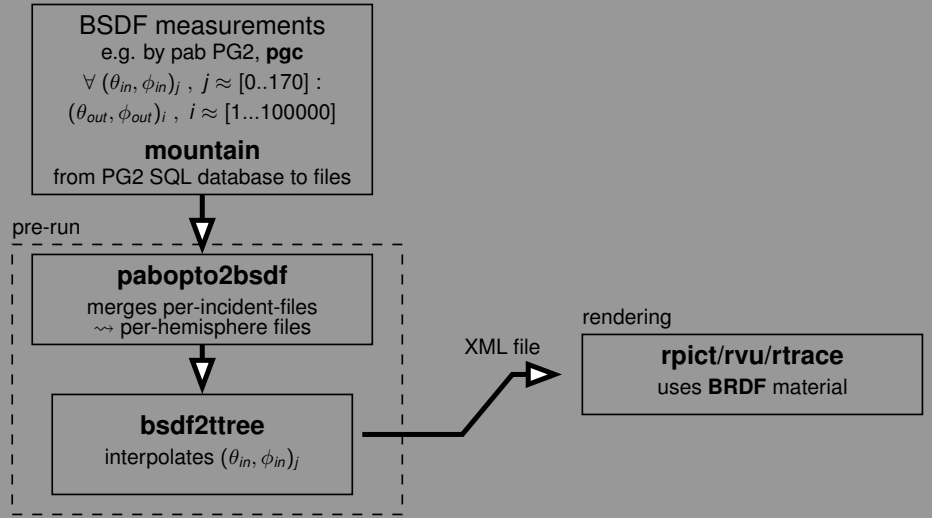
Nearly a catch-all with measured BSDF data: XML based **BSDF**

- very useful if measured BSDF data is available
- no custom functions, no creative thinking-per-sample-type required
- solves interpolation between incident angles
- alternative input: function files

current (2025) drawbacks:

- artefacts limit use in glare studies with sun-path animation

RADIANCE **BSDF** material data flow:



Data-Driven: **BSDF**

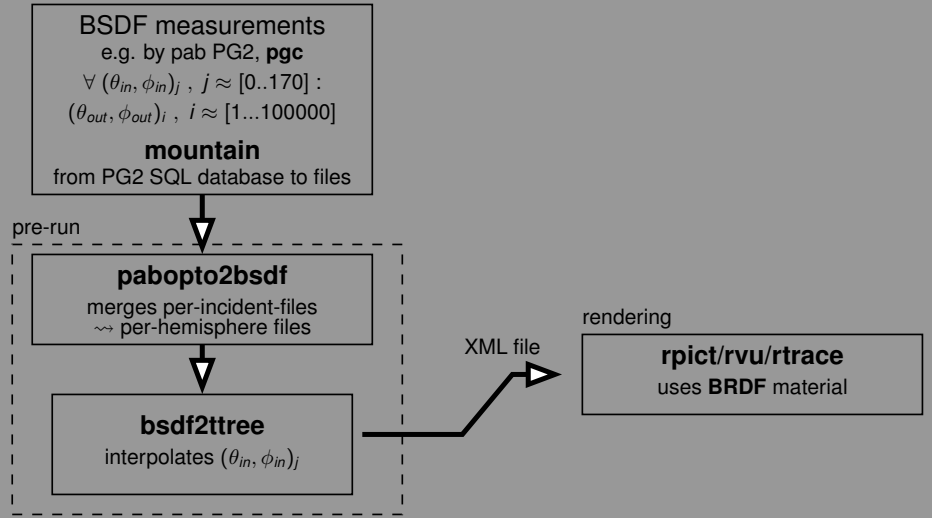
Nearly a catch-all with measured BSDF data: XML based **BSDF**

- very useful if measured BSDF data is available
- no custom functions, no creative thinking-per-sample-type required
- solves interpolation between incident angles
- alternative input: function files

current (2025) drawbacks:

- artefacts limit use in glare studies with sun-path animation
- only limited handling of small-angle-scattering (”peak” extraction)

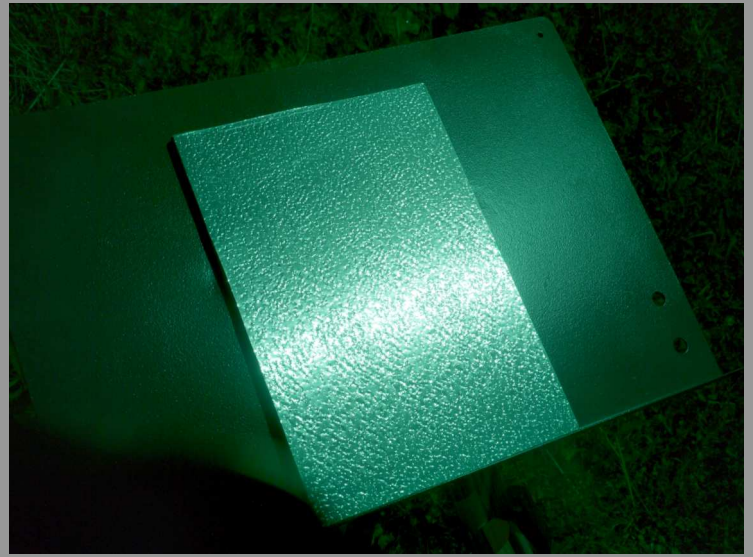
RADIANCE **BSDF** material data flow:



Case Study 1, rolled stainless steel roof sheet

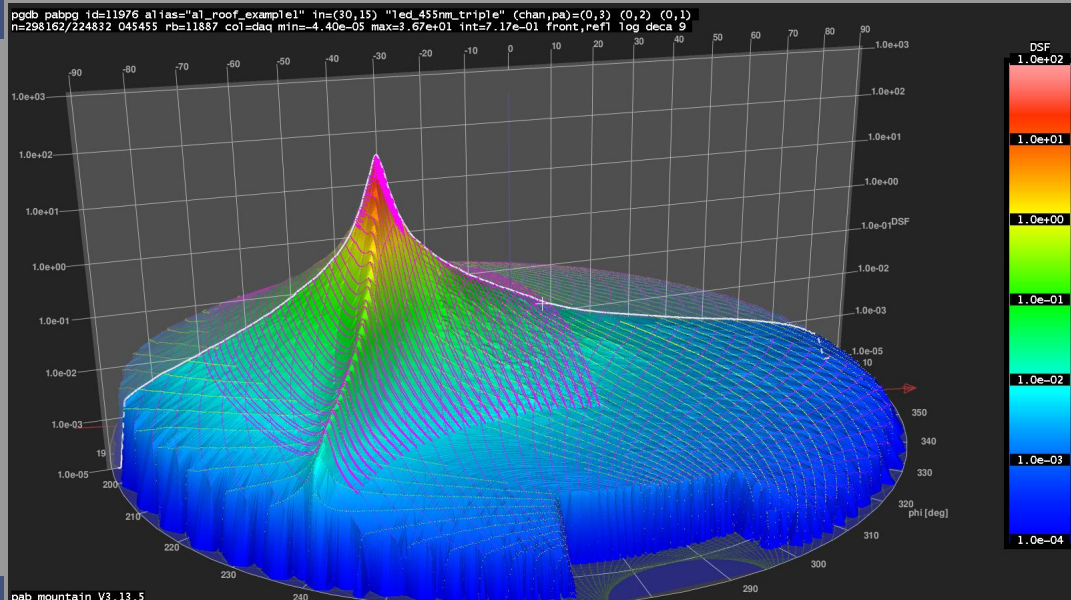
- photo

HDR photo taken through D12 welding glass in summer sunshine



Case Study 1, rolled stainless steel roof sheet

- photo
- measured data



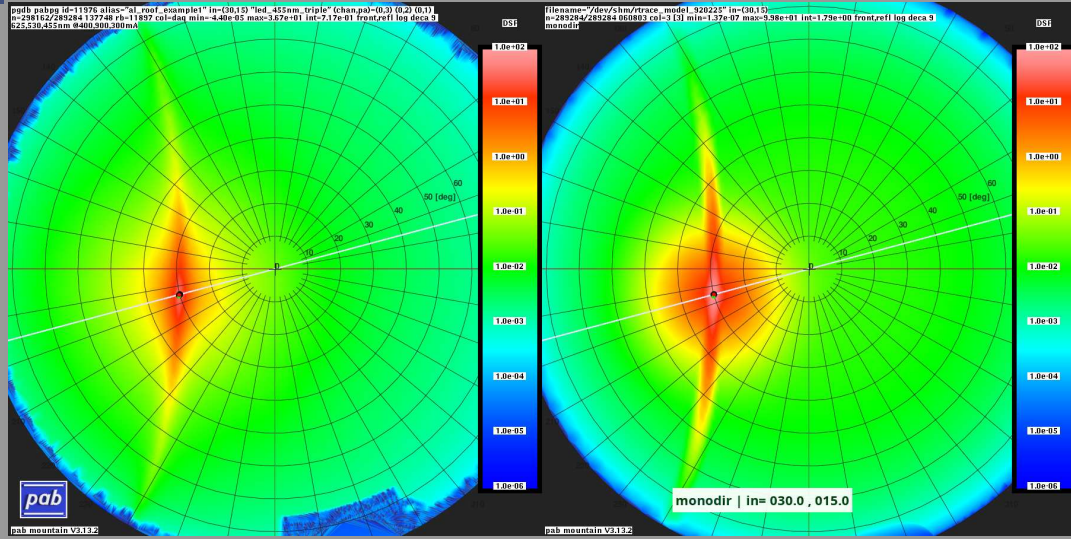
Case Study 1, rolled stainless steel roof sheet

- photo
- measured data

comparison measured / model, using ”virtual `rtrace` gonio-photometer” :

- function file model ”monodir”

comparison: data (left) and function model (right)



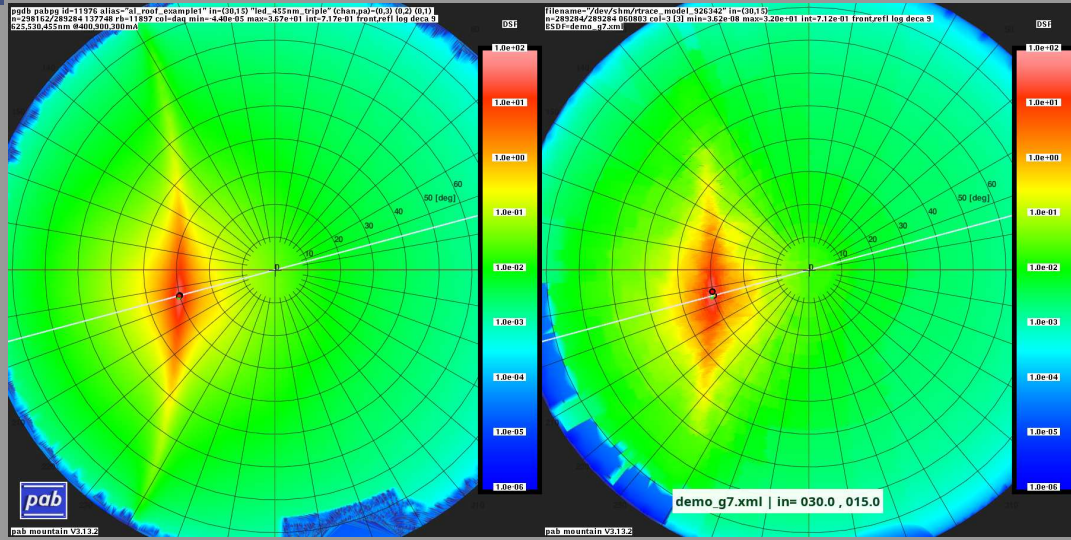
Case Study 1, rolled stainless steel roof sheet

- photo
- measured data

comparison measured / model, using ”virtual `rtrace` gonio-photometer” :

- function file model ”monodir”
- XML model

comparison: data (left) and XML model (right)



Case Study 1, rolled stainless steel roof sheet

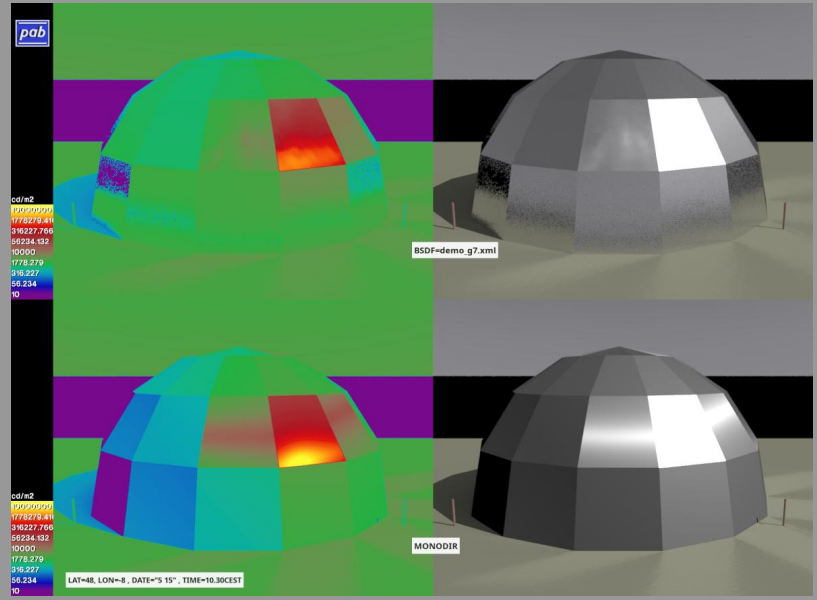
- photo
- measured data

comparison measured / model, using ”virtual `rtrace` gonio-photometer” :

- function file model ”monodir”
- XML model

demo building

- sun-path animation of comparison XML versus function
  slight ”jumps” in incident interpolation cause practical problem at client
  ($N = 7, t = 0.9$)



Case Study 1, rolled stainless steel roof sheet

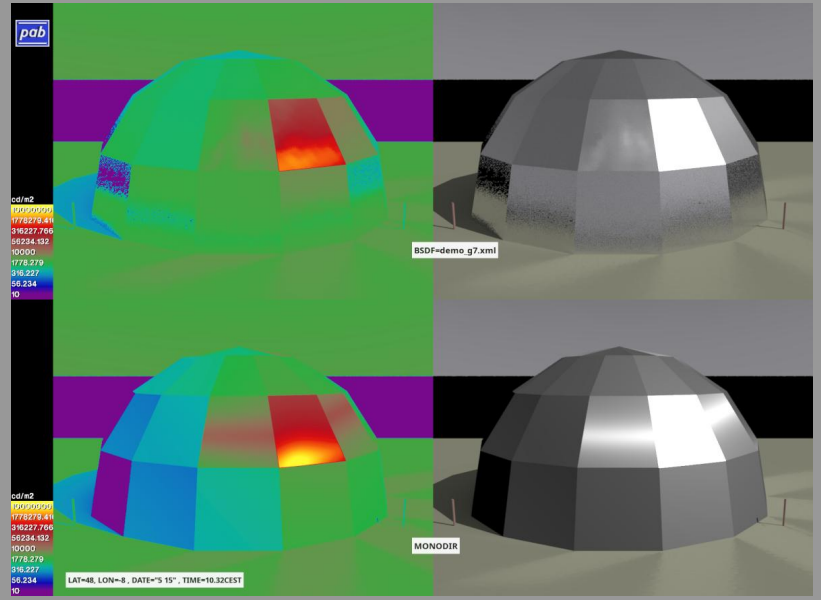
- photo
- measured data

comparison measured / model, using ”virtual `rtrace` gonio-photometer” :

- function file model ”monodir”
- XML model

demo building

- sun-path animation of comparison XML versus function
  slight ”jumps” in incident interpolation cause practical problem at client
  ($N = 7, t = 0.9$)



Case Study 1, rolled stainless steel roof sheet

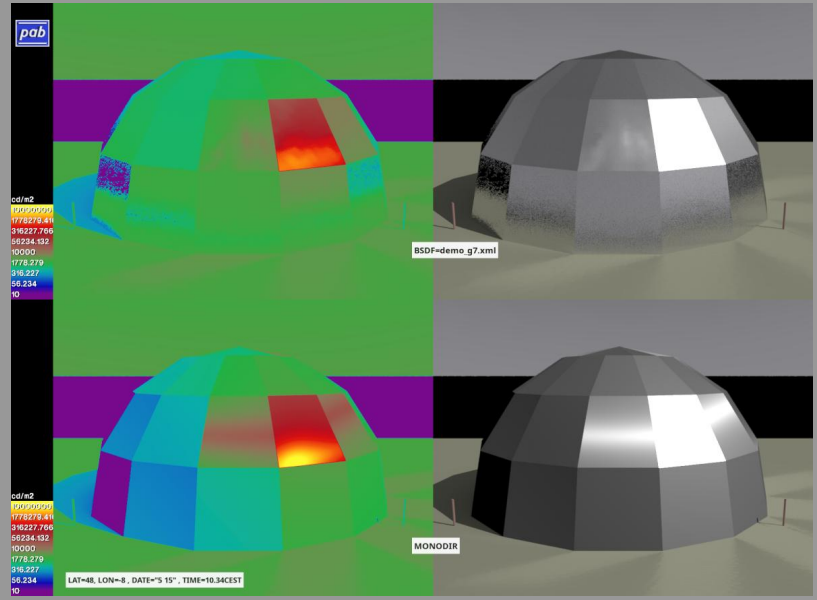
- photo
- measured data

comparison measured / model, using ”virtual `rtrace` gonio-photometer” :

- function file model ”monodir”
- XML model

demo building

- sun-path animation of comparison XML versus function
  slight ”jumps” in incident interpolation cause practical problem at client
  ($N = 7, t = 0.9$)



Case Study 1, rolled stainless steel roof sheet

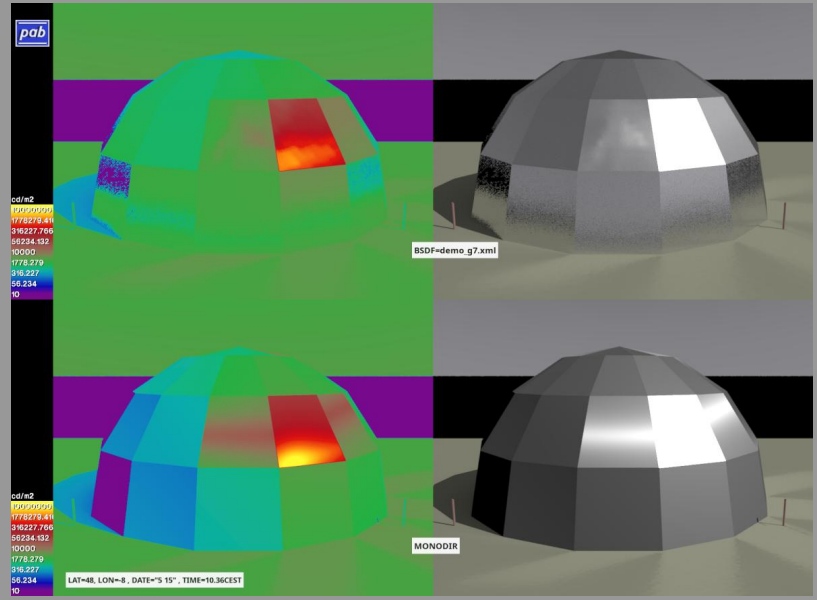
- photo
- measured data

comparison measured / model, using ”virtual `rtrace` gonio-photometer” :

- function file model ”monodir”
- XML model

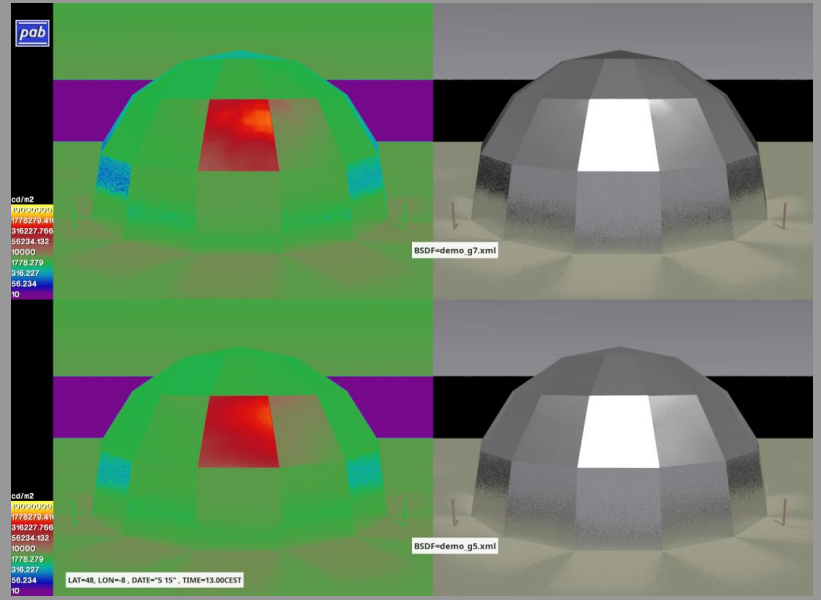
demo building

- sun-path animation of comparison XML versus function
  slight ”jumps” in incident interpolation cause practical problem at client
  ($N = 7, t = 0.9$)



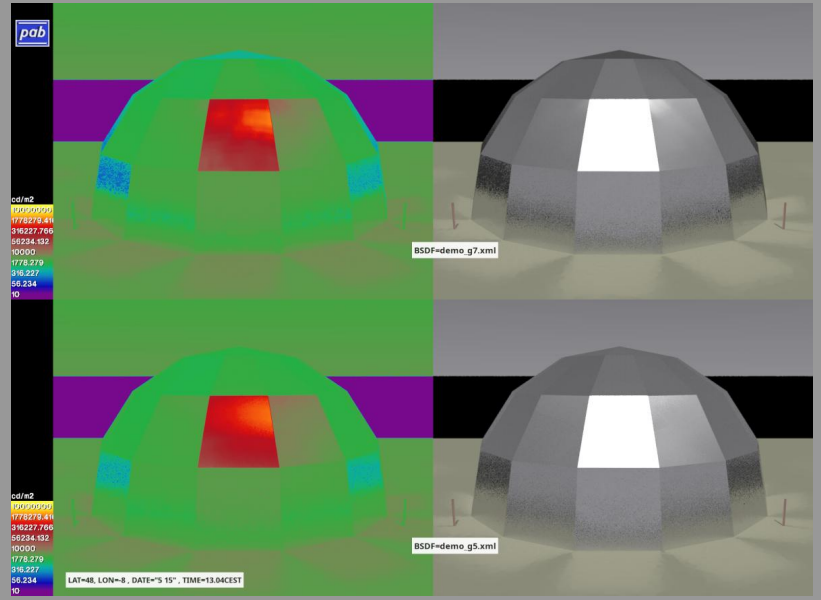
Case Study 1, parameter variation **bsdf2ttree**

- **bsdf2ttree** -g N
  $N = [5, 6, 7]$: only 7 yielded lowest artefacts for $\vec{x}_{in}$
  variation with $\vec{x}_{out}$ smooth for $N = 5$, but ”jumpy” for $\vec{x}_{in}$



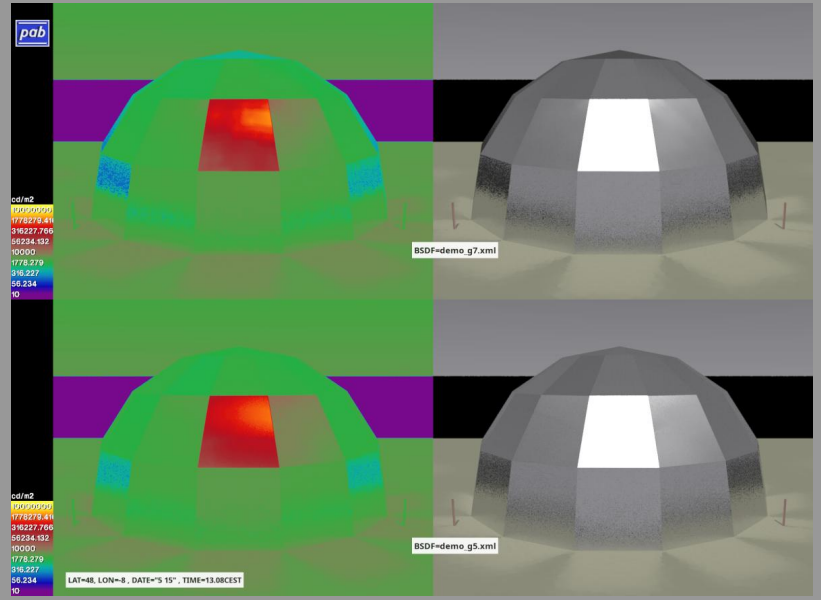
Case Study 1, parameter variation **bsdf2ttree**

- **bsdf2ttree** -g N
  $N = [5, 6, 7]$: only 7 yielded lowest artefacts for $\vec{x}_{in}$
  variation with $\vec{x}_{out}$ smooth for $N = 5$, but ”jumpy” for $\vec{x}_{in}$



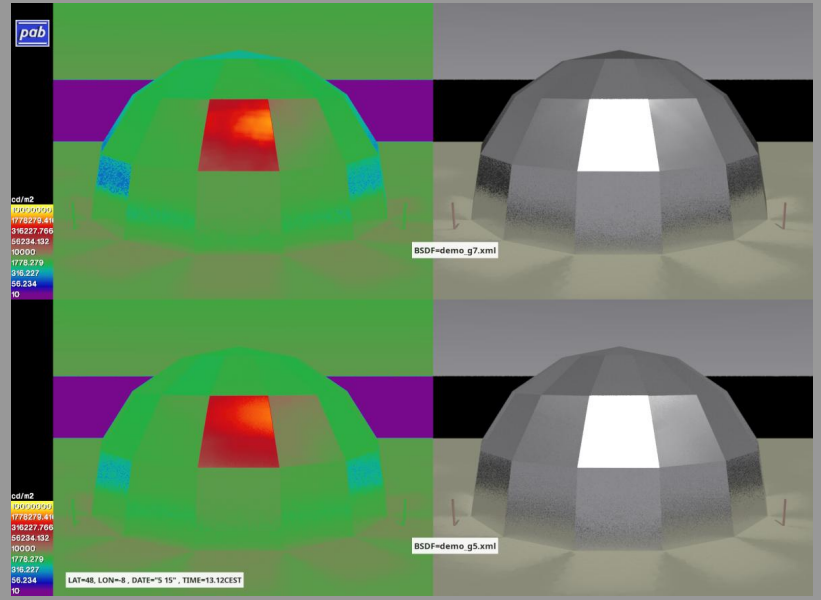
Case Study 1, parameter variation **bsdf2ttree**

- **bsdf2ttree** -g N
  $N = [5, 6, 7]$: only 7 yielded lowest artefacts for $\vec{x}_{in}$
  variation with $\vec{x}_{out}$ smooth for $N = 5$, but ”jumpy” for $\vec{x}_{in}$



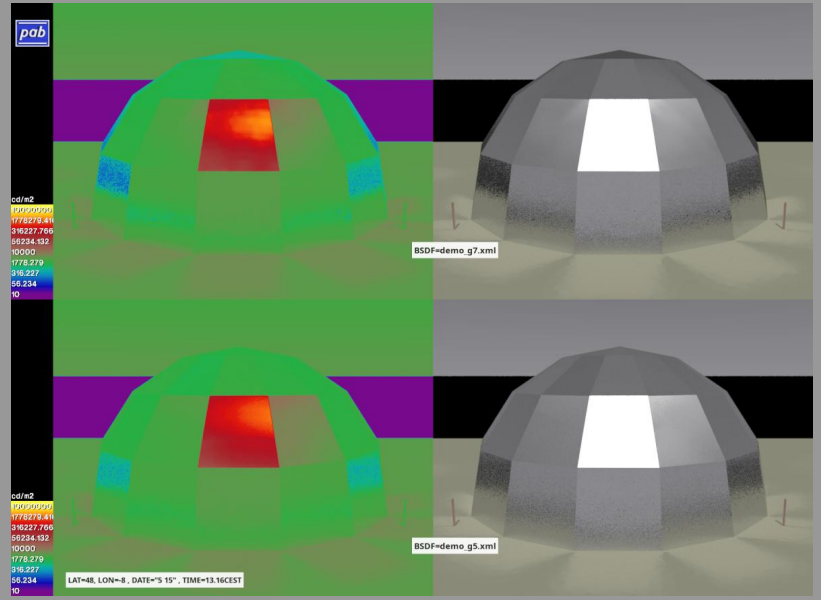
Case Study 1, parameter variation **bsdf2ttree**

- **bsdf2ttree** -g N
  $N = [5, 6, 7]$: only 7 yielded lowest artefacts for $\vec{x}_{in}$
  variation with $\vec{x}_{out}$ smooth for $N = 5$, but ”jumpy” for $\vec{x}_{in}$



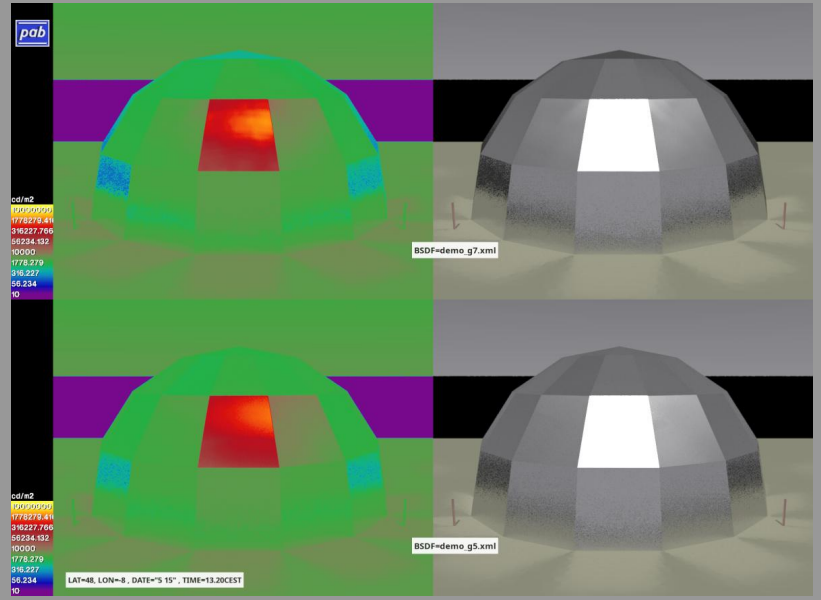
Case Study 1, parameter variation **bsdf2ttree**

- **bsdf2ttree** -g N
  $N = [5, 6, 7]$: only 7 yielded lowest artefacts for $\vec{x}_{in}$
  variation with $\vec{x}_{out}$ smooth for $N = 5$, but ”jumpy” for $\vec{x}_{in}$



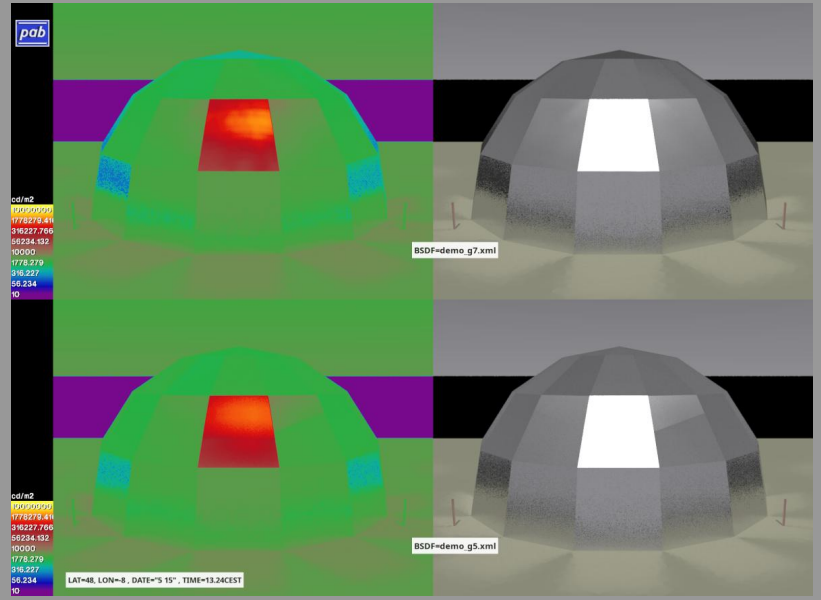
Case Study 1, parameter variation **bsdf2ttree**

- **bsdf2ttree** -g N
  $N = [5, 6, 7]$: only 7 yielded lowest artefacts for $\vec{x}_{in}$
  variation with $\vec{x}_{out}$ smooth for $N = 5$, but ”jumpy” for $\vec{x}_{in}$



Case Study 1, parameter variation **bsdf2ttree**

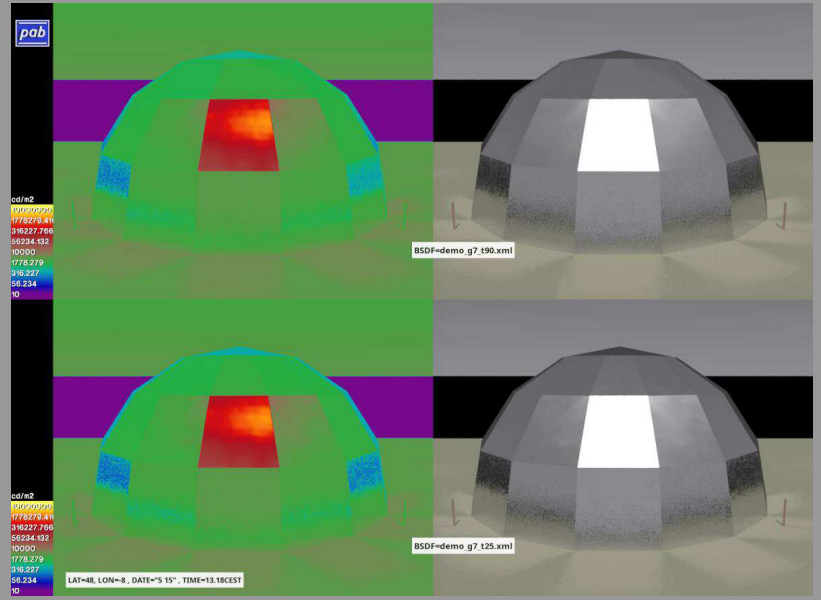
- **bsdf2ttree** -g N
  $N = [5, 6, 7]$: only 7 yielded lowest artefacts for $\vec{x}_{in}$
  variation with $\vec{x}_{out}$ smooth for $N = 5$, but "jumpy" for $\vec{x}_{in}$



Case Study 1, parameter variation **bsdf2ttree**

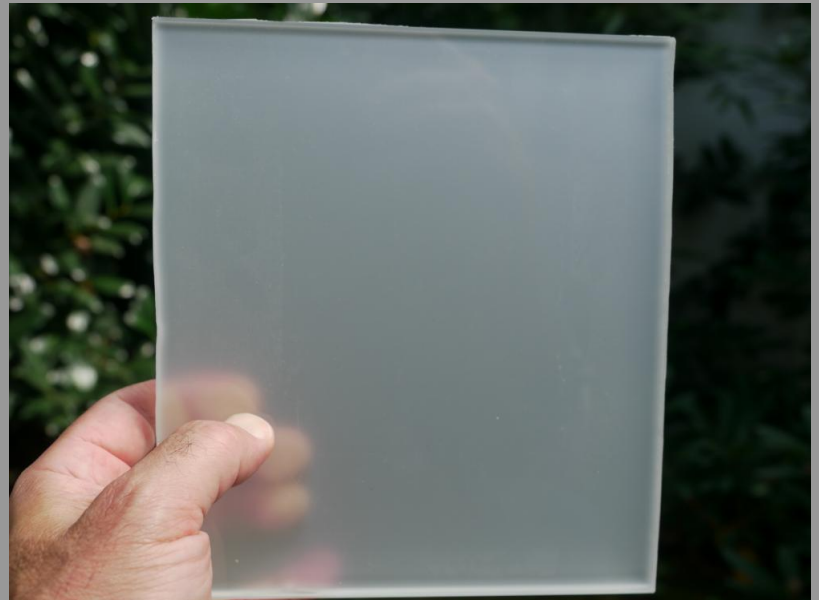
- **bsdf2ttree** -g N
  $N = [5, 6, 7]$: only 7 yielded lowest artefacts for $\vec{x}_{in}$
  variation with $\vec{x}_{out}$ smooth for $N = 5$, but "jumpy" for $\vec{x}_{in}$

- **bsdf2ttree** -t pctcull
  practically zero difference in results for pctcull=$[0.9, 0.25]$
  but larger file-size and longer rendering times with 0.25



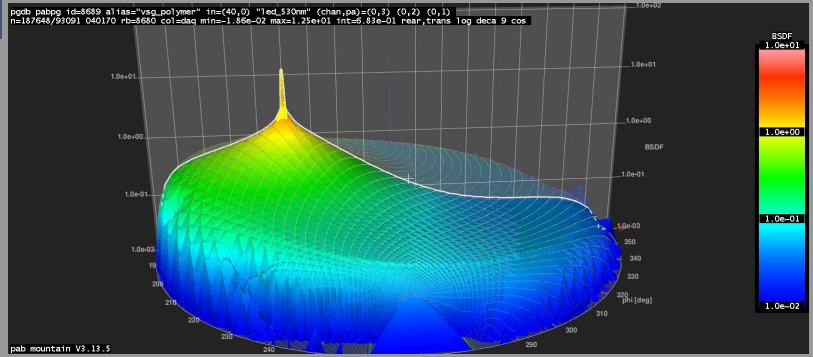
Case Study 2, laminated sheet glass, translucent

- photo: looks inconspicuous at first, but has an unscattered transmission peak



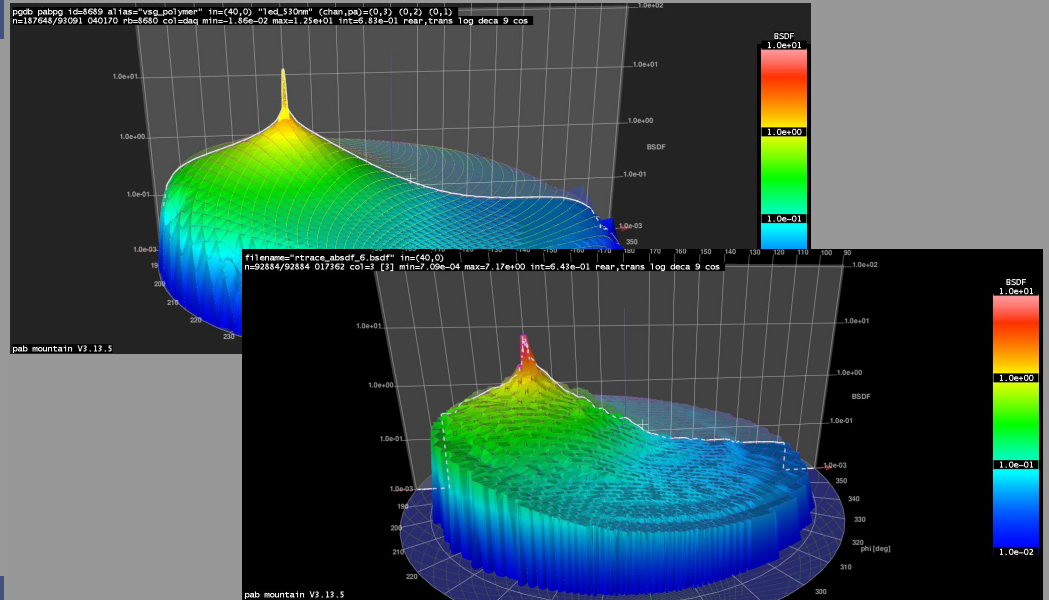
Case Study 2, laminated sheet glass, translucent

- photo: looks inconspicuous at first, but has an unscattered transmission peak
- measured BSDF data



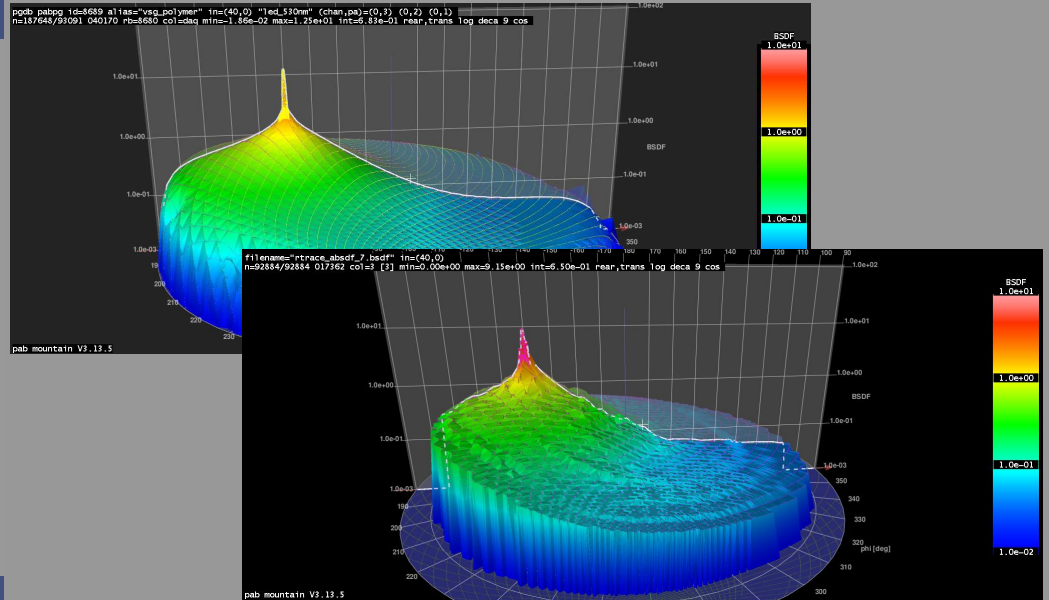
Case Study 2, laminated sheet glass, translucent

- photo: looks inconspicuous at first, but has an unscattered transmission peak
- measured BSDF data
- **aBSDF** XML g=6



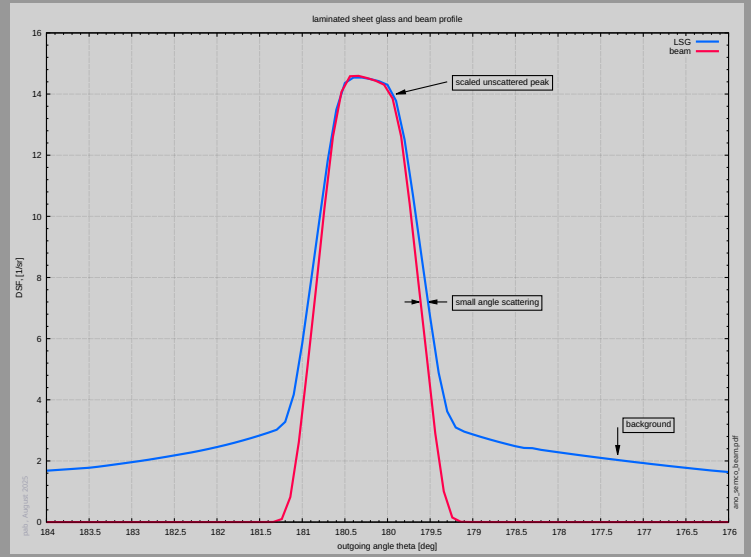
Case Study 2, laminated sheet glass, translucent

- photo: looks inconspicuous at first, but has an unscattered transmission peak
- measured BSDF data
- **aBSDF** XML g=6
- **aBSDF** XML g=8



Case Study 2, laminated sheet glass, translucent

- photo: looks inconspicuous at first, but has an unscattered transmission peak
- measured BSDF data
- **aBSDF** XML g=6
- **aBSDF** XML g=8
- small-angle scattering



Case Study 2, laminated sheet glass, translucent

- photo: looks inconspicuous at first, but has an unscattered transmission peak
- measured BSDF data
- **aBSDF** XML g=6
- **aBSDF** XML g=8
- small-angle scattering
- suggestion:

aBSDF suggestions:

- identify peak by instrument signature, PG2 ”beam measurement”, which is available anyway with any PG2 measurement.
- add peak-extraction for reflective side
- asymmetric resolution $\vec{x}_{in}$, $\vec{x}_{out}$?

Summary

Short Summary of RADIANCE material modelling:

1. Scene language & code structure have been stable for a *very* long time.
   good side: Backward compatible, which is useful when building geometry libraries.
   bad side: scleronomous.

Summary

Short Summary of RADIANCE material modelling:

1. Scene language & code structure have been stable for a *very* long time.
   good side: Backward compatible, which is useful when building geometry libraries.
   bad side: scleronomous.

2. only 2 built-in BSDF models; **plastic** (Ward Gaussian), **ashik2**,
   compared to numerous in general Computer-Graphics,
   no modular plug-in concept

Summary

Short Summary of RADIANCE material modelling:

1. Scene language & code structure have been stable for a *very* long time.
   good side: Backward compatible, which is useful when building geometry libraries.
   bad side: scleronomous.

2. only 2 built-in BSDF models; **plastic** (Ward Gaussian), **ashik2**,
   compared to numerous in general Computer-Graphics,
   no modular plug-in concept

3. has always featured powerful, flexible user function files, way ahead of its time.
   Similar to the newer concept of a ”shader language”

Summary

Short Summary of RADIANCE material modelling:

1. Scene language & code structure have been stable for a *very* long time.
   good side: Backward compatible, which is useful when building geometry libraries.
   bad side: scleronomous.

2. only 2 built-in BSDF models; **plastic** (Ward Gaussian), **ashik2**,
   compared to numerous in general Computer-Graphics,
   no modular plug-in concept

3. has always featured powerful, flexible user function files, way ahead of its time.
   Similar to the newer concept of a ”shader language”

4. XML/Tensortree modelling of measured BSDF works, with some things left to do

Suggestions

What I'd recommend for future developments:

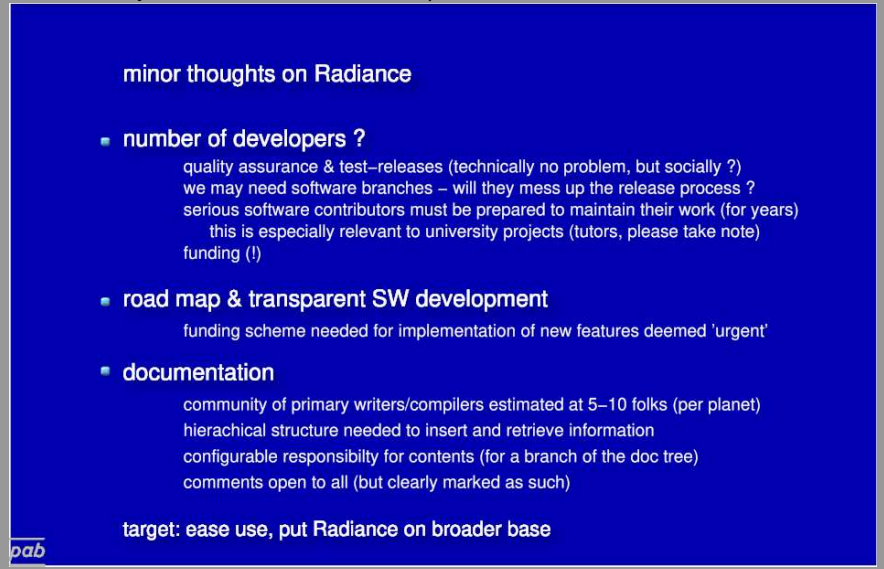
1. The established tool-chain of RADIANCE must be kept:
   Modular, commandline-line oriented, tools for specific tasks.
   Features like **rtrace** have been extremely useful in practice.
2. Better standard materials that fit larger number of BSDF data, with 3-5 parameters.
   especially for "translucent" types, e.g. variants of fritted glass
3. Study on how surface models influence the final result:
   irradiance levels, glare prediction, annual irradiance levels
4. a more cooperative RADIANCE development ?
5. split release date (e.g. January) and workshop date (August),
   giving others a chance to submit work based on the *current* version.

Suggestions

What I'd recommend for future developments:

1. The established tool-chain of RADIANCE must be kept:
   Modular, commandline-line oriented, tools for specific tasks.
   Features like **rtrace** have been extremely useful in practice.
2. Better standard materials that fit larger number of BSDF data, with 3-5 parameters.
   especially for "translucent" types, e.g. variants of fritted glass
3. Study on how surface models influence the final result:
   irradiance levels, glare prediction, annual irradiance levels
4. a more cooperative RADIANCE development ?
5. split release date (e.g. January) and workshop date (August),
   giving others a chance to submit work based on the *current* version.

slide from my talk at RADIANCE workshop 2006:



... thanks

- A very big *Thank You !* to Greg for his year-round support and bug-fixes.

... thanks

- A very big *Thank You !* to Greg for his year-round support and bug-fixes.
- Thanks to Jan for the opportunity for this little summary on materials.

... thanks

- A very big *Thank You !* to Greg for his year-round support and bug-fixes.
- Thanks to Jan for the opportunity for this little summary on materials.

- Thank you for your interest and attention !

... thanks

- A very big *Thank You !* to Greg for his year-round support and bug-fixes.
- Thanks to Jan for the opportunity for this little summary on materials.

- Thank you for your interest and attention !

Happy rendering !

compiled using LaTeX *beamer* class