basket_weave.cal

modeling the view through rollershades (a work in progress)

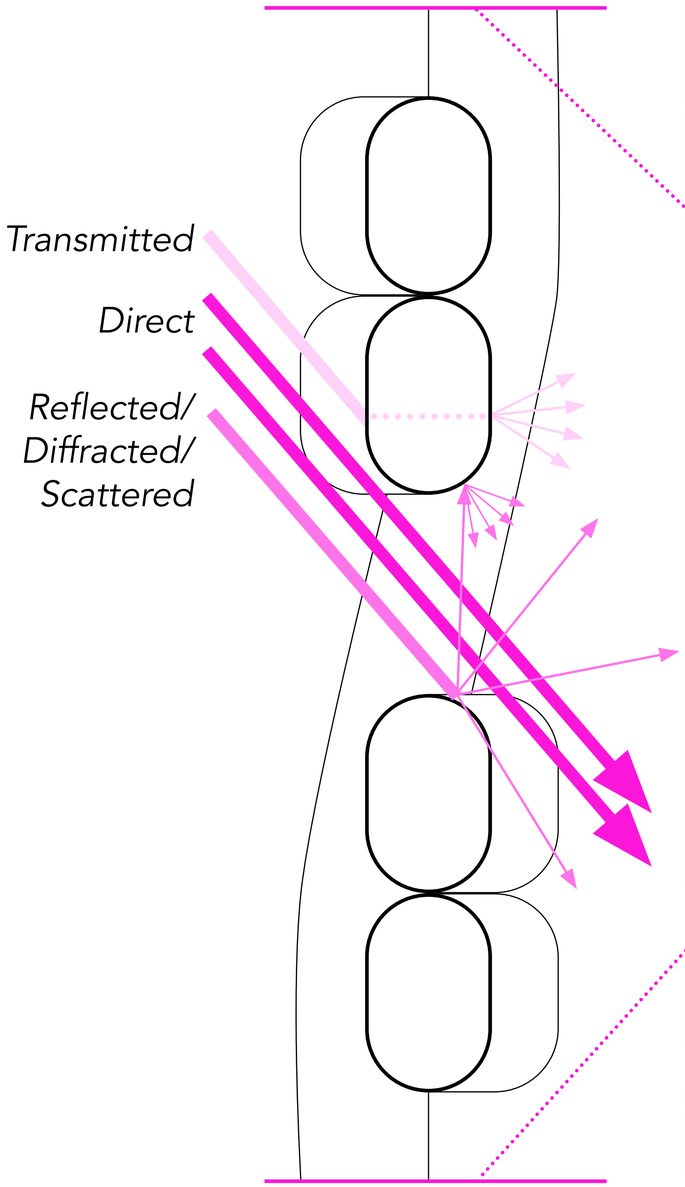
Stephen Wasilewski

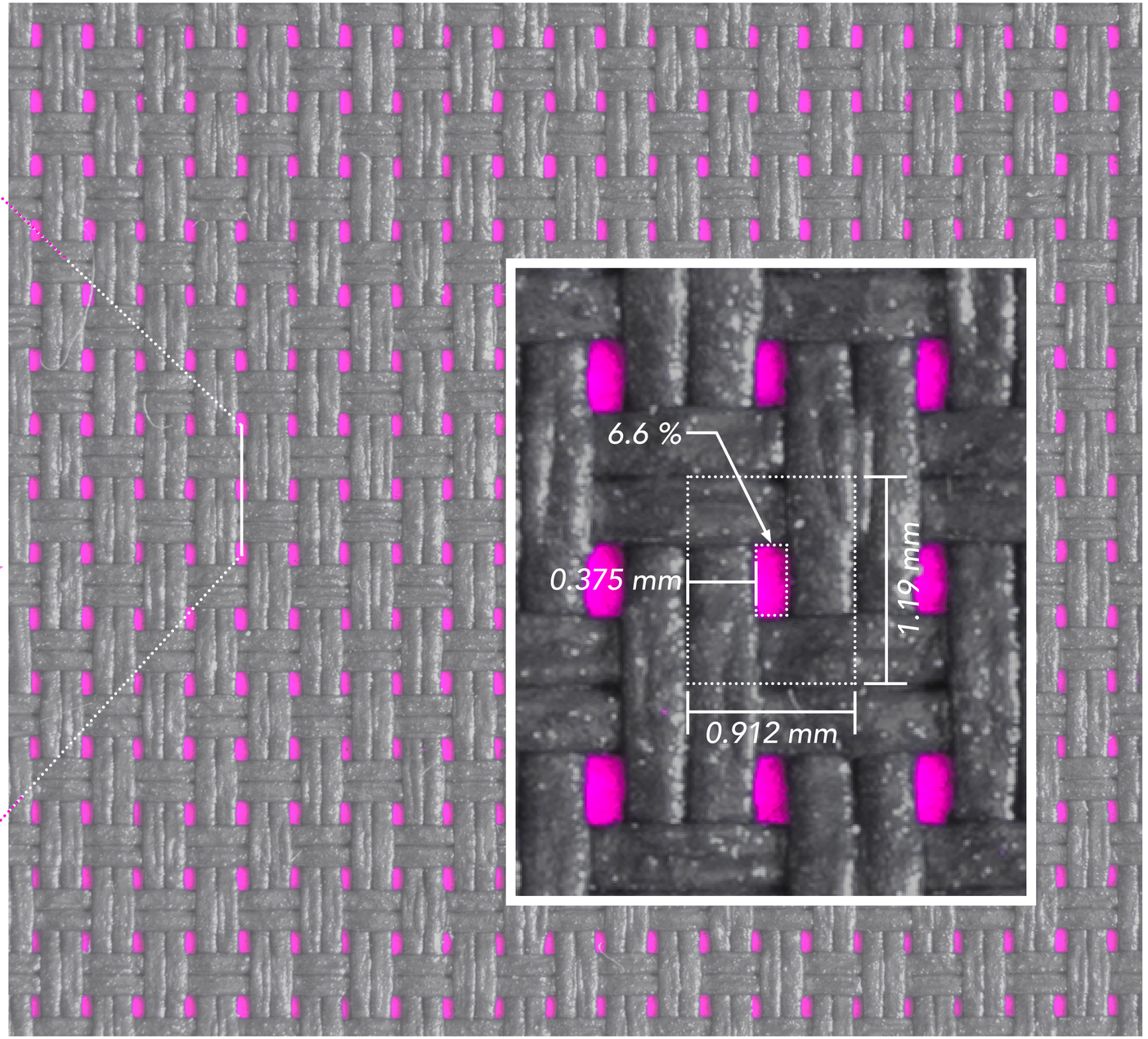
08.27.2024

LIPID
Laboratory of Integrated Performance in Design

Fabric Geometry

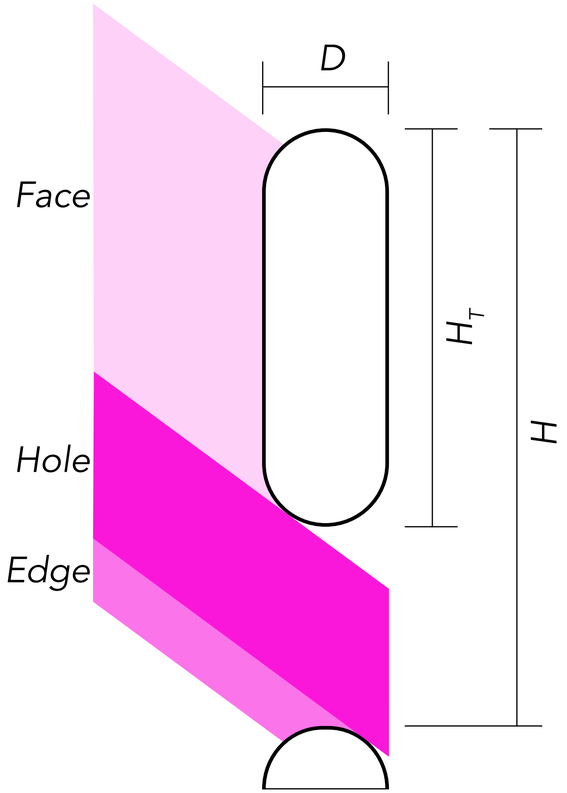
Stephen Wasilewski

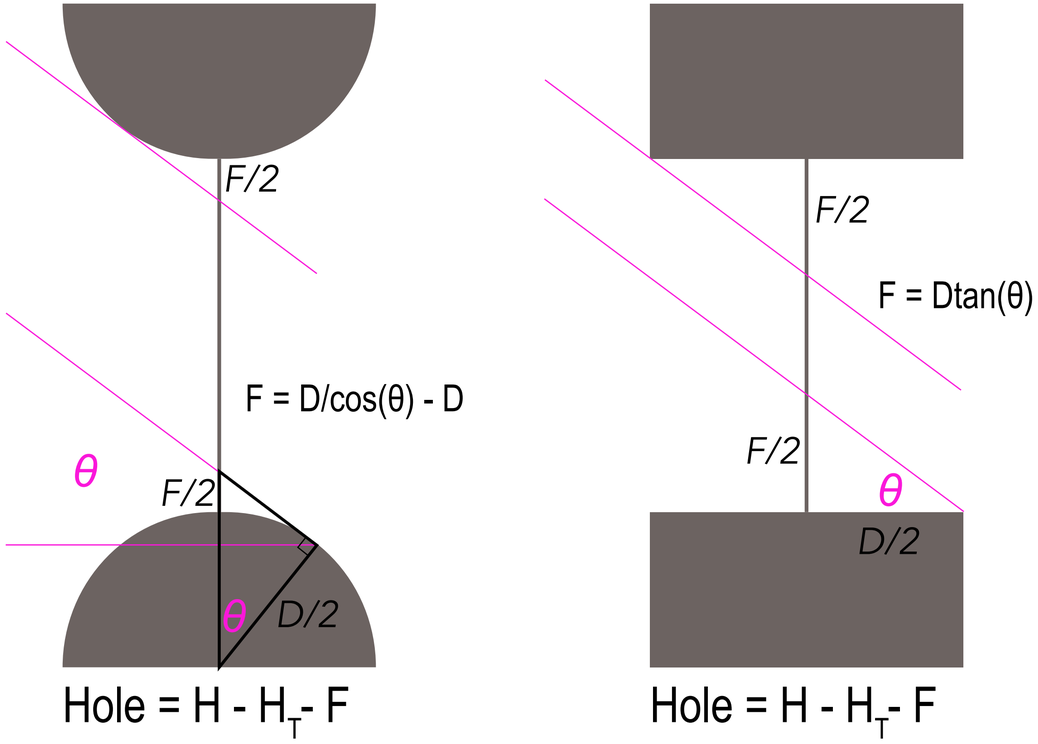


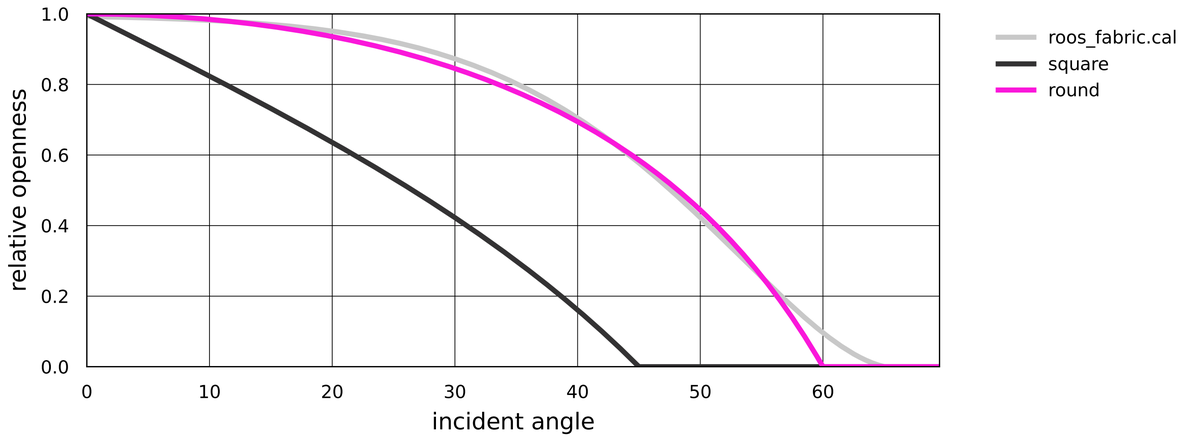


Fabric Geometry

Stephen Wasilewski





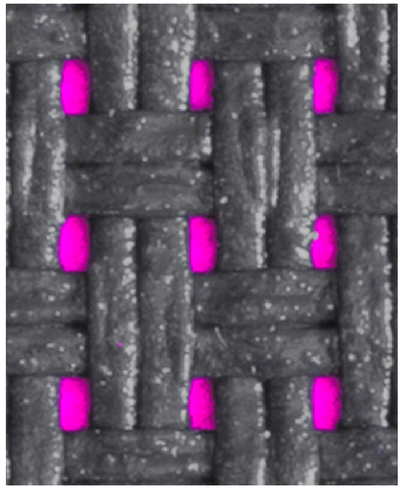


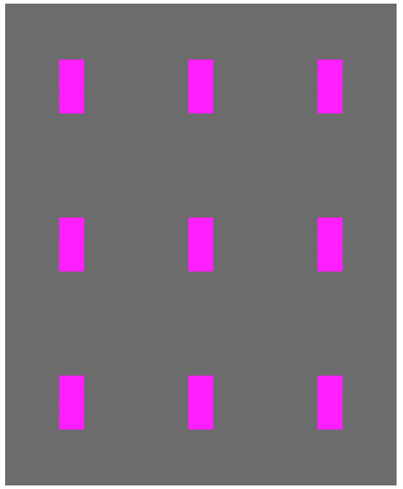
Locate and size holes

```
A1: scaleU (width, in model units)
A2: scaleV (height, in model units)
A3: openness (0 < of < 0.25)
A4: thread ratio (width / height)
A5: depth ratio or, if > 10, cutoff angle (deg)
A6: hole jitter rate (0–1)
```

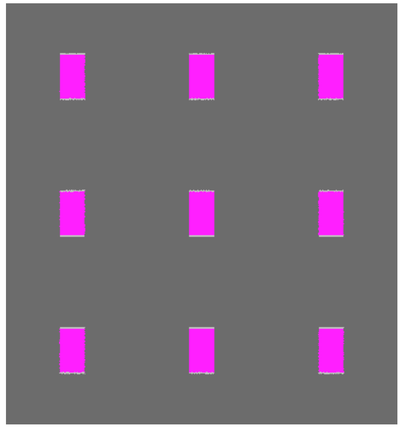
```
void mixfunc shade
4 void thread holes basket_weave.cal
0
6 0.0009 0.0011 0.066 1 0.44 0
```

```
fillx = (threadx+xfor)/A1;
filly = (thready+yfor)/A2;
hU = abs(frac((U+wU)/A1));
hV = abs(frac((V+wV)/A2));
centU = inside(fillx*.5, hU, 1-fillx*.5);
centV = inside(filly*.5, hV, 1-filly*.5);
holes = if(and(centU, centV),1,0);

yfor = if(1e-6-aDz, A2-thready,
          min(A2-thready,
              depthy/cthetay-depthy));
```

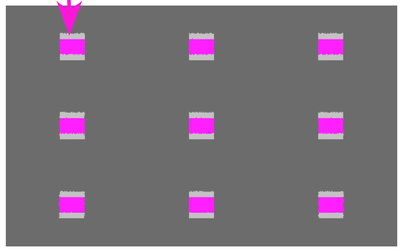




Perpendicular



30 degrees



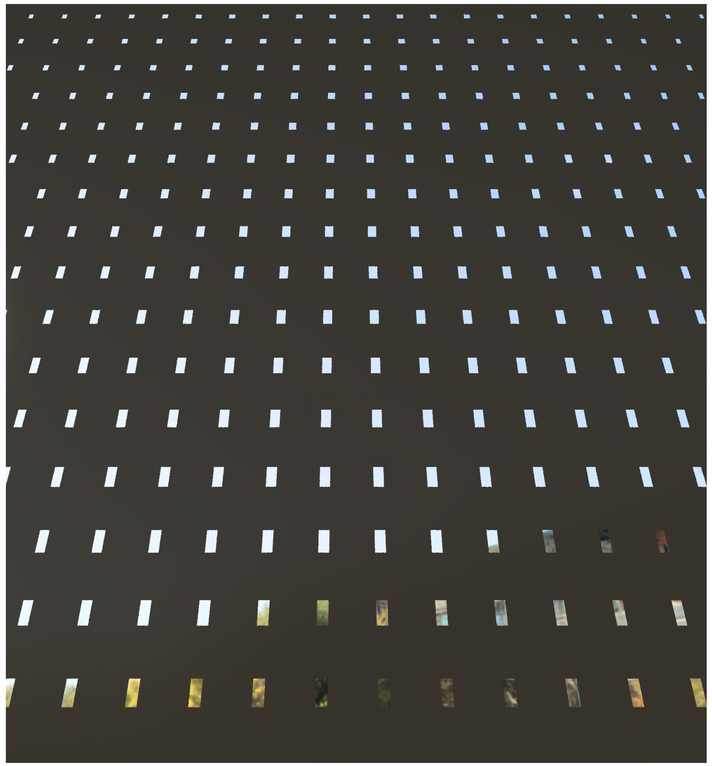
60 degrees

Stephen Wasilewski

Anti-Aliasing: Jitter or Homogenize

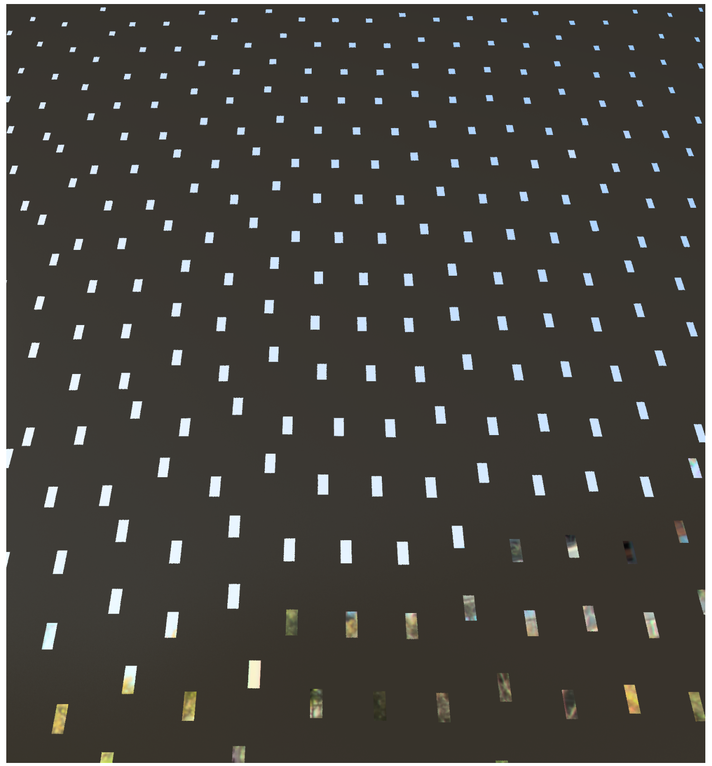
Stephen Wasilewski

```
hU = abs(frac((U+wU)/A1));
```



```
void mixfunc shade
4 void thread holes bw.cal
0
6 0.0009 0.0011 0.066 1 69.0 0
```

```
wU = A6 * threadx *
     (rand(floor(V/A2)) - 0.5);
```



```
void mixfunc shade
4 void thread holes bw.cal
0
6 0.0009 0.0011 0.066 1 69.0 1
```

```
fillx = (threadx + xfor) / A1;
fabric_m = (1 - fillx) *
           (1 - filly);
```

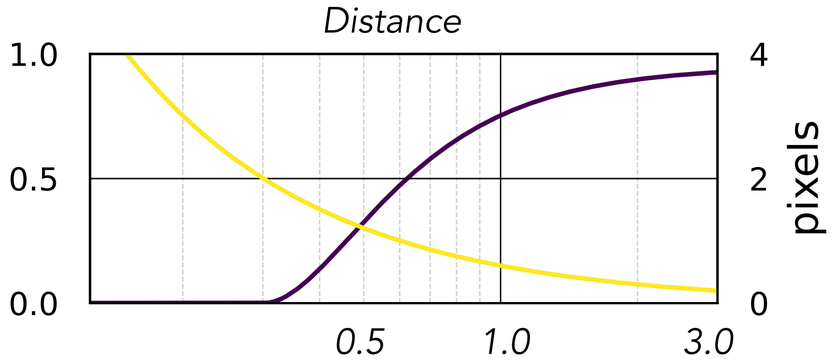
```
void mixfunc shade
4 void thread fabric_m bw.cal
0
6 0.0009 0.0011 0.066 1 69.0 1
```

Matching Acuity / Resolution

Stephen Wasilewski

fabric_m

```
{120 pixels/degree 20/20 vision}
acuity = if(arg(0) - 6, 2*tan(PI/180/A7), 1.75 /
6000);

acuity_falloff(x): sq(cos(min(x,2)*PI/4));
grid2acuity = (min(A1-threadx,A2-thready)/ T)
                acuity;
blurfactor = 0.95 * acuity_falloff(grid2acuity);
fabric = blurfactor * fabric_m +
         (1 - blurfactor) * holes;
```



holes

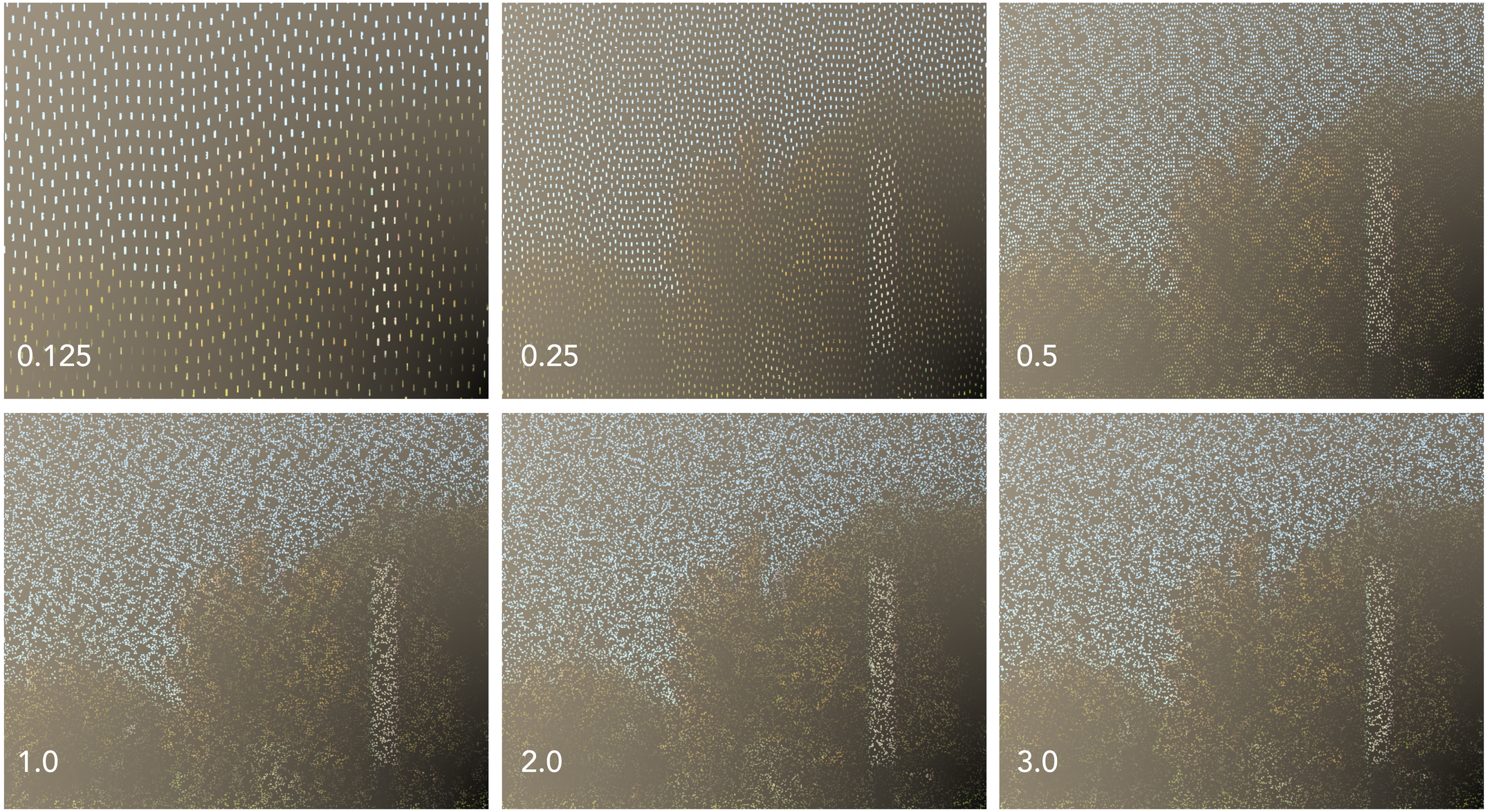
Hole width: (pixels)	4.8	2.4	1.2	0.6	0.3	0.2

fabric
(% fabric_m distance)

0% 0.125	0% 0.25	33% 0.5	75% 1.0	90% 2.0	93% 3.0

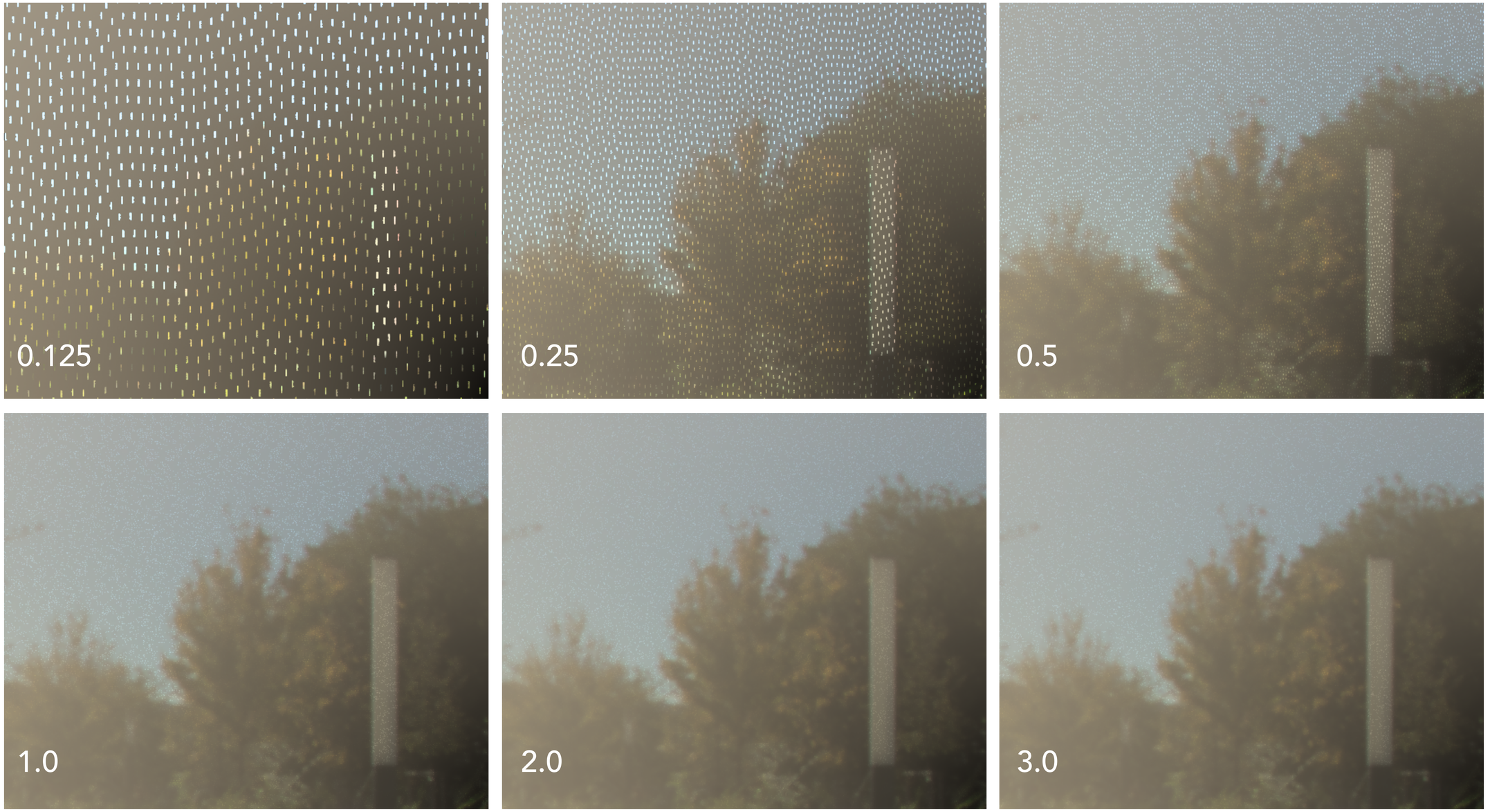
No acuity adjustment

Stephen Wasilewski



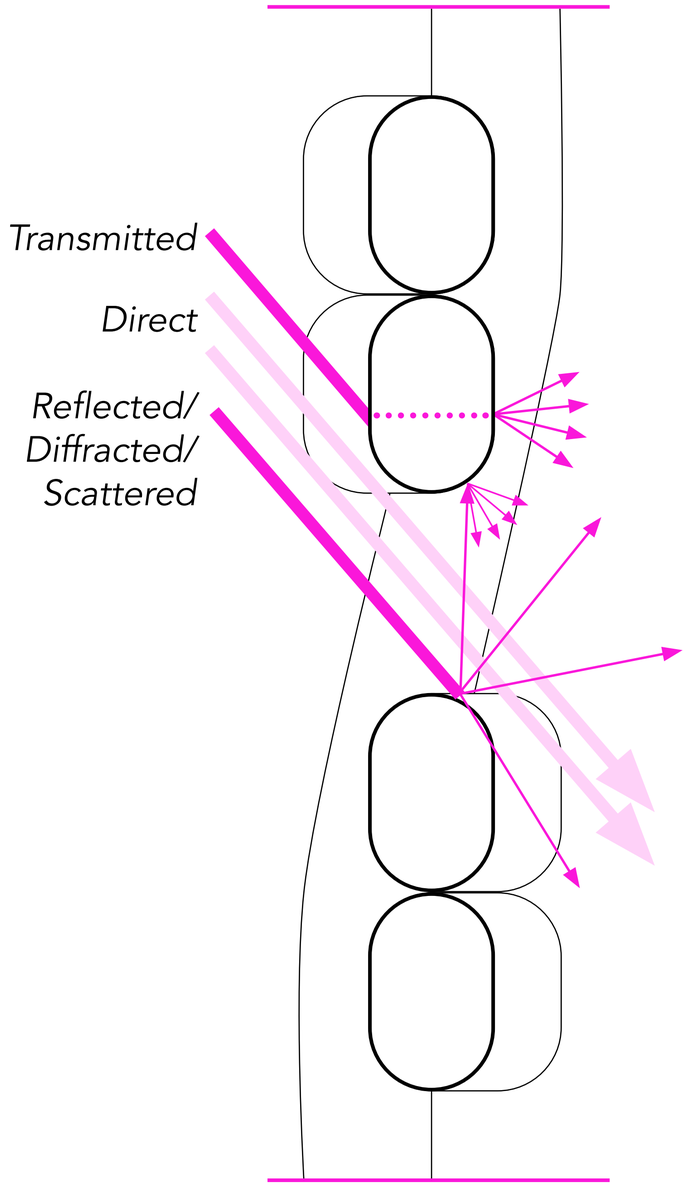
Matching Acuity / Resolution

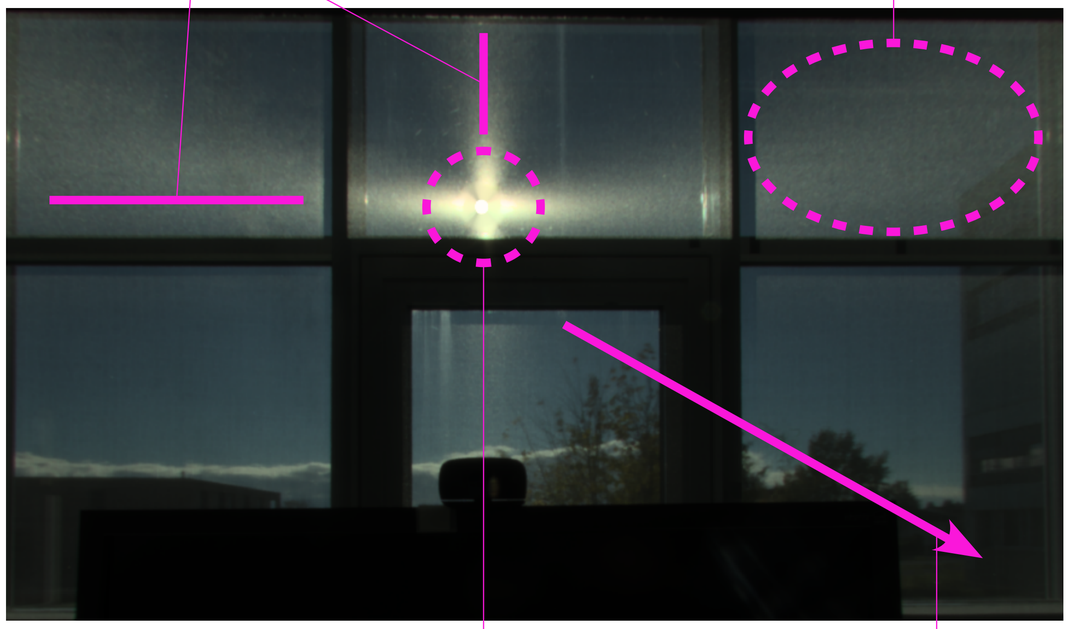
Stephen Wasilewski



Thread Material

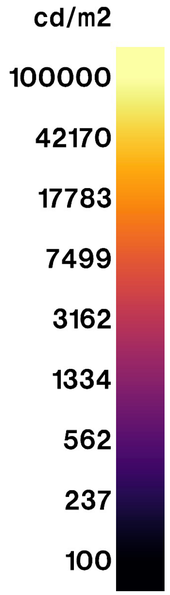
Stephen Wasilewski



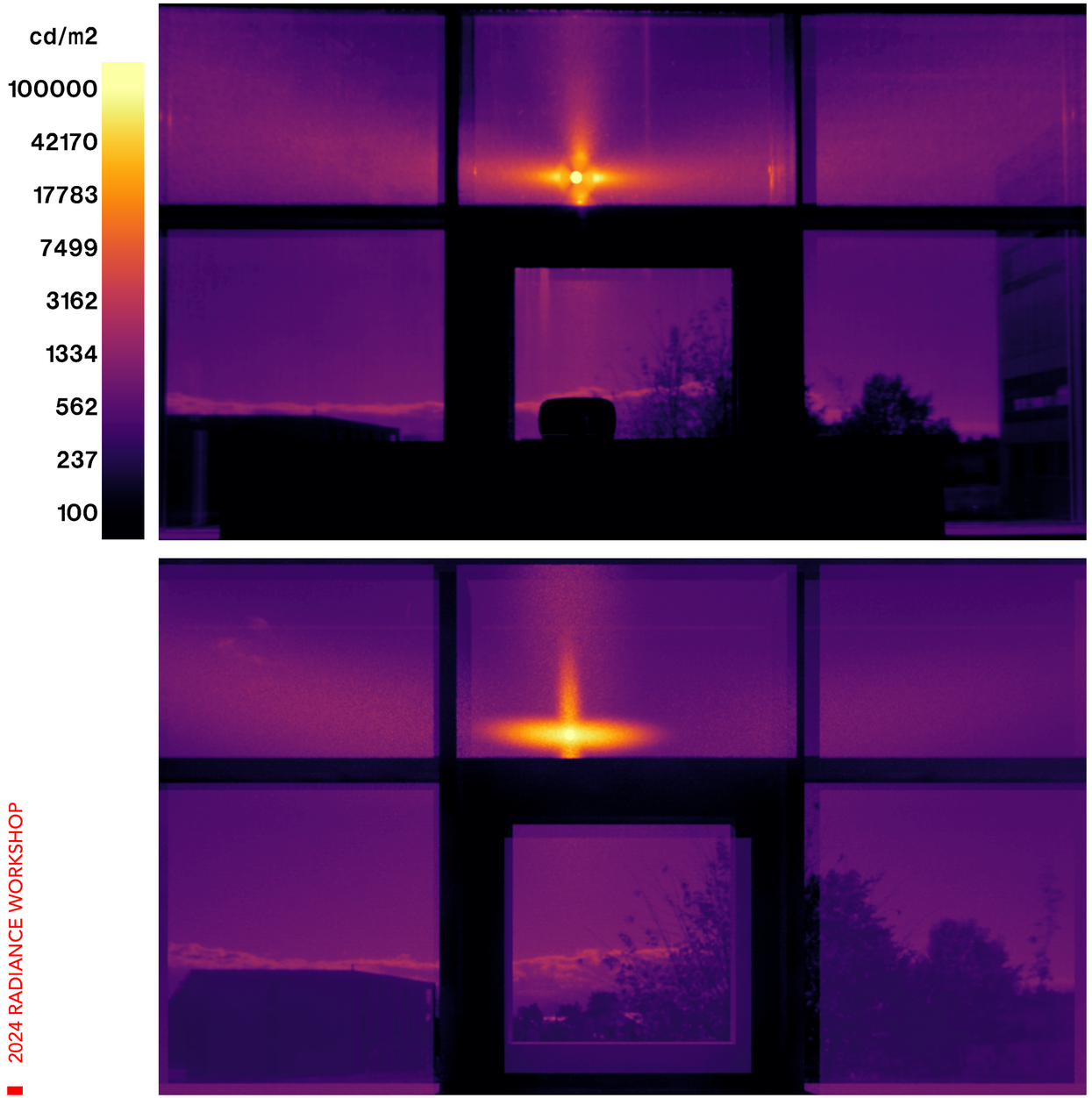


Anisotropic Scattering

Stephen Wasilewski



Anisotropic Scattering

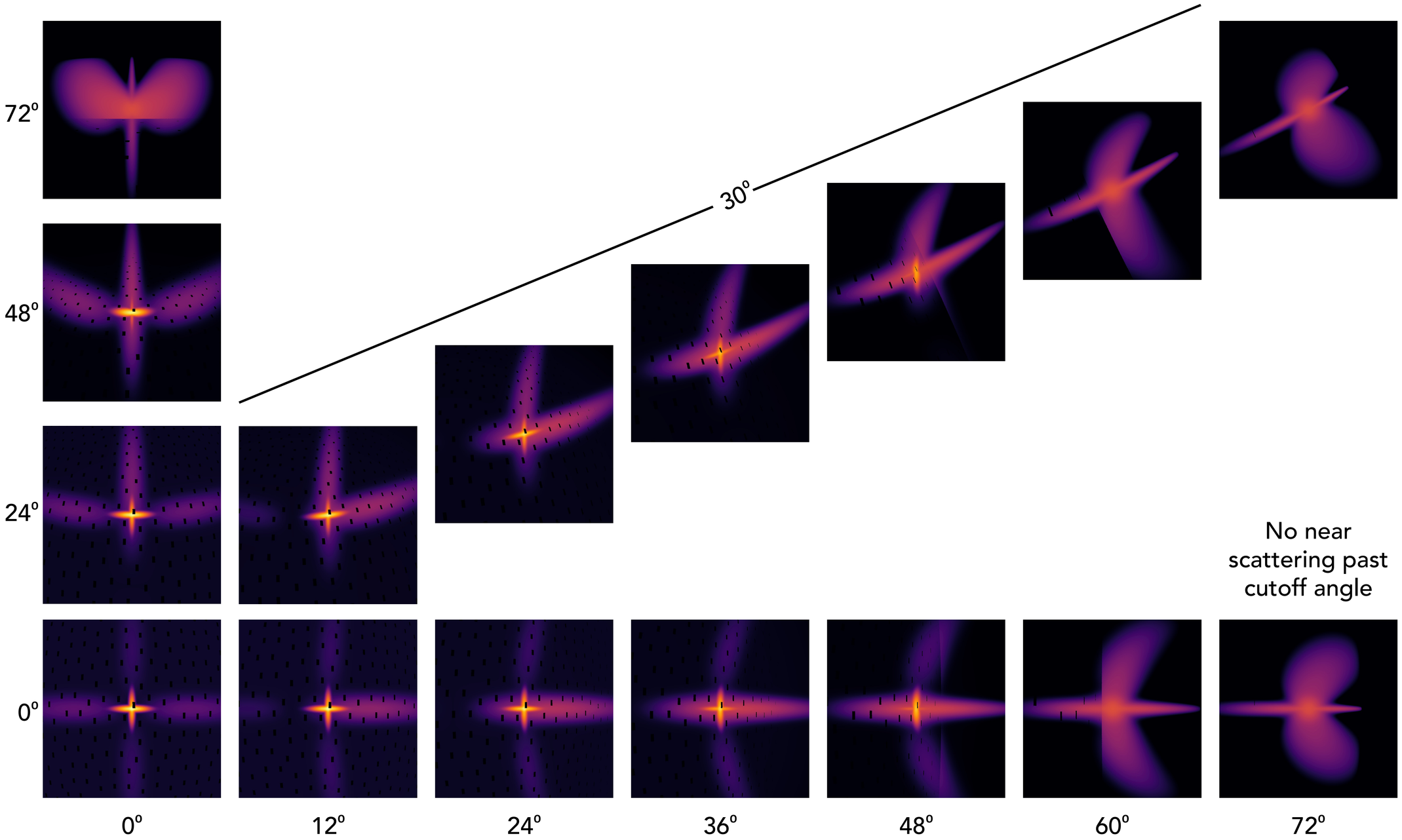



```
{threadx:thready}
{ratio of hole size, the longer side will catch more
highlights, this ratio is taken as constant}
t_orient = (A2-thready)/(A1+A2-threadx-thready);

{low-angle:high-angle scatter}
{blend between two different roughness levels}
scatterx = if(fabric_m,0.8*sq(cos(PI*angx/(2*cutoffx))), 0);
scattery = if(fabric_m,0.8*sq(cos(PI*angy/(2*cutoffy))), 0);

{a dither option that looks better}
scatterx_s = if(rand(V)-scatterx, 0, .5) + 0.5 * scatterx;
scattery_s = if(rand(U)-scattery, 0, .5) + 0.5 * scattery;
```

```
# mix near normal and grazing semi-specular
void mixfunc thread_v
4 thread_v1 thread_v2 scatterx_s basket_weave.cal
0
7 0.00088 0.00105 0.061 1 69.0 1 40.0

# mix vertical (thread_v) and horizontal (thread_h)
void mixfunc thread_t
4 thread_v thread_h t_orient basket_weave.cal
0
7 0.00088 0.00105 0.061 1 69.0 1 40.0
```


Stephen Wasilewski

Anisotropic Scattering



Stephen Wasilewski

Full Material


```
# TVIS=0.0549, RDIFF=0.15, TDIFF=0.0122
void trans2 thread_v1
4 tx ty tz basket_weave.cal
0
8 0.205 0.205 0.205 0.0 0.01 0.05 0.268 0.778

void trans2 thread_h1
4 tx ty tz basket_weave.cal
0
8 0.205 0.205 0.205 0.0 0.05 0.01 0.268 0.778
```

LOW ANGLE SCATTERING

```
# TVIS=0.0183, RDIFF=0.15, TDIFF=0.0122
void trans2 thread_v2
4 tx ty tz basket_weave.cal
0
8 0.168 0.168 0.168 0.0 0.06 0.35 0.109 0.333

void trans2 thread_h2
4 tx ty tz basket_weave.cal
0
8 0.168 0.168 0.168 0.0 0.35 0.06 0.109 0.333
```

HIGH ANGLE SCATTERING

```
# mix near normal and grazing semi-specular
void mixfunc thread_v
4 thread_v1 thread_v2 scatterx basket_weave.cal
0
7 0.00088 0.00105 0.061 1 69.0 1 40.0

void mixfunc thread_h
4 thread_h1 thread_h2 scattery basket_weave.cal
0
7 0.00088 0.00105 0.061 1 69.0 1 40.0
```

SCATTERING
(component incident angle)

```
# mix vertical (thread_v) and horizontal (thread_h)
void mixfunc thread_t
4 thread_v thread_h t_orient basket_weave.cal
0
7 0.00088 0.00105 0.061 1 69.0 1 40.0
```

WARP : WEFT
(hole aspect)

```
# non-transmissive fabric
void plastic thread_p
0
0
5 0.15 0.15 0.15 0 0
```

```
# fall off in thread transmission
void mixfunc thread
4 thread_t thread_p thread basket_weave.cal
0
0
```

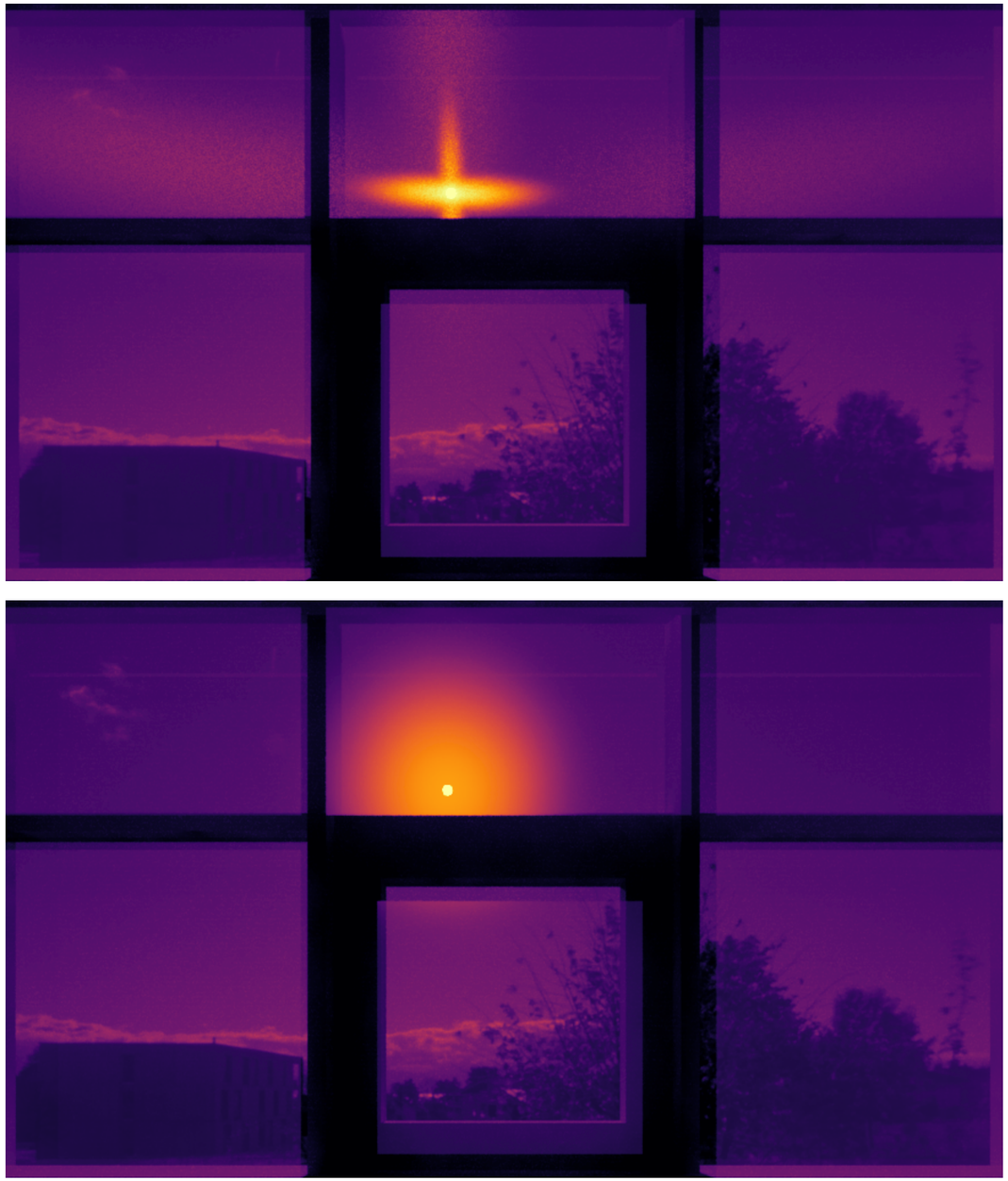
OPAQUE : TRANSPARENT
(incident angle)

```
# the final material
void mixfunc shade
4 void thread fabric basket_weave.cal
0
7 0.00088 0.00105 0.061 1 69.0 1 40.0
```

VOID : THREAD
(apparent hole size)

Stephen Wasilewski

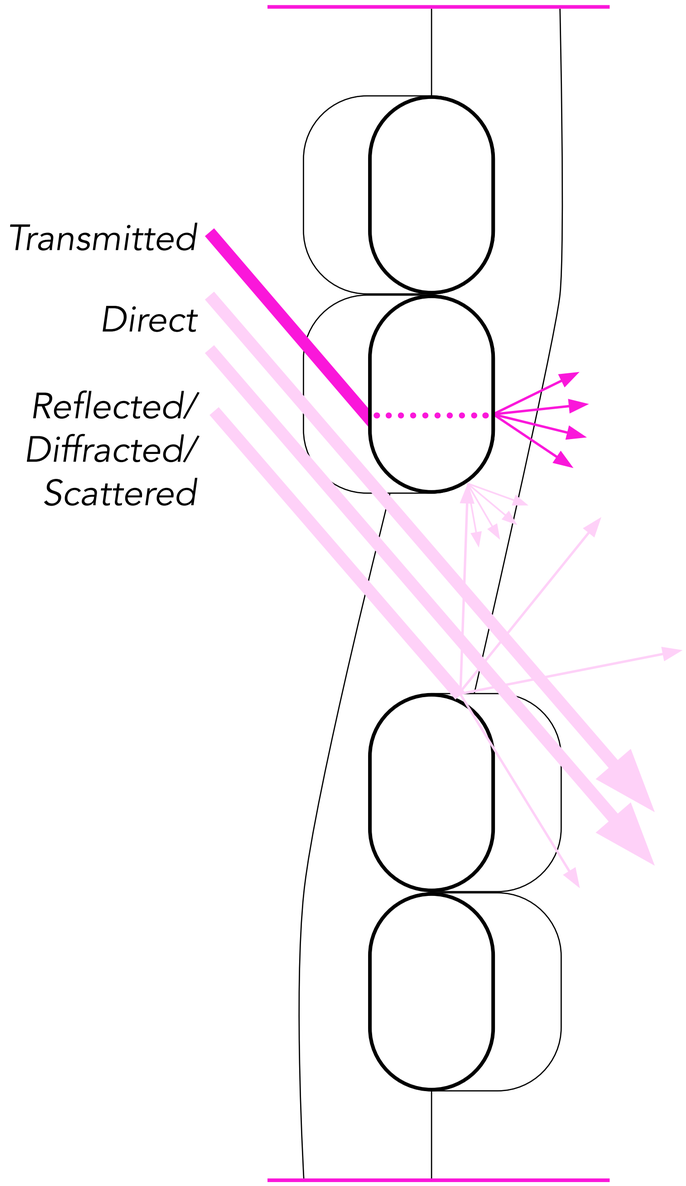
Simplified




```
# effective diff trans is: 0.054
# shade spec: Openness: 6.1% Diffuse Refl: 0.15 Cutoff: 69.0
Diffuse Trans.: 0.0122

# fabric transmission at normal incidence
# TVIS=0.0244, RDIFF=0.15, TDIFF=0.0122
void trans thread_t
0
0
7 0.174 0.174 0.174 0.0 0.08 0.14 0.5

# non-transmissive fabric
void plastic thread_p
0
0
5 0.15 0.15 0.15 0 0

# fall off in thread transmission (due to increased reflection
at grazing angles)
void mixfunc thread
4 thread_t thread_p thread basket_weave.cal
0
0

# the final material
void mixfunc shade
4 void thread fabric basket_weave.cal
0
7 0.00088 0.00105 0.061 1 69.0 1 40.0
```


Stephen Wasilewski

Diffuse Transmission and Material Builder

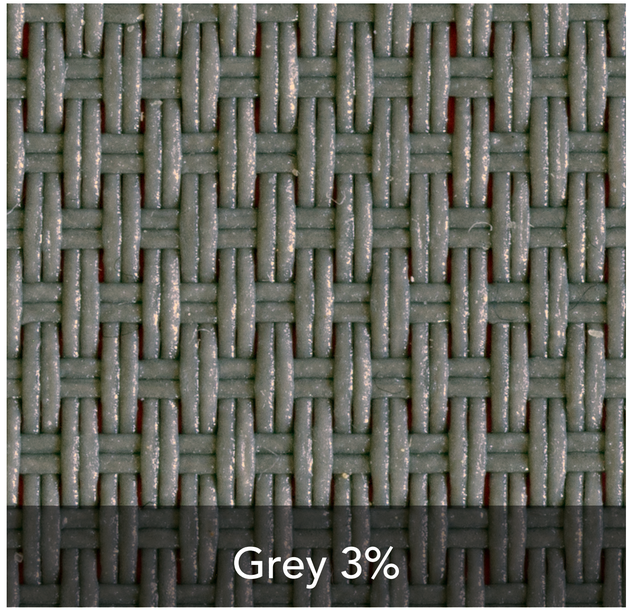


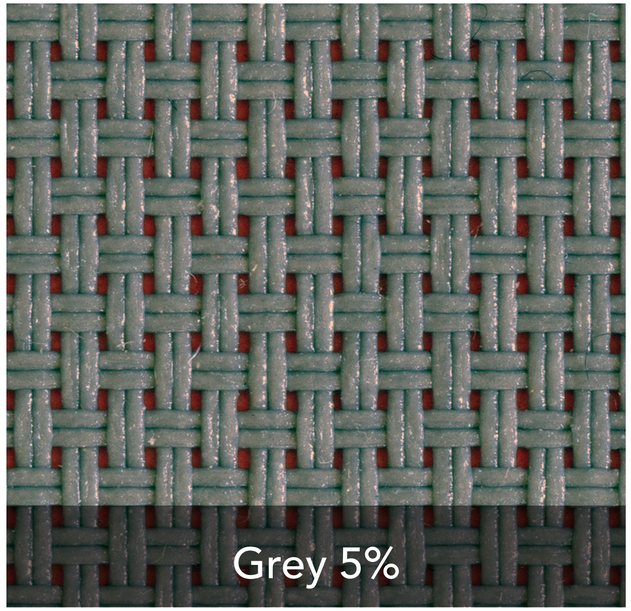

```
./shade_parameter.py --help
./shade_parameter.py basket_weave.cal > material.rad
```

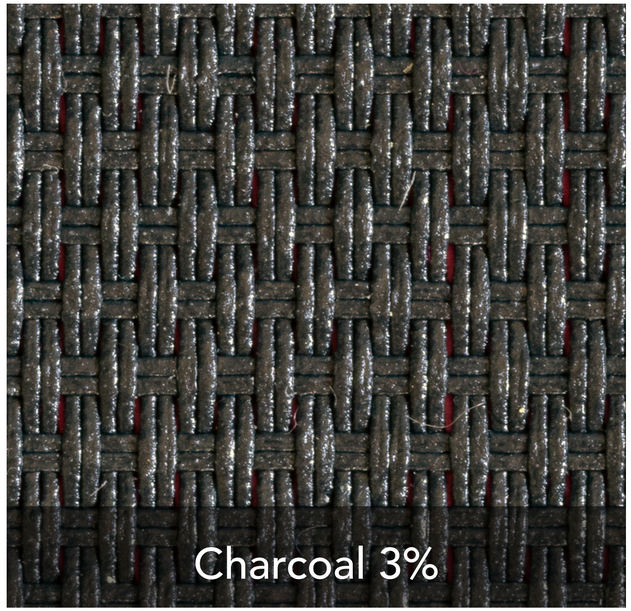
```
def run_shade_min(dt, cutoff, of, dr, ref, calfile="basket_weave.cal",
                  flarescale=1.0, anisotropic=False):
    """return deviation in diffuse transmission from target as a
function of
    additional diffuse transmission, cutoff angle, and openness factor

    parameters:
    dt: additional diffuse transmission
    cutoff: cutoff angle (degrees
    of: openness factor
    dr: diffuse reflectance (one value / grey)
    ref: target diffuse transmission
    calfile: path to basket_weave.cal
    flarescale: adjustment to semi-specular transmission intensity
    anisotropic: use directional scatter model
    """
    # make shade material and calculate diffuse transmission
    result = run_shade(calfile, dt, cutoff, of, dr, flarescale,
anisotropic,
                       "fabric_m")
    # log progress to stderr
    print("{:.04f}\t{:.04f}".format(dt, result/ref), file=sys.stderr)
    # return deviation
    return np.abs(result - ref)
```

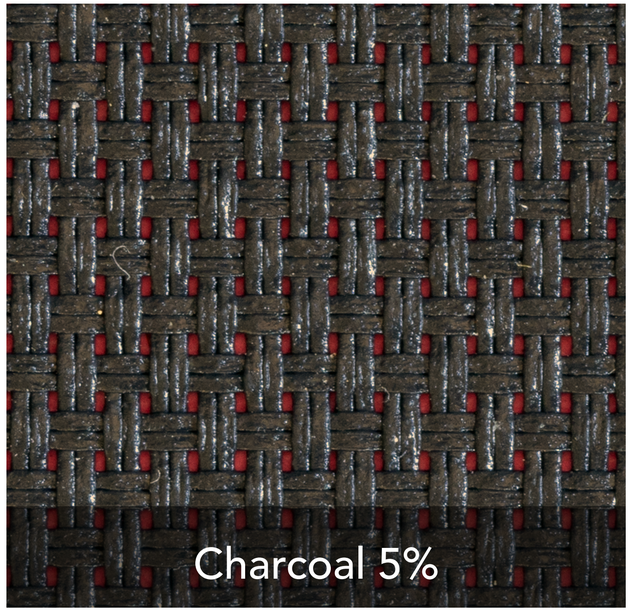
Stephen Wasilewski

Fabric Variations









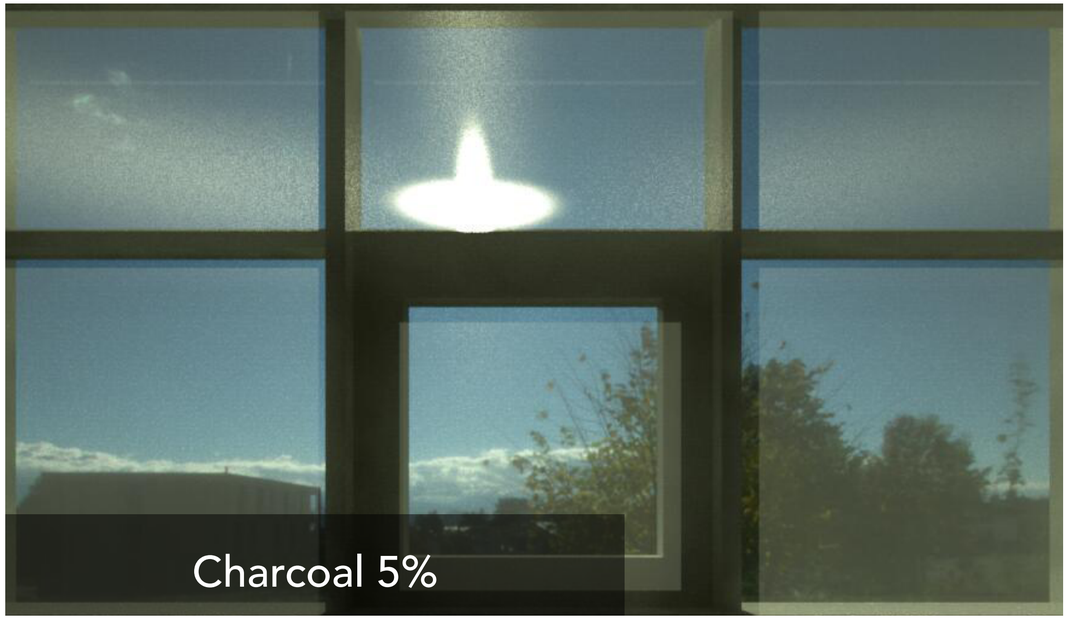

```
A1: scaleU (width, in model units)
A2: scaleV (height, in model units)
A3: openness (0 < of < 0.25)
A4: thread ratio (width / height)
A5: depth ratio or, if > 10, cutoff angle (deg)
A6: hole jitter rate (0-1)
A6: acuity (pixels per degree)

# diff. trans: 0.045
void mixfunc grey03
4 void thread fabric basket_weave.cal
0
7 0.0008 0.0012 0.021 1 69 1 40

# diff. trans: 0.08 (estimated)
void mixfunc grey05
4 void thread fabric basket_weave.cal
0
7 0.0008 0.0012 0.061 1 69 1 40

# diff. trans: 0.017
void mixfunc charcoal03
4 void thread fabric basket_weave.cal
0
7 0.0009 0.0011 0.021 1 69 1 40

# diff. trans: 0.053
void mixfunc charcoal05
4 void thread fabric basket_weave.cal
0
7 0.0009 0.0011 0.061 1 69 1 40
```

OF from: estimated from image (thread count and width)
transmission from: manufacturer data (mermet)
Cutoff angles from:
Karmann, C., Chinazzo, G., Schüler, A., Manwani, K., Wienold, J., Andersen, M., 2023. User assessment of fabric shading devices with a low openness factor. Building and Environment. https://doi.org/10.1016/j.buildenv.2022.109707

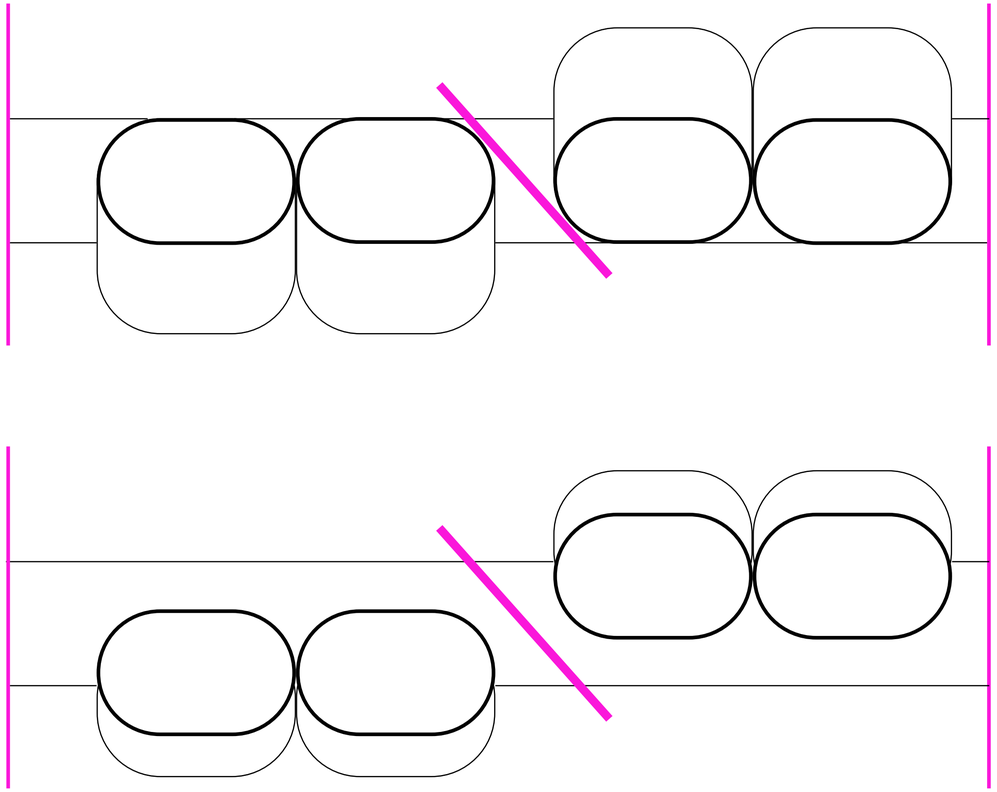
Stephen Wasilewski

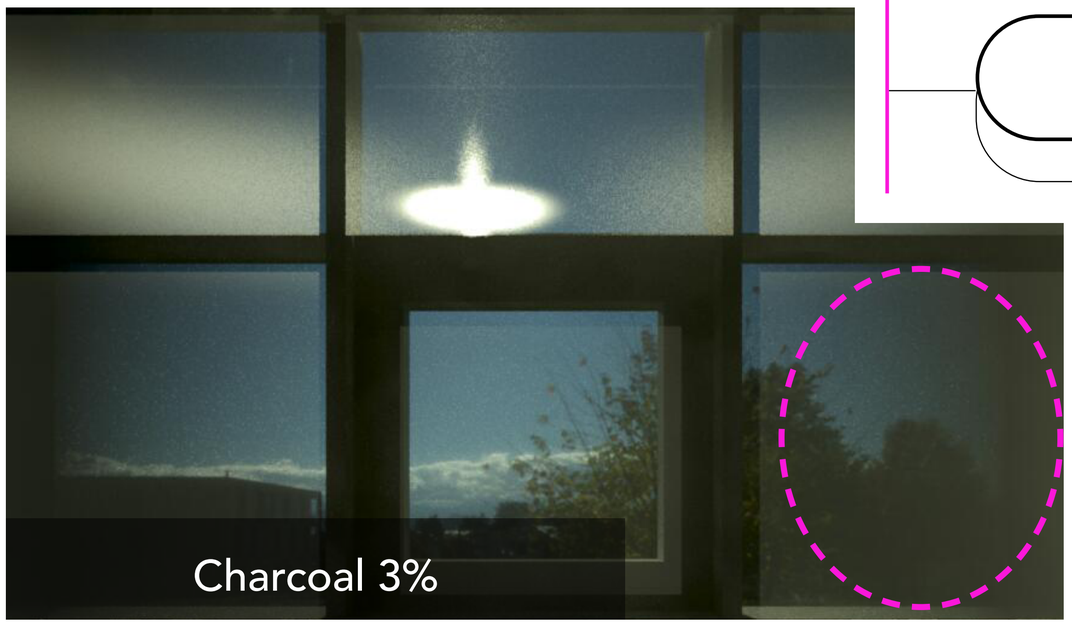
Fabric Variations

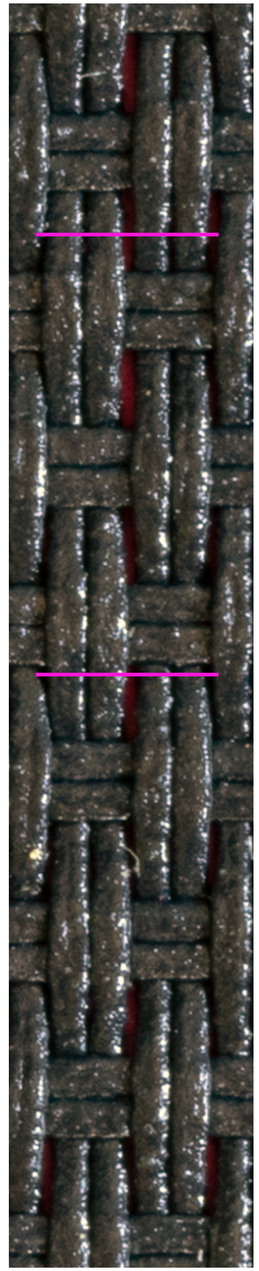
Stephen Wasilewski



Too much cutoff

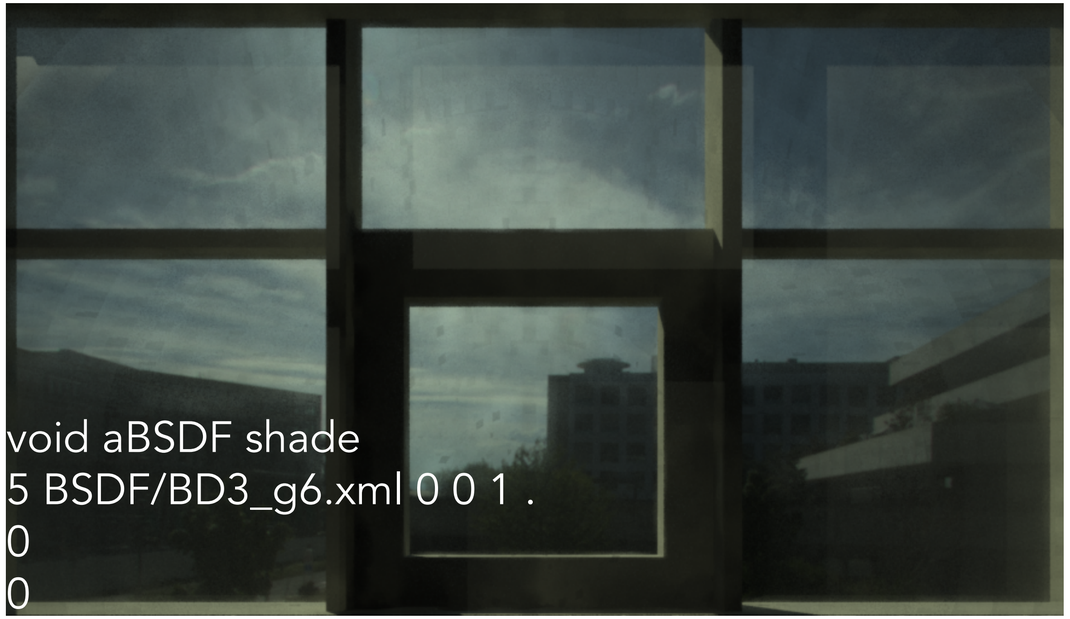


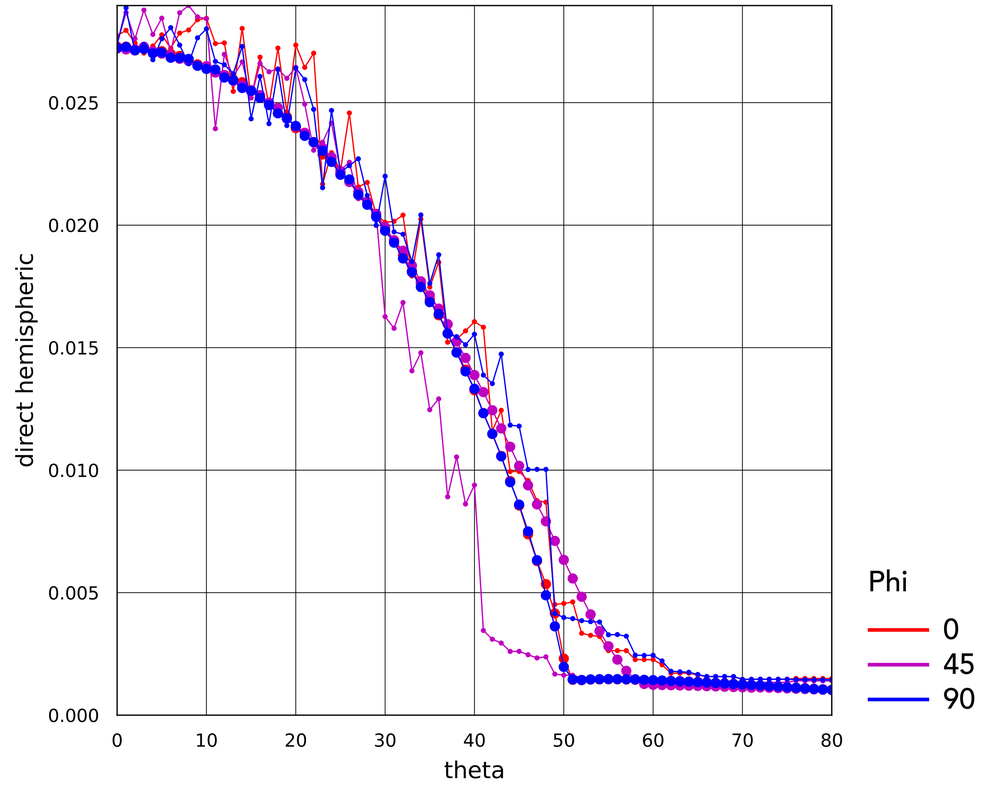


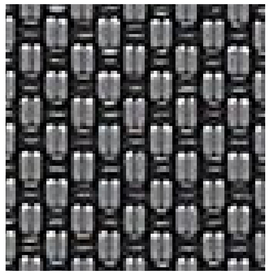


Stephen Wasilewski

Comparison to measured BSDF data - 3% dark





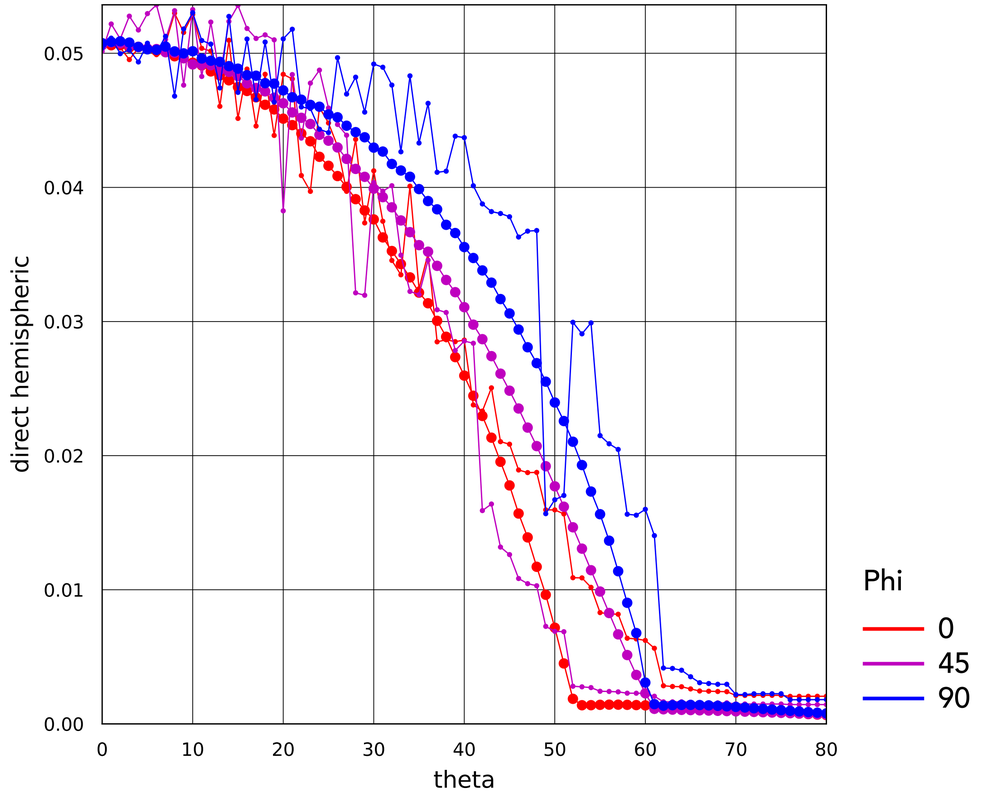


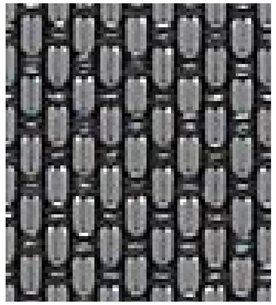
Thanks Taoning!

Wang, T., Lee, E.S., Ward, G.J., Yu, T., 2022. Field validation of data-driven BSDF and peak extraction models for light-scattering fabric shades. Energy and Buildings 262, 112002. https://doi.org/10.1016/j.enbuild.2022.112002

Stephen Wasilewski

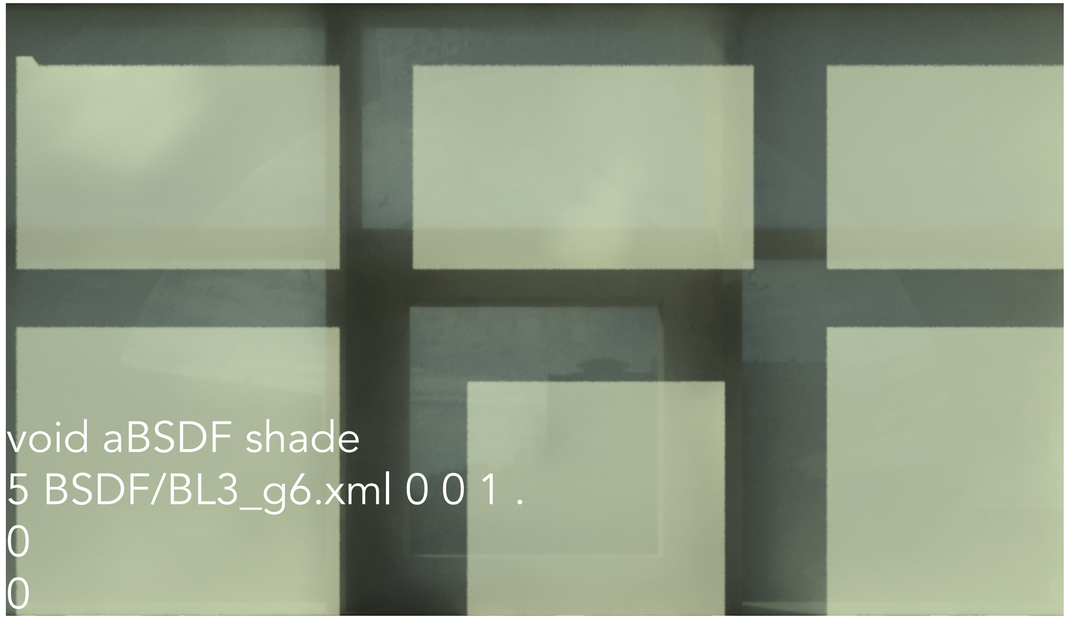
Comparison to measured BSDF data - 5% dark

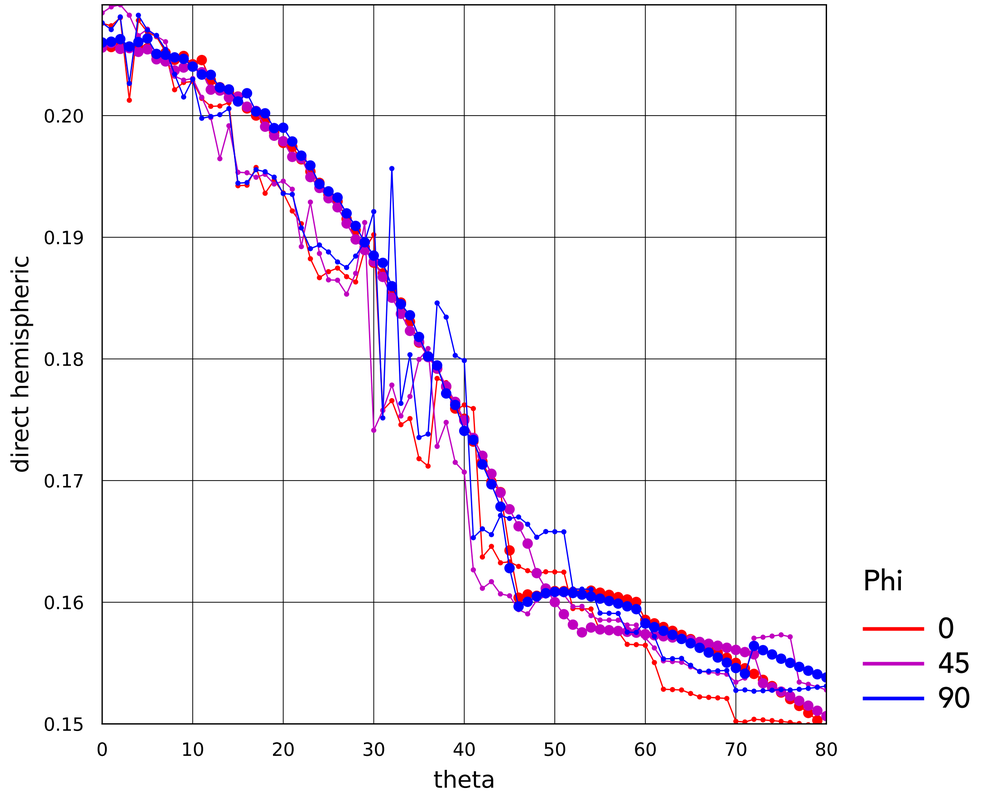




Stephen Wasilewski

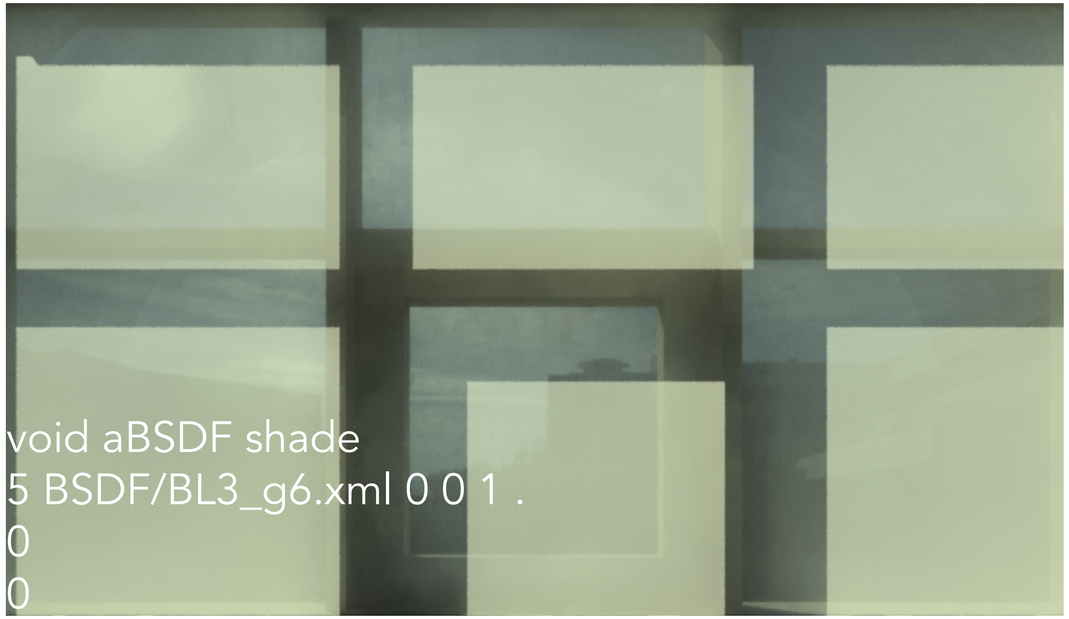
Comparison to measured BSDF data - 3% light

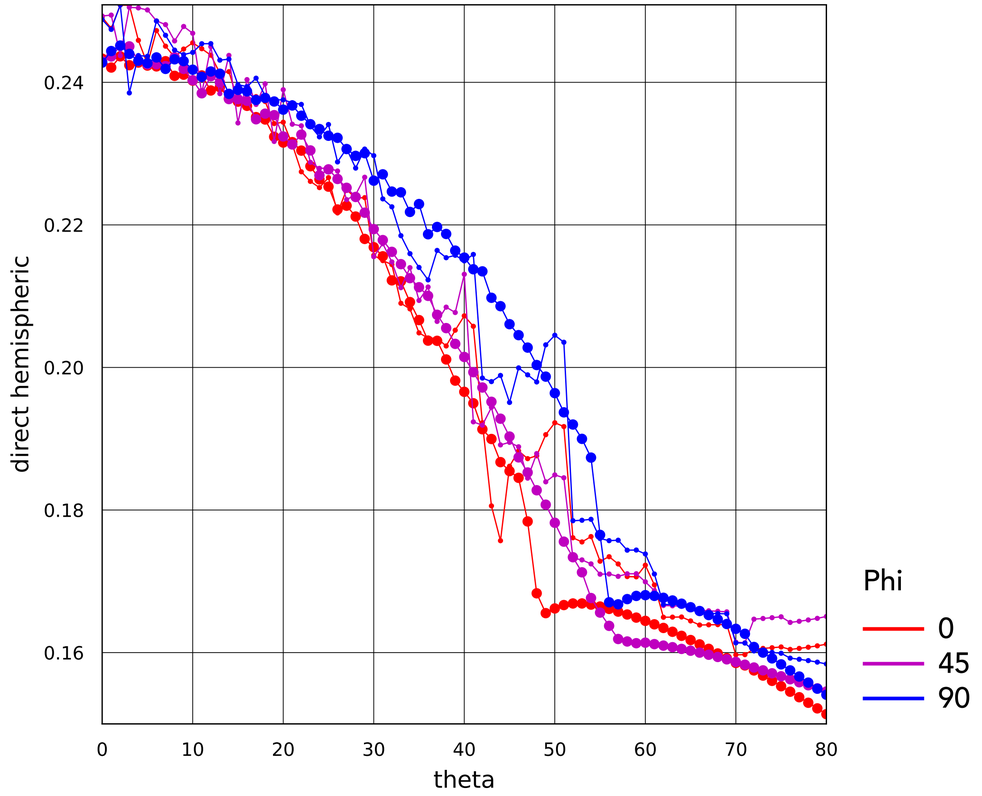




Stephen Wasilewski

Comparison to measured BSDF data - 5% light





Stephen Wasilewski

Thanks for listening

QUESTIONS?

L I P I D

Laboratory of Integrated

Performance in Design

```
{****************** re-projection stuff *********************}

{ this is just to get a normalized coordinate system so Z up is the assumed surface normal for the
rest of the calcs}

rZ = if(1e-6 - Nx, -acos(bound(-1,Ny,1)), acos(bound(-1,Ny,1))); {translation to YZ plane}
rX = if(1e-6 - Ny, -acos(bound(-1,Nz,1)), acos(bound(-1,Nz,1))); {translation to Z axis}

rZx = cos(rZ)*Dx - sin(rZ)*Dy; {apply YZ translation to ray}
rZy = sin(rZ)*Dx + cos(rZ)*Dy;
rZz = Dz;

{all coordinates with Z up normal}
rXx = rZx; {apply Z axis translation to ray}
rXy = cos(rX)*rZy - sin(rX)*rZz;
rXz = sin(rX)*rZy + cos(rX)*rZz;

aDx = abs(rXx); aDy = abs(rXy); aDz = abs(rXz);

{********************** for anisotropy, handle horizontal **********************}

{only used to mix edgex/edgey note that UV/xy relationships are flipped from rest of cal file}
{this only works for vertical or horizontal surfaces, for others use isotropic}

tx = 0;
ty = if(abs(Nz)-.99, 1, 0);
tz = 1-ty;

{************************ fabric shade model stuff ****************************}

{---------- manipulate parameters into necessary geometric values ------------}

{ calculate thread width as function of parameters }
{threadx is the width of the vertical thread, this logic carries throughout}
{thready is the width of the horizontal thread}
threadx = ((A1*A4+A2) - sqrt((A1*A4+A2)*(A1*A4+A2) - 4*A4*A1*A2*(1-A3)))/(2 * A4);
thready = threadx * A4;

{ calculate thread depth based on cutoff angle and opening size or depth ratio }
depthy = if(A5-10, (A2-thready)/(1/cos(A5*PI/180)-1), A5*thready);
depthx = if(A5-10, depthy, A5*threadx);

cutoffy = acos(1/((A2-thready)/depthy+1));
cutoffx = acos(1/((A1-threadx)/depthx+1));

{ amount of hole shrinking) assuming rounded thread
with radius=depth}
angx = Atan2(aDx, aDz);
angy = Atan2(aDy, aDz);
cthetax = cos(angx);
cthetay = cos(angy);
xfor = if(1e-6 - aDz, A1-threadx, min(A1-threadx, depthx*(1/cthetax-1)));
yfor = if(1e-6 - aDz, A2-thready, min(A2-thready, depthy*(1/cthetay-1)));

{fraction of cell that is blocked by thread given incident angle}
fillx = (threadx+xfor)/A1;
filly = (thready+yfor)/A2;
```

```
{----------------- aesthetic and avoid aliasing artifacts ----------------------}

{ jitter hole position }

ru = rand(floor(V/A2));
rv = rand(floor(U/A1));

wU = A6*(threadx)*(ru - 0.5);
wV = A6*(thready)*(rv - 0.5);

{location in periodic UV space perturbed by jitter}
hU = abs(frac((U+wU)/A1));
hV = abs(frac((V+wV)/A2));

centU = inside(fillx*.5, hU, 1-fillx*.5);
centV = inside(filly*.5, hV, 1-filly*.5);

{------------------------------ acuity stuff ---------------------------------}

{120 pixels/degree 20/20 vision}
acuity = if(arg(0) - 6, tan(PI/180/A7), 1.75 / 6000);

{ S-shaped falloff to blend between fabric_m and holes, 100% holes when at least 4 pixels per hole
50% holes at 2 pixel per hole}
acuity_falloff(x): sq(cos(min(x,2)*PI/4));

{relative hole size}
grid2acuity = min(A1-threadx,A2-thready)/max(FTINY,T) / acuity;
blurfactor = 0.95 * acuity_falloff(grid2acuity);

{--------------------------- mixfunc variables ---------------------------------}

{void:thread}
holes = if(and(centU, centV),1,0);
fabric_m = (1 - fillx) * (1 - filly);
fabric = blurfactor * fabric_m + (1 - blurfactor) * holes;
binary_edge = if(and(inside(threadx/A1*0.5, hU, 1 - 0.5*threadx/A1), inside(thready/A2*0.5, hV, 1 -
0.5*thready/A2)),1,0);

{thread_t:thread_p}
Ac : 1;
Bc : 1 / (1-exp(-Ac));
thread = Bc*(1-exp(-Ac*aDz));

{for anisotropic scattering}
{primary:secondary scatter}
{blend between two different roughness levels, the denominator can be used to manipulate the shape
of the inner highlight}
{a larger constant will make the fall off sharper}
scatterx = if(fabric_m,sq(cos(PI*angx/(2*cutoffx))), 0);
scattery = if(fabric_m,sq(cos(PI*angy/(2*cutoffy))), 0);

scatterx_s = if(rand(V)-scatterx, 0, .5) + 0.5 * scatterx;
scattery_s = if(rand(U)-scattery, 0, .5) + 0.5 * scattery;

{threadx:thready}
{ratio of hole size, the longer side will catch more highlights, this ratio is taken as constant}
t_orient = (A2-thready)/(A1+A2-threadx-thready);
t_orient_s = if(rand(U*PI+V)-t_orient,0,0.5) + 0.5 * t_orient;
```