

# Mosaics – Dealing with a precious material



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

**11th International Radiance Workshop 2012**  
**Andreas Noback**



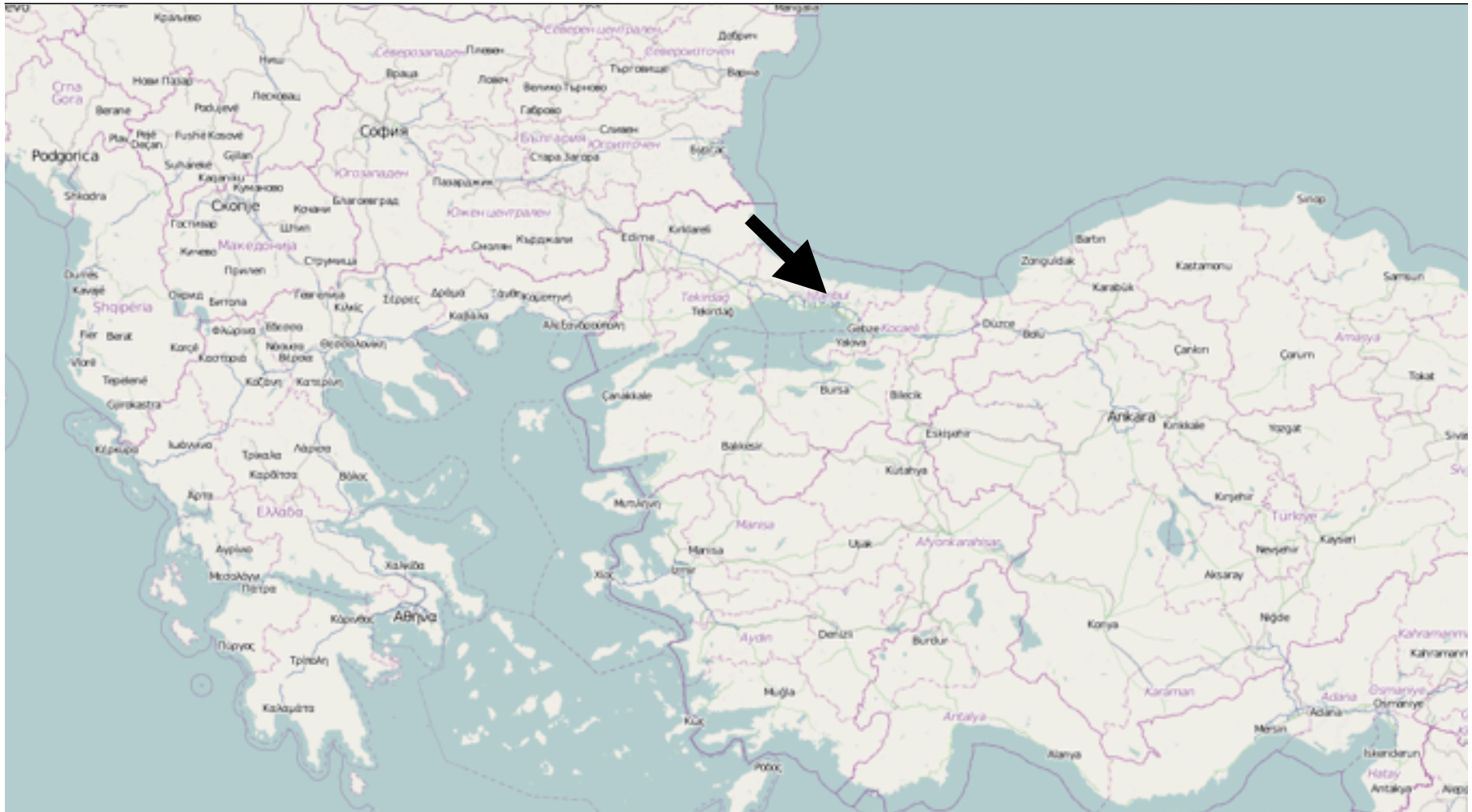
13.9.2012 | Fachbereich Architektur | Fachgebiet Klassische Archäologie | Andreas Noback |

Donnerstag, 13. September 12

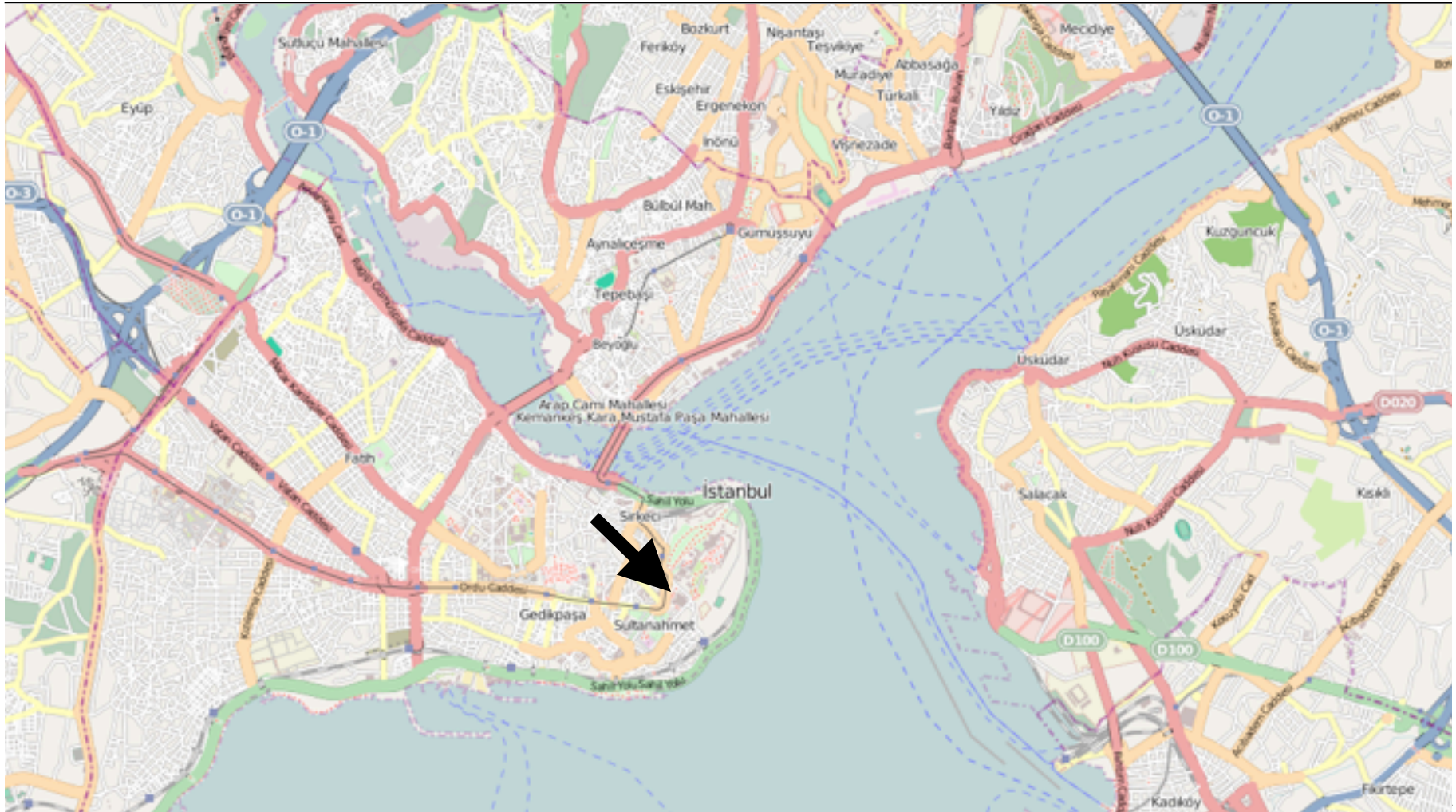
# Radiance Workshop 2008



# Byzantion/Istanbul



# Hagia Sophia



# Hagia Sophia



# Emperor Iustinians Church



---

# Procopius

## De Aedificis 6th century A.D.

---

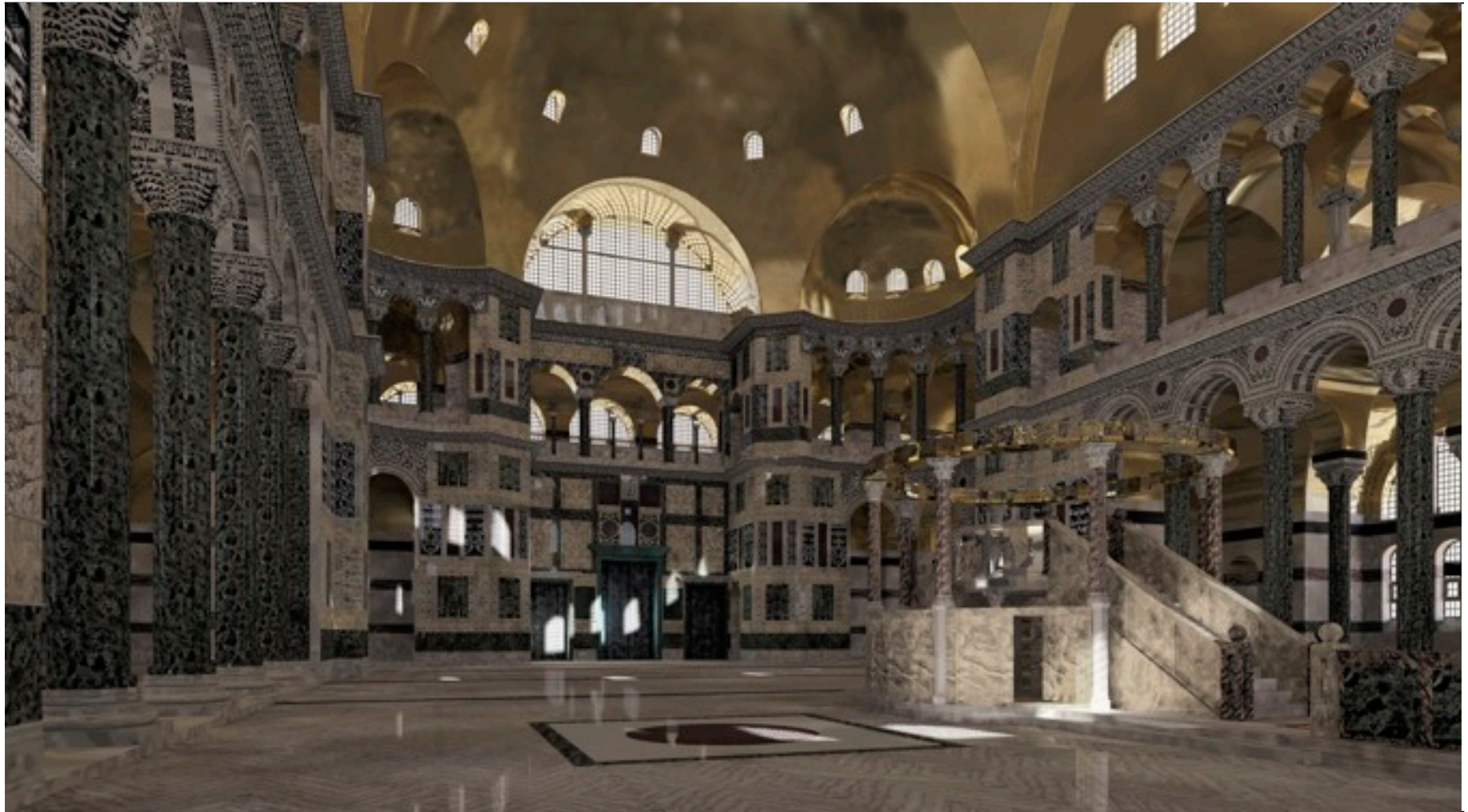


„The church is singularly full of light and sunshine; you would declare that the place is not lighted by the sun from without, but that the rays are produced within itself, such an abundance of light is poured into this church ...”

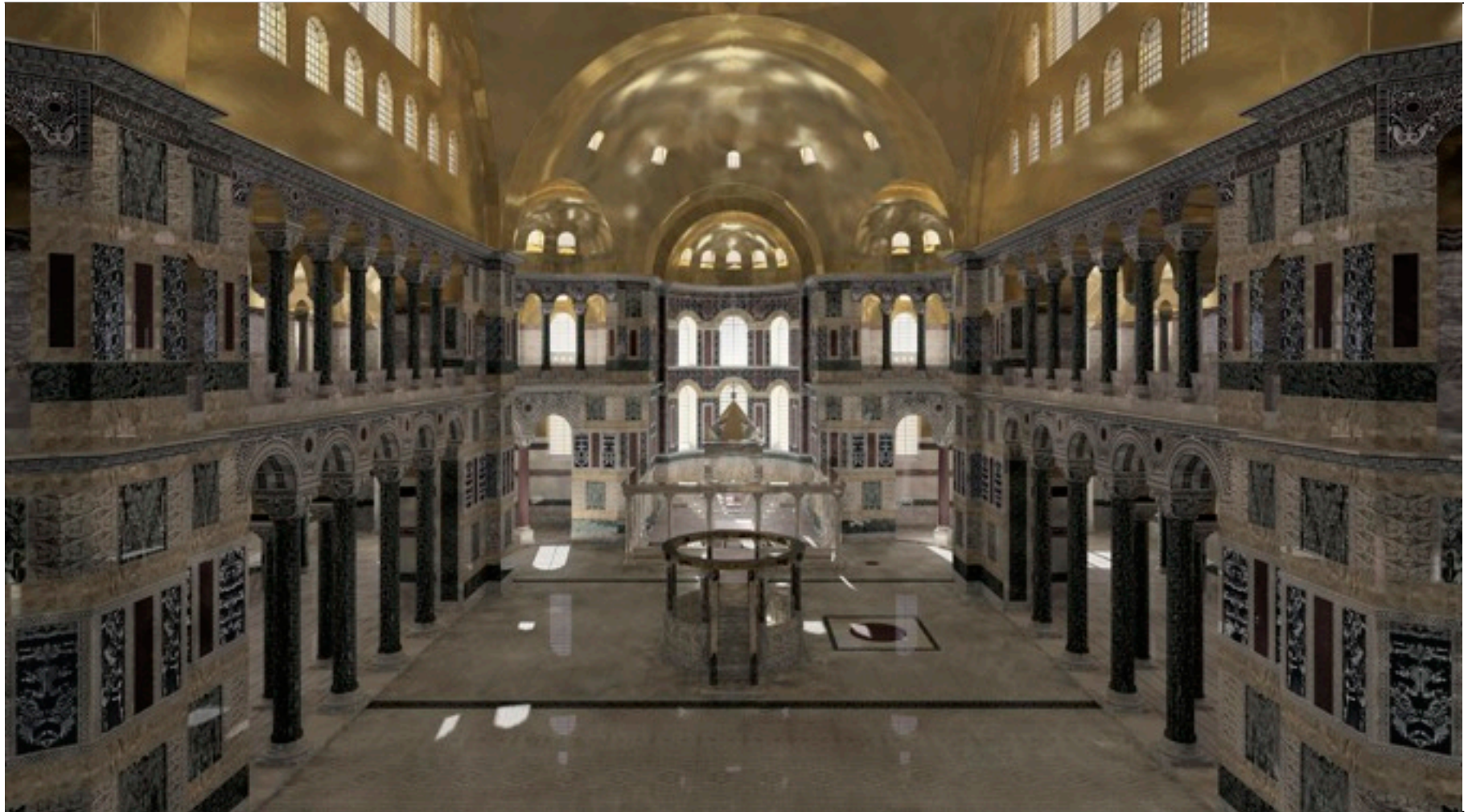




# Reconstruction



# Reconstruction



# Reconstruction



# common.mat

---

```
void metal gold_mosaic_1
0
0
5 0.997303 0.781942 0.334048 0.9 0.02
```

# Mosaics



# Mosaics



# tesserae from Iustiniana Prima



# tesserae orientation

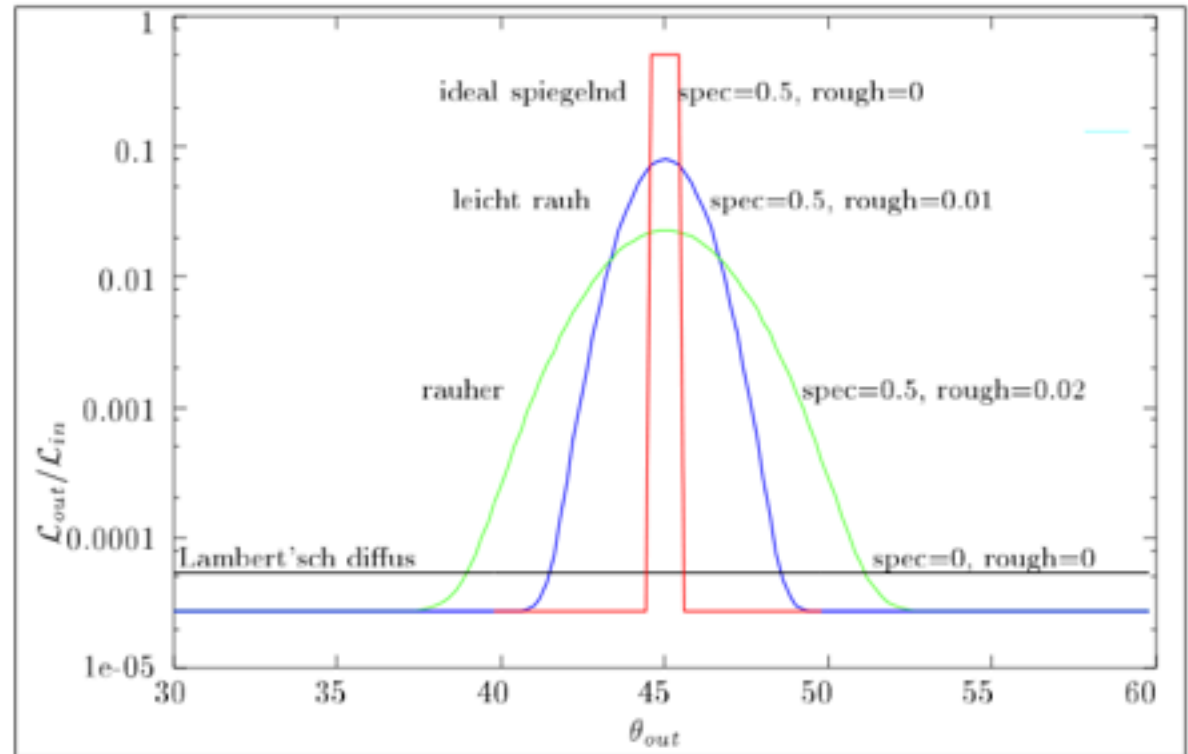
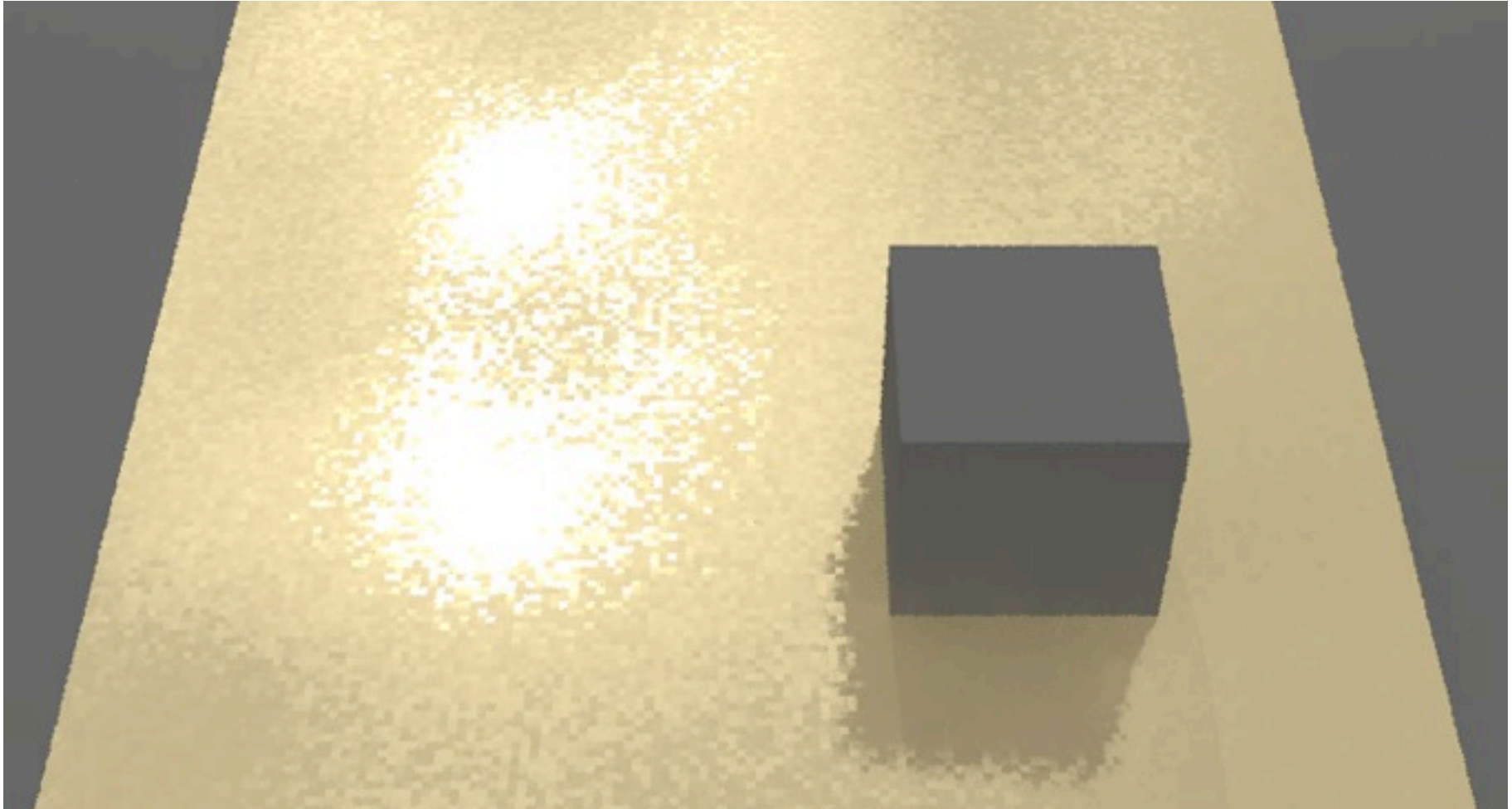


Abbildung A.3: Reflexion für verschiedene Parameter einer "plastic" Oberfläche  $\theta_{in} = 45$

Sources: Antoniadis, Ekphrasis, Peter Apian Bennewitz: Messung und Modellierung von lichtstreuenden Materialien zur Computer-Simulation von Tageslichtbeleuchtungen



```
{
mosaic.cal

void texfunc mosaic
4 aggregate_tilt_dx aggregate_tilt_dy aggregate_tilt_dz mosaic.cal
0
7 0 .1 0 .03 150 .07 5

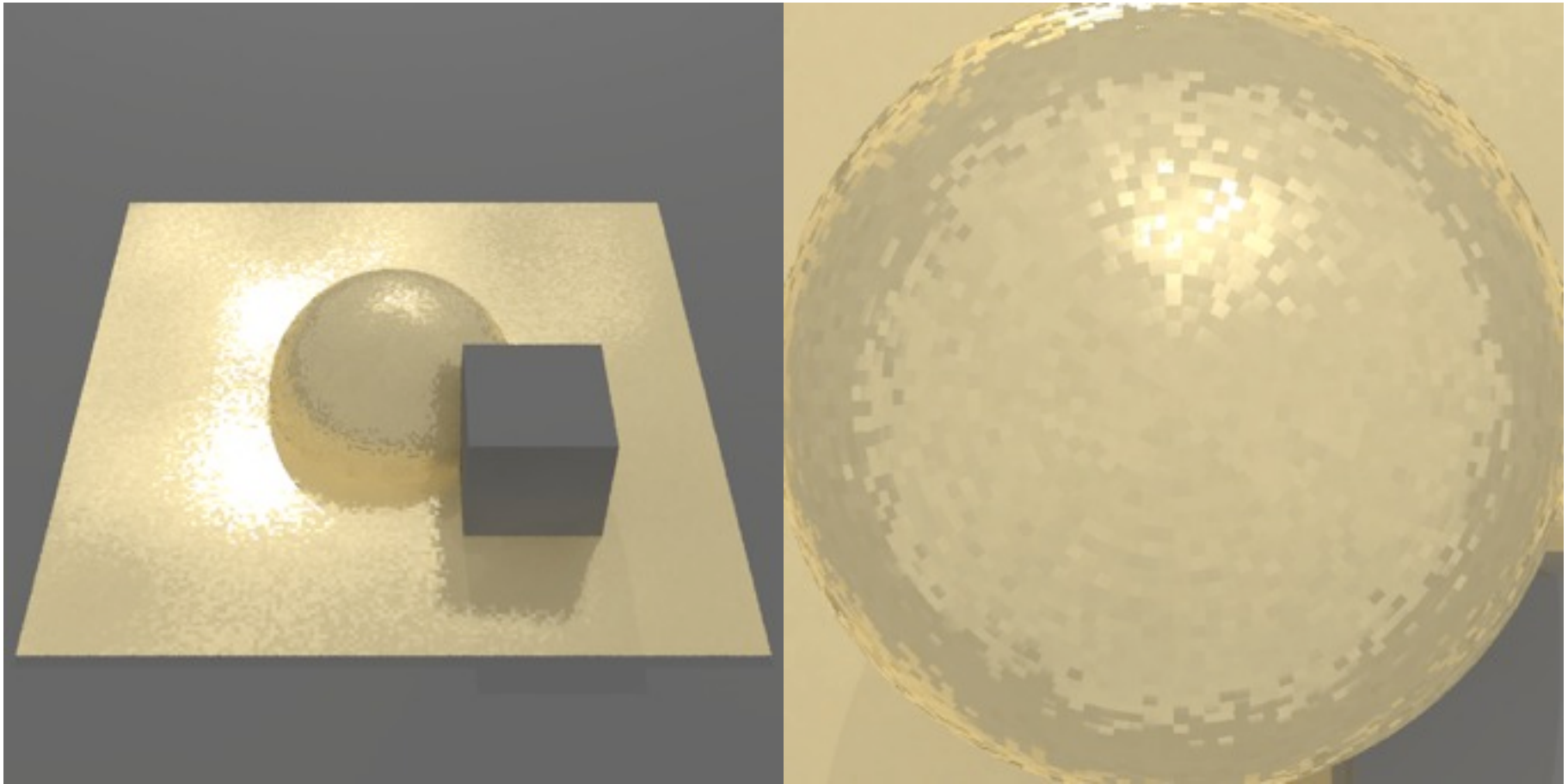
A1-A3 intentional tilt x,y,z
A4 inaccuracy factor
A5 mosaic pieces per meter
A6 bumpiness factor of the plaster
A7 scaling factor of the noise function for bumpiness
    (autocorrelation distances per meter)
}

bumpiness_dx = A6 * noise3a(A7 * Px,A7 * Py,A7 * Pz);
bumpiness_dy = A6 * noise3b(A7 * Px,A7 * Py,A7 * Pz);
bumpiness_dz = A6 * noise3b(A7 * Px,A7 * Py,A7 * Pz);
```

```
inaccuracy_tilt_dx = A4 * noise3a(A5 * floor(A5 * Px)/A5 ,A5 *
floor(A5 * Py) /A5 ,A5 * floor(A5 * Pz) /A5);
inaccuracy_tilt_dy = A4 * noise3b(A5 * floor(A5 * Px)/A5 ,A5 *
floor(A5 * Py) /A5 ,A5 * floor(A5 * Pz) /A5);
inaccuracy_tilt_dz = A4 * noise3c(A5 * floor(A5 * Px)/A5 ,A5 *
floor(A5 * Py) /A5 ,A5 * floor(A5 * Pz) /A5);

aggregate_tilt_dx = inaccuracy_tilt_dx + A1 + bumpiness_dx;
aggregate_tilt_dy = inaccuracy_tilt_dy + A2 + bumpiness_dy;
aggregate_tilt_dz = inaccuracy_tilt_dz + A3 + bumpiness_dz;
```

# mosaic\_sphere.cal



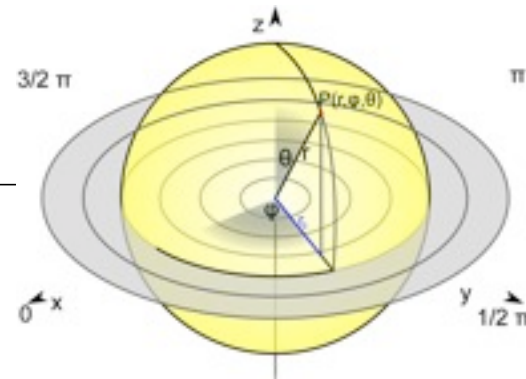
# mosaic\_sphere.cal

```
{  
mosaic_sphere.cal
```

```
void texfunc mosaic_sphere  
10 aggregate_tilt_dx aggregate_tilt_dy aggregate_tilt_dz  
    mosaic_sphere.cal -s 3.425 -t 0 0 1  
0  
8 0 0 0 .01 46 0 0 3.425
```

A1-A3 intentional tilt x,y,z  
A4 inaccuracy factor  
A5 mosaic pieces per unit (meter)  
A6 bumpiness factor of the plaster  
A7 scaling factor of the noise function for bumpiness  
 (autocorrelation distances per unit (meter))  
A8 size of the sphere in units (meter)  
}

```
R = sqrt(Px*Px + Py*Py + Pz*Pz);  
POL = acos(Pz/R);  
AZI = atan2(Py,Px);  
PR = floor(A5 * PI * 2 * A8);
```



TECHNISCHE  
UNIVERSITÄT  
DARMSTADT

```
PDIV = ceil(PR * POL / PI / 2 -.5);  
ADIV = floor(floor(PR * sin(PDIV*PI*2/PR)) * (AZI+PI)/(2*PI));
```

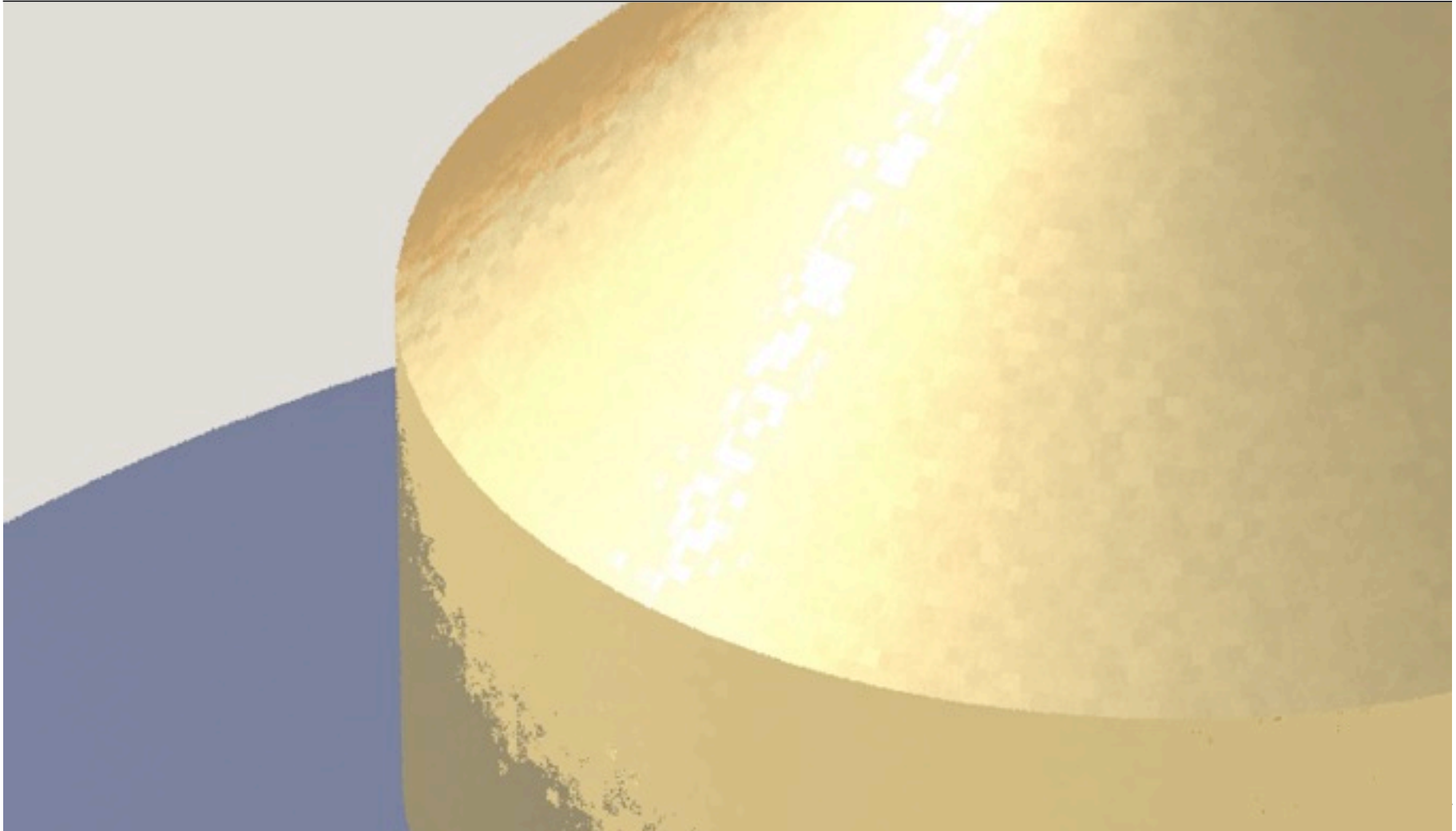
```
inaccuracy_tilt_dx = A4 * noise3a(ADIV, PDIV, 1);  
inaccuracy_tilt_dy = A4 * noise3b(ADIV, PDIV, 1);  
inaccuracy_tilt_dz = A4 * noise3c(ADIV, PDIV, 1);
```

```
PB = POL * A7 * A8;  
AB = AZI * A7 * A8;
```

```
bumpiness_dx = A6 * noise3a(AB, PB, A7);  
bumpiness_dy = A6 * noise3b(AB, PB, A7);  
bumpiness_dz = A6 * noise3c(AB, PB, A7);
```

```
aggregate_tilt_dx = inaccuracy_tilt_dx + A1 + bumpiness_dx;  
aggregate_tilt_dy = inaccuracy_tilt_dy + A2 + bumpiness_dy;  
aggregate_tilt_dz = inaccuracy_tilt_dz + A3 + bumpiness_dz;
```

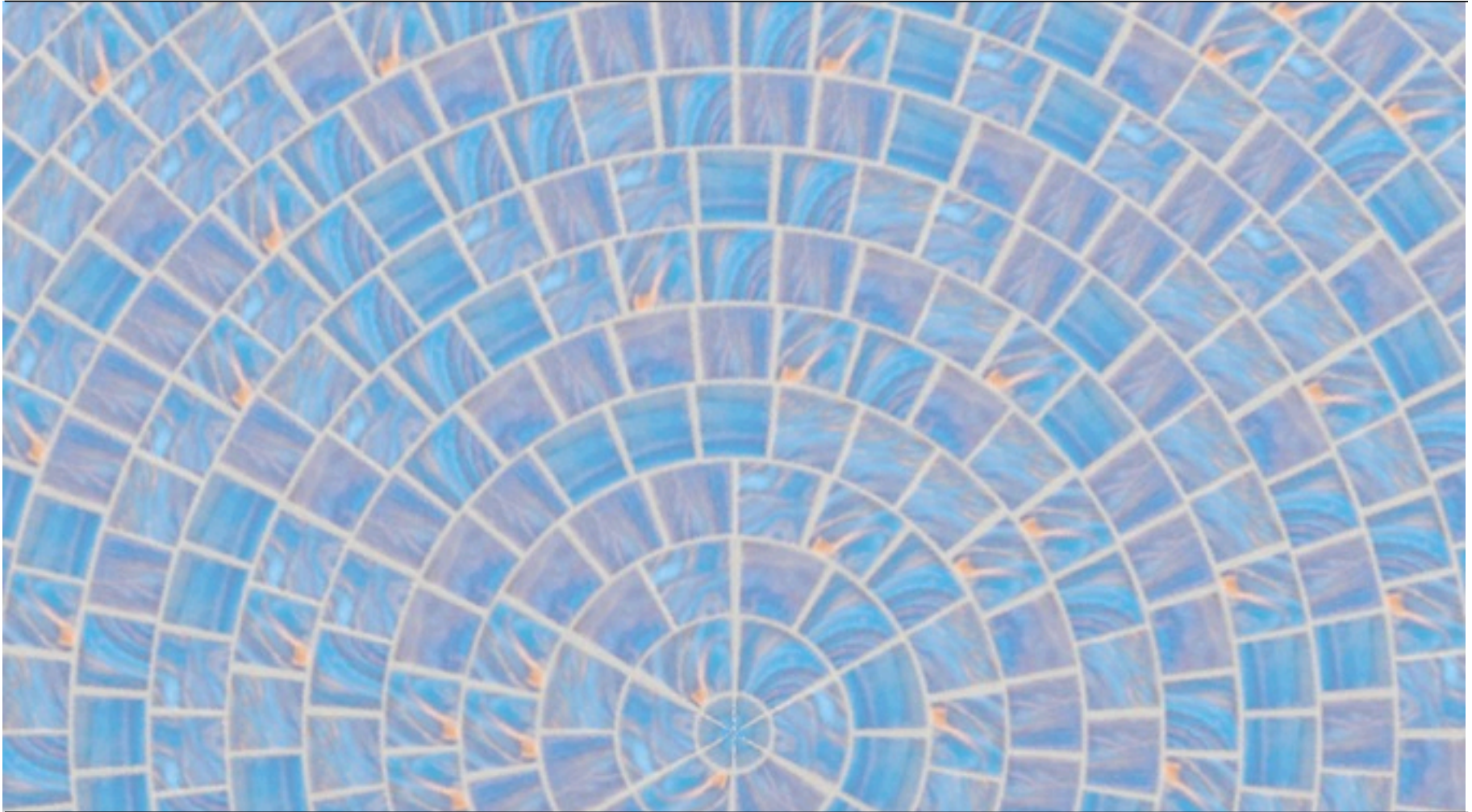
# Cone and Cylinder



13.9.2012 | Fachbereich Architektur | Fachgebiet Klassische Archäologie | Andreas Noback |

Donnerstag, 13. September 12

# pic\_sphere.cal



# pic\_sphere.cal

```
{
pic_sphere.cal

mosaic_sphere colorpict mosaic_sphere_pic
13 noop noop noop picture.hdr pic_sphere.cal tile_u tile_v -s 3.425 -t
0 0 1
0
3 46 3.425 7

A1 mosaic pieces per unit (meter)
A2 scaling
A3 tiles in picture
}
```

```
R = sqrt(Px*Px + Py*Py + Pz*Pz);
POL = acos(Pz/R);
AZI = atan2(Py,Px);
PR = A1 * PI * 2 * A2;
```

```
PDIV = ceil(PR * POL / PI /2-.5);
ADIV = floor(floor(PR * sin(PDIV*PI*2/PR)) * (AZI+PI)/(2*PI));
```

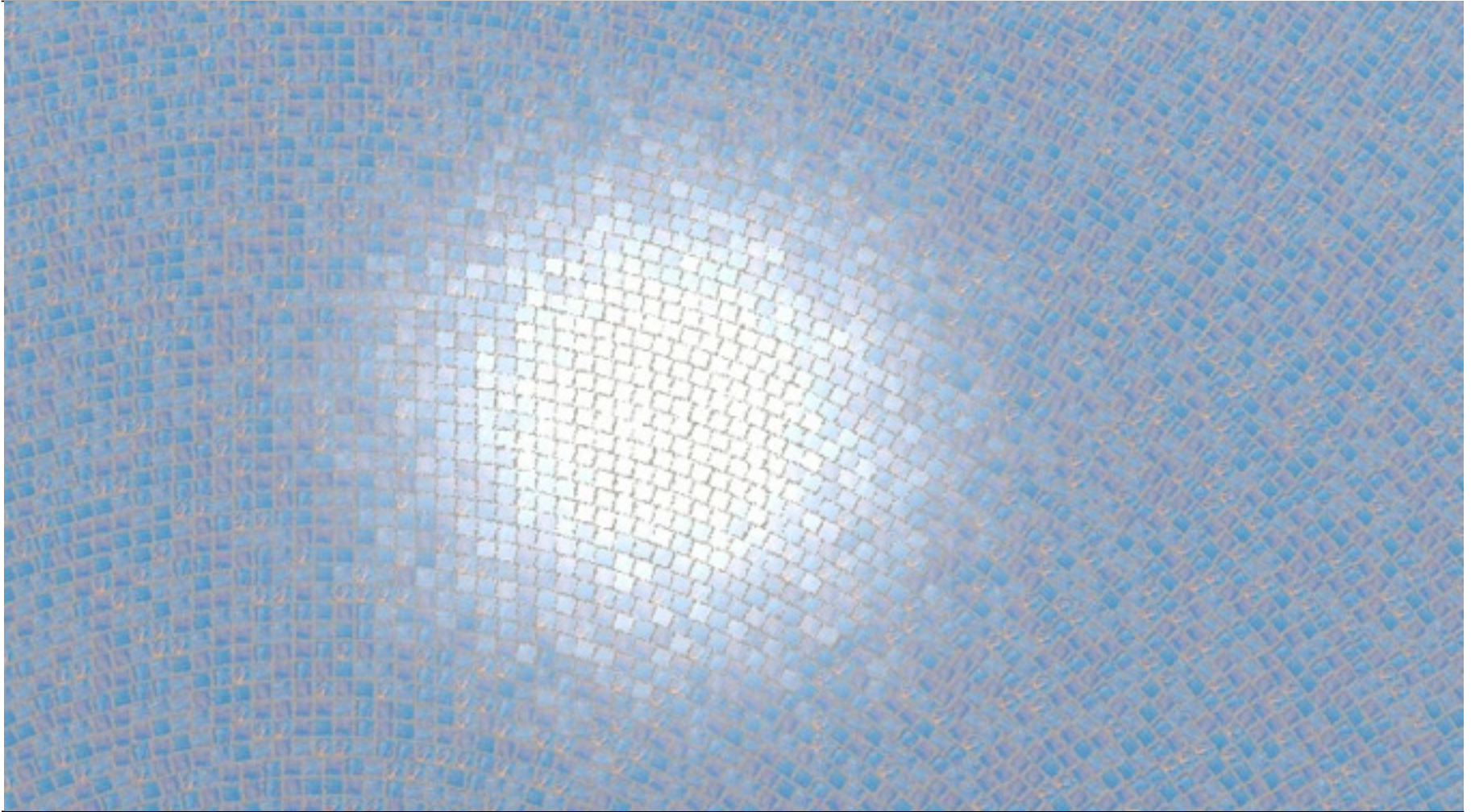
```
AZ = floor(A1 * 2 * PI * A2);
DIV_V = 2 * PI / AZ;
LP = POL - (DIV_V / 2);
PD = ceil(LP / DIV_V);
REST = (PD * DIV_V) - LP;

PPD = if (PD,PD,1);
AZU = floor(AZ * sin(PPD * DIV_V));
DIV_U = 2 * PI / AZU;
LA = AZI+PI;
PA = floor(LA / DIV_U);
REST_A = LA - (PA * DIV_U);
```

```
ptile_u = REST_A / DIV_U;
tile_v = REST / DIV_V;
```

```
incr = A3 * noise3a(ADIV, PDIV, 1);
pincr = if (incr, incr, incr * -1);
tile_u = floor(pincr) + ptile_u;
```





# mixpict

```
void texfunc mosaic_sphere
```

```
10 aggregate_tilt_dx aggregate_tilt_dy aggregate_tilt_dz  
mosaic_sphere.cal -s 3.425 -t 0 0 1
```

```
0
```

```
8 0 0 0 .01 46 0 0 3.425
```

```
mosaic_sphere colorpict mosaic_sphere_pic
```

```
13 noop noop noop _MG_8745_strip.hdr pic_sphere.cal tile_u tile_v -s  
3.425 -t 0 0 1
```

```
0
```

```
3 46 3.425 7
```

```
mosaic_sphere_pic plastic mat_04_mosaik_blaue
```

```
0
```

```
0
```

```
5 1 1 1 .02 .075
```

```
void plastic mat_04_fugen
```

```
0
```

```
0
```

```
5 .20 .19 .16 0 0
```

```
void mixpict mat_04_mix
```

```
13 mat_04_mosaik_blaue mat_04_fugen green _MG_8745_strip_bw.hdr  
pic_sphere.cal tile_u tile_v -s 3.425 -t 0 0 1
```

```
0
```

```
3 46 3.425 7
```



# mixpict & texdata



# material.mat

```
tuerkis_mat texdata tuerkis
9 pass_dx pass_dy nopert xtuerkis.dat ytuerkis.dat ytuerkis.dat
  tex.cal frac(Px) frac(Py)
0
0
```

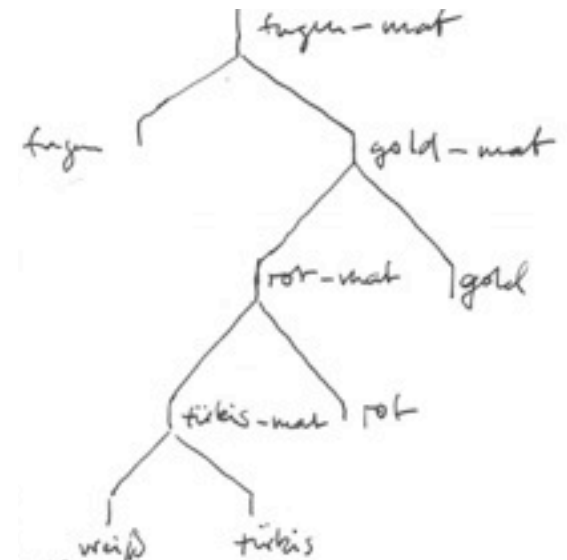
```
gold_mat texdata gold
9 pass_dx pass_dy nopert xgold.dat ygold.dat ygold.dat tex.cal
frac(Px) frac(Py)
0
0
```

```
void mixpict tuerkis_weiss
7 weiss tuerkis green tuerkis-mat.hdr picture.cal pic_u pic_v
0
0
```

```
void mixpict rot_tuerkis_weiss
7 tuerkis_weiss rot green rot-mat.hdr picture.cal pic_u pic_v
0
0
```

```
void mixpict gold_rot_tuerkis_weiss
7 rot_tuerkis_weiss gold green gold-mat.hdr picture.cal pic_u pic_v
0
0
```

```
void mixpict ornament
7 fugen gold_rot_tuerkis_weiss green fugen-mat.hdr picture.cal pic_u
pic_v
0
0
```



# pic2dat.sh

```
#!/bin/bash
xmax=$1;
ymax=$2;
file="$3";

echo "2";
echo "0 1 ${ymax}";
echo "0 1 ${xmax}";
echo "";

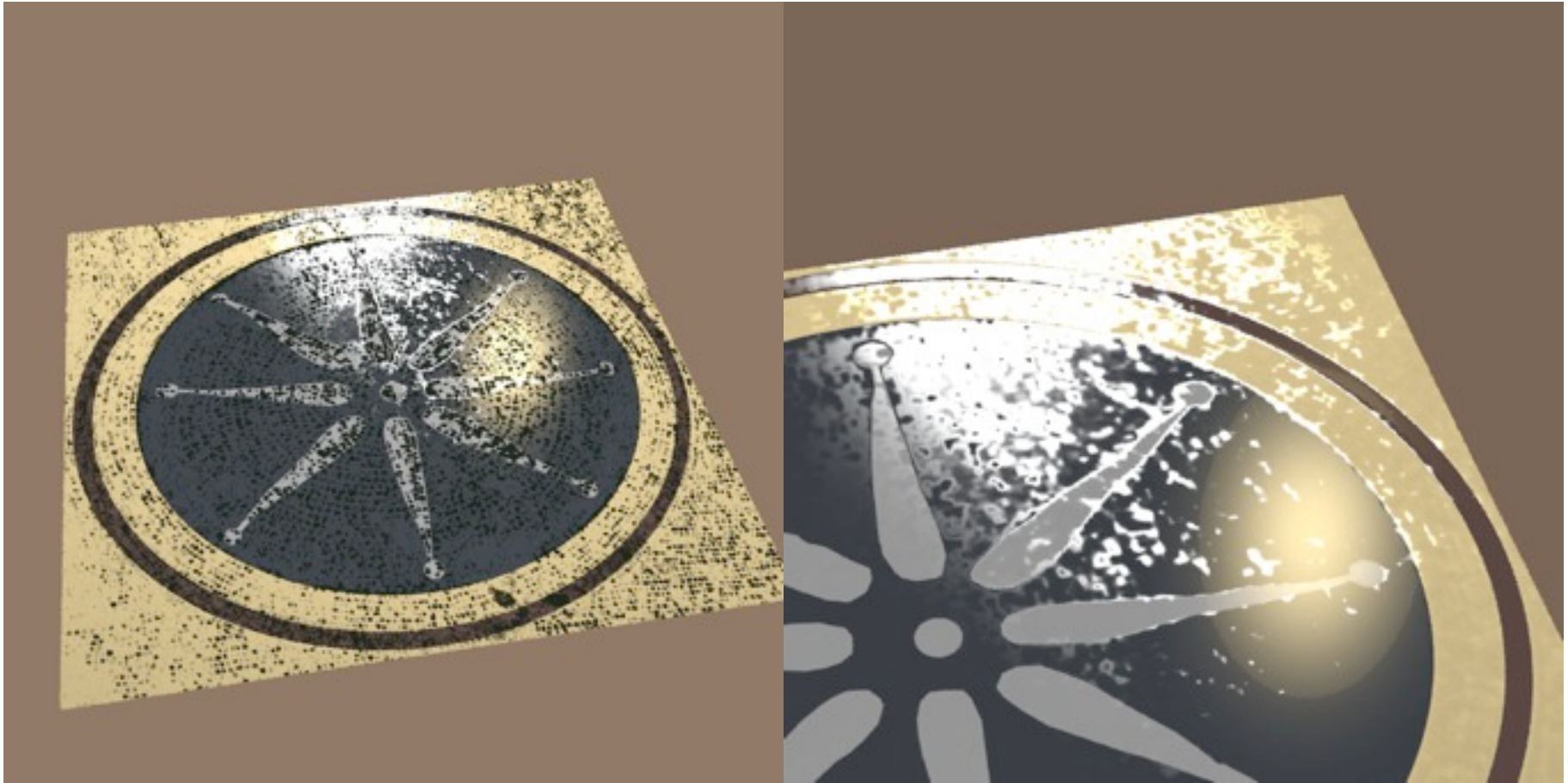
x=1;
y=1;
a=1
for i in $(pvalue -h -H -b -di ${file}); do
    value_a[$a]=$i
    a=$(( $a + 1 ))
done

a=1
y=$ymax
while [ $y -ge 1 ]; do
    x=1
    while [ $x -le $xmax ]; do
        value=$(echo "scale=2; ${value_a[$a]} / 255" | bc)
        echo -n "$value ";
        a=$(( $a + 1 ))
        x=$(( $x + 1 ))
    done;
    echo ;
    y=$(( $y - 1 ))
done;
```

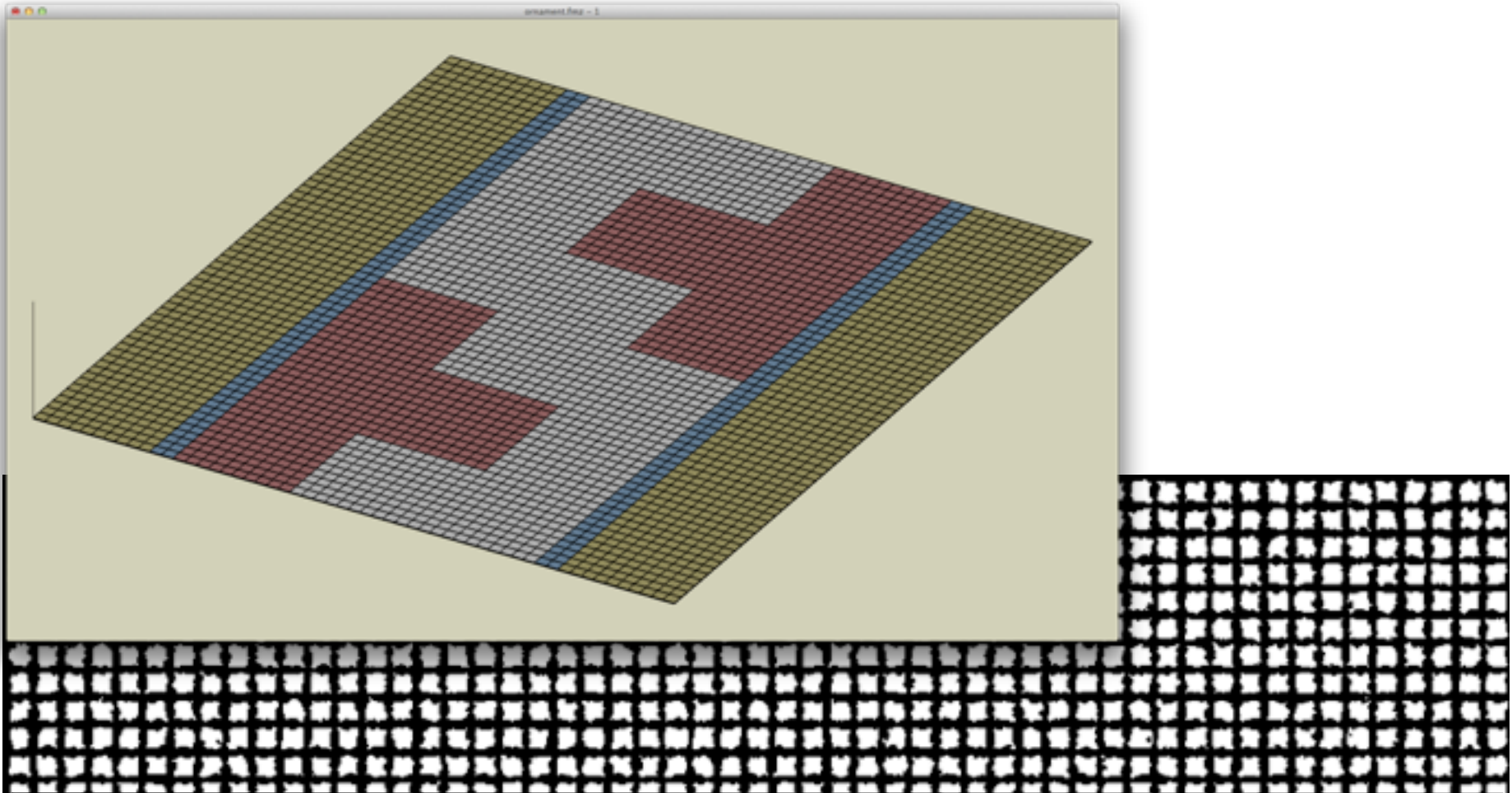
```
{  
tex.cal (based on Radiance Digest Volume 2, Number 4)  
}
```

```
fact=.1;  
pass_dx(dx,dy,dz)=(1-dx)*fact;  
pass_dy(dx,dy,dz)=(1-dy)*fact;  
pass_dz(dx,dy,dz)=(1-dz)*fact;  
nopert(dx,dy,dz)=0;
```

# mixpict & texdata



# Geometry and random materials



# mat.sh & pertub.sh



```
#!/bin/bash
# generates a lot of materials with different perturbation values
```

```
i=1
z=0
mat=$1;

for x in .0 .25 .5 .75 1; do
    for y in .0 .25 .5 .75 1; do

        echo "${mat}_mat texfunc ${mat}_${i}"
        echo "4 aggregate_tilt_dx aggregate_tilt_dy
        aggregate_tilt_dz pert.cal"
        echo "0"
        echo "3 ${x} ${y} ${z}"
        echo ""

        i=$(( i + 1 ))
    done
done
```

```
#!/bin/bash
# usage: ./perturb.sh material 25 geometry.rad

mat=$1
max=$2
file=$3

while read line
do
    let "rand= ${RANDOM} % ${max} +1";
    echo $(echo ${line} | sed -e "s/${mat}/${mat}_${rand}/g")
done < $file
```

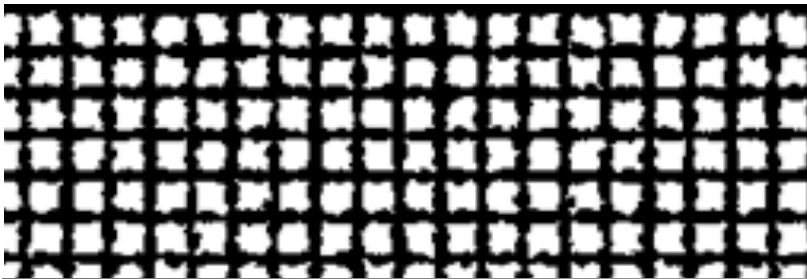
# pert.cal & mixpict

```
{
pert.cal

void texfunc pert
4 aggregate_tilt_dx aggregate_tilt_dy aggregate_tilt_dz pert.cal
0
3 0 .1 0

A1-A3 intentional tilt x,y,z
}

fact = .3;
aggregate_tilt_dx = Px + A1 * fact;
aggregate_tilt_dy = Py + A2 * fact;
aggregate_tilt_dz = Pz + A3 * fact;
```

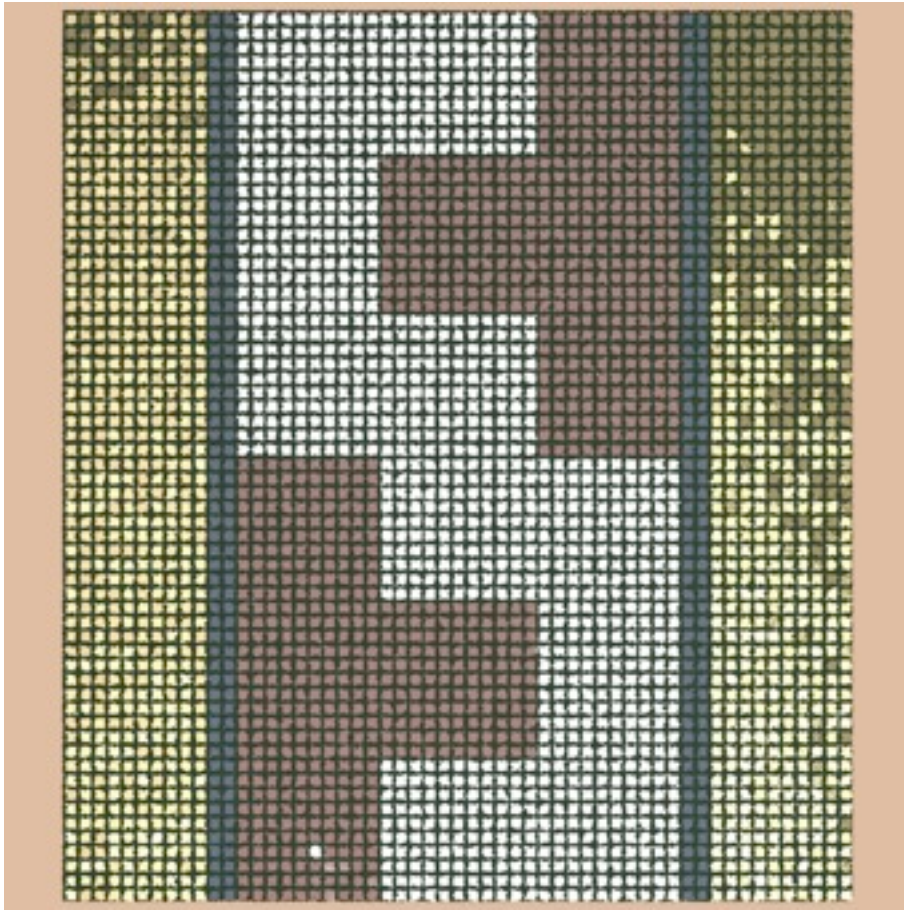


```
void plastic black
0
0
5 .01 .02 .01 0 0
[...]
void metal gold
0
0
5 0.88 0.69 0.28 0.8 0.02

void mixpict gold_mat
9 gold black green ornament.hdr picture.cal tile_u tile_v -s .55
0
1 1.1271
[...]
gold_mat texfunc gold_1
4 aggregate_tilt_dx aggregate_tilt_dy aggregate_tilt_dz pert.cal
0
3 .0 .0 0

gold_mat texfunc gold_2
4 aggregate_tilt_dx aggregate_tilt_dy aggregate_tilt_dz pert.cal
0 [...]
```

# Geometry and random materials



---

# Thank You!

---



- Questions?