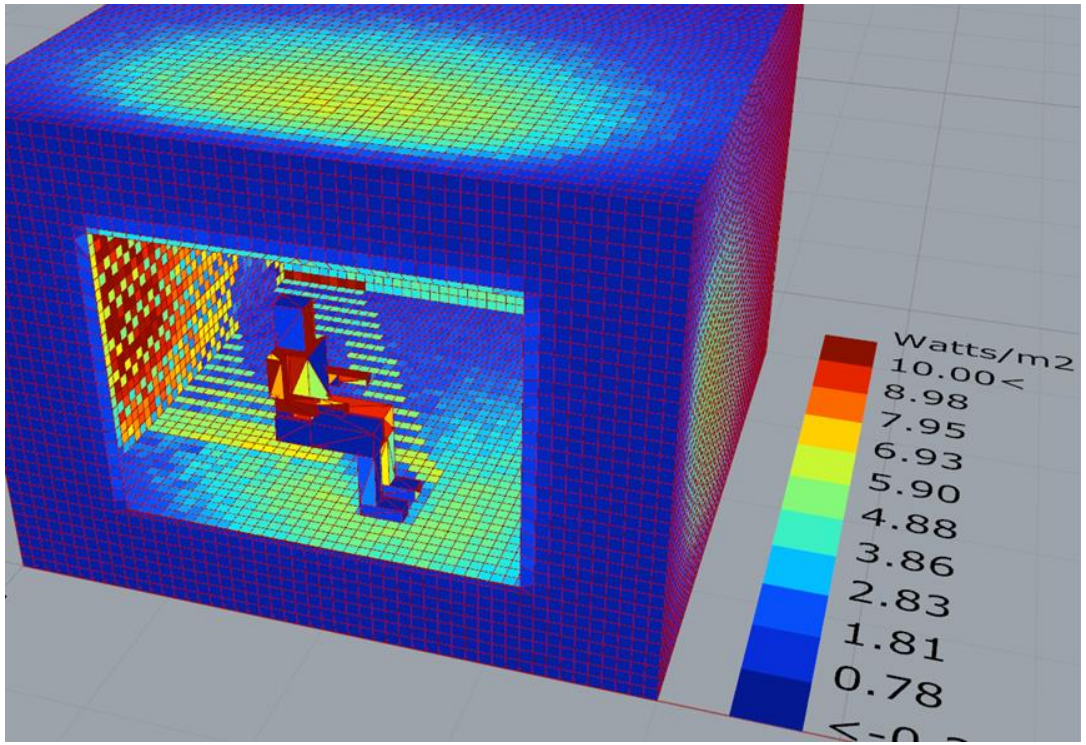
gebäude*
systeme
technik

TECHNISCHE UNIVERSITÄT
KAISERSLAUTERN

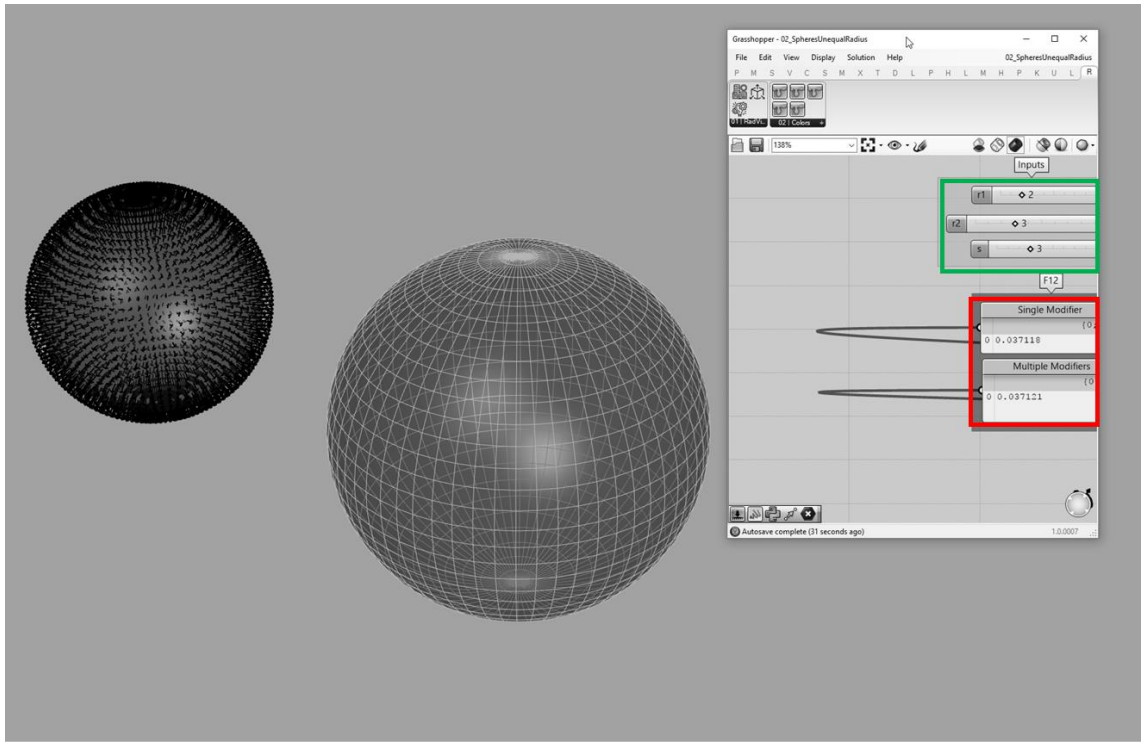
Employing Radiance in Heat Transfer Applications

Sarith Subramaniam
Department of Civil Engineering, TU Kaiserslautern, Germany

Goal: Compute radiative transfer from sky and between surfaces

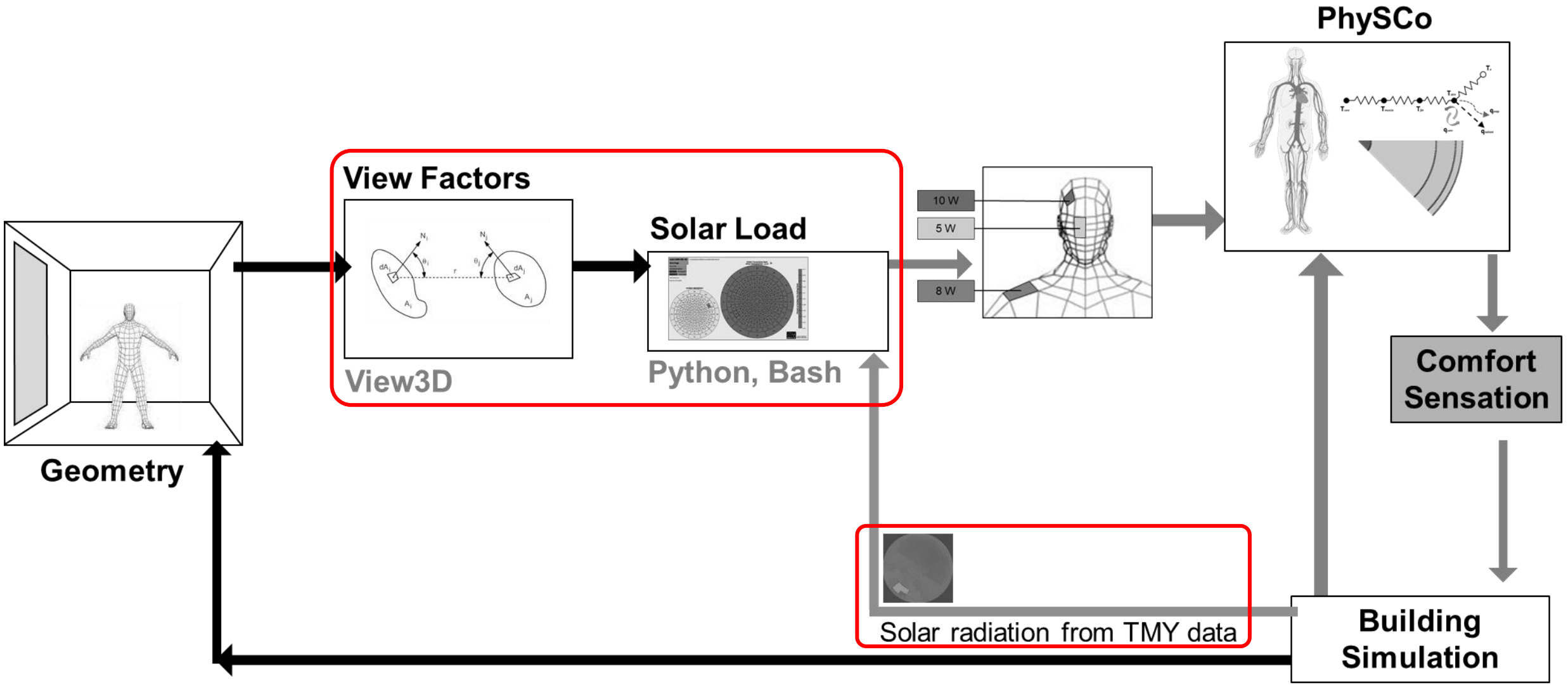


Solar load on human body

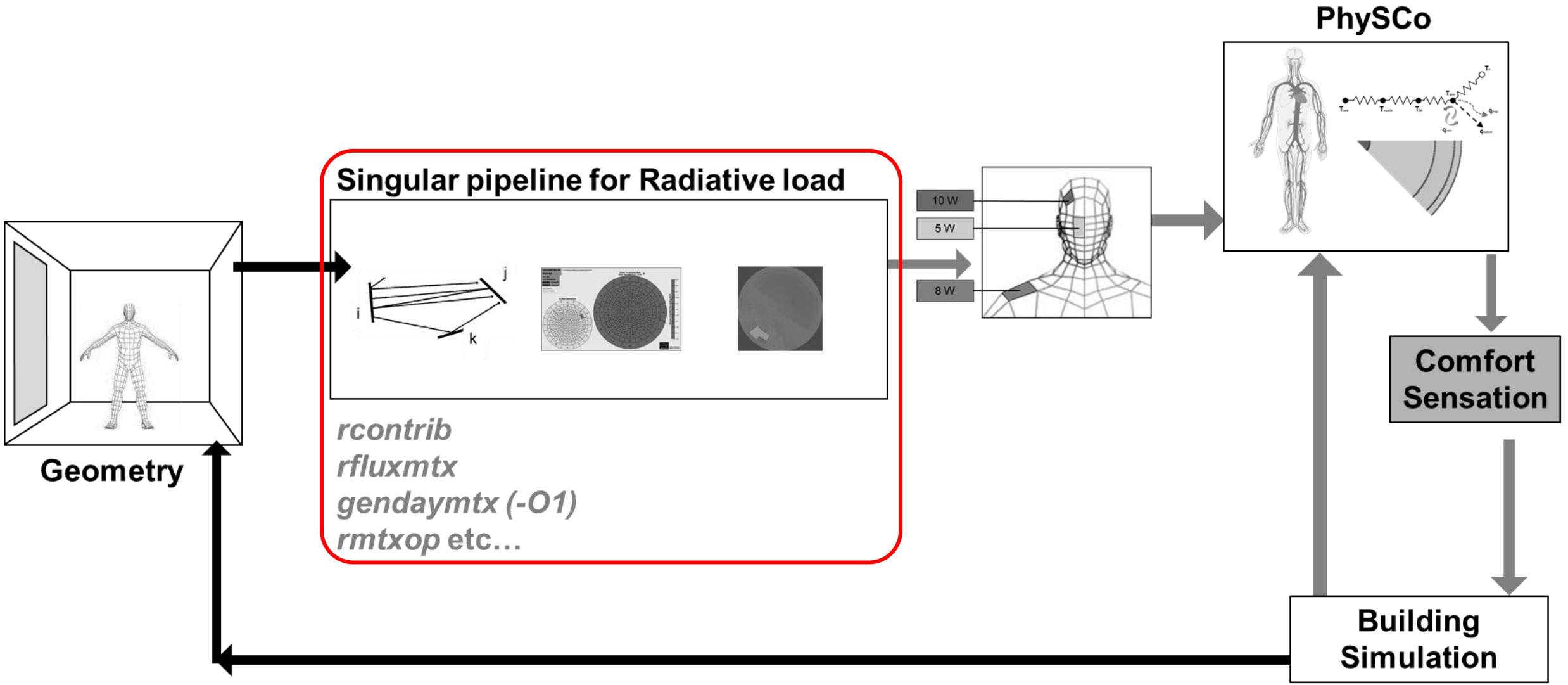


View Factors

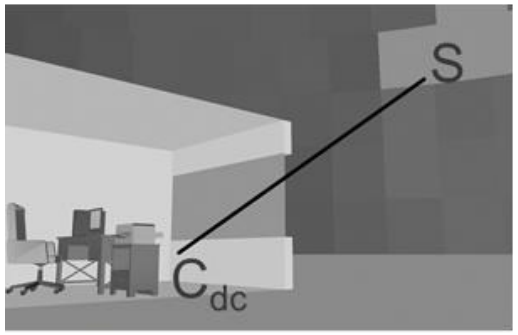
Background: Estimate short-wave radiation for Thermal Comfort Perception

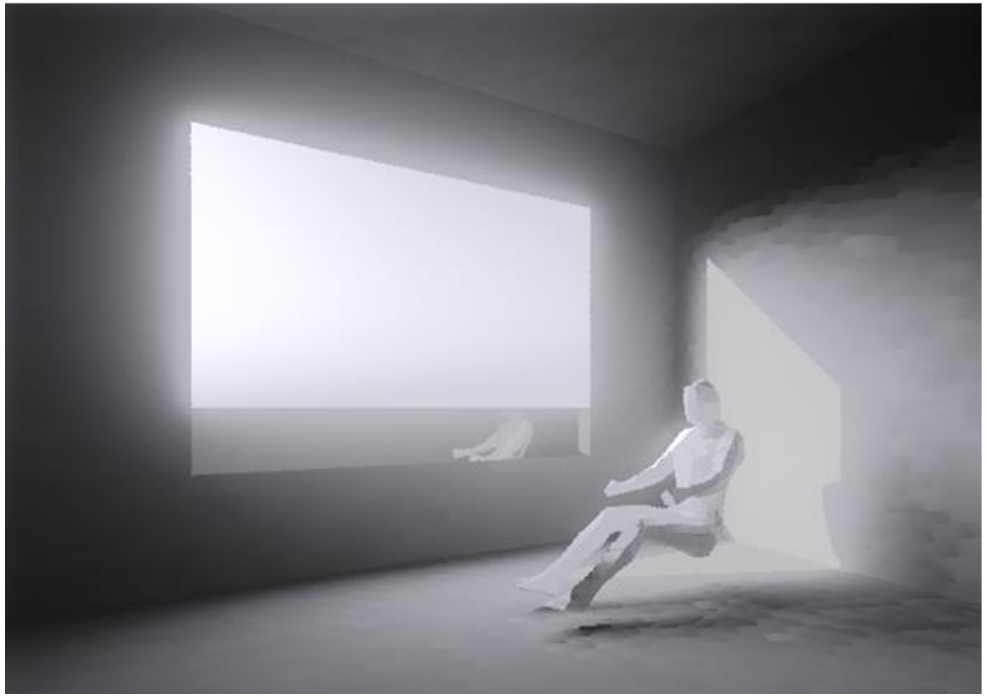


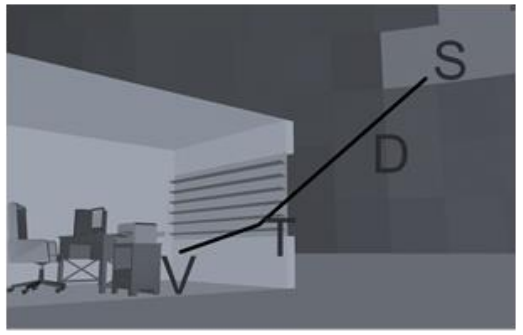
Background: Estimate short-wave radiation for Thermal Comfort Perception

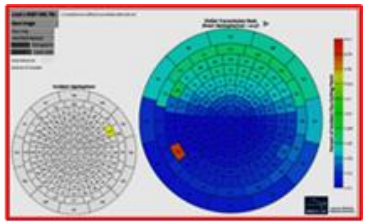


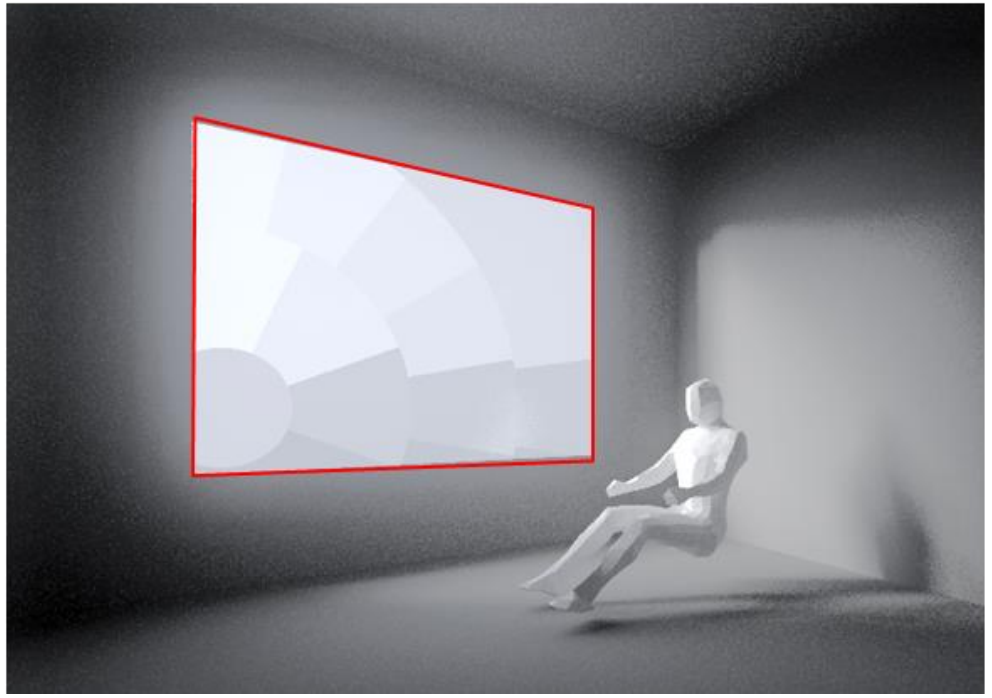
Initial work: Calculation of solar load on Manikin with and without CFS



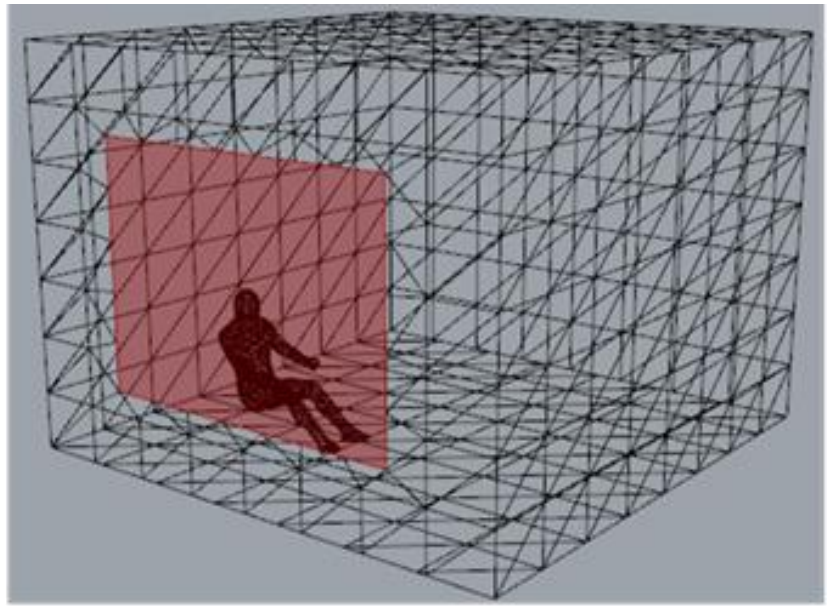


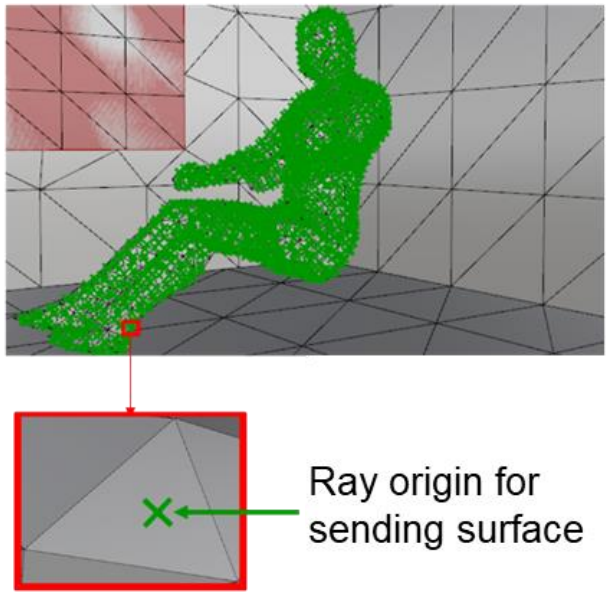


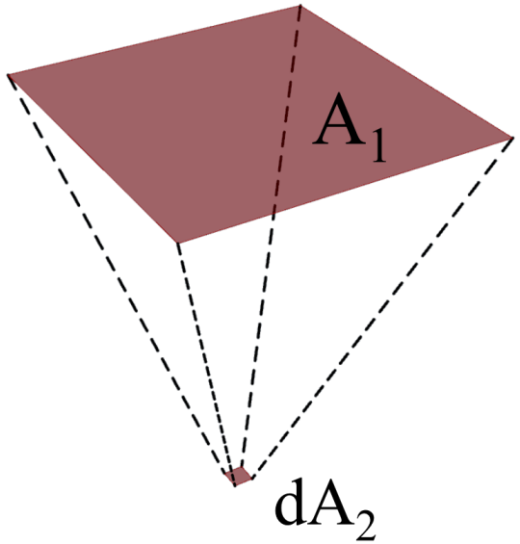




Calculation of contribution factors for annual simulations (-ab 1)

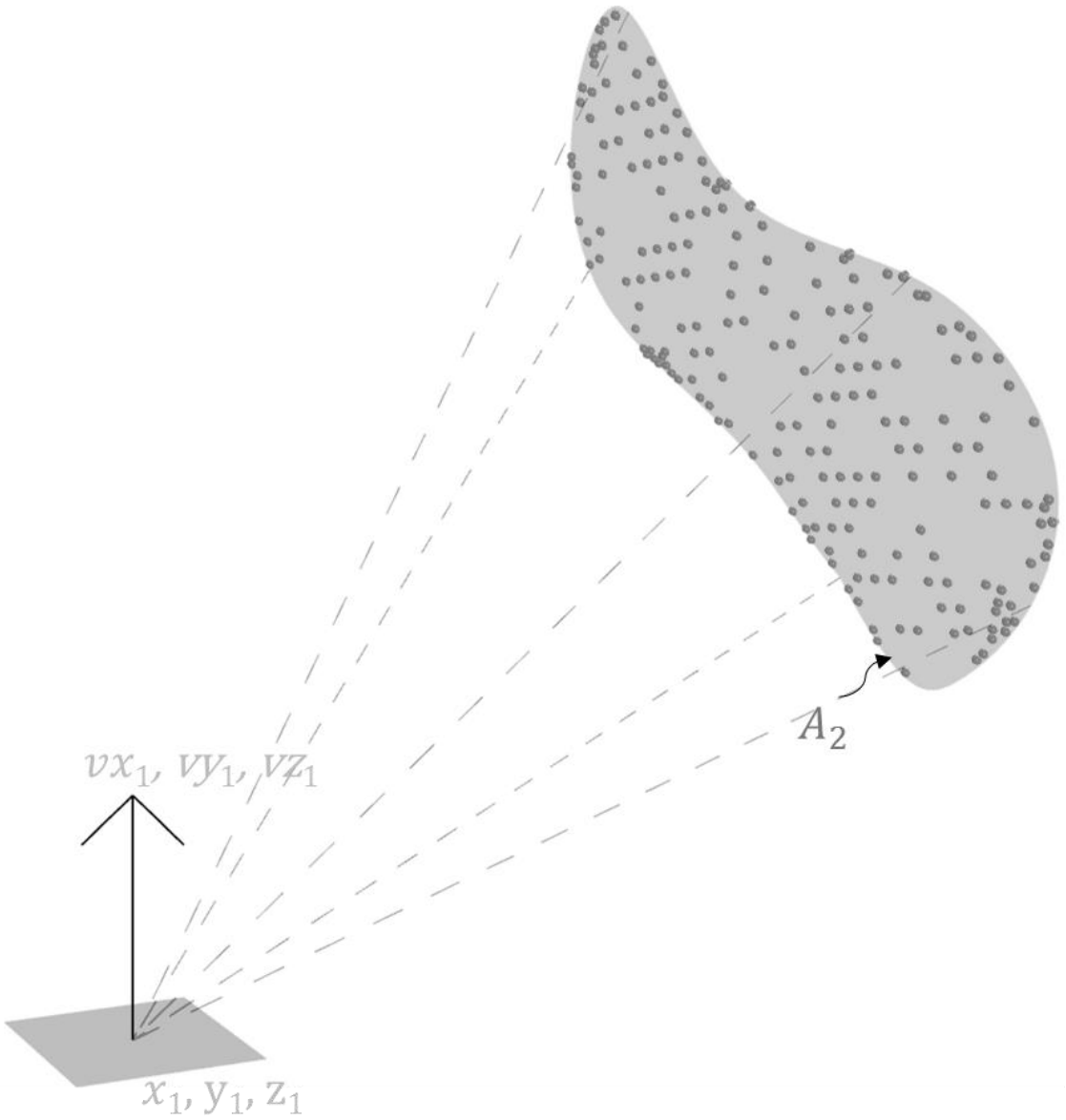


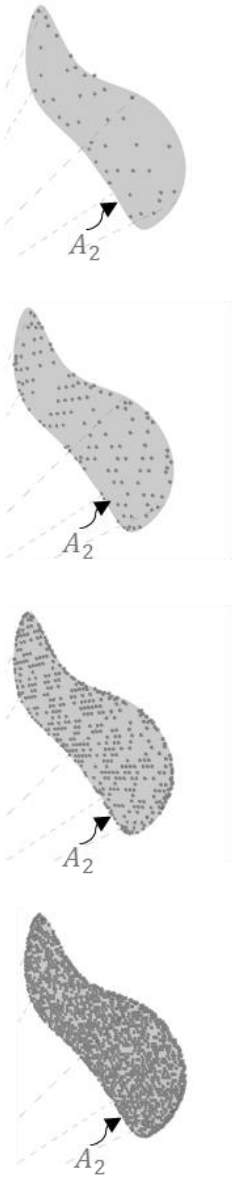




$$A_1 F_{1\text{-}d2} = dA_2 F_{d2\text{-}1}$$

Using Monte Carlo Method to calculate View Factors

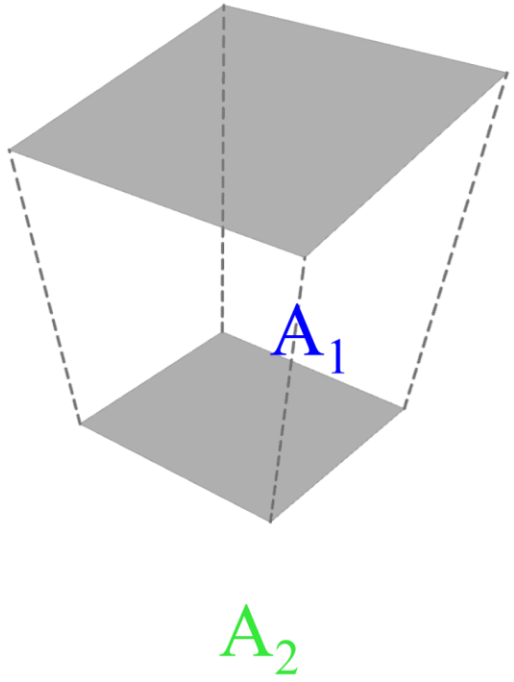


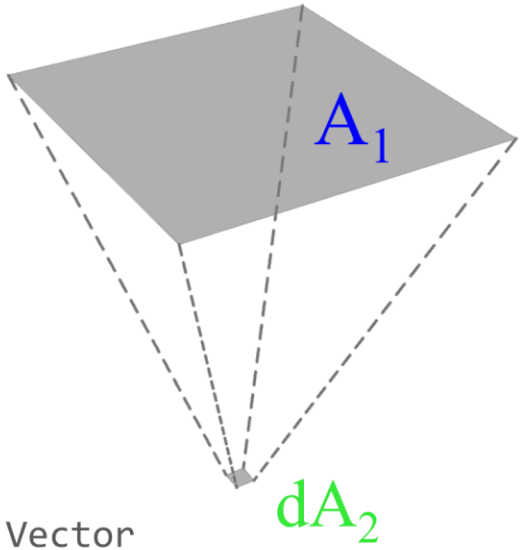


Higher sampling for convergence

Workflow with rfluxmtx (Radiance | Sender | Receiver)


```
rfluxmtx A1.rad A2.rad > result.txt
```






```
Location  Vector
0 0 0 0 0 1 |rfluxmtx -I -ab 1 - A1.rad > result.txt
```

Workflow with rfluxmtx: Limitations and bottlenecks

Multiple modifiers are expanded in commandline while invoking rcontrib.

```
#?RADIANCE
oconv -f rec.rad
rcontrib -fo+ -h+ -ab 1 -ad 100000 -lw 1
1 -b if(-Dx*0-Dy*0-Dz*-1,0,-1) -m r12 -k
if(-Dx*0-Dy*0-Dz*-1,0,-1) -m r14 -bn 1 -
if(-Dx*0-Dy*0-Dz*-1,0,-1) -m r16 -bn 1 -
if(-Dx*0-Dy*0-Dz*-1,0,-1) -m r18 -bn 1 -
if(-Dx*0-Dy*0-Dz*-1,0,-1) -m r110 -bn 1
```

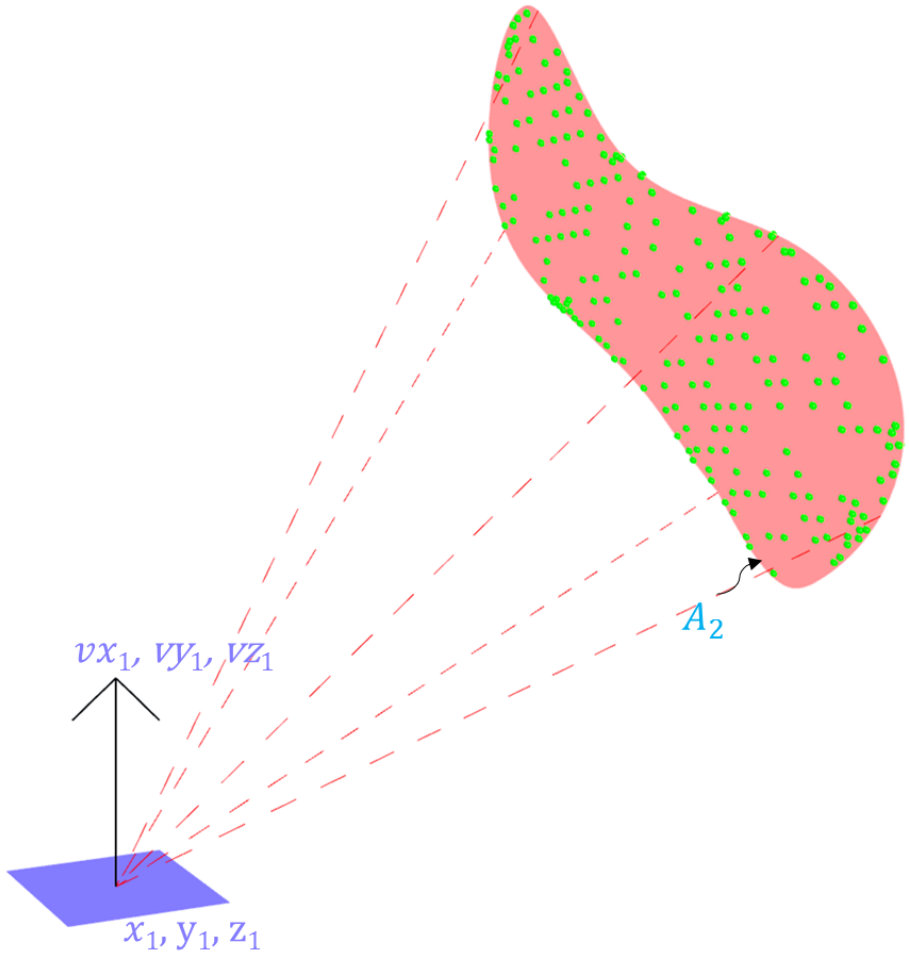
Single sending surface

rfluxmtx [–v][rcontrib options] { sender.rad | - } receivers.rad [-i system.oct] [system.rad ..]

Need to assign rfluxmtx params.*

```
#@rfluxmtx h=u
```

View Factors → Monte Carlo Method ← Rcontrib



$$\boldsymbol{rcontrib} \quad -I \; -ad\ N \; -lw\frac{1}{N} - m \; surfaceIdentifier \quad surfaceGeometry < point(s) > \quad \boldsymbol{ViewFactor} \times \boldsymbol{\pi}$$

View Factor with rcontrib: Simple(st) use-case: Point to Parallel Surface

SurfaceGeometry(sen.rad)

```
void plastic rectangle 0 0 5 1 1 1 0 0

rectangle polygon object 0 0 12

0 0 1   1 0 1   1 1 1   0 1 1
```

point(pts.txt)

```
0 0 0 0 0 1
```

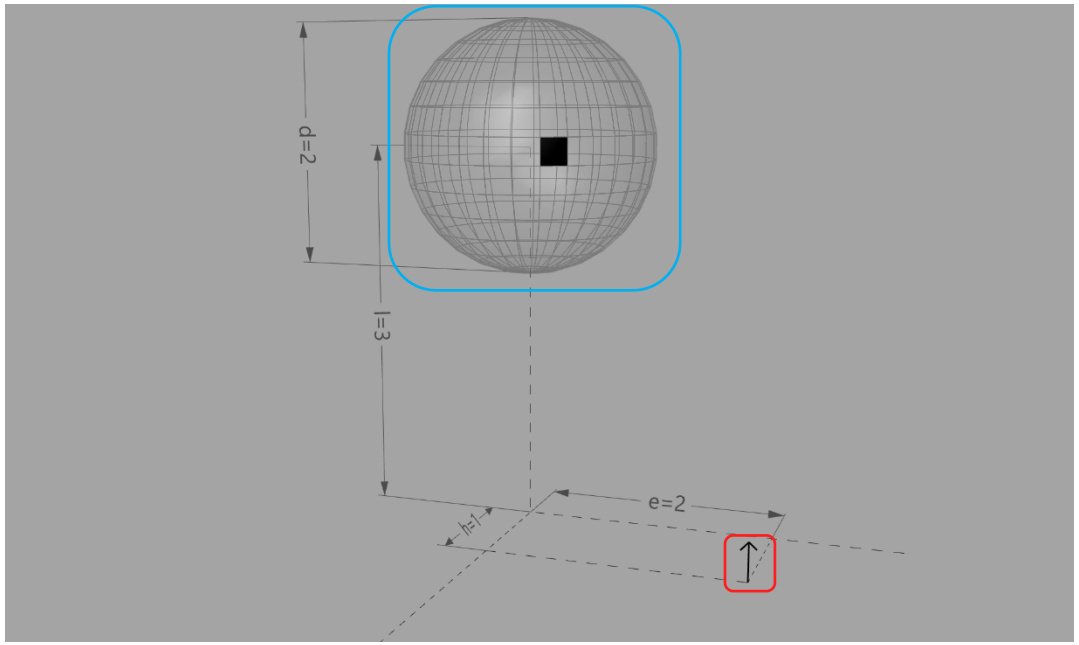
```
D:\>oconv sen.rad > geo.oct

D:\>rcontrib -h -w -I -ad 1000 -lw 0.001 -m rectangle geo.oct <pts.txt
4.325826e-01    4.325826e-01    4.325826e-01
```

surfaceIdentifier

Three Channel Output
(ViewFactor = 0.4325826/π)

Handling curved surfaces: Basis

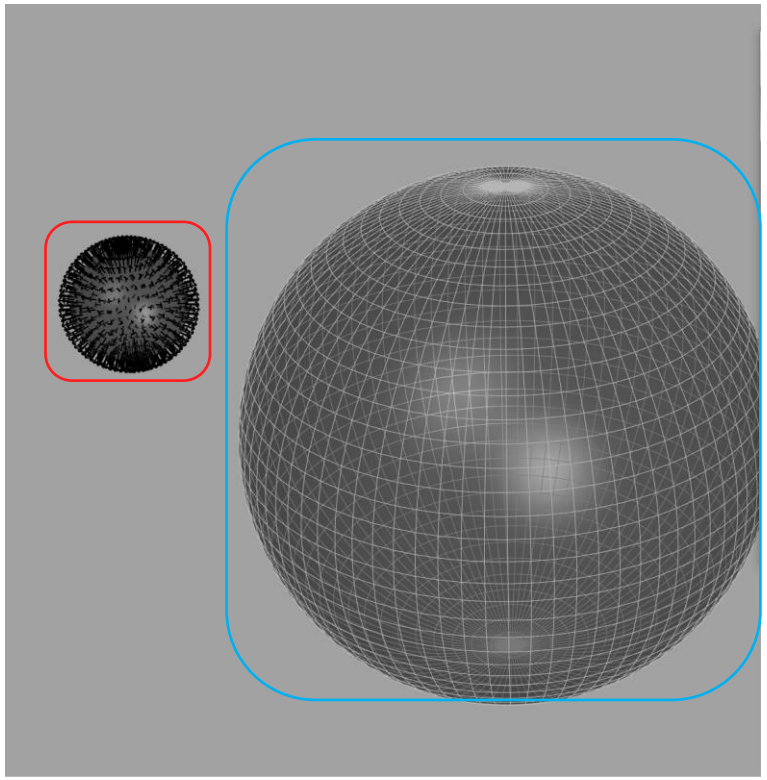


"The view factor from a surface I to a surface j is equal to the sum of the view factors from surface i to the parts for surface j"

$$F_{i-J} = F_{i-J1} + F_{i-J2} + F_{i-J3} + \cdots . + F_{i-JN}$$

$$\boldsymbol{rcontrib} \quad -I \; -ad\,N \; -lw\frac{1}{N} - m \; surfaceIdentifier \quad surfaceGeometry < point(s) > \quad \boldsymbol{ViewFactor} \times \boldsymbol{\pi}$$

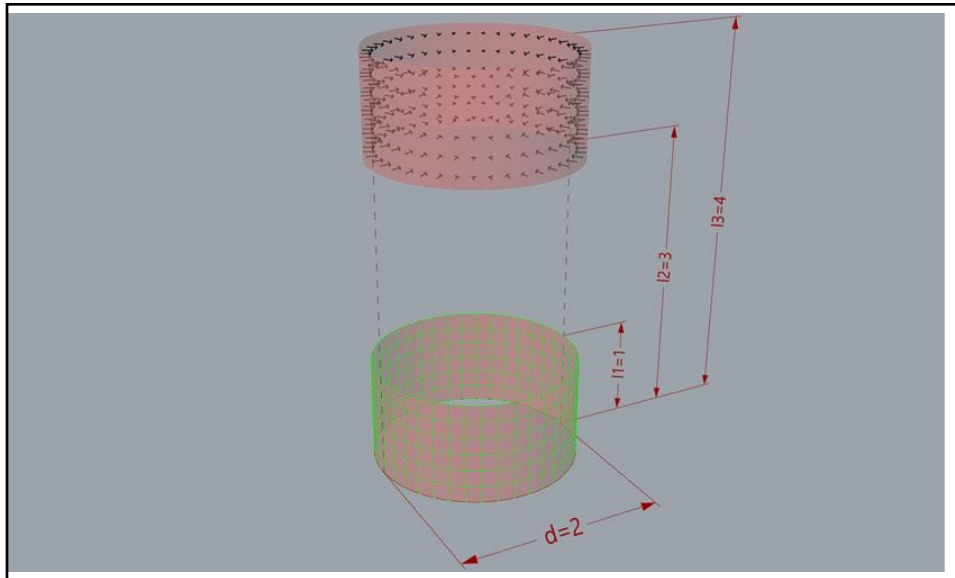
View factor between multiple curved surfaces



$$F_{i,j+k} = \frac{A_i F_{ik} + A_j F_{jk}}{A_i + A_j}$$

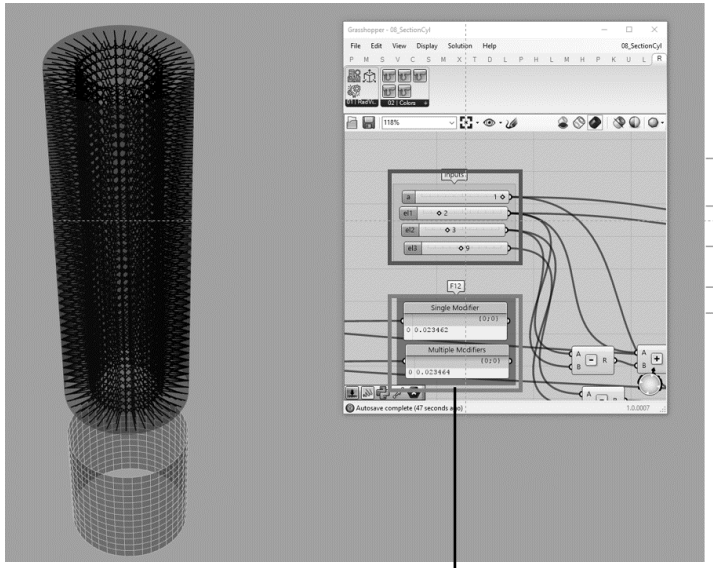
$$\boldsymbol{rcontrib} \quad -I \; -ad\,N \; -lw\frac{1}{N} - m \; surfaceIdentifier \quad surfaceGeometry < point(s) > \quad \boldsymbol{ViewFactor} \times \boldsymbol{\pi}$$

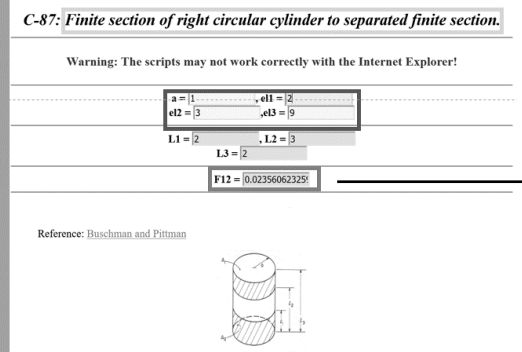
Validation against known analytical solutions



$$a = \frac{d}{2};\ L = \frac{l}{a};\ \ X(L) = \sqrt[2]{L^2 + 4}$$

$$F_{1-2} = \frac{1}{4(L_3 - L_2)}\left[\begin{array}{c} 2L_1(L_3 - L_2) + (L_3 - L_1)X(L_3 - L_1) - \\ (L_2 - L_1)X(L_2 - L_1) - L_3X(L_3) + L_2X(L_2) \end{array}\right]$$





C-87: Finite section of right circular cylinder to separated finite section.

Warning: The scripts may not work correctly with the Internet Explorer!

a = 1 , el1 = 2
el2 = 3 ,el3 = 9
L1 = 2 , L2 = 3
L3 = 2
F12 = 0.02356062325

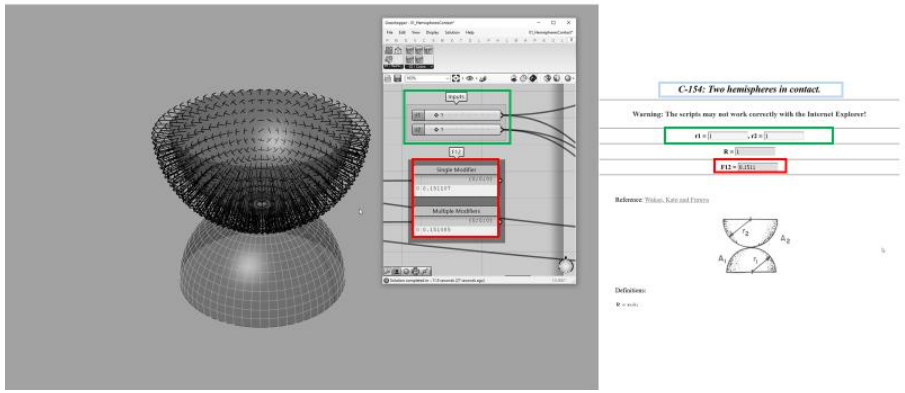
Reference: Buschman and Pittman

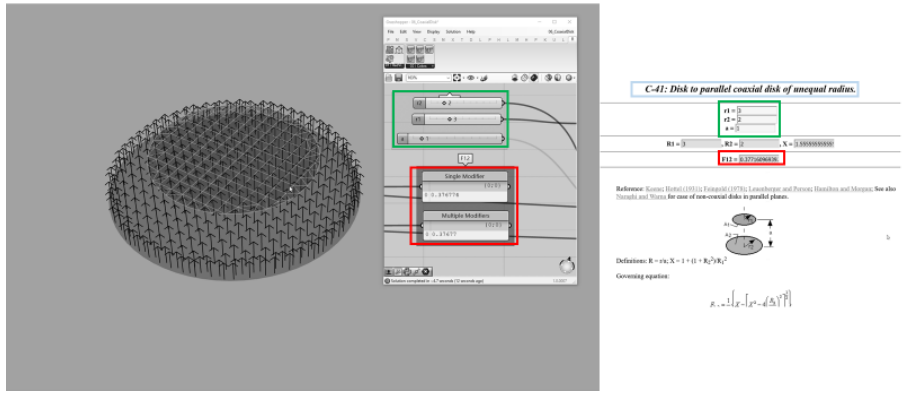
Definitions:

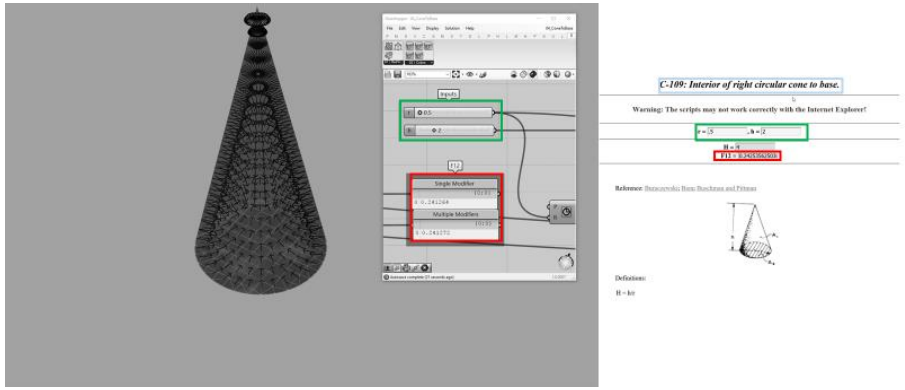
0.02356 (Analytical)

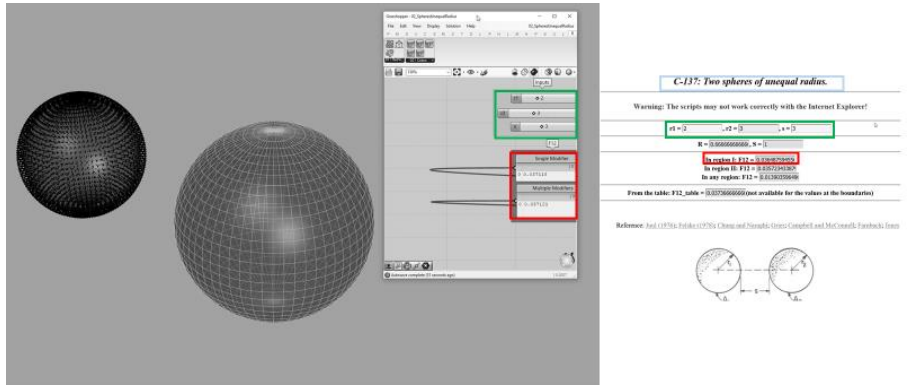
0.02346 (Radiance)

Validation against known analytical solutions

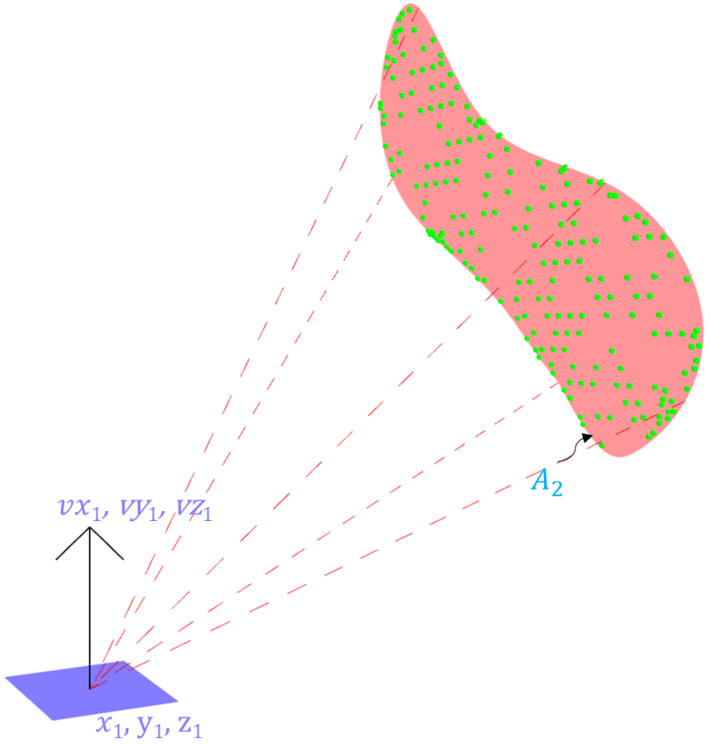


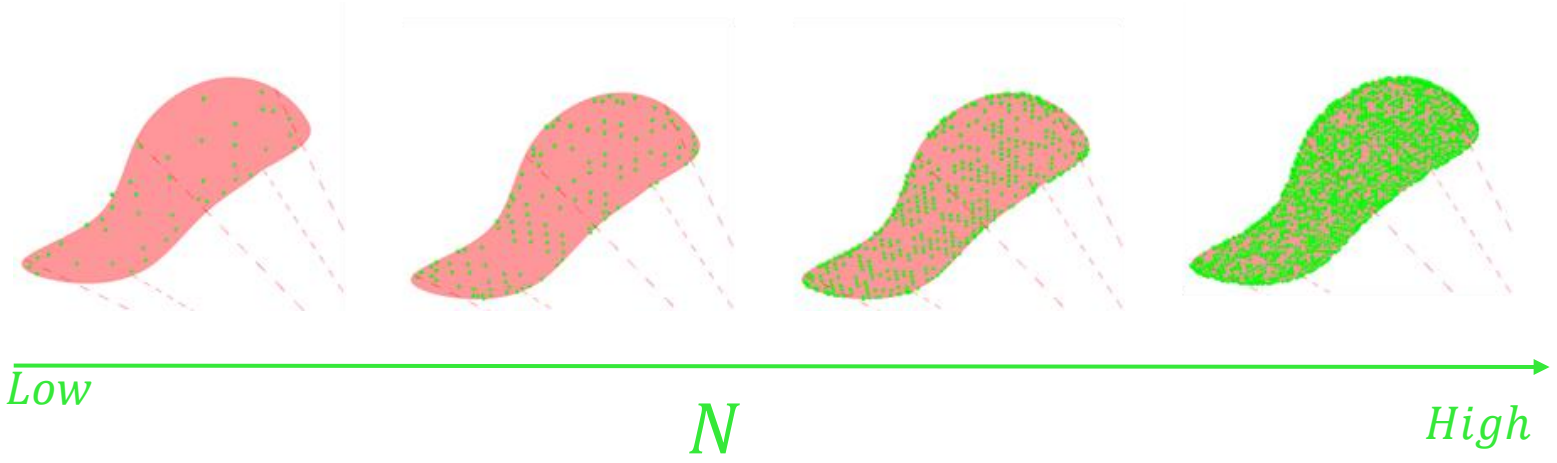






Relevant simulation parameters





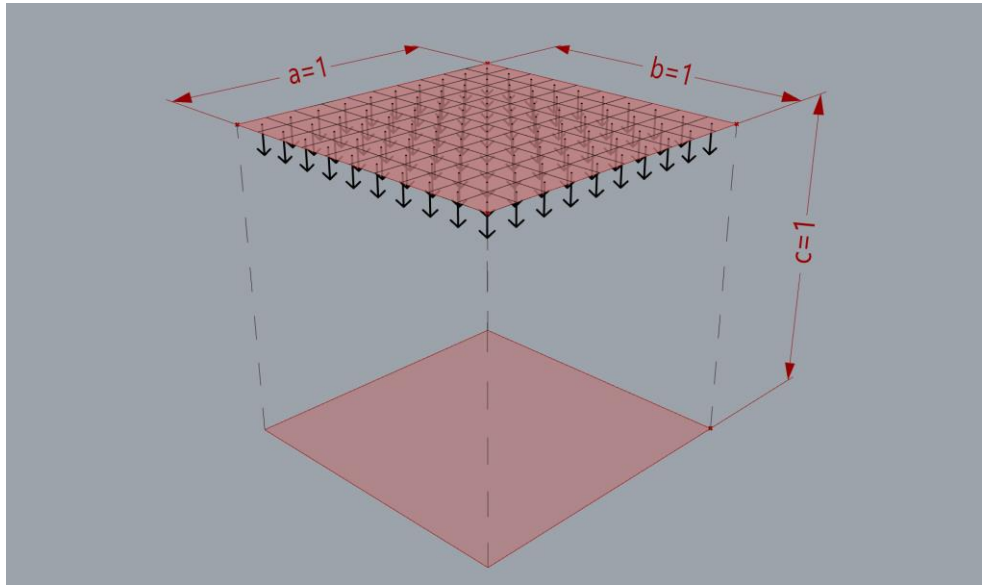
$$\textbf{rcontrib} \quad -I\,[-ad\ N\ -lw\frac{1}{N}]\,-m\ surfaceIdentifier \quad surfaceGeometry < point(s) > \textbf{ViewFactor} \times \pi$$

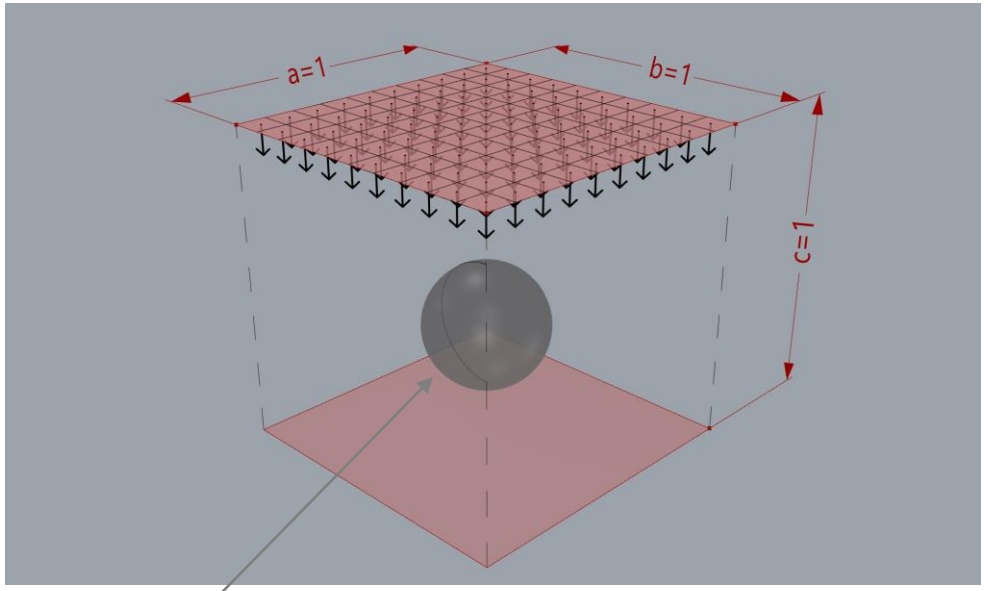
$$-ad\ N\ -lw\frac{1}{N}$$

Obstructed view factor calculation

octree

$$rcontrib \quad -I \; -ad\ N \; -lw\frac{1}{N} -m \; surfaceIdentifier(s) \quad surfaceGeometry < point(s) > \quad ViewFactor \times \pi$$



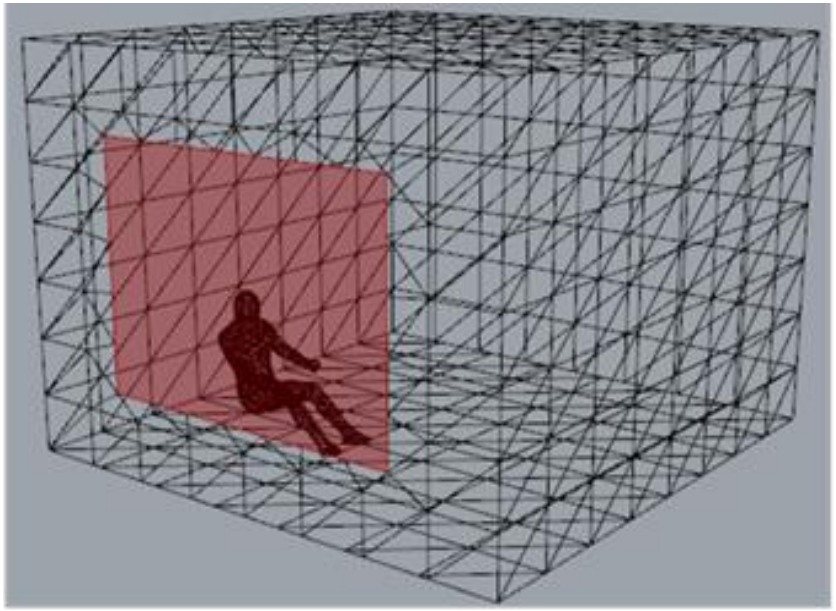


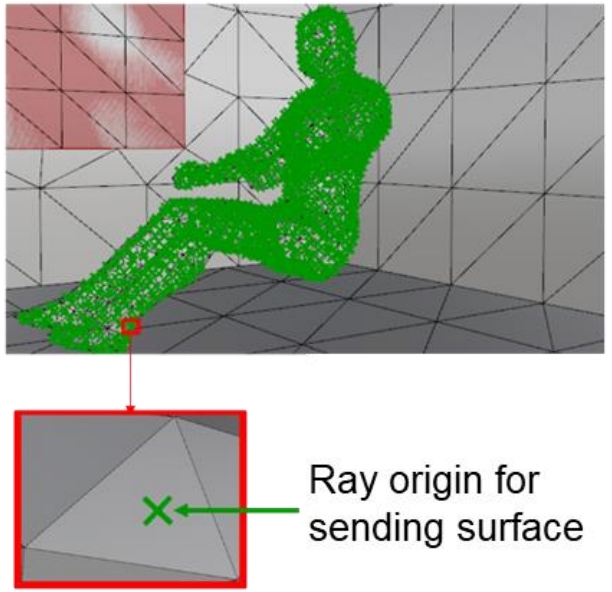
$$rcontrib \quad -I \; -ad\ N \; -lw\frac{1}{N} -m \; surfaceIdentifier(s) \quad surfaceGeometry + obstructing\ geometry < point(s) > \quad ViewFactor \times \pi$$

octree

Comparisons with a standard tool (View3D)

Incident radiation from fenestration to manikin





Average Simulation time (seconds)

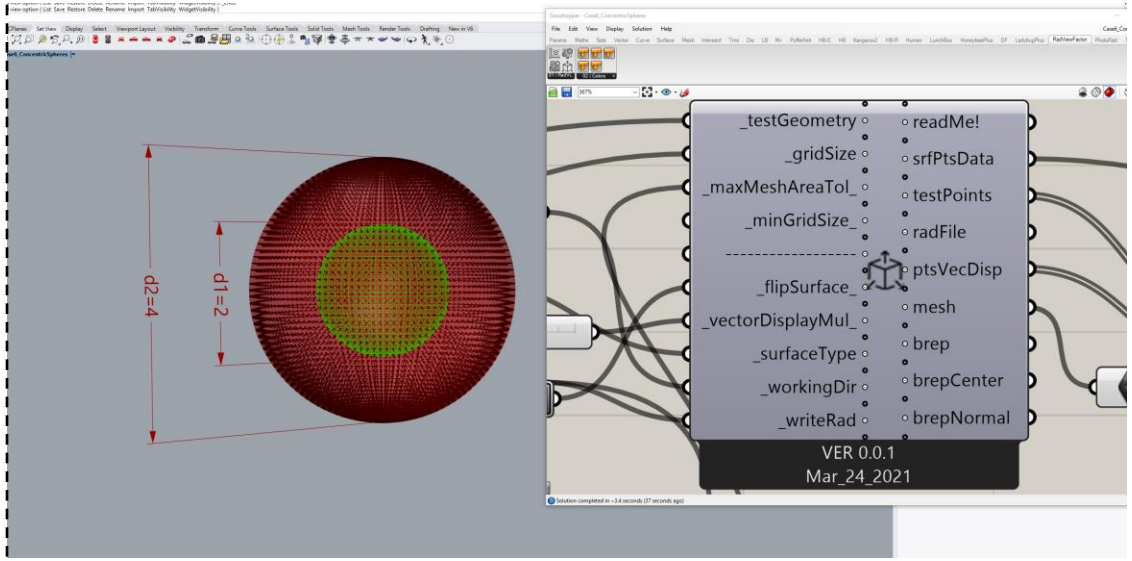
	View3D	Radiance
Total mesh faces	2126	
Mesh faces (manikin)	1336	
View factors (to be calculated)	1336	
View factors (considered)	4519876	1336

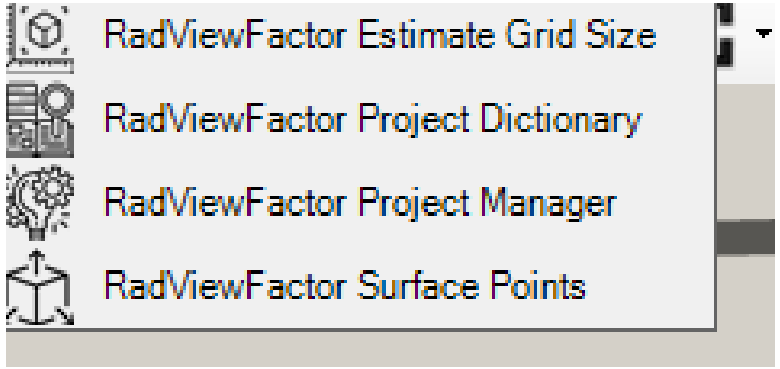
Average Simulation time (seconds)

	1 Proc.	2 Proc.	4 Proc.
View3D	322.5	N.A.	N.A.
Radiance	14.2	7.8	4.1

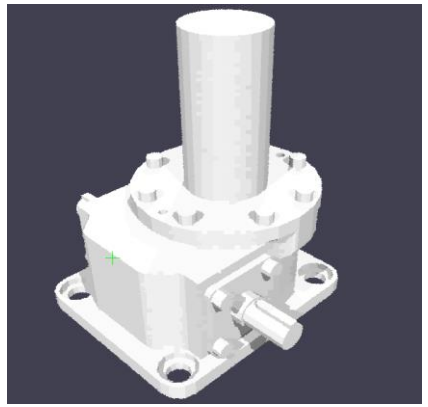
Ongoing work/collaboration

Rhino/Grasshopper Plugin To Handle Geometry/Radiance





Practical Applications in Thermal Engineering



Thank you! Questions or comments?

Sarith Subramaniam (sarith@rhrk.uni-kl.de)