Calculating the IR radiation distribution in a large 3D scene

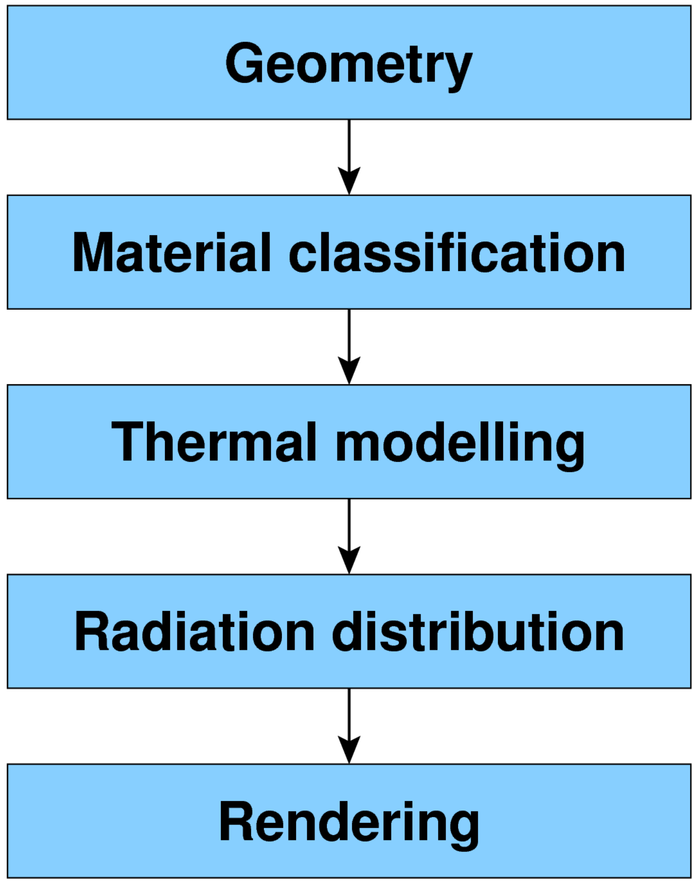
Stefan Hoch, DLR

ITBM&S 2010, Ettlingen

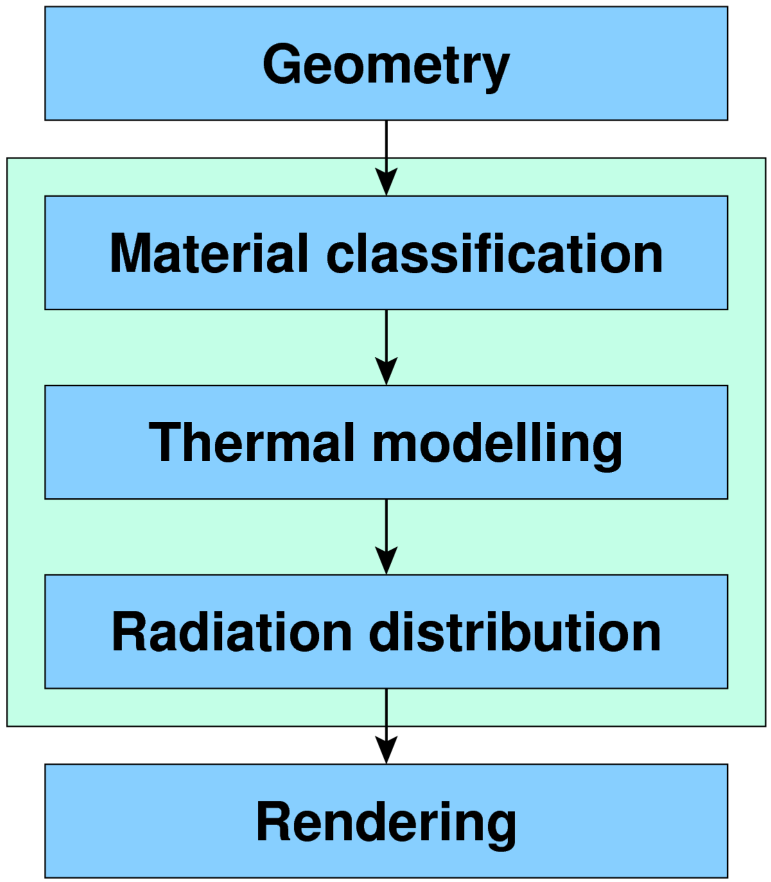
DLR Deutsches Zentrum
für Luft- und Raumfahrt e.V.
in der Helmholtz-Gemeinschaft

MBDA
MISSILE SYSTEMS

Workflow


Geometry
Material classification
Thermal modelling
Radiation distribution
Rendering

Workflow


Geometry
Material classification
Thermal modelling
Radiation distribution
Rendering

Evaluation of radiation distribution (RD) / rendering

- Often calculation of RD and rendering is just one step
- But radiation can be split into
  - a **view-independent** component,
    which is calculated **once** for a scene → ***radiosity method***
  - a **view-dependent** component,
    which has to be **re-calculated** each time the viewer changes position or direction → ***raytracing***

Radiosity Method

- Objective: calculation of the radiation distribution
- Assumptions: 1) Every reflection is **Lambertian**
  2) Geometry is **discretized** into small polygons
  (= *patches*)
- Advantages: 1) **Isotropic** problem
  2) Integration is replaced by a physically sensible
  summation (over patches)

Radiosity Method

- Rendering equation turns into **radiosity equation**

$$B_i = E_i + \rho_i \sum_j F_{ij} B_j$$

- B_i = **radiosity** of patch *i*
- E_i = **emittance** of patch *i*
- ρ_i = **reflectivity** of patch *i*
- F_{ij} = **form factors**: "how much does patch *i* see of patch *j* ?"

For *N* patches in the scene, 2*N*² form factors have to be calculated!

Implementation of radiosity method

- **Radiosity algorithm from scratch?**
  - input/output data format ?
  - several algorithms have to be implemented (occlusion, form factors, numerical solution of the radiosity equation)

→ ***Use existing software, RadThermIR's signature solution***

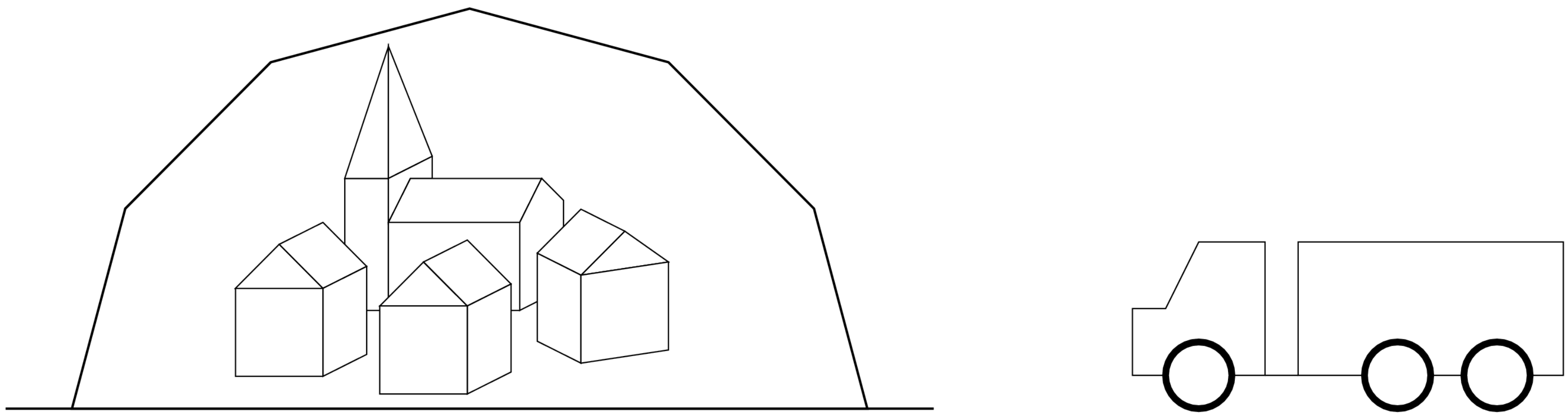
+ thermal solver output can be processed directly

+ all necessary algorithms contained

But how can large scenes be loaded and computed by RadThermIR?

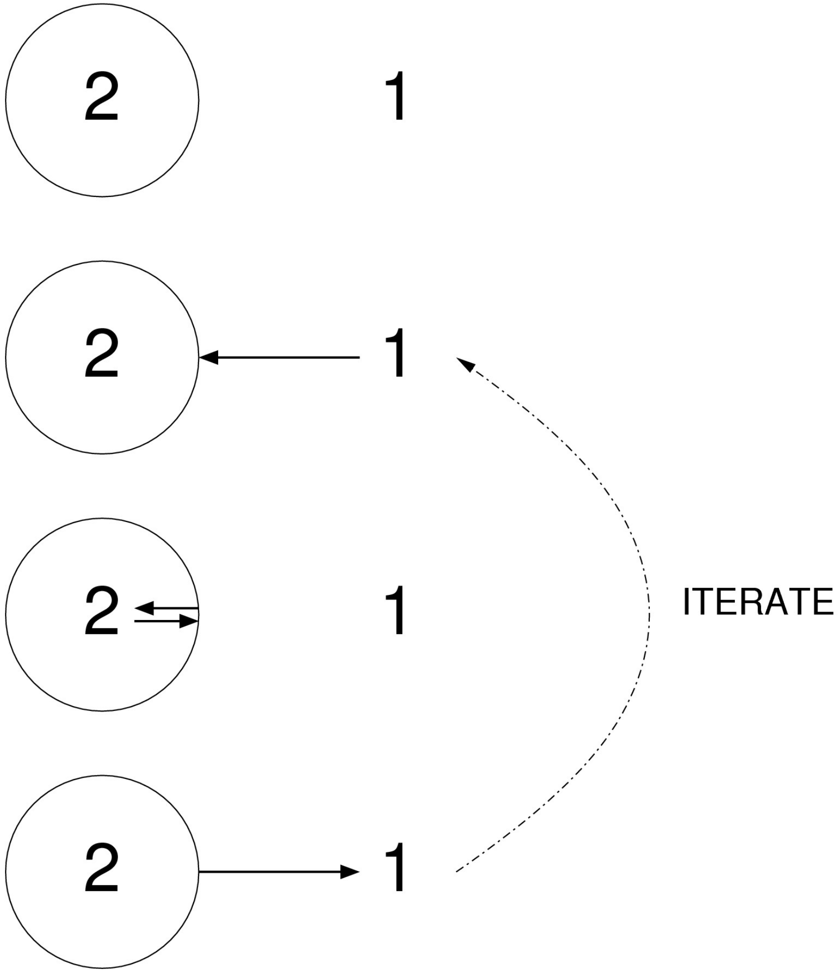
Clustering method

- subdivision of the scene
- fine-structured regions are replaced by domes (= clusters)
- these clusters are modeled separately in RadThermIR (thermal and radiative)
- use an iterative scheme for interaction of subscenes and large scene



Clustering method

- Scheme of interactions:



Clustering method

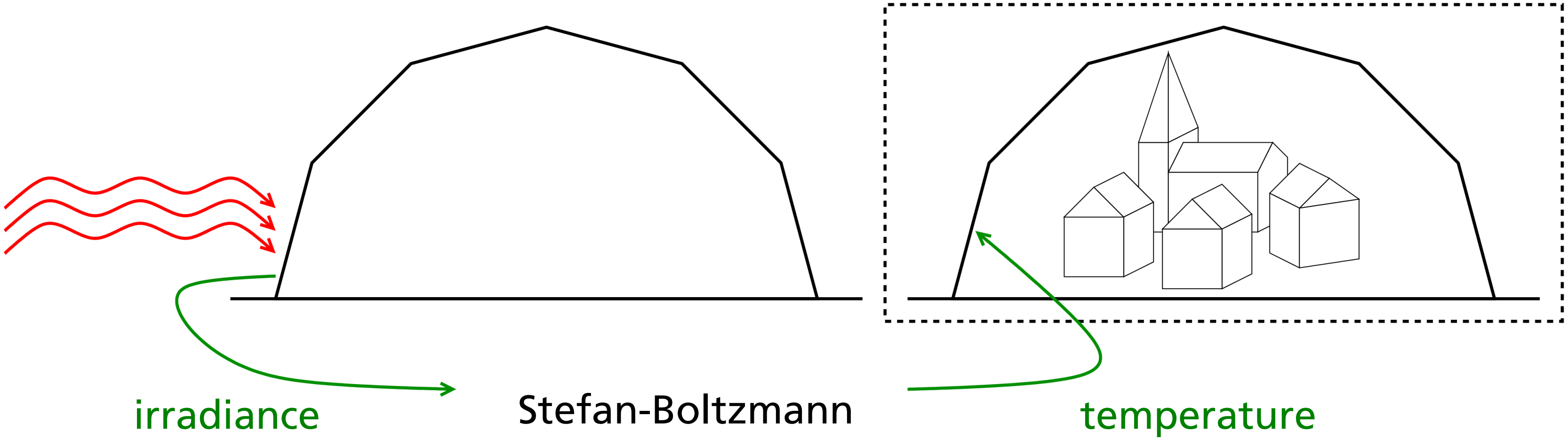
- What are the benefits?
  - number of patches per calculation is reduced
  - less form factors have to be computed
  - parts of the scene can be created independently of the rest

Clustering method

- **Thermal solving**
  - domes are black bodies
  - thermal radiation received by each dome patch is read off via tdfio library function
  - translate these irradiances into temperatures via Stefan-Boltzmann law
  - set these as initial temperatures for the dome in the cluster
  - after processing the cluster, temperatures are transferred back to the main scene in the same way

Clustering method

- transferring thermal radiation

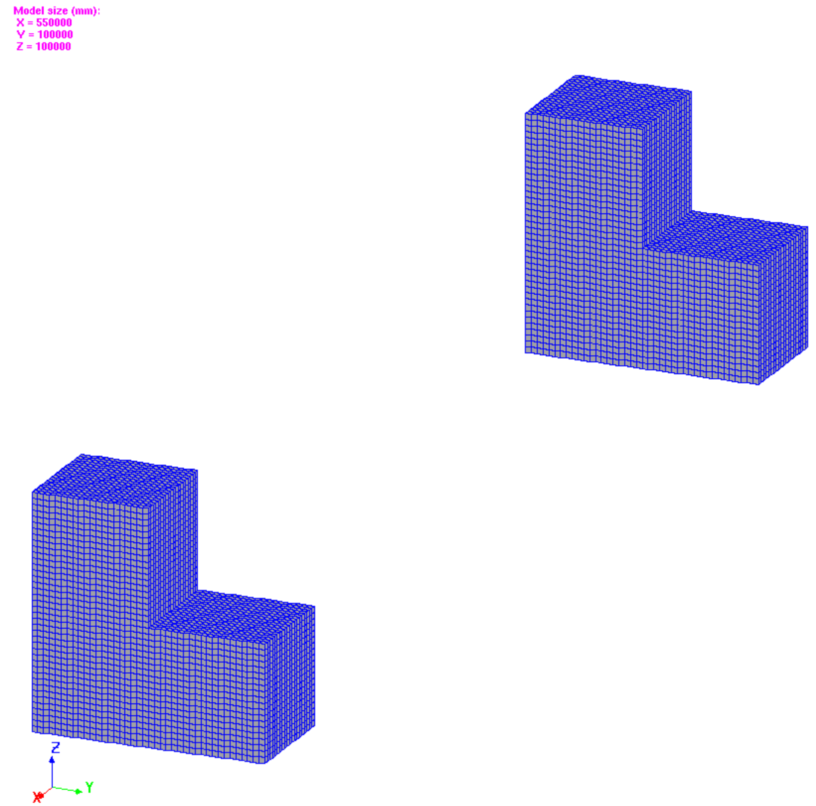


Clustering method

- **signature solution**
  - proceed analogously to the case of thermal solving
  - open problem: spectral radiances on the dome cannot be set by tdfio directly

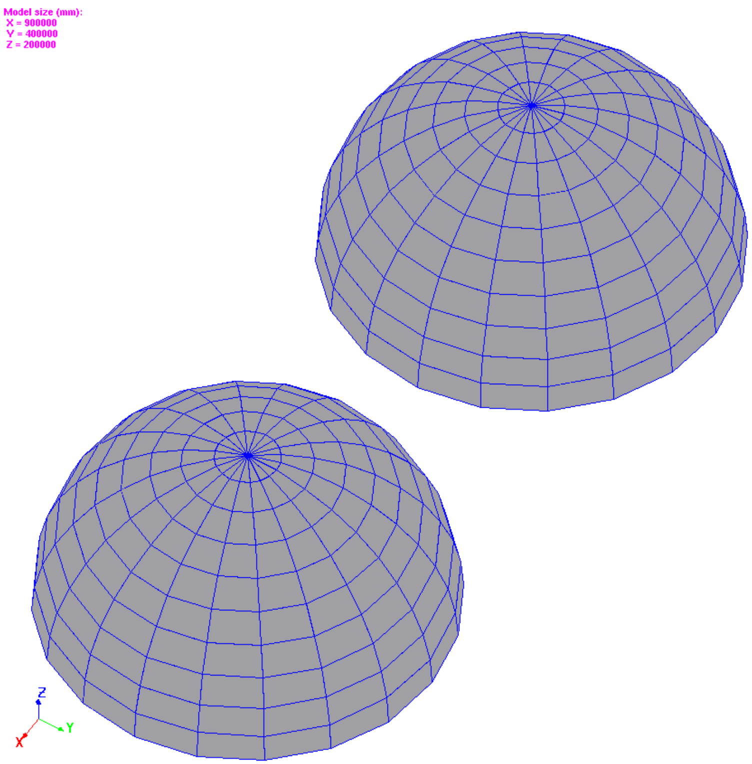
Clustering method

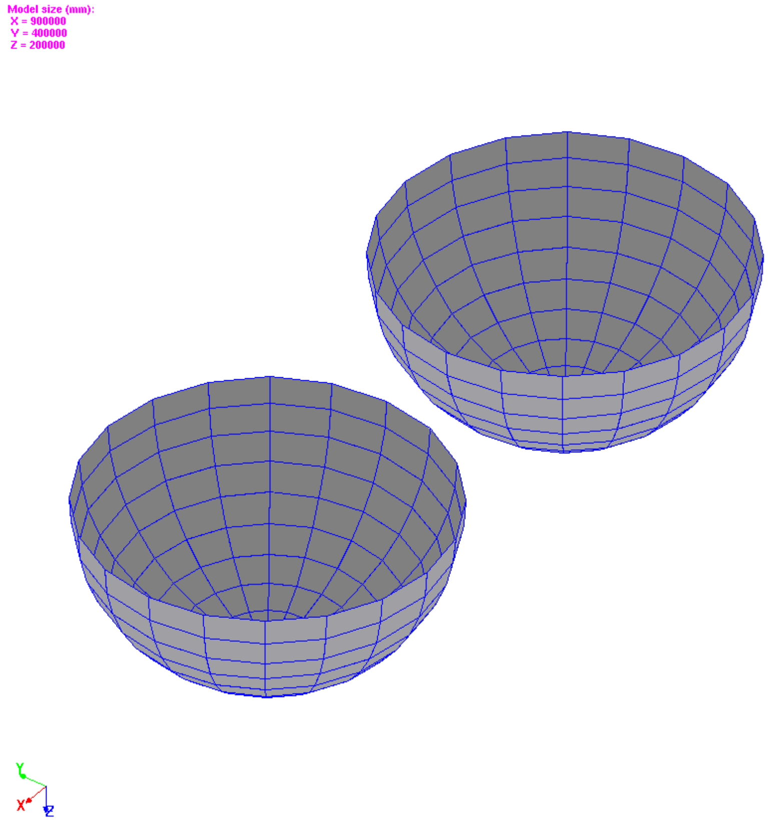
- original scene
- 2 CUBI models,
  scaled by factor 100
  (i.e. base 100 m x 50 m)
- Distance 500 m



Clustering method

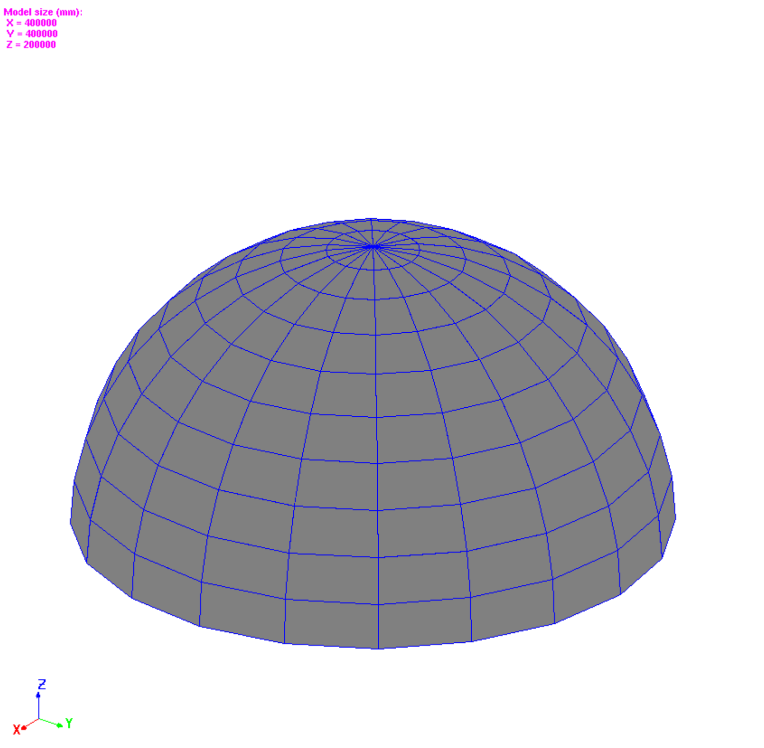
- **clustered scene**, 2 domes representing the geometry

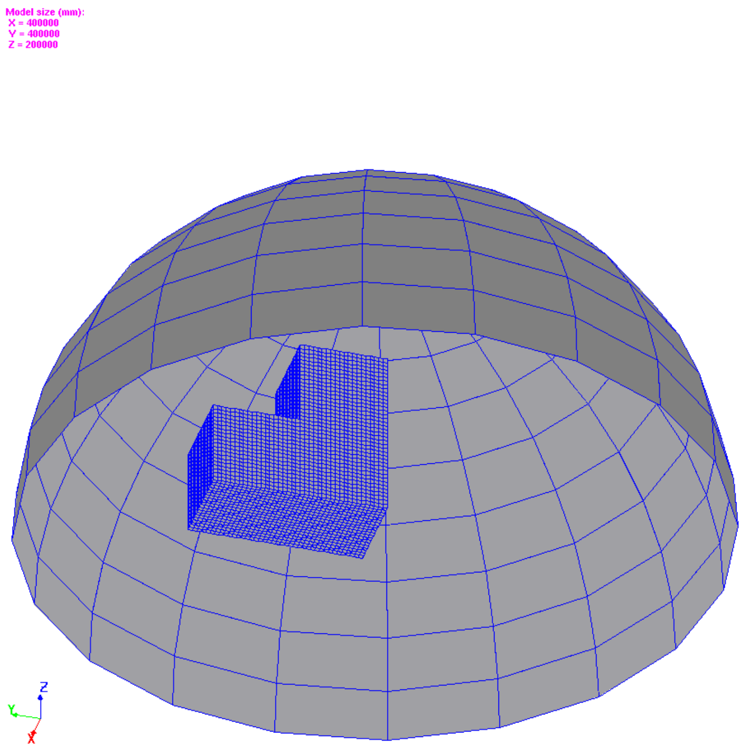




Clustering method

- **one of the clusters** with geometry inside





Summary and Conclusion

- IR scene simulation requires knowledge of RD
- thermal influences on RD are modeled by RadThermIR thermal solver and signature solution
- clustering method is useful to calculate large scenes by subdividing them

What has to be done?

- comparison of temperatures and RD in original vs. clustered scene
- automation (placement of clusters, mapping to the original scene)
- integration into workflow

Thanks to
E. Lindermeir and P. Schätz (LFK)