Deterministic radio propagation modeling and ray tracing

Deterministic ray models (1/3)

"Numerical simulation of multipath propagation according to GTP"

- Ray models compute (some of the) rays linking the two terminals through free space propagation and multiple interactions with buildings. The geometry and the field of such rays must satisfy GTP rules

- Interactions, i.e. reflections, diffractions, scatterings are also called "***events***". Usually all rays experiencing up to a pre-set number of events, N_{ev} are computed

- N_{ev} also defines the so-called ***prediction order***

- Ray models can be fully 3-Dimensional (3D) or can resort to simplified 2-dimensional modelling (2D)

- Ray models's output can be the field (before the Rx antenna) or the received signal (after the Rx antenna). In the latter case, Rx antenna parameters must be input to the model.

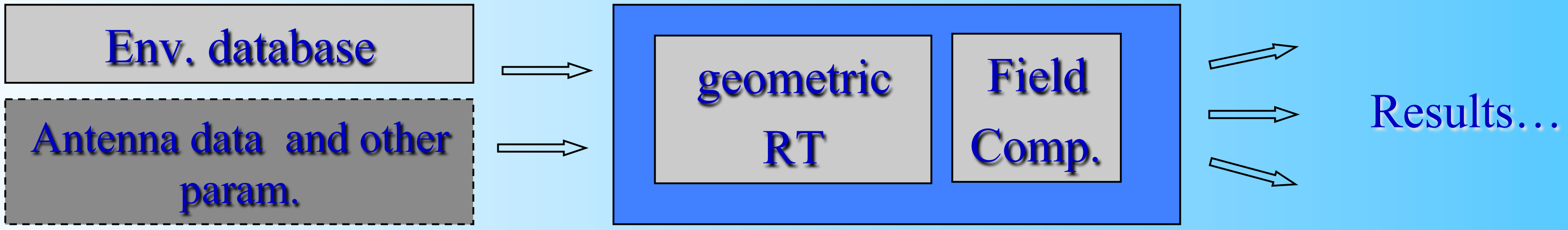
Deterministic ray models (2/3)

- Sometimes beams (tubes of flux) instead of "real" rays are considered. Rays have a null transverse dimension. Beams have a finite transverse dimension and therefore a space discretization is adopted ➔ a limit to space resolution is posed.

- Models adopting rays are usually referred to as ***Ray Tracing*** (***RT***) models. Models adopting beams are usually referred to as ***Ray*** (or ***beam***) ***Launching*** (***RL***) models

- Ray models require in input a detailed description of the environment (***environment database***)

- Usually two different computation steps can be identified: geometric ray tracing and field computation. The former <u>is by far the most CPU-time consuming</u>



Deterministic ray models (3/3)

Advantages over empirical-statistical models:

- 🙂 Multipath simulation → multidimensional prediction
- 🙂 High accuracy
- 🙂 Great versatility

Drawbacks:

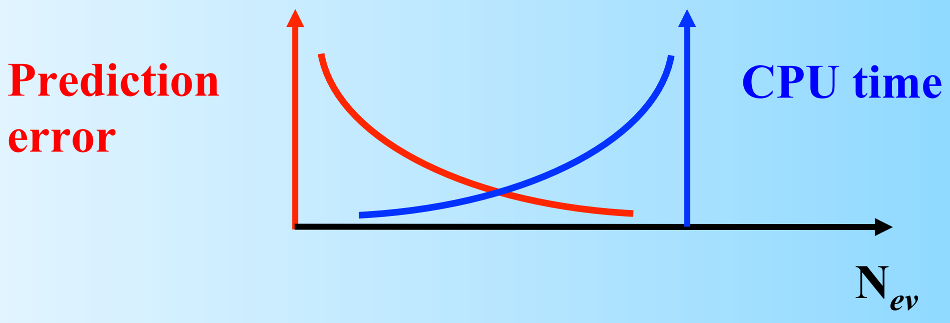
- 🙁 Environment database "cost"
- 🙁 High CPU time (of the order of hours for urban prediction over a cell or area)

Performance

- Prediction accuracy first of all depends on the accuracy of environment database

- Prediction accuracy also depends on N_{ev} : the greater N_{ev} , the better the accuracy. Usually N_{ev} = 3-4 for outdoor, N_{ev} = 2 for indoor prediction are chosen.

- Unfortunately, CPU time grows more than linearly with N_{ev}



Environment databases

- Deterministic models require detailed environment databases

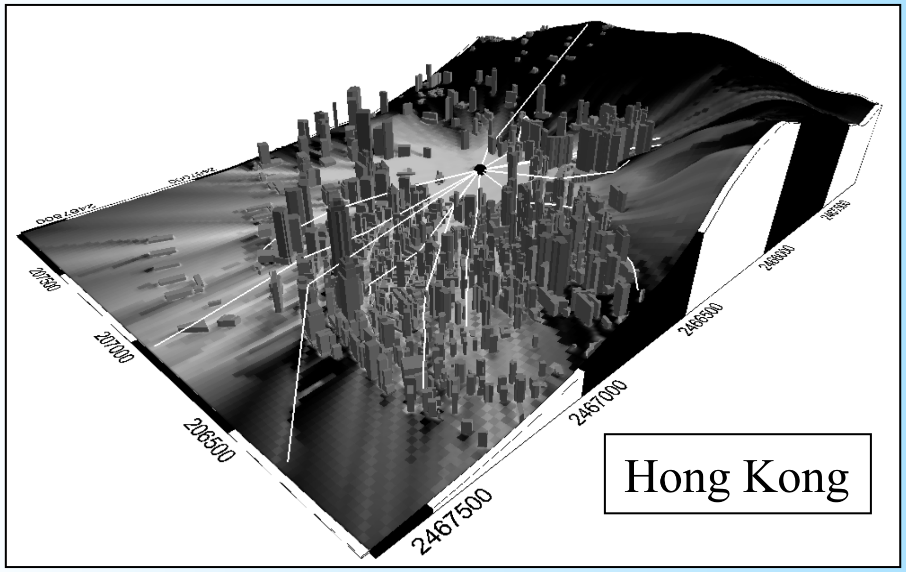
Geometric

Terrain

Buildings (vegetation)

Material param's

Electro magnetic



Environment database issues

- Good environment databases can be expensive (not true for the future)
- Environment databases are difficult to handle by non-expert personnel
- Environment databases are often not very accurate (precision of 0.5m for horizontal coordinates and 1 to 5 m for vertical coordinates)

⇩

☹ Major limitation for ray models

Simplified ray models

- "Simplified ray models" are derived from deterministic ray models. however, the geometry of the multipath pattern is somehow simplified to ease the computational burden while still taking major rays into account

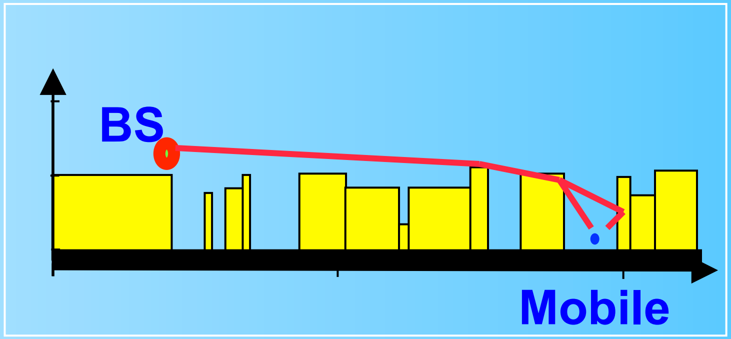
- Usually, the problem is simplified by identifying planes where most rays lie, thus resorting to a 2-dimensional (2D) approach. Such simplifications however are not rigorous since multipath propagation in urban environment is intrinsically 3D

- Of course 2D computation is much simpler and faster than 3D computation

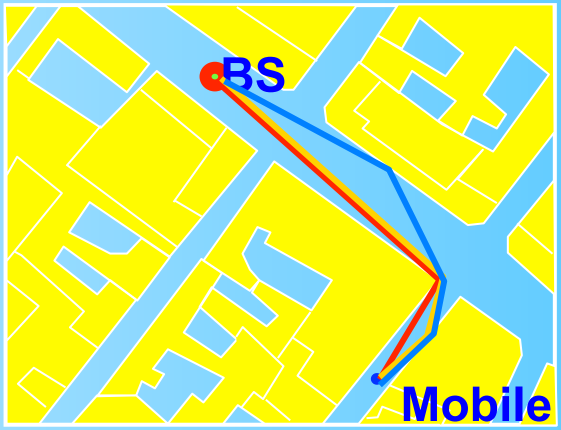
- 2 different approaches can be identified: ***2D+2D*** (lateral plane+vertical plane ray tracing) and ***quasi-3D*** (Vertical Plane Launch, VPL ray tracing)

Ex: 2D+2D ray model

- In the vertical plane either a real 2D ray tracing or an ORT model is applied



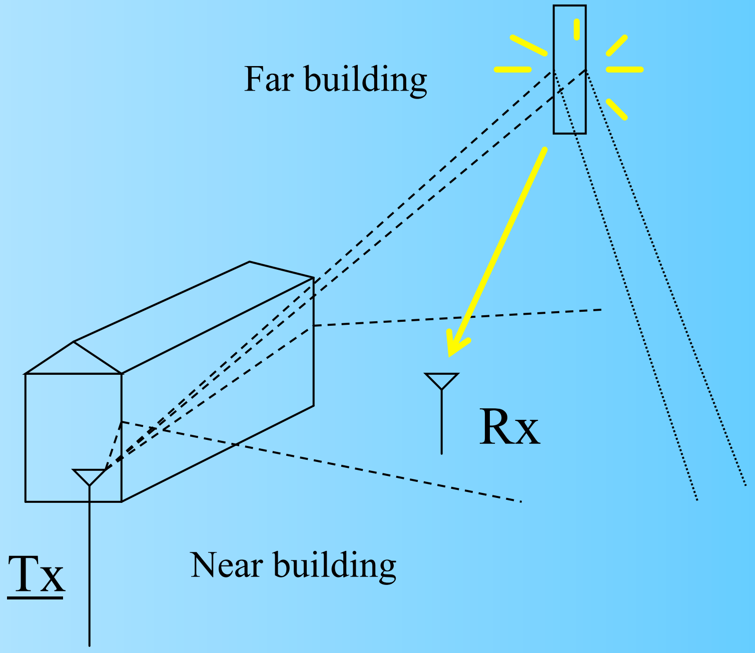
- In the lateral plane a 2D ray tracing is performed



Deterministic ray models: trends

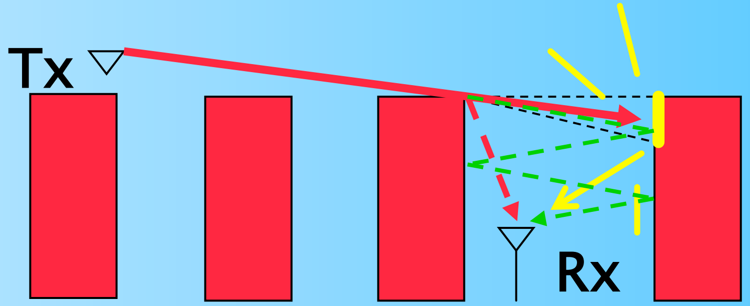
- Speed up techniques to decrease CPU time, e.g. parallelization on Graphical Processing Units
- A-priori determination of computation parameters, “active” database area and ray selection
- Exploitation of the multidimensional prediction potential
- Integration of statistical elements into the deterministic model such as diffuse scattering

Impact of diffuse scattering



·Diffuse scattering dramatically increases signal contribution from far obstacles

·Diffuse scattering is crucial for time and angle dispersion



·Diffuse scattering is important in roof to street propagation

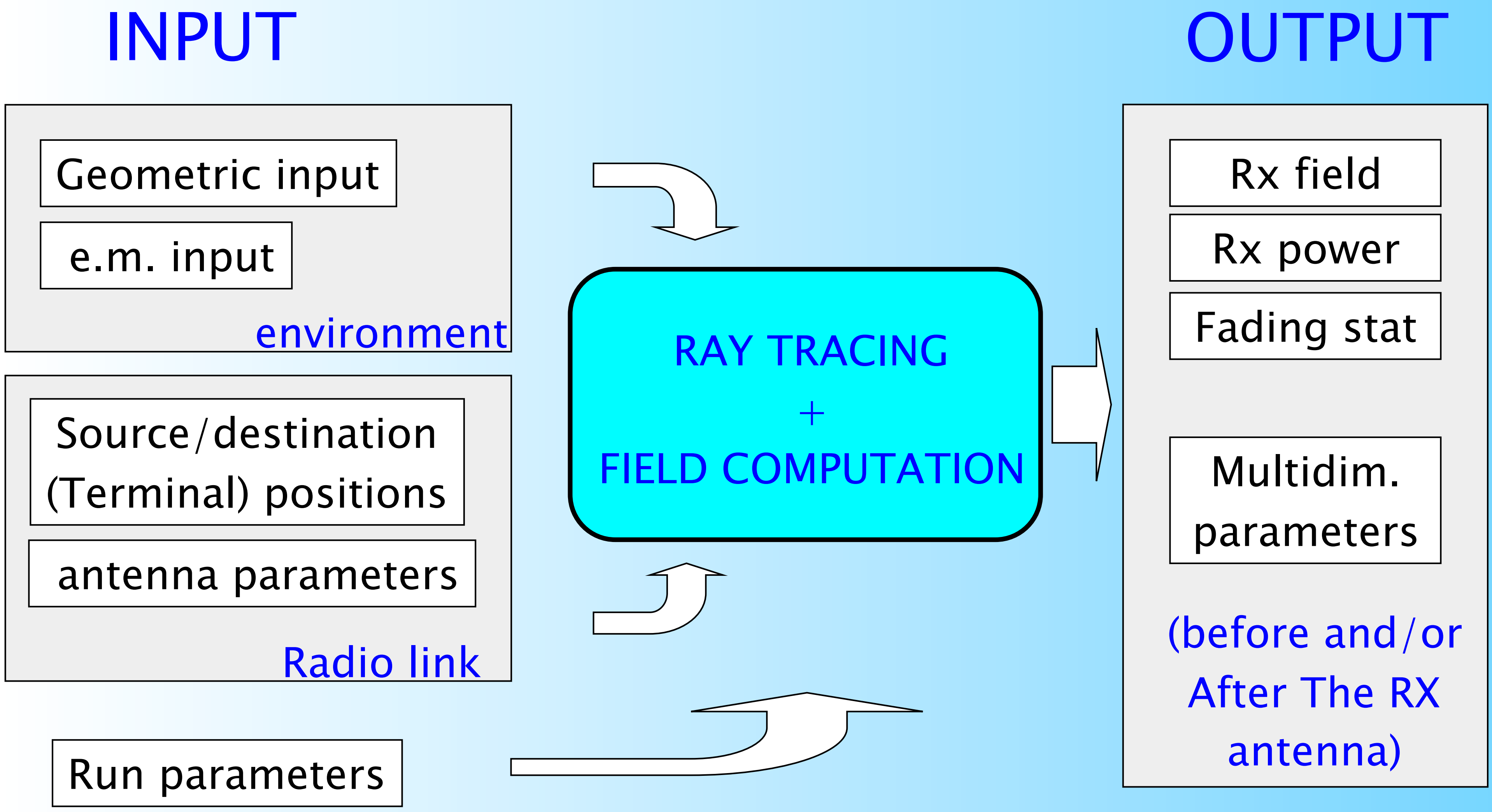
· Diffuse scattering can be modelled in ray models: ex: Effective Roughness model (see further on)

Ray Tracing Example

- Input files, database format
- The view tree - the concept of visibility
- Field Computation
- Example of output file

V. Degli-Esposti, "Urban propagation modelling and ray tracing" 12

Scheme of a ray tracing program



Inputs

“Slow” inputs

- Environment database
- Antenna parameters
- Most run parameters

“Fast” inputs

- Terminal positions
- Antenna orientation
- Some run parameters (es:Nev)

Slow inputs generally do not change during a work session

Fast inputs are generally changed multiple times during a work session

Environment database

Often the ***"PLANET"***[*] ***format*** is used for Environment database
Also the ***"SHAPE" format***, used by public administrations is adopted

The ***UTM system*** is used as a global geographic coordinate system (the one used by GPS)

The ***terrain database*** is a raster data file usually with a pixel of 5x5 m and a vertical resolution of 1 m

The ***urban database*** in PLANET format is composed of the following files:

.map file | .bld file | .atr file | .elm file | .bin file

[*] PLANET is a field prediction software developed by the MSI company, UK, http://www.msiuk.com

The .map file (ex: Helsinki.map)

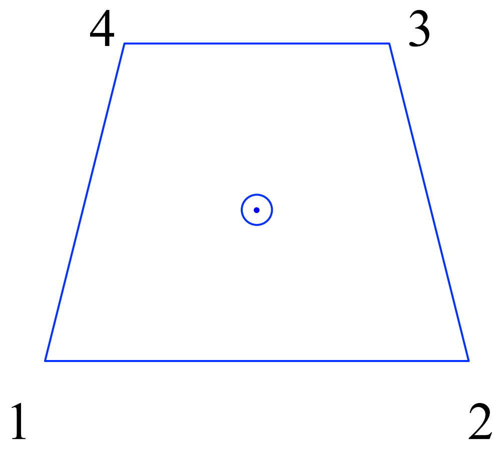
```
# map extension
Upper left corner:    0          5000
Lower right corner: 5000         0
# input files names
Building file:                   INPUT\SCENARI\HELSINKI\helsinki_buildings.bld
# (relative heights)
Height file:                     INPUT\SCENARI\HELSINKI\helsinki_buildings.atr
Electromagnetic file:            INPUT\SCENARI\HELSINKI\helsinki_buildings.elm
Terrain file:                    INPUT\SCENARI\HELSINKI\hki1000.bin
X resolution:                    5 m
Y resolution:                    5 m
```

The .bld file

id n. kind of object n. of records

```
00006       "building"              00005
2701.140000 3289.780000
2702.250000 3260.020000
2735.120000 3261.150000
2734.310000 3290.930000
2701.140000 3289.780000
00007       "building"              00007
2752.400000 3292.380000
2752.930000 3277.880000
2797.170000 3278.730000
2797.620000 3263.370000
2811.350000 3263.780000
2811.030000 3293.620000
2752.400000 3292.380000
```

Horiz. coordinates of the polygon vertexes (the last one is a duplicate of the first one)



The .atr file

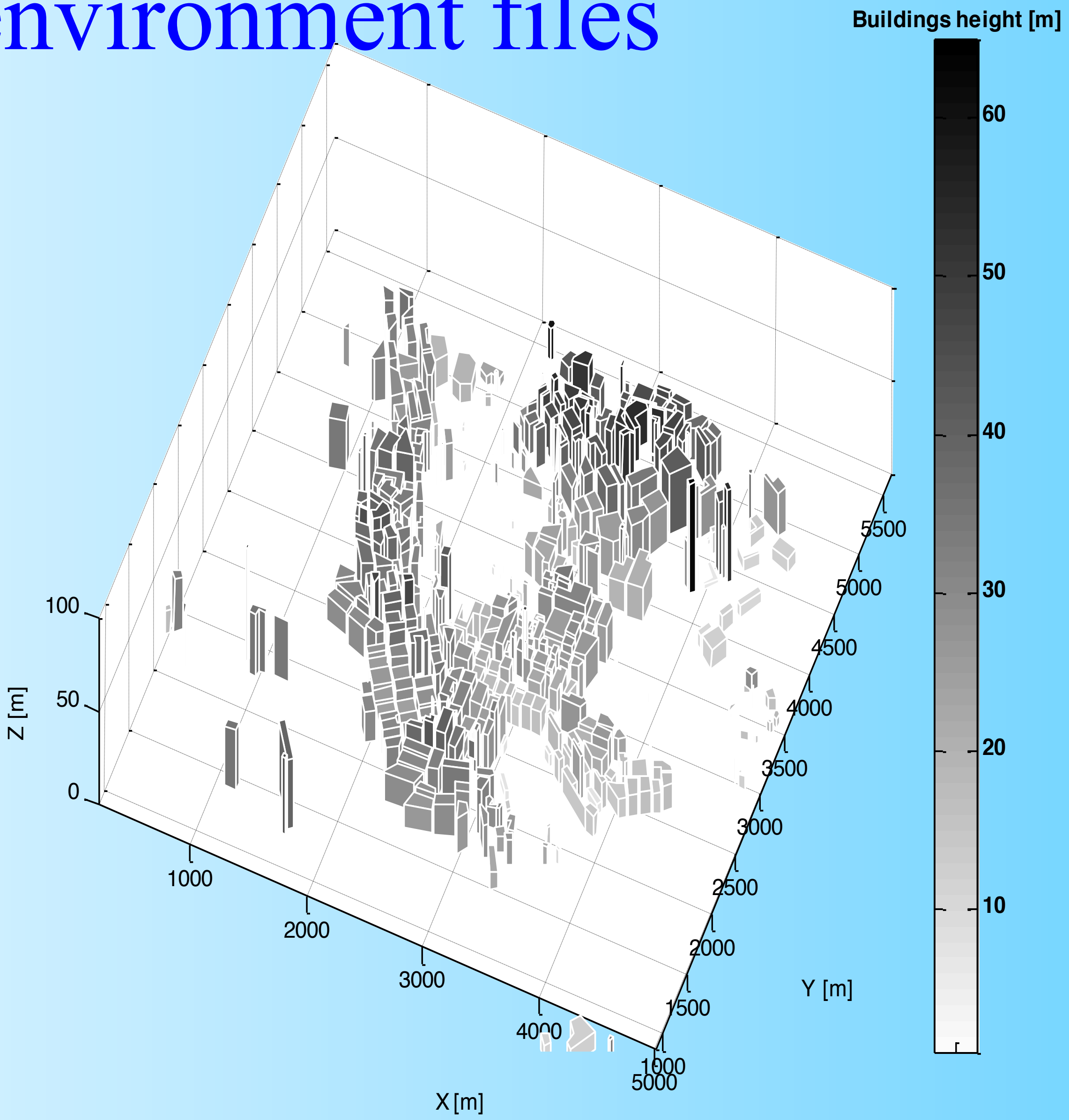
id n. kind of object Height in meters

- 1 "building" 4.250000 … other attributes?...
- 2 "building" 4.250000
- 3 "building" 3.500000
- 4 "building" 8.000000
- 5 "building" 13.125000
- 6 "building" 17.750000
- 7 "building" 21.000000
- 8 "building" 3.000000
- 9 "building" 2.000000
- 10 "building" 7.000000
- 11 "building" 7.000000
- 12 "building" 13.000000

Other environment files

- The ***.elm*** file is similar to the .bld file but ε_r and σ are reported for each side of the polygon (each wall) instead of the coordinates

- The ***.bin*** file is a binary file cointaining raster terrain height data



Other input files

- **The Tx file**

```
Position:                 2825.307 3070. 29.63
Power:                             40 dBm
Frequency:                         2154 MHz
Radiation File:                    INPUT\ANTENNE\antenna_helsinki
Azimuthal Pointing:                0 °
Tilt x:                            0 °
Tilt y:                            0 °
```

- **The Rx file**

```
Point A:   3398    2881    3.65
Point B:   0       0       0
RX Number:            1
Radiation File:                    INPUT\ANTENNE\isotropic
Azimuthal Pointing:                0 °
Tilt x:                            0 °
Tilt y:                            0 °
… other records as above …
```

The Run file

- **(Param.dat)**

```
# computation mode parameters
DEBUG 0
DOUBLE_PRECISION 1.e-6
DOUBLE_AREA_PRECISION 1.e-6
GPC_EPSILON 1.e-12
VERBOSE_RAY 0
MAP_2D 0
MAP_3D 1
NORM 0
TX_RX 0
ANTENNA 0
COHERENT_MODE 1
COHERENT_SCAT 0
COHERENT_ORT 0
PATTERN_EXTRAPOLATION 1
SCAT_AND_DIFF 1
UTD_DIEL 1
S 0.3
SCAT_S_MODEL 1
SCAT_PATTERN_MODEL 1
SCAT_ALPHA_R 3
SCAT_ALPHA_I 3
SCAT_K 0.95
K_FAR 2
```

```
RCS 0
ENABLE_TERRAIN 1
SIGMA_TERRAIN 1.e-3
EPSILON_TERRAIN 15
RAY_SELECTION 0
# power threshold to cut weak rays  (dBunit)
POWER_THRESHOLD_DB -150
BOX_DIM 3
K_FRESNEL 3
EDGE_LENGTH_MIN 1
PARAX_DIST 300
SIGMA 0.001
EPS 8
STEP 10
WALL_WIDTH 50
ORT_SCAT 0
PROFILE 2
REAL_ANGLE 1
# absolute/relative height flag
RELATIVE_H 0
```

The control string

- The control string includes some *fast inputs* which are fundamental *run parameters*
- It is conceived as a command-line parameter to be given when the ray tracing program is run from the operating system prompt

$$\boxed{\underbrace{N_{ev}}_{\text{mandatory}} \quad @\underbrace{[set1][set2]...[setN_{ev}]}_{\text{optional}}@ \quad \underbrace{N_{ev}^{r}\,r\,N_{ev}^{d}\,d\,N_{ev}^{s}\,s}_{\text{optional}}}$$

- @ is a separator
- set1 to setN_{ev} are N_{ev} sets of interaction identifiers whose elements can be r (reflection) d(diffraction) s(scattering): the i-th set defines the interaction kinds allowed for the i-th interaction along a ray. Each set can be replaced by the wildcard '*', with obvious meaning
- $N_{ev}^{r} N_{ev}^{d} N_{ev}^{s}$ are the maximum number of successive reflections, diffractions, scatterings, respectively, regardless of the order of them.

For example:

$$\boxed{3@**[rs]@2d1s}$$

means: 3 events, the first can be any kind as the second, the third can only be a reflection or a Scattering, and overall each ray can undergo max 2 diffractions and 1 scattering

The antenna input file (1/3)

# Name	Trisectorial Helsinki	
# Manufacturer	FB	
# -6 dB Main Lobe (H)	120º	
# -6 dB Main Lobe (V)	40º	
# Frequency	2154	MHz
# Front-To-Back Ratio	0	dB

General informations about antenna (optional)

Polarization:	L	
Angle:	0°	#Azimuth in $(\hat{\theta},\hat{\phi})$ plane

Information about field polarization. In this case: θ-polarization (linear)

Gain:	10	dBi

Antenna (maximum) Gain

The antenna input file (2/3)

Horizontal Pattern:

Azimuth (f) on H-plane (deg)

0	0.
10.1	-7
17.2	-18
24.7	-16.5
112.7	-24
139.7	-26.5
149.7	-35.5
156.7	-25.5
181.7	-18
191.7	-41.5
213.3	-32.5
289.4	-25.5
307.9	-29
318.2	-39.5
338.7	-18
360.	0.

Radiation Function (dB):

$$\frac{g(\pi/2,\phi)}{G}$$

Vertical Pattern:

0	0.
10.8	-8.1
35.5	-19.3
60	-19.9
99.3	-14.7
111	-17.6
139	-25.6
151	-21.8
180	-20.6
200	-23.6
240	-16.4
274.1	-19.7
294	-23.3
344.2	-15.6
357.7	-0.2
360	0.

The antenna input file (3/3)

- The antenna pattern is given only in two planes (H and V). This reflects what is usually supplied by manufacturers.
- The complete pattern in 3D can however be input to the ray tracing program if necessary (Poincaré's sphere data).
- Interpolation is used to get the 3D pattern from the two 2D patterns.
- For convention, ***the maximum of the radiation pattern always corresponds to the positive direction of the x axis, i.e.*** $\boldsymbol{\theta=\pi/2}$, $\boldsymbol{\phi=0}$.
- The actual antenna orientation with respect to the global reference system is given in the Tx and Rx files.

The computation steps

The big problem is how to find all possible ray trajectories… Efficient algorithms must be adopted 'cause the task is demanding. The visibility concept and the corresponding view tree are adopted here.
Computation is carried out in two steps:

- Geometric Ray Tracing
  - The concept of "view-tree"
  - Visibility and view-tree computation

- Field computation
  - Adjustment of ray geometry
  - Computation of the e.m. field

The view tree (1/5)

Environment ***objects*** are radio-terminals, walls, edges or parts of them

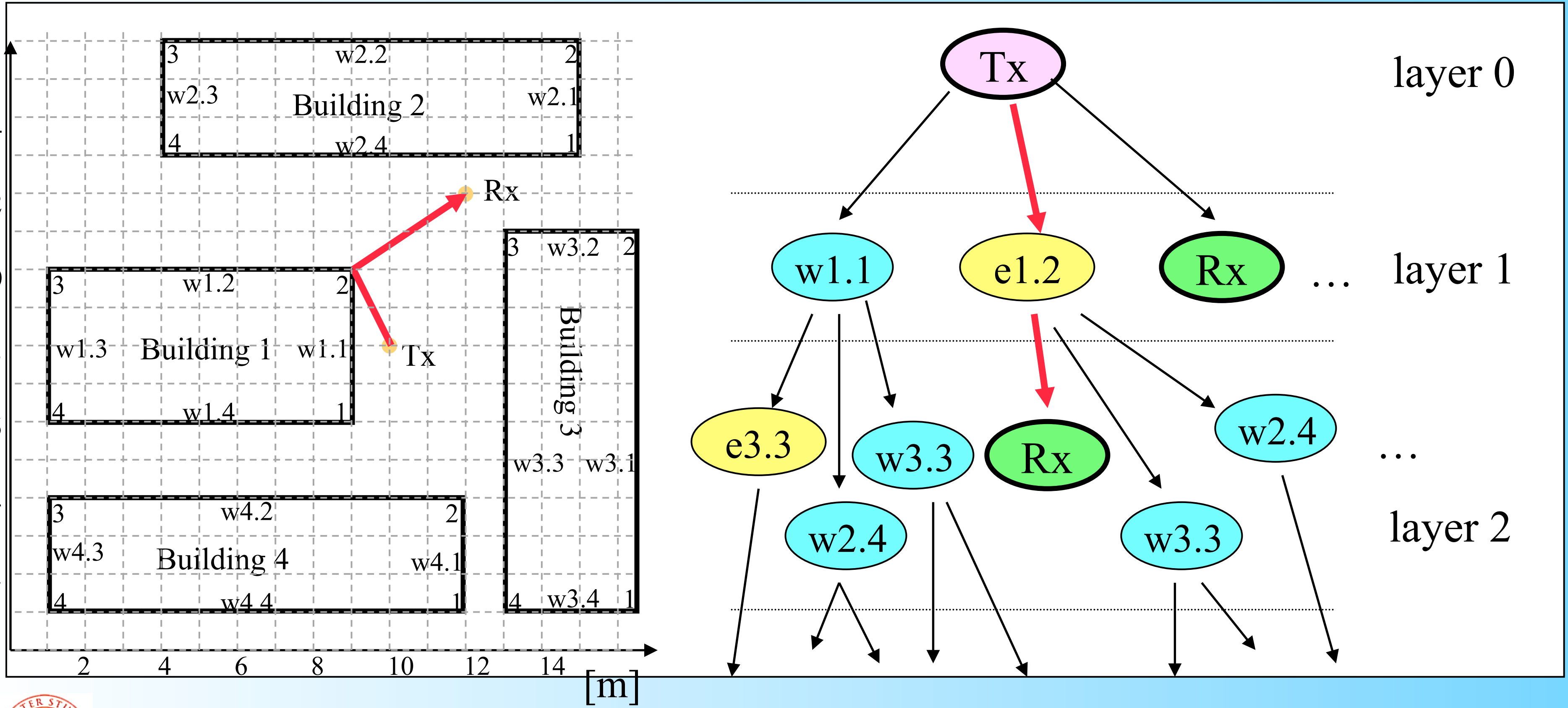
Visibility refer to the existence of a radio propagation path between two objects according to a given interaction on the first one.

The ***view tree*** contains all relevant visibility information for the actual geometrical tracing of rays in a given environment.

Visibility relations do not correspond to exact geometric information. Therefore, after the view tree is constructed, other steps are necessary for the actual computation of both ray geometry and ray field.

The view tree (2/5)

- The environment is decomposed into ***objects*** (walls, edges) + radio teminals
- The view tree is fill-up starting from the Tx on the base of ***visibility relations***
- The tree is layered and the number of layers is equal to Nev+1 (root included)
- The Tx is the root, objects are nodes, interactions are branches, the Rx leaves



The view tree (3/5)

- Actually nodes correspond to objects or parts of them
- The view tree stores all major information for ray tracing to be carried out, however the exact trajectory of rays is still partly undefined, and so is the field. A further “refinement” is therefore necessary
- View tree computation is by far the most time-comsuming part of ray tracing

If N_v is the average number of objects viewed from a previous one then it can be shown that the view tree is composed of M objects with

$$\boxed{\mathrm{M} \approx N_V^{\left(N_{ev}+1\right)}}$$

N_v depends on the kind of enabled events (see further on)

For example if $N_v=10$ and $N_{ev}=3 \rightarrow M=10^4$ This figure is of course proportional to the computation time involved in creating the view tree.

Each time a new object is added to the tree, then the whole environment must be scanned for visibility. It is therefore important to identify ‘smart’ scanning strategies.

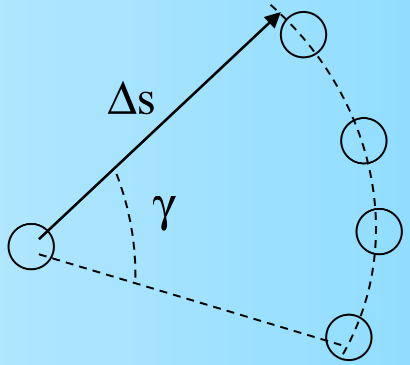
The view tree (4/5)

- One smart technique is ***environment pre-processing:*** objects that cannot be viewed from a given one are a-priori identified on the base of simple geometric considerations

The actual value of N_v depends on the kind of interaction involved

Simple 2D model to estimate N_v :

$$\boxed{N_V \approx \gamma \cdot \Delta s \cdot \delta_w}$$



where γ is *view angle*, Δs *average distance* and δ_w is *linear object density*.

The view angle changes very much with the considered interaction. In particular:

$$\gamma_r \approx \frac{K}{l} \text{ for reflection}$$

$$\gamma_d = \frac{3}{2}\pi \text{ for right wedge diffraction}$$

$$\gamma_s = \pi \text{ for diffuse scattering}$$

where K is a constant and l is *unrolled ray length* (from Tx). γ_r is generally much lower then γ_d and γ_s, and plus descreases with d, i.e. with tree depth.

The view tree (5/5)

- It is therefore evident that N_{ev} alone does not determine CPU time. It is mainly the number of successive diffraction or scattering events out of N_{ev} total successive events, that determines CPU time.

- It is advisable to set separate upper limits to the different kinds of events, and keep the diffraction or scattering ones as low as possible (see "The control string")

- The radio link is reciprocal but ray tracing is not. Since the number of traced rays is limited, then starting tree computation from one terminal can lead to a different computing path than starting from the other terminal $\rightarrow$ different CPU time and slightly different output results

- Environment size, i.e. the total number of buildings or objects determines the shape of the tree, in particular the growing rate of the tree for large values of N_{ev}.

- Moreover, the greater environment size, the higher visibility scanning time, thus total view-tree computation time

RT Computation Basics

The Ray Tracing (RT) algorithm is organised in 4 different steps:

- ***Pre-processing*** of objects

  removal from the environment of all objects which are certainly invisible and selection of the "potentially" visible objects

- ***Visibility*** of selected objects

These two steps are repeated *recursively* each time that a new node (i.e. a "Virtual TX", see further) is added to the "view tree".

Then, the following steps are:

- The ***RT refinement*** procedure
- The ***computation*** of the ***field*** carried by the current ray

  These last two steps are executed each time that a Rx is found (i.e. a "leaf" of the view tree is created).

Pre-processing for outdoor (1)

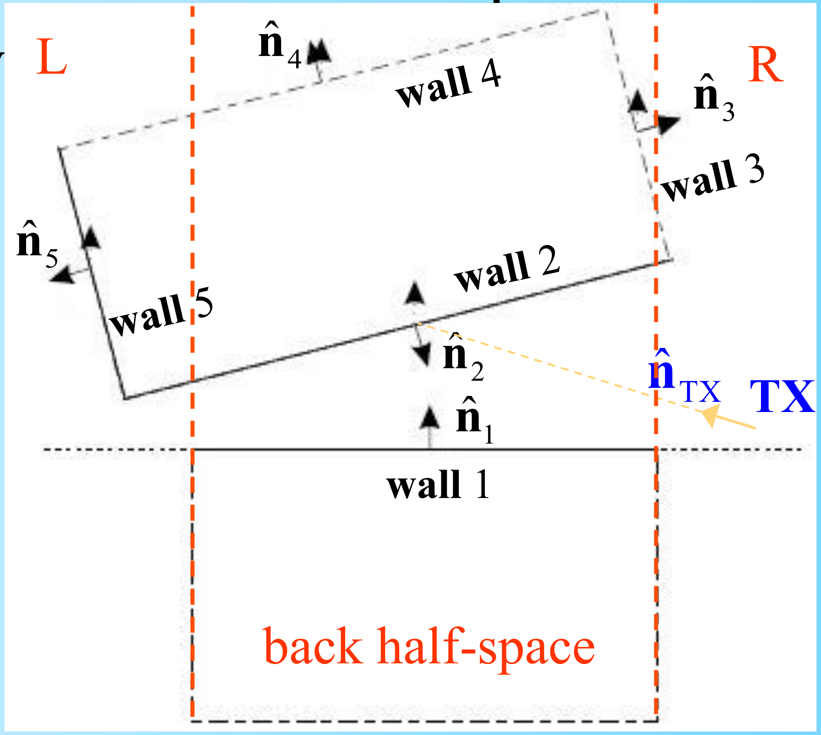
- Some "visibility relations" between the walls (or between the TX and each wall) of the scenario are a priori known.
- The knowledge of these relations can be exploited in order to simplify the visibility

walls i and j $\boxed{0 < \mathrm{acos}\left(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j\right) < \pi/2 \quad in\ R}$

not visible if $\boxed{-\pi/2 < \mathrm{acos}\left(\hat{\mathbf{n}}_i \cdot \hat{\mathbf{n}}_j\right) < 0 \quad in\ L}$

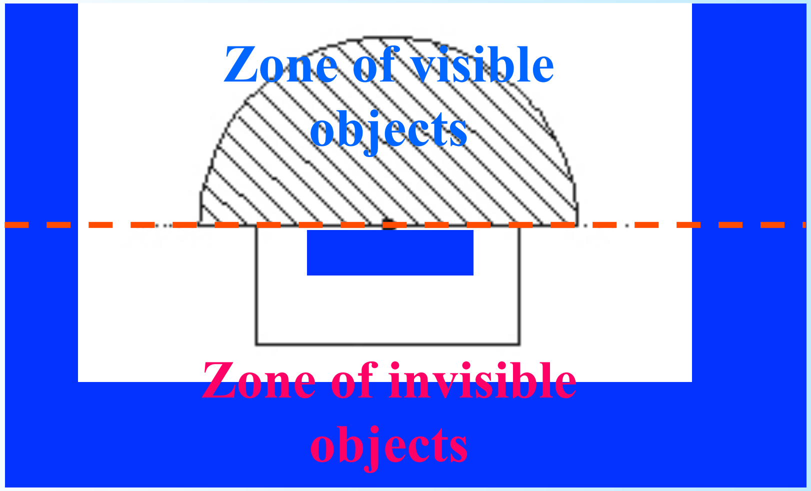
For istance:

- wall 3 and wall 4 are not visible from wall 1, while wall 2 and wall 5 are visible
- wall 2 is visible from the TX ($\hat{\mathbf{n}}_{TX}$ is obtained connecting the TX and the barycentre of wall 2)

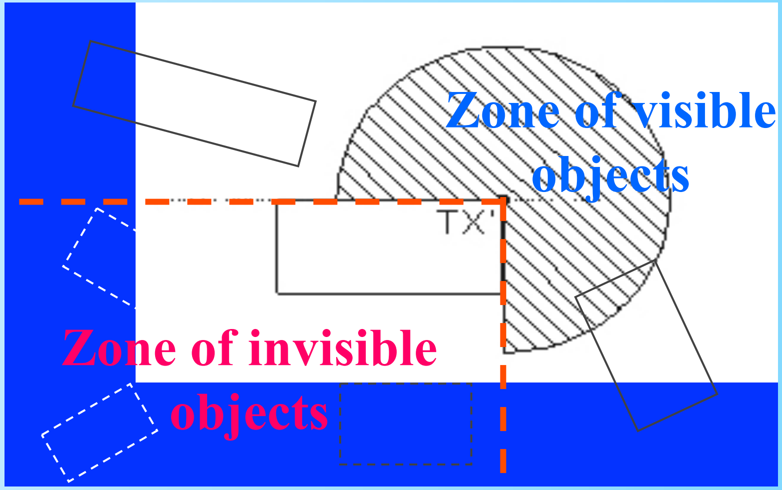


Pre-processing for outdoor (2)

- Reflection and scattering: all the objects which do not lie in the ***half space*** pointed to by the normal are not visible, and then are not considered in the visibility algorithm.
- Diffraction: all the objects which do not lie in the zone external to the (convex) edge are not visible.



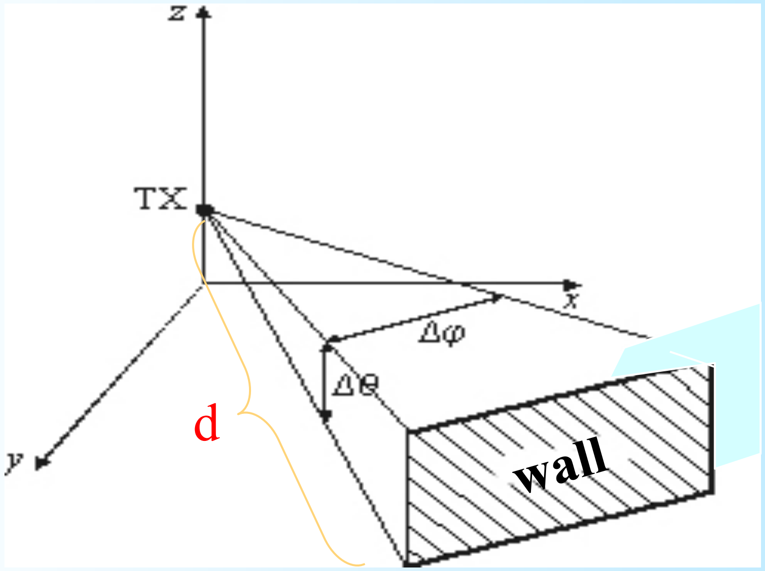
(a) Zones of visible and invisible objects (reflection and scattering)

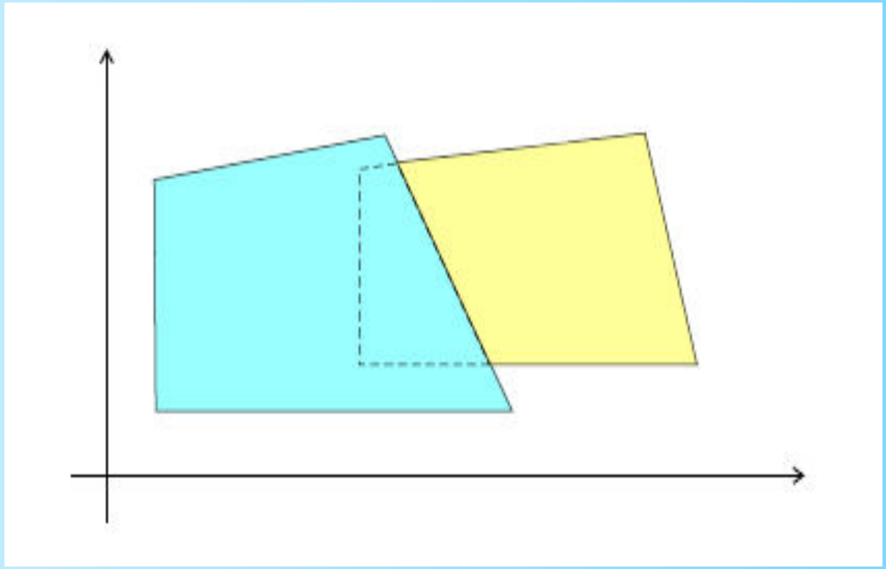


(b) Zones of visible and invisible objects (diffraction)

The Problem of 3D Visibility

- 3D visibility is a quite complex problem
- Each wall can be modelled as a polygon, and the ***visible portion*** of a wall is the result of the superposition of many obstructing objects
- The ***visible portion*** of a single wall (i.e. the "visible polygon") depends not only on the size of the wall, but also on the distance from the view point (i.e. the TX or a Virtual TX) ➡ *"Polygon clipping"* algorithms





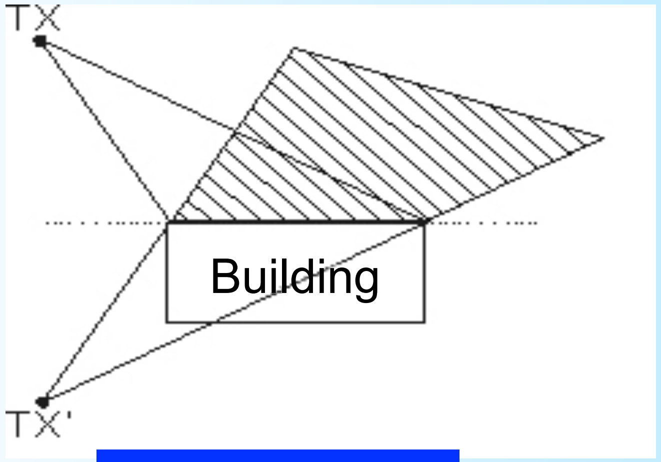
Virtual TX

- When a new object is seen, a new node is inserted in the view tree.
- According to the "Image RT" method, each node of the view tree corresponds to a Virtual TX (VTX).
- Whenever a node is processed, the spherical reference system (r,θ,φ) centred in the VTX is adopted.

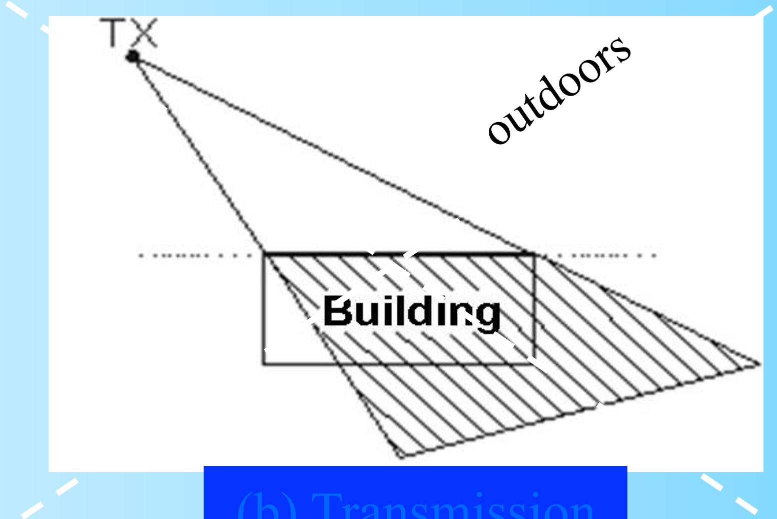
- The position of the VTX w.r.t. the visible object depends on the interaction mechanism. For instance:

  - The reflection VTX is the reflected image of the previous (V)TX w.r.t. the wall plane.
  - The diffraction VTX lies on the edge, and conventionally can be positioned in the middle point of the edge (the actual diffraction point is determined at the end of the process…)
  - The scattering VTX is assumed conventionally in the barycentre of the considered portion of the wall.

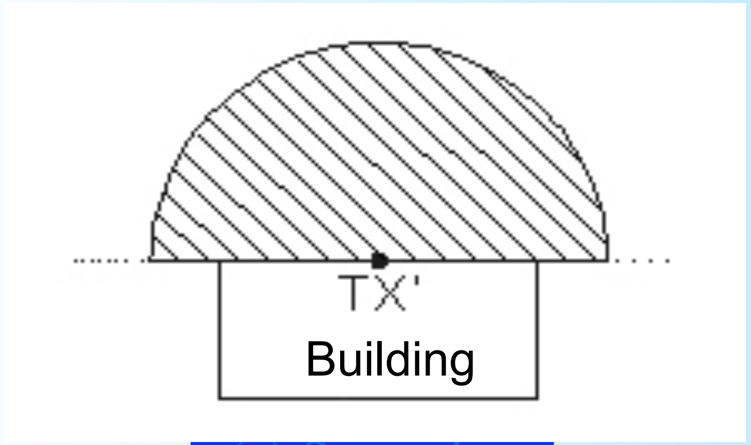
Virtual TXs and Visibility Angles for different propagation mechanisms



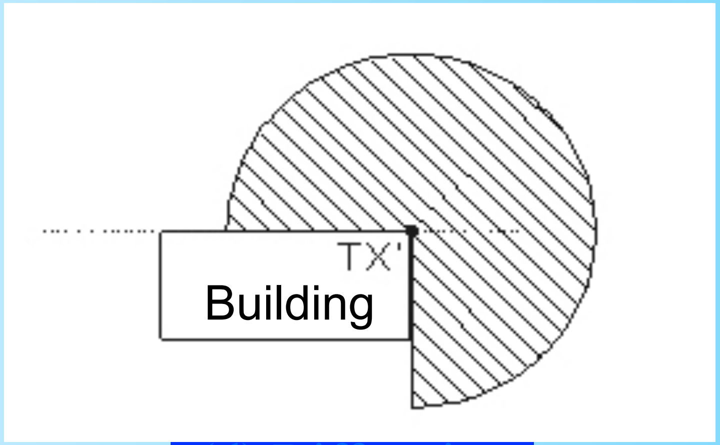
(a) Reflection



(b) Transmission



(c) Scattering



(d) Diffraction

The “RT refinement” Procedure

diffraction points on edges can only be exactly determined after view-tree fill-up (when a Rx is reached)

If the considered ray undergos diffractions and reflections, then also reflection points can only be determined after visibility.

⬇

If is then necessary to exactly determine the cited interaction points starting from the Rx and backtracking the ray until the Tx is reached. (*backtracking*)

There are known, fixed points:

- Tx and Rx positions
- Scattering points (conventionally in the barycentre of each wall element)

The field computation Procedure (1)

- Once all interaction points have been determined then ray field can be computed.
- The far field emitted in the generic point P(r,θ,φ) from the Tx antenna can be computed through the input signal (not only power) as:

$$\vec{E}_T\left(r,\theta_T,\phi_T\right)=I_T\cdot\sqrt{\frac{Z_T\cdot\eta\cdot g_T\left(\theta_T,\phi_T\right)}{16\pi}}\cdot\frac{e^{-j\beta r}}{r}\cdot\hat{p}_T\left(\theta_T,\phi_T\right)=\vec{E}_{T0}\left(\theta_T,\phi_T\right)\cdot\frac{e^{-j\beta r}}{r}\quad(2)$$

where Z_T is the impedance of the antenna, I_T is the current phasor feeding the antenna, g_T the antenna gain function, h is the intrinsic impedance of the medium, $\beta=2\pi/\lambda$ is the wave number and $\hat{p}_T$ is the antenna polarization vector.

- Formula (2) is only valid in free space, can therefore be applied only to the direct ray, with no interactions.
- When interactions occur the already explained procedure of multiplication with the interaction dyadic and re-formulation of the spreading factor must be applied, see following slide.

The field computation Procedure (2)

Considering also interactions the k-th ray field is:

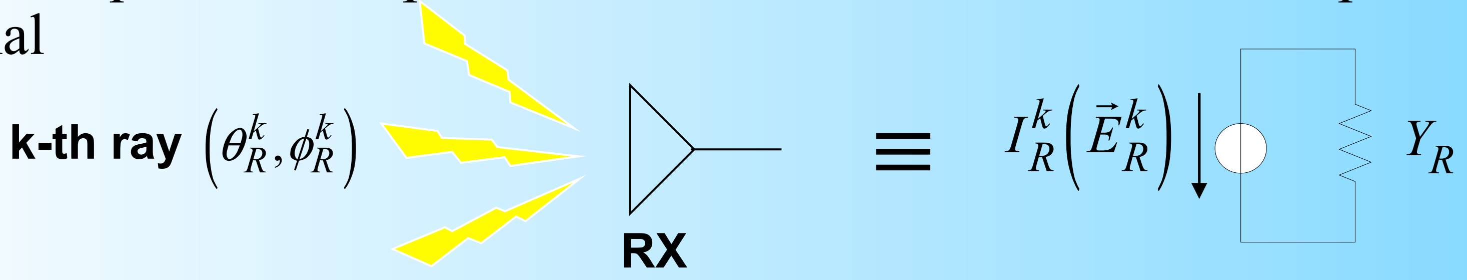
$$\vec{E}_R^k = A_k\left(s_\ell, \ell = 0,1,2,\ldots,N_{EV}^k\right) \cdot \left[\prod_{\ell=\min\left\{1,N_{EV}^k\right\}}^{N_{EV}^k} \underline{\underline{D}}_\ell\right] \cdot \vec{E}_{T0}^k\left(\theta_T^k, \phi_T^k\right) e^{-j\beta s^k} \quad (**)$$

where:

- N_{EV}^k is the number of events experienced by the k-th ray
- s_ℓ is the length of the ℓ-th segment composing the k-th path
- $s^k = \sum_{k=1}^{N_{EV}^k} s_\ell^k$ is the total, unfolded length of the k-th ray
- $\underline{\underline{D}}_\ell$ is the appropriate dyadic to decompose the field into orthogonal polarizations at the ℓ-th interaction point, and includes the interaction coefficients
- A_k is the overall *spreading factor*

The field computation Procedure (3)

- Once the field for each ray has been computed it is necessary to take into account pattern and polarization of the Rx antenna to compute the Rx signal


k-th ray $\left(\theta_R^k,\phi_R^k\right)$
RX
$\equiv$
$I_R^k\left(\vec{E}_R^k\right)$
Y_R

Equivalent circuit of the RX antenna

- The current phasor induced in the RX antenna by the k-th ray can be computed as (Reciprocity Theorem):

$$I_R^k = -j\lambda\sqrt{\frac{\Re e\left(Y_R\right)g_R\left(\theta_R^k,\phi_R^k\right)}{\pi\eta}}\left\{\vec{p}_R\left(\theta_R^k,\phi_R^k\right)\cdot\vec{E}_R^k\right\}$$

where $Y_R = 1/Z_R$ and the subscript "R" refers to the Rx antenna. $\left(\theta_R^k,\phi_R^k\right)$ are the arrival angles of the considered ray in the Rx-based local coordinate system.

The field computation Procedure (4)

- If there is perfect impedance matching between the Rx antenna and the Rx circuitry then received power becomes:

$$P_R = \frac{\left|I_R^{\,TOT}\right|^2}{8\cdot\Re e\left(Y_R\right)}$$

- Therefore, the total coherent power can be expressed as:

$$P_R = \frac{\left|I_R^{tot}\right|^2}{8\cdot\Re e\left(Y_R\right)} = \frac{\left|\sum_{k=1}^{N} I_R^k\right|^2}{8\cdot\Re e\left(Y_R\right)} = \frac{\left|\sum_{k=1}^{N}\left(-j\lambda\sqrt{\frac{\Re e\left(Y_R\right)g_R\left(\theta_R^k,\phi_R^k\right)}{\pi\eta}}\left\{\vec{p}_R\left(\theta_R^k,\phi_R^k\right)\cdot\vec{E}_R^k\right\}\right)\right|^2}{8\cdot\Re e\left(Y_R\right)}$$

where g_R and $\vec{p}_R$ are the antenna gain and the polarization vector of the RX antenna evaluated in the arrival direction of each incoming ray.

The field computation Procedure (5)

Developing the above expression, and simplifying the RX antenna conductance, we obtain:

$$P_R = \frac{\lambda^2}{8\pi\eta} \cdot \left| \sum_{k=1}^{N} \left(f_R\left(\theta_R^k, \phi_R^k\right) \cdot \left\{ \vec{p}_R\left(\theta_R^k, \phi_R^k\right) \bullet \vec{E}_R^k \right\} \right) \right|^2 \quad (1)$$

where $f_R(\theta,\phi) = \sqrt{g(\theta,\phi)}$ is the "radiation function" of the RX antenna.

Assuming a 2-ray model (i.e. direct ray + terrain reflection) and supposing perfect polarization matching between the incident field and the RX antenna eq. (1) becomes:

$$P_R = \frac{1}{2\eta} \cdot \left| \vec{E}_1 f_R\left(\theta_1, \phi_1\right) + \vec{E}_2 f_R\left(\theta_2, \phi_2\right) \right|^2 \cdot \frac{\lambda^2}{4\pi}$$

If only the direct path exists, the eq. 1 reduces to the well known equation:

$$P_R = \frac{\left|\vec{E}\right|^2}{2\eta} \cdot \frac{\lambda^2}{4\pi} g\left(\theta_R, \phi_R\right) = \frac{\left|\vec{E}\right|^2}{2\eta} \cdot A_{eff}\left(\theta_R, \phi_R\right)$$

The field computation Procedure (6)

Due to the incoherent nature of ORT and diffused rays, ORT rays and all rays undergoing at least one diffuse scattering interaction are usually assumed "incoherent".

For incoherent rays, only the module of the field carried by the ray can be estimated (through an incoherent version of formula (**), using incoherent scattering coefficients).

The overall incoherent power can be computed as :

$$P_{incoherent} = \sum_{k=1}^{N_{inc}} \frac{\left| E_{inc}^{k}\left(\theta_R^k, \phi_R^k\right) \right|^2}{2\eta} A_{eff}\left(\theta_R^k, \phi_R^k\right)$$

Finally, the incoherent rays power $P_{\text{incoherent}}$ is summed to the coherent rays power PR (see previous slides) obtaining therefore the total received power:

$$P_R^{tot}(Rx) = P_R + P_{incoherent}$$

Example of output file

File ***.pdp*** → gives all the useful geometric and electromagnetic information for each ray, including ray trajectory, field, power, delay, angle of arrival and departure, etc.

```
Tx_id     Rx_id     Ray_id   N_interactions
x_RX      y_RX      z_RX     R
x_int_1   y_int_1   z_int_1  o
.................
x_int_i   y_int_i   y_int_i   r
.................
x_TX      y_TX      z_TX     T
```

Interaction type (ORT diffraction)

Ray trajectory (backward)

Interaction type (reflection)

(DOD_phi , DOD_theta)   (DOA_phi , DOA_theta) ← **DOA, DOD and Delay**

delay   ERP (DoD) (linear unit)   Rx_Gain (linear)   tau

Depolarization factor

abs(Ex)   arg(Ex)
abs(Ey)   arg(Ey)
abs(Ez)   arg(Ez)

Field components (module and phase)

abs(Ic)   arg(Ic) ← Current at RX (module and phase)

Polarization

Power (dB-unit)