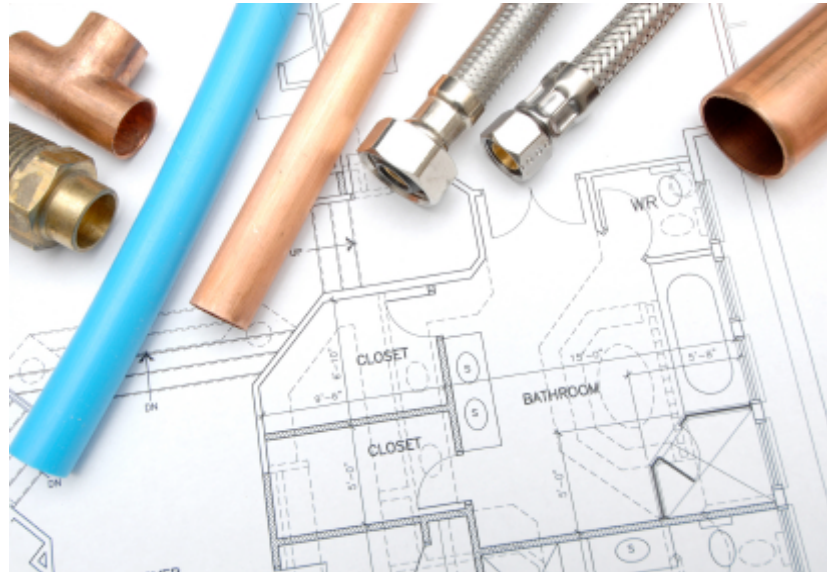


# Simplifying and Exchanging 3D Utility Network Objects Using CityModels



Ihab Hijazi, Bryan Hempen and Manfred Ehlers

Institute for Geoinformatics and Remote Sensing (IGF)  
University of Osnabrück, Germany

***1. Introduction***

***2. Simplifying 3D Utility Network Objects***

***3. Re-Extraction of Boundary Representation***

***4. Implementation and CityGML Output***

# ***Introduction***

Disaster management  
Kreis Recklinghausen

Radio network planning  
Mobile

Noise immision mapping  
Stapelfeldt GmbH

CityGML for 3d city models

Police simulator  
Rheinmetall Defence Electronics

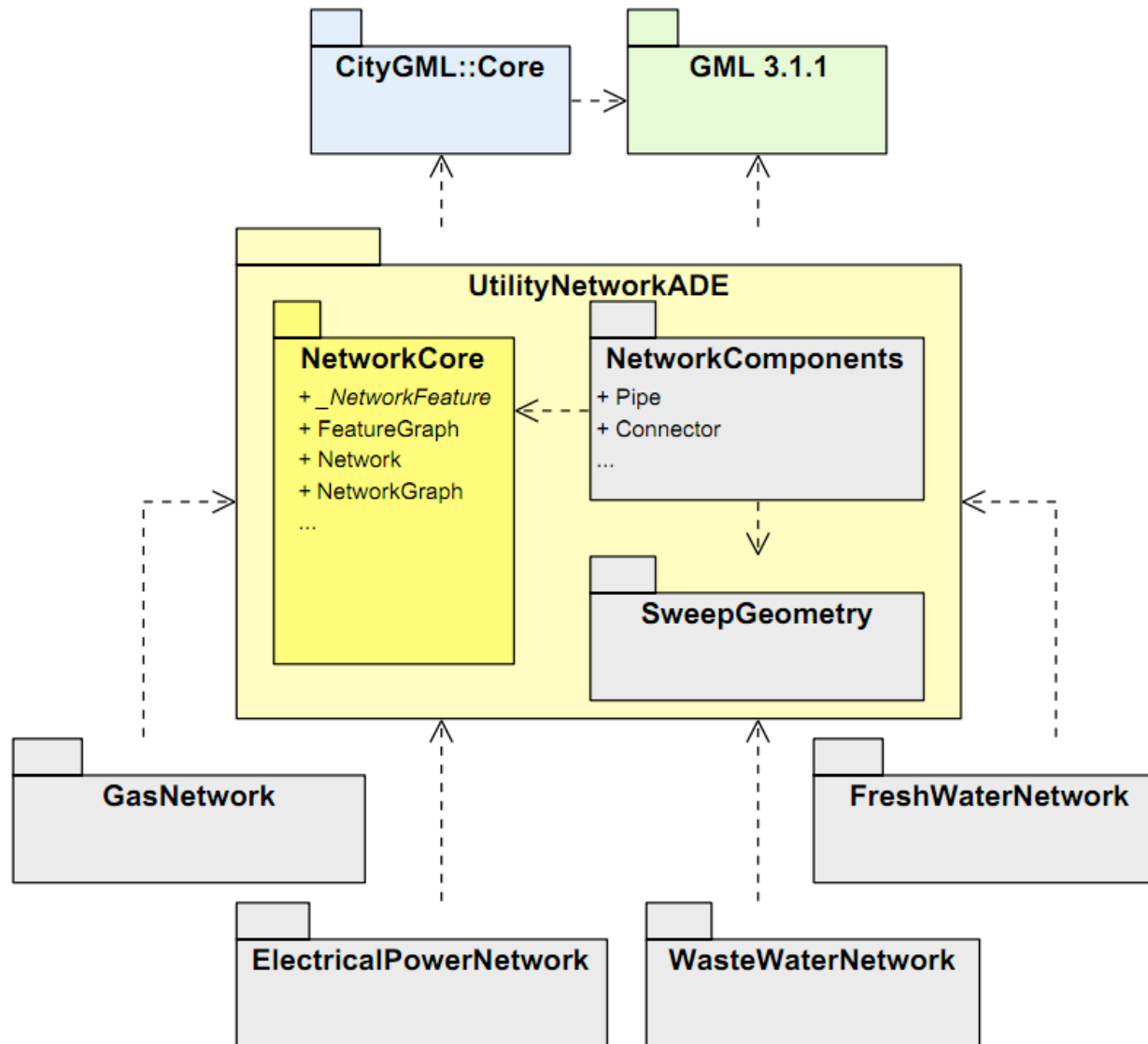
Business development & tourism  
Google

Navigation  
Münsterplatz

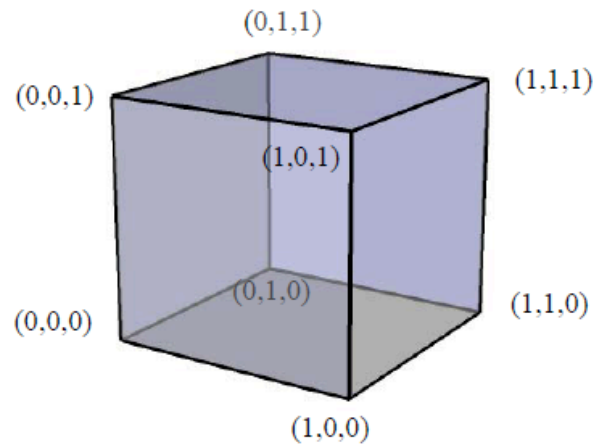
Facility management

Urban planning

Architecture  
Architekturbüro Stadt SenStadt Berlin



# Boundary Representation (BREP)



ID	Vertices
V1	(0,0,0)
V2	(1,0,0)
V3	(1,0,1)
V4	(0,0,1)
V5	(0,1,0)
V6	(1,1,0)
V7	(1,1,1)
V8	(0,1,1)

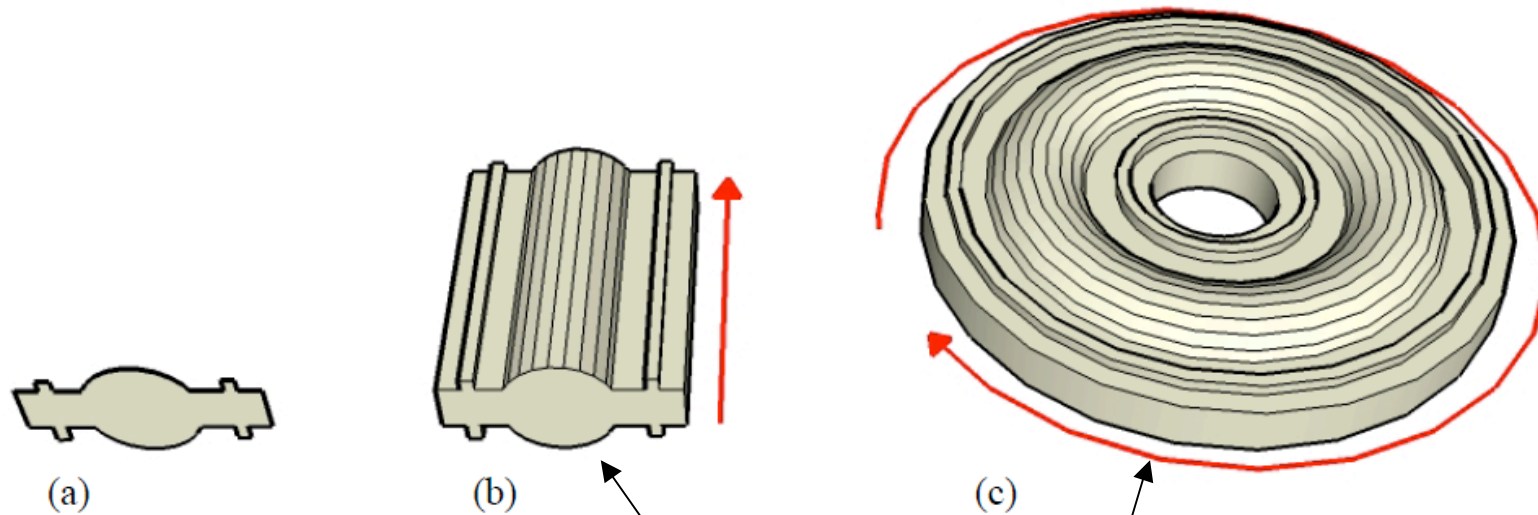
ID	Edges
E1	V1,V2
E2	V2,V3
E3	V3,V4
E4	V4,V1
E5	V1,V5
E6	V2,V6
E7	V3,V7
E8	V4,V8
E9	V5,V6
E10	V6,V7
E11	V7,V8
E12	V8,V5

ID	Faces
F1	E1,E2,E3,E4
F2	E1,E5,E6,E9
F3	E2,E6,E7,E10
F4	E4,E5,E8,E12
F5	E3,E7,E8,E11
F6	E9,E10,E11,E12

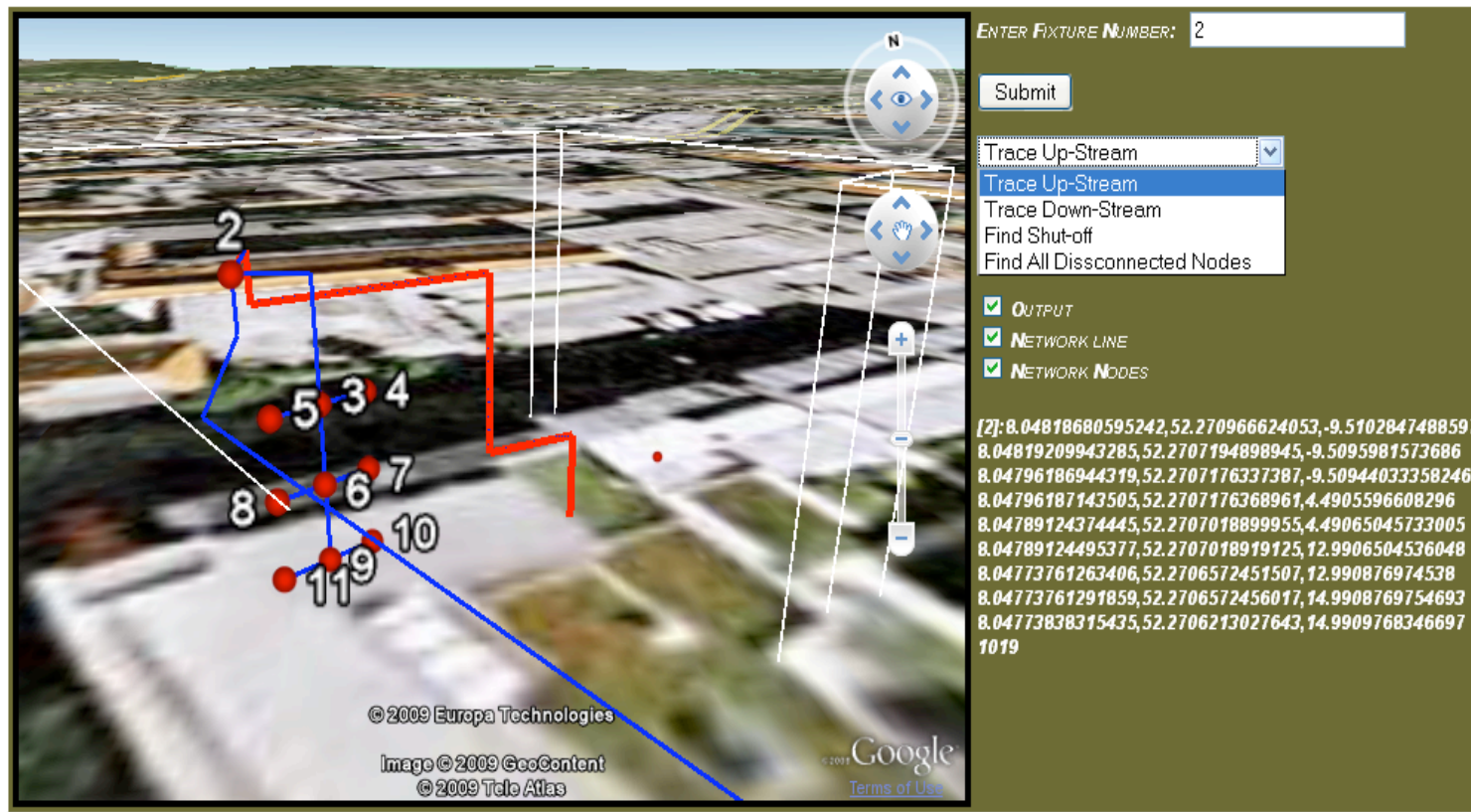
(Ekberg 2007)

- **Commonly used in GIS**
- **Used by CityGML for objects**
- **Requires a lot of storage space, especially for network objects**
- **Low precision**





- **Used in CAD**
  - **Suitable for objects with translational or rotational geometry**
  - **Proposed by CityGML Utility Network ADE**
  - **Requires minimal amount of storage space**
  - **High precision**
  - **Lack of datatype in GIS**
- (Ekberg 2007)



(Hijazi et al 2010)

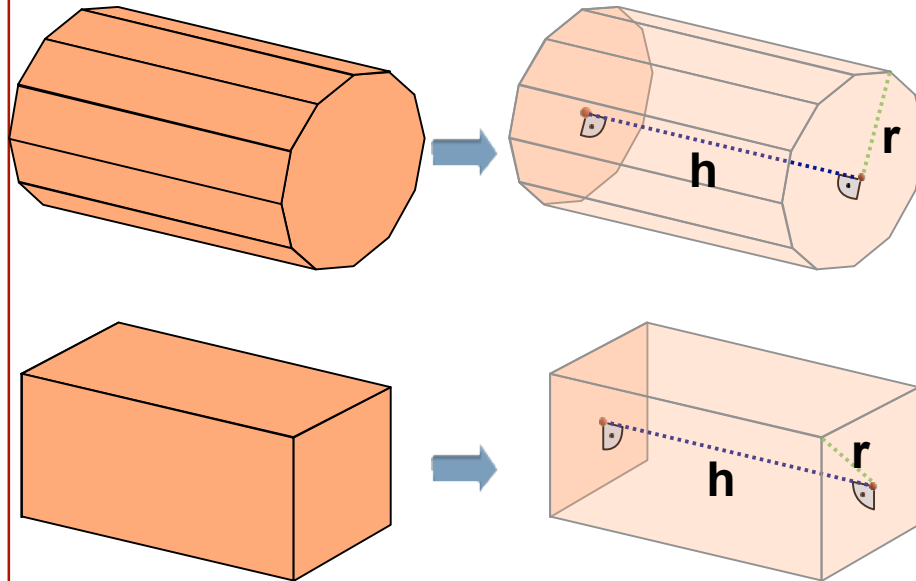
**3D Graph for topological analysis (trace up- and downstream)**



- Suggest **new GIS datatype** compatible to sweep representation  
→ same detail as **BREP**, less storage space
- Propose **methods for simplification** of **BREP** to sweep representation
- To show feasibility: **methods for Re-creation** the **BREP**
- **Output** sweep representation conforming to **CityGML Utility Network ADE**

# ***Simplifying 3D Utility Network Objects***

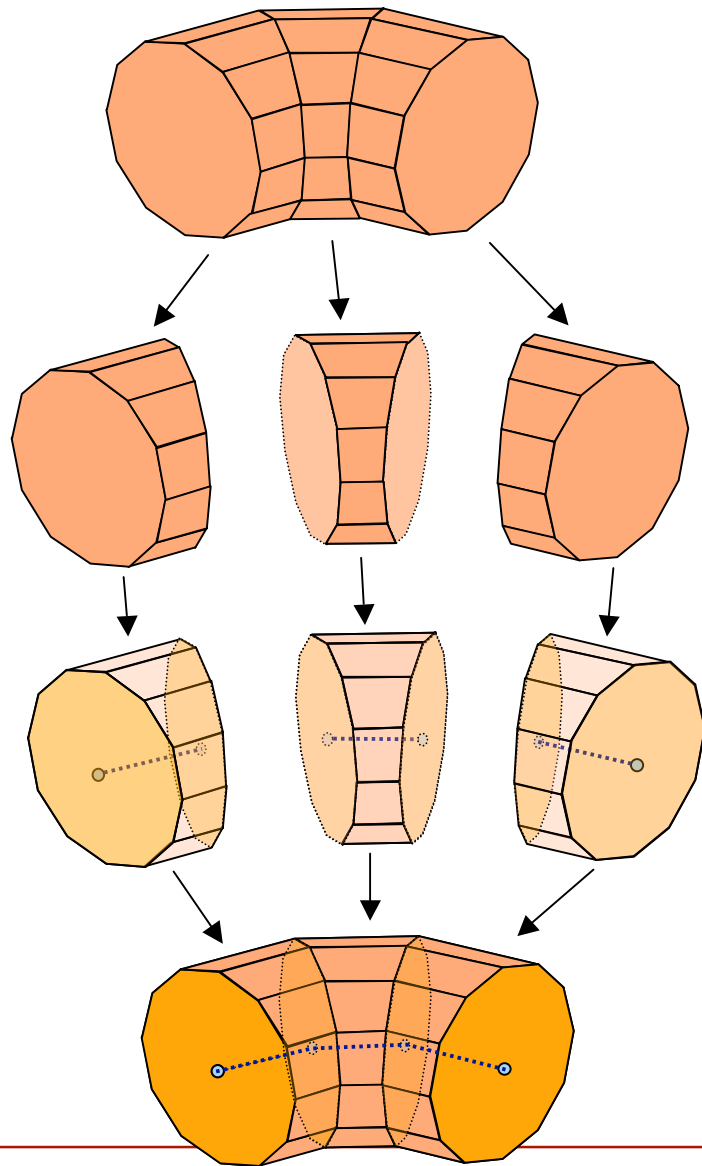
## Basic Idea for Simple Straight Network Objects



- **Centerline = line between ports ( $h$ )**
- **Additionally store radius ( $r$ )**

→ **Storage as sweep representation**

# Network Objects with Turns

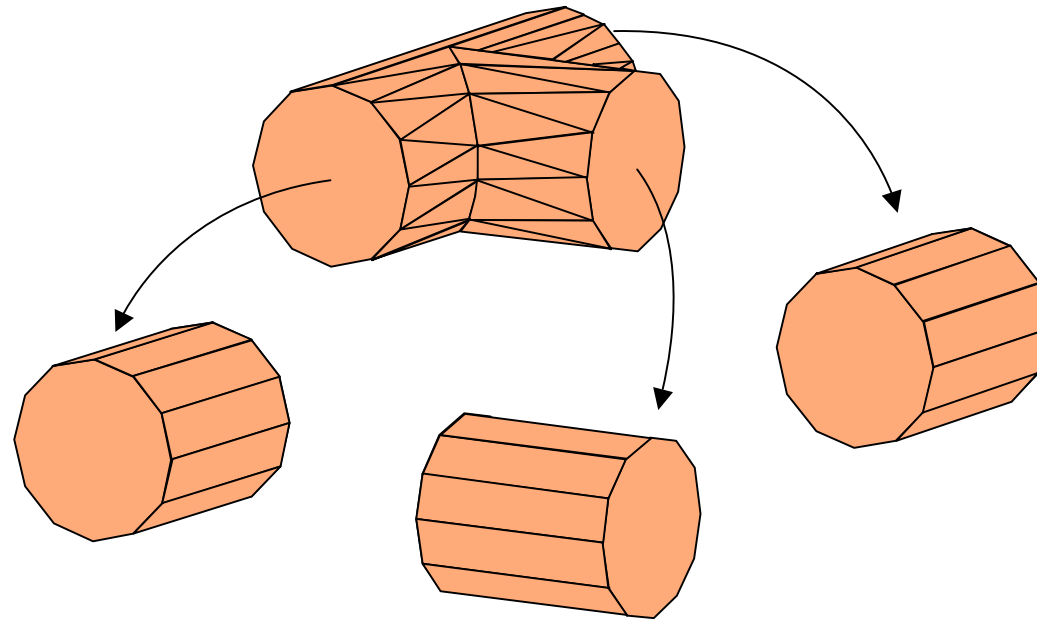


***Each segment = to be treated like singular network object***

***Extract centerlines***

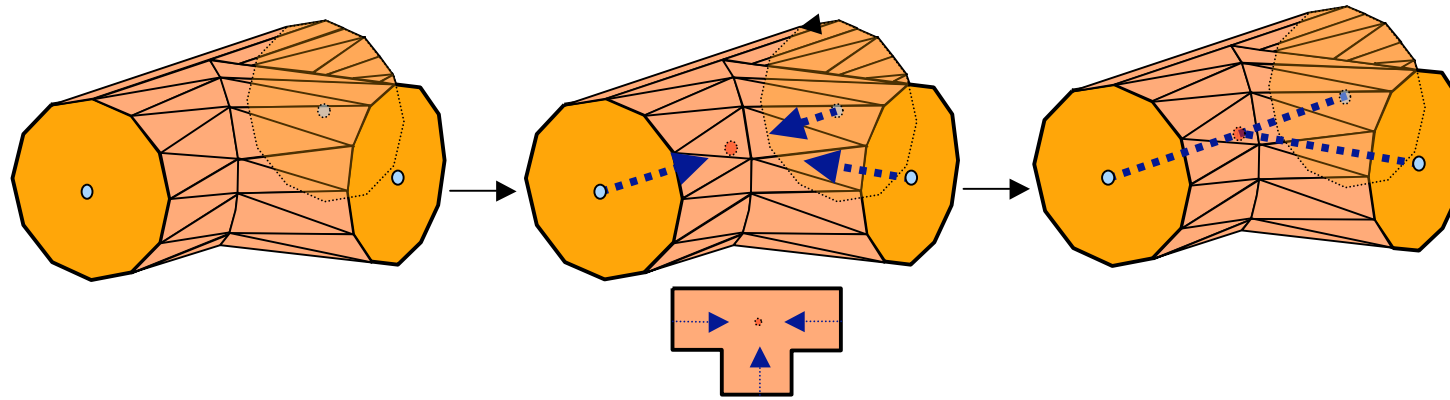
***Concatenate!***

## *Fittings: Same Idea?*



***Problem: underneath overlapping parts the faces are cut!***



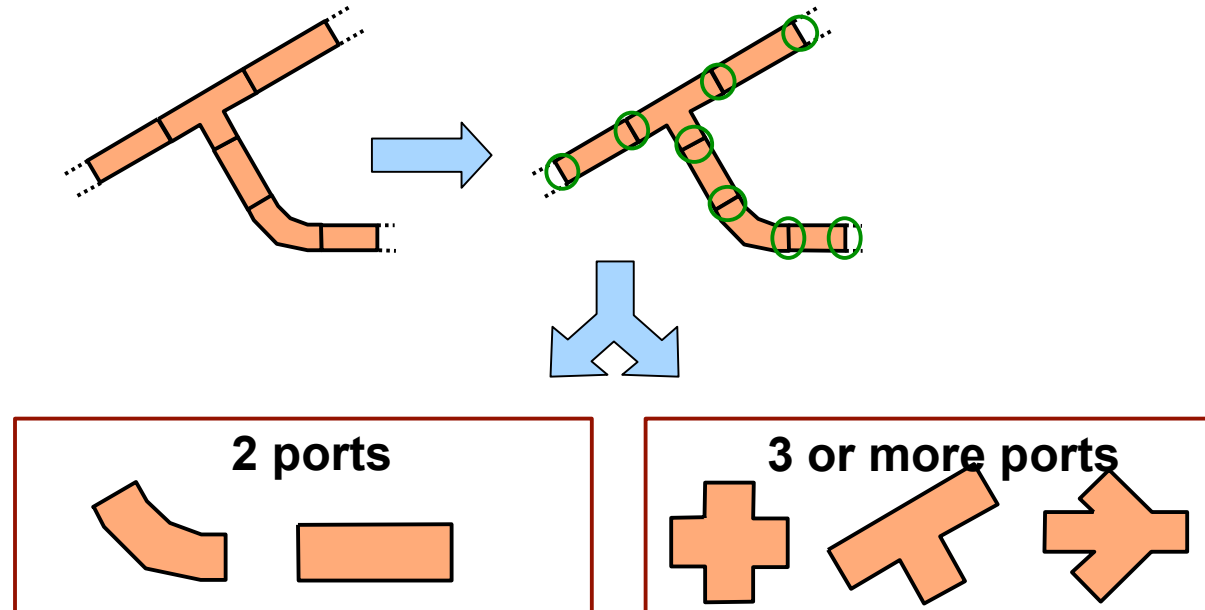


**1. Calculate centroids**

**2. Perpendicular to the ports, move towards the inside and determine the intersection point.**

**3. Connect central point with centroids**

# The Utility Network as a Whole

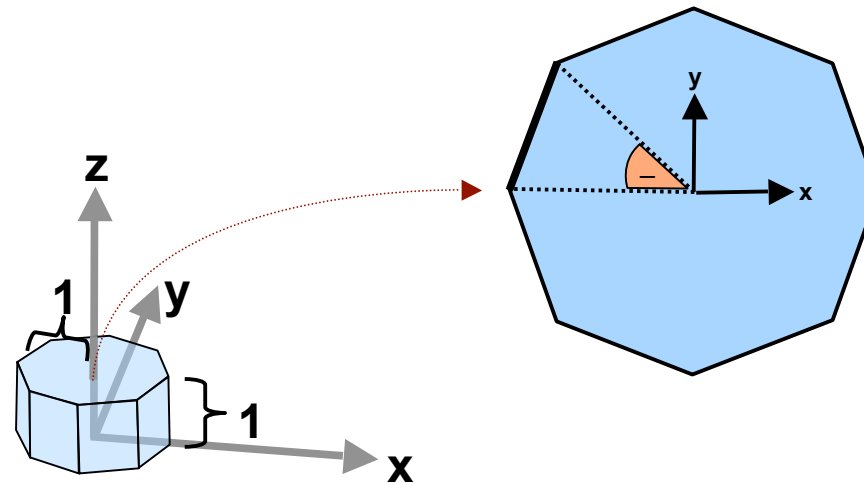


**Determine number of ports. For network objects with...**

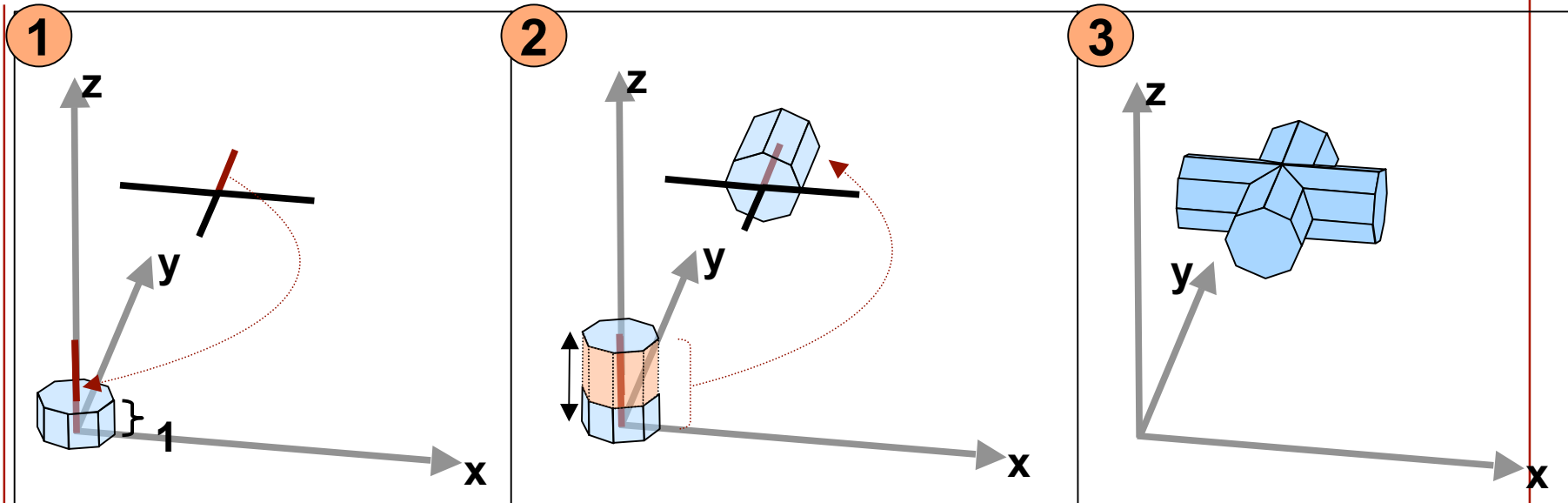
**a) ...2 ports: run algorithm for network objects with turns**

**b) ...3 or more ports: run algorithm for fittings**

# ***Re-Extraction of Boundary Representation***



- 1. Create cylinder with known default parameters.**
- 2. Use 3D transformations to fit each segment of the centerline to the axis of the default cylinder.**
- 3. Use the reverse parameters for transforming the cylinder (and the centerline) back to the centerlines original position.**



**a) Translate into point of origin.**

**b) Rotation around z-axis into x/y-plane.**

**c) Rotation around x-axis into z-axis.**

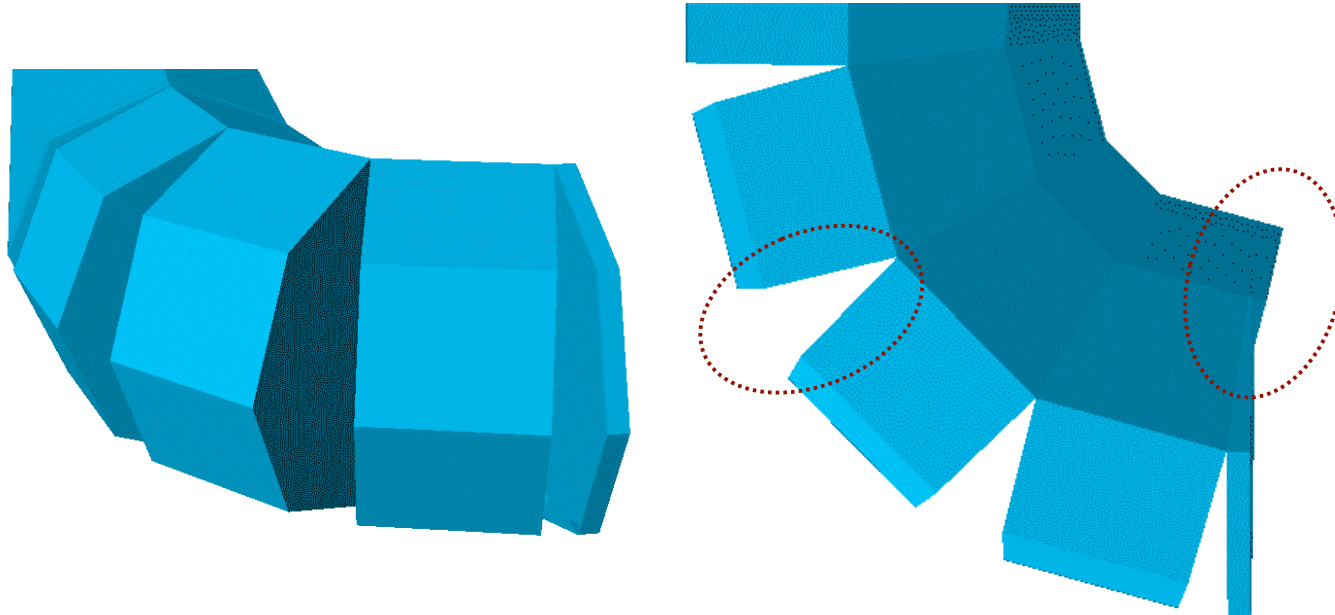
**d) Scale cylinder into positive z-axis direction to length of centerline.**

**e) Reverse transformations for c, b, and a.**

**f) Do this for all segments.**

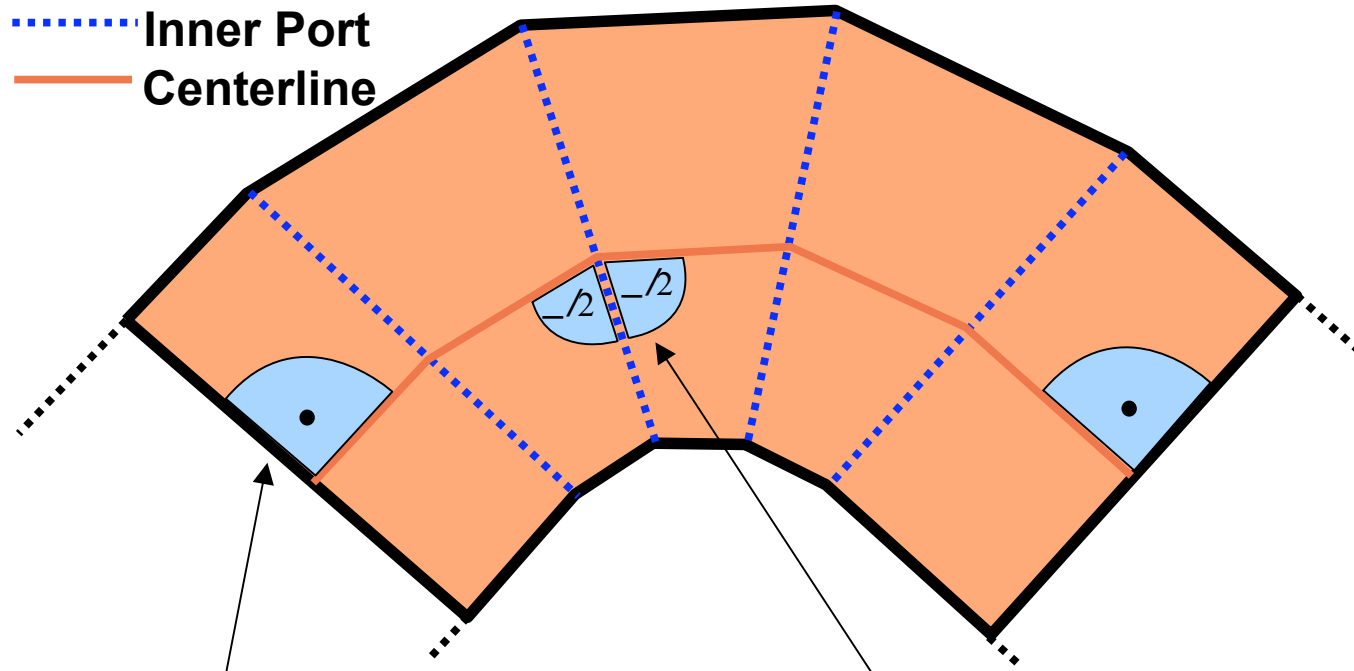


## Network Objects with Turns (1): The Problem



*Problems applying the previously introduced approach*

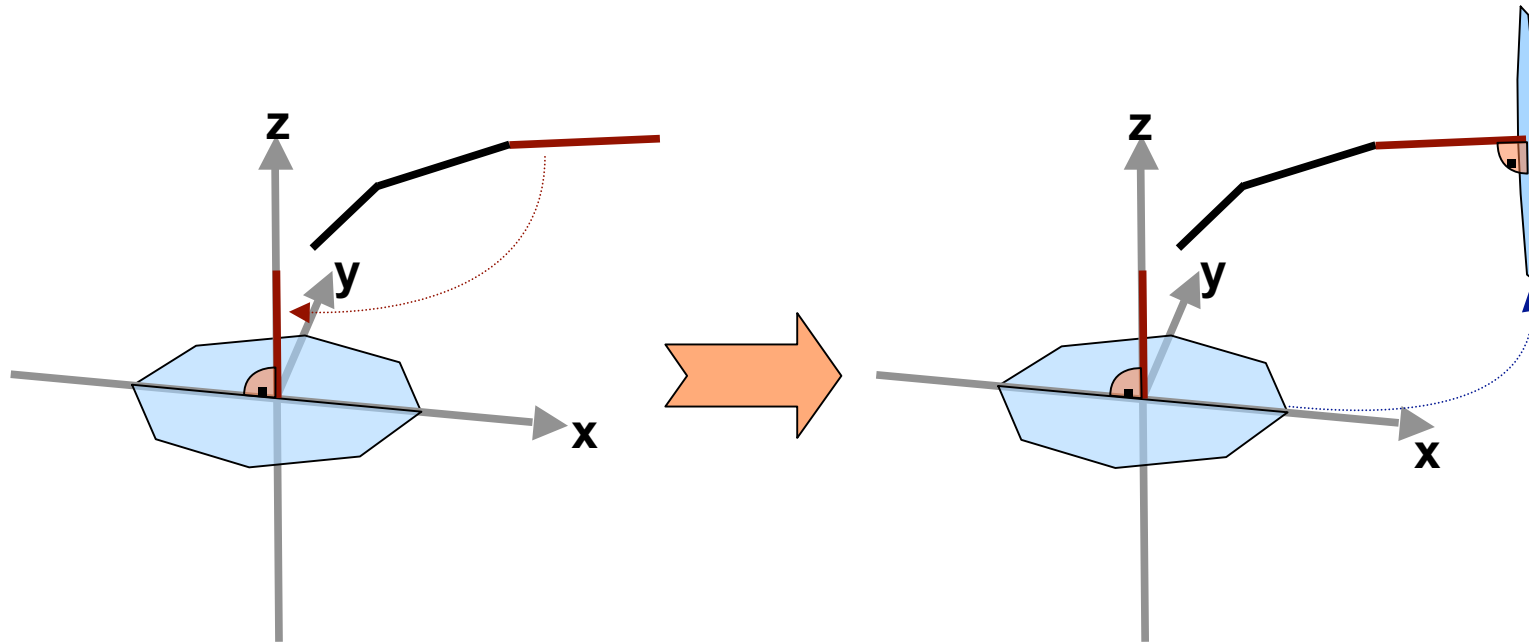
# Network Objects with Turns (2): Solution



*Outer ports perpendicular*

*Inner ports not!*

# Positioning of Outer Ports



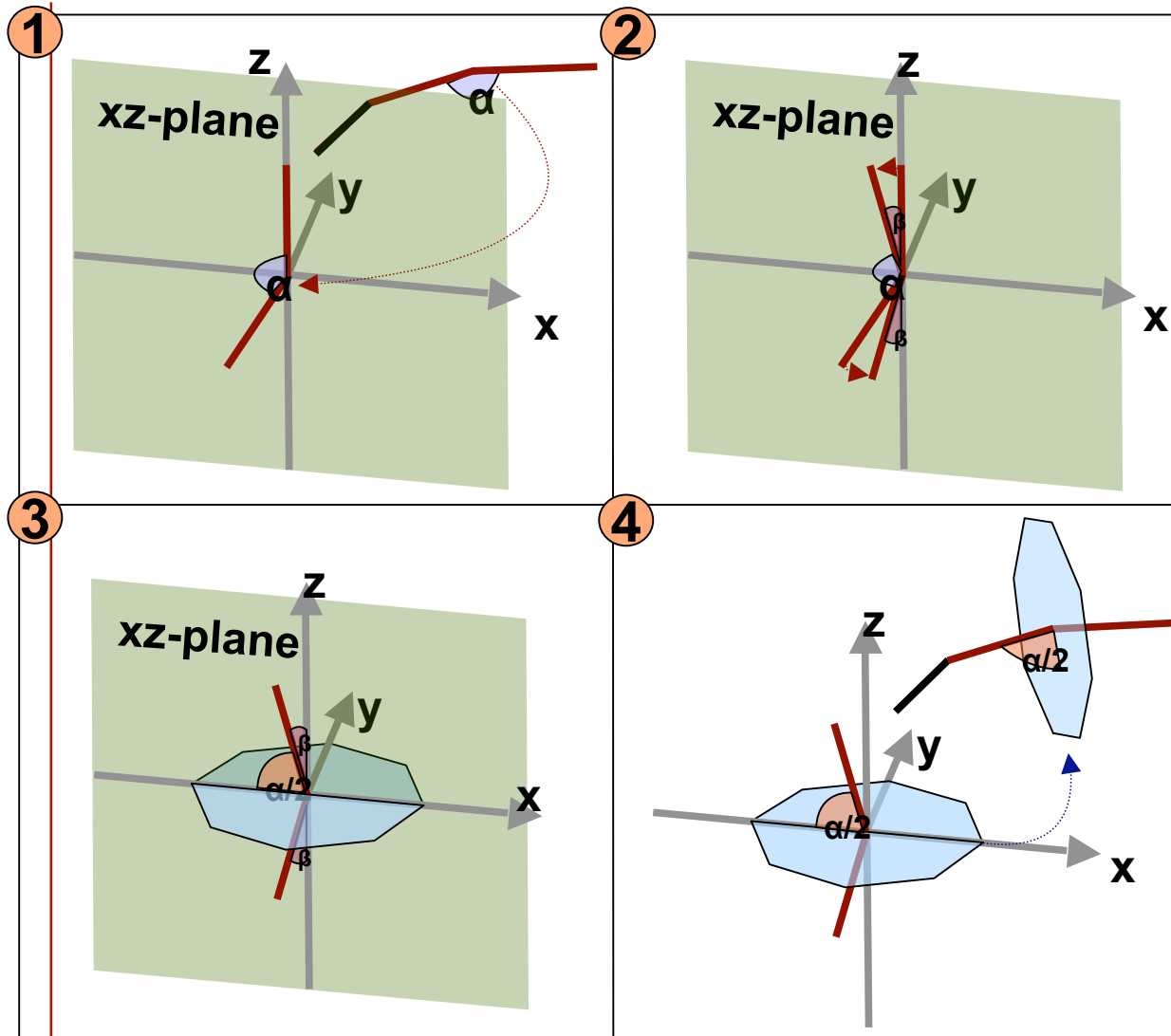
**a) Transform first segment into z-axis with starting point into the point of origin.**

**b) create port in x/y-plane.**

**c) Reverse both transform back.**

**d) Same for last segment and end point.**

# Positioning of Inner Ports



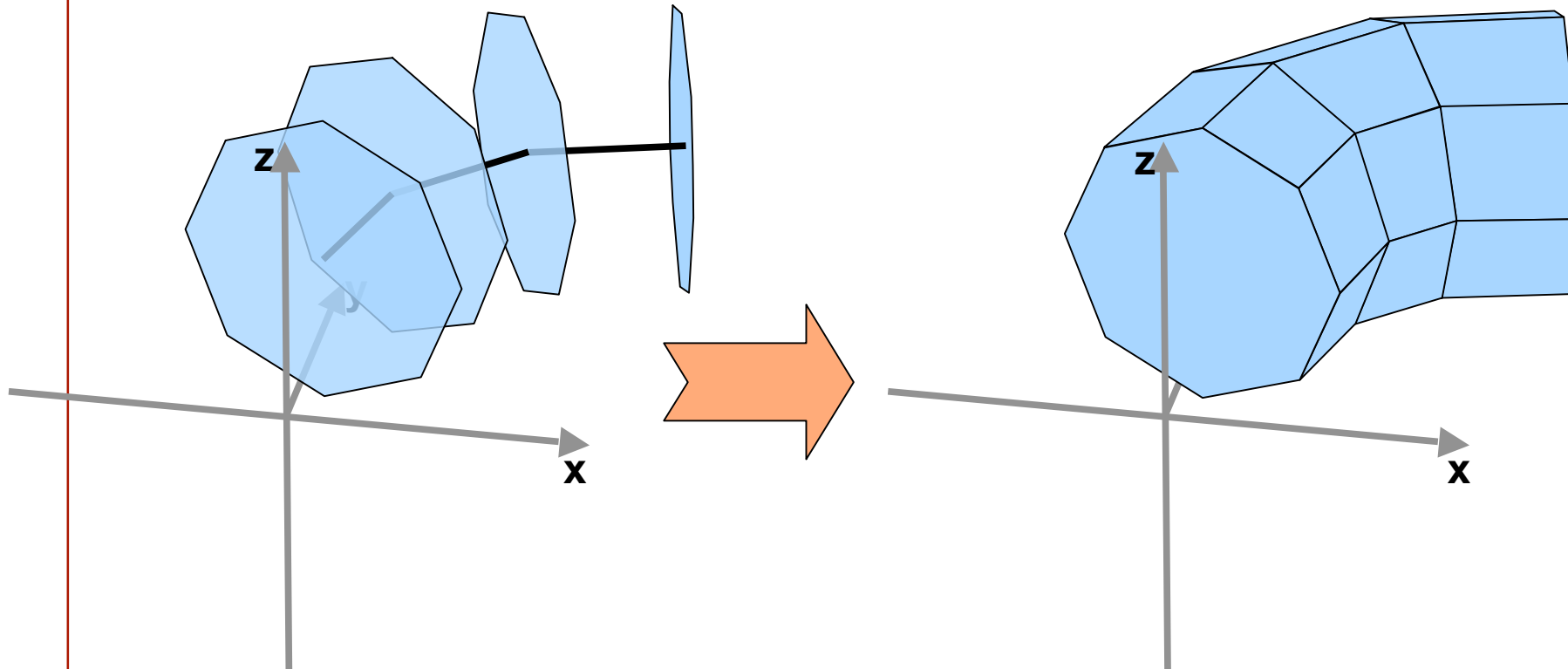
**1. Transform consecutive line segments with sharing point matching point of origin with angle  $\alpha$  staying the same**

**2. Rotate segments around y-axis, so x-axis halves  $\alpha$**

**3. Create port in x/y-plane again**

**4. Reverse transform**

# Creating Surfaces for the Lateral Area



*Create surfaces for lateral area by connecting the ports.*



# *Implementation and CityGML Output*

## DreamCoder

```

DreamCoder for PostgreSQL 2.4 - postgres@localhost[example1] - [PL/pgSQL editor]
File Edit View Database Tools Options Window Help
1 postgresql database - SQLPlex - Database Technology Community for Oracle, MySQL, PostgreSQL D

33 mline text;
34 cof text;
35
36 BEGIN
37 -- call aggregate for points with id of this line
38 SELECT INTO nodes point2ogml(pid) FROM points WHERE pipe =
39
40 -- call aggregate for lines with this line
41
42 SELECT INTO clines lines2ogml(s) FROM lines WHERE pipe = li
43
44 -- wrap this into FeatureGraph and featureGraph
45
46 SELECT INTO fginner xmlelement(name "util_core:FeatureGraph"
47 SELECT INTO fgouter xmlelement(name "util_core:featureGraph"
48

```

DBMS Input

## pgAdmin

```

Query - example1 auf postgreslocalhost:5432 *
Datei Bearbeiten Abfrage Lesezeichen Makros Anzeigen
SQL Editor Graphischer Abfragegenerator
centerline_extraction();
Ausgabefeld
Datenanzeige Zerlegung Meldungen Historie
HINWEIS: DELTA variable is set to 2!
HINWEIS: CREATE TABLE erstellt implizit eine
CONTEXT: SQL-Anweisung »create table centerl:
PL/pgSQL function "centerline_extraction" line

```

Shapefile Output

## pgsql2shp

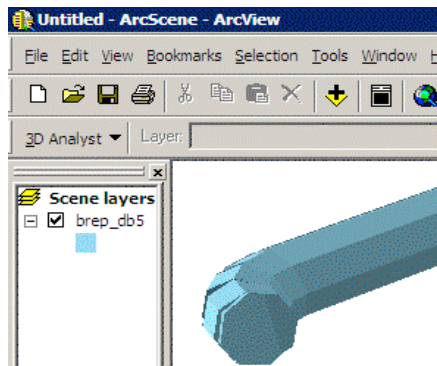
```

C:\WINDOWS\system32\cmd.exe
E:\Programme\PostgreSQL\8.4\bin>pgsql2shp.exe
RCSID: $Id: pgsq12shp.c 5181 2010-02-01 17:35:55
$=1 USE_PROJ=1 USE_STATS=1
USAGE: pgsq12shp.exe [<options>] <database> [<sc
pgsq12shp.exe [<options>] <database> <que
OPTIONS:
-f <filename> Use this option to specify the
to create.

```

Visualization

## ArcScene

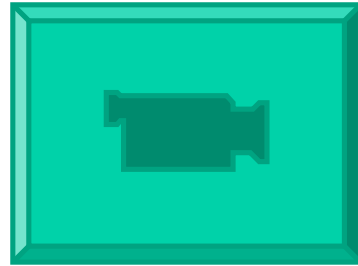


Revision

# Comparison of Memory Requirements

Database	n	Network Objects	Vertices B-rep	Vertices Sweep Rep.	Ratio
example1	12	4 simple, 1 cross fitting (4 segments)	3960	16	0,0040
example2	12	3 simple, 1 t-fitting (3 segments)	2367	12	0,0051
example3	12	3 simple, 1 t-fitting (3 segments)	1955	12	0,0055
example4	12	4 simple, 1 cross fitting (4 segments)	2921	16	0,0055
example5	12	1 simple, 1 with turns (5 segments)	1470	8	0,0054
example6	12	2 simple, 1 with turns (5 segments)	1732	10	0,0058

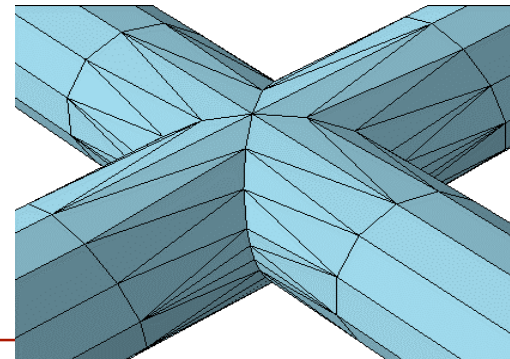
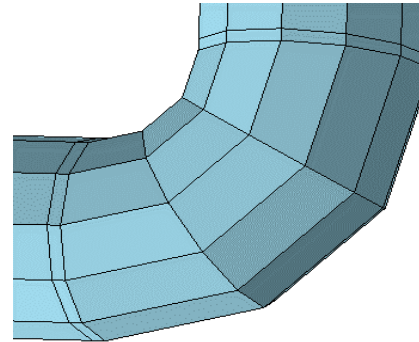
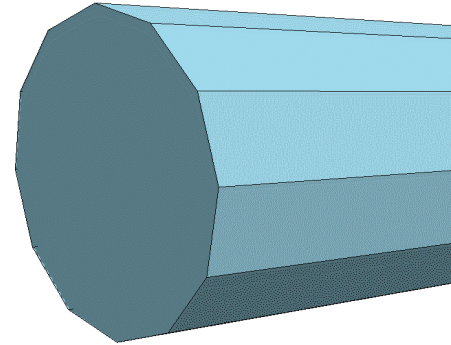
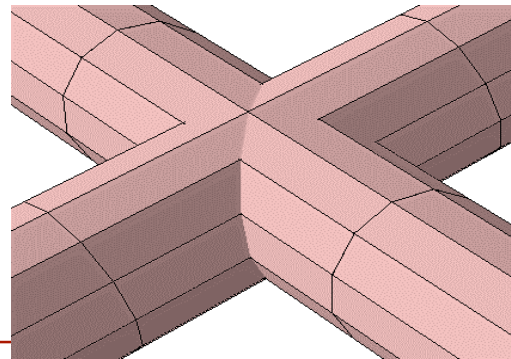
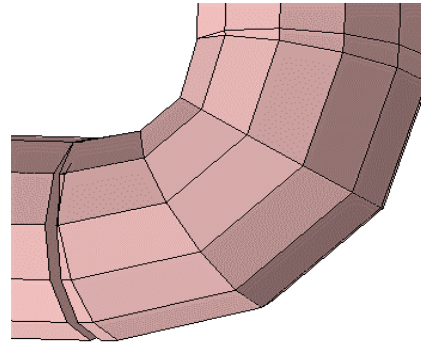
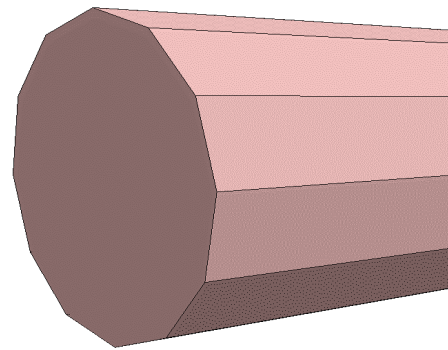
The screenshot displays the Aristoteles3D application window. The left pane shows a hierarchical tree structure for a CityGML file named 'example1.gml'. The tree starts with 'CityModel' and branches into multiple 'util\_comp:Section' elements, each containing 'util\_core:FeatureGraph' and 'util\_core:InteriorFeatureLink' sub-elements, which in turn contain 'util\_core:Node' and 'util\_core:InteriorFeatureLink' members. The right pane is a 3D view area, currently showing a large purple 'X' over a black background, indicating that the 3D model is not rendered. The bottom status bar shows the CRS (Coordinate Reference System) and coordinates: X: 37629.954, Y: 38773.575, Z: 294.930. The Windows taskbar at the bottom includes the Start button, taskbar icons for various applications, and the system tray showing the time as 14:12.





# Comparison to Original BREP

**SWEEP  
REP**



**BREP**

- ***New GIS datatype compatible to sweep representation as used in CAD provides similar detail as BREP***
- ***Storage space is reduced by a factor of 1:300 – 1:500***
- ***BREP can be transformed to sweep representation***
- ***BREP can be recreated from sweep representation***
- ***Output sweep representation conforms to CityGML Utility Network ADE***

For more Information please contact

[ihijazi@igf.uos.de](mailto:ihijazi@igf.uos.de)