Applications of 3D modelling of the built environment

GEO1004: 3D modelling of the built environment



3D geoinformation

Department of Urbanism Faculty of Architecture and the Built Environment Delft University of Technology



Visibility analysis











Other applications

- Visualisation (eg for gaming, tourism, navigation, etc)
- Energy demand estimation (and potential for retrofitting)
- Computational fluid dynamics (eg for wind speeds, air quality, effects on buildings, etc)
- Shadow casting (eg for building permits, visibility analysis, improving energy demand/ solar potential calculations, etc)



Applications based on visualisation?



Some MSc Geomatics theses



- Motivation: create (rough) indoor geometry from widely available outdoor geometry
- Definition of a CityGML LOD2 with interiors (LOD2+)
- Compute interior geometry from exterior geometry + number of storeys
- Compute net internal area

MSc thesis in Geomatics

Automatic enhancement of CityGML LoD2 models with interiors and its usability for net internal area determination

Roeland Boeters

2013

June

A





Exterior in LOD₂

Buildings bodies are prisms Simple roof shapes Thematically classified boundary surfaces No openings in the exterior geometry

Interior in LOD2+

Storeys within building bodies are prisms Attic storey shapes corresponding to roof shapes Thematically classified boundary surfaces No openings in the indoor geometry



Indication of storeys



5.2 GENERATION RULES AND DATA IN

 \wedge





Indication of storeys











Wall thickness

Type	year y	storeys x	$t_{\rm ext} [{\rm cm}]$	$t_{\rm shared} \ [\rm cm]$
Non-stacked	<i>y</i> < 1970	$x \leq 2$	27	11
	C C	$x \ge 3$	27	12
	$1970 \le y \le 1985$	<i>x</i> = 2	27	10
		x = 3	28	12
		x = 4	27	9
	y > 1985	<i>x</i> = 2	28	13
		x = 3	30	12
		x = 4	25	12
Stacked	<i>y</i> < 1970	$x \leq 5$	29	12
		$5 < x \le 10$	38	11
		x > 10	25	9
	$1970 \le y \le 1985$	$x \leq 5$	28	11
		$5 < x \leq 10$	26	11
		x > 10	29	12
	y > 1985	$x \leq 5$	30	12
		$5 < x \le 10$	38	13
- 1		x > 10	35	15
Other types	<i>y</i> < 1970	x = 1	14	14
		$x \ge 2$	31	11
	$1970 \le y \le 1985$	x = 1	14	14
		$x \ge 2$	30	10
	<i>y</i> > 1985	x = 1	14	14
		$x \ge 2$	36	13



This robes can have different shapes. If an exact buffer is required in all direction of the second the desired ki sum is an expensive operation and runs in $O(n^3m^3)$ where n and m are the sum of vertices, halfedges and shalfedges of polyhedron 1 and polyhedron 2 respectively (Hachenberger, 2007). A quick performance test shows that Minkowski sum of a triangular face with an approximated sphere (wi<mark>th 80 triangu</mark>lar facets and 42 vertices) takes about 2-3 times longer than Minkowski sum with a cube whereas the accuracy in the perpendicular offset is then still limited. Therefore a cube is chosen as robot, which is expected to be good eno<mark>ugh as the wa</mark>lls of most buildings are perpendicular to each other. The Minkowski-sum is the vector sum of the point sets of both polyhedra. Therefore when using a cube for applying the offset, a rotation should be applied. This is illustrated for a 2-dimensional case in Figure 39. The offset to the line is not the same for both cases. A rotation thus needs to be applied, such that the square is aligned with



ouffered face from the original solid (set difperation)

erent shapes. If an exact buffer is required in should be a sphere where the radius equals ortunately the Minkowski sum is an expens in $O(n^3m^3)$ where n and m are the sum nd shalfedges of polyhedron 1 and polyhechenberger, 2007). A quick performance test sum of a triangular face with an approximated ar facets and 42 vertices) takes about 2-3 times sum with a cube whereas the accuracy in the then still limited. Therefore a cube is chosen cted to be good enough as the walls of most sular to each other.

is the vector sum of the point sets of both when using a cube for applying the offset, a ed. This is illustrated for a 2-dimensional case to the line is not the same for both cases. A

e applied, such that the square is aligned with perations, may produce multiple disjoint solids. The robot must be scaled, such that the radius is need to be handled separately so that in the outsquals the desired offset.

of which the normal vector ger than 100°, thereby exclosed by conne maximum building height. Which height is marked as characteristic

mption is that each storey has the same height, subtraited inomethics in the eight of the line. total height can be divided by the amount of heights at which the built would be split.

Classifying surfaces









Results





Results

are not snapped to the eaves of the root, because the distance between









Results





Number of buildings



DIFFERENCE NET INTERNAL AREA LOD2+ - BAG FOR STACKED BUILDINGS [%]

Net internal area (stacked)

Cumulative %



Net internal area (non-stacked)



DIFFERENCE NET INTERNAL AREA LOD2+ - BAG [%]

Cumulative



- Motivation: update 3D city models from designed BIM models (including potentially interiors)
- Fill gaps using Minkowski sum to increase size of elements
- Merge elements using Boolean set union
- Reclassify surfaces

Automatic generation of CityGML LoD3 building models from IFC models

MSc thesis in Geomatics by Sjors Donkers

December 2013

Department of GIS TechnologyOTB Research Institute for the Built Environment









Methodology (semantics)







Geometric Transformation Geometric & Semantic Refinement



CityGML

Methodology (geometry)



(a) input

(b) union

(c) dilation

(d) result

(e) erosion

(f) final result













(b) IFC without roof



Results





(d) CityGML without roof











(b) IFC without roof





Results





(d) CityGML without roof













(b) IFC without roof



Results





(c) output CityGML

(d) CityGML without roof











(a) Building where part of the roof is missing

Ssues



(b) A church missing a base slab



- Motivation: repair 3D models so that they can be used in applications
- Voxelisation
- Reconstruction of mesh
- Obtain semantics and export

AUTOMATIC REPAIR OF 3D CITY BUILDING MODELS USING A VOXEL-BASED REPAIR METHOD



A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of

Master of Science in Geomatics

by

Damien Mulder

M.Sc Geomatics Thesis

June 2015



Fixing 3D models















Voxelisation



Voxelisation: overshoot **A**•7 \mathbf{O}





Voxelisation: gap ()()0 0 \mathbf{O} 0 \mathbf{O} \mathbf{O} 0 \bigcirc 0 000000000000000000000000




Voxelisation: shooting rays





Majority counting: overshoot $(\mathbf{0})$ (\mathbf{O}) ()00000000000000000000000000

















Marching cubes







Dual contouring

٦.



exterior voxel

grid edge with sign change

interior voxel



Dual contouring





Dual contouring









(a) The original polygonal model



(c) Dual Contouring result

Full process

0
0
0
0
0
0
0
0

(b) Marching Cubes result

(d) Pressing result







Results









Results









Artefacts









Results





- Motivation: improving the accuracy of the location of personal weather stations for urban heat island research
- Generate potential locations
- Evaluate them through skyview + solar modelling



Delft University of Technology





Air temperature (°C)

32 - 34

Urban heat island



Traditional weather stations





Personal weather stations









Crowdsourced weather data









Behaviour 2018-05-22



Potential locations

Constrained buffer

BGT building

Footprint

BGT road Footprint Potential location Given location



Potential locations





Clipped point

- Generating point
- Initial point





Potential locations





Skyview computation











Analysis



Results





Experiment











Experiment







- Motivation: automate some (simple) building permit checks using a 3DCM
- Formalisation of regulations
- Store necessary data in CityJSON extension
- Automate some checks (car + bicycle parking)

MSc thesis in Geomatics

Automatic building permits checks by means of 3D city models

Jialun Wu 2021





Formalisation of regulations

For residential buildings: BUH40 = Count BU (function."home") AND (A(BU) 40 m2) BUH40-65 = Count BU (function."home") AND (40 A(BU) 65 m2) BUH65-85 = Count BU (function. "home") AND (65 A(BU) 85 m2) BUH85 = Count BU (function."home") AND (85 m2 A(BU)) Rules (must be true) IF BU(function) = "home"THEN MinNPP = (BUH40*2) + (BUH40-65*3)+ (BUH65-85*4) + (BUH85*5)

- NewParkings \geq sum(MinNPP) + sum((MinMQPP/parkingArea))



New attributes to store

	Information	Explanation	Sources	name in sources
7*Attributes	id	Bag id of building	BAG	identificatie
	+function	Function of buildings	BAG	NR_XXX
		included in codelist		(different functions)
	+groundHeight	Elevation above sea level	3D BAG	ground-0.00
		at the ground level		
	measuredHeight	Elevation above sea level	3D BAG	roof-0.75
		at rooflevel		
	+zone	Zone where the building	Digital map	zone
		is located		
	+height_valid	Indicate the height	3D BAG	height_valid
		is valid		
	+total_area	Gross floor area (GFA)	BAG	Calculation results
		of building		on different attributs
2*Geometry	type	geometry type of	BAG	type
		buildings		
	coordinates	a lists contain	BAG	coordinates
		[x,y,z] 3D coordinates		



CityJSON extension





CityJSON extension

"68": {

"type": "Building", "toplevel": true, "attributes": { "+height_valid": 1, "+non_residential": 1, "+groundHeight": 0, "measuredHeight": 28.0, "+total area": 1371.5687999999998, "+discount_factor": 0.95, "+min_bicycle_parking_spaces": 117, "+min_car_parking_spaces": 78, "+function": "catering I"



Generating required info





Programming checks

 $N_40 = int(f['properties']['N_40'])$ $N_40_65 = int(f['properties']['N_40_65'])$ $N_85_120 = int(f['properties']['N_85_120'])$ $N_120 = int(f['properties']['N_120'])$ if f['properties ']['zone'] == 'A': $N_{-}120 * 1.2$) if f['properties']['zone'] == 'B': $N_{-}120 * 1.2$) if f['properties']['zone'] == 'C': $+ N_{-}120 * 1.8$)

```
oneb['attributes']['+min_car_parking_spaces'] = int(
N_40 * 0.1 + N_40_65 * 0.4 + N_65_85 * 0.6 + N_85_120 * 1 + 0.40 = 0.10
oneb['attributes']['+min_car_parking_spaces'] = int(
N_40 * 0.1 + N_40_65 * 0.5 + N_65_85 * 0.8 + N_85_120 * 1 + 0.40
oneb['attributes']['+min_car_parking_spaces'] = int(
N_40 * 0.1 + N_40_65 * 0.6 + N_65_85 * 1.4 + N_85_120 * 1.6
```



Results: tool

Minimum Bicycle and Car Parking Spaces for New Buildings INPUT (Buffer map) INPUT (new buildings) Output

 \times





- Motivation: use 3DCM for space heating demand calculations
- Develop CityJSON extension with all required information
- Implement space heating models
- Use implementation to improve extension design

MSc thesis in Geomatics

Development and Testing of the CityJSON Energy Extension for Space Heating Demand Calculation Özge Tufan | 2022




Development of the CityJSON Energy Extension

Assessing the use of the Energy ADE vs. the Energy ADE KIT profile

Determining the mapping rules

CityJSON Energy Extension as a semi-direct translation from the Energy ADE KIT Profile

Validation through the use case

Validation through cjval

Valid CityJSON Energy Extension

Valid CityJSON + Energy Extension file Space heating demand calculation

Choosing the calculation method to be used

Deciding on the simplifications and assumptions to be made

Collection of needed input data

Creating a CityJSON + Energy Extension file with the needed input data for the use case

Calculation of space heating demand

Enhancing the CityJSON + Energy Extension file with output space heating demand values

Improvements to the CityJSON Energy Extension



Storing new (complex) geometries







New attributes

Net internal area Class Function* Usage Measured height Relative to terrain Roof type Year of construction Footprint area* Storeys above ground* Storeys below ground* Building name* Is single part # of adjacent buildings LoD2 volume* LoD max Building (pand) ID List adjacent buildings Surface ID Parent building ID Surface name Azimuth Inclination Direction LoD2 area Surface normal

Exludes internal structural elements

- Type of use of the building, e.g. residential, mixed-use
- Further description of the class, e.g., health, business
- Whether the building is still in use
- Height of the building, in *m*
- Whether the building is (entirely) above or below the terrain
- E.g. slanted, single/multiple horizontal
- Construction year of the building
- Footprint area, calculated from the LoD0 geometry, in m^2
- Number of storeys situated above ground level
- Number of storeys situated below ground level
- Unique name of the building
- Boolean value to show whether the building has *BuildingParts*
- Number of topologically adjacent buildings
- Building volume, calculated from the LoD2 geometry, in m^3
- Maximum LoD present for the building
- Unique ID of the building
- Building (pand) ID of topologically adjacent buildings
- Unique ID of the BoundarySurface
- Building (pand) ID of the building that the surface belongs to
- Unique name of the BoundarySurface
- Azimuth of the surface, in degrees
- Inclination of the surface, in *degrees*
- Direction of the surface
- Surface area, calculated from the LoD2 geometry, in m^2
- Normal vector of the surface



New attributes

"extraAttributes": { "Building": { "+buildingType": {...}, "+constructionWeight": {...}, "+volume": {...}, "+floorArea": {...}, "+heightAboveGround": {...} }

"Build1": { "type": "Building", "geometry": [...], "attributes": { "+buildingType": "singleFamily", "+constructionWeight": "heavy", } }



New City Objects

```
"extraCityObjects": {
                                     "OutdoorTemperature": {
                                       "type": "+WeatherData",
  "+WeatherData": {
    "type": "object",
                                       "attributes": {
    "properties": {
                                         "weatherDataType": "airTemperature",
      "type": {...},
                                         "values": "RegularTimeSeries1",
                                                   //ID of TimeSeries object
      "attributes": {
        "type": "object",
        "properties": {
          "weatherDataType": {...}, "RegularTimeSeries1": {
          "values": {...},
                                       "type": "+RegularTimeSeries",
          "position": {...}
                                       "attributes": {
                                         "values": [2.61, 4.82, 5.91, 9.32,
        }
                                            14.73, 16.12],
```

}

• • •



Test data







Results: heating energy demand





- Motivation: unreliable or non-existent information in IFC models
- Automatically create shapes of rooms, storeys and apartments
- Built on IfcOpenShell

MSc thesis in Geomatics

Automatic building feature detection and reconstruction in IFC models

Jasper van der Vaart 2022





Current IfcSpaces







(a)







(a)



(c)





(b)



















(C)





(d)







(a



(c)





(b)



(d)























Computing apartments









Results









Results



Results





Recommendations

- GEO5014: Geomatics as support for energy applications
- GEO5015: Modelling wind and dispersion in urban environments
- Your own MSc thesis



Sources of images

- [2-6]: Filip Biljecki (paper on application of 3D city models and PhD thesis)
- [9-20]: Roeland Boeters (MSc thesis and related paper)
- [21-29]: Sjors Donker (MSc thesis)
- [30-48]: Damien Mulder (MSc thesis)
- [49, 51-62]: Yixin Xu (MSc thesis)



Sources of images

- [50]: Anna-Maria Ntarladima (MSc thesis)
- [63-70]: Jialun Wu (MSc thesis)
- [71-78]: Özge Tufan (MSc thesis)
- [79-93]: Jasper van der Vaart (MSc thesis)

