Final Report

Technical Specifications for the Construction of 3D IMGeo-CityGML

Authors:
Jan Blaauboer, Bentley
Joris Goos, Rotterdam Council
Hugo Ledoux, TU Delft
Friso Penninga, Den Haag Council
Marcel Reuvers, Geonovum
Jantien Stoter, Kadaster, Geonovum & TU Delft
George Vosselman, NCG & ITC, U Twente

Date:
8 January 2013
corrections: August 2014

Version:
2.02

List of Figures

Table of Contents

Chapter 1

Introduction

This chapter consists of an introduction to this report and a reader's guide.

This document is Activity 3's Final Report with regards to the six Dutch 3D Pilot Phase 2 activities. Activity 3 was entitled: *A Technical Specifications Document for the Construction of 3D IMGeo-CityGML*.

The 3D Pilot is an initiative of the Cadastre, Geonovum, The Dutch Commission for Geodesy and the Ministry for Infrastructure and the Environment. Over 100 organisations worked together to develop tools to support 3D implementation.

The Pilot Phase 2 consisted of six activities: 1. Generating Sample Data IMGeo 2.0 – CityGML, 2. Designing and Building a 3D Validator, 3 Designing a Standard Specification for the Construction of 3D IMGeo Data 4. Describing a Generic Approach towards the Maintenance and Updating of 3D Data. 5. Collecting Sample 3D Killer Applications 6. Aligning the BIM - CityGML – IMGeo standards. These were run both parallel to and in co operation with each other. Each group has put their experiences into a final document. These results, plus a management summary, make up the seven reports of the 3D Pilot Phase 2.

These reports are intended to provide an introduction to the most important results and are not, therefore, complete. More details can be found in the presentations which were given during the six information sessions (see http://www.geonovum.nl/onderwerpen/3d-geo-informatie?tab=documenten) and the http://www.geonovum.nl/onderwerpen/3d-geo-informatie website. This website will be continuously updated with new insights and developments and as well as documents which refer to this report itself.

This report describes not only what is required in order to begin implementing 3D but also the choices which need to be made before IMGeo topography can be marketed as ready for the addition of 3D Geometry.
This report provides, within the constraints of a number of basic premises and choices, a starting point for the writing of technical texts which can then be adapted to the organisation's specific requirements for 3D IMGeo data.

This report is not, therefore, about legal, contractual agreements, defining a good report layout upon delivery etc. Nor is it about descriptions of procedural agreements, communication or organisational requirements.

Chapter 2

Scope

This chapter describes the scope of the document. It has been written to make clear what can and cannot be expected from this report.

2.1 Providing Help with Basic Premises and Choices.

This report does not provide a design specification document which can be used as a basis for a tender. It does, however, give a number of requirements and recommendations. The authors advise that these requirements would need to be met and the recommendations followed in order to achieve a workable result. Working from a starting point of what applications and data are needed, it gives insight into how to put together a design specifications document from a technical perspective.

2.2 Data

The scope is not restricted to data already at hand. Suggestions are given about how to obtain further data. As far as possible, the report is about extant resources already within organisations or those which can be bought at a relatively low cost.

2.3 IMGeo as Foundation

IMGeo 2.1 2D is used to form the basis for the construction of 3D topography. IMGeo 2.1 is a CityGML Application Domain Extenstion (ADE). This means that the model is a semantic, geometrical and syntactic extension of CityGML. Software packages which support CityGML can also support ADE's such as IMGeo.

A choice could also have been made to work from 2D topograpy such as GBKN (Large Scale Base Data from the Netherlands; a previous version of IMGeo). Our deciding factor was that we felt it was better first to update current 2D topography to 2D IMGeo and then to use this as a foundation for the construction of 3D.

2.4 LODs and Automatic Generation

This report focuses on those LODs which can be automatically generated and which support the most applications. CityGML supports LOD's 0 up to and including 4. LOD0, LOD1 and LOD2 are Levels of Detail which can be automatically generated[1]. LOD3 and LOD4 require intensive manual labour.

The report describes LOD0, LOD1 and LOD2 for all topography. In addition tunnels, constructions, city furniture and trees for LOD3 have also been included. This is for those organisations who would like to take it all one step further.

In CityGML LOD3 is the 3D level where a building's individual windows and doors, constructions or other objects are drawn. The difference between LOD2 and LOD3 is that currently LOD3 can only be executed manually. LOD3 is, due to the technical requirements of the specifications, outside our scope. On the one hand this is because it is not directly necessary for the applications for 3D city models which are currently

[1] The data will need to be, where necessary, manually rectified after the generation process has been completed.

in focus and on the other hand because it is impossible to define in general what kind of technical text an organization will require. This depends entirely on the desired level of detail.

LOD4, the level at which information within the building can be modelled, has a different application to 3D city models and is outside the scope of this report.

The application of LOD3 and LOD4 and the creation of the construction industry's IFC/IFD standard are happening concurrently. BIM is a widely used term. A BIM, however has a far wider context than 3D and is particularly used in relation to the integrated design and maintenance of building elements, contracts, plans, project agreements and such like.

Chapter 3

Requirements and Recommendations as Input to Technical Specifications.

This chapter describes choices for 3D data implementations and their allied Requirements and Recommendations. The chapter provides an organisation with important input of the technical specifications which can then be added to as necessary.

This chapter has been made as comprehensive as possible and is based on the authors' knowledge and experience and a 3D Pilot group's review. It works from the assumption that there is already a 2D IMGeo-CityGML file. 3D IMGeo begins with BGT/IMGeo geometry in 2D. Its 3D extension consists of:

- LOD0 representations of all IMGeo polygons, in which distinctions are made between:
  - Objects at ground level. These together form a topological structure (Chapter 3.2.1 LOD0 for IMGeo Polygons at Ground Level)
  - Objects that are located above or below ground level, including fly-overs which are connected at ground level (Chapter 3.2.2 LOD0 Representations of Polygons at a Level Unequal to 0)
- Volume representations (LOD1, LOD2, LOD3) which connect to LOD0 at ground level:
  - LOD0-LOD1-LOD2 Buildings (BuildingParts, Other Constructions) (Chapter 3.3 Building Specifications)
  - LOD1-LOD3 Bridges, Tunnels (Chapter 3.4 LOD1- LOD3 Tunnels and Bridges)
  - LOD1-LOD2 Plant Cover (Chapter 3.5 Plant Cover in LOD1 and LOD2)
  - LOD2-LOD3 Trees and City Furniture (Chapter 3.6 Trees and Other City Furniture in LOD2 and LOD3)
- Texture (3.7 Texture)

3.1 IMGeo 2.0 CityGML: Generic Requirements

As already stated, IMGeo-CityGML forms the foundation for the procedures outlined in this document. Because of this, it is advised that the required 3D data is delivered in the IMGeo-CityGML format.

Requirement 1. The 3D data should be structured according to IMGeo-CityGML format. The standards can be found at: http://www.geonovum.nl/dossiers/bgtimgeo/destandaard

Requirement 2. The IMGeo-CityGML data must comply with CityGML 2.0[2]. In some cases we have more stringent requirements than CityGML. If this occurs it will be specified in the following requirements.

IMGeo-CityGML can work in two ways; firstly as an exchange format and secondly as an information model in which the IMGeo objects to be maintained are described.

Exchange Format

It may be a good idea to order other exchange formats in addition to IMGeo-CityGML, depending on your application(s). If, for example, CAD-software is being used, then having the data delivered in the relevant

[2] OpenGIS® City Geography Markup Language (CityGML) Encoding Standard, version 2.0, www.opengeospatial.org/standards/citygml

CAD format could be useful. Alternatively, if publication on Google Earth is the goal, then the data can be purchased in Google KML. Having the product delivered in extra formats means having to weigh up the difference between internal conversion costs (in hours) versus the costs charged by an external provider.

Recommendation 1.	If necessary, IMGeo-CityGML can be purchased in different formats.

Conceptual Model

The IMGeo-CityGML conceptual model contains objects that need to be maintained. These IMGeo-CityGML objects need to be represented by the software (storage and user –interface) in use within an organisation.

3.1.1 Reference System

CityGML-IMGeo uses the EPSG:7415[3]Spatial Reference System (SRS). This is a compounded SRS with RD (EPSG:28992) for the XY dimension and NAP (EPSG:5709) for the Z dimension. This must be explicitly referred to within the CityGML file.

Requirement 3.	Use the Spatial Reference System (co ordinate system) EPSG 7415.

3.2 Specifications for LOD0 Representation.

In CityGML's LOD0 a representation in 3D is shown as a surface. In actual fact this is a 2.5D visualisation, with every x,y co ordinate having a z co ordinate as well. In 3D IMGeo at ground level, every polygon stemming from 2D IMGeo has its own 2.5D surface, represented as a Triangulated Surface. This 2.5 surface *per object* is the result of the point data's 'constrained triangulation' in which IMGeo's 2D boundaries are used as breaklines (constraints). The AHN (High Resolution Lidar data set of the Netherlands) can for example be used as height data in the 'constrained triangulation' process. As a result the IMGeo object's boundaries are returned as a (collection of) the sides of a triangle in the Triangular Irregular Network (TIN). In order to illustrate sufficient height, vertices at boundaries need to be added and must also show the height within the surfaces by using a TIN representation. LOD0 IMGeo features come in two types; features at ground level (see Paragraph 3.2.1) and features above or below ground level (Paragraph 3.2.2). Due to a combination of these two types of 2.5D features sometimes more than one z value can be assigned to the same x,y location, something which is not possible within a sole Triangulated Surface. This will be explained in more detail below.

In the rest of this chapter specifications for the following will be given:

- LOD0 for IMGeo polygons at ground level (3.2.1 LOD0 for IMGeo Polygons at Ground Level)
- LOD0 representations of features above or below ground level (3.2.2 LOD0 Representations of Polygons at a Level Unequal to 0)
- The combination of both types of features (3.2.3 Completeness of IMGeo Surfaces at LOD0)
- The closing of a topological LOD0 surface with the help of Terrain Intersection Curves (3.2.4 Closing the topological surface of LOD0 with the help of TIC's)

3.2.1 LOD0 for IMGeo Polygons at Ground Level

The BGT objects at ground level form a planar partition (no gaps or overlaps). This principal is taken over into IMGeo LOD0 representation. That means that all separate features together has to form a 2.5D

[3] In agreement with the 'raamwerk van geo-standaarden' (Geo Standards Framework), Version 2.2.

topological closed structure at ground level. In CityGML a Terrain Intersection Curve is an explicit surface in the terrain TIN which describes the intersection with ground level.

The IMGeo classes which in this manner receive an LOD0 representation as a part of ground level are :

- Traffic Area
- Auxiliary Traffic Area
- Land Use
- Plant Cover
- Waterbody,
- Unclassified Object (no longer relevant after Stage 2[4])
- BuildingPart (for more details about its LOD0 representation, see 3.3.1)
- Other Construction (for more details about its LOD0 representation, see 3.3.1)
- Construction Area
- Boundary (if it is a surface)

With this principal in mind, CityGML-IMGeo extends the modelling principal for topology that CityGML uses for Land Use to other classes. This 2.5D topological principal for Land Use is defined in the CityGML specification as:

> "LandUse objects can be employed to establish a coherent geometric/semantical tesselation of the earth's surface. In this case topological relations between neighbouring LandUse objects should be made explicit by defining the boundary LineStrings only once and by referencing them in the corresponding Polygons using XLinks. The result is a land use tesselation, where the geometries of the land use objects are represented as triangulated surfaces. In fact, they are the result of a constrained triangulation of a DTM with consideration of breaklines defined by a 2D vector map of land use classifications."

A similar principle is used for CityGML-IMGeo, although it is now extended with other feature classes at ground level which are also modelled with a LOD0 representation (modelled for more classes than in CityGML). At the same time the LandUse class is restricted to Land Use as it is used in IMGeo and values which in fact represent water, roads, railways, plant cover and such like are not used but modeled instead with the relevant IMGeo class.

Requirement 4. Every object in IMGeo is represented by a LOD0 geometry i.e. a TIN surface (triangulatedSurface) per object (tessellation of the object's footprint) . The LOD0 terrain is formed by a collection of such adjacent TIN surfaces, with recognizable object boundaries (constrained TIN)[5].

Requirement 5. The LOD0 geometries of all IMGeo polygons (water, road, building, land use, vegetation) at ground level should form a planar partition in 2.5D (no holes or overlap).

[4] In Stage 1 (2012 up to and including 2015) of the construction of the BGT in 2D (a statutory task) objects for which the classification is unknown or which still do not fall within the BGT classification can be classified as an Unclassified Object. In Stage 2 (2016 up to and including 2019) these Unclassified Objects have to become BGT classified. An object can no longer be 'Unclassified' after Stage 2.

[5] Although IMGeo permits Arcs, these are not employed in a 2.5D terrain representation because the TIN is constructed from vertices and edges. Due to this, the requirement that the same breaklines should be employed cannot always apply. This has been communicated to IMGeo (http://www.geonovum.nl/dossiers/bgtimgeo/meldingen-imgeo) with a request to offer a solution.

Requirement 6. The height difference between the terrain in reality and its representation in TINs is allowed to be maximum X cm. X can be dependent on the object type (for example another X can be chosen for hard surfaces with curbs than that for pasture). Individual apexes are acceptable until up to 3 times X, but connected pieces of a TIN of more than Y m2 may deviate no more than this X cm.

In 2D IMGeo objects are divided at ground level. In 3D parts of these boundaries may be delineated by triangulations with a vertical interval where in principal a vertical surface should be (see Figure 1). Many algorithms still cannot work with vertical surfaces, which is also true for TIN implementations in software. Vertical surfaces in a TIN can be avoided by requiring that the upper and under side of such a face are a minimal oblique angle away from each other.

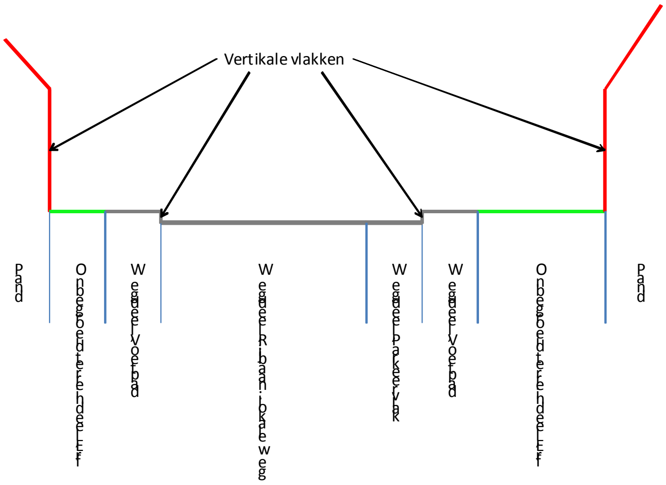
Vertikale vlakken
Pand
Onbegroeid terreindeel Erf
Wegdeel Voetpad
Wegdeel Rijbaan lokale weg
Wegdeel parkeervak
Wegdeel Voetpad
Onbegroeid terreindeel Erf
Pand

Figure 1: Vertical Surfaces which have to be approached with minimal sloping surfaces

Requirement 7. Vertical surfaces in the TIN may not occur, because many GIS software crashes on such data. Instead, vertical surfaces should be approached by maximum sloping surfaces. How this should be done depends on which objects are left and right of the vertical jump. The sloping surfaces need to be attached as follows to the relevant object:

- Boundary of (Auxiliary) Traffic Area - Terrain Area, to the (Auxiliary)Traffic Area
- Boundary of (Auxiliary) Traffic Area - Terrain Area to (Auxiliary)Traffic Area
- Boundary (Auxiliary) Traffic Area – (Auxiliary) Traffic Area, to the highest object
- Boundary Terrain Area- Terrain Area, to the highest object
- Boundary of (Auxiliary) Traffic Area/ Water Body/ Terrain Area/ Division – BuildingPart, to the BuildingPart
- Boundary of (Auxiliary) Traffic Area/Water Body/ Terrain Area/ Division – Other Construction, to the construction
- Boundary of (Auxiliary) Traffic Area/ Water Body/ Terrain Area/ Division – Construction, to the construction
- Boundary of (Auxiliary) Traffic Area/ Water Body/ Terrain Area – Division, to the Division
- Boundary of Object – Waterbody, to the object

Extremely precise vertical intervals can be necessary for particular applications. In Figure 2 the height variations of roadways, footpaths, cycle paths as well as driveways and other entrances are applied. The result could be used to help constrain the chaos resulting from extreme weather such as a heavy rainstorm[6]. It could help determine in which direction the rainwater would flow and what measures could be put in place to manage this situation.

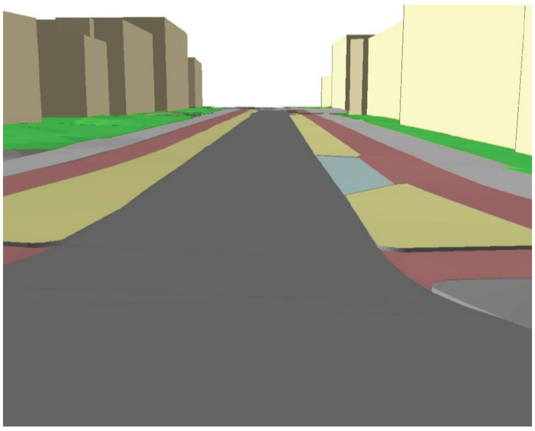

Figure 2: Illustration of traffic areas and its auxiliaries and height variation according to the function of the object. Curbs are represented with vertical surfaces

Requirement 8. When very precise vertical intervals between specific objects are necessary, this should be recorded in the technical specifications. A minimum height should be defined and vertical intervals must be visible. Examples are the height jumps at the location of curbs

Requirement 9. Waterbodies are always flat, horizontal surfaces.

The requirements for LOD0 buildings (BuildingParts and other constructions), are further discussed in 3.3.1.

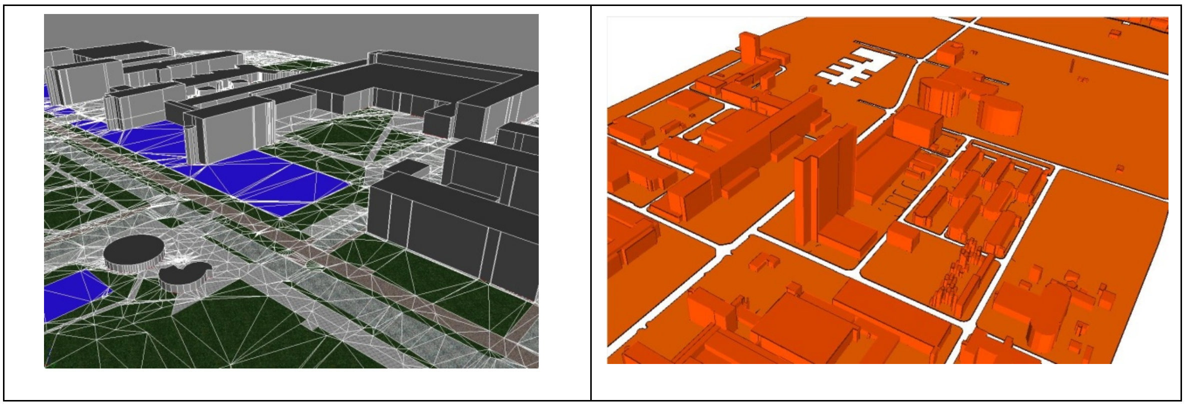

Figure 3: Triangulated terrain with IMGeo surfaces defined upon it and LOD1 buildings.

3.2.2 LOD0 Representations of Polygons at a Level Unequal to 0

An attribute relativeHeightposition was introduced to the 2D BGT in order to model objects which are situated either above or below ground level. According to this rule, water which is open to the sky and above ground is classified as level 0. Objects which are positioned higher (for example something which

[6] Hydrocity is an application which has been developed in response to this problem (built by NEO and ITC U Twente)

bridges water) are assigned a higher level number. Objects which are positioned below ground level (for example a tunnel) receive a lower level number.

This level number can be used to model traffic areas which cross each other at LOD0 or water bodies and traffic areas (for example a road and a road or a road and a river) which do the same as is described in this chapter. The representation of tunnels and bridges in higher LOD’s is discussed in 3.4.

Polygons above or below ground levels are modelled in LOD0 as follows (see Figure 4, taken from Oude Elberink 2010). Ground level objects in LOD0 are modelled as discussed in section 3.2.1. Polygons above and below ground level are positioned at their true height in the third dimension by using their 2.5 LOD0 representation (TriangulatedSurface). It may be necessary to add new 2D boundaries to these junctions to connect these objects to the LOD0 representation at ground level, as can be seen in Figure 4(b), in order to define an extra variation in 3D. It may also be necessary to add terrain surfaces above (or below) these objects in places in 2D where no explicit stacking of objects is modelled so that the LOD0 representation has no holes at ground level; see Figure 4 (c and d).

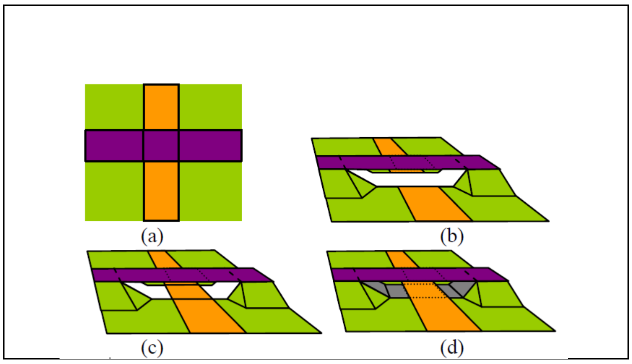

Figure 4: The concept of relativeHeightposition in 3D and adding polygons to close gaps in 3D

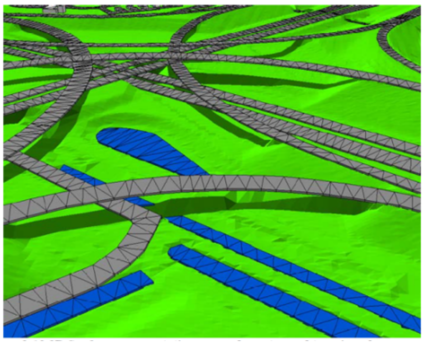	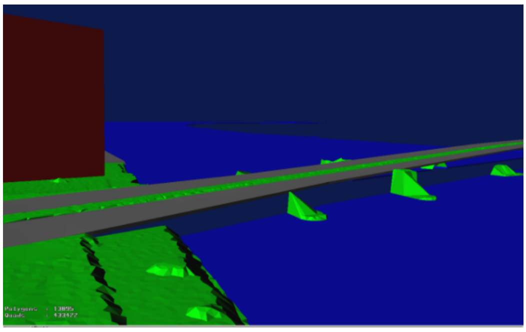
An example of a 3D surface representation of (crossing) traffic areas, water bodies and terrain areas taken from[7]	A 3D model of a 3D Pilot test area, generated by Oude Elberink, ITC U Twente from AHN2 and TOP10NL

Figure 5: gives two real-life examples to which this principle has been applied. (Examples of surfaces above and below ground level.)

The Dutch Ministry of Infrastructure and the Environment’s Digital Topographical Map (DTB)[8] has a similar approach to modelling topography in 2.5D. In addition, this modelling technique is being used in the

[7] http://www.rws.nl/kenniscentrum/contracten/data_eisen/digitaal_topografisch_bestand/

Province of Noord Brabant's 2.5D topographical modeling approach and examples of it can be seen in Figure 6 below. The DTB, however, does not recognise any height information within surfaces. This is in contrast to the LOD0 surface geometries (Triangulated Surfaces) as described in this IMGeo-CityGML implementation guide.

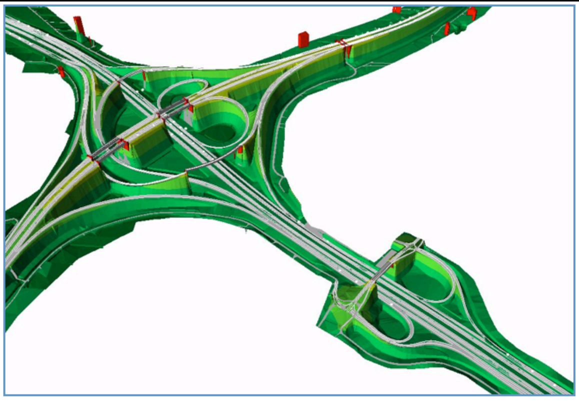	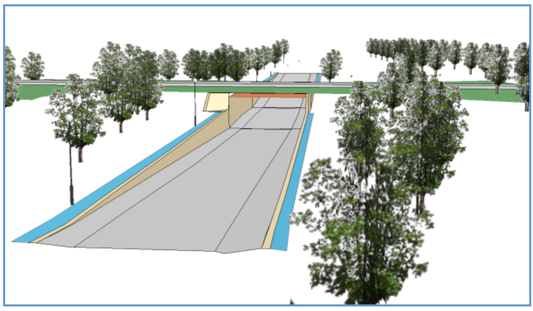
DTB, source: RWS	Topographical map, Province of Noord-Brabant

Figure 6: Further examples of surfaces above and below ground level.

Requirement 10. IMGeo polygons which lie above or below ground should be modelled with a triangulatedSurface which intersects with a topologically consistent LOD0 representation at ground level. The result is a stacking of 2.5D objects

3.2.3 Completeness of IMGeo Surfaces at LOD0

The IMGeo Data Catalogue describes that as a general rule open to the sky, surface water is assigned a relative height level of 0 (meaning: located at ground level). As a result of this, bridges above water are classified at levels higher than 0 and are not included in the LOD0 representation at ground level. A user is often used to a bird's eye view as this is the approach of traditional cartographic reproductions. This view would leave only the bridge visible, and not the water under it, and ensure that the pattern of roads remain recognisable. The figure above shows how features at a level unequal to 0 in LOD0 have to be represented. A correct and complete LOD0 IMGeo representation is only possible if all polygons, irrespective of height level, are added. It is important to realise this, because it goes against the prevailing belief that an LOD0 representation is only a basic 'drape' of a 2D file over a height file. Comparable 'drapes' of 2D files (topgraphical or air photography) such as often occurred when AHN-1 was being introduced, draped the object with the highest elevation (birds-eye view) at each location on the height model and therefore did not deliver a result conforming to an IMGeo LOD0 representation at ground level (height level = 0).

Requirement 11. All IMGeo polygons should be added to IMGeo's LOD0 representation, i.e. both those at ground level as well as the ones above and below ground level

3.2.4 Closing the topological surface of LOD0 with the help of TIC's

3D objects can be placed above or in the terrain model. CityGML has defined a Terrain Intersection Curve (TIC) in order to integrate the terrain model and the 3D objects properly. A TIC indicates where a 3D

[8] http://www.rws.nl/kenniscentrum/contracten/data_eisen/digitaal_topografisch_bestand/

object touches the Terrain Model. In addition to this, a TIC is used to triangulate by using a ClosureSurface value. This results in a closed Terrain Model.

In CityGML[9] TIC's can be used for building and building parts; bridge parts and bridge construction elements; tunnel and tunnel parts; city furniture objects and generic city objects. As has already been mentioned, these TIC's do not need to be used in IMGeo-CityGML in locations where footprints have been included in the terrain. TIC's are, however, necessary to ensure a closed Terrain Model where tunnel entrances are located.

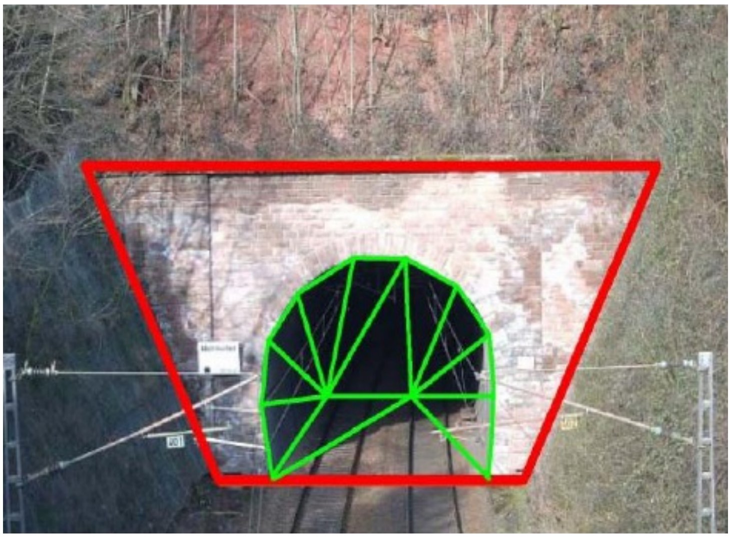

Figure 7: TIC (the opening's outer ring (green) and the triangulated ClosingSurace (green TINs).

Requirement 12. Terrain Intersection Curves (TIC's) should be used to make ClosingSurfaces where 3D objects float above or sink below the Terrain Model. A topologically correct Terrain Model will be the result of this.

3.3 Building Specifications

The word 'Buildings' should be read to include the following object types from 3D IMGeo-CityGML:

- BuildingPart
- Other Construction

When the term 'building' is used in this document, it should be taken to include these both types of IMGeo Objects.

The following LOD's have been developed for buildings: LOD0 for footprints and roof edges (see 3.3.1), LOD1 for simple blocks (3.3.2) and LOD2 where roofs are modelled and texture can be added(3.3.3) etc. LOD3 which models windows, doors and the like or LOD4 which penetrates the building's interior fall outside the scope of this document.

The rest of this chapter describes the following specifications:

[9] See Chapter 6.5 CityGML 2.0 Specifications http://www.opengeospatial.org/standards/citygml

- LOD0 representations of buildings (3.3.1)
- LOD1 representations of buildings (3.3.2)
- LOD2 representations of buildings (3.3.3)
- Solid geometry which models LOD1 and LOD2 representation of buildings (3.3.4)

3.3.1 LOD0

Buildings at LOD0 in CityGML can be represented in two ways; a footprint (in accordance with BGT geometry, i.e. 2D footprint geometry in the large-scale topography) and a roof edge (in general) according to BAG geometry (2D geometry in the building and address register representing the outline of buildings as seen from above). Both the LOD0 representation of a building footprint and a roof edge have to be a 'horizontal surface' pursuant with CityGML specifications. If a footprint is in reality situated on a slope then the lowest value has to be used (as specified in CityGML). It is also stated that the base in LOD2 must be congruent with the LOD footprint.

Although modelling a horizontal surface with footprints has many advantages, this approach also has disadvantages, particularly with buildings where the footprint is not horizontal in reality. These drawbacks have been raised at the OGC CityGML work group and are being discussed. These are:

a. Buildings on a slope (dike, dune) cannot be modelled as such. The sloping footprint has to be approximated by a horizontal surface
b. In order to make sure that in these situations building footprints intersect the terrain, vertical surfaces are required to fill the gap between footprint and terrain. This is specifically true when working with high resolution as is most often the case in the Netherlands (for example the AHN2). These vertical surfaces are not present in reality and moreover a lot of software cannot work with them (see Requirement 7 for more information).
c. Two BuildingParts on a slope that touch another in a vertex cannot be modelled topologically correct. The footprints are modelled with a vertical interval that doesn't exist in real life. In this situation one can choose to put both footprints at the same height. But what should be done with a terrace house on a slope? Neither the artificial differentiations in height nor putting all footprints at the same elevation are true to reality.

Here we recommend following the current CityGML specifications and stipulating that the LOD1 and LOD2 footprint of a building be horizontal. For LOD0 we recommend that the footprint will be determined by the connection of the terrain and the building and thus not to comply to the CityGML specifications. Geonovum is in discussion with OGC on this issue

Requirement 13. The ground surface of a building at LOD1 and LOD2 must be horizontal. The ground surfaces should, though, be determined per individual building and not per block of buildings . This surface is then positioned at the lowest height of the terrain at the location of this surface so that the building sinks "in" the terrain and gaps between ground surface are avoided.

Requirement 14. Notwithstanding the CityGML specification, LOD0 footprint must be determined where the outside wall touches the terrain.

3.3.1 LOD1

LOD1 buildings can best be described as 'block-shaped'. One height, for example an average number, is assigned to 2D building geometry. 3D geometry can be extracted through the use of extrusion. Extrusion is a process by which the original 2D building geometry (polygon) is considered as a ground plan and a duplicate of this polygon is placed at a certain building height as a roof surface. The volume is closed by vertical surfaces which join the ground plan and roof surface together.

Extrusion is based on absolute height positioning; in this case with respect to the terrain. Extrusion is a function that is supported by diverse software packages, which enable users with a little GIS expertise to construct a LOD1 building representation.

From LOD1 representations on, buildings are no longer modelled as surfaces, but as volumes. In order to be able to process and manipulate the volumes correctly, all of the volumes need to be valid (be, among other things, a closed volume). Chapter 3.3.4 surveys in more detail what is required of a 3D Building Model in order for it to be topologically correct.

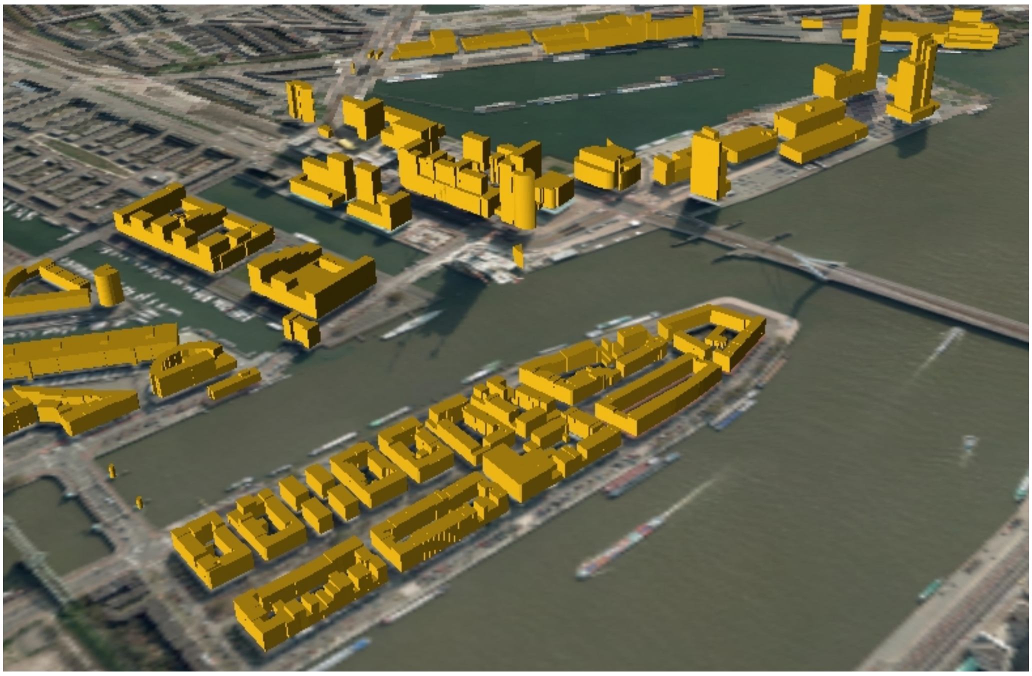

Figure 8: Sample LOD1 buildings

Requirement 15. Every IMGeo building should be delivered with its own LOD1 representation.

Requirement 16. The building height is the median of the height of the points which are positioned within the footprint.

Requirement 17. If a building's roof has significant vertical intervals (for example a church with a tower), then these differing height levels should be distinguished in 3D, particularly if the interval is greater than, for example, 1.5 metres and if the surface area is greater than 4 square metres.

Clarification: This is an important recommendation, but also impacts on the process itself. Some existing algorithms do not take these height intervals into account when constructing LOD1 buildings. Therefore, it could be important that the supplier adjusts the software so that it can.

Requirement 18. The lower surfaces of the building's block geometry must correspond to the 2D and LOD0 geometry in IMGeo

Clarification: This recommendation also implies that the geometry of surfaces which are subdivided into BuildingParts (for example a wall between two adjoining terrace houses) are modelled twice in the data, once for the first (BuildingPart)- object (for example as that object's _WallSurface) and once for the neighbouring (BuildingPart-)object (for example as that object's _WallSurface). It is only in this manner that, during maintenance, each object can be mutated, without affecting the other.

Requirement 19. The lower surface of a LOD1 block should be horizontal, taking the lowest point of the footprint's terrain triangulation as its height (see LOD0 building).

Requirement 20. Through passage must be guaranteed for buildings that bridge either water or roads. This may be achieved artificially, for example by positioning the underside of such a construction five metres above the road. An example of this can be seen in the Figure below which illustrates the Nationale Nederlanden building above the highway A12.

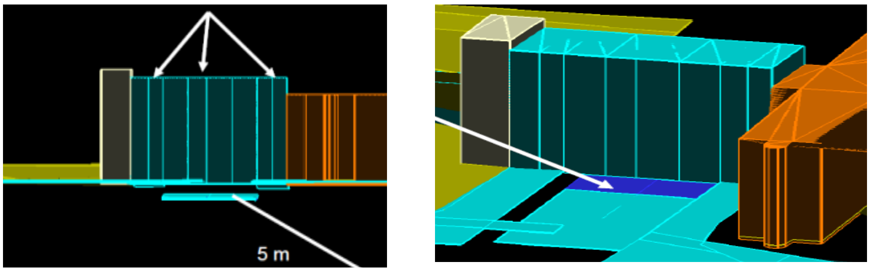

Figure 9: Artificial passage

Requirement 21. The geometry of LOD1 Buildings should be defined in CityGML as GML:Solids (closed volumes, also from below) and not as GML:MultiSurface, which is permitted for LOD1 buildings.

Clarification: A GML:Solid is a composition surface which is again modelled as a collection of adjoining surfaces. A building object modelled as a solid, therefore, does not mean that individual surfaces can no longer be accessed as these have been modelled as a part of the solid geometry. Being able to access individual surfaces can be particularly useful for visualisation purposes, for example in order to give a roof another texture or colour. By contrast, if the LOD1 and LOD2 representations have been modelled as multisurfaces, then there is no topological connection between the surfaces and the solid is also not explicit. If the surfaces of a multisurface geometry form a closed entity/whole, however, then a solid (a composite surface) can be formed. A multi surface is not invariably closed, which is why we recommend that building objects that form close volumes are modelled as such, i.e. as solids.

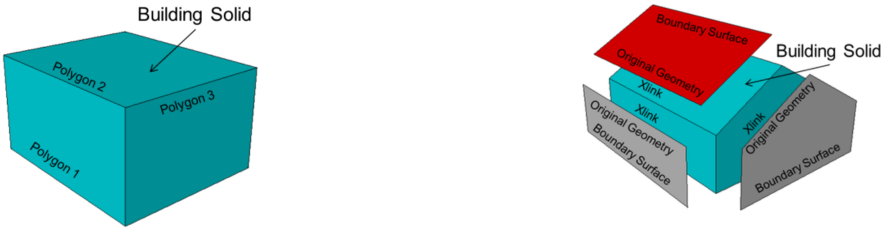

Figure 10: Left: An LOD1 solid (without surfaces) Right: An LOD2 solid with accompanying modelled surfaces (image taken from CityGML specifications)

3.3.3 LOD2

An LOD2 solid has the form illustrated by Figure 10 on the right.

Requirement 22. Each LOD2 IMGeo building is modeled by the GML:Solid geometry type in which the semantics of the boundaries (surfaces) are made explicit (e.g. footprint, roof surface, wall surface). LOD2 buildings can be represented as a collection of a solid with other geometry types such as a multisurface for a roof overhang.

Buildings and, in particular, roofs and roof shapes can be modelled in different ways in LOD2. This can lead to different sorts of deviations between the models and the buildings' true shape. Dependent on the intended application, a modelling approach needs to be chosen which results in the least intrusive deviations. First we will discuss the various options. After that we detail the requirements that can be stipulated in relation to the ambitioned building model.

Working from Existing 2D Building Outlines

The rule is that buildings should be modelled with the location of the walls consistent with the 2D file's building outlines. The BGT and the BAG can be used as a source for 2D building outlines. The BGT registers the geometry of the footprint (where the building touches the ground level) while the BAG registers building geometry as the outline seen from above. Specifically with karge roof overhangs or with buildings on pillars there exist significant differences between the two types of geometry.

The choice for one or the other data has consequences for modeling if the BAG and the BGT boundaries are not the same. This is illustrated in the Figure below by a BuildingPart with a roof overhang. If the 3D model's boundaries are taken from the BGT (second picture), then the model's walls are in the right place, but the roof overhang has been removed. As a result of this is that the model's roof gutters are positioned higher than they really are. If the BAG is used (third picture) then the roof is fully modelled with its gutters in the correct place but the walls have been moved out. Both cases give problems when applying texture to the 3D model. Using the BAG in particular will lead to roof overhangs being shifted by automatically applied texture. This is why it is ideal to use both the BGT and the BAG geometries in the 3D reconstruction of buildings. The BGT can be used to determine the footprint of a building, while the BAG supports the outlines of the roof surfaces. Only in this way can the roof overhangs be modelled explicitly.

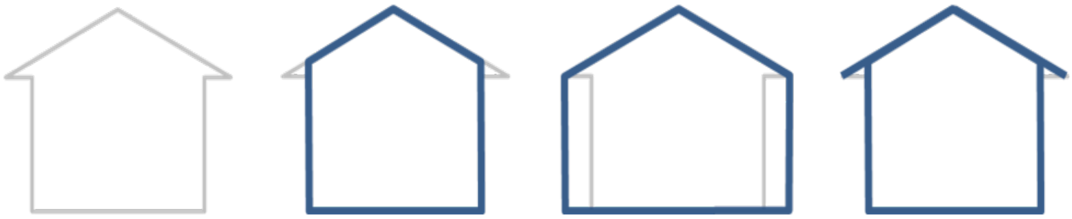

Figure 11: From left to right: a BuildingPart with a roof overhang in light grey, a BuildingPart modelled by working from the BGT, a BuildingPart modelled by working from the BAG and a BuildingPart modelled by working from both the BGT and the BAG.

The Consequences of Working from 2D Outlines.

By working from 2D outlines, 3D models remain consistent with the 2D representation. Small mistakes or deviations in the building outlines can, however, result in intrusive faults or deviations in the 3D Model. When, for example, a rectangular BuildingPart with a pitched roof is not represented as an exact rectangle, this can lead to the pitched roof not being horizontal in the 3D model. When the pitched roof does need to be horizontal, then this can only be achieved by accepting that the plane through the ridge and pitched roof is bent or has a kink.

Combining a number of roof parts can lead to an even more disruptive effect. The Figure below shows a point cloud of a building with two pitched roofs, the colours correspond with points on the four roof surfaces as they are discerned from the point cloud. The location of the corner points as indicated by the arrows are not completely correct in the 2D definition.

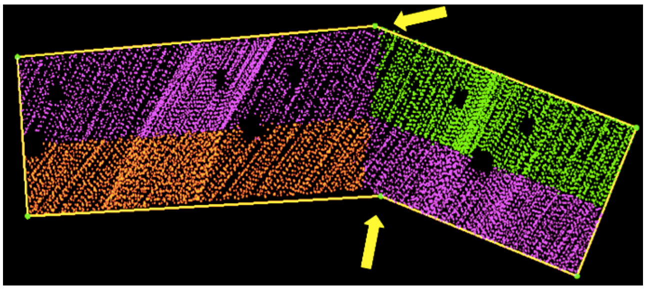

Figure 12: The point cloud of a building with two pitched roofs.

When modelling in 3D a choice has to be made between two ways of drawing a line which joins two pitched roofs together.

1. The line is calculated from the breaklines of the point cloud's surfaces.
2. The line is the connecting line between two corners in the 2D outlines.

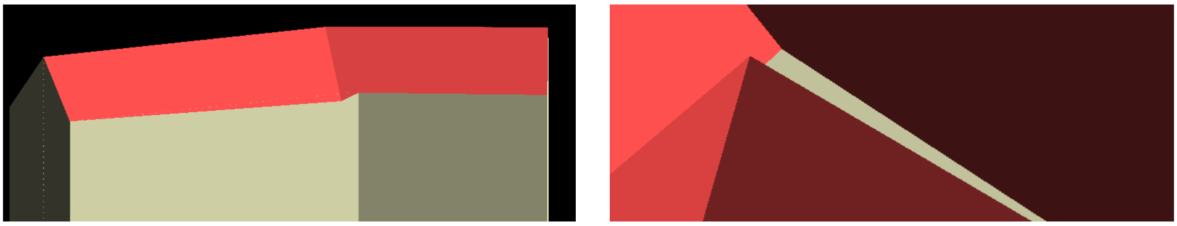

Figure 13: Option 1), Option 2)

Option 1 (Figure on the left) leads to adjoining roof surfaces, but gives the roof overhang a strange kink in case of non-rectangular angles. It can clearly be seen that the breakline between the roof surfaces, as taken from the point cloud, does not connect with the breakline between the walls. In the second case, (Figure on the left) the ridges of the two pitched roofs do not connect. As a result, there are pieces of wall in the 3D model which are not there in reality.

A variant of the second option could be that the ridgelines run, by definition, through the middle of the sides. However, this results in problems with asymmetrical roofs and would still result in pieces of wall on the roof, because due to mistakes in the outline the two pitched roofs cannot be exactly the same width.

In short, whoever commissions the IMGeo CityGML file must realise that retaining the BGT or BAG's 2D geometry means that the 3D model will deviate topologically from reality and that this will result in (small) faults or deviations in the 2D geometry. These topological deviations can only be prevented when the 2D outlines are adapted within the framework of the 3D modelling process. If this is not allowed, then a choice must been made between the two results described above.

Some methods for the reconstruction of roof shapes make use of a library of standard roof shapes (Figure, source: http://www.nachi.org/forum/f11/mitigation-roof-shape-41293/). These methods divide the 2D building outlines in parts, so that each part can be described by a simple standard roof shape. Complex roof shapes consist of combinations of simpler roof shapes. An example of this was seen in the previous chapter where the roof was described as having two pitched roofs. How well complex (compound) roof shapes can be implemented is strongly dependent on the degree to which the 2D building outlines can be successfully segmented. This method has the advantage that roof shapes can be properly modelled by laser altimetrical data (for example the AHN2), even in situations where there is a relatively low point density. A disadvantage, however, is that roof shapes that are not present in the library cannot be reconstructed.

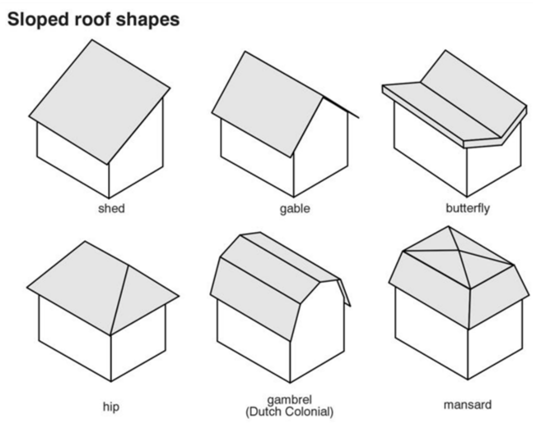

Figure 14: Standard Roof Shapes

An advantage of other methods is that they permit a very wide variation in roof shapes. These methods are frequently only successful in situations of very high point density. It is also possible to combine more than one method. In light of this, it is wise not to prescribe a certain method of building reconstruction, but rather only what the resulting building models need to conform to.

Unity of Modelling

With adjoining properties, such as terrace houses, most people would prefer one 3D model per house when linked with the BAG. It would be, however, unwise to model each house independent of adjoining houses. Due to the noise in the data, small height differentiations may arise between BuildingPart models which are, in reality, the same height. It is therefore recommended that adjoining BuildingParts should first be modelled as one object and, if so desired, to then split them up into one 3D model per BuildingPart.

Building Modelling: What Might be Required

Requirement 23. The locations of the outer walls of 3D building models should be in alignment with the 2D boundaries from the BGT and the BAG (preferably the BGT).

Requirement 24. The roof boundaries of 3D building models should be in alignment with 2D boundaries from the BGT or the BAG (preferably the BAG).

Requirement 25. Building model should be complete in the sense that the combination of all surfaces of one building should form a closed volume, a 3D solid. No surfaces from other buildings should be found within building models. Building models may touch each other but may not overlap.

Clarification: This sometimes occurs when a number of simple roof shapes are combined with each other.

Requirement 26. When a roof overhang is explicitly modeled, roof surfaces should be split at the roof overhang's location to obtain a valid solid geometry. These roof overhangs should be modeled as a (multi)surface and the rest of the roof should form a part of the solid geometry's boundary.

Recommendation 2. If 2D building surfaces are split up by the modelling process, then this should preferably occur in relation to the surfaces in the available point cloud.

Clarification: Option 1) in 3.3.3 (Figure 13)

OR

Recommendation 3. If the 2D building surfaces are split up by the modelling process, this should preferably occur with lines between corner points in the 2D building outlines.

Clarification: Option 2) in 3.3.3 (Figure 13)

Requirement 27. LOD2 roof surfaces with a minimum surface area of X m2 may not deviate more than Y m in height from the corresponding points from the point cloud.

Clarification: Requirement 27 enforces a minimum detail level as well as a precision level of data modeling. Consequently an asymmetrical saddle roof is not modeled as symmetrical because then the deviations would be too high. By keeping the area limit high (for example 4 square meters) this enforces dormers only be modeled if they are large.

Requirement 28. Roof surfaces with a minimum surface area of X m2 may not deviate more than Y degrees in the normal direction from a surface because of the corresponding points from the point cloud. This prevents very flat saddle roofs to be modeled by flat roofs and mansard roofs to be modeled by saddle roofs

Requirement 29. Curved surfaces should be represented by a triangulation where the deviation between the real surface area and the triangulation is no more than X m.

Requirement 30. In the model the roof surfaces' vertices (for as much as they have not been pulled out of alignment by the BAG) must lie within a distance of X m from the closest data points.

Clarification: In Rule 27 it is proposed that data points should be located close by the model surfaces. This is not the case when the boundaries of a roof surface have been modled much too generously, for example when a construction with a 6 m2 flat roof is modelled by a 10m2 construction with the correct height. In order to prevent such deviations, it is also required here that the vertices of a model should be located near the data points

3.3.4 Rules in Relation to the Solid Geometry of LOD1 and LOD 2 Buildings.

All the surfaces of a LOD1 and a LOD2 building should form one or more closed volumes together, represented by the GML type Solid, even if CityGML permits buildings to be modelled with the *Multisurface* type. This is because a solid is the only way a building can be represented as a volume.

A LOD1 building (Building or BuildingPart) can only be represented with a solid. A LOD2 building can be represented by a mixture of a solid and other geometry types such as a multisurface for a roof overhang and a curve for an antenna.

Requirement 31. In LOD1 and LOD2 a building's solids should meet the requirements which are detailed below.

Rules of Validity for a Solid.

Solids which represent an LOD1 or LOD2 building have an exterior boundary which consists of more than one surface. Interior boundaries which represent holes in solids are not permitted (and are also not necessary).

Each surface of an exterior boundary must be a valid Polygon as is defined in the Simple Features Specifications (SFS[10]) for 2D polygons. In this situation the polygons are considered as a 2D surface within the 3D space they are defined in.

According to the SFS these polygons should:

- be (a) planar (surface), i.e. all coordinates should be positioned within the same surface in 3D
- be topologically closed
- not contain double vertices
- be formed by LinearRings which do not intersect and which have no 'spikes'
- have an interior boundary which is made up of a collection of points which are joined to each other

Furthermore, the ground plan's (footprint's) horizontal surface should comply with CityGML2.0's specifications.

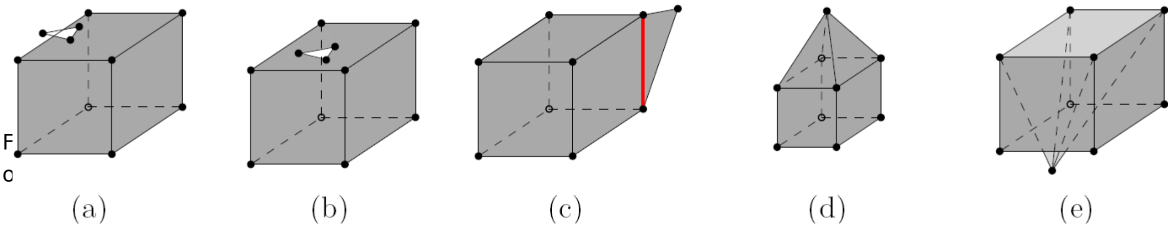

Figure 15: Four invalid solids (a,b,c,e) and one valid solid (d).

If each of a solid's surfaces is a valid Polygon, then the following checklist should be used to determine if the whole solid is valid:

Closed boundary: All surfaces must form a closed boundary that is the solid must be 'watertight'. Each edge of the surface must be shared by two surfaces. Figure 15b shows a solid where this is not the case.

No dangling parts: the boundary surface must not contain so-called dangling pieces. In other words, each edge of the boundary surface must be shared by exactly 2 surfaces, more is not allowed i.e. each edge should be subdivided into two surfaces. Figure 15c illustrates an edge which is divided up into three surfaces and thus results in an invalid solid.

No intersection: the surfaces of a Solid cannot intersect or touch each other. Figure 15e is invalid because the roof's point is positioned under the footprint thus leading to intersections.

[10] OGC. OpenGIS implementation specification for geographic information—simple feature access. Open Geospatial Consortium inc., 2006. Document 06-103r3.

Orientation of surfaces: the orientation of a surface is defined as the ordering of its coordinates. Each surface of a Solid must be oriented such that this ordering is counterclockwise when the surface is viewed from outside the solid. Notice here that this is equivalent to stating that the normals must point 'outwards' (when a right-hand rule system is used). A common mistake is to have the normal of the ground floor point upwards, while to be valid it must point towards the ground.

Overhanging roof

LOD2 buildings can also contain details such as roof overhang, antennae and chimneys. These objects, when not correctly modeled make a solid into an invalid object. See Figure 11 in Chapter 3.3.3 as well.

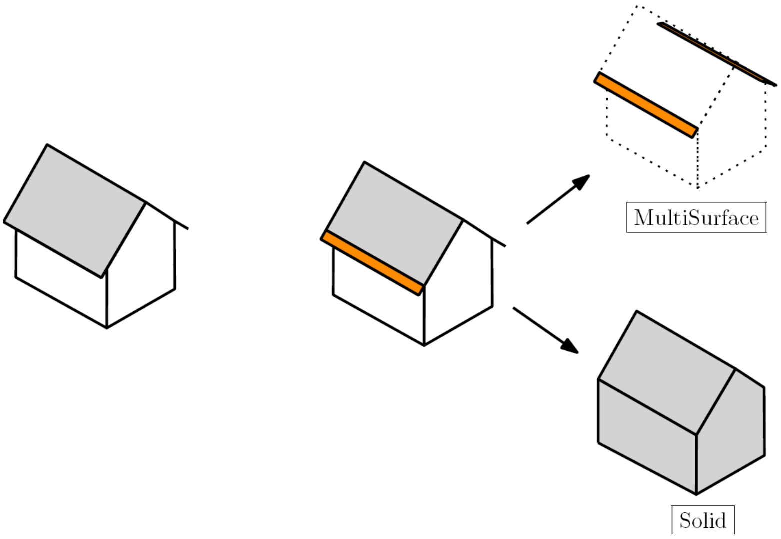

Figure 16: An overhanging roof should be split up so that a building's volume can be represented by a solid.

Figure 16 illustrates the dangling parts of a roof which make the building invalid. According to CityGML specifications the roof should be split up so that the building can be once again represented with a valid solid. Another method, which does not conform to the CityGML standard, is to model the roof as a separate solid. This is illustrated by Figure 14's butterfly roof. Here the roof is modelled as an individual solid. These two solids (one for the roof and one for the rest of the building) would then be aggregated so that there is one solid for the whole building.

NB1: The validator which was developed during the 3D Pilot only checks if a solid adheres to the ISO/OGC rules, including the 2D rules (SFS). But it doesn't check if the roof overhang is connected to the building
NB2: In practice CityGML buildings are still being delivered with multisurface objects instead of solids. Due to this, the validator checks if the multisurface object could be made into a valid solid if it were modelled as such in addition to giving a warning of its presence.

3.4 LOD1- LOD3 Tunnels and Bridges

Modelling Tunnels (Chapter 3.4.1) and Bridges (Chapter 3.4.2) in 3D is completed according to CityGML specifications for Tunnels and Bridges (Sections 10.3 and 10.4 respectively of the CityGML 2.0 specifications).

In IMGeo, it has been decided to model the road which goes through a tunnel or over a bridge separately from its relevant construction for both Bridges and Tunnels. An alternative to this is that the relevant surface only needs to modelled once as geometry and can be referred to from both (construction and road part) objects. This is supported by CityGML. The advantage of saving the geometry twice, however, is that both the road part and the construction are present as separate objects and can be exported independently of each other. A disadvantage is that the roadSurface is modelled redundantly. This means that the roadSurface can be altered independently from the related bridge, which would result in an inconsistent model.

Thus it is important to make sure that both surfaces are in agreement with each other, even if this results in problems with visualisation (for example flickering due to ‘Z fighting’). This is particularly true if both surfaces have a different texture or colour. One possible solution could be that both surfaces are shown with a tick indicating which of the two surfaces are used by the visualisation.

3.4.1 Tunnels

The Tunnel class encompasses representations from LOD1 up to and including LOD4, as illustrated below.

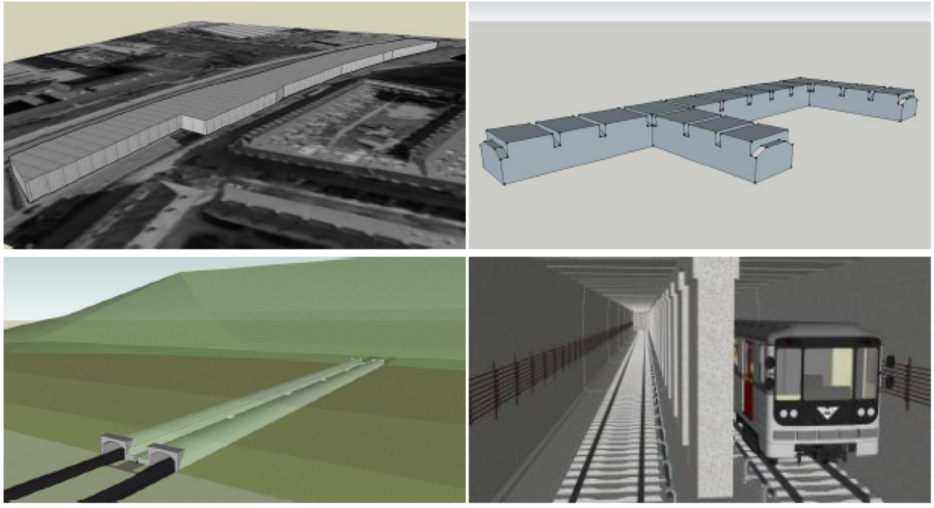

Figure 17: Examples of tunnels in LOD1 (above left), LOD2 (above right), LOD3 (below left) and LOD4 (below right)

TunnelPart is the IMGeo class which should be used for this extension, which is modelled in 2D by a surface. When extending to 3D this surface becomes the footprint of the volume object. How the extension in the different Levels of Detail looks has been described in detail in Section 10.3 of the CityGML 2.0 specifications. Below we have described the most important principles.

Just as with buildings, in LOD1-3 only the outside of the tunnel is defined and consists of the boundary surfaces with the earth, water or air which is situated beside it. The inside of the tunnel is only modelled in LOD4. The result of this is that another object such as a road part or railway line can penetrate it in LOD1-3.

In LOD1 a tunnel consists of a geometry which conveys its volume. The *TerrainIntersectionCurve* (which shows precisely where the tunnel intersects with ground level) is unnecessary because, as mentioned earlier, the footprint of every IMGeo object has to be modelled in the LOD0 representation that defines ground level. The geometry is refined with additional *MultiSurface* and *MultiCurve* geometries in LOD2.

In LOD2 and higher the tunnel’s outer structures are further subdivided by the *_BoundarySurface* and *TunnelInstallation* classes. A boundary surface is a part of the outside of a tunnel with a special function such as a wall (*WallSurface*), roof (*RoofSurface*), ground plan (*Footprint*) or *ClosureSurface. TunnelInstallation* is used for important elements on the outside of a tunnel such as a stairway. In LOD3 openings (doors and windows) can be represented as *_BoundarySurface.*

In LOD4 the space inside the tunnel can also be modelled with *HollowSpace. HollowSpace* allows the tunnel to be actually entered such as by disaster management simulations or incidence of light.

If a tunnel consists of two parts which have different geometry and/or attributes than each other, then the tunnel can consist of two TunnelParts, as illustrated below:

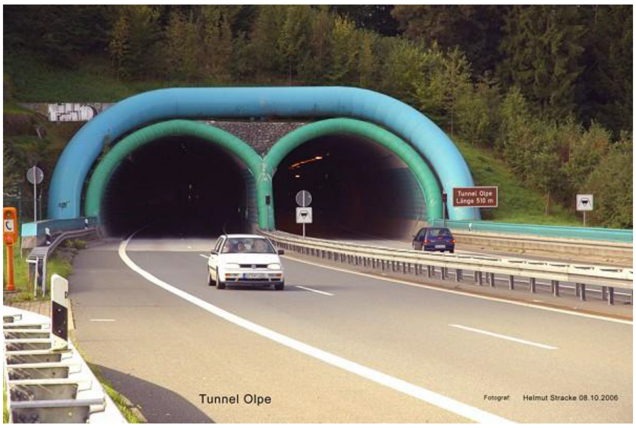

Figure 18: A tunnel modelled by two tunnelparts.

3.4.2 Bridges

Bridges can also be modelled from LOD1 up to and including LOD4, as illustrated below.

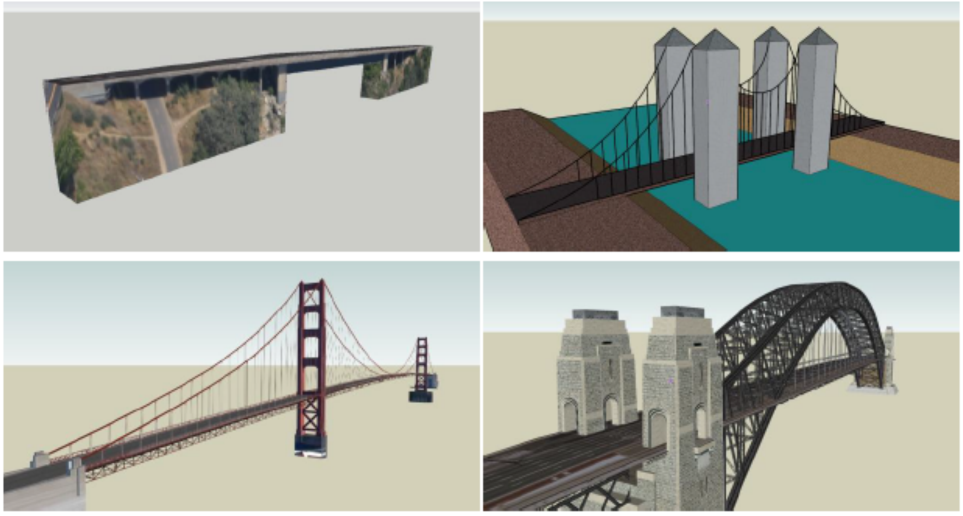

Figure 19: Examples of CityGML bridge models: LOD1 (above left), LOD2 (above right), LOD3 (below left) and LOD4 (below right).

When extending 2D IMGeo to 3D, the different parts of a bridge defined in IMGeo (BridgeConstructionElements) can be used, nl: dek, landhoofd, pyloon, sloof, pijler. These have a surface geometry in 2D. The parts of bridge at level 0 should be integrated as footprint in the terrain (see 3.2.1) in order to ensure that the bridge connects at the right 3D position (and consequently making a TerrainIntersectionCurve unnecessary).

Both moveable and fixed bridges can be modelled by the CityGML class Bridge and – just as with the class Tunnel – a bridge encompasses representations from LOD1 up to and including LOD4. Each LOD encompasses a 'solid' geometry type for a bridge (in the case of IMGeo the 'BridgeConstructionElements'). An example at each LOD is shown below. If there are spaces within a bridge then they are modelled in LOD4.

Representing every part of a bridge as a closed volume is not possible. In situations such as these, a 'multisurface' can be used (*lod1MultiSurface* to *lod4MultiSurface*). In addition to the solids, multicurves can be used for parts of the bridge such as its cables.

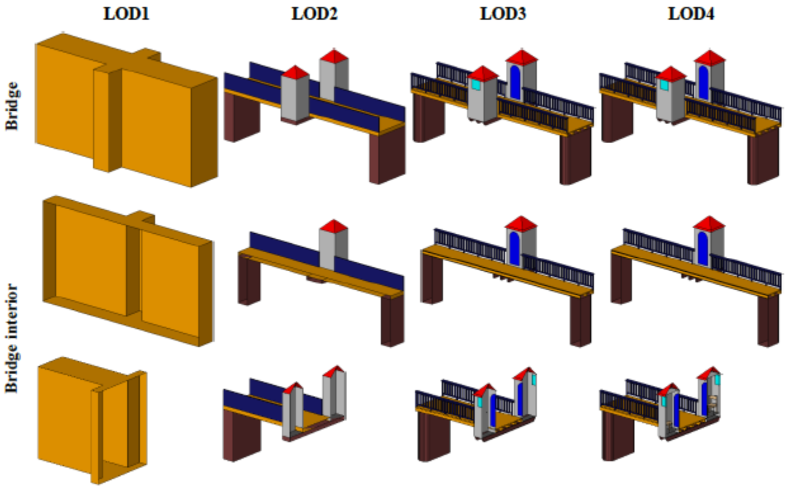

Figure 20: The difference in LOD's for one type of bridge.

3.5 Plant Cover in LOD1 and LOD2

Chapter 3.2 describes how a LOD0 terrain definition can be extracted from polygons at level 0. Volume portrayal of Road Parts, Water Bodies and Land Use terrain parts is of little use. For Plant Cover, however, a volume representation is of interest. An example of a LOD1 Plant Cover representation in a built up area is illustrated in Figure 14. In this LOD1 representation, an average height is determined for each 2D Plant Cover surface. Consequently, as described earlier with relation to LOD1 buildings, each polygon is extruded into a volume.

The LOD2 representation of a Vegetated Terrain Area is not constrained to one height per object, but allows variation in height. This can be achieved through segmentation (where one height is determined per segment) as well as by triangulation where, for example based on laser scan data, the height progression within the Plant Cover Area is defined.

Both LOD1 and LOD2 can be represented, once again in analogy to LOD buildings, with both relative and absolute height positioning, although the relative variant occurs more often. The relative variant is useful if the Vegetated Terrain Area (for example together with LOD1 buildings with relative height positioning) is positioned on a flat surface (2D IMGeo or aerial photograph).

When absolute height positioning is used, connecting up to a LOD0 terrain definition is more complex. A hybrid intermediate form is possible where the Plant Cover is defined in LOD0, but does not use the ground level heights (for example from filtered AHN data) but the crown heights (from unfiltered AHN data). This hybrid form can have additional value from a visualisation perspective, but the possible added value from volume calculations (for example environmental applications: m^3 vegetation) is not supported.

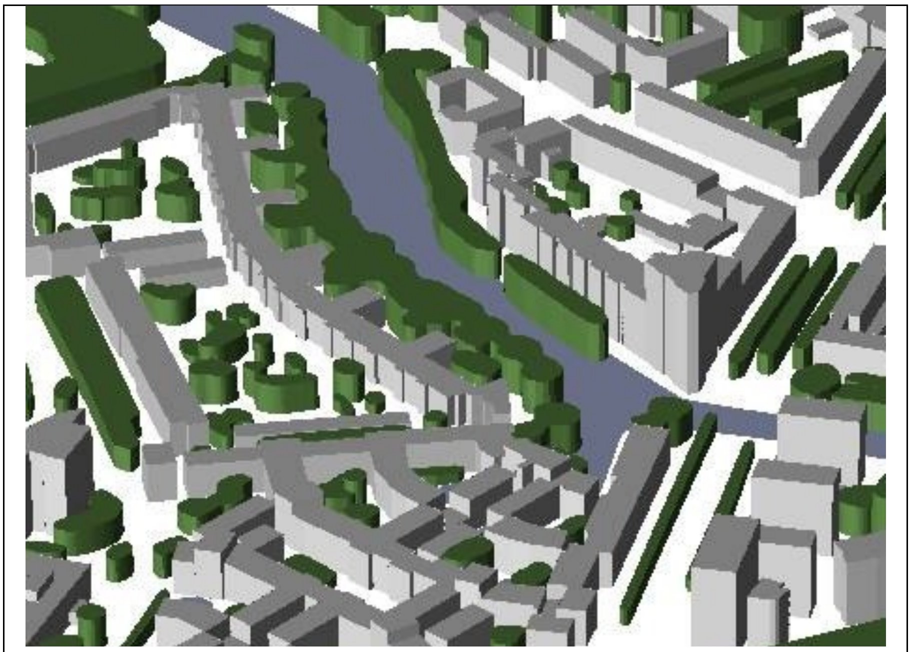

Figure 21: LOD1 of a Vegetated Terrain Area, together with a LOD1 building (source: iDelft)

3.6 Trees and Other City Furniture in LOD2 and LOD3

IMGeo recognises the following City Furniture:

- Container
- Sign
- Installation
- Casing
- Mast
- Pole
- Sensor
- Street Furniture

Picknicktafel
Reclamezuil
Slagboom
Speelvoorziening
Telefooncel

A tree is a separate object type in IMGeo.

These pieces of City Furniture can be described in 3D with a restricted number of base variants. This is essentially also the case for 2D. The visualisation which has been made for 2D IMGeo consists of pictograms.

These objects can also be constructed in 3D object libraries or purchased. For example, IMGeo classifies an “advertising column”. An organisation’s maintenance database may give insight into what sort of advertising column it is, and thus allow different sorts of 3D advertising columns to be represented. The availability of a library of 3D City Furniture symbols would be helpful in this situation.

The difference between LOD2 and LOD3 is that in LOD3 complex parametrical models form the basis for a 3D visualisation while, in comparison, in LOD2 simpler symbol libraries (or models) form the basis.

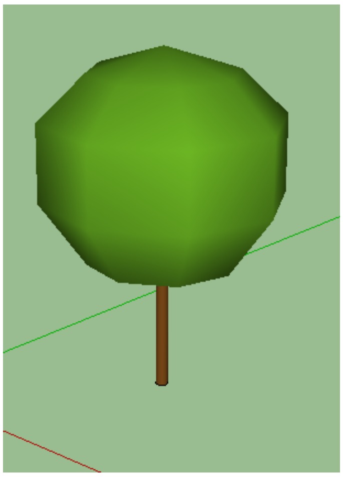	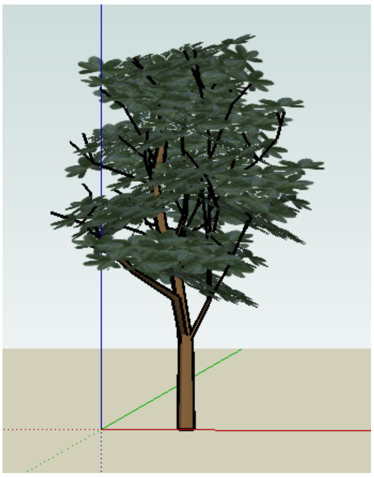
LOD2	LOD3

Figure 22: An example of a tree in LOD2 and LOD3 (source: Alterra)

The use of maintenance information is necessary for trees (in IMGeo only the co ordinates are recorded as attribute). Figure 23 was computed by Alterra and represents trees based on the width of their crown and their height.

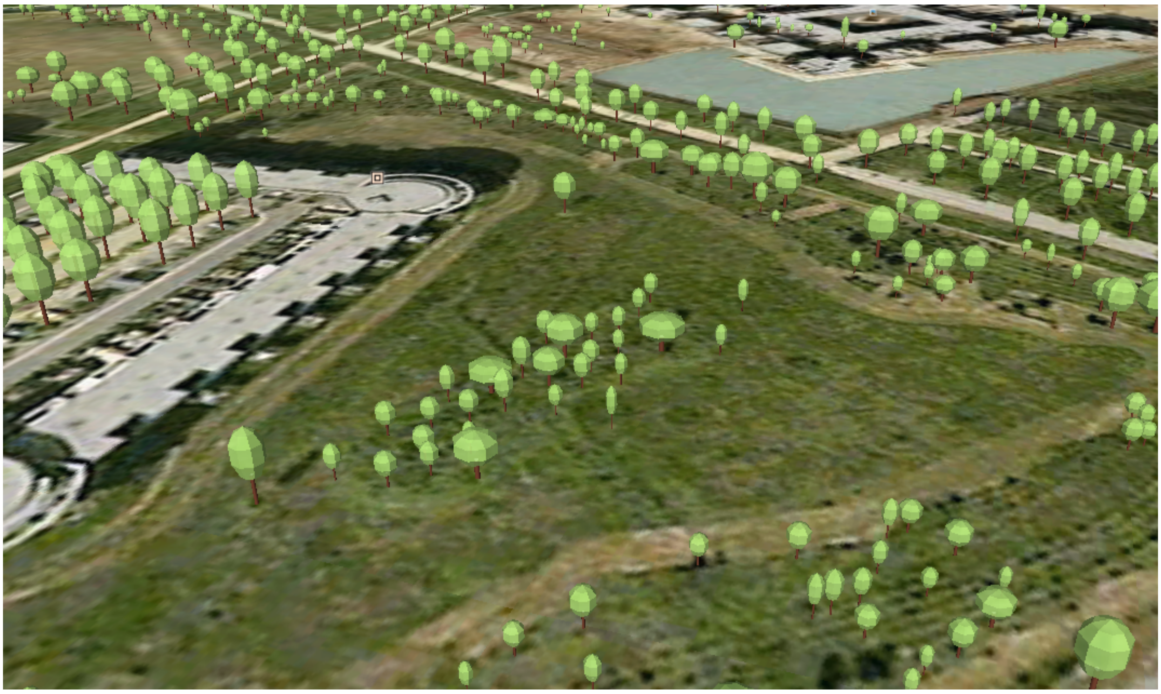

Figure 23: Trees represented according to the width of their crown and their height.

If so desired, this example can be detailed by adding a growth model of each tree species represented. This requires, however, the clear definition in advance of 3D models for each tree species which, as far as we know, are not standardly available. In this way each tree would be portrayed by a unique 3D model. These models would be constructed automatically using a restricted number of basic properties and corresponding parameters, which would enable the demands on IT resources to remain limited, while concurrently delivering a more realistic portrayal of each tree.

Recommendation 4. Dependent on the application, it can be prudent to generate a 3D model for certain items of city furniture (define each one explicitly) and trees. This can be relatively a simple procedure if a good 3D library has been already been created. The authors of this document know of no such library which means that it will need to be discussed within your organisation if this can be included in the questions when the project is put out for tender.

3.7 Texture

Texture information can be added to 3D information. Texture information consists of colours or images. The most important goal of adding texture information is, naturally, the more striking visualisation of the data. That should not, however, be defined as the only goal; texture information in CityGML can also be used for analysis.

Adding texture to 3D geo-information is always to be recommended, specifically if the 3D model should provide a "real-world" like visualization. In addition to this, if more types of texture are used then a wide range of applications can be supported, i.e. the more types of texture the wider the range of applications.

3.7.1 Texture Type

Texture can be assigned in different ways:

1. From image information (mostly photos)
2. From the average point colour per surface
3. From the IMGeo visualisation

re 1. The process of constructing texture information from image information consists roughly of the use of image information from classic aerial photos, oblique aerial photos or panorama photos videos. A process interesting from a technical cost perspective is only reached if all of the orientation data (state and position during recording) are known, and if enough there is enough geometrical quality.

re 2. Sometimes image information can be used in a highly simplified form (one surface receives a single colour which is the average of the colour of the pixels projection on the corresponding surface). This method of texturing is a logical second option if image information is also being worked with.

re 3. 3D geo information's semantic properties can also be used (a surface receives the colour red, because it is a roof surface)

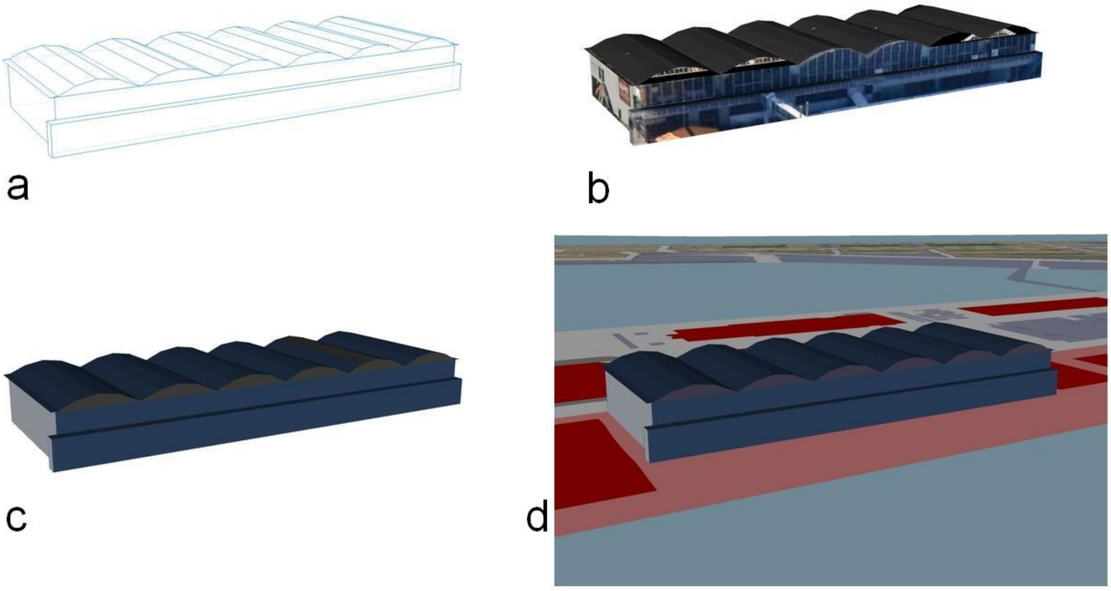

Figure 24: Texturing: a. Without texture; be Texture from (aerial) photoimages; c. Texture on a basis of the average point colour per surface; d. Texture on a basis of average colour and IMGeo's visualisation (including background)

As a rule, texture information derived from image information is saved in separate files in a compromised graphical format (such as jpg) and only attribute data (link to the jpg, projection data for unambiguous mapping on the object) is added. The use of image information means a big claim on hard and software and makes a simple graphical portrayal of 3D geo-information difficult. This means that navigating in a complex 3D model is more difficult to realise if not impossible. In can be readily concluded that the photo's properties (detail or resolution and colour model) have a great impact on the texture information's specification. In addition to this, the quality of the photo's positioning data (angle and orientation) can have an influence on the quality of the textures.

Texture information in the form of computer graphics is many times more efficient to save and to use than the texture information from image information. This method of working derives from the computer game industry and is grafted onto the efficient graphic presentation: navigation in a complex 3D model is much easier then. Texture information in the form of computer graphics does, however, result in a less realistic portrayal in the 3D model. In spite of this, texture information in the form of computer graphics, and specifically on a basis of IMGeo's visualisation, results in a clearer 'map image' (a similar comparison could be made between a TOP10NL representation and an aerial photo) and thus makes interpretation easier.

Recommendation 5. Use textures based on IMGeo's visualisation (from the Visualisation Toolkit NL:*handreiking Visualisatie*), because that makes interpretation easier and visualising more user friendly)

Recommendation 6. In order to get the most realistic image of the city, textures from image information (in combination with an aerial photo positioned on the terrain), are the best solution. This information is, however, only useable on a small scale.

Recommendation 7. If only the image information textures are being requested, then it is also a good idea to ask for textures based on the average point colour per surface. By doing this a photo realistic image of the city as a whole comes into being, this is still quick to visualise and therefore user-friendly.

Recommendation 8. The fact that a CityGML IMGeo file consists of classes and classifications, means that IMGeo's standard visualisation can be employed. This does not need to be specified to the supplier because any GIS and or CAD software package/ viewer can visualise on the basis of class/classification.

Recommendation 9. An average point colour for visualisation purposes can be requested from the supplier. This point colour is a texture which is derived from image information.

The following requirements can be used dependent on what choices have been made:

Requirement 32. Texture information should be delivered on a basis of image information. In each case it is always true that only the image information of the relevant object is used.

Clarification: Which source information should be used should not be specified. The supplier could already have this source information and supplies it on that basis. Alternatively, new source information may need to be collected.

3.7.2 File Properties for Textures Derived from Image Information.

Texture data from image data is raster data and consists of photographs projected on the objects. This sort of information is not actually recorded in CityGML, but is referred to. This reference can be to a web service or to a separate (image-) files.

Requirement 33. The 3D model must contain all the data necessary in order to give a complete and unambiguous relation between texture information and geometry.

There are a number of additional requirements which are necessary with reference to file properties (completeness, cover, format, layout, description). The following requirements can be used dependent on what is desired:

Requirement 34. A texture can be ascribed to all object surfaces, with the exception of the object's footprint (which is level with the object's intersection with ground level) and (parts of) excluded objects for which there is no source data available. Texture data should also be constructed and delivered if there isn't enough source data to fully cover an object's surface.

Clarification: In this situation the missing texture data can be supplied in a colour of choice, or not be supplied at all. In the latter case, these parts can be recorded as transparent in CityGML (with the *wrapMode = border/none*).

When the object's surface cannot be completely covered by texture data alone, then texture data can be delivered in a colour of choice, or the choice can be made to leave the surfaces transparent. This can be done in the same manner.

Requirement 35. If an object's surface cannot be completely covered by texture information alone, then the choice can be made for either some or no texture information to be supplied (in the colour <kleur>).

Requirement 36. All texture information should be delivered in the same format <formaat>.

Clarification: Format specifications such as compression and colour model should be more closely defined. The file format could be, for example, either jpg or png. Due to the fact that 3D models can be difficult to visualise with photo textures, it can be a good idea to choose a format that uses relatively little memory and which can take a great deal of compression.

Requirement 37. The files containing texture information each receive a unique file name according to its structure <*structuur*>.

Clarification: For example, the structure can be defined as {########}.{jpg/png} (# signs should be replaced by numbers and/or letters. A lot of software cannot work with file names longer than 8 characters. Due to this, it is worth recommending that the file names should be made up of (maximum) 8 characters.

3.7.3 Texture in CityGML

Textures should be modelled using the extension module *Appearance* in CityGML. Webservices or (image) files can be referenced from CityGML in relation to textures from image information. Texture information based on IMGeo's visualisation, or a single average colour per surface can be added directly into CityGML.

The *Appearance* module makes it possible to add more textures. This can be done per object as well as for all objects in one go. It can also be repeated for each LOD. Furthermore, it can be done in a number of themes, which for example can refer to IMGeo, or just to photo material or to other unusual qualities which are not only visual (for example noise pollution or infrared radiation).

An Appearance consists of data for each geometrical object, for example each surface in the 3D model has an Appearance. Several colours or textures can be ascribed to a single surface. Alternatively, several surfaces can be referenced to the same colour or texture. This latter method is interesting when colours from IMGeo's visualisation are being used, as they do not then need to be re specified for each object.

CityGML 2.0 uses the term *material* when referring to properties which do not change such as colour, and the term *texture* if it concerns properties which are dependent on their location within the surface such as is the case with textures from image information. Each polygon or object in CityGML can have a *material* as well as a *texture* for both each theme and each side (for example building parts can have both an in and an outside)

CityGML 2.0 provides, therefore, maximum flexibility and texture information can be used for both visualisation and for analysis.

The *Appearance* extension module is a new addition to CityGML 2.0. The class *TexturedSurface* was used in CityGML 1.0. CityGML still provides a replacement module *TexturedSurface,* but its use is not encouraged. The *Appearance* module offers the same functionality and more.

Requirement 38. All texture information should be modelled in CityGML 2.0 with the *Appearance* extension module.

It can be specified that only texture information for the outside of objects is modelled so that the visualisation (rendering) of (BuildingPart) objects does not become too difficult.

Requirement 39. The texture information of (BuildingPart) objects is only modelled for the outer part of the surface. No texture information is defined for the inner part of each surface.

3.7.4 Quality.

The texture information’s geometrical quality is primarily influenced by the source information’s geometrical quality.

When converting from image to texture information, a process can be put in place that utilizes all of the source information’s quality (precision) in the orientation data (angles of cameras and position in the co ordinates of images)

Requirement 40. During the surface texturing process, ie the projection of the source information (image information) onto the surfaces of the 3D model, all existing and supplied orientation data should be used without the numbers being rounded.

A decision can be made to enrich the source information’s orientation data, in order to improve the projection’s quality. This can be done in two ways:

- By doing an additional geometrical correction to the source data, for example when aerial photos are used for an extra and more stringent triangulation and block adjustment.
- By manually instructing (mapping) the surface’s vertices in the source information.

For example, a more stringent triangulation and block adjustment can be performed before purchasing source information such as aerial photos. In this case, a geometrical correction when constructing the 3D model is no longer necessary or wise. Altering the model manually is generally expensive, unless the model has very few objects.

In spite of all this the underlying 2D information can also include geometrical faults, which can have an influence on the geometrical quality of the texturing process. These faults must be accepted because it is required that the walls of a BuildingPart’s location align with the building’s 2D BGT or BAG definition (to assure consistency between 2D and 3D representations).

Recommendation 10. If the source information’s orientation data can be relatively simply improved by an additional geometrical correction (for example by more stringently triangulating and block adjusting aerial photos) then it is recommended that this correction is done.

In general, when (first) constructing a 3D model only one set of source information will be used for texturing. If several source data sets are used, then decisions can be made about how to deal with the geometrical quality for each data set.

In addition to the geometrical quality of image material, radiometrical quality (colour correctness, through drawing) also needs to be taken into account. The source information’s radiometrical quality cannot be improved by converting it to texture information but, at most, be adjusted (made lighter, be supplied with more by drawing).

The source information’s resolution determines the texture information’s resolution.

Finally, the quality (or suitability) of the texturing process is influenced by a number of other factors. There are, of course, all sorts of objects which are captured in images but which are not modelled in the 3D (BuildingPart) model. These include trees, cars, people and chimneys, satellite dishes, dormer windows, bay windows and glasshouses.

Chapter 4

Applications

This chapter gives the necessary requirements for a number of applications and is randomly selected from the many use cases that came up in the 3D Pilot. It is only an indication. The person who uses these requirements in a specifications document will need to determine, through discussion with users, exactly which requirements are necessary to supplement the use cases. The list below can get this process started.

Sample use cases

i. Calculating Flood Areas
ii. Identifying Terrain Structures
iii. Determining Mowing Specifications (With Price Dependent on Gradient)
iv. Drainage Calculations (Municipal Water)
v. Town Planning Visualisations
vi. Spatial Plans in their First Phase
vii. Mid Scale Spatial Analyses (Noise, Solar and Flood Studies)
viii. Photovoltaic Potential
ix. Determining Planning Permission
x. Calculation (x,x,x)
xi. Determining Gutter and Ridge Heights for the Benefit of Zoning Plans
xii. Noise and Environmental Analyses.
xiii. Solar Potential Studies
xiv. Design Processes for City and Region Development
xv. Testing Municipal Designs
xvi. Supporting (Construction) Projects
xvii. Visualising Problems and Solutions in the City
xviii. Developing and Maintaining Outside Objects
xix. Public Order and Safety Simulations
xx. City Promotion (Graphic, Using a Scale Model and Interactive).
xxi. Determining Building Volumes for the OZB
xxii. 3D Cadastre
xxiii. Real Estate Agency (Preselecting Houses for Clients)
xxiv. Telecommunication (locating GSM broadcasting stations)
xxv. Urban Heat Analysis (in conjunction with (vii))
xxvi. Wind Around High Buildings
xxvii. Civil Engineering Planning

Group Requirements	Necessary Requirements	Use cases
A	Requirement 1, Requirement 2, Requirement 3, Requirement 4, Requirement 5, Requirement 6, Requirement 7, Requirement 8, Requirement 9, Requirement 10, Requirement 11, Requirement 12, Requirement 13, Requirement 14, Requirement 15, Requirement 16, Requirement 17, Requirement 18, Requirement 19, Requirement 20, Requirement 21, Requirement 31	A minimal requirement for all use cases.
B	Requirement 22, Requirement 23, Requirement 24, Requirement 25, Requirement 26, Requirement 27, Requirement 28,	v, ix, x, xi, xii, xiii, xiv, xv, xvi, xvii,

	Requirement 29, Requirement 30	xviii, xx, xxi, xxii, xxiii, xxv
C	Requirement 8	iii, iv, x, xxvii
D	Requirement 33, Requirement 34, Requirement 34, Requirement 35, Requirement 36, Requirement 37, Requirement 38, Requirement 39, Requirement 40	v, xvii, xx

It is advised that at least the requirements linked to Group A and B above are stipulated when putting the work of 3DIMGeo data reconstruction out to tender.

Chapter 5

Data

This chapter describes the diverse aspects of the source data which can be used for the reconstruction of 3D information objects. Sometimes suggestions are made concerning further acquisition, but the emphasis is on source material already available at organisations.

5.1 Background to the acquisition and availability of source data

5.1.1 Data, LOD and Scale of the Modelling Process

This report is particularly focused on the construction of LOD0, LOD1 and LOD2 models. In general, a whole governed area or an entire municipality is modelled. In this case there are diverse geo-registrations available which can be used as input data, such as small or large scale topography and the AHN.

This chapter will not give further details about data used for the construction of LOD3 and LOD4 models. In those situations there is always highly detailed information present which can sometimes time-intensively be converted from files, but which also often requires manual construction (for example by surveying or terrestrial laser altimetry). Depending on what can be done with existing registrations as input data, the modelling process and the resultant application, the construction of LOD3 and LOD4 information will more often be small scale and limited to specific projects. However, LOD3 and LOD4 representations can be added to data sets that contain LOD1 or LOD2 representatins: several representations of one BuildingPart can concurrently exist. Even if the ultimate goal is a 3D model in LOD4, construction in smaller LOD's remains an option.

In conclusion: the starting point of this chapter is the construction of 3D information in LOD1 and LOD2 and for the whole governed area.

5.1.2 What is the final goal?

Despite the fact that the application of the data influences the requirements of the data and therefore of the process of 3D data reconstruction, the available source data often determines the content of the 3D model. If data can be used that the organisation already possesses or if data which can be acquired at relatively low cost, then the cost of constructing the 3D model will also be low.

Generally:

- Properties in LOD1 can be modelled if one or several topographical datasets are available which are structured in an object oriented way (such as TOP10NL, GBK or BAG);
- Properties in LOD2 can be modelled if height information of a sufficient resolution is available and includes roof shapes and (bigger) dormer windows (such as taken from AHN2 or own surveying);
- Properties can be given texture if image information including accurate orientation data is available (aerial photos with sufficient length and breadth overlap, oblique photos or panorama photos which are precisely orientated);
- A 2.5D terrain can be modelled if sufficient height information is available (from AHN2, own surveying or image matching in stereo-aerial photos);
- Green objects (for example trees) can be modelled if point topography is available (from outtdoor maintenance systems or from analyses based on, for example, the AHN2).

The 3D Pilot Final Report "werkgroep 3D aanbod van geo-informatie (3D Working Group Supply of Geo Information)" (2011) describes commonly available 3D source data in relation to the construction of 3D geo information. A description of the platforms which are used when surveying source data and the developments in that domain (from static to mobile) are also explained in that report.

5.1.3 A Combination of Sensors

Sensors are being used when surveying more and more often. Although the combination is naturally very interesting from a technical cost perspective, in general there are also extra limitations during the surveying process. The covering/comprehensive surveying/acquisition of oblique aerial photos lays down conditions, for example, the flight plan that can be in opposition to the most suitable configuration for surveying by (stereo) aerial photos. A flight plan can only be optimised once for one of the two resulting datasets.

In city areas in particular, occlusion (the shadow working in falling (light) in photogrammetry and laser altimetry) can play an important role. When arranging a flight plan occlusion should be taken into account, so that few, or no, locations at ground level remain unmeasured or unexecuted and/or so that a sufficient number of property walls are covered by image information.

5.1.4 Benefits and Drawbacks of Source Data Taken Directly from Surveying.

Source data already available in an organisation has an impact on the construction of 3D geo information, because:

- source data taken from surveys offers the customer the possibility to control the surveying so that it delivers the best data for his process (a benefit);
- source data taken from surveys means that the customer has to invest more money than if this source data were either bought in or purchased from pre existing sources (a drawback).

5.1.5 Overview of the Most Important Source Data

These are the most important, most suitable or most often used source data taken directly from (own) surveying:

1. Laser Altimetric Data
2. Laser Scan Data from Dynamic Surveying
3. Oblique Aerial Photos
4. 360° panorama photos or videos
5. Point Clouds from Stereo Aerial Photos (imagematching)

These are the most important, most suitable or most often used source data taken directly from existing processes:

1. GBKN of BGT
2. BAG
3. TOP10NL
4. AHN2
5. DTB (Rijkswaterstaat)
6. Stereo Aerial Photos and derived Orthophotos

As previously mentioned, all of these have been discussed in the 3D Pilot Final Report "werkgroep 3D aanbod van geo-informatie (3D Working Group Supply of Geo Information)" (2011)

5.1.6 Source Data (Point Clouds) from Image Matching.

In addition to laser altimetry (for example in the AHN or AHN2) detailed point clouds can be determined from Stereo Aerial Photos. The result can be a point cloud that defines the bird eye view of a terrain (including objects) just as Laser Altimetry does.

In order to successfully construct detailed point clouds from image matching, stereo aerial photos are necessary:

- which have been surveyed with a relatively big overlap in length (for example 80%), so that points numerically superior can be ascertained (a big overlap in width of, for example, 60% improves the result but is not essential);
- which are sharp, clear and display a broad range of colours
- which are high resolution
- which haven't been (excessively) compressed;
- for which accurate orientation data has been fixed in a high quality triangulation and block adjustment.

The flight plan for stereo aerial photos which are suitable for image matching is often also suitable for producing aerial photos. These photo's are appropriate for putting texture (particularly wall surfaces) into 3D models.

A disadvantage of image matching point clouds is that occlusion can prevent green areas being surveyed. In some cases, with several returns, laser altimetry can survey not only green areas but also dense green (such as shrubs) points at ground level. This will be less of a problem when constructing 3D Geo Information for BuildingParts, except in cases where a tree's foliage is situated above a roof.

The point cloud's quality (its precision in XYZ) is dependent on the quality of the (stereo-) aerial photos orientation and its matching algorithm and these are in general of lower quality than the data sets acquired by laser altimetry, such as the AHN2.On the other hand, a point cloud can often be extracted at little additional cost because usually aerial photos have already been surveyed for other processes.

5.2 Remarks on the Use of Source Data for the Construction of 3D Geo Information.

5.2.1 The Construction of 3D BuildingPart Models (LOD1)

Constructing BuildingParts in LOD1 is simpler (if heigt data is already available) than constructing them in LOD2, and can be either acquired at low cost or done by a GIS analysis in house.

This GIS analysis can vary in complexity if, for example, vertical intervals in roofs are taken into account. Examples of how to approach such an analysis include:

a. Identification of source 2D topography;
b. Clipping the (2D) BuildingPart geometries so that vertical intervals can be handled. Clarification: The required source data can be, for example, BAG or TOP10NL. The most intelligent method is to segment the roof surfaces based on height information which is taken, for example, from laser altimetry. This brings the construction of BuildingParts in LOD2 within reach too.
c. Determining the heights of roof surfaces.
   Clarification: This could be the source data average from either laser altimetry or from photogrammatic measurements. If using laser altimetry, it is wise to take the mismatch between BuildingPart geometry and laser altimetric source data into account by, for example, first buffering the roof surface geometries (for instance with a 1m). Heights can be determined that fall inside these buffers and solution has to be found for exceptions, such as small roof surfaces whose height information, due to the buffer, cannot be recorded. Instead of the average height the maximum height can be used. A maximum height is useful in, for instance, noise research. When determining a maximum height, it is wise to negate outliers in the source data by, for example, discarding the top 10% heights and using the next maximum heights.

5.2.1 The Construction of 3D BuildingPart Models (LOD2)

LOD2 can sometimes be constructed primarily with laser altimetric data, and sometimes with height information from stereo photogrammetry. If both source data sets are available then it is wise not to specify in which manner the LOD2 model should be built but simply what criteria the 3D model must satisfy at the end. In this way, a hybrid approach (such as stereo photogrammetry for gutter heights and after that AHN2 for roof shapes) can still be used.

Segmentation (the determining of groups of points which are situated in the same surface part) is the most successful method to model roof shapes, even those which are more complex, but is more demanding for the software to be used.

The topographical registrations BAG and BGT (GBK) which can be used for the construction of 3D models of BuildingParts in LOD2 have been discussed in Chapter 4. Intelligently using both data sets will result in the most accurate model, but is also hampered by the (definition) differences between BAG and BGT (GBK).

Not all BuildingParts or objects are registered in the BGT (GBK) and/or the BAG, examples are industrial installations such as silos. Therefore one must know that both registrations are not complete with respect to the building objects.

5.2.3 Source Data for Assigning Texture.

Image information without precise orientation data (aerial photos, oblique photos or panorama photos) is not suitable for texturing. The best approach is to take the application of texture into account when purchasing image information. If that is not an option, then alterations to the data such as those described in Chapter 4, can still be used to acquire better orientation data.

In spite of this, automatic texture information made in this way can be uneven in quality

The advantages and disadvantages of using differing image information for applying texture have been discussed in the 3D Pilot Final Report "werkgroep 3D aanbod van geo-informatie (3D Working Group Supply of Geo Information)" (2011)

When using texture information, it is important to take the following unusual effects into account

1. Smearing: Pixels are 'smeared' over the surface when a photo image is used to add texture to surfaces which nearly lie in the extension of the field of view. This looks like long 'smears' or smudges in the textured image. Smearing is particularly common if aerial photos from a less than perfect flight are used to texture the wall surfaces.
2. Z-fighting: If there are two surfaces for a single location and there is texture information for both of them, 'flickering' can occur. This effect is particularly visible if the visualised surfaces both have different textures (or colours). This problem can be solved by adding a parameter (0 or 1) which checks the visualisation of both surfaces and ensures that the surfaces are not visualised concurrently. Software should be able to do this.
3. Visualising CityGML is demanding for hard- and software. Using methods which streamline the visualisation process by not adding texture to both inward and outward facing surfaces or by heavy compression or by using one texture wrap per object (putting an object or BuildingPart's texture information together in one file) is recommended.
4. CityGML is not a suitable format for visualisation, but primarily an exchange format and information model. Other formats, which work from triangulated models (i.e. all of the geometry is in the form of a triangle), are more suitable. Therefore, a conversion is necessary and it is recommended to mention this in the tendering document.
5. If CityGML files have to be converted (and that is recommended), texture files must not have long names. Some graphic formats expect file names no longer than 8 characters (excluding extension). This can be requested in the tendering document.

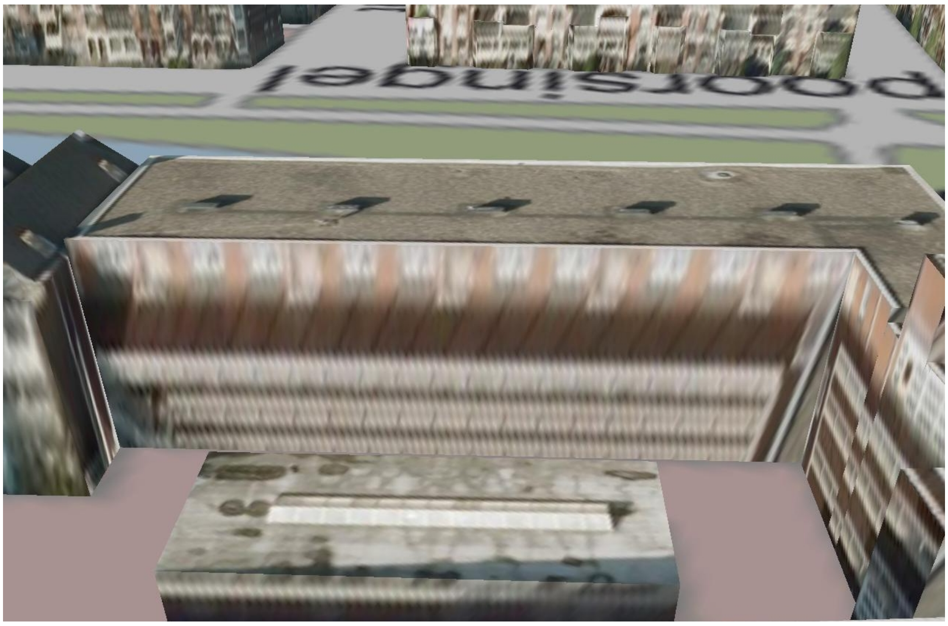

Figure 25: An example of wall surfaces which are smeared after being textured from aerial photos.

5.3 Data which is Supplied by the Customer

5.3.1 Source Data

The person commissioning the 3D data model could supply the following source data if it is available:

1. The governed area's border (such as the municipality boundary) and boundaries of subsectors (such as neighbourhood boundaries)
2. Object oriented, topographic 2D information about the objects to be modeled in 3D (such as main buildings and their annexes), for example, the BGT and/or the BAG.
3. Laser Altimetric Data from, for example, the AHN2 (such as the DSM, DEM and/or DTM)
4. Stereo Photography and/or Orthophotography.
5. 360° photography taken at street level.

The source data can be supplied with descriptions of a variety of characteristics, such as listed below, so that the tendering firm can prepare an appropriate offer:

- number and type of objects
- dimension
- data type
- number of files
- file formats
- actuality
- absolute tolerance/ geometrical quality
- absolute and relative precision in XY and in Z
- reference (co ordinate) system
- camera used
- length and width overlap
- resolution
- orientation format
- precision of horizontal and vertical orientation

5.3.2 The Quality of the Delivered Source Data.

The quality of the source data varies. It is recommended stating in the tender document that imperfections should only be corrected if they are the result of work carried out by the supplier. The supplier does not have to correct errors which are a result of errors in the source material.

5.3.3 Supply in Subareas

The visualisation and maintenance of large areas in 3D can be problematic (given in the hard and software available within the organisation). In addition the construction of large areas by the supplier cannot be achieved by a single operation. It can be useful to come to agreements about dividing up of the area to be modelled. Could this area, for example, be divided according to natural boundaries, neighbourhood or district borders or according to other borders?

5.4 Data Supplied by the Supplier

In addition to the information constructed in CityGML, the supplier can be asked for metadata, a report which includes a description of the data quality, a report about discrepancies in the supplied source data and/or for optional products.

5.4.1 Metadata

The supplier should be asked to supply metadata, for example in XML, which conforms to the Dutch geographical metadata standard. The metadata can then consist of a compulsory core set of metadata elements and an optional set of metadata elements.

5.4.2 Report

A report can include a description of work completed, quality controls and their results and the quality of the supplied 3 D data. It can detail whether the delivery requirements (such as geometric precision, completeness, readability and file characteristics) have been met.

5.4.3 Source Data: Is it Up to Date and What Discrepancies Does it Have?

The source data is never error-free and there are always differences in definitions and actuality of different source data. Due to this, for example:

- BGT (GBK) or BAG surfaces can be bigger or smaller than the object as it is portrayed in the aerial photo and/or laser altimetric data;
- more or fewer surfaces are stored in the BGT (GBK) or BAG than are visible in the aerial photo and/or laser altimetric data.

A report can be requested which lists and categorizes these 'discrepancies'. The supplier can prepare and carry out follow up procedures. This report does not have to detail everything but can be confined to the discrepancies, which the supplier came across during the 3D information's construction.

5.4.4 Optional Products

Supplementary data can be made during the 3D information construction process without additional activities or processes, such as renders, studies, analyses or simulations.

An example of optional information, which the customer can ask for, is the addition of diverse attributes to the 3D model which may ease the use of the 3D data, such as orientation and slope attributes for each surface.

A list of supplementary requisite and optional products can be asked for when tendering which includes a description of the properties and quality of the data.

5.5 File Characteristics, Formats, Layout and Names

A tendering document usually details what file properties are required, for example formats, layout and names. File properties were discussed in Chapter 4. File format can include, for example, the description of the exact(ly) modelled surfaces and require that only outward facing surfaces are modelled (thus no breaklines or break surfaces). File formats can include essential attributes denominated by the CityGML schema and any optional attributes. File layout can describe for which (parts of) maintained areas which file is supplied and in which (if any) file formats supplementary to CityGML. File names can specify, for example, how file names are constructed in CityGML and how file names for texture information are determined from photo images.

Chapter 6

Checklist

This report gives requirements and recommendations. If your organisation contracts out the construction of 3D geo information, then it is essential that checks are done. This Chapter details checks in relation to different requirements. These checks may be:

- Completed in house; or
- Contracted out to an agency specialised in this area who may also do a number of additional checks for you; or
- Completed at your request by the supplier and the results handed over to you.

The supplier can be requested to supply extra products (such as a colour coded point cloud with viewer), which can be easily visually checked.

The table below suggests checks for each requirement. It does not, however, include criteria, which can be used to reject a delivery. Such criteria are dependent on the desired application.

These checks are based on initial findings. Further checks will be added or existing checks modified as we will gain further experience. By using the SIG3D (Special Interest Group 3D) to share additional checks or improvements on those listed below, new and better (automatic) checks will become available.

Requirement	Brief Description	Check
3.1 IMGeo 2.0 CityGML: Generic Requirements		
Requirement 1	The 3D data should be structured according to IMGeo-CityGML format. The standards can be found at: http://www.geonovum.nl/dossiers/bgtimgeo/destandaard	Check this by using the developed validation tool
Requirement 2	The IMGeo-CityGML data must comply with CityGML 2.0. In some cases we have more stringent requirements than CityGML	Check this by using the developed validation tool
Requirement 3	Employ the EPSG 7415 Spatial Reference System (co ordinate system).	Check if the EPSG code 7415 is to be found in the CityGML file.
3.2 Specifications for LOD0 Representation.		
Requirement 4	Every object in IMGeo is represented by a LOD0 geometry i.e. a TIN surface (triangulatedSurface) per object (tessellation of the object’s footprint) . The LOD0 terrain is formed by a collection of such adjacent TIN surfaces, with recognizable object boundaries (constrained TIN)	Check if the number of polygons in LOD0 is the same as that in 2D IMGeo CityGML
Requirement 5	The LOD0 geometries of all IMGeo polygons (water, road, building, land use, vegetation) at ground level should form a planar partition in 2.5D (no holes or overlap).	Check by looking for holes or overlap
Requirement 6	Requirement 6. The height difference between the terrain in reality and its representation in TINs is allowed to be maximum X cm. X can be dependent on the object	It can be requested that a colour coded point file be supplied in which the terrain

	type (for example another X can be chosen for hard surfaces with curbs than that for pasture). Individual apexes are acceptable until up to 3 times X, but connected pieces of a TIN of more than Y m2 may deviate no more than this X cm.	points are coloured as a function of the height deviation with respect to the object surface that the terrain models. It can then easily be seen if areas (greater than Y m2) show greater deviation.
Requirement 7	Vertical surfaces in the TIN may not occur, because many GIS software crashes on such data. Instead, vertical surfaces should be approached by maximum sloping surfaces. How this should be done depends on which objects are left and right of the vertical jump. The sloping surfaces need to be attached as follows to the relevant object:	Testing the Z component of the TIN triangles' normal vectors. These Z components may not be equal to 0. An alternative, but incomplete check, is to look for points with the same XY co ordinates, but the same Z co ordinates.
Requirement 8	When very precise vertical intervals between specific objects are necessary, this should be recorded in the technical specifications. A minimum height should be defined and vertical intervals must be visible. Examples are the height jumps at the location of curbs.	Check randomly if small vertical intervals have been modelled.
Requirement 9	Waterbodies are always flat, horizontal surfaces.	Testing the X and Y component of surfaces' normal vectors. These must be equal to 0.
Requirement 10	IMGeo polygons which are above or below the terrain should be modelled with a triangulatedSurface which connects up to the topologically consistent ground level. The result is the stacking of 2.5 objects.	Overlapping objects with differing levels may not intersect each other in height.
Requirement 11	All IMGeo polygons should be assigned to the IMGeo LOD0 representation, i.e. both those at ground level as well as the ones above and below ground level	Check if a number of polygons in LOD0 are in agreement with the polygons in 2D IMGeo CityGML.
Requirement 12	Terrain Intersection Curves (TIC's) should be used in order to make ClosingSurfaces where 3D objects hang above or in the terrain model. This results in a closed topologically correct terrain model.	
3.3 Building Specifications		
Requirement 13	Requirement 13. The ground surface of a building at LOD1 and LOD2 must be horizontal. The ground surfaces should, though, be determined per individual building and not per block of buildings . This surface is then positioned at the lowest height of the terrain at the location of this surface so that the building sinks "in" the terrain and gaps between ground surface are avoided.	
Requirement 14	Notwithstanding the CityGML specification, LOD0 footprint must be determined where the outside wall	

	touches the terrain.	
Requirement 15	An LOD1 representation should be supplied for every IMGeo building.	Easy to check if the building's IMGeo ID is saved as an attribute to the LOD1 representation.
Requirement 16	The building height is the median of the height of the points which are positioned within the footprint.	Check randomly if the median of the height of the points on one roof lies within a margin of X cm from the height in the model.
Requirement 17	Requirement 17. If a building's roof has significant vertical intervals (for example a church with a tower), then these differing height levels should be distinguished in 3D, particularly if the interval is greater than, for example, 1.5 metres and if the surface area is greater than 4 square metres	
Requirement 18	Requirement 18. The lower surfaces of the building's block geometry must correspond to the 2D and LOD0 geometry in IMGeo.	
Requirement 19	Requirement 19. The lower surface of a LOD1 block should be horizontal, taking the lowest point of the footprint's terrain triangulation as its height (see LOD0 building)	
Requirement 20	For buildings which bridge roads or water, through passage should be guaranteed. This may be artificially applied.	
Requirement 21	The geometry of LOD1 Buildings should be defined in CityGML as GML:Solids (closed volumes, also from below) and not as GML:MultiSurface, which is permitted for LOD1 buildings..	Each building object consists of exactly one solid.
Requirement 22	Each LOD2 IMGeo building is modeled by the GML:Solid geometry type in which the semantics of the boundaries (surfaces) are made explicit (e.g. footprint, roof surface, wall surface). LOD2 buildings can be represented as a collection of a solid with other geometry types such as a multisurface for a roof overhang	Each building object consists of a minimum of one solid.
Requirement 23	The locations of the outer walls of 3D building models should be in alignment with the 2D boundaries from the BGT and the BAG (preferably the BGT).	Randomly test if boundaries from the BGT or BAG have been taken up.
Requirement 24	Roof boundaries of 3D building models are in agreement with 2D boundaries from the BGT or BAG (preferably the BAG).	
Requirement 25	Building models should be complete in the sense that the combination of all of a building's surfaces collectively forms a closed volume, a 3D solid. No surface from another building may be positioned within a building model. Building models may touch each other, but not overlap.	Check by means of the developed validation tool.
Requirement 26	When a roof overhang is explicitly modelled, roof surfaces should be split at the roof overhang's location in order to result in a solid geometry. These roof overhangs should	

	be modeled as a (multi)surface and the rest of the roof should form a part of the solid geometry's boundary.	
Requirement 27	When a roof overhang is explicitly modeled, roof surfaces should be split at the roof overhang's location to obtain a valid solid geometry. These roof overhangs should be modeled as a (multi)surface and the rest of the roof should form a part of the solid geometry's boundary	The supplier can be asked to supply a colour coded point cloud in which the points colour within BAG/BGT polygons a function is of the height difference with the modelled roof. Larger deviations can then be spotted easily. The surface area of each "connected component" of points which deviate too much can be calculated with a little more effort.
Requirement 28	LOD2 roof surfaces with a minimum surface area of X m2 may not deviate more than Y m in height from the corresponding points from the point cloud	Checking can be done with a colour coded file, just as by the previous Requirement, although this time with a colour dependent on the angle difference between normal vectors which have been estimated from points which lie within a certain radius and normal vectors from the modelled surfaces.
Requirement 28	Roof surfaces with a minimum surface area of X m2 may not deviate more than Y degrees in the normal direction from a surface because of the corresponding points from the point cloud. This prevents very flat saddle roofs to be modeled by flat roofs and mansard roofs to be modeled by saddle roofs	Check in the same way as Requirement 31.
Requirement 29	Curved surface areas should be represented by a triangulation in which deviation between the true surface area and the triangulation is not more than Xm.	
Requirement 30	Roof surface corner points in the model (for as much as they haven't been misaligned by the BAG) must lie within a distance of Xm from the closest neighbouring data points	Check if there are data points present within a radius of X m from a vertex (and within a BAG outline). Use a 3D query option or specialised software.
Requirement 31	The solids of buildings in LOD1 and LOD2 should conform to the requirements which are discussed in 4.3.4.	Check with validation tool.
3.7 Texture		
Requirement 32	Texture information should be supplied based on image information. Only image information from the relevant object should be used per object	

Requirement 33	The 3D model must contain all the data necessary in order to give a complete and unambiguous relation between texture information and geometry.	
Requirement 34	A texture is assigned to all object surfaces, with the exception of the object's footprint (equal to the intersection of the object with ground level) and the exception of (parts of) objects for which there is source information available. Texture information is also constructed and supplied in a situation where there is not enough source data available to cover the whole surface of an object.	
Requirement 35	If an object's surface cannot be covered completely with texture information alone, then either some or no texture information should be supplied (in the colour <*kleur*>).	
Requirement 36	All texture information should be supplied in the same format <*formaat*>.	
Requirement 37	Each file containing this texture information should receive a unique file name according to its structure <*structuur*>.	
Requirement 38	All texture information should be modelled in CityGML 2.0 by using the *Appearance* extension model.	
Requirement 39	Texture Information for (BuildingPart) objects should only be modelled for the outward facing side of each surface. No texture information should be defined for the inward facing side of each surface.	
Requirement 40	All existing and/or supplied orientation data should be used without being rounded off during the texturing process, (ie when source information or image information is being projected onto the surfaces of the 3D model).	

Recommendation		Check
3.1 IMGeo 2.0 CityGML: Generic Requirements		
Recommendation 1	Consider having the file supplied not only in IMGeo-CityGML but also other formats.	
3.3 Building Specifications		
Recommendation 2	If 2D building surfaces are to be split up during the modelling process, then this should preferably happen with reference to the surfaces in the available point cloud	
Recommendation 3	If the 2D building surfaces are split up by the modelling process, this should preferably occur with lines between corner points in the 2D building outlines.	
3.6 Trees and Other City Furniture in LOD2 and LOD3		
Recommendation 4	Generating some items of city furniture (each item should be explicitly	

	identified one on one) and trees can be a wise option, dependent on the application. This can be a simple process provided there is a good 3D library to form the basis. These libraries are unknown to the authors of this document and it will therefore require consideration within your organisation as to whether this should be included in the tendering process.	
3.7 Texture		
Recommendation 5	Use texture based on the IMGeo Visualisation (to be found in the *handreiking Visualisatie* (Users Guide Visualisation)) as this makes interpretation simpler and visualisation more user friendly.	
Recommendation 6	The best way to get the most realistic image of the city is to combine textures from image information with aerial photographs positioned on the terrain. This information is, however, only suitable at small scales.	
Recommendation 7	If texture derived from image information is already being requested, then it also a good idea to request textures derived from the average point colour per surface. This results in a complete, photo realistic image of the city, which is both quick to visualise and, consequently, user friendly.	
Recommendation 8	Due to the fact that the CityGML IMGeo file is made up of classes and classifications, IMGeo's standard visualisation can be used (see http://www.geonovum.nl/wegwijzer/standaarden/visualisatie-bgtimgeo-handreiking-versie-12). This doesn't need to be discussed with the supplier because each GIS and/or CAD software package or viewer can visualise based on class/classification.	
Recommendation 9	The supplier can be asked for an average point colour for use in visualisation. This colour stems from the image information texture which is already being supplied.	
Recommendation 10	If the source data's orientation data can be improved by additional geometric correction, for example by making a more stringent triangulation and block adjustment from aerial photos, then it is recommended that this correction also be performed.	

Chapter 7

Costs

The market for the generation of 3D models is developing fast. In light of this it is almost impossible to come up with reliable, quantative information about what the costs in relation to such a model might be. Even so, this chapter is highly relevant to any organisation which is considering starting the development of a 3D model, because the experience has shown the 'early adaptors' that the costs are lower than many people expect.

7.1 Costs: Relevant Factors.

The costs for the creation of a 3D model are dependent on diverse factors:

- How far the production process can be automated. This is dependent on:
    - Level of detail (for example it is easy to construct buildings in LOD1 and LOD2 automatically, but LOD3 often requires manual work)
    - Precision (If large geometric deviations can be tolerated, then manual re editing is less necessary.)
    - Object types to be modelled (for example tunnels and bridges are often more complex than buildings)
    - The Quality and Quantity of Source Data (for example stereo aerial photos and laser scan data can efficiently supplement each other)
    - Texture (good quality texture is difficult to apply automatically)
- The Production Method: methods which work, for example, with a library of roof shapes are cheaper (although poorer quality) than methods which try to detect roof shapes in, for example, point clouds.
- The supplier: because methods are often still in development and the market still has not crystallised, there are price differences between suppliers which are not only based on the quality of the model.

7.2 Rough Cost Indications

By now a number of models have been manufactured in the Netherlands. A number of municipalities have had buildings in LOD2 modelled. Although there is a lot of variation in the costs, most people estimate the costs to be far higher than they are. Automatically building a LOD2 building model can cost between 0.35 and 2 euros per building.