Reconstructing 3D Building Models with the 2D Cadastre for Semantic Enhancement

Frédéric Pedrinis and Gilles Gesquière

Abstract Virtual city models are increasingly used in urban land management processes, which involve the use of different sources of spatial information. This heterogeneous data is, however, often complementary and it may be necessary to give the possibility to join information provided by different sources. This paper presents a method to enhance 3D buildings by using usual 2D vectorial polygon database. These polygons may represent districts, building footprints, or any segmentation of the urban area that adds information to the city model. The enhancement consists in using this polygon database to split the 3D buildings into a set of city objects where each element possesses a 3D geometry and the semantic information of the polygon it is linked to. In this paper, for an illustration purpose, we will present how to create this link between 3D buildings and the cadastre map, in order to create a set of semantically rich 3D building models.

Keywords 3D virtual city · Cadastre · CityGML · Semantic information

1 Introduction

Many cities possess a virtual 3D model representing their territory. They have at their disposal various methods to build their virtual double, depending on the available data and the desired model. Therefore, international standards are currently being developed to manage these 3D city models, like the CityGML standard[1] created in 2008 by the Open Geospatial Consortium[2] (Gröger et al. 2008;

---

[1]http://www.citygml.org/.
[2]http://www.opengeospatial.org/.

F. Pedrinis (✉) · G. Gesquière
LIRIS, University of Lyon, Lyon, France
e-mail: frederic.pedrinis@liris.cnrs.fr

G. Gesquière
e-mail: gilles.gesquiere@liris.cnrs.fr

Kolbe 2009). It is a format that allows the complete description of a building, including 5 different Levels of Details (LoD) and semantic information, but these are rarely present in a newly generated 3D model.

3D urban model construction processes mainly use imagery and/or LiDAR scans to create the 3D geometries of the city (Musialski et al. 2012). They focus on the elaboration of correct 3D geometries in order to propose a visually convincing representation of the city. The semantic richness of buildings defined in the 3D geometries is not really discussed, and these objects can't be directly connected to urban databases. However, other representations of the city can contain the information needed to complete the database. 2D polygonal databases, like for instance the cadastre map, can be used to generate this enhancement of a 3D city model. Such a 2D polygonal database describes an urban area and provides semantic information for each of its elements represented by a polygon. An example would be a partition of the city into polygonal zones according to some characteristics like the buildings date of construction or the average age of their inhabitants. More generally, the method could be used with every 2D database providing a partition relevant to city objects.

In this paper, we choose to focus on the use of cadastre maps to enhance the 3D buildings. In addition to a precise notion of buildings, the cadastre provides semantic information and can grant access to non-georeferenced databases. For example, we can access to the date of construction of a building, its owner's name, its number of housings, or to collect the result of a thermal simulation attached to cadastral buildings. This is why we want to develop a method to link a given 3D building model with the corresponding cadastral polygons. This link is used to create a new model containing, for each building initially defined in the cadastre, a 3D geometry computed from the initial 3D model. The union of these buildings would thus have a 3D model similar to the initial one, and a possible connection with all databases accessible from the cadastre. Figure 1 illustrates this possibility. The cadastral ID provides an access to semantic information. Thanks to our enhancement process described in this paper, the link between the cadastre and the initial 3D model of buildings transfers this accessibility to the resulting model.

Two different ways to generate the link between the cadastre, or any other polygons database, and 3D models can be identified. The first one is to directly involve the 2D polygons in the construction process of 3D geometries, as in Durupt and Taillander (2006), Vallet et al. (2011), and Agugiaro (2014), who used cadastral data to design a robust 3D reconstruction of buildings. The second way consists in a post-process intervention on the geometries in order to associate them to the needed semantic information. In this case, we don't need a process to build a 3D model from raw data, as we can just use an existing one. Ross et al. (2009) propose to add information to 3D city models to create complex 3D Land Information Systems. Among the concerned data, they choose to automatically add attribute information extracted from the cadastre map to the 3D geometries of buildings. The authors do not really describe how this is processed. However, such additions should require a certain overlap between the 3D geometry and the cadastre to be relevant, which is not the case in our datasets. For each object initially called "building" in the 3D model (Fig. 2, top-left), multiple cadastral objects, which also represent "buildings"
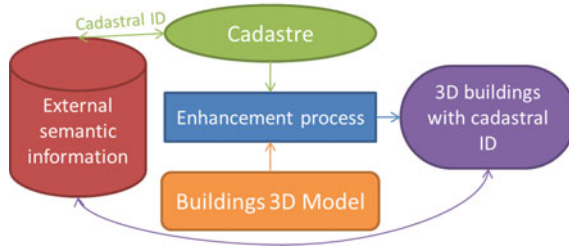
Fig. 1 The link created between 3D buildings and the cadastre generates a new 3D model composed of precise buildings that can access to external databases with their cadastral ID
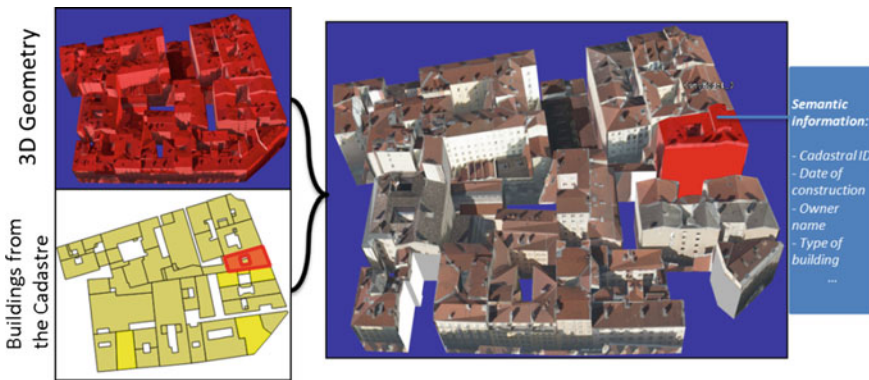

Fig. 2 A single building from a 3D city model (*top-left*) and the cadastral buildings of the same area (*bottom-left*) are linked; a split 3D model according to the cadastre is generated by our method (*right*). A building is picked (highlighted in *red*), and some other ones are hidden (*yellow* polygons on the *bottom-left*). Each of these buildings has a set of semantic information collected thanks to the cadastre

according to their definition (Fig. 2, bottom-left), can match. We have to decide which cadastral object may be linked to the 3D geometry, and which should be ignored. The proposed solution is to split the geometry to provide a new 3D model for each cadastral footprint (Fig. 2, right). The geometry and the texture coordinates are extracted from the initial 3D buildings.

Our tests are made on the data of the city of Lyon (France), which are available on the Lyon Data website.[3] We implemented our method on the software 3D-Use developed by our team.[4]

In this paper, we will first present an overview of our method (Sect. 2). Then, we will present how we create the link between the two datasets with 2D processes (Sect. 3), before involving the 3D to get a better result (Sect. 4). The paper

---

[3]http://data.grandlyon.com/.

[4]http://liris.cnrs.fr/vcity/wiki/doku.php.

continues with the generation process of the new 3D model, which contains the semantic information stored in the cadastre (Sect. 5). We will then propose some tests and discussions (Sect. 6) before concluding and presenting our future works (Sect. 7).

2 Overview of Our Method

To illustrate our method, we will use two data sets. This first one is a CityGML file containing 3D models of buildings in LoD2 (Fig. 3, left). The second one is the French Cadastre which is a 2D vectorial polygons database (Fig. 3, right).

Since the buildings' definition is conveyed by the cadastral polygons, we want to modify these polygons so that they completely cover the surface represented by the footprints of the input 3D models of buildings. Since buildings are modelled in LoD2, covering their footprints means being able to entirely assign their geometry to cadastral polygons and to regenerate 3D geometries, as we will see in Sect. 5. However, if we look closely at the pictures of Fig. 3, we can already identify some inconsistencies between the footprints, such as lacks of courtyards or some differences for small pieces of buildings. This reflects that the two databases do not properly overlap. We illustrate this observation on the left picture of Fig. 4, which is a focus on the rectangular selection of the right picture of Fig. 3.

These differences may have several explanations that are listed below. The first one is linked to the precision that is not the same for the two datasets. A second set of difference is due to the acquisition date. Third, these acquisition methods are simply not comparable. For example, in CityGML LoD2 models of buildings a footprint is defined by the roof shape. As we can see on the right picture of Fig. 4, there are common cases of buildings where such footprints are irrelevant, when

Fig. 3 CityGML buildings are on the *left*, cadastral buildings on the *right*. The *rectangular* selection on the cadastre corresponds to a zoom used in Fig. 4
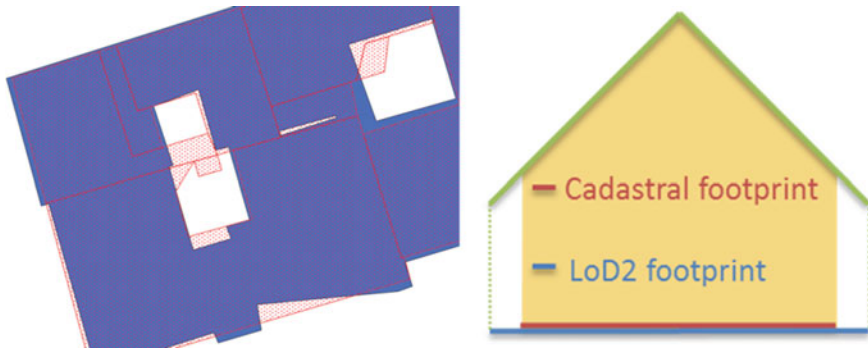
Fig. 4 Inconsistencies between footprints of 3D buildings (*blue*) and cadastral polygons (*red*)

there is an offset between the wall and the roof border, and the cadastre logically does not use them. The footprints are wider than the footprint generated from the real walls that the cadastre probably use. Even if the cadastre footprint logically represents the more realistic representation, our objective here is to split the LoD2 3D geometry and link it with cadastral areas.

The exact process of generating 3D geometries from a footprint and a 3D model will be presented in Sect. 5. If we use this process on the raw data of the cadastre, some wall and roof polygons of the 3D model will not be assigned to the newly generated buildings. More generally, the 3D geometry will be unpredictable, and probably visually unrealistic. We would only obtain a cut of the 3D geometry with an unrelated mask defined by the cadastre, and a part of this geometry would be lost. Figure 5 illustrates this problem: the cut process without any preparation of the footprints is presented on the left, while the initial model is on the right. The lost part of the 3D geometry is illustrated in green, wall and roof polygons that are not entirely on the area defined by the cadastre cannot be naïvely assigned to a building in the final 3D model.

In our dataset based on Lyon CityGML files, we have calculated that the proportion of 3D models, in terms of superficies, that are not covered by the cadastre is about 8 % of the total building superficies. If we cut only with the raw cadastre, we would thus lose a significant part of the 3D model, especially as these losses mainly concern the external envelope of the buildings, which is the most visible part mapped with texture.

Another problem that may appear with a raw cut is the gap between the visible separations of two buildings on the 3D model and the corresponding split defined in the cadastre. A case is illustrated on Fig. 6. A cut is generated with the unchanged footprints of the cadastre (left). The cut generated by our method, which adapt the footprints according to the 3D geometry, is presented on the right and is visually more satisfying.

This means that at least one of the model has to be modified. Since we don't necessary have the data to construct a new 3D textured model, we focus on
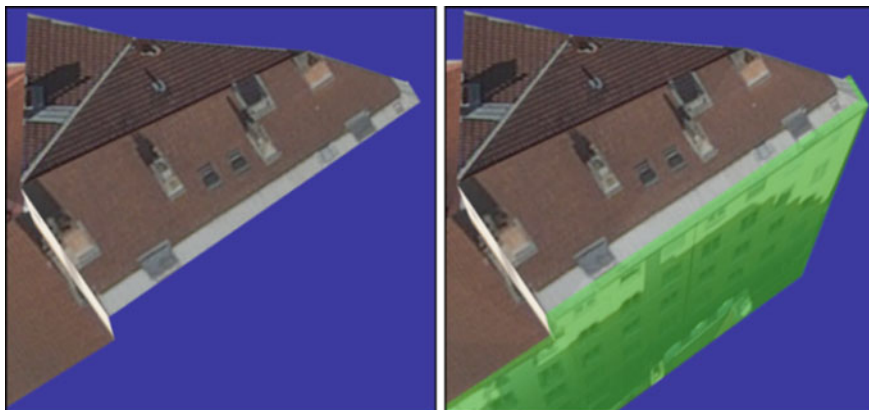
Fig. 5 Cut of the 3D model according to the cadastre without correction will lead to have missing parts (*left*). The lost part of roof and wall polygons are represented in *green* on the initial model (*right*)
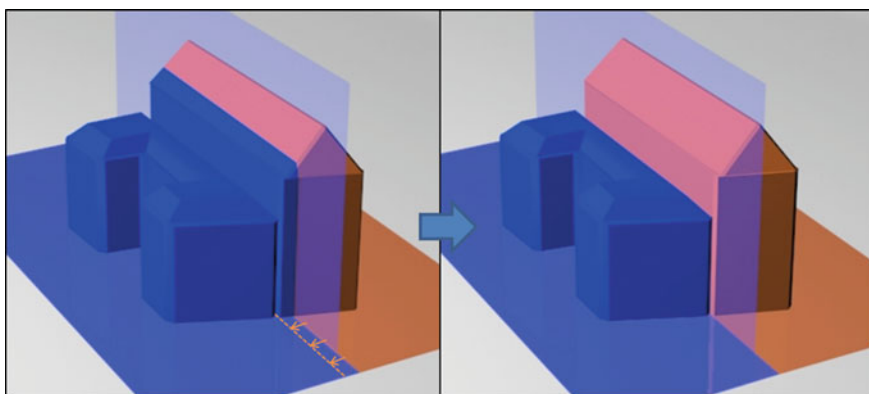

Fig. 6 *Left* image corresponds to a rough cut with the unmodified cadastral footprints. The *right* one is generated with the corrected footprint that takes into account the 3D geometry of the buildings models

adapting the cadastral footprints to the initial 3D geometry, and to keep a link between these new polygons and the original cadastre.

These problems have been identified in Pedrinis et al. (2015) and a first method has been proposed to solve this pairing between the cadastre and 3D buildings in order to provide a better detection of changes between two 3D models available at two different dates. However, this first tentative is not enough to solve all the problems presented in this section.

A global overview of our method is proposed in Fig. 7; it also introduces the notations used in the following of this paper. *G* is a set of 3D geometries that are
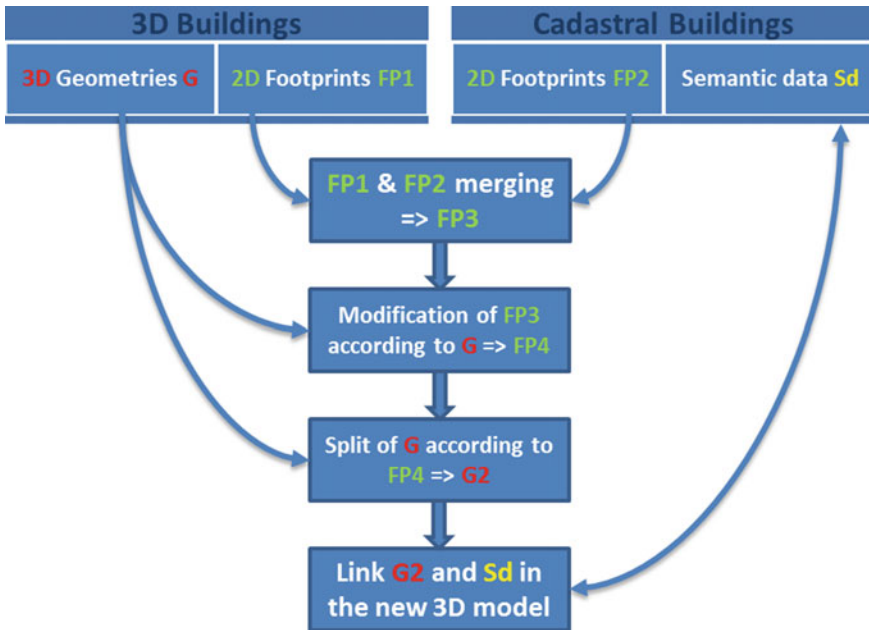
Fig. 7 Presentation of the main steps of our method. The sets of 2D polygons are in *green*, while the 3D geometries are in *red*

used to generate building's footprints by projecting the roof polygons on the ground (*FP1*). On the other side, the set of polygons *FP2* represents the footprints according to the cadastre. Each footprint may have semantic information *Sd*. The two sets of footprints are merged to generate a new one, *FP3*, defining the cadastral footprints from *FP2* that are expanded to cover the entire surface of *FP1*. This step corrects the error shown in Fig. 5. For that presented in Fig. 6, *G* is used to modify *FP3* by adapting footprints to the related 3D geometry. This new set of footprints is called *FP4*. It is then possible to generate a new 3D model *G2* by cutting the 3D polygons of *G* according to polygons of *FP4*. A natural link is now available and *G2* can access to the semantic data *Sd* initially proposed by the cadastre.

3 Merging Footprints from the 3D and Cadastral Models

The previous section has pointed out the need to adapt cadastral polygons to the footprints generated from the 3D models of buildings, in order to be able to use them for the generation of the 3D geometries of new buildings. We thus have two sets of polygons *FP1* and *FP2* and we want to generate a new set of polygons *FP3* that will contain footprints of *FP2* modified to cover the entire surface of polygons from *FP1*. For the example represented on the right picture of Fig. 4, this means
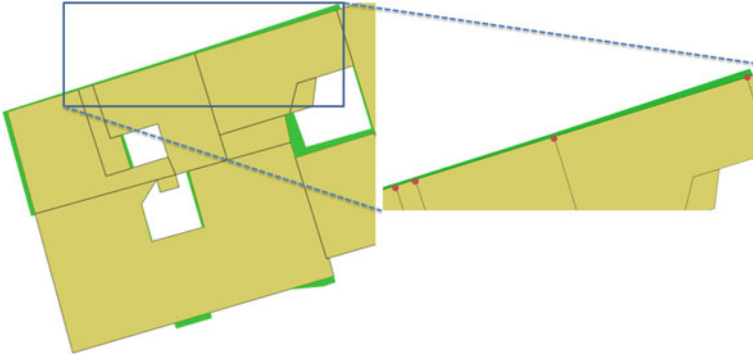
Fig. 8 Difference between *FP1* and *FP2* (beige polygons). The resulting polygons *Diff* are represented in *green*. *Red* points that are on the boundary of two cadastral footprints of *FP2* are then detected

that the red polygons will entirely cover the blue one. To achieve this, the difference between *FP1* and *FP2* is computed to produces a set of polygons *Diff*. These areas correspond to the surface of *FP1* that was not initially covered by *FP2* (presented in green in Fig. 8).

These polygons *Diff* represent the area that needs to be filled by the expansion of the polygons of *FP2*. However, this expansion has to be controlled: the result has to be visually convincing considering that it will define the future division for 3D models of buildings. Each *Diff* polygon is split according to its neighbouring polygons from *FP2*, in order to assign the best possible shape to each of them.

Points on *Diff* that are on the boundary of two cadastral footprints are detected (red points of Fig. 8). These points are used to split *Diff* into distinct polygons, which will be assigned to the concerned cadastral polygons. The resulting footprints need to be visually credible.

The more convincing line passing through P that extends the border line must be chosen to split the *Diff* in two parts. To reach this goal, P is projected on other edges; the best projection P' according to parameters of distance or angles. The angles between [PP'] and the two neighbouring edges of P should be close and the [PP'] segment should be as short as possible. In our implementation, we chose the projection with the shortest segment, except if the two angles are too different. In the example of Fig. 9, P2 and P3 are eliminated by P' and P1 due to their smaller distance to P. P1 is rejected because the angles formed with the neighbouring edges of P are too different. P' is thus chosen and the line defined by P and P' will be used to split *Diff* into two resulting polygons.

This process applied on the selected area of Fig. 8 provides the result presented in Fig. 10. The last step of this process consists in assigning each sub-polygon (blue ones in Fig. 10) to the nearest cadastral footprint (beige polygons) to expand it. The chosen corresponding footprint is the one with which the polygon shares the longest border.
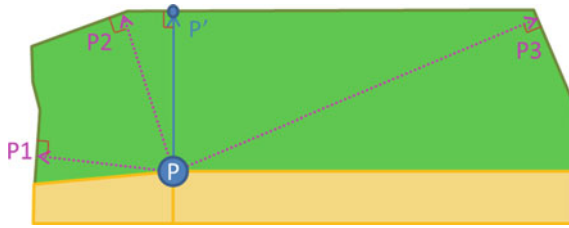
Fig. 9 Orthogonal projections of point P from polygon *Diff* (*green*) on its other edges gives points P1, P2, P' and P3. Due to angle and distance criteria, P' is chosen. Beige polygons are cadastral footprints that were used to detect P
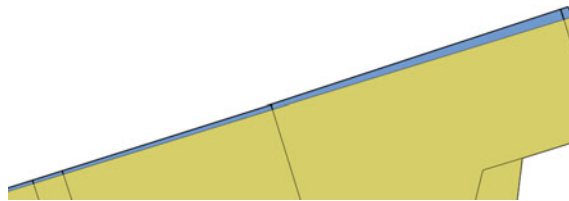

Fig. 10 *Diff* is split into *blue* polygons, which are assigned separately to their nearest cadastral polygon (beige)

Fig. 11 On the *left*, cadastral polygons are extended to cover the footprint *FP1* extracted from the 3D model. On the *right*, *blue* polygons are intersected with *FP1* to delete the non-needed part (not present in the 3D model) and obtaining *FP3*. The *orange rectangle* corresponds to a zoomed area used in Fig. 12

The result on the entire area is shown on Fig. 11 (left). A cadastral footprints covering *FP1* is now available, but they do not perfectly fit: some parts of these footprints are not present in *FP1*, and are not required since they do not correspond to the 3D geometry *G*. The intersection between this set of footprints and *FP1* is then computed (Fig. 11, right). This last result is called *FP3* according to Fig. 7 notations. It represents the exact footprint of *G* split in a set of polygons corresponding to cadastral objects.

4   Correction of Generated Footprints by Using 3D Geometry

This set of polygons *FP3* allows to generate a 3D model, with the process explained in Sect. 5, since these footprints now cover the entire surface of *FP1*. However, as we can see in Fig. 12 (left), which is a focus on the orange selected area of Fig. 11, the result would not be visually satisfactory. Initially, the footprints generated from CityGML buildings only take into account the (x, y) coordinates of roof polygons (Fig. 12, middle). This means that 3D discontinuities of the roof, as seen on the Fig. 12 (left), do not interfere in the choice of splitting edge between two new footprints.

In order to prepare better 3D models, we have to use this information to propose a new version of footprints that will consider the 3D shape of the future generated buildings. The footprints of *FP3* is split according to the roof discontinuities: if a wall of the input 3D model crosses a footprint of *FP3*, it must be split along the 2D line representing this wall (Fig. 12, right). The split results in a set of polygons for each element of *FP3*. These polygons are represented on Fig. 13. Each of them keeps a link to its original footprint of *FP3*. Among these polygons, those that are characteristics of visual inconsistencies are the small or elongated ones: they represent footprints spilling over their neighbours. They are represented in light green in Fig. 13 and we will call them *Sp*. We need to determine which of these polygons must be reallocated to other footprints in order to correct the visual inconsistencies.

The polygons are added to *Sp* if they fall into one of these 3 categories: the very small polygons, the small and non-compact polygons, and the elongated ones. Two parameters are proposed to detect the concerned polygons: the area and the Gravelius Compactness Coefficient (*Kg*) of a polygon. The second parameter *Kg* for a polygon P is defined by $Kg(P) = Perimeter(P)/(2 * \sqrt{\pi * Area(P)})$. This coefficient is initially used to analyse watersheds (Pareta and Pareta 2011). *Kg* is equal to
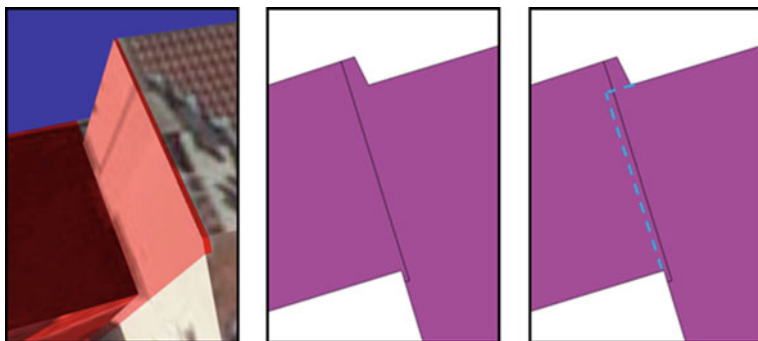

Fig. 12 On the *left* is shown the corresponding 3D cut of the footprints (*middle*), which are taken from the selected area (*orange rectangle*) of Fig. 11. The expected cut corresponding to the wall of the 3D model is represented on *blue* (*right*)

Fig. 13 Split polygons according to 3D discontinuities. *Light green* ones have been selected for their elongated shape or small area

1 if *P* is a circle and is inversely proportional to its compactness (Fig. 14). This parameter is used to detect elongated polygons.

Each of the polygons *Sp* is then compared to the neighbouring footprints of *FP3*. The polygon is reallocated to the footprint with which it shares the longest
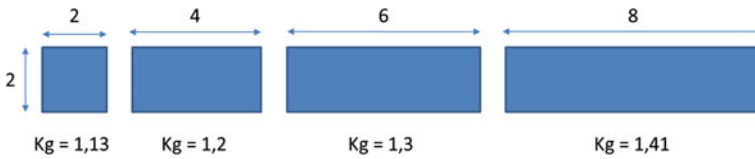

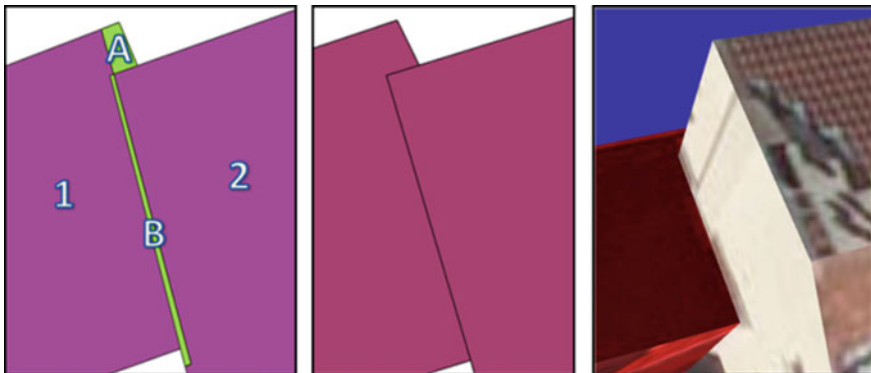**Fig. 14** Some values of the Gravelius Compactness Coefficient (*Kg*)

Fig. 15 Elongated polygons *A* and *B* are detected (*left*) and are respectively merged to footprints *1* and *2* (*middle*) in order to propose two distinct footprints that will give a visually satisfying 3D geometry (*right*)

Fig. 16 Final version of
footprints *FP4* that will be
used to generate 3D models of
buildings

continuous border in the 3D model. If no borders are found, the polygon is left to its
original footprint.

For the illustration presented in Fig. 12, we detect on the left of Fig. 15 the
elongated polygons *A* and *B* (green polygons). We test them with the two cadastral
footprints *1* and *2* (purple polygons). *A* and *1* (resp. *B* and *2*) are considered as the
better neighbours. We can see on the middle of Fig. 15 the two resulting footprints
and, on the right, the resulting 3D models.

The final version of the buildings footprints *FP4* is presented in Fig. 16. This set
of polygons can be used now to generate the 3D model.

5   Creation of the New 3D City Model

Each footprint stored in *FP4* represents a building. We want to create a new 3D city
model containing all these buildings, so we have to generate 3D geometries *G2* for
each of them. Since we want to generate a 3D model for each element of *FP4*, we
split the 3D polygons of *G* accordingly to the footprints of *FP4*, before assigning
the resulting polygons to the concerned footprints in the new CityGML buildings.
Textures coordinates are then computed for each building by using those stored in
the input 3D model. However, existing walls are not enough if we want an inde-
pendent and complete building in the final CityGML file: we have to generate
hidden walls to close each building. Otherwise, we would have holes in the
geometry when a created building would be displayed alone. We thus detect edges
of *FP4* that represent border between two footprints (green edges in Fig. 17), and
we create walls or extend existing walls at their position. For the example of
Fig. 15, we can see a pre-existing wall on the border between footprints 1 and 2.
This wall has to be extended to the ground to generate two closed geometries. These
generated walls do not have textures because they were initially hidden, so they are
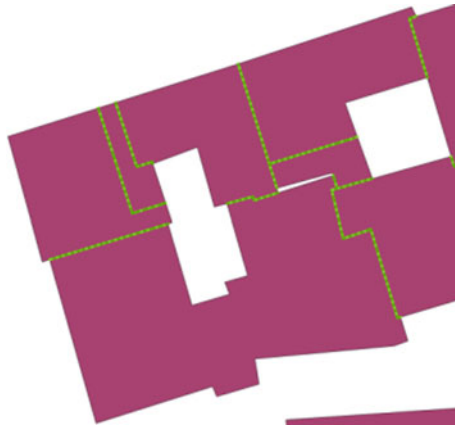
Fig. 17 Detection of missing walls for generated buildings

Fig. 18 Semantic information extracted from a footprint and linked to a 3D building

displayed with a default colour (here white) and can be viewed by hiding their neighbours (Fig. 2, right).

Since polygons of *FP4* are originally extracted from cadastral footprints, the semantic related information can be assigned to the corresponding generated 3D model (Fig. 18). For instance, here, data describing the construction and modification dates, the surface and the perimeter are available.

6 Performance and Discussion

We have tested our method on two districts of the city of Lyon (France). The result on the first district is presented Fig. 19. Table 1 gives some performances data generated with a computer on Windows 7 with an I7-4790K @ 4.00 GHz CPU.
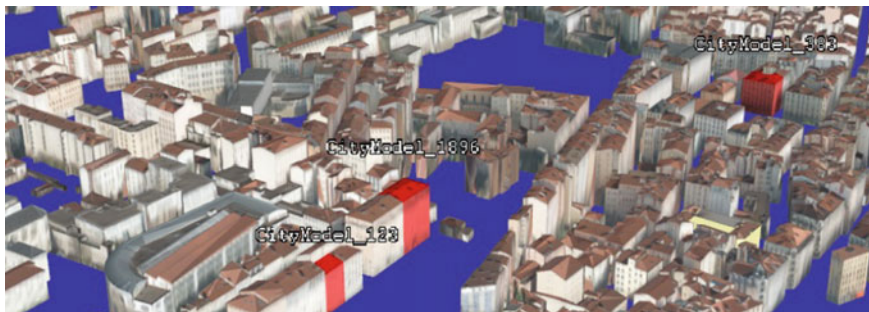
Fig. 19 Enhanced 3D model describing the first district of the city of Lyon. A multi-selection of generated buildings is highlighted in *red*

Table 1 Performance of enhancing CityGML for two districts of the city of Lyon. Each building of the output CityGML file corresponds to a cadastral footprint

District (superficy)	Input CityGML file		Enhancement process	Output CityGML file	
	Number of buildings	File size (Mb)	Time	Nb of buildings	File size (Mb)
Lyon 01 (1.5 km^2)	409	110	1197 s (20 min)	2640	132
Lyon 04 (2.9 km^2)	999	118	1461 s (24 min)	4339	148

We can see on Table 1 that the number of buildings logically increases after the process; however it does not mean that all buildings from the initial CityGML file or those from the cadastre have been kept. It is possible that some footprints did not find any equivalent in the other database. This information can be exported by our software 3D-Use since it may be useful in a validation process of the cadastre for example: these differences can be representative of an out of date cadastre. In our case, this may signify that our process should then require a human intervention to verify the corresponding buildings, on the field or with another model of the city. After discussing with urban operators who manually realize a similar geometry cut, we do think it is anyway necessary to keep a human validation step in our global enhancement process. Some cases lead to disagreements even between different human operators, so changes should be possible on the created model by following the process depicted in Fig. 20. *FP4* are used to generate *G2* but an expert has the possibility to detect inconsistencies in the 3D model cut, eventually guided by computed indicators based on 3D analysis processes. Concerned footprints are then exported to allow modify manual correction by the expert before reinserting them in the set of footprints used to generate a new version of the buildings 3D model. This process can be repeated several time and stopped by the expert or when the possible inconsistencies become under a given threshold.
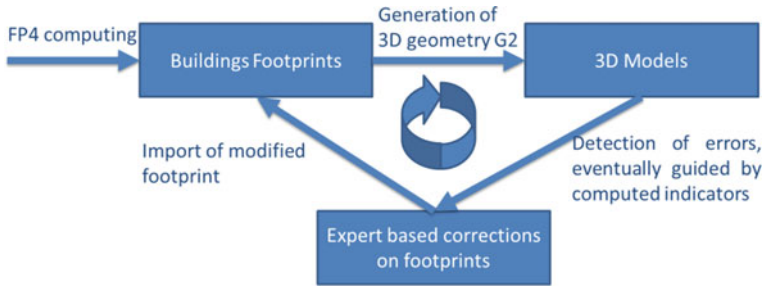
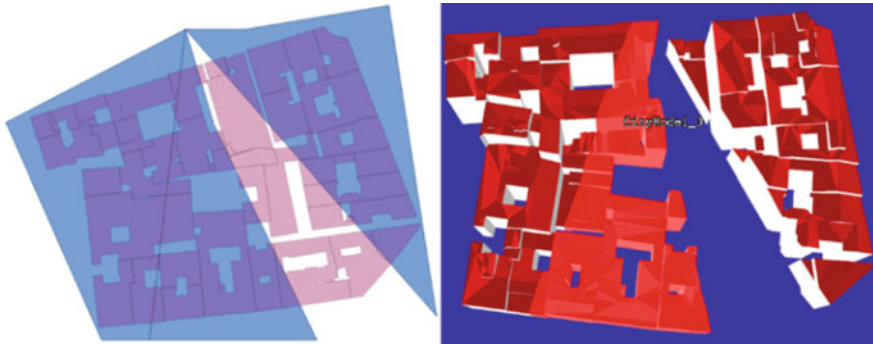**Fig. 20** Semi-automatic validation and correction of the computed footprints

Fig. 21 Cut of a 3D building (on the *right*) with some polygons (*blue* ones, on the *left*)

Decomposing the method in two parts also gives the possibility to use an extra set of polygons in order to cut 3D geometry. We illustrate this possibility in Fig. 21: a set of random polygons (in blue) are used to cut the initial building. In this way, the second part of our method could be used in several other applications like a tiling process for example, or for any method that requires to group buildings accordingly to 2D shapes, which could eventually contain semantic information.

7   Conclusion and Future Works

Nowadays, a large amount of 3D city models are available. The accuracy of reconstruction processes of these models has been increased but the linked semantic is often poor and it may be difficult for a user to link the needed information with these 3D models. On the other side, a large diversity of semantic information linked to 2D models like the cadastre is available. In this paper we propose a method to link these two kinds of data in order to enhance dedicated data linked to 3D buildings.

Since these two models have no reason to be correctly fitted, some processes are required to pair their representation and to generate a merging model. We have proposed corrections on the 2D polygons to make them split the entire 3D geometry and to get convincing new 3D models of buildings. The two datasets are thus closely related. The 3D geometry is involved in the description of the footprints based on the 2D database, while each of them is then used to cut the 3D model and generate a new 3D building. This link between two datasets can also be used to compare two models representing the city at two different dates (inducing a real enhancement of the method proposed in Pedrinis et al. (2015)). It also allows to evaluate one of the model by a second which is deemed true.

The presented processes are not necessarily sufficient to generate a city model wholly convincing. They may need to be followed by a semi-automatic procedure involving an urban expert to produce a better 3D model of the city.

The presented method has been presented to experts. The feedbacks are very positive. Our approach could provide a good alternative to their actual time consuming methods.

Nevertheless, some works remain to be done, in particular to propose a semi-automatic loop for a guided correction by experts in our software. For example, even if the process that automatically corrects error such as 3D inconsistencies in the resulting model can be improved, it should be more relevant to begin by just proposing a detection process of these errors to help the expert correction task. For example, we may propose an evaluation of our results by computing matching indicators between a cadastral footprint and the corresponding 3D geometry.

The method presented in this paper was developed with LoD2 buildings according to the CityGML definition, it may also be necessary to extend it to more precise models.

References

Agugiaro, G. (2014). From sub-optimal datasets to a CityGML-compliant 3D city model: Experiences from Trento, Italy. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, XL-4*, 7–13.

Durupt, M., & Taillandier, F. (2006). Automatic building reconstruction from a digital elevation model and cadastral data: An operational approach. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences, 36*(Part 3), 142–147.

Gröger, G., Kolbe, T. H., Czerwinski, A., & Nagel, C. (2008). OpenGIS city geography markup language (CityGML) encoding standard, Version 1.0.0, International OGC Standard, Open Geospatial Consortium, Doc. No. 08-007r1.

Kolbe, T. H. (2009). Representing and exchanging 3D city models with cityGML. In J. Lee & S. Zlatanova (Eds.), *3D geoinformation sciences*. Berlin: Springer.

Musialski, P., Wonka, P., Aliaga, D. G., Wimmer, M., van Gool, L., & Purgathofer, W. (2012). A survey of urban reconstruction. Eurographics 2012–State of the Art Reports (pp. 1–28).

Pareta, K., & Pareta, U. (2011). Quantitative morphometric analysis of a watershed of Yamuna basin, India using ASTER (DEM) data and GIS. *International Journal of Geomatics and Geosciences, 2*(1), 248–269.

Pedrinis, F., Morel, M., & Gesquière, G. (2015). Change detection of cities. In *3D geoinformation science* (pp. 123–139). Springer International Publishing.

Ross, L., Bolling, J., Döllner, J., & Kleinschmit, B. (2009). *Enhancing 3d city models with heterogeneous spatial information: Towards 3d land information systems. In advances in GIScience* (pp. 113–133). Berlin: Springer.

Vallet, B., Pierrot-Deseilligny, M., Boldo, D., & Brédif, M. (2011). Building footprint database improvement for 3D reconstruction: A split and merge approach and its evaluation. *ISPRS Journal of Photogrammetry and Remote Sensing, 66*(5), 732–742.