



HAL
open science

Parcel Manager: a parcel reshaping model incorporating design rules of residential development

Maxime Colomb, Cécile Tannier, Julien Perret, Paul Chapron, Mickaël Brasebin

► To cite this version:

Maxime Colomb, Cécile Tannier, Julien Perret, Paul Chapron, Mickaël Brasebin. Parcel Manager: a parcel reshaping model incorporating design rules of residential development. *Transactions in GIS*, 2022, 26 (6), pp.2558-2597. 10.1111/tgis.12970 . hal-03700820

HAL Id: hal-03700820

<https://hal.science/hal-03700820>

Submitted on 15 Aug 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Parcel Manager: a parcel reshaping model incorporating design rules of residential development

Maxime Colomb^{a,b,*}, Cécile Tannier^b, Julien Perret^a, Paul Chapron^a, Mickaël Brasebin^a

^a*LASTIG, Univ Gustave Eiffel, IGN-ENSG, Saint-Mande, France*

^b*ThéMA UMR 6049, CNRS-University, Besançon, France*

Abstract

Parcels are the smallest units of land usually considered for urban development purposes. Analyses at the parcel level allow taking into account the administrative and physical environment of buildings with accuracy (roads, surrounding buildings, local design rules for building construction). In this article, we propose a free and open source software application, namely *Parcel Manager*, whose aims are twofold: 1) to assess the effects of different parcel reshaping rules and planning principles on the location, the number and the shape of reshaped parcels, and 2) to provide a basis to assess the potentials for new building constructions. To this end, *Parcel Manager* simulates a large variety of parcel reshaping processes, with or without the joint creation of new roads, and produces realistic parcel layouts that represent either infill urban developments, edge expansions or leapfrog developments. It can be used to determine if the densification of built parcels is feasible or not, regarding planning and design rules as well as the current urban fabric. The current version of *Parcel Manager* only concerns the reshaping of parcels dedicated to the construction of residential buildings but not other types buildings.

Keywords: Urban fabric, computer-based simulation, parcel layout, densification, urban design

1. Introduction

A parcel, defined here as a lot of land owned by one or several owners, is the smallest unit of land usually considered for urban development purposes. Descriptive analyses at the parcel level allow taking into account the current
5 physical and administrative environment of buildings with accuracy. Dynamic

*Corresponding author

Email addresses: maxime.colomb@inria.fr (Maxime Colomb),
cecile.tannier@univ-fcomte.fr (Cécile Tannier), julien.perret@ign.fr (Julien Perret),
paul.chapron@ign.fr (Paul Chapron), mickael.brsebin@gmail.com (Mickaël Brasebin)

simulations using parcels as basic spatial entity are used for demographic census estimations (Jarosz et al., 2008), land-use change analyses (Donnelly and Evans, 2008; Wilson and Song, 2010; Sun and Robinson, 2018), urban growth simulations (Abolhasani et al., 2016), or predictions of future land-use changes (Pocewicz et al., 2008). The increase of available computational resources has even made possible to perform simulations at the parcel scale for continental-sized study areas (Long and Wu, 2017).

Considering the parcels themselves, simple simulation models represent the changes of the state of parcels, e.g. landownership, land use or building density. Parcel reshaping, that is the modification of the boundaries of parcels with respect to a set of morphological constraints (area of reshaped parcels, width, connection to the road network...), is the objective of few applications (Vanegas et al., 2012; Wickramasuriya et al., 2011; Lagrab et al., 2018; Demetriou et al., 2013). Some algorithms contained in those applications have been introduced in the toolboxes of CAD software applications (e.g. the AutoCAD add-on *TOP-parcel* (ARKISoft, 2020)) or in the toolboxes of GIS software applications (e.g. the ArcGIS add-ons *Block Parameter* (City Engine, 2019) and *Parcel Divider* (Dahal and Chow, 2014)). Those toolboxes can be used for creating or reshaping parcels in small- or medium-sized zones that do not contain buildings. Another family of projects focus on the generation of new urban scenes from scratch. The corresponding software applications simulate not only parcel creations but also the creation of buildings and roads (Parish and Müller, 2001), and sometimes leisure and green infrastructures as well (Yang et al., 2013; Yazýcý, 2016).

A common objective of parcel reshaping software applications is to create an optimized parcel plan using an optimization function. Optimization criteria can concern the number of parcels, the types of land-uses (Yang et al., 2013), land ownership and parcel geometry (Demetriou et al., 2012), area of roads (Wickramasuriya et al., 2011), agricultural characteristics and distance from the homestead to the transportation network (Harasimowicz et al., 2017), land prices and land-use sustainability (Demetriou et al., 2013; Brennan and Venigalla, 2016; Yazýcý, 2016), environmental sustainability (Yazýcý, 2016), or even flooding risks (Mustafa et al., 2018). Another common objective is to provide tools that can be used by urban practitioners to design parcel layouts that conform with a given set of rules considering e.g. the size of parcels, their shape or their connection to the road network (Lipp et al., 2011; Vanegas et al., 2012; Yang et al., 2013; Dahal and Chow, 2014).

In many countries in the world, official regulations constrain more or less strictly the reshaping (division and merging) of parcels. For instance, the Land Division Act in Michigan state (USA) sets the following rules for the divided parcels¹: Have an adequate and accurate legal description; Not be narrower than 4:1 (parcel depth to width ratio for parcels less than 10 acres); Meet the minimum parcel width required in the zoning ordinance, if applicable; Meet

¹<https://tinyurl.com/4mcrb884>

the minimum parcel size required in the zoning ordinance, if applicable; Be
accessible by a public road, private road, easement or other similar means (as
50 required by the local land division or zoning ordinance); Not exceed the maximum
number of divisions for the parent parcel, or the number of re-divisions
for the division; Have adequate easements for public utilities from the parcel to
existing public utility facilities; Not result in land-locking a cemetery; and Not
have any unpaid property taxes and/or special assessments due for the last five
55 years. Considering a large area (typically, a large metropolitan area or a city
region), the official regulations may vary in space with respect to each local
authority or to each zone of a zoning plan. Moreover, in some countries, public or
semi-public planning programs can impose the reshaping of a set of contiguous
parcels initially owned by one or several persons, and impose strict rules for this
60 reshaping in order to create a new neighbourhood.

In some other countries, however, no official regulations constrain the
reshaping of parcels. The owner of a parcel can choose the most advantageous
division method to obtain the largest land area that she/he can sell (at a good
price, e.g. for housing purposes) and to minimize the area intended for e.g.
65 access to a plot, green areas, etc.

In all cases, it seems interesting to simulate and assess the effects of varying
morphological rules and methods that determine the reshaping of parcels on
the location, the number and the shape of parcels that could be created and,
ultimately, on the potentials for new building constructions. Obviously, different
70 parcel shapes allow the construction of different types of buildings (individual
houses, blocks of flats, etc): the shape of parcels determines indeed the number,
the shape and the location of buildings that could be constructed within them.
Conversely, the type of buildings targeted to be built should constrain the shape
of the reshaped parcels (measurements of the parcels, width of roads created by
75 simulation, etc). However, none of the existing parcel reshaping models provide
a basis to assess the potentials for new building constructions (Brennan and
Venigalla, 2016).

Additionally, a common planning recommendation is to favour the construction
of buildings in already built parcels (i.e. increasing the local built density)
80 instead of promoting greenfield developments (Cooper et al., 2002; Næss et al.,
2019). Yet none of the existing parcel reshaping models deal with the division
of parcels that already contain buildings.

Last but not least, one major planning issue is to achieve goals that often
contradict one another, for instance, satisfying the housing demand quantitatively
85 and qualitatively while preventing new building constructions in natural
areas and green corridors. However, none of the existing parcel reshaping models
specifically enable taking into account jointly urban design rules set in local
urban master plans and other planning recommendations, set e.g. in regional
housing plans or ecological planning documents.

90 In this article, we propose a free and open source software application for
simulating parcel reshaping, namely *Parcel Manager*, which aim is to assess the

effect of urban design rules and planning recommendations on the location, the number and the shape of new parcels that could be created as well as to provide a basis to assess the potentials for new building constructions. For this purpose, *Parcel Manager* simulates a large variety of parcel reshaping processes, with or without the joint creation of new roads, and produces realistic parcel layouts that represent either infill urban developments, edge expansions or leapfrog developments. Simulations can consider simultaneously different types of input zones (i.e., individual parcels located within an urban fabric or large non-built zones where greenfield development is considered) and can be performed automatically for a large region in which parcel reshaping rules and planning regulations may vary spatially. *Parcel Manager* also enables urban practitioners to determine if the densification of built parcels is feasible or not, regarding both planning and design rules as well as the current urban fabric (parcels, buildings and roads). The targeted built density of parcels created by simulation can be more or less high. The current version of *Parcel Manager* only concerns the reshaping of parcels dedicated to the construction of residential buildings but not the construction of commercial and industrial buildings, and neither other types of activities (agriculture, sport, etc.).

The rest of the paper is organized as follows. In section 2, we present a state of the art of models that simulate parcel reshaping processes and situate *Parcel Manager* with respect to these models. In section 3, we present the *Parcel Manager* model. We test specifically each parcel division algorithm contained in *Parcel Manager* in section 4 and we compare the shape of parcels created by simulation with the shape of parcels in real cases in section 5.

2. State of the art

2.1. Types of parcel division processes

Numerous algorithms automatically divide large parcels into small lots (Table 1). The *OBB* algorithm first generates a bounding box around the parcel under consideration. This bounding box is then split in two and this operation is iteratively repeated until morphological requirements (most often, parcel area or parcel width) are satisfied. It is possible to add an irregularity parameter in order to introduce some diversity in parcel sizes.

Skeleton-based algorithms create a skeleton within each parcel to be split. Two types of algorithms exist: the straight skeleton algorithm (Vanegas et al., 2012) and the algorithm based on the median axis of parcels to be reshaped (Perret, 2006).

Wickramasuriya et al. (2011) propose another algorithm based on the generation of a street grid network on the area under consideration. Streets are created along the boundaries of the grid cells and the grid cells are transformed into parcel lots. This operation is done for three grid positions and the algorithm keeps the grid that minimizes the street area. As the algorithm automatically generates parcel lots and streets (i.e. roads), it is used for parcel division in

Table 1: Existing methods for parcel division

Name	Description of the algorithm	Best cases of use	Quoted in
Oriented Bounding Box (OBB)	Split iteratively an input parcel using boxes oriented regarding the longest width of the parcel or its position with respect to the street. The algorithm stops when a threshold (minimum area or width of the split parcels) is reached	Medium to small-sized non-urbanized zones	Parish and Müller (2001); Vanegas et al. (2012); Mustafa et al. (2018); City Engine (2019)
Skeleton-based method	Parcels are divided according to a skeleton generated by simulation. Strips beneath the edges of the skeleton are divided to create parcels that follow the road they are facing and that conform with some morphological parameters	Medium-sized elongated non-urbanized zones	Vanegas et al. (2012); City Engine (2019)
Offset	When a parcel is divided, a zone in the centre of the parcel is preserved; the OBB algorithm is applied to the border of the parcel only	Medium- to large-sized elongated non-urbanized zones	Vanegas et al. (2012); City Engine (2019)
Template-based algorithm	Road creation follows a pre-defined template having a specific form (cul-de-sac, L-shape or T-shape). Parcels are then generated in-between the roads	Medium- to large-sized zones which shape must be adapted to the chosen template (i.e. T-shape template applies only on a T-bone shaped zone)	Yang et al. (2013); Dahal and Chow (2014)
Voronoi cut	Thiessen polygons are generated around points that represent an approximate position for parcel creation	Medium-sized non urbanized zones	Demetriou et al. (2013)
Fixed parcel number	The user fixes a targeted number of output parcels; then the algorithm operates the parcel division following a given direction or starting from the centre of the zone under consideration	Medium-sized non urbanized zones with a predefined program	Lagrab et al. (2018); ARKISoft (2020)
Street grid	Generation of a street grid network overlaying the zone under consideration; then, creation of parcels in-between the streets. Parcel division can be made with different algorithms: OBB in (Mustafa et al., 2018) or fixed parcel number in (Wickramasuriya et al., 2011)	Large to medium-sized non-urbanized zones	Wickramasuriya et al. (2011); Dahal and Chow (2014); Abolhasani et al. (2016)

135 the cellular automata model of Abolhasani et al. (2016). It is also the basic algorithm of the *Parcel Divider* toolbox (Dahal and Chow, 2014).

The module of the LACONISS model (Demetriou et al., 2013) reshapes parcels using a Voronoi decomposition applied to the centroids of existing parcels. By randomly moving the input centroids, which come from another module of the LACONISS model, different parcel plans are created and compared using a
140 genetic algorithm that optimizes multiple criteria.

The software application of Lagrab et al. (2018) produces equal area parcels starting from the centre of the parcel to be split, or following a particular di-

rection (most often horizontal or vertical), as does the AutoCAD *TOP-parcel* add-on (ARKISoft, 2020).

145 Wiseman and Patterson (2016) have applied and tested several division algorithms on different types of blocks of parcels in North-America. They have analysed the similarity of the simulated parcels with the existing parcels using a series of metrics (parcel shape index (McGarigal and Marks, 1995), area, width of the contact between parcels and roads, proportion of parcels with road access).
150 Regarding the clarity of the algorithms and the possibility of implementing them, Wiseman and Patterson (2016) conclude that only the two algorithms of Vanegas et al. (2012) (*Straight Skeleton (SS)* and *OBB*) and the *street grid* method of Dahal and Chow (2014) are comparable. All three algorithms provide realistic parcel layouts. Other algorithms that simulate parcel division processes
155 can only be used for exploratory purposes because the simulated parcel layouts are not realistic. Still according to Wiseman and Patterson (2016), the *street grid* method provides less realistic parcel layouts than do the *OBB* and the *SS* algorithms. *OBB* tends to be more realistic regarding parcel size while *SS* tends to be more realistic regarding parcel orientation. In any cases, *OBB* produces
160 the most realistic parcel layouts regarding the majority of shapes of parcels to be reshaped.

In view of this, as parcel layouts simulated with *Parcel Manager* have to be realistic, the *Straight Skeleton (SS)* and the *OBB* parcel division algorithms have been selected to be included in *Parcel Manager*. In case of *OBB*, the basic
165 version of the algorithm (Vanegas et al., 2012) is well adapted to divide medium to small-sized parcels located close to an existing road. However some tests we did have shown that is not efficient when parcels are square-shaped or when large parts of the parcels to be reshaped are far from the road since it often creates very elongated parcels and allows that some parcels are disconnected
170 from the road. As such, the *OBB* method has to be improved to overcome these limitations in *Parcel Manager*. In case of *SS*, the division of large-sized square-shaped zones creates unrealistic parcel shapes. Thus the algorithm has to be modified to better manage such cases. To this end, the *Offset* division function, which creates a patio in the centre of a block of reshaped parcels, could
175 be introduced in the *SS* algorithm.

One objective of *Parcel Manager* is to determine if the densification of existing parcels is feasible or not, regarding both planning and design rules as well as the current urban fabric (parcels, buildings, and roads). Yet none of the existing parcel division algorithms deal with parcels that already contain buildings. It
180 is thus necessary to develop for *Parcel Manager* a specific process that enables the division of built parcels without modifying the existing buildings.

2.2. Creation of roads

Parcel reshaping methods often simulate the joint creation of new roads. This simulation occurs either before the creation of parcels, and parcels are then
185 created in-between the roads (Parish and Müller, 2001; Mustafa et al., 2018), or concomitantly as in case of the *street grid* division method (Wickramasuriya

et al., 2011; Dahal and Chow, 2014). In some models, the creation of a road network takes into account more constraints than needed for parcel generation, e.g. water, forest, topology, population density for Parish and Müller (2001) and Chen et al. (2008); minimum cost and minimum travel time for Levi et al. (2019). In other models, the creation of a road network is simulated on the basis of templates (predefined patterns) (Sun et al., 2002; Yang et al., 2013; Dahal and Chow, 2014). The choice of a template can depend on the shape of the block under consideration.

Major roads and local roads are sometimes differentiated (Chen et al., 2008; Mustafa et al., 2018). Driveways are never considered perhaps since they are often (co-)owned and included within the parcels, unlike roads. Yet it is important to taken them into account in order to ensure that all lots have access to the road network.

In case of *Parcel Manager*, the simulation of new road creations is mandatory because planning and design rules often stipulate that all residential parcels must have a direct access to the road network. One parcel division method chosen for *Parcel Manager* is the *OBB* method. A process of road creation has been introduced in this method by Dahal and Chow (2014): a new road is created on the boundary lines of the bounding boxes each time a parcel is split. However, with this process of road creation, a very high number of roads are created when a zone is divided into numerous newly created parcels. A modification of the algorithm is thus needed so that the road creation in *Parcel Manager* does not occur during all steps of the division process. Another parcel division method chosen for *Parcel Manager* is the *SS* method. The generation peripheral roads on the perimeter of each zone under consideration has to be added to this method, as done e.g. by Dahal and Chow (2014) in case of the *street grid* method. Besides, it seems useful to add specific functions in *Parcel Manager* in order to differentiate major roads from local roads when roads are created with either the *OBB* or the *SS* method.

As argued in the previous subsection, *Parcel Manager* must also include a specific process that enables the division of built parcels without modifying the existing buildings. In order to ensure access to the road network for all newly created parcels, a common practice is to create a driveway through one side of the divided parcel that is directly connected to a road. It seems relevant to implement this process, called *flag division*, in *Parcel Manager*.

2.3. Merging of parcels before the division

Merging parcels together, also called *land consolidation*, is another issue of parcel reshaping. Land consolidation can apply in view of both greenfield and brownfield developments (Demetriou et al., 2012). In the model of Harasimowicz et al. (2017) agricultural parcels are merged in order to optimize the location of field lands with respect to the homestead. In the model of Asiamia et al. (2019), the process of merging parcels takes account for land ownership and social relations between economic actors. In the LACONISS model (Demetriou et al., 2012), all contiguous parcels worth reshaping are merged together into

a single block before their division. Then a *land partitioning model* is used to estimate how many parcels should be created within this block and which characteristics the parcels should have (width, area, uniformity of parcel shapes, land price).

235 In case of *Parcel Manager*, the merging of parcels before their division must occur only in the zones where a planning project involved land consolidation.

2.4. Selection of parcels or zones to be reshaped

The selection of parcels or zones to be reshaped is necessary to determine where to apply parcel reshaping processes. In existing models, this selection
240 often involves an optimization process that aims at maximizing environmental or functional criteria (Brennan and Venigalla, 2016; Yazıcı, 2016; Mustafa et al., 2018). In the model of Kilić et al. (2019), numerous criteria are calculated in order to select a set of parcels of interest for designing new sustainable urban operations. Criteria considered are grouped into several categories (geometry,
245 road infrastructure, green area, location in the city, legal owner of parcel) and relate to three different scales (single parcel, block and city). These criteria are weighted with the AHP method of Saaty (1977). In case of the parcel-based cellular automata model of Abolhasani et al. (2016), all non-urbanized zones are divided. Then, a chosen set of parcels is selected for being reshaped
250 regarding the preferences set by the user with the AHP method and the state of neighbouring parcels. In the *LACONISS* model (Demetriou et al., 2012), parcels worth reshaping are selected on the basis of a series of criteria, which mainly relate to land ownership status and the characteristics of the wanted urban scene. Land value can also be a criterion for the selection of parcels to be
255 merged (Demetriou, 2018). It can be obtained from market prices or calculated based on accessibility, irrigation status, parcel size, slope, aspect, and distance to the closest residential zone (Demetriou, 2017).

In case of *Parcel Manager*, the simulations can concern large study areas, typically a metropolitan area or a whole region, in which planning principles
260 and parcel reshaping rules may vary spatially. Thereby, *Parcel Manager* must integrate specific functions for selecting the parcels to be reshaped according to the planning principles and the reshaping rules that concern them.

2.5. Automatic, semi-automatic or interactive reshaping of parcels

Depending on the model, the reshaping of parcels can be automatic, semi-
265 automatic or interactive. With the toolboxes (Dahal and Chow, 2014; City Engine, 2019), the user chooses a particular algorithm and the parameters for each zone in which she/he wants to create or reshape parcels. The parcel reshaping process is then run automatically. Other applications operate in a semi-automatic way. In Yang et al. (2013), for instance, the user first draws the
270 zones where parcel division algorithms will be applied. She/he then chooses a template of block shapes and a set of parameters that will be applied on each zone. If a zone does not fit with the chosen template division, the algorithm

can sometimes generate new roads and new subzones using the streamlines of a tensor field (Chen et al., 2008). Other applications operate interactively. For instance, Lipp et al. (2011) propose a fully interactive model that re-creates parcel plans and road networks when the user adds or moves a road. In Vane-gas et al. (2012), a possible option is to manually operate geometric changes at each step of the parcel division process.

In case of *Parcel Manager*, simulations have to be performed on large study areas (typically, a metropolitan area or a whole region) in which parcel reshaping can concern numerous zones that can be more or less large. As such, the reshaping of parcels in *Parcel Manager* must be automatic.

2.6. Additional requirements for the development of *Parcel Manager*

As exposed in the introduction of the paper, one objective that supports the creation of *Parcel Manager* is the further assessment of potentials for new building constructions in the parcel layouts created by simulation. Obviously, different parcel shapes allow the construction of different types of buildings: the shape of parcels determines the number, the shape and the location of buildings that could be constructed within them. Consequently, the assessment of potentials of new building constructions in the parcel layouts created by simulation requires to introduce in *Parcel Manager* the possibility of choosing which type of urban fabric is targeted to be built in each parcel or set of parcels selected to be reshaped. A targeted type of urban fabric can be defined as a set of morphological parameters determining the shape of the reshaped parcels (measurements of parcels, width of roads created by simulation) that vary according to the type of buildings planned to be constructed. The choice of the type of urban fabric planned to be built in each place must also account for the existing urban environment (surrounding buildings, roads...).

2.7. Computer implementation of parcel reshaping models

Most of the existing applications use proprietary libraries such as ArcPy and can not be freely reusable (Demetriou et al., 2012; Dahal and Chow, 2014; City Engine, 2019). Besides, add-ons for e.g. ArcGIS (Dahal and Chow, 2014; City Engine, 2019) or Grasshoper (Yazýcý, 2016) only concern proprietary software applications. In contrast, Mustafa et al. (2018) have developed their algorithms in C++ and Harasimowicz et al. (2017) have developed their model with the open-source GLPK environment. However, to our knowledge, their source code is not freely available. Indeed, Wiseman and Patterson (2016) highlight that the source code is never provided with the articles that present parcel reshaping software applications. As a consequence, the re-use of any existing computing code for the development of *Parcel Manager* is not possible.

3. Presentation of *Parcel Manager*

Parcel Manager is designed to simulate a large variety of parcel reshaping operations within numerous zones of a single community, a metropolitan area or

315 a whole region (Figure 1). A reshaping operation can consist in the reshaping of a single parcel, which can be built or not, or the reshaping of contiguous parcels, or even the creation of parcels within a large zone (land consolidation).

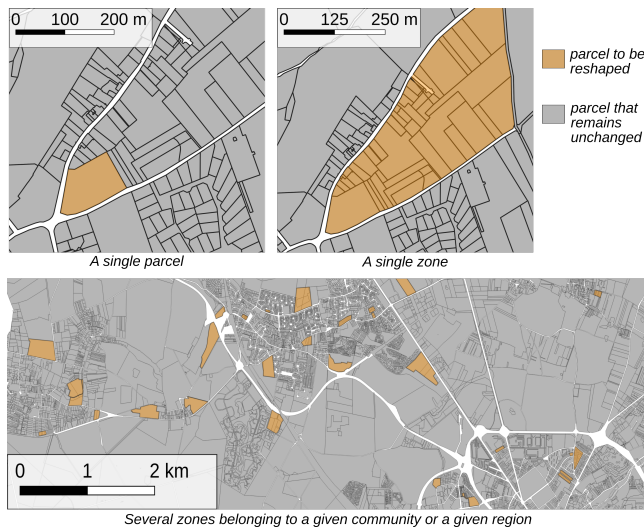


Figure 1: Types of input areas in which *Parcel Manager* can create and reshape parcels

320 In order to perform different types of parcel reshaping operations within numerous zones of an area under study, the architecture of *Parcel Manager* is modular. It involves several parametrizable *parcel division processes* (see section 3.1). Each of them can be enlisted into a *workflow* that simulates the application of a given set of planning and design rules to a unique type of urban fabric (see section 3.2). One or several *workflows* can then be combined within a *scenario*, which sets which workflow applies on which places of the area under study (see section 3.3).

325 *Parcel Manager* deals with both the parcel geometries and their attributes. A simple attribute nomenclature, composed of a community code, a section code and a number, is automatically assigned to parcels and is sufficient to work with *Parcel Manager*. Nevertheless, users can implement their own nomenclature². Concerning the geometries of parcels, topological problems are corrected inter-
330 nally using the JTS (Java Topology Suite) library. As input parcel plans are often derived from multiple digitised sheets, small gaps or overlaps between parcels can exist. They are automatically handled by *Parcel Manager* in order to avoid topological problems in the course of parcel reshaping processes.

335 Following the open science principles (Munafò et al., 2017), *Parcel Manager* algorithms and processes are available in a free and open source library

²See the documentation about the attributes of parcels (<https://tinyurl.com/3tzshp23>)

(see Appendix .1). A technical documentation is available in the Javadoc format (<https://tinyurl.com/yckzcu94>). Besides, *Parcel Manager* can be used by people having no computing skills via a graphical user interface.

3.1. Parcel division processes

340 Three parcel division processes have been implemented in *Parcel Manager*. Two of them, namely *Oriented Bounding Box (OBB)* and *Straight Skeleton (SS)*, are existing processes but we have improved them to overcome some limitations exposed in the previous section of this article. Another process, called *flag division*, has been developed especially for *Parcel Manager* in order to simulate
345 the densification of already built parcels.

3.1.1. Improved Oriented Bounding Box division

The basic functioning of the *Oriented Bounding Box* division (Vanegas et al., 2012) is described in Figure 2. The pseudo-code of the algorithm is provided in Algorithm 1. It can be summarized as follows. First, an oriented bounding
350 box is generated around the parcel or the zone selected to be reshaped. The bounding box is then split at the half of its longest side. If needed, the box is randomly rotated in order to ensure that the two created parcels have a contact with the road. A ξ coefficient represents the probability to rotate the box. The OBB division is recursively applied to the created parcels until either the area
355 of newly created parcels or the length of their side bordering the road comes under a specific threshold.

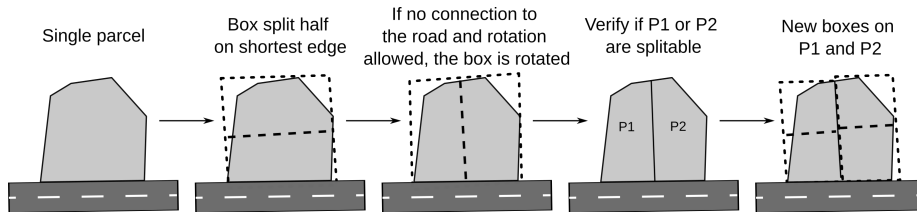


Figure 2: Schema of the OBB parcel division process

This basic version of the algorithm is well adapted to divide medium to small-sized parcels located close to an existing road. However, when parcels are square-shaped or when large parts of the parcels to be reshaped are far from the
360 road, the algorithm is not efficient since it often creates very elongated parcels and allows that some parcels are disconnected from the road. To overcome these limitations, we have introduced a supplementary threshold φ , called 'harmony' threshold, that is compared to the ratio between the longest and the shortest side of the bounding box. When this ratio comes under the φ threshold, the rotation of the box is refused.
365 If φ is set high, reshaped parcels are fairly squared-shaped and can be not connected to a road; if φ is set low, reshaped parcels have a more elongated shape and are imperatively connected to a road. The default value of φ is set to 0.5, which means that, once divided, the parcel can be four times more elongated than wide.

Algorithm 1 Simple recursive Oriented Bounding Box division. *area* computes the area of parcels, *roadContact* computes the length of contact between the parcel and its bordering road, \mathcal{A} is the maximum parcel area, \mathcal{R} is the minimum length of the side of parcels bordering the road, φ is the harmony threshold and ξ is the random box rotation coefficient.

```

function IS_SPLITTABLE(Polygon P)
  return area(P)  $\leq \mathcal{A} \wedge$  (roadContact(P)  $\geq \mathcal{R} \vee$  roadContact(P) = 0)
end function
function SPLIT(Polygon P)
  if !IS_SPLITTABLE(P) then return Set(P)                                 $\triangleright$  Set is a set constructor
  end if
  obb  $\leftarrow$  ORIENTEDBOUNDINGBOX(P)                                     $\triangleright$  standard oriented bounding box
  (cut1, cut2)  $\leftarrow$  DIVIDE_ALONG_OBB(P, obb)

  if !HAS_ACCESS_TO_ROAD(cut1, cut2)  $\wedge$   $\begin{cases} \frac{obb.width}{obb.length} > \varphi & \text{harmony-based version} \\ \text{SAMPLE}(0, 1) \leq \xi & \text{randomised version} \end{cases}$  then
    rotatedObb  $\leftarrow$  ROTATED_ORIENTED_BOUNDING_BOX(P)
    (rotatedCut1, rotatedCut2)  $\leftarrow$  DIVIDE_ALONG_OBB(P, rotatedObb)
    if HAS_ACCESS_TO_ROAD(rotatedCut1, rotatedCut2) then
      (cut1, cut2)  $\leftarrow$  (rotatedCut1, rotatedCut2)
    end if
  end if
  result  $\leftarrow$  SPLIT(cut1)  $\cup$  SPLIT(cut2)                                 $\triangleright \cup$  is the set union
  return result                                                          $\triangleright$  result is a set, possibly typed as a Polygon set
end function

```

370 Road creation is often required when parcels are reshaped in order to ensure
 that all newly created parcels are directly connected to the road network. The
 process of road creation in the OBB algorithm has been firstly introduced by
 Dahal and Chow (2014): a new road is created on the boundary lines of the
 bounding boxes each time a parcel is split. Thereby, when a zone is divided into
 375 numerous newly created parcels, a very high number of roads are created. To
 avoid this, we have modified the algorithm so that the road creation does not
 occur during all steps of the division process. On the contrary, road creation
 can stop *before* the end of the division process, which enables the creation of
 blocks of parcels, i. e. groups of parcels without roads separating them. In order
 380 to generate primary and secondary roads, called respectively *streets* and *lanes*,
 newly created roads can have two different widths set by the user (Figure 3).

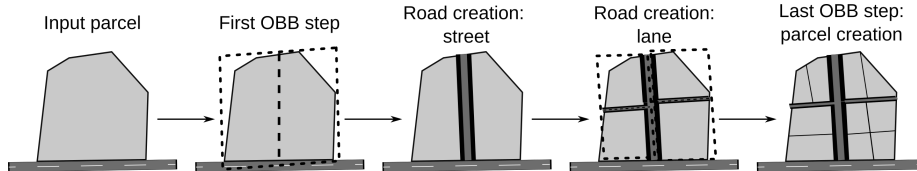


Figure 3: Schema of the new OBB division process with the joint creation of roads

The joint creation of roads, presented in Algorithm 2, begins with the construction of a tree in which each branch corresponds to a split parcel and each leaf is a reshaped parcel obtained at the end of the iterative split (Figure 4).
 385 The depth of each branch of the tree varies according to the number of splits

that has been done for each concerned parcel until the division process stops. Thus branches that correspond to irregularly shaped parts of the initial parcel (parcel $p1$) may have various depth whereas branches corresponding to the core of the parcel to be reshaped (parcel $p0$) usually have the same depth. The algorithm then calculates the median value of the distribution of the depths of the branches of the division tree. The number of iterations until which the road creation occurs is given by the median value of the distribution of the depths of the branches of the division tree *minus* the value of a *block shape* parameter. While this value is higher than the value of a parameter named *street-lane*, streets are created, lanes otherwise (see Figure 4 and examples in Table 2). The value of the *block shape* parameter (integer value) is usually chosen between 0 and 4. The value of the *street-lane* parameter has to be higher than the value of the *block shape* parameter.

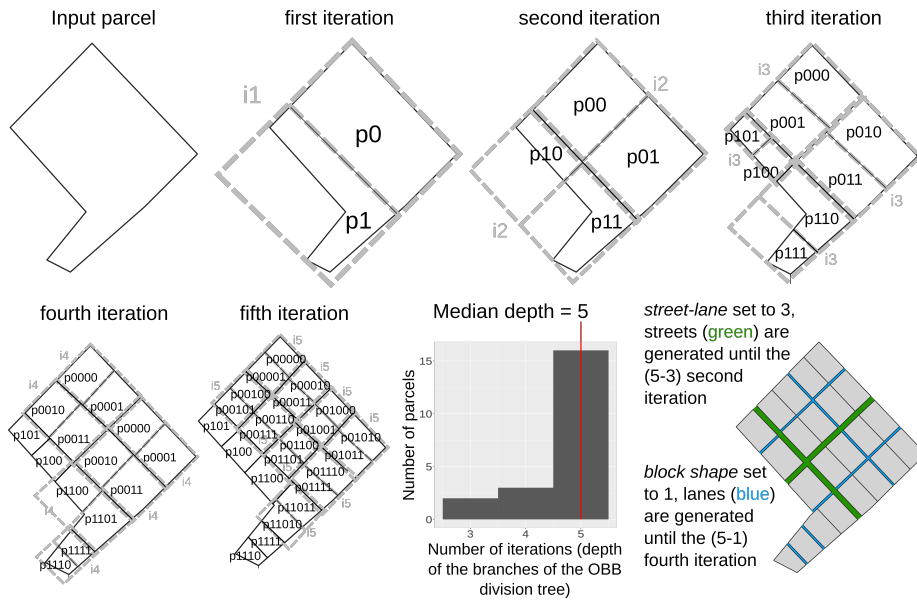


Figure 4: Iteration steps of the OBB parcel division process with the joint creation of roads

3.1.2. Improved Straight Skeleton division

Principles for the creation of a *Straight Skeleton* have been initially introduced by Aichholzer and Aurenhammer (1996). They are illustrated in Figure 5. The creation of a *Straight Skeleton* begins with the iterative construction of waves that translate step by step every edges of the input polygon to its interior. Aligned vertices of the series of waves and nodes of the input polygon are then linked by arcs. The *Straight Skeleton* of the input polygon is the union of all those arcs; polygons formed by the arcs of the skeleton are called *strips*. Several implementations of this method have been proposed (Cheng et al., 2014; Eder et al., 2021). *Parcel Manager* uses the implementation from Kelly and

Algorithm 2 Recursive *Oriented Bounding Box* division with the joint creation of roads. bs is the 'block shape' parameter, sl is the 'street-lane' parameter, lw is the 'lane width' and sw is the 'street width'.

```

function SPLITWROAD(Polygon  $P$ , Int  $bs$ , Int  $sl$ )
   $OBBtree \leftarrow$  SPLIT( $P$ , 0)
   $depths \leftarrow$   $OBBtree.depths[]$   $\triangleright$  Distribution of the depths of the branches of the division tree
   $laneLevelDisabled \leftarrow$  MEDIAN( $depths$ )  $- bs$ 
   $streetLevelDisabled \leftarrow$  MEDIAN( $depths$ )  $- sl$ 
  return SPLIT( $P$ ,  $laneLevelDisabled$ ,  $streetLevelDisabled$ )
end function
 $result \leftarrow \emptyset$   $\triangleright$   $result$  is a list of polygons.

function SPLIT(Tree[Polygon]  $P$ , Int  $laneLevelDisabled$ , Int  $streetLevelDisabled$ )
  if !IS_SPLITTABLE( $P$ ) then  $\triangleright$  is.Splitable: see Algorithm 1
    return  $P$ 
  end if
   $obb \leftarrow$  ORIENTEDBOUNDINGBOX( $P$ )  $\triangleright$  Standard oriented bounding box
  if DEPTH( $P$ )  $\leq$   $streetLevelDisabled$  then
    ( $cut1$ ,  $cut2$ )  $\leftarrow$  DIVIDE_ALONG_OBB_GENERATE_ROAD( $P$ ,  $obb$ ,  $sw$ )
  else if DEPTH( $P$ )  $\leq$   $laneLevelDisabled$  then
    ( $cut1$ ,  $cut2$ )  $\leftarrow$  DIVIDE_ALONG_OBB_GENERATE_ROAD( $P$ ,  $obb$ ,  $lw$ )
  else
    ( $cut1$ ,  $cut2$ )  $\leftarrow$  DIVIDE_ALONG_OBB( $P$ ,  $obb$ )
  end if
  if !HAS_ACCESS_TO_ROAD( $cut1$ ,  $cut2$ )  $\wedge$   $\left\{ \begin{array}{ll} \frac{obb.width}{obb.length} > \varphi & \text{harmony-based version} \\ \text{SAMPLE}(0, 1) \leq \xi & \text{randomised version} \end{array} \right.$  then
     $rotatedObb \leftarrow$  ROTATED_ORIENTED_BOUNDING_BOX( $P$ )
    ( $rotatedCut1$ ,  $rotatedCut2$ )  $\leftarrow$  DIVIDE_ALONG_OBB( $P$ ,  $rotatedObb$ )
    if HAS_ACCESS_TO_ROAD( $rotatedCut1$ ,  $rotatedCut2$ ) then
      ( $cut1$ ,  $cut2$ )  $\leftarrow$  ( $rotatedCut1$ ,  $rotatedCut2$ )
    end if
  end if
   $result \leftarrow$  SPLIT( $cut1$ ,  $laneLevelDisabled$ ,  $streetLevelDisabled$ )  $\cup$ 
  SPLIT( $cut2$ ,  $laneLevelDisabled$ ,  $streetLevelDisabled$ )  $\triangleright$   $\cup$  is the union set
  return  $result$ 
end function

```

Table 2: Role of 'block shape' and 'street-lane' parameters

Targeted type of urban fabric	Medium-sized blocks of flats	Small-sized blocks of flats	Small-sized single-family houses
Value of the 'block shape' parameter	0	2	3
Value of the 'street-lane' parameter	2	4	6
Illustration			

Wonka (2011), which has been adapted for modelling the division of parcels by Vanegas et al. (2012).

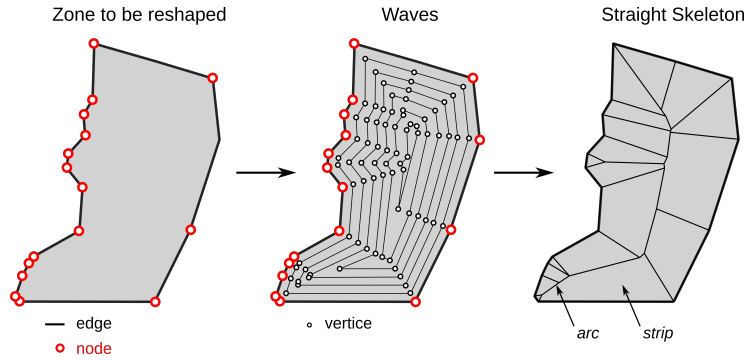


Figure 5: Creation of a Straight Skeleton

410

Following the creation of the *Straight Skeleton*, strips are modified in order to obtain more realistic parcel shapes (Figure 6). α -strips are first generated by merging all input strips that are contiguous and facing the same road. Roads are taken into account within a more or less large neighbourhood, whose size is determined by the parameter *distance to the nearest road*. An attribute of the roads indicates their *level of attraction* (1 being the lowest level). If an input strip is facing several roads, the *level of attraction* of each road is considered in order to choose which road would be considered as being the frontage of the

415

strip³. β -strips are then created by rotating the diagonal edges of the α -strips
 420 that have a triangular shape. A diagonal edge corresponds to a side edge of a
 strip that is not perpendicular to its frontage road. Diagonal edges are rotated
 in order to become perpendicular to their frontage road⁴. Parcels are finally
 obtained by splitting the β -strips perpendicularly to their main road frontage.
 The width of the parcels, i.e. the length of the contact between each parcel
 425 and its road frontage, is randomly chosen from a normal distribution centred
 on $\frac{\text{maximalWidth} - \text{minimalWidth}}{2}$ and a standard deviation parameter σ . β -strips
 that are not linked to a frontage road are not split.

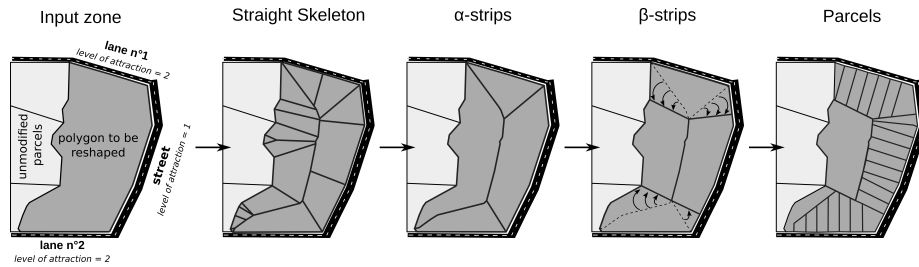


Figure 6: Parcel division process based on a Straight Skeleton

Two options have been added to this algorithm specially for *Parcel Manager*.
 First, as suggested by Vanegas et al. (2012), we have added an *Offset* division
 430 function that enables the creation of a patio in the centre of a block of reshaped
 parcels: during the creation of the *Straight Skeleton*, the drawing of the arcs on
 the waves stops when they reach a predefined length set by a *maximum depth*
 parameter. This lets an empty wave inside the input polygon (see Figure 7).
 Second, as done by Dahal and Chow (2014) for the *street grid* method, we have
 435 added the possibility to generate peripheral roads on the perimeter of each zone
 under consideration. So as to avoid the merging of all α -strips facing the same
 peripheral road, several road segments are created but not one unique peripheral
 road (see pseudocode 4).

3.1.3. Flag division

440 The principle of the *flag division* is to add driveways to the divided parcels to
 ensure that all created parcels are connected to the road network (see Figure 8).
 Usually, the *flag division* is used to divide built parcels but it also applies to
 the division of non built parcels. The principle is as follows: if the divided parts
 of a reshaped parcel do not touch the road, a driveway is created through one
 445 side of the parcel that is connected to a road. It is then possible to create a
 new unbuilt parcel on the back parcel (*flag densification*) or on the front parcel

³Algorithm 3 describes the case where only one frontage is detected.

⁴As detailed in Algorithm 5, the edge is projected perpendicularly to the border edge rather than really rotated.

Algorithm 3 Straight Skeleton algorithm. It requires, as argument, a list of roads composed of road segments with the attributes *name* (optional) and *level of attraction* (mandatory). For details on the implementation of *StraightSkeletonGeneration*, see the pseudo-code in Kelly and Wonka (2011).

```

function STRAIGHTSKELETONDIVISION(Polygon  $P$ , List(Road)  $lR$ , real  $distanceToNearestRoad$ ,
float  $\sigma$ , real  $minWidth$ , real  $maxWidth$ )
  if GeneratePeripheralRoad then
     $P \leftarrow$  GENERATEPERIPHERALROAD( $P$ ,  $laneWidth$ )
  end if
   $frontages \leftarrow$  FRONTAGEDEFINITION( $P$ ,  $lR$ ,  $distanceToNearestRoad$ )
   $P_{strips} \leftarrow$  STRAIGHTSKELETONGENERATION( $P$ ,  $maximumDepth$ )
   $\alpha$ -strips  $\leftarrow$  MERGEINTOALPHASTRIP( $P_{strips}$ ,  $frontages$ )
   $\beta$ -strips  $\leftarrow$  MERGEINTOBETASTRIPS( $\alpha$ -strips,  $frontages$ )
   $parcel \leftarrow$  CUTSTRIPSINPARCEL( $\beta$ -strips,  $minimalWidth$ ,  $maximalWidth$ ,  $\sigma$ )
  return  $parcel$ 
end function

```

Algorithm 4 The *GeneratePeripheralRoad* algorithm generates new road segments around a given polygon (parcel). Three threshold values have to be set. *angleThreshold* is the angle between two segments above which the two segments belong to two different roads. A supplementary condition sets that a segment must be longer than a *minLength* threshold value to be considered as a road. Finally, a segment longer than a *maxLength* threshold value is divided even if no angles are found (which can be the case of circle-shaped parcels). The default values of the three thresholds are respectively $\frac{\pi}{3}$, $2m$ and $\frac{exteriorEdges.length}{2}$.

```

function GENERATEPERIPHERALROAD(Polygon  $P$ , Roads  $existingRoads$ , real  $laneWidth$ , real
 $angleThreshold$ )
   $P_{reduced} \leftarrow$  BUFFER( $P$ ,  $-laneWidth$ )
   $exteriorEdges \leftarrow$  GETEXTERIOREDGES( $P$ ) ▷ outer edges of P in a linked list
   $peripheralRoads \leftarrow$  List()
   $roadSegment \leftarrow$  List()
   $cumulateLength = 0$ 
  for all  $e : exteriorEdges$  do
     $roadSegment.add(e)$ 
     $cumulateLength += e.length$ 
    if ( $ANGLE(e, e.next()) \geq angleThreshold \wedge cumulateLength \geq minLength$ )  $\vee$ 
 $cumulateLength \geq maxLength$  then
       $peripheralRoad.add(roadSegment)$ 
       $roadSegment \leftarrow$  List()
       $cumulateLength = 0$ 
    end if
  end for
   $peripheralRoad \leftarrow$  LINEBUFFER( $peripheralRoads$ ,  $laneWidth/2$ )
  return  $\{P_{reduced}, peripheralRoads\}$ 
end function

```

Algorithm 5 The *FrontageDefinition* algorithm creates a collection of pairs associating each exterior edge with its adjacent road segment.

```

function FRONTAGEDEFINITION(Polygon  $P$ , List(Road)  $lR$ , real  $distanceToNearestRoad$ )
   $frontages \leftarrow \text{Map} \langle \text{Edge}, \text{Road} \rangle$ 
   $exteriorEdges \leftarrow \text{GETEXTERIOREDGES}(P)$  ▷ outer edges of  $P$  in a linked list
  for all  $edge : exteriorEdges$  do
     $r \leftarrow \text{GETFACINGROAD}(edge, lR, distanceToNearestRoad)$ 
    if  $r \neq \emptyset$  then
       $frontages.add(edge, r)$ 
    end if
  end for
  return  $frontages$ 
end function

```

Algorithm 6 The *MergeIntoAlphaStrip* algorithm builds the α -strips by merging the strips connected to the same road segment.

```

function MERGEINTOALPHASTRIP(PolygonSet  $P_{split}$ , Map< Edge, Road >  $frontages$ )
   $\alpha\text{-strips}_{final} \leftarrow \text{List}()$ 
   $roads \leftarrow \text{GETDISTINCTVALUES}(frontages)$ 
  for Road  $r : roads$  do
     $roadFacingEdges \leftarrow frontages.GETKEYSWITHVALUE(r)$ 
     $strips \leftarrow \text{FILTERCONTAINING}(P_{split}, roadFacingEdges)$ 
     $\alpha\text{-strip}_{current} \leftarrow \text{UNION}(strips)$ 
     $\alpha\text{-strips}_{final}.add(\alpha\text{-strip}_{current})$ 
  end for
  return  $\alpha\text{-strips}_{final}$ 
end function

```

Algorithm 7 The *MergeIntoBetaStrips* algorithm fixes the diagonal edges between α -strips and creates β -strips (see Figure 6).

```

function MERGEINTOBETASTRIPS(PolygonSet  $\alpha\text{-strip}$ , Map< Edge, Road >  $frontages$ )
   $exteriorEdges \leftarrow \text{GETEXTERIOREDGES}(\alpha\text{-strip})$ 
   $levelsAttraction \leftarrow \text{Map} \langle \text{Edge}, \text{int} \rangle$ 
  for all  $e : exteriorEdges$  do
    if  $frontages.contains(e)$  then
       $levelsAttraction.add(e, frontages.get(e).getLevelOfAttraction)$ 
    else
       $levelsAttraction.add(e, -\infty)$ 
    end if
  end for
   $\beta\text{-strip} \leftarrow \text{COPY}(\alpha\text{-strip})$ 
   $N_{ext} \leftarrow exteriorEdges.getNodes()$ 
   $N_{\alpha\text{-strip}} \leftarrow \alpha\text{-strip}.getNodes()$  ▷ Vertices of the  $\alpha$ -strip PolygonSet
   $N_{inside} \leftarrow N_{\alpha\text{-strip}} \setminus N_{ext}$ 
   $N_{\alpha\text{-strip-ext}} \leftarrow N_{\alpha\text{-strip}} \setminus N_{inside}$ 
  for all  $n : N_{\alpha\text{-strip-ext}}$  do
     $insideNeighbour \leftarrow \text{GETCLOSEST}(n, N_{inside})$ 
     $adjacentEdges \leftarrow \text{EDGESCONTAINING}(n, exteriorEdges)$ 
     $chosenEdge \leftarrow \text{MINIMUMLEVELATTRACTION}(adjacentEdges, levelsAttraction)$ 
     $newNode \leftarrow \text{ORTHOGONALPROJECT}(insideNeighbour, chosenEdge)$ 
     $\beta\text{-strip}.removeEdge(n, insideNeighbour)$ 
     $\beta\text{-strip}.addEdge(insideNeighbour, newNode)$ 
  end for
  return  $\beta\text{-strip}$ 
end function

```

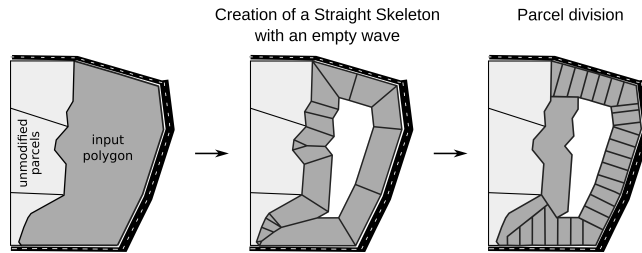


Figure 7: Straight Skeleton division process with the *offset* method

(*porch densification*). Now, if all divided parcels touch a road, no driveways are created.

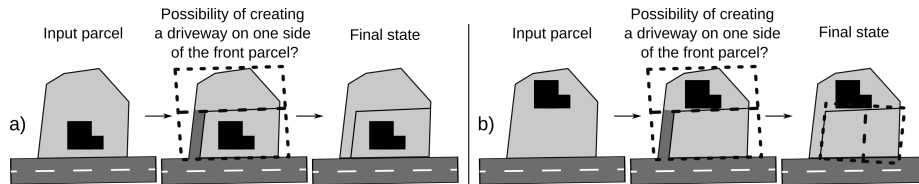


Figure 8: Schema of the *flag division* process: a) flag densification b) porch densification

450 The pseudo-code of algorithm 9 describes how this reshaping process is mod-
 elled in *Parcel Manager*. The *flag division* algorithm splits the input parcel with
 the *OBB* method and simulates the construction of a driveway on one of the
 side of the parcel having a frontage on an existing road. The creation of the
 driveway is first tested along the shortest side of the parcel. If the driveway
 intersects an existing building, the algorithm then tests the construction of the
 455 driveway along the other side of the parcel. If, finally, one newly created parcel
 can not have access to the road network, the flag division of the input parcel is
 considered to be impossible.

Algorithm 8 The *CutStripsInParcel* algorithm splits a β -strip into parcels. $minWidth$ and $maxWidth$ are the upper and lower bounds of the length of the contact between parcels and roads. σ determines the level of diversity (more or less high or low) of the parcel widths.

```

function CUTSTRIPSINPARCEL(PolygonSet  $\beta$ -strip, Map < edge, Road > frontages, float  $\sigma$ , real
   $minWidth$ , real  $maxWidth$ )
  for all f : frontages do
     $frontageEdges \leftarrow f.getEdges()$ 
     $remainder \leftarrow frontageEdges.Length().sum()$ 
     $widths \leftarrow RANDOMWIDTHS(minWidth, maxWidth, \sigma)$   $\triangleright$  Widths drawn from a normal
distribution
     $exteriorEdges \leftarrow GETEXTERIOREDGES(\beta - strip)$ 
     $N_{origin} \leftarrow GETEXTREMITYNODE(frontageEdges)$   $\triangleright$  first node of a frontage edges
     $N_{current} \leftarrow DISTANTNODEALONGEDGES(N_{origin}, widths.next(), frontageEdges)$   $\triangleright$  draws
a Point on a segment at a given distance.
     $currentEdge \leftarrow EDGECONTAINS(frontageEdges, N_{current})$ 
    while  $remainder \geq minWidth$  do
       $normalLine \leftarrow PERPENDICULARLINE(currentEdge, N_{current})$ 
       $N_{proj} \leftarrow INTERSECTS(normalLine, \beta - strip.getEdges() \setminus exteriorEdges)$ 
      if  $N_{proj} \neq \emptyset \wedge remainder \geq minWidth$  then
         $\beta - strip.addEdge(N_{current}, N_{proj})$ 
         $N_{current} \leftarrow DISTANTNODEALONGEDGES(N_{current}, widths.next(), frontageEdges)$ 
         $remainder \leftarrow remainder - distance(N_{current}, N_{proj})$ 
      end if
    end while
     $parcels \leftarrow CONSTRUCTMINIMALPOLYGONSFROMEDGES(\beta - strip)$   $\triangleright$  post-treatment
function creating polygons, whose union forms the  $\beta$ -strip
  end while
end for
return  $parcels$ 
end function

```

Algorithm 9 Flag division algorithm.

Function `GETPARCELSCLOSESTTOROAD`(*Polygon* $P_{toDensify}$, *List*(*Polygon*) P_s) selects surrounding parcels P_s touching $P_{toDensify}$ and the road network. They are then sorted by the distance between the boundaries of $P_{toDensify}$ and the road network across P_s , the first P_s representing the shortest distance. Function `ISLASTELEMENT`($P_{crossed}$) return true if P is the last element analysed, which means it has been impossible to generate a driveway through the parcel.

```

function FLAGDIVISION(Polygon  $P_{ini}$ , List(Polygon)  $P_s$ , List(Building)  $L_{building}$ )
  if !IS_SPLITTABLE( $P$ ) then return Set( $P$ )                                ▷ is.Splittable: see Algorithm 1
  end if
   $P_{result}$  ▷ contains both parcels with flag division, parcels crossed and surrounding parcels
   $obb \leftarrow$  ORIENTEDBOUNDINGBOX( $P$ )                                ▷ standard oriented bounding box
  ( $cut1, cut2$ )  $\leftarrow$  DIVIDE_ALONG_OBB( $P, obb$ )                                ▷ see section 3.1.1
  for all  $cut \in cut1, cut2$  do
    if HASNOACCESSTOROAD( $cut$ ) then
      for all  $P_c \in$  GETPARCELSCLOSESTTOROAD( $cut, P_s$ ) do
         $driveway \leftarrow$  CREATEDRIVEWAY( $P_c$ )
        if !(INTERSECTION( $driveway, L_{building}$ )  $\vee$   $driveway = \emptyset$ ) then
           $crossedParcel \leftarrow$  CREATEDRIVEWAYTHROUGHPARCEL( $driveway, P_c$ )
           $flagParcel \leftarrow$  ADDDRIVEWAYTOPARCEL( $cut$ )
          if HASACCESSTOROAD( $crossedParcel$ )  $\wedge$  HASACCESSTOROAD( $flagParcel$ ) then
             $P_{result}.add(crossedParcel, flagParcel)$ 
          else if ISLASTELEMENT( $P_c$ ) then
            return  $P_{result}.add(P_{ini})$  ▷ the initial parcel is returned and recursion stops
          end if
        end if
      end for
    else
      return Set(FLAGDIVISION( $cut, P_{result}, L_{building}$ )) ▷ If the cut part has a road access,
      we recursive apply the flag cut algorithm to this part
    end if
  end for
  return  $P_{result}$ 
end function

```

3.2. Parcel Manager workflows

In order to simulate complete parcel reshaping processes, workflows in *Parcel*
460 *Manager* combine a chosen parcel division process with one or several other
tasks: parcel selection, parcel merging, and attribute changes of parcels. For
example, if a created parcel is under the minimal parcel area set by the user,
the rule set in a workflow can be either to merge this parcel with the contiguous
parcel which shares its longest edge or to delete this parcel and transform it
465 into a public space.

Three predefined workflows have been implemented knowing that any other
workflow can be created by users.

Parcel densification: this workflow divides parcels while ensuring that they
can be densified. The *flag division* process is first applied to the selected
470 parcels. Tests are then made to ensure that new buildings can be con-
structed within the divided parcels. In particular, the area of each re-
shaped parcel must be above an area threshold otherwise the division is
cancelled. Another condition is that one of the reshaped parcel is unbuilt
to ensure that at least one new building can be constructed. The user can
475 possibly choose to apply a negative buffer around the existing buildings to
avoid data imprecision and to consider only buildings whose area is upper
than a given threshold and thus eliminate from the analysis buildings that
are too small. By default, the data imprecision is set to 1 meter and the
area threshold is set to $20m^2$. If the road access of reshaped parcels is not
480 mandatory⁵, a simple OBB division is operated instead of a *flag division*,
which possibly creates new parcels without any road access.

Zone consolidation: this workflow begins with merging together contiguous
parcels that have been selected to create new urban development zones.
A parcel division process chosen by the user, including the creation of
485 new roads, is then applied to each newly created zone. After the division,
the parcels whose area comes under a given threshold are merged to the
closest parcels with which they share the longest edge, or deleted if they
are isolated.

Zone division: this workflow operates on a zone containing or intersecting
490 several parcels rather than on isolated parcels. Each zone under consider-
ation is usually taken from a zoning plan. First, the parcels that intersect
the zone under consideration are cut and their parts being outside the
zone are set aside. A parcel division process chosen by the user, includ-
ing the creation of new roads, is then applied on all parcels of the zone.
495 The existing roads crossing the zone can optionally be kept in their ini-
tial state. After the division, the parcels whose area comes under a given
threshold are merged with the closest parcels with which they share the
longest edge, or deleted if they are isolated.

⁵In real cases, this can occur in case of an agreement between parcel owners.

500 The creation or reshaping of parcels implies the modification of the attribute table in accordance with the chosen nomenclature. The way the attributes of parcels are added or modified differs according to the workflow (i.e. the densification of a parcel only involves a change of the parcel attribute *number*, while the creation of a whole set of parcels involves the addition of a new *section* name as well as new parcel *numbers*).

505 A workflow is parametrized in view of the simulated development of a unique type of urban fabric. The application of a workflow thus requires to choose the type of urban fabric planned to be built. Table 3 displays the urban fabric parameter profiles included by default in *Parcel Manager*. Two sizes of collective housing units are distinguished that correspond to medium-sized and small-sized blocks of flats (circa 30 or 5 housing units respectively). Two sizes of individual housing units are distinguished that correspond to either villas (medium-sized single-family houses) or housing estate homes (small-sized single-family houses). Users have the possibility to create new urban fabric profiles or to modify the default profiles.

515 The choice of the type of urban fabric planned to be built in each place must often account for the existing urban environment (buildings, roads...). To this end, we have developed a *GetParametersOfScene* package in *Parcel Manager*, which helps users to set the values of morphological parameters for different places of the area under study. This package provides basic information about chosen places: distribution of parcel sizes, attributes of the roads, ratio of the total area of roads on the total surface, etc. Figure 9 shows an excerpt of the graphs obtained with the *GetParametersOfScene* package for a small community located in the east of France (Gennevilliers). The package can also be used for analysing a set of communities, a set of contiguous parcels, or different zones of a zoning plan.

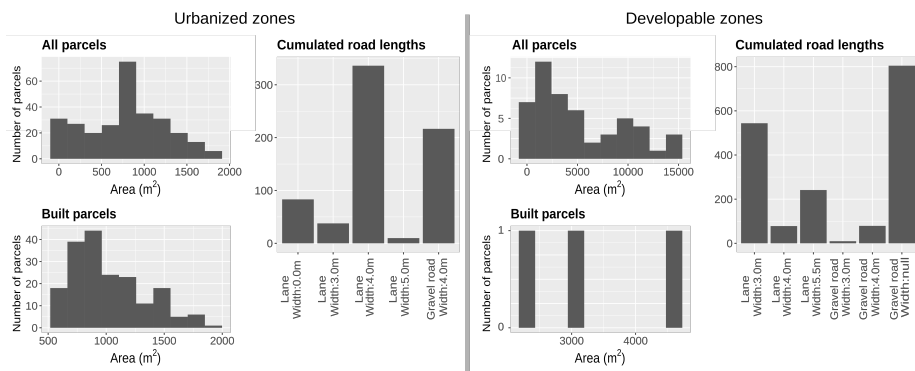


Figure 9: Analysis of two types of zones of the parcel plan of the community of Gennevilliers (France) with the *GetParametersOfScene* package.

525

Table 3: Morphological parameters (i.e. urban fabric profiles) and their default values in *Parcel Manager*

Oriented Bounding Box Process	Medium-sized blocks of flats	Small blocks of flats	Medium-sized single-family houses	Small-sized single-family houses
Maximal parcel area (m^2)	6 000	2 000	1000	550
Minimal parcel area (m^2)	800	400	300	175
Minimum width of contact between road and parcel (m)	15	12	7	7
Value of 'block shape' parameter	2	4	6	6
Value of 'street-lane' parameter	0	2	3	3
Width of streets (m)	20	15	15	15
Width of lanes (m)	12	9	7	7
Flag Division Process		Small blocks of flats	Medium-sized single-family houses	Small-sized single-family houses
Maximal parcel area (m^2)		2 000	1 000	550
Minimal parcel area (m^2)		400	300	175
Minimal width of contact between road and parcel (m)		7	3	3
Width of driveways (m)		7	3	3
Straight Skeleton Division Process	Medium-sized blocks of flats	Small blocks of flats	Medium-sized single-family houses	Small-sized single-family houses
Minimal parcel area (m^2)	800	400	300	175
Width of peripheral roads (m)	12	9	7	7
Maximum depth of parcels	200	100	70	40
Distance to the nearest road	50	30	10	10
Minimum width of contact between parcel and road (m)	15	12	7	7
Maximum width of contact between parcel and road (m)	100	80	50	30

3.3. Scenarios with Parcel Manager

Parcel Manager allows the creation of scenarios in order to run fully automated simulations at a regional scale. Scenarios are composed of a queue of steps. Each step applies a *workflow* on a specific subset of parcels of the region under study. The ordered list of steps is set in a .json file.

In *Parcel Manager*, parcels are selected for a simulation by marking a special boolean attribute called *split*. Parcels can be marked regarding their shape (in particular, their size), or regarding the geometrical relations they have with other objects such as buildings (in order to reshape only unbuilt parcels) and roads (to reshape only the parcels connected to a road), or regarding any exogenous polygons (e.g. zones identified as being worth urbanizing in a zoning plan, communities of a given type, communities located at a given distance from the city centre).

4. Test of each parcel division algorithm

A scenario called *test scenario* has been designed specially to test the three parcel division processes integrated in *Parcel Manager*, namely *Oriented Bounding Box (OBB)*, *Straight Skeleton (SS)* and *flag division*. This *test scenario* can be run directly from the graphical user interface via a dedicated button called *Use cases*. Here we have applied this *test scenario* to the community of Gennes (681 inhabitants) in the east of France. Input data and simulation results can be found on a data server (<https://tinyurl.com/yj8zd8na>). The zoning plan has been specially designed for testing the parcel division algorithms of *Parcel Manager*. Simulation objectives have also been chosen for this test and have nothing to do with the real urban development project of the Gennes's community.

The *test scenario* involves three workflows:

- The workflow *Zone Division* is applied to the AU1 zones (in purple colour on Figure 10) where *Medium-sized blocks of flats* are planned to be built.
- The workflow *Zone Consolidation* is applied to the AU2 zones (in pink colour on Figure 10) where *Medium-sized single-family houses* are planned to be built.
- The workflow *Parcel Densification* is applied to the zones to be densified (in light orange colour on Figure 10) where *Small-sized single-family houses* are planned to be built.

Values of the morphological parameters are taken from Table 3.

4.1. Simulation of the test scenario with the Oriented Bounding Box algorithm. Workflows *Zone Division* and *Zone Consolidation*.

Figure 11 displays the simulation results obtained using the Oriented Bounding Box (OBB) parcel division process improved specially for *Parcel Manager*



Figure 10: Zoning plan and parcel layout of the community of Gennes, east of France.

565 with the joint creation of new roads. The simulated parcel layout is visually
 realistic. 15 parcels have been generated in the AU1 zones by the *Zone Division*
 workflow and 397 parcels have been generated in the AU2 zones by the *Zone*
Consolidation workflow. This difference is logical since the total area of AU1
 zones is much lower than the total area of AU2 zones and the targeted area of
 570 reshaped parcels is larger in the AU1 zones (*medium-sized blocks of flats*) than
 in the AU2 zones (*medium-sized houses*). Every reshaped parcels strictly conform
 with the minimum and maximum area thresholds set in their *urban fabric*
profile (from 300 to 1,000 square meters for *medium-sized houses* and from 800
 to 6,000 square meters for *medium-sized blocks of flats*, see Table 3) (Figure 12).

575 In some cases, roads created by simulation are perfectly linked to the pre-
 existing roads (labels *a* and *b* on Figure 11). In other cases, the linking is not so
 perfect. Figure 13 presents the proportion of road area regarding the total area
 of each reshaping operation. A reshaping operation corresponds to a single zone
 in case of the *Zone division* workflow and a contiguous set of merged parcels
 580 in case of the *Zone consolidation* workflow. Roads under consideration are pre-
 existing roads as well as streets and lanes created by simulation. The ratio of
 road area regarding the total area of a zone ranges usually from 10% to 25%:



Figure 11: Simulation of the *test scenario* on the community of Gennes with the Oriented Bounding Box algorithm. Workflows *Zone Division* in the AU1 zones and *Zone Consolidation* in the AU2 zones.

10% to 30% in cities in the USA (Meyer and Gómez-Ibáñez, 2013), 20% to 25% in occidental capital cities (New York, London, Tokyo, Paris) (Litman, 2020),
 585 5% to 20% in capitals of developing countries (Kolkata, Shanghai, Bangkok, Seoul, Delhi, Sao Polo) (Vasconcellos, 2001), or 7% to 12% in middle-sized German cities (Gössling et al., 2016). Simulation results mostly conform with those values. In the case of reshaping parcels for the construction of medium-sized blocks of flats with the *Zone Division* workflow, the values of the ratio
 590 are in average higher than in the case of reshaping parcels for the construction of medium-sized single-family houses with the *Zone Consolidation* workflow. Besides, the area of roads generated by simulation is generally all the more high given that a reshaping operation is large.

Only four parcels do not have a contact with a road upon a total of 397
 595 reshaped parcels (see Figure 11). This is very satisfying especially since a high number of new roads have been created by simulation. The four parcels without any connection to the road could easily be merged with adjacent parcels, if needed.

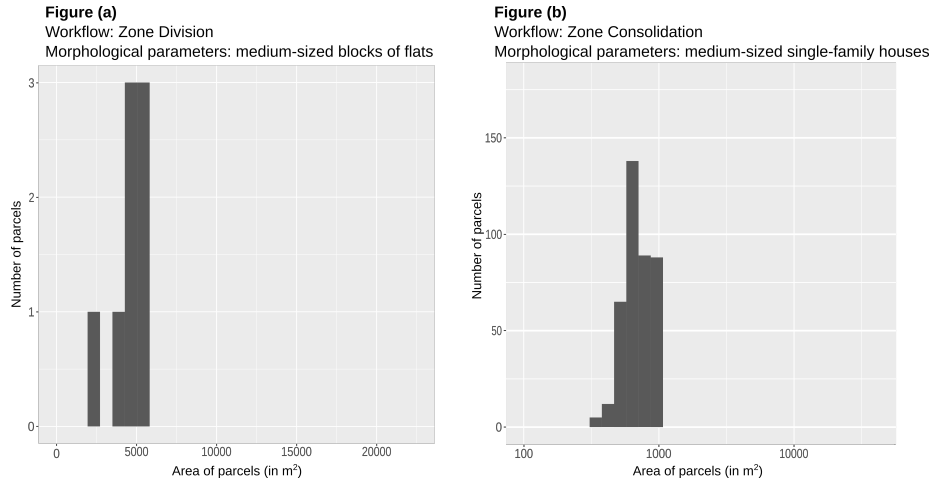


Figure 12: Community of Genes, east of France. Area of parcels reshaped by simulation using the *OBB* division algorithm with the joint creation of new roads. Workflow *Zone Division* applied on the AU1 zones; workflow *Zone Consolidation* applied on the AU2 zones. NB: scales of x-axis and y-axis of figures a and b differ crucially.

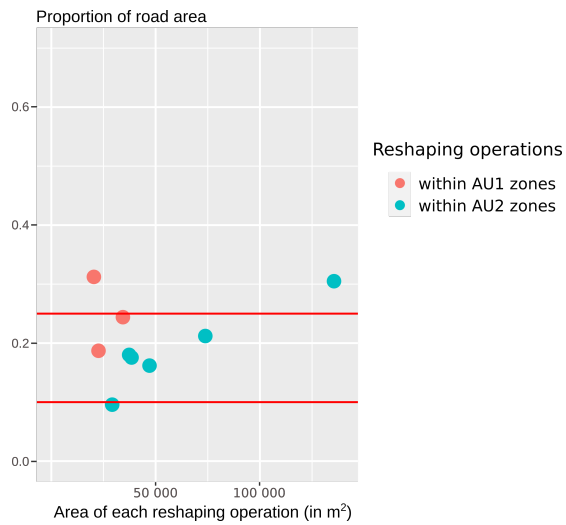


Figure 13: *OBB* parcel division algorithm with the joint creation of new roads: ratio of the area of roads on the total area of each reshaping operation.

600 4.2. *Simulation of the test scenario with the Straight Skeleton algorithm. Workflows Zone Division and Zone Consolidation.*

The *test scenario* has been simulated using three different versions of the *Straight Skeleton* algorithm developed specially for *Parcel Manager*.

605 With version (a) of the *Straight Skeleton* algorithm, a regular *Straight Skeleton* division is applied without the creation of peripheral roads. As the zones selected to be reshaped are often far from existing roads (see labels *a*, *b* and *c* of Figure 14a), the shape of reshaped parcels is most often not realistic.

610 Version (b) of the algorithm generates peripheral roads around each input zone. Figure 14b shows that this method generates unrealistic parcel layouts in the largest zones, in particular the zones located in the south of the area under study (labels *a* and *b*). In smaller zones, in particular the AU1 zones where the creation of *medium-sized blocks of flats* is targeted, the simulated parcel layouts are much more realistic.

615 Version (c) of the algorithm simulates the *offset* division of the parcels, via the introduction of the *maximal depth* parameter, and the joint creation of peripheral roads. It allows the creation of patios, which produces alternative and realistic forms of parcels (see in particular the zone labelled *a* on Figure 14c). Logically, in the AU1 zones where the *Zone division* workflow is applied, simulated parcel layouts are identical to the layouts simulated with versions (b) and (c) of the algorithm. In the largest zones (AU2 zones) where the *Zone consolidation* workflow is applied, very large patios are created.

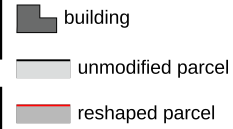
625 Figures 15 and 16 display the area of parcels reshaped with versions *a*, *b* and *c* of the *Straight Skeleton* algorithm. In case of the *zone division* workflow, with versions *b* and *c* of the *Straight Skeleton* algorithm, the number of reshaped parcels and their area are almost identical to the number and area of parcels simulated with the *OBB* algorithm but their shapes differ greatly. With version *a*, however, the number of reshaped parcels is lower and three parcels do not conform with the maximum area threshold set in the *urban fabric profile* (6,000 square meters). In contrast, in case of the *zone consolidation* workflow, the number of reshaped parcels and their area are clearly different from the number and the area of parcels simulated with the *OBB* algorithm. Besides, with version *a* and *b* of the *Straight Skeleton* algorithm, the area of reshaped parcels is often above the *maximum parcel area* parameter value (1000 m) set in the *urban fabric profile* (respectively 114 and 161 parcels are concerned). With version *c* of the algorithm, only 14 reshaped parcels have an area above the *maximum parcel area* parameter value.

630 With versions *b* and *c* of the *Straight Skeleton* algorithm, all reshaped parcels have a contact with a road as this condition is mandatory for the creation of α -strips. Figure 17 displays the ratio of the area of roads calculated for each simulated reshaping operation. This ratio conforms with the expected values in case of the *Zone Consolidation* workflow but roads created with the *Zone Division* workflow seem to be slightly too large.

a) Regular Straight Skeleton algorithm



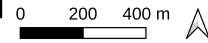
Workflow	Number of reshaped parcels
zone division	9
zone consolidation	140



b) Straight Skeleton algorithm with the creation of peripheral roads



Workflow	Number of reshaped parcels
zone division	13
zone consolidation	216



c) Straight Skeleton offset algorithm with the creation of peripheral roads



Workflow	Number of reshaped parcels
zone division	13
zone consolidation	241

Figure 14: Simulation of the *test scenario* on the community of Gennes with different versions of the *Straight Skeleton* algorithm. Workflows *Zone Division* in the AU1 zones and *Zone Consolidation* in the AU2 zones.

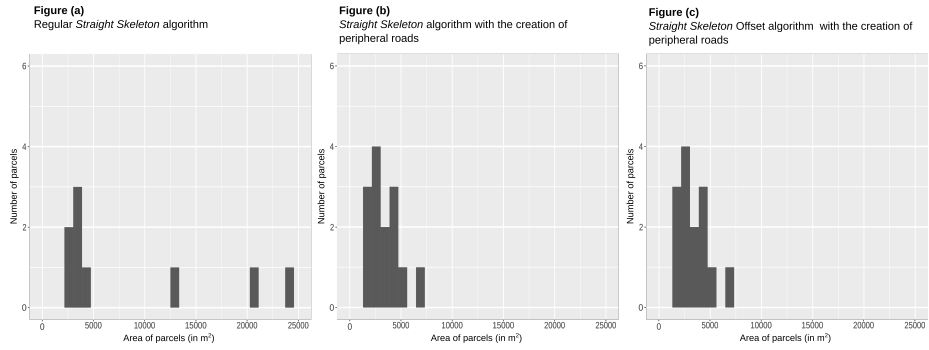


Figure 15: Area of parcels reshaped by simulation with the *Straight Skeleton* algorithm in the AU1 zones of the community of Genes (workflow *Zone Division*).

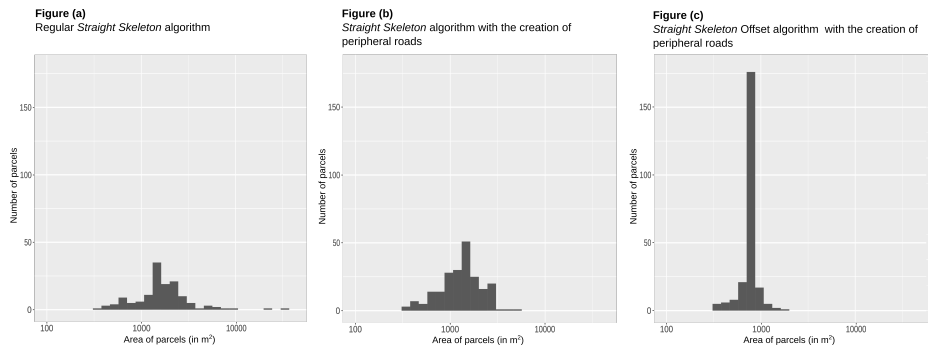


Figure 16: Area of parcels reshaped by simulation with the *Straight Skeleton* algorithm in the AU2 zones of the community of Genes (workflow *Zone Consolidation*).

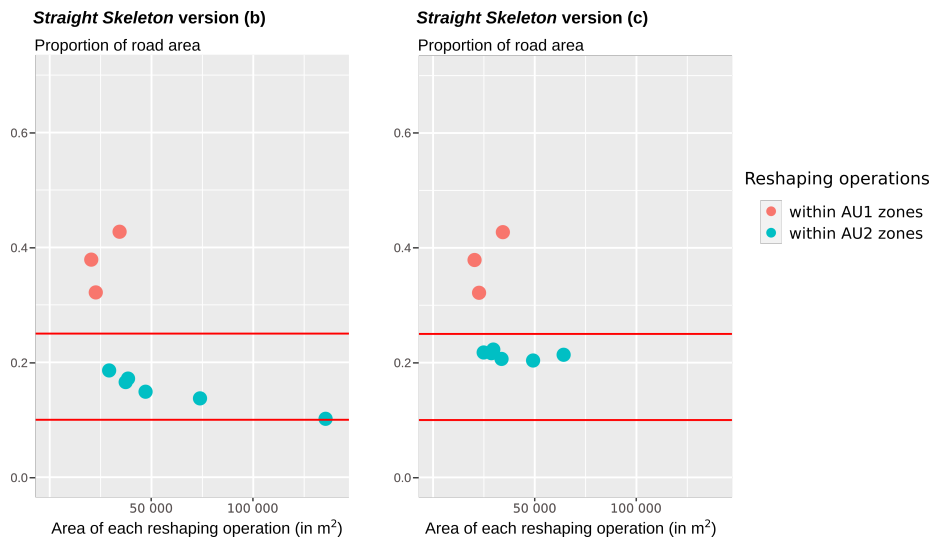


Figure 17: *SS* parcel division algorithm, version *b* (regular division with the creation of peripheral roads) and *c* (offset division with the creation of peripheral roads): ratio of the area of roads on the total area of each reshaping operation. Note: the area of the patios created by simulation with the offset algorithm is not taken into account in the calculation of the ratio.

4.3. Simulation of the Parcel Densification workflow with the associated flag division algorithm

Simulation results obtained with the *Parcel Densification* workflow and the
645 associated *flag division* process applied to the zones to be densified of the com-
munity of Gennes are presented on Figure 18. 81 parcels have been reshaped
by simulation, creating 225 new parcels. Porch densification (see for instance
label *a* on Figure 18) produces smaller parcels than flag densification (label *b*).
Some large parcels have not been densified because of the presence of a building
650 exactly at the centre of those parcels (label *c*) or because they already have a
flag shape (label *d* on Figure 18).

Figure 19 displays the area of the densified parcels. The maximal area of the
parcels when *small-sized houses* are planned to be built (550 m^2 , see Table 3) is
often exceeded. Indeed, with the *Parcel Densification* workflow the division of a
655 parcel is not allowed if the divided parcels are smaller than the *maximal parcel*
area parameter. Moreover, when the connection to a road of a divided parcel is
either lost or impossible, the division of the parent parcel is not allowed even if
the parent parcel is larger than the maximal limit size.

As road connection is mandatory in the *flag division* process, all reshaped
660 parcels are connected to the road network.



Figure 18: Simulation results obtained with the *Parcel Densification* workflow; small-sized houses planned to be built. Focus on the centre of the community of Gennes.

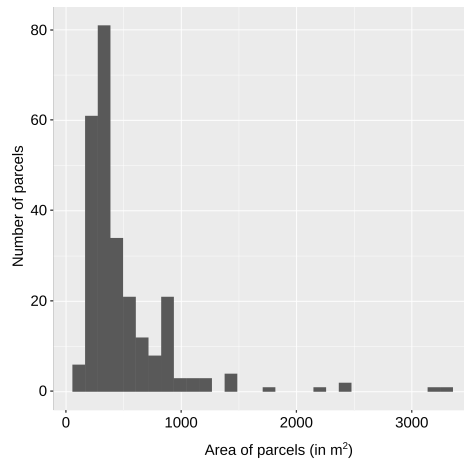


Figure 19: Community of Gennes, east of France. Area of parcels reshaped by simulation with the *Parcel Densification* workflow; small-sized houses planned to be built.

4.4. Assessment of each parcel division algorithm

The topology of the reshaped parcels is correct (reshaped parcels do not overlap) and their geometry is valid.

The *OBB* division algorithm with the joint creation of roads enables the simulation of visually realistic parcel layouts. The area of roads created by simulation may be too large in large zones or when large buildings are planned to be built (e.g. medium-sized blocks of flats) but it is possible to generate thinner roads by modifying the urban fabric profiles (i.e. the values of morphological parameters, see Table 3). The *harmony threshold* introduced in the algorithm enables to efficiently connect almost every reshaped parcels to the road network. In some cases, roads created by simulation are perfectly linked to the pre-existing roads but in other cases, the linking is not perfect. As such, the road creation process should be improved by introducing a mandatory connection to the existing road network. Additionally, the strong regularity of the simulated parcel layouts could be criticized. It would be easy to introduce a kind of irregularity in the *OBB* division algorithm by slightly translating randomly the split line of each divided parcel. Nevertheless, introducing randomness in the parcel division process seems not relevant regarding the objectives of *Parcel Manager*.

In case of the *Straight Skeleton* algorithm, version *a* produces non realistic parcel layouts. Versions *b* (*SS* division with the joint creation of peripheral roads) and *c* (*offset* division with the joint creation of peripheral roads) produce parcel plans that are visually realistic when the parcels to be reshaped are rather small or have an elongated shape. In case of square-shaped medium- or large-sized input parcels, the parcels created by simulation are often much too long with version *b* whereas the patio created with version *c* are often too large. In all cases, the roads created by simulation are perfectly connected to the existing road network. The area of roads created by simulation is too large when large buildings are planned to be built (e.g. medium-sized blocks of flats) but it is possible to generate thinner roads by modifying the urban fabric profiles (i.e. the values of morphological parameters, see Table 3).

Considering the advantages and the limits of both *OBB* and *SS* algorithms, we have implemented in *Parcel Manager* a complementary division process that combines the two algorithms: an *OBB* division is first applied in order to create blocks; then, a *SS* division with the creation of peripheral roads (version *b* of the algorithm) is applied to the blocks. A parameter called *approxNumberParcelPerBlock* sets the approximate number of parcels that have to be created with the *SS* division in each block created by the *OBB* division. The size of the blocks from which the *OBB* division stops is determined by the value of the parameter *maximalArea* multiplied by the value of the parameter *approxNumberParcelPerBlock*. Figure 20 gives an illustration of the parcel layouts obtained with this parcel division process. Interestingly, the simulated parcel layouts combine the advantages of both *OBB* and *SS* algorithms: they are visually realistic; the connection to the road network is good; the shape of the

705 parcels is not too regular; only 3% of the reshaped parcels have an area larger than the *maximal area* value set in the urban fabric profiles.



Figure 20: Simulation of the *test scenario* on the community of Gennes with the *OBB-SS* algorithm. Workflows *Zone Division* in the AU1 zones and *Zone Consolidation* in the AU2 zones. The width of peripheral roads has been set to 7 m. and the minimum width of contact between roads and parcels has been set to 12 m.

710 The *Flag Division* algorithm included in the *Parcel Densification* workflow simulates the complete densification of the zones to be reshaped. Simulated parcel layouts are realistic but we can imagine that such a complete densification is never fully achieved in real cases. A way to improve the realism of the simulation could be to pre-define a limited set of reshaping operations using a random or ad hoc selection process.

715 Finally, parcel shapes created with the *OBB* and *SS* algorithms differ greatly; they also differ from parcel shapes created with the *OBB-SS* algorithm and the *flag division* algorithm. All these simulated parcel shapes are visually realistic taking the limitations exposed above into account. It is thus possible to use different parcel division processes in several workflows to create fine-tuned scenarii in which the shapes of parcels created by simulation is interestingly diversified. Various scenarii can then be compared regarding the number and the shape of parcels created by simulation, and ultimately their potentials for new building

720

constructions.

5. Comparison of the shape of parcels created by simulation with the shape of parcels in real cases

Since an objective of *Parcel Manager* is to generate realistic parcel layouts, we have designed a specific use case to compare the evolution of real parcel plans with the reshaping of parcel plans simulated with *Parcel Manager*. This *comparison use case* concerns the change of the parcel plans of 11 communities of the Seine-et-Marne department, near Paris capital city (France), between 2003 and 2018. *Parcel Manager* is dedicated to the comparison of scenarios that simulate different parcel reshaping processes, which conform with different planning principles, in different places of an area under study. Scenarios are then compared regarding the number, the shape and the location of reshaped parcels. In no instance, however, *Parcel Manager* aims at reproducing past parcel reshaping processes or forecasting future parcel changes. In light of this, the comparison the evolution of the parcel plans of 11 communities of the Seine-et-Marne department with the reshaping of parcel plans simulated with *Parcel Manager* is purely exploratory. A single scenario simulated with *Parcel Manager* is compared to the real changes of parcels between 2003 and 2018. The aim is just to roughly assess if simulation results are globally realistic regarding some quantitative indices.

The *comparison use case* involves three steps:

1. sorting and selecting parcels that have changed between 2003 and 2018 in the real world,
2. simulating the reshaping of this set of parcels with *Parcel Manager*,
3. comparing the geometry and other characteristics of the parcels reshaped by simulation with real parcels whose shape has changed between the two dates.

5.1. Sorting and selecting parcels

The process of sorting and selecting parcels is realized with a specific algorithm (Algorithm 10 in Appendix .2). Two different outputs are produced by the algorithm:

- The *real* dataset is the set of 2018 parcels that have evolved from 2003 to 2018 and that will be compared to the parcels reshaped by simulation.
- The *simulation* dataset is the set of 2003 parcels selected to be reshaped with *Parcel Manager*.

If only a very small part of a parcel has changed from 2003 to 2018 or if the size of a parcel largely exceeds the usual size of residential parcel lots, those parcels are not included in the *simulation* and *real* datasets.

In the case of the 11 communities under consideration, the algorithm selects 930 parcels for the *simulation* dataset and 3,346 parcels for the *real* dataset.

5.2. Description of the scenario simulated with Parcel Manager

The scenario involves three steps:

- 765 • The first step applies the *Parcel Densification* workflow on parcels located in residential zones of the zoning plan and being smaller than a threshold defined as the value of parameter *maximal parcel area* multiplied by 5. This threshold value has been empirically set.
- The second step applies the *Zone Consolidation* workflow on parcels larger than the predefined threshold and located in residential zones of the zoning plan.
- 770 • The third step applies the *Zone Consolidation* workflow on parcels located outside the residential zones of the zoning plan (developable and non developable zones).

As the communities of the area under study have very different parcel layouts, values of parameters *maximal parcel area* and *minimal parcel area* of their urban fabric profile vary according to the community. In order to set those values, we have used the *GetParameterOfScene* package (cf. section 3.2) to calculate the statistical distribution of the area of all built parcels in each community. The *third quartile* of this distribution has been chosen to represent the *maximal parcel area* and the *first decile* has been chosen to represent the *minimal parcel area*. The choice of these two statistical thresholds has been based on case-by-case observations. The other parameter values of the *urban fabric profile* are identical for all communities and correspond to the *small-sized single-family houses* of Table 3.

In view of the requirements of the scenario (i.e. values of parameters *maximal parcel area* and *minimal parcel area* varying according to the community and *small-sized single-family houses* planned to be constructed), the *OBB* parcel division process with the joint creation of new roads has been chosen. The simulation of this scenario creates or reshapes 2,300 parcels from the 930 parcels of the *simulation* dataset.

790 5.3. Comparison of parcels reshaped in the real case with parcels reshaped by simulation

Real and simulated parcels are compared using seven indices: *Area*, *Perimeter*, *Hausdorff distance* and *Aspect ratio* focus on shape similarities; *Number of neighbours*, *Length of contact with road* and *Width of contact with road* are more topological; they characterise the immediate surrounding of the parcels and their connection to the road network. In order to avoid a quantitative bias due to the difference in parcel numbers (2,300 parcels for the *simulated* set and 3,346 parcels for the *real* set), each indice is represented by its density of probability. Particular parcels having an area larger than 3,000 m^2 , a perimeter longer than 500m, an aspect ratio higher than 11, a number of neighbours higher than 5,000, or a width of contact with road longer than 300 m are excluded from the comparison. 35 parcels are concerned.

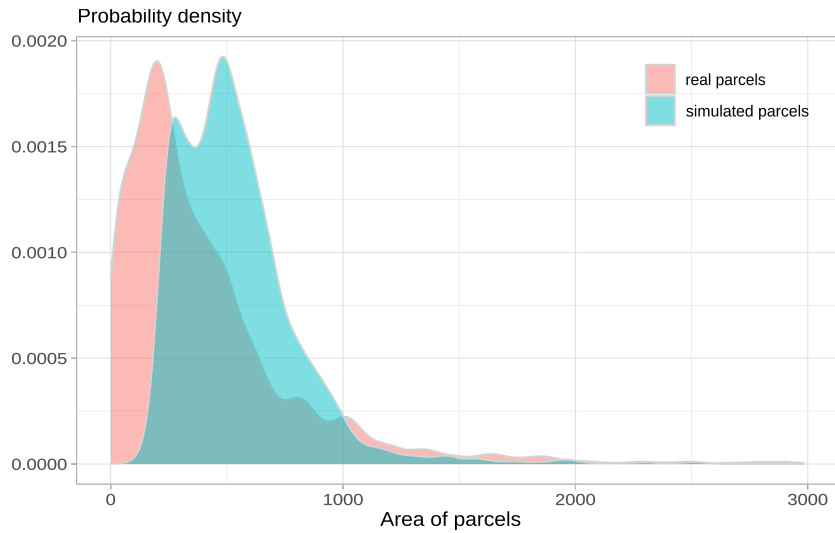


Figure 21: Area of the real and simulated parcels in 2018.

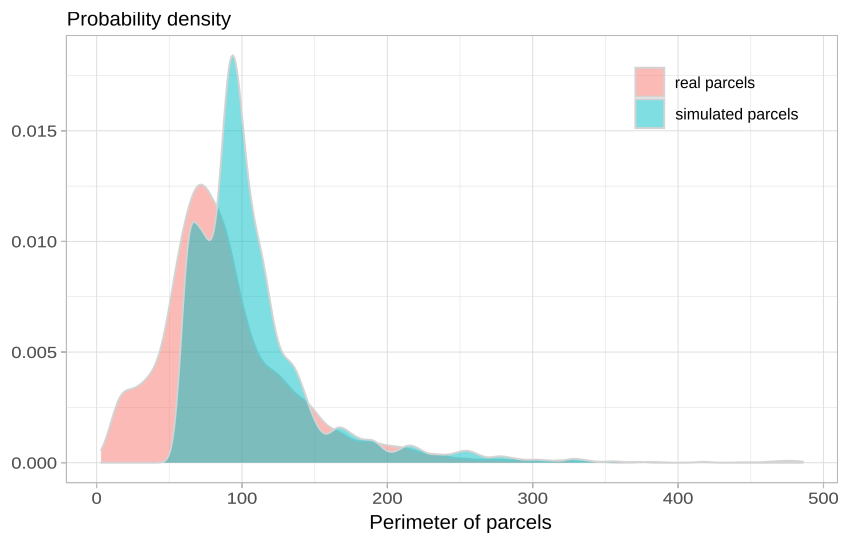


Figure 22: Perimeter of the real and simulated parcels in 2018.

805 The *Area* and *Perimeter* distributions of Figures 21 and 22 are used to assess the similarity of the *simulated* and the *real* sets in terms of parcel surface and length of perimeter. Simulated parcels tend to be systematically larger in surface and perimeter than real parcels. The shape of the distributions differs mostly for small parcels (area smaller than 100 square meters and perimeter smaller than 50 meters). This is explained by the fact that the minimal size of simulated parcels cannot be lower than the first decile of built parcel surfaces of a community,

810 which corresponds to about 100 square meters for most communities of our dataset. In contrast, real parcels may be smaller due to past reconfigurations.

The *Hausdorff similarity distribution* of Figure 23 represents the Hausdorff distance between the *simulated* and *real* parcel shapes calculated using the JTS library (Shekhar et al., 2017). Values are normalized between 0 and 1; the greater the value, the more similar the shapes. The Hausdorff distance is calculated between each simulated parcel and the real parcel with which its intersection is the largest. To provide the reader with an intuition of the shape differences that may be captured by the Hausdorff distance, three examples of pairs of simulated and real parcels with the associated Hausdorff distance are displayed on Figure 24.

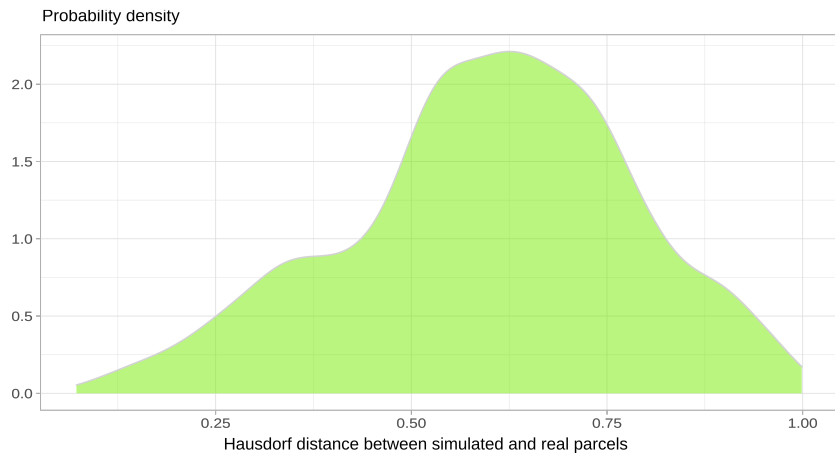


Figure 23: Hausdorff distance between pairs of real and simulated parcels.

The distribution of Figure 23 is quite symmetrical although it exhibits a slightly negative skewness (-0.26). The Hausdorff distance separating the vast majority of pairs of parcels is around 0.5. The distance of few couples is above 0.75.

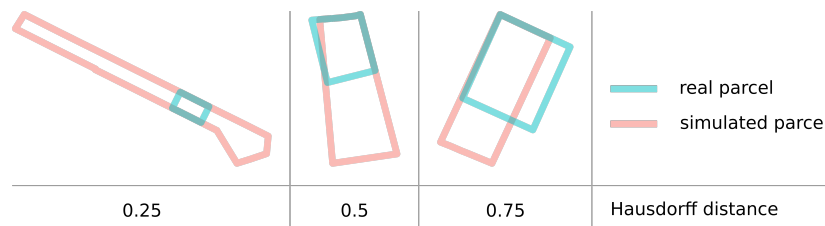


Figure 24: Hausdorff distance of three pairs of real and simulated parcels.

825 The *Aspect ratio* of a parcel is defined as the ratio between the diameter of its smallest circumscribing circle and the diameter of its largest inscribed circle. Round and compact shapes have an aspect ratio close to 1 whereas

more elongated shapes have a higher aspect ratio. As shown on Figure 25, the distribution of aspect ratios of real and simulated parcels are very close although the shape of some real parcels is more compact than the shape of any simulated parcel.

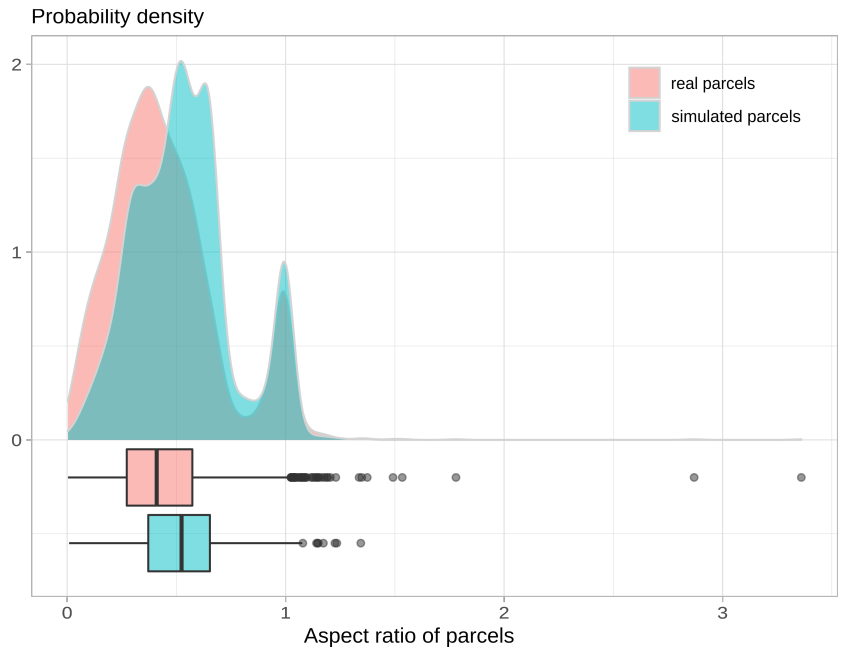


Figure 25: Distributions of aspect ratios of parcels with associated boxplots.

The *Number of neighbours* shown on Figure 26 is a topological measure accounting for the number of adjoining neighbours of each parcel. Neighbouring parcels are defined up to a certain tolerance threshold parameter set to 1 meter.

In order to compare the overall shape of the two histograms, each of both has its own y-axis scale (count) and the same x-axis scale (number of neighbours). The shape of the two histograms is quite similar: few isolated parcels, a majority of parcels having 2 to 4 neighbours, and other parcels having most often 5 to 8 neighbours. Simulated parcels are surrounded by slightly more neighbours than real parcels.

Contact with road is a boolean metric depicting whether a parcel touches or not a road. Only 5.6% of real parcels and 0.6% of simulated parcels have no contact with a road. The very low percentage of simulated parcels is explained by the chosen value of the *block shape* parameter (6, see Table 3), which ensures that almost all simulated parcels have a direct access to a road.

The *Length of contact with road* of a parcel is the cumulated length of the parcel edges that touch a road. The metric is visualized as two superimposed distributions on Figure 27. The two distributions are quite similar. The difference of the peak positions between simulated lengths and real lengths can be

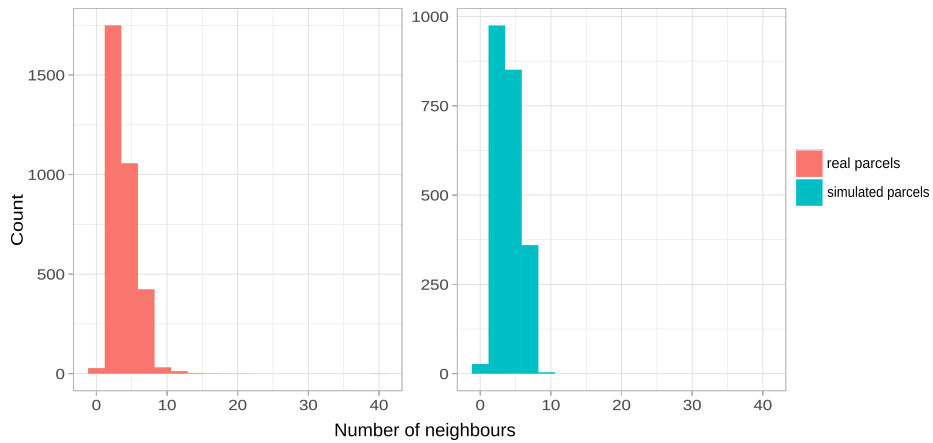


Figure 26: Number of neighbours of each parcel.

850 explained by the fact that almost every simulated parcels are in contact with a road, thus shifting slightly the peak of the distribution to the right, whereas some real parcels are not in contact with a road, thus shifting slightly the peak of the distribution to the left.

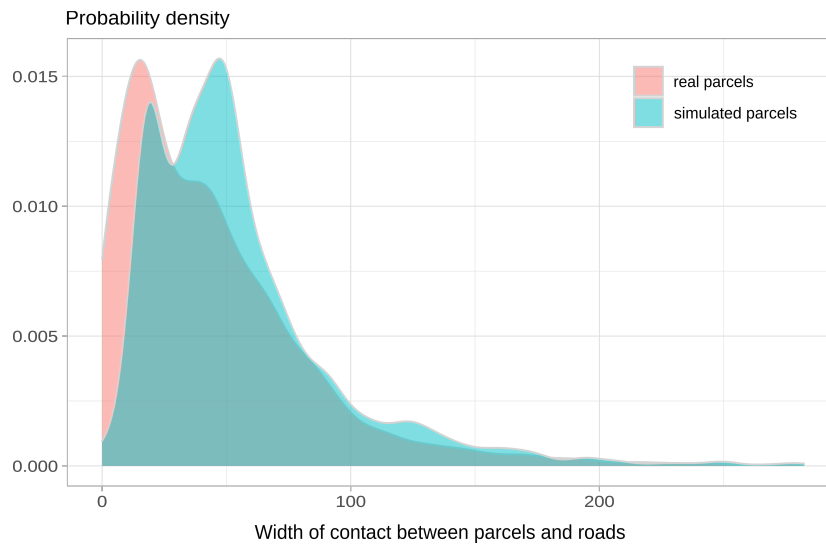


Figure 27: Distributions of the length of the contact with roads of each parcel.

855 We now focus only on the parcels reshaped with the *Parcel Densification* workflow, which involves the *flag division* process. 707 parcels are concerned. Figure 28 shows that the shape of parcels reshaped with the *Parcel Densification* workflow highly resembles the shape of real parcels.

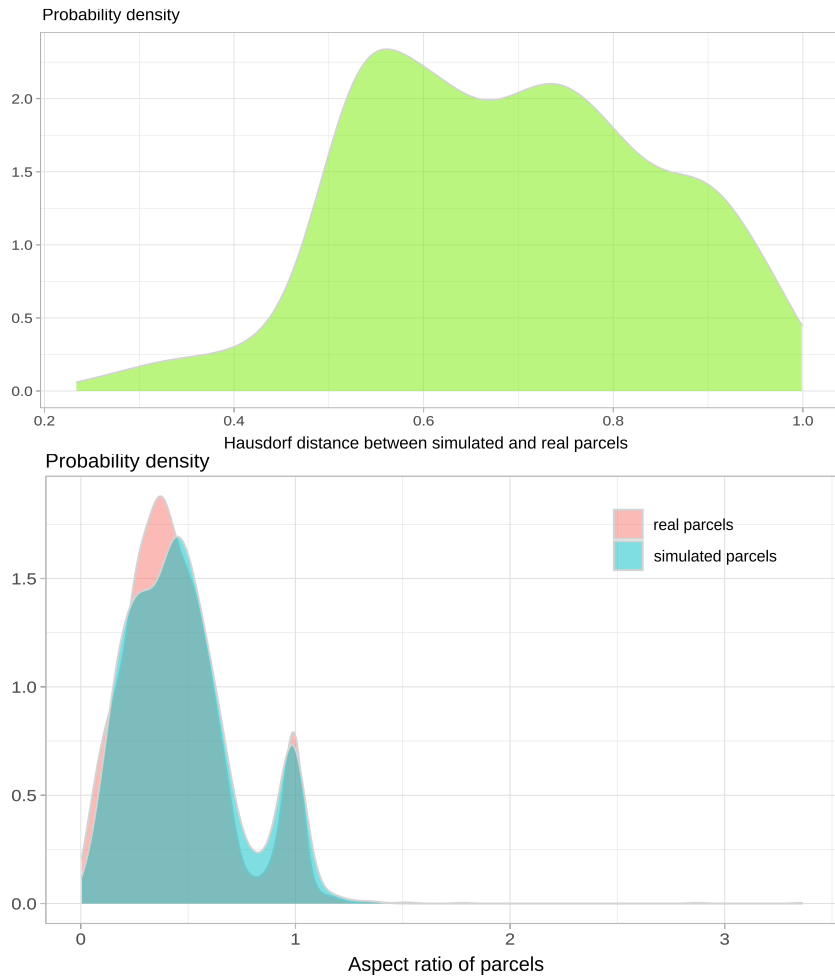


Figure 28: Measures of shape similarity between parcels reshaped in real cases and parcels reshaped with the *Parcel Densification* workflow.

5.4. Discussion of the comparison results

Regarding the seven indices under consideration, it appears that the shapes of parcels reshaped with *Parcel Manager* are quite similar to the shapes of parcels reshaped in real cases. Parcel shapes obtained with the *Parcel Densification* workflow are significantly more similar to real parcel shapes than parcel shapes designed with the other workflows. This is partly due to the fact that densified parcels are usually surrounded by real parcels, whose shape constrains a lot the *flag division* process.

The aim of *Parcel Manager* is not to reproduce past processes of parcel reshaping but to simulate and compare different scenarii of parcel reshaping in view of the comparative assessment of their potentials for building constructions.

The fact that the shapes of parcels reshaped with *Parcel Manager* are quite
870 similar (but not totally similar) to the shapes of parcels reshaped in the real
case suggests that the level of realism of *Parcel Manager* simulation results is
satisfying.

6. Conclusion

In this article, we have presented the *Parcel Manager* software application
875 dedicated to the simulation of a large variety of parcel reshaping processes, with
or without the joint creation of new roads. Simulations can consider simulta-
neously different types of input zones (e.g. individual parcels located within an
urban fabric or large non-built zones where greenfield development is considered)
and can be performed automatically for a large region in which parcel reshaping
880 rules vary spatially. We have implemented some existing parcel division algo-
rithms, namely *Oriented Bounding Box* and *Straight Skeleton*, and improved
them, in particular regarding the creation of new roads and the reshaping of
large square-shaped parcels. These improvements have proven successful. We
have also combined the two algorithms within a new algorithm, namely *OBB-*
885 *SS*, which overcomes some limitations of each of its two predecessors. Last but
not least, we have developed a *flag division* algorithm in order to simulate the
densification of built parcels.

Parcel Manager produces realistic parcel layouts that represent either in-
fill urban developments, edge expansions or leapfrog developments. Nonethe-
890 less, parcel layouts simulated with *Parcel Manager* are not destined for a
direct ready-to-use application by planning practitioners or parcel owners. *Par-*
cel Manager is indeed dedicated to the comparison of scenarios that simulate
different parcel reshaping processes, which conform with different planning and
design principles, in different places of an area under study. Scenarios are com-
905 pared regarding the number, the shape and the location of reshaped parcels.
In no instance, *Parcel Manager* aims at reproducing past parcel reshaping pro-
cesses or forecasting future parcel changes.

As *Parcel Manager* has been developed in Java, it can either be used as a
stand-alone application or integrated within another application as a library.
900 The simulation process is light. The encapsulation of parcel functions and
processes in workflows, and then in scenarios, enables the design of complex
simulations for large areas. Moreover, the free and open-source implementation
allows users to improve existing workflows or to create new workflows. The code
documentation and technical explanations encourage the use of *Parcel Manager*.
905 Additionally, a graphical user interface facilitates its use by people having
no programming skills.

Parcel Manager can be used to assist planners in designing local master
plans. It can help them to evaluate the interest in opening new zones to resi-
dential construction or reducing the number or the size of current developable
910 zones. As planning programs usually try to limit urban sprawl and its numerous
negatives impacts, many planning policies promote the densification of urban

zones so as to avoid greenfield developments. A specific study is often made to analyse the potential to increase locally the built density. *Parcel Manager* can be used to support such analysis. *Parcel Manager* also provides a basis to assess
915 the potentials for new building constructions since almost every reshaped parcels should allow for the construction of the buildings they have been designed for. With this in mind, *Parcel Manager* can be coupled with the software application SimPLU3D (Brasebin et al., 2018) that simulates the building constructibility of a parcel or a group of parcels by generating 3D buildings that respect the
920 rules of local urban master plans as well as other morphological constraints (e.g. maximum building height, front yard setback) (Colomb, 2019). Ultimately, the potentials for new building constructions could be confronted to some targeted objectives of new housing creations, especially the type of housings and the quantity of housing units.

925 The current version of *Parcel Manager* only concerns the reshaping of parcels dedicated to the construction of residential buildings. Yet, thanks to its modular structure, it could be easily adapted to the reshaping of parcels in view of the construction of commercial or industrial buildings. This would only require to modify the urban fabric profiles and the parameters of the parcel division
930 processes. In case of the reshaping of large zones, it might also be interesting to add the possibility of mixing different targeted urban fabric profiles. It could then be relevant to use calibration algorithms in order to automatically generate a specific set of input parameters for each zone. To this end, *Parcel Manager* can already be used with the exploration model OpenMole (Reuillon et al., 2013),
935 which enables computer-intensive automatic calibration analyses.

7. Acknowledgments

This research benefited from the financial support of the I-SITE University Bourgogne Franche-Comté as part of the project PubPrivLands.

Appendix .1. Source code Appendix

940 The source code of *Parcel Manager* is open. It can be freely reused, improved, modified, and shared following the terms of the AGPL3 Licence (<https://www.gnu.org/licenses/agpl-3.0.html>). The source code is available from: <https://tinyurl.com/ywk7y7np>. A technical documentation is available in the Javadoc format: <https://tinyurl.com/yckzcu94>. The graphical user
945 interface (GUI) is available from: <https://tinyurl.com/3yw7srcc>. A pre-compiled and ready to use version of the GUI can be found at: <https://tinyurl.com/mr6dbby3>.

Appendix .2. Pseudo-code of the sorting algorithm used in section 5

950 Algorithm 10 is dedicated to the comparison of the parcel plans of two different dates considering a given study area. It creates the geographical datasets that are used for the comparison of the shape of parcels created by simulation with the shape of parcels in real cases (see section 5). Outputs are four layers:

Table .4: Source code used in the different sections of the article. Links are referenced in relation to the root path of the source code completed by the Java signature folder: "src/main/java/fr/ign/artiscales/pm".

Section	Usage	URL
3.1.1	Oriented Bounding Box process	division.OBBDivision
3.1.3	Flag division process	division.FlagDivision
3.1.2	Straight Skeleton division process	division.StraightSkeletonDivision
3.2	Methods to analyse the general characteristics of a given parcel dataset	analysis.RealUrbanFabricParameters
3.3	Methods to mark the parcels that will be reshaped by simulation	parcelFunction.MarkParcelAttributeFromPosition
4	Test scenario used for the test of each parcel division algorithm	usecase.TestScenario
5	Comparison of the shape of parcels created by simulation with the shape of parcels in real cases	usecase.CompareSimulatedWithRealParcels
5.1	Selection of parcels that have changed between two dates	parcelFunction.ParcelCollection

- the *same* layer contains the *2003 parcels* that have not changed between 2003 and 2018.
- 955 • the *notSame* layer contains the *2003 parcels* that have changed between 2003 and 2018.
- the *simulation* layer contains the *2003 parcels* that have changed between 2003 and 2018. Their geometries have been slightly transformed in order to use them to mark the parcels of the whole parcel map that will be
960 reshaped by simulation.
- the *real* layer contains the *2018 parcels* that have changed between 2003 and 2018.

Algorithm 10 Pseudo code of the Algorithm *sortDifferentParcel* that sorts the *2003 parcels* and compares them to the *2018 parcels*. Parameters *minParcelArea* and *maxParcelArea* correspond to the area bounds of simulated parcels defined in the *urban fabric profile* (i.e. morphological parameters)

```

function ISPARCELSAME(Parcel  $P_1$ , Parcel  $P_2$ )
  return AREA( $P_c$ ) = AREA( $P_r$ )  $\pm$  5%  $\wedge$  HAUSDORFFSIMILARITY( $P_1$ ,  $P_2$ ) > 0.95
end function
function SORTDIFFERENTPARCEL(PolygonCollection  $P_{2003}$ , PolygonCollection  $P_{2018}$ )
  for Parcel  $P_r$  :  $P_{2003}$  do
     $P_{PcompRef} \leftarrow$  GETINTERSECTINGPARCELS( $P_{2018}$ ,  $P_r$ )
    for Parcel  $P_c$  :  $P_{PcompRef}$  do
      if ISPARCELSAME( $P_c$ ,  $P_r$ ) then
         $same.add(P_r)$ 
      else  $\triangleright$  We now check if the parcel has intentionally been deleted
         $P_{Base} \leftarrow$  POLYGONIZE( $P_{compRef}$ )  $\triangleright$  Polygonize function creates new geometries
        from every intersecting parts of a list of geometries
         $P_{Base}.add(P_r)$   $\triangleright$  We add the 2003 parcel to the 2018 parcels
         $add = true$ 
        for Parcel  $P_b$  :  $P_{Base}$  do
          if ISPARCELSAME( $P_b$ ,  $P_r$ ) then  $\triangleright$ 
            If a simplified geometry containing the 2003 parcel has not been put in the same collection, we
            conclude that the 2003 parcel has intentionally been deleted and don't add it to any collection
             $add = false$ 
          end if
        end for
        if  $add$  then
           $notSame.add(P_r)$ 
        end if
      end if
    end for
  end for  $\triangleright$  Create a collection of the 2018 parcels that have evolved

  for Parcel  $P_c$  :  $P_{2018}$  do
    if ISINTERSECTING( $P_c$ ,  $P_{notSame}$ ) then
       $evolved.add(P_c)$ 
    end if
  end for  $\triangleright$  Create a collection of the
  2003 parcels that are of interest for residential parcel simulation, also called simulation. Three
  conditions must be required to transform a non-build parcel into a simulation feature: i) Median
  of the parcel's area distribution must be below minimal parcel's area an upper maximal parcel's
  area. ii) Mean of the parcel's area distribution must be below minimal parcel's area an upper
  maximal parcel's area. iii) The biggest real parcel mustn't represents 80% of the total zone area
  and be higher than 5x the max parcel size. Calibration coefficients have been empirically set.
  for Parcel  $P_{ns}$  :  $P_{notSame}$  do
     $P_{2018-ns} \leftarrow$  GETINTERSECTINGPARCELS( $P_{2018}$ ,  $P_{ns}$ )
     $add = true$ 
    if  $minParcelArea * 0.5 > MEDIAN(P_{2018-ns}.area) > maxParcelArea * 1.5 \vee$ 
 $minParcelArea * 0.25 > MEAN(P_{2018-ns}.area) > maxParcelArea * 2 \vee$ 
 $(BIGESTPARCEL(P_{2018-ns}).area * 0.8 > P_{ns}.area \wedge P_{ns}.area > 5 * maxParcelArea)$ 
then  $add = false$ 
    end if
    if  $add = true$  then
       $simulation.add(P_{ns})$ 
    end if
  end for
   $evolved \leftarrow$  SELECTINTERSECTION( $evolved$ ,  $simulation$ )
return  $simulation$ ,  $evolved$ ,  $same$ ,  $notsame$ 
end function

```

References

- Abolhasani, S., Taleai, M., Karimi, M., and Node, A. R. (2016). Simulating
965 urban growth under planning policies through parcel-based cellular automata
(parca) model. *International Journal of Geographical Information Science*,
30(11):2276–2301.
- Aichholzer, O. and Aurenhammer, F. (1996). Straight skeletons for general
polygonal figures in the plane. In Cai, J.-Y. and Wong, C. K., editors, *Com-
970 puting and Combinatorics*, pages 117–126, Berlin, Heidelberg. Springer Berlin
Heidelberg.
- ARKISoft (2020). Autocad plugin : Top-parcel. Available at <https://www.arkisoft.es/tienda/top-parcel>.
- Asiama, K., Bennett, R., Zevenbergen, J., and Mano, A. D. S. (2019). Respon-
975 sible consolidation of customary lands: A framework for land reallocation.
Land Use Policy, 83:412 – 423.
- Brasebin, M., Perret, J., Mustière, S., and Weber, C. (2018). 3d urban data to
assess local urban regulation influence. *Computers, Environment and Urban
Systems*, 68:37–52.
- 980 Brennan, T. M. and Venigalla, M. (2016). A constructability assessment method
(cam) for sustainable division of land parcels. *Land Use Policy*, 56:47 – 57.
- Chen, G., Esch, G., Wonka, P., Müller, P., and Zhang, E. (2008). Interactive
procedural street modeling. *ACM Trans. Graphics*.
- Cheng, S.-W., Mencil, L., and Vigneron, A. (2014). A Faster Algorithm for
985 Computing Straight Skeletons. In *Algorithms - ESA 2014*, pages 272–283.
Springer, Berlin, Germany.
- City Engine (2019). Esri city engine documentation - block parameters module.
Available at : [https://doc.arcgis.com/en/cityengine/latest/help/help-layers-
block-parameters.htm](https://doc.arcgis.com/en/cityengine/latest/help/help-layers-block-parameters.htm).
- 990 Colomb, M. (2019). *Simulation de formes réalistes de développement résidentiel,
de l'échelle du bâtiment à celle de l'ensemble d'une région urbaine*. Phd thesis,
University Paris-Est (France).
- Cooper, J., Donegan, K., Ryley, T., Smyth, A., and Granzow, E. (2002). Densifi-
cation and urban compaction: Reinforcing the drive for sustainability. *Trans-
995 portation Research Record*, 1817(1):102–109.
- Dahal, K. R. and Chow, T. E. (2014). A gis toolset for automated partitioning
of urban lands. *Environmental Modelling & Software*, 55:222 – 234.
- Demetriou, D. (2017). A spatially based artificial neural network mass valuation
model for land consolidation. *Environment and Planning B: Urban Analytics
1000 and City Science*, 44(5):864–883.

- Demetriou, D. (2018). Gis-based automated valuation models (avms) for land consolidation schemes. In *6th International Conference on Cartography and GIS, Albena, Bulgaria*, pages 43–51.
- 1005 Demetriou, D., See, L., and Stillwell, J. (2013). A spatial genetic algorithm for automating land partitioning. *International Journal of Geographical Information Science*, 27(12):2391–2409.
- Demetriou, D., Stillwell, J., and See, L. (2012). An integrated planning and decision support system (ipdss) for land consolidation: Theoretical framework and application of the land-redistribution modules. *Environment and Planning B: Planning and Design*, 39(4):609–628.
- 1010 Donnelly, S. and Evans, T. P. (2008). Characterizing spatial patterns of land ownership at the parcel level in south-central Indiana, 1928–1997. *Landscape Urban Plann.*, 84(3):230–240.
- 1015 Eder, G., Held, M., and Palfrader, P. (2021). Implementing straight skeletons with exact arithmetic: Challenges and experiences. *Computational Geometry*, 96:101760.
- Gössling, S., Schröder, M., Späth, P., and Freytag, T. (2016). Urban space distribution and sustainable transport. *Transport Reviews*, 36(5):659–679.
- 1020 Harasimowicz, S., Janus, J., Baciór, S., and Gniadek, J. (2017). Shape and size of parcels and transport costs as a mixed integer programming problem in optimization of land consolidation. *Computers and Electronics in Agriculture*, 140:113–122.
- 1025 Jarosz, B., Murdock, S. H., and Swanson, D. A. (2008). *Using Assessor Parcel Data to Maintain Housing Unit Counts for Small Area Population Estimates*, pages 89–101. Springer Netherlands, Dordrecht.
- Kelly, T. and Wonka, P. (2011). Interactive architectural modeling with procedural extrusions. *ACM Transactions on Graphics (TOG)*, 30(2):14.
- 1030 Kilić, J., Jajac, N., Rogulj, K., and Mastelić-Ivić, S. (2019). Assessing Land Fragmentation in Planning Sustainable Urban Renewal. *Sustainability*, 11(9):2576.
- Lagrab, W., Aknin, N., and Alruqimi, M. (2018). An adaptive spatial subdivision of 2-d irregular area for urban planning. In *Proceedings of the 3rd International Conference on Smart City Applications*, SCA '18, pages 37:1–37:7, New York, NY, USA. ACM.
- 1035 Levi, Y., Bekhor, S., and Rosenfeld, Y. (2019). A multi-objective optimization model for urban planning: The case of a very large floating structure. *Transportation Research Part C: Emerging Technologies*, 98:85 – 100.

- Lipp, M., Scherzer, D., Wonka, P., and Wimmer, M. (2011). Interactive modeling of city layouts using layers of procedural content. *Computer Graphics Forum*, 30(2).
1040
- Litman, T. (2020). Transportation land valuation - evaluating policies and practices that affect the amount of land devoted to transportation facilities. Technical report, Victoria Transport Policy Institute.
- Long, Y. and Wu, K. (2017). Simulating block-level urban expansion for national wide cities. *Sustainability*, 9(6).
1045
- McGarigal, K. and Marks, B. J. (1995). FRAGSTATS: spatial pattern analysis program for quantifying landscape structure. *Gen. Tech. Rep. PNW-GTR-351. Portland, OR: U.S. Department of Agriculture, Forest Service, Pacific Northwest Research Station. 122 p*, 351.
- Meyer, J. R. and Gómez-Ibáñez, J. A. (2013). *Autos, Transit, and Cities*. Harvard University Press, Cambridge, MA, USA.
1050
- Munafò, M. R., Nosek, B. A., Bishop, D. V. M., Button, K. S., Chambers, C. D., Du Sert, N. P., Simonsohn, U., Wagenmakers, E.-J., Ware, J. J., and Ioannidis, J. P. A. (2017). A manifesto for reproducible science. *Nature Human Behaviour*, 1(0021):1–9.
1055
- Mustafa, A., Zhang, X. W., Aliaga, D. G., Bruwier, M., Nishida, G., Dewals, B., Erpicum, S., Archambeau, P., Pirotton, M., and Teller, J. (2018). Procedural generation of flood-sensitive urban layouts. *Environment and Planning B: Urban Analytics and City Science*, 0(0):2399808318812458.
- Næss, P., Saglie, I.-L., and Richardson, T. (2019). Urban sustainability: is densification sufficient? *European Planning Studies*, pages 1–20.
1060
- Parish, Y. I. H. and Müller, P. (2001). Procedural modeling of cities. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, page 301–308, New York, NY, USA. Association for Computing Machinery.
1065
- Perret, J. (2006). *Modélisation d'environnements urbains virtuels*. Phd thesis, University of Rennes (France).
- Pocewicz, A., Nielsen-Pincus, M., Goldberg, C. S., Johnson, M. H., Morgan, P., Force, J. E., Waits, L. P., and Vierling, L. (2008). Predicting land use change: comparison of models based on landowner surveys and historical land cover trends. *Landscape Ecology*, 23(2):195–210.
1070
- Reuillon, R., Leclaire, M., and Rey-Coyrehourcq, S. (2013). Openmole, a workflow engine specifically tailored for the distributed exploration of simulation models. *Future Generation Computer Systems*, 29(8):1981 – 1990.

- 1075 Saaty, T. (1977). A scaling method for priorities in hierarchical structures. *Journal of Mathematical Psychology*, 15(3):234 – 281.
- Shekhar, S., Xiong, H., and Zhou, X., editors (2017). *Java Topology Suite (JTS)*, pages 1095–1095. Springer International Publishing, Cham.
- Sun, B. and Robinson, D. T. (2018). Comparison of statistical approaches for
1080 modelling land-use change. *Land*, 7(4).
- Sun, J., Yu, X., Baciú, G., and Green, M. (2002). Template-based generation of road networks for virtual city modeling. In *Proceedings of the ACM Symposium on Virtual Reality Software and Technology, VRST '02*, pages 33–40, New York, NY, USA. ACM.
- 1085 Vanegas, C. A., Kelly, T., Weber, B., Halatsch, J., Aliaga, D. G., and Müller, P. (2012). Procedural generation of parcels in urban modeling. *Computer Graphics Forum*, 31(2pt3):681–690.
- Vasconcellos, E. A. (2001). *Urban Transport Environment and Equity: The Case for Developing Countries*. CRC Press, Routledge.
- 1090 Wickramasuriya, R., Chisholm, L. A., Puotinen, M., Gill, N., and Klepeis, P. (2011). An automated land subdivision tool for urban and regional planning: Concepts, implementation and testing. *Environmental Modelling & Software*, 26(12):1675 – 1684.
- Wilson, B. and Song, Y. (2010). Do large residential subdivisions induce further
1095 development? *Journal of the American Planning Association*, 77(1):5–22.
- Wiseman, N. and Patterson, Z. (2016). Testing block subdivision algorithms on block designs. *Journal of Geographical Systems*, 18(1):17–43.
- Yang, Y.-L., Wang, J., Vouga, E., and Wonka, P. (2013). Urban pattern: Layout design by hierarchical domain splitting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2013)*, 32:181:1–181:12.
- 1100 Yazyćy, S. (2016). A parametric landscape urbanism method: The search for an optimal solution. *ITU J Faculty Arch*, 13(3):155–165.