CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

Testing Block Subdivision Algorithms on Block Designs

Nathalie Wiseman
Zachary Patterson

June 2015

CIRRELT-2015-26

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

Testing Block Subdivision Algorithms on Block Designs

Natalie Wiseman[1], Zachary Patterson[1,2,*]

[1] Department of Geography, Planning and Environment, Concordia University, 1455 de Maisonneuve W., H 1255-26 (Hall Building), Montreal, Canada H3G 1M8

[2] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT)

Abstract. Integrated land use-transportation models predict future transportation demand taking into account how households and firms arrange themselves partly as a function of the transportation system. Recent integrated models require parcels as inputs and produce household and employment predictions at the parcel scale. Block subdivision algorithms automatically generate parcel patterns within blocks. They are useful in preparing missing input parcels to integrated models and in software designed to visualize their results. Evaluating block subdivision algorithms is done by way of generating parcels and comparing them to those in a parcel database. Realistic block subdivision improves input data and could improve spatial accuracy of model predictions, and consequently, transportation demand forecasts. Three block subdivision algorithms are evaluated on how closely they reproduce parcels of different block types found in a parcel database from Montreal, Canada. A standardized block type classification is used that consists of mutually exclusive and comprehensive categories. A statistical method is used for finding a better algorithm and the probability it will perform well for a given block type. Results suggest the Oriented Bounding Box algorithm performs better for warped grids, as well as gridiron and fragmented uniform sites. It also produces more similar parcel areas and widths. The Generalized Parcel Divider 1 algorithm performs better for gridiron and fragmented non-uniform sites. The Straight Skeleton algorithm performs better for loop and lollipop networks and also produces more similar parcel shapes.

Keywords. Automatic block subdivision, algorithm evaluation, urban and regional planning, integrated modeling.

Acknowledgements. This study was funded by the Canadian Research Chairs program, the Canadian Foundation for Innovation and the Agence métropolitaine de transport. This support is gratefully acknowledged.

---

* Corresponding author: Zachary.Patterson@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2015

1. Introduction

Increasingly, planning for transportation infrastructure investments is being integrated with land use planning. Successful integrated planning, it is hoped, can provide competitive alternatives to automobile use and thereby reduce congestion on roadways and greenhouse gas emissions. Integrated land use-transportation modeling is increasingly a method chosen to help forecast transportation demand in this context. It is used because it explicitly accounts for the effect of transportation network performance on population and employment distribution, and thereby overall transportation demand (Waddell et al. 2007; Borning et al. 2008).

Within the field of integrated modeling, there has been a trend toward representing phenomena at increasingly finer spatial units. The finest unit used in integrated modeling to date is the parcel. This is also the spatial unit to which all other data (households, jobs, buildings, etc.) are linked (Waddell 2009). Increasingly fine spatial representation has been motivated by understandings in complex systems, behavioral theory and statistical aggregation bias, and made possible by advances in Geographic Information Systems (Xie and Batty 2003). This increasingly disaggregated spatial representation presents many opportunities but also many challenges within the field of integrated modeling.

For one, parcel-level models require a large amount of highly detailed data (e.g. parcel data), which can be unavailable or incomplete (Schirmer 2010) and whose absence can delay running simulations with integrated models (Patterson and Bierlaire 2010). Moreover, future parcel data is by definition not available, but clearly relevant to forecasting future locations of households and employment with parcel-based models. Recently, a number of algorithms have been developed that can help with both of these problems – so-called *parcel generation or block subdivision* algorithms. Some of these algorithms have been implemented in City Generation Engines, such as Esri's CityEngine and Synthicity's UrbanCanvas, as visualization tools integrated with simplified behavioral models (Vanegas et al. 2012). Others have been developed as stand-alone applications and could potentially be linked to integrated models (Wickramasuriya et al. 2011; Dahal and Chow 2014). Both sets of tools can be used to generate missing data from parcel datasets needed as inputs to parcel-level integrated models.

A central hypothesis of this research is that given the diversity of urban forms and block designs, some algorithms are more likely to perform better on some block types than others. At the same time, these algorithms have yet to be compared with each other in terms of how well they reproduce parcel layouts for different block types.

This research seeks to evaluate which algorithm, if any, performs better on a given block type. To this end, parcels are generated using three algorithms on the same set of blocks from a parcel database for Montreal, Canada. The resulting synthetic parcels are compared statistically to the real-world parcels associated with the blocks from the database. The hope is that use of statistics and a comprehensive block type classification will allow a set of rules to be defined for best matching block subdivision algorithms to block types. A conservative goal of this research is to offer guidelines to users of block subdivision algorithms for preparing missing current or future parcel data for integrated models. A more ambitious goal is to define a set of rules that could be incorporated into an automatic block subdivision program: one that automatically executes the appropriate algorithm for the block type to be subdivided. This could then be incorporated into a future integrated model that simulates land subdivision processes.

The literature review describes previous developments in block subdivision algorithms and in methods used to test their performance. The methodology section outlines the block type classification used, as well as the statistical tests used to evaluate the algorithms' performance. The results show how each algorithm performed on each block type, and a discussion section attempts to outline a set of relationships that define which algorithm is better for which block type. The paper is finished with a few additional remarks and directions for future research in the conclusion.

2. Literature Review

This literature review has two parts. The first describes the literature relating to the history and development of block subdivision algorithms. The second describes how the success of block subdivision algorithms has been evaluated in the past.

2.1. Block Subdivision Algorithms

The idea to use computer programming to model patterns in the urban environment originates with Parish and Müller (2001). They adapted the use of L-systems, which had been successfully used to generate realistic trees in computer graphics programs, to the generation of road and highway networks. Parish and Müller (2001) also demonstrated how such systems could be extended to subdivide land into lots, and create the appropriate geometry for buildings on the lots. Their land subdivision algorithm recursively subdivides blocks along the longest pair of approximately parallel edges, until parcel sizes are under a user-specified threshold area. One of the disadvantages of this algorithm is that parcels with no street access are deleted, leaving holes in the middle of blocks. This method also produces some irregularly shaped parcels. It is worth noting that this appears to fit the description of the block subdivision algorithm used in the integrated modeling system, PECAS (Hunt and Abraham 2009), although details on the exact algorithm used aren't available.

To Parish and Müller's recursive algorithm, Weber et al. (2009) incorporate a varying maximum area threshold depending on a parcel's land use type. More recently, Vanegas et al. (2012) implemented an Oriented Bounding Box (OBB) algorithm as a method for recursively splitting street blocks into parcels. Use of the OBB produces more regularly shaped parcels, and ensures a maximum number are oriented parallel to an adjacent street. Their algorithm also tries to ensure street access by splitting the bounding box along either the longest or widest edge.

Vanegas et al. (2012) also introduced the Straight Skeleton (SS) algorithm. It is based on the straight skeleton shape, formed by collapsing the edges of a polygon inward and tracing the intersection points of each set of edges according to Aicholzer's motorcycle algorithm (Aicholzer and Aurenhammer 1995). This shape is then split at regular intervals determined by a user-specified parcel width. Diagonal edges are then shifted to be perpendicular to roads. Similar to the Straight Skeleton algorithm, the Offset Subdivision algorithm contains an additional parameter specifying the distance to set the far edge of the parcels from the street. This produces a perimeter-block design; one whose parcels surround the outer edge of the block and contain a large central parcel typically occupied by a school or park. All algorithms developed by Vanegas et al. (2012) use block-level descriptive parcel parameters specified by the user, such as minimum parcel area and width.

The above algorithms were developed for City Generation Engines, such as CityEngine and UrbanCanvas. City Generation Engines can structure the forecasts of integrated models, namely future data on population, jobs and buildings, into plausible 3D cities. They also support manual editing of an urban system for more localized planning (Vanegas et al. 2012). In this context, development in an area is simulated according to a predefined process. First a road network is grown according to a road generation algorithm, then a block subdivision algorithm is executed to generate parcels, and finally a rule file determines the type of building to place on each parcel. In contrast, Wickramasuriya et al. (2011) developed a single algorithm to generate both roads and parcels within a block. Their program overlays four different orthogonal grids onto an area of land, selects the one that maximizes the number of parcels or minimizes the length of roads, and then clips the grid to the land's boundaries. Such a system serves as a stand-alone land subdivision tool for use by land developers or urban planners. Being modular and open-source, it could also be incorporated into integrated land use transportation modelling systems.

Building on the work of Wickramasurya et al. (2011), Dahal & Chow (2014) developed the ArcGIS Parcel Divider python toolset, which creates new roads and parcels on previously undeveloped tracts of land. The toolset contains six different algorithms, two of which can be used on any block type and the others on blocks with specific shapes such as T, L, or cul-de-sacs. Of the two more general algorithms, Generalized Parcel Divider 1 is designed for any block type while Generalized Parcel Divider 2 "...tends to generate block pattern with Manhattan-style street network." (p. 6)

Generalized Parcel Divider 1 (GPD1) uses a combination of recursive binary subdivision and grid drawing. First the algorithm recursively subdivides a land tract's oriented bounding box until the width of the bounding box is <= 2.5 times the user specified average parcel length. In a later step, the contours of this final bounding box are turned into roads. A series of grid lines are drawn within the bounding box, perpendicular to its longest edge and at intervals determined by the user-specified average parcel width. Grid lines perpendicular to the shortest edge of the bounding box are then drawn at intervals determined by the user-specified average parcel length. Undersized parcels are merged and the resulting grid is then clipped to the boundaries of the input land area. Generalized Parcel Divider 2 is similar, but skips straight to the grid drawing steps. In so doing, it is nearly identical to Wickramasuriya et al.'s (2011) algorithm.

2.2. Testing Block Subdivision Algorithms

In the presentation of their algorithms, developers typically test their algorithm's ability to generate parcel patterns similar to observed, real-world counterparts. The method used across all studies to determine whether an algorithm performs well, is to subdivide a block with the algorithm and then to compare the resulting parcels to their observed counterparts in a parcel dataset. Each study has, however, used its own methods for selecting test sites, and for comparing the two sets of parcels.

With respect to site selection, each study tested their algorithms on a few sites, chosen to represent distinct types of areas with block characteristics that are expected to affect the algorithms' performance. Vanegas et al. (2012) selected three sites based on land use type and density, block shape and parcel variability. One site was described as "...a mixed-use suburban area composed of rectangular blocks with both straight and curved edges...The set of blocks exhibits significant variability in both the area, the aspect ratio, and the minimum width of the parcels." (p. 9). Wickramasurya et al. (2011) classified sites into two types: one with parallel road and parcel orientations and uniformly shaped parcels, and the other with varying road and parcel orientations and shapes. Since the toolset of Dahal & Chow (2014) has some algorithms designed for particular block shapes (i.e. L-, and T-shapes) they test these algorithms on examples of blocks with these specific characteristics. With respect to their two more general algorithms (GPD1 and GPD2), GPD1 was tested on one site of an irregular and another of a regular shape, while GPD2 was tested visually on a site with an irregular shape.

With respect to the comparison of simulated to real-world parcels, Wikramasuriya et al. (2011) used t-tests and correlation coefficients to statistically compare the number of lots and mean lot sizes of the two sets of parcel distributions. They also compared the standard deviations of the sets of distributions and their mean shape index (MSI) using a standard error calculation. Vanegas et al. (2012) pooled the parcels generated by the OBB, SS and Offset Subdivision algorithms into a single sample, before comparing them to the observed parcel distributions. The statistical component of their method involved overlapping the frequency distributions of metrics of the simulated and real-world parcels to conduct a visual comparison of their similarities. Dahal & Chow (2014) compared the total number of lots and mean lot size of each site using a calculation of standard error ($[Z_{modeled} - Z_{reference}]/Z_{reference} \times 100$).

Given the different methods to test the algorithms, each study also came to different types of conclusions about its algorithms. Vanegas et al. (2012) found that their algorithms produced parcels with similar frequency distributions of metrics as those observed, as well as similar spatial distributions of metrics depicted in color coded maps. Furthermore, all generated parcels were found to have dimensions and aspect ratios that were adequate for containing buildings. Wickramasuriya et al. (2011) found that their algorithm generated parcels with statistically similar area distributions and counts for the first, more regular block type, but failed to do so for the second, more irregular block type. On the first block type, it also produced parcels of highly irregular shapes and sizes adjacent to curved or irregular block boundaries. Dahal & Chow (2014) found their algorithms to produce unrealistically uniform parcel shapes and sizes, with similar widths but longer lengths than those observed. The total number of lots was similar for all sets of parcels.

The above research focused on developing algorithms for generating parcels and also tested their performance on different block types. Each study, however, used a different method for classifying blocks, as well as a different method of testing its algorithms' performance. As a result, it is difficult to perform a meta-analysis of the results, to determine whether or not there is an algorithm that performs better than others, on any given block type. To fill in this gap, this study selects three of the most recently developed general algorithms described in the literature and uses them to generate parcels on the same set of real-world blocks. Before choosing blocks on which to test the algorithms, a standard block classification system was adopted, based on a commonly used road network classification of urban form. Examples of each block type from this classification system were then selected as test cases on which to apply the three algorithms, thus allowing a common basis of comparison.

3. Methodology

The general methodology adopted was to apply three block subdivision algorithms to the same sites, selected as examples of the block type categories, and to perform statistical tests comparing the characteristics of the simulated parcels with their real-world counterparts. The rest of this section describes: how the algorithms themselves and required input parameters were chosen, the block type classification used, and how the sites were selected; what parcel characteristics were compared and what statistical tests used to compare them.

3.1. Selection of Algorithms

In order to conduct a comparison of block subdivision algorithms, a number of candidate algorithms were available. As described in the literature review, there were: the original Parish and Müller algorithm (2001) and its more recent incarnation (Weber et al. 2009); the algorithm used in the PECAS model (Hunt and Abraham 2009); Vanegas et al.'s oriented bounding box (OBB) and straight skeleton (SS) algorithms (2012); Wickramasuriya et al.'s algorithm (2011); and Dahal and Chow's (2014) algorithms. When selecting the candidate algorithms, a number of criteria were used. First, an explicit and detailed description of the algorithm needed to be documented in the academic literature. Because we could not find an explicit and detailed description of the algorithm used in PECAS, we did not include this algorithm. Second, only "general algorithms" were considered in the analysis. The term "general algorithm" is used to distinguish them from algorithms specifically designed to subdivide a particular type of block. For example, the Offset Subdivision algorithm (Vanegas et al. 2012) reproduces a perimeter block design and one could safely assume that it is the best algorithm for this type of block. Similarly, the Cul-de-sac Creator algorithm (Dahal and Chow 2014), could be expected to best subdivide blocks at cul-de-sacs. The more general algorithms were chosen for this study because there is a degree of ambiguity about their performance in relation to each other and to different block types. As a result, a few of Dahal and Chow's algorithms, namely, Cul-De-Sac Creator, L-Shaped Parcel Divider, T-Shaped Parcel Divider, Multi Family Parcel Divider (for multi-family housing lots), Divider with Inner Roads (for blocks with inner looped roads) were not considered in the analysis. Finally, algorithms with documented weaknesses were also removed from consideration. This eliminated Parish and Müller's (holes within blocks and irregularly shaped parcels), Wickramasuriya et al.'s and Dahal and Chow's GPD2 (parcels with irregular shapes and orientations within irregular blocks) algorithms. This left the OBB, SS and GPD1 algorithms as candidates to test.

The application of the OBB and SS algorithms was done through ESRI's implementation in CityEngine. Application of the GPD1 algorithm was done in ArcGIS with the Parcel Divider Toolset after modifying the code to prevent it from generating roads within input blocks and enabling it to read subdivision parameters that vary from block to block.

3.2. Description of Input Parameters

Each algorithm has a different set of input parameters whose values must be specified by the user. For the purpose of this study, input parameters were classified into two types: 1) those which determine the characteristics of the resulting parcels, i.e.: deterministic parameters, and 2) those which constrain the characteristics of the resulting parcels, i.e.: constraint parameters. For example, the width parameter functions as a constraint in the OBB algorithm and as a deterministic parameter in the GPD1 and SS algorithms. In the case of OBB, the width parameter determines the width below which no more subdivision occurs. This leads to many possible width values of the resulting parcels, and so this parameter can be said to be non-deterministic. On the other hand, for SS and GPD1, the input width determines the width of the resulting parcels (before slivers are merged with non-sliver parcels), by drawing a split line at intervals equal to it. To account for this difference, the input width for the OBB algorithm was set as:

(1) $$W_{min} = W_{avg} - 1 \times sd_w$$

Where W_{min} is the minimum width a parcel can take, W_{avg} is the average width of parcels for the block, and sd_w is the standard deviation of parcel widths for the block. For the SS and GPD1 algorithms, the average parcel width by block is the width input value.

This convention was varied slightly for loop & lollipop road networks, with curved blocks and highly irregular parcel shapes. Since the parcel width parameter is difficult to compute, let alone conceptualize, the width of the Minimum Bounding Rectangle of the parcel is used instead. For curved or irregular parcels, the Minimum Bounding Rectangle width appears to be an overestimation of the actual street frontage of the parcel. To compensate for this, for blocks with extreme curves or irregular angles (loop & lollipop) the minimum width parameters were set at 1 standard deviation below the others. That is, for SS and GPD1:

(2) $$W_{input} = W_{avg} - 1 \times sd_w$$

and for OBB:

(3) $$W_{input} = W_{avg} - 2 \times sd_w$$

Where W_{input} is the width input value, W_{avg} is the average width of the bounding box for all the parcels in the block, and sd_w, is the standard deviation of the average width of the bounding box for all the parcels in the block.

For the Straight Skeleton and Oriented Bounding Box algorithms, the width input value is enhanced by a split irregularity parameter, which displaces the parcel's split line from its default position to create less uniform parcels. This displacement distance is sampled proportionally from a distribution defined by a mean equal to the algorithm's input width and a variance equal to 3 times the irregularity parameter. To populate this parameter for the SS algorithm, the blocks with smallest and largest width standard deviations were determined for the entire sample (0 and 12 respectively). The blocks' width standard deviations were then converted to a scale between 0 and 1; the range of this input parameter. The maximum irregularity parameter value (ie: 1) was divided by the maximum width standard deviation (ie: 12) to give a conversion factor of 0.083. This factor was then multiplied by each of the blocks' standard deviations to give their input irregularity values, as follows:

(4) $$\omega_{input} = \left(\omega_{max}/sd_{\omega_{max}}\right) \times sd_{\omega}$$

Where ω_{input} is the input irregularity, ω_{max} is the maximum input irregularity (1), $sd_{\omega_{max}}$ is the maximum width standard deviation (12), and sd_{ω} is the block's width standard deviation. An exception was made for block types with uniform parcel sizes, where the irregularity was set at 0. By definition, these parcels are highly regular and most of the variability in parcel width appears to be a result of irregularities in block shape, rather than an intentional design feature. Further varying these widths in the synthetic parcels through an irregularity parameter was found to have overestimated the width variance.

Since the OBB algorithm's width parameter is a constraint, there is already a degree of variability in the parcel widths it produces. As a result, the split irregularity parameter was generally set at 0, unless the standard deviation of the block's parcel widths was >= 10, in which case it was set at 0.03.

The other input parameters, namely, minimum and maximum area, are constraint parameters and the following maximum and minimums are used:

(5) $$A_{max/min} = A_{avg} \pm 2 \times sd_A$$

Where $A_{max/min}$ are the maximum and minimum values (e.g. of parcel area), A_{avg} is the average value for the parcels in a block, and sd_A is the standard deviation of the average value for the parcels in a block. One exception was for the GPD1 minimum area parameter that was set at:

(6) $$A_{min} = A_{avg}/2$$

Where A_{min} is the minimum parcel area and A_{avg} is the average area of parcels in the block. The minimum area was left open in the paper and it was thought that this option would result in a majority of parcel areas around the mean. It is worth noting that these methods for arriving at the deterministic and constraint parameters were chosen from several trials, based on visual and sometimes statistical analysis.

Finally, there were some required inputs (e.g. length in the GPD1 algorithm) for which there was no ambiguity in the value of the parameter required. For example, the GPD1 algorithm required an average value for the length of the parcel, but this is consistent with the length of the parcel's bounding box. As a result, average parcel length could be used. The *Force street access* parameter required by the OBB algorithm was set to always ensure street access, a parameter value of 1, in light of the general requirement that all residential parcels have access to a road. These input parameters are summarized for each algorithm in the tables below under the actual parameter names used in their respective programs.

TABLE 1 Input parameters used in the SS algorithm

Input parameter	Type	Road network type				Rationale
		Gridiron	Fragmented Grid	Warped Grid	Loops & Lollipops	
lotAreaMin	Constraint	$A_{avg} - 2 \times sd_A$	$A_{avg} - 2 \times sd_A$	$A_{avg} - 2 \times sd_A$	$A_{avg} - 2 \times sd_A$	Used in paper
lotWidthMin	Deterministic	W_{avg}	W_{avg}	W_{avg}	$W_{avg} - 1 \times sd_W$	Used in paper, Experimentation
irregularity	Constraint	$0\ or\ 0.038 \times sd_W$	$0\ or\ 0.038 \times sd_W$	$0\ or\ 0.038 \times sd_W$	0	Trial and Error

TABLE 2 Input parameters used in the GPD1 algorithm

Input parameter	Type	Road network type				Rationale
		Gridiron	Fragmented Grid	Warped Grid	Loops & Lollipops	
Width	Deterministic	W_{avg}	W_{avg}	W_{avg}	$W_{avg} - 1 \times sd_W$	Used in paper, Experimentation
length	Constraint	L_{avg}	L_{avg}	L_{avg}	L_{avg}	Used in paper
AvLotSize	Constraint	A_{avg}	A_{avg}	A_{avg}	A_{avg}	Used in paper
sizeTo Merge	Constraint	$A_{avg}/2$	$A_{avg}/2$	$A_{avg}/2$	$A_{avg}/2$	Experimentation

TABLE 3 Input parameters used in the OBB algorithm

Input parameter	Type	Road network type				Rationale
		Gridiron	Fragmented Grid	Warped Grid	Loops & Lollipops	
lotAreaMin	Constraint	$A_{avg} - 2 \times sd_A$	$A_{avg} - 2 \times sd_A$	$A_{avg} - 2 \times sd_A$	$A_{avg} - 2 \times sd_A$	Used in paper
lotAreaMax	Constraint	$A_{avg} + 2 \times sd_A$	$A_{avg} + 2 \times sd_A$	$A_{avg} + 2 \times sd_A$	$A_{avg} + 2 \times sd_A$	Used in paper
lotWidthMin	Constraint	$W_{avg} - 1 \times sd_W$	$W_{avg} - 1 \times sd_W$	$W_{avg} - 1 \times sd_W$	$W_{avg} - 2 \times sd_W$	Used in paper, Experimentation
forceStreet Access	Deterministic	1	1	1	1	Experimentation
irregularity	Constraint	[0,0.3]	[0,0.3]	[0,0.3]	0	Experimentation

3.3. Block Type Categorization and Test Site Selection

The block type classification used in this study is based on a classification of local road networks developed by Southworth & Owens (1993). This classification has since been used in a number of studies where local road network type categories are required, both quantitative (Rifaat and Tay 2008; Burton et al. 2011) and qualitative (Tasker-Brown and Pogharian 2000; Garde 2008; Sandalack et al. 2013). Southworth & Owens classified residential road networks into five categories at the highest level. These include: gridiron, fragmented parallels, warped parallels, loops and lollipops and lollipops on a stick. The categories are based on the evolution of planning paradigms for local road networks from the 1900s – 1980s and represent a progression from grid, highly connected types to dendritic, highly disconnected types. Schirmer (2010) developed a classification of parcel patterns including two categories for residential blocks: Residential 1, containing parcels of more uniform size and Residential 2, containing parcels of variable sizes. Taking into account these different classifications, the block type classification adopted in this study consisted of two dimensions. The first identified the road network type that defined a group of blocks and determined the range of possible block shapes within it. The second dimension identified the residential block type, namely, Residential 1 (with uniform parcel sizes) or Residential 2 (with variable parcel sizes). Figure 1 shows the seven block types and example sites selected from Montreal's parcel database to represent them. To select the final sites used in this study, all residential road network patches within the CMM were classified into one of the four road network types or else got assigned to a mixed category. Next, 30 test sites were selected for each road network type to encompass blocks most characteristic of that category. Parcel pattern types were identified from a parcel database. The fifth local road network type, namely, lollipops on a stick, was omitted from the study since it doesn't consist of enclosed blocks surrounded by roads; the kind of shape that existing algorithms are capable of subdividing.

3.4. Description of Comparative Tests

The simulated parcels were compared with their observed counterparts on the basis of three different metrics that are the most commonly used in the literature (see literature review): area, mean shape index (MSI) and width. Area was chosen as an indicator of parcel size, MSI as an indicator of parcel shape (FRAGSTATS: McGarigal and Marks 1994) and width as an indicator of the amount of street frontage of the parcels. The t-test was used to determine whether or not the mean metrics of the simulated parcels were statistically similar to the mean metrics of their observed counterparts. The t-test seemed like the most appropriate test and was also consistent with the literature. The Kolmogorov-Smirnov test was used to test whether or not the distributions of metrics of the simulated parcels were statistically similar to the distributions of their observed counterparts.

The distributions of observed and simulated parcel metrics weren't generally normally distributed, yet the t-test is technically designed to compare two t-distributions. However, for any sufficiently large sample ($n > 30$), the central limit theorem states that the mean of a randomly sampled variable drawn from any distribution will itself be normally distributed (Griffiths et al. 1993). Furthermore, this distribution of means will have the same mean as the original sample itself. Therefore, the t-test is used to compare the means of these two distributions.

The Kolmogorov-Smirnov test is a non-parametric test, useful in determining whether two random variables could have been drawn from the same distribution. It computes the empirical cumulative distribution function (ecdf) of a sorted set of sample values, to demonstrate the percentage of the data below each of the values (Crawley 2013). The test statistic, or D value, gives the greatest deviation in percentage below a given value between the two ecdfs being compared. Where a distribution of metrics has duplicate values or ties, the KS-test generates a warning in R that the p-value is approximate. As a result, a bootstrap version of the KS-test (ks.boot), developed by Jasjeet S. Sekhon at UC Berkeley, was used that returns exact p-values in all cases.

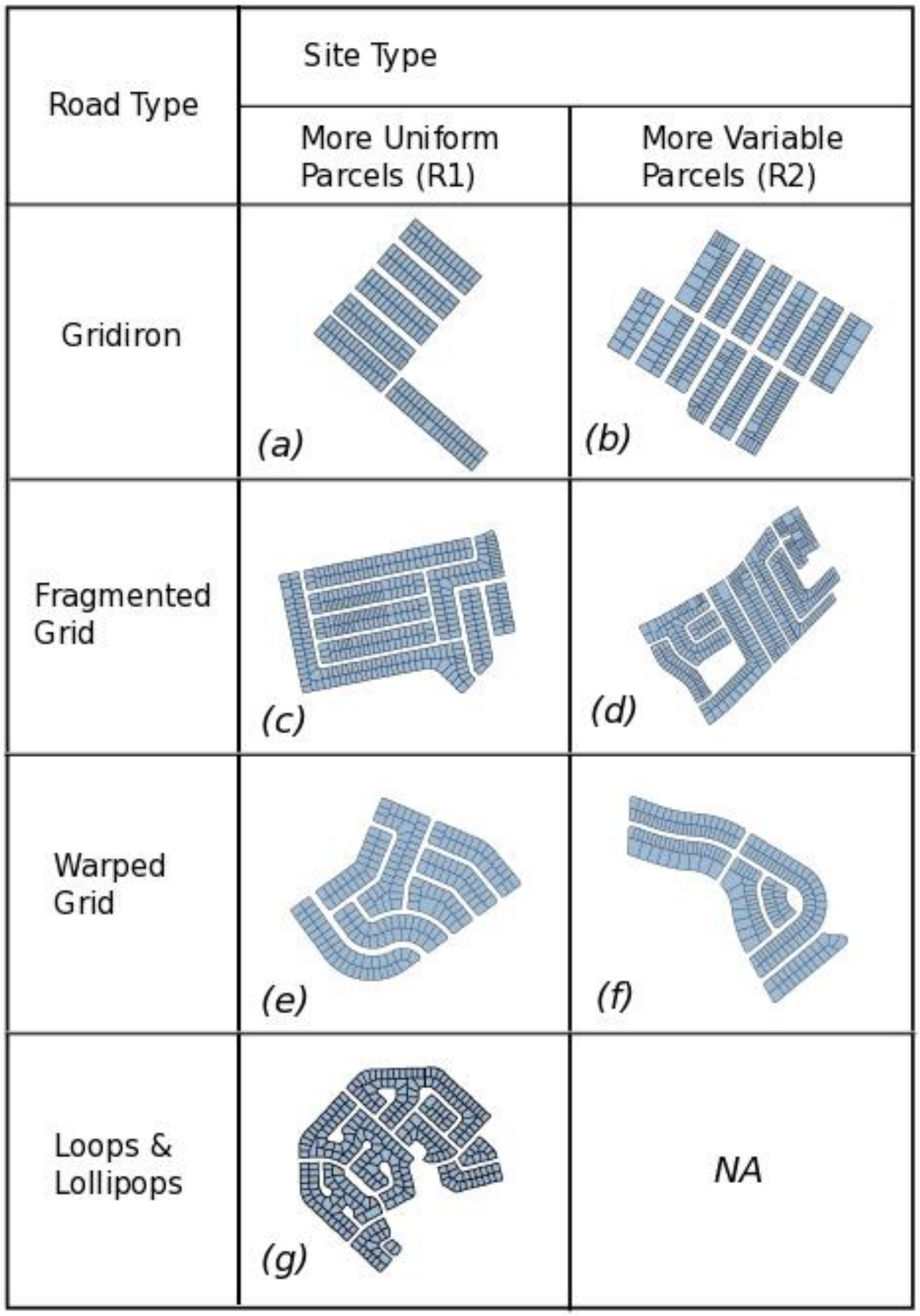

Fig. 1 Block types and sites selected from Montreal's parcel database (Ministère des Ressources naturelles et de la Faune 2009) to represent them. (a) Rectilinear grid - more uniform parcels, (b) Rectilinear grid - less uniform parcels, (c) Fragmented grid - more uniform parcels, (d) Fragmented grid - less uniform parcels, (e) Warped grid – more uniform parcels, (f) Warped grid – less uniform parcels, (g) Loop & lollipop road network – more uniform parcels

4. Data

The observed parcels, to which the simulated parcels are compared, come from the 2009 cadastral data of Québec purchased from the Ministère des Ressources naturelles et de la Faune (2009) with the funding from the Canadian Foundation for Innovation. The database contains parcel shapefiles that cover the entire territory of the Montreal Metropolitan Community (Communauté métropolitaine de Montréal - CMM), excluding some rural areas where data were missing. A road network (DMTI Spatial, CanMap®RouteLogisitics, version 2013.3) was overlaid onto the parcel fabric and used to identify sites from the four road type categories.

The blocks were derived from the most accurate input data available under usual circumstances, namely, the presence of a road network. The road widths were estimated from the number of lanes multiplied by 3.5 m (3.83 yd), a common lane width. The blocks were created in CityEngine by filling in the negative space of the road network. In total 222 sites were selected consisting of between 30 and 35 sites per type, each site containing 3-9 blocks.

5. Results

The null hypothesis of the t-test is that the difference in mean area (or other metric) between the simulated parcels and observed parcels is 0. A p-value less than 0.05 implies that the two means are different and that the algorithm failed to reproduce parcels with statistically similar mean metric. In contrast, a p-value greater than 0.05 doesn't indicate the two means are the same, only that they cannot be concluded to be different. The null hypothesis of the ks-test is that the distributions of metrics for both sets of parcels were sampled from populations with identical distributions. Again, a p-value less than 0.05 implies that the two parent distributions are different and the algorithm failed to produce parcels with a statistically similar distribution of a given metric.

The results of these statistical tests, aggregated by site type, can be found in tables 4-7. Here, each row represents a different site type and each column another algorithm/metric combination. The numbers represent the proportion of non-rejected null hypotheses out of total number of t-tests (or ks-tests). Since the numbers are aggregated by site type, the total number of tests is equal to the number of selected sites per site type and is between 30 and 35. The shaded values indicate the algorithm with highest proportion of non-rejected null hypotheses for a given metric and site type (grey) or for all metrics on average for a given site type (dark grey). Ties are shaded in light grey.

For example, in the first entry in table 4- parcel areas for the gridiron, uniform site types produced by the Straight Skeleton algorithm- 3% of the sites had synthetic parcel areas whose average couldn't be shown to be statistically different from their observed counterparts. Similarly, for the same algorithm and metric, but for the gridiron, non-uniform site types, 7% of the sites had average synthetic parcel areas that couldn't be shown to be statistically different from their observed counterparts in the parcel database. For the gridiron non-uniform site types, 80% of the sites had average mean shape indices (msi) that couldn't be shown to be different from their observed counterparts. Furthermore, this was the highest proportion for this site type and metric so it was highlighted in grey. The average column contains the proportion of sites with non-rejected null hypotheses averaged over all three metrics. This was meant to be a measure of how well each algorithm performed on average for each site type. For example, the Straight Skeleton algorithm produced average parcel metrics that can't be shown to be different from their observed counterparts in 11% of gridiron, uniform sites. For gridiron, non-uniform sites, the same algorithm produced average parcel metrics that couldn't be shown to be statistically different from their observed counterparts 13% of the time. The average proportion of the GPD1 algorithm for this site type was highlighted in dark grey, because it has the highest average proportion of non-failed t-tests (ie: 48%).

The results of the ks-tests are included in table 5 and they indicate that the algorithms don't generally reproduce statistically similar distributions of metrics to their observed counterparts. For example, all sites of type gridiron, uniform parcels subdivided by the Straight Skeleton algorithm were found to have statistically different area distributions to their corresponding observed sites. The GPD1 algorithm also produced parcels whose area distributions were always statistically different from their realistic counterparts in all gridiron uniform sites. Again, the highest proportions of non-rejected null hypotheses for each site type and metric are shaded in grey.

TABLE 4 Proportions of non-rejected null hypotheses for t-Tests for Different Metrics by Algorithm and Site Type[a]

Site type	SS				GPD1				OBB			
	area	width	msi	avg	area	width	msi	avg	area	width	msi	avg
Gridiron - R1	0.03	0.13	0.17	0.11	0.03	0.17	0.13	0.11	0.20	0.30	0.30	0.27
Gridiron - R2	0.07	0.40	0.80	0.42	0.13	0.53	0.77	0.48	0.40	0.50	0.47	0.46
Fragmented - R1	0.03	0.09	0.27	0.13	0.00	0.09	0.30	0.13	0.27	0.21	0.15	0.21
Fragmented - R2	0.12	0.33	0.67	0.37	0.12	0.61	0.52	0.41	0.30	0.24	0.27	0.27
Warped - R1	0.00	0.15	0.24	0.13	0.00	0.09	0.18	0.09	0.18	0.29	0.12	0.20
Warped - R2	0.03	0.00	0.60	0.21	0.00	0.17	0.43	0.20	0.20	0.53	0.17	0.30
Loops, lollipops-R1	0.09	0.84	0.03	0.32	0.22	0.06	0.00	0.09	0.72	0.00	0.19	0.30

[a] Shaded values indicate highest proportions per metric (grey), on average for all metrics (dark grey), and ties (light grey).

TABLE 5 Proportions of non-rejected null hypotheses for ks-Tests for Different Metrics by Algorithm and Site Type[a]

Site type	SS				GPD1				OBB			
	area	width	msi	avg	area	width	msi	avg	area	width	msi	avg
Gridiron - R1	0.00	0.00	0.00	0.00	0.00	0.00	0.03	0.01	0.00	0.03	0.00	0.01
Gridiron - R2	0.00	0.07	0.10	0.06	0.00	0.00	0.03	0.01	0.03	0.07	0.07	0.06
Fragmented - R1	0.00	0.00	0.03	0.01	0.00	0.00	0.09	0.03	0.00	0.00	0.00	0.00
Fragmented - R2	0.03	0.03	0.09	0.05	0.00	0.00	0.03	0.01	0.03	0.03	0.06	0.04
Warped - R1	0.00	0.00	0.09	0.03	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Warped - R2	0.00	0.00	0.13	0.04	0.00	0.00	0.03	0.01	0.03	0.03	0.00	0.02
Loops, lollipops-R1	0.16	0.13	0.03	0.10	0.09	0.00	0.00	0.03	0.00	0.00	0.00	0.00

[a] Shaded values indicate highest proportions per metric (grey), on average for all metrics (dark grey), and ties (light grey).

6. Discussion

One goal of this study is to help users of block subdivision algorithms come to a decision about which algorithm to use for which site type, in order to create the most accurate parcel data possible within integrated models. With that in mind, we can define a "better" algorithm as follows: A better algorithm for a particular site type is one that performed better than the other two. Better performance being measured by the proportion of non-rejected t-test null hypotheses of total number of sites, on average for all metrics for that site type. Because the null hypotheses of the ks-tests were mostly rejected, these results weren't incorporated into the conclusions. It's important to note that this doesn't necessarily mean the better algorithm produces non-statistically different average parcel metrics the majority of the time for a given site type; just that it does so more often than the others.

These results suggest that in all cases, there is an algorithm that performs better for a site type relative to the others. For loop and lollipop sites, the Straight Skeleton algorithm produced, on average, a non-statistically different average parcel metric in 32% of sites. This is a higher percentage than the GPD1 algorithm for the same site type (9% of sites) as well as the OBB algorithm (30% of sites). Similarly, the OBB algorithm performed better than the other two algorithms for the warped site types (both uniform and non-uniform, 20% and 30% probabilities of performing well respectively) and the gridiron (27% probability) and fragmented uniform site types (21% probability). The GPD1 algorithm performed better than the other two for the gridiron and fragmented non-uniform site types (41% and 48% probabilities of performing well). All of these "better" algorithms produced non-statistically different parcel metrics between 20 and 50% of the time. This suggests the algorithms can be further developed to reproduce parcels with similar characteristics to ones that are observed, assuming this is a desirable goal.

The ks-tests indicate that in general, the distributions of area, width and MSI of synthetic parcels aren't statistically similar to those of the observed parcels. With few exceptions, the Straight Skeleton algorithm produced the highest proportion of sites with non-statistically different distributions for all site types. This indicates that the SS algorithm more accurately reproduces distributions of parcel metrics than the other two, though it still does so infrequently.

Differences between synthetic and observed parcels may also be caused by inaccuracies in block shapes derived from the road network when compared with the observed ones from the parcel database. Input blocks with the spatial accuracy of those in the parcel database are not generally available when synthetic parcels are needed. However, as the input data approaches these ideal conditions, the algorithms should perform better and better. Such conditions can be brought about by higher accuracy in road network data. They can also potentially be brought about through the study of municipal guidelines, which determine the range of possible lane widths and sidewalk widths for a given area (Housing Regulations Database, Pioneer Institute Public Policy Research). Knowing these values can improve the representation of road widths, resulting in more accurate block shapes and sizes used in subdivision.

It seems that all the algorithms performed better for site types with non-uniform parcels than for ones with uniform parcels, often by orders of magnitude of 2 or higher. The difference in results between non-uniform and uniform site types was least extreme for the OBB algorithm. An example of a fragmented grid, uniform site and a fragmented grid, non-uniform site subdivided by each algorithm are shown in figures 2 and 3. The t-test results demonstrate that in the uniform site, no non-statistically similar parcel metrics are produced, while in the non-uniform site each algorithm produced 1 or 2 non-statistically different average parcel metrics. These sites are representative of the general trend in the results. The primary reason for this trend is surely that the variance in the indicators for the regular sites is lower than in the irregular sites, and as a result, it is easier to have a significant t-test than in the case of irregular sites. In this sense we can say the t-test is more forgiving in the case of irregular than regular block types. Another possible explanation is that the algorithms produced less uniform parcels where the blocks bend. These parcel irregularities would be more comparable to the non-uniform site types than the uniform ones. Furthermore, since the OBB algorithm recursively subdivides parcels, the parcels in an area are a function of the shapes produced in the previous recursion. In this case, the overall shape of the block would have less of an impact on the parcels than the other algorithms. It seems that the OBB produced more uniform parcels at bends and angles than the other algorithms, which would explain why the differences between results of non-uniform and uniform site types are less extreme here.

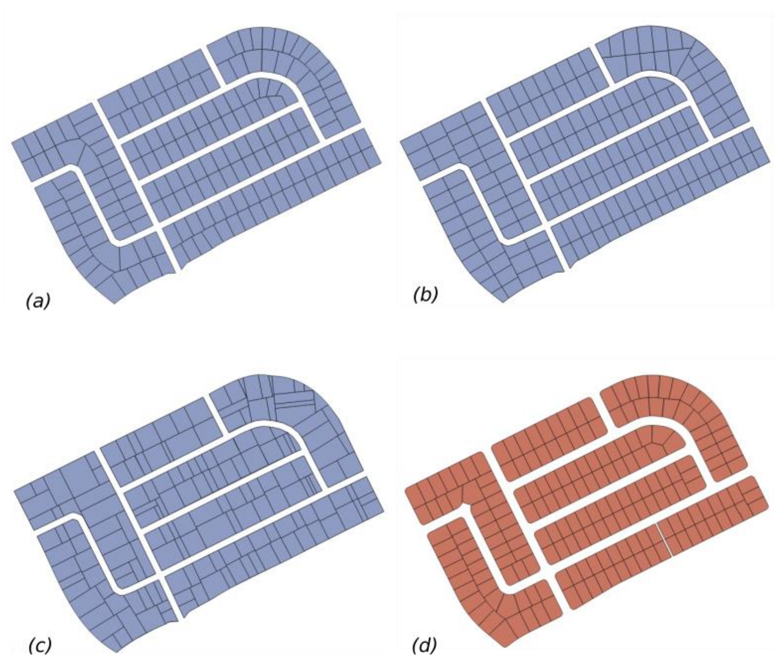

Fig. 2 Parcels generated from blocks of type Fragmented grid - uniform parcel sizes by the 3 different algorithms: (a) Straight Skeleton, (b) Generalized Parcel Divider 1, (c) Oriented Bounding Box, (d) Observed parcels in database. None of the algorithms produced any non-statistically different average parcel metrics.

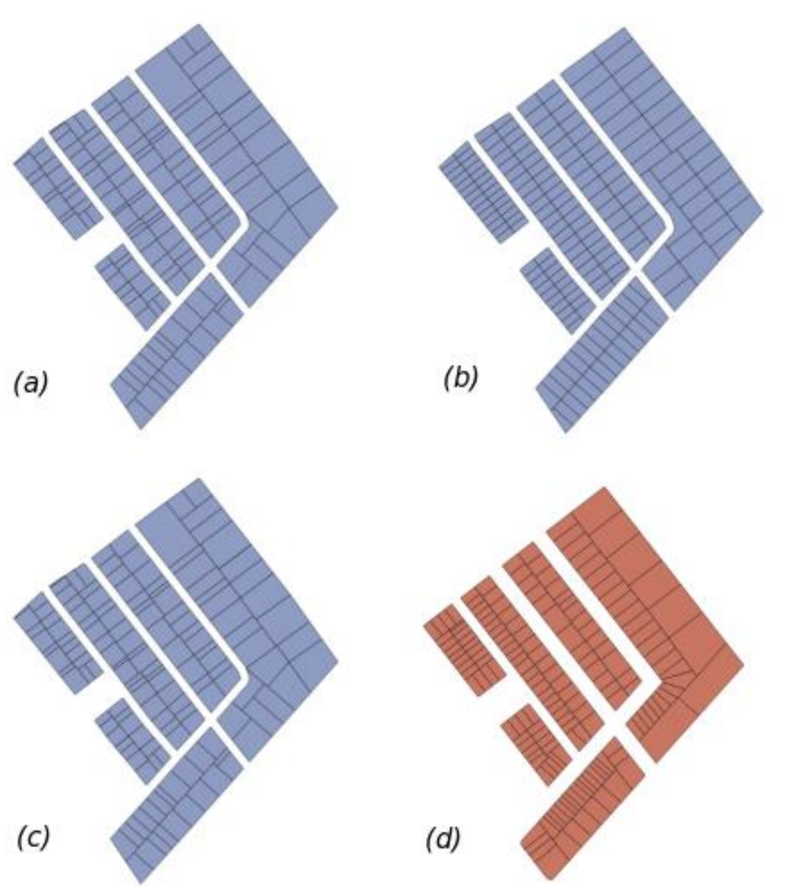

Fig. 3 Parcels generated from blocks of type Fragmented grid - variable parcel sizes by the 3 different algorithms: (a) Straight Skeleton (area: p=0.39, width: p=0.37, msi: p=0.65), (b) Generalized Parcel Divider 1 (area: p=0.49, width: p=0.43, msi: p=0.06), (c) Oriented Bounding Box (msi: p=0.26), (d) Observed parcels in database

One consistent trend is that certain algorithms tended to perform better for certain metrics. This is most apparent from the t-test results (table 4). For example, in 4 out of 7 site types, the SS algorithm produced the highest proportion of non-rejected null hypotheses for the MSI metric. The OBB algorithm performed better for the area metric (7 out of 7 site types) and width metric (4 out of 7 site types). Moreover, for site types where it was the better algorithm, the SS algorithm usually reproduced non-statistically different MSI metrics >50% of the time. These results make sense given the algorithm designs. For example, the SS algorithm was designed by looking at how planners approach the geometric problem of subdividing street blocks using hand drawn sketches. It's not surprising that the resulting parcel shapes are most similar to those observed. Unlike the other two, the OBB algorithm has two area input parameters that are constraints as opposed to deterministic, as well as a constraint width parameter, possibly an indication of why it most often produced more accurate parcel areas and widths.

This finding may be especially useful in light of the fact that planning policy constraints often correspond directly to parcels. A maximum parcel area is often set to regulate development density and a street frontage (ie: parcel width) is used to regulate density and aesthetic characteristics of a neighbourhood. The results of this study can be used to select a better algorithm for simulating development under such policies. For a development scenario that specifies a certain minimum parcel width, the OBB algorithm would most closely reproduce those widths in the synthetic parcels. For a scenario that specifies a certain minimum or maximum parcel area, the OBB algorithm would also be a better choice. On the other hand, the SS algorithm most accurately reproduces parcel shapes and patterns, which is most helpful with visualizing aesthetic characteristics of a future city.

These results not only suggest a better algorithm to use in each of these scenarios, but they also allow us to predict the probabilities that these algorithms will perform well. Testing on 30 sites allows us to form predictions about how likely the algorithms will perform well on members of the population of sites of that type. This is particularly helpful if the geometries they produce contribute to things that are ultimately measured, for example in estimates of the spatial distribution of future populations, which are partly a function of parcel layout. These results could help make best use of the available algorithms within integrated models for most accurately predicting population, as well as contribute to estimating uncertainties of these predictions.

Furthermore, qualitative studies can suggest which algorithm produces parcels that look most similar to their observed counterparts. However, that algorithm doesn't necessarily produce parcels with most similar counts, areas or widths. In fact, visual observation of the synthetic parcels produced in this study suggests they often don't. In the example below (figure 4), a visual inspection might lead one to conclude the SS algorithm reproduces parcels most similar to their observed counterparts. A t-test would however lead us to the conclusion that the OBB algorithm reproduces parcels that are not statistically different from their observed counterparts with respect to average width (p=0.19) and area metrics (p=0.86), while the GPD1 reproduced non-statistically different width (p=0.89) and MSI (0.06). The SS reproduces non-statistically different average MSI (p=0.25). Furthermore, a statistical comparison over multiple sites of this type would lead us to the conclusion that the OBB algorithm most often reproduces non-statistically different average parcel metrics for this site type (OBB – 30%, GPD1 - 20%, and SS - 21%). Several sites subdivided by the three algorithms are included for comparison in figures 4-7 below.

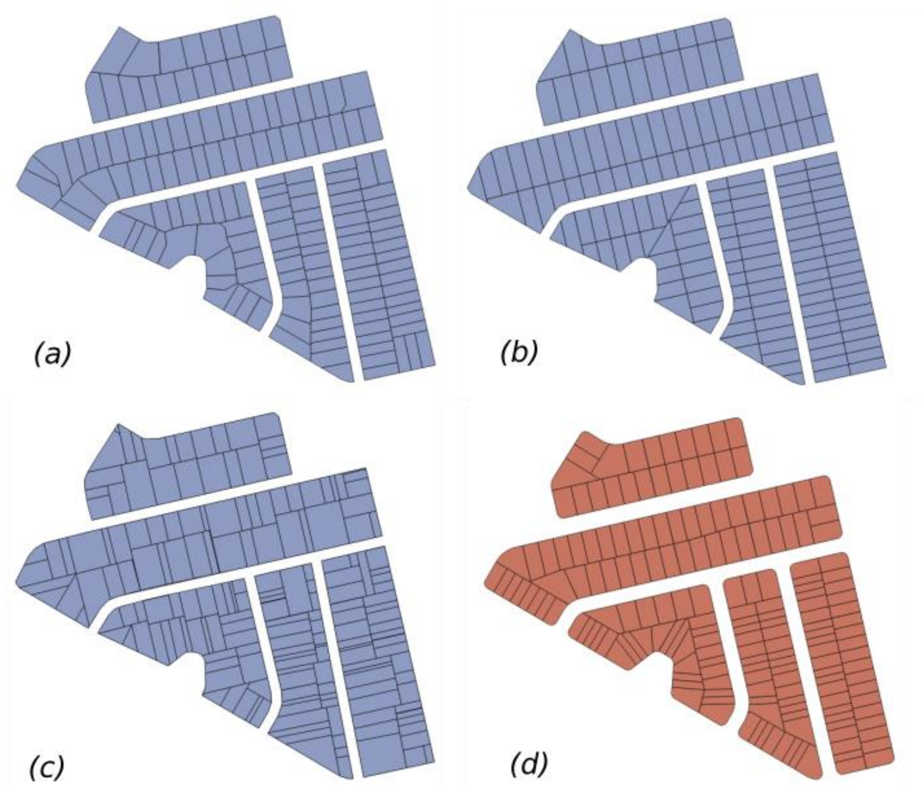

Fig. 4 Parcels generated from blocks of type Warped grid - variable parcel sizes by the 3 different algorithms: (a) Straight Skeleton (msi: p=0.25), (b) Generalized Parcel Divider 1 (width: p=0.07, msi: p=0.89), (c) Oriented Bounding Box (area: p=0.86, width: p=0.19), (d) Observed parcels in database

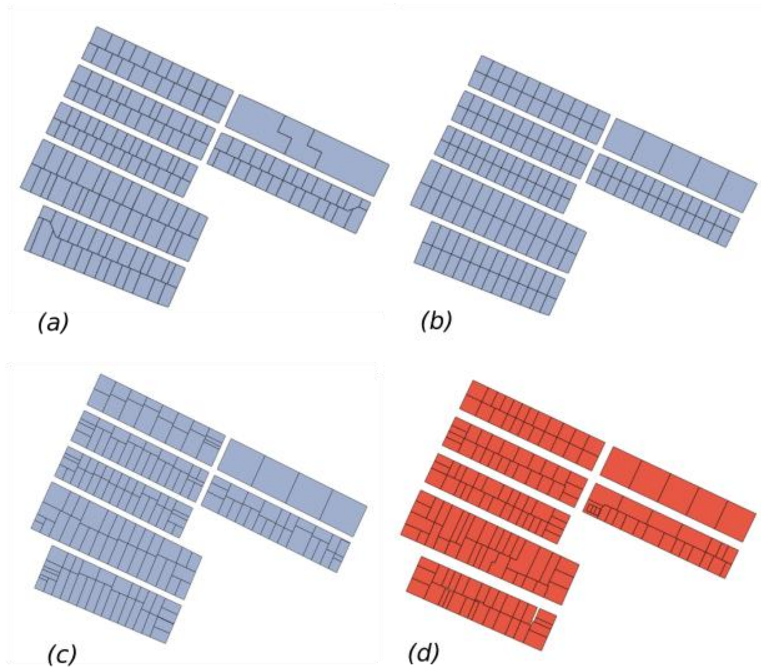

Fig. 5 Parcels generated from blocks of type Gridiron grid - variable parcel sizes by the 3 different algorithms: (a) Straight Skeleton (area: p=0.178, width: p=0.535, msi: p=0.415), (b) Generalized Parcel Divider 1 (area: p=0.139, width: p=0.356), (c) Oriented Bounding Box (area: p=0.308, width: p=0.322, msi: p=0.109), (d) Observed parcels in database

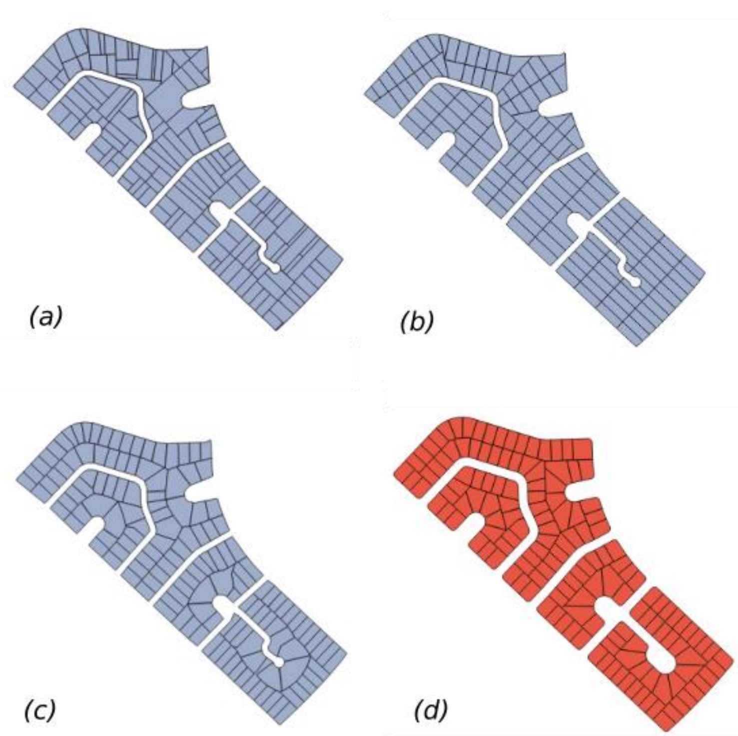

Fig. 6 Parcels generated from blocks of type Loops & Lollipops - uniform parcel sizes by the 3 different algorithms: (a) Straight Skeleton (width: p=0.869), (b) Generalized Parcel Divider 1, (c) Oriented Bounding Box (area: p=0.132), (d) Observed parcels in database

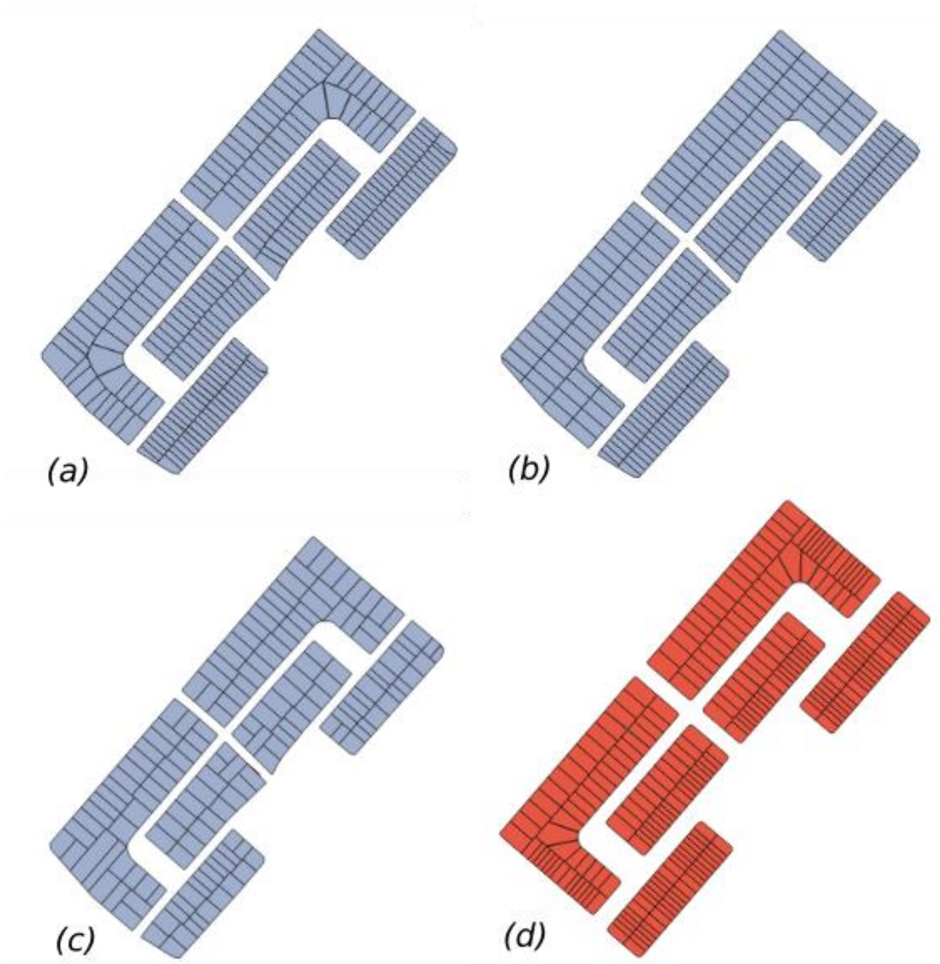

Fig. 7 Parcels generated from blocks of type Fragmented grid – variable parcel sizes by the 3 different algorithms: (a) Straight Skeleton (msi: p=0.09), (b) Generalized Parcel Divider 1 (width: p=0.06, msi: p=0.1), (c) Oriented Bounding Box, (d) Observed parcels in database

7. Conclusions

This study tested how well parcels simulated with three block subdivision algorithms compared with their observed counterparts from a parcel database from Montreal, Canada. In order to do this, it also presented a block type classification. By comparing the different block subdivision algorithms according to different metrics and across the same 30 sites of a given type, it has helped to work toward defining a relationship between block type and an algorithm better suited to subdividing it. Finally, by testing each algorithm on 30 sites per type, it presented a method for finding the likelihood this algorithm will perform well either on the whole or for a given metric. This has the potential to increase the accuracy of synthetic parcel data produced using existing algorithms.

The results suggest that the OBB algorithm more often produces non-statistically different parcels on warped (both uniform and non-uniform) grids as well as on gridiron and fragmented uniform sites. Meanwhile the SS algorithm more often produces non-statistically different parcels on loop and lollipop networks. The GPD1 algorithm more often produces non-statistically different parcels for gridiron and fragmented non-uniform sites. These better algorithms performed well between 20 and 50% of the time. In addition, the SS algorithm tends to produce most non-statistically different average parcel shapes and often does so the majority of the time, while the OBB algorithm does so for average parcel areas and widths. Assuming that it's a desirable goal to reproduce statistically similar parcel characteristics rather than parcels with realistic characteristics more generally, there is room for further development of these algorithms.

The variation in performance of the different algorithms suggests that a larger program that incorporates them all and executes the best one for a given block type or metric could stand to improve the spatial accuracy of parcels in input data and in simulations of urban expansion. An area of future research could be to develop a multinomial logit model for predicting the spatial distribution of road network types in a future residential area. Road generation algorithms could be used to generate roads networks of the predicted types. Such a model could then be integrated with the current findings to predict the future road network type, generate it, and then execute the better subdivision algorithm suited to its blocks. This would be a more accurate method for completely automating the block subdivision process.

8. Acknowledgements

This study was funded by the Canadian Research Chairs program, the Canadian Foundation for Innovation and the Agence métropolitaine de transport.

References

Waddell, P, Ulfarsson, GF, Franklin, JP and Lobb, J (2007) Incorporating land use in metropolitan transportation planning. Transportation Research Part A: Policy and Practice, 41(5): 382-410. doi: 10.1016/j.tra.2006.09.008

Borning, A, Waddell, P and Förster, R (2008) Digital Government: E-Government Research, Case Studies, and implementations. Springer US

Waddell, P (2009) Parcel-Level Microsimulation of Land Use and Transportation: The Walking Scale of Urban Sustainability. Workshop on Computational Algorithms and Procedures for Integrated Microsimulation Models. International Association for Travel Behaviour Research, Jaipur, India.

Xie, Y and Batty, M (2003) Integrated Urban Evolutionary Modeling UCL Working Paper Series, University College London London 68.

Schirmer, P (2010) Options and constraints of a parcel based approach to 'UrbanSimE'. Swiss Transportation Research Conference. STRC, Monte Verita/Ascona, pp 1-43.

Patterson, Z and Bierlaire, M (2010) Development of prototype UrbanSim models Environment and Planning B: Planning and Design, 37(2): 344-368.

Vanegas, CA, Kelly, T, Weber, B, Halatsch, J, Aliaga, DG and Muller, P (2012) Procedural Generation of Parcels in Urban Modeling. Computer Graphics Forum, 31(2): 681-690. doi: 10.1111/j.1467-8659.2012.03047.x

Wickramasuriya, R, Chisholm, LA, 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. doi: 10.1016/j.envsoft.2011.06.003

Dahal, KR and Chow, TE (2014) A GIS toolset for automated partitioning of urban lands. Environmental Modelling & Software, 55: 222-234. doi: 10.1016/j.envsoft.2014.01.024

Parish, Y and Muller, P (2001) Procedural Modeling of Cities. Proceedings of the 28th annual conference on Computer graphics and interactive techniques. ACM SIGGRAPH, Los Angeles, CA, pp 301-308. doi: 10.1145/383259.383292

Weber, B, Muller, P, Wonka, P and Gross, M (2009) Interactive Geometric Simulation of 4D Cities. EUROGRAPHICS, 28(2).

Aicholzer, O and Aurenhammer, F (1995) A Novel Type of Skeleton. Journal of Universal Computer Science, 1(12): 752-761.

Southworth, M and Owens, P (1993) The evolving metropolis: Studies of community, neighborhood, and street form at the urban edge Journal of the American Planning Association, 59(3): 271-287.

Rifaat, SM and Tay, R (2008) Effect of Street Pattern on the Severity of Crashes Involving Vulnerable Road Users Proceedings of the 18th Canadian Multidisciplinary Road Safety Conference Canadian Association of Road Safety Professionals, Whistler, British Columbia pp 1-15.

Burton, EJ, Mitchell, L and Stride, CB (2011) Good places for ageing in place: development of objective built environment measures for investigating links with older people's wellbeing BMC Public Health 11: 839-851. doi: 10.1186/1471-2458-11-839

Tasker-Brown, J and Pogharian, S (2000) Learning from Suburbia: Residential Street Pattern Design. Socio-economic Series. Canada Mortgage and Housing Corporation Ottawa, Canada, pp. 1-6.

Garde, A (2008) City Sense and Suburban Design: Planners' Perceptions of the Emerging Suburban Form. Journal of the American Planning Association, 74(3): 325-342.

Sandalack, BA, Alaniz Uribe, FG, Eshghzadeh Zanjani, A, Shiell, A, McCormack, GR and Doyle-Baker, PK (2013) Neighbourhood type and walkshed size. Journal of Urbanism: International Research on Placemaking and Urban Sustainability, 6(3): 236-255. doi: 10.1080/17549175.2013.771694

Griffiths, WE, Hill, RC and Judge, GC (1993) Learning and Practicing Econometrics John Wiley & Sons, Hoboken, New Jersey.

Crawley, MJ (2013) The R Book John Wiley & Sons, West Sussex, United Kingdom.