

Linear building pattern recognition in topographical maps combining convex polygon decomposition

Zhiwei Wei^{a,b*}, Su Ding^c, Lu Cheng^d, Wenjia Xu^{a,b}, Yang Wang^{a,b}, Lili Zhang^{a,b}

^aKey Laboratory of Network Information System Technology (NIST), Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100190, China;

^bThe Aerospace Information Research Institute, Chinese Academic of Sciences, Beijing 100190, China; ^cCollege of Environmental and Resource Science, Zhejiang A & F University, Zhejiang 311300, China; ^dSchool of Resource and Environment Science, Wuhan University, Wuhan 430079, China.

Address for correspondence: Zhiwei Wei. E-mail: 2011301130108@whu.edu.cn

Linear building pattern recognition in topographical maps combining convex polygon decomposition

Building patterns are crucial for urban form understanding, automated map generalization, and 3D city model visualization. The existing studies have recognized various building patterns based on visual perception rules in which buildings are considered as a whole. However, some visually aware patterns may fail to be recognized with these approaches because human vision is also proved as a part-based system. This paper first proposed an approach for linear building pattern recognition combining convex polygon decomposition. Linear building patterns including collinear patterns and curvilinear patterns are defined according to the proximity, similarity, and continuity between buildings. Linear building patterns are then recognized by combining convex polygon decomposition, in which a building can be decomposed into sub-buildings for pattern recognition. A novel node concavity is developed based on polygon skeletons which is applicable for building polygons with holes or not in the building decomposition. And building's orthogonal features are also considered in the building decomposition. Two datasets collected from Ordnance Survey (OS) were used in the experiments to verify the effectiveness of the proposed approach. The results indicate that our approach achieves 25.57% higher precision and 32.23% higher recall in collinear pattern recognition and 15.67% higher precision and 18.52% higher recall in curvilinear pattern recognition when compared to existing approaches. Recognition of other kinds of building patterns including T-shaped and C-shaped patterns combining convex polygon decomposition are also discussed in this approach.

Keywords: building patterns; convex decomposition; automate map generalization; computational geometry; visual perception theory.

1. Introduction

Buildings, as the main entities in urban areas, are important objects in topographical maps. The buildings arranged regularly will form patterns which are important local structures characterizing built-up spaces (Zhang et al., 2013). These patterns may have political, economic, cultural, and functional implications. Thus, building patterns can be

used in many involved applications, such as automatic map generalization (Ruas & Holzapfel, 2003; Steiniger, 2007; Zhang et al., 2013c; Renard & Duchêne, 2014; Zhao et al., 2020), semantic classification of urban scenes (Zhang et al., 2015; Du et al., 2019), prediction of building attributes (Niu et al., 2017; Rosser et al., 2019), and visualization of 3D city models (Mao et al., 2012; Zhang et al., 2013a; Pepe et al., 2013). It is therefore important to recognize building patterns automatically in topographical maps.

Building patterns are considered as visually salient structures (Du et al., 2015a). Many approaches are developed to recognize building patterns based on visual perception rules mainly involving proximity and similarity between buildings. For example, alphabet-like patterns such as Z-shaped and H-shaped patterns are recognized based on template-matching approaches (Yang, 2011; Du et al., 2015a; Gong et al., 2018). Collinear and curvilinear patterns are recognized with Gestalt principles based on different kinds of proximity graphs (Christopher & Ruas, 2002; Mao et al., 2012; Zhang et al., 2013; Gong et al., 2013; Wang & Burghardt, 2019a). Grid-like patterns are recognized based on refinement on a proximity graph or mesh idea (Anders, 2006; Gong et al., 2014; Wang & Burghardt, 2019b). With the development of machine learning, algorithms including random forest and graph convolutional network are also applied for building pattern recognition (He et al., 2018; Yan et al., 2018; Zhao et al., 2020). As proximity relation between buildings can be represented by different kinds of proximity graphs, the effectiveness to detect collinear building patterns with various proximity graphs is compared by Wei et al. (2018). The similarity between buildings can be defined based on their characteristics, such as size, orientation, and shape, and parameters to measure these characteristics have also been studied by many scholars (Agent, 1999; Duchêne, 2003; Basaraner & Cetinkaya, 2017).

However, buildings are considered as a whole in all the above approaches. Some

visually aware building patterns may fail to be recognized. Psychological studies have shown that human vision is a part-based system, i.e., a shape sometimes may be decomposed into parts for a visual task (Singh et al., 1999). For example, the building c on a street corner may be a combination of several buildings, a part of c may form a collinear pattern with the buildings along one side of the street by visual perception, e.g., (a, b, c_1) or (d, e, c_2) in Figure 1(a); and it is similar for building g and the collinear pattern (f, g_1, h) can also be aware by the visual perception in Figure 1(b). While these patterns will not be recognized using existing approaches if the buildings are considered as a whole. Furthermore, building shapes have become more and more complex with the development of cities, such as commercial buildings with podiums and annexes, multi-family buildings with different room semantics, and buildings with holes (Hu et al. 2021). This makes it challenging to describe the characteristics or relations of complex buildings in pattern recognition.

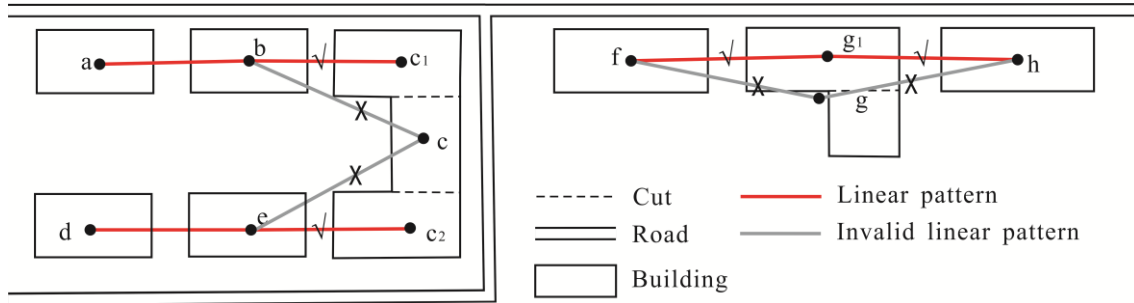


Figure 1. Pattern recognition combining convex polygon decomposition.

Convex shape decomposition is a widely used technique for object retrieval, classification, and pattern recognition in computer vision (Juengling & Mitchell, 2007). For example, handwritten digits and human body gestures are recognized based on convex shape decomposition (Dash et al. 2018). It becomes easier to understand complex shapes if they are decomposed into simpler parts. This idea is also effective for building polygon understanding and some researches have been performed. For example, building footprints are decomposed into rectangles for their roof recommendation in 3D model

visualization (Hu et al. 2018), and complex relational contexts (e.g., “between” and “among”) between buildings are defined with polygon decomposition (Du et al. 2019). If buildings are also decomposed into sub-buildings to recognize their patterns, more visually aware building patterns can then be recognized.

Motivated by the above thoughts, we try to recognize linear building patterns including collinear and curvilinear ones by combining convex polygon decomposition. Because linear building patterns are the most common patterns and form the basis for the recognition of other, higher-level patterns, e.g., grid-like patterns (Wang and Burghardt 2019). Linear building patterns are defined according to perception rules including proximity, similarity, and continuity. The linear patterns are then recognized with convex polygon decomposition, in which buildings can not only be considered as a whole, but also can be a whole made of sub-buildings. As buildings in topographical maps may be represented as a polygon with holes, a node concavity definition is developed based on polygon skeletons which is applicable for a polygon with holes or not. The building polygon is decomposed by removing the node with the largest concavity each time, and the orthogonal features of the building polygon are also considered in the decomposition.

The remainder of this paper is structured as follows. Section 2 reviews previous works for building pattern recognition and convex polygon decomposition; Section 3 defines the linear patterns including collinear ones and curvilinear ones; Section 4 introduces the linear pattern recognition process combining convex polygon decomposition; Section 5 shows and evaluates the experimental results of the proposed approach, and analyzes the sensitivity of parameters; Section 6 compares with previous works, discusses the effectiveness of the proposed approach for buildings with complex shapes and recognition of other kinds of building patterns; Section 7 concludes the paper and identifies future works.

2. Related work

2.1 Building pattern recognition

A building pattern is a specific arrangement of buildings. Whether a group of buildings forms a pattern depends on how the arrangement is defined.

If the patterns are defined as a template based on structural parameters. The patterns are then recognized by matching building groups with the defined templates. For example, Rainsford & Mackaness (2002) used this idea to detect linear patterns in rural buildings; Yang (2011) used defined templates to detect alphabet-like (e.g., Z-shaped and H-shaped) patterns. As different alphabet-like patterns may be a combination of low-level units, Gong et al. (2018) detected higher-level alphabet-like patterns by defining elementary units. However, template-based patterns are usually defined based on strict formalization, which may sometimes fail to recognize all potential patterns.

Some mathematical definitions of building patterns have also been introduced. For example, linear building patterns can be defined as a group in which buildings are arranged along a line. Wang & Burghardt (2021) used the stroke to detect linear building patterns based on a proximity graph. Collinear and curvilinear patterns were then distinguished in the detected patterns based on the ratio of overlap between their auxiliary polygon and the oriented bounding box. Sahbaz & Basaraner (2021) detected linear patterns through buffer analyzes for a generalization purpose. Wei et al. (2018) detected linear patterns based on a similar idea by using four proximity graphs, where the relative nearest graph (RNG) is proved to be the most accurate. To visualize a 3D city model, Mao et al. (2012) detected linear patterns on the ground plan of a building model using the minimum spanning tree (MST). However, patterns recognized in the above approaches may not satisfy Gestalt principles because they only consider proximity relations between buildings, whereas the similarity between buildings is also an important

factor in building pattern recognition. Christopher & Ruas (2002) detected buildings along a road by considering both proximity and similarity between buildings, although their results contain some false positives and require manual processing. Zhang et al. (2013b) recognized collinear and curvilinear patterns based on the MST with Gestalt principles by applying seven algorithms. However, this approach may not be appropriate to detect linear patterns in dense areas for MSTs usually comprise trees with short branches. Pilehforooshha & Karimi (2018) proposed a framework for detecting linear building patterns based on a defined similarity index, and these patterns were then refined by a defined pattern interaction index. Because different linear building patterns may share the same buildings, Gong et al. (2014) detected multi-connected linear patterns by deleting edges from a DT-like proximity graph with structural parameters. Grid-like patterns were then recognized based on the detected linear patterns. However, the similarity between buildings' shapes is not considered in their approach because buildings are approximated as their smallest bounding rectangle (SBR), and the recognized linear patterns by using the approach may be inconsistent with visual perception. Xing et al. (2021) also detected the collinear patterns by representing the buildings with the axes of their SBR. Anders (2006) detected grid-like patterns based on the RNG; while Wang & Burghardt (2019) and Wang & Burghardt (2020) used a proximity graph to detect grid-like building patterns for typification based on mesh ideas. However, their definitions of grid-like patterns are vague.

Reasoning-based approaches have also been used for building pattern recognition. Du et al. (2016) proposed a graph partitioning approach with graph coarsening for building clustering. Collinear patterns, curvilinear patterns, parallel and perpendicular groups, and grid-like patterns were then detected in obtained clusters based on four integrated strategies. Du et al. (2015a) also developed a three-level relational approach

based on spatial reasoning. Qualitative angle and qualitative size relations between buildings were defined to link buildings to building patterns. As their pattern definitions were strictly formalized, some potential patterns may fail to be recognized. Lüscher et al. (2009) defined the concept of the English terraced house with ontological modeling and a supervised Bayesian inference was then used to determine complex concepts for pattern classification.

Developments in computer science have also led to the introduction of some machine learning algorithms for building pattern recognition. He et al. (2018) recognized regular (e.g., collinear, curvilinear, and rectangular) patterns and irregular (e.g., L-shaped, H-shaped, and high-density) patterns in topographic maps based on graph partitioning and the random forest algorithm. Yan et al. (2018) and Zhao et al. (2020) recognized patterns of building groups using the graph convolutional network, where a building group is represented with a graph. However, more training data from different regions may be needed to improve the accuracy of these machine learning algorithms. An examination of reasons for why these algorithms are useful may also be required.

In all the above approaches, buildings are only considered as a whole. Some visually aware building patterns may fail to be recognized by these approaches because human vision is also proved to be a part-based system. Urban buildings are also getting more and more complex nowadays, and the parameters or relational descriptions used in these approaches may be difficult to apply for buildings with complex shapes. Thus, decomposing the buildings into convex polygons will help us to understand the building shapes easily and accurately recognize building patterns.

2.2 Convex polygon decomposition

Representing a shape in terms of meaningful parts is a fundamental problem in shape analysis and recognition (Rodrigues et al., 2018). As convexity plays a vital role in human

cognition, shape polygons are usually decomposed into convex or approximate convex sub-polygons based on user requirements (Lien & Amato, 2006).

Different requirements can be developed into rules for different purposes in convex polygon decomposition. The most commonly used rules are the minimum component rule and the short-cut rule based on psychophysical findings (Singh et al., 1999; Luo et al., 2014). The minimum component rule states that the polygon needs to be decomposed into a minimum number of sub-polygons, while the short-cut rule says that the shorter cut is more likely to be selected. To maintain a balance in the area of the sub-polygons after decomposition, a symmetry axis or an angle bisector is also preferred as a cut in polygon decomposition (Mi & DeCarlo, 2007). If the critical points are considered, the lines connecting two critical points may be more likely to be selected as a cut (Qu et al., 2016). If the polygons are decomposed for a classification task, the number of sub-polygons of each polygon after decomposition may need to be similar (Wang & Lai, 2016).

Different rules may conflict with each other and need to be optimized, and some optimization algorithms are applied for polygon decomposition. Lipson & Shpitalni (2001) used the A* algorithm to obtain a hierarchical decomposition of the polygons. Juengling & Mitchell (2007) compared the decomposition results obtained by using a dynamic programming and a greedy algorithm. Ren et al. (2013) used an integer linear programming algorithm to decompose polygons with concavity constraints. Wang & Lai (2016) used the minimum energy algorithm to decompose polygons by searching an optimal pruning sequence. If these rules are transformed into constraints of clustering algorithms, some clustering algorithms can also be applied. Liu et al. (2004) proposed an improved k-means clustering for a 3D mesh, and Li et al. (2020) applied a spectral clustering approach for the near-convex decomposition of 2D polygons.

Polygons often have holes in practice, and polygon decomposition becomes an NP-

hard problem if the minimum component rule and the short-cut rule need to be met at the same time (Lien & Amato, 2006). Optimization algorithms or clustering algorithms may be inflexible on these occasions. Another commonly used strategy is to decompose the polygons step by step, or in a divide-and-conquer strategy. Concavity needs to be defined in these strategies as polygons are usually cut with a decrease in concavity. Concavity can be measured by estimating the similarity of a polygon to its convex hull (Wei et al., 2018). But it is a global measure for the polygon, and it is difficult to determine where and how to decompose the polygons with the measure. Thus, some concavity parameters have been proposed for the concave nodes of the polygon. For example, the concavity of the concave nodes can be measured by the longest distance from the nodes to the polygon's convex hull along the polygon's boundary, or the largest "visible distance" from the nodes to the polygon's convex hull (Zunic & Rosin 2002). As the polygon is decomposed step by step, node concavity can be defined according to the reduction in concavity after each cut (Ghosh et al., 2013). Different measures can also be combined (Lien & Amato 2006). But above measures are not applicable for a polygon with holes.

3. Linear building pattern definition

Linear pattern definition mainly involves two tasks, pattern criteria, and pattern rules. The criteria mean how to measure a linear pattern, the rules represent how to define the pattern based on the criteria.

3.1 *Criteria*

Linear building patterns are visually salient structures, in which similar buildings arrange along a line. They can be ruled by visual perception criteria as summarized in Table 1, according to Li et al. (2004), Zhang et al. (2013), Du et al. (2016), and Wang & Burghardt (2018). But unlike the buildings that are all defined as a whole in the existing

approaches, the buildings which follow these criteria can also be a part of a building in the proposed approach.

Table 1. Criteria for linear building patterns.

Criteria	Description
Proximity	Buildings adjacent to each other tend to be in a linear pattern.
Similarity	Buildings similar in area, shape, orientation tend to be in a linear pattern.
Continuity	Buildings that distribute continuously along a line tend to be in a linear pattern.

In the following, we would introduce the properties of each criterion.

(1) Proximity

The proximity relation between buildings can be represented as a proximity graph, and two buildings connected by an edge are defined as being adjacent to each other. As roads and buildings are the two main features in a topographical map and roads sometimes may naturally divide these buildings into groups (Li et al., 2004). A proximity graph based on constrained Delaunay triangulation (CDT) skeleton that considers roads is used to model the proximity relation between buildings, the buildings that share the same skeleton are considered as being adjacent to each other (Liu et al. 2014), as shown in Figures 2(a) and 2(b). To avoid generating extremely narrow triangles in CDT construction, the segments of roads and buildings are checked with an empirical threshold (T_l) which can be set by considering the average length ($\overline{L_s}$) of all segments as Equation 1 (Liu et al., 2014).

$$T_l = \lambda * \overline{L_s} \quad (1)$$

λ is a weight and here $\lambda=1$. Suppose a segment with length as L_{Si} , if $L_{Si} > T_l$, then additional points are inserted by applying a linear interpolation method (Liu et al., 2014).

Different sub-graphs can be obtained based on the proximity graph, e.g., MST and RNG. As compared by Wei et al. (2018), RNG is proved to be more effective for linear building pattern recognition. RNG is used to model the proximity relation between

buildings in the proposed approach, Figure 2(b). The RNG can be represented as $G=(E,V)$, where $V=\{v_1, v_2, \dots, v_m, \dots\}$ is the node set, v_m represents building B_m , and $E=\{e(v_a, v_b), v_a \in V, v_b \in V\}$ is the edge set. Two buildings B_a and B_b connected by $e(v_a, v_b)$ are defined as being adjacent to each other. Length of $e(v_a, v_b)$ (Le) is defined as the nearest distance between B_a and B_b .

Because buildings may be decomposed into sub-buildings in the proposed approach, two adjustments are added. 1) A threshold (T_e) for Le is set because the Le between two decomposed sub-buildings may be zero. If $Le \leq T_e$; Then set $Le=T_e$. T_e can be set according to the legibility constraints in building map generalization, e.g., $T_e=0.2\text{mm}$ at 1: 10k (Ruas, 1998). 2) RNG edges need to be locally adjusted if a building is decomposed into sub-buildings. As shown in Figure 2(d), the buildings B_1 and B_2 are decomposed into sub-buildings, their connected RNG edges need to be adjusted to connect the obtained decomposed sub-buildings. If an adjusted edge crosses over a building, it needs to be deleted. The local adjustment for the RNG is performed as Algorithm 1 in case a building is decomposed into parts.

Algorithm 1. Local adjustment for RNG in case a building is decomposed into parts

Input: Building set as $BS=\{B_1, B_2, \dots, B_m, \dots\}$, its RNG as $G=(E,V)$; a building B_k is decomposed into a polygon set and represented as $DB^k=\{B^k_1, B^k_2, \dots, B^k_n, \dots\}$, its corresponding nodes as $DV^k=\{v^k_1, v^k_2, \dots, v^k_n, \dots\}$

Output: $G=(E,V)$

1. **Update V :**

Remove v_k from V , add each node v^k_n in DV^k to V

2. **Update E :**

For each node v_i in V **Do**

If the edge $e(v_k, v_i)$ is an edge of E **Then**

For each node v^k_n in DV^k **Do**

If the edge $e(v_n^k, v_i)$ doesn't across over any building B_m in BS **Then** add $e(v_n^k, v_i)$

to E

For each sub-polygon B_n^k in DB^k **Do**

Find any building B_p^k in DB^k touches with B_n^k **AND** $e(v_p^k, v_n^k)$ is not an edge of E **Then**

add $e(v_p^k, v_n^k)$ to E

Return $G=(E,V)$

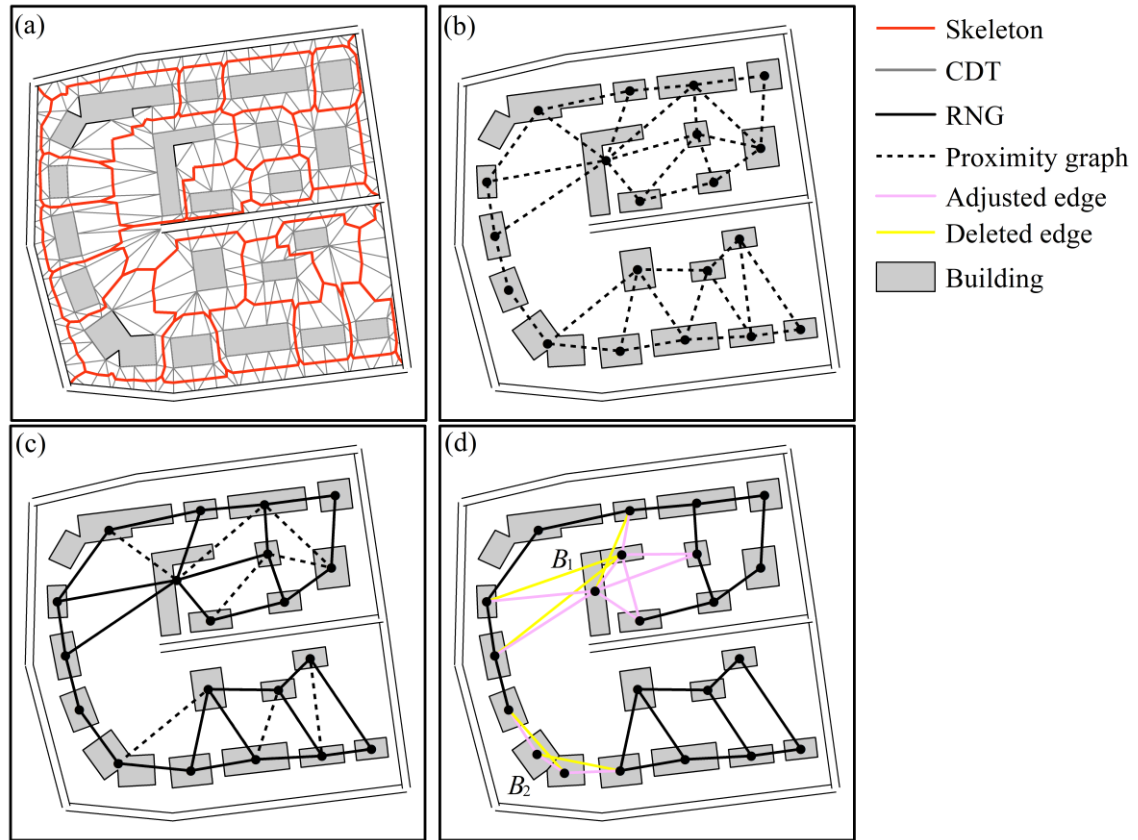


Figure 2. Proximity graph. (a) CDT skeleton considering roads; (b) Proximity graph; (c) RNG; (d)

Local adjustment for RNG while polygons are decomposed.

(2) Similarity

The similarity between two buildings (B_a and B_b) can be defined based on their similarities in area, orientation, and shape, as illustrated in Table 2.

Table 2. Parameters for the similarity between two buildings (B_a and B_b). Large values for similarity parameter mean low similarity between B_a and B_b (\downarrow).

Measure	Description and definition	Source
$As \downarrow$	$As(B_a, B_b) = \max(Area_{B_a}, Area_{B_b}) / \min(Area_{B_a}, Area_{B_b})$, $Area$ is the area of a polygon	Yan et al. (2008)
$Os \downarrow$	$Os(B_a, B_b) = \begin{cases} Ori_{B_a} - Ori_{B_b} & (Ori_{B_a} - Ori_{B_b} \leq 90) \\ 180 - Ori_{B_a} - Ori_{B_b} & (Ori_{B_a} - Ori_{B_b} > 90) \end{cases}$, Ori is the orientation of the SBR of a polygon	Zhang et al. (2013b)
$Ss \downarrow$	$Ss(B_a, B_b) = \max(EC_{B_a}, EC_{B_b}) / \min(EC_{B_a}, EC_{B_b})$, EC is the edge count of a polygon	Yan et al., (2008)

(3) Continuity

Continuity rules whether the buildings arrange along a line, and can be defined between the three buildings (B_a , B_b , and B_c) based on the differences in the orientation and length of their connecting RNG edges, as defined in Table 3 (Wei et al., 2018). For a more detailed classification of linear patterns, the aligned angle (*alignAngle*) and the facing ratio (*FR*) between two buildings (B_a and B_b) are also introduced by Zhang et al. (2013) and Wang & Burghardt (2019a), as defined in Table 3.

Table 3. Parameters for continuity between three buildings (B_a , B_b , and B_c).

Measure	Description and definition	Source
Do	The angle between the two RNG edges connecting B_a , B_b , and B_c	
D_L	$D_L(B_a, B_b, B_c) = \max(Le_1, Le_2) / \min(Le_1, Le_2)$, Le_1 and Le_2 are the lengths of the two RNG edges connecting B_a , B_b , and B_c	Wei et al. (2018)
<i>alignAngle</i>	The <i>alignAngle</i> is defined for a building B_a and a RNG edge e which connects B_a as: the angle between the long axis of B_a and e	Zhang et al. (2013)
<i>FR</i>	$FR = \text{Overlap}(Pro_{B_a}, Axis_{B_b}) / \text{Union}(Pro_{B_a}, Axis_{B_b})$, Pro_{B_a} is the projection length of B_a on B_b 's axis, $Axis_{B_b}$ is the length of B_b 's axis	Wang & Burghardt (2021)

3.2 Rules

Different linear patterns can be obtained according to different definitions for their aligned lines. The typology for linear building patterns according to Wang & Burghardt (2019a) is adopted in our approach as collinear patterns and curvilinear patterns, and a collinear pattern can also be considered a special kind of curvilinear pattern.

(1) Collinear patterns

Collinear patterns are considered as patterns in which similar buildings are along a straight line. The proximity criteria for a collinear pattern is defined as whether the two buildings (B_a and B_b) are connected by a RNG edge as $P(B_a, B_b)$. The similarity criterion for a collinear pattern can be defined as whether the two buildings (B_a and B_b) are similar in size, orientation, and shape as $S(B_a \text{ and } B_b)$, Equation 2.

$$S(B_a, B_b) =_{\text{def}} \{A_s(B_a, B_b) \leq \delta_1, O_s(B_a, B_b) \leq \delta_2, S_s(B_a, B_b) \leq \delta_3\} \quad (2)$$

δ_1 , δ_2 and δ_3 are the thresholds. The continuity criterion for a collinear pattern can be defined based on the differences in the orientation and length of the RNG edges which connect the three adjacent buildings (B_a , B_b , and B_c) as $C(B_a, B_b, B_c)$, Equation 3.

$$C(B_a, B_b, B_c) =_{\text{def}} \{Do(B_a, B_b, B_c) \geq \eta_1, DL(B_a, B_b, B_c) \leq \eta_2\} \quad (3)$$

η_1 and η_2 are the thresholds. Collinear patterns sometimes can further be classified as straight-line patterns and oblique patterns, and can be ruled by setting a range $[\beta_1, \beta_2]$ for the FR . The straight-line pattern may have a range $[0.4, 1.0]$, the oblique patterns may have a range $[0.0, 0.4]$. A collinear pattern is defined as Equation 4.

$$ColP(B_a, B_b, B_c) =_{\text{def}} P(B_a, B_b) \wedge P(B_b, B_c) \wedge S(B_a, B_b) \wedge S(B_b, B_c) \wedge C(B_a, B_b, B_c) \wedge FR \in [\beta_1, \beta_2] \quad (4)$$

(2) Curvilinear patterns

Curvilinear patterns are considered as patterns in which similar buildings along a curve, which means the pattern gradually alters their heading angles and orientations of composing buildings (Zhang et al., 2013). The proximity and similarity criteria for a curvilinear pattern can be defined as the same as those for a collinear pattern. As for the continuity criterion for a curvilinear pattern, a tendency to describe the gradual alternation of the heading angles and orientations of composing buildings should be defined.

According to Zhang et al. (2013), the tendency $T(B_a, B_b)$ for two adjacent buildings (B_a , B_b) connected by an RNG edge can be ruled by the *alignAngle* as Equation 5.

$$T(B_a, B_b) =_{\text{def}} \{ \text{alignAngle}(B_a) \leq \beta, \text{alignAngle}(B_b) \leq \beta \} \quad (5)$$

β is a threshold, and β may decrease with the increase of the Do according to Zhang et al. (2013). A curvilinear pattern can be defined as Equation 6.

$$\text{ColP}(B_a, B_b, B_c) =_{\text{def}} \{ P(B_a, B_b) \wedge P(B_b, B_c) \wedge S(B_a, B_b) \wedge S(B_b, B_c) \wedge C(B_a, B_b, B_c) \wedge T(B_a, B_b) \wedge T(B_b, B_c) \} \quad (6)$$

The thresholds include η^1 and η^2 are usually set larger in a curvilinear pattern than those in a collinear pattern.

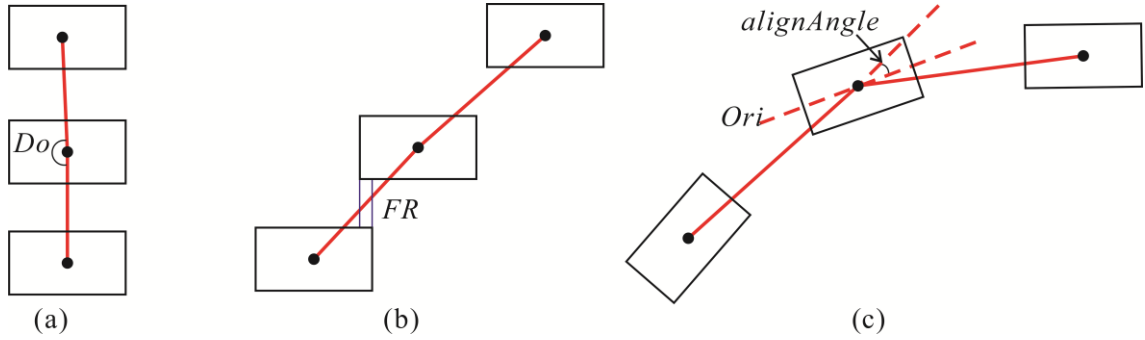


Figure 3. Linear patterns. (a) Straight-line pattern; (b) Oblique patterns; (c) Curvilinear pattern.

4. Linear pattern recognition combining convex polygon decomposition

With the definitions as in Equations 4 and 6, a linear pattern can be recognized with two adjacent buildings B_a and B_b as a start, and grows by adding other buildings. A pattern grows if B_a and B_b and the newly added building B_c form a pattern. And whether B_a , B_b , and B_c form a pattern needs to be detected by combining convex polygon decomposition in the proposed approach. The pattern growth is repeated until no pattern is formed while a new building is added. All linear patterns in a building cluster can be recognized by repeating this process. Thus, convex polygon decomposition and linear pattern growth are the two keys in linear pattern recognition.

(1) Convex polygon decomposition: The algorithm to decompose the selected polygon into two sub-polygons is defined in Section 4.1.

(2) Linear pattern growth: The algorithm to determine whether a linear pattern can grow or not is defined in Section 4.2, and performed by evaluating whether the three buildings form a linear pattern combining convex polygon decomposition.

4.1 *Convex polygon decomposition*

Convex polygon decomposition aims to remove the polygon's concave nodes. A commonly used strategy is to decompose the polygons with a decrease in node concavity (Lien & Amato, 2006). The node with the largest node concavity is selected to perform a cut to decompose the polygon into two sub-polygons step by step. The following two steps will be performed: (1) Node concavity is defined first; (2) A cut is then selected to decompose the polygon into two sub-polygons.

4.2.1 *Node concavity definition*

Here we first define the variables used in our method for clarity. A building polygon with holes is denoted as a polygon set as $PS = \{PB_1, PH_1, \dots, PH_m, \dots\}$, where PB_1 is the boundary and PH_m is a hole. Nodes of a polygon in PS are denoted as $NS = \{v_1, v_2, \dots, v_n, \dots\}$. The angle (Ag_n) at node v_n is defined as the angle of edge $v_{n-1}v_n$ rotates clockwise around node v_n to edge v_nv_{n+1} . If v_n is on PB_1 and $Ag_n \in (180, 360)$, v_n is a concave node; if v_n is on PH_m and $Ag_n \in (0, 180)$, v_n is a concave node. The concave area of a building polygon associated with a concave node v_n is defined as CA_n . If a building B_i has concave nodes, then B_i is a concave polygon.

The implicit hierarchy of polygon skeletons can well represent the hierarchy of CAs in a polygon with holes and has been widely applied in shape analysis (Liu et al., 2014). Thus, polygon skeletons are used to define the node concavity in the proposed approach.

(1) Polygon skeletons and their implicit hierarchy

The polygon skeletons are obtained based on constrained Delaunay triangulation (CDT), as shown in Figure 4(a) (Liu et al., 2014). The additional points are inserted into the long edges of the buildings in CDT construction as defined in Equation 1 with $\lambda=0.2$. The triangles in CDT can be classified into four types according to their neighbors: one neighbor (type I), two neighbors (type II), three neighbors (type III), and connects to a non-enclosed external area (type IV). According to Ai et al. (2017), a type IV triangle serves as the entry to a CA, and a type III triangle acts as the crossing triangle, where a larger CA is divided into two sub-CAs, the type II triangles consist of the body of CAs; a type I triangle serves as the terminal area of a CA. The implicit hierarchy of these CAs and their corresponding skeletons can then be built as trees based on the triangle classification, Figures 4(b) and 4(c).

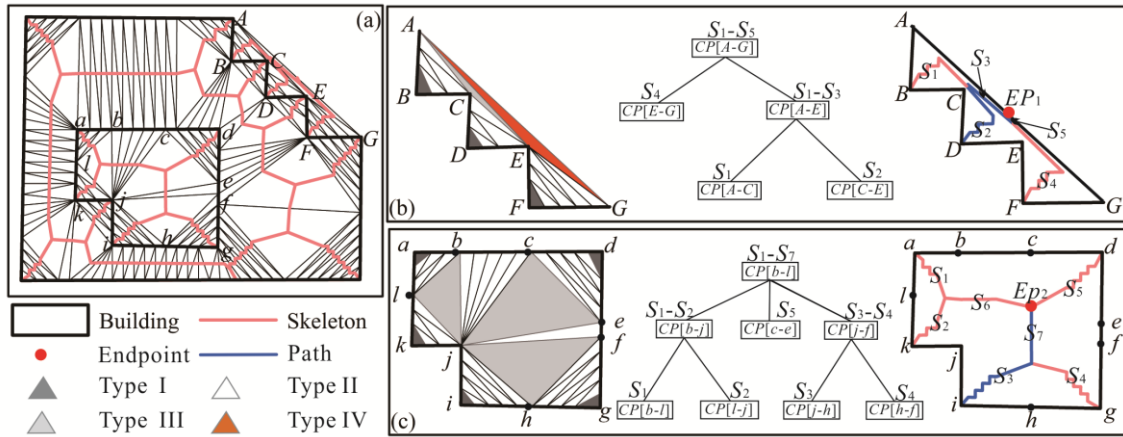


Figure 4. Node concavity definition. (a) The skeleton of the building polygon based on CDT. (b) Concavity of node on the building's boundary. (c) Concavity of node on a hole.

(2) Node concavity definition

An intuitive way to define node concavity of v_n is to obtain the path length of v_n extending outward to the convex hull of PB_1 (if v_n is on PB_1), or inward to the center of PH_m (if v_n is on PH_m) (Lien & Amato, 2006). As shown in Figure 4, the skeletons can be used as paths for v_n extending outward or inward to an endpoint from a high-level skeleton

to a low-level skeleton. If v_n is on PB_1 , the edge adjacent to the non-enclosed area in the type IV triangle acts as the entry of the CA associated with v_n . The midpoint of the edge is an endpoint of the extending path along with the skeletons from v_n . For example, the extending path of node D on the boundary can be obtained as S_2 - S_3 - S_5 to the endpoint EP_1 , Figure 5(b). If n_p is on PH_m , the endpoint is the center point of PH_m for the extending path along with the skeleton from v_n . The center point of PH_m which locates on the skeletons is obtained with the centrality of vertices based on skeletons according to Lu et al. (2020). For example, the extending path of node i on the hole can be obtained as S_3 - S_7 to the endpoint EP_2 , Figure 4(c). The concavity of v_n ($NCon_n$) is then defined as the length of the extending path from v_n to an endpoint along with the skeletons as Equation 7.

$$NCon_n = LP_n / Per_n \quad (7)$$

where LP_n represents a length of the extending path, and Per_n represents the perimeter of corresponding CA associated with v_n .

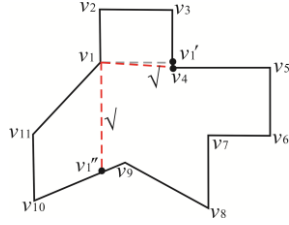
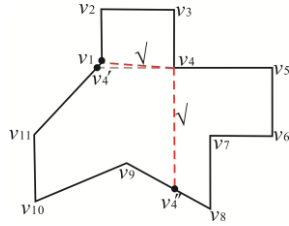
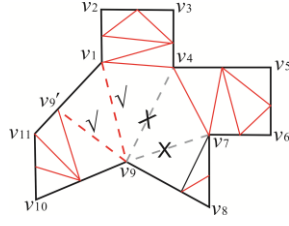
4.2.2 Cut selection

A cut is a line that can decompose the polygon into two sub-polygons or remove a hole of the polygon. For a given node v_n , the best cut needs to be selected from the candidate cuts to implement the decomposition.

(1) Candidate cuts

Buildings are man-made objects which tend to have a simple shape and orthogonal features (OFs) (Sester, 2005). A cut can't destroy the OFs , and the OFs can be defined as a node set $\{v_{n-1}, v_n, v_{n+1}, v_{n+2}\}$, in which the angles at node v_n and v_{n+1} are right angles (Wei et al., 2021). The right angle is an angle (Ag) that satisfies $|Ag - 90| < \alpha \cup |Ag - 270| < \alpha$, α is a constant. Suppose the angle at v_n as Ag_n , the candidate cuts for v_n in a building B_i are defined as Table 4.

Table 4. Candidate cuts for a node v_n in a building B_i , its angle is denoted as Ag_n , $OS = \{v_{m-1}, v_m, v_{m+1}, v_{m+2}\}$ is denoted as the node set of an orthogonal feature.

v_n	Candidate cuts	Examples
$v_n \in OS$ and Ag_n is not a right angle	1) Extending edge $v_n v_k$ of B_i , $v_k \in OS$, and intersects B_i as a node $v_{n'}$, the candidate cut is $v_n v_{n'}$; 2) Draw a line parallel to edge $v_m v_{m+1}$ from node v_n , and intersects B_i as a node $v_{n''}$, the candidate cut is $v_n v_{n''}$. For example, $v_1 v_4$ and $v_1 v_{1''}$ are candidate cuts for concave node v_1 .	
$v_n \in OS$ and Ag_n is a right angle	Extending the edges $v_{m-1} v_m$ and $v_m v_{m+1}$ of B_i , and intersect B_i as a node $v_{n'}$ and $v_{n''}$, the candidate cuts are $v_n v_{n'}$ and $v_n v_{n''}$. For example, $v_4 v_1$ and $v_4 v_{4''}$ are candidate cuts for concave node v_4 .	
$v_n \notin OS$	The CDT edges which are inside B_i and connect v_n are candidate cuts, and the cuts destroy the OFs or can't remove their related concave nodes are deleted from the candidate cuts. For example, $v_9 v_1$ and $v_9 v_{9''}$ are candidate cut for concave node v_9 .	

Remarks: suppose a candidate cut as $v_n v_{n'}$, and $v_{n'}$ may be close to a node of B_i . If there is a node v_x of B_i , the angle (Ag_x) between $v_n v_x$ and $v_n v_{n'}$ satisfies $Ag_x \leq \delta$, then the node $v_{n'}$ is replaced by node v_x , and $\delta = 3^\circ$. For example, the cut $v_1 v_{1'}$ is changed into $v_1 v_4$, and the cut $v_4 v_{4'}$ is changed into $v_4 v_1$.

(2) Best cut

The most commonly used rules for convex polygon decomposition are the minimum component rule and the short-cut rule based on psychophysical findings (Singh et al., 1999; Luo et al., 2014). The two rules are also adopted in this approach.

Minimum component rule: The polygon needs to be cut into minimum convex polygons. The rule can be measured by a reduction in the number of concave nodes after the execution of the cut, denoted as $RCut$. The cut with a larger $RCut$ is more likely to be selected.

Short-cut rule: A cut with shorter edges is more likely to be selected. The rule can be measured by the length of the cut, denoted as $LCut$. The cut with a smaller $LCut$ is more likely to be selected.

The minimum component rule has a higher priority in many convex polygon decomposition algorithms. The optimization function for a Cut_i is defined as Equation 8.

$$f_{Cut_i} = \alpha_1 * (RCut_i - 2) * LCut_i \quad (8)$$

where α_1 is a large constant, and the cut with smaller f_{Cut_i} is more likely to be selected.

4.2.3 Algorithm

The node with the largest node concavity is selected to perform a cut to decompose the building polygon into two sub-polygons each time. If no cut is available for the node, the node with the next largest concavity is selected; as building polygon may have holes, not all cuts can cut a polygon into two sub-polygons. On this occasion, an added cut is performed. Furthermore, a preprocess also needs to be performed to delete the unnecessary nodes before each cut, e.g., repeating nodes, collinear nodes, and sharp nodes (Wei et al., 2021). For details of the polygon preprocess see Wei et al. (2021). The algorithm to cut a polygon into two sub-polygons is performed as **Algorithm 2**.

Algorithm 2: Decompose the given polygon (B_i) into two sub-polygons

Input: B_i , denoted as a polygon set $PS_i = \{PB_i, PH_1, \dots, PH_m, \dots\}$, its concave nodes denoted as CN_i

Output: the polygon set obtained after decomposition, denoted as RP

Initial $RP = Null$

While B_i is not decomposed into two sub-polygons **AND** there are concave nodes in CN_i

Get the concave node v_n in CN_i with the largest node concavity (defined as Equation 7)

Get the candidate cuts for v_n , denoted as CS

If there are cuts in CS **Then**

Get the best cut Cut_b in CS , and perform the cut Cut_b for B_i

Update PS_i , CN_i and RP

Else Remove the node v_n from CN_i

If there are two polygons in RP **Return** RP

4.2 Linear pattern growth

As illustrated in Section 4.1, a linear pattern can be recognized with two adjacent buildings B_a and B_b as a start, and grows by adding other buildings. A pattern grows if B_a and B_b and the newly added building B_c form a pattern. The key is to determine whether B_a , B_b , and B_c form a pattern by combining convex polygon decomposition. If B_a , B_b and B_c can't form a pattern, a building in $\{B_a, B_b, B_c\}$ is selected to be decomposed into two sub-polygons. The decomposition process is repeated until a stop condition is reached or a linear pattern is obtained. The process of linear pattern growth is shown in Figure 5.

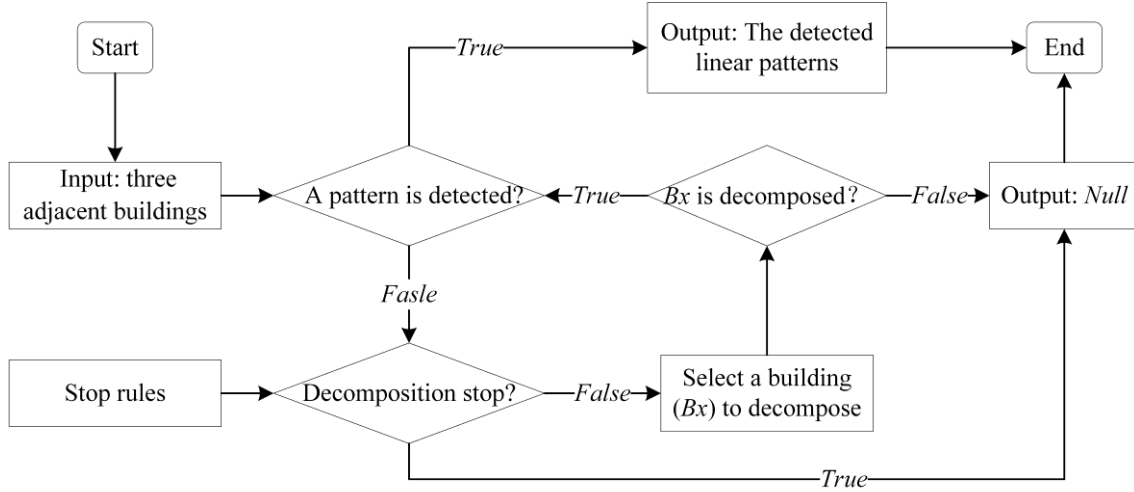


Figure 5. Process of linear pattern growth.

(1) Select a building to decompose

The polygon with the most complex shape is selected to be cut each time. The shape complexity can be described as edge count (EC) defined in Table 2, i.e., the polygon with the largest EC is selected to be cut.

(2) Stop rules

Stop conditions rule when to stop the decomposition process, mainly involved in whether the selected building can be decomposed or not. With consideration of the cognitive burden, two adjacent buildings can't be decomposed at the same time in a linear

pattern (*Rule 1*); and a building won't be decomposed into many sub-buildings for a visual task (*Rule 2*) (Dash et al., 2018). Because a linear building pattern is recognized with two buildings as a start, and grows by adding other buildings. While recognizing a specific pattern with given two buildings as a start, if a building is already a part of the specific pattern, it can't be decomposed in subsequent pattern growth (*Rule 3*). And a polygon won't be decomposed into strict convex polygons, but into approximate convex polygons in practice. It means a building won't be decomposed if its largest node concavity is smaller than a threshold (*Rule 4*). Furthermore, if the building's boundary is a convex polygon, it also can't be decomposed (*Rule 5*). These rules are summarized in Table 5.

Table 5. Rules to stop the decomposition process.

Rules	Descriptions
<i>Rule 1</i>	A building can't be decomposed if an adjacent building has been decomposed
<i>Rule 2</i>	A building can't be decomposed if it has been decomposed T_k times, where T_k is a threshold
<i>Rule 3</i>	In recognition of a specific pattern with given two buildings as a start, if a building is already a part of the specific pattern, it can't be decomposed in subsequent pattern growth
<i>Rule 4</i>	A building can't be decomposed if its largest node concavity is smaller than a threshold (T_{nc})
<i>Rule 5</i>	A building can't be decomposed if the building's boundary is already a convex polygon

(3) Algorithm

Given three adjacent buildings as $\{B_a, B_b, B_c\}$, in which B_c is a newly added building.

Whether $\{B_a, B_b, B_c\}$ will grow is implemented as **Algorithm 3**.

Algorithm 3: Linear pattern growth, decide three adjacent buildings will grow or not

Input: Two adjacent buildings B_a and B_b , and a newly added building B_c

Output: $\{B'_a, B'_b, B'_c\}$ as a linear pattern, means $\{B_a, B_b, B_c\}$ can grow; *Null*, means $\{B_a, B_b, B_c\}$ can't grow

If $\{B_a, B_b, B_c\}$ forms a pattern **Then**

Return $\{B_a, B_b, B_c\}$

Else

Get the building B_y with the largest EC in $\{B_a, B_b, B_c\}$

If a stop condition is reached **Then Return** *Null*

Else Polygon decomposition is applied for B_y , and obtained polygons are denoted as RP

If there are no polygons in RP **Then Return** $Null$

Else the two polygons in RP are represented as $RP = \{RB^1, RB^2\}$

If there is a polygon RB^i in RP forms a pattern with another two buildings in $\{B_a, B_b, B_c\}$

Then return $\{\{x|x \in \{B_a, B_b, B_c\} \wedge x \neq B_y\}, RB^i\}$

Else repeat linear pattern growth algorithm for $\{\{x|x \in \{B_a, B_b, B_c\} \wedge x \neq B_y\}, RB^1\}$ and $\{\{x|x \in \{B_a, B_b, B_c\} \wedge x \neq B_y\}, RB^2\}$

4.3 Linear pattern recognition

The linear pattern recognition algorithm in a building cluster consists of two steps.

Step 1: A linear pattern is recognized with linear pattern growth by adding other buildings. All linear patterns are obtained by repeating Step 1.

Step 2: A post-process is performed to delete the duplicate patterns.

Given a building set as $BS = \{B_1, B_2, \dots, B_m, \dots\}$, its RNG as $G = (E, V)$, $V = \{v_1, v_2, \dots, v_m, \dots\}$, $E = \{e(v_a, v_b), v_a \in V, v_b \in V\}$, the linear building patterns combining convex polygon decomposition are recognized as Algorithm 4.

Algorithm 4: Linear building pattern recognition combining convex polygon decomposition

Input: the building set as $BS = \{B_1, B_2, \dots, B_m, \dots\}$, its proximity graph as $G = (E, V)$

Output: A collection of linear patterns (LS)

1. Linear building pattern recognition

For each edge $e(v_a, v_b)$ in E **Do**

Set the start edge as $SE = e(v_a, v_b)$: v_a as a start node, v_b as an end node; the linear pattern with SE as a start is denoted as $LP = \{B_a, B_b\}$, and a deep copy of BS is denoted as $CopyBS$, a deep copy of G is denoted as $CopyG = (CopyE, CopyV)$

Loop initial the start edge SE

For each edge $e(v_b, v_c)$ in $CopyE$ **Then**

Linear pattern growth algorithm is applied for $\{B_a, B_b, B_c\}$, and the output of linear pattern growth algorithm is denoted as GB

If there are three buildings in GB , it means a pattern with three buildings can be obtained by linear pattern growth **Then**

The three buildings in GB are represented as $GB = \{B'_a, B'_b, B'_c\}$;

Local adjustment algorithm of RNG is applied for $CopyG$, and update $CopyBS$;

Set the start edge as $SE = e(v'_b, v'_c)$ Update the linear building pattern LP ;

Break;

End Loop

If there are three or more buildings in LP , it means LP is a pattern **Then** add LP to LS

2. Post process: delete the duplicated patterns in LS .

5. Experiments

5.1 Experiment data

The study area is an area with 765961 buildings represented by an ID as ‘TL’ in the map product named ‘OS OpenMap-Local’ (Ordnance Survey, 2021). The nominal viewing scale of the map product is 1:10,000. Statistics analysis includes building count (BC), average area ($AveArea$), average edge count ($AveEC$), the rate of buildings with fewer edges no more than 8 (ER), and the rate of right angles (RR) for all buildings in the study area are shown in Table 6. From Table 6, we have the following observations: (1) Buildings have an average EC of 4.47, and 98.03% of the buildings have fewer edges no more than 8, which means most buildings are with simple shapes. Thus, EC is an effective parameter that can be used for shape similarity measurement due to the simple shape of buildings (Yan et al., 2008). Furthermore, EC also has a low computational cost which can be easily applied to buildings. (2) The rate of right angles reaches 96.06%, which means most buildings are with multiple orthogonal features and need to be well considered in polygon decomposition, the definition of right angle and orthogonal features see Section 4.2.

Two datasets in which buildings have similar characteristics as the study area were collected and used for the experiments. Dataset A is with 896 buildings and 161 roads, Figure 6(a); and Dataset B is with 767 buildings and 122 roads, Figure 6(b).

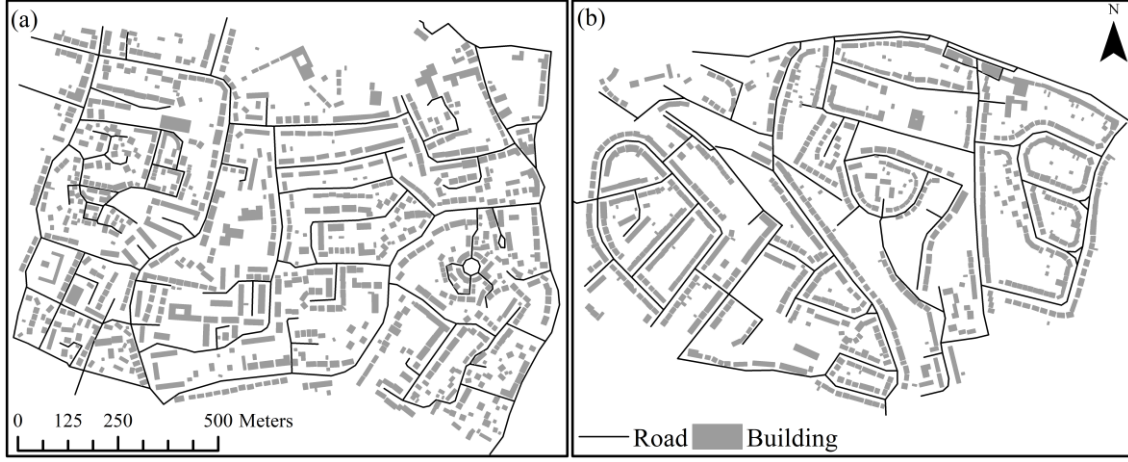


Figure 6. The two datasets for experiments.

Table 6. Analysis on the characteristics of buildings in the study area.

Data	BC	$AveArea/m^2$	$AveEd$	$ER\% (EC \leq 8)$	$RR\%$
The study area	765961	216.94	4.47	98.03	96.06
Dataset A	896	186.30	4.30	98.10	97.09
Dataset B	767	183.46	4.34	97.39	95.29

5.2 Result and evaluation

Collinear patterns were recognized in Dataset A with parameters setting as: $\delta_1=2$, $\delta_2=15^\circ$, $\delta_3=1.4$, $\eta_1=165^\circ$, $\eta_2=2$, $FR \in [0.4, 1.0]$. 88 collinear patterns were recognized in Dataset A, and 28 buildings were decomposed for collinear pattern recognition (Figure 7). Curvilinear patterns are recognized in Dataset B with parameters set as $\delta_1=2$, $\delta_2=25^\circ$, $\delta_3=1.4$, $\eta_1=155^\circ$, $\eta_2=2$, $\beta=20^\circ$, and 81 curvilinear patterns were recognized in dataset B, and 21 buildings were decomposed for curvilinear pattern recognition (Figure 8).

To evaluate the efficiency of our approach, we compared our results with those identified manually by 6 graduate students who have experience in cartography and spatial databases. They annotated the patterns separately with informed that the polygons can be decomposed, and a voting system was adopted when the labeling results were inconsistent. The recognized patterns that were in line with those identified by the users are denoted by tp , those that were not in line with the ones identified by the users are denoted by fp , and patterns that were not recognized by the proposed approach are denoted by fn . *Precision* and *Recall* were defined as Equations 9 and 10 (Du et al., 2015a).

$$Precision = tp / (tp + fp) \quad (9)$$

$$Recall = tp / (tp + fn) \quad (10)$$

The statistical analysis of our approach is shown in Table 7. The results show that (1) Most linear patterns can be recognized in the two datasets with the proposed approach, and the *Recall* for collinear pattern recognition in Dataset A is 95.56%, and the *Recall* for curvilinear pattern recognition in Dataset B is 92.59%. (2) A high *Precision* can be reached while linear patterns are recognized with the proposed approach, and the *Precision* for collinear pattern recognition in Dataset A is 97.72%, and the *Precision* for curvilinear pattern recognition in Dataset B is 92.59%. (3) 2 collinear patterns are misrecognized and 4 collinear patterns are not recognized in Dataset A; and 6 curvilinear patterns are misrecognized and not recognized in Dataset B. These linear patterns are not recognized or misrecognized for our fixed parameter setting: If $FR \in [0.5, 1.0]$, then P_1 in Figure 7(b) will not be recognized; if $\delta_1=3$, then P_2 in Figures 7(b), P_3 and P_4 in Figures 7(c) and P_5 in Figure 8 (b) will be recognized; If $\eta_1=145^\circ$ and $\beta=20^\circ$, then P_6 and P_7 in Figure 8(c) will be recognized.

Table 7. Evaluation of the linear pattern recognition results.

	<i>tp</i>	<i>fp</i>	<i>fn</i>	<i>Precision</i> %	<i>Recall</i> %
Collinear patterns	86	2	4	97.72	95.56
Curvilinear patterns	75	6	6	92.59	92.59



Figure 7. (a) Collinear patterns recognized in Dataset A. (b), (c) Enlarged views.

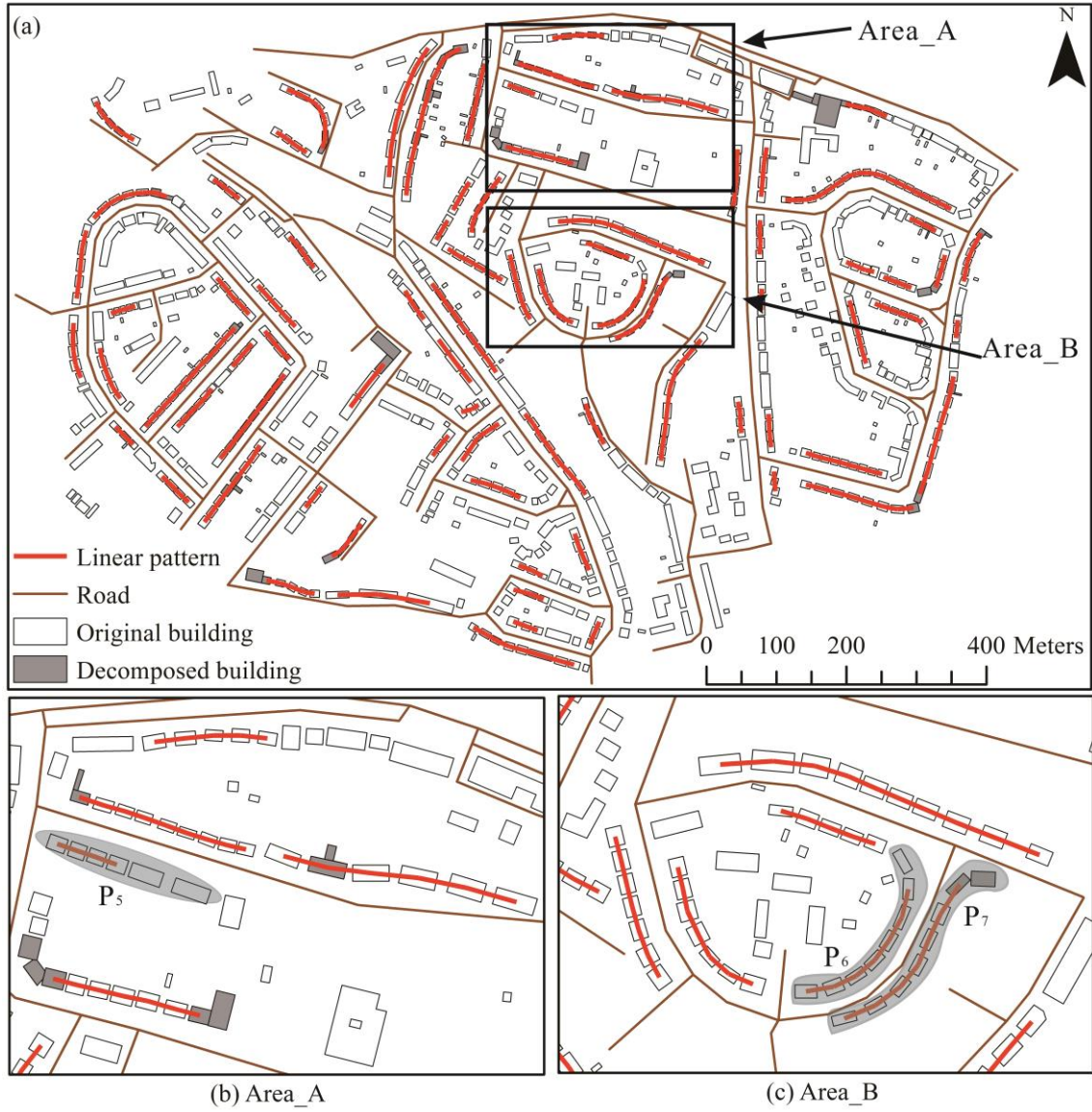


Figure 8. (a) Curvilinear patterns recognized in Dataset B. (b), (c) Enlarged views.

5.3 Parameter analysis

In this section, we analyze the influence of parameter setting on proposed results, mainly involving the parameters to measure the proximity, similarity, and continuity between buildings.

(1) Proximity

We used the RNG to represent the proximity relation between buildings in our approach. The proximity relation can also be described by other proximity graphs (Wei et al., 2018). If the DT-like proximity graph is applied to recognize linear building

patterns, some unexpected patterns may be obtained, e.g., the pattern in Figure 9(a). It is because the DT-like proximity graph may have some visually denied edges for linear pattern recognition. If the MST is applied to recognize linear patterns, some visually aware patterns may fail to be recognized, e.g., the pattern in Figure 9(b). It is because only the local shortest edges are connected in MST (Zhang et al, 2013). As compared by Wei et al. (2018), RNG is proved to be the most sufficient to detect linear building patterns, and RNG is recommended for linear pattern recognition.

(2) Similarity

δ_1 , δ_2 and δ_3 are the thresholds to rule the similarity between buildings in size, orientation, and shape. If large δ_1 , δ_2 and δ_3 are set, buildings with a large difference in size, orientation, or shape may be recognized in a linear pattern. For example, if $\delta_1=4$, the two buildings connected by the blue line in Figure 9(f) will be recognized as a linear pattern; or if $\delta_2=35^\circ$, the two buildings connected by the blue line in Figure 9(c) will be recognized as a linear pattern. But these two patterns may be denied by users. If small δ_1 , δ_2 and δ_3 are set, some visually aware patterns may fail to be recognized. For example, if $\delta_1=1.2$, the pattern in Figure 9(d) can't be recognized. As shown in Table 6, most buildings in topographical maps tend to have simple shapes, and $\delta_1=2$, $\delta_2=15^\circ$ and $\delta_3=1.4$ are recommended in collinear pattern recognition; δ_2 may be a little larger in curvilinear pattern recognition, and $\delta_2=25^\circ$ is recommended in practice.

(3) Continuity

η^1 and η^2 are the main thresholds to rule whether the buildings distribute along a defined line. η^2 is set smaller in collinear pattern recognition than it is in curvilinear pattern recognition, and a threshold for *alignAngle* is also set in curvilinear pattern recognition. *FR* is mainly defined to classify the collinear patterns into straight-line patterns and oblique patterns, and *FR* can also be set as $FR \in [0,1]$ in practice. If large

η^1 and η^2 are set, some unexpected patterns may be recognized. For example, if $\eta_2=150^\circ$, the two buildings connected by the blue line in Figure 9(g) will be recognized as a linear pattern. But the pattern may be denied by users. If small η^1 and η^2 are set, some visually aware patterns may fail to be recognized. For example, if $\eta_1=1.2$, the pattern in Figure 9(e) can't be recognized. And $\eta_1=2$, $\eta_2=165^\circ$ are recommended in collinear pattern recognition, and $\eta_1=2$, $\eta_2=155^\circ$ are recommended in curvilinear pattern recognition according to our experiments.

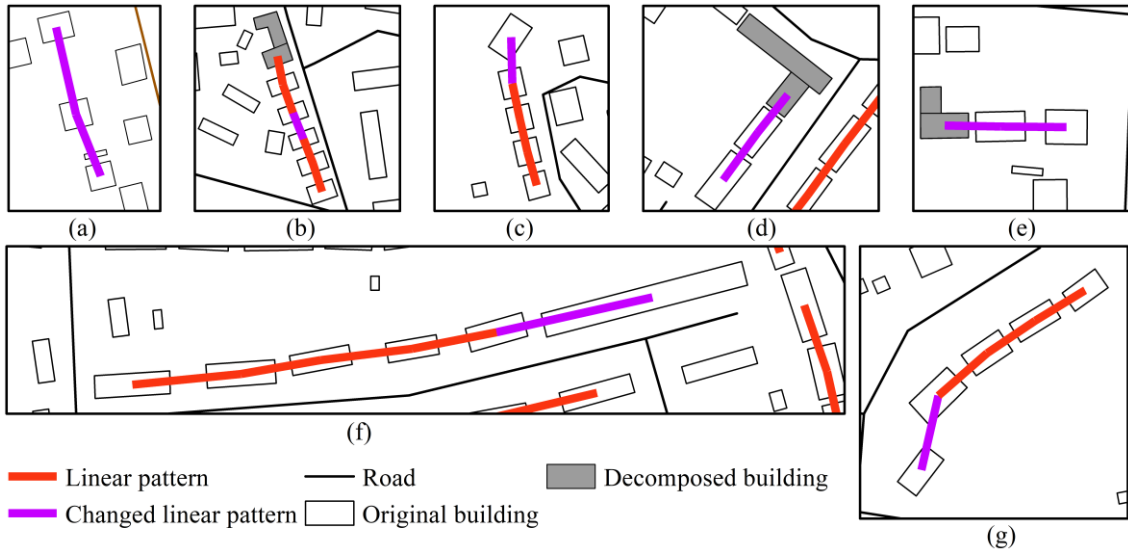


Figure 9. Examples for recognized linear patterns with changed parameters, and the changed linear patterns compared to Figures 7 and 8 are colored blue. (a) DT-like proximity graph; (b) MST; (c) $\delta_2=35^\circ$; (d) $\delta_1=1.2$; (e) $\eta_1=1.2$; (f) $\delta_1=4$; (g) $FR \in [0,1]$, and $\eta_2=150^\circ$.

6. Discussion

6.1 Comparisons with existing approaches

In this section, we compared the proposed approach with existing approaches. The collinear patterns are recognized in Dataset A according to Wei et al. (2018) with the same parameters with RNG; and curvilinear patterns are recognized according to Zhang et al. (2013) with the same parameters with RNG. The compared results are shown in

Table 8. From Table 8 we have the following observations: (1) 22 collinear patterns are misrecognized and 33 collinear patterns can't be recognized with the existing approach; and the proposed approach has 25.57% higher *Precision* and 32.23% *Recall* by comparing to the existing approach; (2) 18 curvilinear patterns are misrecognized, and 21 curvilinear patterns can't be recognized with existing approach; and the proposed approach have 15.67% higher *Precision* and 18.52% *Recall* by comparing to the existing approach.

Four typical areas are also selected for a detailed analysis (Figure 10). As shown in Figure 10, the linear patterns P₁, P₂, P₄, P₆, P₇, and P₉ can only be partially recognized, and P₃, P₅, and P₈ can't be recognized with existing approaches. But these linear patterns can all be recognized with the proposed approach combining convex polygon decomposition, and these linear patterns can all be aware by users. As shown in Figure 10, these linear patterns can only be partially recognized or can't be recognized mainly because buildings may sometimes have different sizes, shapes, or orientations in a local area. These occasions are very common in urban areas. Our approach may be more efficient for the areas in which buildings are not all with similar characteristics by comparing to existing approaches. The comparisons show that it is helpful to decompose building for linear pattern recognition, which means the part-whole hierarchies can enhance a richer set of semantic concepts and relations of map data to some extent. These semantic concepts and relations can then improve their based applications such as urban form understanding, automated map generalization, and 3D city model visualization.

Table 8. Compared results between the proposed approach and existing approaches.

		<i>tp</i>	<i>fp</i>	<i>fn</i>	<i>Precision</i> %	<i>Recall</i> %
Collinear patterns	The proposed approach	86	2	4	97.72	95.56
	Existing approach	57	22	33	72.15	63.33
Curvilinear patterns	The proposed approach	75	6	6	92.59	92.59
	Existing approach	60	18	21	76.92	74.07

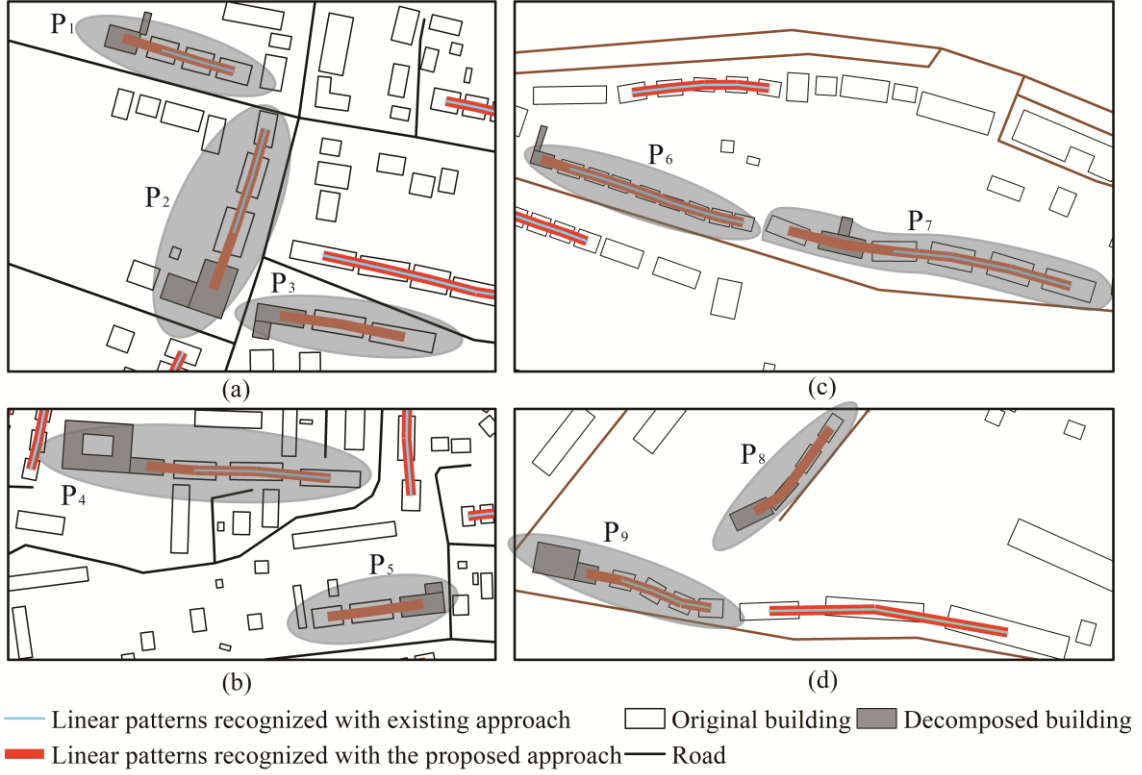


Figure 10. Compared results between the proposed approach and existing approaches in four typical areas, in which the patterns circled out can be recognized by the proposed approach, but can't be recognized or only can be partially recognized with existing approaches.

6.2 Effectiveness for the area with complex buildings

To evaluate the effectiveness of the proposed approach for buildings with complex shapes, an area (346 buildings and 223 roads) in which buildings are with complex shapes is also selected for the experiment. Collinear patterns are recognized in this area with parameters setting as: $\delta_1=2$, $\delta_2=15^\circ$, $\delta_3=1.4$, $\eta_1=165^\circ$, $\eta_2=2$, $FR \in [0.4, 1.0]$. 13 collinear patterns are recognized in the area with 8 buildings decomposed, in which 6 collinear patterns can only be recognized combining convex polygon decomposition (Figure 11). It means our approach can also be applied to buildings with complex shapes.

But three points need to point out here: (1) Buildings with complex shapes are more likely to arrange irregularly, and regular patterns including collinear patterns and curvilinear patterns may not be common in areas with complex buildings, e.g., Area_A

and Area_B in Figure 11; Furthermore, the recognized patterns in Figure 11 mostly tend to have simple shapes, which means a building with complex shape may not be easy to form linear patterns with adjacent buildings. (2) We measure the shape similarity between buildings based on the edge count (EC), but EC may not be proper for buildings with complex shapes; and other parameters need to be introduced, e.g., similarity measurement based on turning angle (Yan et al., 2018), but it may also increase the computational cost. (3) Buildings with complex shapes may have many concave nodes, and more cuts may be needed to recognize a pattern; a strategy to reduce the cuts may be needed. For example, some cuts may be superfluous for pattern recognition and can be deleted.

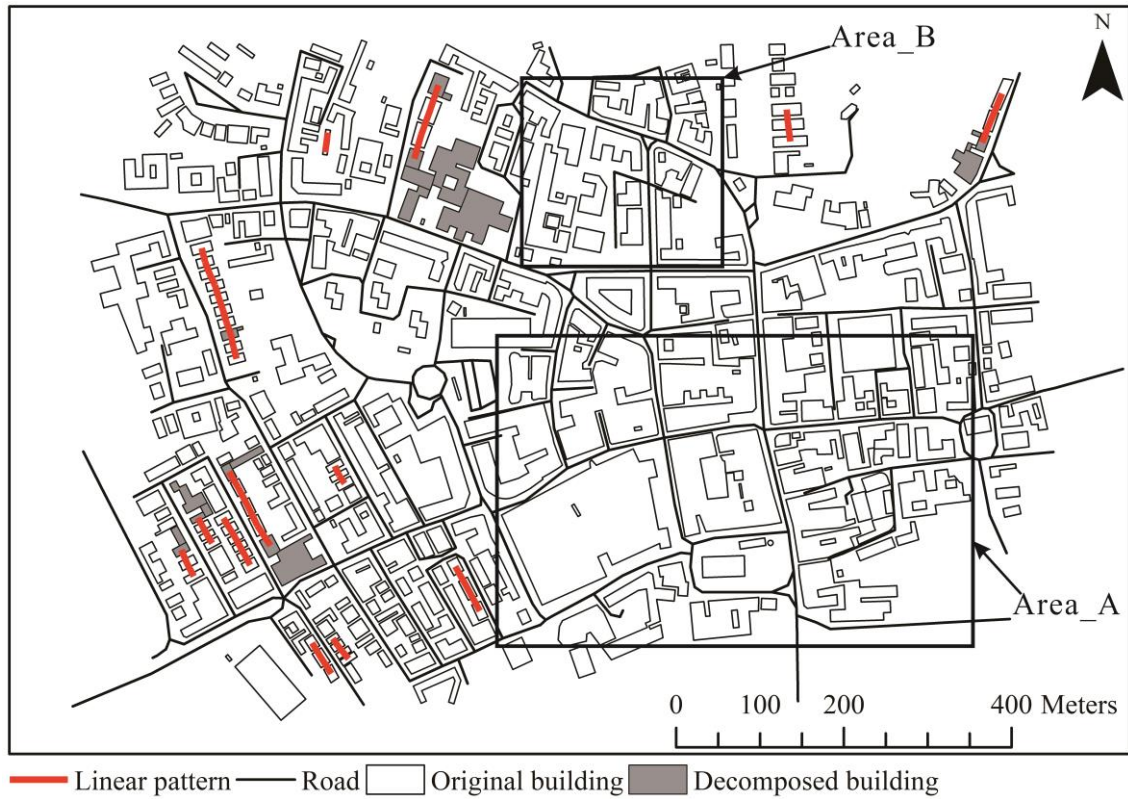


Figure 11. Collinear patterns are recognized in an area where buildings are with complex shapes.

6.3 Recognition of other kinds of building patterns

Buildings are considered as a whole in all existing building pattern recognition approaches, and we have recognized collinear patterns and curvilinear patterns successfully with the proposed approach combining convex polygon decomposition. If

the buildings are decomposed first, other building pattern recognition approaches can also be applied, e.g., C-shaped and T-shaped pattern recognition. As shown in Figure 12, C-shaped and T-shaped patterns are recognized based on the approach of Gong et al. (2018) with building decomposed first. And the C-shaped pattern (P_1 - P_5) and T-shaped pattern (P_6 - P_9) can only be recognized with polygon decomposition, and these patterns are also user-aware.

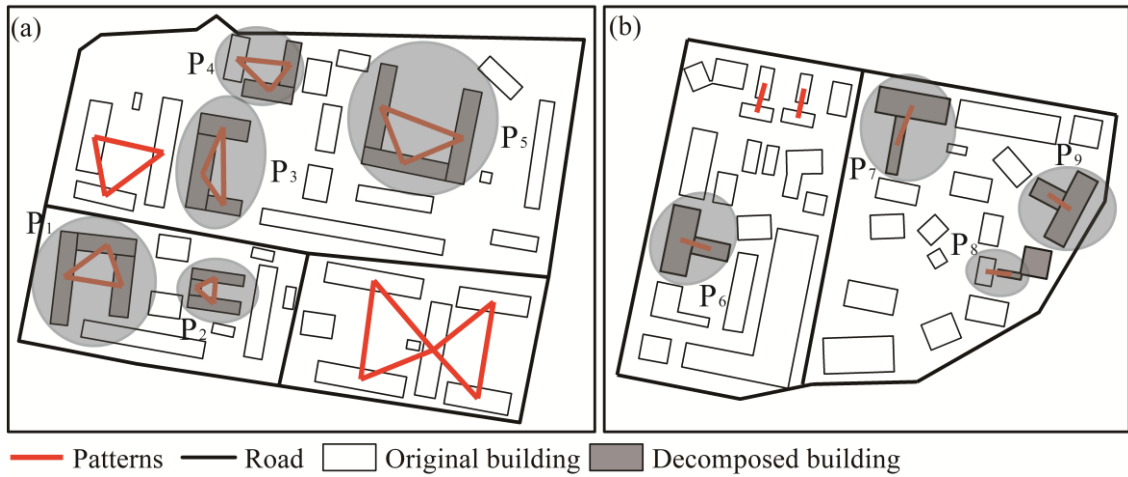


Figure 12. Recognition results of C-shaped and T-shaped patterns combining convex polygon decomposition.

7. Conclusions

To recognize linear building patterns in topographical maps, this study first proposed an approach to recognize building patterns combining convex polygon decomposition according to the visual perception process. In this approach, the buildings can be a whole or decomposed into parts for linear pattern recognition. Linear building patterns including collinear ones and curvilinear ones were recognized successfully in two datasets. And the results indicated that a higher accuracy and recall can be achieved with our approach when compared with the existing approaches. We further proved that other kinds of building patterns including T-shaped and C-shaped patterns can also be recognized with convex polygon decomposition. In the building polygon decomposition, a new node concavity is also developed which is applicable for a polygon with holes. Our future

works will focus on refining our approach. Decomposition rules or pattern recognition rules may vary according to different user requirements or applications. But these rules are defined in our approach based on strict graph-isomorphism, and these rules can be learned with data enrichment and machine learning algorithms in the future.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Data and code availability statement and data deposition

The data and code that support the findings of this study are all openly available, website as: <https://github.com/TrentonWei/Linear-Building-Pattern-Recognition->

Geolocation information

Two datasets at a nominal viewing scale of 1:10, 000 are collected from the study area with an ID as ‘*TL*’ in the ‘OS OpenMap Local’ product. Website for the product as: <https://osdatahub.os.uk/downloads/open/OpenMapLocal>

References

- Ai T, Ke S & Yang M, et al. (2017). Envelope generation and simplification of polylines using Delaunay triangulation[J]. International Journal of Geographical Information Science, 31(2): 297-319.
- Anders K H & Sester M. (2001). Parameter-free cluster detection in spatial databases and its application to typification. In ISPRS Congress, Amsterdam. Retrieved from: <http://www.ifp.uni-stuttgart.de/publications/2000/Anders1573.pdf>
- Anders K H (2006). Grid typification. Proceedings of the 12th International Symposium on Spatial Data Handling Springer, Berlin, 633–642.

- Basaraner M & Cetinkaya S. (2017) Performance of shape indices and classification schemes for characterizing perceptual shape complexity of building footprints in GIS, *International Journal of Geographical Information Science*, 31(10): 1952-1977.
- Christophe S & Ruas A. (2002). Detecting building alignments for generalization purposes. *Advances in Spatial Data Handling*, Heidelberg.
- Dash K S, Puan N B, Panda G. (2018). Unconstrained handwritten digit recognition using perceptual shape primitives[J]. *Pattern Analysis and Applications*, 21(2): 413-436.
- Department of Geography, University of Zurich. (1999). Selection of basic measures. Retrieved from: <http://agent.ign.fr/deliverable/DC1.html>.
- Duchêne C, Bard S & Barillot X. (2003). Quantitative and qualitative description of building orientation. 5th Workshop on Progress in Automated Map Generalization, Paris.
- Du S, Shu M & Feng C. (2015a). Representation and discovery of building patterns: A three-level relational approach. *International Journal of Geographical Information Science*, 30(6): 1161–1186.
- Du S, Zhang F & Zhang X. (2015b). Semantic classification of urban buildings combining VHR image and GIS data: an improved random forest approach. *ISPRS Journal of Photogrammetry and Remote Sensing*, 105: 107–119.
- Du S, Luo L & Cao K, et al. (2016). Extracting building patterns with multilevel graph partition and building grouping[J]. *ISPRS Journal of Photogrammetry and Remote Sensing*, 122: 81-96.
- Du S, Shu M & Wang Q. (2019). Modelling relational contexts in GEOBIA framework for improving urban land-cover mapping[J]. *GIScience & Remote Sensing*, 56(2): 184-209.

- Ghosh M, Amato N M & Lu Y, et al. (2013). Fast approximate convex decomposition using relative concavity[J]. CAD Computer Aided Design, 45(2): 494-504.
- Gong X, Wu F & Qian H, et al. (2014). The parameter discrimination approach to multi-connected linear pattern recognition in building groups. Geomatics and Information Science of Wuhan University, 39(3): 335–339.
- Gong X & Wu F. (2018). A graph match approach to typical pattern recognition in urban building groups. Geomatics and Information Science of Wuhan University, 43(1): 159–166.
- He X, Zhang X & Xin Q. (2018). Recognition of building group patterns in topographic maps based on graph partitioning and random forest[J]. ISPRS Journal of Photogrammetry and Remote Sensing, 136: 26-40.
- Hu X, Fan H & Noskov A. (2018). Roof model recommendation for complex buildings based on combination rules and symmetry features in footprints[J]. International Journal of Digital Earth, 11(10): 1039-1063.
- Hu X, Fan H & Noskov A, et al. (2021) Room semantics inference using random forest and relational graph convolutional networks: A case study of research building[J]. Transactions in GIS, 25(1): 71-111.
- Juengling R & Mitchell M. (2007) Combinatorial shape decomposition[C]// Advances in Visual Computing, Third International Symposium, Isvc, Lake Tahoe, Nv, USA, November. DBLP.
- Li Z, Hu J & Stojmenovic M, et al. (2020). Revisiting spectral clustering for near-convex decomposition of 2D shape[J]. Pattern Recognition, 105:107371.
- Lien J M & Amato N M. (2006). Approximate convex decomposition of polygons[J]. Computational Geometry, 35(1-2):100-123.

- Lipson H & Shpitalni M. (1996). Decomposition of a 2D polygon into a minimal set of disjoint primitives[C]//CSG96 Conference on Set-Theoretic Solid Modeling, 65-82.
- Liu R & Zhang H. (2004). Segmentation of 3D meshes through spectral clustering[C]//12th Pacific Conference on Computer Graphics and Applications, 2004. PG 2004. Proceedings. IEEE, 298-305.
- Liu Y, Guo Q & Sun Y, et al. (2014). A combined approach to cartographic displacement for buildings based on skeleton and improved elastic beam algorithm[J]. Plos one, 9(12): e113953.
- Lu W & Ai T. (2020). Center point extraction of simple area object using Triangulation skeleton graph[J]. Geomatics and Information Science of Wuhan University, 45(3): 337-343.
- Luo L, Shen C & Liu X, et al. (2014). A computational model of the short-cut rule for 2D shape decomposition[J]. IEEE Transactions on Image Processing, 24(1): 273-283.
- Lüscher P, Weibel R & Burghardt D. (2009). Integrating ontological modelling and Bayesian inference for pattern classification in topographic vector data[J]. Computers, Environment and Urban Systems, 33(5): 363-374.
- Mao B, Harrie L & Ban Y. (2007). Detection and typification of linear structures for dynamic visualization of 3D city models[J]. Computers, Environment and Urban Systems, 6(3): 233-244.
- Mi X & DeCarlo D. (2007). Separating parts from 2d shapes using relatability[C]// 2007 IEEE 11th International Conference on Computer Vision. IEEE, 1-8.2012, 3.
- Nedas K A & Egenhofer M J. (2008). Spatial-scene similarity queries. Transactions in GIS, 12 (6): 661-681.

- Niu N, Liu X & Jin H, et al. (2017). Integrating multi-source big data to infer building functions[J]. *International Journal of Geographical Information Science*, 31(9): 1871-1890.
- Ordnance Survey. OS OpenMap-Local [OS data © Crown Copyright], 2021. Data source as: <https://www.ordnancesurvey.co.uk/business-government/tools-support/open-map-local-support>.
- Pepe M, Costantino D, Alfio VS, et al. (2021). A Novel Method Based on Deep Learning, GIS and Geomatics Software for Building a 3D City Model from VHR Satellite Stereo Imagery. *ISPRS International Journal of Geo-Information*, 10(10):697.
- Pilehforooshha P & Karimi M. (2019). An integrated framework for linear pattern extraction in the building group generalization process[J]. *Geocarto International*, 34(9): 1000-1021.
- Qu W, Ma M & Li Z, et al. (2017). A coarse-to-fine shape decomposition based on critical points[J]. *Concurrency and Computation Practice and Experience*, 29(2): e4088.
- Rainsford D & Mackaness W. (2002). Template matching in support of generalization of rural buildings. *The 10th International Symposium on Spatial Data Handling*, Springer, Berlin, 137–151.
- Ren Z, Yuan J & Liu W. (2013). Minimum near-convex shape decomposition[J]. *IEEE transactions on pattern analysis and machine intelligence*, 35(10): 2546-2552.
- Renard J & Duchêne C. (2014). Urban structure generalization in multi-agent process by use of reactional agents. *Transactions in GIS*, 18 (2): 201–218.
- Rodrigues R S V, Morgado J F M & Gomes A J P. (2018). Part-based mesh segmentation: a survey[C]//*Computer Graphics Forum*. 37(6): 235-274.
- Rosser J F, Boyd D S & Long G, et al. (2019). Predicting residential building age from map data[J]. *Computers, Environment and Urban Systems*, 73: 56-67.

- Ruas A. (1998). A method for building displacement in automated map generalization. *International Journal of Geographical Information Science*, 12(8): 789-803.
- Ruas A & Holzapfel F. (2003) Automatic characterization of building alignments by means of expert knowledge[C]//21th ICC conference. 1604-1616.
- Saeidi S, Mirkarimi S H, Mohammadzadeh M, et al. (2019). Assessing the visual impacts of new urban features: coupling visibility analysis with 3D city modelling[J]. *Geocarto International*, 34(12): 1315-1331.
- Sahbaz K, Basaraner M. (2021). A zonal displacement approach via grid point weighting in building generalization[J]. *ISPRS International Journal of Geo-Information*, 10(2): 105.
- Sester M. (2005). Optimization approaches for generalization and data abstraction[J]. *International Journal of Geographical Information Science*, 19(8-9): 871-897.
- Singh M, Seyranian G D & Hoffman D D. (1999). Parsing silhouettes: The short-cut rule[J]. *Perception & Psychophysics*, 61(4): 636-660.
- Steiniger S. (2007). Enabling pattern-aware automated map generalization[D]. University of Zurich.
- Taillandier P, Duchêne C & Drogoul A. (2011) Automatic revision of rules used to guide the generalization process in systems based on a trial and error strategy, *International Journal of Geographical Information Science*, 25(12): 1971-1999.
- Wang X & Burghardt D. (2019). A mesh-based typification method for building groups with grid patterns[J]. *ISPRS International Journal of Geo-Information*, 8(4): 168.
- Wang X, Burghardt D. (2020). Using stroke and mesh to recognize building group patterns[J]. *International Journal of Cartography*, 2020, 6(1): 71-98.
- Wang X, Burghardt D. (2021). A typification method for linear building groups based on stroke simplification[J]. *Geocarto International*, 36(15): 1732-1751.

- Wang C & Lai Z. (2016). Shape decomposition and classification by searching optimal part pruning sequence[J]. *Pattern Recognition*, 206-217.
- Wei Z, Guo Q & Wang L, et al. (2018). On the spatial distribution of buildings for map generalization[J]. *Cartography and Geographic Information Science*, 45(6): 539-555.
- Wei Z, He J & Wang L, et al. (2018). A collaborative displacement approach for spatial conflicts in urban building map generalization[J]. *IEEE Access*, 6: 26918-26929.
- Wei Z, Liu Y, Cheng L, Ding S. (2021). A Progressive and Combined Building Simplification Approach with Local Structure Classification and Backtracking Strategy. *ISPRS International Journal of Geo-Information*, 10(5): 302.
- Xing R, Wu F, Gong X, et al. (2021). An axis-matching approach to combined collinear pattern recognition for urban building groups[J]. *Geocarto International*, 1-20.
- Yan H, Weibel R & Yang B. (2008). A multi-parameter approach to automated building grouping and generalization. *Geoinformatica*, 12(1): 73–89.
- Yan X, Ai T & Yang M, et al. (2019) A graph convolutional neural network for classification of building patterns using spatial vector data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 150, 259-273.
- Yang W. (2011). Identify building patterns. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*. Göttingen: Copernicus GmbH, 391–398.
- Zhang L, Deng H & Chen D. (2013a). A spatial cognition-based urban building clustering approach and its applications. *International Journal of Geographical Information Science*, 27(4), 721–740.
- Zhang X, Ai T & Stoter J. (2013b). Building pattern recognition in topographic data: Examples on collinear and curvilinear alignments. *Geoinformatica*, 17(1), 1–33.

Zhang X, Stoter J & Ai T, et al. (2013c). Automated evaluation of building alignments in generalized maps. *International Journal of Geographical Information Science*, 27(8): 1550–1571.

Zhang X, Du S & Wang Y. (2015). Semantic classification of heterogeneous urban scenes using intra-scene feature similarity and inter-scene semantic dependency. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 8 (5): 2005-2014.

Zhao R, Ai T & Yu W, et al. (2020). Recognition of building group patterns using graph convolutional network. *Cartography and Geographic Information Science*, 47(5): 400-417.

Zunic J & Rosin P L. (2002). A convexity measurement for polygons, in: *British Machine Vision Conference*, 173–182.