International Journal of
Geo-Information

MDPI

Article

Employing Incremental Outlines for OpenStreetMap Data Updating

Hanfa Xing [1,*], Yuan Meng [2,*], Jun Chen [3], Jie Song [2] and Kaixuan Fan [2]

[1] School of Geography, South China Normal University, Guangzhou 510631, China
[2] College of Geography and Environment, Shandong Normal University, Jinan 250300, China; songjie@stu.sdnu.edu.cn (J.S.); fankaixuan@stu.sdnu.edu.cn (K.F.)
[3] National Geomatics Center of China, Beijing 100830, China; chenjun@nsdi.gov.cn
[*] Correspondence: xinghanfa@163.com (H.X.); mengyuan@stu.sdnu.edu.cn (Y.M);
Tel.: +86-020-852-113-80 (H.X.)

Received: 23 April 2018; Accepted: 19 July 2018; Published: 20 July 2018

Abstract: The updating of changing information plays a significant role in ensuring the quality of OpenStreetMap, which is usually completed by mapping the whole changing objects with a high degree of uncertainty. The incremental object-based approach provides opportunities to reduce the unreliability of data, while challenges of data inaccuracy and redundancy remain. This paper provides an incremental outline-based approach for OpenStreetMap data updating to solve this issue. First, incremental outlines are delineated from the changed objects and distinguished through a spatial classification. Then, attribute information corresponding to incremental outlines is proposed to assist in describing the physical changes. Finally, through a geometric calculation based on both the spatial and attribute information, updating operations are constructed with a variety of rules to activate the data updating process. The proposed approach was verified by updating an area in the OpenStreetMap datasets. The result shows that the incremental outline-based updating approach can reduce both the time and storage costs compared to incremental objects and further improve data quality in the updating process.

Keywords: OpenStreetMap; updating; incremental outlines; incremental changes; spatial data

1. Introduction

Developments in information and communication technologies have made access to spatial data easily available and inexpensive [1]. As one of the major sources in the spatial data contribution, OpenStreetMap (OSM), which was launched in 2004 [2], has gained much attention and been used increasingly in a variety of domains, including natural resource management, urban planning and regional decision making [3–5]. Its immense potential has also been used in studies and has become a subject in the academic sphere [6]. Through an online Flash-based editor, users assuming the role of spatial data providers are capable of updating geographical features [7]. With the increase in registered users, the updating rate of OSM data has risen dramatically, and the contributed data have been mapped across the world.

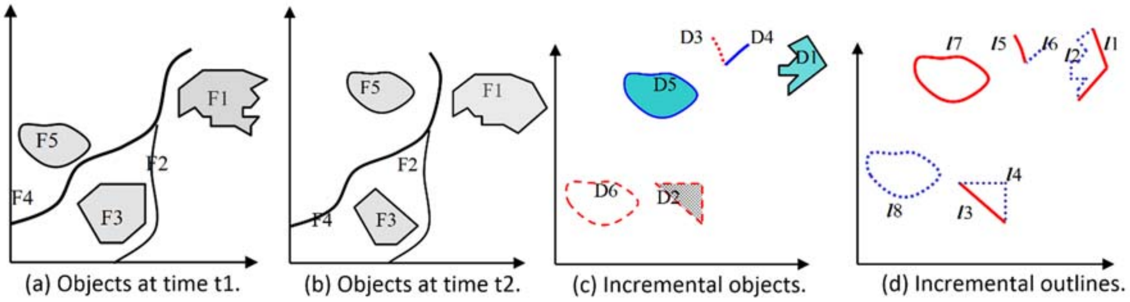
Although the user-oriented updating process has been proven to be an effective method for data collection, quality assurance remains a serious issue within the use of OSM data for creating reliable sources. The major weaknesses that limit further applications of OSM data have been investigated in existing studies. The first issue that has been highlighted is the geometric accuracy, which is the evaluation of both the positions and shapes of OSM data. Mooney et al. [8] implemented a shape similarity metric to assess OSM polygons based on the ground truth data. Shape assessment on building footprint in OSM data has also been focused by proposing quality indicators [9,10]. Results from these studies reveal a considerable proportion of dissimilarity in OSM data, indicating the obvious limitation in shape accuracy. In terms of position accuracy, Girres and Touya [11] proposed comparisons between the OSM data in France and a reference dataset. By utilizing different distance measures, position inconsistencies were found in point, linear and polygon features. The second problematic aspect lies in the spatial heterogeneity of OSM data, including country-level, region-level and city-level distributions. Yang et al. [12] explored the distribution inequality by comparing the OSM data in four countries: Germany, France, the United States and the Netherlands. In addition, OSM data have also been assessed in different regions in China, where a density analysis was proposed to visualize the spatial heterogeneity more specifically [13]. Regarding the city-level data distribution, Hagenauer and Helbich [14] mentioned that urban areas have been mapped with more sufficient data than other areas in terms of both quality and quantity.

The abovementioned two issues reveal the key challenges in the current OSM data updating process. On the one hand, the OSM data are mainly updated through the mapping of the whole changing objects, including both the changed and unchanged parts of the objects. This updating strategy significantly influences the data quality, as the repeated mapping by different users can cause inconsistencies in the same object in terms of both shape and position accuracy [15,16]. For example, one object, which only changed shape without moving in a short time period (such as one type of land use area), might be updated with the wrong position by other users. On the other hand, users play a significant role in the updating process, and there are no spatial restrictions and limitations for OSM data contribution. In other words, users are allowed to map whole objects regardless of the changing situation. This further causes the major issue of data inequality, namely spatial heterogeneity in variable scales of areas, as the mapping of the whole objects generates redundant data in partial regions and thus results in an uneven spatial distribution [17]. In summary, effective approaches and improvements are required in the updating of OSM data drawn from the discussed issues.

In fact, updating approaches of such spatial data, which basically involves two available modes, whole object-based updating and incremental object-based updating, have been a concern over the past few decades [18–21]. Generally, whole object-based updating aims to delineate change information based on the existing status of the entire object, whereas incremental object-based updating aims to label the changing part of the object solely. For a further inspection, these two different modes can be illustrated via Figure 1a–c. Specifically, Figure 1a,b presents the objects before changing at time *t1* and after changing at time *t2*, respectively. Objects captured by whole object-based updating are all the changed objects during the time intervals *t1*–*t2*, including objects *F1*, *F3*, *F4* and *F5* in Figure 1b. *F2* does not need to be captured for updating because it remains unchanged from *t1* to *t2*. Figure 1c demonstrates the objects generated by the incremental object-based updating. Objects in solid lines and polygons, including *D1*, *D4* and *D5*, represent the increased parts of the changed objects, whereas *D2*, *D3* and *D6* refer to the decreased parts. The above two modes represent the most common updating process and have stimulated many studies to further investigate updating strategies within the two modes.

Currently, a variety of effective approaches for data updating have been proposed in the existing studies. With regard to whole object-based updating, Claramunt and Thériault [22] divided the changing objects into eight types according to the basic, transformation and movement processes of the single entity. Hornsby and Egenhofer [23] proposed nine combinations of change detection operations based on the states of existence and non-existence for the identifiable objects. However, since only change information is required in the data updating process, the delineation of the whole object can actually result in data redundancy. On the other hand, the incremental object-based updating has the potential to address this limitation. Basically, the incremental changes described by existing studies can be summarized as three aspects: topology-driven changes, spatial overlapping-driven changes and event-driven changes. Topology-driven changes concentrate on topological relations among incremental changes. Chen and Zhou [24] designed an incremental updating method based on topological linkage and further illustrated it with the updating process of a cadastral database. Li and

Wang [25] automatically recognized the changes in residential areas through the topological relations between the old and new boundaries. Zhou et al. [26] proposed a computation model to refine the non-empty topological relations among polygon features. Overlapping-driven changes aim to capture the incremental changes through non-topology attributes such as size, shape, position and direction. Research proposed by Xing et al. [27] designed a descriptive model to capture incremental changes via multi-parameters, including identity, dynamics, dimension, shape, etc. Moreover, Wang et al. [28] proposed a hierarchical matching approach with the utilization of variable object attributes to identify different incremental changes. Event-driven changes are mainly based on identifying the types of changes before proposing incremental object-based updating. Fan et al. [29] proposed a new incremental change detection based on event semantics for cadastral database updating. Another study focused on the updating process and utilized position, geometry and attribute information to build a new description model. In summary, despite the variety of incremental object-based updating approaches, they all avoid updating the whole objects repeatedly and thus eliminate data inaccuracy and redundancy to some extent.



(a) Objects at time t1. (b) Objects at time t2. (c) Incremental objects. (d) Incremental outlines.

Figure 1. Comparison between incremental objects and outlines. *F1*–*F5* denote the linear and polygon objects in both times *t1* and *t2*. *D1*–*D6* denote the changing parts of the objects between time *t1* and *t2*. In particular, *D1*, *D4* and *D5* in solid lines and blue color represent the increased objects, while *D2*, *D3* and *D6* in dotted lines represent the decreased objects. *l1*, *l3*, *l5* and *l7* with red solid lines refer to the increased outlines, while *l2*, *l4*, *l6* and *l8* with blue dotted lines refer to the decreased outlines. (**a**) Objects at time *t1*; (**b**) Objects at time *t2*; (**c**) Incremental objects; (**d**) Incremental outlines.

According to the fact that the current OSM data generating strategies are based on the whole object-based updating process, which causes the abovementioned issues of OSM data quality, one can deduce that the incremental object-based updating can fill the gap between updating OSM data and ensuring data quality. However, there remain fundamental questions in capturing incremental changes utilizing the existing approaches. One is that the OSM website has implemented dedicated editing tools and technical infrastructures to support the current whole object-based updating strategy, and it may take effort for the OSM developers to modify the existing environment. Apart from that, the more important issue is that the proposed incremental object-based approaches cannot completely meet the needs of OSM updating for ensuring data quality. Because massive amounts of new data could be generated with high frequency on a worldwide scale, extensive data storage is still inevitable, even though only incremental objects need to be mapped. More specifically, approaches including topology, spatial overlapping and event-driven changes need to delineate both polygons and outlines of the changed objects, including the outlines that are shared with the unchanged parts of the objects. As a result, issues about data inaccuracy and redundancy still remain in this type of data updating process.

Faced with these challenges, it is necessary to develop extensive approaches for effectively updating OSM data. In fact, mapping newly-generated outlines (namely, incremental outlines) solely instead of the corresponding object might provide an opportunity to solve these challenges. Illustrations are shown in Figure 1c,d. Figure 1c represents the incremental objects, namely the changing parts of the objects in the time period *t1*–*t2*. Figure 1d displays the outlines of these changing objects in Figure 1c. While the object entities need to be labelled in Figure 1c, such as *D1*, *D2* and *D5*, Figure 1d only requires corresponding outlines *l1*, *l3* and *l7*, with *l2* and *l4* not needing to be mapped in the updating process. Therefore, both costs and time can be greatly reduced in updating large quantities of OSM data.

Based on this hypothesis, this paper proposes a novel approach to employ incremental outlines for OSM data updating. First, incremental outlines are classified into different types based on the spatial information. Second, attribute information is concerned with describing the physical changes of the incremental outlines. Finally, both geometric calculation and updating rules are involved in helping to construct different operations. These operations are considered triggers to activate the target incremental outlines for OSM data updating.

The contribution of this paper lies in the following:

(1) A novel data updating framework, namely an incremental outline-based approach, is proposed for OSM data updating to solve the problem of data inaccuracy and redundancy.
(2) Spatial classification and attribute information are proposed to describe both the spatial and physical changes of the incremental outlines.
(3) Updating operations are designed based on the utilization of incremental outlines to activate the updating process of the OSM data.

The rest of the paper is organized as follows. Section 2 presents the novel methodology of employing incremental outlines for OSM data updating. Section 3 applies the proposed approach to an experimental dataset and discusses the results by describing the updating process using incremental outlines and further comparing the efficiency with incremental objects. Section 4 discusses the research results and provides conclusions for this study.

2. Methodology

2.1. Overview

The overall framework of the proposed approach consists of three main parts, which are shown in Figure 2. Generally, both spatial and attribute information are inevitable in the data updating process. Thus, the first two steps focus on describing the incremental changes from the perspective of space and attributes, while the third step constructs effective rules to update incremental changes based on the above spatial and attribute information.

In the first step, the concept of incremental outlines is introduced to delineate the incremental changes. This is illustrated by the incremental changes including lines and areas, which are extracted from OpenStreetMap in different time periods. Based on this concept, spatial classification is proposed to classify the incremental changes effectively.

In the second step, the changes are distinguished based on incremental outlines within a variety of attributes. In particular, 5 attributes are considered for the incremental outlines' description, including object identity and transitions, topological relations, shape, position and orientation.

In the third step, strategies in data updating are proposed based on the delineation of the incremental outlines. First, we involve geometric calculations including "intersection" and "difference" to quantify the incremental outlines. Based on these quantifications, 7 operations are proposed to distinguish different kinds of change updating, which include "Create", "Delete", "Update", "Merge", "Split", "Move" and "Rotate". In addition, incremental updating rules are constructed to activate 7 operations for incremental updating.

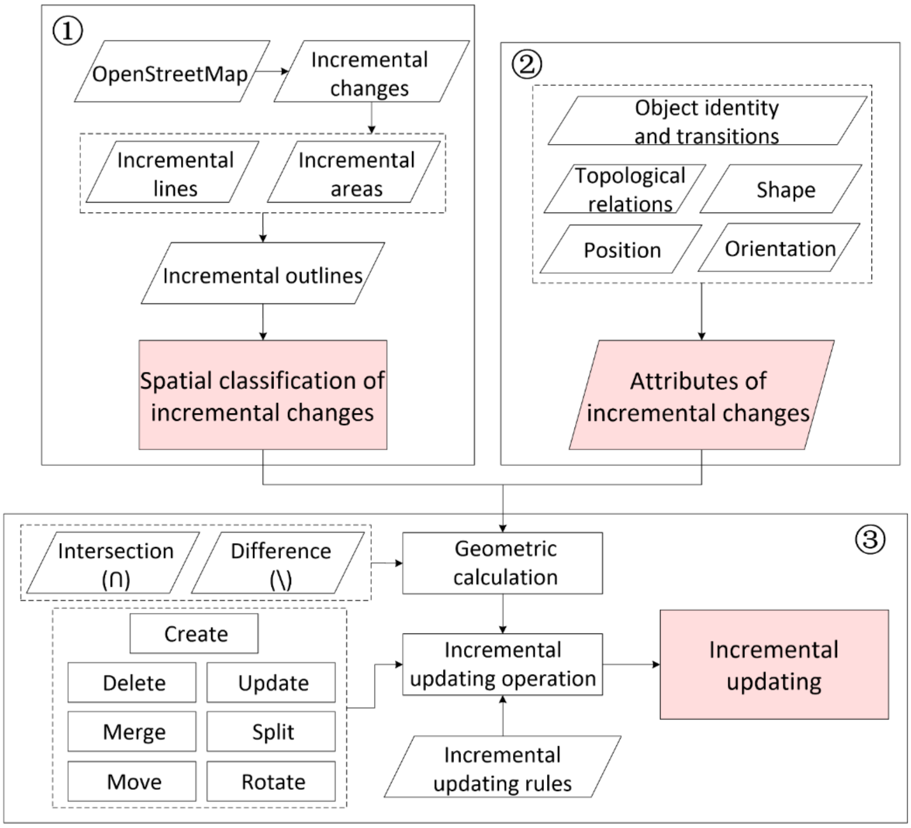


Figure 2. The framework of the methodology.

2.2. *Spatial Classification of Incremental Changes*

The detection and classification of information about object changes are vitally important in the process of incremental updating [30]. General methods for delineating the changes depend on depicting the incremental object. This section introduces a novel approach, namely incremental outlines, to effectively capture the incremental changes, reducing the complexities and costs of utilizing an incremental object. Section 2.2.1 illustrates different conditions of incremental outlines derived from the definition of incremental objects, while Section 2.2.2 summarizes spatial classification types of incremental outlines.

2.2.1. Introduction of the Incremental Outlines

As discussed in Section 1, Figure 1d demonstrates the basic idea of constructing incremental outlines. Taking object *F1* as an example, for *F1* changing during the time period of *t1*–*t2*, the incremental change can be captured as the blue-colored area. This changed area consists of three parts, including outlines *l1*, *l2* and area *D1*. In the whole object-based updating process, the whole area *D1* must be mapped for capturing the changing object. However, as *l2* already existed in *F1* in time *t1*, it is not necessary to map l_2 repeatedly for labelling the changing part in time *t2*. Therefore, the changes can be described effectively by solely labelling the increased outline *l1* in time *t2*. Here, we define *l1* as an incremental outline.

Based on the abovementioned assumption, we distinguish different conditions of incremental outlines' labelling. These conditions are derived and compared with the classification of incremental objects [31], which are displayed in Tables 1 and 2. We first divide the change information into incremental lines and incremental areas. Then, each condition in labelling incremental outlines is explained based on the six types of incremental objects, including "positive increments", "negative increments", "deviant positive and negative increments", "extending positive increments", "shrinking negative increments" and "extending and shrinking increments". Thus, seven conditions for incremental lines labelling and seven conditions for incremental area labelling are summarized by utilizing incremental outlines.

2.2.2. Spatial Classification of Incremental Changes Based on Incremental Outlines

For the purpose of facilitating incremental changes' updating, we aim to classify different types of incremental outlines spatially based on the labelling conditions in Section 2.2.1.

However, challenges remain in utilizing the existing incremental outlines to form new objects. Taking one type of incremental outline, the "split target line", as an example, we illustrate this issue in Figure 3. Basically, the split operation separates the original object into two parts, and only one part remains as a new object. Based on this concept, the object outline is divided into two parts, namely, line *a* and line *b*, after the line splitting operation. The line *L* refers to the incremental outline after the change of the target object. Despite the incremental change being successfully captured by *L*, it remains a question to identify which part of the object outlines (line *a* and line *b*) should be reserved to form the new object. Adding an identifiable point on the remaining part of the object is an effective way of solving this problem. As shown in Figure 3, line *b* is labelled as the remaining outline, which will be utilized to form the new object. In other words, we define this process as "reserve identifiable points", which is included in the spatial classification of the incremental outlines.

Table 1. Distinguishing incremental outlines from incremental lines.

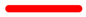
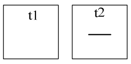
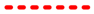
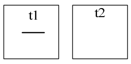
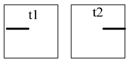
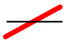
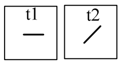
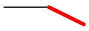
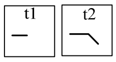
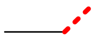
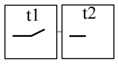
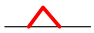
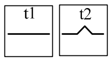
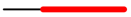
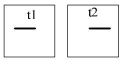
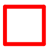
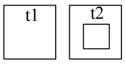
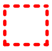
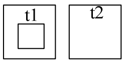
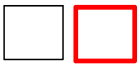
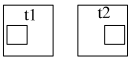
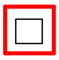
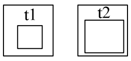
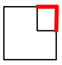
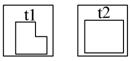
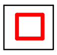
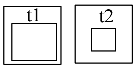
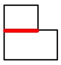
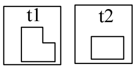
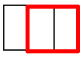
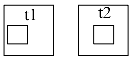
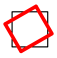
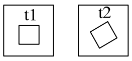
Incremental Types	Incremental Objects	Incremental Outlines	Legends
Incremental Lines	Positive increments	 Create target line	 Appearance of line
	Negative increments	 Delete target line	 Vanishing of line
	Deviant positive and negative increments	Move target line	 Movement of line, objects disjoint
		 Rotate target line	 Rotation of line
	Extending positive increments	 Extend target line	 Extending of line
	Shrinking negative increments	 Split target line	 Shrinking of line
	Extending and shrinking increments	 Divert target line	 Partial changes of line
		 Move target line	 Movement of line, objects intersect

Table 2. Distinguishing incremental outlines from incremental areas.

Incremental Types	Incremental Objects	Incremental Outlines	Legends
Incremental areas	Positive increments	 Create target line	 Appearance of area
	Negative increments	 Delete target line	 Vanishing of area
	Deviant positive and negative increments	 Move target line	 Movement of area, objects disjoint
	Extending positive increments	 Replace target line	 Expanding of area
		 Expand target line	 Partial expanding of area
	Shrinking negative increments	 Replace target line	 Shrinking of area
		 Split target line	 Partial shrinking of area
	Extending and shrinking increments	 Move target line	 Movement of area, objects intersect
		 Rotate target line	 Rotation of area

Based on the incremental differences in Section 2.2.1 and the above condition of reserving points, the labelling of incremental outlines can be spatially classified as 10 types, including "create target line", "delete target line", "extend target line", "merge target line", "move target line", "divert target line", "replace target line", "split target line", "rotate target line" and "reserve identifiable point", which are displayed in Figure 4.

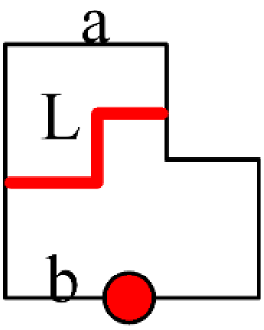


Figure 3. Reserving identifiable points.

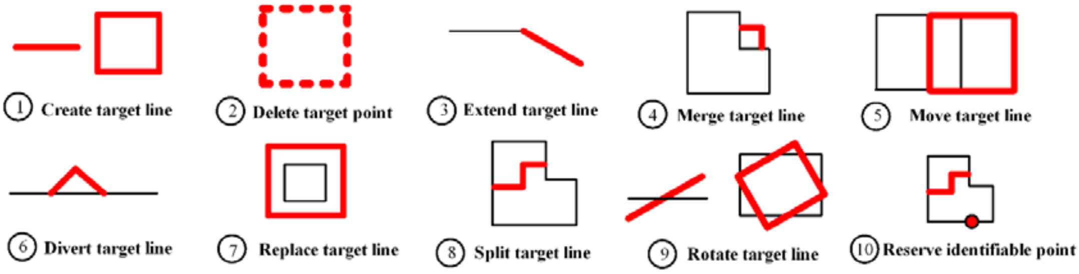


Figure 4. Spatial classification of incremental outlines.

2.3. *Attributes of Incremental Changes*

Information about incremental changes involves not only spatial classification of the changed objects, but also the attribute information that describes the physical changes of the objects [27,32]. Existing studies have employed a variety of attributes, including object identity and dynamics, topological relations and geometric features (such as shape, size, position and orientation) [33–35]. However, since these attributes were proposed to describe the whole object and the incremental changes solely, they cannot meet the requirement of providing effective information about incremental outlines. Thus, five attributes are modified for incremental outline descriptions, which are shown in the following aspects:

(1) Object identity and transitions.

The identity and transitions of objects form one of the fundamental elements in the change detection and updating process. Specifically, object identity aims to distinguish an object from others, indicating the individuality and uniqueness of the object [36], while transitions refer to the object dynamics from one identity state to another [23]. Here, the identity and transitions of incremental outlines T_{Ic} are defined as the following four types:

$$T_{Ic} = \{Create, Elimination, Continue - existence, Reincarnate\} \tag{1}$$

where the corresponding values of four types are defined as:

$$T_{Ic}(Value) = \{02, 21, 22, 12\} \tag{2}$$

Thus, the identity and dynamics of incremental outline I_c can be defined as:

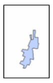
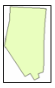
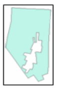
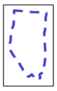
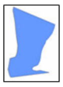
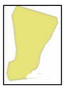
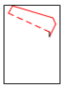
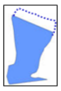
$$I_c = (T_{Ic}(Value), O_{id}) \tag{3}$$

where O_{id} refers to the identified number of incremental outline I_c.

(2) Topological relations.

Topological relations serve as a significant indicator in depicting incremental updating [33]. Examples are illustrated in Table 3. From the perspective of incremental objects, the changes can be classified as "positive increments", which are represented in the fourth column in Table 3. On the other hand, as the changed object contains the original object in terms of topological relations, the incremental object can be updated by solely labelling the outline of the changed object (see Column Incremental Outlines in Table 3). Therefore, topological relations are considered effective information about incremental change attributes.

Table 3. Topological relations among objects before and after changes.

Incremental Type	Before Changes	After Changes	Incremental Objects	Incremental Outlines	Attributes
Positive increments					Topological relations
					Topological relations

Basically, five types are considered to describe the topological relations of incremental outlines, which are defined as:

$$T_{Tp} = \{Disjoint,\ Touch,\ Intersect, Contain,\ Overlap\} \tag{4}$$

where the corresponding values of the five types of topological relations are defined as:

$$T_{Tp}(Value) = \{1,2,3,4,5\} \tag{5}$$

The description of the topological relations of the incremental outlines T_p is:

$$T_p = \left(T_{Tp}(Value), O_{id}\right) \tag{6}$$

(3) Shape, position and orientation.

Shape, position and orientation are utilized to describe incremental outlines from the perspective of geometric properties [35]. In particular, the description of shape S_c is defined as:

$$S_c = \begin{cases} (\{0,1\}, O_{id}),\ Shape\ of\ points \\ (\{1,2\},\ O_{id}),\ Shape\ of\ lines \\ (f_E), O_{id},\ Shape\ of\ areas \end{cases} \tag{7}$$

where the set of values $\{0,1\}$ indicates the existence and non-existence of points. Value set $\{1,2\}$ refers to the straight lines and curves, respectively. f_E represents the Euler number, which aims to refer to the variety of area shapes based on the research proposed by Zhou et al. [26].

Moreover, the position P_c and orientation A_c of the incremental outlines are defined as:

$$P_c = (\{0,1\}, O_{id}) \tag{8}$$

$$A_c = (\{0,1\}, O_{id}) \tag{9}$$

where value set $\{0,1\}$ refers to the unchanged and changed state of the target incremental outline.

Based on the aforementioned attributes, attribute information of the incremental outline can be summarized in Equation (10):

$$C_{IO} = \left[I_c, T_p,\ S_c, P_c, A_c\right] \tag{10}$$

where C_{IO} represents the set of descriptions belonging to incremental outline IO. I_c and T_p refer to the description of the identity and dynamics, as well as the topological relations, respectively. S_c, P_c and A_c indicate the attributes of the incremental outlines in terms of shape, position and orientation.

2.4. Incremental Updating Using Incremental Outlines

After both spatial and attribute information about incremental changes are labelled, a geometric calculation is required to distinguish the spatial features of incremental outlines automatically. Utilizing the geometric calculation result, the incremental updating process is proposed based on different operations along with the updating rules. Thus, this section consists of three main parts, including the geometric calculation, incremental updating operations and updating rules.

2.4.1. Geometric Calculation Based on Incremental Outlines

Based on the spatial classification in Figure 4, one can see that it is the endpoints of the incremental outlines that determine the updating of the object changes. Therefore, common set operations of intersection (∩) and difference () are considered for geometric calculation, the usability of which has been discussed in the current literature [37,38].

Generally, assuming that A and B are different objects, the intersection of A and B, denoted by $A \cap B$, is the object that belongs to both A and B. Meanwhile, the difference of A and B, denoted by AB, is the object that belongs to A and does not belong to B. Utilizing these two operations, a geometric calculation is proposed based on Equation (11):

$$f(L', L) = [f_N(L' \cap L), f_N(L \backslash P)] \tag{11}$$

where L and L' refer to the object outlines before changes and the incremental outlines. P indicates the endpoint of one incremental outline. $f_N(L' \cap L)$ represents the number of the generated points when proposing an intersection operation on one object outline L and one incremental outline L'. According to the spatial classification of incremental outlines, $f_N(L' \cap L)\epsilon\{0, 1, 2\}$ including L and L' disjointing without generating points or intersecting with one or two points. In addition, $f_N(L \backslash P)\epsilon\{1, 2, 3\}$ represents the object number after the object outline L is split by endpoint P.

Thus, the spatial information about different types of incremental outlines in Section 2.2.2 can be calculated as follows:

a. Extend target line: $f_N(L' \cap L) = 1$, $f_N(L \backslash P) = 1$
b. Split target line: $f_N(L' \cap L) = 2$, $f_N(L \backslash P) = 2$
c. Merge target line: $f_N(L' \cap L) = 2$, $f_N(L \backslash P) = 2$
d. Divert target line: $f_N(L' \cap L) = 2$, $f_N(L \backslash P) = 3$
e. Replace target line: $f_N(L' \cap L) = 0$, $f_N(L \backslash P) = 1$

2.4.2. Incremental Updating Operations

Once the incremental outlines are discriminated, which provides geometric information drawn from both spatial classification and attributes, basic operations need to be defined to distinguish different updating processes. However, existing operations are mainly designed for geographic entity changes' updating and cannot meet the updating requirement in terms of incremental changes [39,40]. Here, 7 effective operations are proposed based on incremental outlines, which are shown in Figure 5. It should be noted that among the proposed operations, "Delete" and "Update" correspond to deleting historical objects and replacing these objects with new ones. Despite the proposed strategy technically removing the changing parts for data updating, the original objects and incremental outlines remain in each time period of the database; thus, historical changes can be captured at any time.

Operation 1: Create. The task of this operation is to add a new object to the database. It is associated with one type of incremental outline "create target line" and one attribute "object identity and transitions" I_c.

Operation 2: Delete. The task of this operation is to permanently destroy an object in the database. Specifically, this is usually done by deleting the identified number of the corresponding object. The type of incremental outline titled "delete target line" and the attribute of "object identity and transitions" I_c are included in this operation.

Operation 3: Update. This operation aims to replace a historical object with a new object. The type of incremental outline "replace target line" with the attribute "shape" S_c is considered to utilise this operation for data updating.

Operation 4: Merge. This operation aims to form a new object by merging historical object with incremental outlines, which can be utilized in "extend target line", "merge target line", "divert target line" and "reserve identifiable point". Meanwhile, "topological relations" T_p and "shape" S_c are involved as necessary attributes.

Operation 5: Split. This operation aims to form a new object by removing an object that has disappeared. Incremental outlines "split target line" and "reserve identifiable point" can be proposed, with two required attributes "position" P_c and "orientation" A_c.

Operation 6: Move. This operation aims to modify an object with different positions disregarding shapes and orientations. Operations including "Create" and "Delete" are not required in this process. The detailed incremental outline type and attribute involve "move target line" and "position" P_c respectively.

Operation 7: Rotate. This operation aims to modify an object with different orientations without changing the shapes and positions of the target object. The detailed incremental outline type and attribute involve "rotate target line" and "orientation" A_c, respectively.

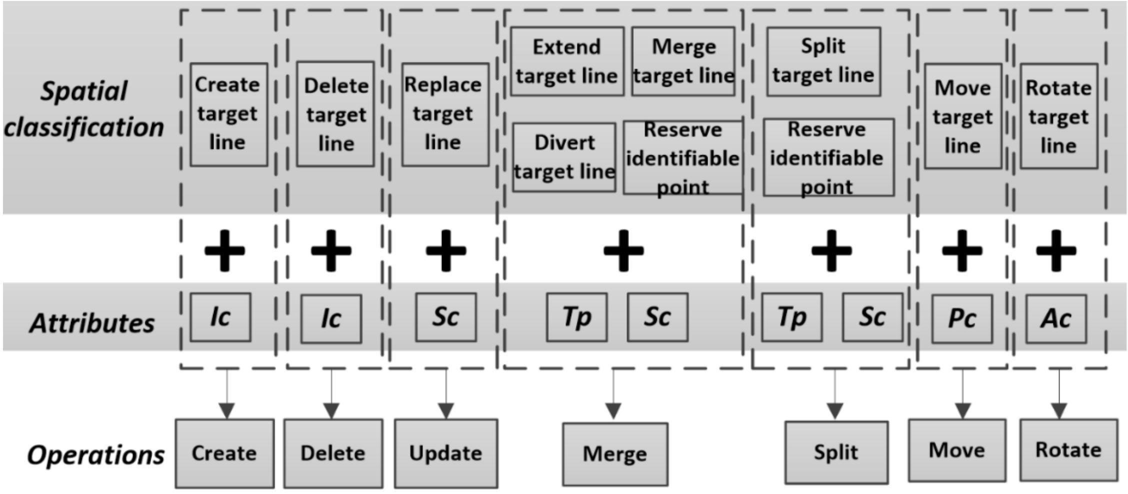


Figure 5. Updating operations based on incremental outlines.

2.4.3. Construction of Incremental Updating Rules

Based on the above operations, effective rules are needed to activate certain operations for incremental updating. We use an example in Figure 6 to illustrate the basic annotations in the incremental updating rules. Assuming that L (the lines in black) and L' (the line in red) are object outlines and incremental outlines, respectively, we define P_1 and P_2 as two endpoints of L'. P_3 refers to a reserved identifiable point, which is defined in Section 2.2.2 to identify the reserved line of the changing object. Endpoints P_1 and P_2 split object outline L into two parts. One is object outline L_R containing reserved identifiable point P_3. The other part is object outline L_E. In particular, L_R will be reserved in the changing object, while L_E will be replaced by the incremental outline L'. Meanwhile, endpoints P_1 and P_2 form a straight line L_{P1P2}. In this case, line L_{P1P2} shares the same line with L_E. However, L_E and L_{P1P2} are not always the same in many other scenarios. In addition, we utilise E to define the incremental change. Moreover, PlotGeometry refers to the incremental line plotting on the map.

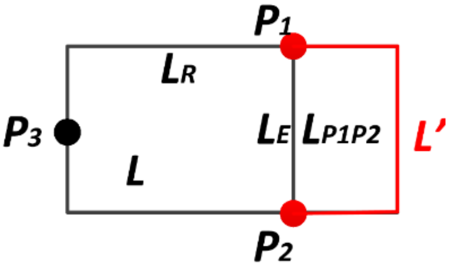


Figure 6. Annotations of incremental updating rules.

Here, 10 incremental updating rules are constructed as follows:

Rule 1: If the type of incremental outline is "create target line", the incremental operation "Create" is proposed.

If (PlotGeometry(L') = "Create L") and (I_c = 02), then (Create(E))

Rule 2: If the type of incremental outline is "delete target line", the incremental operation "Delete" is proposed.

If (PlotGeometry(L') = "Delete L'") and (I_c = 21), then (Delete(E))

Rule 3: If the type of incremental outline is "move target line", the incremental operation "Move" is proposed.

If (PlotGeometry(L') = "Move L") and ((I_c = 22) and (A_c = 0) and (P_c = 1), then (Move(E))

Rule 4: If the type of incremental outline is "rotate target line", the incremental operation "Rotate" is proposed.

If (PlotGeometry(L') = "Rotate L") and ((I_c = 22) and (A_c = 1) and (P_c = 0), then (Rotate (E))

Rule 5: If the types of incremental outline of the changed line are "split target line" and "reserve identifiable point", the incremental operation "Split" is proposed to split object outline L and delete L_E.

If (PlotGeometry(L') = "Split L'" and PlotGeometry(P) = "Reserve P") and (I_c = 22) and (T_p = 4), then (Split(L') and Delete (L_E))

Rule 6: If the type of incremental outline is "extend target line", the incremental operation "Merge" is proposed.

If (PlotGeometry(L') = "Extend L") and (I_c = 22) and (T_p = 4), then (Merge(L'))

Rule 7: If the types of incremental outline of the changed area are "split target line" and "reserve identifiable point", the incremental operations "Split", "Delete" and "Merge" are proposed.

If (PlotGeometry(L') = "Split L" and PlotGeometry(P) = "Reserve P") and (I_c = 22) and (T_p = 5), then (Split(L) and Delete (L_E) and Merge (L_R, L'))

Rule 8: If the types of incremental outline are "merge target line" and "reserve identifiable point", the incremental operations "Split", "Delete" and "Merge" are proposed.

If (PlotGeometry(L') = "Extend L" and PlotGeometry(L') = "Reserve P") and (I_c = 22) and (T_p = 4), then (Split(L) and Delete (L_E) and Merge (L_R, L'))

Rule 9: If the type of incremental outline is "divert target line", the incremental operations "Split", "Delete" and "Merge" are proposed.

If (PlotGeometry(L') = "Divert L") and ((I_c = 22) and (S_c = 1)), then (Split(L) and Delete (L_{p1p2}) and Merge (L_R, L'))

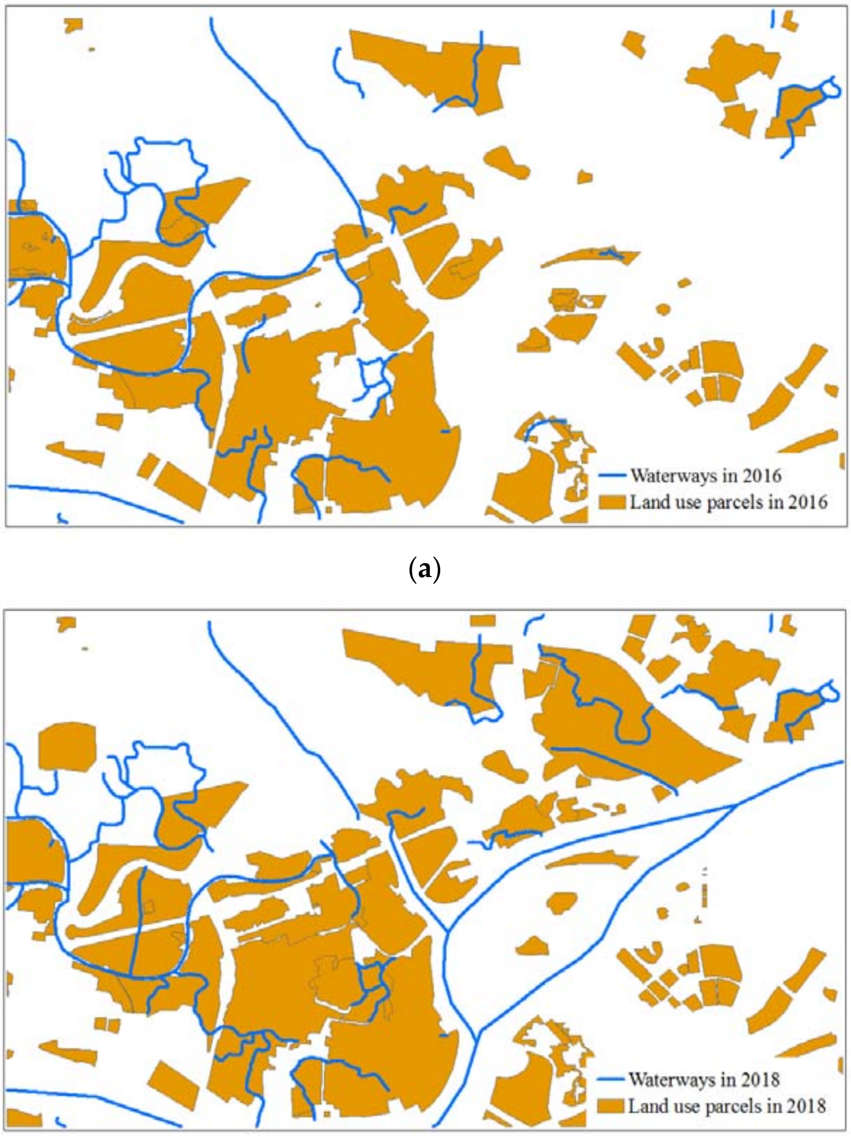
Rule 10: If the type of incremental outline is "Update target line", the incremental operation "Update" is proposed.

If (PlotGeometry(L') = "Update L") and (I_c = 22) and (S_c = 1) or (T_p = 4) or (T_p = 5)), then (Update (E)).

3. Results

3.1. Experimental Data

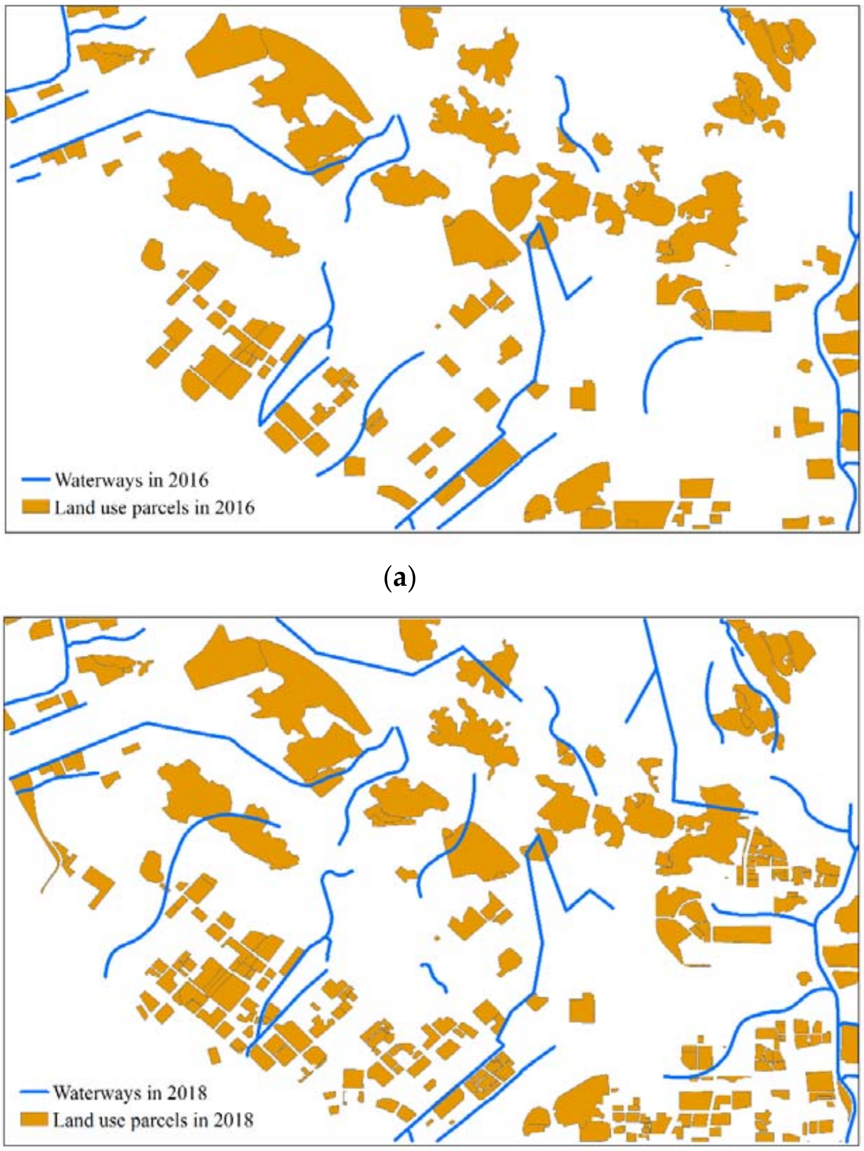
The OSM datasets in two experimental areas derived from the years 2016 and 2018 were utilized to illustrate the updating process and to evaluate our proposed approach. The first experimental area occupied 47.24 km^2, as shown in Figure 7a,b. It consisted of 45 waterways and 93 land use parcels in the year 2016. Until 2018, the numbers of waterways and land use parcels increased to 56 and 107, respectively. Figure 8a,b displays the second experimental area in the years 2016 and 2018. The number of waterways increased from 65 to 81, while the number of land use parcels rose from 134 to 275. The statistic numbers of the OSM data in two experimental areas reveal obvious changes in the period of 2016–2018. Therefore, based on these datasets, the following experiments show detailed applications in delineating different types of incremental outlines and further build effective operations and rules to assist the updating process.



(**a**)

(**b**)

Figure 7. OSM data derived from the years 2016 and 2018 in Experimental Area #1, (**a**) OSM data in 2016 in Experimental Area #1. (**b**) OSM data in 2018 in Experimental Area #1.



(**a**)

(**b**)

Figure 8. OSM data derived from the years 2016 and 2018 in Experimental Area #2. (**a**) OSM data in 2016 in Experimental Area #2. (**b**) OSM data in 2018 in Experimental Area #2.

3.2. Incremental Updating for OpenStreetMap

To verify the feasibility of the proposed methodology, an OpenStreetMap updating-based prototype system was designed through the combination of Visual Studio 2012 and ArcGIS Engine using C# programming. The flowchart of the data updating process is shown in Figure 9. For the changed object, both incremental outlines and the corresponding attributes needed to be uploaded in the system. Then, based on one incremental line chosen for object updating, the outline of the unchanged part of the object was extracted. Finally, both the incremental and extracted lines were merged to form the new object by inferring the updating rules proposed in Section 2.4.3 and the calculation of the geometric information in Section 2.4.1.

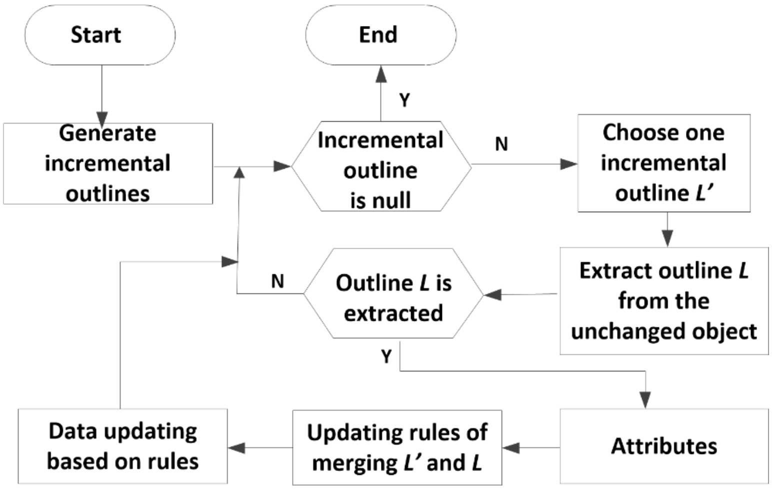
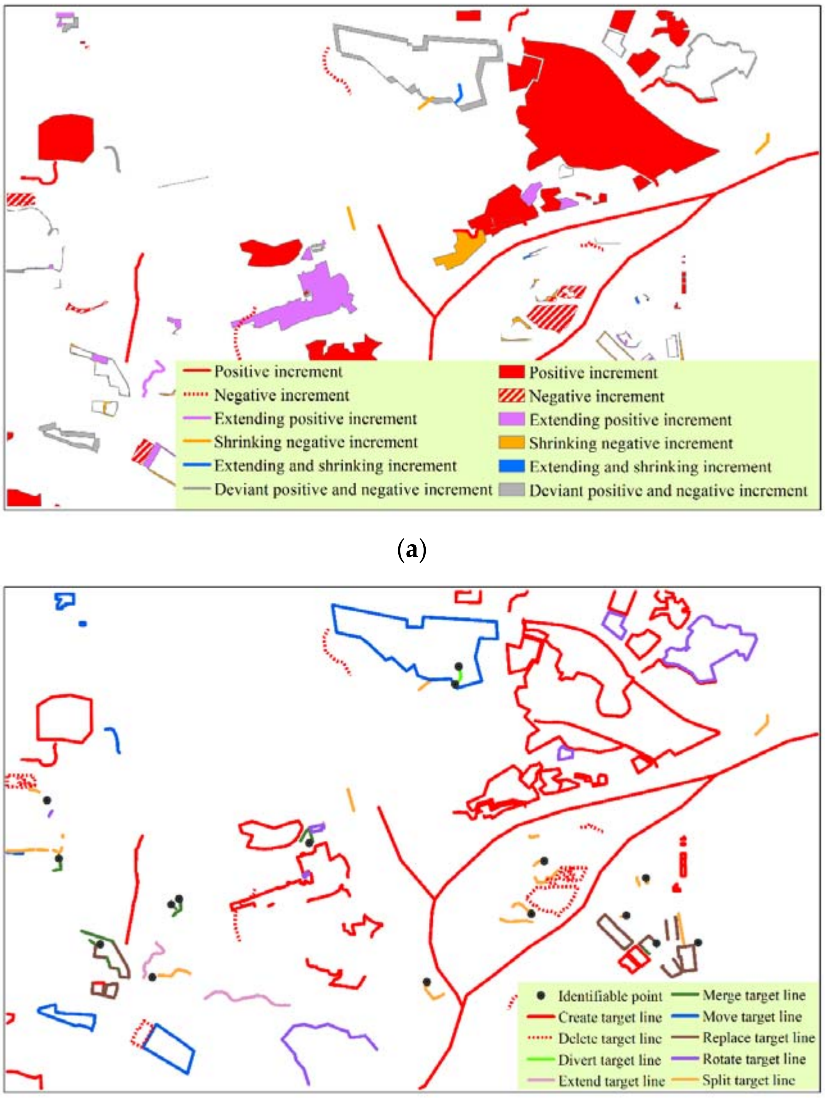

Start
End
Y
Generate incremental outlines
Incremental outline is null
N
Choose one incremental outline *L'*
N
Outline *L* is extracted
Extract outline *L* from the unchanged object
Y
Data updating based on rules
Updating rules of merging *L'* and *L*
Attributes

Figure 9. The flowchart of OSM data updating using incremental outlines.

Based on the above prototype system, the incremental updating process was proposed using OSM data in the two experimental areas. In addition, an incremental object-based approach was also utilized as comparative experiments for updating these two experimental datasets. One should note that the incremental object-based approach proposed here followed the classification strategy proposed by Chen, Lin and Liu [31], in which seven kinds of incremental lines and areas were mapped in detail. Figure 10a,b displays the classification of incremental objects and outlines, respectively, in Experimental Area #1. For the classification of incremental objects in Figure 10a, incremental areas needed to be mapped and classified. Statistically, among the 13.46 km^2 and 16.64 km^2 land use parcels in 2016 and 2018, the changing areas (both increased and decreased areas) were 4.62 km^2, occupying 34.32% and 27.76% of the total area in 2016 and 2018, respectively. Meanwhile, the classification of the incremental outlines, as presented in Figure 10b, only highlighted the changed outlines instead of mapping the whole areas. In particular, each type of incremental outline represented a different updating process during the period of 2016–2018. A similar situation was shown in Experimental Area #2. The incremental objects, as shown in Figure 11a, contained 5.53 km^2 in terms of land use parcels during this time interval, occupying 26.55% and 23.31% of the whole parcel areas in 2016 and 2018, respectively. Incremental outlines are proposed in Figure 11b to avoid mapping these areas to improve the updating efficiency.

With regard to the classification of both incremental objects and outlines, further statistics were calculated and are displayed in Table 4. For the classified incremental objects in Experimental Area #1, "positive increment", "extending positive increment" and "shrinking negative increment", taking 44, 30 and 28 of the total number of incremental changes, were the dominant types among the six types of incremental changes. Others, including "negative increment", "extending and shrinking increment" and "deviant positive and negative increment", occupied 14, 4 and 18 changes, respectively. Meanwhile, Experimental Area #2 presented different proportions of incremental objects. In particular, 130 changes were classified into "positive increment", which made up most of the changes in this area. On the other hand, another five types of incremental objects only occupied a relatively small proportion of the total changes. For the classification of the incremental outlines, it was the "create target line" that took up the most in these incremental outlines, with a total number of 45 and 128 in Experimental Areas #1 and #2, respectively. In addition, "delete target line", "replace target line" and "split target line" were classified with relatively higher frequency in Experimental Area #1, whereas "merge target line" and "split target line" occurred with more changes, namely, 25 and 29 incremental outlines, in Experimental Area #2.

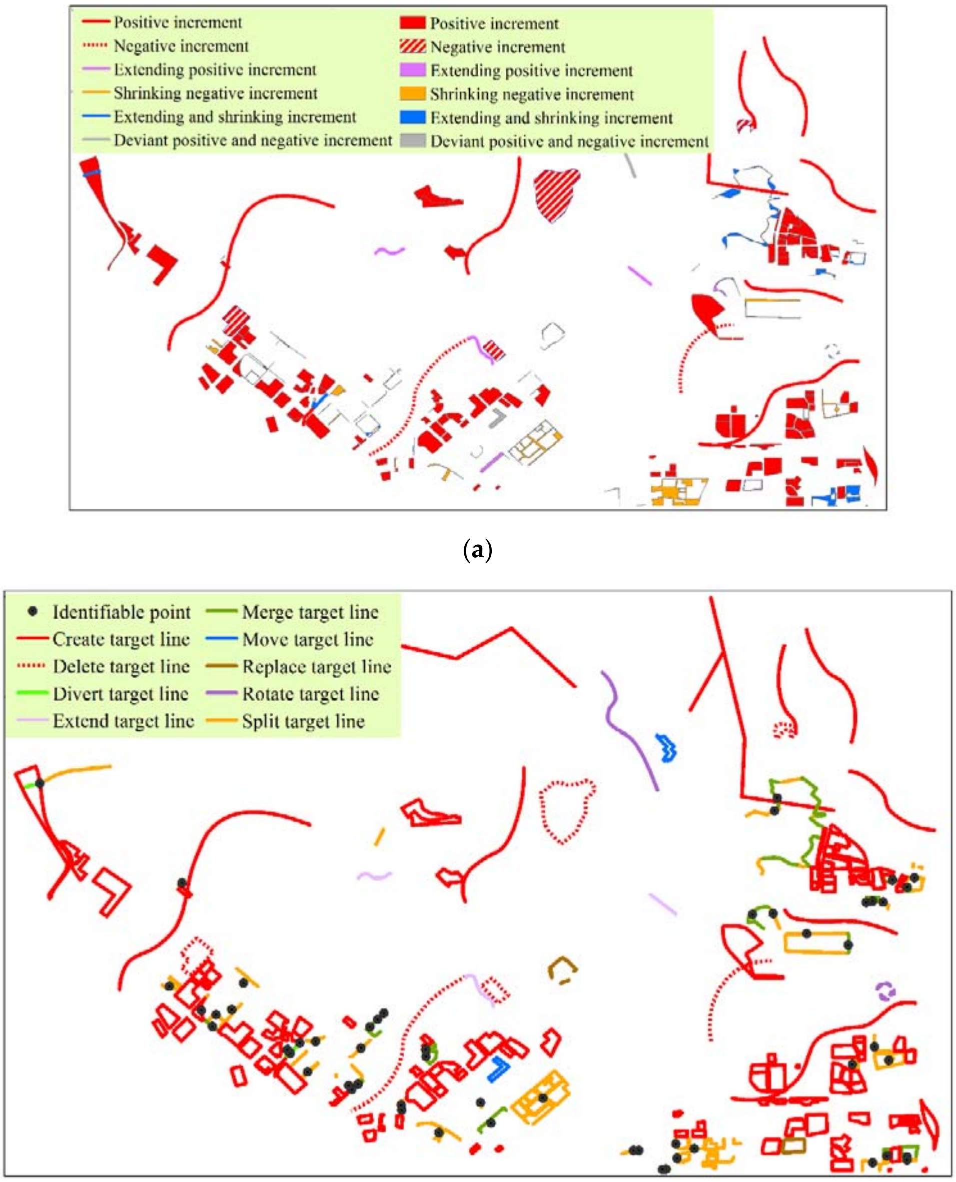
The above updating processes based on incremental objects and outlines were compared in three aspects to evaluate the efficiency of the proposed methodology, with the results summarized in Table 5. The first was to assess the number of incremental changes. In Experimental Area #1, a total number of 138 changes was mapped based on the incremental object-based approach, whereas 102 changes, 36 changes fewer, were captured using incremental outlines. A similar scenario was shown in Experimental Area #2, with 219 incremental objects and 210 incremental outlines required for updating. The second was the time cost of updating. While the incremental object-based approach took approximately 1.2 s and 2.6 s in the two experimental areas separately, only 0.4 s and 0.9 s were needed based on the incremental outlines. For the third aspect, namely the storage for updating, the incremental outline-based approach still had advantages, with 54.8 KB and 81.2 KB, approximately 43.2 KB and 45.6 KB less storage than that based on incremental objects. The above comparison indicated that the incremental outline-based approach showed advantages not only in the numbers of incremental changes, but also in time and storage costs. The results proved the updating efficiency of using an incremental outline-based approach. Moreover, with less storage being required, it further showed potential to better ensure the quality of the OSM data.


Positive increment
Negative increment
Extending positive increment
Shrinking negative increment
Extending and shrinking increment
Deviant positive and negative increment
Positive increment
Negative increment
Extending positive increment
Shrinking negative increment
Extending and shrinking increment
Deviant positive and negative increment
Identifiable point
Create target line
Delete target line
Divert target line
Extend target line
Merge target line
Move target line
Replace target line
Rotate target line
Split target line

(a)

(b)

Figure 10. Changes classification in Experimental Area #1. (**a**) Classification of incremental objects in Experimental Area #1. (**b**) Classification of incremental outlines in Experimental Area #1.


Positive increment
Negative increment
Extending positive increment
Shrinking negative increment
Extending and shrinking increment
Deviant positive and negative increment
Positive increment
Negative increment
Extending positive increment
Shrinking negative increment
Extending and shrinking increment
Deviant positive and negative increment
Identifiable point
Create target line
Delete target line
Divert target line
Extend target line
Merge target line
Move target line
Replace target line
Rotate target line
Split target line

(**a**)

(**b**)

Figure 11. Changes classification in Experimental Area #2. (**a**) Classification of incremental objects in Experimental Area #2. (**b**) Classification of incremental outlines in Experimental Area #2.

Table 4. Statistics of incremental objects and outlines classification in the two experimental areas.

Incremental Changes		Experimental Area #1	Experimental Area #2
Incremental objects classification	Positive increment	44	130
	Negative increment	14	10
	Extending positive increment	30	13
	Shrinking negative increment	28	18
	Extending and shrinking increment	4	40
	Deviant positive and negative increment	18	8

Table 4. *Cont.*

Incremental Changes		**Experimental Area #1**	**Experimental Area #2**
Incremental outlines classification	Create target line	45	128
	Delete target line	11	10
	Divert target line	1	4
	Extend target line	3	3
	Merge target line	7	25
	Move target line	7	5
	Replace target line	9	3
	Rotate target line	7	3
	Split target line	12	29

Table 5. Comparison between incremental objects and incremental outlines for OSM updating in the two experimental areas.

	Incremental Types	**Numbers of Incremental Changes**	**Time Costs for Updating**	**Storage for Updating**
Experimental Area #1	Incremental objects	138 incremental objects	1.2 s	98.0 KB
	Incremental outlines	102 incremental outlines	0.4 s	54.8 KB
Experimental Area #2	Incremental objects	219 incremental objects	2.6 s	126.8 KB
	Incremental outlines	210 incremental outlines	0.9 s	81.2 KB

4. Discussion and Conclusions

The updating of information about object changes provides the fundamentals to ensure OpenStreetMap data quality and further contributes to variable applications. Approaches proposed by existing studies involve two available modes: whole object-based updating and incremental object-based updating. While whole object-based updating can result in data redundancy, incremental object-based updating shows advantages in improving both updating accuracy and efficiency because it only requires labelling the changing part of the objects. However, the challenge of data redundancy still exists when updating massive amounts of OSM data because it requires mapping outlines that are shared with the unchanged parts of the objects. Faced with this issue, this paper proposed a novel concept called incremental outlines for OpenStreetMap data updating. In particular, only the changing outlines were considered for labelling in the updating process. We classified 10 types of incremental outlines, defined attribute information in terms of object identity and transitions, topological relations, shape, position and orientation. Then, geometric calculations were proposed based on the classified outlines with certain attributes. Finally, seven operations, including "Create", "Delete", "Update", "Merge", "Split", "Move" and "Rotate", were constructed for OSM data updating. Particularly, we considered 10 rules involving the geometric calculation results to activate the updating operations. The effectiveness of the incremental outlines was verified through the updating of OSM data with two experimental areas. The result showed that many fewer changes were needed to be mapped and updated using incremental outlines compared with the incremental object-based approach. In addition, it further revealed significant progress in terms of both time and storage costs.

The contribution of this paper is a new methodology for OSM data updating using incremental outlines. We systematically classified the incremental outlines according to the spatial distribution of the changed objects and further assigned attributes to each incremental outline. Through the above spatial and physical information, we have technically offered a guideline for users to update changes of OSM data correctly. Based on the pre-defined attributes, we proposed different operations to activate the updating process. It will be helpful for OSM developers in designing the technical infrastructure for storing and managing the changing information with higher accuracy and efficiency.

Despite the satisfactory results that were obtained, several issues remain to be discussed, which are listed in the following aspects. First, the proposed spatial classification and attributes of the incremental outlines tend to be complex for users to upload. Technically, when uploading the changing information of the OSM data, it is challenging for users to distinguish the target information among 10 types of incremental outlines and label corresponding attributes correctly. Thus, simplifying the current classification nomenclature is necessary to assist users with the data updating process. Second, as mentioned in Section 1, there may be difficulties in integrating the proposed approach in the existing OpenStreetMap system. Since this worldwide project has developed a complete system including editing tools and technical infrastructures for data updating [2], changes including the user editing rules and data management techniques remain unsettled. This issue will be further investigated in future studies. Third, the incremental outline-based approach is not limited to OpenStreetMap updating. In other words, it holds the potential to be used in the updating process of a variety of spatial data, which are related to government, industrial or private sectors. Future work will employ incremental outlines in other fields such as updating cadastral databases [41] and land cover [42] to further improve and evaluate our proposed approach.

Author Contributions: H.X. implemented the method, performed the major part of the experiments and drafted the manuscript. Y.M. developed the framework and wrote the manuscript. J.C. made substantial contributions to the conceptual design and methodological development. J.S. and K.F. implemented the experiments.

Funding: This research was supported by the National Natural Science Foundation of China (Grant No. 41501420); Engineering Research Centre of Geospatial Information and Digital Technology, NASG (Grant number SIDT20170202).

Acknowledgments: We would like to thank all reviewers for their valuable comments and constructive suggestions.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Goodchild, M.F. Citizens as voluntary sensors: Spatial data infrastructure in the world of web 2.0. *Int. J. Spat. Data Infrastruct. Res.* **2010**, *2*, 24–32.
2. Haklay, M.; Weber, P. OpenStreetMap: User-generated street maps. *IEEE Perv. Comput.* **2008**, *7*, 12–18. [CrossRef]
3. Estima, J.; Painho, M. Exploratory Analysis of OpenStreetMap for Land Use Classification. In Proceedings of the Second ACM SIGSPATIAL International Workshop on Crowdsourced and Volunteered Geographic Information, Orlando, FL, USA, 5 November 2013; pp. 39–46.
4. Liu, X.; Long, Y. Automated identification and characterization of parcels with OpenStreetMap and points of interest. *Environ. Plan. B Plan. Des.* **2016**, *43*, 341–360. [CrossRef]
5. Fonte, C.C.; Martinho, N. Assessing the applicability of OpenStreetMap data to assist the validation of land use/land cover maps. *Int. J. Geogr. Inf. Sci.* **2017**, *31*, 2382–2400. [CrossRef]
6. Ali, A.L.; Falomir, Z.; Schmid, F.; Freksa, C. Rule-guided human classification of volunteered geographic information. *ISPRS J. Photogramm. Remote Sens.* **2017**, *127*, 3–15. [CrossRef]
7. Hashemi, P.; Abbaspour, R.A. Assessment of logical consistency in OpenStreetMap based on the spatial similarity concept. In *OpenStreetMap in GIScience*; Springer: Berlin, Germany, 2015; pp. 19–36.
8. Mooney, P.; Corcoran, P.; Winstanley, A.C. Towards quality metrics for OpenStreetMap. In Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, CA, USA, 2–5 November 2010; pp. 514–517.
9. Fan, H.; Zipf, A.; Fu, Q.; Neis, P. Quality assessment for building footprints data on OpenStreetMap. *Int. J. Geogr. Inf. Sci.* **2014**, *28*, 700–719. [CrossRef]
10. Fan, H.; Yang, A.; Zipf, A. The intrinsic quality assessment of building footprints data on OpenStreetMap in baden-württemberg. In *Flächennutzungsmonitoring VIII Flächensparen-Ökosystemleistungen-Handlungsstrategien*; RHOMBOS: Berlin, Germany, 2016; pp. 253–260.
11. Girres, J.F.; Touya, G. Quality assessment of the French OpenStreetMap dataset. *Trans. GIS* **2010**, *14*, 435–459. [CrossRef]
12. Yang, A.; Fan, H.; Jing, N.; Sun, Y.; Zipf, A. Temporal analysis on contribution inequality in OpenStreetMap: A comparative study for four countries. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 5. [CrossRef]

13. Shen, J.; Yang, S.; Zang, K.; Zhou, J.; Fan, H.; Mao, B. The opinion of crowd participatory for OpenStreetMap: A survey in china. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2018**, *42*, 1525–1531. [CrossRef]
14. Hagenauer, J.; Helbich, M. Mining urban land-use patterns from volunteered geographic information by means of genetic algorithms and artificial neural networks. *Int. J. Geogr. Inf. Sci.* **2012**, *26*, 963–982. [CrossRef]
15. Zhao, Y.; Zhou, X.; Li, G.; Xing, H. A spatio-temporal VGI model considering trust-related information. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 10. [CrossRef]
16. Yang, A.; Fan, H.; Jing, N. Amateur or professional: Assessing the expertise of major contributors in OpenStreetMap based on contributing behaviors. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 21. [CrossRef]
17. Fonte, C.C.; Bastin, L.; See, L.; Foody, G.; Lupia, F. Usability of VGI for validation of land cover maps. *Int. J. Geogr. Inf. Sci.* **2015**, *29*, 1269–1291. [CrossRef]
18. Spéry, L.; Claramunt, C.; Libourel, T. A spatio-temporal model for the manipulation of lineage metadata. *Geoinformatica* **2001**, *5*, 51–70. [CrossRef]
19. Claramunt, C.; Theriault, M. Toward semantics for modelling spatio-temporal processes within GIS. *Adv. GIS Res. I* **1996**, 27–43.
20. Xing, H.; Chen, J.; Changhui, L.I. Design and implementation of network marking and reporting system of topographic feature change information. *Geomat. World* **2013**, *5*, 59–63.
21. Zhu, H.; Wu, H. Description and representation model of spatial object incremental update. *J. Spat. Sci.* **2014**, *59*, 47–61. [CrossRef]
22. Claramunt, C.; Thériault, M. Managing time in GIS an event-oriented approach. In *Recent Advances in Temporal Databases*; Springer: Berlin, Germany, 1995; pp. 23–42.
23. Hornsby, K.; Egenhofer, M.J. Identity-based change: A foundation for spatio-temporal knowledge representation. *Int. J. Geogr. Inf. Sci.* **2000**, *14*, 207–224. [CrossRef]
24. Chen, J.; Zhou, X. Incremental updating of spatial database based on topological linkage, taking cadastral database's updating as an example. *Acta Geod. Cartogr. Sin.* **2008**, *37*, 321–329.
25. Li, P.; Wang, H. Automatic recognition of change types of residential areas using topology relations model. *Geomat. Inf. Sci. Wuhan Univ.* **2009**, *3*, 12.
26. Zhou, X.-G.; Chen, J.; Zhan, F.B.; Li, Z.; Madden, M.; Zhao, R.-L.; Liu, W.-Z. A Euler number-based topological computation model for land parcel database updating. *Int. J. Geogr. Inf. Sci.* **2013**, *27*, 1983–2005. [CrossRef]
27. Xing, H.; Chen, J.; Li, C.; Zhou, X. Parametric approach to classification of spatial object change. *J. Cent. South Univ.* **2014**, *45*, 495–500.
28. Wang, Y.; Zhang, Q.; Guan, H. Incrementally detecting change types of spatial area object: A hierarchical matching method considering change process. *ISPRS Int. J. Geo-Inf.* **2018**, *7*, 42. [CrossRef]
29. Fan, Y.; Yang, J.; Zhang, C.; Zhu, D. A event-based change detection method of cadastral database incremental updating. *Math. Comput. Model.* **2010**, *51*, 1343–1350. [CrossRef]
30. Harrie, L. *Generalisation Methods for Propagating Updates between Cartographic Data Sets*; Department of Physical Geography and Ecosystem Science: Lund, Sweden, 1998.
31. Chen, J.; Lin, Y.; Liu, W. Formal classification of spatial incremental changes for updating. *Acta Geod. Cartogr. Sin.* **2012**, *41*, 108–114.
32. Badard, T.; Richard, D. Using xml for the exchange of updating information between geographical information systems. *Comput. Environ. Urban Syst.* **2001**, *25*, 17–31. [CrossRef]
33. Zhang, X.; Guo, T.; Huang, J.; Xin, Q. Propagating updates of residential areas in multi-representation databases using constrained Delaunay triangulations. *ISPRS Int. J. Geo-Inf.* **2016**, *5*, 80. [CrossRef]
34. Egenhofer, M.J.; Al-Taha, K.K. Reasoning about gradual changes of topological relationships. In *Theories and Methods of Spatio-Temporal Reasoning in Geographic Space*; Springer: Berlin, Germany, 1992; pp. 196–219.
35. Zhao, Z.; Stough, R.R.; Song, D. Measuring congruence of spatial objects. *Int. J. Geogr. Inf. Sci.* **2011**, *25*, 113–130. [CrossRef]
36. Khoshafian, S.N.; Copeland, G.P. *Object Identity*; ACM: New York, NY, USA, 1986; Volume 21.
37. Li, Z.; Zhao, R.; Chen, J. A Voronoi-based spatial algebra for spatial relations. *Prog. Nat. Sci.-Mater. Int.* **2002**, *12*, 528–536.
38. Xing, H. Models of spatial marking and reporting for geographic updating. *Acta Geod. Cartogr. Sin.* **2014**, *43*, 880.
39. Beyen, J.; Henrion, J. Updating topographic databases with arc info; client-fitted creation of change-only information. In *International Archives of Photogrammetry and Remote Sensing*; Citeseer: Stuttgart, Germany, 1998.

40. Badard, T. On the automatic retrieval of updates in geographic databases based on geographic data matching tools. *Bulletin Comité Français Cartographie* **1999**, *162*, 34–40.
41. Zhou, X.; Chen, J.; Madden, M. Classification and identification of cadastral structure change. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2008**, *37*, 1437–1446.
42. Chen, J.; Chen, J.; Liao, A.; Cao, X.; Chen, L.; Chen, X.; He, C.; Han, G.; Peng, S.; Lu, M. Global land cover mapping at 30 m resolution: A POK-based operational approach. *ISPRS J. Photogramm. Remote Sens.* **2015**, *103*, 7–27. [CrossRef]