Utilizing Geometric Coherence in the Computation of Map Transformations

Padraig Corcoran[a,*], Peter Mooney[b], Michela Bertolotto[a]

[a] *School of Computer Science and Informatics, University College Dublin. Ireland*
[b] *Department of Computer Science, National University of Ireland Maynooth. Ireland*

Abstract

Adaptive mapping and real-time spatial data delivery are currently major research topics in the fields of Web-GIS and Location Based Services (LBS). In order to successfully implement these paradigms a methodology to progressively adopt or transform an arbitrary map representation in an arbitrary fashion on-the-fly is necessary. Such a methodology was previously considered not feasible due to the high computational complexity associated with existing automated map generalisation methodologies. This paper presents a methodology, inspired by related research in the field of computer graphics, which offers the potential to reduce such associated computational complexity. This is achieved using an approach to map transformation which takes advantage of the geometric coherence between a user's spatial data requirements.

Keywords: Adaptive, Geometric Coherence, Map Transformation

1. Introduction

The Internet has become an important medium for the dissemination of spatial or map data and in the future most developments in the field of Geographical Information Science (GIS) will centre on the Internet (Peng and Zhang, 2004). This new medium, commonly referred to as Web-GIS, has introduced mapping and GIS to a large audience with little or no knowledge

*Corresponding Author
Email address: padraig.corcoran@ucd.ie (Padraig Corcoran)

of cartographic principles. Therefore map representations which require a reduced cognitive load to interpret are necessary. It has been recognized that in order to communicate spatial information effectively and reduce cognitive load a map representation must be adapted or personalized to an individual's requirements (Foerster, 2010). Wilson et al. (2010) performed studies which showed that adapting or personalizing map content results in greater user efficiency when performing many map based tasks. Kopf et al. (2010) showed that adapted navigation maps communicate direction information more effectively.

As an example consider the case where a motorist is navigating along the road coloured purple in Fig. 1. This map contains much information which is irrelevant to the motorist. Such information includes the display of other roads in high levels of detail and the inclusion of features which are not located spatially close to the purple road. In order to adapt the map in question a suitable map transformation must be applied. We define a map transformation as a function which transforms a map representation in the function domain into a different map representation in the function range. A map transformation in turn corresponds to the application of a sequence of map transformation operators. By applying the following sequence of transformation operators the map in Fig.1 is transformed to that of Fig. 2(a) which, it could be argued, meets the motorist's requirements. Firstly the detail used to represent the pink road is reduced while the detail used to represent the purple road remains high. Secondly, only those features which are spatially close to the purple road are represented in the resulting map; all other irrelevant features are removed. Finally, due to the importance of topological properties in navigation, the map in Fig. 2(a) is topologically accurate. Unfortunately most existing Web-GIS and LBS do not offer map representations which are adapted to an individual's needs. Instead such services pre-compute map representations and therefore can only provide non-adaptive generic solutions. This is the approach employed by map providers such as Google Maps and OpenStreetMap.

It must also be noted that generally an individual's spatial information requirements will change with high frequency. Consider again the example of the motorist but in this case the event where the motorist alters their journey from travelling on the purple road to travelling on the pink road. The causal effect of this event is that the map in Fig. 2(a) is no longer adapted to the motorists requirements. Instead a map such as that displayed in Fig. 2(b), it could be argued, is necessary to meet these requirements where the the pink

road is represented in high detail while the purple road is represented in low detail and only those features spatially close to the pink road are shown.

Unfortunately current methodologies for automated map generation cannot be considered real-time and therefore would not tolerated by the average user who demands real-time responses. This also makes such methodologies inappropriate for time-critical applications such as emergency response services or real-time traffic management which require instant access to up-to-date spatial data (Zhang and Li, 2005). Jones et al. (2000) noted that, due to computational complexity, real-time map generation is not regarded as a practical solution in the near future. This can be attributed to the fact that such approaches have focused primarily on achieving high cartographic quality irrespective of computational complexity and associated running time (Weibel, 1997; Weibel and Burghardt, 2007). Consequently the automated map generation process could last several hours (Lehto and Sarjakoski, 2005). The exact time constraint a map generation process must obey in order to be classified as real-time varies between authors. Weibel and Burghardt (2007) specify a constraint of a few seconds, Lehto and Sarjakoski (2005) a constraint of five seconds and Cecconi and Galanda (2002) a constraint of ten seconds. Yang et al. (2005) discussed the use of several methods for enhancing the performance of Web-GIS. However all these methods are based on the assumption that map representations are not adapted to a user's requirements. Foerster (2010) proposed an adaptive map generation methodology which produces map representations of high quality but the author's approach is not real-time even for small datasets. The author identified reducing computational complexity as a major challenge facing the development of such a system. The aim of the work presented in this paper is to make a scientific contribution towards tackling this issue.

In order to transform a given map representation to one which has been adapted to an individual's requirements two computational components must be implemented. Firstly a set of transformation operators, which when applied correctly are capable of transforming a given map to the required representation, must be defined. Secondly a method is necessary for determining a suitable application of these operators; that is determining the subset of operators to be applied, the order in which they should be applied and how they should be parametrised (Shea and McMaster, 1989). In this paper we focus exclusively on how the first of these components can be implemented in a computationally efficient manner and assume that an implementation of the second component is available. We propose that two possible strategies

may be used to implement this component; we now discuss each of these in turn.

The first possible approach is to define transformation operators which are based on the assumption that each transformation is computed in isolation where the transformation domain and range are equal to the original map representation and the required representation respectively. In the context of computing the map representations in Fig. 2(a) and Fig. 2(b) this corresponds to setting the transformation ranges equal to the maps in Fig. 2(a) and Fig. 2(b) respectively and both transformation domains equal to the map in Fig. 1. Given that each map is computed in isolation it must also be transmitted in isolation. All existing approaches to automated map generation use this first strategy of computing each map in isolation (Lehto and Sarjakoski, 2005).

The second possible approach is to define transformation operators which are based on the assumption that the spatial datasets requested by a user within a small window of time will generally exhibit geometric coherence. That is to say, a user will generally request a map representation containing much of the same geometric data as a previous representation received but with the addition or removal of specific information. In order to exploit such geometric coherence to achieve reduced computation time and data transmission, the previously computed representation may be used as a starting point for computation of the required representation. In other words, the domain of the required transformation may be assigned to the previously computed representation and the range assigned to the required representation. To illustrate this consider the situation where the map in Fig. 2(a) has previously been computed but no longer fulfils the user's requirements. Instead a map similar to that displayed in Fig. 2(b) is required. The two maps in question exhibit geometric coherence; Fig. 2(a) may be transformed into Fig. 2(b) with the addition and subtraction of some details. On the other hand the geometric coherence between the maps in Fig. 1 and Fig. 2(b) is weak. Therefore in this situation the optimal strategy, in terms of both computation and transmission costs, is to compute a transformation where the map in Fig. 2(a) is the domain and the map in Fig. 2(b) is the range. Assuming that most pairs of consecutive map requests will exhibit geometric coherence, this strategy reduces the computational complexity of the corresponding transformation. Since only updates to transform the previous representation into the required representation need be transmitted the overall transmission costs are subsequently reduced. This concept of only

transmitting updates has previously been proposed in the context of progressive transmission (Bertolotto and Egenhofer, 2001; Yang et al., 2007).

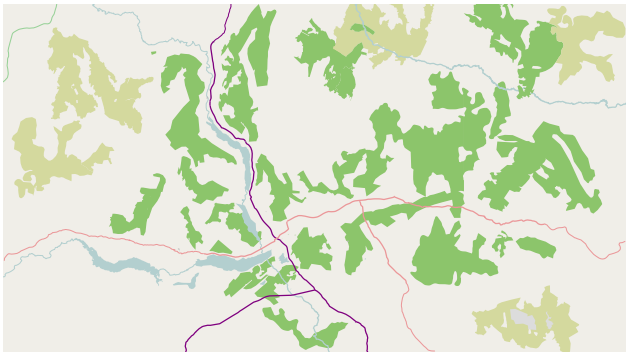
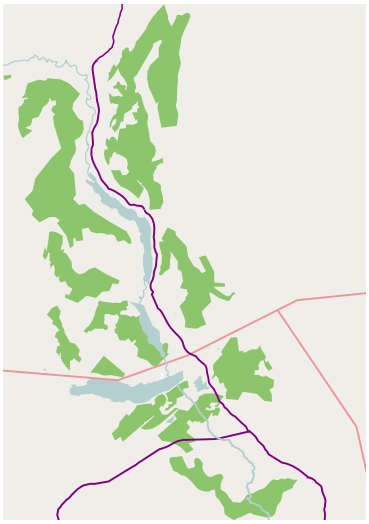
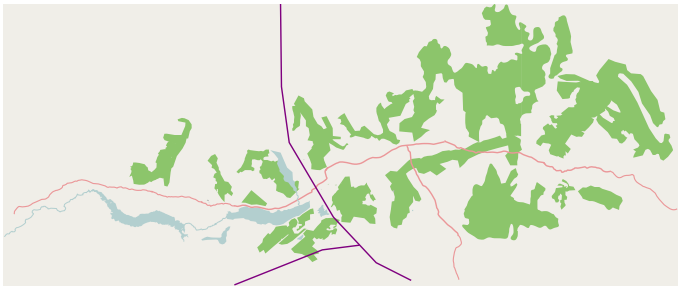


Figure 1: The original map from which the adapted maps in Fig 2 are derived is displayed.



(a)



(b)

Figure 2: The maps in (a) and (b) are adapted for a motorist travelling on the purple and pink coloured roads receptively.

It is evident from the above discussion that exploiting the geometric coherence between user requirement's offers the potential to reduce the computational complexity of applying map transformation operators. This concept of exploiting the geometric coherence has previously been proposed by many authors in the field of computer graphics (Badt, 1988; Luebke and Erikson, 1997; Oh et al., 2001). Despite this fact geometric coherence has yet to be considered in the field of automated map generation. The development of a set of transformation operators, which exploit this concept, is the focus of this paper.

The remainder of this paper is outlined as follows. In section 2 we review

existing approaches to automated map generation which attempt to provide real-time performance. Such methods do not exploit the geometric coherence between user requirements. In Section 3 we describe a set of transformation operators which can transform a map representation in an arbitrary manner and in turn allow the exploitation of geometric coherence. Section 4 describes a methodology for applying these operators such that certain constraints are satisfied. Section 5 presents results which demonstrate the reduction in computational time achieved using the proposed operators. Finally in Section 6 we draw conclusions and discuss possible future research on this topic.

2. Existing Map Generation Methods

To overcome the issue of high computational complexity associated with automated map generation a number of solutions have been proposed. All such solutions are based on generalisation transformations which are defined as the process of creating a selected and simplified map representation of detail appropriate to the scale and/or the purpose of the map (Weibel, 1997). An important attribute of any generalisation operator is that it does not add information to the transformation range not contained in the transformation domain. To remove the requirement to compute a required map representation upon request Han and Bertolotto (2004) and Weibel and Burghardt (2007) propose to represent the original data set using a hierarchical data structure which contains a multi-scale representation created through generalisation. When a user requires a dataset at a particular scale this is then extracted from the data structure. Examples of such an approach include the Binary Line Generalisation (BLG) Tree (van Oosterom and van den Bos, 1989) and the tGAP structure (Haunert et al., 2009). The main disadvantage of using a hierarchical data structure is that the map representations are pre-computed and therefore cannot be completely adapted to a user's requirements. Hierarchical data structures generally only provide map representations where all objects are represented at an equal scale (Bertolotto et al., 1995). There are many situations where a user may require a representation containing objects represented at different scales. Consider again the example above where a motorist is travelling on the purple road in Fig. 2(a). In this situation, it could be argued, that the purple road should be represented in high detail and other features, for example the pink road, should be represented in low detail. A second strategy, which has been proposed to

overcome the issue of computational complexity, is to generate map representations on-the-fly without ensuring topological accuracy. Implementations of this approach include the works of Sarjakoski and Sarjakoski (2004), Lehto and Sarjakoski (2005) and Mustafa et al. (2006). Many geographical decisions are a function of map topology; Kopf et al. (2010) stresses this point in the context of navigation maps. Therefore it is important that all map representations are topologically accurate. Cecconi and Galanda (2002) propose a final strategy to overcome the issue of computational complexity. This approach integrates both a hierarchical data structure and on-the-fly generalisation. A hierarchical data structure is used to generate the required representation for those objects with a corresponding high computational cost for generalisation. While on-the-fly generalisation is used to create the required representation for those objects with a corresponding low computational cost for generalisation. Again, the major limitation of this approach is that it does not ensure topologically accurate results.

3. Transformation Operators

As discussed in section 1 in order to transform a given map representation a set to transformation operators must be defined. In this section we define a set of transformation operators which allow the geometric coherence between user requirements be exploited. All previous approaches to map transformation have been based on the application of exclusively generalisation operators or exclusively enhancement operators. An enhancement operator is an operator which allows the addition of map detail. Unlike a generalisation operator it adds information to the transformation range which is not contained in the transformation domain. That is, information required to compute the transformation is also drawn from a data source other than the transformation domain. For example consider a line which has been previously transformed by a generalisation operator which in turn reduces the number of vertices used to represent the line. An operator which subsequently re-inserts these vertices at a later stage would be classified as an enhancement operator because the vertices in question are not contained in the transformation domain. In situations where the required map transformation involves exclusively the reduction of map detail generalisation operators are suitable. On the other hand, in situations where the required map transformation involves exclusively the addition of map detail enhancement operators are suitable. However an arbitrary map transformation may

involve both the simultaneous reduction and addition of detail. For example consider the transformation required to transform the map in Fig. 2(a) to that in Fig. 2(b). Such a transformation would remove objects which are spatially close to the purple road and add objects whivh are spatially close to the pink road. This transformation would also remove detail from the purple road and add detail to the pink road. In such situations a set of generalisation and enhancement operators which can be applied simultaneously is necessary. In the remainder of this section we define two generalisation and two enhancement operators which meet this requirement.

Jones (1997) identified eight categories of generalisation operators. These are selection, simplification, typification, exaggeration, enhancement, collapse, aggregation and displacement. In this article we are primarily concerned with selection and simplification operators. Before defining these operators it is convenient to define a number of variables. Let M be the set of objects contained in the original map before any transformation has been applied; in the context of the example presented in Section 1, M would correspond to the set of objects contained in the map in Fig. 1. Let D and R be the sets of objects contained in the transformation domain and range respectively. Let T be the set of objects to which we wish to apply a generalisation or enhancement operator. Depending on the operator T may be a subset of D or R. The function $V(a)$ returns the set of vertices representing the object a. Finally the function $|A|$ returns the cardinality of the set A.

Informally, selection transforms a map by reducing the number of objects which are represented. For example the map in Fig. 3(a) is generalized using a selection operator which reduces the number of objects represents by a single polygon to return the result in Fig. 3(b). Selection is defined formally in the following definition.

Definition 1. *Let T be the set of objects to be removed from D through generalisation by selection where $T \subseteq D$. The range of this transformation is equal to the set difference between D and T. That is $R = D \setminus T$.*

Informally, simplification transforms a map by reducing the number of vertices used to represent objects. That is, if an object a is simplified to b, the set of vertices representing b will be a subset of those representing a. The map in Fig. 3(b) is subsequently generalized by applying simplification to both objects to return the result in Fig. 3(c). Simplification is defined formally in the following definition.

Definition 2. *Let T be the set of objects to be simplified where $T \subseteq D$. The range of this transformation R is such that* $\forall s(s \in T) \rightarrow \exists r((r \in R) \wedge (V(r) \subseteq V(s)))$.

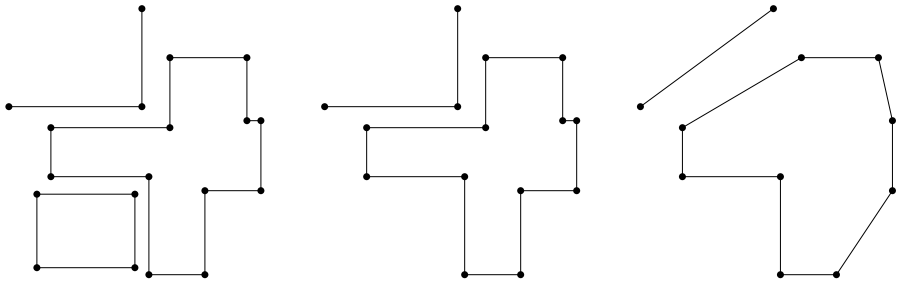


Figure 3: The map in (a) is generalized by selection to produce (b). The map in (b) is generalized by simplification to produce (c).

Both selection and simplification operators provide a more abstract representation and therefore reduce user cognitive load (Corcoran et al., 2012). For each generalisation operator there exists a corresponding enhancement operator. These operators add detail in a manner which is opposite to the generalisation operator in question. We define the corresponding enhancement operator to selection as addition. Informally, addition adds objects which were previously removed through selection. In Fig. 4(a) the map in question is enhanced using an addition operator which adds a single polygon to return the result in Fig. 4(b). Addition is defined formally using the following definition.

Definition 3. *Let T be the set of objects to be added to D by addition where $T \subseteq M$ and $T \cap D = \emptyset$. The range of this transformation is equal to the set union of D and T. That is $R = D \cup T$.*

We define the corresponding enhancement operator to simplification as refinement. Informally, refinement adds vertices to objects which were previously removed through simplification. That is, if an object a is refined to b, the set of vertices representing b will be a superset of those representing a. The map in Fig. 4(b) is subsequently enhanced using a refinement operator which is applied to a single polygon to return the result in Fig. 4(c). Refinement is defined formally using the following definition.

Definition 4. *Let T be the set of objects to be refined in D where $T \subseteq D$. The range of this transformation R is such that* $\forall s(s \in T) \rightarrow \exists r((r \in R) \wedge (V(r) \supseteq V(s)))$.

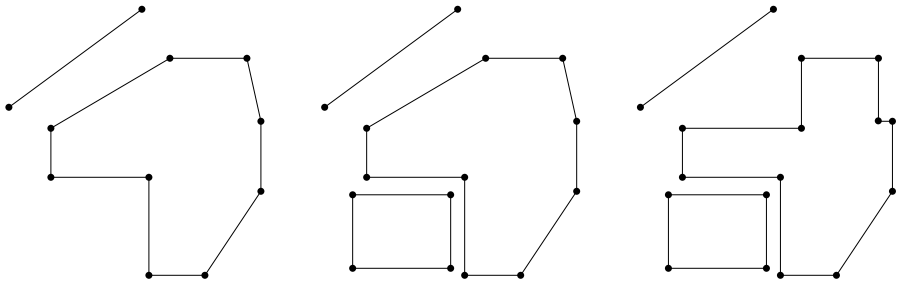


Figure 4: The map in (a) is enhanced by addition to produce (b). The map in (b) is enhanced by refinement to produce (c).

A property that follows from definition 4 is that the set of vertices which represent a refined object will be a subset of the set of vertices which represent the corresponding object in the original map M. That is, $\forall s(s \in T) \rightarrow \exists m((m \in M) \wedge (V(r) \subseteq V(m)))$.

As will be illustrated in the forthcoming section when performing a map transformation it is important to maintain topological relations between all objects and not just those in D and R. In order to facilitate this we maintain an additional set of objects S where all transformations, except selection and addition, applied to D are also applied to S. That is to say the set S will contain a representation of all objects. For example, in the context of the two step generalisation illustrated in Fig. 3, following generalisation the set of objects S is equivalent to those in Fig. 3(b) without the application of the selection operation.

4. Transformation Constraints

The intention of any generalisation process is to produce the best result possible subject to a set of constraints (Jones and Ware, 2005). Such constraints may also be considered in the context of map enhancement. Weibel (1996) identified four classes of constraints which generalisation, and in turn

enhancement, may aim to satisfy. These are shape (Gestalt), semantic, metric and topological constraints. A Gestalt constraint maintains important object shape characteristics. Semantic and metric constraints perform generalisation or enhancement in a manner which is a function of object semantics and an error function respectively. Finally a topological constraint ensures that all resulting maps are topologically equivalent to the original detailed map (Corcoran et al., 2011). Two maps are topological equivalent if all corresponding topological relations, such as disjoint and meet, are equal. If a generalisation or enhancement is topologically equivalent to its corresponding original detailed map we say it is topologically consistent.

In this paper we propose a map transformation methodology which is based on the four generalisation and enhancement operators presented in the previous section. In each of the following subsections we describe an implementation of each operator which satisfies shape and topological constraints.

4.1. Simplification

Removing a single vertex from $V(T)$ will simplify the set of objects T by a single step. Depending on how many times this step is repeated varying degrees of simplification may be obtained. In this section we describe a methodology for simplifying T by a single step such that shape and topological constraints are satisfied. To simplify T by a single step such that shape constraints are satisfied the element of $V(T)$ which contributes least significantly to the shape characteristic of any object should be removed (Latecki and Lakmper, 1999). The significance of a vertex a is determined using the metric of Latecki and Lakmper (1999) defined in Equation 1 where $l(a)$ and $r(a)$ return the respective lengths of the two line segments adjacent to a and $\beta(a)$ returns the the turning angle between these segments.

$$K(a) = \frac{\beta(a)\, l(a)\, r(a)}{l(a) + r(a)} \tag{1}$$

Removing the least significant vertex in terms of the function K will ensure a shape constraint is satisfied but will not ensure a topological constraint is also satisfied. For example consider the scene in Fig. 5(a) where $D = T = \{p, r\}$, $V(T) = \{p_1, p_2, p_3, p_4, p_5, r_1, r_2, r_3, r_4\}$, $V(p) = \{p_1, p_2, p_3, p_4, p_5\}$ and $V(r) = \{r_1, r_2, r_3, r_4\}$. If the least significant vertex in $V(T)$, that is the vertex r_2, were removed the simplification in question would be topologically inconsistent. That is, the topological relationship between p and r is altered through the introduction of an intersection. If a single vertex is removed

from an object then a triangular shaped bounded face exists between the original and simplified object. Consider again the scene in Fig. 5(a). If the vertex p_3 is removed from p the bounded face $\{p_2, p_3, p_4\}$ is formed; this is represented by the grey region in Fig. 5(b). It has been proven that if an object vertex lies inside such a bounded face the simplification in question is topologically inconsistency otherwise it is topologically consistent (Corcoran et al., 2011). From Fig. 5(b) we can see that no vertex lies in the bounded face in question and therefore we can infer that the removal of p_3 results in a topologically consistent simplification. On the other hand, if the vertex r_2 was removed from r, the vertex p_3 would lie in the corresponding bounded face and we could infer that the corresponding simplification is topologically inconsistent.

When performing simplification it is necessary to maintain topological consistency with respect to all objects and not just those which are elements of D, R and T. Consider again the scene in Fig. 5(a) where $D = T = \{p\}$; unlike the previous example r is no longer a member of these sets. The topological relationship between p and r must still be maintained so that the introduction of r at a later stage through an addition operator can be achieved without the introduction of a topological inconsistency. This is achieved by maintaining topological relations between all objects in S, where S contains a representation of all objects in the original map M (see end of Section 3).

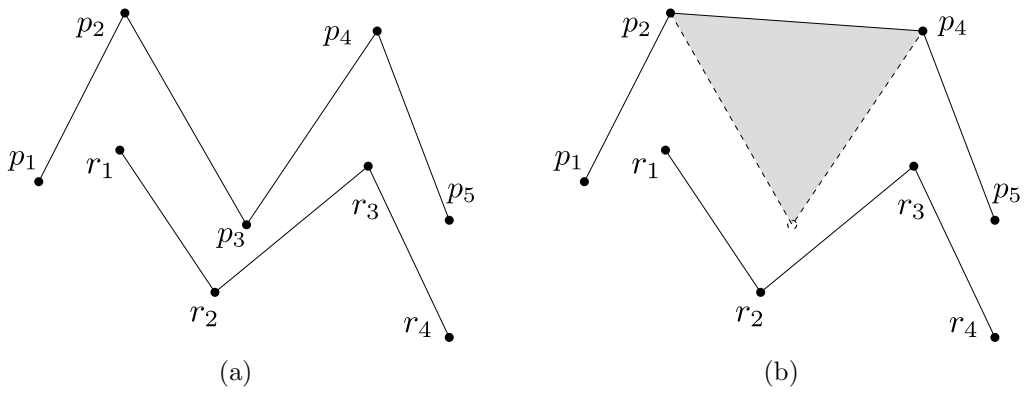


Figure 5: Simplifying the line object p in (a) creates a triangle shaped bounded face which is represented by the grey region in (b).

The previous discussion demonstrates that topological consistency can

be maintained if at each simplification step no element of $V(S)$ lies in the corresponding bounded face. The naive approach would be to determine if each element of $V(S)$ lies inside the bounded face. It has been proven that the topological relationship between two objects cannot change through simplification if their corresponding convex hulls do not intersect (Saalfeld, 1999). For example consider the scene in Fig. 6 where $S = \{p, q, r\}$. It is evident that the convex hulls of p and r intersect while the convex hulls of p and q do not. Therefore if an element of $V(p)$ is removed, one only needs to evaluate if any element of $V(p)$ or $V(r)$ lies in the corresponding bounded face in order to determine if the result is topologically consistent. Elements of $V(q)$ do not need to be considered. For $s \in S$ we define $C(s)$ as the subset of S such that the convex hull of each element in this subset intersects the convex hull of s. Due to the fact that objects contained in a map are generally well distributed over space, the magnitude of $|C(s)|$ and $|V(C(s))|$ will be relatively small compared to $|S|$ and $|V(S)|$ respectively. For each $s \in S$ the set $C(s)$ is computed a single time using the original detailed map before any transformation has been computed.

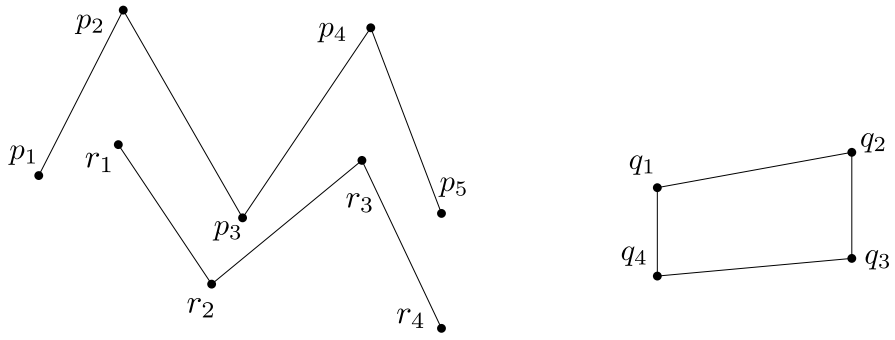


Figure 6: A scene containing two line objects and a single polygon object.

In order to simultaneously satisfy shape and topological constraints at each simplification step we propose to remove the least significant vertex of T, in terms of the function K, such that the resulting map is topologically consistent. We now describe how this strategy is implemented. For each $s \in S$ we maintain a variable denoted $OS(s)$ which indicates the least significant element of $V(s)$, in terms of the function K, such that its removal returns a topologically consistent result. For example if $S = \{p, q, r\}$ as represented in Fig. 6, the variables in question would be $OS(p) = p_3$, $OS(q) = q_1$ and $OS(r) = r_3$. Each time an element is removed from $s \in S$ through

simplification, the variable $OS(.)$ corresponding to each element of $C(s)$ must be updated. For example if the element p_3 is removed from p in Fig. 6, r_2 subsequently becomes a new candidate for the $OS(r)$; therefore the value of $OS(r)$ must be updated. For an arbitrary element of S the time complexity required to compute $OS(s)$ is presented in the following theorem.

Theorem 1. *For $s \in S$ the least significant vertex of $V(s)$, in terms of the function K, such that its removal returns a topologically consistent result, can be determined in $O(\,|V(s)|\ log\ |V(s)| + |V(s)|\,|V(S)|\,)$ time.*

Proof. The elements of $V(s)$ are initially sorted by K value; this requires $O(|V(s)|\ log\ |V(s)|)$ time. These elements are then evaluated in ascending order until an element is found such that its removal returns a topologically consistent result. At most $|V(s)|$ vertices will be evaluated. Each evaluation requires at most $|V(S)|$ time to determine if any vertex lies inside the bounded face in question. Therefore the overall time complexity is $O(\,|V(s)|\ log\ |V(s)| + |V(s)|\,|V(S)|\,)$. □

4.2. Selection

Selection is performed by subtracting the set T from the set D; that is $R = D \backslash T$. This operation preserves the topological relationships between all objects contained in R and therefore the topological constraint is satisfied. Also selection does not alter the shapes of the objects contained in R and therefore the shape constraint is satisfied. The set subtraction $D \setminus T$ can be performed in $O(|V(D)| + |V(T)|)$ time (Cormen et al., 2009).

4.3. Refinement

Adding a single vertex to $V(T)$ will refine the set of objects T by a single step. In this section we describe a methodology for refining T by a single step such that shape and topological constraints are satisfied. In order to perform refinement in a manner which satisfies shape constraint the most significant element of $V(T)$ must be added first. As in the case of simplification, the function K in Equation 1 is used to determine the significance of such elements. Adding the most significant element of $V(T)$ will ensure a shape constraint is satisfied but will not ensure a topological constraint is also satisfied. Consider the scene $D = \{p, r\}$ represented in Fig 7(a) and which corresponds to a simplification of Fig. 5(a). If one wishes to perform a refinement of D with $T = \{p, r\}$ simply adding the most significant

element of $V(T)$, that is p_3, returns a result which satisfies shape constraints but not topological constraints. This is illustrated in Fig. 7(b) where it is evident that an intersection has been introduced between p and r.

As discussed in Section 4.1 when one performs simplification a bounded face is formed between the original and resulting objects. Such a bounded face is also created when refinement is performed. The bounded face corresponding to the refinement in Fig. 7(b) is represented by a grey region. It was demonstrated in Section 4.1 that when performing simplification topological consistency can be determined by evaluating if an object vertex lies in the corresponding bounded face. It was proved by Corcoran and Mooney (2011) that this procedure cannot be used to determine topological consistency when performing refinement. For example consider again the refinement of Fig. 7(b). Although the refinement in question changes the topological relationship between p and r no object vertex lies in the corresponding face. It was demonstrated by Corcoran and Mooney (2011) that in order to determine the topological consistency of a refinement one must explicitly evaluate if an intersection of objects was introduced. In the case of the refinement in Fig. 7(b) this corresponds to evaluating if the added line segments $\overline{p_2p_3}$ and $\overline{p_3p_4}$ intersect objects in the scene. The topological relationship between two objects cannot change through refinement if the corresponding convex hulls of the original detailed objects do not intersect. That is, the topological relationship between p and r cannot change if $r \notin C(p)$ or $p \notin C(r)$. Akin to the simplification procedure described above, this property is used to reduce the number of segment intersections which must be evaluated.

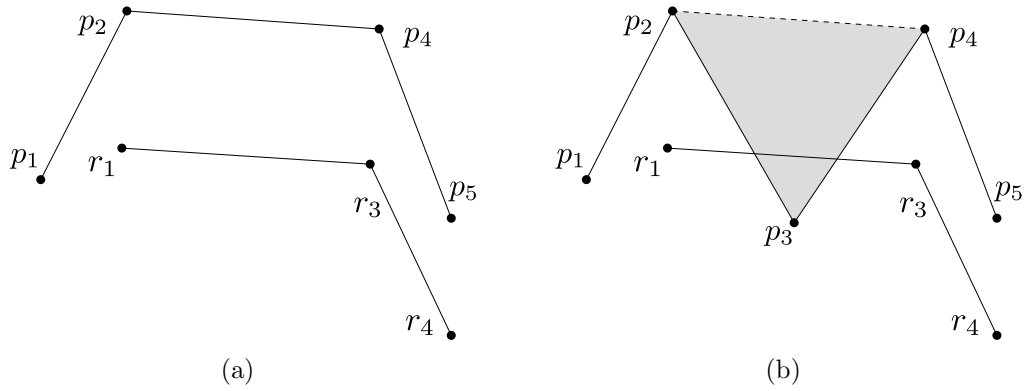


Figure 7: The scene in (a) is refined in (b) by the addition of the vertex p_3. This refinement creates a triangle shaped bounded face which is represented by the grey region in (b).

In order to simultaneously satisfy shape and topological constraints at each refinement step we propose to add the most significant vertex of T, in terms of the function K, such that the resulting map is topologically consistent. We now describe how this strategy is implemented. For each $s \in S$ we maintain a variable denoted $OR(s)$ which indicates the most significant element of $V(s)$, in terms of the function K, such that its addition returns a topologically consistent result. For example if $S = \{p, q\}$ as represented in Fig. 7(a), the variables in question would be $OR(p) = \emptyset$ and $OS(r) = r_2$. Each time an element is added to $s \in S$ through refinement, the variable $OR(.)$ corresponding to each element of $C(s)$ must be updated. For example if the element r_2 is added to r in Fig. 7(a), p_3 subsequently becomes a new candidate for $OR(p)$; therefore the value of $OR(p)$ must be updated. For an arbitrary element of S the time complexity required to compute $OR(s)$ is presented in the following theorem.

Theorem 2. *For $s \in S$ the most signification vertex of, in terms of the function K, such that its addition to $V(s)$ returns a topologically consistent result, can be determined in $O(\,|V(s)|\ log\ |V(s)| + |V(s)|\,|V(S)|\,)$ time.*

Proof. The elements of $V(s)$ are initially sorted by K value; this requires $O(|V(s)|\ log\ |V(s)|)$ time. These elements are then evaluated in descending order until an element is found such that its addition returns a topologically consistent result. At most $|V(s)|$ vertices will be evaluated. Each evaluation requires at most $|V(S)|$ time to determine if the added segments intersect

any object. Therefore the overall time complexity is $O(\,|V(s)|\;log\;|V(s)| + |V(s)|\,|V(S)|\,)$. □

4.4. Addition

Addition is performed by taking the union of the sets T and D; that is $R = D \cup T$. Simplification and refinement operators preserve topological relations between all objects in S and in turn all objects in R; therefore the topological constraint is satisfied. Also addition does not alter the shapes of the objects contained in R and therefore the shape constraint is satisfied. The set union $D \cup T$ can be performed in $O(|V(D)| + |V(T)|)$ time (Cormen et al., 2009).

5. Results

As discussed in section 1 in order to construct an automated map generation methodology two computational components must be implemented. The first corresponds to the definition of a set of transformation operators while the second corresponds to a methodology for determining a suitable application of these so that fit-for-purpose map representations are returned. The aim of the work presented is to implement the first of these components through the definition of a set of transformation operators which facilitate the exploitation of geometric coherence and in turn reduce the computation of applying transformations. As such, we do not aim to provide a complete solution to the challenge of automated map generation and in turn provide maps which are necessarily fit-for-purpose. As reviewed in section 2, all existing approaches to automated map generation have been based on the application of exclusively generalisation operators which do not exploit geometric coherence. Therefore in order to evaluate the proposed set of transformation operators relative to existing generalisation operators in terms of computational complexity a number of trials were performed.

Each trial was performed using the following methodology. Firstly a large scale dataset M was created. Next two adapted representation of M entitled D and R were defined such that they were different but exhibited significantly geometric coherence. Next we computed R using the proposed operators by computing a transformation where the corresponding domain and range were equal to D and R respectively. Finally we computed R using exclusively generalisation operators by computing a transformation where the corresponding domain and range were equal to M and R respectively.

For each of these transformations the corresponding computation time and the volume of data which was added and removed through enhancement and generalisation respectively was determined. This corresponds to the volume of data which must be transmitted to the client in order for them to received the desired representation R.

The proposed methodology was implemented using the following technologies. The server used was Python simpleHTTPServer. However any HTTP server capable of running Python can be used. All server-side components were written in Python while all client-side components were written in JavaScript. Extensive use was made of many HTML5 API's (Lubbers et al., 2010). Data transmission was performed using the HTML5 WebSocket API which defines a full-duplex communication channel between client and server. Client map visualization was performed using the HTML5 Canvas API.

The first trial performed was based on computing the adapted representation of Fig. 1 displayed in Fig. 2(b) which was discussed in Section 1. To evaluate the proposed set of operators D and R were assigned the representations displayed in Fig. 2(a) and Fig. 2(b) respectively. The transformation in question was computed in 4 seconds. It involved the addition of 2,159 object vertices through addition and refinement operators. It also involved the removal of 1,923 object vertices through selection and simplification operators. The total volume of data requiring transmission is therefore 2,159+1,923 = 4,082 vertices. To evaluate the use of exclusively generalisation operators M and R were assigned the representations displayed in displayed in Fig. 1 and Fig. 2(b) respectively. The transformation in question was computed in 7 seconds. The result contained 6,382 vertices and this corresponds to the volume of data which must be transmitted. It is possible to quantify the percentage of geometric coherence between two map representations as the percentage the data which does not require retransmission. In this example the percentage of geometric coherence between Fig. 2(a) and Fig. 2(b) is 36%. The above information corresponding to this specific trial is contained in the second row of Table 1.

The next trial performed involved computing the adapted representation of Fig. 8 displayed in Fig. 9(b). To evaluated the proposed set of operators D and R were assigned the representations displayed in displayed in Fig. 9(a) and Fig. 9(b) respectively. The sole difference between these two representations is that the orange line is represented in low detail in Fig. 9(a) and high detail in Fig. 9(b). The transformation in question was computed in 2 seconds. This fast execution can be attributed to the fact that the trans-

formation in question only involves the application of a refinement operator to the orange line. The total number of vertices added and removed by the transformation in question was 300. To evaluate the use of exclusively generalisation operators M and R were assigned the representations displayed in displayed in Fig. 8 and Fig. 9(b) respectively. All objects in Fig. 9(b) apart the orange line are simplified versions of their original forms in Fig. 8. The transformation in question was computed in 19 seconds. This result contained 14,245 vertices and corresponds to the volume of data which must be transmitted. The percentage of geometric coherence between Fig. 9(a) and Fig. 9(b) is 97%. The above information corresponding to this specific trial is contained in the third row of Table 1.

Many other trials using this testing procedure were performed; three of which are shown in the final three rows of Table 1. In all cases the proposed methodology which exploits geometric coherence offered superior performance compared to the traditional methodology of computing and transmitting each representation in isolation. It should also be noted that, in all trials performed the proposed methodology returned a solution within a time constraint which many authors would refer to as real-time (see section 1). However this is not the case for almost all trials performed using the traditional methodology.

Prop. Time	Prop. Data	Trad. Time	Trad. Data	% Coherence
4	4,082	7	6,382	36%
2	300	19	14,245	97%
3	3,268	5	5,872	44%
6	6,982	10	10,295	32%
1	100	13	8,874	98%

Table 1: Each row corresponds to a particular test trial.

6. Conclusions

The introduction of Web-GIS has brought about a major shift in the research goals of automated map generation. Users of Web-GIS demand map representations which are adapted to their requirements and computed in real-time. This is currently not possible due to the high computational

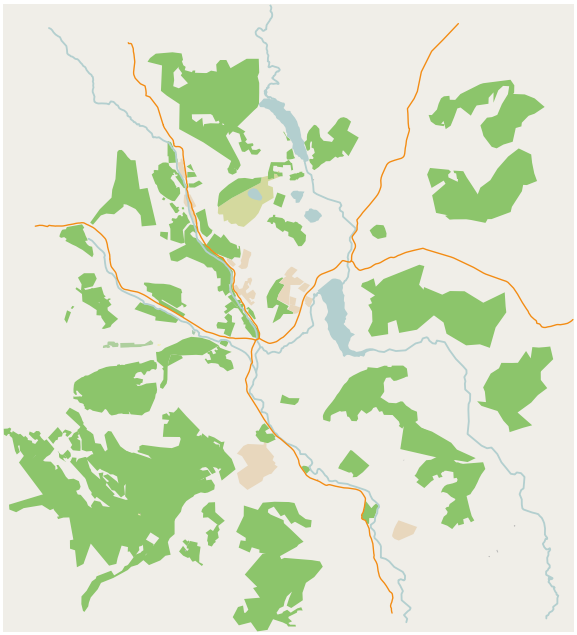


Figure 8: The original map from which the adapted maps in Fig 9 are derived.

complexity associated with automated map generation. To construct an automated map generation methodology one must firstly define a set of transformation operators which, when applied in a suitable manner, are capable of transforming a given map representation into the required adapted representation. In this paper we propose a new paradigm which exploits the geometric coherence in user spatial data requirements to reduce the computational cost associated with applying such transformation operators. Quantitative results demonstrate that relative to existing approaches, which do not exploit such geometric coherence, this approach offers reduced computation and transmission costs.

Due to the fact that the concept of exploiting geometric coherence is new to the field of automated map generation, the proposed work potentially represents the starting point for many possible research paths. Some of these are as follows. It must be noted that the proposed methodology is based on the assumption that a user's spatial data requirements will exhibit a geometric coherence. In future work the authors hope to perform user trials to evaluate if in fact this assumption is true. The proposed implementation

could be extended to provide more satisfying results by integrating more generalisation and enhancement operators. Another possible research path would be to integrate existing methods which attempt to predict a user's future spatial data requirements (Weber, 2010). If such a requirement could be accurately predicted computation could be initiated before the actual request is made by the user. This would in turn reduce the time delay between the user request and receipt of data. Finally in order to deliver a complete solution to the challenge of automated map generation, a method is also necessary for determining the subset of transformation operators to be applied, the order in which they should be applied and how they should be parametrised. The authors hope to built on the work presented in this paper and develop such a methodology.

Acknowledgements

Research presented in this paper was funded by the Irish Research Council for Science Engineering and Technology (IRCSET) EMPOWER program and the Irish Environmental Protection Agency (EPA) STRIVE programme (grant 2008-FS-DM-14-S4). The authors would like to sincerely thank the anonymous reviewers who's efforts significantly improved the quality of the research presented.

References

Badt, S., 1988. Two algorithms for taking advantage of temporal coherence in ray tracing. The Visual Computer 4, 123–132.

Bertolotto, M., De Floriani, L., Marzano, P., 1995. A unifying framework for multilevel description of spatial data. In: Frank, A., Kuhn, W. (Eds.), Spatial Information Theory A Theoretical Basis for GIS. Vol. 988 of Lecture Notes in Computer Science. Springer Berlin / Heidelberg, pp. 259–278.

Bertolotto, M., Egenhofer, M. J., 2001. Progressive Transmission of Vector Map Data over the World Wide Web. GeoInformatica 5 (4), 345–373.

Cecconi, A., Galanda, M., 2002. Adaptive Zooming in Web Cartography. Computer Graphics Forum 21 (4), 787–799.

Corcoran, P., Mooney, P., 2011. Topologically consistent selective progressive transmission. In: Geertman, S., Reinhardt, W., Toppen, F., Cartwright, W., Gartner, G., Meng, L., Peterson, M. P. (Eds.), Advancing Geoinformation Science for a Changing World. Vol. 1 of Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg, pp. 519–538.

Corcoran, P., Mooney, P., Bertolotto, M., 2012. Line Simplification in the Presence of Non-Planar Topological Relationships. In: AGILE International Conference on Geographic Information Science (Lecture Notes in Geoinformation and Cartography). No. In Press. Springer.

Corcoran, P., Mooney, P., Winstanley, A. C., 2011. Planar and Non-Planar Topologically Consistent Vector Map Simplification. International Journal of Geographical Information Science 25 (10), 1659–1680.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2009. Introduction to Algorithms. The MIT Press, New York.

Foerster, T., 2010. Web - based architecture for on - demand maps : integrating meaningful generalization processing. Ph.D. thesis, Enschede, University of Twente Faculty of Geo-Information and Earth Observation ITC.

Han, Q., Bertolotto, M., 2004. A multi-level data structure for vector maps. In: Proceedings of the 12th annual ACM international workshop on Geographic information systems. ACM, New York, USA, pp. 214–221.

Haunert, J.-H., Dilo, A., van Oosterom, P., 2009. Constrained set-up of the tgap structure for progressive vector data transfer. Computers and Geosciences 35 (11), 2191 – 2203, progressive Transmission of Spatial Datasets in the Web Environment.

Jones, C., Abdelmoty, A., Lonergan, M., van der Poorten, P., Zhou, S., 2000. Multi- Scale Spatial Database Design for Online Generalisation. In: International Symposium on Spatial Data Handling. Beijing.

Jones, C. B., 1997. Geographical information systems and computer cartography. Prentice Hall.

Jones, C. B., Ware, J. M., 2005. Map generalization in the Web age. International Journal of Geographical Information Science 19 (8-9), 859 – 870.

Kopf, J., Agrawala, M., Bargeron, D., Salesin, D., Cohen, M., 2010. Automatic generation of destination maps. ACM Transactions on Graphics 29 (6), 1–12.

Latecki, L. J., Lakmper, R., 1999. Convexity rule for shape decomposition based on discrete contour evolution. Computer Vision and Image Understanding 73 (3), 441–454.

Lehto, L., Sarjakoski, L. T., 2005. Real-time generalization of XML-encoded spatial data for the Web and mobile devices. International Journal of Geographical Information Science 19 (8&9), 957–973.

Lubbers, P., Albers, B., Salim, F., 2010. Pro HTML5 Programming: Powerful APIs for Richer Internet Application Development. Apress.

Luebke, D., Erikson, C., 1997. View-dependent simplification of arbitrary polygonal environments. In: Proceedings of the 24th annual conference on Computer graphics and interactive techniques. SIGGRAPH '97. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, pp. 199–208.

Mustafa, N., Krishnan, S., Varadhan, G., Venkatasubramanian, S., 2006. Dynamic simplification and visualization of large maps. International Journal of Geographical Information Science 20 (3), 273–302.

Oh, K. S., Shin, B. S., Gil Shin, Y., 2001. Mobility culling: an efficient rendering algorithm using temporal coherence. The Journal of Visualization and Computer Animation 12 (3), 159–166, enter text here.

Peng, Z.-R., Zhang, C., 2004. The roles of geography markup language (GML), scalable vector graphics (SVG), and Web feature service (WFS) specifications in the development of Internet geographic information systems (GIS). Journal of Geographical Systems 6 (2), 95–116.

Saalfeld, A., 1999. Topologically Consistent Line Simplification with the Douglas-Peucker Algorithm. Cartography and Geographic Information Science 26 (1), 7–18.

Sarjakoski, L. T., Sarjakoski, T., 2004. A Use Case Based Mobile GI Service with Embedded Map Generalisation. In: ICA Workshop on Generalisation and Multiple Representation.

Shea, K. S., McMaster, R. B., 1989. Cartographic generalization in a digital environment: When and how to generalize. Autocarto 9, 56–67.

van Oosterom, P., van den Bos, J., 1989. An object-oriented approach to the design of geographic information systems. Computers and Graphics 13 (4), 409–418.

Weber, B., 2010. Mobile Map Browsers: Anticipated User Interaction for Data Pre-fetching. Ph.D. thesis, The University of Maine.

Weibel, R., 1996. Advances in GIS Research II (Proceedings 7th International Symposium on Spatial Data Handling). London: Taylor & Francis, Ch. A Typology of Constraints to Line Simplification, pp. 533–546.

Weibel, R., 1997. Generalization of spatial data: Principles and selected algorithms. In: Algorithmic Foundations of Geographic Information Systems. Springer Berlin / Heidelberg, pp. 99–152.

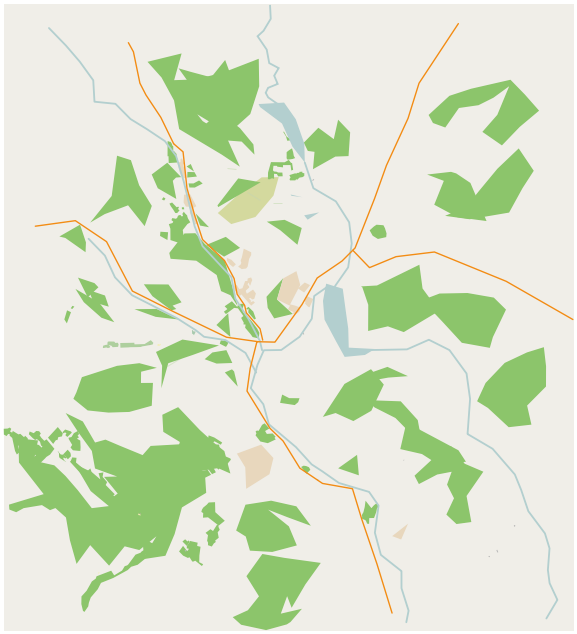
Weibel, R., Burghardt, D., 2007. Generalization, On-the-Fly. In: Encyclopedia of GIS. Springer, pp. 339–344.

Wilson, D., Bertolotto, M., Weakliam, J., 2010. Personalizing map content to improve task completion efficiency. International Journal of Geographical Information Science 24 (5), 741–760.

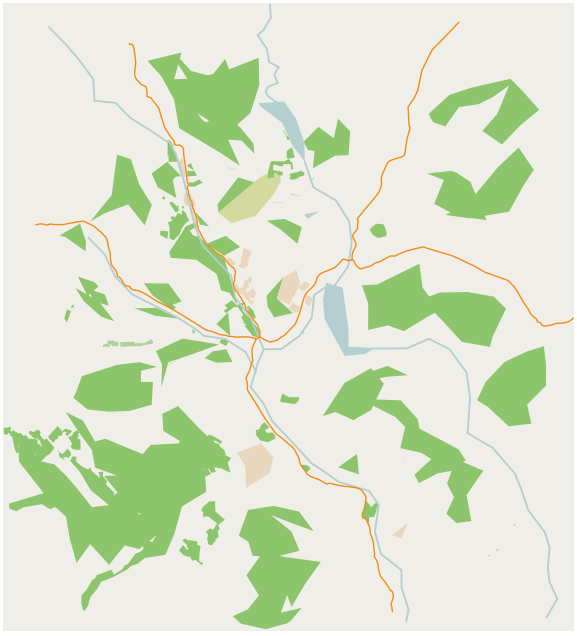
Yang, B., Purves, R., Weibel, R., 2007. Efficient Transmission of Vector Data Over the Internet. International Journal of Geographical Information Science 21 (2), 215–237.

Yang, C., Wong, D., Yang, R., Kafatos, M., Li, Q., 2005. Performance-improving techniques in web-based GIS. International Journal of Geographical Information Science 19 (3), 319–342.

Zhang, C., Li, W., 2005. The roles of web feature and web map services in real-time geospatial data sharing for time-critical Applications. Cartography and Geographic Information Science 32 (4), 269–283.



(a)



(b)

Figure 9: The maps in (a) and (b) are adapted in order to fulfil user requirements.