

Hierarchical GML Modeling of Transportation Networks

Keywords: Web mapping, Hierarchical representations, Cartographic generalization, Project OneMap

Gunnar Misund

Associate Professor
Østfold University College
Halden
Norway
gunnar.misund@hiof.no
<http://www.ia.hiof.no/~gunnarmi/>

Biography

Gunnar is teaching and researching at Østfold University College, Halden, Norway. His main research interests are digital (web) mapping and distributed computing. He is the founder of Project OneMap.

Knut-Erik Johnsen

M.Sc. Student
Østfold University College
Halden
Norway
knut.e.johnsen@hiof.no

Biography

Knut-Erik is studying computer science at Østfold University College, Halden, Norway. He is working mainly with solutions for web services, management of large data repositories and web mapping. He is a member of the Project OneMap Team.

Mats Lindh

M.Sc. Student
Østfold University College
Halden
Norway

mats.s.lindh@hiof.no

Biography

Mats is studying computer science at Østfold University College, Halden, Norway. Among his interests are algorithm design, game programming, web computing and component based systems. He is a member of the Project OneMap Team.

Abstract

Management, presentation and analysis of huge amounts of geodata calls for hierarchical representations. In this paper we discuss simplification strategies for networks, where both topology, geometry and thematic properties are taken into consideration. We present GML based methods applied to the complete US road network derived from US Census data represented in Tiger Line files. The issues discussed in the paper should be of interest in wide range of applications. However, the main target application is the geodata Repository in Project OneMap (1). One of the objectives of this project is to incrementally build a large, distributed global online map.

Table of Contents

1. Background

2. OneMap Generalization

2.1 Project OneMap

2.2 Extreme Generalization

2.3 Grid Approximation

3. Related Work

3.1 Selection

3.2 Clustering

4. Proposed Methods

4.1 Geometric Generalization

4.2 Semantic Generalization

4.2.1 Road Type Grouping

4.2.1.1 Road Class

4.2.1.2 Road Name

4.2.2 Stroke Extent Grouping

5. GML Road Modeling

5.1 GML Modeling

5.2 Existing GML Road Models

5.3 OneMap Design Considerations

5.4 OneMap Storage Schemas

6. Tiger Line Examples

6.1 Data Source

6.2 Generalization

6.2.1 Grid Approximation

6.2.2 Road Class

6.2.3 Stroke Extent

7. Future Work

8. Final Remarks

Footnotes

Acknowledgements

Bibliography

1. Background

The work presented in this paper [1] is part of an ongoing long term effort named Project OneMap [MIS1]. The main objective of this project is to incrementally build a large, distributed global online map. The intention is to provide free access to a repository of geographic information that may be used as base data for a wide variety of applications, such as GIS (Geographic Information Systems) analyses, environmental planning, tourist guides, transportation management and the like. Project OneMap is an open project in every respect, as it is based on Open Source [OSR] and Open Content [OCN] principles, and also adopts a distributed management model. The project is currently hosted by and coordinated from Østfold University College, School of Computer Science

Much of the background work was carried out as part of a student project in the Digital Map course at Østfold University College in the spring of 2003 [DIG] [MIS5]. The authors of the paper were connected to the course, as supervisor and students.

The first part of our paper focuses on network simplification. We give a brief outline of Project OneMap, point to some related work on road system generalization and describe the general principles of our proposed methods. In the second part we discuss GML (Geographic Markup Language) as a geodata modeling tool. We point to some existing GML road models, and highlight some OneMap design considerations with implications for our modeling strategy. We then present the schema family defining our internal storage of road systems. In the third part we assess the proposed simplification methods by applying them to a data set encompassing the complete system

of US roads, as present in the US Census Tiger Line files. Finally we suggest directions for future development and draw some conclusions.

2. OneMap Generalization

2.1 Project OneMap

In order to clarify the main reasons for investigating hierarchical network representations, we give a brief outline of Project OneMap, which is our main target application. The OneMap infrastructure has three main components:

Repository

The repository is a distributed storage infrastructure. This is basically a (huge) set of GML files structured to efficiently support retrieval and updating of the geodata comprising the world map. The total amount of base data is currently in the magnitude of 50 M points, and is expected to expand to around 500 M points within the next few months. The data is hierarchically structured in levels of detail to facilitate global overviews as well as street level inspections. The files are distributed redundantly on a set of servers. This enables both storage scalability and parallel processing of retrieval queries and updating procedures. The main interface to the repository is defined as an [OGC](#) (Open GIS Consortium) Web Feature Service [\[WFS\]](#) implemented with a (slightly modified) open source framework from Deegree [\[DEG\]](#).

Gateway

The Gateway is a browser based user interface for retrieval of OneMap data. We are currently using an SVG/JavaScript implementation, which provides simple but sufficient navigation and query possibilities [\[MIS4\]](#). The Gateway is based on the OGC Web Feature Server Specification [\[MIS4\]](#).

Clearinghouse

One of the main objectives of OneMap is to make it possible to build a huge map in an incremental and uncoordinated manner, with contributions from a wide variety of parties, but still guarantee a reasonable level of reliability and quality. We address the challenge of integrating submissions with existing geodata by applying the well known principles of peer review [\[MIS3\]](#).

The alpha versions of Repository and Gateway has been up and running for a year, and the beta version will be released later this year. The initial version of Clearinghouse will be launched later this year.

2.2 Extreme Generalization

The design of the OneMap infrastructure relies on three crucial assumptions, which have some important implications.

Global coverage

Each featuretype collection is assumed to have global coverage. Not only do the obvious features, like the world coastline, follow this requirement, but also featuretypes like roads will in principle span the whole globe. This has some clear advantages for the user of OneMap, e.g. that it will be possible to extract the whole road transportation network in Europe as a single topologically consistent entity, which may serve as valid input to a variety of network algorithms, like shortest paths computations. However, this requirement poses some serious challenges for design and implementation of the infrastructure components.

Implications

The various featuretype collections in the OneMap Repository will inherently be of substantial size.

Extreme zooming

Long range zooming, from satellite view to street level detail, may be realized by two different techniques: 1) On-the-fly generalization of the original (most detailed) dataset, or 2) Preprocessed sets of approximations representing the original data in levels of decreasing detail.

Implications

The first approach may be technologically feasible, but they are inherently computational intensive. Since our feature collections are of global coverage and high detail, we regard preprocessed levels of detail as the only sound strategy for the OneMap infrastructure.

Incremental updating

The OneMap Repository is designed to facilitate incremental growth, by a set of smaller contributions rather than one-time installments of large quantities of data. This is orthogonal to the more common centralized strategy used by most mapping agencies.

Implications

Updates and modifications have to be carried out by *local* methods involving only relatively small subsets of the total amount of information. However, to the authors' knowledge, the majority of commonly used generalization methods are indeed of global nature, like the popular Douglas-Poiker line thinning algorithm.

All in all, the OneMap design calls for what we call *extreme generalization*, regarding both the size of the base data and the zoom range. This implicates that the OneMap Repository has to store the geodata in a sufficiently large number of levels of detail. Further, each of these levels have to (directly or indirectly) be generated by generalization methods working in a local manner, which

makes it possible to regenerate only small subsets of the original storage structure to incorporate submissions of new data sets or modifications of the existing data.

So far, only one strategy has been found meeting these requirements, and will be discussed in the next section.

2.3 Grid Approximation

The strategy chosen for the OneMap infrastructure is conceptually very simplistic, on the verge of being naive. Consider each level of detail as a resolution grid, where the grid cell size reflects how small objects the system is able to handle. One may conceptually view this as the display matrix in a computer monitor. The assumption is that all information "smaller" than a grid cell ("pixel") is superfluous, and should be discarded in the approximation process. This is accomplished by "snapping" each coordinate in the original data set to its nearest grid point. Obviously, many consecutive points may snap to the same point (at least at the coarser levels), and only one of these points is used in the approximation. In [Figure 1](#) we have a 3x3 grid, and point A snaps to gridpoint (1,1). The points B, C and D also snap to the same place, but these points are discarded.

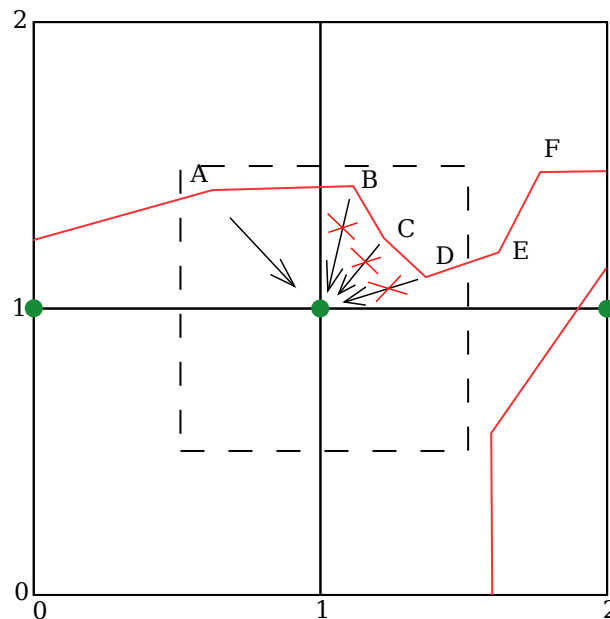
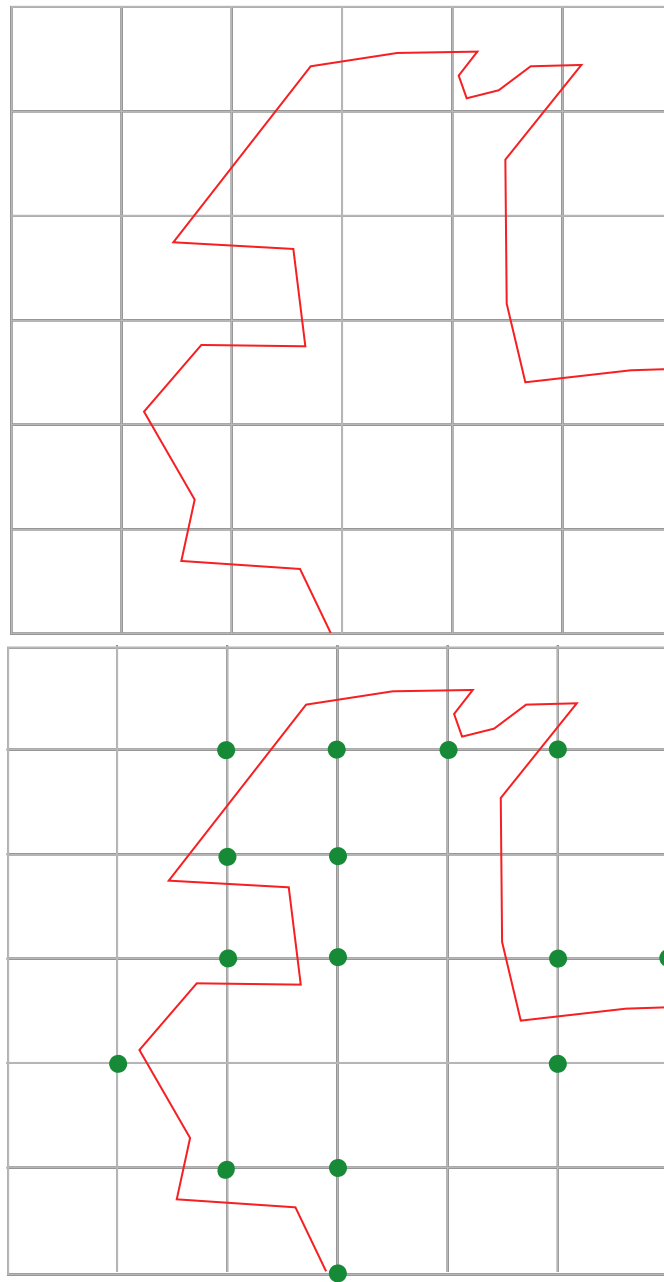


Figure 1: Point Snapping

The snapping procedure, which we call *grid approximation*, is the main building block in the different OneMap generalization methods. A more complete example is given in [Figure 2](#).



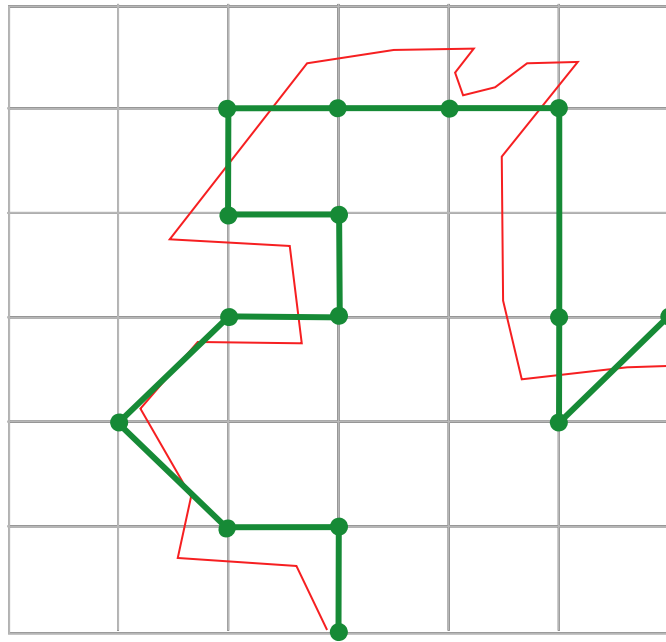


Figure 2: Grid Approximation

For further reduction of the number of points, it is convenient to remove redundant points in straight lines. In addition, it is possible to convert "staircase" segments to diagonals (**Figure 3**).

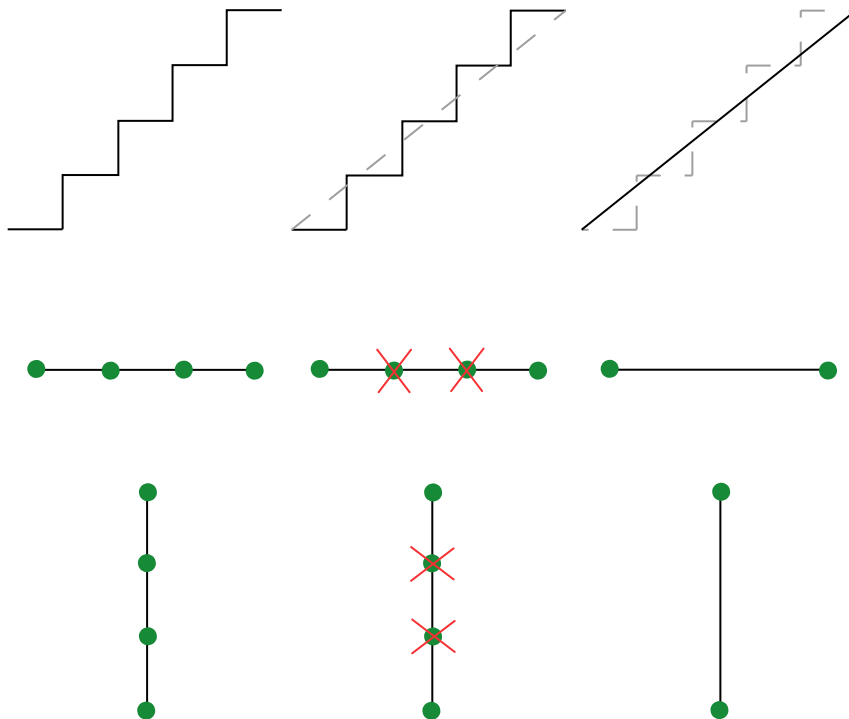


Figure 3: Additional Point Removal

The OneMap generalization strategy was first published in July 2002 [MIS2] . A similar approach was published simultaneously by Doihara et. al. [DOI] , where they in addition to the point removal property also highlight two other important characteristics:

Elimination

A feature with a geometry that entirely collapses to the same gridpoint, may be completely removed from the resulting dataset.

Aggregation

Adjacent features, say two buildings, may "melt" together, in the sense that they share some part(s) of their boundaries. This may be used to merge a set of smaller objects into a larger one.

The most important property in the OneMap context is the elimination effect. However, we are currently investigating how to exploit the aggregation property. Different feature types and different geometries call for variants of the outlined grid approximation method. In the following, we focus on how to apply the strategy on feature types represented as networks, i.e. sets of nodes more or less connected with edges.

3. Related Work

Most map generalization methods may roughly be decomposed in two relatively independent processes: 1) Semantic simplification, i.e. selection of relevant types of information, and 2) Geometric simplification, i.e. adjustment of the level of visual detail of the selected information.

Geometric network approximation is basically carried out by reducing the number of nodes and edges in the original model. In this section we briefly review two main directions in geometric generalization of networks, from which our proposed methods borrow selected elements.

3.1 Selection

The most common way to reduce the complexity of a geographic network is to remove some of the edges and the nodes, or in other words, select a subset of the nodes and edges (see **Figure 4**).

Typically, a node and its incident edges may be removed if the spatial extension of a node (defined by the geometry of the incident edges) and its edges is below a certain threshold. The selection/removal process often incorporates sophisticated heuristics based on application specific parameters, such as the relative importance of edges and/or nodes. An example of this strategy is documented in [KRE] .

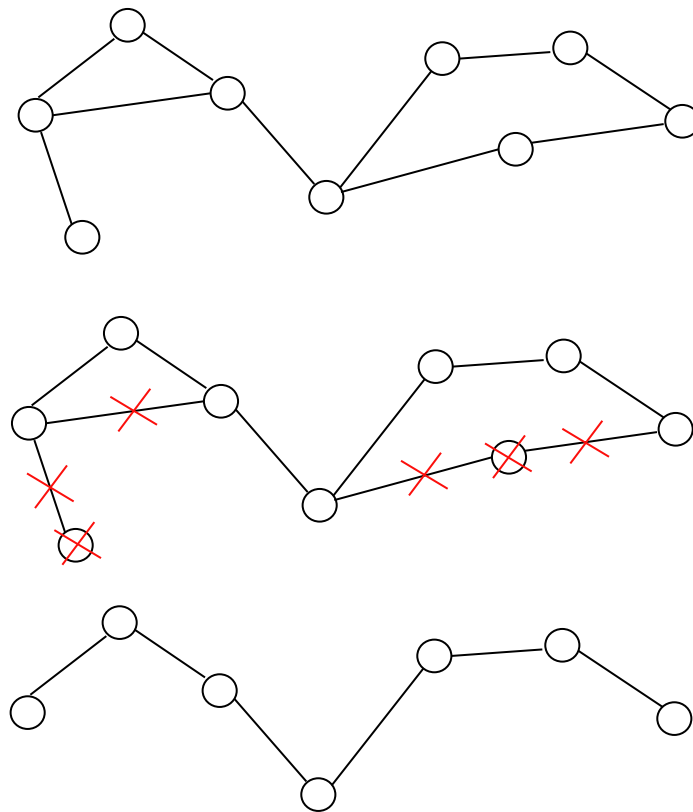


Figure 4: Node and edge selection

Thomson et. al. [THO] have proposed and implemented an edge selection algorithm inspired from classical results in hydrology. Horton [HOR] classified river system by indexing each branch according to a set of criteria, like straightness, junction angles etc., all based on natural flow characteristics of streams. Thomson et. al. build on this technique, and merges consecutive road segments into so-called *strokes*. Their overall criterion is to ensure "good continuation". The method generates a ranking of the strokes according to their importance. The selection process is carried out by only keeping strokes with a significance above a certain threshold. The method is illustrated in **Figure 5**, where the original network is decomposed to three strokes.

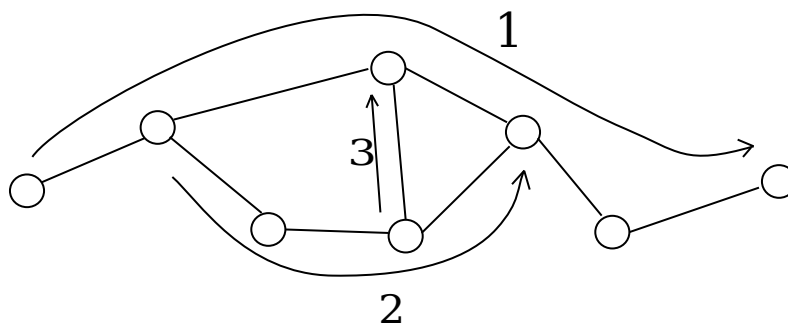


Figure 5: Stroke grouping

3.2 Clustering

Graph clustering and its close relative, graph partitioning, are well studied techniques used in many applications, such as VSLI design, molecular biology and data mining. This approach has also been applied to cartographic network generalization. The main idea is to identify clusters of nodes, e.g. according to Euclidian proximity. Most often the subgraph defining the cluster has to be connected such that there is a path from all nodes to any other node in the cluster. The clusters are represented by virtual nodes. Virtual edges are introduced between clusters that are considered neighbors according to some definition, e.g. that there exists an edge from one of the nodes in the first cluster to one node in the second ([Figure 6](#)). See for example [\[MAC\]](#) for details of this technique.

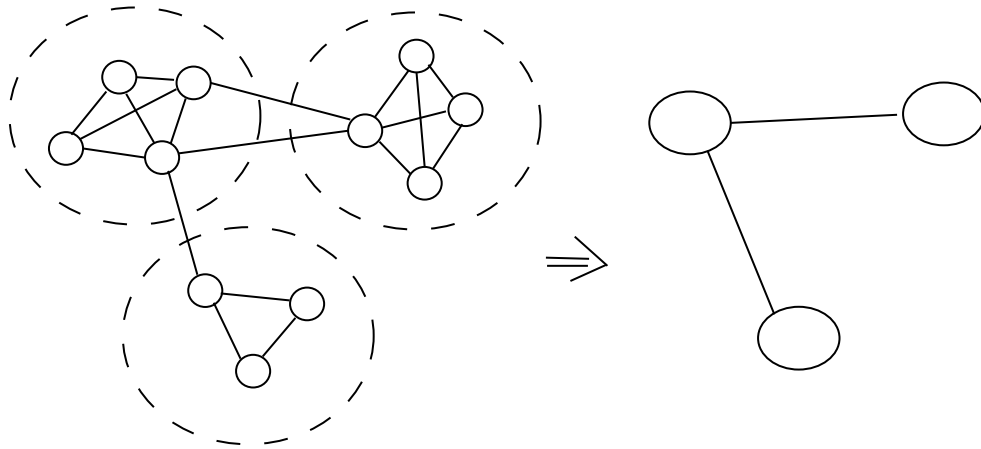


Figure 6: Node and edge clustering

4. Proposed Methods

In the following we describe our proposed approximation methods. We call the general approach *network gridding*. Note that we are primarily concerned with the geometric properties of the resulting approximations, and leave the topological aspects for future research.

The geometric component of all methods is based on the grid approximation paradigm. However, the methods differ in the semantic view of the road system.

4.1 Geometric Generalization

It's worth noting that the grid approximation method to some extent inherently supports both subset selection and node clustering. Edges collapsing to a single point may be discarded. On the other

hand, a set of nodes snapping to the same grid point may be conceptually viewed as a cluster of nodes represented by the virtual node corresponding to the grid point. The methods described in the next section take advantage of these properties of the grid approximation.

4.2 Semantic Generalization

In order to provide means for choosing a semantically "right" selection of roads for a given level of detail, we need to decompose the entire network into subsets of differing importance. We suggest two methods, one based on the types of roads, and the other on the extent of strokes, i.e. some logical grouping of consecutive road segments.

4.2.1 Road Type Grouping

The first approach is based on decomposition according to road type. Each subset will then constitute a relatively homogenous feature collection, regarding semantic significance. It will be up to the user (or the application) to select the appropriate road type(s) for the given purpose. We suggest two ways of generating semantic homogenous network subsets.

4.2.1.1 Road Class

It is common to categorize roads according to some authoritative catalogue. The roads mapped by the US Census Bureau are categorized by the Census Feature Class Code, CFCC for short [TIG]. **Figure 7** shows all roads of the CFCC classes *A11* to *A18*, which together defines the road type "Primary Highway With Limited Access". The representation is approximated with grid resolution of approximately 0.125 degrees (roughly 11 km).

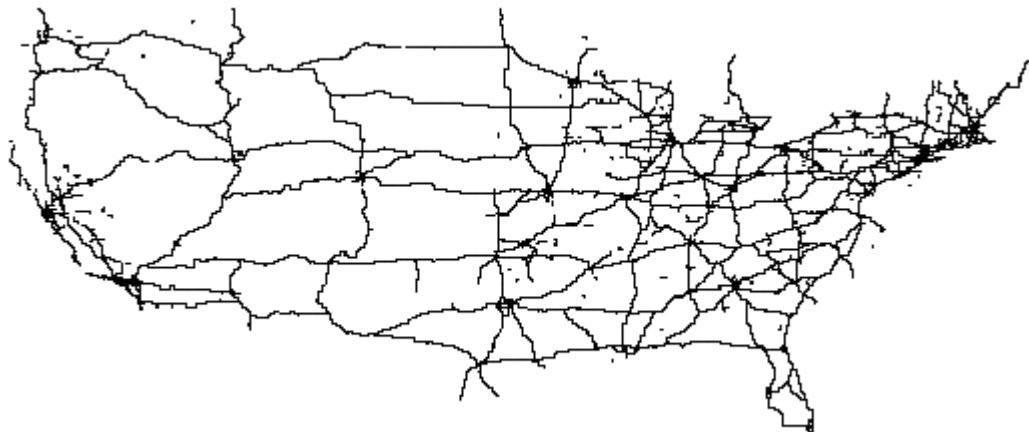


Figure 7: CFCC classes A11 - A18, resolution

ca. 11 km

The "official" web gateway to the US Census data, the Tiger Map Server [\[MAP\]](#) , is not able to display the continental wide road system, as illustrated in [Figure 8](#) . The authors believe that this is due to missing efforts on generalization of the census data.



Figure 8: Tiger Map Server response on requesting continental wide map of highways

4.2.1.2 Road Name

A common attribute in road models is the name of a segment. Most often the name carries information of the type of the road, such as "Sunset Boulevard", "Main Street", "Memorial Drive" and "US Highway 11". We propose to use the type component of the name as a key, and consider e.g. all roads containing the string "US Highway" (or legal variants) to belong to the same feature class. In cases where alternative names are provided, we choose the name with highest "precedence" as the classification key (as an example, "US Highway 11" has higher precedence than "Main Street"). In [Figure 9](#) all "US Highway" roads are represented with 0.125 degree (roughly 11 km) resolution.



Figure 9: US Highways, resolution ca. 11 km

4.2.2 Stroke Extent Grouping

In this method we follow the principles of generating *strokes* [THO]. Conceptually we connect consecutive segments carrying the same name and compute the bounding boxes of the strokes, and the extent is set to the largest box side. We define a set of feature classes such that each class will contain strokes where the extent fall within a given size interval. The left picture in **Figure 10** shows all roads in New York County with stroke extent larger than 10.5 km. The center map shows strokes in the interval 657 to 1314 meter, and in the last one we find road strokes with extent in the interval 164 to 328 meter.

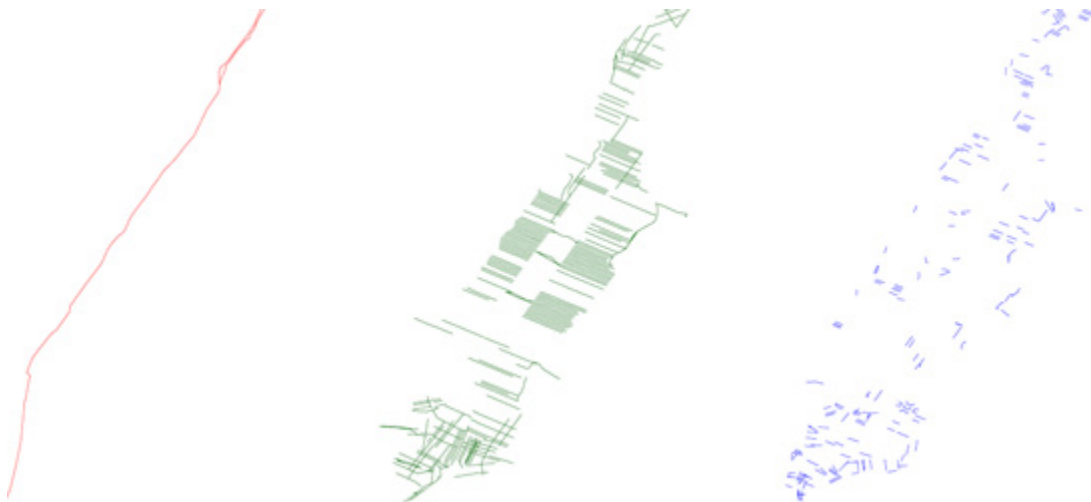


Figure 10: New York County roads sorted on stroke extent

5. GML Road Modeling

Before giving details on the OneMap Road Model, we discuss briefly some aspects on GML modeling in general and point to a few other road modeling efforts. We also emphasize some general design goals in OneMap with important implications for our application schemas.

5.1 GML Modeling

Interoperability is the key rationale for the work of OGC, and GML is meant to be *the* tool for allowing geodata to flow freely between various applications. It is however a fairly young metastandard, and very few large scale GML projects have been undertaken. There is one notable exception, the OS MasterMap series from British Ordnance Survey (see the following section). This initiative has shown that it is possible to model and represent large and real life topographic datasets. On the other hand, smaller demonstrator projects also indicate that GML may be able to act as the integrating element when mixing data and formats from different sources.

Since GML basically is a metastandard, the different application areas are supposed to develop their own specific profiles and/or extensions. This is both a strength and a weakness of GML. On one hand, GML has the potential to significantly penetrate the geodata scene, in both a vertical and horizontal manner. On the other hand, since a GML document is not following an unambiguous standard, such as e.g. SVG and MathML, it may be extremely difficult to handle in multi purpose applications, like for example an editor. For the same reason, the market has not yet seen too many all-round GML viewers, able to treat instances of arbitrary GML schemas. With the advent of GML 3, the scope of the standard dramatically widened, including aspects like proper topology representation and coverage modeling. However, the added complexity will in many ways increase the problems of using GML in general purpose settings, handling simultaneously a plethora of differing GML schemas.

5.2 Existing GML Road Models

There have been some scattered efforts on modeling road networks with GML, of which we mention a few.

GML Specifications Examples

The specifications for both GML 2.x [\[GML2\]](#) and GML 3.0 [\[GML3\]](#) use road models as examples. The GML 2.x model is quite limited, but the 3.0 edition extends the models by adding complex geometry and topological constructs.

OS MasterMap

The British mapping agency, Ordnance Survey, are offering their major topographic series, the OS MasterMap, on a GML 2 format. The map sheets include various road information, and the interested reader may download and study their schemas from [\[OSM\]](#). The MasterMap series is, to the authors knowledge, the first and only full-scale deployment of GML technology.

TOP10NL

The Dutch Topographic Service has carried out a comprehensive prototype project to model and represent their basic topographic data sets with GML [\[TOP\]](#). A TOP10NL data set were used as a case under the 2nd GML Relay [\[REL\]](#). The main objective of this event is to demonstrate interoperability issues by passing GML data back and forth between various GML enabled software from leading vendors. The specifications for both GML 2.x and GML

3.0 [TOP] uses road models as examples. The GML 2.x version is quite limited and basic, but the 3.0 edition elaborates the models by adding more complex geometry and topological constructs.

Tiger Line Files

Daisy [DAI] reports an effort to translate US Census Tiger Line files (including road information) to GML 2. He refers to an open source Java conversion framework made by Galdos Systems Inc and a OGC demonstrator distributing GML converted Tiger Files over New York City in a WFS application.

5.3 OneMap Design Considerations

The OneMap Repository stores geodata in large filesystems. Each featuretype is preprocessed and translated to GML, and multiple levels (typically 25) of generalized versions are generated. Level #n has a resolution of 2^{-n} degrees. A level is often referred to as layer, according to cartographic tradition. Each layer has global extent, and to facilitate efficient queries and retrieval, the layers are tiled by adaptive quadtree subdivision, such that the size of each tile is below a given threshold (see Figure 11). The tiles are termed sheets.

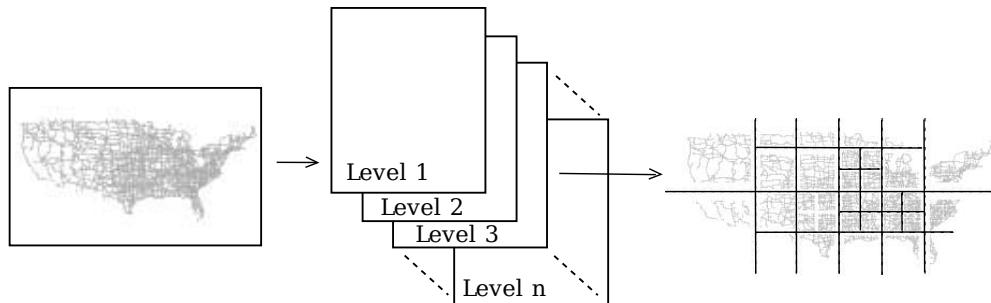


Figure 11: OneMap Repository file structure

The repository of generalized and subdivided feature collections is the heart of OneMap. This is the core legacy part of the infrastructure that should gracefully survive the rapid shifts in the information technology society. Ideally, a single OneMap sheet should be self explanatory, even taken out of context. A consequence of this is that external linking should be kept at a minimum. Without extensive external linking, the ability to model topology is quite limited. However, due to the grid generalization method and the tessellation of each layer, we may construct the necessary topology on-the-fly when needed.

Another reason for keeping things simple, is that the OneMap infrastructure aims at managing base geodata, and not necessarily the most complex objects and topologies. We consider it our main task to supply raw material which should be refined to fit the specific needs from various application

areas. Hence, models of moderate complexity should suffice.

OneMap is designed to handle a wide variety of different geographic objects in a reasonably homogenous manner. Hence, we can not afford to develop specialized methods for every feature type. Rather, we have to rely on "one-fit-all" models and algorithms. This is however a challenging trade-off situation, and obviously, corners have to be cut to follow this requirement.

The grid approximation methods make it difficult to use anything else than pointset geometry with linear interpolation. The tessellation process and incrementally updating requires splitting of pointsets into ordered sets of segments. The GML 3 specification lacks specific support for general geometry segmentation (except for the Curve type).

As a summary, we may formulate the paramount guiding principle in Project OneMap: *One should try to make things as simple as possible*. The authors think that any infrastructure has a better chance to survive, and indeed, evolve, if it is kept as simple as possible. This applies in particular to the GML schemas, as they are considered the main design tool for structuring the content of OneMap.

5.4 OneMap Storage Schemas

The foundation for our schemas is the most basic definition of a geographic feature, which goes as follows. A feature has a *type* that semantically should give an idea of what kind of real life object it represents. Moreover, it is defined by a set of *properties*. Each property has a (semantically meaningful) name, optionally a datatype, and finally a value. The properties have traditionally been geometric or non-geometric. This is the bottom line of the [ISO](#) (International Standardization Organization) Simple Feature Specification.

Our case example, the road network, is modeled as a set of independent road segments, with a few required thematic properties, such as CFCC class and name. Zero or more properties may be added. The geometry of the segment is a ordered set (one or more) linestrings with piecewise linear interpolation. Note that no nodes are present in the model, and no explicit topology is imposed on the set of edges. However, given that two edges share a common point, e.g. a junction, they will in all the simplified levels still share exactly this node. This is a direct consequence of the grid approximation method which creates a set of detail levels of finite precision. Hence, assuming that the selected area is not excessively large (relative to the resolution), it is not too computationally expensive to build a proper graph topology from the scattered edges.

Since we adopt the idea of modeling generic feature types rather than a set of specific and differing feature classes, we may apply the following schema family for many types of geodata, including the road network. Note that the OneMap internal storage schemas differ from the schemas used when exporting data from of the infrastructure, but a description of these are beyond the scope of this paper.

The GML schemas are not meant to be used as is. The main way to develop specific application schemas is to *extend* the base schemas to support ones specific needs. However, since the scope of GML is rather broad, another strategy is to only use a subset of the standard, commonly known as a *profile*. The resulting application schemas are then derived by extending this profile.

Our main modeling strategy is to work with a narrow GML profile, and to derive specific schemas by minor extensions, mainly to support geometry segmentation. In addition we adopt a simplistic base model of a feature, as a set of geometric and non-geometric properties. We provide minimal support for topology.

We have chosen a model with a relatively large number of small schemas with narrow scope, rather than a few schemas defining many elements. The majority of the schemas are extensions of a rather radically simplistic GML 3 profile, in fact conceptually a subset of GML 2. However, we take advantage of some of the added functionality in GML 3, and avoid completely the deprecated constructs included to ensure backwards compatibility.

In the following we give a brief description of the different schemas extending our base profile. The interested reader may explore the family of schemas in [\[SCH\]](#).

om_generic_feature.xsd

The GML 2 specification did in fact provide a "one-fit-all" schema, the `gmlpacket.xsd`, proposed as a possible exchange and transport format for a broad class of geodata. It is based on the simplistic property-value model, where the feature type is represented as a property, and not as the element type. We have built on this schema when designing the multipurpose `om_generic_feature.xsd` schema.

om_geometry.xsd

All geometry in OneMap has to be defined as point wise linear interpolation of pointset structures. The only only accepted basic geometry types are points (0 dimensional), polylines (1 dimensional) and polygons (2 dimensional). However, both polylines and polygons are defined by *segments*, due to arguments presented in [Section 5.3](#). Hence, we have made a narrow profile of the geometry schemas in the GML 3 specification. Further, we have added the `om_geometry.xsd` to extend the basic geometry types to support geometry segmentation.

om_metadata.xsd

We have stripped the GML 3 metadata and extended it with constructs from the Dublin Core repertoire [\[DUB\]](#). The main purpose with the OneMap metadata is to point to the original source of the data and the person or party which has submitted the data. It is equally important to state the modification level (e.g. if and how the data may be updated).

om_dependencies.xsd

This schema defines how to express dependencies among feature instances. This could e.g. be used to ensure that a crude representation of a coastal highway still remains onshore if the coastline is refined and potentially may intersect the highway geometry.

om_layer.xsd

Each layer in the OneMap Repository is represented as a set of levels of details, and this schema defines the necessary elements to characterize each layer.

om_sheet.xsd

Each level of detail of each layer is split into a set of sheets. A sheet has a local discrete coordinate system based on resolution and location, and this schema ensures that all sheets carry the necessary information to transform it back to the global geographic coordinate system.

6. Tiger Line Examples

We present some examples, all based on the 2002 TIGER/Line® Files from U.S. Census Bureau [\[UCS\]](#) .

6.1 Data Source

The Tiger Line files are in the public domain, and may be downloaded and used at no cost. The files carry information on a large number of feature types, e.g. roads, buildings, lakes and county borders. The total size is 3.5 GB compressed and roughly 50 GB unzipped. The format is in plain ascii and well described. In our case we have only extracted road related data.

The census data is distributed county-wise, with up to 17 files for each county. In our methods, in particular the stroke based algorithm, we have to build the topology which is implicitly modeled in the base data. One of the principles in OneMap is that it must be possible to run all parts of the infrastructure on standard hardware. The census data contains around 32 million road segments defined by roughly 124 million points. It is obvious that it is not possible to handle the complete road model by using internal memory based algorithms, hence our methods use external memory (disk) as temporary storage during processing.

6.2 Generalization

6.2.1 Grid Approximation

As previously described, the basic geometrical simplification is based on the grid approximation paradigm. In order to quantify the effects of this method, we logged the results from applying it to the complete road system in New York County. The original data contain 8329 segments and 17451

points. In the table below we list the percentages of segments and points (relative to the original data) on the different levels of detail.

Resolution (meter)	11	22	43	87	173	346	692	1384	2769	5539
Segments	100	100	96	78	41	13	3.7	1.04	0.35	0.07
Points	99	97	92	74	40	13	3.5	1.00	0.33	0.07

Table 1

6.2.2 Road Class

In **Figure 12** we show an example of using the road classification as a grouping criterion.

The left map shows the CFCC classes A1 (red) and A2 (blue) in New York State. A1 is "Primary Highway With Limited Access", and A2 "Primary Road Without Limited Access".

In the center map we have zoomed in and included class A3 (green), which is "Secondary and Connecting Road".

The last map shows Lower Manhattan with the remaining classes added: A4 (purple, "Local, Neighborhood, and Rural Road"), A6 (brown, "Road with Special Characteristics") and A7 (cyan, "Road as Other Thoroughfare"). Class A5 ("Vehicular Trail") is, by no surprise, not present in the dataset.



Figure 12: Road Class Grouping

For a reference we include a map (**Figure 13**) corresponding to the right picture in **Figure 12** downloaded with the Tiger Map Server [\[MAP\]](#) .



Figure 13: Manhattan roads from Tiger Map Server

6.2.3 Stroke Extent

Figure 14 shows an example of using stroke extent, or the "size" of contiguous segments with the same name, as generalization parameter. The left map includes all strokes, i.e. the complete road system. In the center map we are zooming out, and select the strokes with extent larger than 346 meter. In the last map there are only strokes larger than 5539 meter.



Figure 14: Stroke Extent Grouping

Figure 15 shows a map generated with the Tiger Map Server. It corresponds to the middle picture in **Figure 14**. This illustrates the potential benefits of using the stroke extent property as the semantic generalization criterion.

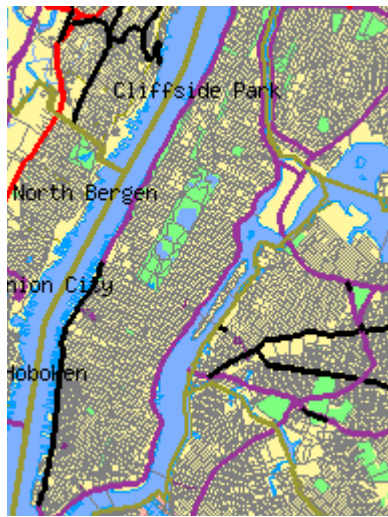


Figure 15: NY City roads from Tiger Map Server

7. Future Work

The US census data includes other types of network, such as river systems, and we are planning to adapt our methods to also be applicable to these featurtypes. This would be part of an effort to incorporate the complete US Census data in the OneMap Repository.

On our agenda is also to deploy the methods on road data from other sources, the first candidate being the Norwegian Road Authority. This would be an important step to to ensure generality and applicability of our algorithms.

However, we think the most important extension of our work is to investigate how our strategies could support efficient graph algorithms, like shortest path computations and solutions to the Traveling Salesman Problem. The initial approach will be to use the multiple representations to compose optimally structured networks as input to standard graph algorithms. As part of this work we also have to explore and take advantage of the topological components in GML 3.

8. Final Remarks

GML is used as modeling language in virtually all components of OneMap. Schemas define formats used in input, exchange between different components, storage and output. By using GML as the

glue in the OneMap infrastructure we are able to manage a set of apparently quite complex tasks. We may reuse components, such as parsers, and take advantage of many high grade tools (often free and open source). The use of GML schemas provides an efficient means for defining sound and (relatively) unambiguous formats. We have also experienced that GML, being an XML application, is able to bridge the gap between the cartographic/GIS oriented community and the more mainstream computer science people [MIS4].

We have used road networks as the case application. However, the authors are of the opinion that the same methods, with minor modifications, may be adapted for other networks, like railways and canals, and even river systems.

Finally, we think our methods may enhance existing web based interfaces to large repositories of network data, such as the Tiger File Server.

Footnotes

1. This paper is part of the OneMap Documentation Series (OMD 00503). The OneMap publications serve two purposes: 1) To disseminate knowledge and experiences that might be useful for other parties and applications, and 2) To encourage feedback and discussion in a broad setting. The presented results are *not* to be regarded as "final solutions", rather as a starting points for further discussions and developments.

Acknowledgements

The authors are grateful for the financial support offered by Østfold University College, which made it possible to finalize and refine results from the Digital Maps course. It also made it possible for part of the OneMap Team to attend the GML Dev Days conference.

Bibliography

[DAI]

Implementing GML Schema for U.S. Census TIGER/Line files. Paul Daisey, in Proceedings, Digital Earth 2001, Fredericton, Canada, June 2001 http://www.digitalearth.ca/ses11_25.html.

[DEG]

deegree 1.0.11. Project Deegree <http://deegree.sourceforge.net/index.html>.

[DIG]

Digital Maps. Gunnar Misund, Østfold University College <http://www.ia.hiof.no/digmap/>.

[DOI]

An Adaptive Lattice Model and its Application to Map Simplification. T. Doihara, P. Wang and W. Lu, in Proceedings of Geospatial Theory, Processing and Applications, ISPRS Commission IV, Symposium 2002, Ottawa, Canada, July 9-12, 2002, IAPRS Vol. XXXIV, part 4, ISSN 1682-1750 <http://www.isprs.org/commission4/proceedings/pdfpapers/301.pdf>.

[DUB]

Dublin Core Metadata Element Set, Version 1.1: Reference Description. Dublin Core Metadata Initiative <http://dublincore.org/documents/dces/>.

[GML2]

OpenGIS® Geography Markup Language (GML) Implementation Specification, version 2.1.2. Open GIS Consortium, Inc., 2003-17-09 (OpenGIS Project Document Number 02-069) <http://www.opengis.net/gml/02-069/GML2-12.html>.

[GML3]

OpenGIS® Geography Markup Language (GML) Implementation Specification, version 3.0. Open GIS Consortium, Inc., 2003-01-29 (OGC 02-023r4) <http://www.opengis.org/technology/documents/02-023r4.pdf>.

[HOR]

Erosional development of streams and their drainage basins. H. Horton, in Bulletin of the Geological Society of America, 56, 275-370, 1945.

[KRE]

On the Automated Generalization of Road Network Maps. Marc van Kreveld and Jarno Peschier, in Proceedings of the 3rd International Conference on GeoComputation, University of Bristol, United Kingdom, 17 - 19 September 1998 http://www.geocomputation.org/1998/21/gc_21.htm.

[MAC]

Detection and simplification of road junctions in automated map generalisation. W. A. Mackaness and G.A. Mackechnie, in Proceedings 18th International Cartographic Conference, Stockholm, Sweden, 1013-1021, 1997.

[MIS1]

The One Map Project. Gunnar Misund and Knut-Erik Johnsen. In online proceedings from GML Dev Days, Vancouver, July 2002 http://www.gmldev.org/GMLDev2002/presentations/MakingMapsfromGML/johnsen/one_map_misund_johnsen.html.

[MIS2]

The One Map Project. Gunnar Misund and Knut-Erik Johnsen. Presentation, GML Dev

<http://www.census.gov/geo/www/tiger/tiger2002/tgr2002.pdf>.

[TOP]

The TOP10NL demonstration Website. De Topografische Dienst http://kartoweb.itc.nl/top10nl/TOP10NL_eng/index2.htm.

[UCS]

2002 TIGER/Line® Files. U.S. Census Bureau <http://www.census.gov/geo/www/tiger/tiger2002/tgr2002.html>.

[WFS]

Web Feature Service Implementation Specification. Open GIS Consortium, Inc., 19-September-2002 (OGC 02-058) <http://www.opengis.org/techno/specs/02-058.pdf>.