Feature-driven generalisation of isobaths on nautical charts: a multi-agent system approach

Eric Guilbert

Department of Geomatics Sciences
Laval University, Québec QC, G1V 0A6 Canada
eric.guilbert@scg.ulaval.ca

Abstract

A nautical chart provides a schematic view of the seafloor where isobaths (contour lines joining points of same depth) and depth soundings are generalised to highlight undersea features that form navigational hazards and routes. Considering that the process is ultimately driven by features and their significance to navigation, this article proposes a generalisation strategy where isobath generalisation is controlled by undersea features directly. The seafloor is not perceived as a continuous depth field but as a set of discrete features composed by groups of isobaths. In this article, generalisation constraints and operators are defined at feature level and composed of constraints and operators applying to isobaths. In order to automate the process, a multi-agent system is designed where features are autonomous agents evaluating their environment in order to trigger operations. Interactions between agents are described and an example on a bathymetric database excerpt illustrates the feasibility of the approach.

1 Introduction

A nautical chart provides a simplified representation of the seafloor. The chart is used by navigators to plan their route and must ensure navigation safety. The main two elements depicting the bathymetry are isobaths and soundings. Navigators identify undersea features such as reefs and shoals as groups of isobaths and soundings on the chart. Isobaths provide general information about the shape of the features while soundings provide accurate information about high and low points. In comparison with topographic maps, nautical charts provide a more schematic representation of landforms: a highly detailed description of the seafloor is not required as it is not visible to the navigator. The objective being to provide information for navigation, shallow undersea features that mark navigation hazards are emphasised while deeper features may be ignored. The construction process is strongly constrained by standards defined by the International Hydrographic Organisation (IHO, 2009).

In contrast with other topographic maps which rely on mostly or fully automated processes, nautical chart generalisation automation is still in an early stage. Starting from the seafloor modelled by a set of soundings and isobaths extracted from the bathymetric database, the cartographer would work by selecting spot soundings and isobaths according to the relevance of the undersea features they model and so great prominence is given to lesser depths. As reported in (NOAA, 1997, p. 4-11), "[cartographers] do, deliberately and knowingly, and on behalf of the navigator, include all lesser depths within a contour even if it means that [their] catch includes many deep ones as well".

Isobaths on the chart are modified to emphasise characteristic features of the seafloor. An isobath modelling a hazard is kept and may be caricatured while an isobath marking a depression may simply be omitted if it is too small or does not mark a relevant location such as an anchorage or a fairway. Referring to Ruas and Plazanet (1997)'s classification, main constraints in nautical chart generalisation are categorised into (Guilbert and Zhang, 2012):

- The legibility constraint: generalised objects must be legible by observing a minimal size or distance between them;
- The functional constraint: related to navigation safety, the generalised representation of the relief must be higher than the original representation so that the depth interpreted from the chart cannot be greater than the real depth;
- The structural and shape constraints: morphological details of the seafloor (slope, roughness) should be maintained as far as possible. Characteristic features of the relief are emphasised;
- The topological and position constraints: spatial relationships, location and relative distances between objects must be maintained.

In cartography, generalisation is performed to select the amount of information according to the scale of the map (model generalisation) and to adapt the representation to the map purpose with consideration for map aesthetics and legibility (cartographic generalisation). Two approaches are considered for the generalisation of contour lines. As contours model a terrain, a first approach consists in generalising the original DTM or a DTM generated by contours and extracting the contours from the simplified terrain. The second approach simplifies the contour lines directly according to some geometrical criteria.

In comparison to common terrain generalisation approaches, seafloor generalisation is conducted with different consideration on the relief. The safety constraint means that any local deformation of the seafloor must be done upward only. Furthermore, in order to address relevance to navigation, undersea features must be identified and generalised according to their meaning.

Similarly to terrain contour generalisation, two approaches were developed for isobaths. In the first approach, the seafloor surface is generalised and isobaths are extracted from the simplified surface (Peters et al., 2014). This approach pertains to model generalisation as it is robust and fast and allows generating isobaths at various scales. In the second, isobaths are directly generalised through line deformation and selection (Guilbert and Saux, 2008). This approach is mostly appropriate for cartographic generalisation as it can consider legibility and take into account feature characteristics. However, as mentioned above, these features need to be identified and considered in the decision process to automate their generalisation.

This work pertains to the second approach and presents a model for selecting undersea features formed by groups of isobaths according to their relevance. Although delineating features with isobaths may be seen as a limitation, it provides robust identification of eminences and depressions and it corresponds to how features are perceived on the chart. Soundings are not included in the identification process, they can be considered when computing geometrical properties of the features. The contribution of this research is to present a model which extracts further topographic knowledge from the terrain model and uses it to drive the generalisation process. The method is based on a multi-agent system (MAS) where features are considered as autonomous agents which can choose and perform generalisation operations on themselves and on isobaths. Zhang and Guilbert (2011) introduced a MAS model structuring features and isobaths at different levels to perform feature selection. This work extends their model by describing and implementing the complete framework, performing not only selection but also feature deformation, and adding communication between agents to prioritise the actions and choose the operations.

This article first reviews existing works related to isobath generalisation and then presents the generalisation model (Section 3) and the multi-agent system (Section 4). Section 5 presents results obtained on a case study and the last section concludes and discusses perspectives.

2 Related works on isobath generalisation

2.1 DTM generalisation

Terrain generalisation methods rely mostly on global filtering methods and selection methods eliminating non relevant points of the DTM. Global filtering does not take any consideration for and therefore cannot integrate cartographic constraints. They are mostly appropriate for resampling. Selection methods usually preserve terrain features by preserving drainage lines or by selecting characteristic points. Both approaches rely on mathematical principles and are used to derive a smaller scale map (Weibel, 1992).

Most DTM generalisation methods do not apply to nautical chart generalisation as they do not observe the safety constraint. Peters et al. (2014) developed a simplification method where the surface is simplified by pushing points upward. A triangulated surface is first generated from the isobaths then for each point, if the depth interpolated from its neighbours is shallower than the point's depth, the computed depth replaces the point depth pushing the surface upward. The process can be repeated several times, each pass providing a higher representation of the seafloor. The method is robust and can handle both large and small scale changes. It also has the advantage of always yielding smoothed contours.

2.2 Contour line generalisation

Cartographic generalisation also requires the application of local operations to cartographic elements according to their meaning or the purpose of the map (Weibel, 1992). These operations depend on the information to emphasise. Contour line generalisation belongs to this group. Existing works define dedicated line generalisation methods to improve legibility and smoothness. Due to the safety constraint, specific methods are required for isobaths. Wang and Müller (1998) present a simplification method applicable to isobaths, but such method cannot be combined with other deformation operations and does not prevent topological conflicts with other lines (Dyken et al., 2009). Specific methods have also been developed for simplifying sets of contours (Li and Sui, 2000; Gökgöz, 2005; Matuk et al., 2006). Refering to the classification above, these methods can satisfy the legibility and topological constraints but do not handle the functional constraint.

A specific smoothing method respecting the safety constraint was developed by Guilbert and Lin (2006). The method is based on a snake modelwhere cartographic constraints are expressed by energies (Burghardt and Meier, 1997) and belongs to continuous optimisation approaches (Harrie and Weibel, 2007). Energies can be combined to perform both smoothing and displacement (Guilbert and Saux, 2008). Safety and distance conflicts are evaluated by defining an admissible zone where the isobath can lay. Energy terms are related to the line position and curvature. An optimal solution is obtained when the total energy of the system is minimised.

Guilbert and Saux (2008)'s work can perform automatic smoothing and displacement of individual isobaths satisfying legibility and safety. In comparison with Peters et al. (2014) which applies systematically to the whole model, the method can be applied to isobaths to improve the legibility locally by smoothing or deforming the lines and to emphasise some characteristics of the seafloor according to the cartographer's need. The method is semi-automatic and the user has to control the process by indicating which operation can be applied. Dealing only with isobaths, the system cannot always infer which side of the isobath is deeper (because surrounding isobaths are at the same depth for example). The user has to pass this information so that the system can choose an operation. Indeed, the choice of a generalisation operation does not depend on the isobath but mainly on the type of undersea feature (whether an eminence or a depression) the isobath belongs to. The objective is not to generalise each isobath but to generalise the features they model. Therefore further automation of the process requires the development of a strategy centred on undersea features where both constraints and operators are defined at feature level so that generalisation can be conducted in a systematic way.

2.3 Generalisation strategy

An overall generalisation strategy for chart bathymetry is discussed by Tsoulos and Stefanakis (1997). An expert system is designed and operations are performed in a sequence so that the result depends on their order. It also requires the definition of complex rules to handle interactions between elements, leading to a highly complex system. This issue has been indeed observed more generally for map generalisation in (Harrie and Weibel, 2007). A different approach, based on a multi-agent system, was proposed by Gaffuri (2008) for terrain generalisation. Agents represent elements of the TIN including nodes and edges. Each agent can perform a generalisation operation chosen from a plan of action, making use of information from other agents or the environment. If the action presents an improvement of the representation, the result is kept. The method is applied mostly for cartographic generalisation and interaction between terrain data and other map objects, such as rivers, roads or buildings.

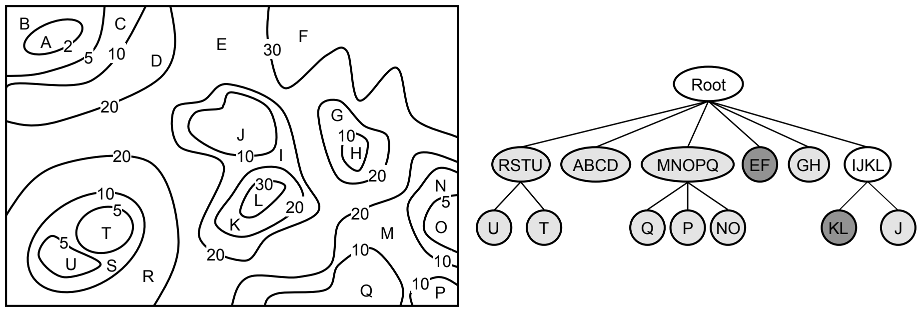
Zhang and Guilbert (2011) proposed a model based on a MAS that is applicable to isobath selection. Undersea features are considered as agents formed by sets of isobaths and features were selected based on contextual information (type of feature, distance between isobaths, feature area). Other operations (deformation, aggregation) were not handled. Topological relationships between contours are expressed in a contour tree from which depression and eminence features are identified. However the contour tree cannot deal correctly with open contours and with the characterisation of terrain features as objects structuring the terrain. An improvement was brought by Guilbert (2013) who defines a hierarchical data structure, the feature tree. The structure models topological relationships between features which are formed by groups of open and closed contours.

A further interest for working with isobaths rather than the DTM is that features on the chart are portrayed by groups of isobaths. Hence features defined in the feature tree correspond to what is perceived on the chart. This article builds up from these contributions and proposes a multi-agent system where undersea features are active communicating agents which can select and apply different types of operators. The next section describes the methods for modelling features. It makes use of the feature tree for feature classification (Section 3.1) and provides a set of constraints (Section 3.2) and operations (Section 3.3) defined on features. The multi-agent system built around features is described in Section 4.

3 Undersea feature modelling

3.1 Undersea feature classification

In order to extract knowledge about undersea features, they must be identified and classified from the set of isobaths. Safety constraint requires mainly that features are classified as depressions (features lower than their surroundings) and eminences (higher than their surrounding). The feature tree (Guilbert, 2013)



(a) Isobath map (b) Feature tree

Figure 1: (a) Isobath map: numbers indicate depth, letters are inter-contour regions; (b) feature tree obtained by aggregating regions: eminences (light grey), depressions (dark grey) and unclassified features (white). Modified from (Guilbert, 2013).

provides a hierarchical structure based on inclusion where features are groups of isobaths identified as eminences and depressions. An eminence is a feature whose inner isobaths are strictly above the isobaths forming the boundary, a depression is a feature whose inner isobaths are strictly deeper than the bounding isobaths (Figure 1). If a feature is defined by only one isobath, its class is inferred by neighbouring features or features at the level above in the tree. Features that are neither depressions nor eminences remain undefined and are classified as such. The tree is built from a contour graph and can deal with open and closed contours indifferently. Each feature is an object on its own composed by a set of isobaths. It has its own attributes defining feature properties which are its type (eminence or depression), its area and its height.

Features are used to drive the generalisation process and ensure the safety constraint is satisfied. Instead of considering the seafloor as a field (i.e. with depth as a function of the position), the seafloor is represented by a set of undersea features. Each feature is an individual object with an exact location defined by its boundary contours which can be stored explicitly to enrich the bathymetric database. Hence, even if features are not elements drawn on the chart, they are objects holding the topographical knowledge and they control the isobaths and soundings they are composed of. The generalisation process is driven by assessing constraints and applying operators on features.

3.2 Generalisation constraints

Generalisation constraints are evaluated both to decide if an operation is needed and to evaluate the result of an operation. As features are defined as objects, they have their own constraints to satisfy. They correspond mainly to global criteria related to the characterisation of undersea features portrayed on the chart, assessing if a feature shall be preserved, modified or omitted. They take

Safety	The depth at any position on a feature cannot be deeper than the original depth
	An eminence cannot be removed
Legibility	A minimum distance must be observed between features
	A feature must be large enough to contain a sounding marking its lowest or highest point
Shape	The overall morphometry of the feature shall be preserved
Topology	The topology defined by the feature tree must remain consistent
Position	The outline of a feature shall be preserved

Table 1: List of constraints applicable to features.

into account feature attributes and relationships between features.

Constraints applying to features are classified in Table 1. Constraints related to safety and legibility are strong constraints which cannot be violated. Both shape and position constraints are weak constraints which are measured to estimate the quality of a generalisation. The topological constraint is a preservation constraint to guarantee that no inconsistency occurs in the process.

Constraints on isobaths apply at a more local level, considering the geometry and the spatial relationships of individual isobaths defined in the contour graph. Constraints are listed in Table 2. As isobaths are components of features, constraints on isobaths are used to define constraints on features. For example, if an isobath violates safety then all features containing this isobath also violate the safety constraint and the distance between two neighbouring features is evaluated by assessing the distance between their enclosing isobaths. As for features, they can be classified as strong (safety and legibility) and weak (shape and position). Legibility, shape and position constraints can be evaluated locally as only the isobath and its adjacent isobaths are needed. Topological constraints are also constraints assessed during an operation to preserve from any topological inconsistency.

3.3 Generalisation operators

3.3.1 Feature operators

Similarly a set of generalisation operators are defined at both feature and isobath levels. As mentioned earlier, features model global information about the seafloor and are defined by sets of isobaths. Therefore, operations applied to features trigger operations on isobaths and modify the structure of the terrain. Both the feature tree and the contour tree may be modified as isobaths and features can be deleted.

The simplest operation is the deletion of a feature. It consists in deleting all the isobaths contained by the feature (including the boundary). If the feature has some descendants in the feature tree, they are also deleted. Deletion is performed when a feature is too small or not important enough to be kept. Due to the safety constraint, this operation can apply only to depressions. For

Safety	An isobath cannot be pushed towards shallower depth
	Removing an isobath cannot result in a deeper seafloor representation
Legibility	A minimum distance must be observed between two isobaths or between two disjoint segments of an isobath
	The area enclosed by an isobath must be large enough to contain at least one sounding
Shape	The shape of the isobath shall be preserved
Topology	The topology defined by the contour graph must remain consistent
Position	An isobath shall remain close to its original position
	Distance between isobaths shall be maintained as much as possible

Table 2: List of constraints applicable to isobaths.

example, in Figure 1, eminence J cannot be removed but depression KL can be removed. In that case, feature IJKL which was undefined becomes feature IJ and is an eminence. Similarly, feature IJKL cannot be removed directly because removing it would mean that the area is below 20 m depth while the area within feature J is above 10 m.

Deformation consists in modifying the area of a feature by displacing its boundary. It applies to both depressions and eminences to correct legibility conflicts between two features. If an eminence feature has a legibility conflict with its descendant, the isobaths in the feature are pushed outward to make more room between the isobath and the descendant feature. Deformation of a depression is done by pulling the isobath inward and it applies when a feature is in conflict with an adjacent feature or with its parent feature.

Features can also be aggregated together to form a new larger feature. The operation is performed by merging their two boundary isobaths into one, which becomes the boundary of the new feature. It imposes therefore that both isobaths are at the same depth. Aggregation is performed on features which are too close and cannot be removed or to highlight a part of the seafloor by grouping several features together. Aggregation can lead to the removal or the modification of original features. For example, older features may no longer exist after aggregation (Figure 2, left). If the new feature is of the same type as its ascendant, it is redundant and not added to the feature tree (Figure 2, right). Because of safety, aggregation applies to eminences only: if applied to depressions, the area between the two depressions would be part of the aggregated depression and would then appear deeper.

Finally, caricature consists in preserving the most important characteristic of a feature and removing other details. In nautical chart generalisation, it applies to eminences which are not large enough to contain a sounding marking their highest point. The feature boundary is enlarged to a large enough area. If it contains other isobaths, they are removed and the boundary isobath is assigned

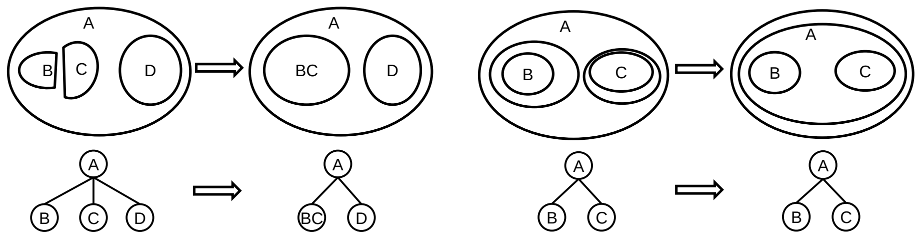


Figure 2: Aggregation of two features. Left, original features B and C are removed to form a new feature BC; right, feature tree structure remains but feature boundaries are modified.

Name	*Description*	*Apply to*
Deletion	All isobaths included in the feature are deleted	Depression
Deformation	Isobaths are pushed away from conflicting features	Both
Aggregation	Two features are aggregated by merging their boundary isobaths forming one new feature	Eminence
Caricature	Area of the boundary isobath is enlarged	Eminence

Table 3: List of operators on features.

the depth of the highest isobath. However this operation must be done only once no other conflict occurs between the feature and its neighbours as the isobath cannot be aggregated anymore. The list of feature operators is summarised in Table 3.

3.3.2 Isobath operators

Operations on isobaths are of two kinds: they can be local, not modifying the information at feature level, or be part of an operation performed on a feature modifying the topology. In the first category are the smoothing and the displacement of an isobath. These operators are applied to improve the legibility and aesthetic of the map and do not alter the information conveyed in the feature tree. Enlargement is applied when caricaturing a feature. It is in that case triggered by the feature but it does not modify the structure of the contour and feature tree. It can also be applied at isobath level if an inner isobath is not large enough. In a caricature, the depth attribute can be modified and set to the depth of the shallowest isobath which is removed. The last two operations, deletion and merge, are performed only as parts of feature operations. They cannot be triggered directly at isobath level as they are parts of operations modifying the feature structure. Deletion can be called by any of the feature operators: deleting a feature requires the deletion of its isobaths, merging features imposes the deletion of boundary isobaths and caricature may delete too small isobaths. Merge is only called by the aggregation operator and

Name	*Description*	*Level*
Smoothing	The isobath is smoothed towards greater depth	Isobath
Displacement	The isobath is pushed away from another isobath	Both
Enlargement	Area of the isobath is enlarged to a minimum size	Both
Deletion	The isobath is deleted	Feature
Merge	Two neighbouring isobaths are merged into one new isobath	Feature
Depth change	The isobath depth is changed to a smaller value	Feature

Table 4: List of operators on isobaths. Last column indicates at which level the operator is called.

consists in creating one isobath from two adjacent isobaths. The list of isobath operators is summarised in Table 4.

Operations can also be decomposed into two groups of operation: discrete operations (deletion and creation of contours and features, depth attribution, merge) and continuous operations (enlargement, smoothing, displacement). Discrete operations are triggered at feature level only because they are parts of operations which modify the topology while continuous operations are performed at isobath level as they modify the geometry but not the topology.

3.4 The snake model

Continuous operations integrate constraints of Table 2 and are defined using a continuous optimisation approach, the snake (or active contour) model (Harrie and Weibel, 2007). A snake is a parametric function $(x, y) = f(t)$ which has its own energy related to its position. As the snake is in a stable position when its energy is minimal, it will deform itself in order to minimise its energy and reach an equilibrium. An isobath is considered a snake f whose total energy $E(f)$ is composed of an internal energy $E_{\texttt{int}}$ related to the shape of the line and an external energy $E_{\texttt{ext}}$ defined by cartographic constraints that apply on the line, upsetting the initial balance of energy (equation 1 where s is the curvilinear abscissa and l the length of the line). If a cartographic constraint is violated, the external energy is high and the line is deformed to bring down the external energy to zero and minimise the total energy of the system, providing a new equilibrium. Different operations can be performed together by defining different energy terms (Guilbert and Saux, 2008).

$$E(f) = \int_0^l [E_{\texttt{int}}(s) + E_{\texttt{ext}}(s)]ds \tag{1}$$

The internal energy is defined by the derivative of the isobath itself (equation 2), so that reducing the internal energy tends to smooth the isobath. Parameters α and β control the tension and curvature of the snake. As smoothness is characterised by the curvature at any point of the snake, α is set to zero and β

is set to a constant.

$$E_{\texttt{int}}(s) = \frac{1}{2}\left(\alpha(s)\left|f'(s)\right|^2 + \beta(s)\left|f''(s)\right|^2\right) \tag{2}$$

Different terms of external energy are defined, each of them corresponding to different constraints. The first constraint of area indicates that an isobath must contain at least one sounding o and so that any point of the isobath must be at a distance $\varepsilon_{\texttt{radius}}$ from the sounding. If a point is closer than this threshold, a penalty in proportion to the violation is assigned (equation 3). Legibility also requires a minimum distance $\varepsilon_{\texttt{dist}}$ to be maintained between lines. Therefore, an external energy is added to points of an isobath which are too close from another line and need to be displaced (equation 4). The energy is a function of the distance $d(f(s), g)$ between the point $f(s)$ and the adjacent isobath g.

$$E_{\texttt{area}}(s) = \begin{cases} \varepsilon_{\texttt{radius}}^2 - ||f(s) - o||^2 & \text{if } ||f(s) - o|| < \varepsilon_{\texttt{radius}} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$E_{\texttt{dist}}(s) = \begin{cases} \varepsilon_{\texttt{dist}}^2 - d(f(s), g)^2 & \text{if } d(f(s), g) < \varepsilon_{\texttt{dist}} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

As the position of the isobath needs to be checked against all its relatives, i.e. adjacent, parent and descendant isobaths from the contour graph, and against the circle C of centre o and radius $\varepsilon_{\texttt{radius}}$, instead of checking the position against each neighbouring isobath one by one during the deformation, an admissible area A_a combining all constraints in which the snake can move is defined before the start of the deformation. The definition depends on the direction of the deformation. If the isobath has to be pushed towards its parent, A_a is bounded by the original position of the isobath (as it must be pushed towards one side only) and by its parent and adjacent isobaths forming a polygon. If the isobath is to be pulled inward, A_a is defined by inner isobaths. If the snake moves outside A_a, topology or safety is violated and the solution is not valid. Further, a second polygon, defining the free energy area is used. This polygon is defined by an offset of $\varepsilon_{\texttt{dist}}$ of all relative isobaths and by C. It represents the area where the external energy is zero. Parts of the snake outside this area are in conflict and so have a positive external energy.

3.5 Merging

Finally, merging two isobaths f and g is done by filling in the space between them. The maximum distance that is filled in between the curves is a parameter noted $\varepsilon_{\texttt{aggreg}}$. Here also, an admissible area A_a is defined by the parent isobath and other adjacent features. Considering two polygons F and G delineated by f and g and noting

$$P_F = \{p \in F : d(p, g) < \varepsilon_{\texttt{aggreg}}\}$$

$$P_G = \{p \in G : d(p, f) < \varepsilon_{\texttt{aggreg}}\}$$

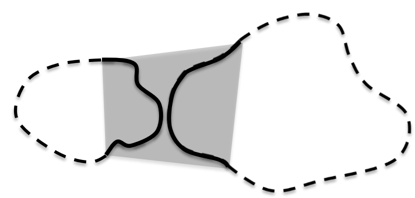


Figure 3: Merging two isobaths. Dotted lines: isobaths f and g, thick lines: P_F and P_G, grey shape: $\mathrm{Conv}(P_F \cup P_G)$.

the merged isobath h is defined as the boundary of the polygon

$$H = (F \cup \mathrm{Conv}(P_F \cup P_G) \cup G) \cap A_a \tag{5}$$

where Conv is the convex hull of a set of points (Figure 3).

The different parameters $\varepsilon_{\mathtt{radius}}$, $\varepsilon_{\mathtt{dist}}$ and $\varepsilon_{\mathtt{aggreg}}$ are related to the legibility constraints. The first two define the minimum size and the minimum distance between two isobaths. Aggregation also relates to legibility but the third parameter is used to control the process. In order to be effective, $\varepsilon_{\mathtt{aggreg}}$ must be bigger than $\varepsilon_{\mathtt{dist}}$.

4 The multi-agent model

4.1 Modelling with agents

The generalisation process can be modelled by a multi-agent system where cartographic objects are seen as autonomous agents which can make their own decisions and trigger generalisation operations. Three main generalisation models were developed. Ruas (1999) first proposed the AGENT model where three levels of agents are defined: micro agents representing individual objects, meso agents formed by groups of objects and macro agents controlling a population of agents. An agent can evaluate its own constraints and decide if it needs to generalise itself. Meso agents are responsible for what is done within a group and can eliminate individual objects.

The CartACom model (Duchêne, 2004) defines relational constraints as objects associated to agents so that one constraint can be shared between two agents. It also allows communication between agents so that an agent can perform an operation following another agent's demand. The GAEL model (Gaffuri, 2008) was developed specifically to handle field variables such as elevation. The field agent is decomposed into points, edges and triangles, on which constraints are defined, adding a sub-micro level. The field agent can interact with other agents which are vector objects. Therefore, interactions can be hierarchical as in AGENT and transversal as in CartACom.

As discussed by Duchêne and Gaffuri (2008), each model has its own area of application. AGENT is based on hierarchical interactions between single objects (micro agents) and groups of objects (meso agents). CartACom focuses instead on transversal interactions between single objects allowing them to communicate. Finally, GAEL focuses on field variables and their relationships with objects.

This research is concerned with the seafloor representation which is not modelled as a field but by undersea features composed of isobaths. As the objective is to select these features based on their characteristics, the approach is centred on features which are able to evaluate their environment and perform operations. Isobaths are components which are controlled by features and act upon their request. As in AGENT, a third type of agent, the controller, is required to set priorities between different features to avoid inconsistent operations.

4.2 The isobath agent

Isobath agents are reactive agents which act only when triggered by features. They can perform two actions: evaluate their environment and perform a generalisation operation. In both cases, features pass them information about their environment. Evaluation is done by estimating legibility constraints of Table 2 and returns a value to the feature such as an area or a distance. Isobaths do not make any decision whether a conflict occurs. The feature makes a decision upon the different feedbacks it gets and may ask the isobath to perform an operation. Only continuous deformations presented in 3.3.2 are performed by isobaths.

4.3 The controller

The controller is an agent which can communicate with feature agents. As it operates at a global level, it can hold metadata information about the map including threshold values for constraints. It is in charge of controlling the generalisation process by setting rules and priorities between the features. Its main role is to avoid inconsistent operations that may occur when features are performing concurrent operations. Each time a feature needs to perform a generalisation operation, the feature needs to ask a permission to the controller. The controller grants permission to the feature if no relative feature (parent, child and adjacent feature from the feature tree) is performing an operation or an evaluation. Once permission is given, all relative features are locked by the controller and they wait for the operation to be over. Once the operation is completed, the controller releases the lock and asks the features to start again the evaluation process.

The controller maintains a priority queue so that permission among neighbours is given to the feature with the highest priority. Priority is set based on conflict information communicated by each feature. It is decided according to the type of feature, the type of conflict and also based on the success of the last operation. Priorities make the system more efficient by limiting cascading effects and avoid dead locks where the same operations may repeat. Operations on depressions are given precedence because they have less impact on their neighbourhood as a depression cannot be enlarged. Also, if a feature performs an operation but fails to improve its situation, its priority is lowered so that other conflicts in the vicinity are corrected first, offering new solutions.

Finally, the controller assesses if the whole process is completed by checking if any feature still requests an operation. If no, it means that no more conflict occurs and the controller sends a message asking all feature agents to take down.

4.4 The feature agent

4.4.1 The feature life cycle

The whole generalisation process is driven by the features which are the objects making the decisions. Features are active agents able to communicate with their inner isobaths in order to evaluate their state and decide upon further actions, and with the controller to ask permissions and report their actions. Each feature agent goes through a series of behaviours representing its life cycle shown in Figure 4. The four following behaviours are considered:

- *Evaluating*: the feature evaluates the existence of conflicts by checking its area and its distance with other features;
- *Waiting*: if an operation must be done, the feature needs to request permission from the controller and waits for an answer.
- *Performing* an operation: once permission has been granted, the generalisation operation starts up. The feature checks a set of plans and performs the best one;
- *Listening*: no conflict has been detected and so no action is taken. The feature does not perform any action and waits for a message from the controller. This message can ask to start again the cycle and re*evaluate* their constraints if a modification happened around or the controller can ask the feature to be taken down when the whole process is over.

During its *evaluation*, the feature asks its isobaths to check for distance or area conflicts. At the end of the process, each conflict is described by the conflict type, the type of feature involved and their relationship (adjacent, parent or descendant). If no conflict occurs the feature passes directly in *listening* behaviour as no action is required; otherwise an operation must be done and the feature sends a request to the controller and *waits* for an answer.

When *waiting*, the feature can receive three kinds of answer from the controller:

- Permission is rejected because a relative feature with higher priority is already waiting for a permission. The feature does not resubmit its request but keeps *waiting* until it is told by the controller to reevaluate its environment. Once it receives such message, the feature starts again its *evaluation*;

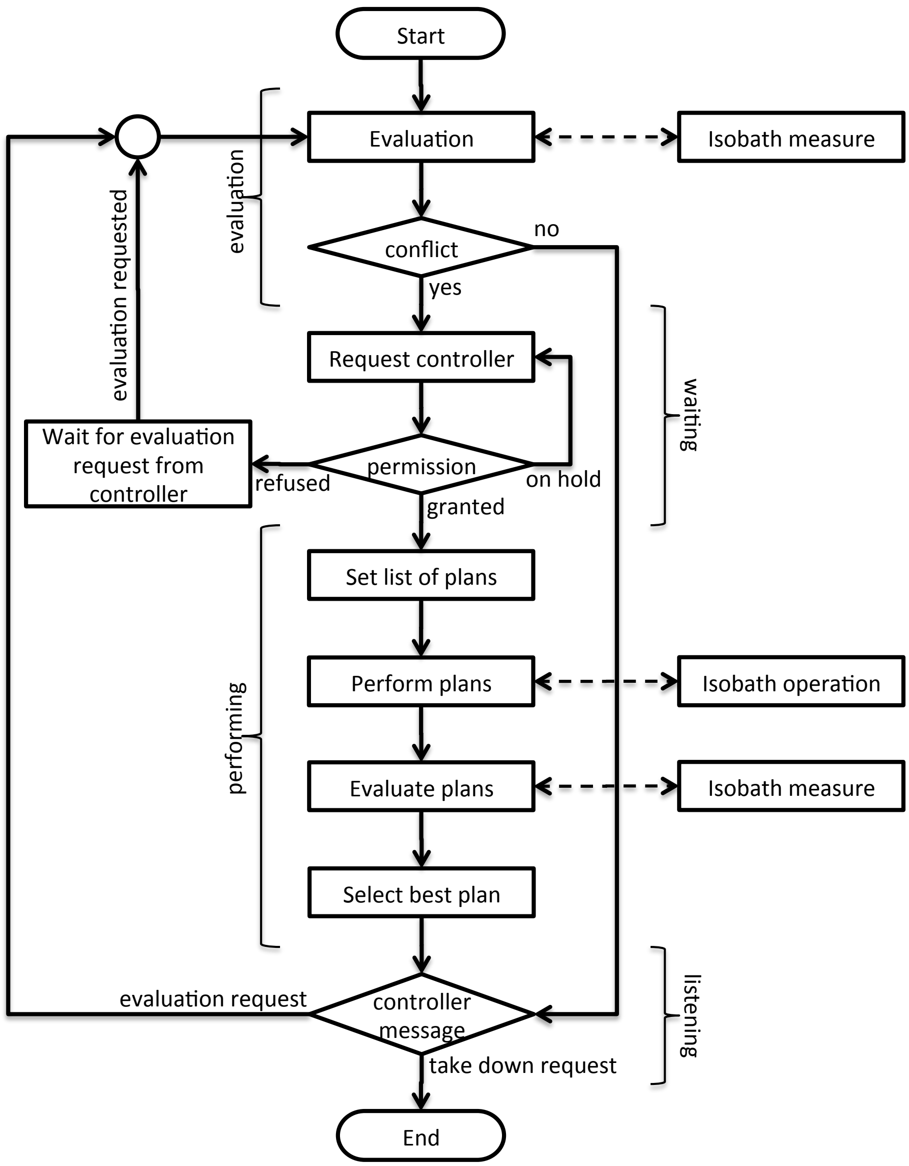


Figure 4: Life cycle of a feature agent including its behaviour and interactions with controller and isobath agents.

- Permission is upheld: the operation cannot be done yet because relative features have not finished their evaluation. The feature keeps *waiting* and resubmits its request. The permission may be rejected later if a relative feature submits a request with a higher priority;

Conflict	*Feature type*	*Relationship*	*Action*
Area	Depression	-	Deletion
Distance	Depression	Adjacent	Deformation, deletion
Distance	Depression	Parent	Deformation, deletion
Distance	Depression	Descendant	-
Area	Eminence	-	Caricature, aggregation
Distance	Eminence	Adjacent	Aggregation
Distance	Eminence	Parent	-
Distance	Eminence	Descendant	Deformation

Table 5: List of possible actions for each type of conflict and feature with the relationship of the other conflicting feature.

- Permission is granted. All other evaluations among relatives have been completed and no operation at a higher priority has been requested. The operation can be *performed*.

4.4.2 Plans of action

Performing an operation goes through three steps. First, a list of actions applicable to the conflict is set following the list of Table 3. The set of operations is limited by the type of the feature and by the relationship between conflicting features. The list of actions is summarised in Table 5. When a feature is conflicting with its parent or descendant, only one feature takes action with respect to the safety constraint. For example, if a depression is conflicting with its descendant, it cannot be deformed as it would enlarge the feature and violate the safety.

Second, each operation is performed in parallel. Continuous operations require the application of isobath operators (Section 3.3.2). All three operations (smoothing, displacement and enlargement) are performed in the same snake model with the feature passing the list of isobaths and safety information required for the computation of the admissible area and energy terms. Result of each operation is evaluated according to the constraints of Table 1. If an operation violates safety or topology, it is rejected. Results are then ranked according to their legibility by measuring how close to the threshold the feature is, and finally by checking how well they preserve their shape and position. This is done by measuring the variation of area. In the case of an aggregation, the difference between the area of the new feature and the total area of older features is taken.

Third, the best result is selected and if needed, the data structure is updated. The feature indicates to the controller whether it managed to improve its situation. If not, the controller lowers the feature priority in the queue. Once the process, is over, the controller releases the lock on the feature relatives so that they can restart their evaluation to take into account the change.

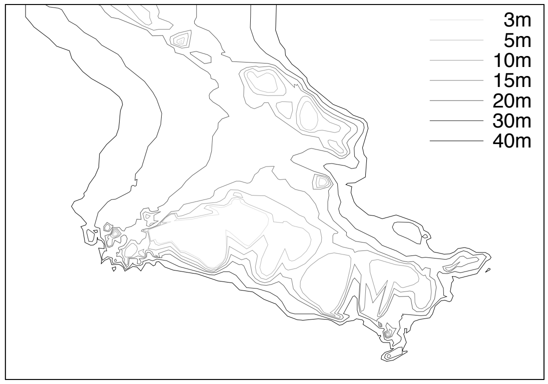


Figure 5: Initial map.

5 Results

The multi-agent system was implemented in Java with the Jade platform[1] and tested on a set of isobaths provided by the French Hydrographic and Oceanographic Service (SHOM). Isobaths have been extracted from the bathymetric database at 1:50,000 but no cartographic generalisation was performed. Figure 5 shows the excerpt at real map size. As soundings were not available, the spot sounding for enlargement was considered to be in the centre of the feature. Parameters set by the user are the three parameters required for isobath operations $\varepsilon_{\texttt{radius}}$, $\varepsilon_{\texttt{dist}}$ set respectively to 2 mm and 0.5 mm to satisfy legibility and $\varepsilon_{\texttt{aggreg}}$ set to 1 mm. The result is presented on Figure 6 and was obtained automatically.

Figure 7 further illustrates the process on features on the far left of Figure 5. At the beginning, each feature agent evaluates its constraints. In the bottom left portion of the original map, several features are in conflict. On the top left, the depression in the centre of the figure is too small and too close to the eminences. The eminence on the left is also too small. 30 m isobaths which bound the eminences are also too close from the 40 m isobaths which belong to their parent feature. Both eminences and the depression request permission to the controller which prioritises them. Depression has higher priority and is deleted (top right). The controller asks then adjacent eminences to reevaluate their constraints. There is no distance conflict but the left eminence is too small and performs an enlargement which is constrained by the parent and adjacent eminence. Finally, the enlargement creates a legibility conflict and both eminences are merged. At the end of the whole process, the isobaths will be further pushed away to allow enlargement of the higher eminences (Figure 6).

[1]http://jade.tilab.com/

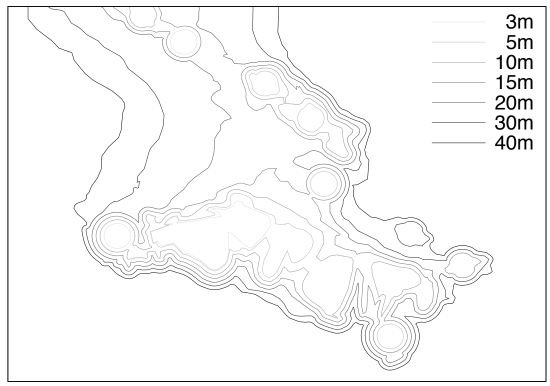


Figure 6: Generalised map.

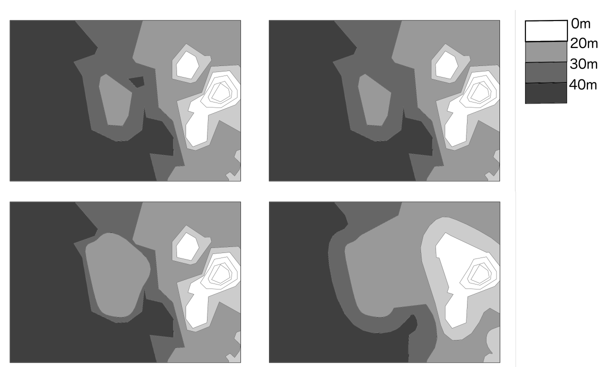


Figure 7: Illustration of the generalisation process on a small area. From top left to bottom right: depression removal, eminence enlargement and aggregation.

Overall, constraints set at the beginning are respected. About 300 operations were performed in total, most of them in the lower part of the map where isobaths were too close and progressively pushed down. In most cases, lists of plans set by the feature agent consisted of one action. In the case both aggregation and caricature could be performed, caricature lead to a bigger deformation and aggregation was chosen as it gave better results.

Although the result satisfies the legibility constraint, it leads to too large a deformation with many isobaths gathered in the lower part of the chart. As isobaths are all at regular interval, the slope appears regular and more gentle than what it was, modifying the morphology of the seafloor. A cartographer would have instead have removed some isobath segments to avoid too large displacements. The result could therefore be improved by extending the definition of features to deal with segmented contours and adding a segment removal operator.

6 Conclusion and future works

This model demonstrates the interest of an object representation of the terrain in designing a feature-driven generalisation process. Although a limitation, the characterisation of features from isobaths alone has the benefit that it provides a robust definition of features that matches with their perception on the chart. It allows for a qualitative approach where undersea features can be stored in the bathymetric database and can be used for reasoning. A set of operators were defined at feature level and the generalisation process was coordinated by a multi-agent system. Results respect legibility and safety constraints but operators may be added to better preserve the slope and the morphology in general. The limited choice of operators lead to too large displacements in cluttered areas to yield an acceptable result for deployment in applications.

So far, isobaths are defined as reactive agents because only constraints at feature level have been considered. More local constraints specific to isobaths such as the smoothness could be evaluated at the isobath level so that isobaths could perform their own evaluation and trigger operations or potentially request actions at the feature level.

In this case study, one controller was set up to manage all features and the computation time was within three minutes. In the case of a larger map, performances may be slowed down because of the much larger number of objects to handle. A solution to facilitate parallel processing is to define several controllers in charge of separate areas of the map. A hierarchical partitioning is already defined by features so that controllers could handle independent processes at separate levels.

The next step following this work is adding soundings in the model as other feature components. That will provide a more accurate representation of the seafloor and allow generalisation of both soundings and isobaths at the same time through the generalisation of features. Feature operations will be extended to trigger sounding selection together with isobath generalisation.

By adding soundings, the seafloor is no longer described by a network of isobaths but by a surface model formed by soundings and isobaths. Isobaths can still be used to delineate features but other seafloor elements can be added such as reefs or seamounts marked by isolated soundings or the bottom of a fairway marked by a series of soundings extending the model by adding other types of features, leading to a better consideration of the bathymetry and navigation rules.

Beside generalisation, the use of a hierarchical data structure classifying features on the surface model can apply to other domains in oceanography and the classification of features into more specialised concepts such as reefs, shoals or canyons. Such a description would require the definition of formal feature properties that can be computed from the terrain model. This formalisation can be achieved through the development of an ontology which can structure the knowledge and assist in the classification (Yan et al., 2014).

References

D. Burghardt and S. Meier. Cartographic displacement using the snakes concept. In W. Förstner and L. Plümer, editors, *Semantic modeling for the acquisition of topographic information from images and maps*, pages 59–71. Birkhauser Verlag, 1997.

C. Duchêne. *Généralisation cartographique par agents communicants: le modèle CartACom*. PhD thesis, Université Paris 6, 2004.

C. Duchêne and J. Gaffuri. Combining three multi-agent based generalisation models: AGENT, CartACom and GAEL. In A. Ruas and C. Gold, editors, *Headway in Spatial Data Handling, 13th international symposium on Spatial Data Handling, 23–25 June, Montpellier, France*, pages 277–296, Berlin, 2008. Springer.

C. Dyken, M. Dæhlen, and T. Sevaldrud. Simultaneous curve simplification. *Journal of Geographic Systems*, 11:273–289, 2009.

J. Gaffuri. *Généralisation automatique pour la prise en compte de thèmes champs: le modèle GAEL*. PhD thesis, Université Paris-Est, 2008.

T. Gökgöz. Generalization of contours using deviation angles and error bands. *The Cartographic Journal*, 42(2):145–156, 2005.

E. Guilbert. Multi-level representation of terrain features on a contour map. *Geoinformatica*, 17(2):301–324, 2013. doi: 10.1007/s10707-012-0153-z.

E. Guilbert and H. Lin. B-spline curve smoothing under position constraints for line generalisation. In S. Nittel and R. De By, editors, *ACM GIS'06*, pages 3–10, New York, 2006. ACM press.

E. Guilbert and E. Saux. Cartographic generalisation of lines based on a B-spline snake model. *International Journal of Geographical Information Science*, 22 (8):847–870, 2008.

E. Guilbert and X. Zhang. Generalisation of submarine features on nautical charts. In M. Shortis, J. Shi, and E. Guilbert, editors, *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume I-2, pages 13–18, 2012. doi: 10.5194/isprsannals-I-2-13-2012.

L. Harrie and R. Weibel. Modelling the overall process of generalisation. In W. Mackaness, A. Ruas, and L. Sarjakoski, editors, *Generalisation of Geographic Information: Cartographic Modelling and Applications*, pages 67–87. Elsevier Science, Amsterdam, 2007.

IHO. *Chart Specifications of the IHO and Regulations for International Charts*. International Hydrographic Organization, 3.007 edition, 2009.

Z. Li and H. Sui. An integrated technique for automated generalization of contour maps. *The Cartographic Journal*, 37(1):29–37, 2000.

K. Matuk, C. Gold, and Z. Li. Skeleton based contour line generalization. In A. Riedl, W. Kainz, and G. A. Elmes, editors, *Progress in Spatial Data Handling*, pages 643–658. Springer Berlin Heidelberg, 2006.

NOAA. *Nautical Chart User's Manual*. National Oceanic and Atmospheric Administration, U.S. Department of Commerce, 1997.

R. Peters, H. Ledoux, and M. Meijers. A voronoi-based approach generating depth-contours for hydrographic charts. *Marine Geodesy*, 37(2):145–166, 2014.

A. Ruas. *Modèle de généralisation de données géographiques à base de contraintes et d'autonomie*. PhD thesis, Université de Marne-la-Vallée, 1999.

A. Ruas and C. Plazanet. Strategies for automated generalization. In M. Kraak, M. Molenaar, and E. Fendel, editors, *Advances in GIS research II, Proceedings of the 7th International Symposium on Spatial Data Handling*, pages 319–336, London, 1997. Taylor & Francis.

L. Tsoulos and K. Stefanakis. Sounding selection for nautical charts: an expert system approach. In *Proceedings of the 18th International Cartographic Conference*, pages 2021–2029, 1997.

Z. Wang and J.-C. Müller. Line generalization based on analysis of shape characteristics. *Cartography and Geographic Information Systems*, 25(1):3–15, 1998.

R. Weibel. Models and experiments for adaptive computer-assisted terrain generalization. *Cartography and Geographic Information Systems*, 19(3):133–153, 1992.

J. Yan, E. Guilbert, and E. Saux. An ontology for submarine feature representation on charts. In J. Parsons and D. Chiu, editors, *Advances in conceptual modeling*, volume 8697 of *Lecture notes in computer science*, pages 91–100. Springer, 2014.

X. Zhang and E. Guilbert. A multi-agent system approach for feature-driven generalization of isobathymetric line. In A. Ruas, editor, *Advances in Cartography and GIScience. Selection from ICC 2011, Paris*, volume 1, pages 477–495. Springer, 2011.