Project Number	**ESPRIT / LTR /24 939**
Project Title	**AGENT**
Deliverable Type	**Report**

Deliverable Number	**D D3**
Internal ID	
Contractual Date of Delivery	
Actual Date of Delivery<	
Version	**1.0**
Workpackage/Task contributing	**D2**
Authors	**UNI-ED, IGN, UNIZH**
Confidentiality	**Public**

Title of Deliverable	**Strategic Algorithms Using Organisations**

Abstract

The reports discusses operators and algorithms which require a consideration of contextual information for their operation and which are utilised by the organisational (meso) level to orchestrate and influence the activities of individual micro agents.

Keyword List

Generalisation algorithms, Meso-agent organisations, contextual information

1 Executive Summary

This report describes the research conducted for the D2 task of work package D.

The objectives of this report follow those itemised in the AGENT Technical Annex: WP D.2 Strategic algorithms using organisations (p.49). As such, the reports discusses operators and algorithms which require consideration of contextual information for their operation and which are utilised at the organisational (meso) level to orchestrate and influence the activities of individual agents in a directed manner. The aim of this analysis is to identify and develop a set of algorithms "able to utilise information provided at the organisational level" and satisfy the goals of a meso agent as described in task A2.

The report follows the terminology and framework of previous research on cartographic algorithms, principally that described in WP D D1: Selection of basic algorithms. Within this framework, it introduces the notions of contextual information and algorithmic strategy, providing an analysis of this terminology and how it relates to research in this work-package. It defines the role of contextual algorithms within the AGENT system as providing procedures to allow for the visual re-organisation of information in such a way as to communicate the aspects of the space that are salient to the information at a different scale or for a specific map theme.

In the second section of the report detailed analysis of contextual operators and algorithms is presented. Four operators are discussed; Selection, Displacement and Aggregation (Amalgamation and Typification). The requirement for selection operators for meso-organisations is twofold. To simplify data to reduce visual complexity, whilst retaining the information most pertinent to the scale of the intended map, without any cost in locational accuracy. To identify and generate other meso-organisation for both better characterisation of the geographic information and to reduce processing complexity by a *divide and conquer* approach. The operation of displacement is used to remedy problems of conflicting symbology and violation of proximity constraints relating to the perceptibility of symbology. Aggregation is performed to simplify data in areas of high densities whilst maintaining the overall gestaltic impression of density of information across the intended map. The requirement of typification is to reduce data whilst maintaining the impression of density of information in areas of medium density and maintaining the impression of the structure of the original information.

The final section of the report provides a scoping of the needs of the project with respect to contextual algorithms. It makes recommendations for implementation of specific algorithms with respect to the organisations identified in WP A A4: Geographic Object Modeling.

Authors

Alistair Edwardes (Principal Author); Introduction, Aggregation, Amalgamation, Selection of Algorithms for Organisations.

Emmanuel Fritsch; Selection.

Mats Bader; Displacement

Jean-Francois Hangouet; Typification

2 Introduction

2.1 Objectives of the report

The objectives of this report follows those itemised in the AGENT Technical Annex: WP D.2 Strategic algorithms using organisations (p.49). As such the reports discusses operators and algorithms which require a consideration of contextual information for their operation and which are utilised at the organisational (meso) level to orchestrate and influence the activities of individual agents in a strategic manner. The aim of this analysis it to identify and develop a set of algorithms "able to utilise information provided at the organisational level" and satisfy the goals of the meso agents, as defined in task A2. Key to this are the notions of contextual information and algorithmic strategy which integrate this reports with those of WP A4: Geographical object modelling and WP C2 Measures for organisations.

2.1.1 Previous AGENT research.

In WP D1, Report D D2 : Selection of basic algorithms, an analysis of cartographic operators was introduced. Whilst the ultimate aim of the research for this work package was to, "identify a set of generalisation algorithms and their conditions of use according to the nature and geometric characteristics of an agent", the report also laid out a common framework within which to describe and discuss cartographic operators and cartographic algorithms at both the independent and contextual levels. In section 2.1 Typology (pp. 6-8), the report outlines a "hierarchical decomposition of generalisation operators", using the two main categories of "Traditional Operators", referring to the type of procedures employed in manual generalisation and "Digital Operators", referring to those operators that result from the translation of traditional operators into the digital domain. This report again uses these concepts and classifications in the further discussion of contextual algorithms.

2.2 Definition of contextual algorithms

2.2.1 Contextual Algorithms

In Report D D2 : Selection of basic algorithms (p. 4), a generalisation algorithm was defined as, "A formal mathematical construct that solves a generalisation problem by changing an object's geometry or attribute". This definition is again remembered here, but it is necessary to differentiate between this and the goals of contextual generalisation in order to provide the synthesis required to conceive of the system in its entirety. Independent generalisation considers geographical entities as existing essentially in isolation. The generalisation of these entities seeks to provide a cartographic representation that is perceptible at a target scale but that does not deviate too far from the veracity of their surveyed representation. Contextual generalisation aims to remove the assumption of isolation, by providing an awareness of the environment in which these entities exist. Within a multi agent system (MAS) it essentially provides the 'eyes' of the system. It is this environment as an emergent property of the data that provides the information that is communicated in a generalised map. Thus, the purpose at this level is to reduce the visual complexity of the spatial information rather than that of the data. If we adopt a view of cartographic space as *relative*, then the contextual properties of the information is the space. Describing the properties of this space is a complex issue, ranging from absolute aspects of space and context (such as absolute location), to emergent aspects such as perceptual groupings and information density (cf. Boffet 1999). These issues are discussed in more detail in report A4 and in section 1.3.2.

The role of contextual algorithms within this system is to provide procedures that allow for the visual re-organisation of information in such a way as to communicate the aspects of the space that are salient to scale and specific map theme.

The algorithms provide mechanisms to select salient groupings of information or, by its negation, eliminate unimportant information and to re-arrange or transform the representation of these groupings. Three types of operation are used to make these changes; Selection (usually together with elimination or differential symbolisation), Displacement and Aggregation, by either Typification or Amalgamation. These operation are described in detail in section 3 - under the heading 'State of the Art'.

2.2.2 Context and Strategy

Generalisation aims to provide an abstraction of geographic reality to enhance comprehension and communication of information. Generalisation algorithms, within this process, provide a method for abstracting phenomena within the database (themselves a first abstraction of reality). Generalisation is, as such, about altering the view of the model we have of geographic reality, implicit within the database. The different 'views' are defined by the strategies at the meso level that ensure logical consistency through the optimal treatment of the information that is to be communicated in the final map. For example, the use of an index such as Horton's (1945) or Strahler's (1960) for the selection of edges of a river network has the objective of optimising the geomorphologic properties of a river system (Rusak Mazur and Castner, 1990). However, it does not address the social, cultural or economical properties of a river system such as being a human resource or, explicitly, the perceptual properties of the river as a cartographic entity. In effect, the model of geographic reality, is optimised towards a single representation of the information specific to the scale of representation. Generalisation is about enhancing particular properties of a given entity, with respect to anticipated map use. Thus, in the representation of a river network in a mountainous region it is precisely these geomorphologic properties that are the most salient and important to communicate. As a further example, in the selection of roads in an inter-urban road network, the algorithm of Mackaness and Beard (1993) has the objectives of minimising the total line length of the network, but ensuring that all nodal points on the network remain connected. Here the property of the network as a resource to inter-connect places in the most direct way is enhanced. Alternatively, the algorithm of Reynes (1997) aims at maximising the connectivity to certain key 'attractive' nodes, such as places of social, economic or cultural importance. Here, the properties of the network as infrastructure are enhanced. The application of each of these algorithms will depend on the geographic context in which the entity exists and the form of the entity. The algorithm of Mackaness and Beard, is therefore perhaps more appropriate where the road network is weakly connected and settlement is sparsely distributed. The algorithm of Reynes is more appropriate, where the network is strongly connected and settlement more dense.

To attempt to optimise for more than just a subset of the possible properties of a geographic entity, at a reduced scale, runs counter to the purposes of generalisation. The information will become confused and chaotic, resulting in a net decrease in the overall communication of the map. At the meso level, algorithms therefore implement cartographic operators for different contexts by optimising for differing objectives salient to the information. The operator itself will always be the same but the strategy by which the operator is encoded will be dependent on the algorithm. Further, the selection of the algorithm is the responsibility of the meso organisation, which is best placed to evaluate the context and hence make strategic decisions about generalisation. Clearly, it is important that the objectives employed by an algorithm are therefore known to the meso agent.

The preceding discussion has focused on the use of an objective function in strategic algorithms to optimise the communication of information within, what the authors have called, different geographic contexts. This context could equally be called a phenomenological context. However, we can also identify two other contexts, which we will call the cartographic context and the perceptual context. The cartographic context is the result of the specific symbolic system used to encode the geographic information; the cartographic representation of space and the representation of the surveyed real world entities as symbols. This could be thought of as an ontological or semiotic context through which the rest of the mapped information is described. This representation of space imposes specific cartographic laws that govern how symbols can be arranged within it. Principally, this relates to the proximity between objects and the density of information per unit area of the map. In order to ensure that the cartographic model is consistent with the surveyed representation of the world, these laws will also enforce rules on location and topological relationships. Likewise, the form of symbolic representation of information also places constraints on how the information can be described, e.g. the level of line-work detail. Enforcing these symbol constraints is principally the responsibility of the micro agent. The perceptual context relates to the perceptual interpretation of spatial arrangement and the entities that exist through these relationships. The term spatial context, or perhaps more accurately relative spatial context, could also be used here but this is avoided since it introduces complexities concerning the understanding of space. Such arrangements consist of the spatial correlation in the 'values' of perceptual properties of groupings of objects. These properties include, proximity, orientation and symmetry and the patterns created by their spatial intercorrelations which are interpreted as representing specific geographic information (Delucia and Black, 1989). Whilst, the justification for both the perceptual and the cartographic contexts concerns the perception of information, the cartographic context is separated from the perceptual context since it relates to enhancement of the precognitive visualisation of data rather than the enhancement of the cognitive interpretation of geographic information. Hence the cartographic context exists more objectively and with less reference to scale whilst the perceptual contexts exist more subjectively and varies more according to scale.

2.2.3 Conflict

Since all generalisation algorithms alter the state of the data they are applied to, they all have the potential for *side-effects*, in the form of conflicting symbology and changes in metric and topological qualities. To minimise the potential for creating these conflicts and to remedy data that has become corrupted in this way, contextual algorithms must be aware of the rules that govern the space in which they have to operate, the cartographic context. In altering the data the algorithms must also be aware of the perceptual and geographic relationships that exist between the objects and groups of objects in a map, through which the geographic information is communicated[1]. The common approach is to satisfy more than one context simultaneously. For example, Regnauld (1998), in his generalisation of building clusters, uses specific perceptual properties, which he terms gestalt, of geographic information to solve the cartographic conflicts and hence satisfy both contexts. Similarly Hangou⬟t (1998) identifies groupings of objects based on geographical principles, which he terms phenomenological, and uses this information to direct the solution to satisfy the cartographic context. The selection of an optimal strategy for the generalisation of specific examples of geographic information or 'situations' (Ruas, 1999) will be the responsibility of the meso organisation. This is the best level at which to evaluate what elements of the geographic scene are most important and decide which algorithmic strategy should be employed to maintain or enhance these characteristics. Likewise, it is at this level where an evaluation of goals can be made and contextual conflicts identified. What is necessary for these decisions to be made, is a clear definition of the particular strategy that each algorithm employs.

2.3 Algorithmic Techniques

[1] In the terminology of Anderson (1993) the knowledge of this kind of information would be called 'procedural knowledge', since the semantic existence of the entities is created because of their spatial / perceptual organisation. This compares to 'declarative knowledge' where the opposite is true, the semantic entity exists as geographical knowledge apart from any real world existence. The terminology is not used here since it is ambiguous in generalisation, where 'procedural knowledge' means knowledge about sequencing of generalisation operations (McMaster, 1991) . Also, the distinction is slightly false since with familiarity, procedural knowledge ultimately becomes declarative.

This section, provides a brief consideration of the computational aspects of algorithmic techniques in relation to current generalisation algorithms. There are two main purposes to this discussion; to consider the issue of logical suitability of the computational approaches given the context that the algorithm is operating in, and to consider the issue of time-complexity of algorithms. The discussion is necessary to assist in decision making at the meso level and to consider issues of correctness of the logical model in representing the 'real world'.. For example, to answer the question; given an organisation are there algorithms which should not be used because the size of the organisation prohibits computation within a reasonable period of time? This is also necessary in the evaluation of an algorithm with respect to its own objectives. This is a different problem to the evaluation of the algorithm with respect to the generalisation it produces.

2.3.1 Logical correctness

Logical correctness refers to the suitability of an algorithmic approach to the problem. Here the two main concerns are the logical formulation and the strategy with regard to the aims and philosophy of the AGENT project.

The logical formulation relates to three main issues;

1. The symbolic system used to encode the objectives of the algorithm and deduce the solution for a given set of inputs.

Commonly, the system will be arithmetic but it may also be based on symbolic logic. Pertinent issues here are the degree to which expressed preferences may be captured as numeric values or if these are better handled in a system closer to natural language or qualitative description. Leading on from this is the issue of how different objectives can be combined, for example, in a cost function. An example of a numeric approach is Harrie (1999) where constraints are 'analytically' expressed as a system of linear equations, which are then solved together. The danger of this kind of system is that compromise is not necessarily the best approach in all cases of generalisation. Since some cartographic constraints that must be solved absolutely (e.g no symbology conflicts) and these may not be solved for sufficiently if the system is trying to look only for as solution that provides the best compromise amongst the more preferential constraints.

2. The logical model and data-structures used to represent space and spatial relationships.

To be made computationally tractable space and spatial relations must be modelled. For example, space may be discretised using voronoi polygons or a finite element mesh and perceptual or spatial relations may be represented using a vector analogy such as a Delaunay triangualtion network or minimum spanning tree (MST). In using these formulations their assumptions about space must be considered and their suitability evaluated.

3. The logical model used to represent geographic or cartographic processes.

In a similar manner to the representation of space, the representation of geographic process and cartographic process, for manual generalisation, need to be represented in a computable form. In general, we can identify characteristics that describe the algorithmic approach. Characteristics include global and local, empirical, goal-directed, idealistic and analogue, deterministic and non-deterministic. Global approaches may be global with regard to analysis, treatment or both. Global analysis considers the statistical properties of a set of objects in aggregate, perhaps for the entire data set. Conversely, local analysis considers only objects that are close together either spatially or in attribute space. Strictly speaking, local analysis is determined only on neighbours, in order to avoid the aggregation of data values entirely and hence the erosion of information. Global treatment refers to the scope of the domain of application of the algorithm. An algorithm that handles its input in its entirety is generally considered global. A good example of this concept is the Douglas and Peucker (1973) algorithm for line vertex reduction. Treatment here is performed in a global top-down manner on the entire extent of the line inputted to this algorithm. For contextual algorithms, a good example is the area-patch algorithm of Muller and Wang (1993). In this example both analysis and treatment are global. Local treatment takes a reductionist approach to processing, usually by applying the process sequentially on sub-parts of the inputted data. Empirical approaches are those based on observation and direct experience of the process of manual generalisation. Rule based approaches are empirical to the extent that the processing of data is pre-defined using rules formulated from studying examples of manual generalisation or interviewing cartographers about their experiences of generalisation. Rule based approaches can also be considered global in terms of treatment, since the rules are defined globally for all observed situations of a certain map type. Goal-directed approaches aim to ensure the output of an algorithm satisfies certain goals, based on map specifications and cartographic knowledge. Constraint based approaches are examples of these. Idealistic approaches treat the inputted data as representing consciously experienced mental objects and attempt to preserve the experience of these entities during the process, phenomenological and gestalt (perceptual) based algorithms are examples of this approach. Analogue approaches, rather than attempting to mimic the actual processes themselves attempt to find a suitable analogy with which to represent the process and solve the generalisation. The types of method vary widely, ranging from the modeling and simulation of a society by MAS, (Baeijs 1996; Morisset and Ruas, 1997) to the use of physics based energy principles (Burghardt 1997). Deterministic and non-deterministic relate to the predictability of the outcome of an algorithm. Hence, if an algorithm is deterministic the output can be known at the start of the treatment. Rule based approaches using production rules, where the action is tied to the condition, are examples of this kind of approach. Non-deterministic approaches usually involve a back-tracking operation and a range of different possible actions at every decision node, meaning that, practically speaking, the result cannot be predicted from the input.

The concern of strategy in relation to the aims and philosophy of the AGENT project raises a number of issues, some of which are also addressed in considering logical correctness. Three main issues are pertinent for the selection and design of algorithms for the project, which relate to MAS principles, such as goal directed behaviour, autonomy and local control:

1. Goal directed behaviour. This issue relates to whether the approach can be implemented within the agent framework. To be goal directed an algorithm must have specific constraints which it can solve and through this, goals which to be satisfied. Hence, these must be relevant to the global process as defined in A2 and A4. Whilst this seems characteristic of all generalisation algorithms, the discussion by Dutton (1999) on algorithms for line generalisation would indicate this not necessarily the case.

2. Scope of the generalisation operation. This issue relates to both the autonomous nature of an agent and its need to be goal directed. Essentially, it questions the extent over which the algorithm is expected to operate and the suitability of this to the philosophy of the project. Locally defined algorithms operate on a single or a comparatively small group of objects, whereas, globally defined algorithms are usually applied on a class wide basis. Global algorithms can thus defeat the objective of a MAS to make local decisions based on varied local contexts. In addition, the processing used by global algorithms usually pre-defines a sequence of actions. This deterministic approach removes the ability of an agent to act in a goal directed manner since its actions are pre-coded. According to MAS philosophy, the system is therefore made unstable. However, a global approach also has advantages. It ensures consistency in the treatment of features across the map and therefore can maintain gestalt properties. In addition, at a high level pre-defined sequential processing may be an entirely appropriate course of action. An example is in Brazile and Edwardes (1999), who make the assertion that it is necessary to sequence the generalisation of high priority linear features before any other features, even though this generates an inherent priority hierarchy. Arguably, sequential operations at this high level are consistent with a MAS approach that aims to mimic cognition. Since high level thinking may involve the use of, often well defined, sequential processing techniques.

3. Integration with previous research. This issue is important to the compatibility of the contextual algorithms operating at the meso level with agents operating at the micro level. Where relevant therefore, in translating a contextual algorithm into the AGENT paradigm, the implementation and conceptualisation of the problem should remember the nature of the individual objects as agents and the philosophy of recursive decomposition.

2.3.2 Time complexity

Descriptions of time complexity of algorithms are usually used to assist in the quantitative evaluation of an algorithmic approach. The measure relates to the amount of computational time (CPU time) required as the size of input to an algorithm grows[2] (Helleman, 1996). Whilst at the present time, we are not principally concerned about the efficiency of an algorithm, only it's suitability to perform a given task (its *correctness*), in terms of pragmatics we must pay attention to the issue. Since, contextual algorithms generally involve the interaction amongst both sets of entities and sets of objectives, the issue is of greater significance here than in the discussion of independent algorithms. Whilst, we can expect the majority of algorithms to have polynomial running times often, theoretically or without the use of any optimisation strategy, these will be at least of the complexity n^2. This means for the number of inputs n, the number of calculations that must be performed is n^2. Whilst, this is not problematic for small values of n, for much larger values of n and/or higher exponents, it becomes more difficult to justify.

Agents offer the exciting potential of parallel processing and the use of multi-threading. Such work lies beyond the remit of the current project. A more common approach, adopted in contextual generalisation to handle issues of time complexity, is through the creation of a dynamic indexing structure. The nature of this structure depends largely on the specific problem, but the principles are common. The aim is to create an index that allows for the rapid retrieval and manipulation of the contextual information, time complexity is thus decreased at the acceptable cost of an increase in space complexity (see footnote 2.). The intrinsic properties of the structure are then used to process the information. Important in this process is the type of index used, as this needs to relate to the objectives of the algorithm. In the partitioning proposed in the agent project (Brazile and Edwardes, 1999), the data structure is based on cycles that can be found in the selected road network. This generates the geographic context of a city block. The approach here is one of *divide-and-conquer*. All the possible candidate inputs can divided into smaller logical subsets, which are then computed separately as individual units. The approach works by reducing n and then summing the final computation times for each unit. So, if the time to compute the function for the entire problem was n^2 and n ($n > 0$) is divided into four separate units, the final computation time will be $4 * (n/4)^2$. Clearly, this will take as long or less time than solving the problem in its entirety. Partitioning has several applications in contextual urban generalisation (see A4). The index assists the process by defining geographical units in which certain objects can be considered as separate from the rest of the candidate objects. An example where this is employed is in Ruas (1999) for displacement. Since the partition represents a defined geographic entity, a city-block, the structure can also be used for district amalgamation, and street selection, based on aggregated building density properties to define the cartographic context (Peng and Muller, 1997). Likewise, Regnauld (1998) uses the approach of a dynamic index to create a Minimum Spanning Tree (MST) with which to index linear building clusters. The creation of this structure intrinsically stores the perceptual context upon which the typification is based. The usefulness of this approach will depend on the number of inputs, the original complexity of the problem and most importantly the complexity of creating the index in the first place. This is essentially a separate problem that requires its own consideration of time complexity. For MST, Regnauld uses a ‘greedy strategy’. This strategy speeds up the computation by always making the most locally advantageous decisions, where local refers to the available information at any given point in the processing. This strategy is also used by Burghardt (1997) uses in creating a ‘snake’ data structure for displacement. The approach has the advantage that it reduces computation by removing the need to look ahead for better more global solutions, however it has the disadvantage in some cases, though not MST, that it cannot be guaranteed to find the globally optimal solution.

[2] The issue of time complexity is usually associated with the issue of space complexity. Space complexity relates to the amount of memory (RAM) required by an algorithm and is determined by the amount of resources an algorithm uses in terms of variables and data-structures. Usually the requirements of the two components for an algorithm are inversely related, where an increase in running time can be afforded at a decrease in space or a decrease in running time at an increase in space. This situtation is known as the space-time tradeoff. However, given the size of available memory in most systems, the issue of evaluating algorithms on their space complexity is usually not relevant.

3 State of the Art

The state of the art of contextual generalisation, aims to provide a review and analysis of research on the implementation of contextual cartographic operators as algorithms in automated generalisation. The section is divided along the lines of the cartographic operators identified by the WP D1 report, D D2: Selection of basic algorithms. It follows the same typology, making the distinction between 'Traditional operators' and 'Digital operators'. A discussion of each cartographic operator is first proposed, followed by a description of the algorithms that have been designed to implement the operator. The aim of the review is to scope the current knowledge surrounding the subject and identify the areas where this knowledge is lacking with respect to the AGENT project.

The following operators are discussed in order;

- Selection / Elimination
- Displacement
- Aggregation
  - Amalgamation
  - Typification

3.1 Selection Algorithms

3.1.1 Introduction

Although the considerations developed in this chapter could be applied to selection for any feature class - even the meso-agents constituted through the generalisation process- the examples presented come principally from road network generalisation, from urban buildings and from hydrographics networks. These three kinds of feature classes present a wide range of cartographic constraints: connectivity, pattern reconstitution, feature density and no conflicts. For each kind of constraint, we will present a set of corresponding measures.

The complexity of algorithm and the level of abstraction, i.e. the depth of analysis that drives the selection process, is also considered. Although selection is the most contextual of generalisation operations, some purely geometrical operators could also be described as performing selection processes.

For more complex process, cartographic constraints must be modelled. Compliance to these constraints are evaluated through measures. Thus, we have to relate selection process to measurement, first by identification of reliable measures (see task C2), then by description of process allowing an optimisation of these measures.

3.1.2 typification vs selection

Consideration in this chapter relates to selection algorithms. However, AGENT project conventions require that a distinction is made between typification and selection. Both operators aim to preserve the legibility of a set of features by elimination of a subset, and conservation of the other. But, selection aims to preserve feature identity, and keeps each remaining object on its original footing, whilst, typification mainly targets a conservation of global set constitution, and allows displacement to ensure it.

Typification algorithms are considered in this report as a subset of aggregation. However of note here is that in some holistic process (for instance [Müller & Wang, 91], see below) separation between selection and typification is not clear, since no basic algorithm can be described.

In order to allow the multi agent system to chose automatically between typification and selection algorithm, it is necessary to express some rules of choice. For the time being, only basic ones are defined An example are those described in Bolletino Geodesico (1998). In this experiment reported here, after recognition of building alignment, a typification algorithm is chosen instead of a selection one if in this alignment, the biggest building area is inferior to twice the smallest one.

3.1.3 Duality selection/elimination

From a different point of view, selection and elimination appear to be dual operations. They are distinguished only to assist the different user perspective that they are related to.

The notion of elimination is used when focusing on eliminated features, and selection is preferred when remaining features are the subject of focus. Focusing on eliminated features is most useful when the selection/elimination operation is triggered by the conflict of features that therefore have to been eliminated. Whilst, selection highlights the features that need to be preserved for the integrity of cartographic message.

3.1.4 Criteria of Classification

Selection algorithms may be classified according several criteria.

Here a typology of selection processes, according commonly used object classification on which selection processes will be applied, is proposed. Typologies and classification in generalisation are always pretexts to discussion and critics. Selection does not infringe this rule, with many arguments inspired by object classifications. For instance, elimination of bends can be seen as a selection algorithm when each bend is considered as a single object, but it will be classified as an internal shape deformation if the basic object level is a line, such as in classical vector databases. In keeping with the logic of the agent project, it is noted that selection/elimination of detail into a simple object is more relevant to the micro level, and therefore should not be developed here. Nevertheless, two such algorithms are briefly recalled, in order to present the concept of emergent behaviour.

For disjoint atomic objects, three kinds of tools used for selection are presented; cost function, underlying fixed structure, and iterative processing. These are detailled and for each examples presented.

Classical algorithms are then presented classified according to the object type on which they are applied. These criteria have been retained in order to present algorithms, with respect to their applicability to the project. The tools on which they are built are detailed, showing that some may use two or the three kinds of tools. Advantages and drawbacks and condition of use are also detailed where possible.

3.1.5 Algorithmic tools for selection

3.1.5.1 Erasing algorithms

These algorithms are not formally dedicated to selection and elimination. They compute first geometrical transformations, which can result in selection. Two examples of this category are:

- Whirlpool algorithm (developed by (Dougenik, 1980) and first proposed by (Chrismann, 1983) (presented in the DD1 report) .
- Area-patch generalisation process proposed by (Müller and Wang, 1991), and discussed in the chapter on “typification algorithms” and in this report.

Whirlpool algorithm is a line simplification algorithm. It works by using a cluster computation that generates a priority ranking for loop elimination. This is similar to the use of erosion to remove the thinnest features in the Müller & Wang algorithm.

These algorithms are interesting as examples of new behaviour generated but not directly related to the purpose of computation. The selection/elimination behaviour is a serendipitous side effect (in contrast to the bad side effects of algorithms often found in generalisation) since the property was not expected.

The algorithms presented here are fairly basic. They do not take into account any semantic considerations. They have been described in previous report, and are mentioned here only as archetypes of selection. Their particular interest lies in the property of emergent behaviour. In the best cases, multi agent selection should present such emerging behaviours.

3.1.5.2 Selection through short-cut of cost functions

Automation of generalisation needs measures to drive the process. Also selection measures are used, to distinguish features that have to be preserved from those that can be eliminated.

In many cases, selection is computed through an indicator of relevance: most relevant features are kept, less relevant ones are eliminated. This indicator of relevance is used as a cost function, and the definition of the algorithm is mainly based on the computation of this indicator. Many simple generalisation algorithms use basic measures as indicators: elimination of smallest buildings or shortest cul-de-sacs use, as cost function, a measure respectively on area and length. The ability of such a process lies in the constitution of more complex measures that express the more abstract reality of the data.

Each measure involved in a cost function is related to a corresponding cartographic constraint. The selection may consist either in successive short cuts of each measure, or in a single short cut of a global cost function aggregating a whole set of measures.

A basic example of this method is given in Bolletino Geodesico (1998), which eliminates, when required (cf. see above), the smallest building. The equivalent measure used for the cost function is the area of each feature, which reflects the basic law of generalisation.

A short cut according more complex constraints is demonstrated by Mackaness (1995). Here he defines measures related to urban road network analysis. Three measures are developed: depth, connectivity, and control. It should be noted that these measures are topological, but seem quite far from classical measures of graphs: since they are inspired by a concrete problem, they are probably better adapted. The parameterisation of the short cut threshold on the three measures is not defined by Mackaness,, instead all possibilities are left open for the use of these measures.

Ideally, all cartographic constraints should be taken into account in a short cut selection. But in fact, selection/elimination processing on other features may modify the relevance of the remaining features. For example, if an important connection between two points is insured by three different roads, each road could be easily eliminated, but one of them has to be kept : the elimination of each road depends on the selection of the others, and requires a global view of the relationships. Such a view may be insured by a global structure.

3.1.5.3 Selection derived from a fixed structure

Some selection algorithms are directly based on a predefined underlying data structure. The example we present here comes from Beard and Mackaness (1993) and is based on MST (minimal spanning tree). The selection provided by the MST is a minimal well connected subset of the original network. All points of the network remain connected, but transit path are changed into cul-de-sacs.

Such algorithms are highly dedicated, fixed, and seem hard to modify in order to abide by a large set of cartographic constraints. In particular, parameterisation for a different level of generalisation is not directly possible, and requires another mechanism. For that reason, they must be combined with other type of selection.

For instance, Beard & Mackaness complete the MST selection by other arcs, according a cost function, taking into account the size of roads. They also use the ratio of completeness to limit the change that occurs form the elimination process.

3.1.5.4 Iterative selection

Since an indicator of feature relevance needs to depend on the other remaining features, some authors use iterative selection. At each step, a feature is eliminated, and measures used for the selection are then recomputed for this new situation. The process is repeated until the situation satisfies a predefined criteria.

An example of this method is given by [Ruas, 99] for the aggregation of urban blocks, which is modelled as the elimination of separating streets (cf. below).

Though this method, with a recomputation of each measure at each step, may look heavy , in fact it could be lightened, since most measure are local, and have to be recomputed only on the area. However, it still remains time-costly compared to a simple short cut.

3.1.6 Control of selection

The control of selection is led in most cases by the same constraints, and equivalent measures, as other generalisation operations. Nevertheless, the specificity of selection, whose goal is a reduction of number of features, implies the need to determine if this number is relevant to the scale and the purposes of the targeted map. Even though this selection constraint highly depends on the considered class, and the specification of the map, Töpfer (1974) Töpfer and Pillewizer (1966) propose a law to lead selection, known as radical law. The general form of the law is:

$$n_t = n_o . \sqrt{s_o / s_t}$$

with : n_t : number of features in the targeted data set

n_o : number of features in the original data set

s_t : scale of targeted map

s_o : scale of original data set

3.1.7 Description of algorithms

Here a selection of elimination algorithms according their domain of use is presented.

3.1.7.1 Hydrologic Networks

Various research has looked at the use of geomorphological indicators for the selection and elimination of tributaries. Rusak Mazur and Castner (1990) and Richardson (1993) use Horton's (1945) method of assigning orders to stream segments that takes into account the shape of junctions. The order is determined by a consideration of the angle at which an upstream segment joins a downstream segment. The segment that joins with the least angular change is considered the continuation of the downstream segment, the ordering is then developed from this. Of note is that Thompson and Richardson (1999) observe this is a very similar model to that of the 'good continuation' principle. Likewise the ordering of Strahler (1960) has also been suggested (Mackaness and Beard, 1993). The problem with these methods is that the geomorphological phenomena is placed primary and they fail to account for graphic constraints such conflicting symbology. However, in principle at least, they produce results very similar to those found in manual generalisation.

3.1.7.2 Building

As was outlined in introduction, most building elimination is performed through typification, since buildings are commonly organised into blocks and aligned to roads. Ruas (1999) presents a pure building selection algorithm. The process is iterative, since elimination is computed step-by-step. At each step, an elimination cost of each building is computed. This cost function is the sum of various measures on buildings and their contexts, which is detected through a Delaunay triangulation structure.

This method takes numerous constraints into account:

- **semantic** : particular buildings are automatically kept
- **size** : smallest building are preferentially eliminated.
- **distances** : buildings too close to other buildings are preferentially eliminated.
- **road network interaction** : buildings close to the road network are preferentially kept
- **multi-directional overlap** : buildings overlapping in a wide range of directions are preferentially eliminated.

According to these constraints, a cost function is proposed for each building. The algorithm is then computed iteratively. At each step:

1- The most conflicted building (i.e. the highest value of the cost function) is eliminated.

2- The cost function of contiguous buildings is updated

Iterations are repeated until constraints all are satisfied, i.e. cost function is inferior to a predefined threshold.

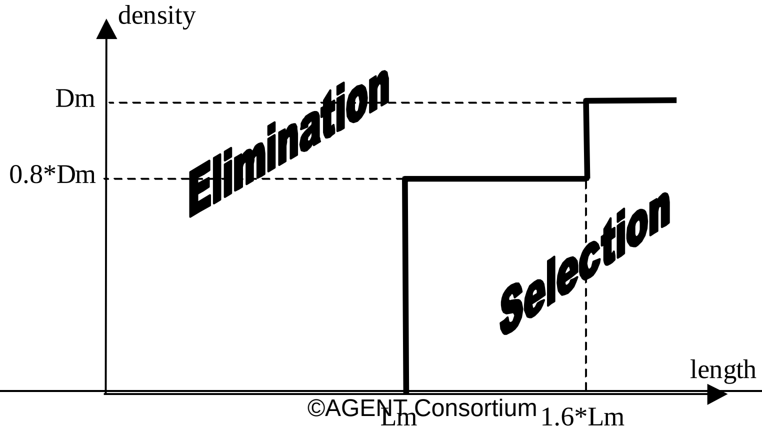
3.1.7.3 City road network

3.1.7.3.1 Cul-de-sac

The elimination of cul-de-sac used by Ruas (1999) is performed as a pre-process, according a cut of simple cost functions. In fact, it depends on both the length of roads and the local density, cul-de-sac are eliminated according to three conditions:

- when their size is less than a minimal size (Lm) or
- when the local density of road is higher than a density threshold (Dm) or
- when the size is inferior to 1.6*Lm, and the density is higher than 0.8*Dm.

The planar separation of points (size, density) between selection and elimination domains is shown in the figure below.



As pointed out in Ruas (1999), this algorithm is dedicated to urban situations. For inter-urban networks, cul-de-sac are mostly used for connection, and they have to be treated conjointly with features that are connected with them.

3.1.7.3.2 connecting roads

In Ruas (1999), the previous algorithm is used as a preprocess for the next one.

<u>City road elimination (Ruas, 1999)</u>

Ruas (1999) presents a street elimination tool that could also be described as an aggregation operator, See section 'Aggregation'. It relies on two operations: first a most conflicted block is identified. Then a neighbour of this block is selected for merging. The separating road is then eliminated causing the blocks to amalgamate. The process is then repeated.

The operations proceed as follows:

1- A constraint violation indicator is computed, as the density of features in each block. The indicator concerns only blocks whose area is less than a given threshold, which is a function of density. Larger surfaces have to be treated by other processes.

The indicator takes into account :

- the area of the block
- the density of the block

2- For the most conflicted block, the first elimination process is launched: for each bounding street of the block, a cost of elimination is computed. This takes into account :

- the area of the block - the minimal area is preferred
- its density - the highest density is preferred
- the compaction of the resulting block if this aggregation is achieved - most compact results are preferred.

Bearing in mind that several functions could fit with these requirements. A formula is used to compute the cost function used for merging:

$$f(\mathit{neighbour}) = (\mathrm{area}(\mathit{neighbour}) / \texttt{mac\_area}) \quad * (1 + \texttt{max\_density} - \mathrm{density}(\mathit{neighbour})) \\ * (1.1 - \mathrm{compactness}(\mathit{resulting}))^2$$

For each block an indicator of constraint violation is computed. The most conflicted block is then launched at the first step. Similarly, for each neighbour of that block, a merging cost is computed. It takes into account the internal homogeneity of merging blocks and the importance of the dividing street. Finally, the minimal cost operation is retained. The process is then iterated, with updated indicators of constraint violation.

By this method, consequences of each elimination are taken into account in the successive iterations.

It should be noted that only complex situations require this treatment: Ruas 99 uses concurrently, short-cut selection, for instance by elimination of cul-de-sacs if they are too short or located in a high density area.

This algorithm is highly efficient, since it takes into account a wide range of constraints in a fairly simple way, producing very satisfactory results. But it has been design for a complex process : most large blocks are not touched, there conflicts are dealt with usingthe building selection algorithm described hereafter. In fact, it appears more a block aggregation process than a road selection algorithm.

3.1.7.4 interurban road network

3.1.7.4.1 connecting roads

3.1.7.4.1.1 Selection based on connectivity

Connectivity is the primary constraint required for a network. For that reason, some selection algorithms are based on minimal spanning trees, which insures the connectivity of the resulting selection. Among these algorithms, we present that of, Beard & Mackaness (1993).

The minimal spanning tree defines a first set of selected arcs. A second set can be then computed in order to reach the selection ratio defined by the user. The choice of this second set is done according to the semantic importance of the roads and the level of connectivity of their extremities, but the corresponding function is not given.

Even if this method did take into account several constraints, one is privileged, which leads to the potential for heavy violations of the others. The interest of the method lies in the heavy compliance of the constraint related to the underlying data structure.

It has to be noted that such a method uses the geometry of the network, but does not take into account pattern and spatial analysis : for that reason, results would be very poor for city road network.

3.1.7.4.1.2 Selection constrained by travelling

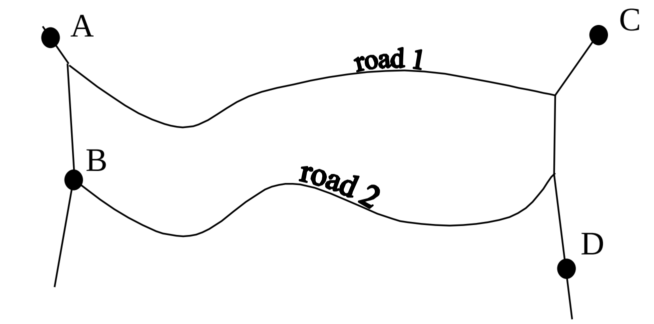
The algorithms of Morisset & Ruas (1997) and Reynes (1997) are short cut algorithms based on consideration of travelling connectivity. Reynes, (1997) uses a direct approach, through a short cut of an accessibility function. Whilst Morisset & Ruas, (1997) use an model of travelling by multi-agent computation. Both require the definition of attracting points.

We detail the Reynes process here:

1. A set of important connecting points is pre-defined.
2. All shortest paths between connecting points are computed, by the Djikstra’s algorithm
3. For each arc, the amount of travel using it is computed. This measure on each arc constitutes a cost function
4. The most used arcs are kept - the others eliminated. The set of features is cut according the number of arcs that have to been eliminated, or a predefined threshold on the amount of travel.

Drawbacks :

Double paths are not always well detected since equivalent short paths may be computed separately on parallel arcs (cf. fig below).



Road 1 and 2 are parallel, and achieve the same traversal role, since substitution of road 1 by road 2 does not perturb cross circulation from A to C and from B to D. With the Reynes algorithm, both road 1 and road 2 are kept, since both are traversed, by the shortest path from A to C, for road 1, and from B to D for road 2, whereas one of them may be eliminated. Elimination of roads 1 and 2 are inter-dependent.

On the contrary, segments joining A to B and C to D are independent. Each one is needed for the local inter-connections.

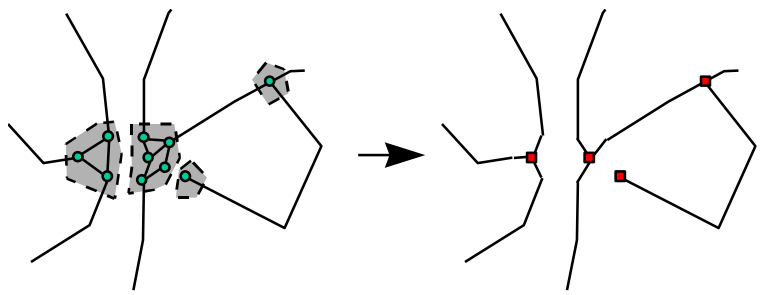
Morisset & Ruas propose a parallel way to achieve a similar selection taking account the communication functionality of a road network. Instead of computing globally the minimal paths from a point to another one, this computation is made by agent, each one being responsible for a given path. Possible traversals between the whole set of point pairs are not necessarily all computed: a random subset can be choosen for this method, shortening the computation time. This is the principal advantage of the method.

3.1.7.5 intersection

Selection of intersections could be seen as a part of road selection, since road selection implies intersection selection, whereas the opposite is false (selection of intersection points does not imply arc selection).

But in some case, it appears as a specific operation. Particularly for complex junctions, without any urban context. For such a case, Mackaness & Mackechnie (1997) propose an algorithm based on clustering. Clusters are constituted using a distance constraint, and by their topological connections.

1. clusters are constituted though computation of a dendrogram, based on Euclidean distance.
2. each cluster is decomposed into connected subset
3. for each sub-cluster, a centre of gravity replaces the other points.
4. each arc joining two points in the sub-cluster is eliminated
5. each arc joining a point of the cluster is then connected to the new node



This operator could be described as close to the topological unification operators used by geographic information systems to constitute topology . But in fact, these operators change the network topology, whereas Mackaness & Mackechnies' intersection selection respects it.

3.1.8 Conclusion

Selection algorithms have been examined from two perspectives. First a consideration of the cartographic constraints that the selected sub-set has to respect has to be made. This point of view leads to the definition of measures to evaluate the compliance of a solution relative to its constraints. Next the algorithm has been characterised according to how it finds a satisfactory solution. We have detailed some of the methods used for such an optimisation.

Short cut selection is adapted for clear and simple violations. In fact, it is close to class elimination in the map schema definition. Selection through a fixed structure is needed when the constraints relating to structure take a high degree of priority compared to the others. Finally, iterative selection seems the most precise, since consequences of selection/elimination on a given feature are taken into account at each step for the successive operations.

3.2 Displacement

3.2.1 Definitions and Overview

3.2.1.1 Introduction

Displacement is used to counteract the problems that arise when two or more features are in conflict. Conflicts arise through lack of distance between objects or due to inconsistencies caused by other generalization operations. The way to counteract these problems is to shift or deform objects.

Such a first definition seems very broad, but it is exactly this variety of situations that makes the problem of displacement difficult. The following characterisations will make things clearer.[3]

3.2.1.2 Reasons for displacement:

Three reason for displacement can be distinguished (see Figure 1):

1. Conflicts due to the *decrease of absolute empty space* between two objects, when moving from one scale to another. (Figure 1a.)
2. Conflicts due to *increased relative symbol width* (mainly of linear features). (Figure 1b)
3. Conflicts arising when *other generalization algorithms change geometry* of (line) objects without adjusting the neighborhood to this change. (Figure 1c)

Note: Generalization due to reasons 1 and 2 are also necessary in traditional cartography. Conflicts of type 3 arise only in automated generalization when algorithms introduce new conflicts.

---

[3] Note that we also have a displacement at the micro-scale (see '*polygon_dispalace_by_vector*'). Displacement at the micro-level is necessary due to inadequate generalization, altering an object's position. Such objects need to be moved back to their initial position. As there is no conflict *between* objects, such a shift is not an act of displacement at the meso scale (thus in correspondence to our distinction of algorithms at the micro- and meso-level.

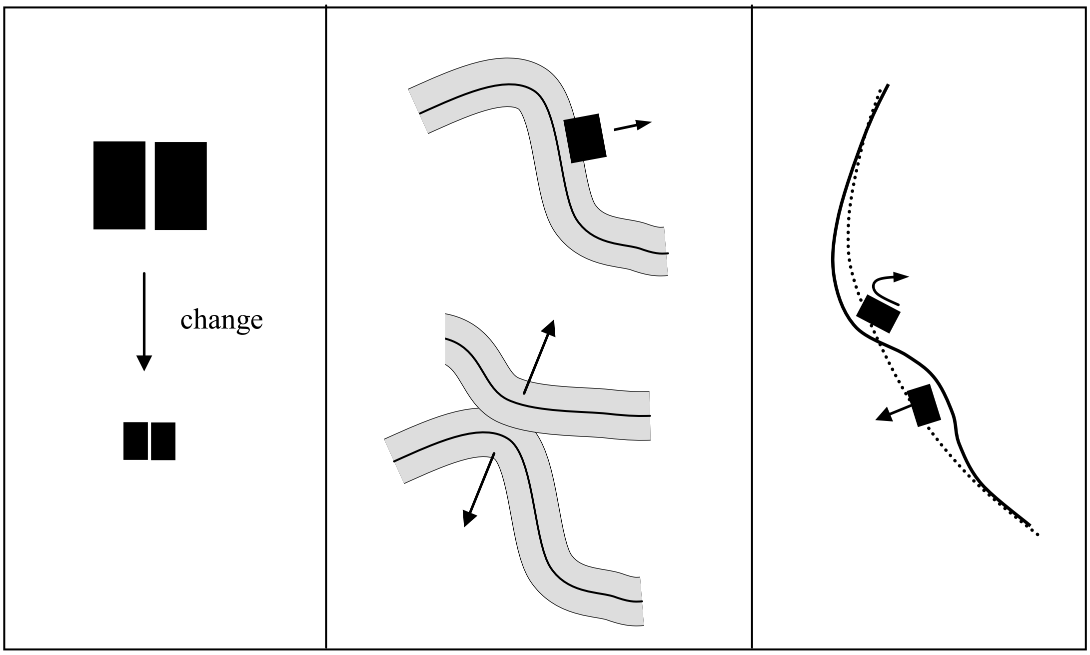


Figure 1a) b) c)

From an algorithmic point of view, we can also interpret the increased symbol width as a change in geometry: the border of the road symbol is moved. However, the displacement due to changed object geometry is more difficult to treat, as displacement can be very inhomogeneous along a line, varying in both direction and magnitude (Figure 1c) .

3.2.1.3 Displacement focus and approach

Two *displacement focuses* are distinguished:

- *Immediate displacement*, is the conflict solution of objects directly interfering.
- *Displacement propagation*, is the adaptation of the environment to this altered situation. We can distinguish two types of propagation:
    1. Propagation within the displaced object itself: When displacement conflicts are resolved for large objects, normally only the conflicting parts are shifted. To ensure connectivity to the other parts of the objects and to maintain the object's shape, the other parts of the object need to be adapted to the altered situation. Or to say it the other way round: the displacement has to be propagated.
    2. Propagation to neighboring objects.

       Note: It could be said that this type of propagation is just the iterative use of a displacement algorithm to all conflicts. This is not strictly true, as propagation should not only resolve all proximity conflict, but needs also to maintain the relative relationships (such as relative proximity, alignments) amongst objects.

Further, two displacement approaches can be distinguished:

- *Translative displacement*, is the movement of entire objects.
- *Deforming displacement* (anamorphism), is the solution of proximity conflicts by changing object geometry.

Mathematically: Translative displacement is the shift of objects by adding the same displacement vector to each vertex. Deforming displacement is performed calculating a shift of each vertex by an individual displacement vector. As such, it keeps the displacement distances minimal at the cost of altered shape.

Strict displacement is mainly applied on small polygons, such as buildings. Linear objects (such as roads) and bigger polygons (such as land use parcels) instead undergo the process of deformation.

3.2.1.4 Structure of this report

In section 3.2.2, the most important approaches to displacement are described. The algorithm description is kept very short. It is directed to the reader familiar with displacement methods.

Section 3.2.3, intends to compare displacement algorithms. Evaluation criteria are described. However, it is seen that a comparison can not easily be made, as algorithms address different problems.

Section 3.2.4, is a first step towards a recommendation. The algorithms are placed in the context of a geographic situation. This makes different situations visible, for which algorithms need to be implemented. The main goal of the section is therefore to classify displacement problems.

In Section 3.27, a method for each type of conflict classified in Section 3.2.2.4 is recommended. While methods seem to be robust enough for integration, a lot of knowledge is missing to guide their process.

3.2.2 Description of main approaches

Methods for displacement have a long tradition in computational cartography. In the 1970s various researchers from Hannover (Hannover-Schule, especially Lichtner (1976)), focused on displacement methods. Based on several measures between conflicting objects, he computed displacement vectors for each vertex to solve a proximity conflict. Such methods, computing displacement vectors for vertices based on proximity measures within their neighborhood, are named '*mechanistic'* in this report. Nickerson (1988) followed the same ideas, but concentrated on the displacements of lines. Mackaness (1994) proposed radial displacement from a conflict center. Ruas (1999) extended and improved the mechanistic approach to a complex system of constraint-driven iterative displacements.

Bobrich (1995) first viewed the problem of displacement as *optimisation* issue. Displacement, within this framework, is described in terms of energies used to perform positional changes and deformations of objects. The solution of a displacement problem is then associated with the problem of energy minimization. The subject of optimisation is a well-studied problem in computer science. As varied are the approaches in that domain, so to are the proposed methods for displacement: Burghardt *et al.* (1997) used active splines, Harrie (1999) a least squares method, Hojholt (1998) a finite element method and Ware and Jones (1999) simulated annealing. A new approach was presented by Baeijs (1996), bringing Multi-Agent systems into cartographic generalisation.

Before addressing a comparison and evaluation of displacement methods, the most relevant mechanistic and optimization approaches are briefly described.

Mechanistic approaches:

3.2.2.1 Nickerson (1988)

Nickerson's algorithm starts with buffering roads. Intersecting buffers indicate roads with insufficient distance. A classification of buffers helps to distinguish different problems, especially buffer road intersections (polylines with common nodes).

The intersecting buffers are used as a basis for computing displacement vectors for each vertex. A main displacement direction is determined and, using a triangular-shaped filter, a smooth displacement magnitude is achieved. Special treatment at road intersection makes the algorithm stable and valuable.

When dealing with different object classes, the authors propose a hierarchical displacement, moving 'weaker' objects (e.g. contour lines, power lines) while keeping important objects (rivers, railroads) fixed (*one-sided propagation*). The algorithm by Nickerson is implemented in PlaGe (IGN).

3.2.2.2 Ruas (1999)

Ruas's approach focuses on the displacement of buildings due to symbolization of roads. As such, it tries to solve the same problem as most optimisation problems (see below) -find the new positions of buildings within a partition built by roads. The problem is thus twofold: Displace buildings such that they do not overlap the increased road-symbol, but also ensure minimal distance between buildings.

The fundamental displacement vectors are created by the widened road symbolization. The increase of symbol width is interpreted as forces, which are propagated to the adjacent buildings. The algorithm recursively displaces the building with greatest proximity conflict. After a building is displaced, it is kept fixed during displacement of other buildings.

The displacement of a building is guided by constraints regarding its shape and absolute position, as well as its minimal distance and relative position to other buildings. The description how individual displacement vectors result from each constraint, and how they are aggregated to a final displacement vector, is beyond the scope of this state of the art.

In summary, it can be said that the algorithm by Ruas includes a lot of cartographic knowledge which is missing from other algorithms. Thus, the algorithm produces good cartographic solutions. The algorithm fails not only in the degenerate case where not enough space is available to perform a correct displacement, but also in simpler cases due to inappropriate displacement sequence. Such cases are rare and could be avoided in the AGENT system by using aggregation and elimination earlier in the process.

3.2.2.3 Baeijs (1996)

Baeijs designed a Multi-Agent System using reactive agents. Each vertex is associated with an agent. Each agent has a proximity scope, wherein the agent is looking for other agents to identify proximity conflicts. As an agent should never be in conflict with agents of the same geographical object, each vertex makes part of a group identifying all agents of the same object.

Using this initialization, repulsion forces (electrostatic forces) are computed between agents. Each agent tries to push the other agents out of its proximity scope. By displacing the agents according to these repulsion forces, an equilibrium state is reached, identifying the end of each cycle.

The approach by Baeijs is interesting from a computer science point of view and brings the idea of MAS into generalization. However, it lacks sufficient cartographic constraints to guide the system to a good solution. Only a 'proportional following' is introduced, that spreads a fraction of the repulsion forces to the other agents of the same group, thus not ensuring shape control.

Optimization approaches:

3.2.2.4 Burghardt (1997)

Snakes (active splines) are used in pattern recognition to detect fuzzy object contours. Hereby, lines are attached to an object in an energy minimizing way (attraction). The displacement of lines in cartography can be viewed as the opposite, thus as energy-minimized repulsion.

The total energy of a line consists of inner and outer energy: outer energy comes into play when a line segment is too close to another line (potential of displacement). Inner energy is generated when the shape of a line is distorted (potential of Gestalt). The optimal shape and position of the displaced lines are given when their energy is minimal. Burghardt enriched this model by introducing two weighting factors: one factor forces lines forming junctions to intersect in an orthogonal way, a second weighting factor allows changing the magnitude of displacement regarding an object's importance.

The numeric conversion of this problem was done using the principle of variation as well as the Greedy-algorithm. The implementation of the Greedy-algorithm is somewhat straightforward, but makes an optimization for each point only. On the contrary, the implementation of the variational method computes the optimum over thee entire line in one step (producing better results), but is resource intensive and complicated.

3.2.2.5 Harrie (1999)

Harrie describes a method to resolve readability conflicts arising from the insertion of roads or buildings in existing datasets. As such, the problem is wider than the problem analyzed by Ruas, as in Harrie's study roads also might be displaced (or at least locally adjusted).

In a first step, a series of geometric and topologic constraints (e.g. stiffness of buildings, road intersections, curvature of roads) are expressed (translated) as linear functions of the object coordinates. In a second step, theses equations are assembled to a design matrix, stating the displacement problem now as linear equation. This equation has to be solved for the displacement vector of all vertices. As some of the constraints are contradictory, a displacement vector fulfilling all constraints will never exists. A residual vector needs to be added. Because the equation system is now overdetermined, i.e. there is no unique solution. The 'best solution' is the one that agrees, as much as possible, with the constraints; i.e. we face a minimization problem of a function of the residual vector. To solve the equation system, the least squares method is used.

The approach is promising, as cartographic constraints can be formulated one-to-one in a system. The translation of 'spoken' constraints into mathematical language is not always straightforward and needs further research. However, the main problem is to find good weighting factors when assembling different constraints to a big design matrix.

3.2.2.6 Hojholt (1998)

Hojholt's algorithm intends to displace buildings due to increased road symbol width. The main goal is to displace the buildings closest to the conflict source. It does not look for the best allocation of all buildings in a partition. The displacement is propagated into the neighborhood without taking new conflicts into consideration.

Hojholt interprets the map space as deformable material: the increased symbol width pushes the partition boundary towards its interior, compressing the space with its buildings. To model this problem, the Finite Element Method (FEM) is adopted. FEMs are used in most engineering domains to solve (partial) differential equations. The deformation of material is herein a well-studied problem in structural mechanics.

To achieve a discretisation of the problem domain, a triangulation is computed. Different stiffness values of the triangle edges help the modeling of different material properties, such as rigid buildings.

The idea of continuous deformation is intuitive - it keeps main object relations and forms, and the propagation problem is solved without introducing new problems. With FEM, there is a tool to model this idea. The power of the approach is not yet well explored. A significant problem is the fact that the method does not solve a conflict for certain; if the resistance of the environment is too high, not all initial conflicts are resolved. A way to handle this problem could be to process the method iteratively. Other open questions are: how to introduce additional cartographic constraints in the model? Is there a way to use the information of strain that rests in the deformed area?

3.2.3 Evaluation of algorithms

3.2.3.1 Quality, reliability and ease of use

Judging the quality of the presented approaches is not an easy task. Most of the algorithms are not implemented in platforms that go beyond testbeds. Illustrations in proceeding papers are usually limited to one example, which differ in data. Therefore a comparison is nearly impossible.

The results of the road-vs-building displacement methods seem generally to meet the cartographic requirements. Every method has its degenerated cases, where the method fails. As Ruas (1999) points out, the quality is directly related to the change of scale. With bigger scale changes, the algorithms must fail. It is then the task of the aggregation and elimination operators to reduce detail to a readable level.

The Nickerson-Freeman algorithm for the displacement of lines is integrated in PlaGe and has been shown to produce relatively good and robust results. Burghardt's snake algorithm is presented with good results. However, there is no knowledge about the robustness of this method.

Every method is based on cartographic knowledge to guide the process: While Ruas's algorithm can be performed step by step, the optimization methods need to define one evaluation function - integrating all constraints - at the beginning. There is little experience of how to tune weights when assembling such overall cost functions. A desired change of parameter (as a result of agent communication) is therefore more easily performed in mechanistic approaches.

Of course, the aim of this evaluation would be to find criteria to judge the quality of the methods. This is not possible, as the algorithms do not address the same problems. This point will be discussed in more detail in Section 2.3.3.

3.2.3.2 AGENT-philosophy vs. algorithm approach

Displacement is a process at the Meso-Level: a Meso-Agent is responsible for the proximity conflicts within a partition and this will guide the process of displacement. Agent communication and various measures will ensure (hopefully) good results. From this point of view, a global optimisation approach such as the one by Harrie or Jones-Ware is contradictory. The iterative computing towards a final solution (such as that described by Ruas) is better suited, as agents can guide and influence the process. The algorithm by Hojholt as a starting point for further corrections would also be adequate.

In conjunction with the idea of backtracking and internet-capable response, the issue of the problem of CPU-resources claimed by the methods needs also to be addressed. Optimisation algorithms usually need to solve huge linear equation systems, which results in exhaustive manipulations. Their cost in terms of CPU-time is therefore prohibitive. The problem area has to be reduced drastically to allow fast computation.

3.2.4 A view towards the entire problem domain

In this section, a looking towards the entire problem is made. Here, it is not the algorithm design that is really important, but the conflicts that can be resolved and the solution obtained.

All methods described above intend to formalize displacement. Nevertheless, *they address different problems*.

3.2.4.1 Object Types

A conflict may exist between features of any data type. Point vs. line, line vs. line, line vs. polygon and polygon vs. polygon are common problems.

As mentioned in the Section 3.2.1, lines and bigger polygons are normally deformed. As polygons consist of a boundary line, they can also be treated as lines, thus building the group of 'deformable lines'.

On the other hand, small polygons and points are shifted. When a representative point of a polygon is identified, these objects can be grouped as 'rigid point objects'.

In a first approach, the following object interactions can be identified for a displacement process:

1. Deformable line interfering with deformable line (e.g. two conflicting roads)
2. Fixed line pushing rigid point objects (e.g. displacing houses along a road)
3. Fixed point object pushing a deformable line (e.g. triangulation mark urging a road to displace)

4. A group of interfering rigid point objects (e.g. a group of buildings)

Note that, it is not only the displacement itself where different data types need to be handled. Also in displacement propagation, different data types present themselves. For example, the method by Ruas propagates displacement in urban areas, but is not intended to compute propagation also for streets of rivers. On the other hand, the algorithm by Hojholt can handle (theoretically) all object types.

3.2.4.2 Conflict Potential

Displacement may be performed in situations with *different conflict potential*: The 'conflict potential' of a situation is the probability with which a displacement of one object will trigger successive propagation. This is mainly dependent on the density of objects, but can also be due to alignment of objects or the straightness of line. For example, the conflict potential of an isolated building along a road in landscape of very low building density is minimal. Whereas, the methods described by Ruas, Harrie or Ware focus on urban partitions with high conflict potential.

Closely related to the topic conflict potential is the topic of sequencing (see below).

An algorithm in a partition with high conflict potential can fail because there is not enough space to display all objects with sufficient distances. Such natural failure is not a problem, as long as this problem is detected. Related to this problem is the question, of whether an algorithm will solve the initial conflict definitively or not. When applying mechanistic approaches, the initial problems are normally solved for certain (maybe introducing new problems). But when applying optimization methods, the solution over the entire problem domain does not necessarily resolve the initial conflict for certain. For example, when the space is too constrainted, the method by Hojholt can fail to solve the displacement for the triggering objects.

3.2.4.3 Sequencing

Two sources of problems related to sequencing can be distinguished:

1. *Convergence in propagation*: When dealing with heavily covered map regions, displacement propagation may never end, as the translation of one object results in new displacement problems. The conflicts are only pushed to and fro, no convergence towards a satisfied solution is found.

   Two ways to overcome this problem can be conjectured. A meso-agent needs to handle this problem, which might be difficult, as sequencing is a problematic task in multi-agent systems. More adequately, a displacement algorithm that resolves conflicts within a whole partition, by having a sequencing procedure inside it, can be used. There are two approaches to this kind of task: Ruas's algorithm iteratively displaces the objects, avoiding an infinite loop by keeping objects positions fixed after having displaced them once. Harrie's and Ware-Jones' method determines the final position of all objects in one optimization step.

2. *Order*: Suppose that it is necessary to change the geometry of an object (e.g. a road) in such a way that this change forces another object to be displaced. When this change of geometry is made without any immediate processing of the affected objects, a map that is in an unstable state is produced, as many conflicts are in the map still waiting to be solved. In even more degenerated cases, the objects lose their initial topological and semantic meaning, and are, therefore, impossible to correct later.

   Therefore, a change of geometry or symbolization has to release all processes for solving subsequent proximity conflict. For example, if a road is displaced, all objects along the displaced road need to be adapted immediately to this new situation or at least need to be marked for treatment in a subsequent step. Therefore, the original relation to the displaced road needs to be memorised, as this knowledge is no longer available in the altered map.

   The advantage of triggering immediate corrections is not only to keep the map in a state with as few conflicts as possible. It also allows a response by the neighborhood to a desired displacement. If no solution is found for propagating the displacement, there is no use in making the initial displacement. Other solutions have to be found.

3.2.5 An attempt at tabular comparison

Table 1 intends to compare displacement algorithms in a tabular form. However, it is difficult to judge between alternative algorithms as most deal with different cases.

As a recommendation, it is necessary to find which conflict situations are most common. Thus, Section 3.2.4 tries to identify the most relevant situations.

3.2.6 Distinction of displacement problems

It is also attempted to extract a set of different situations that will cover hopefully most displacement problems in the AGENT project. For the AGENT-project, we make the assumption that linear features (mainly the transportation network) are treated with priority. This means that, after having eliminated all objects that are of no immediate use, generalisation starts by symbolizing roads at their final size. Linear features trigger displacement, but are not displaced themselves. There are no other objects that will make roads move (this excludes triangulation marks).

It is suggested that this sequence will be useful, as i) it strongly facilitates the process, ii) it is in strong correlation with the partition approach, where the line network builds the basis for a partition, and iii) for the purposes of most maps the transport network has higher priority than other feature classes.

These observations lead to a distinction of 4 displacement scenarios, for which displacement functionality must be available (see figure 2 and 3):

1. Displacement of linear features

2. Propagation of linear displacement in the neighborhood

2a. In open neighborhoods (small conflict potential)

2b. In closed neighborhoods (high conflict potential)

3. Displacement due to change of geometry

4. Displacement of objects due to ‘natural’ proximity conflicts (Refinement)

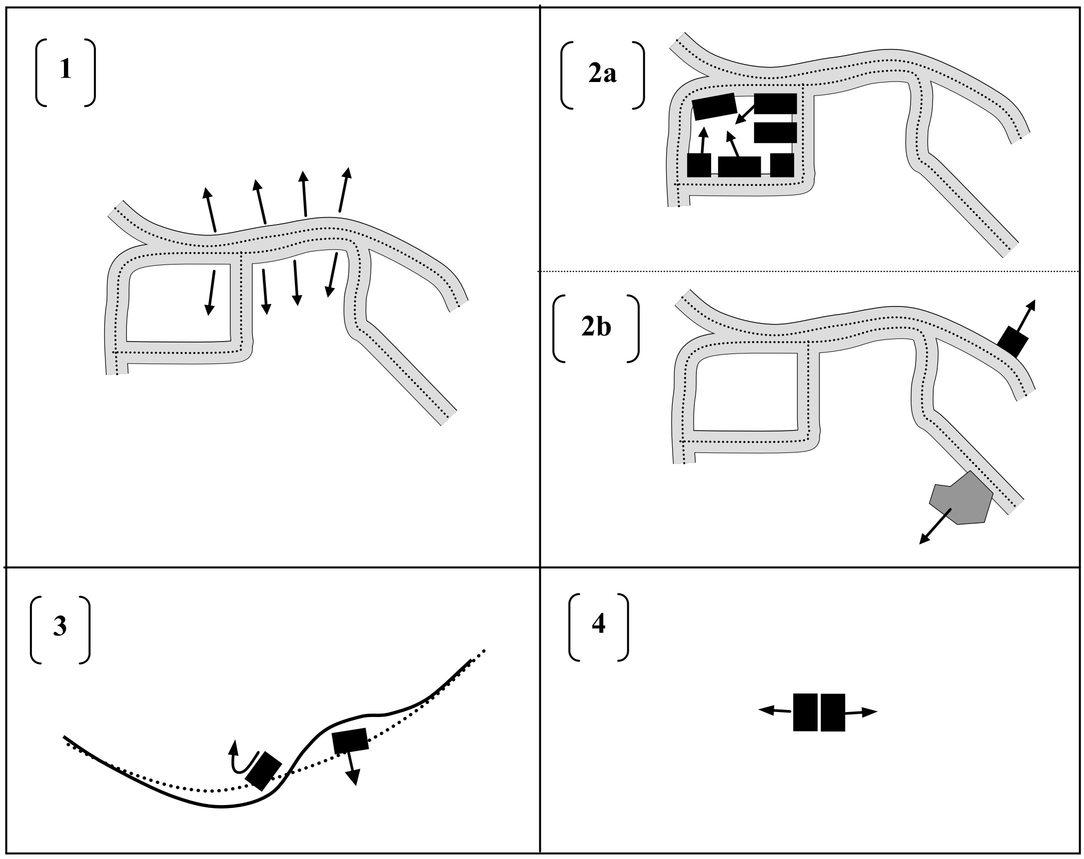


Figure 2. Displacement Scenarios

1: The increased symbol width results in readability (proximity) conflicts between lines. An algorithm needs to determine the new positions and geometries of these lines.

Once the new positions are determined, the neighborhood has to be adjusted for this new geometry. Therefore algorithms of class 2 must follow this action.

2: The change of symbolization width or the displacement of a line results in the same effect for the environment: the information needs to be displaced with less space. Therefore, one can think of a deformation of the neighborhood. As mentioned above, there are propagations within areas with low conflict potential (2a) or within areas with high potential (2b). For the latter, special attention is necessary as space is rare and the displacement vectors need to be chosen wisely to avoid subsequent interference with other buildings.

1 and 2: As mentioned, every algorithm of type 1 needs a 'postprocessing' of type 2 to keep the map in a stable position. Note that this subsequent correction may fail. It is therefore necessary to include a feedback mechanism: if the propagation fails or is only valid when violating other conflicts (every problem could be solved by deleting all participating object!), it has to be asked whether this displacement was really necessary.

3: As in 2, this kind of displacement is a correction of the neighborhood due to change of (linear) geometry. However, it differs from 2, as: i) the width of displacement can be bigger, ii) the displacement direction can change heavily along the line, and iii) it is rather a translation and adaptation of individual objects than a deformation. Therefore it is not known if such conflicts can be treated with the same tools, as is present for situation of type 2b in the next section. (See recommendations).

4: The space between objects can become too small, without any change of symbolization or geometry, just by change of scale. Such problems require very little correction (or typification and elimination operations).

In the recommendation section, for each situation an algorithm is suggested. The recommendation of several algorithms to translate one operator is hereby not extravagant, but rather, reflects the complexity of the displacement process. In satisfying several cases the process becomes significantly more tractable with the choice of a selection of different tools. In particular, the process of choosing parameters will benefit from this decomposition.

3.2.7 Recommendations

3.2.7.1 Algorithms for displacing linear features (type 1)

For solving proximity conflicts between lines, the algorithm by Nickerson-Freeman is proposed. As proven by the IGN, the algorithm succeeds in making adequate generalizations. The algorithm is well known and parameters are easy to set. Few computing resources need to be allocated for this method. The IGN has enriched the algorithm by a function to correctly reconnect segments after displacement.

For subsequent improvement, Burghardt's snake concept is recommended. There is potential to treat more than two lines simultaneously and also the ability to add constraints for intersection. However, at this point the approach is not yet tested and sufficently robust for integration.

Beyond the algorithm: the problems

The Nickerson algorithm itself will not ensure good results. The following questions to improve the results need to be answered:

- What describes a *conflict zone*? It is not sufficient to only concentrate on those arcs that interfere directly. The character of entire lines needs also to be respected (see characterisation below). The definition of conflict zones helps to limit the complexity, describing a displacement order and the displacement direction.

  It should be remembered that conflict zones of roads are not managed by partitions, defined in DD4, but stand perpendicular to them.

- What describes the displacement *order*? With two lines, the question is to know which line to move. With more than two conflicting lines, a displacement order also needs to be defined.

  This problem could be avoided using optimization techniques that deal with several lines at once. However, robust tools for such tasks are not yet available. Though, the algorithm by Burghardt points in the right direction.

- What describes the *division of displacement magnitude*? A conflict could be resolved by displacing both lines, or by transferring the dilation to one line only. This problem is arguably less important, as displacement usually only deals with small magnitudes.

- Do *intersections* need special treatment?

- How is the displaced segment continued (propagated) such that it adequately falls back to the original geometry? At the IGN, an algorithm for this propagation is available (Lecordix and Barrault, undocumented), but this works only interactive mode, thus further improvements need to be found.

Of note is that most of these problems are closely related to the line shape. One possible way of approaching these problems is to calculate a '*deformation index*' for each line, quantifying its resistance against a deformation and displacement. This can only be achieved by characterising the line. In an advanced stage, this deformation index could also integrate the compulsions of its neighborhood.

3.2.7.2 Propagation of linear displacement in the neighborhood (type 2)

3.2.7.2.1 Algorithms in closed areas (high object density) (type 2b)

To solve the problems for allocating buildings in a reduced space - without running into convergence problems - Ruas's algorithm is recommended. The algorithm can be integrated as a big monolithic unit (integrating also a lot of measures), or extractions could be implemented as core algorithm, with the rest of the method integrated as measures and procedural knowledge. The first approach has the advantage that good results are ensured but it somewhat contradicts the agent philosophy. The second approach also poses many problems to the agents, but allows a better control of the process.

Note that using a 'global' method for a whole area does not obviate the need for a detailed evaluation of the result, as a global optima is searched for which might not necessarily solve all (initial) conflicts.

3.2.7.2.2 Algorithms in open areas (low object density) (type 2a)

Two methods that could handle these problems are proposed:

1. As the idea of deformation is appropriate to this kind of problem, the algorithm by Hojholt is recommended. Besides its intuitive model, the method has the advantage that it is robust (introducing no topological inconsistencies) and that all kinds of objects can lie in the deformed zone (including polygons and other line objects). However, it is note clear that this is not a case of using a cannon to shoot a bird (to use a Swiss saying, expressing that the method is much too complex for the relatively 'easy' task). There is a need for clear problem delimitation, and the CPU-resources may go beyond what is available. The method also deforms the space, recognizing neither important shapes and structures, nor new proximity conflicts. Therefore shapes can be distorted, alignments diffused. A careful analysis of the result is necessary. Further research in the consortium is being undertaken to include more cartographic constraints in the method.

2. As an alternative, a vertex-based method is proposed using a radial approach to detect a displacement direction (see Mackaness) with a linear decrease of magnitude. This method has the advantage that it can be easily adapted to other problems, thus made more flexible. However, the method also has disadvantages. It does not deal with all feature types and is less robust.

For the prototype, the use of Mackaness's method is recommended. It offers a flexible method that can be adapted to all kinds of problems. For later improvement, the Hojholt method is recommended.

Beyond the algorithms: the problems

As in the case of road displacement, *conflict zones* need to be detected. In this case, this means the 'empty' space has to be determined. Such constraining zones are also needed as a pre-requisite to the Finite Element Method: Where a closed polygon is required to delimit the deformable space for the method. The FEM has to compute the stressees in the deformed area when computing new object positions. These stresses could also be used to characterise the solution: high stresses indicating new problems. A feedback to the displacement triggering process could be provided by this and an unresolvable configuration could therefore be avoided.

3.2.7.3 Correction with displacement due to change of geometry (type 3)

Knowledge is limited about this kind of displacement and very difficult cases, where complex relations between both sides of the altered road need to be maintained, can be conjectured.

It is recommended to try to resolve these problems with the methods already suggested. Experimentation will show how far these methods are suited, and what additions are necessary.

As it is hard to preview the necessary processes associated to a change of geometry, a processing sequence for each situation cannot be defined. Therefore, it is likely to be more appropriate, to treat this as a type 3 problem with a global optimization method. There is also potential for a FEM to handle this kind of problem.

3.2.7.4 Remaining situations

As classified in Section 3.2.3, solutions for 'natural' proximity conflicts are necessary. Moreover, it must be remembered that not all situations are yet classified using our proposals. However the authors believe that these remaining problems can be resolved using one of the methods already suggested (Mackaness or FEM). These additional cases are not yet specified, as is better to start with the well-defined problems, hence increasing a knowledge of algorithms and problems and subsequently addressing the completion of a simple classification at a later date. Eventually, an improved version of Harrie's method or the FEM could cover all these problems.

To summarize (see also figure 3), the implemention of the following is proposed:

- Nickersons's algorithm: lines vs. lines, lines vs. rigid objects

- Ruas's algorithms: Rigid objects in areas with high conflict potential

- Mackaness (vertex-based) method: Adaptation to all remaining cases

For later improvement, further study of the Finite Element Method is recommended.

As can be seen in figure 3, the focus of the classification and of the choice of algorithm is on situations, where the increased road symbol-width is triggering displacement. On the one hand, this is the main source of displacement. On the other hand, many of the remaining conflicts can be transformed to these situations, once they have been set in place. Therefore concentrating on these cases is recommened.

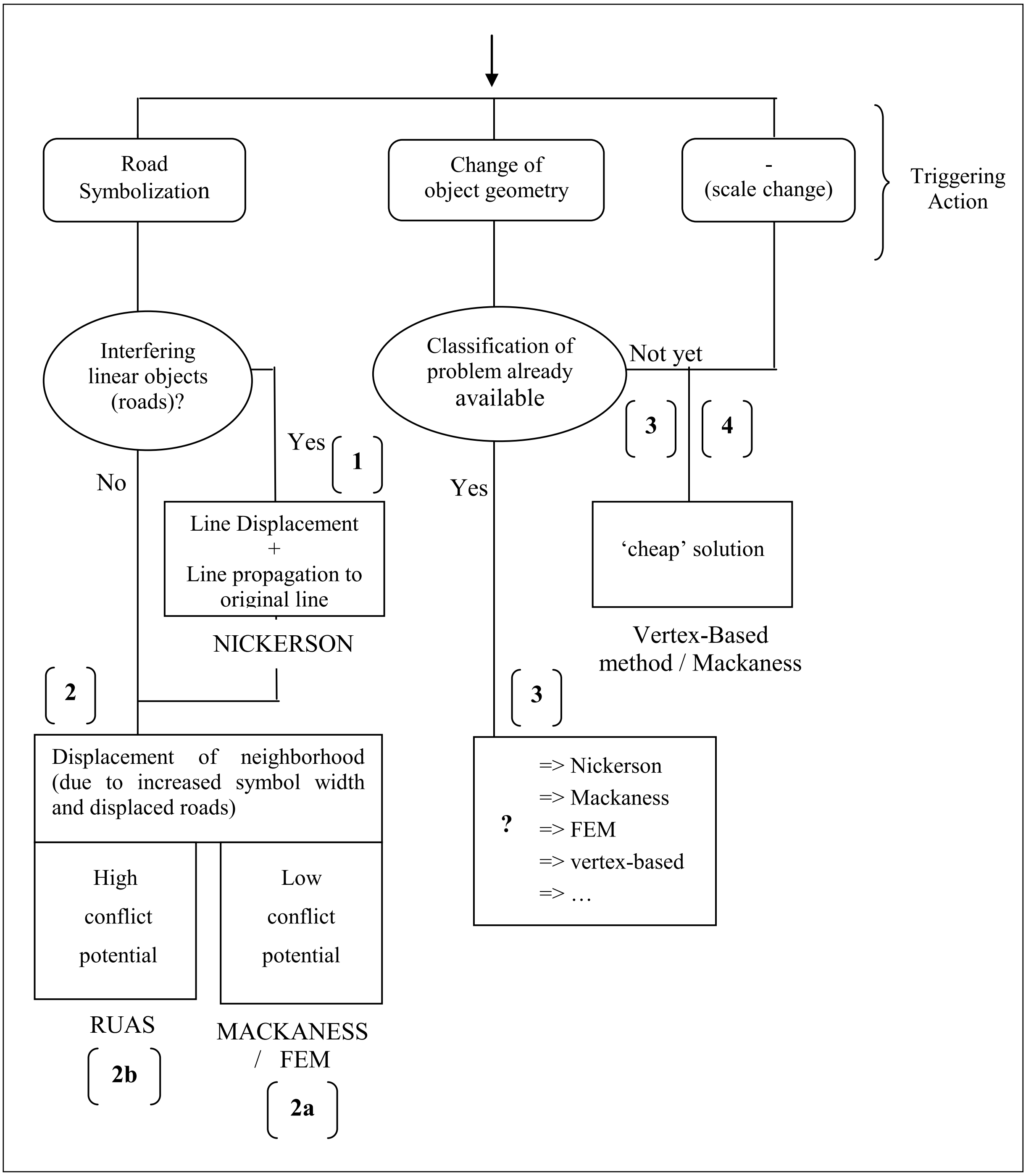


Figure 3. Classification of displacement algorithms

		Mackaness	**Nickerson**	**Baejis**	**Ruas**	**Burghardt**	**Harrie**	**Hojholt**
General	**Example in paper**	point set	line vs. Line	various object classes	Set of buildings in road partition	line vs. Line		Set of buildings in road partition
	Quality of results	+/0 test with simple example	0	0	+	+ only one example available	+ only one example availeable	+/0 only one example available
	Parameters	+	+	+	- many	- weighting parameters	- (?) weighting functions	0/- applied forces
	Ease of implementation	+	0	-	0	-	-	-
	CPU-time and storage	+	+	0	?	-	0/-	0/-
	Robustness	-	+	?	?	+ (?)	?	+
	AGENT-philosophy	0	0	+	+	0	0	-
Object types	**rigid <-> rigid**	+	0	+	+	-	+	+
	deformable -> rigid	0	+	0	+	-	+	+
	rigid -> deformable	-	0	-	-	0/+	+	+ (?)
	deformable <-> deformable **Propagation in object itself?** **(not necessary for rigid obj)**	-	+ - (see remarks)	-	-	+ +	- (?)	0 (?)
Propagation	**Propagation in neighbourhood?** **(to what type of objects?)**	+ (points only)	-	0 (all vertices)	+ (buildings only)	-	+ (all types) (?)	0/+ (all types)
	characterised propagation?	-	-	-	+	-	0/+	-
	avoid new conflicts in propagation? **(approach in propagation)**	-	-	+ (iterative approach)	+ (iterative approach)	+ (?) (one step)	+ (one step)	+ (one step)
	resolve all conflicts in a region	-	-	0	+	+ (?)	+	- only initial conflict resolved
Remarks		Simple and clear idea for small problems	Propagation method available: see IGN [Lecordix, Barrault, not documented]	Missing a lot of cartographic constraints for good solutions	Method consists of several measures and algorithms, which can also be of value individually		If all constraints could be modelled by mathematic terms, method very powerful	Only initial conflict are resolved. However, propagation does not include new problems

- **- bad result**
- **+ good result**

3.3 Aggregation

3.3.1 Definition

The principle of aggregation is to represent a group of objects with simplified composite representation. Two types of aggregation can be distinguished:

1) Aggregation of several objects into one object this is termed amalgamation .

2) Aggregation of several objects into a new group of objects this is termed typification.

Analysis of these two types of operator are considered separately in the next two sections.

Aggregation represents the fusion of a group of objects into one or more objects. For example a group of buildings may be amalgamated into a single block. The process requires two sets of rules two define its operation (Molenaar, 1998).

Linkage rules - which define the spatial relations between objects that must exist for their aggregation.

Aggregation rules - which define the, usually semantic, relationships that must exist for aggregation.

Linkage rules include topological and metric relationships such as;

- Adjacency - two objects must share a common link if they are to be aggregated,
- Touching - two objects must share a common node if they are to be aggregated, and
- Containment - one of the objects must be contained by the other if they are to be aggregated
- Proximity - two objects must be disjoint but within a certain distance of each other for their aggregation. This distance is often termed epsilon (ε).

Clearly, a combination of these rules may be defined for the aggregation. Typically, the implementation of linkage rules is through a clustering mechanism, for example , a minimum spanning tree (Regnauld, 1998).

Aggregation rules include relationships such as;

- Class - two objects must be of the same class or share a common named super-class (inheritance) to be amalgamated,
- Functionality - two objects have a common functional role (part of or aggregation), e.g. farm buildings and a farm fields are a common unit farm.
- Geometry - one of the objects must be smaller than a certain size
- Structural - a group of objects form a common geographic or perceptual structure
- Association - an object or group of objects is associated with one or more other objects

Again, it is conceivable that more than one of these rules is defined.

3.3.2 Purpose

The principle of aggregation is to preserve semantic or structural properties of the distribution of a group of objects but to simplify its representation over the original footprint. Aggregation may be performed for a number of reasons;

- When one or more objects are too small to be represented individually the objects may be enlarged. This enlargement of objects may lead to an impression of over-occupation in an area.

To counter this, if are adjacent to or in close proximity of one another aggregation of the objects into a composite representation may be performed. Generally, it is attempted to maintain the structure and distribution of the group following the operation.

- When the density of objects within an area is high, resulting in conflicts such as overlapping symbology, the objects may be aggregated into a simplified representation. This may be performed with a view to reducing the overall impression density, for example if the objects have been enlarged It may also be performed to enhance the impression of density, for example, through the in filling of an urban block into a single built up area.
- When the semantic level of detail of the database is made more abstract (database or model generalisation), on account of scale or theme, objects may be aggregated to produce more abstract entities.
- When it is wished to enhance or exaggerate the impression of density distribution of information across a map, either for reasons of enhancing communication or to maintain consistency in representation amongst different parts of the map, aggregation may be performed. This operation needs to be controlled at a strategic level, where such gestaltic issues may be considered. Regnauld (1999) discusses this issue in an urban context.

A review of the algorithms for the two forms of aggregation are hereafter presented. Amalgamation is presented first followed by typification.

3.4 Amalgamation Algorithms

3.4.1 Definition

Two types of amalgamation operations exist; amalgamation and combination. Amalgamation operations combine objects to create a new composite object of the same dimensionality. The single composite object produced should maintain the structural and geographic information conveyed by the original group. The change in nature that occurs because of this operation will depend on the feature class. As a general rule though, feature classes that are represented by mainly single disjoint area will change their nature with amalgamation and features that are represented by patches will retain the same nature. For example, Buildings amalgamated will become a more abstract representation, a sub-block or a built up area, however forest patches amalgamated will retain the same semantic identity of a forest patch. Combination creates single entities of a higher dimensionality than the original ones. This compares with the operation of collapse, which results in the representation of a group of objects with an object of a lesser dimension, for example, the symbolisation of a city with a point. For the purpose of the project discussion of combination and collapse is withheld, the reader is referred to DD1 for more details on these operators.

3.4.2 Purpose

The use of amalgamation can occur for four main reasons;

1) When one or more objects of a group, which are adjacent to or in close proximity of one another, are too small to be represented individually the objects are amalgamated into a single representation. The object instantiated by the process may often represent a more abstract entity of a different class to the original objects.

2) When the density of objects within an area is high, resulting in the violation of constraints, the objects may be amalgamated together into a single, usually more abstract representation. For example, a group of buildings may be aggregated to form an entity 'Built-up Area'.

3) When the semantic level of detail is made more abstract (database or model generalisation) on account of scale or theme, objects may be amalgamated to produce a more abstract entity.

4) When it is wished to enhance or exaggerate the impression of density of information across a map either for reasons of enhancing communication or to maintain consistency in representation amongst different parts of the map, amalgamation may be performed.

3.4.3 Algorithms for connected areas

3.4.3.1 Feature types

Connected areas use the shared primitives of a link-node data structure to generate space-exhaustive areal blocks. They consist two distinct elements, the area and the boundaries. The spatial and topological relationships that influence the amalgamation process are usually adjacency, touching and containment. However, some feature types will also require the more complex operation of disjoint amalgamation between areal blocks of objects, described in section 3.4.4.

Examples of feature types termed connected areas are:

- Urban blocks and sub-blocks
- Urban districts
- Land cover units
- Forest parcels
- Administrative or ownership boundaries
- Land parcels, for example, demarked by hedgerows or drainage

Amalgamation most commonly occurs for two main reasons; an element is too small or two adjacent elements hold the same information, for example, because a change in the class of one or both the objects. The process may also triggered by the removal of a boundary segment or the removal of a feature representing a boundary (e.g. a road separating urban blocks).

3.4.3.2 Procedure

The operation of amalgamation is performed by the union of two or more areas, to remove any links that are used by more than one of the areas. This can either be done topologically using a XOR operation on the two sets of boundary links or geometrically by performing an OR operation on the two areas.

The main differences in algorithmic processing are concerned with how areas to be amalgamated are found. The differences occur according to whether the primary entity of concern is the boundary or the area. Essentially, this could be described as a selection operation, though this distinction is not felt important here.

If the primary entity is a boundary segment, the information stored on the supporting links and nodes is used to find the areal units to be aggregated. Ruas (1999) uses this approach in the selection and elimination of a minor streets through the amalgamation of urban block partitions. The algorithm is applied when the symbolisation of both the roads and the buildings causes conflicts and there is insufficent space to reorganise the buildings or where the streets are too small to be represented. The links and nodes of the streets then provide the information about which city-block partitions they support and these are then amalgamated.

If the primary entity of concern is the areas, there are two possible modes of selection for amalgamation. The areas could be selected in a prior operation and the amalgamation performed or the selection and amalgamation is performed simultaneously.

Peng and Muller (1996), use the area as the primary concern for the amalgamation of city-blocks that are too small. However, whilst area is used primarily, since the nature of problem is areal, they also place important emphasis on the boundary segments of the areas, which is treated as a dual problem. The method they use is a dynamic decision tree to embed the rules for classification and aggregation. Three rules are observed;

1) Road connectivity must be maintained, the road name is used to determine the connectivity of individual road segments.

2) Amalgamation of a city block should be performed with its smallest adjacent neighbour, subject to the non-violation of rule 1.

3) Map boundaries must not be broken.

The results reproduced in their paper are visually pleasing and have potential for use in the project at scales where city-block become too small for representation. However, the method to determine road connectivity based on semantics is problematic, since this information may not exist and even if it does exist is not sufficient to characterise the contextual properties of a street network. This could be easily improved by the addition of character information on the links, for example by using Thompson and Richardsons' (1999) algorithm for network selection based on the principle of 'good continuation'.

The amalgamation may be required because of a change in the semantic level of information, or because an area is too small. For example, in the generalisation of land use an area may be too small to hold any information relevant to the map at a new scale e.g a very small hydrologic unit within a forest maybe amalgamated to the forest, since the information is no longer significant. Edwardes (1999, undocumented) uses this approach for the amalgamation of categorical maps driven by a change in class or the need to remove areas less than a threshold value. The two algorithms use slightly different approaches to the problem.

The algorithm for class driven amalgamation uses a depth first recursive search. For each object not yet included in the solution, the algorithm finds the object's adjacent neighbours. This enforces the linkage rules. A filter is then performed on this set of objects using the aggregation rules to determine which of the neighbours are to be amalgamated. For each neighbour in this filtered collection of the process is then recursively called until no more objects are found that satisfy the aggregation rules. At each recursive call the current amalgamated area is passed by reference as an input to the recursive function and this is then combined with the newly found area. Hence, the output of the entire search is the entire amalgamated area.

The algorithm for geometry driven uses an iterative search. All objects that are too small to be represented are selected into a set. For each object in this set the largest adjacent neighbour that satisfies the conditions of the aggregation rules is found and amalgamated with the area which is too small. If no neighbours satisfy the aggregation rules then the object is amalgamated with the largest neighbour. The largest neighbour is used so that the amount of error incurred through the amalgamation is minimised. If this area is also in the set of too small objects it is removed. Following the amalgamation the newly created area is measured. If this is still too small it is then added back into the set of too small objects and the processing is iteratively continued. Currently the danger of this processing is that contextual implications of the elimination of objects that are too small is not considered. For example, it is not tested whether an object is part of a cluster of other disjoint objects that should be preserved and hence generalised by another means. In order to avoid the removal of small areas, some meso control could analyse the distribution of areas in term of size and type in order to guide the amalgamtion process and to preserve some thematic ratio.

Currently the application of these algorithms within the project is limited since they have been defined for feature types not currently being considered. However, there is potential for their use to create district organisations by the amalgamation of urban blocks.

3.4.4 Algorithms for disjoint areas

3.4.4.1 Feature types

Disjoint areas are any groups of areas that are not connected, but within a defined proximity of each other. This proximity is often termed epsilon (ε). The distance of proximity to allow amalgamation will usually be the less than the minimum perceptible distance that can be represented. This is commonly measured between a vertex and a vertex, a vertex and an edge or the projection of the perpendicular of an edge to an edge. The operator will usually result in an area that is greater than the sum of the areas of the group of objects being processed, due to the addition of area between the amalgamated objects. Two types of operation can be distinguished; Amalgamation to a composite object and amalgamation into a district or region. In the amalgamation of objects into a single object, the composite will be located across the original footprint of the group of the objects and share many of the structural properties of the group. This type of amalgamation is used to preserve the distribution of different groups of objects within an area and compares well with the objectives of typification. The principle consideration in choosing between these two operators is how much density of information it is wished to convey. For example, a city could be composed of three zones of density. In the center amalgamation to built-up areas could be mainly used to convey the highest density. Between the centre and the outskirts medium density could be conveyed by the use of mainly amalgamation of buildings into discrete blocks. Typification could then be mainly applied in the outskirts to give the impression of less density. Regnauld (1999) discusses the issue of urban density in more detail. In the amalgamation of a group of objects into a district or region, the region will no longer contain any of the structural information of the group. Instead, the region will communicate semantic information about the group and information about the density of that semantic information within an area. An example of this is the amalgamation of buildings into a single built-up district. As a district subsequent amalgamation will be performed using the operations described in section 3.4.3.

The types of feature class termed disjoint areas includes;

- Buildings
- Lakes
- Forest patches

3.4.4.2 Procedure

In the amalgamation of groups of objects to a single discrete area two principle techniques are commonly employed; amalgamation by displacement and amalgamation be addition of area. In amalgamation by displacement, objects are displaced so they overlap or become adjacent and the geometries combined. The area produced is either the same or less than the aggregate area of the original objects, but the created area is no longer over the original footprints. In the amalgamation by addition of area, the gaps between objects are filled to result in the amalgamated geometry. Here, additional area is generated, but the composite object covers the original footprints.

3.4.4.2.1 Amalgamation by displacement

Ware *et al.* (1995) use displacement and rotation to amalgamate anthropogenic objects that have selected for amalgamation. Figure 4. Describes the process.

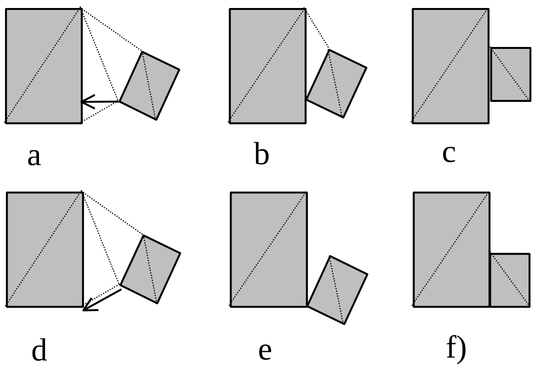


Fig 4 Amalgamation by displacement (Ware *et al.* 1995)

The procedure first computes a Delaunay triangulation network on the vertices of the contours of the objects. The displacement vector is then computed as either the shortest distance between the objects (Fig. 4a) or the shortest edge of the triangulation (Fig. 4d). A cost function is used to decide if one or both objects will be displaced and by how proportionally how much. The displacement is then computed bringing the two objects in contact at a vertex and an edge (Fig. 4b) or at two vertices (Fig. 4e). The rotation operation is then performed to bring the objects together (Figs. 4c, 4f) and the amalgamation is performed to join the geometries. Generally the displacement along the shortest edge of the triangulation provides better results (Fig. 4d) since it does not introduce violation of contraints, such as edges of the contour which are too small.

Regnauld (1998) proposes a method of amalgamation by displacement (Figure 5). Amalgamation in this example is performed by displacing the objects until they overlap. The displacement vector is computed along the minimal distance, either between a vertex and an edge, an edge and an edge or two vertices. This is the distance D0B0 in Figure 4.

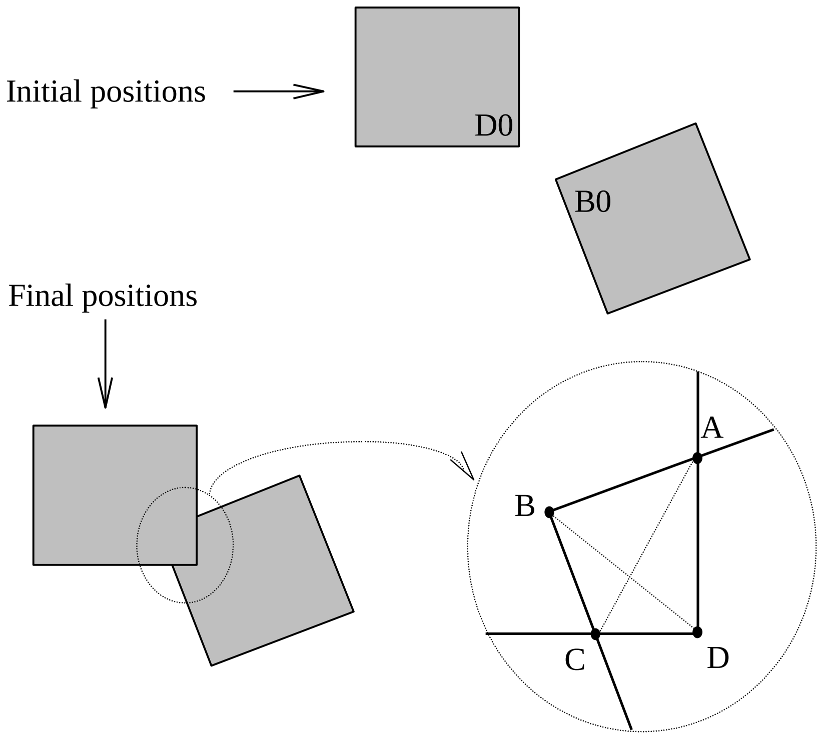


Figure 5. Amalgamation by displacement (Regnauld, 1998)

The displacement is applied to the extent that the width of overlap is greater than the minimum local width. In figure 4 this is shown by the dotted line AC. Again a cost function is used to determine the degree to which each object will be displaced.

3.4.4.2.2 Amalgamation by addition of area

Ware *et al.*(1995) use a Delaunay triangulation network to compute the area that needs to be added to join to disjoint area. Figure 6 describes this process

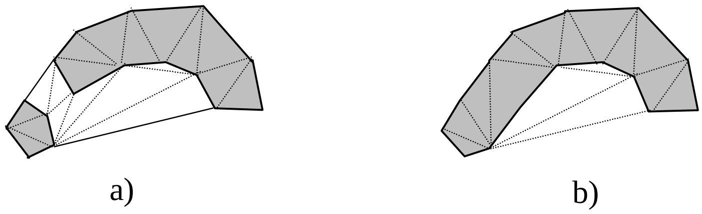


Figure 6. Amalgamation by addition of area (Ware *et al.* 1995)

The triangulation network is computed on the vertices of the contours of the original objects. This discetises the space between the objects (Fig. 6a). Amalgamation is then based on infilling of the area bounded by the smallest edges (Fig 6b).

Regnauld (1998) provides an alternative solution using the convex hull of the group to guide the amalgamation. The operator works on two objects at a time, if more objects are to be amalgamated than this they may be subsequently joined with the newly generated area. The algorithm is described in Figure 7.

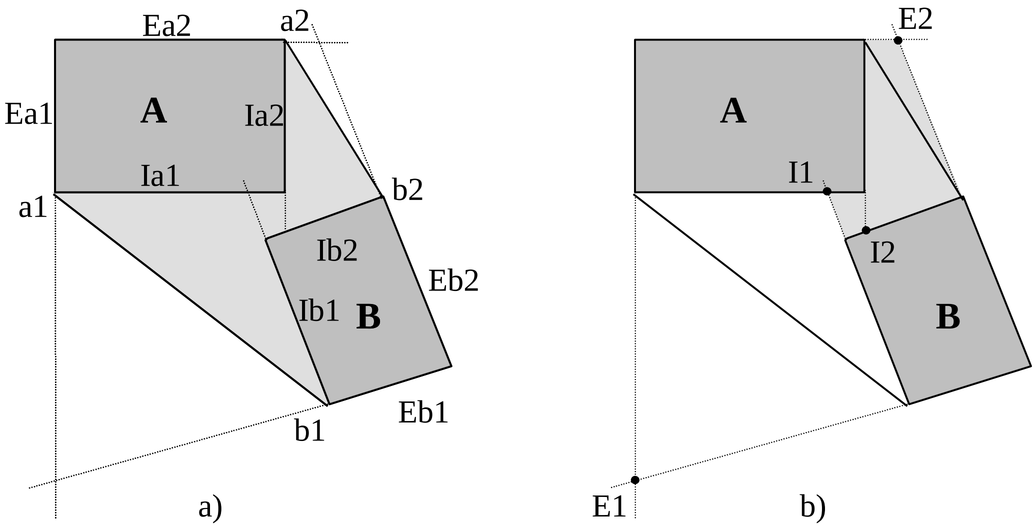


Figure 7. Amalgamation by addition of area (Regnauld, 1998)

The process works by first finding the convex hull of the group. The aim of the algorithm is then to replace the edges of the convex hull with edges created by prolongation and intersection of the original edges of the objects and retain the perceptual structure of the group. In Figure 7, to replace the convex hull edge a1b1, two candidates exist; the intersection of the internal edges (Ia1 and Ib1 in Fig. 7a) I1 in Fig. 7b, or the intersection of the external edges (Ea1 and Eb1 in Fig. 7a) E1 in Fig. 7b. The same principle applies to the replacement of a2b2. Combining these candidates gives four possible solution. The internal candidates are chosen in preference, but in cases of topological consistencies or violation of the constraint for minimum width, the external candidate that minimises the overall change in size may be used.

Morphological operators have been used extensively for generalisation and shape characterisation in the raster GIS (Schylberg, 1992; Su *et al.* 1997), though their application in the vector domain is still fairly limited. The operator commonly used for aggregation is *closing*. *Closing* consists of the successive application of two transformation operators *dilation* and *erosion.* Figure 8 describes theses operators.

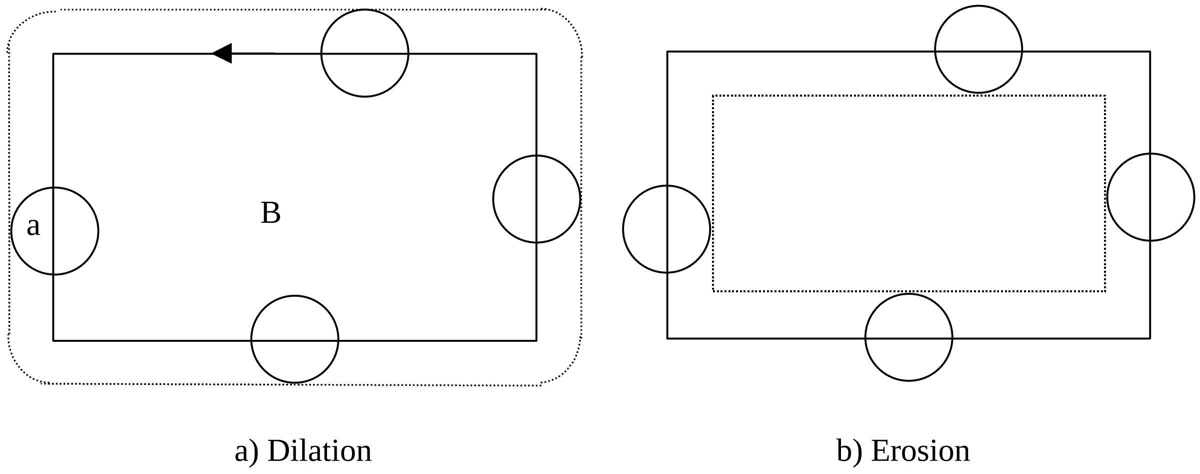


Figure 8 Morphological operators.

For dilation, the principle involves computing the Minkowski addition for a structuring element, labelled a in Fig. 8a, and an object, labelled B in Fig. 8b. The structuring element may be any shape, though commonly for aggregation it is a disk. The structuring element is translated along the vector defined by the contour of the object, this is shown by the arrow. Essentially, the operation generates a buffer outwards from the object. The operation of Erosion is the same except it uses Minkowski subtraction. The areas created by the operators are shown as dashed.

Closing represents the application of the dilation operator to generate a new contour followed by the erosion operator on this contour. When the operator is applied to a group of disjoint objects it can be used to amalgamate them across their original footprints. Fig 9 describes the process.

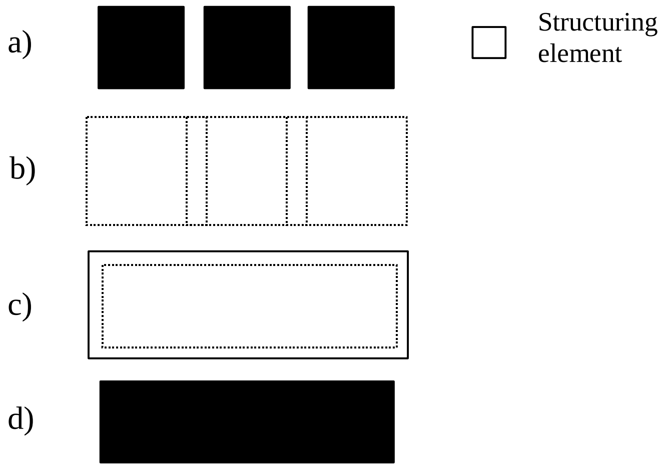


Figure 9 Amalgamation by closing

Fig. 9a shows three objects within proximity for amalgamation. The structuring element used in this example is a square, shown on the right. Fig. 9b describes the overlapping geometries generated by the dilation. In Fig. 9c, the geometries have been combined and the erosion operator applied, the new geometry is shown by the dashed contour. The erosion of the amalgamated geometries returns the amalgamated and enlarged geometry to the original footprints but retains the area created between the objects, this is shown is Fig. 9d. A desirable or undesirable side effect of the operation is that it removes any concave detail from the objects that is smaller than the structuring element.

Ormsby (1999) uses dilation of buildings for generation of a graphic boundary of a city by amalgamation. The process works by simply applying dilation, using a disk structuring element, to each of the objects and OR-ing the geometries together where they overlap. Selection is then performed to remove isolated geometries that are too small to represent a town. Subsequently, any holes are removed from the generated areas and a small Douglas-Peucker simplification is applied. This gives the generated contour a desirable anthropogenic appearance. The full closing operation is not performed since it is wished to create an expanded area that intersects with other feature classes of the town, such as the road network. However, currently the operation can violate several geometric, such as minimum edge length and minimum local width, and topological constraints, such as, self-intersection. A better solution may be to use the full capabilities of morphological operators to perform the boundary simplification and ensure against self-intersection.

For the generation of built-up regions by amalgamation Glover and Mackaness (1999) polygonise the road network and use semantic information aggregated to the polygons to control block filling. The issue here is one of ensuring consistency in the representation of density and this is achieved by the use of a scale band rule base to decide when to apply different types of aggregation operator.

3.4.4.3 Recommendations

3.4.4.3.1 Buildings

For the aggregation of groups of buildings into a single sub-block a number of algorithms have been described. Regnauld (1998) and Ware (1995) both describe methods using displacement of two buildings. However, both these methods result in an unacceptable change in the structure of the original group. The may also both result in the violation of micro constraints such as level of detail. For these reasons they are not recommended for implementation. Regnauld (1998) also suggest a recursive method for amalgamation of groups of buildings by addition of area. The method works best with convex buildings, implying an initial characterisation of the group before deciding if it is appropriate may be advisable. However, the results are good for many configurations of buildings and structure is retained without the side-effect creation of conflicts. Ware et al (1995) also suggest an area additive method for amalgamating areas by LDT. The results seem good, with structure of the group being retained. Whilst, the examples given by the authors are mainly for area patches there is no reason why the algorithm cannot be used for buildings. However, there is a processing overhead involve in calculating the LDT, which must be considered if it is to be applied to large numbers of buildings. Also the boundary segments added are always straight lines from the triangulation and this may be undesirable in some instances. Morphological operators have been discussed for amalgamation of sets of building and these have the potential to be very useful. Of note, is that they will intrinsically remove detail smaller than the structuring element thereby preventing the creation of additional micro constraints. However, there is a lack of knowledge of their application in the vector domain to fully evaluate them. More knowledge is required on the type of configuration they should be applied to, the effects of different shapes of structuring element and the procedural knowledge about sequencing of them. The algorithm of Ormsby et al (1999) for the amalgamation of sets of buildings to generate the graphical area of a city has produced useful results. The algorithm also has a dual purpose of being a functional entity as well as a graphic one. However, as a graphic entity the results generate secondary conflicts that must be handled. The process therefore requires more research.

3.4.4.3.2 Urban blocks

Two types of amalgamation operator are required for the project based on different goals. There is a need for amalgamation in response to urban blocks that are too small and a need for amalgamation to preserve semantic district information. Many of the amalgamation algorithms can theoretically perform both operations but there is a lack of knowledge about how to characterise semantic districts and this information needs to be formalised before valid algorithms can be implemented. This problem is part of the task D3. For the amalgamation of urban blocks the most important consideration is whether the algorithm preserves the road network structure. Both Ruas (1999) and Peng and Muller (1996) have mechanisms to perform this. However, the approach of Ruas is more locally based, being more with the philosophy of the project, and does not rely on the presents of semantic information to retain the network structure. Her approach relies on the partition to preserve the network structure, though this could be improved by the application of the continuity principle.

Figure 10. Summarises the different algorithms with regard to their usefulness to the project and makes recommendations for their implementation.

Algorithm	**Input**	**Output**	**Pros**	**Cons**	**Desirability**
Regnauld (1998) Area additive	Buildings	Sub-block	Retains structure Creates no additional conflicts	Works best with convex buildings, less stable with no convex	Desirable Now
Regnauld (1998) Displacement	Buildings	Sub-block	Easy to implement Creates no additional area	Can cause additional conflicts Structure of grouping is not well retained	Not Desirable
Ware et al (1995) Displacement	Buildings	Sub-block	Creates no additional area	Can cause additional conflicts Structure of grouping is not well retained Significant overhead in processing LDT	Not Desirable
Ware et al (1995) Area additive	Disjoint areas	Sub-block	Retains structure of group	Segments added are always straight lines. Significant overhead in processing LDT	Desirable Later
Morphological Operators	Buildings	Sub-block	Retains the structure of the groups Should not create additional conflicts	Not sufficently researched - problems defining sequencing and structuring element. Overhead in processing could be significant with a complex structuring element	Desirable Later
Ormsby *et al. (1999)* Morphological Operator Hybrid	Buildings	Urban area	Easy to implement Fast	Can cause additional conflicts Needs further research	Desirable Now
Ruas (1999) Amalgamation by Selection	Urban Blocks	Districts or Urban Blocks	Preserves structure of the road network Conflict driven	Could cause semantic constraint violations	Desirable Now
Peng and Muller (1996)	Urban Blocks	Districts or Urban Blocks	Conflict driven Easy to implement	Doesn't preserve structure of the road network sufficiently Could cause semantic constraint violations	Desirable Later
Edwardes (1999)	Connected areas	Connected areas	Implemented Easy to use	Needs further research on semantic constraint violations and consequences of generalisation Designed for feature types not in project	Desirable Later

Figure 10. Summary of the different amalgamation operation and their suitability to the project

3.5 Conclusion

The preceeding discussion has reviewed and evaluated theory, techniques and algorithms for amalgamation. To conclude, there are still areas where further research and development needs to be made. In particular there is a need for better tools for spatial analysis with which to characterise context and generate better rules for aggregation. Examples of this are, the characterisation of the road network to provide information about for the amalgamation of urban partitions and the characterisation of configurations of buildings to make choices between the different amalgamation to sub-block algorithms.

Additionally, for the maintenance and communication of district information there is a particular lack in what information to characterise and how it relates to areas across the map.

The recommendations for implementation that have come out of this discussion are;

Regnauld (1998) - Area additive amalgamation for buildings,

Ormsby et al. (1999) - Morphological Operator Hybrid for generating urban areas,

Ruas (1999) - Selection of roads to amalgamate urban blocks that are too small.

3.6 Typification Algorithms

Author: J.F. Hangouët, Cogit Lab, 10 June 1999

3.6.1 What is typification

3.6.1.1 A Definition

Typification is a generalisation operator that aims at ensuring that the configuration of a group is represented in the generalised version of a map. Whilst, *typification* operators often intricately mix the operators of selection and displacement within groups of objects of a same nature, we are more concerned with the nature of the phenomena it produces rather than the process by which this is achieved.

Typification methods can be described under two main classes, according to the kind of configuration to be transformed: methods on linear groups and methods on 2-D groups. It is of note that far more research has been dedicated to the former.

Typification was once the name for the symbolization of a building by template-matching (c.f. (Staufenbiel 73), (Meyer 89), (Powitz 93)). Usage of the term has changed since the late 1980s and “typification” now applies to the generalisation of groups of objects. The following definition sums up the different definitions or descriptions given in (McMaster 91: p.36), (McMaster-Shea 92: p.62), (Ruas-Lagrange 95: p.88), (Hangouët-Regnauld 96: pp.224-225), (Hangouët 96: p.224), (Regnaud 97: pp.1396sqq.), (Ruas-Mackaness 97: p.1391), (Regnauld 98: pp.115sqq.):

> ***Typifying*** *consists in bringing into being a cartographic representation, where the identified distribution is preserved, features of a same nature that happen to be grouped locally by some identifiable geographical process* (Hangouët-Lamy 99)

Comments

What is transformed is a group of features of a similar nature that occupy a circumscribed area (current transformation algorithms do not manage for the possible presence of features of other kinds amongst the features of a group; or, in other words, the different features are supposed to be non-constraining).

The group is organised by an identification of an underlying geographical process, the very process that is envisioned when the group is named and sensed to be important. The group is transformed making an effort not to deface the identified organisation. The variety of geographical phenomena explains why there are different typification methods, each one specific to the kind of group involved.
The typification algorithms presented hereafter mix selection and displacement of components. Selection-only methods complying with the definition above are described in another section of the report, and displacement-only in yet another section.

3.6.1.2 General principles

Why typification is necessary *(this paragraph is reproduced from (Hangouët-Lamy 1999: § 4.1))*

With scale reduction, small objects may no longer be legible if not enlarged. When objects are isolated, enlargement has no side effect. When density is very high, individual enlargements lead to the infilling of an area, which is visually sound and, computationally speaking, easily reprocessed into an aggregated object for subsequent handling. In intermediate densities however, individual enlargements of features lead to an impression of over-occupation: in fact, free 'white spaces' should also be enlarged to be legible — but the enlargement of both objects and spaces is physically impossible. The traditional solution consists in creating room by removing some of the objects and reorganising the remaining objects across the original footing. Essentially, this is what typification is about; lakes harmoniously spaced out, bends obstinately succeeding one another crawling up the steep mountain slope, streets that criss-cross the town, houses in the residential area, all are liable to be typified.

Note that, as with any generalisation operator, typification may also be induced by map purpose (thematic cartography) and not only by change of scale.

3.6.1.3 General procedure

- The group is identified.
- All features' output sizes are simulated.
- The final number of features to be represented is computed.
- Individual characteristics of the original features are measured and their variations from the average or expected values over the group are computed.
- Features are classified based on their characteristics, onto a 'must-be-represented' to 'may-be-discarded' scale that is dependent on the specifications of the required map.
- Important features are selected up to the allowed number, individually generalised, and scattered over the group's original footing.

(All operations are made either explicitly - i.e. they are programmed as such - or implicitly - i.e. they belong with the designer's choice).

3.6.2 A note on the naming of typification algorithms

Two (intuitive) methods have been used for naming typification algorithms:

Approach-oriented naming: The approach used to identify the group to be typified gives the name the typification algorithms (eg. "Gestalt typification"),

Feature-oriented naming: The nature of the components of the group to be typified gives the name to the typification algorithm (eg. "bend typification").

3.6.3 Typification of linear groups

Perceptually, a linear group is can be apprehended as a group whose components visually follow a distinguishable soft curve. In this section, such groups are supposed to be distributed in linear structures, i.e. the features to be transformed can be said to occupy stretches of ground. In cognitive terms this means that they come in succession along obvious and obviously unique perceptual or geographical paths, not in a clutter. This is an intuitive description only, the general mathematical formulation we think of requires somewhat lengthy interpretations of the Voronoï diagram drawn between the features, and, as the subject of a forthcoming article, will be spared here, suffice to say, that Regnauld's segmentation of the Minimal Spanning Tree, itself a graph mathematically deductible from the Voronoi diagram, is an example for buildings.

An important characteristic of linear groups is that current methods first retrieve the order of the components within the group.

3.6.3.1 Buildings : Gestalt typification or structuration

Regnauld (Regnauld 1998), uses a method for typification of linear groups of buildings which he terms structuration. The method uses a Minimal Spanning Tree to elicit clusters of buildings that are visually consistent, hence the name of the method, it then applies the typification operation to these groups of buildings. A Minimum Spanning Tree is defined as a cycle-free graph connecting all elements, whose total edge length is minimal, within the set of all cycle-free graphs connecting all elements.

The Minimal Spanning Tree generated on separate objects elicits "linear" structures ("linear": very much like tortuous boughs on a branch, and the branch itself, are linear). More specifically, it can be segmented into (linear) groups of homogeneous components (i.e. of regular shapes, sizes, interdistances etc.).

The typification is then applied on these groups in order of those that are;

positioned firstly: the buildings which connect the group to the neighbouring groups; the extremities of the group and (if possible) the buildings where the group forks.

positioned secondly (and possibly): buildings whose position along the group suddenly departs from the interpolated succession ("cusps").

positioned possibly: intermediary buildings, placed next to each other, with an interdistance deduced form both the original spacing and the final scale.

Each represented building is individually generalised, with a possible emphasis on its remarkability within the group.

The following is a short summary of the description provided in (Regnauld 1998).

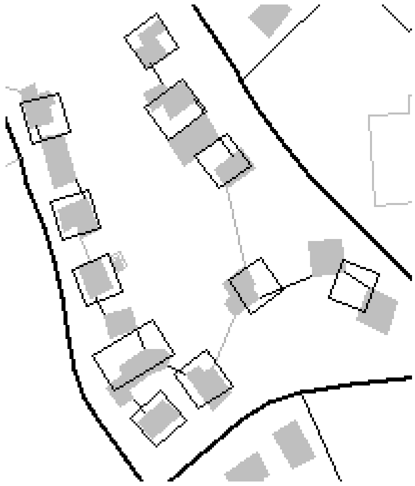
3.6.3.1.1 Method

- **Input**
  - A set of groups of buildings resulting from the analysis of the MST computed on all buildings.
  - The analysed MST (shape, size and orientation characteristics described for each group).
  - Input scale.
  - Output scale.
  - Minimal interdistance allowed at input scale.
  - Separation threshold at output scale.
- **Output**
  - A set of groups that can be represented.

- A set of individual characteristics: mean size of the final buildings over the area, number of buildings exceptionally different from the mean size, exhaustiveness ratio (final number of buildings / initial number of buildings).

- A set of contextual characteristics, computed from the comparison between the MSTs computed on the initial and final configurations.

- A set of land use characteristics: an occupation ratio (cumulated area of buildings / total area of the street-delimited block) and a spacing deviance ratio (effective final interdistance / theoretical final interdistance).



full: original buildings,

edges between buildings: MST
grey edges: between groups
black edges: within a group

contours: buildings typified

Vocabulary :

The MST computed on all buildings is called the "all-MST" in the following.

Each group is structured by the local portion of the all-MST, this portion is called "group-MST".

Structurally speaking, a group is composed of;

- extremity buildings (buildings with only one group-MST edge),
- fork buildings (buildings with at least three group-MST edges),
- intermediary buildings (buildings with exactly two group-MST edges).

A "group tender" is a building of a group connected by the all-MST to a building of another group.

Two buildings are "typically too close" when the minimal distance between them is smaller than the typical distance over the group.

Two buildings are "distinctly too close" when the minimal distance between them is smaller than the distinction threshold.

An exceptional building is a building whose size or orientation (or shape) is remarkably different from the recurring value over the group (cf. II.1-2 for the definition of "remarkably different").

Each group will be transformed quite independently as follows.

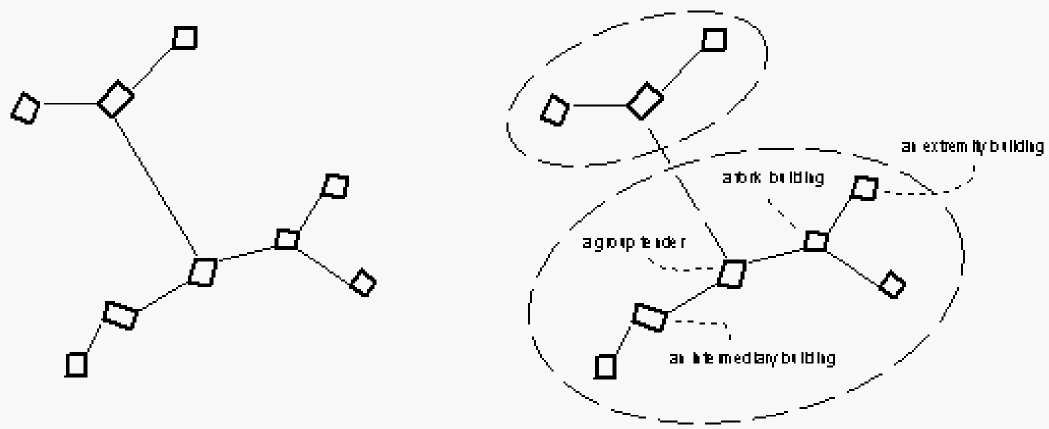


Figure II.1-1a/ All-MST, group MSTs and definitions

- **Transformation**

<u>- Prepositioning of group tenders</u>

Group tenders are imposed to be represented in the generalised version of the area.

A group tender is represented in its original position if it is an intermediary building.

Otherwise it is displaced so that the final distance to the other group (df) in the interval [dmf-DM] is homothetical to the initial distance (di) in the interval [dmi-DM]:

$$df = dmf + (di - dmi) * (DM-dmf) / (DM-dmi)$$

(dmi is the minimal distance allowed between buildings at the initial scale, DM is the longest length of an edge from the all-MST, dmf is the separation threshold at the final scale). The displacement is performed as follows:

If the group tender and its counterpart from the other group are both extremity buildings, either is brought toward its own group neighbour by half the necessary additional distance projected along the direction to the neighbour.

If the group tender is an extremity, and the counterpart from the other group is not, the group tender is displaced by the necessary amount of translation toward its group neighbour.

Figure II.1-1b/ Distancing buildings (geometrical elements and result)

<u>- Positioning of particular buildings</u>

Particular buildings are positioned first. Particular buildings include obligatory buildings and desirable buildings.

Obligatory buildings :

Obligatory buildings include: buildings at the extremities and buildings at the forks of the group.

- Extremity buildings are left in their positions. If extremity buildings are distinctly too close to each other, there are three possibilities:

~ unique or shared erosion is applied if it increases the distance without shattering either building's appearance, otherwise,

~ amalgamation of the two buildings is performed if they are not individual houses, otherwise,

~ one building is eliminated.

- Fork buildings may be slightly translated:

If two fork buildings are typically too close to each other, the procedure described above for extremity buildings is applied.

If a fork building and an extremity building are typically too close to each other, two configurations must be distinguished:

~ Y fork (all angles between successive radiating edges smaller than π) the fork building is allowed to move "vertically down" the Y's leg (the edge not involved in the smallest angle)

If its distance to this extremity is typically too close, the procedure described above for extremity buildings is applied.

~ T fork (one of the angle between radiating edges greater than π) the fork building is allowed to move "vertically down" the T's leg (the edge not involved in the largest angle)

If the fork building conflicts with several buildings, all are amalgamated, except in the case of a Y fork when the fork building conflicts with the "upper" buildings, which is processed as the simple Y-fork case.

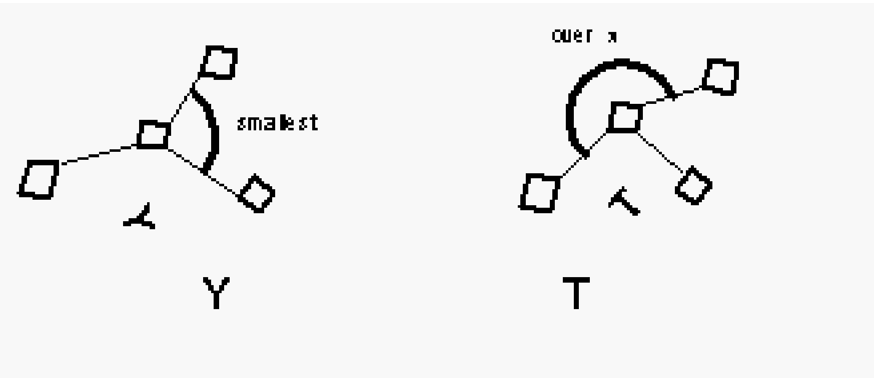


Figure II.1-1c/ "Y-fork" and "T-fork"

Desirable buildings :

Desirable buildings include exceptional buildings and buildings at cusps in the group (a cusp is where the angle between two successive edges is smaller than 90°).

Each combination (one desirable building, two desirable buildings… all desirable buildings) is tested in turn and the non-conflicting possibilities only are selected. These make up a set of possible solutions. Each is processed with the positioning of casual buildings, and the solution where interdistances most approach the theoretical interdistance is preferred.

- Positioning of casual buildings

Casual buildings are positioned into the final group section after section, a section being a branch in the group-MST between two already positioned particular buildings.

Each building along the section is considered in turn.

The building is positioned as follows:

If the group is regularly spaced out: a regular interdistance is expected (RI, computed from the original regular interdistance with the same interpolation used for the prepositioning of group tenders).

If dcf [ie. distance (current building - former building)] < RI

then

move current building toward the next building in the initial group until dcf = RI

check if current building closer to next building than to initial position

in which case remove current building

else

move current building toward former building in the initial section

The procedure continues with next building as current building until the second particular building positioned is met. If the distance between this building and the last building positioned is smaller than the regular interdistance, the last building is removed, leaving a somewhat extra-large hole.

The extra-largeness of the hole is brought back to the expected interdistance:

- by moving the second particular building toward the last building positioned. This is possible only when the second particular building is not a tender-building.
- by shifting all positioned buildings along the final group to harmonise interdistances.

If the group is not regularly spaced out:

If dcf [ie. distance (current building - former building)] < separation threshold

then

move current building toward the next building in the initial group until dcf = separation threshold

check if current building closer to next building than to initial position

in which case remove current building

else

nope (no operation)

The procedure continues with next building as current building until the second particular building positioned is met. If the distance between this building and the last building positioned is smaller than the separation threshold, the last building is removed.

<u>- Representation of buildings</u>

Each building, when it is not requested to take part to an amalgamation or to cancel itself, is represented by individual simplification. Its orientation, notably relatively to the road, must be dealt with carefully, as the translations described above may bring the buildings closer to non-parallel parts of the nearby road.

<u>- Computation of indicators</u>

Together with the final buildings, several indicators are provided in the output of the method, to be used by the generalisation process to take decisions.

- Individual characteristics of the final set:
- mean size of the final buildings over the area,
- number of buildings exceptionally different from the mean size,
- exhaustiveness ratio (final number of buildings / initial number of buildings).
- Contextual characteristics of the final set (*a new all-MST is computed over the final buildings and its segmentation is compared to the initial all-MST*):
- number of different group-connectors (MST edge between homologous group-tenders)
- number of initial groups that happen to be segmented into smaller groups in the final all-MST.
- number of final groups that happen to contain smaller groups in the initial all-MST.
- Land-use characteristics of the final set:
- occupation ratio (cumulated area of buildings / total area of the street-delimited block)
- for each final group: spacing deviance ratio (effective final interdistance / theoretical final interdistance).

- Control

There may remain conflicts between groups in dense and not linear areas.

- Validity through scale

The method is valid as long as obligatory buildings can be represented without conflicting with each other.

3.6.3.1.2 Necessary information on a group

- Constitution

The buildings within a city-block are linked by a MST where the distance used is the minimal distance from contour to contour (not from centroid to centroid).

This all-MST is segmented into groups of homogeneous interdistances and individual characteristics. The full, recursive procedure (over-segmentation, characterisation, regrouping and re-segmenting) is explained in [Regnauld 98]). Only two necessary steps are described hereafter.

<u>- Severing the all-MST at overlong edges</u>

For each edge in the all-MST, and for a node of this edge, the set *S* of “neighbouring” edges is retrieved: the other edges connected by this node and again the other edges connected to these edges. *S’* is computed similarly from the other node of the edge. In the following, *ml* (*ml*’) is the mean length of the edges in *S* (*S’*), and *sd* (*sd*’) the standard deviation of the lengths. The edge is valued with:

$$ol = \max \left[\; lg / (1.2 * ml + 2 * sd) \;\; ; \;\; lg / (1.2 * ml' + 2 * sd') \; \right]$$

All edges are ordered from highest ol (edges whose lengths most contrast with their neighbours) to lowest ol.

The edge with highest ol, if it is greater than one, is removed (which yields two groups).

Neighbouring edges are valued anew and the process is reiterated in each group.

<u>- Characterizing a group</u>

The size and orientation of each building (but other characteristics could be added, such as shape), the median value (m) and deviation (σ) for either measure is computed over the group. Buildings with a measurement *v* lying outside the [m - coef*σ , m + coef*σ] of the measure are marked as "exceptional" for this measure. The exceptional building scoring the highest (*v*-m) / σ "atypicity" ratio is extracted from the group; the median and deviation are computed again and the "exclusion on account of atypicity" is repeated until no building is atypical in the remaining group.

If $h / e > 5$ (h: number of buildings in the remaining homogeneous group over e: number of buildings excluded), the initial group is said to show the characteristic at stake (regularity of size or of orientation), and the characteristic is stored as the median of the measure within the homogeneous group together with a list of the exceptional buildings and their atypicity values.

The buildings in the group are eventually given a global atypicity value (size atypicity + orientation atypicity).

A global atypicity measure is computed for each building in the group as the sum of the atypicity values for the different measures (size, orientation - shape possibly…), and the buildings in the group are ordered on this global atypicity.

- Order

Buildings within a group come in succession according to the edges of the MST.

- Individual measures

~ Size: the initial size of a building, si, is measured as its area. A building's final size, sf, is set as:

<u>if</u> si < smax, sf = smin + (si - MinS) * (smax - smin) / (smax - MinS)

<u>else</u> sf = si

where smin is the minimal size threshold, smax the size above which buildings are no longer enlarged, and MinS the minimal building size in the original buildings.

~ Orientation: the orientation of a building is measured as the weighted mean of the two longest segments found on the points of the contour (the two segments are not allowed to have a common end-point).

3.6.3.2 Buildings: phenomenological typification

The kind of grouping that phenomenological typification is applied to is a succession of buildings that lie between two crossroads and on the same side of a street (Hangouët 1998). The naming of the algorithm is based on geographical principles; the principle being that buildings are commonly found in succession along streets.

The number of buildings to be represented is deduced from the original number of buildings, the original 'black/white' ratio, and the enlargement factor applied to the buildings. The guiding rule is that the 'black/white' ratio must be preserved as far as possible.

Buildings are placed at positions interpolated from their original locations. Whether an original building is represented depends on it being originally closer to the interpolated position and on its remarkability in the group.

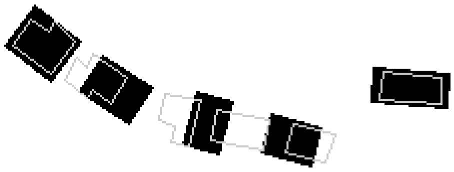
3.6.3.2.1 Method

- Input :

- *f*, enlargement factor to be applied to a building
- *group*, an informed group of buildings (cf. II.2-2 for the necessary information)
- *preferences*, a list of preferences ("buildings with these characteristics must be favoured over other buildings" - cf. II.2-2 for the characteristics that can be taken into account).

- Output :

- a group of buildings



contours: original buildings

full: buildings typified

- Transformation

n is the initial number of buildings.

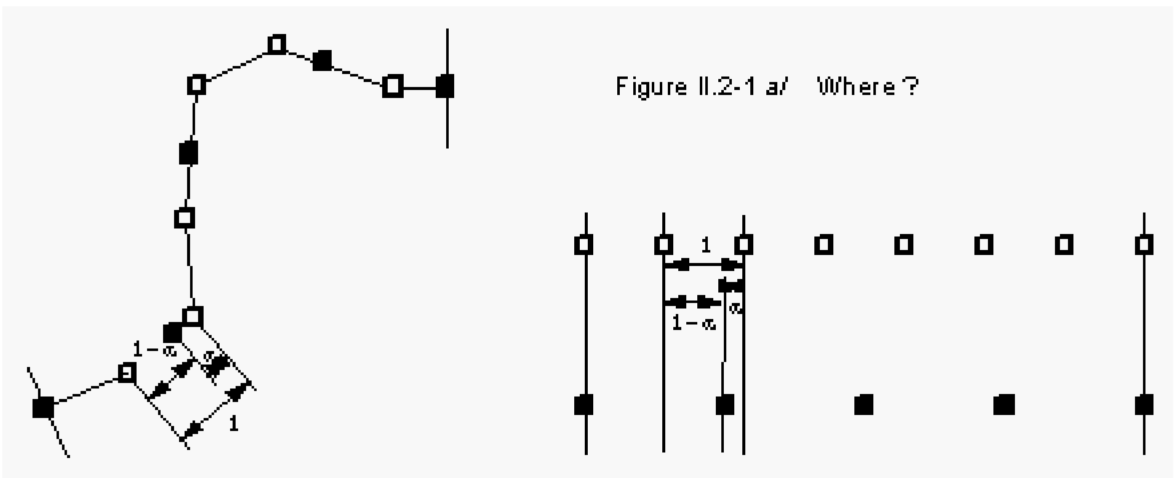
<u>- How many buildings are to be represented ?</u>

m, the number of buildings to be represented, is computed as : m = NI [n/f] (with NI[x] : Nearest Integer to x).

<u>- Where are the buildings to be represented ?</u> (cf. fig. II.2-1 a/)

The final group is represented in the footing of the original, i.e. is made to start and stop where the original group starts and stops.

Each *i*th final position is interpolated between the *j*th and (*j*+1)th initial positions, where j = GSI [$i*n/m$] (GSI[x] : Greatest Smaller Integer to x). The interpolation ratio is computed as if the group was originally evenly distributed: $\alpha(i)$ = FRC [(i-1) * (n - 1) / (m - 1)]. (with FRC[x] = x - GSI [x]). If $\alpha(i) = 0$ then j+1 is brought back to j.



- Which buildings are to be represented ?

The first and last buildings in the row are preserved and represented in their own positions (a far-fetched explanation for keeping those two is because they usually are corner buildings).

The two buildings in the original group that lie straight before and straight after a final position are selected and the one with the most respective remarkable characters is chosen (cf. fig. II.2-1 b/).

(A building is declared *remarkable* for such characteristic when it is *extraordinary* for the relevant measure - cf. II.2-2 - and when the characteristic is specified in the *preferences* of the input).

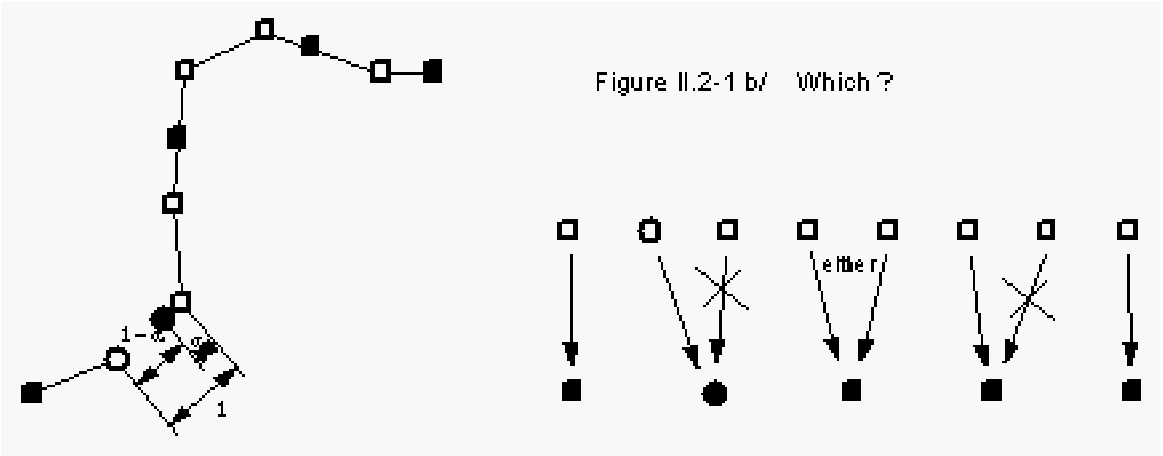


Figure II.2-1 b/ Which ?

- How buildings are to be represented ?

Each building to be represented is enlarged by factor *f* and represented as such, or as a best-fitting rectangle, or as a simplified building: in fact, any available individual transform applies.

- Control

This method makes it possible to dispense with after-controls on building-to-building topology: buildings won't bump into each other except when the enlargement factor is so huge that only two buildings remain. In most cases however, provided that the original group itself is safe from overlaps, and that scale ratios are small, the method is safe, building-to-building topologically speaking.

Building-to-road topology however has to be checked (buildings when enlarged may crawl over the street).

Features other than buildings may occur within the original group and the re-positioning of buildings may induce topological inconsistencies.

- Validity through scale

The method is valid as long as the two extremity buildings can be represented without conflicting with each other.

3.6.3.2.2 Necessary information on the group

- Constitution (one method among others)

For each building, the closest road section is retrieved.

A given road section thus comes with a group of buildings on one side and another group of buildings on the other side.

A group of buildings thus comes with the road section it follows.

- Order (make-do chaining, one method among others)

1- In the group, the two closest buildings are looked for. This is the pair B1-B2. Pi is initialised with B1 and Pf with B2 (cf. fig. II.2-2 a/).

2- (Df) being the straight line perpendicular to (Pi Pf) in Pf, and Hf the half-plane limited by (Df) and not containing (Pi), Pp is searched for as the building closest to Pf within Hf.

3- Step 2 starts again with Pi ← Pf and Pf ← Pp until there's no building left in current Hf. This makes up chain Ch1-2.

4- Steps 2 and 3 are reiterated with Pi ← B2 and Pf ← B1. This makes up chain Ch2-1

5- The two chains are brought together as a single chain ["Chain ← Ch1-2 + Reverse (Ch2-1)"].

6- The group is declared linear iff no building remains, and its order is that of Chain.

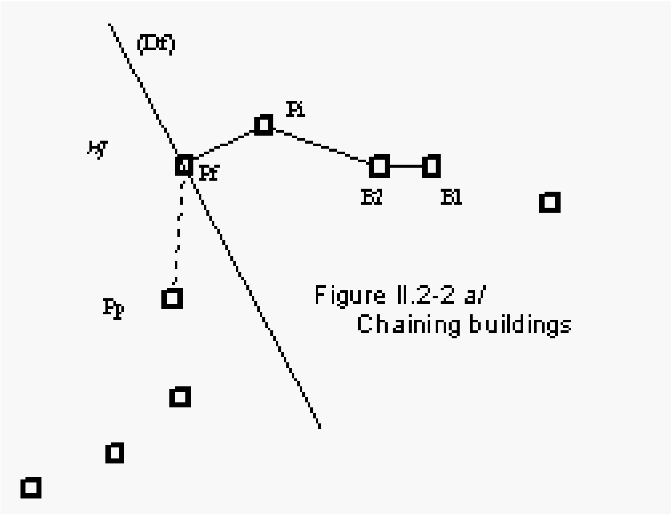


Figure II.2-2 a/
Chaining buildings

- Individual measures (some measuring methods among others)

The nature of a building is read from its semantic attribute.

The size of a building is measured as its area.

Its shape as its number of corners.

Its orientation as that of the best-fitting cross computed on its sides.

Its orientation to the road as the angle between orientation and tangent to the road where the building projects.

Its distance to the road as the pair (minimal distance to the road, maximal distance to the road).

Its distances with neighbours as the pairs (minimal distance, maximal distance).

When these are computed for each building, the mean value and deviation for each kind of measure is computed over the group. Buildings with a measurement lying outside the [mean-deviation , mean + deviation] of the measure are marked as "extraordinary" for this measure.

3.6.4 Road bends

This typification method has been applied to a succession of bends along a road (Mustière et al. 1999). It consists in removing two successive bends: not the first nor the last nor the highest), those that best cumulate large and long and symmetrical characteristics. The two pieces that remain of the road-arc are anamorphosed toward each other into reconnection. Originally, this typification method has been applied by their designers to a succession of bends identified within a road arc by means of the detection of inflection points.

3.6.4.1 Method

- Input

A series of successive bends in a road arc.

- Output

A road arc with two bends missing.

- Transformation

<u>- Principle</u>

Two successive bends are removed from the arc and the two arc pieces are anamorphosed into re-connection.

<u>- Removal of two successive bends.</u>

The pair of successive bends to be discarded is that that includes neither first nor last nor highest bend and that hits the highest value :

$\alpha \cdot hv(j) + \beta \cdot sv(j) + \gamma \cdot lv(j) \quad + \quad \alpha \cdot hv(j+1) + \beta \cdot sv(j+1) + \gamma \cdot lv(j+1)$

where hv(x) is the height value of bend x, sv the symmetry value and dv the length.

The exact weighting is unknown to me for the moment.

The pair of successive bends to be discarded is supposed in the following to lie between inflection points Ii, Ii+1 and Ii+2.

<u>- Repositioning of remaining bends.</u>

O is the barycentre of Ii and Ii+2, weighted respectively by the length after Ii+2 and by the length before Ii.

(Di-) is the direction of the bend before Ii [(Di+2+) is the direction of the bend after Ii+2].

Vi- is the vertex of the bend before Ii [Vi+2+ is the vertex of the bend after Ii+2].

The curve before Ii is transformed as follows:

Vector **IiO** is projected into vector **ti** on (Di-), and vector **li** is defined as: **li** = **IiO - ti**.

In the following, cd (M,N) is the distance along the curve between M and N on the curve.

For any intermediary point P on the curve between Vi- and Ii, its transformed self P' is computed as:

PP' = cd (Vi-,P) **ti** + cd (I1,P) **li**

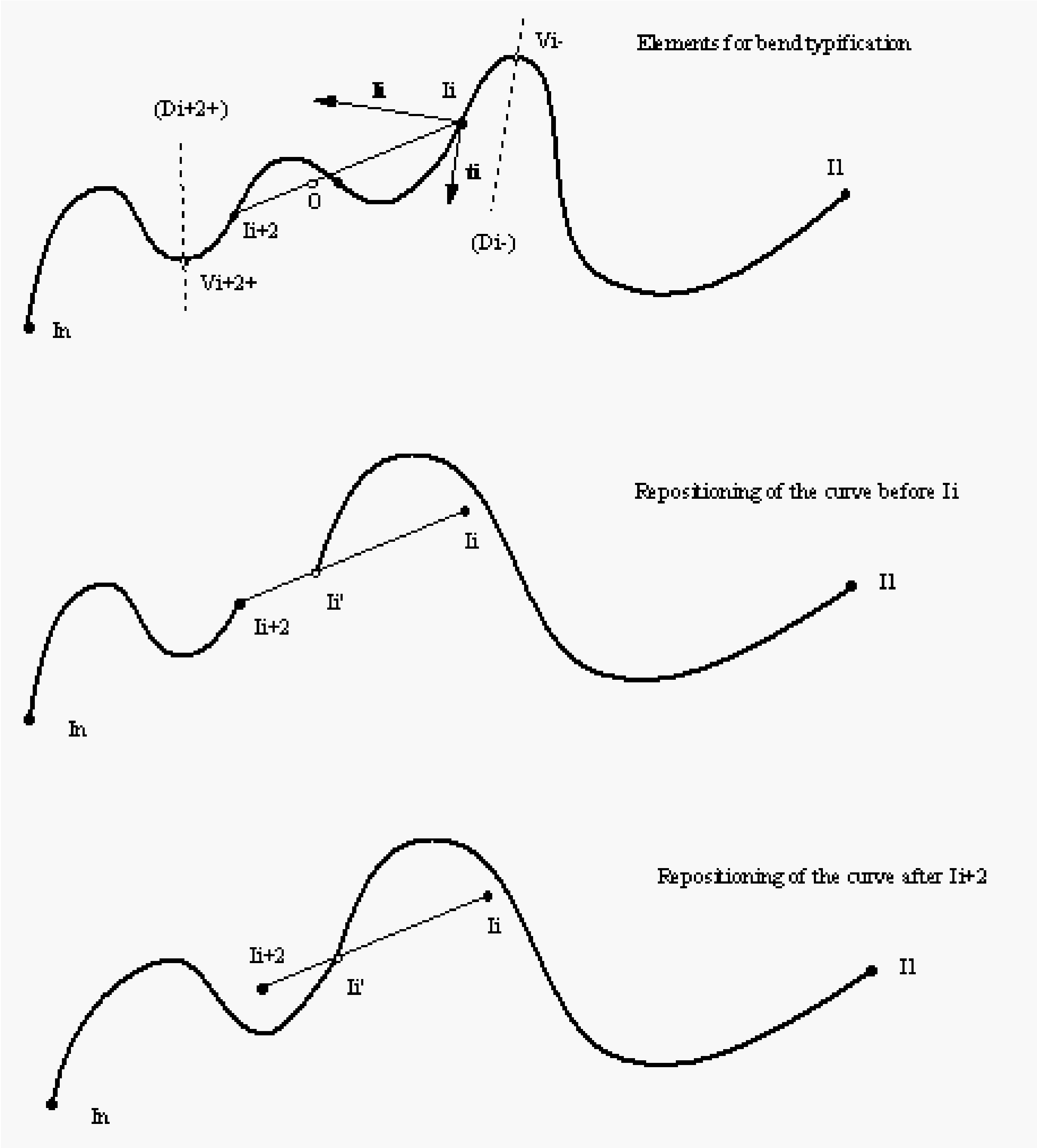
For any intermediary point P on the curve between I1 and Vi-, its transformed self P' is computed as:

PP' = cd (I1,P) **li**

The curve after Ii+2 is processed symmetrically.

<u>- Smoothing the curve</u>

The two parts of the curve usually join in O with an angle. The whole curve is smoothed lightly.



- Control

Environment is not taken into account yet. But bends following particular terrain features could be pre-selected as bends to-be-preserved.

- Validity through scale

The method is valid as long as the three bends (first, last, biggest) can be represented without conflicting with each other.

3.6.4.1.1 Necessary information on the group

- Constitution

The bends are retrieved from the road arc as the road segments occurring between two successive inflection points.

Inflection points are computed after a Gaussian smoothing parametrized by σ, as described in report C1.

- Order

The order of the bends along the road is naturally that of the progression of the road.

- Individual measures (detailed in report D1, cf. description of the “Accordion” algorithm).

Bend height : Distance from vertex to base (line joining the two successive inflection points).

Bend symmetry : The distance from the vertex’s projection on the base to the midpoint between the successive inflection points (the closer to 0, the more symmetric) is used to compute the symmetry of the bend (I don’t know the exact computation yet).

Bend length : length of the curve between the two successive inflection points.

Bend direction : half the mean value of the doubled tangents in each point between the successive inflection points.

3.6.5 Typification of 2-D groups

In this section, groups are supposed to be made of features scattered over an area of ground. While highly geometrical configurations occur in the real world (rows in rows, staggered rows etc.), and identification methods are used in other domains (e.g. automated image processing), we know of no such methods applied for generalisation purposes yet, and current typification methods dispense with the first step of eliciting specific order over the group.

3.6.5.1 Ponds

This method, originally implemented on raster data, applies to vector data as well in so much as the operations *erosion* and *dilation* are mathematically independent from their raster or vector representation. It was designed for the typification of pond-studded areas (Müller-Wang 1992). The components of the group (“patches”) are areal contours.

Patches with a size greater than a threshold (fixed indirectly through the choice of map specifications) are dilated, the others are eroded. Erosion and dilation radii vary with the patches sizes, according to the *a priori* principle: the richer get richer, the poorer get poorer. Patches in isolate small groups that are brought too close from each other are displaced apart. Patches that are too small that remain are eliminated, unless they are isolated.

3.6.5.1.1 Method

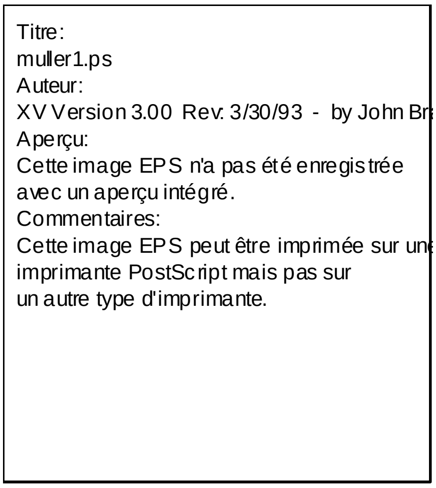
- Input

- Set of patches (ponds)
- Selection model
- Source scale (1/Ms)
- Target scale (1/Mt)
- Maximal dilation radius (k)
- Smallest tolerable size of patch on target map (*sts*, may be fixed to 0.5 mm²)

- Separability distance on target map (*sd*, may be fixed to 0.4 mm)

- Output

-Set of patches



Input patches Output
patches

- Transformation

The input patches are ordered by area.

The patches' total area is *ia*.

The total area of patches to be dilated (*fa*) is fixed by the selection model : $fa = ia * f(Ms/Mt)$

where f(x) is one of the four models available : $f(x) = x^2$; $f(x) = (x^3)^{1/2}$; $f(x) = x$; $f(x) = (x)^{1/2}$.

T is the size threshold across which patches will be dilated or eroded. It is defined as the mid-size of the last patch to be dilated (deduced from the order on the patches and the total area to be dilated) and the following patch in the ordered list.

A patch with area > T is eroded, otherwise it is thickened ; unless the patch is located in isolation [no definition is given, one could be : there is no other patch within a distance > (mean distance + 2*deviation) between patches] or the patch overlaps another patch

The erosion/dilation radius for a patch with size *s* and contour length *p* is computed as :

$$rad = [\, s / (p^2 / 4\pi)\,] * k * [\, |\, s - T\, | / (\, smax - T\,)\,]^{1/2}$$

where smax is the greatest of all patch areas.

Eliminate eroded patches whose area < *sts* * $(Mt/Ms)^2$

Reselect from these eliminated patches those that are located in isolation, and enlarge them to threshold size T.

Displace patches closer to each other than *sd* * (Mt/Ms), each by a distance proportional to the other's area, and propagate displacement to their neighbours.

Each patch is smoothed independently.

- Control

Topological distortion may occur. Check for overlaps

- Validity through scale

The method is valid throughout scales.

3.6.5.1.2 Necessary information on the group

- Constitution

The group is given as such, it is not retrieved from a bundle of data.

- Individual measures

A patch’s area.

3.6.6 Summary

The typification algorithms have been described with an insistence on the separation between:

a/ the transformation method itself and

b/ the constitution and information of the group to be transformed.

The following table reviews the algorithms described above:

- what the elements in the group are (“components”)
- what ultimate quality of the group the method relies on (“fundamental quality of the group”)
- the strong points of the transformation method (”pros”)
- the weak points of the transformation method (“cons”)
- the author’s opinion on the inclusion of the method in AGENT (”desirable?”)

Distinct somehow from the transformation methods themselves are summarised the constitution methods used by the designers of the transformation methods:

- method for constituting the group (group constitution)
- the strong points of the constitution method (”pros”)
- the weak points of the constitution method (“cons”)

Transformation method	**components**	**fundamental quality of the group**	**pros**	**cons**	**desirable?**	**group constitution**	**pros**	**cons**
Gestalt typification	buildings	linear	powerful, selected components could be generalised individually	buildings not displaced away from roads	yes	MST on buildings	elicits linear groups within a given set	qualification of the linearity / planarity of the given set to be automated
Phenomenological typification	buildings	linear	1-go, selected components could be generalised individually	buildings not displaced away from roads	yes	access criteria buildings + streets	elicits topological relations between buildings and streets	identification of corner buildings to be automated
bend typification	bends	linear	1-go, selected components could be generalised individually	2-bend removal only	yes	road segmentation at inflection points	scale-adaptable	hard to distinguish between road shapes and digitalisation noise
pond typification	ponds or round, convoluted contours ("patches")	planar	tunable	designed for raster data	not now	(given)	(-)	how can it be automated?

4 Selection of Algorithms for Organisations

4.1 Introduction

The analysis in A4 : Geographic Object Modelling, has distilled the essential organisations that are required to model contextual information for the automated generalisation of an urban infrastructure.

Region Organisations		**Network Organisations**	
Entire map extent	✓	Road	✓
Urban	✓	River	
Inverse of urban		Railway	
Urban District	✓		
Urban Block	✓		
Rural block			
Sub block (urban)	✓		
Forest			
National Park			
Mountain			

Table 1: Proposed organisations

This sections examines needs for generalisation operators and algorithms of these organisations with regard to the evaluation in section 2, State of the art.

4.2 Decomposition by organisation

4.2.1 Entire map extent

The operators that are used by this organisation are

- Creation
    - Urban organisations
    - Rural organisations

The Entire Map Extent organisation is a temporary organisation that is used in the absence of a macro level. Research on the macro level of control is the subject of future research in task D3. The organisation is required for the creation of urban and rural organisations that will ultimately be controlled by urban and rural macro level agent. The areas not generated as urban consist the rural (or inverse of urban) organisation. These are required for the creation of inter-urban organisations, such as, the road network organisations.

For the selection and creation of urban areas it is proposed to use the algorithm of Ormsby *et al* (1999, undocumented) to define the graphical boundary of an urban area (see A4). Currently no other algorithm is known to perform this operation. The algorithm is document in DD4 section 'CityBoundary'.

4.2.2 Urban

The operators that are available to this organisation are;

- Selection;
  - All Intra-urban roads and access nodes
  - Intra-urban street network
- Creation;
  - Urban blocks organisations
  - Urban districts organisations
- Global Amalgamation
- Collapse

Urban organisations are created and administered by the urban macro agent. The nature of this macro control is the subject of task D3. The urban organisations are required to control the representation of the city as a geographic entity and ensure the connection between the city and the external areas, which ultimately will control access to other urban areas. To achieve this it is necessary to have operations that can select points of connection between an urban area and the rest of the map. The selection of inter-urban access (attractive) nodes is undocumented and is likely to require further research. Currently the algorithm involves a geometric intersection operation between the city boundary and all roads of the dataset. This generates a collection of nodes on the access roads entering and exiting the city. Further research needs to determine whether this set is too large to effectively describe the access points of the city and how a second selection should be performed to generate a sub-set if necessary.

The two main geographical aspects of an urban area are occupation and an access network. To maintain this information it is necessary to have operations to select the optimal access network that can be represented at the intended scale, preserving connectivity, shape and distribution of arcs. It is proposed that this operation of intra-urban street selection uses the gestalt principle of 'good continuation' as defined by Thompson and Richardson (1999), weighted by semantic to identify the city street network as a perceptual entity. In addition the algorithm proposed uses the inter-urban nodes of the city to ensure the connectivity of the urban infrastructure across the entirety of the map. Two algorithms described in DD4 are defined for this purpose. The algorithm 'CityRoads' selects the subset of all roads that are within the city boundary. The algorithm also cuts and marks with nodes the access points at which these roads exit and enter the city. The second algorithm, described in DD4 under the section 'StreetSel1', then selects the street network from amongst these roads. Selection alone cannot be guaranteed to resolve any cartographic conflicts, such as symbology overlap amongst selected street. To achieve this displacement operators are required. However, this is a complex problem requiring further research. Possibly the algorithm of Nickerson (1989) used for inter-urban road displacement could be used.

To maintain the impression of occupation it is necessary to select city block regions within the city which can be used to control the distribution of information density across the map as well as providing a logical geographical unit to characterise neighbourhood information. Its is proposed that the selection and instantiation of these urban block organisations is performed using partitioning as defined by Brazile and Edwardes (1999) and Ruas and Plazanet (1996). The algorithm for this uses the output of the street selection process as its input. This algorithm is described in C2 in the section 'Partitioning'.

The aspects of the city as a place of economic, cultural and social importance must also be controlled for by this organisation. In order to perform this operation, it is necessary to have selection operations that can characterise and maintain this information during the generalisation process. Some of this type of information is to be maintained using aggregations of urban blocks to create urban districts. Further research needs to be undertaken on the selection and generation of urban districts within the city, an algorithm of Edwardes (1999, undocumented) for the aggregation of continuous area patches could be adapted for this purpose. However, the mechanics for the physical creation of these organisations is not as important as the characterisation to define these districts, and it is this area which needs additional research (c.f. Boffett, 1999).

For small scales or particular themes, the most appropriate representation of a city is often a single built up block or a symbolised point. As a block, information must be must preserved about the extent of the urban area and it's connectivity to other areas. Likewise, as a symbolised point information must be preserved about access to the town and its connectivity to other places. The algorithm proposed for global amalgamation to an area is again that of Ormsby *et al.* (1999, undocumented). This algorithm is defined in DD4 under the section 'CityBoundary'. The area produced then undergoes a simplification using the Douglas-Peucker (1973) algorithm to generate a visually clean and anthropogenic appearance. However, this post-processing approach is poor as it fails to address any topological, metric or structural constraints. Whilst, this algorithm therefore requires additional research for improvement, for example the use of the simplification algorithm of de Berg *et al.* (1998), this is not a priority since the current considered scale range over which generalisation will be performed makes this operation very rare. At larger scales this operation may also be performed by aggregating the filled blocks described in section 4.2.4. There exists no specific algorithm to perform the collapse and symbolisation operation though an adaptation of the algorithm of Mackechnie and Mackaness (1999) for simplification of network junctions could possibly be used. However, this area requires further research.

4.2.3 Rural (Inverse of Urban)

The operators that are currently required for this organisation are;

- Selection;
    - Inter-urban road network organisation
- Displacement of roads

Currently only operators for the selection and placement of the inter-urban road network organisations are required by the rural organisation. The set of roads on which to perform the selection is determined as those roads not in the urban organisations. The access nodes of the urban organisation will again be required to ensure that connectivity between urban places is maintained. For the selection of the road network the algorithm of Reynes (1997) is proposed. This seems to offer the most aesthetically pleasing results as well as satisfying the requirement for connectivity of attractive nodes. This algorithm is documented in DD4 in the section 'RoadNetSel'. In the later stage of implementation the algorithm of Mackaness and Beard (1993) is also proposed. This algorithm has particular properties that make it attractive for specific contexts such as sparsely inhabited landscapes. The algorithm is documented in DD4 in the section 'RoadNetSel2'. For the displacement of roads to solve proximity conflicts between lines, the use of the algorithm by Nickerson (1988) is proposed. As proven by the IGN, the algorithm succeeds in making adequate generalizations. The algorithm is well known and parameters are easy to set. This algorithm is documented in DD4 section 'RoadDisp'.

4.2.4 Urban Block

The operators that are used by this organisation are

- Selection;

- individual buildings
- minor streets and cul-de-sacs

- Creation
    - building clusters sub-block organisations
- Displacement of buildings
- Amalgamation
    - block infilling

The urban block organisations are instantiated by the urban areas. The role of the urban blocks is to manage structural information about a neighbourhood, such as, building density distribution, architectural form and access to points within the unit, for example by cul-de-sacs. According to the density of information this organisation needs tools to manage symbology conflicts either by; creation of sub-block building clusters, the selection and elimination of buildings and minor roads, the reorganisation of objects within the block by displacement. To achieve this the explicit representation of structural information is necessary. This is performed through the creation of sub-block building clusters.

For the creation of building clusters (sub block organisations) it is proposed to implement two algorithms; that of Regnauld (1998) based on gestalt principles of perception and that of Hangouet (1998) based on phenomenological principles. Both algorithms provide robust and pleasing results and are well integrated with the philosophy of the project. DD4 documents the algorithm of Regnauld (1998) section 'BldgGestalt' and Hangouet (1998), section 'BldgPhenom'. To perform the operation of building selection and elimination the algorithm of Ruas (1999) is proposed. This removes individual buildings using a cost function based on congestion criteria. The algorithm is described in section 'BldgRemoval' of DD4. To solve the displacement problem of reorganising buildings in a reduced space - without running into convergence problems - Ruas's (1999) algorithm is recommended. This algorithm is robust and has been shown to produce good results. The algorithm is documented in DD4 section 'BldgDisp'.

Where information density is too high to allow for the reorganisation of information the only approach is to amalgamate buildings into a single built-up area. For example, a city center. To perform the amalgamation the algorithm of Glover and Mackaness (1999) is proposed. This is documented under the section 'Blockfilling' in DD4. The control of this needs to be performed by a strategic agent to ensure the consistency in treatment, which could otherwise result in a false impression of over-occupation and under-occupation in different areas. The nature of this control is the subject of task D3. At small scales urban blocks become too small to be represented individually. To maintain readability it is proposed to amalgamate of urban blocks to create larger areas, using the algorithm of Ruas (1999) which is documented in DD4 section 'StreetSel2'. The mechanism for controlling this process is still the subject of further research in task D3. However, it is likely that it will either be performed by negotiation between urban blocks to control the amalgamation or by the creation of district meso-organisations that can control the process with regard to network structure and semantics.

4.2.5 Urban Sub-Blocks

The operators that are used by this organisation are

- Typification
    - Building clusters
- Amalgamation
    - Building clusters

The building cluster sub-blocks are used to represent structural and architectural information. In situations of medium information density where symbology conflicts exist, operators are also required for the re-organisation the data either by typification of the arrangement of the buildings or by the amalgamation of groups of buildings subject to structural constraints. Typification of the building clusters created will be performed again using the algorithms of Regnauld (1998) and Hangouet (1998). Clearly, these algorithms are best integrated with the clustering mechanism and data structures used to index the groupings. These algorithms are documented respectively in DD4 sections 'BldgTypif' and 'BldgTypifPhenom'. For amalgamation of buildings the displacement algorithm of Regnauld (1999) is proposed. The algorithm is used to amalgamate buildings two at a time. This algorithm is documented in DD4 section 'BldgAmalg1'.

4.2.6 Road Network

The operators that are currently required for this organisation are;

- Control and reconnection of line segments during and after generalisation

The role of the road network organisations is to manage the segmentation of roads and the reconnection of linear road segments following independent generalisation. The algorithm to perform this operation is documented in DD4 section 'RoadSegConnect'.

5 References

Anderson, J. R. (1993). *Rules of the Mind.* Hillsdale, N.J.: Lawrence Erlbaum.

Baeijs, C., Y.Demazeau and L.Alvares (1996): SIGMA: Application of Multi-Agent Systems to Cartographic generalization. *MAAMAW'96*, Berlin, Germany, p.163-176

Bobrich, J. (1995): *Promotion - Ein neuer Ansatz zur kartographischen Verdraengung auf der Grundlage eines mechanischen Federkraftmodells*. PhD thesis, ????, Germany.

Bolletino Geodesico [BG] (1998*) "Derivazione cartographica alla scala 1:50000 - Generalizzazione automatica dei fabbricati"* Bolletino Geodesico, 1998, no 2.

Boffet, A. (1999) A Framework for Automated Spatial Analysis Based on Spatialization Principles. *Workshop of the ICA Working Group on Map Generalisation*

Brazile, F and Edwardes, A (1999) Map Space Partitioning for Agent Process Organization *Workshop of the ICA Working Group on Map Generalisation*

Burghardt, D. and S. Meier (1997): Cartographic Displacement using the Snakes Concept. In: Foerstner, W. and L. Pluemer (eds.): *Semantic modelling for the acquisition of Topografic information from Images and Map*. Birkhaeuser-Verlag, Basel, Switzerland.

Chrismann, N.R.(1983) *"Epsilon Filtering : A technique for automated scale changing".* Proceedings of ACSM Annual Conference, pp322-331.

De Berg, M., van Kreveld, M., and Schirra, St. (1995): A New Approach to Subdivision Simplification. Technical Papers, ACSM/ASPRS Annual Convention, 4 (Auto-Carto 12): 79-88.

Delucia and Black (1989) A comprehensive approach to automatic feature generalisation. *Proc. 13th Int. Cartographic Conf.* (Morelia, Mexico), 168-191.

Dougenik J.A. (1980) *"Whirlpool ; A program for polygon overlay".* Proceedings Auto-Carto 4, 1980, pp 304-311.

Douglas, D and Peucker T. (1973). Algorithms for the reduction of the number of points required to represent a digitised line or its caricature. The Canadian Cartographer, vol **10**. pp. 112-122

Dutton, G. (1999). Scale, sinuosity and point selection in digital line generalization. *Cartography and Geographic Information Science*. **26(1)**, 33-53.

Glover, E. and Mackaness, W. (1999) Dynamic Generalisation from Single Detailed Database to Support Web Based Interaction *Proceeding of the 19th International Cartographic Conference, Ottwawa* pp. 1175-1185

Hangouët J-F (1996)*: Automated Generalization Fed on Latent Geographical Phenomena*, Proceedings of InterCarto 2, Irkutsk June 1996, pp.125-128

Hangouët J-F (1998): *Approche et méthodes pour l'automatisation de la généralisation cartographique; application en bord de ville*, thèse de doctorat, université de Marne-la-Vallée, novembre 1998

Hangouët J-F, and Lamy S (1999): *Automated Cartographic Generalization: Approach and Methods* (to be published in the proceedings of ICA/ACI 1999, Ottawa Canada)

Hangouët, J-F and Regnauld, N (1996) : Analyse des îlots urbains pour la généralisation contextuelle, Revue Internationale de Géomatique, vol.6, n.2-3, 1996, pp.219-226.

Harrie, L.E. (1999): The Constraint Model for Solving Spatial Conflicts in Cartographic *Generalization. Cartography and GIS*, 26 (1), p.55-69.

Helleman, G. L, (1996) *Data structures, Algorithms and Object-Oriented Programming*. MCGraw-Hill:Singapore.

Hojholt, P. (1998): Solving Space Conflicts in Map Generalisation using a Finite Element Method. *Cartography and GIS* (To appear)

Horton (1945) Erosional development of streams and their drainage basins, *Bulletin of the Geological Survey of America,* Vol **56**, pp. 275-370.

Lichtner, W. (1976): *Ein Ansatz zur Durchfuehrung der Verdraegung bei der EDV-unterstuetzten Generalisierung in Topographischen Karten*. PhD thesis, Hannover, Germany.

Mackaness, W. A., and Beard, M. K. (1993), Use of Graph Theory to Support Map Generalization. *Cartography and Geographic Information Systems*, **20**, 210 - 221.

Mackaness, W.A. (1994): *"An Algorithm for Conflict Identification and Feature Displacement in Automated Map Generalization".* Cartography and GIS, 21 (4), p.219-232.

Mackaness W.(1995) *"Analysis of Urban Road Networks to support Cartographic Generalisation".* Cartography and GIS, Vol 22, no 4, 1995, pp 306-316.

Mackaness & Mackechnie, 1997 *Detectiond and Simplification of Road Junctions in Automated Map Generalization.* ICC'97 Proceedings Vol.2 pp 1013-1021

Mackaness, W. A., and Mackechnie, G. (1999), Detection and Simplification of Road Junctions in Automated Map Generalization. *GeoInformatica*, **3**, 185-200.

McMaster R (1991): *Conceptual frameworks for geographical knowledge*, in 'Map Generalization', edited by Buttenfield and McMaster, Longman Scientific & Technical, 1991 pp.21-39

McMaster R, and Shea K S (1992): *Generalization in Digital Cartography*, Association of American Cartographers, 1992.

Meyer U (1989)*: Generalisierung der Siedlungsdarstellung in digitalen Situationsmodellen*, thèse de doctorat, université de Hanovre, 1989.

Morisset B & Ruas. A (1997) *"Simulation and Agent Modelling for road selection in generalisation".* ACI 1997 Vol 3 pp 1376-1380 Stockholm.

Müller J.C. & Wang Z. (1991) *"Area-Patch Generalization : a competitive approach".* Rapport technique, ITC Enschede, 1991.

Müller, J-C and Wang Z (1992): *Area-patch generalization: a competitive approach*. 1992, The Cartographic Journal, n.29, pp.137-144

Mustière S, Lecordix and F Skarbinck S (1999),: *Bend removal*, Algorithm Grid, Cogit, 1999.

Nickerson, B.G.(1989): Automatic cartographic generaliszation for linear features. *Cartographica*, 25 (3), p.15-66.

Ormsby, D. Edwardes, A and Mackaness W. (1999) See DD4 Section 'CityBoundary'.

Peng, W. and Muller, J. C., (1996) A Dynamic Decision Tree Structure Supporting Urban Road Network Automated Generalisation. *The Cartographic Journal.* Vol **33** (1) pp. 5-10.

Powitz B (1993): *Zur Automatisierung der Kartographischen Generalisierung topographischer Daten in Geo-Informationssystemen,* thèse de doctorat, université de Hanovre, 1993.

Regnauld N (1997): *Structuration du bâti pour sa généralisation*. Proceedings of ICA/ACI 1997, Stockholm, Vol.3 pp.1395-1401.

Regnauld N (1998): *Généralisation du bâti : Structure spatiale de type graphe et représentation cartographique*, thèse de doctorat, université de Marseille, mai 1998

Regnauld N (1999) Preserving Density Constrasts During Cartographic Generalisation. *Innovations in GIS VII: GeoComputation*. Atkinson and Martin (eds) Taylor and Francis:London (in press)

Reynes J.L. (1997) *"Selection du reseau routeir en vue de la selection".* DESS Report, University Paris VI, COGIT, 1997.

Richardson, D. (1993) *Automatic Spatial and Thematic Generalisation using a Context Transformation Model*, Doctoral Dissertation, Wageningen Agricultural University, Netherlands.

Richardson, D., and Thomson, R. C. (1996), Integrating Thematic, Geometric and Topological Information in the Generalisation of Road Networks. *Cartographica*, **33**, 75-84.

Roberts, J. (1997): *The Development of an Autonomous Self Evaluating Algorithm for Map Generalisation: Automated Feature Displacement*. MSc Dissertation, Departement of Geography, University of Edinburgh, Scotland.

Ruas A, and Mackaness W (1997): *Strategies for urban map generalisation*, Proceedings of ICA/ACI 1997, Stockholm, Vol.3 pp.1387-1394.

Ruas A, Lagrange J-P (1995)*: Data and knowledge modelling for generalization* in 'GIS and Generalization', edited by Müller, Lagrange & Weibel, Gisdata vol.1, series editor Masser & Salgé, Taylor & Francis 1995.

Ruas A. and Plazanet, C. (1996) Stratergies for automated map generalisation. *Spatial Data Handling'96* Delft, Netherlands. *Advances in GIS Research II*: Taylor and Francis pp. 319-336

Ruas A. (1999) *"Modèle de généralisation de données géographiques à base de contraintes et d'autonomie"*. Thèse de doctorat, Université de Marne-la-Vallée, avril 1999.

Rusak Mazur, E. and Castner, H. W., (1990). 'Horton's ordering scheme and the generalisation of river networks', *The Cartographic Journal*, Vol. **27**, pp. 115-125.

Strahler, A, N. (1960) *Physical Geography*. Wiley:New York.

Staufenbiel W (1973): *Zur Automation der Generalisierung topographischer Karten mit besonderer Berücksichtigung großmaßstäbiger Gebäudedarstellungen*, thèse de doctorat, université de Hanovre, 1973.

Thomson R C. & Richardson D E. (1995) "*A graph Theory approach to road network generalisation*". Proc. ACI - Barcelona 1995. pp 1871-1881.

Thompson, R. C, and Richardson, D. E. (1999) The 'Good Continuity' Principle of Perceptual Organisation applied to the Generalisation of Road Networks. *Proceeding of the 19th International Cartographic Conference, Ottwawa* pp. 1215-1225

Töpfer, F. (1974) "Kartographische Generalisierung" VEB Hermann Haack, geographisch-Kartographische Anstalt Gotha Leipzig.

Topfer, F and Pillewizer W. (1966). "The principles of selection", *The Cartographic Journal*. Vol. **3** (1) pp. 10-16.

Ware, J.B. amd C.B. Jones (1999): Conflict Reduction in Map Generalization Using Iterative Improvement. *GeoInformatica*, 2 (4), p.383-407.

Ware M., Jones. C and Bundy L. (1995) A spatial model for detecting (and resolving) conflict caused by scale reduction. Proceeding SDG 96, pp. 547-558