HOW THE QUALITY OF CARTOGRAPHIC GENERALIZATION IS ASSURED – THE APPROACH WITH AN AUTOMATIC CARTOGRAPHIC GENERALIZATION CHAIN

H. Z. Qian[a, *], Y. Xu[b], L. Q. Meng[c], Q. Y. Song [a]

[a]Zhengzhou Institute of Surveying and Mapping, 450052, Longhai middle road NO.66, Zhengzhou, China – haizhongqian@163.com
[b]Dalian Surveying and Mapping group, China, 116000 – Xuyun@163.com
[c]Department of Cartography, Technical University of Munich, Arcisstr. 21, D-80333 Munich, Germany – meng@bv.tum.de

Commission WG II/3

KEY WORDS: Cartographic Generalization, Generalization Knowledge, Process Control, Automatic Cartographic Generalization Chain

ABSTRACT:

As one of the key techniques in multi-representation database, GIS, mapmaking and Digital Earth, automatic cartographic generalization has been intensively researched since decades. Current developments worldwide have been concentrated on the efforts towards adaptive and easily embeddable automatic generalization modules or services. Based on an analysis of the common rules concerning the quality issues that a cartographic generalization system has to obey, this paper put forward an "automatic cartographic generalization chain". This chain integrates cartographic generalization knowledge, algorithms, operations, real-time monitoring, quality assessment and process control together. With examples, we show how to apply the common rules in an automatic cartographic generalization system.

1. INTRODUCTION

With the rapid development of GIS, mapmaking automation, and Digital Earth, researchers have become increasingly aware of the role of automatic cartographic generalization. Indeed, there exists already a rich knowledge base in current computer-aided generalization systems along with a large number of easy-to-use generalization modules. In spite of their different applications and user groups, all generalization systems share some commonality, especially the general rules they have to obey. Therefore, it is a fundamental task for the involved developers to gain an insight into these common rules (Wang, 1992).

In this paper, the authors introduce a feasible strategy of quality assurance - "automatic cartographic generalization chain" (ACGC) on the basis of their experiences with the development and application of a generalization system termed as "MiniGIS". The following sections will address this strategy in two parts: common rules a generalization system has to obey, and the realization of ACGC in "MiniGIS" (Qian & Meng, 2007).

2. COMMON RULES FOR CARTOGRAPHIC GENERALIZATION SYSTEMS

If we compare a GIS with a cartographic generalization system, we will notice that the latter is more difficult to develop because of more constraints it has to satisfy. For example, user will typically expect the cartographic generalization system to provide the following additional functions which are not necessary in a GIS:
(1) assessment and management of generalization algorithms.

(2) intelligent map compilation.

(3) iterative retrieval and optimization of generalization operations

(4) quality assessment of generalization results.

These functions are associated with a number of design rules which in turn have to take a number of quality criteria into account.

(1) The effectiveness and efficiency
A good generalization system is not necessarily a fully automatic system which is anyway not yet possible in current stage according to the research community. But it must be effective and efficient. In other words, That means, the system with its functions must be able to solve individual generalization problems and provide satisfactory results on the one hand, and reduce the efforts of human cartographers and accelerate the mapmaking process (Li Z et al., 2004).

(2) The generalization quality
Whether a generalization system is successful or not is decided by its quality in the end. We might improve the generalization quality by the measures such as (Qian et al., 2005; Qian et al., 2006d):
. Careful selection of generalization algorithms
· Careful assignment of generalization operations
. Adoption of a generalization knowledge base
. Assessment of generalization results in real-time.
. Control the whole process of generalization

(3) System intelligence
It is a well-known fact that different cartographic experts might generalize the map in different ways although they obey the same essential rules. Likewise, in a comprehensive

generalization system, different generalization modules, their different combinations or different calling sequences may also lead to different results as well. Therefore, the system developer faces a fundamental challenge of how to minimize the heterogeneity and the uncertainty of human operators. That is to say, we should let computer do as much work as possible and minimize the decision load of human cartographers (Qian et al., 2006c).

(4) Integration
In order to perform a generalization task, many individual operations or functions must be combined and iteratively activated. This requires on the one hand the close cooperation between cartographers and system developers, on the other hand the interaction between the users and the system.

(5) Adaptability
The adaptability is a quality factor that reflects how well the system suits users' need and the applications. As soon as missing functions are detected or required, the generalization system should be further developed so as to fill gaps between supply and demand. The system should allow to insert their own functions via interface protocols. Likewise, it should allow the developer or the user to pick up the relevant functions for a particular application without having to re-install the system. Finally, the system should assure a smooth user interaction and data share (Mustière S, 1998).

3. EXAMPLE – ACGC

3.1 The definition of ACGC

With the source data as the starting point for the generalization, the identified cartographic generalization task can be divided into several subtasks. Every subtask can be further divided into even smaller subtasks (see Figure.1). So there are hierarchical relations among tasks, with the root on the top and leaves at the bottom (Qian et al., 2006a).

Accordingly, an ACGC contains many subchains which in turn can be decomposed into even smaller subchains. This is again a hierarchical structure. The relation between each parent node and child node is actually the relation of parent-child task. Subtasks at the same hierarchical level are regarded as equivalent. The tasks lying in the intermediate levels have preceding and subsequent tasks.

These chains can be ordered and translated into a particular chain which can be recognized and executed by computer. We term this executable chain the ACGC.

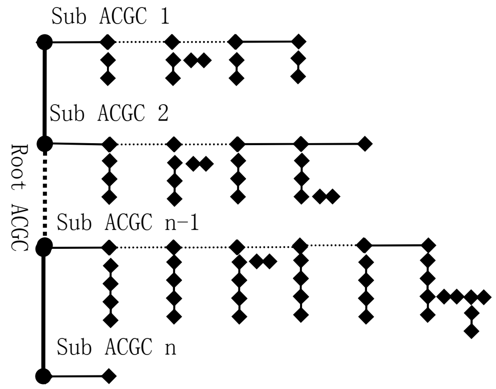

Figure.1 Structure of ACGC

The ACGC reveals a number of characteristics as follows (Qian et al., 2006b):

(1) **Complexity**. Every ACGC may deal with many generalization tasks, and have a lot subnodes.

(2) **Single direction.** A parent chain may contain many subchains, but a subchain can only be pointed to one parent. That means the ACGC is structured in a single direction and generalization operations should be executed one after another.

(3) **Multiple results**. Any generalization operation will affect its next operations in ACGC. Therefore, different generalization operation orders will produce different generalization results.

(4) **Non single optimal solution**. Map generalization is an over-constrained and ill-structured problem, that is, there does not exist the best solution for a certain conflict per se (Weibel and Jones, 1998)

(5) **Composition.** An ACGC can be divided into subchains. The similar tasks may be organized into a similar chain, and different tasks in different chains.

3.2 Defining and organizing CGK

Cartographic generalization knowledge in combination with the map purpose, spatial characteristics of mapping area and map scale etc. provides important supports to generalization tasks, algorithms and operations. It is an ordered collection of common and recessive generalization rules. In this paper, cartographic generalization knowledge is defined to contain five types of information: characteristic of map features, generalization thresholds, generalization conditions, related generalization operations, and generalization algorithms (see Table1).

GenerKnowledge<ID, Cod, GQ, GC, GO, GA>

ID	record number
Cod	data coding
GQ	generalization threshold
GC	generalization condition
GO	generalization operation
GA	generalization algorithm

Table 1 Knowledge structure

a ID coding length width tension area relative-elevation elevation selection transformation simplification mergence displacement edition map-purpose map-character note
b 1 30210 8<2/1/3> 0 0 0 0 0 must-sel can-tran can-simp can-merg can-dis can-edit common conmmon-area street-sideline
<1> selection <2> simplification
<3> mergence <4> displacement ……
</1> TIN algorithm </2> genetic algorithm
</3> mathematical morphology
</4>Agent algorithm ……

Table 2 An example of knowledge structure

Table2 shows an example of cartographic generalization knowledge (see record (a)) and an example (see (b)). In this record, "coding" means the type and characteristics of the map feature. The items "length", "width", "tension", "area", "relative-elevation" and "elevation" can be assigned with generalization thresholds. For example, 8<2/1/3> shows when the length of object is larger than 8mm, system will begin the simplification operation (<2>) which cites TIN algorithm (</1>) or mathematical morphology (</3>) (refer to Table2). The items of "selection", "simplification", "mergence", "displacement" and "edition" are generalization operations. The items "map purpose" and "map characteristic" indicate the applicability of cartographic generalization knowledge and they belong to generalization condition.

3.3 Mining cartographic generalization tasks automatically

A given dataset should be matched with cartographic generalization knowledge in order to identify the generalization tasks. If the dataset does not fit the requirements expressed in the knowledge, a generalization process will be activated. A typical workflow of mining cartographic generalization tasks is show in Table3.

```
//Source：data、CGK lib
        Source：<Data><Knowledge Lib>
//Method：Compare measure
Method：<Compare>
//Get original data
Group<Data>
//Initialize generalization condition
Set<GC>
//Order data with importance(priority)
Order(Data ID) with Data Attribute
// Ransack data
For(Data ID from 0 to max)
  //Ransack CGK lib
  For(Knowledge ID from 0 to max)
     //Compare data and CGK
     If(Data Attribute< Knowledge GQ)
     //Problem data is put into lib1
     Then Put Data ID to Lib1
       //Related CGK is put into lib2
        Put Knowledge to Lib2
//Order CGK of lib2 with importance(priority)
Order(Lib2) with Impt
// Distill operation and algorithm information from lib2
Generalize(GO,GA) from Lib2
```

Table 3 process of mining generalization tasks

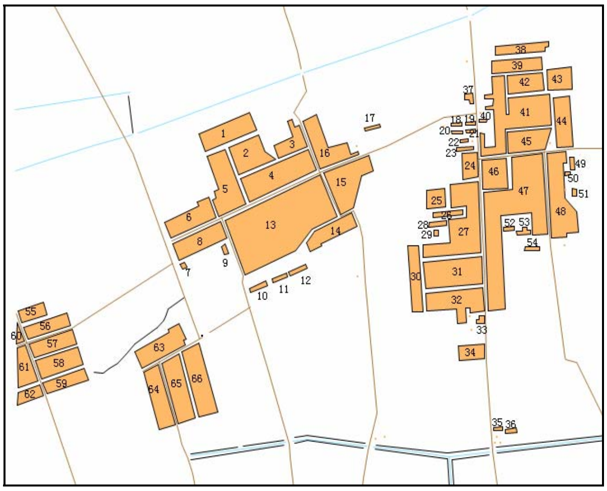

Figure.2 Source data of a city map at 1:10000 scale

Step1: "Source" means to initialize the dataset and the knowledge library.

Step2: In the item "Method", the "*Compare*" method of mining data is set, which means to mine the generalization tasks by comparing all attributes of data with records in the knowledge library.

Step3: Initialize generalization conditions such as generalization purpose, data scale, area characteristics, road characteristics, hydrographic characteristics and map load etc.

Step4: Order data with priority so as to reduce the complexity of process.

Step5: iteratively compare the dataset with knowledge library to identify the generalization need and the associated knowledge records. That means if any data does not satisfy the requirements in the knowledge libarary, the dataset will be regarded as generalization candidate and put into *lib1*, and the related knowledge records are put into *lib2*.

Step6: the information in *lib1* will undergo a generalization process.

For example, a dataset at 1:10000 scale (Figure2) will be generalized to 1:25000 scale. A part of generalization tasks mined from the dataset are listed in Table4.

```
……
20, 130204, 0, 0.3<3/3>, 0, 0, 0, 0, MS, NT, CSi, CM, CD, CE,
GM, Street-block, border upon 21
21, 130204, 0, 0.3<3/3>, 0, 0, 0, 0, MS, NT, CSi, CM, CD, CE,
GM, Street-block, border upon 20
22, 130204, 0, 0, 0, 12, 0, 0, MD, NT, NSi, NM, ND, CE, GM,
Street-block, Small
23, 130204, 0, 0.3<3/1>, 0, 0, 0, 0, MS, NT, CSi, CM, CD, CE,
GM, Street-block, border upon 24
24, 130204, 0, 0.3<3/1>, 0, 0, 0, 0, MS, NT, CSi, CM, CD, CE,
GM, Street-block, border upon 23
26, 130204, 0, 0.3<3/1/3/4>, 0, 0, 0, 0, MS, NT, CSi, CM, CD,
CE, GM, Street-block, border upon 27/28
27, 130204, 0, 0.3<3/1/3/4>, 0, 0, 0, 0, MS, NT, CSi, CM, CD,
CE, GM, Street-block, border upon 26/28
28, 130204, 0, 0.3<3/1/3/4>, 0, 0, 0, 0, MS, NT, CSi, CM, CD,
CE, GM, Street-block, border upon 26/27
29, 130204, 0, 0, 0, 12, 0, 0, MD, NT, Nsi, NM, ND, CE, GM,
Street-block, Small
……
--------------------------------------------------------------------------------
-----------
Must selecte—MS, Must delete—MD, Can not translate—NT
Can simplify—CSil Can not simplify—NSil Can merge—CM
Can not merge—NM, Can displace—CD, Can not displace—ND,
```

Table 4 Result of mining tasks from original settlement dataset Figure2(parameter of width(distance)=0.3mm, area=12mm^2) (part)

3.4 Forming and executing ACGC automatically

The cartographic generalization tasks can not be identified and executed by computer directly. All the tasks must be assembled, organized and managed so as to form ACGC which then can be executed by computer with a particular cartographic generalization compiler.

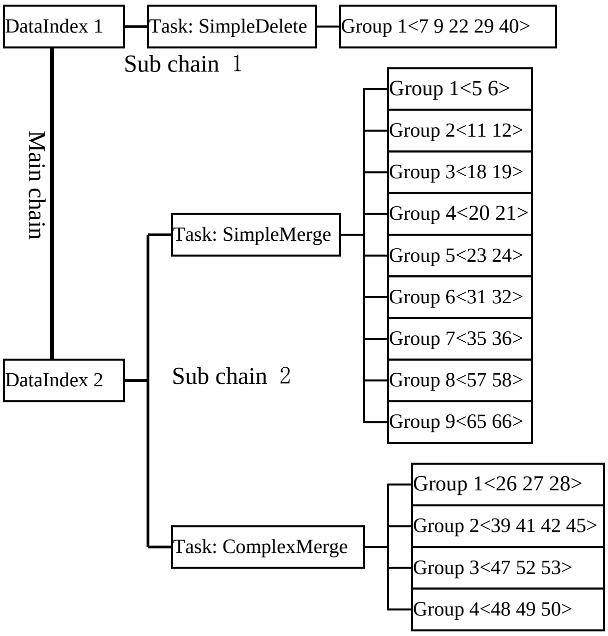

Figure.3 Translation of generalization tasks of table 4 into ACGC

This process contains two steps:

(1) Divide generalization tasks into generalization operation types.

In Table4, two generalization operations are listed which are “deletion” and “merging”. So all generalization tasks are segemented into two parts i.e. “DataIndex 1 (deletion)” and “DataIndex 2 (merging)”.

(2) In each part, tasks are divided into several groups.

Tasks related to the same operation are set in one group. For example, task 5 and task 6 have the same operation and are set in one group. They will undergo merging operation.

Figure3 is the result of ACGC transformed from Table4. The “DataIndex 1”and “DataIndex 2” constitute the main chain of ACGC, and every DataIndex contains subchains.

Table5 shows the executable codes derived from ACGC of Figure3. With a particular cartographic generalization compiler, the codes of Table5 can be executed automatically. Figure4 is the result after the execution.

```
START
   DataIndex 1
      Task: SimpleDelete
                           Group 1<7 9 22 29 40>
   DataIndex 1
   DataIndex 2
      Task: SimpleMerge
    Group 1<5 6> Operate <3/1>
    Group 2<11 12> Operate <3/3>
        Group 3<18 19> Operate <3/3>
       Group 4<20 21> Operate <3/3>
       Group 5<23 24> Operate <3/1>
       Group 6<31 32> Operate <3/3>
       Group 7<35 36> Operate <3/1>
          Group 8<57 58> Operate <3/3>
          Group 9<65 66> Operate <3/3>
      Task: ComplexMerge
    Group 1<26 27 28> Operate <3/1/3/4>
          Group 2<39 41 42 45> Operate <3/1/3/4>
          Group 3<47 52 53> Operate <3/1/3/4>
          Group 4<48 49 50> Operate <3/1/3/4>
   DataIndex 2
END
```

Table 5 Translation of settlement generalization chain of Table4 into executable code

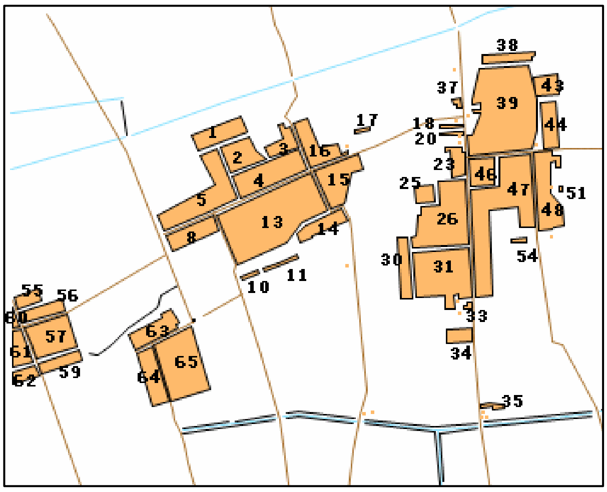

Figure.4 Tthe result of automatic generalization.

3.5 Monitoring cartographic generalization

Different human cartographers possess different generalization knowledge. If an automatic generalization system can monitor cartographers' generalization operations, wrong decisions could be reduced, thus the generalization quality will be improved. Being supported by the cartographic generalization knowledge, our system can monitor users' operations in real-time. That is, if any operation disobeys the requirements of the knowledge, it will be interrupted.

The monitoring model mainly consists of perception, log lib, analysis, action and cartographic generalization knowledge.

(1) Perception: to detect the change of outer environment in real-time, and send the perceived information to analysis engine and log lib modules.

(2) Log lib: to record and save all information sent from perception module.

(3) Analysis engine: to analyze the received information depending on the generalization knowledge and produce corresponding instructions which will be sent to action module.

(4) Action: to execute the instructions received from analysis engine module.

(5) CGK: to offer support to all other modules.

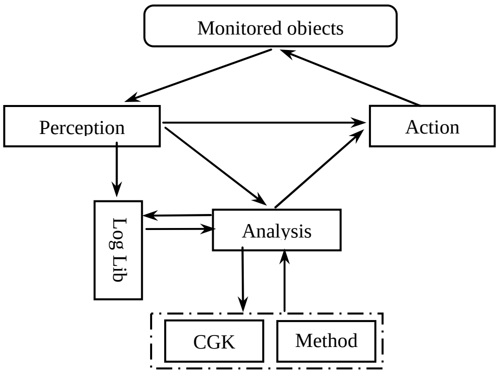

Figure.5 Structure of monitor model

Figure 5 demonstrates the process. Perception module monitors objects in real-time and sends its result to log module and analysis engine module; according to rules and methods defined in the cartographic generalization knowledge base, analysis engine creates the instructions to guide the generalization in action module.

3.6 Save ACGC based on CASE Technique

A successful ACGC can be saved as a CASE in CASE lib. If we can not achieve fully satisfied generalization results at last, a most promising ACGC will be selected from CASE lib and be executed to get the final generalization result. Only in this way can our system iteratively achieve desired generalization result at the end.

4. THE WHOLE MODEL OF ACGC

Now we can divide the whole ACGC process into the following steps (Figure6):

(1) Data preparation
It checks the input data, the nomenclature, unit, range and district of data, and classifies data, e.g. into main roads and rivers etc.

(2) Quality assessment of source data. If the source data does not need generalization, the system will terminate automatically.

(3) Initializing system. The various components of the generalization system are set to their initial states with the default parameters and values.

(4) Set and adjust state before generalization. A number of generalization conditions are set and adjusted to the actual task, they include cartographic purpose, source scale, target scale, characteristic of data area; generalization parameters, algorithms and operations etc.

(5) Mining cartographic generalization tasks automatically
(6) Forming ACGC automatically

(7) Executing ACGC automatically

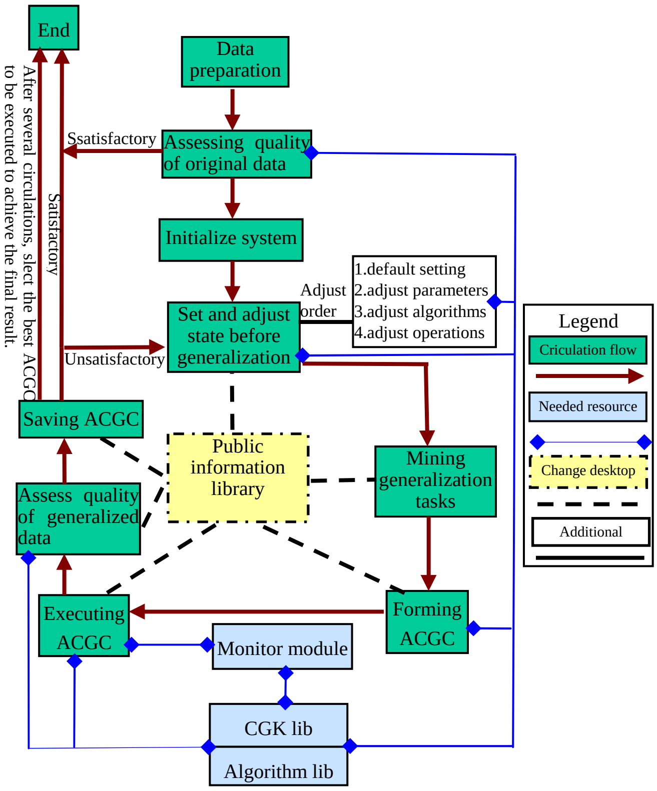

Figure.6 The complete generalization system based on ACGC

(8) Assess the quality of generalized data, and judge whether the result satisfies the requirements of CGK or not.
If the result does not satisfy the requirement, system will go back to step (4) and adjusts generalization parameters, algorithms and operators, then start from step (4) again. Otherwise, system goes to steps (11).

(9) Saving ACGC formed after each iteration as a CASE in CASE lib.

(10) If system can not achieve satisfied result after a certain number iterations, the system will select the most promising solution from CASE lib and execute it.

(11) Terminating.

5. CONCLUSTION

Automatic cartographic generalization is an integration of many theories, methods and techniques. In comparison to a GIS, a generalization system needs to satisfy more strict quality requirements, still with less chance of success. Based on the experiences with the development of the generalization system MiniGIS, the paper introduces a strategy of quality assurance - "automatic cartographic generalization chain" (ACGC). The first results have proved its feasibility.

ACKNOWLEDGEMENTS

The work described in this paper was supported by National Natural Science Foundation of China (NNSFC) (No.40701157) and the joint venture between Information Engineer University, China, and Technical University of Munich, Germany, within the frame of the bundle project "Interoperation of 3D Urban Geoinformation" co-sponsored by NNSFC and Deutsche Forschungsgesellschaft (German Research Foundation) (DFG).

REFERENCES

Wang, J., 1992. *Theory of General Map Generalization*, Map Publishing House, Beijing, China, pp.1-12.

Qian H.Z. and Meng L.Q., 2007. "Polarization Transformation" as an algorithm for automatic generalization and quality assessment. In: *the 15th International Conference on Geoinformatics*, Nanjing, China.

Weibel R. and Jones, C.B., 1998. Special Issue on Map Generalization, *GeoInformatica*, Vol. 2, Part 4.

Qian, H.Z., Wu, F. and Wang, J.Y., 2006a. Automated Generalization-Chain and Cartographic-Generalization Process-Control Model. *Journal of China University of Mining & Technology*, 35(6), pp.787-792.

Qian, H.Z., Wu, F. and Wang, J.Y., 2006b. Theory and Technique Model of Automated Cartographic Generalization Chain. *ACTA GEODAETICA et CARTOGRAPHICA SINICA*, 35(4), PP.400-408.

Qian, H.Z., Wu, F., Guo, J. and Wang, J.Y., 2006c. Spatial Data Checking with Generalization Knowledge. *ACTA GEODAETICA et CARTOGRAPHICA SINICA*, 35(2), PP.184-191.
Li, Z., Yan, H., Ai, T. and Chen, J., 2004. Automated building generalization based on urban morphology and gestalt thoery. *International Journal of Geographical Information Science*, 18, pp.513~534.

Qian, H.Z., Wu, F., Zhang, L. L. and Chen, B., 2005. Quality Assessment of Point Group Geometry Generalization with Polarization Transformation. *ACTA GEODAETICA et CARTOGRAPHICA SINICA*, 34(4), PP.361-370.

Qian, H.Z., Meng, L.Q., Wu, F. and Wang, J.Y., 2006d. The Generalization of Point Clusters and Its Quality Assessment Based on a Polarization Approach. *Mapping and Image Science*, 2006.4, pp.55-63.

Mustière S, 1998. "GALBE: Adaptive Generalisation. The need for an Adaptive Process for Automated Generalisation, an Example on Roads", In: *Proceedings of 1st GIS'PlaNet conference*, Lisbonne.