Map Generalization in GIS: Practical Solutions with Workstation ArcInfo™ Software

An ESRI® Technical Paper • July 2000

ESRI 380 New York St., Redlands, CA 92373-8100, USA • TEL 909-793-2853 • FAX 909-793-5953 • E-MAIL info@esri.com • WEB www.esri.com

J-8512

Map Generalization in GIS: Practical Solutions with Workstation ArcInfo Software

An ESRI Technical Paper

Contents **Page**

Generalization in a GIS Environment 1

The Nature of Computer-Assisted Generalization 2

Enriching the Database to Support Generalization 3
Associating Features with Scales 3
Collecting Features in Meaningful Ways
(Adding Intelligence to Features) 4
Case I: Street Network 4
Case II: Buildings in Street Blocks 5

Generalization with Workstation ArcInfo 6

New Generalization Operators, Commands, and Enhancements 6
BENDSIMPLIFY Operator 6
ORTHOGONAL Operator and BUILDINGSIMPLIFY
Command 7
FINDCONFLICTS Command 9
CENTERLINE Command 9
AREAAGGREGATE Command 10
Enhancements to the GENERALIZE Command 11
Correcting Some Label Errors 11
Fixing the Root of the Problems—Topological Errors 12

Deriving Procedures to Solve Generalization Problems....................... 12
Area Extend 13
Constrained Area Aggregation... 15

Conclusions and Looking Ahead... 16

References.. 18

Map Generalization in GIS: Practical Solutions with Workstation ArcInfo Software

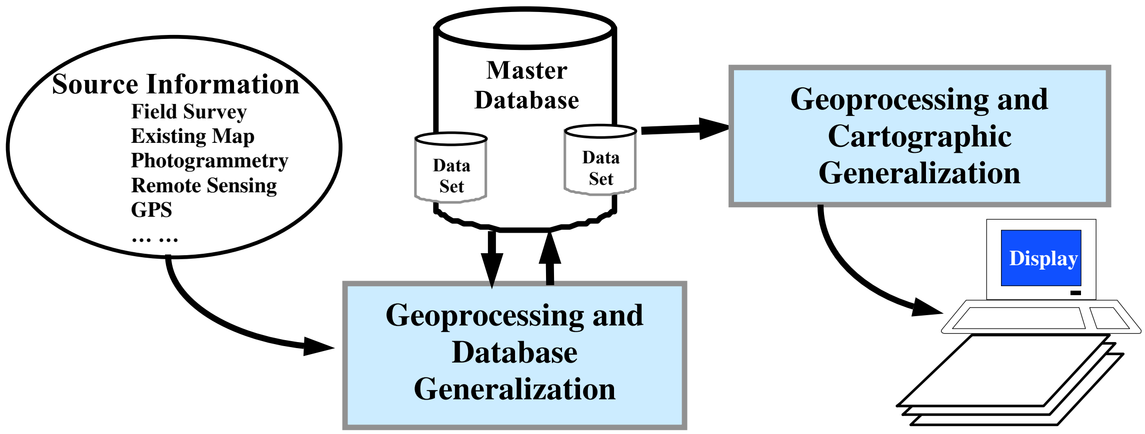
In recent decades, the mapping industry has taken the revolutionary move of replacing traditional manual work by computer-driven processes. As Joel Morrison stated in his recent review of the state of government cartography, "The last thirty years have seen cartographers digitize and automate analog mapmaking processes. We are essentially done with this task." Geographic information system (GIS) technology has been widely used by geographers and cartographers to create databases that serve long-term, multiple-purpose applications. Map generalization has become an indispensable component of today's GIS systems. Whatever the task, from compiling data in different sources into a master database to composing an output from it for data compression, analysis, or representation, generalization is most likely involved. The basic concepts and the state of the art of computer-assisted generalization were reviewed in a previous paper (ESRI). The present paper further explains the nature of map generalization in GIS and presents our newly developed generalization functions and sample procedures derived for specific generalization tasks in Workstation ArcInfo™ software.

Generalization in a GIS Environment

Generalization in GIS is no longer merely a cartographic task. Today's cartographers "end up compiling two things: the geographic database, and maps derived from this database" (Morehouse). Weibel and Jones came up with two terms to distinguish two types of generalization tasks: database generalization and cartographic generalization. Database generalization can help with both extracting appropriate information from source data to put into a master database and deriving new databases or data sets with less detail from the master database for analysis or applications at reduced scales. Cartographic generalization produces graphic products or visualization of the database, as maps or computer displays, usually also at a reduced scale. Figure 1 shows where database generalization and cartographic generalization may take place in a GIS environment.

J-8512

Figure 1
Generalization in GIS



Although both database generalization and cartographic generalization reduce data complexity, the former focuses more on the content, completeness, and accuracy of the derived data, while the latter deals with map space and resolution, symbol conflicts, visual quality, and readability. For example, if we need to derive a reduced database to represent a given level of information, we can execute a selection query to obtain a subset of data from the master database. But if we need to put the subset data onto a map, we have to satisfy certain cartographic specifications, that is, the minimum spacing and minimum sizes of symbols, the balance of feature density, and so on, which directly affect the readability of the map. Some mapped features may have to be displaced, excluded, or simplified. It is quite common that in order to produce a cartographic output, database generalization has to be done as a preprocess to obtain the desired subset of data, followed by cartographic generalization.

The Nature of Computer-Assisted Generalization

A digital solution to a given problem greatly depends on whether or not that problem and the expected outcome can be precisely described and the processing rules and logic can be translated into a computer program. Some geoprocesses can be easily automated based on clearly defined objectives, rules, and straightforward techniques, for example, converting data from a vector format to a raster format or transforming a set of data from one map projection to another. But generalization, especially the cartographic aspect of it, which combines the art and science of mapping communication, is among the most vaguely defined tasks. There are general principles, such as "As scale decreases, lines should become less irregular. The number of hydrographic features decreases and the streams are shortened appropriately" (Swiss Society of Cartography), but there is no complete set of explicit rules that guide the way to a definite solution. A major part of our effort, therefore, is to analyze existing specifications and maps and try to "see" what a cartographer thinks about when applying different actions to generalize maps. The outcome of the analysis is a set of explicit rules and logical steps defined according to our understanding about human actions in generalization, which have made the implementation of new generalization tools possible. (See Generalization with Workstation ArcInfo.)

To implement the rules and steps of generalization, the challenge is to develop algorithms that mimic human vision, decision, and action. Some simple algorithms are purely based on mathematical models and geometry, but more advanced algorithms take into account the characteristics of features such as spatial relationships and patterns. Sometimes a fully automated solution may not be possible owing to the lack of knowledge or technical limitations. In this case, we let the computer do as much as possible and flag wherever possible the incomplete and unresolved features or areas for interactive operations. (See more details in New Generalization Operators, Commands, and Enhancements.) These computer-assisted solutions have led to quite promising results with great time savings (Lee).

The success of generalization processes relies not only on our understanding of the problem and the use of powerful techniques but also on spatial information that supports the decision making. Existing databases may not have been purposely structured and equipped to output data for many themes and at many scales. Geographic features are typically stored in databases as geometric elements: points, line, and polygons, with or without topology and attributes. The main technical limitations in developing generalization tools lay in identifying feature structures and relationships beyond these geometric elements, handling operations involving different feature classes across data layers, and preserving data integrity in features that have been reduced. Without sufficient information about the features, relying on algorithms and computation to solve the problems can lead to expensive processing and inaccurate results; sometimes it may not be possible to reach a solution. Enriching GIS databases with appropriate feature attributes and structures to support automated generalization has become necessary.

Enriching the Database to Support Generalization

One of the remaining issues in generalization is that "if the decision-support information is missing or hard to derive computationally, can we supply it in the database?" (Lee). For generalization, the idea of enriching the database is to make certain information is explicitly available in the database so that the generalization decisions can be easily programmed and therefore compensate for the technical limitations of relying only on algorithm-based solutions. The examples below illustrate typical cases where computer programs cannot determine the proper level of detail for displaying data or the clear relationship among different features for processing. In these cases, additional database information, such as scale range, feature hierarchies and priorities, and structural information, would make direct and precise decisions possible.

Associating Features with Scales

An essential task in generalization is to selectively include or exclude features for a desired output. Such selections are usually based on feature classifications, geographic attributes, and cartographic constraints and are tied to scale. The feature-scale relation varies with the application and the emphases of the output and is mostly a subjective decision. Associating features with the scale range can guide a generalization program to easily and precisely select or display features at an appropriate level of detail or to process a generalization operation accordingly, especially in run time. For example, a type of point feature can be included and represented by point symbols on a map up to a certain scale; beyond that scale, point aggregation could automatically be performed so that they are represented as areas.

Very few geographic databases have scale factors linked to features. One way to store the feature-scale relation is to use an attribute—for example, "scale-range"—in the database. A feature with a scale-range value of 5–25k would mean that it should appear on a map of 1:5,000 to 1:25,000 scales. Another way is to embed the scale range as part of the feature code—for example, fcode-5-25, meaning the same as above. The hard-coded feature-scale information in a database ensures consistent and repeatable generalization decisions and processes.

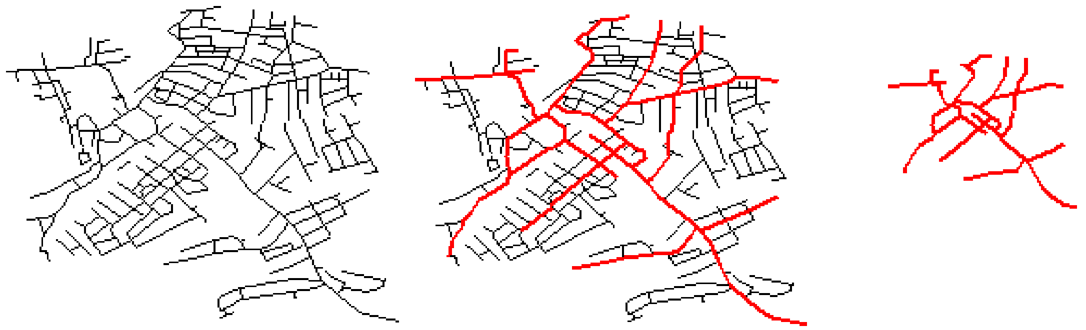
Collecting Features in Meaningful Ways (Adding Intelligence to Features)

Generalization should not always be a geometry-based process, but rather a feature-based process. A group of features may appear in certain spatial patterns. For example, a hydrographic network follows topographic structures, therefore consisting of main streams and branches in a logical hierarchy. Individual trees may be scattered randomly, although they might indicate a certain geographic environment. Buildings in a city are generally divided by street patterns. The essential objective of generalization is to preserve the characteristics and integrity of geographic data while reducing the level of detail in its representation. However, with only the geometry of the features in databases, it would be very difficult to automatically analyze the geographical structures and construct a properly generalized version of it. The examples below illustrate the necessity of intelligent data for generalization.

Case I: Street Network

One common generalization operation is typification, which reduces feature density and simplifies the structural pattern without destroying the overall impression of a feature group at a smaller scale. A computer program can only do this successfully if it has geographic information about the features. For example, a local street network is collected as a set of lines (Figure 2-A) without any ranking. A cartographer wants a subset of the lines (shown in red in Figure 2-B) as the main streets that best represent the street network at a reduced scale (Figure 2-C). It would be difficult to automatically recognize these main streets and form the typified version of the network. Although the street length, connectivity, number of connections, and spacing between streets can perhaps be used as criteria in determining the main pattern automatically, the result might not match the geographic reality. It would be helpful if the digitized lines were ranked.

Figure 2
Digitized Street Network and the Expected Typification



A. A street network to be generalized

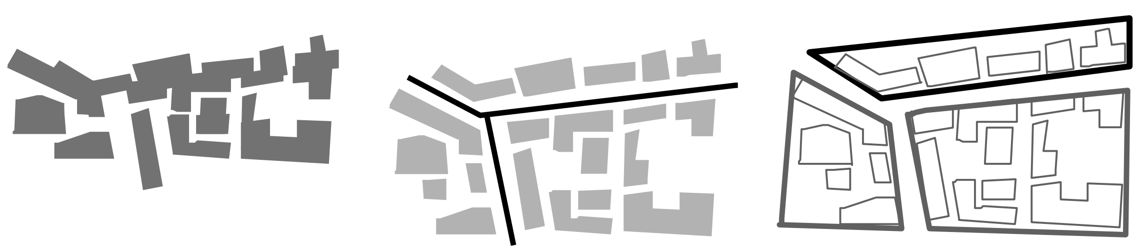
B. Expected typification shown as thicker red lines

C. The expected result at a reduced scale

Case II: Buildings in Street Blocks

In area aggregation, another common type of generalization, individual area features in close proximity or sharing the same geographic attributes are combined into larger areas. Quite often this process has a restriction that the aggregated results cannot spatially cross another type of feature. This is called constrained aggregation. Taking urban areas at large scales as an example, aggregation of buildings can be constrained by main streets. Many maps show buildings only (Figure 3-A) and rely on human eyes to interpret where the major and minor streets are (Figure 3-B) and which buildings belong to what street blocks (Figure 3-C), if aggregated.

Figure 3
Example of Buildings in Street Blocks and Desired Aggregation Result



A. Buildings in urban areas

B. Human-interpreted major streets

C. Street blocks and desired building aggregation

Technically it is not difficult to group buildings automatically according to distance and to form new areas. However, to avoid crossing main streets when aggregating buildings, the street data has to be either available in the database or derivable so that the aggregation process can use it as a constraint. If a street layer exists in the database, spatial analysis functions such as INTERSECT in the Workstation ArcInfo system can detect streets interfering with building aggregation and aggregates only buildings on the same side of the street. (See more details in Deriving Procedures to Solve Generalization Problems.) Some researchers have suggested constructing and maintaining building–street relationships to support constrained aggregation (Regnauld, Jones et al.). Others proposed partitioning buildings by streets so that buildings contained within a street block are aggregated independently from others (Brazile and Edwardes). Without street data, however, an alternative way would be to assign buildings with feature codes or attribute values according to street blocks.

Enriching databases for generalization calls for long-term design and planning. You can store road widths to make it easier to collapse road casings to centerlines, flag landmark features to be preserved, rank features in the order of importance to be considered in resolving spatial conflicts, and so on. In other words, some rules, spatial relationships, and criteria used to define the end products can be turned into explicit attributes and attached to features. Having meaningful and intelligent data could help the automated process select features, recognize feature relationships, and produce more accurate results in a more efficient way than the algorithmic computation.

Generalization with Workstation ArcInfo

Workstation ArcInfo contains a full range of geoprocessing tools to transform data from the database to a specific output including tools used in generalization, selection of features, raster and vector data conversion, spatial analysis with buffering and overlapping, and so on. However, special tools are needed to deal with tasks specific to generalization: tools to compress data, to reduce the level of detail in different types of features, and to satisfy the requirements for both database generalization and cartographic generalization. The geoprocessing tools and special generalization tools can be made up as procedures to solve more complicated generalization problems.

New Generalization Operators, Commands, and Enhancements

The generalization tools developed in the last few years have been designed for the data model and software technology of the Workstation ArcInfo system. We have focused on the most requested generalization functions for outputs at large- to medium-scale range. In this section, each of the following new functions will be described in more detail.

- BENDSIMPLIFY operator—released in ArcInfo 7.1.2
- ORTHOGONAL operator—released in ArcInfo 7.2.1
- BUILDINGSIMPLIFY command—released in Workstation ArcInfo 8.0.1
- FINDCONFLICTS command—released in Workstation ArcInfo 8.0.1
- CENTERLINE command—released in Workstation ArcInfo 8.0.1
- AREAAGGREGATE command—released in Workstation ArcInfo 8.0.2
- Enhancements to the GENERALIZE command
  - A minor enhancement released in Workstation ArcInfo 8.0.1
  - A major enhancement to be released in Workstation ArcInfo 8.1

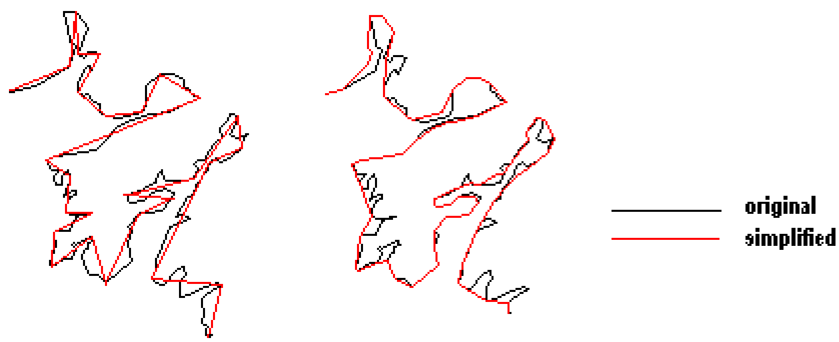
BENDSIMPLIFY Operator

BENDSIMPLIFY is a line simplification operator using an in-house algorithm (Wang). It can be specified through the following commands:

- GENERALIZE in ARC and ARCEDIT
- WEEDOPERATOR in ARCPLOT

One might ask: "There exist many line simplification routines. Why try a new one, and how is this better than the others?" Well, the early-developed algorithms (Lang, Douglas and Peucker) simplify a line by keeping the so-called critical points that depict the essential shape of a line and removing all other points. They are effective for data compression and easy to implement. ArcInfo, along with many other GIS software programs, has adopted the Douglas–Peucker algorithm (named POINTREMOVE in our software) for removing redundant points along lines and reducing data volume. However, the resulting lines appear angular and could self-cross (Figure 4-A). BENDSIMPLIFY reduces a line by detecting and removing extraneous bends from the original line and therefore preserving the main shape of the feature and cartographic quality (Figure 4-B).

Figure 4
Line Simplification Operators



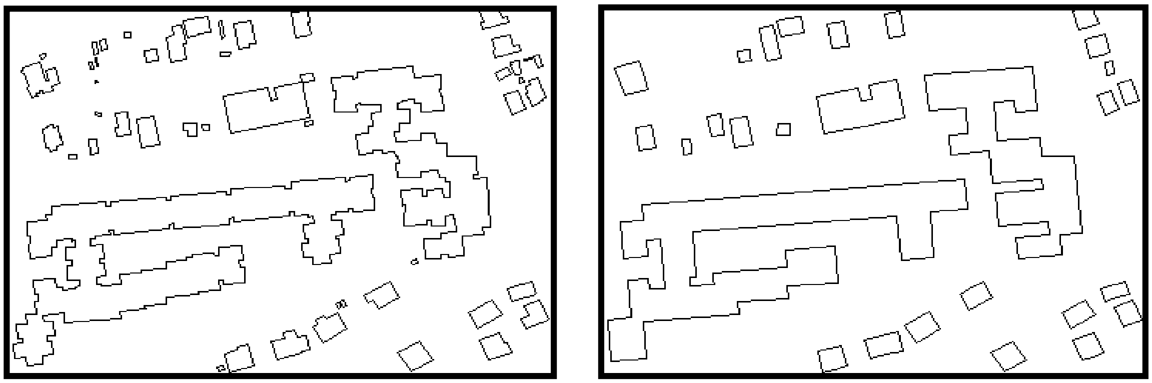
A. POINTREMOVE **B. BENDSIMPLIFY**

ORTHOGONAL Operator and BUILDINGSIMPLIFY Command

The ORTHOGONAL operator in the ARCPLOT™ command, WEEDOPERATOR, and the ARC command, BUILDINGSIMPLIFY (BDS), simplifies buildings or other features with mostly square corners by reducing details in their boundaries while maintaining their essential shape and size (Wang and Lee). Buildings are generally orthogonal areas; therefore, simplification preserves and enhances orthogonality (Figure 5).

J-8512

Figure 5
Before and After Simplification



The ORTHOGONAL operator was added to the WEEDOPERATOR command and supported by polygon-drawing commands to draw simplified building polygons. BUILDINGSIMPLIFY was created so that the simplified buildings can be stored in a new coverage with attributes.

BUILDINGSIMPLIFY works well on individual buildings. Although to a certain extent buildings connected with straight, near-parallel lines (such as dividing walls) can be simplified, it is not suitable to use the building simplification function on features such as parcels or county boundaries. These features may have some orthogonal corners, but they tend to be connected in complicated ways. (See details in the command reference in Workstation ArcInfo Online Help.)

The output coverage from BUILDINGSIMPLIFY will contain simplified buildings as preliminary regions with two new items, BDS-STATUS and BDS-GROUP. The item BDS-STATUS uses the numbers 1 through 5 to record the simplification status.

1—simplified separate building
2—separate building partially simplified due to spatial conflict
3—a short side found in the resulting building
4—simplified or partially simplified buildings connected with straight lines
5—not simplified

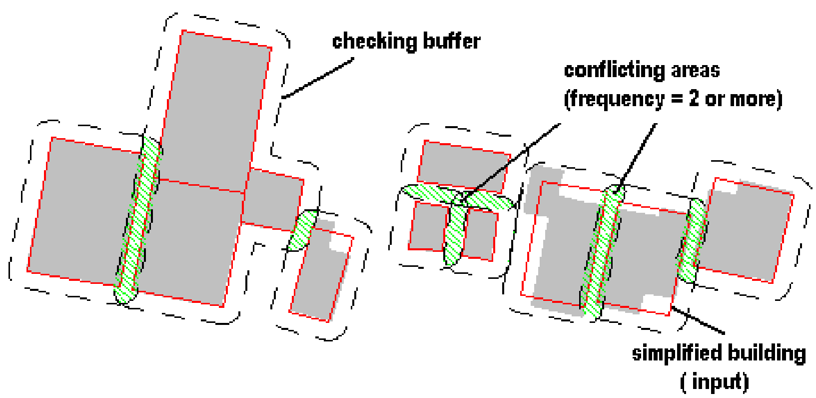
The item BDS-GROUP stores a unique positive value for each group of connected buildings. A single building will receive a BDS-GROUP value of zero. A single building with a hole will receive a unique negative value for both outer and inner boundaries. (See details in the command reference in Workstation ArcInfo Online Help.) The status and group information are the feedback from the automatic process and can be used to facilitate postediting or postprocessing. You can easily select features by these item values and check the quality of the result or perform necessary interactive editing.

FINDCONFLICTS Command

FINDCONFLICTS is a new ARC Atool command. It takes the simplified buildings (see above) as input and finds where they overlap or are too close to each other based on a specified distance (in order to maintain a minimum spacing between features on a map) and the BDS-GROUP item values.

To find the spatial conflicts, region buffers are created around each building or group of connected buildings. Overlapping buffers indicate a conflict. An output will then be produced, storing these region buffers with an item FREQUENCY for polygons. A polygon gets a FREQUENCY value of 2 or more according to how many region buffers overlap (Figure 6). All nonconflicting areas receive a FREQUENCY value of 1.

Figure 6
Finding Spatial Conflicts by Buffering



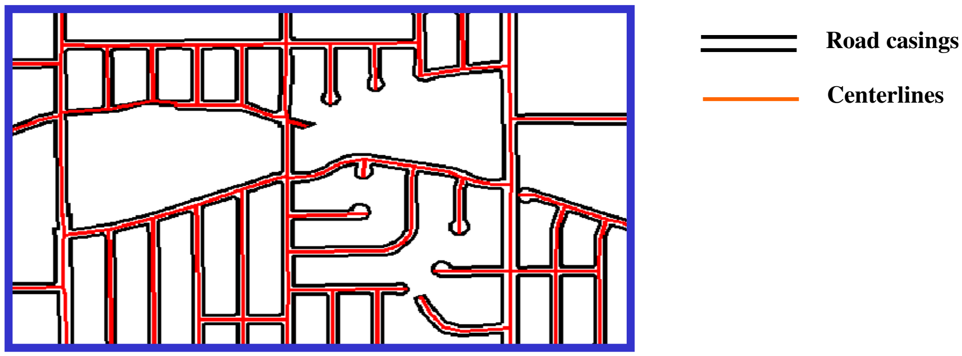
Since FINDCONFLICTS requires the information and data structure produced by BUILDINGSIMPLIFY, you cannot use it to detect conflicts in just any data.

CENTERLINE Command

The CENTERLINE command, implemented in ARC and ARCEDIT™ software, produces centerlines (single lines) from relatively regular dual-line features, such as road casings, based on specified width tolerances (Figure 7).

J-8512

Figure 7
Creating Centerlines



The output coverage will have an item, LTYPE, in the .aat file to flag line types. The created centerline will have an LTYPE value of 1. Unused lines (such as a single casing or casings wider than the specified range) and outlines around complicated intersections will be flagged with an LTYPE value of 2 for editing them further.

The centerline process will partition input data that exceeds 500 arcs. Centerlines are created for each partition and then merged. The partition lines will be included in the result with LTYPE = 3 so that you can check the connections along these lines.

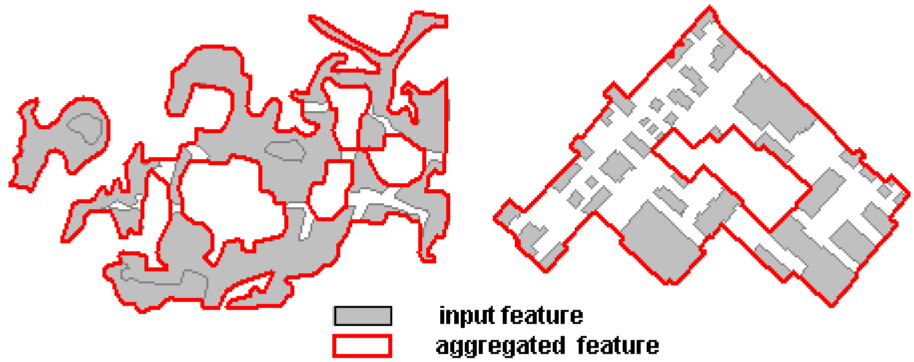
The resulting centerlines are linked to their source casings; therefore, it is easy to derive attributes, such as road names and other information, from input to output. (See details in the command reference in Workstation ArcInfo Online Help.)

AREAAGGREGATE Command

AREAAGGREGATE is an ARC Atool command that combines adjacent and disjoint polygonal features in close proximity into new area features and preserves the distinctive characteristics of the features whether orthogonal or nonorthogonal (Figure 8).

J-8512

Figure 8
Area Aggregation



A. NONORTHOGONAL **B. ORTHOGONAL**

The AREAAGGREGATE command first converts the input polygons to grid (raster), and then uses GRID functions EXPAND, SHRINK, and so on, to group features within the specified distance of each other. The result is then converted back to vector with proper construction of new boundaries.

The choice of cell size for raster and vector conversions depends on the feature type, source data resolution, and intended output data resolution. For orthogonal features (building-like features), set the cell size so that the shortest side of the feature can be made up by two or three cells from which a vector line can be reconstructed. For natural features such as forests and soils, use a cell size that will not cause positions to shift too much but that is not so small as to make a very long processing time and take up a lot of storage space.

The output coverage of AREAAGGREGATE contains new areas as preliminary regions. It also has a one-to-many relation table that links the new regions to source polygons so that attributes can be derived. (See details in the command reference in Workstation ArcInfo Online Help.)

Enhancements to the GENERALIZE Command

The GENERALIZE command simplifies lines or polygon boundaries by removing extraneous details from them without destroying their essential shapes. It has long been known that the GENERALIZE command can produce label errors, either no labels or multiple labels, when it simplifies polygon boundaries. It can also produce topological errors, including line crossing, line overlapping, and zero-length lines, which can also cause label errors. These problems have cost a lot of manual checking and editing time on the user's end. We have taken steps to address these problems.

Correcting Some Label Errors

A label point could fall outside its polygon after the arcs that form the polygon are simplified using the GENERALIZE command. Therefore, this polygon gets a "no label" error, and the adjacent polygon gets a "multiple labels" error. Also, GENERALIZE can produce "collapsed" polygons—that is, it simplifies a small polygon to a point (a line with zero length) or a two-arc polygon to a two-point line. Such polygons will disappear when building polygon topology, but the labels remain and become multiple labels in the neighbor polygons. An enhancement was made for Workstation ArcInfo 8.0.1 to correct the above label errors. A label falling outside its polygon will be moved inside the nearest line segment in the polygon boundary. The label of a polygon that has disappeared will be eliminated.

Fixing the Root of the Problems—Topological Errors

In addition to the above cases, where lines are simplified so much that the polygons formed by these lines disappear, another topological error, crossing lines produced by the simplification process, also introduces new polygons with no labels. One of the essential rules of simplification is to preserve topology—that is, a polygon should remain a polygon and a line to the west of another line should remain to the west after simplification. An option, {NOERRORCHECK | ERRORCHECK}, has been added to the ARC GENERALIZE command for the upcoming Workstation ArcInfo 8.1 release. This option specifies whether to check for topological errors or not including line-crossing, line-overlapping, zero-length lines (or collapsed polygons), and polygon holes falling outside of their polygons.

When NOERRORCHECK (default) is used, GENERALIZE will act as it has always done—that is, it will not check for topological errors. If ERRORCHECK is specified, the command will find and avoid errors generated by the line simplification. If any topological errors are found, the arcs involved will be regeneralized using a reduced tolerance. Then the result will be checked for topological errors again. The process iterates until no more errors are found.

If the input coverage has an arc attribute table (in_cover.aat), the out_cover.aat will contain a new item, TOLFLAG, which stores the tolerance in decimal numbers used for each arc. Tolerances smaller than the weed_tolerance indicate arcs that are undersimplified to avoid line errors. (See details in the command reference in Workstation ArcInfo Online Help.)

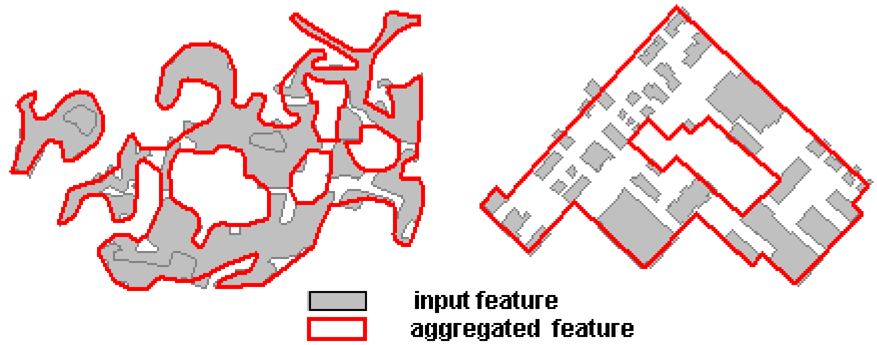
This enhancement complies with generalization rules and eliminates topological errors and label errors caused by the process. Use BUILD to obtain polygon topology instead of CLEAN, which could introduce new label errors and sliver polygons.

Deriving Procedures to Solve Generalization Problems

In deriving a reduced data set from input data, each feature will go through either a few straightforward steps or require a complicated procedure to arrive at the desired form. For example, to transform individual features into an areal form with reduced complexity may take the following two steps: use AREAAGGREGATE to obtain the combined areas (as in Figure 8) and then apply the proper simplification operator to get simplified boundaries (Figure 9).

J-8512

Figure 9
Simplified Boundaries After AREAAGGREGATE Shown in Figure 8



A. Nonorthogonal features with boundary simplified by GENERALIZE (BEND)

B. Orthogonal features with boundary simplified by BUILDINGSIMPLIFY

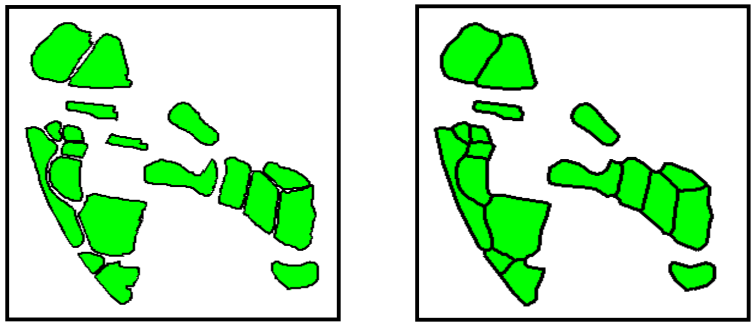
Other possible ways of chaining generalization steps can be set according to the feature type and the criteria. For example, to generalize rivers, you may first use SELECT or RESELECT by attribute to select only main streams; then apply GENERALIZE with POINTREMOVE or BENDSIMPLIFY to simplify them; finally, use SPLINE (in ARCEDIT) to smooth the resulting lines.

Complicated generalization problems can be solved by longer procedures set for specific requirements. In response to the benchmark requirements from Hong Kong (HK) Land Information Center (LIC) on converting a 1:1,000 scale database to a 1:5,000 scale database, a number of generalization tools were needed to simplify and combine features for the reduced level of detail (HK LIC). With limited time and little new development, a few special procedures were derived using existing geotools to obtain the desired generalized results. Among them, the procedures for area extend and constrained polygon aggregation described below were the most challenging and interesting. The BUFFER command involved in both procedures for feature grouping and the program for restoring orthogonal shapes in the second procedure may be replaced by the new command, AREAAGGREGATE, in Workstation ArcInfo 8.1.

Area Extend

Area extend takes place where two polygon features need to remain distinct, but the space between them cannot or need not be represented at the reduced scale; both polygons are, therefore, extended so that their boundaries are merged. Figure 10 shows that the space between ponds (a feature in the Hong Kong Land Information Center project) disappeared as the result of area extend.

Figure 10
Before (Left) and After (Right) Area Extend
(A small polygon was eliminated.)



In order to achieve the desired result, the key actions are to group features by distance and to find the middle lines between features. The procedure for solving this task uses a combination of raster and vector techniques as follows:

BUFFER the pond polygons with a positive buffer distance (outward).
This results in an aggregated polygon.

BUFFER that polygon with the same distance inward (negative).
This makes the aggregated polygon remain connected even though it has returned to the source polygon positions (Figure 11-A).

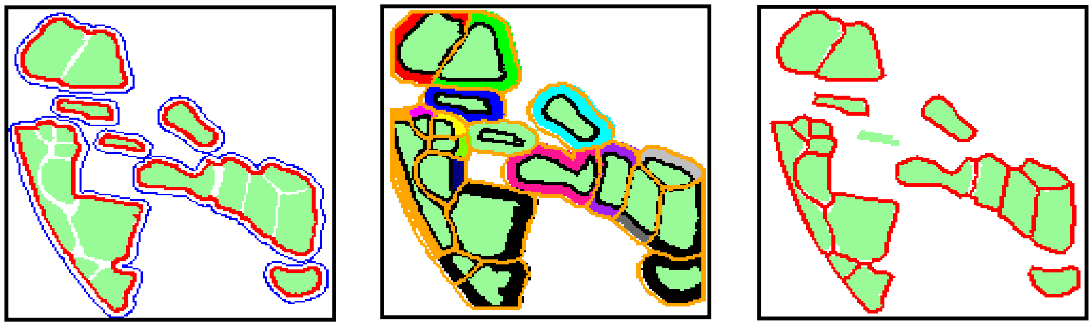
In GRID, use LINEDIST to convert the polygons (which are vector) to raster form and expand the polygons to a given maximum distance.
This finds the middle lines between polygons.

Convert the expanded polygons, including the lines where two polygons should meet, back to vector form (Figure 11-B).

CLIP the expanded polygons by the aggregated (buffered) polygons.
APPEND the polygon meeting lines with the aggregated polygon to form the desired results (Figure 11-C).

BUILD polygon topology for the output coverage.

Figure 11
The Procedure of Area Extend



A. Grouping **B. Finding middle lines** **C. Resulting polygons**

Constrained Area Aggregation

One of the key rules of generalization is to preserve spatial relationships among features. Constrained aggregation is a typical example of maintaining geographic integrity in generalization. As mentioned earlier, area aggregation needs to take a constraint feature into account in some cases. For example, when combining buildings in the HK LIC project, some buildings are close enough to each other by distance but located on either side of a road. If the road is a constraint feature, then the buildings should be aggregated only on either side of it, not across it.

Given that the building is the feature to be aggregated and the road is the constraint feature, the following procedures were used:

BUFFER the building polygons with a positive buffer distance (outward).
This results in an aggregated polygon.

BUFFER that polygon with the same distance inward (negative).
This makes the aggregated polygon remain connected even though it has returned to the source polygon position.

INTERSECT the aggregated polygon with roads (constraint features).
The roads divide the aggregated polygon in parts.

Form regions for the separated polygons on both sides of the road.

REGIONBUFFER the building polygons with a positive buffer distance (outward).
This results in aggregated polygons of the correct grouping of buildings without crossing the roads.

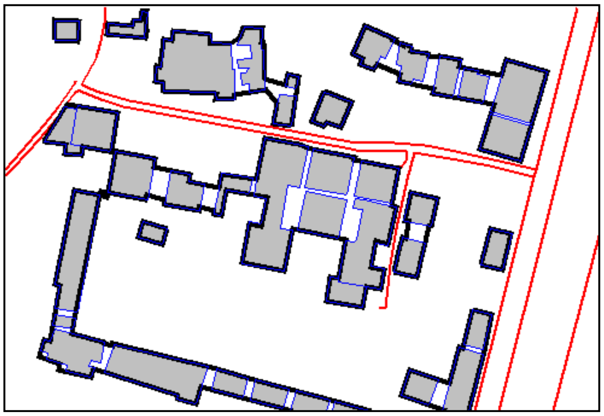
REGIONBUFFER the above result polygon with the same, but negative, distance (inward).
This makes the above aggregated (buffer) polygon return to source polygon positions and remain connected.

Restore orthogonal shapes in the aggregated (REGIONBUFFER) polygons by the C program written for the benchmark.

BUILD polygon topology for output coverage.

The constrained aggregation result is shown in Figure 12. Buildings within the aggregation distance are aggregated only if they do not cross the roads.

Figure 12
Building Aggregation Constrained by Road



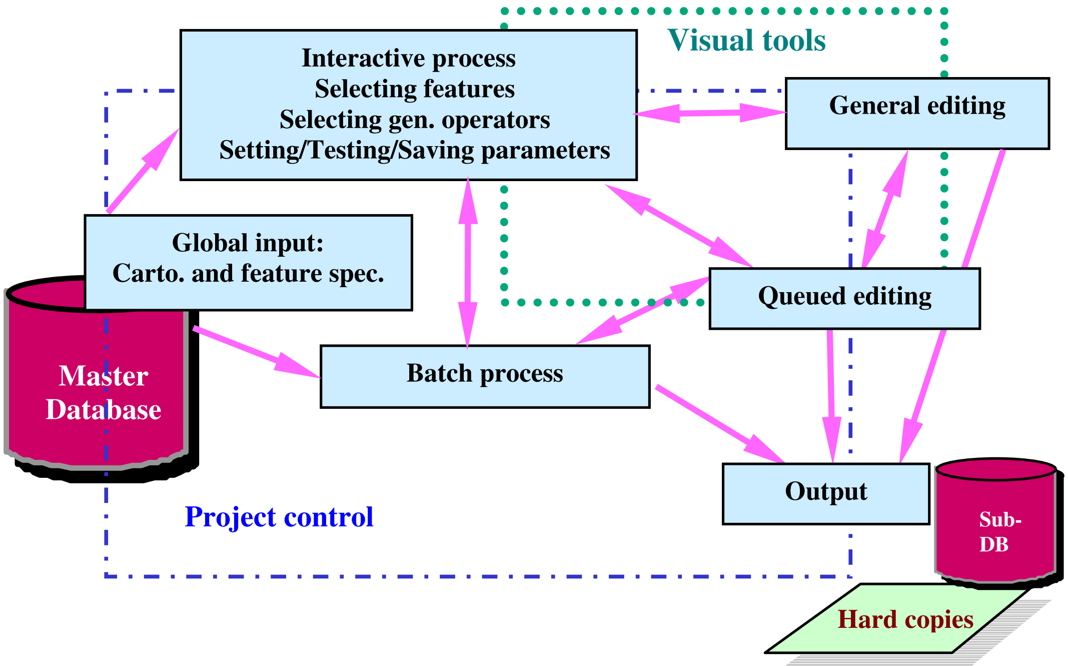
Conclusions and Looking Ahead

With GIS functionality and the newly developed generalization tools in Workstation ArcInfo, several common generalization tasks can now be done in more automated ways with less cost in labor and time. We will continue to provide necessary enhancements, derive procedures and solutions for special cases, and research the remaining challenges.

The new generation of ArcInfo GIS will address the integration of map generalization capabilities. Our ultimate goal is to support any data transformation and map production with maximum automation, flexibility, and productivity.

Figure 13 shows our vision of how generalization can work in GIS. It will allow interactive and batch processes for selecting, analyzing, generalizing, and postprocessing data based on user-supplied specifications. A set of generalization operators will be implemented to reduce the level of detail in all feature types. If an automated process could not produce complete results due to conflicts or complex situations, an editing queue will be generated with flags to question areas or features. The queued editing and other general editing can be performed in the interactive environment where a set of visual tools will be available to assist the analysis and modification of data. All parameters and process sequences can be saved, modified, and retrieved easily for repeated use. A generalization session will be managed with security, consistency, and efficiency.

Figure 13
Generalization Framework



From the system design to the development of methods to solve difficult problems, such as spatial conflicts, there are many challenging areas in GIS-based map generalization. In the short term, we will focus on the most demanded requirements and create an easy-to-use environment that facilitates generalization processes. We want to welcome users' input and suggestions.

References

Brazile, Francisco and Alistair Edwardes, 1999, "Organizing Map Space for Generalization Through Object-Primary Partitioning," working paper for the ICA '99 Workshop on Progress and Development in Automated Map Generalization, Ottawa, Ontario, Canada, August 12–14.

Douglas, David H. and Thomas K. Peucker, 1973, "Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature," *The Canadian Cartographer,* Vol. 10, No. 2, pp. 112–122.

ESRI, 1996, "Automation of Map Generalization: The Cutting-Edge Technology," technical paper. It can be found in the Technical Papers section of ArcOnline at http://arconline.esri.com.

Jones, Christopher B. et al., 1995, "Map Generalization with a Triangulated Data Structure," *Cartography and Geographic Information Systems,* Vol. 22, No. 4, pp. 317–331.

Land Information Center, Land Department of Hong Kong, 1996, "1:5,000 Mapping Specifications," internal document, p. 19.

Lang, T., 1969, "Rules for the Robot Draughtsmen," *The Geographical Magazine,* Vol. 42, No. 1, pp. 50–51.

Lee, Dan, 1996, "Making Database Support Map Generalization," GIS/LIS '96 Proceedings, Denver, Colorado, USA, pp. 467–480.

Lee, Dan, 1999, "New Cartographic Generalization Tools," ICC '99 Proceedings (on CD), Ottawa, Canada.

Morehouse, Scott, 1995, "GIS-Based Map Compilation and Generalization," *GIS and Generalization—Methodology and Practice,* GISDATA 1, Taylor & Francis, pp. 21–30.

Morrison, Joel L., 1999, "The State of Government Cartography in 1998," *Cartography and Geographic Information Sciences,* Vol. 26, No. 3, pp. 167–199.

Regnauld, Nicolas, 1996, "Recognition of Building Clusters for Generalization," *Proceedings of 7th International Symposium on Spatial Data Handling,* Section 4B, pp. 1–14.

Swiss Society of Cartography, 1987, *Cartographic Generalization—Topographic Maps,* Second Edition.

Wang, Zeshen, 1996, "Manual Versus Automated Line Generalization," GIS/LIS '96 Proceedings, pp. 94–106.

Wang, Zeshen, and Dan Lee, 2000, "Building Simplification Based on Pattern Recognition and Shape Analysis," to be published in the *Proceedings of the Ninth Spatial Data Handling Symposium.*

Weibel, Robert, and Christopher B. Jones, 1998, "Computational Perspectives on Map Generalization," *GeoInformatica,* Vol. 2, No. 4, Kluwer Academic Publishers, pp. 307–314.