"HENRI COANDA"
AIR FORCE ACADEMY
ROMANIA

GERMANY

"GENERAL M.R. STEFANIK"
ARMED FORCES ACADEMY
SLOVAK REPUBLIC

INTERNATIONAL CONFERENCE of SCIENTIFIC PAPER
AFASES 2011
Brasov, 26-28 May 2011

ALGORITHMS FOR MAP GENERALIZATION WITH ARCGIS SOFTWARE

Florin Virgil DULGHERU

Geoint Center, Military Technical Academy, Bucharest, Romania

Abstract: Generalization tools allow cartographers to alter geographic feature placement and visualization parameters. Generalization is used when you need to derive small-scale maps from large-scale data. Large-scale maps cover a small physical area, whereas small-scale maps cover a larger physical area. Small-scale maps are usually rendered with less detail, and this process is known as generalization. Generalization requires geoprocessing algorithms and problem solving, which are performed at the topology level before feature symbolization. ArcGIS software solution, development by ESRI USA, 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.

1. INTRODUCTION

The generalization tools developed in the last few years have been designed for the data model and software technology of the ArcGIS system. The cartographers have focused on the most requested generalization functions for outputs al large to medium scale range. In this paper, each of the following new functions will be described in more detail: bendsimplify operator, orthogonal operator, building-simplify command, findconflicts command, centerline command, areaaggregate command and generalize command.

2. GENERALIZATION TOOLS

2.1 Bendsimplify Operator. 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) for removing redundant points along lines and reducing data volume. However, the resulting lines appear angular and could selfcross (Figure 1a).

BENDSIMPLIFY reduces a line by detecting and removing extraneous bends from

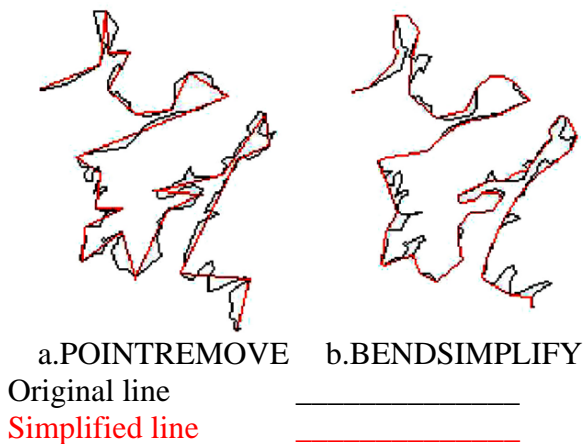the original line and therefore preserving the main shape of the feature and cartographic quality (Figure 1b).

a.POINTREMOVE   b.BENDSIMPLIFY
Original line   _____
Simplified line   _____

Figure 1. Line Simplification Operators

2.2 Orthogonal operator and building-simplify command. The ORTHOGONAL operator in the ARCPLOT command, WEED-OPERATOR, 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 (algorithm Wang and Lee). Buildings are generally orthogonal areas; therefore, simplification preserves and enhances orthogonality. 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.
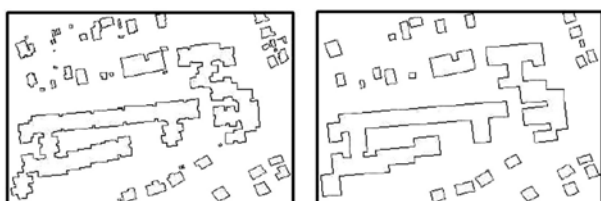

Figure 2. Before and after simplifications

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. The output coverage from BUILDINGSIMPLIFY will contain simplified buildings as preliminary regions with two new items, BDS-STATUS and BDS-GROUP. The item BDSSTATUS 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. 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.

2.3 Findconflicts command. FIND-CONFLICTS is a new ARC tool command. It takes the simplified buildings 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.
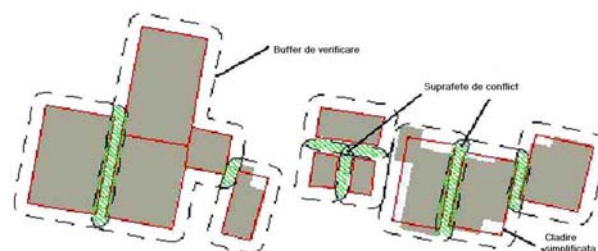

Figure 3. Finding spatial conflicts by buffering

To find the spatial conflicts, region buffers are created around each building or group of

"HENRI COANDA"
AIR FORCE ACADEMY
ROMANIA

GERMANY

"GENERAL M.R. STEFANIK"
ARMED FORCES ACADEMY
SLOVAK REPUBLIC

INTERNATIONAL CONFERENCE of SCIENTIFIC PAPER
AFASES 2011
Brasov, 26-28 May 2011

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. All nonconflicting areas receive a FREQUENCY value of 1. Since FIND-CONFLICTS requires the information and data structure produced by BUILDING-SIMPLIFY, you cannot use it to detect conflicts in just any data.

2.4 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. 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.
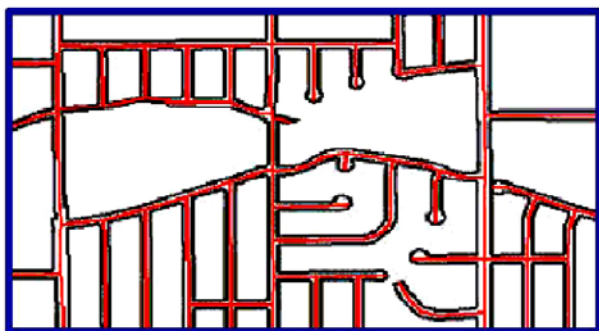

Figure 4. Create centerlines

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.

2.5 Areaaggregate command. AREA-AGGREGATE is an ARC tool 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.
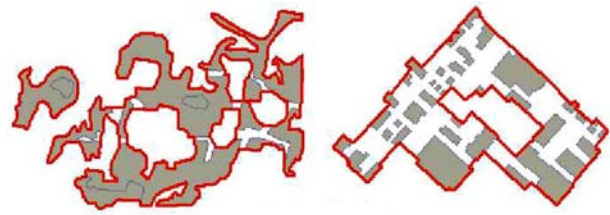

Figure 5. Areaaggregation

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 AREA-AGGREGATE 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.

2.6 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. 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.

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 Workstation ArcInfo 8.1 release. This option specifies whether to check for topological errors or not including linecrossing, 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.

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.

3. CONCLUSIONS & ACKNOWLEDGMENT

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.

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

REFERENCES

1. ESRI, *Automation of Map Generalization – The Cutting-Edge Technology*, White Paper Series, May, 1996;

2. ESRI, *Map Generalization in GIS: Practical Solutions with Workstation ArcInfo Software*, White Paper Series, July, 2000;