

Entorno para la generalización cartográfica en Sistemas de Información Geográfica

Miguel R. Luaces, José R. Paramá, Isabel Pérez-Urria Lage

Laboratorio de Bases de Datos
Universidade da Coruña
Campus de Elviña S/N
{luaces, parama, iperezurria}@udc.es

Resumen

En el marco de los sistemas de información geográfica la generalización cartográfica juega un papel esencial. Por una parte permite construir un mapa más claro y representativo para el usuario, eliminando los detalles superfluos o innecesarios a determinada escala. Por otra parte, en el ámbito computacional las técnicas de generalización permiten reducir la cantidad de datos, minimizando así el tiempo de espera en la visualización del mapa y evitando sobrecargar los recursos. Sin embargo, las herramientas existentes de generalización cartográfica son productos comerciales de elevado coste y no existen herramientas de software libre para realizar estas tareas. Por ello hemos desarrollado un módulo básico de generalización para sistemas de información geográfica basado en técnicas de simplificación que ha sido diseñado de tal manera que nos va a permitir integrar nuevas técnicas de generalización en el futuro. Este módulo fue integrado en dos herramientas de visualización de información geográfica implementadas en su totalidad en software libre, JUMP y gvSIG. Por otra parte, siguiendo los estándares definidos por el Open Geospatial Consortium definimos una extensión del servicio Web Feature Service (WFS) para soportar simplificación, al que llamamos Web Simplification Service (WSS).

Palabras clave: Sistemas de Información Geográfica (SIG), generalización cartográfica, Web Feature Service (WFS), software libre.

1 Introducción

La gran complejidad de la superficie terrestre supone un grave problema para cualquier representación digital ya que una representación exacta implicaría un almacenamiento excesivo de información. A pesar de los recientes progresos en el desarrollo de dispositivos de almacenamiento masivos éstos se ven rápidamente desbordados incluso por representaciones de baja precisión espacial. De esto se deduce que cualquier implementación de un SIG debe hacer uso de la generalización, aproximación y otras técnicas para lograr representaciones geográficas con una precisión adecuada.

El nivel de generalización debe variar en función de la escala, es decir, el contenido del mapa deberá ser reducido a aquello que es necesario y posible representar en función de los cambios de escala. Por ejemplo, en un mapa a escala 1:1.000.000 no podemos mantener todos los detalles que se representarían con una escala 1:100.000 pues la densidad gráfica aumentaría exageradamente. Además, en ocasiones tanto nivel de detalle resulta redundante pues a la hora de visualizarlo en una pantalla varios puntos de la geometría se representarán en un mismo pixel. Debido a la gran cantidad de técnicas que se pueden aplicar y a la heterogeneidad en la morfología de la superficie terrestre, la generalización cartográfica es una de las tareas más complejas que existen en la generación de mapas, además de consumir una gran cantidad de tiempo y recursos.

Dada la gran complejidad en este campo, en este artículo describimos la construcción de una infraestructura básica para la generalización en la que tengan cabida la mayor parte de las técnicas existentes de generalización cartográfica. Por otra parte, en la actualidad existen diversas herramientas alternativas para la visualización y mantenimiento de información geográficas, algunas comerciales y otras ofrecidas como software libre. Hasta lo que alcanzan nuestros conocimientos, ninguna de las alternativas de software libre existentes incluye herramientas para generalización, por lo que es necesario recurrir a software propietario, de coste muy elevado y sin posibilidades de extensibilidad o mejora del mismo. Por esta razón podemos afirmar que la herramienta propuesta en este artículo llena un vacío en cuanto a sistemas de software libre que implementen herramientas de generalización cartográfica.

En este artículo se describe el análisis, diseño e implementación de un componente para la generalización de geometrías en sistemas de información geográfica (SIG) que se integra en dos entornos diferentes: como servicio web independiente, y en una herramienta de visualización de información geográfica. Por ello, en el artículo se consideran tres aspectos independientes. En primer lugar se describe un módulo de generalización en el que se pueden integrar diversos algoritmos y técnicas. A continuación, este módulo de generalización se integra en un servicio web cuyo objetivo es la simplificación de geometrías en tiempo real y que funciona como una pasarela entre la aplicación y un servicio *Web Feature Service* (WFS). Si al servicio se le indica un algoritmo de simplificación proporcionará las geometrías simplificadas, en otro caso será transparente. La comunicación entre los servicios web se realizará utilizando estándares siempre que sea posible (como el *Geography Markup Language* para la representación de información geográfica o utilizando los tipos de datos recomendados por el *OpenGeospatial Consortium*). Por último, se presenta el componente de integración del módulo de generalización en una herramienta interactiva de visualización de información geográfica (como gvSIG o JUMP).

El resto de este artículo se organiza de la siguiente manera: en la Sección 2 se describe el trabajo relacionado que sirve de base a nuestro desarrollo, en la sección 3 se presenta con más detalle el algoritmo que se ha utilizado para simplificar coberturas poligonales del espacio sin generar errores topológicos, en la sección 4 se describe la arquitectura del sistema implementado, en la sección 5 se muestran resultados experimentales de la aplicación del algoritmo, y, finalmente, en la Sección 6 se presentan conclusiones y trabajos futuros.

2 Trabajo relacionado

A la hora de representar cartografía sobre un medio físico o electrónico la generalización cartográfica representa un papel muy importante: se trata de adaptar todos los elementos que aparecen en el mapa a la escala y objetivo del nuevo mapa. El propósito es lograr una representación clara del mapa, eliminando el exceso de información y ciñéndose a lo que es

necesario y posible representar.

El concepto de generalización cartográfica no es nuevo, de hecho, los cartógrafos la han empleado desde siempre de forma manual para representar la superficie terrestre a diferentes niveles de detalle. Pero es con el nacimiento de los sistemas de información geográfica cuando surge la necesidad de buscar algoritmos y soluciones computacionales a esta tarea.

En las últimas décadas hemos experimentado grandes avances en las tecnologías informáticas, tanto en dispositivos de almacenamiento masivo como en capacidad computacional. Sin embargo, el procesamiento de información espacial es una tarea muy pesada hasta para los equipos más modernos. Ya hemos enumerado las ventajas de la generalización a la hora de presentar la información más clara y representativa para el usuario, pero centrándonos en los SIG podemos decir que ésta supondrá una reducción significativa de la carga de datos a transmitir entre los distintos módulos de nuestro sistema, con el correspondiente ahorro en el tiempo de espera en la visualización de los mapas.

De acuerdo con la clasificación de McMaster y Shea (1992) [1] podemos identificar los siguientes métodos de generalización:

- *Simplificación*: por ejemplo, eliminar puntos del contorno de un polígono para crear una geometría más simple.
- *Suavizado*: sustitución de formas agudas y complejas por otras más suaves.
- *Ruptura*: reemplazo de un área por una combinación de puntos y líneas.
- *Agregación*: sustitución de un conjunto de objetos por un número menor de objetos.
- *Fusión*: sustitución de varios objetos por un único objeto.
- *Mezcla*: sustitución de un conjunto de líneas por un menor número de líneas.
- *Refinamiento*: sustitución de un patrón complejo de objetos por un selección que preserva la forma general del patrón.
- *Exageración*: aumento relativo de un objeto para preservar sus características, que se perderían si el objeto se mostrase a escala.
- *Aumento*: mediante la alteración del tamaño y forma de los símbolos.
- *Desplazamiento*: movimiento de objetos de sus posiciones originales para preservar su visibilidad.

Como primera aproximación estas técnicas nos hemos centrado en el método de simplificación de líneas. Existen varias técnicas y algoritmos que implementan este método, uno de ellos es el método propuesto por McMaster en 1987 llamado ventana deslizante de tres puntos [2]: el método elimina puntos que están muy próximos a sus vecinos, o aquellos cuyos vectores forman un ángulo muy pequeño. El método consiste en una ventana que se desliza por la línea y que abarca tres puntos. Los tres vértices se examinan con dos criterios: si la distancia de un vértice al anterior es muy pequeña, el segundo punto se elimina. Por otra parte, si el ángulo entre una línea proyectada que une los dos primeros puntos y la línea que une los dos segundos puntos es pequeño, entonces el tercer vértice puede ser eliminado.

Sin embargo, el algoritmo de simplificación más extendido e implementado en diversos paquetes comerciales es el de Douglas-Peucker [3]. Su funcionamiento es muy simple, y consiste en deshacerse de todos los puntos que se encuentren a una distancia menor que cierta tolerancia de la línea que une los puntos inicial y final de la línea a simplificar.

En nuestro caso hemos aplicado este algoritmo tanto en líneas (carreteras, ríos, calles) como en

polígonos (municipios, comarcas, núcleos de población). Sin embargo, el problema principal con que nos hemos encontrado radica en la incapacidad de este algoritmo para mantener las líneas adyacentes entre polígonos que forman una cobertura del espacio, como ocurre con los municipios, dado que el algoritmo de Douglas-Peucker simplifica de forma independiente cada uno de los polígonos. La aproximación a este problema la abordamos a través de la solución propuesta en el artículo de M. R. Jhonston, C. D. Scout y R.G. Gibb en 1999 [4]. En su artículo proponen utilizar un grafo para la representación de coberturas del espacio y simplificar arcos, en vez de geometrías. Explicaremos tanto el algoritmo de Douglas-Peucker como su adaptación a coberturas poligonales del espacio con más detalle en la siguiente sección.

Siguiendo una filosofía basada en software libre y utilizando estándares del *OpenGeospatial Consortium* (OGC) [5], hemos decidido implementar el sistema de simplificación como una extensión del estándar *Web Feature Service* (WFS) [6] que reciba peticiones con parámetros adicionales de simplificación. WFS es un estándar definido por el OGC que permite a un cliente obtener y actualizar información geográfica codificada en *Geographic Markup Language* (GML) a partir de una o varias fuentes de datos de diverso tipo.

3 El algoritmo de simplificación

Examinando la base de datos de un SIG cualquiera podemos encontrar información espacial de diversos tipos que es preciso simplificar. Podemos realizar la siguiente clasificación:

- *Líneas*: como la red de carreteras, la de ferrocarril, o la red eléctrica, etc.
- *Polígonos*: como los edificios, los núcleos de población, etc.
- *Colecciones*: contienen geometrías de diversos tipos, como líneas y polígonos. En este grupo podemos incluir la hidrografía (líneas para los ríos y polígonos para embalses y lagos).
- *Coberturas del espacio*: son conjuntos de polígonos adyacentes que cubren la totalidad del espacio, por ejemplo las fronteras políticas (municipios, parroquias o provincias)

Debido a esta diversidad se hace patente la necesidad de emplear métodos de generalización específicos orientados a cada tipo de entidad. Sin embargo, como primera aproximación decidimos aplicar el algoritmo de Douglas-Peucker de simplificación de líneas a todos los tipos. Evidentemente, esto tiene algunos problemas, que comentaremos más adelante.

3.1 El algoritmo de Simplificación de Douglas-Peucker

Este algoritmo, propuesto independientemente por Ramer (1972) [7] y Douglas & Peucker (1973) [3] construye una nueva línea simplificada que se mantiene a una distancia predefinida δ de la línea original. A continuación se muestra el funcionamiento de este algoritmo:

1. Unir los vértices inicial y final por una línea recta \overline{ab}
2. Hallar las distancias en perpendicular desde cada vértice hasta la línea \overline{ab}
3. Eliminar los vértices cuya distancia a \overline{ab} sea menor que la tolerancia δ
4. El vértice más distante se elige como el nuevo punto para continuar iterando hasta que no existen vértices a una distancia mayor que δ

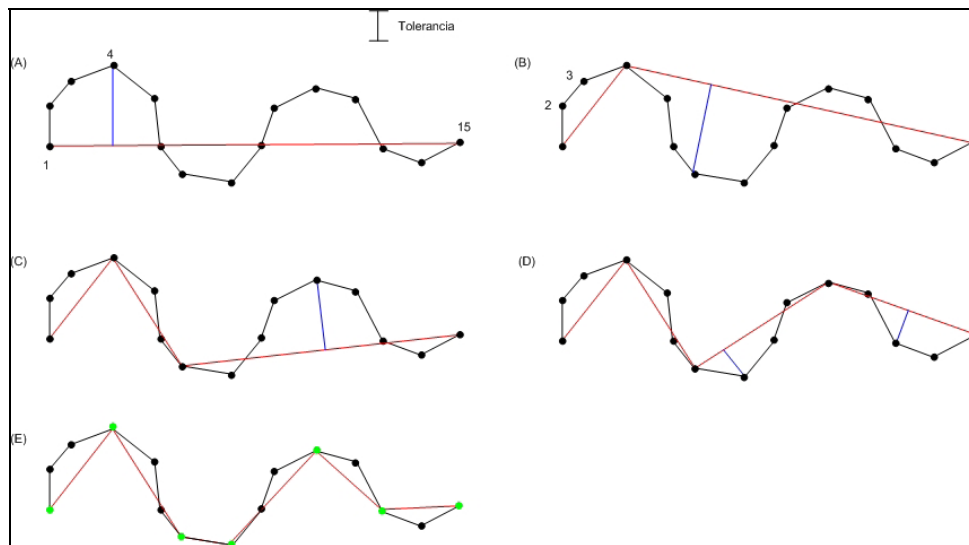


Figura 1: Algoritmo de Simplificación de Douglas-Peucker

En la figura 1 vemos un ejemplo del funcionamiento del algoritmo de Douglas-Peucker. La línea original consta de 15 puntos. En (A) se unen los puntos 1 y 15 con una línea y se busca el punto que se encuentre a mayor distancia de esta línea, en este caso el punto más lejano está marcado en la imagen como 4. La distancia al punto 4 supera la tolerancia δ definida por el usuario, por lo que escogemos este punto como el siguiente para continuar iterando. En (B) se unen los puntos 1 y 4, y vemos que los puntos 2 y 3 están a una distancia menor que la tolerancia δ por lo que pueden ser eliminados. A continuación (B) se unen los puntos 4 y 15 y se repite el proceso. En el último paso (E) ya no hay ningún punto que se encuentre a una distancia mayor que la tolerancia δ , la línea simplificada la obtenemos uniendo todos los puntos que quedaban sin eliminar en cada iteración.

3.2 Simplificación de Polígonos Adyacentes empleando Grafos

A la hora de simplificar coberturas del espacio nos encontramos con el problema de que Douglas-Peucker no mantiene las adyacencias entre polígonos, y obtenemos una imagen como la de la Figura 2b. Vemos que aparecen huecos y solapamientos entre los polígonos. Para la solución de este problema decidimos aplicar teoría de grafos para la representación de las geometrías, tal y como se propone en [4]

La idea básica es identificar cada coordenada del mapa con un vértice del grafo. Los vértices de grado ≥ 3 más un nodo determinado de cada polígono aislado constituyen el conjunto de *nodos*. Definimos una *polilínea* como un camino cualquiera en el grafo y un *arco* como una polilínea entre dos nodos que no contiene a otros nodos.

La técnica utilizada consiste en simplificar los arcos del grafo en vez de los polígonos completos. Al final del proceso se reconstruyen los polígonos uniendo todos los arcos que lo forman. De esta forma aseguramos que cada frontera entre dos polígonos es simplificada una sola vez, evitando que sean simplificadas de forma independiente. El resultado deseado es el que se muestra en la Figura 3b.



Figura 2: (a) Municipios originales. (b) Municipios simplificados con Douglas-Peucker



Figura 3: (a) Municipios originales. (b) Municipios simplificados con la aproximación de grafos

3.3 Eliminación previa de conflictos en las geometrías iniciales

El método explicado en el apartado anterior es muy sensible a la congruencia y consistencia de los datos geográficos. Es muy común que en las representaciones de coberturas del espacio no sean perfectas, sino que existan huecos y solapamientos imperceptibles a la vista entre los distintos polígonos, resultado de una introducción manual de los datos. Esto da lugar a que se construirán en el grafo muchos más arcos que si no existieran dichos huecos, y el algoritmo de simplificación, por tanto, será mucho menos efectivo, ya que mantendrá en las geometrías simplificadas más vértices de los necesarios.

Para solucionar esto se ha decidido implementar un filtrado del grafo construido previo a la simplificación, que consta de dos partes:

- Por una parte, la reducción en la precisión de los datos. En muchos casos, el problema sólo consiste en que la precisión de las coordenadas es excesiva y dos vértices sólo difieren en uno o dos dígitos poco significativos. Por ejemplo, si tenemos las coordenadas 557695,432210577 y 557695,432210576 en el grafo serían insertados como vértices

diferentes.

- Por otra parte, hemos implementado un algoritmo para la eliminación de huecos que se encuentran por debajo de cierta tolerancia definida por el usuario (ϵ).

Algoritmo para la eliminación de huecos entre polígonos que forman una cobertura del espacio:

1. Creación del grafo G correspondiente a la cobertura
2. Crear un índice espacial con todas las aristas de G
3. Para cada vértice de grado 2:
 - 3.1. Buscar los arcos candidatos: creamos un cuadrado alrededor del nodo a una distancia ϵ (el cuadrado tiene lado 2ϵ). Buscamos en el índice espacial los arcos que intersecan con ese cuadrado.
 - 3.2. Eliminamos de la lista de arcos aquellos que pertenecen a la misma geometría que el nodo que estamos procesando y los arcos marcados como borrados en iteraciones anteriores.
 - 3.3. Buscamos el arco más próximo al nodo (midiendo distancia en perpendicular del vértice al arco)
 - 3.4. Dividimos el arco seleccionado en dos y los llevamos hasta el nodo. Sólo creamos nuevos arcos si no existían anteriormente.
 - 3.5. Insertamos los nuevos arcos en el índice espacial y marcamos los antiguos como borrados.

4 Arquitectura del Sistema

La arquitectura básica del sistema se describe en la Figura 4. Los componentes desarrollados por nosotros se muestran sombreados en gris. En la zona inferior izquierda podemos ver el módulo de generalización, al que se accede a través de una fachada. El módulo de simplificación ha sido desarrollado de forma que añadir nuevos algoritmos sea fácil e intuitivo, además de estar pensado como un módulo, de forma que resulta relativamente sencillo integrarlo en distintos entornos. Las herramientas de visualización de información geográfica JUMP y gvSIG (zona superior izquierda) acceden al mismo a través de un *plugin* desarrollado específicamente para cada una de ellas. Por otra parte, en la zona superior derecha tenemos el servicio *Web Simplification Service* (WSS), que obtiene los datos de las *Features* geográficas a través de un *Web Feature Service* (WFS) y que utiliza el módulo de generalización para simplificar las geometrías antes de enviárselas al usuario. Por último, tanto las herramientas de escritorio como el WFS obtienen sus datos a través de una conexión a un sistema gestor de bases de datos con extensión espacial.

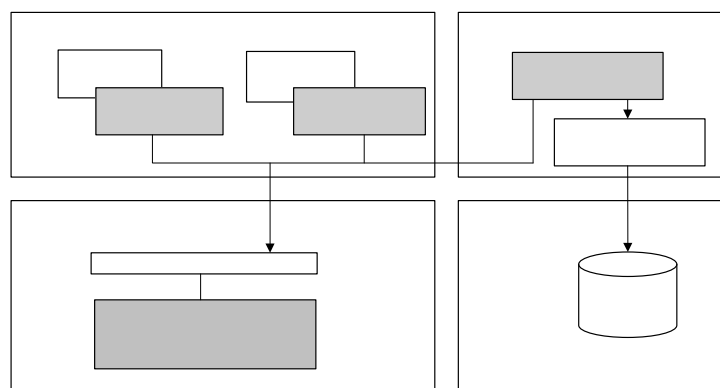


Figura 4: Arquitectura del Sistema

4.1 Web Simplification Service

El servicio *Web Simplification Service* (WSS) implementa la funcionalidad básica de un servidor WFS [6], es decir, puede resolver peticiones HTTP *getCapabilities*, *GetFeatureType* o *GetFeature*, entre otras, que permiten obtener metadatos acerca de las entidades que puede proporcionar el servidor, además de información acerca del tipo de una entidad, o la propia entidad codificada en un documento GML. Además de las operaciones básicas, nuestro servicio consta de una extensión que proporciona entidades con su información geográfica simplificada. Acepta tanto peticiones *GET* como *POST*. Por debajo del WSS funciona un WFS, que accede a los datos mediante una conexión con la base de datos espacial.

Como ejemplos de peticiones a nuestro servidor WSS mostramos los siguientes, en donde vemos que los parámetros de simplificación se concatenan en la URL que constituye la petición al servicio. Si es una petición *GET* a continuación del path del WSS se incluye el atributo *algorithm* para indicar el algoritmo de simplificación, el atributo *params* para indicar los parámetros separados por comas en caso de ser más de uno, y por último se incluirán los atributos de la petición WFS:

```
http://localhost:81/geosimplifier/wss?
algorithm=DouglasPeucker&params=100&
service=wfs& version=1.0.0&request=getfeature&typename=concellos
```

En este caso se trata de una petición *GetFeature* que obtiene todos los municipios simplificados con el algoritmo de Douglas-Peucker con una tolerancia de 100m.

En caso de acceder al servicio mediante una petición de tipo *post* el proceso es muy similar: a continuación del path del WSS se incluirán los atributos *algorithm* y *params* descritos anteriormente. Usando la operación del ejemplo anterior:

```
http://localhost:81/geosimplifier/wss?algorithm=DouglasPeucker&params=100

<wfs:GetFeature service="WFS" version="1.0.0"
  outputFormat="GML2"
  xmlns:topp="http://www.openplans.org/topp"
  xmlns:wfs="http://www.opengis.net/wfs"
  xmlns:ogc="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfs
  http://schemas.opengis.net/wfs/1.0.0/WFS-basic.xsd">
  <Query typeName="topp:concellos"></Query>
</wfs:GetFeature>
```

4.2 Integración en herramientas de visualización

El módulo de generalización ha sido desarrollado para ser integrado fácilmente en entornos de visualización de información geográfica. En concreto lo hemos integrado en JUMP y gvSIG, dos de las herramientas más potentes de código abierto que podemos encontrar hoy en día.

JUMP (Java Unified Mapping Platform) [8] es una aplicación de escritorio especialmente diseñada para procesar y visualizar datos espaciales, y patrocinado por varias organizaciones, entre las que se encuentran GeoConnections, el Centro Canadiense de Información Topográfica- Sherbrooke y el Ministerio de Recursos Naturales de Ontario. Por otra parte, gvSIG [9] es una herramienta de

características similares que ha sido desarrollada por la Generalitat Valenciana. La gran ventaja de gvSIG es la rapidez en el renderizado de la información espacial, lo que la convierte en una potente herramienta para el procesado de grandes volúmenes de datos.

En la Figura 5 se muestra el *plugin* de generalización integrado en JUMP. Éste permite escoger la capa que se desea simplificar, el algoritmo de simplificación y los parámetros (distintas tolerancias y los parámetros que emplean los algoritmos). Las geometrías simplificadas se mostrarán en una nueva capa, que permita al usuario exportar los datos en distintos formatos como *shapefile*, volcarlos a base de datos, etc.

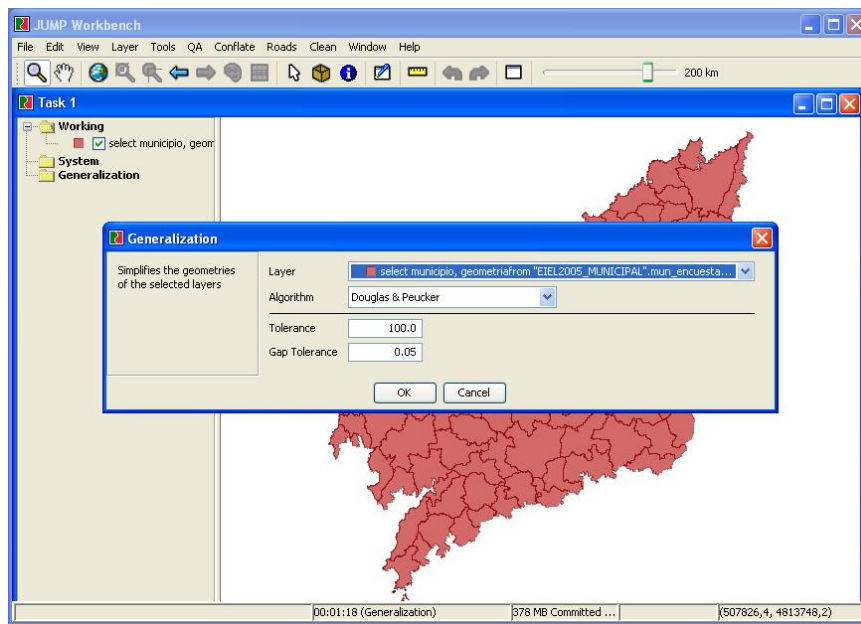


Figura 5: Integración del módulo de generalización en JUMP

5 Resultados experimentales

En esta sección expondremos los resultados obtenidos con algunas de las tablas de la base de datos del proyecto EIEL (Encuesta de Infraestructuras y Equipamientos Locales) [10].

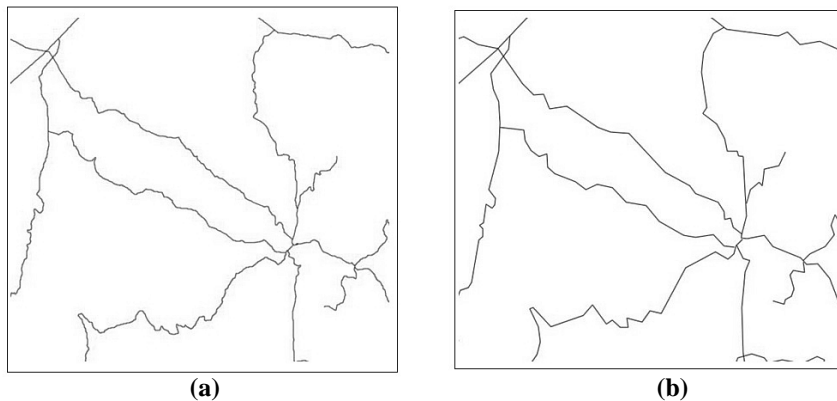


Figura 6: Simplificación de carreteras con Douglas-Peucker y tolerancia 100m

Como primer resultado, se muestra la simplificación de la red de carreteras de la provincia de A Coruña, que son de tipo línea (Figura 6). Para este tipo de datos utilizamos únicamente el algoritmo de Douglas-Peucker. En la Tabla 1 se pueden ver resultados bastante ilustrativos de los porcentajes de reducción de coordenadas que se obtienen con respecto a las geometrías originales:

RED DE CARRETERAS DE LA PROVINCIA DE A CORUÑA		
TOLERANCIA (m)	Nº COORDENADAS	% REDUCCIÓN
Geom. originales	1431725	-
1	547574	61,75%
10	163498	88,58%
100	54584	96,19%

Tabla 1: Porcentajes de reducción en la simplificación de la red de carreteras de la Provincia de A Coruña con Douglas-Peucker y distintas tolerancias

En la Figura 7 se muestra la simplificación de municipios (polígonos adyacentes que forman una cobertura del espacio). La Figura 7a representa las geometrías originales. A primera vista no se observa nada anormal, pero realizando comprobaciones coordenada a coordenada descubrimos que la frontera de la esquina inferior derecha contiene unos 600 huecos, el mayor de los cuales mide 20cm de ancho, por lo que a este nivel de escala (1:10.000) son imperceptibles.

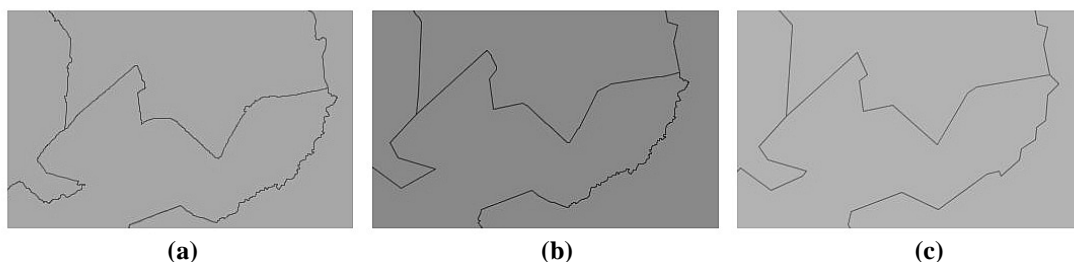


Figura 7: (a) Parte de los municipios originales. (b) Simplificación utilizando grafos y Douglas-Peucker. (c) Simplificación con grafos y Douglas-Peucker y eliminación previa de huecos

Al intentar simplificar con Douglas-Peucker y la aproximación de grafos explicada en el apartado 3.2 vemos que esta frontera apenas se ve simplificada (Figura 7b). Lo que ocurre es que el algoritmo identifica los extremos de los huecos como nodos (vértices de grado ≥ 3) que deben ser mantenidos en la geometría final, por lo que esta aproximación no nos sirve. En la Figura 7c vemos que este problema se soluciona con la eliminación previa de los huecos entre las fronteras. La tabla 2 ilustra la diferencia en los porcentajes de reducción entre ambos algoritmos. Vemos que, lógicamente, en el segundo caso los porcentajes son mayores.

La webEIELv2.0 [10] es un proyecto realizado para la Diputación de A Coruña para la distribución de los datos contenidos en la Base de Datos Territoriales de la EIEL. La primera versión fue puesta en producción y abierta a la totalidad de la comunidad web en julio de 2004, y recientemente, en mayo de 2007 ha sido revisada y ampliada con nuevas funcionalidades. Una de las herramientas principales de la web es el Mapa Temático Interactivo, que permite cargar y descargar capas de información geográfica relativas a la provincia de A Coruña, así como obtener información no geográfica asociada a cada elemento del mapa.

MUNICIPIOS DE LA PROVINCIA DE A CORUÑA				
TOLERANCIA (m)	D&P + GRAFO		D&P + GRAFO + ELIMINACIÓN DE HUECOS	
	Nº COORDENADAS	% REDUCCIÓN	Nº COORDENADAS	% REDUCCIÓN
Geom. originales	497029	-	497029	-
1	180348	63,72%	172322	65,34%
10	63624	87,2%	50891	89,76%
100	25042	94,96%	8811	98,22%

Tabla 2: Comparativa entre los porcentajes de reducción en la simplificación de los municipios de la Provincia de A Coruña con Douglas-Peucker y teoría de grafos, sin eliminar huecos y eliminándolos.

Para esta herramienta sería inviable cargar todas las capas de información con todo el detalle, por eso fue necesario implementar dos soluciones: por una parte cargar únicamente las capas que se consideran necesarias para cada escala de visualización, por ejemplo, las edificaciones únicamente se cargan en escalas inferiores a 1:4000. Por otra parte, la simplificación de geometrías: se aplicaron 4 niveles diferentes de simplificación, cada uno de los cuales se carga en una escala diferente. En la Figura 8 se puede ver el mapa interactivo en el que están cargadas, entre otras, las siguientes capas simplificadas: municipios, núcleos de población, suelo urbano, suelo residencial y carreteras.

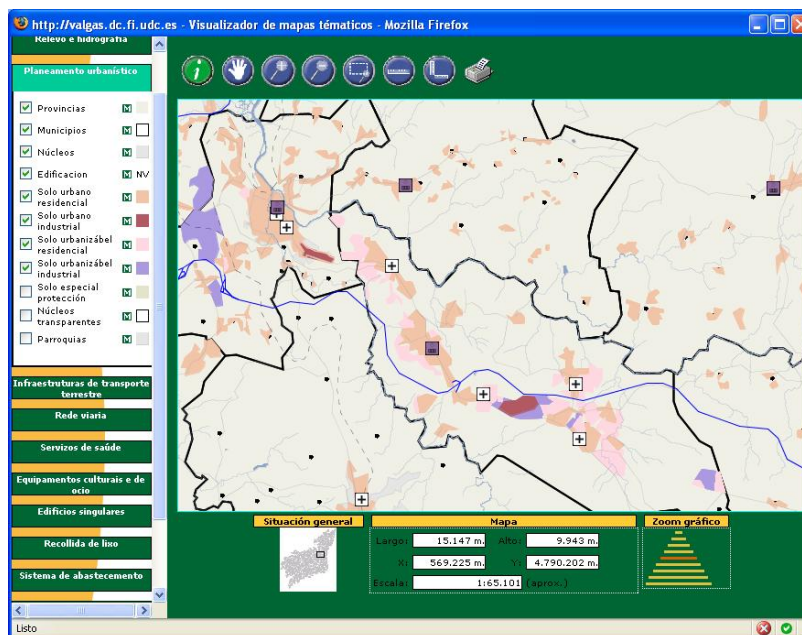


Figura 8: Mapa Temático Interactivo de la webEIEL v2.0

6 Conclusiones y trabajo futuro

En este trabajo hemos desarrollado un marco básico que permita abarcar diversos métodos de generalización, integrándolo en dos herramientas de visualización de información geográfica de código abierto implementadas en Java, como JUMP y gvSIG. Siguiendo los estándares del OGC,

hemos realizado una extensión de la operación *GetFeature* del estándar WFS para soportar generalización.

En el campo de los algoritmos de generalización hemos empleado un método sencillo para la simplificación de líneas mediante reducción de puntos, como Douglas-Peucker, introduciendo una variante empleando teoría de grafos en el caso de coberturas del espacio, como las fronteras políticas entre municipios, comarcas, etc.

Somos conscientes de todo el trabajo que aún nos queda por hacer en este campo. Si tomamos como referencia los métodos implementados en paquetes comerciales, como Dynagen, de Intergraph podríamos enumerar diversas herramientas esenciales para mantener la calidad de la cartografía simplificada, por poner algún ejemplo, métodos de suavizado para dar un aspecto menos anguloso y más natural a las líneas simplificadas, o la agregación de áreas en otras de más alto nivel (para el caso de las edificaciones). Si quisiéramos ir más allá podríamos contemplar la posibilidad de generalizar varias capas simultáneamente, con el objetivo de conservar la consistencia entre las mismas, una vez simplificadas (por ejemplo, en el caso de simplificación de carreteras los objetos cercanos queden del mismo lado de la carretera en el que se encontraban originalmente). Nuestros esfuerzos en el futuro se encaminarán en investigar y desarrollar nuevos métodos que cubran las deficiencias que encontramos en nuestro sistema y que proporcionen soluciones de generalización más específicas y orientadas a las entidades presentes en nuestras bases de datos.

Agradecimientos

Para el desarrollo de este trabajo de investigación hemos empleado las bases de datos geográficas de la *Encuesta de Infraestructura y Equipamientos Locales* (EIEL), elaborada según metodología común por las diputaciones provinciales, con la colaboración técnica del Ministerio de Administraciones Públicas (MAP). En concreto, estos datos pertenecen a la Diputación Provincial de A Coruña y pueden ser consultados libremente, siempre y cuando se respeten sus condiciones de uso [10].

La migración a software libre de la EIEL de la Diputación Provincial de A Coruña se llevó a cabo mediante un convenio de colaboración entre la Diputación de A Coruña y el Laboratorio de Bases de Datos de la Universidad de A Coruña dirigido por Nieves R. Brisaboa. El trabajo fue dirigido por un comité de seguimiento integrado por Xosé Francisco Freire Corzo, Pedro A. González Pérez, Miguel Lorenzo Turbón y Miguel R. Luaces.

Referencias

- [1] R.B. McMaster and K.S. Shea. "Generalization in Digital Cartography". Association of American Cartographers, Washington DC. 1992
- [2] Laurini R., Thompson D. "Fundamentals of Spatial Information Systems". Academic Press 141-145, 1992.
- [3] D. Douglas, T.K. Peucker. "Algorithms for the reduction of the number of points required to represent a digitalized line or its caricature". Canadian Cartographer 10(2), 112-122. 1973.
- [4] M.R. Johnston, C. D. Scout, R.G. Gibb "Problems Arising From a Simple GIS Generalisation Algorithm", SIRC 99, University of Otago, Dunedin, New Zealand, 1999.
- [5] OpenGeospatial Consortium: <http://www.opengeospatial.org/>

- [6] Web Feature Service (WFS) specifications: <http://www.opengeospatial.org/standards/wfs>
- [7] U. Ramer. "An iterative procedure for the polygonal approximation of plane curves". *Computer Graphics and Image Processing* 1, 244-256. 1972
- [8] Java Unified Mapping Platform: <http://www.vividsolutions.com/JUMP/>
- [9] GIS de la Generalitat Valenciana: <http://www.gvsig.gva.es/>
- [10] Web de la EIEL (Encuesta de Infraestructura y Equipamientos Locales): <http://www.dicoruna.es/webeiel/>
- [11] Longley P. A., Goodchild M. F., Maguire D. J., and Rhind D. W. "Geographical Information Systems. Vol. 1, Principles and Technical Issues". John Wiley & Sons, 1999
- [12] Longley P. A., Goodchild M. F., Maguire D. J., Rhind D. W. "Geographic Information Systems and Science". John Wiley & Sons, 2005.