

CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

Mapa virtual de la República Mexicana

Tesis que presenta

Héctor Javier Acosta Corro

para obtener el Grado de

Maestro en Ciencias

en la Especialidad de

Ingeniería Eléctrica

Director de la Tesis

Dr. Luis Gerardo de la Fraga

México, D.F.

marzo 2007

Resumen

La geografía como toda ciencia emplea las herramientas que tiene a su alcance. Una herramienta muy importante es la computación y una de las aplicaciones de ésta es el usar multi-etiquetas para convertir las representaciones en papel (un mapa) a representaciones digitales como lo son los GIS, buscando siempre una mejor representación de la información de la porción de tierra a la cual se alude. También es posible guardar la información, empleando distintos formatos, además existen distintas formas de representación como imágenes, mapas de contorno o mallas de polígonos.

En esta tesis se trabajó con una cantidad considerable de datos (algunos millones de vértices), los cuales representan curvas de nivel, dichos vértices son el punto de partida para la construcción de los volúmenes a visualizar. De esta manera el trabajo se divide en dos etapas: (1) un método que nos permite una reconstrucción tridimensional adecuada de la porción de la tierra representada por las curvas de nivel y (2) manejar de forma adecuada los datos, para sólo visualizar los datos contenidos dentro de un área de visualización evitando el dibujo de aquellas secciones del modelo que no se encuentran contenidas dentro de dicha área de visión.

Para la primera etapa se muestrean las curvas de nivel del mapa, para ello se calculan las curvas splines de los datos. Posteriormente las construcciones geométricas de los diagramas de Voronoi y Delaunay para generar el esqueleto: se eligen aquellas secciones del esqueleto que eliminan los triángulos planos, con esto se obtiene un mejor modelo tridimensional. Además se clasifican las secciones del esqueleto, para el adecuado cálculo de las alturas de los vértices del esqueleto. La segunda etapa consiste en un método para ordenar los datos en regiones, también se establece un mecanismo para la selección de las regiones que se encuentren dentro del área de visión.

La construcción se realiza partiendo de datos vectoriales, en este caso curvas de nivel donde cada curva de nivel es una polilínea. Finalmente se realizó una interfaz gráfica para la visualización de la construcción tridimensional, empleando las bibliotecas de Qt y OpenGL.

Abstract

Geography, as all sciences, uses tools that are close to it. A very important tool is computation and one of its applications is the use of multi-labels for transforming map illustrations to digital representations as a GIS (Geographic Information System). This kind of systems is made in order to get a better representation of the mention Earth information. It is also possible to store information using a lot of formats, also there are several representations forms such as images, contour maps or poligonal meshes.

This thesis is based on huge data (millions of vertices). The vertices represent level curves and they are the starting step in the volume constructions. This work is divided in two parts: (1) A method that allows a good 3D reconstruction of a portion of the Earth (represented by level curves) and (2) the realization of a better data handle in order to only visualize contents data inside of a visualization area, without drawing those model's sections of does are not contents inside of the visualization.

First part started with re-sampling map level curves, splines are calculated for that. Geometrical constructions of Voronoi and Delaunay diagrams are calculated to generate the skeleton; skeleton's sections were choosen in order to eliminate plain triangules. Also skeleton's sections are classified, for a better calculation of skeleton vertex high. Second part is a method to organize data in sections, establishing mechanisms to select sections that are inside of a vision area.

The construction starts from vector data; the level curves are in the form of polylines. Finally an graphic interface to visualize the three-dimensional constructions was made using Qt and OpenGL libraries.

Agradecimientos

Al **CONACyT**, por haber proporcionado los recursos económicos para la realización de este trabajo.

Al **CINVESTAV** por haber prestado los recursos materiales útiles en el desarrollo de esta tesis.

Al **proyecto 45306 del CONACyT** por el apoyo económico.

Al **Dr. Luis Gerardo de la Fraga** por la asesoría prestada en la realización de esta tesis, pero sobre todo su infinita paciencia.

Al **Dr. Francisco Rodríguez Henríquez**, cuya clase de códigos y criptografía fue la que más me agrado durante mi estancia en el CINVESTAV, no importando las sesiones en más de una ocasión de más de tres horas.

A **Sofía Reza** por la ayuda incondicional que me brindo durante mi estancia en el CINVESTAV.

A mi hermano **Ulises**, quién más allá de ser un dolor de muelas, ha hecho mi vida en muchos momentos más interesante.

A mis padres **Vicelina y Héctor** y mi tía **Maria** quienes me han apoyado siempre, o casi siempre, eso ya depende de cuán descabellada sea la empresa que desee realizar, pero por lo general me apoyan.

A mis compañeros de carrera y amigos **Moc, Mike, Lao y Benjas**, quienes por muchos años me han soportado y como van las cosas me soportarán por varios más.

A mis compañeros **Zombie Lopez, Chablé, Zacapala, Julio, los Oscares, Mario, Mireya, Noel, Vite, Patch, Jorgito, Gil, Edna**, quienes compartieron conmigo esta aventura que fue la maestría.

Agradezco a los revisores de mi tesis el **Dr. Guillermo Benito Morales Luna** (CINVESTAV-IPN) y al **Dr. Francisco Rodríguez Henríquez** (CINVESTAV-IPN),

VIII

por sus observaciones.

Índice general

Abstract	IX
Índice de figuras	XIII
Índice de tablas	XIII
1. Introducción	1
1.1. Antecedentes	1
1.2. Descripción del problema	2
1.3. Tipos de datos	3
1.3.1. Características de los datos vectoriales	5
1.3.2. Modelo de rejilla (<i>raster</i>) vs modelo de malla de triángulos	5
1.4. Triángulos planos	6
1.5. Esqueleto	9
1.6. Organización de la tesis	10
2. Eliminación de los <i>triángulos planos</i>	11
2.1. Método	11
2.2. Carga de datos a memoria	12
2.3. Splines	13
2.4. Muestreo de curvas de nivel	16
2.4.1. Método de muestreo por fuerza bruta	16
2.4.2. Método de Harada y Nakamae	17
2.5. Eliminando <i>triángulos planos</i>	21
2.5.1. Condición del muestreo para calcular el esqueleto	23
2.5.2. Cálculo del <i>esqueleto</i>	26
2.5.3. Seleccionando las secciones del esqueleto que evitan <i>triángulos planos</i>	29
2.5.4. Cálculo de alturas a los vértices del <i>esqueleto</i>	31
3. Agrupamiento de datos	39
3.1. Planteamiento del problema	39
3.2. Utilizando una rejilla	40
3.2.1. Dimensiones de la rejilla	41
3.2.2. Estructuras de datos	41

3.2.3.	Mapeo del plano	43
3.2.4.	Orden y vecindad	45
3.2.5.	Características	48
3.3.	Utilizando árboles de particionamiento espacial binario	48
3.3.1.	Creación del árbol	49
3.3.2.	Llenado del árbol	50
3.3.3.	Características	50
3.4.	Triangulación	50
4.	Resultados	55
4.1.	Diseño de la interfaz	55
4.1.1.	Vista de la interfaz	55
4.2.	Área de visualización	55
4.3.	Área de controles	56
4.3.1.	Restricción de las translaciones	56
4.3.2.	Restricciones de la rotación sobre el eje x	58
4.3.3.	Restricciones de la rotación sobre el eje z'	58
4.3.4.	Descripción del escalamiento	59
4.4.	Área de referencia	61
4.5.	Asignación de colores a vértices	64
5.	Conclusiones y trabajo a futuro	67
5.1.	Conclusiones	67
5.2.	Trabajo futuro	68
A.	Descripción de programas	71
A.1.	Programa de muestreo	71
A.2.	Cálculo de las aristas del esqueleto	71
A.3.	Cálculo de alturas	72
A.4.	Muestreo del esqueleto	72
A.5.	Visualización del modelo	72
B.	Experimento	77

Índice de figuras

1.1.	Se muestra la arquitectura de la aplicación.	4
1.2.	La figura muestra los puntos A,B,C todos sobre una misma circunferencia, estos puntos cumplen con la condición de <i>Delaunay</i>	7
1.3.	La figura muestra los puntos A,B,C,D todos sobre una misma circunferencia, estos puntos cumplen con la condición de <i>Delaunay</i>	7
1.4.	La triangulación de puntos sobre curvas de nivel donde se observa la presencia de triángulos planos.	8
1.5.	Se muestra el <i>esqueleto</i> correspondiente a la superficie.	9
2.1.	La estructura en que son cargadas las curvas de nivel.	12
2.2.	La línea es la aproximación al círculo, mientras que + representan al conjunto muestra.	18
2.3.	La línea es la aproximación del círculo, mientras que + son el conjunto muestra.	21
2.4.	La línea es un conjunto de funciones <i>splines</i> , mientras que las + son el conjunto muestra.	22
2.5.	Triangulación empleando solo puntos muestreados de las curvas de nivel . .	23
2.6.	(a) Curvas que definen el dibujo de un ganzo (b) Conjunto de puntos muestra de las curvas (c) Reconstrucción poligonal de las curvas.	24
2.7.	La mínima distancia entre el conjunto de curvas de nivel, TCL y el eje medio. Las curvas deben muestrearse a 0.42 por la mitad de la distancia mínima entre curvas de nivel.	25
2.8.	a) El caso en que la arista de <i>Voronoi</i> pertenece al esqueleto b) El caso en que la arista de <i>Delaunay</i> pertenece al <i>esqueleto</i>	28
2.9.	Los cuatro casos en que la arista de Voronoi pertenece al esqueleto	28
2.10.	Se muestra con * las <i>curvas de nivel</i> , mientras que con × el <i>esqueleto</i>	30
2.11.	Las <i>curvas de nivel</i> son respresentadas con *, mientras que con × se representan las secciones del <i>esqueleto</i> que evitan a los triángulos planos.	31
2.12.	Los puntos del esqueleto que se utilizan para eliminar los <i>triángulos planos</i> forman grafos conectados, estos grafos pueden clasificarse en cuatro tipos. El tipo 1 es dentro de una cima, el tipo 2 es entre dos curvas de nivel, el tipo 3 es reentrante fuera de una curva, el tipo 4 es reentrante dentro de una curva de nivel.	33

2.13. Se muestran una sección del *esqueleto* que se encuentra en una cima. 35

2.14. La figura muestra el caso en el cual una sección del *esqueleto* se encuentra entre más de una *curva de nivel* con la misma altura. 36

2.15. La figura muestra tres casos de secciones reentrantes etiquetados como A, B y C. 37

3.1. La figura muestra un plano dividido en 4 regiones, en azul curvas de nivel, en rojo los puntos de intesección de las curvas de nivel con las fronteras de las regiones. 40

3.2. El plano que se muestra en la figura tiene sobrepuesto una rejilla de dimensiones 2×2 , los datos están divididos en 4 regiones. 40

3.3. La figura muestra el plano x, y con dimensiones $c \times d$, mientras que los elementos de las rejilla tienen dimensiones $a \times b$ 42

3.4. Cada una de las regiones es una lista de estructuras de tipo *Regiones* en este caso se muestra la región R_i , donde cada elemento de esta lista es la raíz de una lista de la estructura del tipo *Segmentos*. 43

3.5. Cada uno de los elementos de la rejilla denotadas por R_k ha sido mapaeado a un elemento del arreglo $R_{i,j}$ 45

3.6. La figura muestra las coordenadas que delimitan a la región. 46

3.7. La figura muestra que regiones son mayores o menores que la región R . . . 46

3.8. La figura muestra arriba una rejilla con 16 elementos, mientras que abajo muestra el orden de dichos elementos. 47

3.9. La figura muestra los vecinos de R , los Vx son los vecinos sobre el eje x , mientras que los Vy son los vecinos sobre el eje y 47

3.10. La figura muestra el como se realiza el particionamiento usando un árbol **PEB**. 49

3.11. El plano se ha dividido en las regiones A, B, C y D , mientras que e, f, g y h son puntos que se han agregado a las fronteras entre las regiones. 52

3.12. Triangulación sin puntos sobre la frontera 52

3.13. Triangulación con puntos sobre la frontera 53

4.1. La figura muestra la visualización. 56

4.2. La figura muestra el área de control de la interfaz gráfica. 57

4.3. La figura muestra el área de visión, después de realizar algunas translaciones muestra que el área de visión ha llegado a uno de los extremos. 57

4.4. La figura muestra cuatro casos: la vista superior, una rotación de 45° sobre el eje x , una rotación de 90° sobre el eje x y una rotación de 135° sobre el eje x la cual no es permitida. 58

4.5. En a) el plano de visión se encuentra paralelo al plano $x, y, z = z'$ en b) el plano ha sido rotado sobre el eje x , dando como resultado que z' cambie en c) el plano de visión se encuentra paralelo al plano x, z y $z' = y$ 59

4.6. La figura muestra, el área de visión contenida en el área de datos. 60

4.7.	La figura muestra, el área de visión que esta vez no se encuentra contenida en el área de datos.	60
4.8.	La figura muestra, el área de visión que tras algunas translaciones queda contenida dentro del área de datos.	61
4.9.	La figura muestra, el área de referencia en ella se encuentran elementos que representan la latitud y la longitud, la escala a la cual se encuentran los datos.	61
4.10.	La figura muestra las combinaciones de los colores partiendo de los colores primarios, el color negro es la ausencia de color, es decir todos los canales tienen valor cero.	64
4.11.	La altura se ha dividido en niveles, a cada nivel se le asigna un color base.	65
5.1.	La figura muestra un área de datos, un área de <i>buffer</i> y un área de visión, estando sólo en memoria las últimas dos.	70
B.1.	(a) los datos iniciales (b) los <i>splines</i> que corresponden al conjunto de datos iniciales.	77
B.2.	El conjunto de puntos muestreados a una razón de 0.42.	78
B.3.	(a) Esqueleto de los datos remuestreados (b) Secciones útiles para eliminar <i>triángulos planos</i>	78
B.4.	Secciones seleccionadas remuestreadas, y <i>curvas de nivel</i>	79
B.5.	Triangulación de los puntos de las curvas de nivel y los puntos del esqueleto.	80
B.6.	La reconstrucción correspondiente al ejemplo de este apéndice	80

Índice de tablas

2.1.	Número de coeficientes para cada término de la Ec. 2.30.	19
2.2.	Se muestra el formato del archivo de entrada para el <i>script qvoronoi</i>	26
2.3.	Se muestra el formato del archivo de salida del <i>script qvoronoi</i>	27
2.4.	Se muestra los casos en que una arista de <i>Voronoi</i> pertenece al <i>esqueleto</i>	29
2.5.	Se muestra el formato del archivo de entrada requerido por el <i>script</i> que calcula el <i>esqueleto</i>	33
A.1.	Formato del archivo que representa al esqueleto.	73
A.2.	Formato del archivo que representa al diagrama de <i>Delaunay</i>	73
A.3.	Formato de entrada para los datos el programa de muestreo	74
A.4.	Formato del archivo de listas duales s_i y a_i son los vértices que forman la i -ésima arista de la corteza, mientras que p_i y q_i son los vértices diagrama de Voronoi.	75
A.5.	El formato de salida del programa s_i y a_i son los vértices que forman la i -ésima arista.	75

Capítulo 1

Introducción

1.1. Antecedentes

Un mapa es una representación gráfica y métrica de una porción de territorio sobre una superficie bidimensional, generalmente plana, pero que también puede ser esférica como ocurre en los globos terráqueos. El que el mapa tenga propiedades métricas significa que ha de ser posible tomar medidas de distancias, ángulos o superficies sobre él y obtener un resultado aproximadamente exacto [22].

El uso de las técnicas basadas en la fotografía por satélite ha hecho posible no sólo conocer el contorno exacto de un país, de un continente o del mundo, sino también aspectos Etnológicos, históricos, estadísticos, hidrográficos, orográficos, geológicos y económicos que llevan al hombre a un conocimiento más amplio de su medio, del planeta en el que vive [22].

La cartografía cubre la necesidad del hombre de poder hacer referencia a algún lugar mediante el uso de un conjunto de símbolos previamente establecidos. Este conjunto de símbolos ha ido evolucionando para satisfacer las necesidades del hombre, hasta convertirse en un estándar.

Lo esencial en la elaboración de un mapa es la expresión gráfica, la cual debe ser clara sin sacrificar por ello la precisión. El mapa es un documento que tiene que ser entendido según los propósitos que intervinieron en su preparación. Todo mapa tiene un orden jerárquico de símbolos; la expresión gráfica debe permitir visualizar esta jerarquía [22].

Para satisfacer estas necesidades, la información es dividida en capas o planos de lectura, cada una de estas capas representa la misma porción de la tierra, pero resaltando distintos objetos de interés, como pueden ser, vías de comunicación, hidrografía, curvas de nivel, etc. En todo momento se debe tener presente las técnicas de color, también debe relacionarse la cantidad de información a escala y detalle. A un mayor área de representación debe ser mayor la cantidad de elementos informativos del lugar.

Como resultado del incremento en los medios de difusión de información y de la alta disponibilidad de datos en la Internet, los centros de investigación e instituciones privadas cuentan con personal (investigadores), grupos expertos en múltiples áreas. En la actualidad estos grupos de expertos han desarrollado herramientas para la representación y/o análisis de datos; las aplicaciones de estas herramientas se pueden ver en diferentes áreas, por ejemplo:

1. En medicina, se usan técnicas para visualizar el cerebro humano en 3D a partir de una resonancia magnética, permitiendo que los especialistas puedan observar con mayor facilidad el estado en que se encuentra el cerebro [23].
2. En la arquitectura, existen herramientas para realizar simulaciones de estructuras, con el objetivo de probar la resistencia de las estructuras a las oscilaciones [12].
3. En sistemas de seguridad el reconocimiento de rostros no solamente se hace sobre una imagen plana, sino que últimamente se tiende a generar un modelo 3D del rostro, con el objetivo de trabajar sobre el modelo 3D ya que éste presenta más características útiles para diferenciar un rostro de otro [6].
4. En la geografía se tiende a representar los mapas con volumen al representarse de esta manera presentan mayores facilidades para el análisis del terreno. Además este tipo de representación brinda más información visual acerca del terreno que las representaciones tradicionales.

El camino ha sido largo hasta llegar a esta última tendencia, las representaciones han sido desde los trazos en arena hasta los modelos tridimensionales, pasando por representaciones planas, como lo son las cartas topográficas o los globos terráqueos.

1.2. Descripción del problema

La tesis se enfocó en resolver dos problemas:

1. La construcción de un modelo tridimensional que no contenga *triángulos planos*. Para eliminar estos triángulos se introdujeron algunos puntos que pertenecen a la construcción geométrica llamada *esqueleto*. Se hace énfasis en que el modelo se construye partiendo de datos vectoriales.

2. En la visualización se tiene el problema de manejar una cantidad considerable de datos (millones de puntos). El problema consiste en seleccionar aquellos datos se encuentran dentro del área de visión, para de esta manera sólo dibujar estos datos.

Los modelos tridimensionales que contienen *triángulos planos* no representan adecuadamente al terreno al que hacen referencia, en las secciones en las que se presentan *triángulos planos* el modelo no sigue las tendencias de las curvas de nivel.

El problema de los *triángulos planos* (descrito en la sección 1.4) se solucionó al introducir más puntos antes de construir la malla de polígonos los cuales deben pertenecer al eje medio, llamado también *esqueleto*; no todos los puntos que pertenecen al *esqueleto* son necesarios para eliminar los *triángulos planos*. Se buscó un método para seleccionar sólo aquellos puntos del esqueleto que son útiles para eliminar los *triángulos planos*.

En cuanto a la visualización resulta más costoso dibujar todo el modelo, que seleccionar y dibujar aquellas secciones que se encuentran en el área visión.

Para resolver el problema de visualización, que no es otro que la selección de aquellos polígonos que se encuentran en el área de visión existen diversas alternativas, algunas de éstas son analizadas en el capítulo 3.

En el presente trabajo de tesis se siguió la metodología siguiente:

1. Seccionar los datos. Los datos se dividen en regiones, cada región contiene un conjunto de puntos que cumple ciertas características espaciales. Este seccionamiento es descrito en el capítulo 3.
2. A partir de los datos seccionados, se construyó en cada una de las regiones la malla de polígonos correspondiente al conjunto de puntos de cada región. Las mallas de polígonos en conjunto son el modelo 3D.
3. Posteriormente, se implementó un mecanismo que permite seleccionar las regiones que se encuentran dentro del área de visión.
4. Finalmente se realizó la interfaz de usuario.

La figura 1.1 muestra la arquitectura de la aplicación, tal como se aplica realmente para la visualización 3D de mapas vectoriales.

1.3. Tipos de datos

Actualmente existen distintos tipos de datos de los cuales se puede partir para crear un modelo tridimensional. Algunos de estos son las imágenes de curvas de nivel, los datos

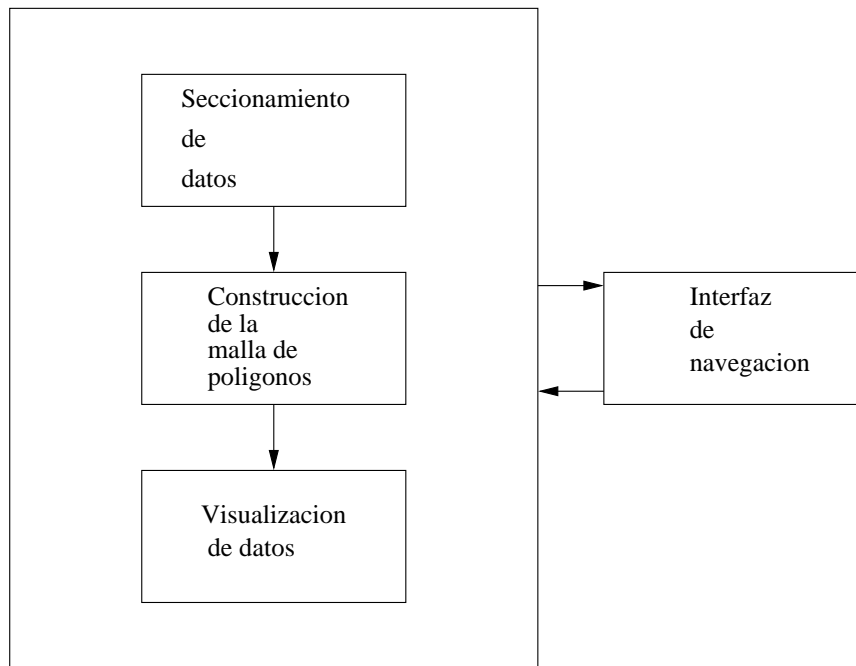


Figura 1.1: Se muestra la arquitectura de la aplicación.

raster o rejilla, datos vectoriales, etc.

Los datos con que se trabajó en esta tesis fueron comprados al INEGI [15], dicho Instituto maneja tres tipos de datos de los cuales se eligió trabajar con el tipo de datos llamado *conjunto de datos vectoriales*. Del INEGI se puede obtener los siguientes tipos de datos:

1. Conjuntos de datos vectoriales, corresponden a la presentación digital de los mapas que tradicionalmente ha elaborado el INEGI, consignan los rasgos u objetos geográficos mediante una representación de puntos, líneas y polilíneas que conforman áreas. Estos datos se encuentran separados por temas en diferentes capas de información tales como vías de comunicación, localidades, hidrografía, curvas de nivel, etc. [15].
2. Conjuntos de datos *raster*, corresponden a datos en formato teselar (*raster*) en los que se incluyen las ortofotos y los modelos de elevación del terreno. La estructura de estos archivos está conformada por un arreglo matricial de valores de un atributo particular, en el caso de las imágenes, los valores pueden ser el de la reflectancia del terreno para cada elemento de imagen o bien, valores de altura del terreno cuando se trata de los modelos de elevación del terreno [15].
3. Conjuntos de datos alfanuméricos, corresponden a archivos de tipo texto con diferentes atributos considerados de interés, relativos a los diferentes rasgos existentes en los conjuntos de datos vectoriales. En esta clase se incluyen archivos de nombres

geográficos (topónimos y localidades) puntos geodésicos, puntos de muestreo para mapas de recursos naturales, descripciones de unidades temáticas, etc. [15].

4. Mapa de contornos, los cuales pueden ser adquiridos del sitio del INEGI.

1.3.1. Características de los datos vectoriales

A continuación se describen algunas características que se deben tomar en cuenta para realizar una visualización adecuada.

1. Los valores de z son de altitud o de alturas ortométricas, en unidades enteras de metros y están referidos al nivel medio del mar, con base en el Datum Vertical para Norteamérica de 1929 [22] (NAVD29) [15].
2. Los puntos están referenciados horizontalmente al sistema de coordenadas de UTM (*Universal Transverse de Mercator*). El Sistema de referencia geodésico es NAD27 o ITRF92 época 1988.0 [15].

El UTM es un sistema de coordenadas geográficas alternativo al empleo de latitud y longitud. Una de sus ventajas es que sus magnitudes se expresan en metros; a diferencia de otros sistemas en que las medidas son angulares pudiendo variar la dimensión lineal [22]. La referencia o estación base del NAD27 se encuentra ubicada en Meades Ranch Kansas, en Estados Unidos.

3. El cubrimiento de cada carta corresponde al formato regular de 15' de latitud por 20' de longitud de la cartografía elaborada en la escala 1:50000 (que corresponden a una sección de 28×35 Km.) mientras que en la escala 1:250000 corresponde al formato de 1° de latitud por 2° de longitud [15] (que corresponden a una sección de 112×280 Km.).

1.3.2. Modelo de rejilla (*raster*) vs modelo de malla de triángulos

Un conjunto de datos vectoriales consiste de un conjunto de polilíneas tridimensionales que mediante el método descrito en el capítulo 2 es posible obtener una malla de polígonos, la cual es el modelo tridimensional a visualizar.

Los datos de tipo rejilla (*raster*) representan un modelo tridimensional (cada uno de los vértices de la rejilla es tridimensional), por lo que el cómputo de este tipo de datos se enfoca únicamente a su visualización.

A continuación se muestran algunas ventajas y desventajas que tiene un modelo sobre el otro.

1. El modelo de rejilla es más eficiente en cuanto a cálculos que el modelo de malla de triángulos. La rejilla al construirse de manera uniforme (los vértices de ésta se encuentran a cierta distancia) permite que las diferentes transformaciones (rotación, escalamiento, etc.) puedan ser realizadas eficientemente mediante la implementación de algún algoritmo incremental, mientras que en la malla de triángulos al no tener una construcción uniforme complica dichos cálculos.
2. El modelo de rejilla es más eficiente en memoria que el modelo de mallas de triángulos. Por lo regular siempre se necesitan más vértices en los modelos de malla de triángulos que en los modelos de rejilla.
3. El modelo de malla de triángulos es más exacto que el modelo de rejilla, el modelo de malla de triángulos permite un mayor grado de detalle, debido a que se construye directamente sobre los datos vectoriales (mapas de contornos) cada triángulo refiere a una zona con una variación de altura permitiendo detallar valles, cuevas, cañones, etc. mientras que el modelo de rejilla carece de esta cualidad.

Aun cuando computacionalmente es más eficiente el modelo de rejilla, la calidad de la visualización es la que nos interesa, por lo que se optó por trabajar con el modelo de malla triángulos para contar con la cualidad de detalle.

También debemos estar concientes que uno de los objetivos de este trabajo de tesis es el seccionar los datos de tal manera que sólo se apliquen las distintas transformaciones a los datos a visualizar, evitando de esta manera aplicar las distintas transformaciones al conjunto total de datos.

Otro punto que debe considerarse es que si bien es cierto que se debe construir la malla de triángulos a partir del conjunto de datos vectoriales para posteriormente realizar la visualización, esta construcción es considerada como precomputo.

1.4. Triángulos planos

El modelo a visualizar es el resultado de triangular los puntos que representan los datos vectoriales sobre el plano (x, y) , utilizando para dicha triangulación el diagrama de *Delaunay*.

Sea P un conjunto de puntos con $P \subset R^2$, sea τ una colección de triángulos. τ es una triangulación de *Delaunay* para P si ningún punto de P es interior al círculo que circunscribe a cualquier triángulo [22].

La circunferencia circunscrita de un triángulo, es la circunferencia que contiene a los tres vértices del triángulo.

Decimos que circunferencia circunscrita de un triángulo se encuentra vacía cuando la circunferencia no contiene ningún otro punto del conjunto de datos.

La Fig. 1.2 muestra al círculo que circunscribe al triángulo ABC , este círculo es el centrado en el punto en que se cortan las mediatrices de los lados del triángulo. En algunas ocasiones algunos puntos presentan ambigüedades, formando en este caso polígonos con cuatro o más aristas, dependiendo de cuantos puntos se encuentren sobre la circunferencia, cumpliendo de esa forma la condición de *Delaunay*.

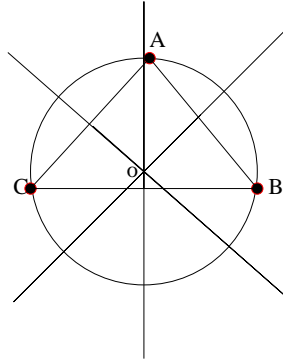


Figura 1.2: La figura muestra los puntos A,B,C todos sobre una misma circunferencia, estos puntos cumplen con la condición de *Delaunay*.

En la Fig. 1.3 existen los triángulos ABC , CDA , DBC y ABD inscritos sobre la misma circunferencia, este caso es un ejemplo de una ambigüedad ya que hay dos posibles triangulaciones de *Delaunay* para los puntos $ABCD$.

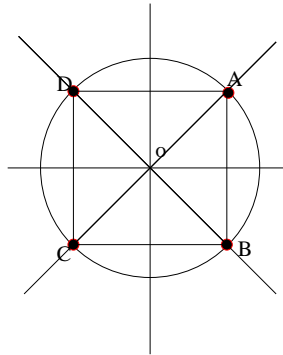


Figura 1.3: La figura muestra los puntos A,B,C,D todos sobre una misma circunferencia, estos puntos cumplen con la condición de *Delaunay*.

Los datos vectoriales constan de un conjunto de puntos P con $P \subset R^3$, la triangulación en este trabajo de tesis se realiza sobre el plano (x, y) , de esta manera los polígonos

cubren el plano (x, y) . Llamamos *triángulos planos* a aquellos polígonos cuyos vértices tengan el mismo valor en la ordenada z .

Los modelos que presentan *triángulos planos* ocasionan visualizaciones pobres, con ésto nos referimos a que los modelos tridimensionales que contienen *triángulos planos* no representan adecuadamente la porción de la Tierra a la que hacen referencia, ya los *triángulos planos* no siguen la tendencia del terreno a representar.

La figura 1.4 podemos observar una vista de un modelo, en el cual se presentan *triángulos planos*, en este trabajo se clasificó a los *triángulos planos* en los siguientes tipos:

1. Cimas, son *curvas de nivel* que no contienen otras *curvas de nivel*.
2. Secciones reentrantes por fuera de la *curva de nivel*.
3. Secciones reentrantes por dentro de la *curva de nivel*.
4. Entre *curvas de nivel* con la misma altura.

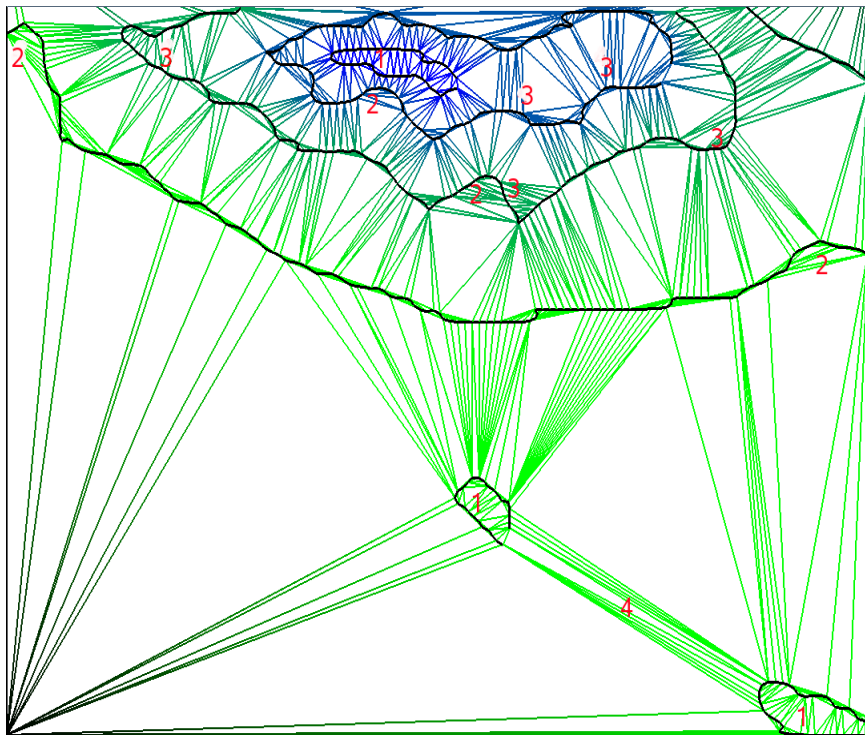


Figura 1.4: La triangulación de puntos sobre curvas de nivel donde se observa la presencia de triángulos planos.

1.5. Esqueleto

Esqueleto. El *esqueleto* $ES(X)$ de un objeto $X \in R^2$ son los lugares geométricos ubicados en los centros de los círculos máximos incluidos dentro del objeto X [3].

En la figura 1.5 se muestra una superficie a la cual se le ha calculado su *esqueleto*.

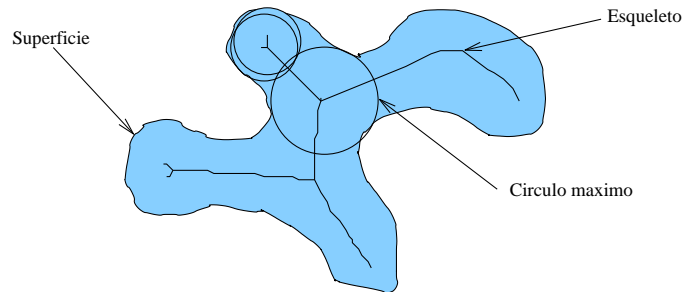


Figura 1.5: Se muestra el *esqueleto* correspondiente a la superficie.

Existe más de una forma de calcular el *esqueleto*, éstas dependen de la fuente de datos con que se este trabajando, algunos ejemplos de las instancias de este problema son:

1. Trabajando con imágenes, el eje medio puede obtenerse mediante el empleo de un mapa de distancias, en el trabajo de Abigail Martínez [17] se trata este problema, otro trabajo que aborda específicamente este problema es el de Alexandru Telea, Cristian Sminchisescu y Sven Dickinson [21] en el que se propone un método para obtener este el *eje medio*.
2. Cuando la fuente de datos son puntos contenidos en el plano, el esqueleto es calculado apoyándose en la relación que presentan las construcciones geométricas del diagrama de *Voronoi* y el diagrama de *Delaunay* que mediante la prueba de *dentro del círculo* descrita en [9] se puede decidir si la arista del diagrama de *Voronoi* asociada a la arista de *Delaunay* pertenecen o no al esqueleto.

En el trabajo de D. Attalo y A. Montanvert [3] se analizan tres métodos para calcular el esqueleto, estos métodos se basan en el cálculo del diagrama de *Voronoi* y la intersección de éste con la superficie a la cual se le desea calcular el esqueleto, la diferencia entre cada uno de los métodos de este trabajo radica en si se toma las aristas, los vértices o la intersección del diagrama de *Voronoi* con la superficies a las que se les desea calcular el esqueleto.

El esqueleto se usa para obtener una mejor reconstrucción 3D. Para ésto se seleccionarán las partes del esqueletos que eliminan los *triángulos planos*.

1.6. Organización de la tesis

Tomando en cuenta los problemas descritos en la sección 1.2 este trabajo puede ser dividido en dos partes, la descripción del método empleado para realizar la construcción del modelo tridimensional y la descripción del método empleado para el manejo de los datos a visualizar.

En el capítulo dos se describe el método empleado para la construcción del modelo tridimensional, dicho método es una extensión del método del trabajo de Abigail Martínez [18] cambiando la forma en que realiza el muestreo de las curvas de nivel, también se cambia la clasificación de los grafos que pertenecen al esqueleto, además el cálculo de la altura de los puntos del esqueleto toma en cuenta el contexto de dichos grafos.

El capítulo tres se dedica a analizar dos métodos que sirven para el seccionamiento de los datos a trabajar, dichos métodos son los árboles binarios espaciales (*binary space trees BSP*) y un método basado en una rejilla.

El capítulo cuatro describe el cómo se lleva a cabo la visualización del modelo, así como la interacción entre el modelo, el módulo de selección de datos y la interfaz de usuario.

Finalmente el capítulo cinco muestra las conclusiones a las que se llegó con este trabajo, así como las mejoras que se le pueden hacer.

Capítulo 2

Eliminación de los *triángulos planos*

El modelo tridimensional construido con el método descrito en este capítulo, consiste en una malla de polígonos cuyos vértices son puntos tridimensionales.

La malla de polígonos se construye usando la triangulación de Delaunay sobre puntos en el plano (x, y) . Posteriormente se les debe asignar la altura a estos puntos. En este capítulo se explicará ha detalle el proceso de asignación de altura.

2.1. Método

Los pasos para obtener el modelo tridimensional son los siguientes:

1. Obtener las curvas de nivel de los archivos del INEGI [15]. Esto es, acceder a los puntos que corresponden a los vértices de cada polilínea que conforma cada una de las curvas de nivel contenidas en el formato estándar CAD. Dicho de otra manera, la primera parte consiste en leer archivos de tipo DXF.
2. El siguiente paso es calcular la aproximación de cada una de las curvas de nivel mediante la técnica de splines (esta técnica es descrita detalladamente en la sec. 2.3, en la Pág. 13) este proceso da como resultado que las curvas de nivel ya no estén representadas por líneas rectas; ahora se tiene que entre cada par de vértices se encuentran dos polinomios que representan esa parte de la curva.
3. Un vez que se tiene la representación de las curvas de nivel mediante polinomios cúbicos, éstas deben ser re-muestreadas de forma equiespaciada. Las características que deben cumplir los puntos se describirán en la sección 2.4.
4. Para evitar el problema de los triángulos planos se procede a obtener el esqueleto de las curvas de nivel, debido a que la construcción de éste nos sirve para generar más puntos; lo que garantiza que cuando se calcule el diagrama de Delaunay al menos un

punto no pertenezca a una curva de nivel con la misma altura, ésto se explicará más a detalle en la sección 2.5.

5. Existen casos en los que no es necesario agregar puntos del *esqueleto*, debido ha que hay secciones del diagrama de *Delaunay* en las que no se presentan *triángulos planos*.
6. Construir el diagrama de *Delaunay* a partir del conjunto de puntos remuestreados y del *esqueleto*, ya que estos triángulos construyen el modelo.

2.2. Carga de datos a memoria

En esta sección se describe el proceso de cargar a memoria los datos provenientes de los archivos DXF.

Los requerimientos son:

1. Un archivo en formato DXF.

Para carga los datos vectoriales a memoria se utilizan las estructuras de datos `CurvaNivel` y `Punto`. En memoria se tiene una lista de curva de nivel donde cada curva de nivel tiene una lista de puntos (los vértices de la polilínea). La Fig. 2.1 muestra el estado de los datos en memoria.

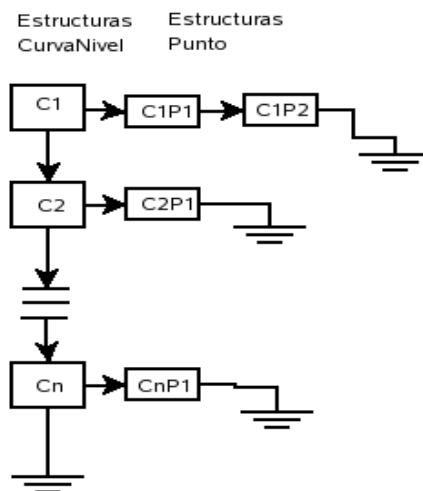


Figura 2.1: La estructura en que son cargadas las curvas de nivel.

La estructura utilizadas son:

```
typedef struct CurvaNivel
{
    unsigned int altura;
    unsigned int xmax,ymax,xmin,ymin;
    struct punto *siguiente;
    struct CurvaNivel*sigNvl;
};
```

La estructura de datos `CurvaNivel` contiene los siguientes elementos:

Los elementos `(xmin,ymin)`, `(xmax,ymax)` que vistos de esta manera son los vértices inferior izquierdo y superior derecho de la caja que contiene a la curva de nivel.

El elemento `altura`, es la altura de la curva de nivel. Al colocar este elemento aquí y no en la estructura `Punto` se elimina la redundancia de la información, esto se debe a que todos los puntos de la curva de nivel tienen la misma altura.

Los elementos `siguiente` y `sigNvl` son apuntadores a los puntos de la curva y a la siguiente curva de nivel, respectivamente.

```
typedef struct Punto
{
    unsigned int x[N],y[N];
    struct punto *siguiente;
};
```

La estructura de datos `Punto` sólo contiene información acerca de los puntos, esta información consiste únicamente en el par de ordenadas organizadas en un par de arreglos (X,Y) de longitud N , de tal forma que el punto i -ésimo es representado por la pareja $X[i]$, $Y [i]$.

2.3. Splines

La técnica *spline* es una de las muchas técnicas de interpolación, ésta consiste en lo siguiente:

A partir de un conjunto de n puntos genera n polinomios si la curva se encuentra cerrada, si no se encuentra cerrada genera $n - 1$ polinomios, este conjunto de polinomios al ser evaluados entre cada par de puntos generan una curva.

Una curva se dice ser cerrada si los vértices forman un ciclo; en otro caso, la curva se encuentra abierta.

Entonces una función *spline* está formada por varios polinomios, cada uno definido sobre un sub-intervalo, estos polinomios se unen entre sí obedeciendo a ciertas condiciones de continuidad. La más utilizada es aquella en donde la primera y segunda derivadas paramétricas son iguales en los puntos de unión y que la segunda derivada es cero en los puntos terminales (a esta formulación se le llama *spline* natural) [14].

Consideremos una función *spline* unidimensional para un conjunto de puntos de cardinalidad $n + 1$, sean (y_0, y_1, \dots, y_n) las ordenadas correspondientes a estos puntos, el polinomio que representa cada uno de los segmentos de la curva entre cada uno de estos puntos.

El *spline* está definido por un polinomio de tercer grado:

$$Y_i(t) = a_i + b_it + c_it^2 + d_it^3 \quad (2.1)$$

Evaluando la Ec. (2.1) en cada intervalo:

$$Y_i(0) = y_i = a_i \quad (2.2)$$

$$Y_i(1) = y_{i+1} = b_i + c_i + d_i \quad (2.3)$$

Calculando la primera derivada a la Ec. (2.1) y evaluando en los extremos del intervalo:

$$Y'_i(t) = b_i + 2c_it + 3d_it^2 \quad (2.4)$$

$$Y'_i(0) = b_i = D_i \quad (2.5)$$

$$Y'_i(1) = b_i + 2c_i + 3d_i = D_{i+1} \quad (2.6)$$

Con las siguientes restricciones :

$$Y_{i-1}(1) = Y_i(0) \quad (2.7)$$

$$Y'_{i-1}(1) = Y'_i(0) \quad (2.8)$$

$$Y_i(0) = y_i \quad (2.9)$$

$$Y''_{i-1}(1) = Y''_i(0) \quad (2.10)$$

La ecuación (2.7) indica que las curvas se unen en los puntos de unión. Las Ecs. (2.8) y (2.9) indican que las primeras derivadas paramétricas se igualan entre cada curva y en los puntos de unión. Finalmente la Ec. (2.10) indica que las segundas derivadas paramétricas se igualan en los puntos de unión.

Por otra parte, tanto el punto inicial como el final deben satisfacer las siguientes condiciones:

$$Y_0(0) = y_0 \quad (2.11)$$

$$Y_{n-1}(1) = y_n \quad (2.12)$$

Se tiene que encontrar los términos a_i , b_i , c_i , d_i . De la Ec. (2.2) tenemos que $a_i = y_i$, mientras que b_i está dada por (2.5). Entonces necesitamos encontrar las expresiones para c_i y d_i . Restando (2.2) a (2.3)

$$\begin{aligned} y_{i+1} - y_i &= b_i + c_i + d_i \\ y_{i+1} - y_i &= D_i + c_i + d_i \end{aligned} \quad (2.13)$$

Restando (2.5) a (2.6)

$$D_{i+1} - D_i = 2c_i + 3d_i \quad (2.14)$$

Las ecuaciones (2.13) y (2.14) forman un sistema de ecuaciones con dos incógnitas. Resolviendo el sistema se tienen las expresiones para c_i y d_i . Multiplicando (2.13) por dos y sumándola a (2.14) resulta:

$$D_{i+1} - D_i - 2(y_{i+1} - y_i) + 2D_i = d_i \quad (2.15)$$

Multiplicando (2.13) por -3 y sumándola a (2.14):

$$\begin{aligned} c_i &= D_{i+1} - D_i - 3(y_{i+1} - y_i) + 3D_i \\ c_i &= 3(y_{i+1} - y_i) - 2D_i - D_{i+1} \end{aligned} \quad (2.16)$$

Las segundas derivadas deben ser iguales en los puntos de unión, como lo indica la Ec. (2.10). La segunda derivada es

$$\begin{aligned} Y_i''(0) &= 2c_i \\ &= 6(y_{i+1} - y_i) - 4D_i - 2D_{i+1} \end{aligned} \quad (2.17)$$

$$\begin{aligned} Y_{i-1}''(1) &= 2c_{i-1} + 6d_{i-1} \\ &= 2[3(y_i - y_{i-1}) - 2D_{i-1} - D_i] + 6[2(y_i - y_{i-1}) + D_{i-1} + D_i] \\ &= 6(y_i - y_{i-1}) - 4D_{i-1} - 2D_i + 12(y_i - y_{i-1}) + 6D_{i-1} + 6D_i \\ &= 6y_{i-1} - 6y_i + 2D_{i-1} + 4D_i \end{aligned} \quad (2.18)$$

igualando (2.17) y (2.18)

$$\begin{aligned} 6(y_{i+1} - y_i) - 4D_i - 2D_{i+1} &= 6y_{i-1} - 6y_i + 2D_{i-1} + 4D_i \\ 6y_{i+1} - 6y_i - 6y_{i-1} + 6y_i &= 2D_{i-1} + 4D_i + 4D_i + 2D_{i+1} \\ 6y_{i+1} - 6y_{i-1} &= 2D_{i-1} + 8D_i + 2D_{i+1} \end{aligned} \quad (2.19)$$

Ésto nos da como resultado $4(n-1) + 2$ ecuaciones, y tenemos $4n$ incógnitas; por lo que son necesarias otras dos restricciones para poder resolver el sistema de ecuaciones, para esto se toman las segundas derivadas en los puntos terminales iguales a cero, como sigue:

$$Y_0''(0) = 0 \quad (2.20)$$

$$Y_{n-1}''(1) = 0 \quad (2.21)$$

La Ec. (2.19) y las expresiones resultantes de aplicar las restricciones de (2.20) y (2.21) pueden escribirse en forma matricial si la curva se encuentra abierta como:

$$\begin{bmatrix} 2 & 1 & 0 & . & . & . & 0 & 0 \\ 1 & 4 & 1 & . & . & . & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & . & . & . \\ . & . & 0 & 1 & 4 & 1 & 0 & . \\ . & . & . & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & . & . & . & 1 & 4 & 0 \\ 0 & 0 & . & . & . & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ . \\ . \\ . \\ D_{n-1} \\ D_n \end{bmatrix} = \begin{bmatrix} 3(y_1 - y_0) \\ 3(y_2 - y_0) \\ 3(y_3 - y_1) \\ . \\ . \\ 3(y_n - y_{n-2}) \\ 3(y_n - y_{n-1}) \end{bmatrix} \quad (2.22)$$

y cuando se encuentra cerrada se escribe como:

$$\begin{bmatrix} 4 & 1 & 0 & . & . & . & 0 & 1 \\ 1 & 4 & 1 & . & . & . & 0 & 0 \\ 0 & 0 & 1 & 4 & 1 & . & . & . \\ . & . & 0 & 1 & 4 & 1 & 0 & . \\ . & . & . & 0 & 1 & 4 & 1 & 0 \\ 0 & 0 & . & . & . & 1 & 4 & 0 \\ 1 & 0 & . & . & . & 0 & 1 & 4 \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ . \\ . \\ . \\ D_{n-1} \\ D_n \end{bmatrix} = \begin{bmatrix} 3(y_1 - y_n) \\ 3(y_2 - y_0) \\ 3(y_3 - y_1) \\ . \\ . \\ 3(y_n - y_{n-2}) \\ 3(y_0 - y_{n-1}) \end{bmatrix} \quad (2.23)$$

2.4. Muestreo de curvas de nivel

El muestreo de las curvas de nivel es una técnica que consiste en tomar un conjunto de puntos pertenecientes a cada una de las curvas de nivel, con el objetivo de que dicho conjunto de puntos sea capaz de representar a cada una de las curvas de nivel, dichos puntos deben cumplir con ciertas características las cuales se mencionarán en la sección 2.5.1.

2.4.1. Método de muestreo por fuerza bruta

Si la longitud sobre una curva está definida por la integral de longitud de arco

$$l = \int_A^B \sqrt{x'(t)^2 + y'(t)^2} dt \quad (2.24)$$

donde $y(t)$ y $x(t)$ son polinomios cuyas características coinciden con la ecuación 2.1

de esta forma $y'(t) = \frac{dy(t)}{dt}$ y $x'(t) = \frac{dx(t)}{dt}$

l es la longitud sobre la curva.

Una solución directa sería sustituir la integral (2.24) por la integral siguiente:

$$m = \int_A^s \sqrt{x'(t)^2 + y'(t)^2} dt \quad (2.25)$$

donde m es el paso de muestreo, s es una variable que al resolver esta integral de manera simbólica es el valor que satisface la igualdad entre la integral (2.25) y el paso de muestreo m . De esta manera nos queda por resolver una ecuación con una incógnita.

Para solucionar la integral (2.25) se decidió emplear un método numérico, incrementando el intervalo de integración hasta dar con el paso de muestreo.

El algoritmo 1 emplea el método de *Newton* para encontrar el conjunto de puntos muestra, en cada iteración del método verifica si la longitud sobre la curva en el punto t actual corresponde a un punto muestra, en caso de serlo se agrega este punto al conjunto muestra en caso de no serlo se desecha.

Algoritmo 1 Muestreo

Entrada: El conjunto de polinomios *splines*, el número en que se ha dividido el intervalo de integración, el paso de muestreo.

Salida: El conjunto muestra.

```

1: float s=0
2: float A,B
3: float intervalo
4: int i=0
5: mientras PasoMuestra > s e i < numParticiones hacer
6:   intervalo = (B - A)/NumParticiones
7:   s+ = Newton(A + (intervalo * i), A + (intervalo * (i + 1)))
8:   i ++
9: fin mientras

```

En la figura 2.2 se muestra la aproximación de un círculo y su respectivo conjunto muestra la razón de muestreo es 0.7 del segmento más corto.

Debido a que es una integración numérica sólo se obtiene una aproximación, la precisión depende del número de particiones en que se ha dividido el intervalo de integración.

La complejidad del algoritmo 1 es lineal con respecto al número de particiones en que se ha dividido el intervalo de integración.

2.4.2. Método de Harada y Nakamae

En el año de 1989 Harada y Nakamae desarrollaron un método para obtener muestras equiespaciadas, basándose en la distancia euclidiana a partir de la relación entre el

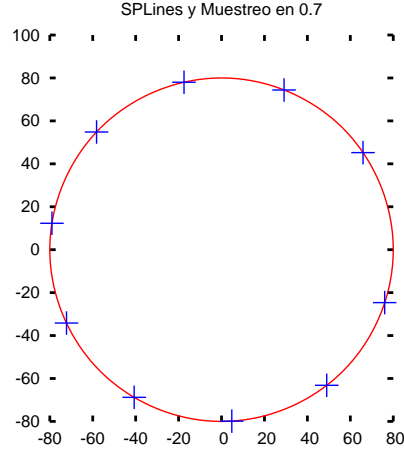


Figura 2.2: La línea es la aproximación al círculo, mientras que + representan al conjunto muestra.

despliegue de la curva y los espacios paramétricos. Para ello es necesario cambiar la representación paramétrica de los *splines* a una implícita [11]. Este método al basarse en la distancia euclidiana es también una aproximación. Una de las aplicaciones del método de Harada y Nakamae es la de muestrear una curva a una velocidad de recorrido constante.

Sean $x_i(t)$ y $y_i(t)$ las ecuaciones paramétricas en el intervalo i definidas como:

$$x_i(t) = a_3^i t^3 + a_2^i t^2 + a_1^i t + a_0^i \quad (2.26)$$

$$y_i(t) = b_3^i t^3 + b_2^i t^2 + b_1^i t + b_0^i \quad (2.27)$$

donde $0 \leq t \leq 1$ e $i = 0, 1, \dots, n-1$, i es el número de segmentos, y a_0^i hasta b_3^i están definidos entre el intervalo P_i y P_{i+1} .

En el trabajo de Senderberg [19] la ecuación paramétrica que representa al spline es re-escrita como:

$$a_3^i t^3 + a_2^i t^2 + a_1^i t + a_0^i - x = 0 \quad (2.28)$$

$$b_3^i t^3 + b_2^i t^2 + b_1^i t + b_0^i - y = 0 \quad (2.29)$$

la implicitización de las ecuaciones 2.28 y 2.29 se obtienen por:

término	variable	número de coeficientes
A	x^3	1
B	y^3	1
C	x^2y	1
D	xy^2	1
E	x^2	7
F	y^2	7
G	xy	8
H	x	21
I	y	21
J	constantes	34

Tabla 2.1: Número de coeficientes para cada término de la Ec. 2.30.

$$\begin{vmatrix} a_3^i & a_2^i & a_1^i & a_0^i - x & 0 & 0 \\ 0 & a_3^i & a_2^i & a_1^i & a_0^i - x & 0 \\ 0 & 0 & a_3^i & a_2^i & a_1^i & a_0^i - x \\ b_3^i & b_2^i & b_1^i & b_0^i - y & 0 & 0 \\ 0 & b_3^i & b_2^i & b_1^i & b_0^i - y & 0 \\ 0 & 0 & b_3^i & b_2^i & b_1^i & b_0^i - y \end{vmatrix} = 0 \quad (2.30)$$

La Ec. 2.30 es una ecuación bicúbica con respecto a x y y . Con la ayuda de un programa matemático simbólico (en este trabajo se usó Mathematica) se puede expandir la ecuación y el valor de los coeficientes, la tabla 2.4.2 muestra el número de coeficientes por variable.

A continuación se listan los coeficientes:

$$A = b_0^3 x^3$$

$$B = -a_0^3 y^3$$

$$C = -3b_0^2 x^2 a_0 y, \quad D = 3b_0^2 a_0^2 x y^2$$

$$E = 3a_0 b_3 b_0^2 x^2 - 3a_3 b_0^3 x^2 + a_0 b_1^3 x^2 - 3a_0 b_2 b_1 b_0 x^2 - a_1 b_1^2 b_0 x^2 + 2a_1 b_2 b_0^2 x^2 + a_2 b_1 b_0^2 x^2$$

$$F = 3a_0^3 b_2 y^2 - 3a_3 a_0 b_0 y^2 - a_1 a_0^2 b_2 y^2 - 2a_2 a_0^2 b_1 y^2 - a_1^3 b_0 y^2 + 3a_2 a_1 a_0 b_0 + a_1^2 a_0 b_1 y^2$$

$$G = -6a_0^2 b_3 b_0 x y + 3a_0^2 b_2 b_1 x y - 2a_1 a_0 b_1^2 x y + 2a_1^2 b_1 b_0 x y - a_2 a_0 b_1 b_0 x y + a_3 a_0 b_1 b_0^2 x y - 3a_2 a_1 b_0 x y - a_1 a_0 b_2 b_0$$

$$H = -3a_0^2 b_3 b_2 b_1 x + 2a_1 a_0 b_3 b_1^2 x + 3a_0^2 b_3^2 b_0 x - 2a_1^2 b_3 b_1 b_0 x - a_2 a_0 b_3 b_1 b_0 x + 3a_2 a_1 b_3 b_0^2 x - 6a_3 a_0 b_3 b_0^2 x - 2a_3 a_0 b_1^3 x + 6a_3 a_0 b_2 b_1 b_0 x + 2a_3 a_1 b_1^2 b_0 x - 4a_3 a_1 b_2 b_0^2 x - 2a_3 a_2 b_1 b_0^2 x + 3a_3^2 b_0^3 x +$$

$$a_0^2 b_2^3 x - a_1 a_0 b_2^2 b_1 x + a_2 a_0 b_2 b_1^2 x + a_1 a_0 b_3 b_2 b_0 x + a_1^2 b_2^2 b_0^2 x - a_2 a_0 b_2^2 b_0 x - a_2 a_1 b_2 b_1 b_0 x + a_2^2 b_2 b_0^2 x$$

$$I = -3a_0^3 b_3^2 y + 3a_1 a_0^2 b_3 b_2 y - 2a_1^2 a_0 b_3 b_1 y + 4a_2 a_0^2 b_3 b_1 y + 2a_1^3 b_3 b_0 y - 6a_2 a_1 a_0 b_3 b_0 y + 6a_3 b_0^2 b_3 b_0 y - a_2 a_0^2 b_1^2 y + a_2 a_1 a_0 b_2 b_1 y - a_2^2 a_0 b_1^2 y - a_2 a_1^2 b_2 b_0 y = 2a_2^2 a_0 b_2 b_0 y + a_2^2 a_1 b_1 b_0 y - a_3 a_2 a_0 b_1 b_0 y - a_2^3 b_0^2 y - 3a_3 a_0^2 b_2 b_1 y + 2a_3 a_1 a_0 b_1^2 y + a_3 a_1 a_0 b_2 b_0 y - 2a_3 c_1^2 b_1 b_0 + 3a_3 a_2 a_1 b_0^2 y - 3a_3^2 a_0 b_0^2 y,$$

el término J no es necesario ya que sólo se emplean los términos que dependen de x o y por esta razón J no se muestra aquí.

Los términos que dependen de las variables x y y son necesarios para evaluar las Ecs. 2.31 y 2.32, estas Ecs. son derivadas de la Ec. implícita 2.30.

$$F_x(p, q) = 3Ap^2 + 2Cpq + Dq^2 + 2Ep + Gq + H \quad (2.31)$$

y

$$F_y(p, q) = 3Bq^2 + Cp^2 + 2Dpq + 2Fq + Gp + I \quad (2.32)$$

estas ecuaciones nos sirven para evaluar las ecuaciones que definen el parámetro para cada uno de los puntos muestra.

$$|p - p'| \sqrt{1 + \{(F_x/F_y)_{x=p, y=p}\}^2} = T_k \quad (2.33)$$

Para obtener una mejor aproximación numérica si el valor absoluto de F_y es menor que el de F_x se utiliza la siguiente ecuación

$$|q - q'| \sqrt{1 + \{(F_y/F_x)_{x=p, y=p}\}^2} = T_k \quad (2.34)$$

es decir se busca T_k en base a las ecuaciones paramétricas $x(t)$ y $y(t)$ según sea conveniente, p y q son las ordenadas del punto actual y p' y q' son las ordenadas del punto siguiente, siendo estas últimas las dos incógnitas.

El término $|p - p'|$ es evaluado asumiendo que el parámetro evaluado para p debe ser t y p' debe ser $t + \Delta t$ entonces se puede reescribir la Ec. 2.33 de la siguiente forma:

$$a_3^i \Delta t^3 + (3a_3^i t_p + a_2^i) \Delta t^2 + (3a_3^i t_p + 2a_2^i + a_1^i) \Delta t - |J_i| = 0 \quad (2.35)$$

donde $J_i = T_k \sqrt{1 + \{(F_x/F_y)_{x=p, y=p}\}^2}$, T_k es el paso de muestreo elegido, de esta manera nos queda por encontrar Δt .

Se realizaron algunos experimentos para probar el algoritmo, uno de los experimentos consistió en aproximar un círculo partiendo de nueve puntos, las ecuaciones paramétricas son $x(t) = \cos(x)$ y $y(t) = \sin(t)$, t se encuentran espaciadas 45° empezando en 0° .

En la figura 2.3 la aproximación de un círculo y en azul se muestra el conjunto muestra también resultado de un muestreo con paso de 0.7.

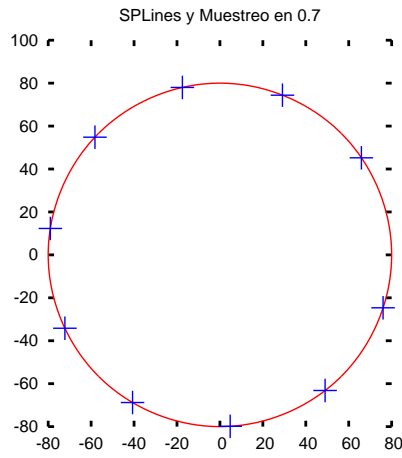


Figura 2.3: La línea es la aproximación del círculo, mientras que + son el conjunto muestra.

Un algoritmo que ha mostrado resultados aceptables para resolver la ecuación 2.35 es el de bisección, tal como lo argumentan Harada y Nakamae en [11].

La metodología que desarrollaron Harada y Nakamae, consiste en resolver el determinante 2.30 para encontrar los términos de la tabla 2.1, éstos son utilizados para evaluar las Ecs. 2.31 y 2.32, que dependiendo de los valores obtenidos en estas ecuaciones se resuelve la Ec. 2.35 o su variante. Al encontrar Δt de la Ec. 2.35 se evalúan las Ecs. 2.26 y 2.27 sustituyendo t por $t + \Delta t$, obteniendo de esta forma p' y q' . El método se repite cambiando p y q por p' y q' recorriendo toda la curva.

La figura 2.4 una aproximación a una curva irregular, las + son los puntos que pertenecen al conjunto muestra. Al observar los puntos muestreados, es posible observar que los puntos de la figura están equi-espaciados, es decir, que para cualquier punto en una curva, la distancia que existe entre el punto actual y el que le sigue es igual a la distancia que existe entre el punto actual y el anterior.

2.5. Eliminando *triángulos planos*

Como se mencionó en 1.4 se debe construir un modelo tridimensional sin la presencia de *triángulos planos* ya que los modelos que presentan *triángulos planos* no se apegan adecuadamente a las tendencias del terreno.

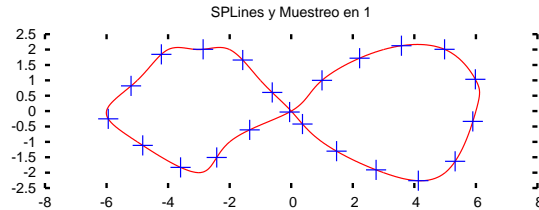


Figura 2.4: La línea es un conjunto de funciones *splines*, mientras que las + son el conjunto muestra.

En la figura 2.5 podemos observar una triangulación, la cual se ha hecho sólo con los puntos muestreados de las curvas de nivel.

La triangulación obtenida mediante el empleo del diagrama de *Delaunay* se realiza sobre el plano (x, y) . Al construir la triangulación empleando los puntos muestreados que representan a las curvas de nivel, se crean *triángulos planos*, existen tres casos en que los triángulos planos pueden aparecer:

1. Cimas, son *curvas de nivel* que no contienen dentro otras *curvas de nivel*, la Fig. 2.5 (denotado con el número uno) se muestra un ejemplo de este caso.
2. Secciones reentrantes, en este caso los *triángulos planos* se encuentran sobre una misma *curva de nivel*, pero a diferencia de las cimas las curvas de nivel de este caso contienen otras curvas de nivel. La Fig. 2.5 el número dos denota a una sección reentrante que se encuentra fuera de la curva, y el número tres muestra a una sección reentrante dentro de la curva de nivel.
3. Secciones entre curvas con la misma altura, aquí los polígonos tienen vértices en dos o más *curvas de nivel*, pero todas estas curvas tienen la misma altura.

La Fig. 2.5 muestra cuatro casos, el número uno hace referencia a las cimas, el número dos hace referencia a secciones reentrantes por fuera de la curva de nivel, el número tres hace referencia a secciones reentrantes por dentro de la curva de nivel, las secciones etiquetadas con el número cuatro son aquellas con una misma altura que están entre curvas de nivel. Es importante poder distinguir cada uno de estos casos, ya que más adelante será necesario clasificar las secciones del esqueleto, tomando en cuenta estos cuatro casos.

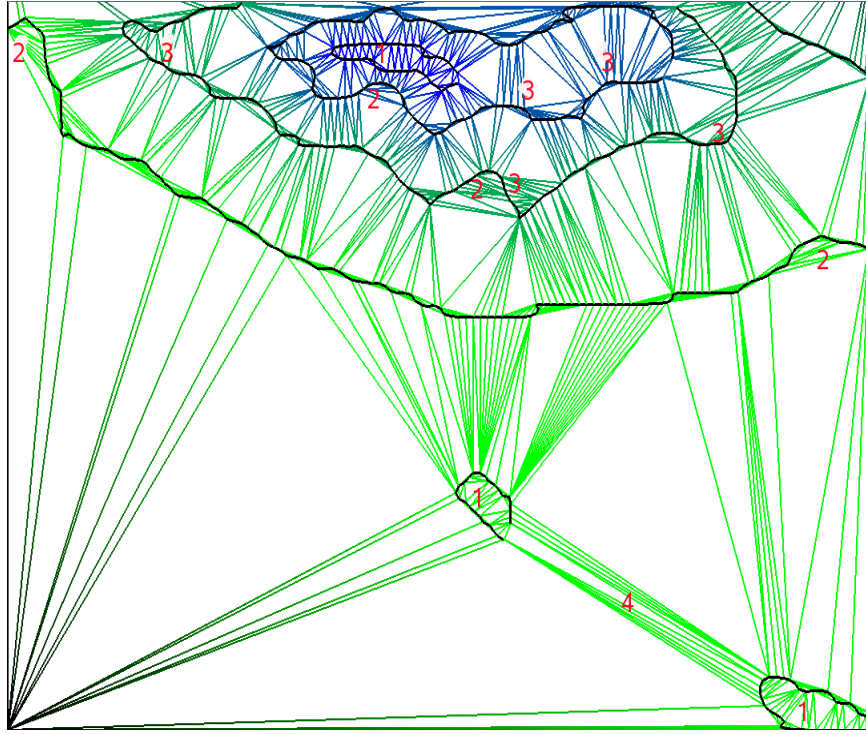


Figura 2.5: Triangulación empleando solo puntos muestreados de las curvas de nivel

Para evitar la presencia de los *triángulos planos* nos auxiliamos en el elemento geométrico llamado *esqueleto*.

2.5.1. Condición del muestreo para calcular el esqueleto

Hasta este momento se tiene que las curvas de nivel representadas por polinomios cúbicos (*splines*), se tiene que calcular el esqueleto a las curvas de nivel. Para ésto se utilizó el procedimiento propuesto en [17]. Este procedimiento requiere que las curvas *splines* se encuentren muestreadas de tal forma que sea posible calcular el esqueleto. Veamos primero algunas definiciones necesarias para entender el problema.

Sea F una curva en el plano, tal que F no tiene ramificaciones ni intersecciones consigo misma entonces se dice que F es una *curva suave*.

Las curvas de nivel cumplen las características de *curva suave*. Si la curva suave se muestrea con la densidad suficiente, la curva sobre las muestras es una *reconstrucción poligonal* de la curva.

Sea F una *curva suave*, y sea $S \subset F$ un conjunto finito de puntos muestras sobre F .

La reconstrucción poligonal de F a partir de S es un grafo que conecta cada par de puntos adyacentes a lo largo de F , y no otras. A esta reconstrucción poligonal se le conoce como *corteza* [2].

En la figura 2.6 (a) se presenta, a modo de ejemplo, el dibujo de un ganzo formado por tres curvas (ojo, patas y cuerpo) en la parte (b) el conjunto muestra para cada una de estas curvas (c) la reconstrucción poligonal de cada curva.

Nótese que el número de muestras en la espalda de la figura es menor que el número de muestras en la cabeza y patas de ésta. Este ejemplo fue construido manualmente para ilustrar el proceso de reconstrucción poligonal [17].

La característica de muestreo que debe cumplir S para garantizar una reconstrucción poligonal adecuada de F a partir de S , está basada en una función del *tamaño de la característica local*, la cual, en cierto sentido, cuantifica el “nivel de detalle” local en un punto sobre F .

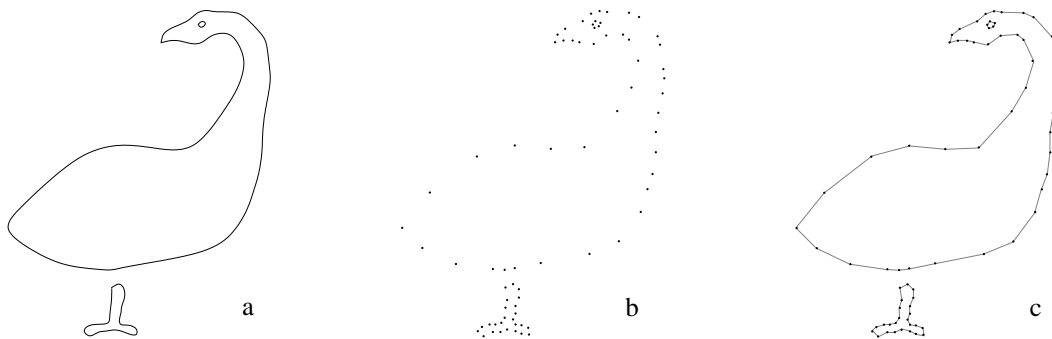


Figura 2.6: (a) Curvas que definen el dibujo de un ganzo (b) Conjunto de puntos muestra de las curvas (c) Reconstrucción poligonal de las curvas.

El *tamaño de la característica local*, $TCL(p)$, de un punto $p \in F$ es la distancia euclidiana de p al punto más cercano en el eje medio de la curva [2]. Puesto que se hace uso del eje medio, la definición del TCL depende tanto en p como de la cercanía de ciertos rasgos característicos de la curva.

Entonces la condición de muestreo queda como sigue:

F está r -muestreada por un conjunto muestra S si: para cada punto $p \in F$, p guarda una distancia no mayor a $r \cdot TCL(p)$ con una muestra $s \in S$, para valores $r \leq 1$, donde r es la razón de muestreo.

De acuerdo a [17] si se emplea un valor para $r < 0.42$ se garantiza una densidad de muestreo adecuada, de modo que en el proceso de la reconstrucción de la corteza del

conjunto de puntos de muestra se formen aristas sólo entre cada par de puntos adyacentes.

En la tesis de Abigail Martínez [17] (apéndice C) se demostró de forma experimental que si se usa un $r > 0.42$ en algunas zonas de la corteza, la construcción del *esqueleto* no es la correcta.

La construcción del esqueleto es una aproximación al esqueleto real (eje medio) la cual presenta ciertas variaciones en función del conjunto muestra; según [2] y verificado experimentalmente en la tesis de Abigail Martínez no se garantiza que estas variaciones a lo largo de esta construcción sean aceptable cuando en el muestreo se usan valores de $r > 0.42$.

Ahora bien, si en la tesis de Abigail Martínez se aproximó el eje medio usando procesamiento de imagen; en este trabajo de tesis no fue posible utilizar este proceso debido a que las curvas están en forma vectorial.

La solución al problema de elegir la razón de muestreo que se propuso es la de muestrear las curvas de nivel a 0.42 por la mínima distancia entre curvas de nivel. De esta manera se garantiza que el esqueleto se aproxime al eje medio entre un par de curvas como se puede observar gráficamente con la figura 2.7.

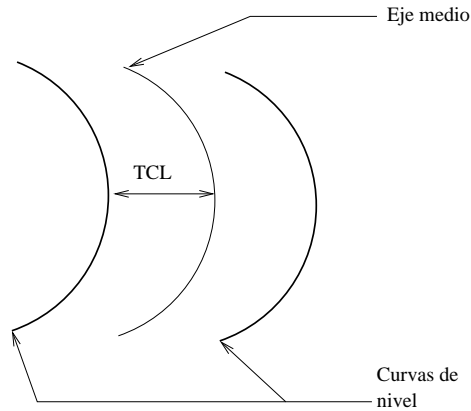


Figura 2.7: La mínima distancia entre el conjunto de curvas de nivel, TCL y el eje medio. Las curvas deben muestrearse a 0.42 por la mitad de la distancia mínima entre curvas de nivel.

En resumen el procedimiento usado en esta tesis es el siguiente:

1. Se encuentra la mitad de la distancia mínima entre curvas de nivel en todo el conjunto de curvas.

Número de dimensiones de los puntos
Número de puntos, en este caso supondremos que tiene n puntos
x_1 x_1 x_2 y_2 x_3 y_3 \dots \dots \dots x_n y_n

Tabla 2.2: Se muestra el formato del archivo de entrada para el *script qvoronoi*.

2. Se calcula 0.42 por la distancia encontrada en el paso uno.
3. Se muestrean todas las curvas de nivel al valor obtenido en el paso dos.
4. Se calcula el esqueleto.
5. Se seleccionan solo las secciones del esqueleto que eliminan los *triángulos planos*.
6. Se remuestrea el esqueleto al valor dado en el paso uno (la mitad de la mínima distancia entre curvas) para reducir el número de puntos utilizados en la visualización.

2.5.2. Cálculo del *esqueleto*

El cálculo del *esqueleto* se realizó empleando el método de [9], al cual se le añadieron un par de condiciones que nos permite seleccionar únicamente las aristas del diagrama de *Voronoi* que son útiles para eliminar los *triángulos planos*.

Prueba dentro del círculo

La prueba es aplicada a cuatro puntos en el plano, de los cuales dos pertenecen a una arista del diagrama de *Delaunay* y los restantes pertenecen a la arista dual del diagrama de *Voronoi*.

Lo que se requiere para el cálculo del esqueleto es una lista de pares de aristas duales entre el diagrama de *Voronoi* y el diagrama de *Delaunay*. Para calcular esta lista se emplea la biblioteca de *qhull* [4]. El *script* que se encarga de este calculo es `qvoronoi` con la siguiente sintaxis `qvoronoi p Fv <Archivo de puntos >Archivo de salida`. El formato del archivo de salida se describe en la tabla 2.2.

n	número de vértices <i>Voronoi</i>
x_1	y_1
x_2	y_2
x_3	y_3
..	.
..	.
..	.
x_n	y_n
m	número de pares de aristas duales
$r_1 s_1$	$p_1 q_1$
$r_2 s_2$	$p_2 q_2$
$r_3 s_3$	$p_3 q_3$
..	.
..	.
..	.
$r_m s_m$	$p_m q_m$

Tabla 2.3: Se muestra el formato del archivo de salida del *script qvoronoi*.

Donde la opción **p** indica que el archivo de salida contendrá la lista de los vértices del diagrama de *Voronoi*.

La opción **Fv** indica que el archivo de salida también tendrá la lista de los valores de los índices que hacen referencia a los pares de aristas duales.

Finalmente el formato del archivo que obtendremos del *script qvoronoi* es el de la tabla 2.3. Donde r_i y s_i corresponden a los vértices de la arista de *Voronoi*, mientras que p_i y q_i son los vértices de *Delaunay*.

La arista de *Delaunay* pertenece a la corteza cuando existe un círculo a través de sus dos vértices que no contiene a ninguno de los vértices de *Voronoi* asociados, de otro modo la arista de *Voronoi* pertenece al *esqueleto* [7].

La forma de evaluar esta prueba se basa en las coordenadas de los puntos a los cuales se les aplica.

$$DentroDelCirculo = \begin{vmatrix} 1 & X_A & Y_A & X_A^2 + Y_A^2 \\ 1 & X_B & Y_B & X_B^2 + Y_B^2 \\ 1 & X_C & Y_C & X_C^2 + Y_C^2 \\ 1 & X_D & Y_D & X_D^2 + Y_D^2 \end{vmatrix} > 0 \quad (2.36)$$

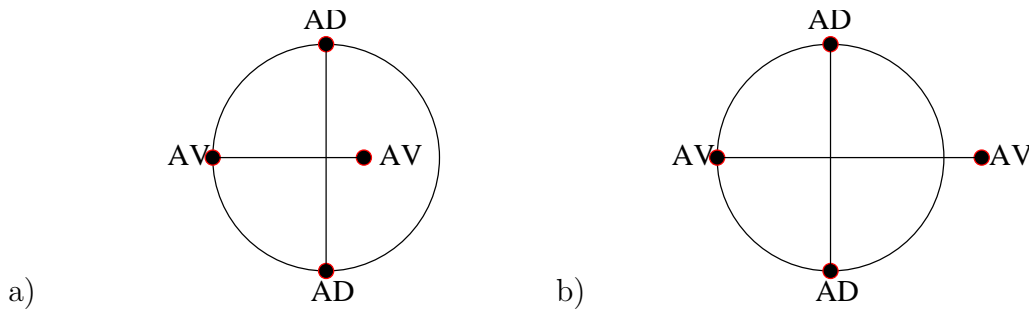


Figura 2.8: a) El caso en que la arista de *Voronoi* pertenece al esqueleto b) El caso en que la arista de *Delaunay* pertenece al *esqueleto*.

tal que $A = (x_A, y_A)$, $B = (x_B, y_B)$, $C = (x_C, y_C)$, $D = (x_D, y_D)$; donde A y B son los vértices que pertenecen a la arista del diagrama de *Delaunay*, mientras que C y D son los vértices de la arista dual del diagrama de *Voronoi*; la prueba es mayor que cero si el vértice D se encuentra dentro del círculo, la Ec. 2.36 es el resultado del trabajo de [10].

Al aplicar la prueba de *dentro del círculo* debemos considerar que no conocemos el orden en que se encuentran los puntos que representan al par de aristas. Tomando en cuenta el orden en que se presentan los puntos existen cuatro posibles casos en los que la arista de *Voronoi* pertenece al *esqueleto*.

En la figura 2.9 el primer caso ocurre cuando el punto D (D es un vértice del diagrama de *Voronoi*) queda contenido en el círculo que se traza en dirección opuesta a las manecillas del reloj, los puntos A , C y B están sobre el círculo, donde (A y B) son los vértices de la arista de *Delaunay*.

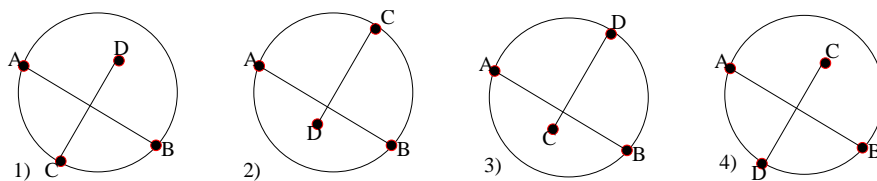


Figura 2.9: Los cuatro casos en que la arista de *Voronoi* pertenece al esqueleto

El segundo caso de la Fig. 2.9 corresponde al caso en que el punto D queda dentro del círculo trazado en dirección de las manecillas del reloj, los puntos que se encuentran sobre el círculo son A , C y B .

En el tercer caso de la figura 2.9 el punto C queda dentro del círculo que se traza en la dirección de las manecillas del reloj, donde A , D y B están sobre el círculo.

Finalmente el cuarto caso la figura 2.9 muestra que el punto C está contenido en el círculo que esta vez se traza en sentido contrario a las manecillas del reloj quedando sobre

Caso	Prueba del ciclo	Dirección del ciclo
1	$PruebaDelCirculo(A, C, B, D) > 0$	en dirección contraria a las manecillas del reloj
2	$PruebaDelCirculo(A, C, B, D) \leq 0$	en dirección de las manecillas del reloj
3	$PruebaDelCirculo(A, D, B, C) \leq 0$	en dirección de las manecillas del reloj
4	$PruebaDelCirculo(A, D, B, C) > 0$	en dirección contraria a las manecillas del reloj

Tabla 2.4: Se muestra los casos en que una arista de *Voronoi* pertenece al *esqueleto*.

el círculo los puntos A , D y B .

Como puede observarse en las figuras ??, ??, ?? y ?? el orden en el cual se presentan los vértices no siempre es el mismo. Por sí sola esta prueba no es capaz de establecer el orden en que se encuentran los vértices del par de aristas; para solucionar este problema es necesario evaluar el siguiente determinante:

$$\Delta = \begin{vmatrix} X_A & Y_A & 1 \\ X_B & Y_B & 1 \\ X_C & Y_C & 1 \end{vmatrix} \quad (2.37)$$

donde Δ es mayor o igual que cero si y sólo si A , B y C forman un ciclo en sentido contrario a las manecillas del reloj [16].

Al calcular los determinantes 2.36 y 2.37 es posible calcular el *esqueleto*, las condiciones que se deben cumplir para que una arista de *Voronoi* pertenezca al esqueleto se muestran en la tabla 2.4.

Con las condiciones expresadas en la tabla 2.4 es posible seleccionar todas las aristas del diagrama de *Voronoi* que pertenecen al esqueleto.

2.5.3. Seleccionando las secciones del esqueleto que evitan *triángulos planos*

Hasta este punto se ha calculado en su totalidad el esqueleto. Al construir el diagrama de *Dealaunay* con los puntos de las curvas de nivel y los puntos que forman al esqueleto se garantiza que no existirán *triángulos planos*; sin embargo no todos los puntos del *esqueleto* son necesarios para evitar la presencia de los *triángulos planos*.

El problema que se presenta es el de seleccionar sólo aquellas secciones que sean útiles para eliminar los *triángulos planos*, con el objetivo de construir el diagrama de *Dealaunay* con estas secciones y los puntos que conforman a las curvas de nivel.

En el trabajo [7] se muestra una posible solución la cual consiste en los siguientes pasos:

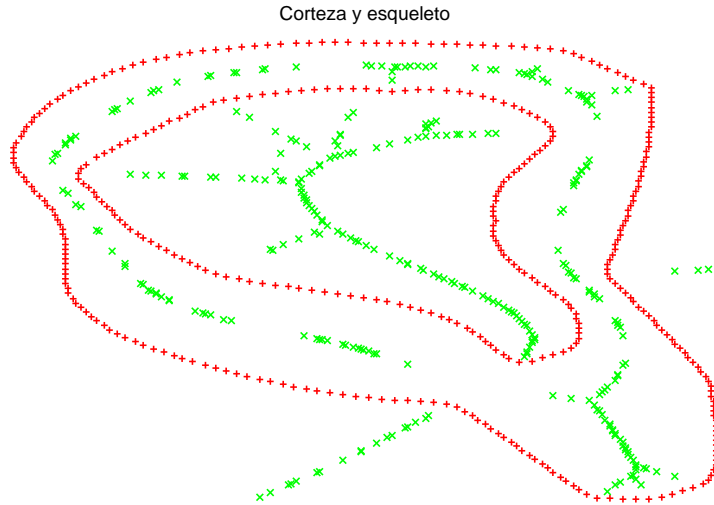


Figura 2.10: Se muestra con * las *curvas de nivel*, mientras que con × el *esqueleto*.

1. Calcular el esqueleto.
2. Ordenar al esqueleto en grafos.
3. Calcular el diagrama de Delaunay con los puntos que conforman a las curvas de nivel y el esqueleto.
4. Clasificar las ramas de los grafos dependiendo la conectividad que presenten en el diagrama de Delaunay.

Las secciones del esqueleto que eliminan a los *triángulos planos* son representadas por ramas que presentan conectividad con puntos (de las curvas de nivel) con la misma altura.

Cuando se tiene un conjunto de datos suficientemente grande (cientos de miles de puntos) la clasificación del *esqueleto* resulta ineficiente.

Para no tener que realizar la clasificación al esqueleto completo se buscaron una serie de características que deben cumplir las aristas de Delaunay a las que se les aplica la prueba *dentro del círculo*, de tal forma que sólo se le aplique esta prueba a las aristas de Voronoi que posiblemente pertenezcan al esqueleto y que eliminen los *triángulos planos*. Se le aplica la prueba *dentro del círculo* a los pares de aristas duales con las siguientes características:

1. Los vértices de la arista de Delaunay deben tener la misma altura (el mismo valor en la ordenada z) ya que los vértices de los *triángulos planos* se caracterizan por tener el mismo valor en la ordenada z .

2. Los vértices de la arista de Delaunay no deben ser vértices contiguos sobre la misma polilínea que representa a una determinada *curva de nivel*.
3. La distancia entre los vértices de *Voronoi* no debe superar un umbral.

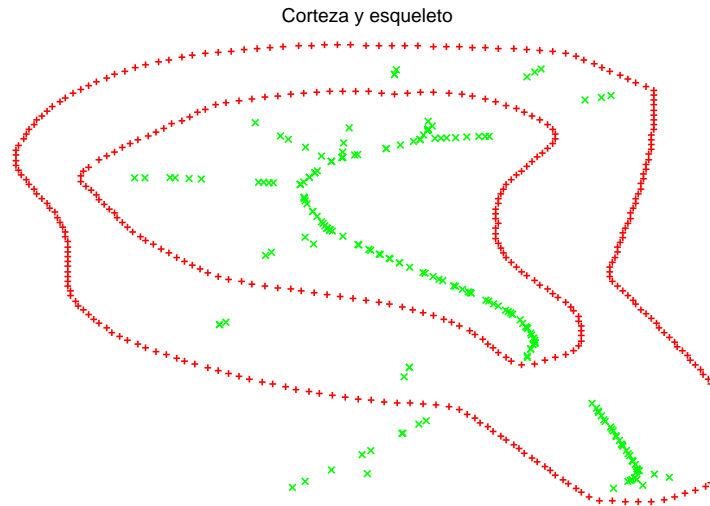


Figura 2.11: Las *curvas de nivel* son respresentadas con *, mientras que con × se representan las secciones del *esqueleto* que evitan a los triángulos planos.

Al agregar estas condiciones se seleccionan aquellas aristas que se encuentran en las cimas, en las secciones reentrantes o entre curvas de nivel con la misma altura.

El umbral no es otro que la mitad de la razón de muestreo, ya que si la distancia entre los vértices de *Voronoi* es mayor no quedarán conectados al construirse el diagrama de *Delaunay*.

La figura 2.11 muestra el esqueleto calculado mediante el algoritmo 2.

2.5.4. Cálculo de alturas a los vértices del *esqueleto*

En este momento los vértices que forman las aristas del *esqueleto* carecen de altura, ya que estos puntos son vértices del diagrama de *Voronoi*.

El cálculo de la altura de cada uno de los vértices se debe realizar tomando en cuenta el contexto en el que se encuentra, en otras palabras, se debe tomar en cuenta la conectividad que presenta en el diagrama de *Delaunay*, considerando también su posición dentro del esqueleto. A todo lo anterior es lo que llamamos contexto.

Algoritmo 2 Cálculo del esqueleto

Entrada: Los puntos que representan a las curvas de nivel, los puntos del esqueleto, la lista de pares duales.

Salida: Las secciones del esqueleto útiles para eliminar a los *triángulos planos*.

- 1: **mientras** haya elemento en la lista de aristas duales **hacer**
 - 2: **si** los vértices de la arista del Diagrama de *Delaunay* tienen la misma altura y no son contiguos sobre la curva de nivel. **entonces**
 - 3: **si** los vértices de la aristas del Diagrama de *Voronoi* no superan el umbral. **entonces**
 - 4: **si** se cumple alguna de las condiciones de la tabla 2.4 **entonces**
 - 5: Agregar la arista de *Voronoi* al esqueleto
 - 6: **fin si**
 - 7: **fin si**
 - 8: **fin si**
 - 9: **fin mientras**
-

Para encontrar el contexto de cada punto se debe solucionar los dos siguiente problemas:

1. Organizar las aristas en grafos con el objetivo de establecer un orden ya que hasta este momento sólo hemos seleccionado las aristas de *Voronoi* útiles para eliminar a los *triángulos planos*.
2. Establecer una vecindad, esto es conocer cual es la conectividad sobre el diagrama de *Delaunay* que tiene el grafo, no sólo se toma en cuenta el vértice al que se le calcula la altura; sino todos los vértices que conforman el grafo. Con dicha conectividad es posible establecer una clasificación de los grafos. Al tener una clasificación de los grafos se puede elegir entre alguna de las ecuaciones del trabajo Dakowicz y Gold [7].

Para solucionar el problema de organizar los vértices en grafos se empleó uno de los *scripts* resultado del trabajo de [17], éste requiere como entrada un archivo con el formato de la tabla 2.5.

Originalmente los vértices eran bidimensionales, lo cual se modificó , ahora en la ordenada z se guarda el valor del radio del círculo menor. El radio del círculo menor es un parámetro que nos sirve para calcular la altura del vértice en cuestión, ya que nos brinda una relación del vértice dentro del grafo; dicho parámetro es obtenido durante la *prueba de dentro del círculo*.

Una vez que los vértices del grafo están organizados, se construye el diagrama de *Delaunay* utilizando los puntos que conforman a las curvas de nivel y los puntos del esqueleto. Posteriormente se procede a clasificar cada grafo del esqueleto con base conectividad que

número de vértices del esqueleto
vértice uno
vértice dos
..
..
..
vértice n-ésimo
número de aristas
se listan las parejas de los vértices que forman cada una de las aristas

Tabla 2.5: Se muestra el formato del archivo de entrada requerido por el *script* que calcula el *esqueleto*.

presente en el diagrama de Delaunay, los cuatro tipos en los que se clasifican los grafos son:

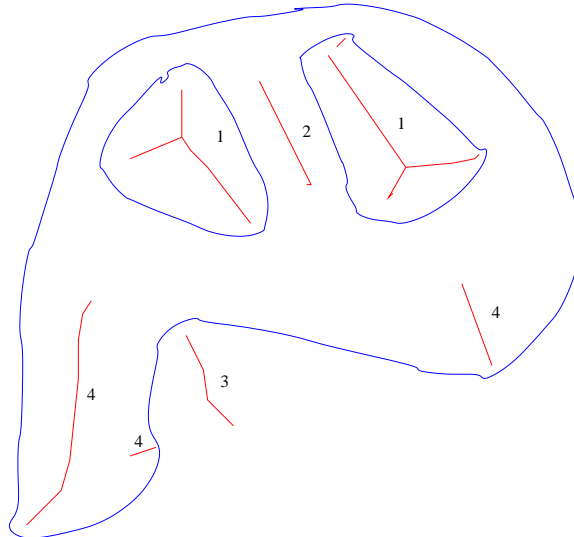


Figura 2.12: Los puntos del esqueleto que se utilizan para eliminar los *triángulos planos* forman grafos conectados, estos grafos pueden clasificarse en cuatro tipos. El tipo 1 es dentro de una cima, el tipo 2 es entre dos curvas de nivel, el tipo 3 es reentrante fuera de una curva, el tipo 4 es reentrante dentro de una curva de nivel.

1. El primer caso corresponde a las Cimas, estos grafos son aquellos que están rodeados por puntos que pertenecen a la misma *curva de nivel*.
2. El segundo caso corresponde a los grafos entre dos o más *curvas de nivel* con la misma altura.

3. El tercer caso corresponde a los grafos que se encuentran en secciones reentrantes, pero el grafo se encuentra por fuera de la *curva de nivel*.
4. El cuarto caso corresponde a los grafos que se encuentran en secciones reentrantes pero donde el grafo se encuentra por dentro de la *curva de nivel*.

La figura 2.12 muestra cada uno de los caso enumerados en el mismo orden en que se han listado los casos.

Cada uno de estos casos corresponde a un contexto distinto, ya que cada uno de estos contextos indica una tendencia distinta del terreno. La altura de los vértices debe ser calculada tomando en cuenta el caso en cual se encuentra el grafo al cual pertenece el vértice.

Cálculo de alturas a vértices de cimas

Como se ha mencionado anteriormente las cimas son aquellas secciones del *esqueleto* que se encuentran rodeadas de una sola *curva de nivel*, la figura 2.13 muestra este caso, además muestra el mayor de los radios mínimos, este es el vértice que tendrá mayor altura, también muestra el radio del punto P .

El cálculo de las alturas se hace empleando la Ec. 2.38.

$$z = z_1 + a \frac{R_i}{R_{max}} \quad (2.38)$$

z es la altura que se desea calcular.

z_1 es la altura de la *curva de nivel* que rodea al grafo.

R_i es el radio mínimo de los círculos centrados en el vértice al que se le desea calcular, este radio es la arista de Delaunay de menor longitud, teniendo como uno de sus vértices, el vértice del esqueleto al que se le calcula la altura.

R_{max} es el valor del radio con mayor longitud. Éste pertenece al vértice con mayor altura, por lo que las alturas se normalizan con respecto a este vértice.

a es la mitad de la diferencia de alturas los un nivel al siguiente nivel.

Las alturas están normalizadas entre z_1 y $z_1 + a$, el comportamiento que tiene la Ec. 2.38 es que entre más lejanos se encuentren los vértices del esqueleto de la curva de nivel a la que se encuentren asociados, las alturas de los vértices tendrán una magnitud mayor.

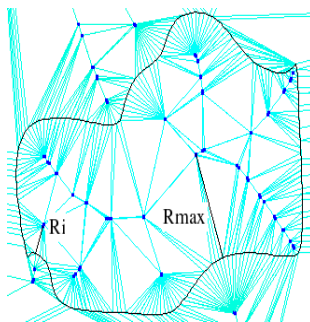


Figura 2.13: Se muestran una sección del *esqueleto* que se encuentra en una cima.

Cálculo de alturas a vértices entre curvas con en el mismo nivel

Las secciones del esqueleto que caen en este caso se caracterizan porque los vértices que conforman a estas secciones tienen conectividad con más de una curva en el mismo nivel. Las curvas en el mismo nivel tienen la misma altura.

La altura para estos vértices se generalizó a la siguiente fórmula:

$$z = z_1 + \frac{a}{2} \quad (2.39)$$

donde a es la mitad de la diferencia de alturas los de un nivel al siguiente nivel y z_1 es la altura de la curva de nivel asociada. Se eligió este valor, ya que esta sección del esqueleto representa a una depresión entre las curvas.

La figura 2.14 muestra una sección del esqueleto que se encuentra rodeado por dos curvas con la misma altura.

Cálculo de alturas a vértices sección reentrante

Las secciones reentrantes en general corresponden a lo que comúnmente, en algunas zonas del país, se conoce como las *faldas de los cerros*, éstas son declives que parten de una cima o de un valle, dependiendo el punto de referencia que se desee tomar.

El problema que se tiene con estas secciones reentrantes es decidir si la sección del esqueleto tiende a aumentar su altura con respecto a la curva que tiene más cercana o tiende a disminuirla.

Para poder decidir si la sección del esqueleto tiende a aumentar o disminuir, se debe verificar si la sección del esqueleto se encuentra dentro o fuera de la curva que tiene cerca.

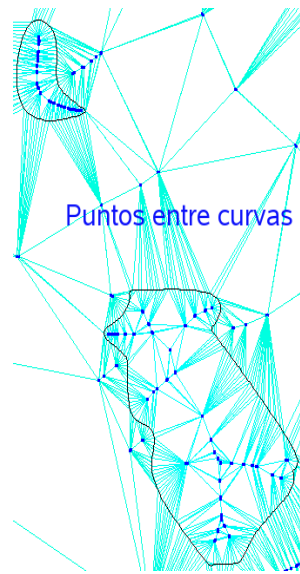


Figura 2.14: La figura muestra el caso en el cual una sección del *esqueleto* se encuentra entre más de una *curva de nivel* con la misma altura.

Los puntos dentro de la curva de nivel deben tener mayor altitud que los puntos de la curva; mientras que los que estén fuera de ella deben tener menor altitud.

Para calcular la altura de los vértices de las secciones reentrantes que se encuentran fuera de la curva de nivel, se utilizó la siguiente ecuación:

$$z = z_1 - a \frac{R_i}{R_{max}} \quad (2.40)$$

mientras que para calcular la altura de los vértices de las secciones del esqueleto que se encuentran dentro de la curva se realizó la siguiente ecuación:

$$z = z_1 + a \frac{R_i}{R_{max}} \quad (2.41)$$

donde z es la altura que se desea calcular.

z_1 es la altura de la *curva de nivel* más cercana al grafo.

R_i es el radio mínimo de los círculos centrados en el vértice al que se le desea calcular, este radio es la arista de Delaunay de menor longitud, teniendo como uno de sus vértices, el vértice del esqueleto al que se le calcula la altura.

R_{max} es el valor del radio con mayor longitud.

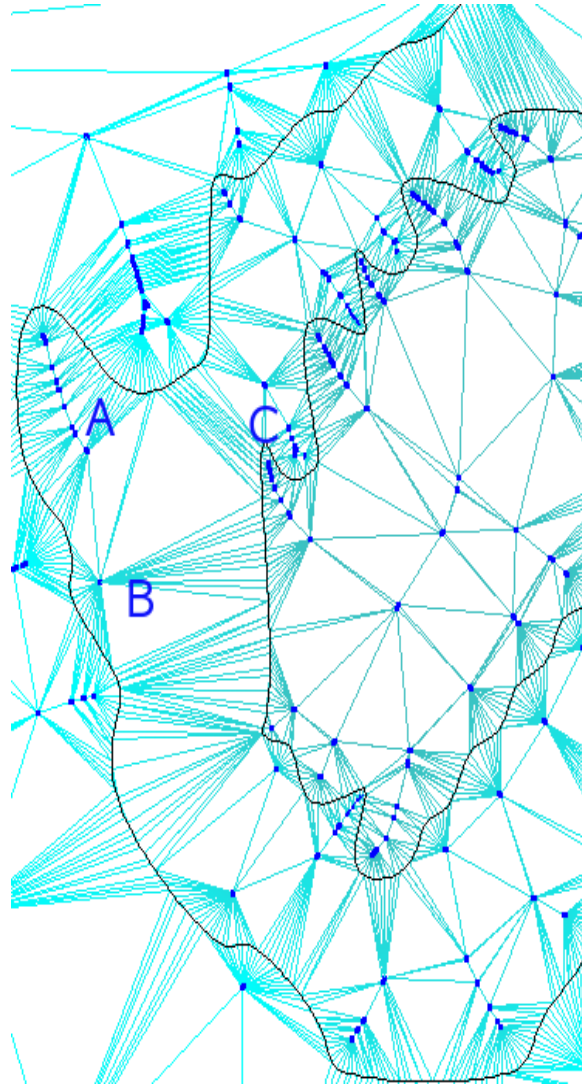


Figura 2.15: La figura muestra tres casos de secciones reentrantes etiquetados como A, B y C.

a es la mitad de la diferencia de alturas de un nivel al siguiente nivel.

Tanto R_{max} como R_i se calculan como en el caso de las cimas.

Para establecer si la sección del esqueleto se encuentra dentro o fuera de la curva de nivel más cercana, es necesario definir una vecindad. Una sección de esqueleto es vecina de otra si y sólo si existe un camino de cualquier vértice de la sección a la otra sección pasando únicamente por vértices del esqueleto.

En la figura 2.15 A y B son vecinos; mientras que C no es vecino ni de A ni de B , ya que si C fuese vecino de A también sería vecino de B ; por otro lado si C fuese vecino de B también sería vecino de A .

Al establecer una vecindad se determina entre que curvas de nivel se encuentra la sección del esqueleto, ésto nos permite decidir si esta sección está por dentro o por fuera de la curva.

En la figura 2.15 A y B se encuentran dentro de la curva de nivel que tienen cerca, mientras que C se encuentra fuera de la curva de nivel que tiene más cerca.

Capítulo 3

Agrupamiento de datos

Otro objetivo del este trabajo es analizar e implementar un algoritmo que permita: ordenar un conjunto de datos vectoriales con respecto a sus características espaciales y seleccionar del conjunto ordenado un subconjunto que cumpla ciertas características espaciales.

Lo que se busca es particionar el espacio de datos en sub-espacios con el proposito de trabajar con un subconjunto de los datos para facilitar la manipulación de éstos.

Por particionar nos referimos al proceso de dividir un espacio (normalmente un espacio euclidiano) en dos o más conjuntos disjuntos, en otras palabras, el particionando divide un espacio en regiones no superpuestas; cualquier punto se encuentra en una, y sólo una de las regiones [22].

Si el algoritmo cumple con estas características, será posible acceder a los subconjuntos de datos de interés.

3.1. Planteamiento del problema

Se parte de un conjunto de datos vectoriales masivo (3758 curvas de nivel, con un total de 391534 vértices; el número de vértices fue obtenido antes de re-muestrear las curvas de nivel) contenidos en un plano (x, y) , el problema es proponer una estructura de datos, de tal forma, que organice el conjunto de datos masivo en regiones del plano (x, y) .

La figura 3.1 muestra varias curvas de nivel sobre un plano el cual se ha dividido en cuatro regiones; como se puede observar una curva está contenida en una o más regiones, por lo que cada curva puede ser representada en al menos un segmento.

En este capítulo se muestran dos alternativas para solucionar los problemas antes descritos. La primera consiste en mapear el plano a un arreglo bidimensional, donde cada entrada representa una región del plano a manipular y la segunda consiste en construir un árbol cuyas hojas representan las regiones en que se ha dividido el plano (x, y) .

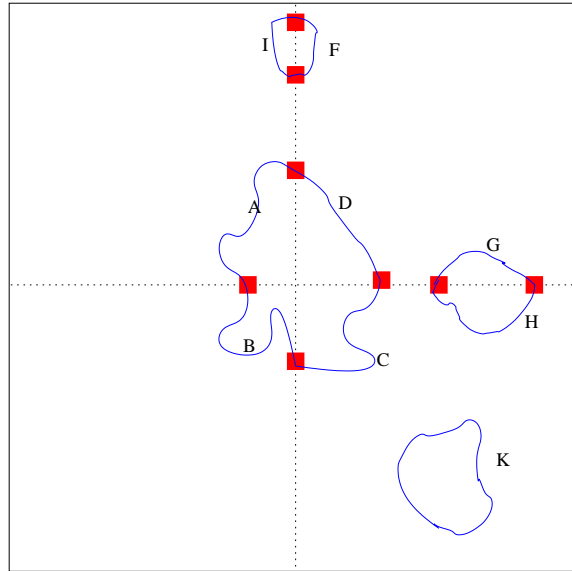


Figura 3.1: La figura muestra un plano dividido en 4 regiones, en azul curvas de nivel, en rojo los puntos de intersección de las curvas de nivel con las fronteras de las regiones.

3.2. Utilizando una rejilla

El método descrito en esta sección consiste en dividir el plano en regiones, esto puede idealizarse como el sobreponer una rejilla sobre el plano (véase figura 3.2) donde cada elemento de la rejilla es una región del plano, de esta manera todas y cada una de las curvas de nivel quedan agrupadas en una o más regiones.

Los requerimientos son :

1. Las dimensiones de los elementos de la rejilla.
2. Lista de curvas de nivel.
3. Dimensiones del plano (x, y) .

1	2
3	4

Figura 3.2: El plano que se muestra en la figura tiene sobrepuesto una rejilla de dimensiones 2×2 , los datos están divididos en 4 regiones.

Al implementar este método se representó la rejilla mediante un arreglo bidimensional, donde cada uno de los elementos del arreglo representa una región del plano; por lo que se debe mapear la rejilla al arreglo mediante una función uno a uno entre los elementos de la rejilla y las regiones del plano.

En las subsecciones siguientes se describirán a detalle el tipo de dato del arreglo.

Cada una de las regiones, representadas por las entradas del arreglo, contiene una lista con los segmentos que contiene dicha región.

3.2.1. Dimensiones de la rejilla

Las dimensiones de la rejilla es el número de renglones y el número de columnas que tendrá la rejilla, al conocer ésto se puede saber en cuantas regiones se ha dividido el espacio de datos.

El número de columnas se calcula con la siguiente ecuación:

$$e = \lceil c/a \rceil \quad (3.1)$$

e es el número de columnas.

c es la longitud del plano de datos sobre el eje x véase la Fig. 3.3.

a es la longitud de las regiones sobre el eje x véase la Fig. 3.3.

El número de renglones se calcula con la siguiente ecuación:

$$f = \lceil d/b \rceil \quad (3.2)$$

f es el número de renglones.

d es la longitud del plano de datos sobre el eje y véase la Fig. 3.3.

b es la longitud de las regiones sobre el eje y véase la Fig. 3.3.

3.2.2. Estructuras de datos

En esta subsección se describen los tipos de datos empleados durante la implementación.

La estructura de datos que representa a cada uno de los elementos de la rejilla es la siguiente:

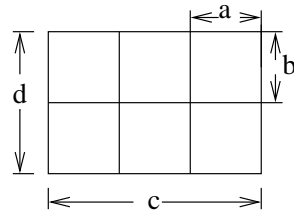


Figura 3.3: La figura muestra el plano x, y con dimensiones $c \times d$, mientras que los elementos de las rejilla tienen dimensiones $a \times b$.

```
typedef struct Regiones {
    struct Segmentos *siguiente;
    struct Regiones *CurvaSiguiente;
    struct CurvaNivel* curva;
};
```

CurvaNivel *curva: es la referencia a la curva de nivel a la cual se le está añadiendo a la lista de regiones.

Segmentos *siguiente: es la raíz de lista de segmentos de la curva de nivel que son contenidos dentro la región.

Regiones *CurvaSiguiente: este elemento nos permite crear la lista de elementos de tipo Regiones en cada una de las entradas del arreglo.

Cada una de las regiones es la raíz de una lista de elementos de este tipo, esta estructura nos sirve para construir una lista de segmentos en cada uno de los nodos de la lista de regiones.

La estructura **Segmentos** se muestra a continuación así como una breve descripción de ésta.

```
typedef struct Segmentos{
    int puntoInicial, NumPuntos;
    int x1,x2,y1,y2;
    struct Segmentos *siguiente;
};
```

Esta estructura proporciona la siguiente información:

1. El punto a partir del cual la curva pertenece a la región.
2. El número de puntos de la curva que están contenidos en la región.

3. Indica si la curva inicia en otra región.

4. Indica si la curva continua más de esta región.

`int puntoInicial` indica el punto a partir del cual la curva de nivel tiene intersección con la región actual, especificando el punto inicial del segmento.

`int NumPuntos` indica el número de elementos contenidos partiendo del punto inicial.

Los elementos `int x1 y1` tendrán un valor distinto de cero si la curva de nivel inicia en otra región, de esta manera, estas coordenadas indican el punto sobre la frontera de la región en que inicia el segmento.

Los elementos `int x2 y2` tendrán un valor distinto de cero si la curva de nivel continua más allá de la región actual, de esta manera, estas coordenadas indican el punto sobre la frontera de la región en que finaliza el segmento.

Cada uno de los elementos del arreglo es la raíz de una lista de elementos de tipo *Regiones* véase la figura 3.4.

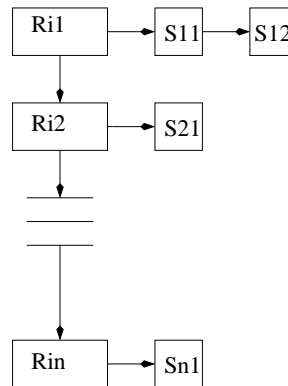


Figura 3.4: Cada una de las regiones es una lista de estructuras de tipo *Regiones* en este caso se muestra la región R_i , donde cada elemento de esta lista es la raíz de una lista de la estructura del tipo *Segmentos*.

3.2.3. Mapeo del plano

En esta subsección se describe cómo es mapeada la rejilla a un arreglo y las operaciones empleadas para seleccionar el elemento de interés del arreglo.

El mapear la rejilla al arreglo consiste en asignar una única región a cada uno de los elementos del arreglo tomando en cuenta la posición de la región dentro de la rejilla, esto

puede hacerse numerando las regiones de la misma forma en que se numeran los elementos del arreglo; de esta forma el problema radica en indexar los elementos del arreglo.

El plano ha sido representado mediante un arreglo bidimensional por lo que se cuenta con dos índices. A los cuales denotaremos como i y j respectivamente, los intervalos en los que se encuentran definidos son: $0 \leq i < e$ y $0 \leq j < f$ donde e es el número de columnas del arreglo y f es el número de renglones.

Al indexar a las regiones las combinaciones de los valores de estos índices deben referir a una única región.

La ecuación empleada para realizar la indexación es la siguiente:

$$R_k = i_k + j_k * e \quad (3.3)$$

R_k es la k -ésima región.

i_k es el valor del índice i de la región R_k .

j_k es el valor del índice j de la región R_k .

Al emplear la Ec. 3.3 la indexación se realiza cómo en la figura 3.5.

Una vez que se tienen las dimensiones del arreglo con que se representa el plano el siguiente paso consiste en llenar este con los segmentos de las curvas de nivel, el algoritmo que realiza esto es el siguiente:

Algoritmo 3 Seccionado de datos

Entrada: La rejilla, la lista de curvas.

Salida: Ordena a las curvas en regiones.

- 1: **mientras** Haya elemento en la lista de curvas **hacer**
 - 2: Encontrar el elemento inicial y final de la rejilla que contengan a la curva.
 - 3: Encontrar las secciones que tengan intersección con algún segmento de la curva de nivel.
 - 4: Asignar los segmentos de la curva a las regiones correspondientes.
 - 5: **fin mientras**
-

La búsqueda de la región que contiene al punto p con coordenadas x_1, y_1 consiste en encontrar el valor de los índices de dicha región, para ello se emplean las siguientes ecuaciones:

$$i_{x_1} = \frac{a}{x_1} \quad (3.4)$$

R3	R4	R5
R0	R1	R2

R01	R11	R21
R00	R10	R20

Figura 3.5: Cada uno de los elementos de la rejilla denotadas por R_k ha sido mapeado a un elemento del arreglo $R_{i,j}$.

i_{x_1} es el valor del índice i de la región que contiene al punto p .

a es la longitud de las regiones sobre el eje x véase la Fig. 3.3.

$$j_{y_1} = \frac{b}{y_1} \quad (3.5)$$

j_{y_1} es el valor del índice j de la región que contiene al punto p .

b es la longitud de las regiones sobre el eje y véase la Fig. 3.3.

Al encontrar los valores i_{x_1} y j_{y_1} es posible calcular el número de la región que contiene al punto p mediante la Ec. 3.3

3.2.4. Orden y vecindad

Para entender cómo están ordenadas las regiones es importante describir los criterios que se emplearon para este fin. También es importante definir una vecindad, en otras palabras resaltar qué regiones de las que rodean a una determinada región son importantes. Así como es necesario establecer las operaciones para pasar de una región a una de sus vecinas.

Si se desea ordenar las regiones tomando en cuenta su posición dentro de la rejilla primero se debe especificar que criterio se va a emplear.

Antes de describir qué criterio de orden se usó en este trabajo, es importante resaltar algunas características de las regiones.

1. Las regiones son rectangulares y con las mismas dimensiones.
2. Todos los puntos contenidos dentro de las regiones son positivas.
3. Las regiones están delimitadas por las coordenadas x_1, y_1 y x_2, y_2 , donde x_1 es la ordenada de la región más cercana al origen con respecto al eje x , y y_1 es la ordenada de la región más cercana al origen en el eje y , mientras que las ordenas x_2 y y_2 son aquellas ordenas de la región más lejanas al origen en el eje x y y respectivamente, véase la figura 3.6.

4. Las regiones se encuentran contiguas, con esto nos referimos a que no hay ningún punto en el plano que se ha cubierto con la rejilla que no pertenezca al menos a una región.

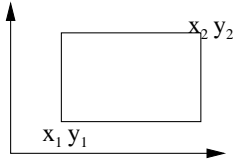


Figura 3.6: La figura muestra las coordenadas que delimitan a la región.

Tomando en cuenta estas características, se puede describir el criterio de *menor que* ó *mayor que* el cual se siguió en este trabajo, dicho criterio es el empleado para ordenar las regiones.

La región R_i es menor que la región R_k si $y_{2R_i} < y_{1R_k}$, pero también R_i es menor que la región R_k si $y_{2R_i} = y_{1R_k}$ y $x_{2R_i} < x_{1R_k}$, en cualquier otro caso R_i es mayor que R_k .

La figura 3.7 muestra la región R , la cual se encuentra rodeada por ocho regiones, de las cuales cuatro son menores y cuatro son mayores.

>	>	>
<	R	>
<	<	<

Figura 3.7: La figura muestra que regiones son mayores o menores que la región R .

Una vez que se tiene un criterio que permite establecer si una región es mayor o menor que otra es posible ordenar las regiones.

La figura 3.8 muestra cómo han sido ordenadas las dieciséis regiones en las que se ha dividido el espacio de datos. El ordenamiento se realiza implícitamente al momento de construir la rejilla.

Hasta ahora se han mencionado los criterios que se han seguido para establecer si una región es mayor o menor que otra, ésto nos sirvió para ordenar las regiones. Ahora falta por definir la vecindad de una determinada región y las operaciones para pasar de una región a una de sus vecinas.

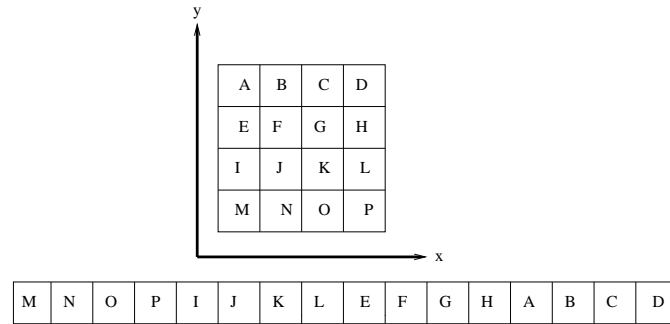


Figura 3.8: La figura muestra arriba una rejilla con 16 elementos, mientras que abajo muestra el orden de dichos elementos.

Como se mencionó al inicio de este capítulo, la importancia de seccionar los datos es crear cúmulos de datos más pequeños que compartan ciertas características espaciales, en este caso se seccionan mediante la característica de pertenecer a una determinada región del plano.

En este punto no es importante describir las operaciones que se le aplican a los cúmulos de datos; lo que si es importante es resaltar que los datos agrupados en las distintas regiones son independientes entre regiones y que el acceso a los datos contenidos en una región se realiza mediante una operación de complejidad constante (véase la subsección 3.2.3).

En este trabajo no sólo nos interesa acceder a una determinada región; sino que en la mayoría de los casos nos interesará acceder a elementos contiguos, ya sea con respecto al eje x o el eje y ; nunca nos interesará acceder a una región que esté contigua variando sobre ambos ejes. Esto tiene que ver con el como se lleva acabo la visualización del área de visión lo cual se describe en el capítulo 4.

La figura 3.9 muestra la vecindad de una región, los vecinos de la región R son aquellas regiones que tengan únicamente una variación sobre el valor de uno de sus índices de ± 1 , en otras palabras las regiones contiguas sobre los ejes x y y .

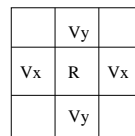


Figura 3.9: La figura muestra los vecinos de R , los Vx son los vecinos sobre el eje x , mientras que los Vy son los vecinos sobre el eje y .

De esta forma el pasar de una región a una de sus vecinas se reduce a incrementar o decrementar el valor del índice del eje sobre el cual queremos desplazarnos.

Si nos interesara saber cuales son los números asignados a las regiones vecinas a una determinada región, esto es posible tomando en cuenta sobre que eje se quiere obtener el vecino.

La k -ésima región denotada por R_k con valores en sus índices de i_k y j_k puede ser ubicada dentro de la rejilla mediante la operación $k = i_k + j_k * e$ (e es el número de columnas de la rejilla).

De esta forma al incrementar o decrementar i_k o j_k según sea el caso es posible encontrar los vecinos de la región R_k .

3.2.5. Características

A continuación se listan algunas ventajas y desventajas de utilizar esta técnica de agrupamiento de datos.

1. Las entradas del arreglo necesarias que tendrá N , donde N es el número de zonas en que se ha dividido el plano (x, y) .
2. La operación de búsqueda de las regiones es constante.
3. El algoritmo puede ser implantado a más dimensiones.
4. Es relativamente sencillo moverse de una región a una de sus vecinas.
5. La desventaja de este método es que las regiones tienen las mismas dimensiones.
6. Como se mencionó, los elementos de la rejilla pueden ser numerados, esta numeración puede ser útil si se almacenase la rejilla en el disco duro.

3.3. Utilizando árboles de particionamiento espacial binario

Los árboles de particionamiento espacial binarios son un método para subdividir recursivamente un espacio en elementos convexos empleando hiperplanos. Esta subdivisión permite obtener una representación de la escena mediante un árbol conocido como árbol PEB [22].

El elemento con el que se particiona el espacio n -dimensional es un elemento con una dimensión menor que el espacio a particionar, por ejemplo, si desea particionar un espacio tridimensional el elemento usado es un plano, por otro lado si se desea particionar un plano, el elemento será una línea, dicho elemento nos sirve para dividir el espacio en dos sub-espacios.

Esto se puede hacer recursivamente hasta satisfacer las necesidades de la aplicación en la cual se vaya a emplear dicho particionamiento.

Los requerimientos para implantar esta alternativa son los siguientes:

1. La descripción de cómo se dividirá el plano (x, y) .
2. Lista de curvas de nivel (la lista generada en la sección anterior).

En esta sección se describirán los módulos necesarios para utilizar el ordenamiento de los datos vectoriales utilizando PEB.

Este método se divide en dos partes:

1. La creación del árbol.
2. Llenado del árbol.

3.3.1. Creación del árbol

El árbol se crea siguiendo las reglas de un árbol binario, en cada uno de los nodos se almacena una división del plano siendo las hojas las regiones que contienen los datos. En la figura 3.10 se muestra un ejemplo del particionamiento de un plano en tres regiones, en primera instancia tenemos el plano a y su respectivo árbol PEB. Posteriormente el plano es particionado sobre el eje x formandose las regiones b y c en este momento se agregan estas dos regiones al árbol PEB. Finalmente se realiza una partición más sobre una de las regiones lo cual da como resultado que esta región se divida en dos, en ese momento se agregan dos regiones más al árbol PEB.

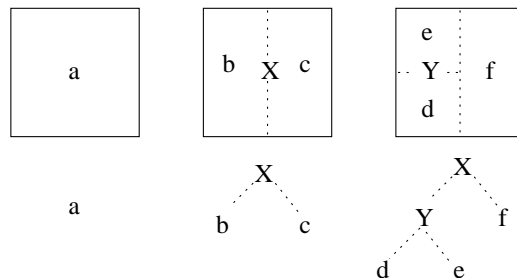


Figura 3.10: La figura muestra el como se realiza el particionamiento usando un árbol **PEB**.

3.3.2. Llenado del árbol

El llenado del árbol PEB consiste en asignar los segmentos de las curvas de nivel a la hoja que representa a la región en la cual el segmento se encuentra contenido.

Para particionar los datos se busca que regiones contienen a cada curva de nivel, posteriormente se le asigna a cada hoja los segmentos que le corresponden.

3.3.3. Características

Este método fue tomado de [8], en este lugar se pueden ver más detalles.

1. Los nodos que tendrá el árbol son $2N - 1$ donde N es el número de zonas en que se ha dividido el plano (x, y) .
2. La búsqueda de las hojas a dibujar son en el mejor de los casos logarítmicas y en el peor de los casos lineales, dependiendo del estado del árbol.
3. El algoritmo puede ser implantado a n-dimensiones.
4. La evaluación de orden es complicada.

La evaluación de orden es descrita en la sección 3.2.4, aún cuando los criterios son los mismos que se emplearon en el método de la rejilla, para los PEB se requiere evaluar explícitamente estas condiciones, mientras que en el otro método el orden se obtiene al construir la rejilla.

3.4. Triangulación

La malla de polígonos que representa el modelo a visualizar no es otra que el diagrama de Delaunay, el cual se construye con el conjunto de puntos que representan los vértices de las curvas de nivel y los vértices de las secciones del esqueleto seleccionadas.

Sin embargo los datos se encuentran seccionados, como hemos visto en este capítulo los vértices se encuentran contenidos en distintas regiones. Al crear los diagramas de Delaunay sobre los vértices contenidos dentro de cada una de las regiones ya no es necesario ordenar los polígonos, pero las uniones de las regiones deben cumplir las siguientes características:

1. Las uniones de las secciones no deben contener huecos.
2. Las uniones entre regiones deben ser suaves, esto es que el cambio de una región a otra sea lo menos visible posible.

Para que las uniones de las regiones cumplan estas características se agregaron algunos puntos a las regiones. La figura 3.11 muestra cuando y donde son introducidos estos puntos.

Solo hay dos casos en los cuales se agregan puntos sobre alguna frontera, estos son:

1. Se agregan puntos en la posición de los vértices de las regiones en la figura 3.11 son los puntos e , g y h .
2. También se agregan puntos en donde haya intersección, intersección entre las *curvas de nivel* y las fronteras de las regiones.

En la figura 3.11 el plano ha sido dividido en las regiones A , B , C y D , los puntos e , f , g y h agregados y considerados en la construcción del diagrama de Delaunay.

Siguiendo con la figura 3.11 e es incluido en las regiones A , B , C y D , f es incluido en las regiones A y C , g es incluido dentro de las regiones A y B , y h es incluido dentro de la región B .

Los puntos sobre los vértices son fáciles de ubicar, ya que estos son los vértices de la rejilla que se construyo en el capítulo 3; sin embargo éstos carecen de altura, lo cual no afecta a la construcción del diagrama de Delaunay; pero es importante asignarle altura a cada uno de estos puntos ya que de no hacerlo estos vértices crearían hundimientos.

La altura de estos vértices es el promedio de las alturas de aquellos vértices con los cuales se construye una arista de Delaunay, tomando en cuenta los diagramas de Delaunay de cada una de las regiones en las que ha sido incluido el vértice al que se le está calculando su altura.

La altura que se le asignan a los puntos de las intersecciones entre una curva de nivel y una frontera de las regiones, es la misma altura de la curva de nivel.

Como podemos observar en la figura 3.11 los puntos agregados a las regiones, se agregan sobre la cubierta del conjunto de puntos de cada región. Al agregar los puntos sobre la cubierta se garantiza que si los puntos se encuentran contiguos, con respecto a la cubierta, formarán una arista, dicha arista es la frontera entre dos regiones.

La figura 3.12 muestra el resultado de realizar la triangulación sin introducir puntos sobre las fronteras de la región, las líneas negras representan las fronteras. Al ser visibles las fronteras quiere decir que no hay aristas sobre estas fronteras, lo que significa que hay huecos entre las regiones.

La figura 3.13 muestra el resultado de realizar la triangulación introduciendo puntos sobre las fronteras de la región, podemos observar que ahora las líneas de las fronteras

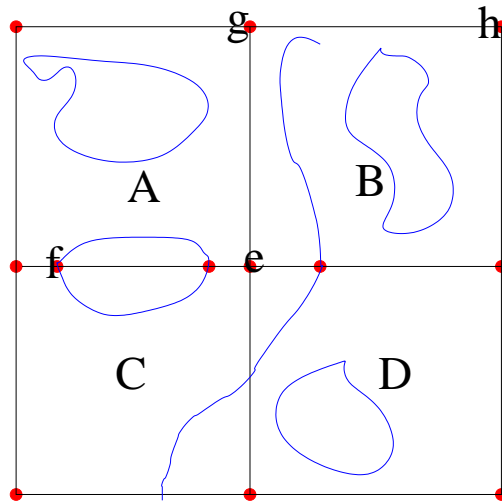


Figura 3.11: El plano se ha dividido en las regiones A , B , C y D , mientras que e , f , g y h son puntos que se han agregado a las fronteras entre las regiones.

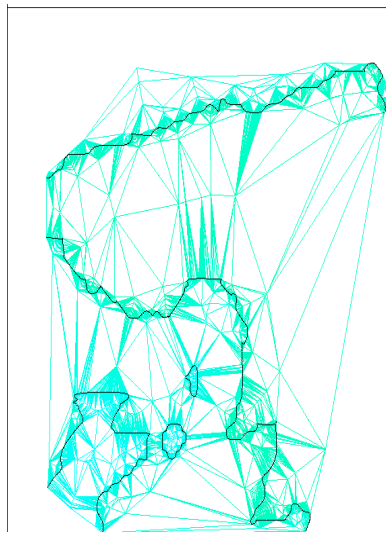


Figura 3.12: Triangulación sin puntos sobre la frontera

han sido cubiertas por aristas, cubriendo los huecos entre las regiones.

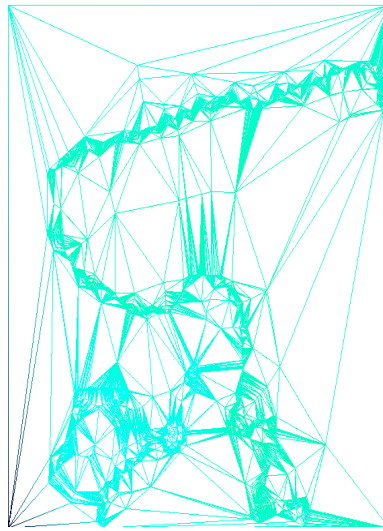


Figura 3.13: Triangulación con puntos sobre la frontera

Capítulo 4

Resultados

Como resultado de este trabajo se realizó una interfaz para visualizar la reconstrucción del mapa de México. Se usó en las pruebas el mapa a escala 1:250,000 E1403 del INEGI.

En este capítulo se describe el diseño de la interfaz gráfica.

4.1. Diseño de la interfaz

La interfaz fue implementada con el uso de las bibliotecas OpenGL y QT.

La interfaz consta de cuatro partes:

1. El área de visualización es un *Widget* en el que se dibuja el modelo tridimensional.
2. El área de controles consta de varios elementos que permiten realizar las distintas transformaciones con las que se manipula el modelo tridimensional.
3. El área de referencia muestra la latitud y longitud del terreno que se visualiza.
4. Una brújula o *rosa de los vientos*.

4.1.1. Vista de la interfaz

La interfaz en su totalidad está orientada a cubrir dos objetivos: visualizar el modelo tridimensional construido mediante el método descrito en el capítulo 2 y la manipulación de éste.

4.2. Área de visualización

El área de visualización consta de un *widget* en el que se dibuja la proyección paralela del modelo tridimensional.

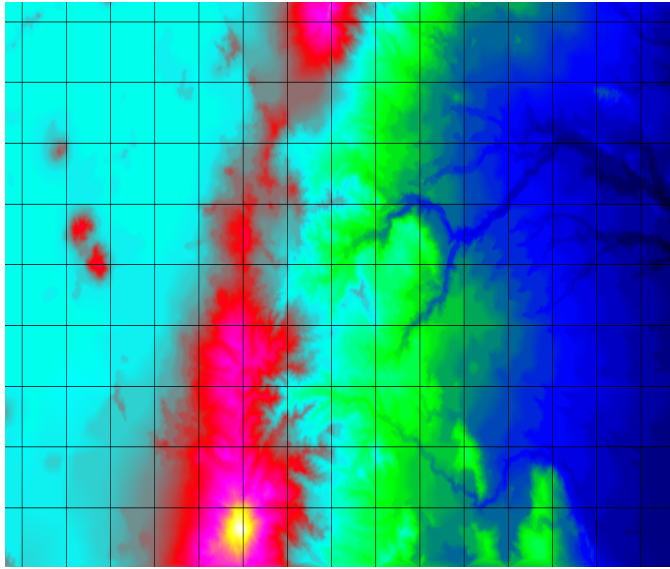


Figura 4.1: La figura muestra la visualización.

El área de dibujo tiene dos opciones, dibujar polígonos como se muestra en la figura 4.1 y dibujar aristas.

4.3. Área de controles

El área de controles consta de una serie de elementos que permiten desplazarse sobre el área de visualización, en otras palabras, permite que el usuario se mueva dentro del mapa, las funciones que tiene son: translaciones sobre el eje x sobre el eje y , rotaciones sobre los ejes x y un eje z' , también se permite realizar escalamientos. Todas estas transformaciones tienen una serie de restricciones.

Las transformaciones se encuentran restringidas, con el objetivo de evitar que el usuario ocasione visualizaciones erróneas del terreno como resultado de aplicar estas transformaciones, las restricciones se detallarán adelante.

4.3.1. Restricción de las translaciones

Las translaciones serán posibles, siempre y cuando haya datos en el sentido en el que se deseen hacer, una vez que se ha llegado a un extremo del mapa el botón que corresponde a dicha translación debe ser desactivado y coloreado con otro color.

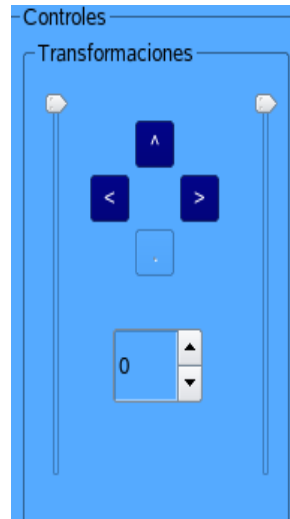


Figura 4.2: La figura muestra el área de control de la interfaz gráfica.

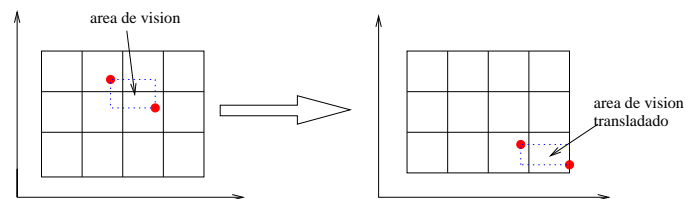


Figura 4.3: La figura muestra el área de visión, después de realizar algunas translaciones muestra que el área de visión ha llegado a uno de los extremos.

Dicho en otras palabras la translación es válida siempre y cuando haya datos en el sentido que se realice, de otra manera no la translación no se lleva a cabo ya que no tiene sentido visualizar zonas en las que no hay datos.

4.3.2. Restricciones de la rotación sobre el eje x

Esta rotación tiene sentido ya que los puntos son tridimensionales, lo cual nos permite que el usuario no sólo observe la vista superior del terreno sino que tenga más vistas.

La limitación esta rotación está ligada con la coherencia de las vistas, la figura 4.4 muestra un ejemplo de esta transformación.

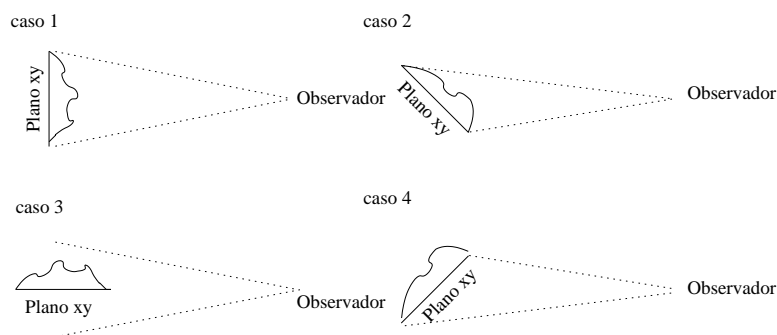


Figura 4.4: La figura muestra cuatro casos: la vista superior, una rotación de 45° sobre el eje x , una rotación de 90° sobre el eje x y una rotación de 135° sobre el eje x la cual no es permitida.

Como se puede observar en la figura 4.4, al rotar el modelo más 90° el observador empieza a ver los datos por debajo de las montañas; lo cual no tiene ningún sentido.

Una vez se ha descrito el caso en el que se encuentra esta rotación vemos que el ángulo se encuentra en el intervalo de 0° a 90° . En la interfaz el elemento encargado de representar la apertura del ángulo de rotación es un *qslicer*, el cual permite limitar los valores que puede tomar dicho ángulo.

4.3.3. Restricciones de la rotación sobre el eje z'

Antes de describir esta rotación, debemos saber a que le llamamos eje z' .

El eje z' es un vector perpendicular al plano de visión que se ha obtenido tras rotarlo sobre el eje x , además dicho vector debe ser positivo y debe pasar por el centro de dicho plano.

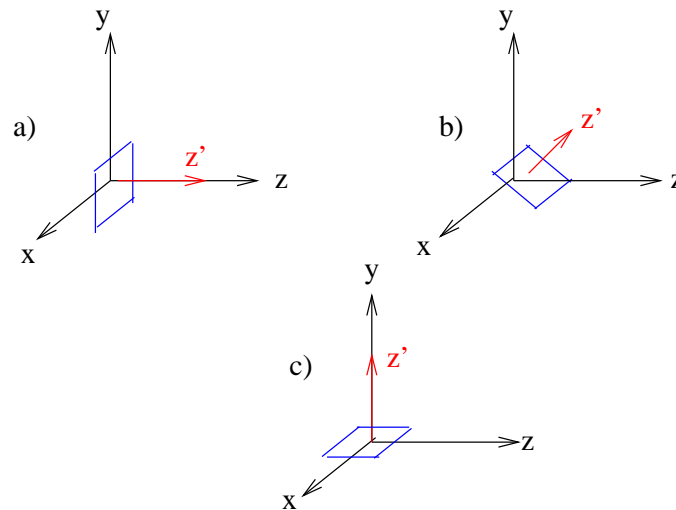


Figura 4.5: En a) el plano de visión se encuentra paralelo al plano x, y , $z = z'$ en b) el plano ha sido rotado sobre el eje x , dando como resultado que z' cambie en c) el plano de visión se encuentra paralelo al plano x, z y $z' = y$.

Como se puede observar en la Fig. 4.5, el ángulo del eje z' es el ángulo en que se ha rotado el plano de visión con respecto al eje x .

La primera alternativa es realizar dicha rotación una vez que se ha hecho la rotación del eje x ; sin embargo computacionalmente esta forma no es la más adecuada, lo más adecuado es realizar la rotación sobre el eje z y luego realizar la rotación sobre el eje x .

La rotación sobre el eje z' es cíclica se encuentra definida entre 0° y 360° .

4.3.4. Descripción del escalamiento

Al realizar el escalamiento se toman en cuenta dos aspectos, éstos son:

- Escalar el modelo que se está visualizando, esto es, aplicar la transformación a los datos.
- Escalar el área de visión con el objetivo de seleccionar aquella o aquellas regiones que contengan al área de visión.

La relación que existe entre los datos y el área de visión es la siguiente:

1. Si se hace un escalamiento positivo (*Zoom In*), a los datos se les aplica tal cual la transformación; mientras que al área de visión lleva la transformación inversa con el objetivo de que ésta se reduzca abarcando menos datos.
2. Si se hace un escalamiento negativo (*Zoom Out*), a los datos se les aplica tal cual la transformación, mientras que el área de visión lleva la transformación inversa.

El escalamiento también se encuentra limitado por los datos, ya que no tiene caso realizar una escala menor a uno ya que esto traería como resultado que algunas secciones de la pantalla se encuentren sin datos a visualizar.

Sin embargo esto no es lo único que se debe cuidar, al realizar el escalamiento se debe cuidar que los vértices del área de visión queden contenidos dentro del área de datos; de no ser así se realiza una serie de translaciones con el objetivo de llevar a los vértices dentro del área de datos.

La figura 4.6 muestra un área de datos en la cual el área de visión está delimitada por las líneas punteadas; mientras que los puntos son los vértices que sirven para marcar el inicio y el fin del barrido de regiones a visualizar.

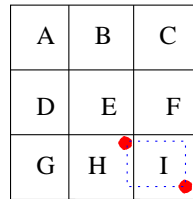


Figura 4.6: La figura muestra, el área de visión contenida en el área de datos.

La figura 4.7 muestra el caso en que debido al escalamiento el área de visión no se está totalmente contenida dentro del área de datos.

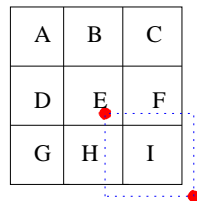


Figura 4.7: La figura muestra, el área de visión que esta vez no se encuentra contenida en el área de datos.

Tras hacer una serie de translaciones (a lo sumo dos), el área de visión se encuentra de nuevo en el área de datos, esto puede verse en la figura 4.8.

Debido a que se debe acotar los valores del escalamiento, el elemento que se presta para representar los valores del escalamiento es un *qslicer*.

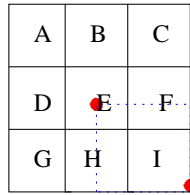


Figura 4.8: La figura muestra, el área de visión que tras algunas translaciones queda contenida dentro del área de datos.

4.4. Área de referencia

El área de referencia contiene una serie de elementos mediante los cuales se relaciona el mundo real al modelo que visualizamos, estos elementos son la latitud y la longitud.

Latitud

Longitud

Escala

Curvas de nivel

Figura 4.9: La figura muestra, el área de referencia en ella se encuentran elementos que representan la latitud y la longitud, la escala a la cual se encuentran los datos.

Latitud y longitud se representan con el formato de grados, minutos y segundos; por lo que hay que convertir los datos que se tienen de metros a grados, minutos y segundos. Para realizar la conversión se utilizaron las ecuaciones de Coticchia-Surace [1], las cuales son deducidas mediante el transversal cilíndrico terrestre de Mercator, tomando como referencia un elipsoide de revolución y dividiendo la Tierra en 60 husos iguales de 6 grados cada uno [13].

El origen de latitudes es el ecuador y el de longitudes es el meridiano de Greenwich.

Para realizar el cálculo es necesario conocer sobre que elipsoide se encuentran las coordenadas que se desean transformar, las ecuaciones requieren como datos el semieje mayor y el semieje menor del elipsoide.

El desarrollo del cálculo se baso en la referencia [13] y tiene los siguientes pasos:

Semieje mayor = a

Semieje menor = b

Calculamos la excentricidad, la segunda excentricidad, el radio polar de curvatura y el aplanamiento:

Segunda Excentricidad:

$$e' = \frac{\sqrt{a^2 - b^2}}{b} \quad (4.1)$$

Se calcula el radio polar de curvatura y el aplanamiento:

Radio polar de la curvatura:

$$c = \frac{a^2}{c} \quad (4.2)$$

Aplaneamiento:

$$\alpha = \frac{a - b}{a} \quad (4.3)$$

Ahora debemos ubicar las coordenadas dependiendo su posición con respecto al origen:

Primero se translada la ordenada x mediante la operación

$$x = x - 500000 \quad (4.4)$$

Para la ordenada y se usa el siguiente criterio:

1. Si y se encuentra al norte del ecuador, y no es modificada.
2. Si y se encuentra al sur del ecuador $y = y - 10000000$.

Debemos conocer el huso UTM (o Zona UTM) al que pertenecen las coordenadas a convertir. El siguiente paso es obtener el meridiano central del huso en el que caen las coordenadas geodésicas sobre las que operamos. Con la siguiente operación:

$$\lambda_0 = huso * 6 - 183 \quad (4.5)$$

A continuación se calculan todos los parametros de las ecuaciones:

$$\varphi = \frac{y}{6336197.724 * 0.9996} \quad (4.6)$$

$$v = \frac{c}{(1 + e'^2 \cos^2 \varphi)^{\frac{1}{2}}} \quad (4.7)$$

$$a = \frac{x}{v} \quad (4.8)$$

$$A_1 = \text{sen}(2\varphi) \quad (4.9)$$

$$A_2 = A_1 \cos^2 \varphi \quad (4.10)$$

$$J_2 = \varphi + \frac{A_1}{2} \quad (4.11)$$

$$J_4 = \frac{2J_2 + A_2}{4} \quad (4.12)$$

$$J_6 = \frac{5J_4 + A_2 \cos^2 \varphi}{3} \quad (4.13)$$

$$\alpha = \frac{3}{4}e'^2 \quad (4.14)$$

$$\beta = \frac{5}{3}\alpha^2 \quad (4.15)$$

$$\gamma = \frac{35}{27}\alpha^3 \quad (4.16)$$

$$B_\Phi = 0.9996 * c * (\varphi - \alpha J_2 + \beta J_4 - \gamma J_6) \quad (4.17)$$

$$b = \frac{y - B_\Phi}{v} \quad (4.18)$$

$$\zeta = \frac{e'^2 a^2 \cos^2 \varphi}{2} \cos^2(\varphi) \quad (4.19)$$

$$\xi = a[1 - \frac{\zeta}{3}] \quad (4.20)$$

$$\eta = b(1 - \zeta) + \varphi \quad (4.21)$$

$$\text{senh } \varepsilon = \frac{e^\varepsilon - e^{-\varepsilon}}{2} \quad (4.22)$$

$$\Delta\lambda = \arctan\left(\frac{\text{senh } \varepsilon}{\cos \eta}\right) \quad (4.23)$$

$$\tau = \arctan(\cos(\Delta\lambda) \tan(\eta)) \quad (4.24)$$

$$\lambda = 180 \frac{\Delta\lambda}{\pi} + \lambda_0 \quad (4.25)$$

$$\varphi' = \varphi + [1 + e'^2 \cos^2 \varphi - \frac{3}{2}e'^2 \text{sen } \varphi \cos \varphi (\tau - \varphi)](\tau - \varphi) \quad (4.26)$$

$$\varphi'' = \frac{180\varphi'}{\pi} \quad (4.27)$$

La siguiente tabla muestra cómo calcular la longitud y la latitud empleando los valores λ y φ'' calculados arriba.

longitud grados	$\lfloor \lambda \rfloor$
longitud minutos	$\lfloor (\lambda - \text{grados}) * 60 \rfloor$
longitud segundos	$\lfloor ((\lambda - \text{grados}) * 60 - \text{minutos}) * 60 \rfloor$
latitud grados	$\lfloor \varphi'' \rfloor$
latitud minutos	$\lfloor (\varphi'' - \text{grados}) * 60 \rfloor$
latitud segundos	$\lfloor ((\varphi'' - \text{grados}) * 60 - \text{minutos}) * 60 \rfloor$

4.5. Asignación de colores a vértices

Se cuenta con un conjunto de puntos tridimensionales los cuales pertenecen a las curvas de nivel, se desea asignar un color a los puntos dependiendo de la altura que tenga el punto en cuestión.

Por otro el modelo de color que usa *OpenGL* es el modelo de RGB, el cual emplea como colores primarios los colores rojo, verde y azul, usando las combinaciones de estos para formar los demás colores. Cada uno de los colores pueden tomar 256 intensidades de colores, estas intensidades pueden ser normalizadas al intervalo de 0 a 1, combinando dichas intensidades podemos formar 2^{24} colores.

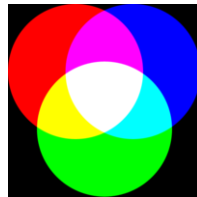


Figura 4.10: La figura muestra las combinaciones de los colores partiendo de los colores primarios, el color negro es la ausencia de color, es decir todos los canales tienen valor cero.

Del conjunto de puntos es posible establecer una altura máxima y una altura mínima con el fin de determinar el intervalo en el que se encuentran los puntos del conjunto de datos.

Una vez que se ha determinado este intervalo, éste es dividido en niveles dependiendo del número de colores con los que se cuenta.

Sea n el número de colores entonces se tienen $n - 1$ niveles, esto lo podemos ver en la figura 4.11, la cual muestra un intervalo de alturas dividido en $n - 1$ niveles, ya que se cuenta con n colores.

Donde la longitud de cada nivel a la que llamamos l la podemos calcular mediante

$$l = \frac{Z_{max} - Z_{min}}{n - 1} \quad (4.28)$$

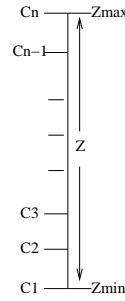


Figura 4.11: La altura se ha dividido en niveles, a cada nivel se le asigna un color base.

donde Z_{min} es la altura mínima, Z_{max} es la altura máxima y n es el número de colores.

Entonces si se tiene un vértice con la altura z' , el primer paso es determinar en que intervalo se encuentra, a este intervalo se le denota por i , esto se obtiene mediante la Ec. siguiente:

$$i = \lfloor \frac{z' - z_{min}}{l} \rfloor \quad (4.29)$$

Cada nivel se encuentra entre dos colores, por lo que cuando una altura cae dentro de un intervalo, el color que le corresponde a dicha altura debe interpolarse entre los colores en los que se encuentra, para ello usamos la Ec. siguiente:

$$C_{z'} = C_i * (1 - t) + C_{i+1} * t \quad (4.30)$$

donde t se encuentra definida en el intervalo $0 \leq t \leq 1$, dependiendo de la posición de la altura dentro del nivel en que se encuentra, para encontrar el valor de t evaluamos la Ec. siguiente:

$$t = \frac{z' - z_{min} - l * i}{l} \quad (4.31)$$

donde z' es la altura a la que se le desea asignar un color, z_{min} es la altura mínima, i es el nivel en que se encuentra la altura y l es la longitud de los niveles.

Los colores son el resultado de las combinaciones de intensidades de los colores rojo, verde y azul, por lo que la implementación de la ecuación es evaluada con respecto a cada una de las intensidades.

Capítulo 5

Conclusiones y trabajo a futuro

5.1. Conclusiones

En el presente trabajo de tesis se ha tratado el problema de visualizar reconstrucciones tridimensionales partiendo de datos vectoriales de los mapas.

Este trabajo se dividió en dos partes:

1. La construcción del modelo tridimensional eliminando los *triángulos planos*.
2. El seccionamiento de datos, para permitir que sólo se seleccione aquellas secciones de datos dentro del área de visión.

El primer punto se resolvió utilizando la siguiente metodología:

- Calcular los *splines* con los datos vectoriales de las curvas de nivel.
- Muestrear cada una de las curvas a una razón constante igual a la mitad de la distancia mínima entre dos curvas distintas.
- Calcular el esqueleto.
- Seleccionar aquellas partes del esqueleto que eliminan *triángulos planos*.
- Calcular las alturas de los vértices del esqueleto. Para esto se toma el contexto en cual se encuentran cada uno de éstos.
- Triangular todo el conjunto de pares de coordenadas (x, y) de los puntos.

En el trabajo de Dakowicz y Gold [7] se discute cómo calcular la altura a las cimas y las secciones reentrantes; sin embargo este trabajo no distingue si las secciones reentrantes se encuentran dentro o fuera de la curva, simplemente generaliza estos casos.

Para realizar el cálculo de las secciones reentrantes se toma en cuenta si la sección del esqueleto se encuentra dentro o fuera de la curva de nivel que tienen cerca, de esta manera las alturas de los vértices del esqueleto tienden a aumentar o disminuir (dependiendo cual sea el caso) conforme se aleja de la curva de nivel a la que está asociada.

La segunda parte se dedicó al seccionamiento de datos por regiones. Para esta parte se analizaron dos técnicas empleadas para el seccionamiento de datos, eligiéndose la más adecuada para el problema que se planteó en este trabajo de tesis. El método empleado consiste en sobreponer una rejilla sobre el mapa con la que se trabaja, cada uno de los elementos de la rejilla representa una región, de esta manera los datos se dividen en regiones. La construcción de la rejilla es simple, ya que sólo se debe dar las dimensiones de las regiones y mediante dos operaciones la rejilla es construida, faltando el llenado de ésta. El llenado de las regiones es lineal esto es tiene una complejidad n , con respecto a los vértices de las curvas de nivel, ya que sólo verifican la intersección de las curvas de nivel con las regiones. La selección de las regiones que se encuentran dentro del área de visión es constante.

El modelo tridimensional se calcula de manera independiente en cada región lo cual fue posible al introducir puntos sobre los bordes de las regiones.

Finalmente, se construyó una interfaz gráfica para visualizar la reconstrucción tridimensional de la carta E1403 del INEGI.

Actualmente el conjunto de datos se encuentra cargado totalmente en memoria y sólo se dividen de manera lógica.

5.2. Trabajo futuro

En la siguiente lista se presentan una serie de características y posibles mejoras para nuestro sistema:

1. Obtener la razón de muestreo en cada una de las regiones de manera independiente, esto permitirá que las regiones se encuentren muestreadas de forma independiente y las regiones no estarán sobre muestreadas.
2. Almacenar los datos en disco, ya que con un conjunto de datos mayor es necesario.
3. Encontrar un mecanismo que permita conservar en memoria sólo aquellas regiones que se encuentren en el área de visión.
4. Actualmente los datos se encuentran sobre un plano; por lo que al proyectarlos sobre la superficie de una esfera mejoraría la forma en que se visualiza la información.

5. Proyectar los datos sobre una esfera implica desarrollar un mecanismo de selección que tome en cuenta la nueva superficie sobre la cual se proyectan los datos, el trabajo [5] se habla al respecto.

Podemos poner un ejemplo, supongamos que se desea visualizar un área que abarque más de una carta topográfica, como puede ser un país, también se dará por sentado que se cuentan con los datos vectoriales. Los pasos a seguir son los siguientes:

1. Establecer sobre qué sistema de referencia se está trabajando, esto con el objetivo de establecer una relación entre el modelo visualizado y la porción de tierra que representan.
2. Definir las dimensiones del área que se desea visualizar.
3. Establecer las dimensiones de la rejilla, buscando que haya intersección entre algunas de las fronteras entre regiones y las divisiones de las cartas topográficas.
4. Dividir los datos en las regiones, esto es, asignarle a cada región los puntos que quedan contenidos en cada una de las regiones.
5. Establecer la razón de muestreo en cada región, tomando únicamente las curvas de nivel contenidas en cada región.
6. Muestrear cada una de las regiones de manera independiente.
7. Calcular el esqueleto en cada una de las secciones.
8. Agregar los puntos de unión entre secciones, como se mostró en el capítulo 3.
9. Establecer un mecanismo que permita tener en memoria sólo aquellas regiones que se encuentren en el área de visión, estableciendo un *buffer* para ello.
10. Proyectar los datos sobre una esfera.
11. Establecer un mecanismo para recortar las secciones que no se dibujan, tomando la superficie sobre la cual se proyectan los polígonos.

Los pasos dos y tres son enfocados a la creación de la rejilla, al crear la rejilla es posible numerar las regiones.

El paso cuatro divide los datos, con la numeración relacionando un sólo archivo a cada una de las regiones.

El objetivo de los pasos cinco, seis y siete es la creación del modelo tridimensional.

El punto ocho se enfoca a la unión de las regiones, si buscamos que las fronteras de las regiones se intersecten con los límites de las cartas topográficas, no se necesitarán

consideraciones adicionales para la unión entre cartas topográficas.

El punto nueve es importante implantarlo, debido a que resultaría imposible tener en memoria el conjunto completo de datos. Para ésto se sugiere que los datos se almacenen en disco empleando la numeración de la rejilla para organizarlos, esto se esquematiza en la Fig. 5.1.

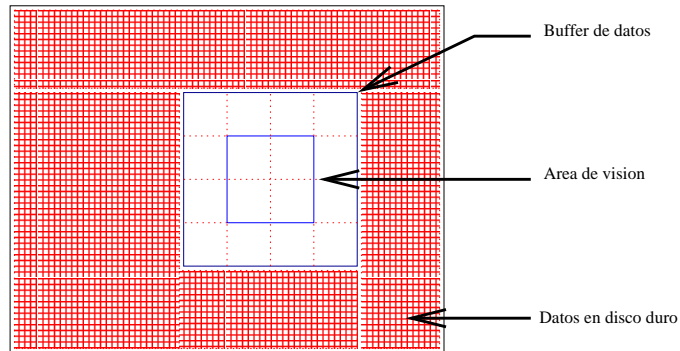


Figura 5.1: La figura muestra un área de datos, un área de *buffer* y un área de visión, estando sólo en memoria las últimas dos.

Por ultimo es necesario establecer los mecanismos para hacer el manejo del *buffer*, quizá una arquitectura de hilos sería la indicada, ya que la carga de datos no tiene costo computacional considerable. La mayor parte del trabajo es el acceso al disco duro; mientras que la visualización sí requiere cálculos por lo que quizá con un par de hilos podría implantarse, un hilo dedicado a la carga de datos y otro el otro dedicado a la visualización.

Apéndice A

Descripción de programas

En este apéndice se muestra una descripción de los programas empleados para la construcción del modelo tridimensional, estos son:

1. Un programa que muestrea los datos.
2. Un programa que calcula las aristas del esqueleto.
3. Un programa que calcula los puntos las alturas de los vértices del esqueleto.
4. Un programa que muestrea el esqueleto.
5. Un programa que visualiza el modelo tridimensional.

A.1. Programa de muestreo

El programa de muestreo de datos se divide en dos partes, primero calcula los polinomios mediante los splines, y luego los muestrea.

Este programa requiere como entrada un archivo con el formato de la tabla A.3.

A la salida de este programa se obtiene un archivo DXF.

A.2. Cálculo de las aristas del esqueleto

Este programa calcula las aristas del esqueleto que se rodean de puntos con la misma altura, para ello se requieren la listas de pares de aristas duales entre los diagramas de Voronoi y Delaunay y las curvas de nivel. Empleando el script de qvoronoi se obtiene el archivo con el formato de la tabla A.4.

El formato del archivo de las curvas de nivel es el de la tabla A.3.

A la salida de este programa se obtiene un archivo que lista los vértices pertenecientes a las aristas del esqueleto, después de la lista de vértices, lista las aristas, el formato del archivo se muestra en la tabla A.5.

A.3. Cálculo de alturas

Este programa calcula las alturas de los vértices del esqueleto, el cálculo se realiza tomando los casos vistos en el capítulo 2, para ello se requiere:

1. El diagrama de Delaunay que se calcula con un script de qhull, con los puntos del esqueleto y los puntos de las curvas de nivel el formato es el de la tabla A.2.
2. Un grafo que representa el esqueleto, este grafo se calcula con un script del trabajo de las tesis de Abigail Martínez [17], cuya entrada es el archivo de salida del programa de la sección A.1 la tabla A.1 muestra el formato.
3. Las curvas de nivel el, archivo tiene el formato de la tabla A.3.

A la salida este programa se ofrece un archivo con el mismo formato del grafo de entrada, solo se le anexa la altura a cada grafo, véase la tabla A.1.

A.4. Muestreo del esqueleto

Este programa recibe como entrada un grafo que representa al esqueleto, el formato del archivo de entrada se muestra en la tabla A.1.

A.5. Visualización del modelo

Esta aplicación es la interfaz que permite interactuar al usuario con el modelo tridimensional construido. Esta aplicación recibe como entrada un archivo DXF el cual tiene la información vectorial de las curvas de nivel. este archivo es el obtenido con el programa de muestreo, además requiere el esqueleto remuestreado, que también es obtenido con el programa de muestreo, pero teniendo como entrada el esqueleto.

n vértices del esqueleto
v_0
\cdot
\cdot
v_{n-1}
m grafos
l ramas del grafo uno
$s_1 r_1 p_1 q_1$
\cdot
\cdot
$s_{l-1} r_{l-1} p_{l-1} q_{l-1}$
\cdot
\cdot
l' ramas del grafo m
$s_1 r_1 p_1 q_1$
\cdot
\cdot
$s_{l-1} r_{l-1} p_{l-1} q_{l-1}$

Tabla A.1: Formato del archivo que representa al esqueleto.

n número de poligonos
se listan los vértices de los poligonos

Tabla A.2: Formato del archivo que representa al diagrama de *Delaunay*

n Número de curvas de nivel.
k Número de puntos curva de la uno.
h Altura de la curva uno.
$ \begin{array}{c} Px_0 \ Py_0 \\ Px_1 \ Py_1 \\ \cdot \\ \cdot \\ \cdot \\ Px_{k-1} \ Py_{k-1} \end{array} $
k' Número de puntos de la curva dos.
h' Altura de la curva dos.
$ \begin{array}{c} Px_0 \ Py_0 \\ Px_1 \ Py_1 \\ \cdot \\ \cdot \\ \cdot \\ Px_{k'-1} \ Py_{k'-1} \\ \cdot \\ \cdot \\ \cdot \end{array} $
k^n Número de puntos de la n-ésima curva.
h^n Altura de la curva n-ésima curva.
$ \begin{array}{c} Px_0 \ Py_0 \\ Px_1 \ Py_1 \\ \cdot \\ \cdot \\ \cdot \\ Px_{k^n-1} \ Py_{k^n-1} \end{array} $

Tabla A.3: Formato de entrada para los datos el programa de muestreo

n número de vértices del Voronoi
$Px_0 Py_0$ \cdot \cdot \cdot $Px_{n-1} Py_{n-1}$
k número de pares de aristas duales
$a_0 s_0 p_0 q_0$ $a_1 s_1 p_0 q_0$ \cdot \cdot \cdot $a_{k-1} s_{k-1} p_{k-1} q_{k-1}$

Tabla A.4: Formato del archivo de listas duales s_i y a_i son los vértices que forman la i -ésima arista de la corteza, mientras que p_i y q_i son los vértices diagrama de Voronoi.

n número de vértices del esqueleto
$Px_0 Py_0$ $Px_1 Py_1$ \cdot \cdot \cdot $Px_{n-1} Py_{n-1}$
k número de aristas del esqueleto
$a_0 s_0$ $a_1 s_1$ \cdot \cdot \cdot $a_{k-1} s_{k-1}$

Tabla A.5: El formato de salida del programa s_i y a_i son los vértices que forman la i -ésima arista.

Apéndice B

Experimento

En este apéndice se muestra un experimento, el cual consiste que a un conjunto de puntos se le calcula las secciones del esqueleto útiles para la eliminación de los triángulos planos. Mostraremos algunas imágenes las cuales corresponden a los siguientes pasos:

1. Cálculo de los Splines.
2. Cálculo de la distancia mínima.
3. Remuestreo sobre los Splines a $0.42 \cdot \frac{DistMin}{2}$.
4. Cálculo del esqueleto.
5. Selección de las secciones útiles para la eliminación de *triángulos planos*.
6. Remuestreo del esqueleto.

Partimos de un conjunto de puntos a los cuales se le calculan los *splines*, la figura B.1 muestra el en (a) conjunto de puntos iniciales mientras que (b) muestra la aproximación a la curva usando *splines*.

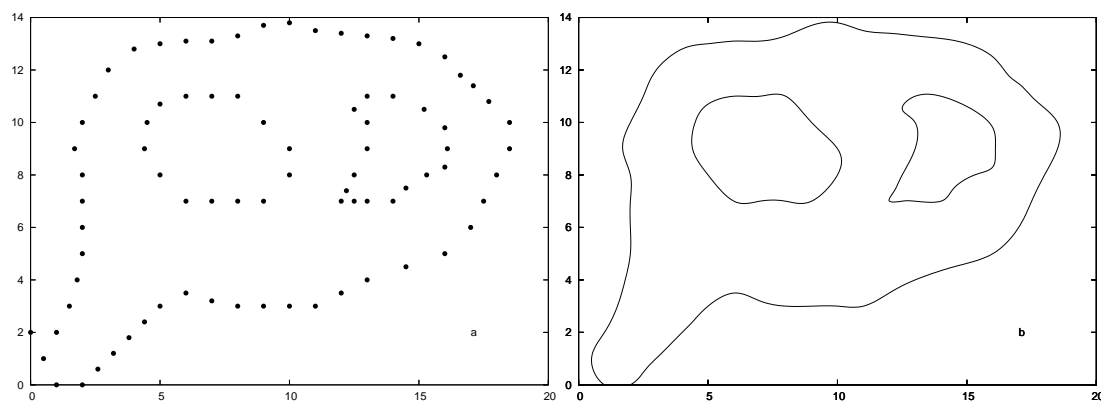


Figura B.1: (a) los datos iniciales (b) los *splines* que corresponden al conjunto de datos iniciales.

Tras leer los datos calculamos la distancia mínima para este conjunto la distancia mínima entre curvas es 1. Entonces los datos los remuestreamos a razón de 0.42 la figura B.2 muestra los splines muestrados a esta razón.

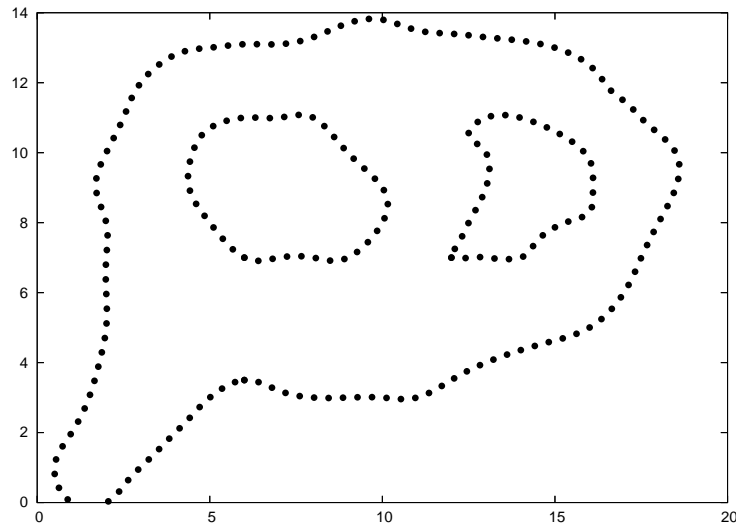


Figura B.2: El conjunto de puntos muestrados a una razón de 0.42.

Una vez que sea ha remuestreado los datos, se calcula el esqueleto, la figura B.3 muestra el esqueleto correspondiente al conjunto de datos remuestreados.

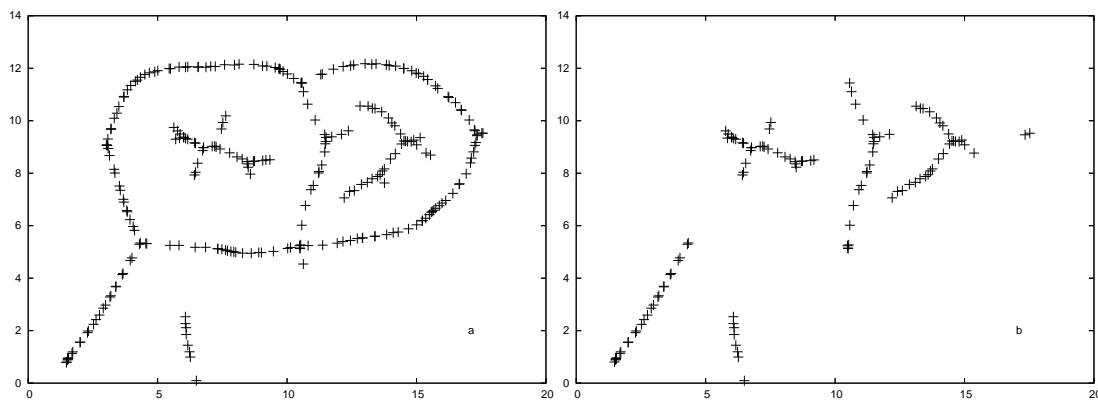


Figura B.3: (a) Esqueleto de los datos resmuestrados (b) Secciones útiles para eliminar *triángulos planos*.

Sin embargo no todas las secciones del esqueleto son necesarias para eliminar los *triángulos planos*, aquellas secciones del esqueleto que no sirvan para este fin deben ser

eliminadas, la figura B.3 muestra aquellas secciones del esqueleto útiles para la eliminación de los *triángulos planos*.

Se remuestrea el esqueleto para reducir los puntos de las secciones seleccionadas, la figura B.4 muestra el remuestreo de estas secciones.

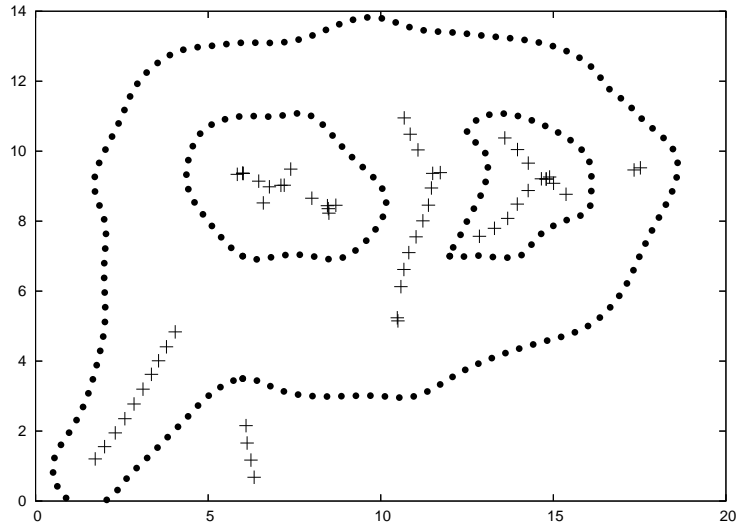


Figura B.4: Secciones seleccionadas remuestreadas, y *curvas de nivel*.

Se triangula empleado los puntos de las curvas de nivel, y los puntos seleccionados del esqueleto ésto se ve en la figura B.5.

Finalmente se muestra en la Fig. B.6 la reconstrucción correspondiente a las tres curvas de nivel.

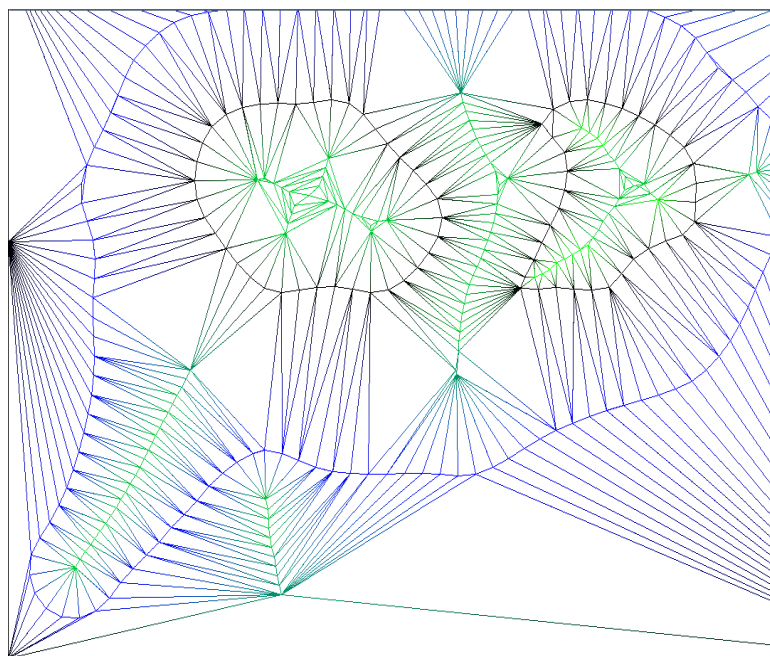


Figura B.5: Triangulación de los puntos de las curvas de nivel y los puntos del esqueleto.

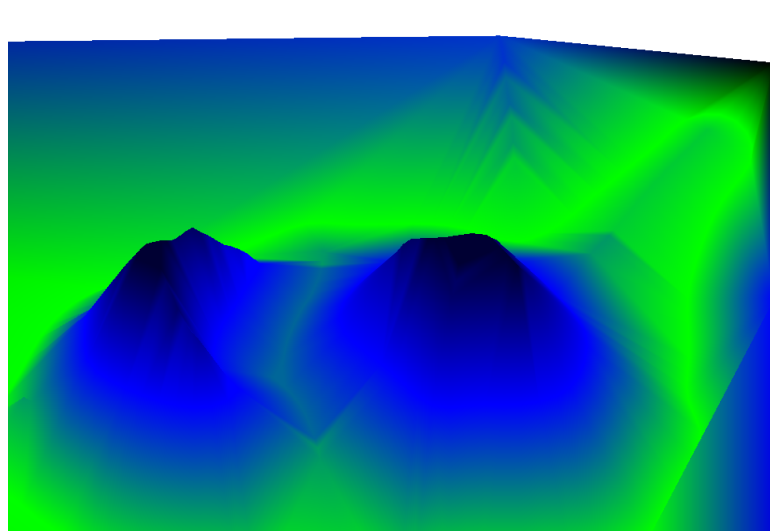


Figura B.6: La reconstrucción correspondiente al ejemplo de este apéndice

Bibliografía

- [1] Coticchia A and Surace L. La trasformazione delle coordinate dal sistema utm. *Bolletino di Geodesia e Science Affini*, 4, 1950.
- [2] Nina Amenta, Marshall Bern, and David Eppstein. The crust and the beta-skeleton: Combinatorial curve reconstruction. *Graphic Models and Image Processing*, 60(2 of 2):125–135, 1998.
- [3] Dominique Attali and Annick Montanvert. Computing and simplifying 2d and 3d continuous skeletons. *Comput. Vis. Image Underst.*, 67(3):261–273, 1997.
- [4] C. Bradford Barber, David P. Dobkin, and Hannu Huhdanpaa. The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software*, 22(4):469–483, 1996.
- [5] Malte Calsen and Han-Christian Hege. Terrain rendering using spherical clipmaps. *Eurographics/IEEE-VGTC Symposium on Visualization*, 2006.
- [6] Kyong I. Chang, Kevin W. Bowyer, and Patrick J. Flynn. An evaluation of multimodal 2d+3d face biometrics. *IEEE Trans. Pattern Anal. Mach. Intell.*, 27(4):619–624, 2005.
- [7] Maciej Dakowicz and Christopher M. Gold. Visualizing terrain models from contours — plausible ridge, valley and slope estimation. In *Proc. Int. Worksh. Visualization and Animation of Landscape*, 2002.
- [8] Binary Space Partitioning Trees FAQ. <http://www.faqs.org/faqs/graphics/bsptree-faq/>.
- [9] Christopher M. Gold. Crust and anti-crust: A one-step boundary and skeleton extraction algorithm. In *Symposium on Computational Geometry*, pages 189–196, 1999.
- [10] Leonidas Guibas and Jorge Stolfi. Primitives for the manipulation of general subdivisions and the computation of voronoi. *ACM Trans. Graph.*, 4(2):74–123, 1985.
- [11] Koichi Harada and Eihachiro Nakamae. Sampling point setting on cubic splines for computer animation. *The Visual Computer*, 5(1&2):14–21, 1989.

- [12] Vineet R. Kamat and Julio C. Martinez. Dynamic three-dimensional visualization of fluid construction materials. *Journal of Computing in Civil Engineering*, 18(3):237–247, 2004.
- [13] Leo Livares. <http://www.bloggratis.org/blog/leolivarestar>.
- [14] mathworld. <http://mathworld.wolfram.com/>.
- [15] Instituto Nacional de Estadística Geografía e Informática INEGI. www.inegi.gob.mx.
- [16] Franco P. Preparata and Michael I. Shamos. *Computational Geometry: An Introduction (Monographs in Computer Science)*. Springer, August 1985.
- [17] Abigail Martínez Rivas. Reconstrucción del volumen a partir de su mapa de contornos. Master's thesis, Centro de Investigación y Estudios Avanzados del IPN Departamento de Ingeniería Eléctrica Sección de Computación, Noviembre 2005. Director Dr. Luis Gerardo de la Fraga.
- [18] Abigail Martínez Rivas and Dr. Luis Gerardo de la Fraga. Terrain reconstruction from contours maps. *Memoria del XIV Congreso Internacional de Computación CIC 2005*, pages 167–175.
- [19] Anderson DC Senderberg TW. Implicit representation of parametric curves and surfaces. *Comput Vis Graph Image*, 28:72–84, 1984.
- [20] Yeong-Ching Chen Shinn-Ying Ho. An efficient evolutionary algorithm for accurate polygonal approximation. In *GRAPHITE '04: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 2305–2317. Pattern Recognition, 2001.
- [21] Alexandru Telea, Cristian Sminchisescu, and Sven Dickinson. Optimal inference for hierarchical skeleton abstraction. In *ICPR '04: Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 4*, pages 19–22, Washington, DC, USA, 2004. IEEE Computer Society.
- [22] wikipedia. www.wikipedia.org.
- [23] Burkhard C. Wunsche and Richard Lobb. The 3d visualization of brain anatomy from diffusion-weighted magnetic resonance imaging data. In *GRAPHITE '04: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 74–83, New York, NY, USA, 2004. ACM Press.