

**ESTIMACIÓN DE NORMALES CON REORGANIZACIÓN
DE VECINDADES PARA RECONSTRUCCIÓN 3D**

TESIS

Que para obtener el grado de
MAESTRÍA EN CIENCIAS DE LA INGENIERÍA ELÉCTRICA

presenta

Ubaldo Ruiz López

Félix Calderón Solorio

Director de Tesis

Universidad Michoacana de San Nicolás de Hidalgo

Enero 2008

Agradecimientos

En primer lugar debo agradecer a mi familia. A mis padres Ubaldo y Ma. Guadalupe por inculcarme una cultura de trabajo y apoyarme de manera incondicional en todos mis proyectos. A mis hermanas Ana y Vianey por esa sana competencia que nos ha hecho ser mejores y alegrar mi vida con sus sonrisas y abrazos.

Debo agradecer a una persona especial en mi vida. Gracias Isis-Talia por estar siempre ahí para ayudarme, regañarme y orientarme cuando sentía que las cosas no iban bien.

Quiero agradecer a mis profesores Félix Calderón, Edgar Leonel Chávez y Leonardo Romero por compartir generosamente sus conocimientos y sembrar en mi el gusto por la investigación.

Gracias al Dr. Juan José Flores y al Dr. Mariano Rivera por sus excelentes comentarios y sugerencias que han contribuido de manera sustancial a enriquecer este trabajo.

Finalmente, gracias a todos mis compañeros y amigos con los cuales he tenido la fortuna de compartir esta experiencia.

Resumen

Los algoritmos de reconstrucción de superficies a partir de conjuntos de puntos, con mayor rapidez, mejor uso de memoria y buena tolerancia al ruido, requieren del conocimiento del vector normal a la superficie en cada punto. En la literatura, las normales son calculadas suponiendo que la superficie puede modelarse localmente con un plano en cada punto. El plano es calculado utilizando mínimos cuadrados sobre una vecindad definida alrededor del punto correspondiente. En la actualidad, las vecindades se crean considerando únicamente la distancia euclidiana entre los puntos. Diversos autores han observado que este criterio falla en regiones cercanas a bordes y esquinas, donde es posible que puntos que se encuentran en planos diferentes sean considerados como parte de la vecindad. Estos puntos generan una estimación errónea de la normal, la cual se traduce en una baja calidad de la reconstrucción.

En este trabajo se propone un nuevo algoritmo para estimación de normales con reorganización de vecindades. Nuestra propuesta cambia el tamaño de la vecindad en aquellos lugares donde están presentes bordes y esquinas, descartando los elementos que no pertenezcan al mismo plano. Para observar el desempeño de la propuesta, las estimaciones de la normal se comparan de manera cuantitativa y cualitativa con las que se obtienen utilizando el algoritmo propuesto por Hoppe et al. en [Hoppe92]. Las pruebas se realizan sobre conjuntos de datos con diferentes niveles de ruido. Los resultados muestran que el algoritmo de estimación con reorganización de vecindades tiene un mejor desempeño en las regiones cercanas a bordes y esquinas, sin afectar la estimación en regiones suaves. La mejor estimación de las normales se traduce en reconstrucciones con bordes y esquinas bien definidos.

Abstract

The surface reconstruction algorithms from point clouds with greater speed, better use of memory and good noise tolerance, require an estimation of the normal vector to the surface at each point. In the Literature, the normal is computed assuming that the surface can be modeled locally with a plane at each point. The plane is calculated using least squares on a neighborhood defined around the corresponding point. The neighborhoods are created considering the Euclidean distance between the points. Several authors have noticed that this idea fails in regions near edges and corners, where it is possible to take points in different planes as part of the neighborhood. These points generate an incorrect estimation of the normal, which produce a low quality reconstruction.

In this work, we propose a new algorithm for normal estimation with neighborhood reorganization. Our proposal changes the neighborhood size in those places where edges and corners are presented, discarding the elements that do not belong to the same plane. To observe the performance of our algorithm, its normal estimations are compared against the ones computed using the algorithm proposed by Hoppe et al. in [Hoppe92]. Several tests were executed in models with different noise levels. The results show that the algorithm of normal estimation with neighborhood reorganization has a better performance in regions near the edges and corners, without affecting the estimation in smooth regions. The normal estimations of our algorithm produce a surface reconstruction with well defined edges and corners.

Contenido

Agradecimientos	V
Resumen	VII
Abstract	IX
Contenido	XI
Lista de Figuras	XIII
Lista de Tablas	XVII
Lista de Símbolos	XIX
Lista de Publicaciones	XXI
1. Introducción	1
1.1. Antecedentes	2
1.2. Planteamiento del problema	4
1.3. Motivación	5
1.4. Objetivos de la tesis	6
1.5. Descripción de la tesis	6
2. Reconstrucción de superficies	7
2.1. Algoritmos geométricos	8
2.1.1. Algoritmo de corteza	9
2.1.2. Algoritmo de coconos	13
2.1.3. Tratamiento de bajas densidades en la muestra	14
2.2. Algoritmos volumétricos	16
2.2.1. Métodos de nivel cero	16
2.2.2. Funciones de base radial	20
2.2.3. Partición de unidad	22
2.2.4. Mínimos cuadrados móviles	25
2.2.5. Técnicas basadas en conceptos físicos y modelos deformables	26
2.3. Conclusiones	27
3. Métodos para la estimación de normales	29
3.1. Cálculo de las normales	30
3.2. Planteamiento alternativo	33
3.3. Orientación consistente	34
3.4. Ajuste con peso	37

3.5. Ajuste adaptativo	37
3.6. Conclusiones	39
4. Propuesta de estimación de normales con reorganización de vecindades	41
4.1. Sistema de resortes con condición de reposo adaptable	42
4.2. Regularización de normales	47
4.3. Cálculo de las normales regularizadas	54
4.4. Conclusiones	55
5. Experimentos y Resultados	57
5.1. Configuración experimental	57
5.2. Experimentos con modelos sintéticos	59
5.2.1. Cubo con medias esferas en dos de sus caras	59
5.2.2. Pirámide	61
5.2.3. Cilindro	61
5.3. Experimentos con modelos sintéticos con ruido	64
5.3.1. Ruido local	64
5.3.2. Ruido global	74
5.4. Conclusiones	81
6. Conclusiones	85
6.1. Conclusiones generales	85
6.2. Trabajo futuro	86
A. Conceptos básicos	87
A.1. Geometría	87
A.1.1. Diagrama de Voronoi	87
A.1.2. Triangulación de Delaunay	88
A.1.3. Complejo simplicial	89
A.1.4. Forma α	91
A.2. Espacios topológicos	92
A.2.1. Topología	92
A.2.2. Homeomorfismo	93
A.2.3. Triangulación	93
A.2.4. Variedades	94
A.2.5. Variedades con fronteras	95
A.2.6. Orientabilidad	96
Referencias	97

Lista de Figuras

1.1. Ajuste de una superficie a un conjunto de puntos en 3D.	2
1.2. Estimación de normales.	4
2.1. Reconstrucción de un conjunto con una densidad variable de puntos.	10
2.2. Algoritmo de corteza en dos dimensiones.	11
2.3. Celda de Voronoi.	12
2.4. Creación de cocono.	14
2.5. Reconstrucciones efectuadas con dos métodos tolerantes a bajas densidades en la muestra.	15
2.6. Reconstrucciones efectuadas sobre datos con ruido utilizando dos algoritmos geométricos.	15
2.7. Distancia signada.	17
2.8. Planos tangentes.	18
2.9. Proyección sobre el plano tangente.	18
2.10. Reconstrucciones generadas con la propuesta de Hoppe et al.	20
2.11. Ejemplo de reconstrucciones empleando funciones de base radial.	23
2.12. Ejemplo de reconstrucciones con el método MPUI.	25
3.1. Plano tangente.	30
3.2. Vectores propios.	31
3.3. Normales calculadas a partir de un conjunto de puntos.	32
3.4. Acercamiento sobre la esquina superior izquierda de la Figura 3.3.	32
3.5. Árbol Euclidiano de Expansión Mínima.	35
3.6. Grafo Riemanniano calculado de acuerdo al método propuesto por Hoppe et al.	35
3.7. Normales orientadas de acuerdo al método propuesto por Hoppe et al.	36
4.1. Ejemplo de una función observada y la función esperada correspondiente.	42
4.2. Funciones de ajuste suaves para distintos valores de η	45
4.3. Función de ajuste con ARC.	47
4.4. Valores de l_i para la función en la Figura 4.1(a).	47
4.5. Ejemplo de un conjunto de puntos muestreado sobre una superficie en 2D.	48
4.6. Normales correctas.	48
4.7. Gráfica de las componentes de las normales.	49

4.8. Normales calculadas por el método de Hoppe et al.	49
4.9. Gráfica de las componentes de las normales calculadas por el método de Hoppe et al.	50
4.10. Ejemplo de una normal atípica.	50
4.11. Nueva estimacion de la normal.	51
4.12. Normales regularizadas empleando una función de energía con ARC.	53
4.13. Gráfica de las componentes de las normales regularizadas.	54
4.14. Valores de l_i para las normales en la Figura 4.8.	54
5.1. Superficies reconstruidas para el modelo de un cubo modificado.	60
5.2. Superficies reconstruidas para el modelo de una pirámide.	62
5.3. Superficies reconstruidas para el modelo de un cilindro.	63
5.4. Gráfica de los errores de estimación en ambos métodos para el modelo de un cubo modificado con ruido local.	65
5.5. Reconstrucciones agregando ruido local, con un factor de 0.1.	65
5.6. Reconstrucciones agregando ruido local, con un factor de 0.2.	66
5.7. Reconstrucciones agregando ruido local, con un factor de 0.3.	66
5.8. Reconstrucciones agregando ruido local, con un factor de 0.4.	67
5.9. Gráfica de los errores de estimación en ambos métodos para el modelo de una pirámide con ruido local.	68
5.10. Reconstrucciones agregando ruido local, con un factor de 0.1.	69
5.11. Reconstrucciones agregando ruido local, con un factor de 0.2.	69
5.12. Reconstrucciones agregando ruido local, con un factor de 0.3.	70
5.13. Reconstrucciones agregando ruido local, con un factor de 0.4.	70
5.14. Gráfica de los errores de estimación en ambos métodos para el modelo de un cilindro con ruido local.	71
5.15. Reconstrucciones agregando ruido local, con un factor de 0.1.	72
5.16. Reconstrucciones agregando ruido local, con un factor de 0.2.	72
5.17. Reconstrucciones agregando ruido local, con un factor de 0.3.	73
5.18. Reconstrucciones agregando ruido local, con un factor de 0.4.	73
5.19. Gráfica de los errores de estimación en ambos métodos para el modelo de un cubo modificado con ruido global.	75
5.20. Reconstrucciones agregando ruido global, con un factor de 0.005.	75
5.21. Reconstrucciones agregando ruido global, con un factor de 0.0075.	76
5.22. Reconstrucciones agregando ruido global, con un factor de 0.01.	76
5.23. Reconstrucciones agregando ruido global, con un factor de 0.0125.	77
5.24. Reconstrucciones agregando ruido global, con un factor de 0.015.	77
5.25. Gráfica de los errores de estimación en ambos métodos para el modelo de una pirámide con ruido global.	78
5.26. Reconstrucciones agregando ruido global, con un factor de 0.005.	79
5.27. Reconstrucciones agregando ruido global, con un factor de 0.0075.	79
5.28. Reconstrucciones agregando ruido global, con un factor de 0.01.	80
5.29. Reconstrucciones agregando ruido global, con un factor de 0.015.	80
5.30. Gráfica de los errores de estimación en ambos métodos para el modelo de un cilindro con ruido global.	82

5.31. Reconstrucciones agregando ruido global, con un factor de 0.005.	82
5.32. Reconstrucciones agregando ruido global, con un factor de 0.0075.	83
5.33. Reconstrucciones agregando ruido global, con un factor de 0.01.	83
5.34. Reconstrucciones agregando ruido global, con un factor de 0.0125.	84
5.35. Reconstrucciones agregando ruido global, con un factor de 0.015.	84
A.1. Diagrama de Voronoi.	88
A.2. Triangulación de Delaunay.	88
A.3. Diagrama de Voronoi (naranja) y Triangulación de Delaunay (negro).	89
A.4. Tipos de simplejos.	90
A.5. Espacios en una dimensión con diferentes tipos topológicos.	93
A.6. Triangulación de un disco cerrado.	94
A.7. Ejemplos de variedades.	95
A.8. Ejemplos de variedades con fronteras.	95
A.9. Ejemplos de variedades no orientables con y sin fronteras.	96

Lista de Tablas

5.1. Factores de escala para los tipos de ruido en los experimentos.	58
5.2. Errores de estimación en ambos métodos para el modelo de un cubo modificado.	59
5.3. Errores de estimación en ambos métodos para el modelo de una pirámide.	61
5.4. Errores de estimación en ambos métodos para el modelo de un cilindro.	62
5.5. Errores de estimación en ambos métodos para el modelo de un cubo modificado con ruido local.	64
5.6. Errores de estimación en ambos métodos para el modelo de una pirámide con ruido local.	68
5.7. Errores de estimación en ambos métodos para el modelo de un cilindro con ruido local.	71
5.8. Errores de estimación en ambos métodos para el modelo de un cubo modificado con ruido global.	74
5.9. Errores de estimación en ambos métodos para el modelo de una pirámide con ruido global.	78
5.10. Errores de estimación en ambos métodos para el modelo de un cilindro con ruido global.	81

Lista de Símbolos

S	Superficie original.
\mathbf{p}	Punto cualquiera en \mathbb{R}^3 .
X	Conjunto de puntos en un espacio tridimensional adquiridos a partir de S .
\mathbf{x}_i	i -ésimo elemento de X .
vor	Conjunto de vértices del diagrama de Voronoi de X .
vor_x	Celda de Voronoi asociada al punto \mathbf{x} .
\mathbf{p}_x^+	Vértice de vor_x más alejado de \mathbf{x} .
\mathbf{p}_x^-	Vértice de vor_x más alejado de \mathbf{x} en la dirección opuesta a \mathbf{p}_x^+ .
\mathbf{n}_x^+	Vector de \mathbf{x} a \mathbf{p}_x^+ .
U	Conjunto de todos los vértices \mathbf{p}^+ y \mathbf{p}^- .
coc_x	Cocono. Complemento del cono doble con vértice en \mathbf{x} , el vector \mathbf{n}_x como eje y un ángulo de apertura de $\frac{3\pi}{4}$.
$Z(f)$	Conjunto de nivel cero de la función f .
T_i	Plano tangente asociado al punto \mathbf{x}_i .
V_i	k -vecinos más cercanos al punto \mathbf{x}_i .
\mathbf{c}_i	Centro asociado al plano T_i .
\mathbf{n}_i	Vector normal unitario al plano T_i .
k	Número de elementos de V_i .
$ds(\mathbf{p})$	Distancia entre un punto \mathbf{p} y el punto más cercano, $\mathbf{z} \in S$, multiplicado por ± 1 dependiendo del lado de la superficie en que se encuentra.
$d(\mathbf{p}, X)$	Distancia de Hausdorff de un punto \mathbf{p} a X .
ρ_A	Densidad de adquisición de la muestra X .
δ_A	Magnitud del error de adquisición más grande en X .
$\Phi(\mathbf{p}_j, \mathbf{p}_i)$	Función centrada en \mathbf{p}_i .
β	Vector con pesos desconocidos.
P	Polinomio.

C_i	Matriz de covarianza de V_i .
λ_i^j	j -ésimo valor propio de la matriz C_i .
\mathbf{v}_i^j	j -ésimo vector propio de la matriz C_i .
\mathbf{x}_j	j -ésimo elemento de V_i .
N_i	Nodo i -ésimo del grafo.
(i, j)	Lado entre los nodos N_i y N_j .
r	Radio de una esfera centrada en \mathbf{x}_i .
σ_n	Desviación estándar.
ρ_L	Densidad local de muestreo.
κ_L	Curvatura local.
s	Distancia de \mathbf{x}_i a su k -vecino más cercano.
ν	Distancia promedio de \mathbf{x}_i a sus k -vecinos más cercanos.
d	Distancia de \mathbf{x}_i al plano de mejor ajuste de los k -vecinos.
E	Energía potencial almacenada en un sistema de resortes.
g	Función observada.
f	Función esperada.
∇f_i	Gradiente de la función f_i .
l_i	Término de pertenencia a la vecindad.
η	Factor de suavidad.
μ	Término de penalización.
\mathbf{m}_i	Normal regularizada.
\mathbf{n}'_i	Nueva estimación del vector normal.
γ	Multiplicador de Lagrange.

Lista de Publicaciones

Calderon Felix, Ruiz Ubaldo y Rivera Mariano. Surface–Normal Estimation with Neighborhood Reorganization for 3D Reconstruction. En *12th Iberoamerican Congress on Pattern Recognition, CIARP 2007. Lecture Notes in Computer Science LNCS 4756 : 321-330*. Valparaiso Chile, Noviembre 13-16, 2007.

Capítulo 1

Introducción

La representación tridimensional por computadora de un objeto físico tiene una amplia variedad de aplicaciones en áreas como el diseño industrial, simulación por computadora y medicina, entre otras. La creación de esta representación cae dentro del área denominada modelado geométrico y se realiza en una serie de pasos [Söderkvist99]. Estos pasos se ejecutan en un orden fijo y son típicamente los siguientes:

Adquisición de datos: Consiste en obtener las coordenadas en 3D de puntos sobre la superficie del objeto [Saleem04], generalmente empleando dispositivos ópticos de medición. Estos dispositivos frecuentemente se conocen como cámaras de rango y producen imágenes que contienen información sobre las distancias de la cámara a la superficie. Es necesario operar las cámaras de rango desde distintas vistas para poder obtener datos sobre la superficie completa. Cada imagen de rango normalmente contiene una estructura de los datos que puede ser utilizada cuando éstos son procesados.

Registro e integración: Si se generan múltiples vistas durante la adquisición de los datos, es necesario realizar un proceso de registro [Rusinkiewicz01] para encontrar las transformaciones que conectan las distintas imágenes de rango. Una vez que se conocen las transformaciones, éstas son usadas para poner los datos de las distintas vistas en una representación unificada expresada en un sistema común de coordenadas.

Ajuste de la superficie: El objetivo de este paso es construir una aproximación concisa de la superficie del objeto. Este paso generalmente incluye determinar la topología del objeto cuando no es conocida a priori y ajustar una superficie con esta topología. Como ocurre comúnmente, existe una relación entre la precisión y la simplicidad de un modelo. Los métodos de ajuste típicamente tratan de minimizar una función que captura esta relación. El ajuste puede realizarse en cada una de las vistas generadas durante el proceso de adquisición, para después unificarse durante el proceso de integración. Sin embargo, es preferible realizarlo sobre un único conjunto de puntos en 3D (ver Figura 1.1), problema que será abordado en este trabajo.

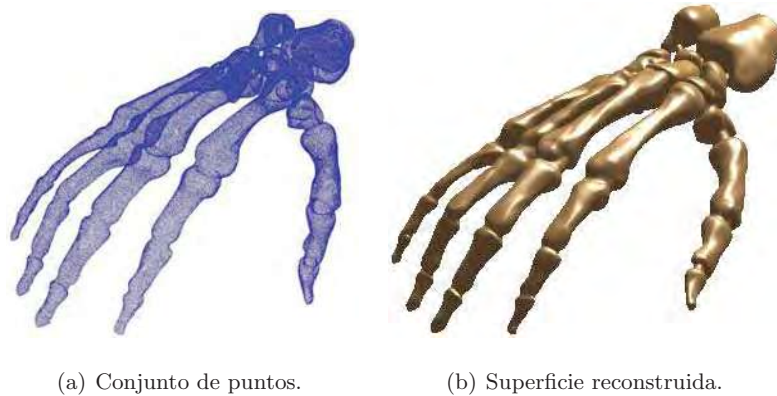


Figura 1.1: Ajuste de una superficie a un conjunto de puntos en 3D. Fuente: [Carr01].

1.1. Antecedentes

La reconstrucción de superficies a partir de conjuntos de puntos en 3D fue tratada probablemente por primera vez por Edelsbrunner et al. [Edelsbrunner83] y Boissonnat [Boissonnat84] a mediados de los 80s. Actualmente es una área activa donde es posible distinguir dos corrientes principales de investigación [Schall05] que son la geométrica y la volumétrica.

La corriente geométrica se basa en conceptos y algoritmos [Amenta98c, Dey03] desarrollados en el campo de la geometría computacional [O'Rourke00], especialmente los

diagramas de Voronoi y su dual, la triangulación de Delaunay. Esta corriente, a pesar de ser la única que garantiza sus resultados teóricamente [Amenta98a] para superficies suaves, depende de una densidad adecuada de puntos en la muestra y la ausencia de ruido para obtener buenos resultados. Estos requisitos limitan su uso con modelos capturados a partir de objetos reales. Existen varias líneas de investigación en esta dirección, tratando de reducir la dependencia de la densidad de la muestra [Amenta01], aumentar la tolerancia al ruido [Dey04, Kolluri04] y disminuir su elevado costo computacional [Dey01].

La corriente volumétrica se basa en la idea de representar la superficie mediante el conjunto de nivel¹ cero de una función implícita. Actualmente esta corriente se utiliza ampliamente debido a una mejor tolerancia al ruido, la posibilidad de trabajar con grandes conjuntos de puntos y su capacidad para manejar superficies con bordes y esquinas. La representación resultante es fácil de manipular, sin embargo, para poder representarla en pantalla es necesario aplicar un algoritmo de extracción [Lorensen87, Bloomenthal94, Hart96] que la convierta en una malla triangular.

Es importante mencionar que los algoritmos existentes en esta corriente [Hoppe92, Carr01, Ohtake03b] requieren de información exacta sobre las normales asociadas al conjunto de puntos para definir una orientación y construir un función de distancia con signo, la cual toma valores positivos o negativos al proyectar puntos del espacio sobre la superficie dependiendo del lado de ésta en que se encuentren. El conjunto de nivel cero de la función de distancia se toma como una aproximación a la superficie real.

En algunos casos una estimación de las normales se puede obtener durante el proceso de adquisición, sin embargo, es deseable y en ocasiones necesario poder inferirlas directamente del conjunto de puntos. Varios trabajos han sido desarrollados en este sentido [Hoppe92, Pauly03, Mitra03, Dey05], sin embargo, a pesar de funcionar adecuadamente en regiones suaves tienen deficiencias en la regiones cercanas a esquinas y bordes, que repercuten directamente en la calidad de la superficie reconstruida.

¹Sea H un conjunto y $f : H \rightarrow \mathbb{R}$ un campo escalar sobre H . El conjunto de nivel C_k para la función f es el subconjunto de los puntos x en H para los cuales $f(x) = k$. En símbolos: $C_k = \{x \in H | f(x) = k\}$. Si $H = \mathbb{R}^2$ los conjuntos de nivel son en general curvas y se las llama curvas de nivel. Si $H = \mathbb{R}^3$ los conjuntos de nivel suelen ser superficies y se les llama superficies de nivel.

1.2. Planteamiento del problema

Generalmente, la estimación de normales se realiza en dos etapas [Hoppe92]. En la primera etapa, se asigna un plano tangente a cada punto y la normal se aproxima con el vector ortogonal unitario a este plano. Es importante notar que la aproximación de la normal tiene dos orientaciones posibles [Hoppe92]. En la segunda etapa, se calcula una orientación espacialmente consistente de las normales, tomando como base que las normales de puntos geoméricamente cercanos deben ser casi paralelas y su producto interno deber tender a uno. Los planos tangentes se determinan mediante un ajuste por mínimos cuadrados de los k -vecinos más cercanos a cada punto.

En la actualidad, la selección de estos vecinos toma en cuenta únicamente su proximidad geométrica. No obstante, se ha observado que esta técnica falla al considerar elementos que pertenecen a una parte de la superficie que tiene características diferentes. En la Figura 1.2(a) se muestra un ejemplo de la estimación errónea efectuada por los métodos actuales, con los cuales la normal en el punto 3 se calcula ajustando un plano a los puntos 1, 2, 3, 4 y 5, seleccionados tomando en cuenta su distancia euclidiana al punto 3. En la Figura 1.2(b) se puede observar la normal estimada cuando el ajuste se realiza únicamente a los puntos 1, 2 y 3, seleccionados considerando su distancia euclidiana y su contribución a la estimación de la normal. En la Figura 1.2(c) se presenta una estimación alternativa de la normal, en ella el ajuste se realiza únicamente a los puntos 3, 4 y 5. Los dos últimos criterios de selección permiten obtener la estimación de la normal más adecuada.

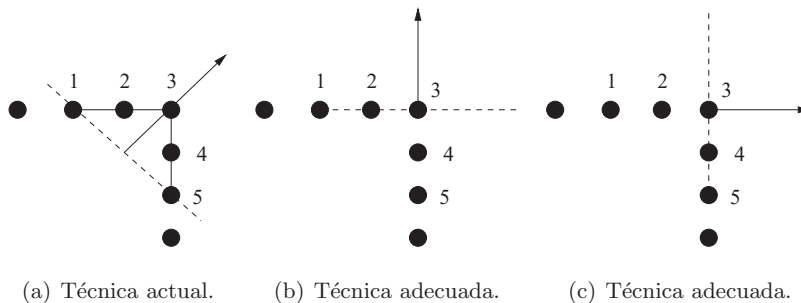


Figura 1.2: Estimación de normales.

Por lo tanto, es necesario desarrollar una estrategia para determinar los elementos que deben formar parte de la vecindad, que no solo tome en cuenta su proximidad geométrica sino que además cuantifique el efecto de su contribución a la estimación.

1.3. Motivación

La representación digital de un objeto del mundo real tiene ventajas sobre el objeto mismo dado que puede ser replicada, diseminada, modificada o estudiada. Además los objetos reales sufren del deterioro asociado con el paso del tiempo y es necesario darles mantenimiento. La representación digital no sufre de este mal y puede ser almacenada en su forma original por un tiempo indefinido. Por lo tanto, es conveniente tener esta representación. A continuación se dan ejemplos de algunas aplicaciones:

Ingeniería inversa: Muchas industrias tiene grandes catálogos de partes que fueron desarrolladas sin herramientas CAD y que aún pueden ser de mucho interés para sus creadores. Por esta razón, es deseable obtener una representación digital la cual puede ser incorporada o modificada en nuevos diseños o simplemente archivada.

Diseño industrial: A pesar de que en estos días la mayoría del diseño y modelado es realizado empleado herramientas CAD hay ocasiones donde es más conveniente realizar prototipos en madera o arcilla como por ejemplo en el diseño de automóviles o modelado de características faciales. En estas situaciones el prototipo final puede ser representado digitalmente y el modelo por computadora puede entonces ser sometido a modificaciones, simulaciones o enviado a manufactura.

Ambientes virtuales: La creación de ambientes virtuales simulando el mundo físico alrededor de nosotros requiere de modelos para poblar estos mundos. Actualmente, los ambientes virtuales contienen caracteres con apariencia de caricatura en parte debido a la carencia de modelos realistas generados por computadora de los objetos cotidianos. La representación digital de objetos reales contribuye a mejorar la calidad de los ambientes virtuales.

1.4. Objetivos de la tesis

El objetivo de este trabajo es desarrollar una estrategia para mejorar el cálculo de las normales asociadas a un conjunto de puntos, que se verá reflejada en una mayor calidad de la reconstrucción tridimensional. La propuesta debe:

- Determinar que elementos pertenecen a las vecindades asociadas a cada punto, tomando en cuenta no solo su proximidad geométrica sino también su contribución a la estimación.
- Tolerar el ruido presente en la mayoría de los procesos de adquisición.
- Igualar la efectividad de los métodos existentes en regiones suaves.
- Mejorar la estimación cerca de los bordes y esquinas.

1.5. Descripción de la tesis

El capítulo 2 consiste en un recuento de los algoritmos más significativos para reconstrucción de superficies a partir de conjuntos de puntos en 3D. En ese capítulo se destacan los beneficios de usar una representación implícita y la necesidad de una estimación correcta de las normales asociadas a cada punto. El capítulo 3 describe los algoritmos existentes para el cálculo de normales a partir de conjuntos de puntos en 3D. Ahí se hará notar la importancia de una elección correcta de los elementos en las vecindades asociadas en cada punto para la estimación de la normal correspondiente. El capítulo 4 expone la propuesta desarrollada en este trabajo para mejorar el cálculo de las normales en regiones cercanas a bordes y esquinas. La propuesta está basada en un sistema de resortes con condición de reposo adaptable presentado en [Rivera00]. El capítulo 5 muestra resultados y comparativas de los métodos existentes y la propuesta de este trabajo. Finalmente, en el capítulo 6 se exponen las conclusiones generales del trabajo y enumeran algunas de las líneas de investigación en futuros desarrollos.

Capítulo 2

Reconstrucción de superficies

A partir de la investigación realizada por Hoppe et al. [Hoppe92, Hoppe93, Hoppe94], el problema de la reconstrucción de superficies en conjuntos de puntos ha recibido mayor atención, convirtiéndose en un área activa de investigación. En la actualidad, se han propuesto una gran cantidad de algoritmos, la mayoría de ellos basados en ideas tomadas de la física y las matemáticas. Entre los algoritmos existentes se pueden distinguir dos líneas principales de investigación [Schall05]: los algoritmos geométricos y los algoritmos volumétricos.

Los algoritmos geométricos [Edelsbrunner92, Amenta98c, Amenta01, Dey03] toman como piezas fundamentales en el proceso de reconstrucción varios de los conceptos (ver Apéndice A.1) desarrollados en el área conocida como geometría computacional [O'Rourke00], especialmente, la llamada triangulación de Delaunay (ver Apéndice A.1.2). Estos algoritmos resultan atractivos al ser los únicos algoritmos de reconstrucción de superficies que garantizan teóricamente sus resultados. Desafortunadamente, estas garantías imponen una serie de restricciones en la muestra y el tipo de superficie a reconstruir, que difícilmente son satisfechas en modelos reales. Es necesario que las muestras sean lo suficientemente densas y libres de ruido, además, las superficies a reconstruir deben ser suaves. En la sección 2.1, se presenta un resumen de los algoritmos considerados como fundamentales en esta categoría y que están relacionados con el cálculo de normales. Para un análisis a detalle sobre los algoritmos geométricos existentes se sugiere consultar [Cazals04, Cazals06, Dey06].

Los algoritmos volumétricos [Hoppe92, Carr01, Ohtake03b] aproximan las muestras mediante una función implícita en 3D. La superficie puede ser reconstruida explícitamente extrayéndola como el conjunto de nivel cero de la función calculada. Estos algoritmos compensan el hecho de no garantizar sus resultados teóricamente con una mayor flexibilidad en la calidad de la muestra y el tipo de superficie a reconstruir. Son capaces de obtener buenas aproximaciones en muestras que contienen ruido, además de manejar bordes y esquinas de manera adecuada en una propuesta reciente [Ohtake03b]. En la sección 2.2, se describen las ideas detrás de algunos algoritmos fundamentales en esta corriente.

2.1. Algoritmos geométricos

Los algoritmos geométricos se basan en la idea de que reconstruir la superficie consiste en establecer conexiones entre los puntos. Cualquier construcción geométrica que defina un complejo simplicial (ver Apéndice A.1.3) sobre las muestras es una estructura auxiliar para la reconstrucción. Una de estas estructuras es la triangulación de Delaunay; algunos métodos calculan un subconjunto de esta triangulación para encontrar la superficie.

Uno de los primeros trabajos en esta corriente está basado en formas α (ver Apéndice A.1.4) y fue desarrollado por Edelsbrunner et al. [Edelsbrunner83]. Las formas α [Edelsbrunner92] son construcciones parametrizadas de los diagramas de Voronoi (ver Apéndice A.1.1) y las triangulaciones de Delaunay que asocian una forma poliédrica a un conjunto de puntos. Un simplejo (ver Apéndice A.1.3) se incluye en una forma α si su circunferencia¹ de radio no mayor que α no contiene puntos de la muestra. El espectro de formas α , ésto es, el conjunto de formas α para todos los valores de α , da una idea de la forma general del conjunto de puntos. Las formas α se usan ampliamente para la reconstrucción de superficies. Bajaj et al. [Bajaj95b] las usan para obtener una representación volumétrica intermedia del conjunto. También han sido usadas por Sakkalis et al. [Sakkalis99] para la reconstrucción de curvas.

¹La esfera de menor radio que contiene un sólido dado.

Otro trabajo pionero en el área fue presentado por Boissonnat [Boissonnat84]. Su algoritmo elimina progresivamente elementos de la triangulación de Delaunay del conjunto de puntos, basándose en sus circunferencias. Para el caso bidimensional, Amenta et al. en [Amenta98b, Attali98] y Bernardini et al. en [Bernardini97] presentan resultados para reconstrucciones suaves basadas en Delaunay. Dey et al. en [Dey99a, Dey99b] y Giesen et al. en [Giesen99] lo hacen para técnicas de reconstrucción de curvas basadas en el problema del agente viajero y otras ideas tomadas de la geometría computacional. Amenta et al. en [Amenta98a, Amenta98c] fueron los primeros en dar resultados teóricos con su algoritmo de corteza en el caso tridimensional. Su algoritmo de corteza es similar al algoritmo de Boissonnat. El algoritmo presentado en [Amenta98c] ha derivado a otros que son considerados como fundamentales en el área y serán abordados con más detalle en este capítulo.

Otra propuesta interesante fue presentada por Mencl [Mencl95], donde se rellenan contornos en una extensión del Árbol Euclidiano de Expansión Mínima² (EMST por sus siglas en inglés) del conjunto de puntos. Attene et al. en [Attene00] emplean el Árbol Euclidiano de Expansión Mínima y el grafo de Gabriel³ para definir nuevas operaciones para eliminar tetraedros en algoritmos similares a [Boissonnat84].

2.1.1. Algoritmo de corteza

Este algoritmo fue propuesto por Amenta et al. [Amenta98c], siendo el primero con garantías demostrables para el problema de la reconstrucción de superficies. Dada una *buena muestra* a partir de una superficie suave, la salida está garantizada para ser topológicamente correcta y converger a la superficie original, mientras la densidad de muestreo aumente. La definición de *buena muestra* resulta interesante ya que la densidad de muestreo requerida varía localmente, capturando la noción intuitiva de que las áreas con menos características de una superficie pueden ser reconstruidas a partir de menos muestras, la Figura 2.1 da una idea general.

²Es el árbol con costo mínimo conectando todos los puntos en el conjunto X , donde el costo de un lado corresponde a la distancia euclidiana entre los nodos. Este árbol es un subgrafo de la triangulación de Delaunay de X .

³Es un grafo que conecta un conjunto de puntos en el espacio euclidiano. Dos puntos L y M están conectados por un lado siempre que el círculo que tiene el segmento LM como su diámetro no contenga otros puntos del conjunto.

El algoritmo produce un conjunto de triángulos que se denominan la corteza de la muestra y cuyos vértices son elementos de ella, todos los triángulos en la corteza forman parte de la triangulación de Delaunay de los puntos en la muestra.

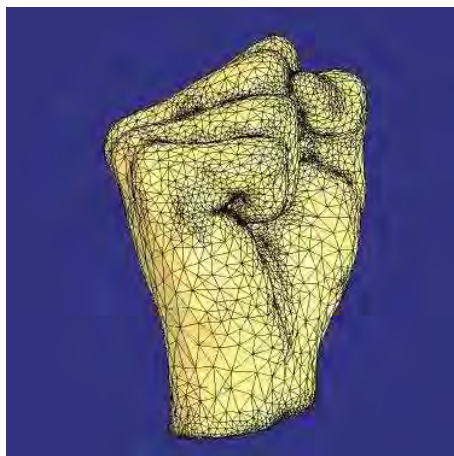
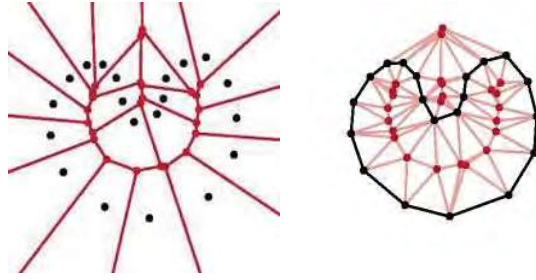


Figura 2.1: Malla reconstruida empleando el algoritmo de corteza. Fuente: [Amenta98c].

Amenta et al. comienzan con una versión bidimensional del algoritmo. En este caso, la corteza es un grafo sobre el conjunto de puntos X de una superficie S y la definen de la siguiente manera:

Un lado e pertenece a la corteza si e tiene un circuncírculo vacío no solo de todos los otros puntos en la muestra sino además de todos los vértices de Voronoi de X . La definición anterior conduce al siguiente algoritmo, el cual se ilustra en la Figura 2.2. Primeramente, se calcula el diagrama de Voronoi de X y se denota como vor al conjunto de vértices de Voronoi. A continuación se calcula la triangulación de $X \cup vor$. La corteza consiste en los lados de Delaunay entre los puntos de X , ya que estos son los lados con circuncírculos vacíos de puntos en $X \cup vor$. Se debe notar que la corteza es un subconjunto de la triangulación de Delaunay de los puntos de entrada. Al añadir los vértices de Voronoi se quedan fuera los lados no deseados de la triangulación de Delaunay. Esta técnica recibe el nombre de filtrado de Voronoi.



(a) Diagrama de Voronoi. (b) Triangulación de Delaunay.

Figura 2.2: Algoritmo de corteza en dos dimensiones. En la imagen izquierda, se muestra el diagrama de Voronoi de un conjunto X adquirido a partir de una curva. En la imagen derecha, se presenta la triangulación de Delaunay de $X \cup V$, con los lados de la corteza en negro. Fuente: [Amenta98c].

En el caso tridimensional, para cada punto \mathbf{x} se usan únicamente los dos vértices de la celda de voronoi asociada (vor_x) que están más alejados a ese punto, un vértice en cada lado de la superficie S . Amenta et al. llaman a los dos vértices seleccionados los polos de \mathbf{x} y los denotan como \mathbf{p}_x^+ y \mathbf{p}_x^- . El polo \mathbf{p}_x^+ se toma como el vértice de vor_x más alejado de \mathbf{x} . La observación en [Amenta98c] de que vor_x es larga y delgada (ver Figura 2.3), implica que el otro polo \mathbf{p}_x^- debe caer casi en la dirección opuesta. Por lo tanto, se elige \mathbf{p}_x^- simplemente como el vértice en vor_x tal que $\mathbf{p}_x^- - \mathbf{x}$ y $\mathbf{p}_x^+ - \mathbf{x}$ tienen un producto punto negativo.

A continuación se presenta el algoritmo básico tomado de [Amenta98c].

1. Se calcula el diagrama de Voronoi del conjunto de puntos X .
2. Para cada punto \mathbf{x} en el conjunto:
 - a) Si \mathbf{x} no cae sobre la envolvente convexa, sea \mathbf{p}_x^+ el vértice más alejado de \mathbf{x} en vor_x . Sea \mathbf{n}_x^+ el vector $\mathbf{p}_x^+ - \mathbf{x}$. Amenta et al. llaman vector polar a \mathbf{n}_x^+ .
 - b) Si \mathbf{x} cae en la envolvente convexa, sea \mathbf{n}_x^+ el promedio de las normales de los triángulos adyacentes.
 - c) Sea \mathbf{p}_x^- el vértice de Voronoi de vor_x con proyección negativa sobre \mathbf{n}_x^+ que está más alejado de \mathbf{x} .

3. Sea U el conjunto de todos los polos \mathbf{p}_x^+ y \mathbf{p}_x^- . Calcular la triangulación de Delaunay de $X \cup U$.
4. Mantener solo aquellos triángulos para los cuales sus tres vértices son puntos de la muestra X .

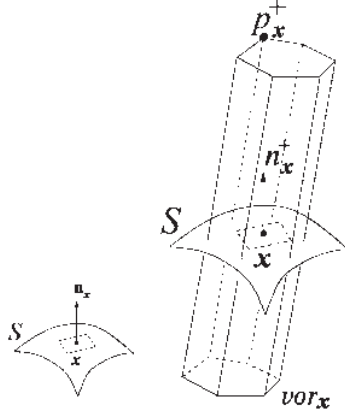


Figura 2.3: La celda de Voronoi vor_x es larga y delgada en la dirección de la normal real \mathbf{n}_x . El vector $\mathbf{p}_x^+ - \mathbf{x}$ aproxima a \mathbf{n}_x . Fuente: [Dey03].

Amenta et al. mencionan que se requiere un filtrado adicional para producir una variedad (ver Apéndice A.2.4) lineal a pedazos homeomorfa (ver Apéndice A.2.2) a S y garantizar que la salida converge a la superficie real mientras la densidad de muestreo se incrementa. Un teorema presentado en [Amenta98a], afirma que los vectores $\mathbf{n}_x^+ = \mathbf{p}_x^+ - \mathbf{x}$ y $\mathbf{n}_x^- = \mathbf{p}_x^- - \mathbf{x}$ a partir de un punto de la muestra a sus polos están garantizados a ser casi ortogonales a la superficie en \mathbf{x} . Estos vectores pueden ser utilizados en un proceso adicional de filtrado de normales eliminando todos aquellos triángulos cuyas normales difieren demasiado de \mathbf{n}_x^+ y de \mathbf{n}_x^- . Al usar el filtrado de normales, las normales de los triángulos de salida se aproximan a las normales de la superficie a medida que se incrementa la densidad de muestreo. En [Amenta98a] prueban que los triángulos restantes contienen un subconjunto formando una superficie plana a pedazos homeomorfa a la superficie S .

Amenta et al. afirman que el filtrado de normales puede ser peligroso cerca de los bordes y esquinas. Las direcciones de \mathbf{n}_x^+ y \mathbf{n}_x^- no son ortogonales a todos los planos tangentes cercanos y algunos triángulos deseables pueden ser eliminados. Esto limita la aplicación del método a conjuntos de puntos cuya superficie no sea suave. Otra limitante es la complejidad del algoritmo, que es $O(n^2)$, dado que ese es el tiempo en el peor de los casos, requerido para calcular la triangulación de Delaunay en tres dimensiones. Es importante notar que el número de puntos en la muestra más los polos es a lo más $3n$.

2.1.2. Algoritmo de coconos

Fue desarrollado por Amenta et al. en [Amenta00] a partir del algoritmo anterior. De manera similar al algoritmo de corteza, calcula un conjunto que contienen la triangulación de Delaunay de X si la muestra dada es lo suficientemente densa. Ésto se logra calculando el diagrama de Voronoi de X para encontrar los vectores polares de cada uno de los puntos en la muestra.

Amenta et al. definen un cocono coc_x para todo punto \mathbf{x} (ver Figura 2.4), como el complemento del cono doble con vértice en \mathbf{x} , el vector polar \mathbf{n}_x^+ como eje y un ángulo de apertura de $\frac{3\pi}{4}$. Como el vector polar \mathbf{n}_x^+ aproxima la normal a la superficie en \mathbf{x} el cocono determina una vecindad alrededor del plano tangente en \mathbf{x} . El algoritmo detecta todos los lados de Voronoi en vor_x que son intersectados por el cocono coc_x para todo punto \mathbf{x} . Los triángulos duales de Delaunay constituyen el conjunto de triángulos a formar la superficie. Después de que se determina completamente el conjunto de triángulos se ejecuta un paso de extracción para buscar la superficie.

Este algoritmo tiene las mismas garantías teóricas que el algoritmo de corteza, sin embargo, al igual que ese algoritmo requiere del cálculo de la triangulación de Delaunay de X , lo que implica que la complejidad de ejecución es $O(n^2)$. Presenta además los mismos problemas para reconstruir superficies con bordes y esquinas.

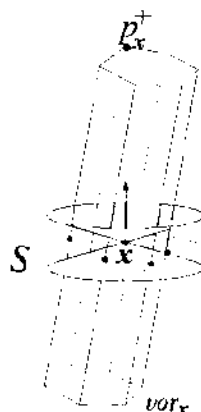


Figura 2.4: Cocono con vértice en \mathbf{x} . Los puntos indican los lados de la celda que intersecta. Fuente: [Dey03]

2.1.3. Tratamiento de bajas densidades en la muestra

Los métodos anteriores están respaldados por resultados matemáticos rigurosos y dan garantías de que la superficie reconstruida es correcta bajo las condiciones ya presentadas. Sin embargo, los algoritmos geométricos enfrentan dificultades al tratar con datos ruidosos y muestreos pobres.

Un problema común cuando las condiciones de muestreo no se satisfacen es que las superficies resultantes pueden presentar hoyos. Se han desarrollado varias extensiones importantes de los métodos anteriores, las cuales producen superficies sin hoyos. Los métodos dentro de estas extensiones generalmente son algoritmos basados en algún tipo de marcado interior/exterior de la superficie y de forma general funcionan de la siguiente manera:

Primero calculan la triangulación de Delaunay de los puntos en la muestra, los tetraedros resultantes son marcados como interiores o exteriores dependiendo de si el tetraedro está dentro del sólido acotado por los datos o fuera de él. Una vez que todos los tetraedros han sido marcados la superficie final es extraída eliminando todos aquellos triángulos que no son compartidos por un tetraedro marcado como exterior y otro marcado como interior. Del procedimiento anterior se tiene que la superficie no tiene hoyos.

Dos de los métodos que generan este tipo de reconstrucciones son el PowerCrust presentado por Amenta et al. [Amenta01] y el TightCocone por Dey et al. [Dey03]. Un par de reconstrucciones generadas con ambos métodos se muestran en la Figura 2.5. Al igual que los algoritmos geométricos descritos anteriormente, la complejidad de ambos métodos es de $O(n^2)$ lo que imposibilita su uso en conjuntos con millones de puntos, tienen baja tolerancia al ruido (ver Figura 2.6) y presentan errores al reconstruir regiones cercanas a bordes y esquinas.

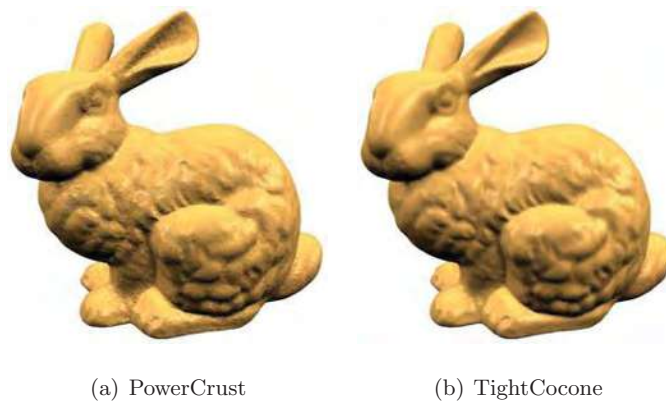


Figura 2.5: Reconstrucciones efectuadas con dos métodos tolerantes a bajas densidades en la muestra. Fuente: [Schall05].

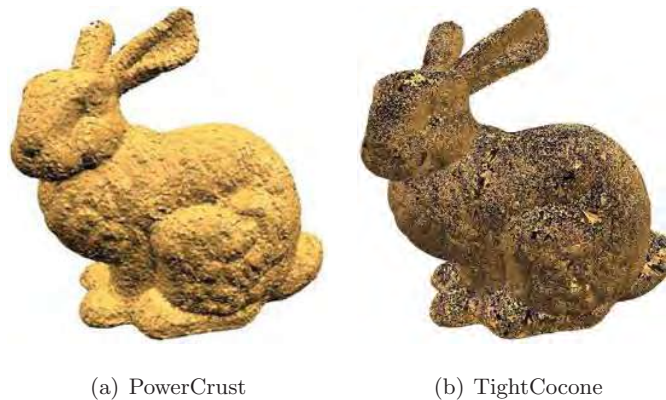


Figura 2.6: Reconstrucciones sobre datos con ruido utilizando dos algoritmos geométricos. Fuente: [Schall05].

2.2. Algoritmos volumétricos

Los métodos implícitos son una extensión de la idea de Blinn [Blinn82] de mezclar primitivas locales implícitas. En forma contraria a las técnicas que reconstruyen representaciones lineales a pedazos de X , éstos típicamente ajustan una función sobre todo el conjunto de datos. Al hacerlo, inherentemente están interpolando. Si se desea, la función puede ser aproximada para obtener una representación lineal (poligonal) a pedazos [Lorensen87].

Frecuentemente, la diferencia entre los distintos métodos implícitos es simplemente la elección de la función empleada para representar la superficie. Por ejemplo, Krishnamurthy y Levoy en [Krishnamurthy96] calculan vectores detallados de desplazamiento y ajustan superficies empleando B-splines [Watt99]. Lim [Lim95] hace un ajuste de los datos mediante la unión de esferas. Las esferas son configuradas inicialmente empleando una triangulación de Delaunay de X . Ohtake et al. [Ohtake03b] emplean funciones cuadráticas pesadas para un ajuste local multinivel. Bajaj et al. [Bajaj95a, Bajaj95b] emplean una versión modificada de parches de Bernstein-Bezier [Watt99]. Otros trabajos [Guo91, Dahmen93, Guo93] modelan la superficie empleando splines [Watt99].

2.2.1. Métodos de nivel cero

Una clase de técnicas implícitas aproxima los datos con una función de mejor empate. Una función es más propensa a ser elegida que otra si su conjunto de nivel cero, $Z(f)$, es más cercano al conjunto de puntos. En [Pratt87, Taubin91] minimizan la suma de las distancias cuadradas de Hausdorff del conjunto de puntos a $Z(f)$ donde f es un polinomio en tres variables. Muraki et al. [Muraki91] toman f como una combinación lineal de tres núcleos gaussianos. Además de la cercanía de f a 0 sobre los datos, su función de ajuste también mide que tan bien las normales unitarias a $Z(f)$ empatan a las normales estimadas a partir de los datos. Uno de los algoritmos sobresalientes en esta categoría es el propuesto por Hoppe et al. [Hoppe92]. Además de ser considerado fundamental en el área de reconstrucción de superficies, este algoritmo está fuertemente ligado al problema de la estimación de normales.

El algoritmo toma como entrada un conjunto de puntos X y produce como salida una superficie simplicial (ver Apéndice A.2.3) que aproxima a S . La topología (ver Apéndice A.2) de S es inferida directamente de los datos. El algoritmo consta de dos fases:

- En la primera fase, se define una función $f : D \rightarrow \mathbb{R}$, donde $D \subset \mathbb{R}^3$ es una región cerca de los datos, tal que f estima la distancia geométrica signada (ver Figura 2.7), a la superficie desconocida S . El conjunto $Z(f) = \{\mathbf{p} \mid f(\mathbf{p}) = 0\}$ es la estimación de S .
- En la segunda fase, emplea un algoritmo de contorno [Lorensen87] para aproximar S por una superficie simplicial. El paso clave para definir la función de distancia signada es asociar un plano orientado a cada uno de los puntos en el conjunto de datos, estos planos tangentes sirven como una aproximación local a la superficie.

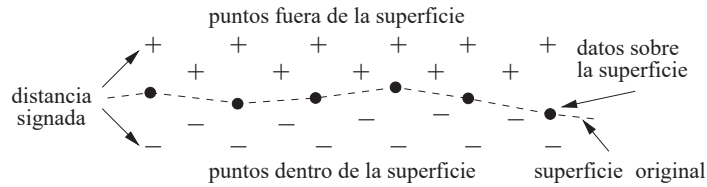


Figura 2.7: Distancia signada.

El plano tangente T_i , asociado con el punto $\mathbf{x}_i \in X$, está representado por un punto \mathbf{c}_i llamado centro y un vector normal unitario \mathbf{n}_i . Calculados de forma que el plano es el de mejor ajuste por mínimos cuadrados a los k -vecinos más cercanos a \mathbf{x}_i . La distancia con signo $ds(\mathbf{p})$ de un punto arbitrario $\mathbf{p} \in \mathbb{R}^3$ a la superficie S es la distancia entre el punto \mathbf{p} y el punto más cercano $\mathbf{z} \in S$, multiplicada por ± 1 , dependiendo del lado de la superficie en que se encuentra.

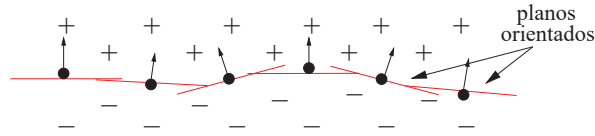


Figura 2.8: Planos tangentes.

Como S es desconocida, este procedimiento se simula con ayuda de los planos orientados (ver Figura 2.8). Se selecciona el plano tangente T_i cuyo centro \mathbf{c}_i es el más cercano a \mathbf{p} . Este plano tangente es una aproximación lineal local a S , por lo que se toma la distancia con signo $ds(\mathbf{p})$ a S como la distancia con signo entre \mathbf{p} y su proyección \mathbf{z} sobre T_i (ver Figura 2.9), esto es:

$$ds(\mathbf{p}) = (\mathbf{p} - \mathbf{c}_i) \cdot \mathbf{n}_i \quad (2.1)$$

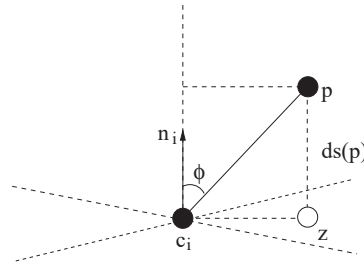


Figura 2.9: Proyección sobre el plano tangente.

Si S es una variedad (ver Apéndice A.2.4) sin fronteras esta regla funciona bien; sin embargo, debe extenderse al caso de variedades que pueden tener fronteras (ver Apéndice A.2.5). Supóngase que el conjunto X tiene una densidad de muestreo ρ_A y una magnitud δ_A para el error máximo de adquisición. Sea $d(\mathbf{p}, X)$ la distancia de Hausdorff de \mathbf{p} a X . Si no existe ruido, se puede deducir que un punto \mathbf{z} con $d(\mathbf{z}, X) > \rho_A$ no puede ser un punto de S dado que violará la condición de que X tiene densidad ρ_A . Los puntos de la muestra no dejan hoyos de un radio mayor que ρ_A . Si la muestra tiene una magnitud del error δ_A el radio de los hoyos puede incrementarse, pero no por más de δ_A .

Por lo tanto, se puede concluir que un punto \mathbf{z} no puede ser un punto de S si $d(\mathbf{z}, X) > \rho_A + \delta_A$. Si la proyección \mathbf{z} de \mathbf{p} sobre el plano tangente más cercano tiene $d(\mathbf{z}, X) > \rho_A + \delta_A$ se marca $ds(\mathbf{p})$ como indefinida. Los valores indefinidos son usados por el algoritmo de contorno para identificar las fronteras. El algoritmo crea un conjunto que es plano a pedazos pero contiene discontinuidades. Las discontinuidades resultan del particionamiento implícito del espacio en regiones dentro de las cuales un único plano tangente es usado para definir la función de distancia con signo. Afortunadamente, las discontinuidades no afectan al algoritmo.

El algoritmo de cubos en marcha (Marching Cubes) [Lorensen87] evalúa la función en la Ecuación (2.1) en los vértices de un malla cúbica y encuentra las intersecciones del contorno dentro de las celdas. En este trabajo, se emplea una variante propuesta por [Lewiner03], que elimina las ambigüedades del algoritmo original y garantiza una topología correcta del resultado. Para estimar de forma precisa las fronteras, el tamaño de los cubos se elige de forma tal que los lados tienen una longitud menor que $\rho_A + \delta_A$.

Hoppe et al. en [Hoppe92] sugieren establecer el tamaño de los cubos algo mayor que este valor, para incrementar la velocidad de ejecución y reducir el número de las caras triangulares generadas. El algoritmo visita únicamente los cubos que intersectan el conjunto cero, creando una malla con los cubos apropiados. De esta manera, la función de distancia con signo es evaluada únicamente en puntos cercanos a los datos. No se reporta ninguna intersección dentro de un cubo si la función de distancia con signo está indefinida en cualquier vértice del cubo, lo que genera las fronteras de la superficie simplicial. En la Figura 2.10 se muestran un par de reconstrucciones generadas con este método.

Curless y Levoy en [Curless96, Curless98] emplean una propuesta similar al algoritmo anterior ya que usan la función de distancia con signo a partir de los planos tangentes estimados. Bajaj et al. en [Bajaj95a] emplean formas α [Edelsbrunner83, Edelsbrunner92] para construir f , mientras Boissonnat y Cazals en [Boissonnat00] usan interpolación basada en los vecinos naturales los cuales pueden ser fácilmente calculados a partir del diagrama de Voronoi de X .

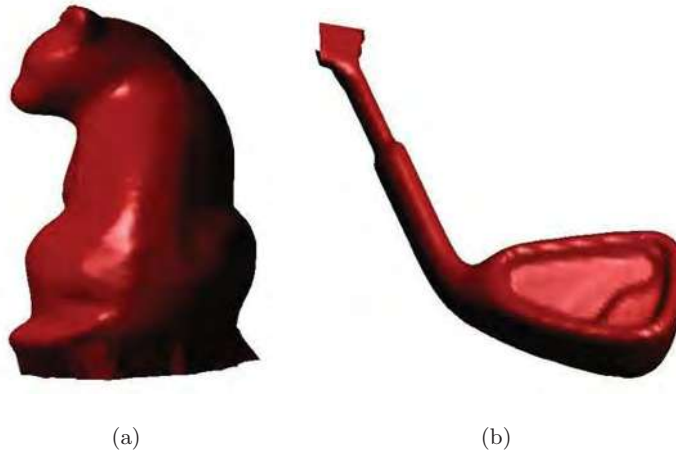


Figura 2.10: Reconstrucciones generadas con la propuesta de Hoppe et al.

2.2.2. Funciones de base radial

Otro tipo de algoritmos de reconstrucción de superficies emplean funciones de base radial (RBFs por sus siglas en inglés). A continuación, se describe el funcionamiento general de los algoritmos de reconstrucción basados en RBF.

Dado un conjunto de puntos, se interpola o aproxima por la superficie de nivel de una función compuesta $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ definida como una combinación lineal de primitivas simples.

$$f(\mathbf{p}_j) = \sum_{i=1}^m \beta_i \Phi(\mathbf{p}_j, \mathbf{p}_i) \quad (2.2)$$

donde $\Phi(\cdot, \mathbf{p}_i) : \mathbb{R}^3 \rightarrow \mathbb{R}$ son funciones centradas en \mathbf{p}_i y β_i son pesos desconocidos [Girosi93]. En las primeras propuestas los \mathbf{p}_i correspondían a los puntos \mathbf{x}_i de la muestra X . En las propuestas actuales, éstos se pueden obtener a partir de un proceso de selección [Carr01].

Comúnmente se restringe a soluciones estables a translaciones y rotaciones del conjunto de puntos, por lo tanto, las funciones Φ están dadas por:

$$\Phi(\mathbf{p}_j, \mathbf{p}_i) = \phi(\|\mathbf{p}_j - \mathbf{p}_i\|) \quad (2.3)$$

donde $\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$.

En esta sección se presentarán varias de las funciones ϕ convencionales y se discutirán algunas de sus limitaciones. La reconstrucción basada en funciones de base radial produce una superficie implícita suave cuyo conjunto de nivel cero tiene las mismas propiedades de continuidad de las funciones base Φ .

Sea $F = [f_i]$ el vector con los n valores de la función f sobre el conjunto de puntos X . Se quiere encontrar una función $f : \mathbb{R}^3 \rightarrow \mathbb{R}$ tal que $\forall j = 1, \dots, n$

$$f(\mathbf{x}_j) = \sum_{i=1}^n \beta_i \phi(\|\mathbf{x}_j - \mathbf{x}_i\|) = f_j \quad (2.4)$$

El problema de la reconstrucción se reduce a determinar un vector $\beta = \{\beta_1, \dots, \beta_n\}$ resolviendo un sistema lineal de ecuaciones dado por la restricción de la Ecuación (2.4). Debido a que todas las restricciones están localizadas sobre la superficie, todas las f_i tienen valor cero. Para evitar la solución trivial $\beta = \vec{0}$, se agregan restricciones interiores y exteriores donde la función no es cero y se asignan a ellas los valores $-d$ y d , respectivamente. Carr et al. [Carr01] proponen un método para agregar estas restricciones. Los pesos $\beta = [\beta_i]$ se calculan empleando la Ecuación (2.4) y denotando $[\phi(\|\mathbf{x}_i - \mathbf{x}_j\|)] = A_{X,\Phi}$, se tiene que resolver el siguiente sistema lineal:

$$A_{X,\Phi} \cdot \beta = F \quad (2.5)$$

Para obtener una solución única, la matriz $A_{X,\Phi}$ se requiere que sea invertible, al menos sobre el subespacio de β vectores donde se busca la solución. Una solución común es usar un subespacio tal que:

$$\forall \beta \in \mathbb{R}^n \quad \sum_{i=1}^n \beta_i P(\mathbf{x}_i) = 0 \quad \forall P \in \mathbb{P}^q \quad (2.6)$$

donde \mathbb{P}^q es el conjunto de polinomios de orden superior a q . Con esta restricción, las funciones ϕ son condicionadas a ser positivas definidas [Buhmann03]. Además, para recuperar el número correcto de variables, se agrega a la Ecuación (2.2) un polinomio $P \in \mathbb{P}^q$

$$f(\mathbf{x}_j) = \sum_{i=1}^n \beta_i \Phi(\mathbf{x}_j, \mathbf{x}_i) + P(\mathbf{x}_j) \quad (2.7)$$

Algunas funciones convencionales de base radial son:

$$\phi : \mathbb{R}^+ \rightarrow \mathbb{R}$$

- biarmónica $\phi(r) = r$ con un polinomio lineal
- pseudocúbica $\phi(r) = r^3$ con un polinomio lineal
- triarmónica $\phi(r) = r^3$ con un polinomio cuadrático
- placa fina $\phi(r) = r^2 \log(r)$ con un polinomio lineal

Las funciones anteriores tienen soporte no acotado⁴. Las ecuaciones correspondientes conducen a un sistema lineal denso, siendo posible recuperar una solución únicamente para conjuntos de puntos pequeños. Para resolver este problema Morse et al. [Morse01] hacen uso de funciones gaussianas como funciones de base radial con soporte compacto para obtener una matriz dispersa de interpolación. Para trabajar con conjuntos de puntos grandes o incompletos un par de estrategias han sido desarrolladas. La propuesta descrita en [Carr01] emplea funciones de base radial poliarmónicas (i.e. funciones sin soporte compacto), reduce el número de centros en un proceso de selección y ejecuta una evaluación rápida usando el Método Rápido de Multipolo [Greengard87]. Otra propuesta, consiste en usar funciones con soporte local [Tobor04], la partición de unidad es empleada para mezclar las funciones y el soporte de la función es calculado localmente para todos los centros, como se describe en [Ohtake04]. Una versión con multiescala de este método fue propuesta en [Ohtake03c].

2.2.3. Partición de unidad

Algunas de las propiedades deseables de un método de reconstrucción de superficies son velocidad, baja carga de memoria, una reproducción de características como bordes y esquinas y tolerancia ante la presencia de hoyos y una baja densidad de muestreo.

⁴Se dice que una función tiene soporte compacto si el conjunto donde no es nula conforma un conjunto cerrado y acotado. Por ejemplo, si se tiene una función $f(x)$ cualquiera, se define el soporte de ésta como sigue: $\text{supp}(f) = \{x \in \mathbf{R} \mid f(x) \neq 0\}$. Si el supp de f es un conjunto cerrado y acotado, entonces es compacto.

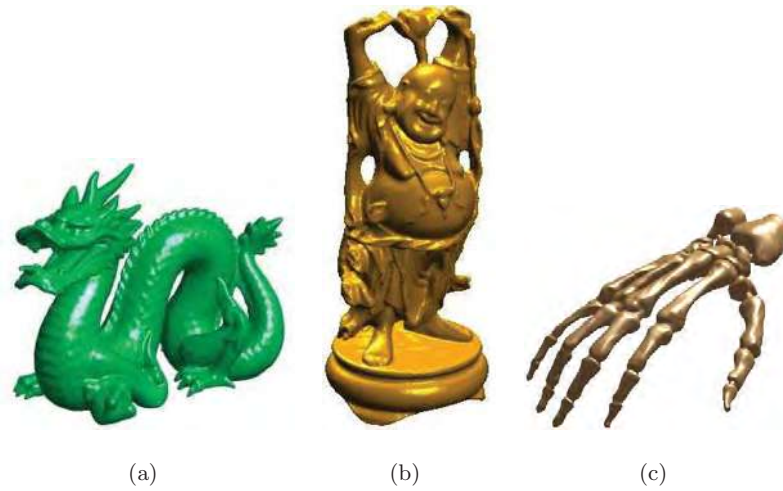


Figura 2.11: Ejemplo de reconstrucciones empleando funciones de base radial. Fuente: [Carr01].

Ohtake et al. [Ohtake03b] desarrollaron un método que nombran Partición Multi-nivel de Implícitas Unitarias (Multilevel Partition of Unity Implicits, MPUI por sus siglas en inglés), diseñado para cumplir estos requisitos y crear de forma rápida y precisa superficies a partir de grandes conjuntos de puntos. La superficie es aproximada por el conjunto de nivel cero de la función de distancia.

El MPUI está basado en una técnica conocida como Partición de Unidad y la idea principal consiste en dividir el dominio del problema en subdominios más pequeños donde el problema puede ser resuelto localmente. Primeramente, los datos son aproximados por separado en cada subdominio y posteriormente las soluciones son unificadas empleando una suma pesada de aproximaciones locales. Los pesos son funciones suaves y suman uno sobre todo el dominio.

Considérese un dominio global acotado Ω en un espacio euclídeo. Se divide Ω en M subdominios con un pequeño traslape $\{\Omega_i\}_{i=1}^M$ con $\Omega \subseteq \cup_i \Omega_i$. Por un pequeño traslape se entiende que la intersección de dos subdominios incidentes contienen a lo más un punto. Junto con esta cobertura se construye una partición de unidad, ésto es, una familia de funciones continuas no negativas con soporte compacto $\{\omega_i\}_{i=1}^M$ tal que el $Supp(\omega_i) \subseteq \Omega_i$ y

$\sum_{i=1}^M \omega_i = 1$ en todos lados. Sea X un conjunto de n puntos sobre la superficie. Para cada celda Ω_i , se construye un conjunto $X_i = \{\mathbf{x}_i \in X \mid \mathbf{x}_i \in \Omega_i\}$ y la superficie es aproximada en cada subdominio por una función local f_i . La función global está definida por una combinación de funciones locales como:

$$f(\mathbf{x}_i) = \sum_{i=1}^n \omega_i(\mathbf{x}_i) f_i(\mathbf{x}_i) \quad (2.8)$$

La condición $\sum_{i=1}^M \omega_i = 1$ puede obtenerse a partir de cualquier otro conjunto de funciones suaves W_i por un proceso de normalización:

$$\omega_i(\mathbf{x}_i) = \frac{W_i(\mathbf{x}_i)}{\sum_{j=1}^n W_j(\mathbf{x}_i)} \quad (2.9)$$

La función de peso W_i determina la continuidad de la función de reconstrucción global f . Estas funciones pueden ser generadas usando la geometría local de la celda correspondiente (función de distancia, centro y radio de la celda, etc.). Para la descomposición del dominio generalmente se propone una subdivisión del espacio basada en árboles (binarios, octrees [Watt99], etc.). Las celdas pueden tener formas diferentes como esferas, cubos [Ohtake03b] o elipsoides.

La elección del método de ajuste local es otro grado de libertad. Algunos métodos emplean funciones de base radial [Tobor04] o cuadráticas [Ohtake03b]. Es posible adaptar la estrategia de ajuste para cada celda de acuerdo al número de puntos y distribución de las normales asociadas. De esta manera es posible reconstruir superficies con bordes o esquinas. Ohtake en [Ohtake03b] emplea la función más apropiada de las tres aproximaciones locales siguientes:

1. Un cuádrica general en 3D
2. Un polinomio cuadrático bivariado en la coordenadas locales
3. Un superficie cuadrática a pedazos que ajusta un borde o esquina

Un conjunto de pruebas simples son ejecutadas para realizar la selección de la función adecuada [Ohtake03b]. Es importante destacar que la complejidad del método es sensitiva a la salida en el sentido que el tiempo de creación y el consumo de memoria dependen de la complejidad de la superficie reconstruida y no del número de puntos en la muestra. En la Figura 2.12 se muestran algunas reconstrucciones generadas con este método. Notése como es capaz de generar bordes bien definidos.

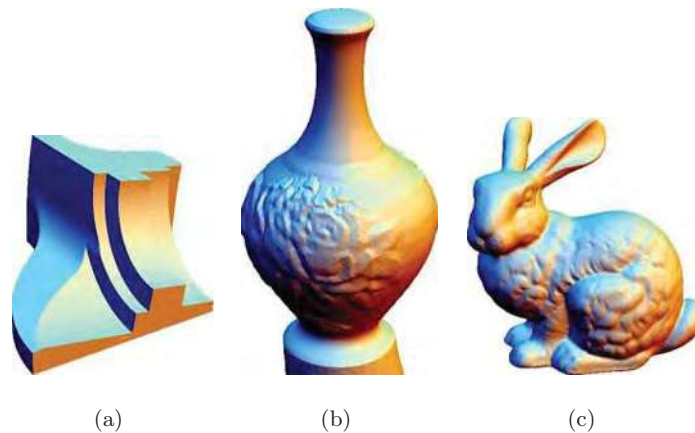


Figura 2.12: Ejemplo de reconstrucciones con el método MPUI. Fuente: [Ohtake03b].

2.2.4. Mínimos cuadrados móviles

Las superficies basadas en mínimos cuadrados móviles fueron presentadas por Levin [Levin98]. Estas superficies están definidas por un operador que proyecta los puntos en un vecindad tubular sobre la superficie. La vecindad se define usualmente como una unión de bolas centradas en los puntos de entrada X . Un resumen de las técnicas existentes que incluye además una breve descripción del operador tradicional de proyección puede encontrarse en [Kobbelt04].

Adamson y Alexa [Adamson03] proponen un técnica simple de proyección para la definición de una superficie implícita a partir del conjunto de puntos. Iterativamente proyectan un punto \mathbf{p} sobre un plano definido por un promedio pesado de los puntos vecinos

$$\mathbf{a}(\mathbf{p}) = \frac{\sum_j \Phi(\|\mathbf{p} - \mathbf{x}_j\|) \mathbf{x}_j}{\sum_j \Phi(\|\mathbf{p} - \mathbf{x}_j\|)} \quad (2.10)$$

y la normal $\mathbf{n}(\mathbf{p})$. Si las normales \mathbf{n}_j para los puntos de entrada son dadas $\mathbf{n}(\mathbf{p})$ puede ser calculada promediando las normales de entrada

$$\mathbf{n}(\mathbf{p}) = \frac{\sum_j \Phi(\|\mathbf{p} - \mathbf{x}_j\|) \mathbf{n}_j}{\|\sum_j \Phi(\|\mathbf{p} - \mathbf{x}_j\|) \mathbf{n}_j\|} \quad (2.11)$$

en caso contrario, la normal puede ser determinada con la normal al plano de ajuste por mínimos cuadrados pesados de los puntos \mathbf{x}_j . Después de converger el punto \mathbf{p} alcanza una posición sobre la superficie por mínimos cuadrados móviles.

Amenta y Kil [Amenta04] analizan la estabilidad del operador de proyección original para los puntos que no están lo suficientemente cerca de la superficie por mínimos cuadrados móviles. Además, presentan una definición explícita de las superficies anteriores en términos de los puntos críticos de una función de energía E_{MLS} sobre líneas determinadas por un campo vectorial.

2.2.5. Técnicas basadas en conceptos físicos y modelos deformables

Las técnicas basadas en modelos deformables hacen crecer/decrecer una malla inicial sujeta a algunas restricciones hasta que ésta rodea satisfactoriamente a X . Sclaroff y Pentland [Pentland91] describen un método para ajustar un conjunto de puntos deformando una supercuádrica [Watt99]. Kobbelt et al. [Kobbelt99] simulan la envoltura del objeto con una membrana plástica. Liao y Medioni en [Liao95] emplean una versión modificada de splines bidimensionales a los cuales llaman B-snakes [Brigger00].

Los métodos basados en conceptos físicos describen un término de energía cuya minimización controla el crecimiento inicial del modelo. Varios términos de energía miden la cercanía del modelo a X , su suavidad y otros atributos. Hoppe et al. en [Hoppe93] deforman una malla inicial bajo estas condiciones. Chen y Medioni en [Chen95] inicializan sus mallas con un modelo simple de un globo que está contenido totalmente en X . Posteriormente,

inflan el globo hasta llegar a X . Sethian [Sethian99] y Zhao et al. [Zhao00] emplean métodos de conjunto de nivel variacionales para deformar una membrana rodeando a X bajo ciertas condiciones de energía.

2.3. Conclusiones

En este capítulo, se presentó un repaso de las dos corrientes principales en el área de la reconstrucción de superficies a partir de conjuntos de puntos.

La primera corriente está respaldada bajo resultados matemáticos rigurosos pero solo trabaja adecuadamente en los casos específicos de superficies suaves y ausencia de ruido en la muestra. Es incapaz de garantizar sus resultados para muestras con ruido y superficies con bordes, problemas típicos con el uso de modelos reales. Otro problema es la complejidad de los algoritmos en que está basada, imposibilitando el manejo de conjuntos con millones de puntos, comunes en la actualidad.

La segunda corriente introduce una mayor flexibilidad para generar la reconstrucción. Es capaz de manejar conjuntos de puntos adquiridos de modelos reales. Esta flexibilidad está supeditada a un requisito adicional en el conjunto de entrada, la información sobre las normales asociadas a la superficie descrita por el conjunto de puntos. Estas normales forman un rol central en la reconstrucción, ya que son necesarias para definir una orientación de la superficie y en muchas ocasiones calcular una función de distancia de puntos en el espacio a la superficie.

La propuesta presentada por Ohtake et al. basada en la Partición de Unidad, se distingue por manejar adecuadamente los bordes y esquinas, problema que ha sido reportado frecuentemente en la literatura. Esta característica aunada a su velocidad de ejecución y un uso eficiente de memoria, la convierte en uno de los métodos considerados como parte del estado del arte en reconstrucción de superficies. En esa propuesta el rol de las normales va más allá de definir la orientación de la superficie, al ser usadas para determinar el tipo de función de ajuste a utilizar. De lo anterior, se concluye que la estimación correcta de

las normales asociadas al conjunto de puntos juegan un rol preponderante en el área de la reconstrucción de superficies. En el próximo capítulo, se presenta un resumen de las técnicas y problemas de los métodos existentes para calcular normales a partir de conjuntos de puntos.

Capítulo 3

Métodos para la estimación de normales

La calidad de la aproximación de la superficie de salida con los métodos actuales [Ohtake03b] de reconstrucción de superficies, depende de la precisión con que las normales calculadas a partir del conjunto de puntos aproximan a las verdaderas en la superficie original. Al ser parte central en el problema de la reconstrucción de superficies, la estimación de normales merece atención especial por sí misma.

Como lo mencionan Dey et al. [Dey05], existen dos propuestas dominantes para estimar normales a partir de conjuntos de puntos en 3D, una numérica [Hoppe92, Mitra03, Pauly03], basada en mínimos cuadrados y análisis de componentes principales y la otra en su mayor parte geométrica [Amenta98c, Dey06], basada en diagramas de Voronoi. La propuesta numérica es eficiente permitiendo trabajar con grandes cantidades de puntos, la otra por el contrario, es costosa, lo que limita sus posibilidades en aplicaciones con modelos reales. Este trabajo se centra en la propuesta numérica, donde la técnica ampliamente utilizada consiste en encontrar un conjunto de puntos en la vecindad de cada punto \mathbf{x} y entonces calcular el plano de mejor ajuste a los puntos seleccionados; la normal en cada plano se toma como una aproximación de la normal en cada punto. En las siguientes secciones se da una descripción más detallada de la técnica para el cálculo de normales además de algunas de las modificaciones que fueron propuestas para hacerla más efectiva.

3.1. Cálculo de las normales

Hoppe et al. [Hoppe92] desarrollaron uno de los primeros métodos para el cálculo de normales a partir de conjuntos de puntos, el cual continúa siendo ampliamente utilizado. Definen un plano tangente T_i para cada punto \mathbf{x}_i en X . El plano T_i está representado por un punto \mathbf{c}_i llamado centro y un vector normal unitario \mathbf{n}_i (ver Figura 3.1). El centro y el vector normal unitario para el plano T_i son determinados empleando los k vecinos más cercanos de X a \mathbf{x}_i . Este conjunto está denotado por V_i y es llamado la vecindad de \mathbf{x}_i .

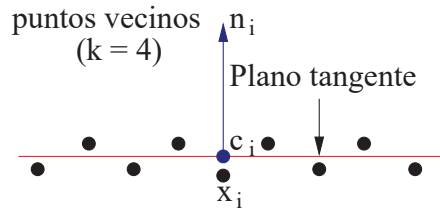


Figura 3.1: Plano tangente.

El centro y la normal unitaria se calculan de forma que el plano T_i corresponde al ajuste por mínimos cuadrados de V_i . El centro se toma como el centroide de V_i y la normal \mathbf{n}_i se determina usando análisis de componentes principales. Para estimar \mathbf{n}_i , se calcula la matriz de covarianza C_i de V_i , que es una matriz simétrica de 3×3 semidefinida positiva.

$$C_i = \sum_{j \in V_i} (\mathbf{x}_j - \mathbf{c}_i) \otimes (\mathbf{x}_j - \mathbf{c}_i) \quad (3.1)$$

donde \otimes denota el operador del producto externo vectorial¹. Sean $\mathbf{x}_j = (x_j, y_j, z_j)$ y $\mathbf{c}_i = (\bar{x}_i, \bar{y}_i, \bar{z}_i)$ las componentes correspondientes de ambos vectores, entonces la Ecuación 3.1 toma la forma:

¹Si \mathbf{v} y \mathbf{u} tienen componentes v_i y u_i respectivamente, entonces la matriz $\mathbf{v} \otimes \mathbf{u}$ tiene a $u_i v_j$ como su ij -ésimo elemento.

$$C_i = \begin{pmatrix} \sum_{j \in V_i} (x_j - \bar{x}_i)^2 & \sum_{j \in V_i} (x_j - \bar{x}_i)(y_j - \bar{y}_i) & \sum_{j \in V_i} (x_j - \bar{x}_i)(z_j - \bar{z}_i) \\ \sum_{j \in V_i} (y_j - \bar{y}_i)(x_j - \bar{x}_i) & \sum_{j \in V_i} (y_j - \bar{y}_i)^2 & \sum_{j \in V_i} (y_j - \bar{y}_i)(z_j - \bar{z}_i) \\ \sum_{j \in V_i} (z_j - \bar{z}_i)(x_j - \bar{x}_i) & \sum_{j \in V_i} (z_j - \bar{z}_i)(y_j - \bar{y}_i) & \sum_{j \in V_i} (z_j - \bar{z}_i)^2 \end{pmatrix} \quad (3.2)$$

Si $\lambda_i^1 \geq \lambda_i^2 \geq \lambda_i^3 \geq 0$ son los valores propios de la matriz en la Ecuación (3.2), asociados con los vectores propios $\mathbf{v}_i^1, \mathbf{v}_i^2$ y \mathbf{v}_i^3 respectivamente (ver Figura 3.2), se elige \mathbf{n}_i como \mathbf{v}_i^3 o $-\mathbf{v}_i^3$. La Figura 3.3 muestra un ejemplo de las normales calculadas por el método anterior para un conjunto de puntos que representan una superficie rectangular.

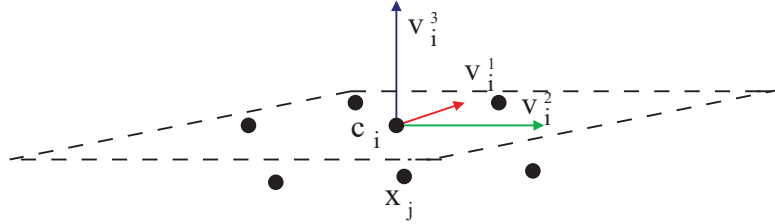


Figura 3.2: Vectores propios.

En la Figura 3.4 se hace un acercamiento de la esquina superior izquierda del conjunto de puntos presentado en la Figura 3.3, esto facilita apreciar un par de problemas que surgen al utilizar el algoritmo descrito en esta sección. El primero, que las normales no tienen una orientación globalmente consistente. Hoppe et al. [Hoppe92] proponen un algoritmo para obtener consistencia en las orientaciones, esta propuesta tiene amplia aceptación en la actualidad y será descrita en la siguiente sección. El segundo problema ya fue presentado en el Capítulo 1 y consiste en la estimación de las normales cerca de los bordes y esquinas. En el Capítulo 4, se hace una propuesta que mejora los resultados del algoritmo descrito a lo largo de esta sección.

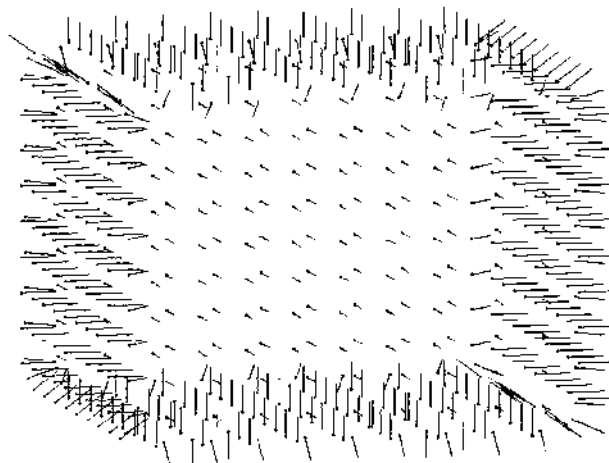


Figura 3.3: Normales calculadas a partir de un conjunto de puntos.

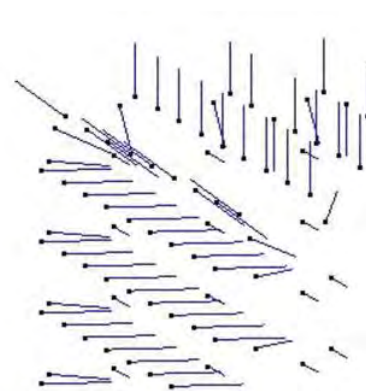


Figura 3.4: Acercamiento sobre la esquina superior izquierda de la Figura 3.3.

3.2. Planteamiento alternativo

Gopi et al. [Gopi00] presentan un planteamiento alternativo para la estimación de normales; proponen que el vector normal \mathbf{n}_i es el vector que minimiza la varianza del producto punto entre si mismo y los vectores de \mathbf{x}_i a sus k -vecinos más cercanos. Si los k -vecinos más cercanos son $\mathbf{x}_1, \dots, \mathbf{x}_k$, entonces los vectores de \mathbf{x}_i a sus k -vecinos más cercanos son $\mathbf{v}_j = \mathbf{x}_j - \mathbf{x}_i$, $1 \leq j \leq k$. Se desea encontrar \mathbf{n}_i tal que minimiza

$$\frac{\sum_{l=1}^k (D_l - \frac{\sum_{m=1}^k D_m}{k})^2}{k} \quad (3.3)$$

donde $D_l = \mathbf{n}_i \cdot \mathbf{v}_l$ y $D_m = \mathbf{n}_i \cdot \mathbf{v}_m$.

Los vectores \mathbf{v}_j pueden ser vistos como las coordenadas de los k -vecinos más cercanos con \mathbf{x}_i como el origen. Eliminando el factor de escala $\frac{1}{k}$ de la Ecuación (3.3), se obtiene

$$\min \left\{ \sum_{l=1}^k \left(\mathbf{n}_i \cdot \mathbf{v}_l - \frac{\sum_{m=1}^k \mathbf{n}_i \cdot \mathbf{v}_m}{k} \right)^2 \right\} \quad (3.4)$$

factorizando

$$\min \left\{ \sum_{l=1}^k \left[\left(\mathbf{v}_l - \frac{\sum_{m=1}^k \mathbf{v}_m}{k} \right) \cdot \mathbf{n}_i \right]^2 \right\} \quad (3.5)$$

Si \mathbf{x}_i está en el origen, el centroide de los k -vecinos más cercanos es $\mathbf{c}_i = \frac{\sum_{m=1}^k \mathbf{v}_m}{k}$. Por lo tanto, la Ecuación (3.5) puede ser reescrita como,

$$\min \left\{ \sum_{l=1}^k \left[(\mathbf{v}_l - \mathbf{c}_i) \cdot \mathbf{n}_i \right]^2 \right\} \quad (3.6)$$

Si A es la matriz de $k \times 3$ donde $\mathbf{v}_l - \mathbf{c}_i$ define los vectores fila, entonces la Ecuación (3.6) se reduce a

$$\min(\|A\mathbf{n}_i\|) \quad (3.7)$$

El problema de minimización puede ser planteado como un problema estándar de descomposición en valores singulares. El vector propio que corresponde al valor propio más pequeño de A es el vector normal que minimiza la ecuación anterior.

Así que a pesar de que el planteamiento es diferente a la propuesta en [Hoppe92], resulta que las normales calculadas por ambos métodos son las mismas.

3.3. Orientación consistente

Supóngase, que dos puntos $\mathbf{x}_i, \mathbf{x}_j \in X$ son cercanos geoméricamente. Idealmente, si el conjunto X es denso y la superficie es suave, los planos tangentes correspondientes T_i y T_j son casi paralelos, esto es, $\mathbf{n}_i \cdot \mathbf{n}_j \approx \pm 1$. Si los planos están orientados consistentemente entonces $\mathbf{n}_i \cdot \mathbf{n}_j \approx 1$, en otro caso, ya sea \mathbf{n}_i ó \mathbf{n}_j debe invertirse. La dificultad en encontrar una orientación globalmente consistente es que esta condición debe mantenerse entre todos los pares de puntos suficientemente cercanos.

Hoppe et al. modelan este problema como la optimización de un grafo. El grafo contiene un nodo N_i por plano tangente T_i , con una arista (i, j) entre N_i y N_j si los centros \mathbf{c}_i y \mathbf{c}_j de los planos tangentes son lo suficientemente cercanos. El costo en cada arista (i, j) codifica el grado de consistencia en la orientación de N_i y N_j y se toma como $\mathbf{n}_i \cdot \mathbf{n}_j$. El problema radica en seleccionar las orientaciones de los planos tangentes tales que se maximiza el costo total del grafo, dado como la suma de los costos de cada arista. Desafortunadamente, en [Hoppe95] muestran que este problema es *NP*-completo y para resolverlo plantean una aproximación.

El primer paso es decidir cuando un par de nodos están conectados en el grafo. Dado que la superficie se supone que consiste de un único componente conectado, es decir, es posible ir de un punto a otro en la superficie sin salir de ella, el grafo también debe estarlo. Un grafo simple que está conectado para un conjunto de puntos y que tiende a conectar a los vecinos es el Árbol Euclidiano de Expansión Mínima (EMST) (ver Figura 3.5).

En [Hoppe92] afirman que el EMST sobre los centros $\mathbf{c}_1, \dots, \mathbf{c}_n$ de los planos tangentes no tiene una cantidad suficientemente de aristas para la orientación. Por lo tanto, lo enriquecen añadiéndole un número determinado de aristas. Específicamente, se añaden una arista (i, j) si \mathbf{c}_i está en la k -vecindad de \mathbf{c}_j ó \mathbf{c}_j está en la k -vecindad de \mathbf{c}_i . El grafo resultante (ver Figura 3.6) es llamado Grafo Riemanniano y está diseñado para ser un

grafo conectado que codifica la proximidad geométrica de los centros de los planos tangentes.

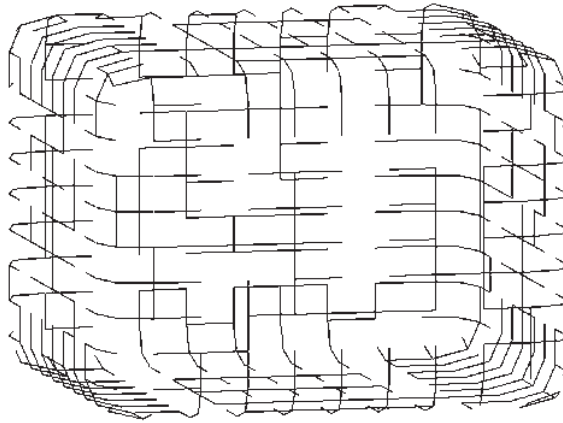


Figura 3.5: Árbol Euclidiano de Expansión Mínima.

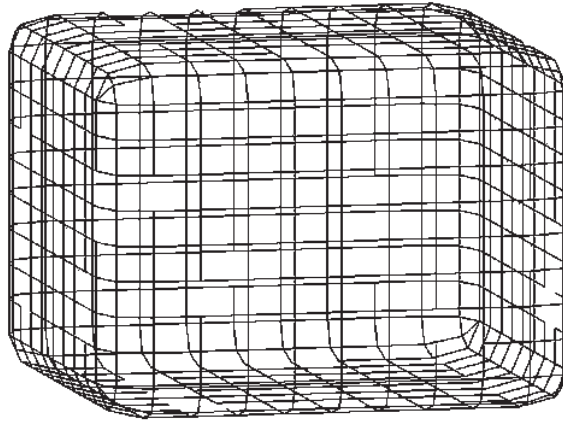


Figura 3.6: Grafo Riemanniano calculado de acuerdo al método propuesto por Hoppe et al.

Intuitivamente, se desea elegir un recorrido que favorezca la propagación de T_i a T_j si los planos tangentes no orientados son casi paralelos. En [Hoppe92], asignan a cada arista (i, j) en el Grafo Riemanniano el costo $1 - |\mathbf{n}_i \cdot \mathbf{n}_j|$. Además de ser no negativa, esta asignación tiene la propiedad de que el costo es pequeño si los planos tangentes no orientados son casi paralelos. Un orden de propagación favorable puede obtenerse recorriendo el Árbol de Expansión Mínima (MST) del Grafo Riemanniano.

De acuerdo con [Hoppe92], para asignar una orientación a un plano inicial, la normal unitaria del plano cuyo centro tiene la coordenada mayor en z es forzada a dirigirse hacia el eje z positivo. Tomando este nodo como inicial, se recorre el árbol en profundidad, asignando a cada plano una orientación que es consistente con la de su padre. Si durante el recorrido, al plano actual T_i le ha sido asignada la orientación \mathbf{n}_i y T_j es el siguiente plano a visitar, entonces \mathbf{n}_j es reemplazada con $-\mathbf{n}_j$ si $\mathbf{n}_i \cdot \mathbf{n}_j < 0$. La Figura 3.7 muestra el resultado de aplicar el algoritmo de orientación a las normales presentadas en la Figura 3.3.

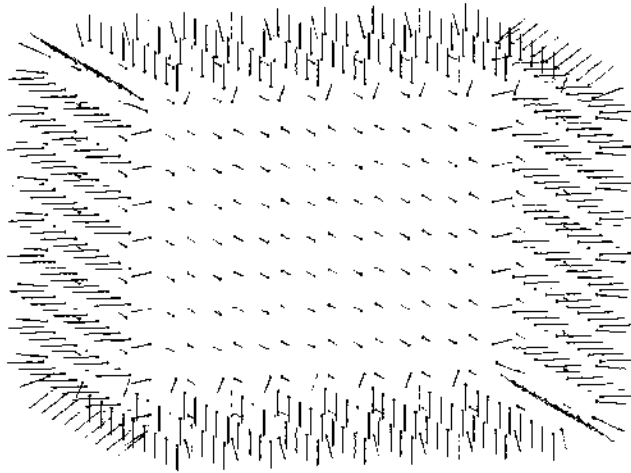


Figura 3.7: Normales orientadas de acuerdo al método propuesto por Hoppe et al.

3.4. Ajuste con peso

Pauly et al. [Pauly03] sugieren que el plano de ajuste T_i debe dar más importancia a los puntos de V_i más cercanos a \mathbf{x}_i . Por lo tanto en [Pauly03], a los puntos $\mathbf{x}_j \in V_i$ les asignan distintos pesos basados en sus distancias a \mathbf{x}_i . Entre más pequeña sea la distancia de un punto \mathbf{x}_j a \mathbf{x}_i , mayor es el peso que tiene. La función de peso está dada por

$$w(\|\mathbf{x}_j - \mathbf{x}_i\|) = e^{-\frac{\|\mathbf{x}_j - \mathbf{x}_i\|^2}{h^2}} \quad (3.8)$$

donde h^2 se elige como un tercio de la distancia al cuadrado entre \mathbf{x}_i y el elemento en V_i más lejano a este. La matriz de covarianza toma la forma

$$C_i^* = \sum_{j \in V_i} w(\|\mathbf{x}_j - \mathbf{x}_i\|) [(\mathbf{x}_j - \mathbf{c}_i) \otimes (\mathbf{x}_j - \mathbf{c}_i)] \quad (3.9)$$

El cálculo del vector normal a partir de C_i^* se realiza de manera análoga a [Hoppe92]. Este algoritmo mejora la estimación de la normal cuando la densidad de los puntos no es uniforme. Sin embargo, es posible observar que falla al calcular normales cercanas a bordes y esquinas, ya que no hace una verificación de la pertenencia de los elementos a un plano único.

3.5. Ajuste adaptativo

Mitra et al. [Mitra03] notaron que una selección adecuada del valor de k es crucial para obtener una buena estimación de la normal. Tomar el mismo valor de k para todos los puntos como en [Hoppe92] puede perjudicar el ajuste, especialmente en lugares donde las muestras son arbitrariamente densas. Por lo tanto, en lugar de usar los k vecinos del punto \mathbf{x}_i , consideran las muestras dentro de una esfera con un cierto radio r . Bajo la suposición de que el ruido tiene media cero y desviación estándar σ_n , ellos obtienen una cota sobre el ángulo entre la normal estimada y la normal verdadera con una probabilidad casi uno. El radio que puede obtenerse minimizando esa cota tiene la siguiente expresión en el caso tridimensional siempre que la probabilidad sea $1 - \epsilon$.

$$r = \left(\frac{1}{\kappa_L} \left(c_1 \frac{\sigma_n}{\sqrt{\epsilon \rho_L}} + c_2 \sigma_n^2 \right) \right)^{\frac{1}{3}} \quad (3.10)$$

donde ρ_L es la densidad local de muestreo, κ_L es la curvatura local, c_1 y c_2 son un par de constantes.

Mitra et al. asumen que la desviación estándar σ_n del ruido es conocida y las constantes c_1 y c_2 dependen de la distribución de muestreo del conjunto de puntos. En [Mitra03] no se presenta un método para calcular los valores exactos de c_1 y c_2 y simplemente los suponen. El valor de ϵ lo fijan en 0.1. Dado un conjunto de puntos, estiman la densidad local de muestreo de la siguiente manera:

Para un punto \mathbf{x}_i dado, calculan la distancia s de \mathbf{x}_i a su k -vecino más cercano para un valor pequeño de k , en sus experimento $k = 15$. La densidad local de muestreo en \mathbf{x}_i es aproximada como $\rho_L = k/(\pi s^2)$ muestras por unidad de área. Para estimar la curvatura local, emplean el método propuesto en [Gumhold01]. Sean $\mathbf{x}_j \in V_i$ los k -vecinos más cercanos a \mathbf{x}_i y sea ν la distancia promedio de \mathbf{x}_i a todos los puntos \mathbf{x}_j . Calculan el plano de mejor ajuste para esos k -vecinos y la distancia d de \mathbf{x}_i al plano de mejor ajuste como en [Hoppe92]. La curvatura local en \mathbf{x}_i puede ser aproximada como $\kappa_L = 2d/\nu^2$. Una vez que todos los parámetros fueron calculados, se obtiene el valor de r usando la Ecuación (3.10).

Nótese que el valor de r puede ser usado para obtener un mejor valor de k que sirve para estimar nuevamente la densidad local y la curvatura local. Esto sugiere un esquema iterativo en cual se pueden estimar repetidamente la densidad local, la curvatura local y el tamaño de la vecindad. En [Mitra03] afirman que tres iteraciones son suficientes para obtener buenas aproximaciones para todas las cantidades.

3.6. Conclusiones

En este capítulo se han descrito las técnicas existentes para la estimación de normales a partir de conjuntos de puntos. La idea detrás de ellas es la creación de vecindades para cada punto y el ajuste de un plano en cada vecindad, la normal asociada a ese plano se toma como una aproximación a la normal en el punto. Sin embargo, como se hizo notar, esta técnica tiene problemas cerca de bordes y esquinas, los cual se derivan de un ambigüedad en la creación de las vecindades, ya que éstas pueden incluir puntos que pertenecen a planos diferentes en esas regiones. En la actualidad, las vecindades son creadas únicamente a partir de la distancia euclidiana entre los puntos sin que se tomen en cuenta otras características de la geometría del conjunto. En el siguiente capítulo, se presenta una propuesta para la creación de las vecindades que emplea información adicional. Las vecindades calculadas permiten obtener una mejor estimación de la normal cerca de los bordes y esquinas, la cual se ve traducida en una mejora significativa de la calidad de la reconstrucción cerca de esas regiones.

Capítulo 4

Propuesta de estimación de normales con reorganización de vecindades

Los métodos descritos en el capítulo anterior obtienen resultados satisfactorios en regiones suaves. Sin embargo, ante la presencia de bordes y esquinas [Gumhold01], las normales calculadas generalmente son incorrectas. Esto repercute directamente en la calidad de la reconstrucción de la superficie ya que se obtienen bordes y esquinas que lucen redondeadas. El problema radica en que el conjunto V_i puede contener puntos en dos planos diferentes si \mathbf{x}_i está dentro o cerca de un borde y en tres planos si \mathbf{x}_i está dentro o cerca de una esquina. El método que se presenta en este capítulo obtiene una nueva estimación de la normal para cada punto \mathbf{x}_i , seleccionando los elementos $\mathbf{x}_j \in V_i$ que están únicamente en el mismo plano que \mathbf{x}_i . El algoritmo calcula la aproximación a partir de las normales en la vecindad, limitando la influencia de aquellas normales que pueden ser consideradas atípicas. Lo anterior permite identificar y manejar regiones cercanas a bordes y esquinas.

4.1. Sistema de resortes con condición de reposo adaptable

Para mejorar la estimación de las normales se propone una técnica de regularización [Neumaier98] con una función de energía basada en un sistema de resortes con condición de reposo adaptable (Adaptive-Rest Condition, ARC). La idea se origina del trabajo presentado por Rivera y Marroquin en [Rivera00], siendo utilizada en el cálculo del flujo óptico en [Calderon03] y registro de imágenes en [Calderon06b].

El sistema de resortes basado en ARC tiene la propiedad de que en aquellos lugares donde se quiere un suavizado homogéneo, la condición de reposo es cercana a cero, mientras que en los bordes donde la condición de suavidad no es constante, se debe ajustar de manera automática. A continuación, se realiza una representación matemática de esta idea. Sea f la función esperada y g la función observada de un experimento, representadas por los vectores columna con n componentes $[f_i]$ y $[g_i]$ respectivamente. En la Figura 4.1(b) se muestran los valores esperados como resultado de aplicar un filtrado a los valores mostrados en la Figura 4.1(a). Nótese que la función, en la Figura 4.1(b), es suave a pedazos.

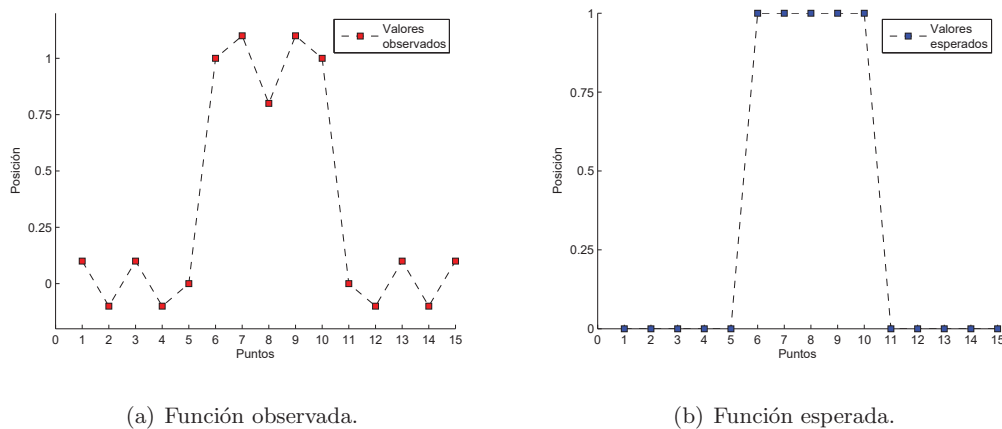


Figura 4.1: Ejemplo de una función observada y la función esperada correspondiente.

La Ecuación (4.1) mide las diferencias al cuadrado entre f y g . Esta representa la energía potencial almacenada en un conjunto de resortes con constantes de rigidez

igual a dos, donde la longitud del i -ésimo resorte equivale al error entre las componentes correspondientes f_i y g_i .

$$E(f) = \sum_i (f_i - g_i)^2 \quad (4.1)$$

Minimizando la Ecuación (4.1) obtenemos la solución $f_i = g_i$. Esta solución es equivalente a la función en la Figura 4.1(a). Sin embargo, deseamos una solución similar a la función en la Figura 4.1(b). Penalizar el gradiente de f_i permite obtener una función suave controlada por η . Modificando la Ecuación (4.1) podemos incluir el parámetro η en nuestra función de energía como se muestra en la Ecuación (4.2). Esta ecuación representa un filtro pasa bajas o de membrana [Calderon06a] controlado por el parámetro η y es utilizada en las técnicas de regularización.

$$E(f) = \sum_i (f_i - g_i)^2 + \eta \sum_i \|\nabla f_i\|^2 \quad (4.2)$$

En nuestro caso, se trabaja con valores discretos por lo que el gradiente se aproxima mediante diferencias finitas [Chapra97], así la Ecuación (4.2) toma la forma de la Ecuación (4.3).

$$E(f) = \sum_i (f_i - g_i)^2 + \eta \sum_i (f_i - f_{i-1})^2 \quad (4.3)$$

Para calcular el i -ésimo valor de f que minimiza la Ecuación (4.3), hacemos $\frac{\partial E(f)}{\partial f_i} = 0$, lo que conduce a la Ecuación (4.4).

$$\frac{\partial E(f)}{\partial f_i} = 2(f_i - g_i) + 2\eta(f_i - f_{i-1}) - 2\eta(f_{i+1} - f_i) = 0 \quad (4.4)$$

Para cada i -ésima observación en nuestro conjunto de datos, tenemos que resolver la i -ésima Ecuación (4.4), lo cual conduce a resolver un sistema lineal de ecuaciones dado por la Ecuación (4.5). Este sistema de ecuaciones es disperso y la matriz R es definida semipositiva con dominancia en la diagonal, por lo cual para valores grandes de η se resuelve utilizando el método de Gauss-Seidel [Chapra97] o Gradiente Conjugado [Chapra97].

Los parámetros l_i descartan las contribuciones con ∇f_i grandes. Para calcularlos y resolver el sistema con la matriz de la Ecuación (4.8) debemos encontrar el i -ésimo valor de l que minimiza la Ecuación (4.6). Haciendo $\frac{\partial E(f,l)}{\partial l_i} = 0$ obtenemos la Ecuación (4.9).

$$\begin{aligned} \frac{\partial E(f,l)}{\partial l_i} &= 2\eta(1-l_i)(f_i - f_{i-1}) = 0 \\ l_i &= \frac{f_i - f_{i-1}}{f_i - f_{i-1}} = 1 \end{aligned} \quad (4.9)$$

De la Ecuación (4.9) notamos que l_i siempre es uno para todos los casos, conduciendo nuevamente al resultado de la Ecuación (4.1). Por lo tanto, es necesario imponer un término de penalización μ por cada $l_i = 1$. De esta manera, se restringe a tener $l_i = 1$ únicamente cuando sea más recomendable pagar la penalización μ a considerar el valor determinado por ∇f_i . El parámetro μ dependerá del valor del gradiente en aquellas regiones que se desean considerar como bordes. En la Ecuación (4.10) podemos ver la inclusión de la penalización para l_i a la Ecuación (4.6).

$$E(f,l) = \sum_i (f_i - g_i)^2 + \eta \sum_i [(f_i - f_{i-1})(1-l_i)]^2 + \eta\mu \sum_i l_i^2 \quad (4.10)$$

Los valores f_i que minimiza la Ecuación (4.10) están dados por el sistema lineal mostrado en la Ecuación (4.8). Para encontrar los valores de l_i necesarios para resolver el sistema, calculamos el valor de l_i que minimiza la Ecuación (4.10), haciendo $\frac{\partial E(f,l)}{\partial l_i} = 0$.

$$\begin{aligned} \frac{\partial E(f,l)}{\partial l_i} &= 2\eta(1-l_i)(f_i - f_{i-1}) + 2\eta\mu l_i = 0 \\ l_i &= \frac{f_i - f_{i-1}}{\mu + f_i - f_{i-1}} \end{aligned} \quad (4.11)$$

nótese que l_i está acotada entre $[0, 1]$ como se pretendía inicialmente.

Los valores de l_i nos permitirán determinar cuales son los puntos en los bordes de la función. En la Figura 4.3 se muestra la función de ajuste con ARC con $\eta = 1000$, para los datos en la Figura 4.1(a). Es fácil observar que reproduce de manera correcta a la función esperada en la Figura 4.1(b), debido a un suavizado de los datos con ruido y un manejo adecuado de los bordes de la función. La Figura 4.4 presenta los valores calculados

para l_i . Se aprecia que en los escalones localizados en los elementos 6 y 11 tenemos que $l_i = 1$ y para el resto de los datos $l_i = 0$. Esto significa que regularizamos desconectando las observaciones 5 y 6 y 10 y 11.

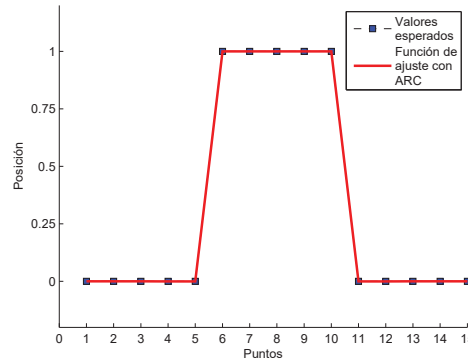


Figura 4.3: Función de ajuste con ARC.

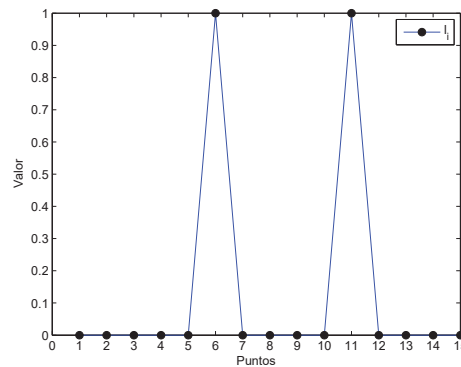


Figura 4.4: Valores de l_i para la función en la Figura 4.1(a).

4.2. Regularización de normales

Supongamos que tenemos el conjunto de puntos mostrados en la Figura 4.5(a), los cuales son el resultado de muestrear la superficie en 2D en la Figura 4.5(b). Los bordes de la superficie están localizados en los elementos 6, 10, 16 y 20 de acuerdo con la Figura 4.6,

donde se muestran las normales calculadas en cada punto de la Figura 4.5(a) utilizando la información de la Figura 4.5(b).

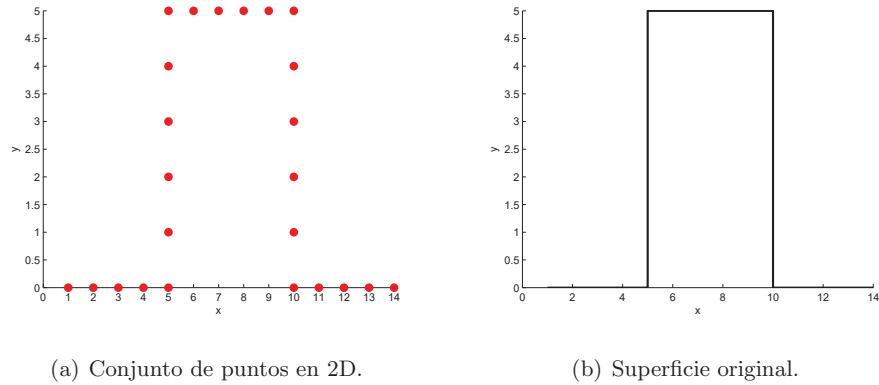


Figura 4.5: Ejemplo de un conjunto de puntos muestreado sobre una superficie en 2D.

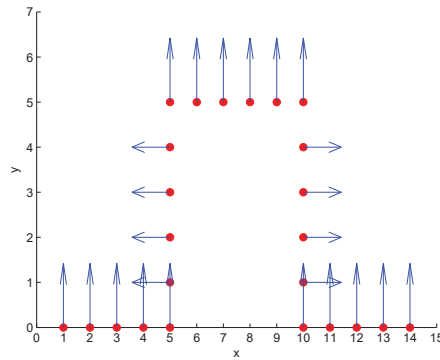


Figura 4.6: Normales correctas.

Intuitivamente, si la superficie es suave, las normales asociadas a cada punto \mathbf{x}_i y a sus k -vecinos más cercanos deben ser parecidas entre si, es decir, la diferencia entre las componentes de las normales debe tender a cero. Si la superficie tiene bordes, se espera un cambio grande en la magnitud de cada componente cerca de estas regiones. Los valores de las componentes x y y de las normales en la Figura 4.6 se muestran en la Figura 4.7. En esta figura observamos que las componentes de las normales se comportan como funciones

suaves a pedazos, con bordes bien definidos tal como lo sugiere la Figura 4.5(b). Una región suave indica que los puntos se encuentran en planos paralelos, mientras los escalones de la función indican la presencia de un plano con una orientación diferente.

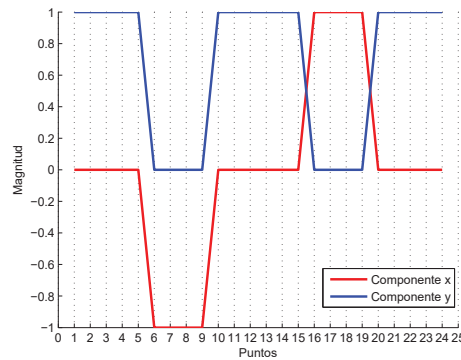


Figura 4.7: Gráfica de las componentes de las normales.

En la Figura 4.8 se muestran las normales calculadas con el método de Hoppe et al. para los puntos en la Figura 4.5(a).

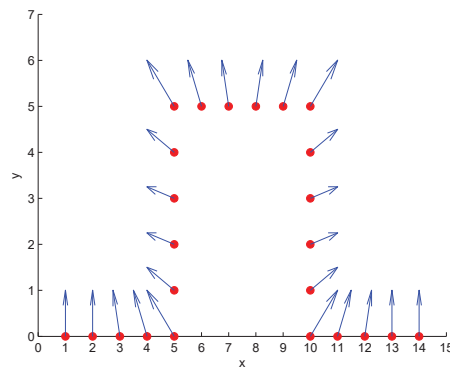


Figura 4.8: Normales calculadas por el método de Hoppe et al.

Una gráfica con las componentes de las normales calculadas por el método de Hoppe et al. se presenta en la Figura 4.9. En esta figura las componentes de las normales estimadas se comportan como funciones más suaves a diferencia de los resultados en la Figura 4.7. El suavizado es producto de la influencia de puntos en planos diferentes, que

ocasionan que la estimación se modifique.

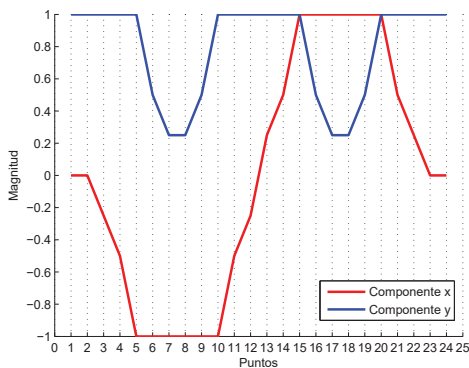


Figura 4.9: Gráfica de las componentes de las normales calculadas por el método de Hoppe et al.

De la Figura 4.9 proponemos un filtrado de las componentes que preserve los bordes. En la sección anterior se mostró que la minimización de una función de energía con ARC nos permite obtener este tipo de filtrado.

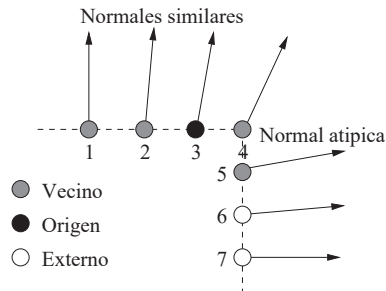


Figura 4.10: Ejemplo de una normal atípica dentro de una vecindad. La normal en 5 difiere totalmente al resto de las normales asociadas a los elementos de la vecindad de 3.

La idea fundamental detrás de la técnica de regularización consiste en marcar las normales atípicas asociadas a los \mathbf{x}_j en cada V_i (ver Figura 4.10) y calcular una nueva estimación de la normal denotada como \mathbf{m}_i , a partir de las normales de los puntos en V_i que no fueron marcadas (ver Figura 4.11). Se propone calcular las normales a partir de la minimización de una función de energía con ARC. Deseamos manejar el caso más general en tres dimensiones, por lo tanto, la Ecuación (4.10) es modificada para considerar a los

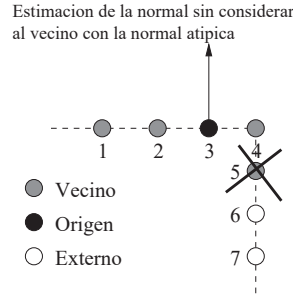


Figura 4.11: Nueva estimación de la normal al eliminar al vecino con la normal atípica de la Figura 4.10.

elementos en una vecindad V_i . La ecuación final es la siguiente:

$$E(\mathbf{m}, l) = \sum_{i \in X} \|\mathbf{m}_i - \mathbf{n}_i\|^2 + \eta \sum_{i \in X} \sum_{j \in V_i} [\|\mathbf{m}_i - \mathbf{m}_j\| (1 - l_{ij})]^2 + \eta \mu \sum_{i \in X} \sum_{j \in V_i} l_{ij}^2, \quad (4.12)$$

sujeto a $\|\mathbf{m}_i\| = 1$

donde \mathbf{n}_i es la normal calculada con el método propuesto por Hoppe et al. (descrito en el Capítulo 3) para un cierto tamaño de vecindad. El término l_{ij} indica la pertenencia ($l_{ij} = 0$) ó no ($l_{ij} = 1$) del j -ésimo elemento a la vecindad del i -ésimo elemento. Este término brinda la información sobre los elementos a descartar en cada vecindad para calcular la nueva estimación de la normal. Dado que las normales son vectores unitarios se agrega la restricción $\|\mathbf{m}\| = 1$, la cual es incluida con un multiplicador de Lagrange γ en la función langrangiana de la Ecuación (4.13).

$$L(\mathbf{m}, l, \gamma) = E(\mathbf{m}, l) - \sum_{i \in X} \gamma_i (\|\mathbf{m}_i\| - 1) \quad (4.13)$$

La solución se calcula resolviendo las condiciones de Karush-Kuhn-Tucker dadas por las Ecuaciones (4.14), (4.15) y (4.16):

$$\nabla_{\mathbf{m}} L(\mathbf{m}, l, \gamma) = 0 \quad (4.14)$$

$$\nabla_l L(\mathbf{m}, l, \gamma) = 0 \quad (4.15)$$

$$\|\mathbf{m}_i\| = 1 \quad (4.16)$$

donde ∇_x es el operador gradiente parcial. Se propone emplear el algoritmo de Gauss-Seidel para resolver este sistemas de ecuaciones, en virtud de que el sistema lineal de ecuaciones tiene una matriz diagonalmente dominante y semidefinida positiva.

Resolviendo la Ecuación (4.15) se tiene que:

$$[\mu + \|\mathbf{m}_i - \mathbf{m}_j\|^2]l_{ij} - \|\mathbf{m}_i - \mathbf{m}_j\|^2 = 0 \quad (4.17)$$

$$l_{ij} = \frac{\|\mathbf{m}_i - \mathbf{m}_j\|^2}{\mu + \|\mathbf{m}_i - \mathbf{m}_j\|^2} \quad (4.18)$$

Ahora resolviendo la Ecuación (4.14) se tiene que:

$$m_{i,d}[1 + \gamma_i + \eta \sum_{j \in V_i} (1 - l_{ij})^2] = n_{i,d} + \eta \sum_{j \in V_i} m_{j,d}(1 - l_{ij})^2 \quad (4.19)$$

$$m_{i,d} = \frac{n_{i,d} + \eta \sum_{j \in V_i} m_{j,d}(1 - l_{ij})^2}{1 + \gamma_i + \eta \sum_{j \in V_i} (1 - l_{ij})^2} \quad (4.20)$$

La t -ésima iteración de Gauss-Seidel está dada por las siguientes ecuaciones:

$$\widehat{\mathbf{m}}_i^{(t)} = \frac{\mathbf{m}_i^{(t)}}{\|\mathbf{m}_i^{(t)}\|} \quad (4.21)$$

$$l_{ij}^{(t)} = \frac{\|\widehat{\mathbf{m}}_i^{(t)} - \widehat{\mathbf{m}}_j^{(t)}\|^2}{\mu + \|\widehat{\mathbf{m}}_i^{(t)} - \widehat{\mathbf{m}}_j^{(t)}\|^2} \quad (4.22)$$

$$m_{i,d}^{(t+1)} = \frac{n_{i,d} + \eta \sum_{j \in V_i} \widehat{m}_{j,d}^{(t)}(1 - l_{ij}^{(t)})^2}{1 + \gamma_i + \eta \sum_{j \in V_i} (1 - l_{ij}^{(t)})^2} \quad (4.23)$$

La restricción para la Ecuación (4.12) se satisface en cada iteración de acuerdo a la Ecuación (4.21), por lo tanto, el multiplicador de Lagrange γ_i puede tomar cualquier valor, debido a que su contribución a la función de costo será cero en cada iteración, por simplicidad se hace $\gamma_i = 0$.

En la Figura 4.12 se muestra el resultado de aplicar la regularización a las normales presentadas en la Figura 4.8. Las componentes de las normales filtradas se presentan en la Figura 4.13. Podemos observar como estas componentes reproducen el comportamiento real de las normales en la Figura 4.7. Como es deseable para esta superficie, las regiones son suaves a pedazos con los bordes claramente definidos. En la Figura 4.14 se muestran los valores de los términos de pertenencia l_{ij} . Nótese como los valores $l_{ij} = 1$ corresponden únicamente a los bordes de la Figura 4.5(b).

Es importante mencionar que para superficies que contienen solo regiones planas y bordes, como el caso de la Figura 4.5(b) o conjuntos de puntos con niveles altos de ruido la estimación de la normal \mathbf{m}_i es una buena aproximación. Sin embargo, si los datos no contienen ruido o son una combinación de bordes y regiones con variaciones suaves en la normal, por ejemplo un escalón con una media circunferencia en su parte más alta, es mejor utilizar los valores de l_{ij} para descartar elementos en las vecindades y calcular una nueva aproximación de la normal utilizando el método presentado por Hoppe et al. Esto evita que los detalles suaves se pierdan como consecuencia del filtrado inherente a la función con ARC, la cual interará hacer de una superficie suave, una superficie suave a pedazos. En la siguiente sección se presenta el algoritmo general para la regularización.

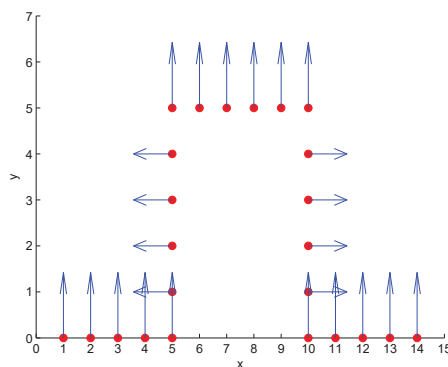


Figura 4.12: Normales regularizadas empleando una función de energía con ARC.

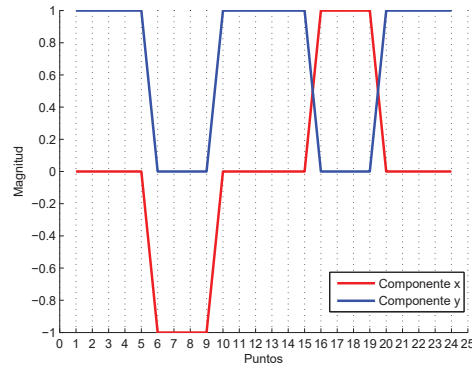


Figura 4.13: Gráfica de las componentes de las normales regularizadas.

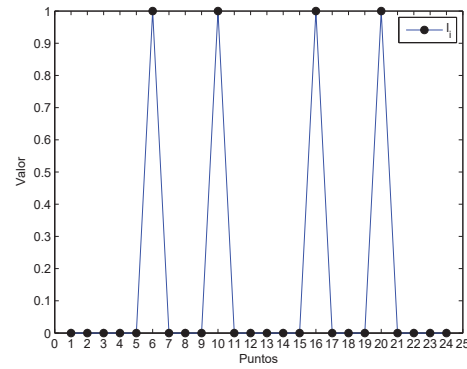


Figura 4.14: Valores de l_i para las normales en la Figura 4.8.

4.3. Cálculo de las normales regularizadas

De lo expuesto en la sección anterior, el algoritmo general de regularización se resume de la siguiente manera:

1. Calcular las normales orientadas \mathbf{n}_i para alguna vecindad k empleado el método de Hoppe et al. descrito en el Capítulo 3.
2. Calcular la vecindad reorganizada y la normal filtrada \mathbf{m}_i haciendo:

a) Sea $\mathbf{m}_i^{(0)} = \mathbf{n}_i$ y $t = 0$

- b) Calcular los vectores $\mathbf{m}_i^{(t)}$ aplicando la Ecuación (4.21)
 - c) Calcular los términos de pertenencia $l_{ij}^{(t)}$ usando la Ecuación (4.22)
 - d) Actualizar las normales $\mathbf{m}_i^{(t+1)}$ aplicando la Ecuación (4.23)
 - e) Hacer $t \leftarrow t + 1$
 - f) Repetir los pasos (b-e) hasta que $\|\mathbf{m}_i^{(t+1)} - \mathbf{m}_i^{(t)}\| < \epsilon$
3. Si los puntos están libres de ruido o representan superficies suaves calcular una nueva estimación de las normales orientadas \mathbf{n}'_i usando el método de Hoppe et al., a partir de la siguiente matriz de covarianza empleando los términos l_{ij}^* finales (óptimos)

$$C_i^* = \sum_{j \in V_i} (1 - l_{ij}^*) [(\mathbf{x}_j - \mathbf{c}_i^*) \otimes (\mathbf{x}_j - \mathbf{c}_i^*)] \quad (4.24)$$

donde

$$\mathbf{c}_i^* = \frac{\sum_{j \in V_i} (1 - l_{ij}^*) \mathbf{x}_j}{\sum_{j \in V_i} (1 - l_{ij}^*)} \quad (4.25)$$

4. Si se desea refinar la estimación, repetir el algoritmo desde el paso 2 empleando las nuevas normales \mathbf{n}'_i .

4.4. Conclusiones

En este capítulo se presentó un nuevo método para la estimación de normales a partir de conjuntos de puntos. El algoritmo está diseñado para obtener un mejor desempeño en las regiones cercanas a bordes y esquinas. Esto se logra a partir de la minimización de una función de energía basada en ARC. La función de energía nos permite identificar a los mejores elementos en la vecindad de cada punto para la estimación de la normal. En el siguiente capítulo, se hace una comparación cualitativa y cuantitativa, de las normales estimadas con el método desarrollado por Hoppe et al. descrito en el Capítulo 3 y la propuesta de este trabajo.

Capítulo 5

Experimentos y Resultados

En este capítulo, se comparan de forma cuantitativa y cualitativa el esquema para estimación de normales desarrollado por Hoppe et al. descrito en el Capítulo 3 y la propuesta para estimación de normales con reorganización de vecindades presentada en el Capítulo 4.

5.1. Configuración experimental

Para comparar los métodos de manera cuantitativa, es necesario medir la desviación entre la normal estimada y la normal correcta sobre la superficie. Desafortunadamente para conjuntos de puntos en 3D frecuentemente no se conoce la superficie original. Para solucionar este problema Dey et al. [Dey05] proponen calcular el conjunto de normales de referencia reconstruyendo la superficie a partir del conjunto de datos originales (sin ruido) usando el algoritmo de Coconos [Amenta00]. El promedio de las normales de los triángulos incidentes a un vértice \mathbf{v} se toma como la normal de referencia. La propuesta anterior funciona razonablemente para superficies suaves. Sin embargo, tal como se mencionó en el Capítulo 2, el algoritmo de Coconos no maneja adecuadamente las superficies con bordes y esquinas, por lo tanto, la estimación de la normal de referencia es deficiente para estos casos. Al ser el tema central de este trabajo, se decidió crear algunos modelos sintéticos de superficies con bordes y esquinas que incluyan al conjunto de normales de referencia.

Como todas las normales calculadas en este trabajo son unitarias, definimos el error de una normal estimada en el punto \mathbf{x}_i como el ángulo entre la normal de referencia $\mathbf{n}_{r,i}$ y la normal estimada $\mathbf{n}_{e,i}$. El ángulo está dado como:

$$\theta_i = \text{acos} \left(\frac{\mathbf{n}_{r,i} \cdot \mathbf{n}_{e,i}}{\|\mathbf{n}_{r,i}\| \|\mathbf{n}_{e,i}\|} \right) \quad (5.1)$$

Los comparación cualitativa de los métodos se realiza generando reconstrucciones de la superficie original. Se decidió utilizar la implementación de Ohtake [Ohtake03a] del algoritmo MPUI para obtener estas reconstrucciones. El algoritmo MPUI emplea las normales estimadas por cada método para efectuar la reconstrucción. Se crea además la reconstrucción correspondiente al conjunto de normales de referencia, la cual sirve como punto de comparación.

De manera adicional, se efectuaron pruebas para observar la tolerancia de los métodos al ruido. Para estas pruebas se generaron nuevos conjuntos de puntos agregando ruido controlado a los conjuntos originales. Las componentes x , y y z del ruido son independientes y uniformemente distribuidas. En los experimentos, se elige un nivel de ruido tomando como base dos escalas: una local y otra global [Dey05]. Para la escala global, se toma la amplitud del ruido como un factor del lado más largo de una caja paralela a los ejes que contiene al conjunto de puntos. Dado que esta medida es grande, el factor debe ser pequeño para que el conjunto de puntos después de las perturbaciones aún pueda ser usado razonablemente para las reconstrucciones. Dey et al. [Dey05] sugieren que la escala global para las perturbaciones es más cercana a la realidad. Para la elección del factor en la escala local, se emplea la distancia promedio de cada punto \mathbf{x} a sus cinco vecinos más cercanos. El punto \mathbf{x} es perturbado con un factor de esta distancia. Los factores de escala para ambos tipos de ruido se muestran en la Tabla 5.1.

Tipo de ruido	Factores de escala				
Local	0.1	0.2	0.3	0.4	
Global	0.005	0.0075	0.01	0.0125	0.015

Tabla 5.1: Factores de escala para los tipos de ruido en los experimentos.

5.2. Experimentos con modelos sintéticos

Los parámetros para el algoritmo MPUI (ver [Ohtake03b]) en todas las pruebas de estos experimentos fueron un tamaño mínimo de celda de 0.005 para la subdivisión espacial y un error máximo de la aproximación de 0.005 en cada celda. Para el resto de los parámetros del algoritmo se tomaron los valores por defecto de la implementación de Ohtake [Ohtake03a]. Estos parámetros conducen a la mejor reconstrucción para cada modelo empleando las normales de referencia. Para el caso del algoritmo con reorganización de vecindades tomamos los valores de $\eta = 1000$ y $\mu = 0.01$.

5.2.1. Cubo con medias esferas en dos de sus caras

La superficie en la Figura 5.1(a) corresponde a la reconstrucción generada con el conjunto de normales de referencia de este modelo y será utilizada como base de comparación. El tamaño de la vecindad fue de 15 elementos para ambos algoritmos de estimación de normales.

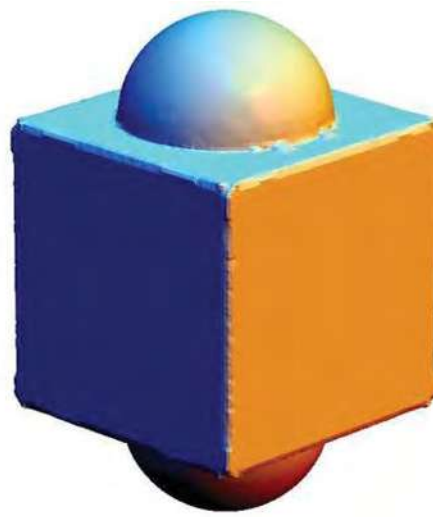
La superficie reconstruida empleando las normales calculadas por el método de Hoppe et al. se presenta en la Figura 5.1(b). En este caso, se observa que las esquinas tienen problemas serios, como consecuencia directa de una mala estimación de la normal cerca de estas regiones. La Figura 5.1(c) muestra la superficie reconstruida usando la propuesta con reorganización de vecindades. Es fácil notar como el método presentado en este trabajo incrementa la precisión de la estimación de las normales cerca de las esquinas sin afectar las áreas suaves. En la Tabla 5.2 se muestra el promedio del error angular y desviación estándar para ambos métodos de estimación de normales.

Algoritmo	Modelo	
	Promedio del error	Desviación Estándar
Hoppe et. al.	7.5544°	13.5544°
Propuesta	0.9973°	2.0968°

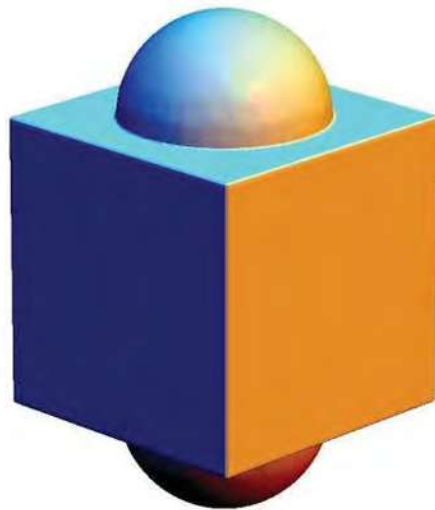
Tabla 5.2: Errores de estimación en ambos métodos para el modelo de un cubo modificado.



(a) Reconstrucción empleando las normales de referencia.



(b) Reconstrucción empleando las normales estimadas por el método de Hoppe et al.



(c) Reconstrucción empleando las normales estimadas con la propuesta de este trabajo.

Figura 5.1: Superficies reconstruidas para el modelo de un cubo modificado.

5.2.2. Pirámide

En la Figura 5.2(a) se muestra la superficie reconstruida a partir del conjunto de normales de referencia de este modelo. Para ambos algoritmos de estimación de normales, el tamaño de la vecindad fue de 18 elementos.

La superficie reconstruida empleando las normales calculadas por el método de Hoppe et al. se presenta en la Figura 5.2(b). Nuevamente se observa que los bordes tienen problemas debido a una mala estimación de la normal cerca de estas regiones. En la Figura 5.2(c) se observa la superficie reconstruida usando la propuesta con reorganización de vecindades. Podemos notar que el método presentado en este trabajo incrementa la precisión de la estimación de las normales cerca de las esquinas sin afectar las regiones planas. En la Tabla 5.3 se muestran los promedios de los errores angulares obtenidos para ambos métodos de estimación de normales.

Algoritmo	Modelo	
	Promedio del error	Desviación Estándar
Hoppe et. al.	4.0338°	7.8453°
Propuesta	2.2055°	10.5999°

Tabla 5.3: Errores de estimación en ambos métodos para el modelo de una pirámide.

5.2.3. Cilindro

La superficie que se muestra en la Figura 5.3(a) fue reconstruida a partir del conjunto de normales de referencia de este modelo. El tamaño de la vecindad fue de 20 elementos en ambos algoritmos de estimación de normales.

En la Figura 5.3(b) se tiene la superficie reconstruida empleando las normales calculadas por el método de Hoppe et al. Una vez más se observa que los bordes tienen problemas debido a la mala estimación de la normal. En la Figura 5.3(c) se muestra la superficie reconstruida usando la propuesta con reorganización de vecindades. El método presentado en este trabajo incrementa la precisión de la estimación de las normales. En la Tabla 5.4 se muestran los promedios de los errores angulares obtenidos para ambos métodos.

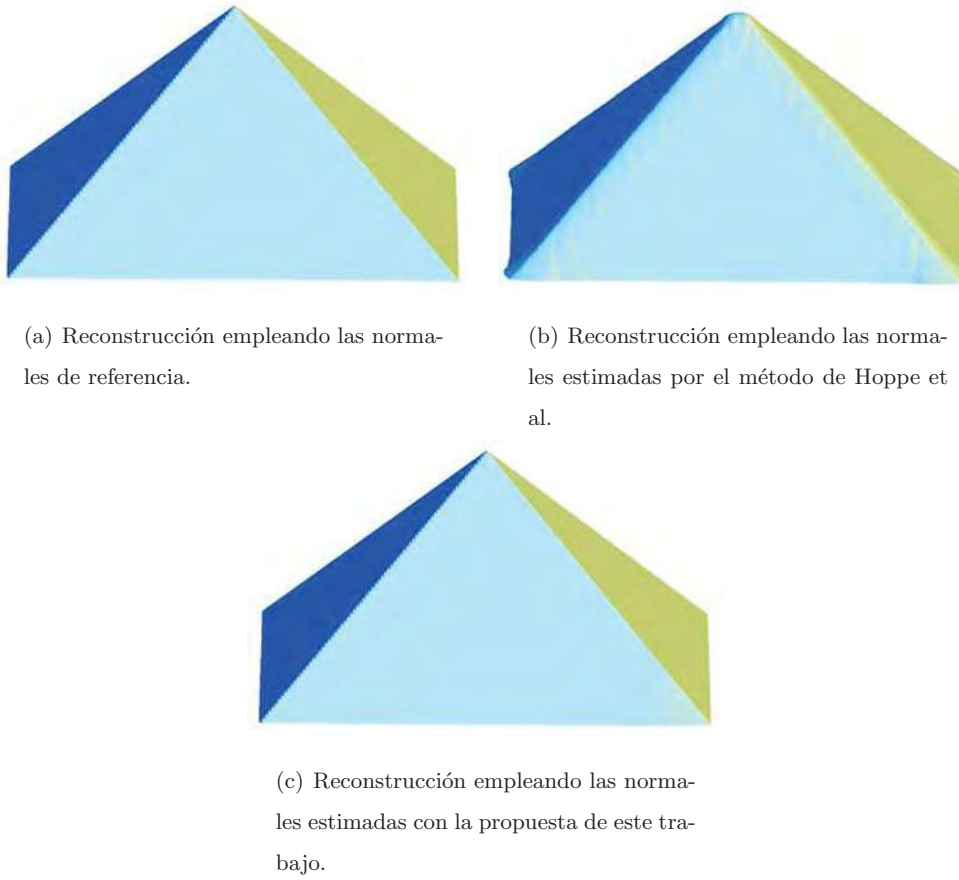
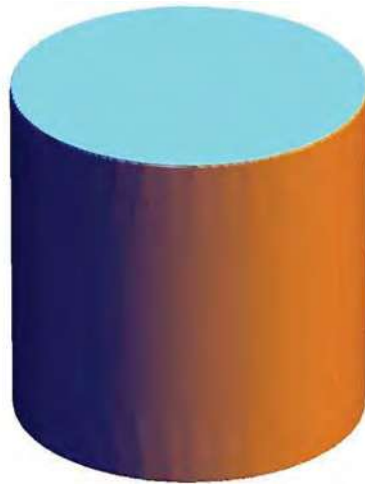


Figura 5.2: Superficies reconstruidas para el modelo de una pirámide.

Algoritmo	Modelo	
	Promedio del error	Desviación Estándar
Hoppe et. al.	4.7679°	9.7428°
Propuesta	1.7282°	2.2406°

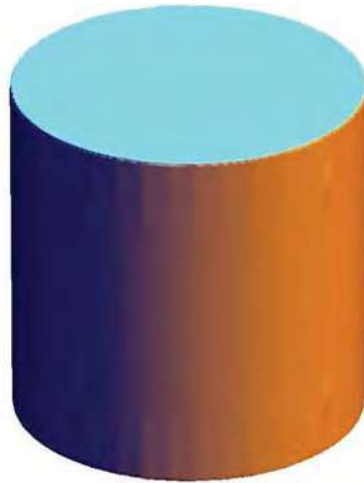
Tabla 5.4: Errores de estimación en ambos métodos para el modelo de un cilindro.



(a) Reconstrucción empleando las normales de referencia.



(b) Reconstrucción empleando las normales estimadas por el método de Hoppe et al.



(c) Reconstrucción empleando las normales estimadas con la propuesta de este trabajo.

Figura 5.3: Superficies reconstruidas para el modelo de un cilindro.

5.3. Experimentos con modelos sintéticos con ruido

Para observar el comportamiento de los métodos bajo la presencia de ruido, se aplicaron distintos tipos y factores de ruido a cada uno de los modelos sintéticos. Los parámetros del algoritmo de reconstrucción son los mismos que fueron utilizados en la Sección 5.2. A continuación, se presenta los resultados obtenidos.

5.3.1. Ruido local

Cubo con medias esferas en dos de sus caras

La Tabla 5.5 muestra el promedio del error angular para ambos métodos.

Factor de ruido	Hoppe et al.		Propuesta	
	Promedio	Desviación Estándar	Promedio	Desviación Estándar
0.1	7.8458	13.4154	1.3050	2.0654
0.2	8.2543	13.3499	1.9141	3.1797
0.3	8.8031	13.4122	2.3852	2.3967
0.4	9.3044	13.4514	3.0691	2.8251

Tabla 5.5: Errores de estimación en ambos métodos para el modelo de un cubo modificado con ruido local.

En la Figura 5.4 se muestra una gráfica de los valores en la Tabla 5.5. En este caso, podemos observar que la estimación de las normales empleando el método propuesto supera al método de Hoppe et al. Es de esperarse que la reconstrucción final también sea superior. En las Figuras 5.5-5.8 se muestran las reconstrucciones finales empleando ambos métodos de estimación de normales con los distintos factores de ruido. Es importante observar la calidad de la reconstrucción cerca de los bordes y esquinas, donde es posible apreciar una mejoría de la propuesta de este trabajo aún con niveles altos de ruido.

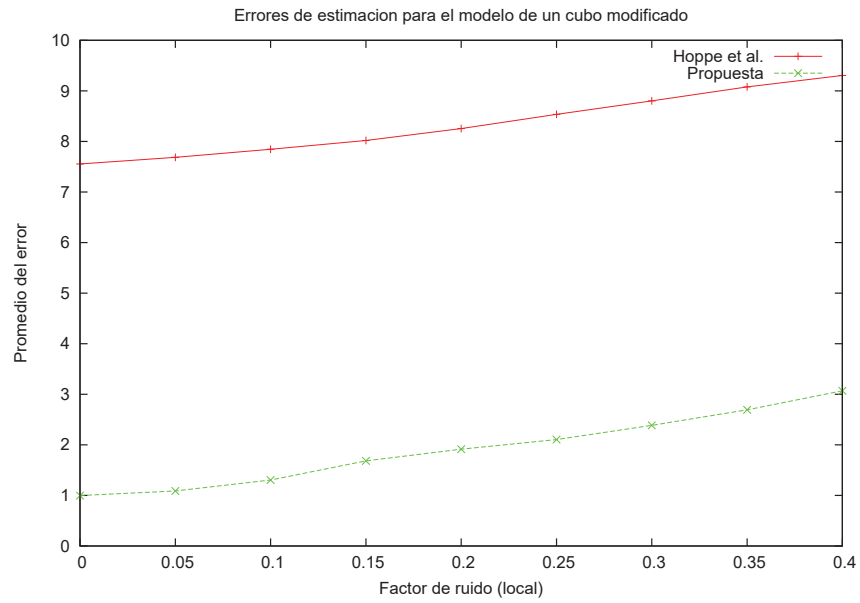
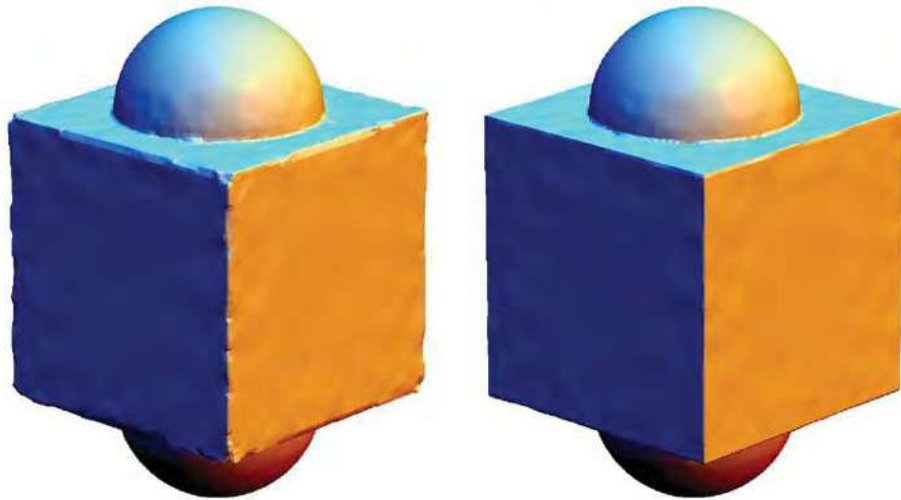


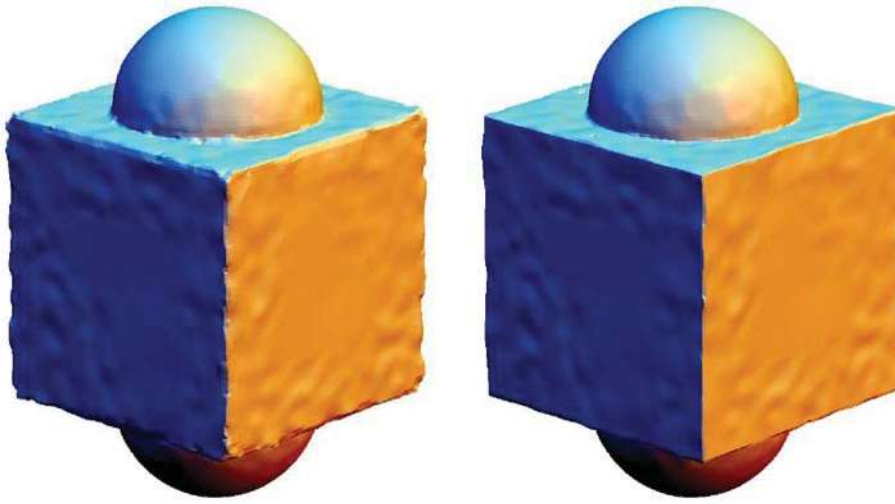
Figura 5.4: Gráfica de los errores de estimación en ambos métodos para el modelo de un cubo modificado con ruido local.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

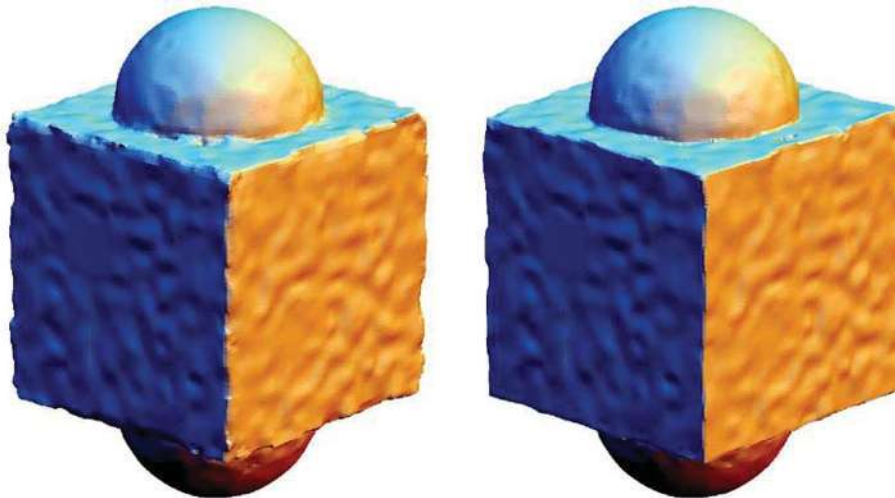
Figura 5.5: Reconstrucciones agregando ruido local, con un factor de 0.1.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

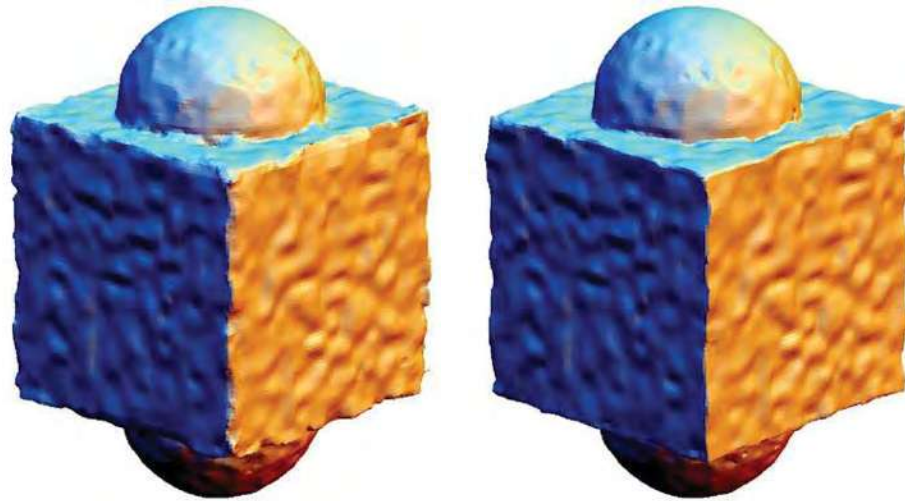
Figura 5.6: Reconstrucciones agregando ruido local, con un factor de 0.2.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.7: Reconstrucciones agregando ruido local, con un factor de 0.3.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.8: Reconstrucciones agregando ruido local, con un factor de 0.4.

Pirámide

En la Tabla 5.6 se muestra la estimación del error angular para ambos métodos. La gráfica correspondiente se tiene en la Figura 5.9. Nuevamente se puede observar que la estimación de las normales empleando el método propuesto supera al método de Hoppe et al. La reconstrucción final también es superior. En las Figuras 5.10-5.13 se muestran las reconstrucciones finales empleando ambos métodos de estimación de normales con los distintos factores de ruido. La calidad de la reconstrucción cerca de los bordes y esquinas presenta una mejoría usando la propuesta de este trabajo.

Factor de ruido	Hoppe et al.		Propuesta	
	Promedio	Desviación Estándar	Promedio	Desviación Estándar
0.1	4.2788	7.8542	2.2042	9.6963
0.2	4.5589	7.7810	2.8274	9.8444
0.3	4.8545	7.6846	3.3457	10.1939
0.4	5.1937	7.6401	3.9615	9.9197

Tabla 5.6: Errores de estimación en ambos métodos para el modelo de una pirámide con ruido local.

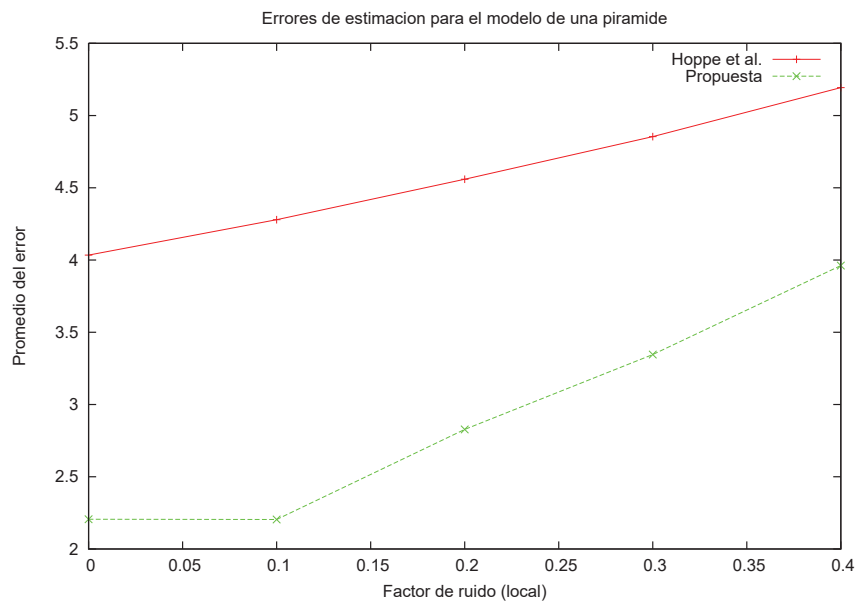
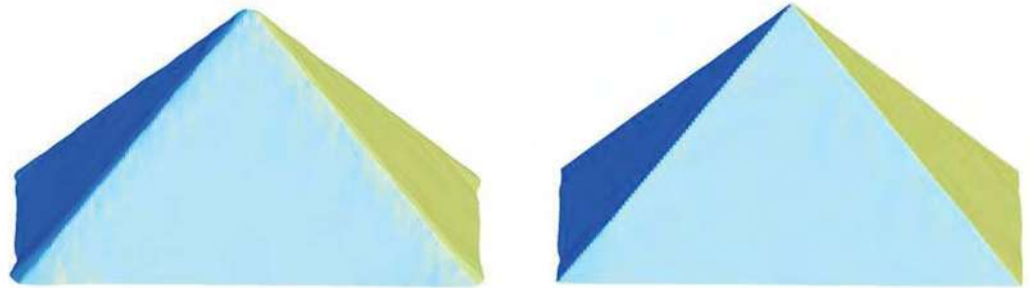


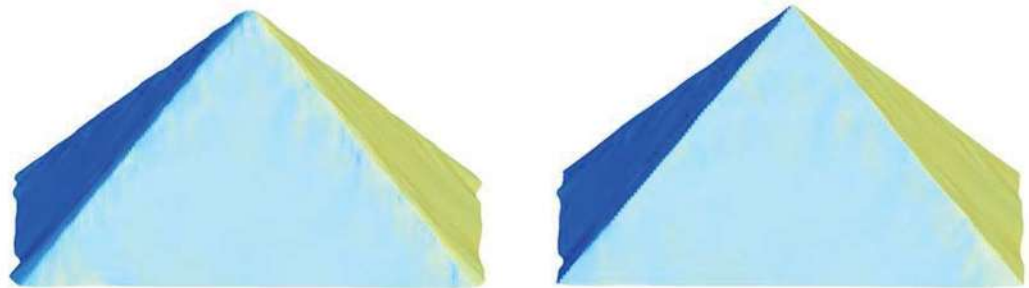
Figura 5.9: Gráfica de los errores de estimación en ambos métodos para el modelo de una pirámide con ruido local.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.10: Reconstrucciones agregando ruido local, con un factor de 0.1.



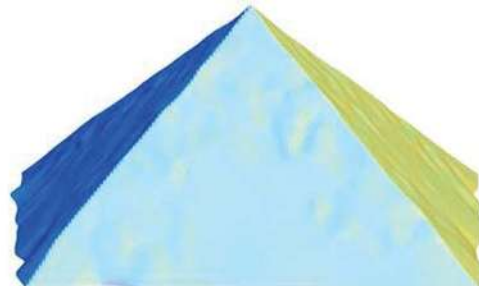
(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.11: Reconstrucciones agregando ruido local, con un factor de 0.2.

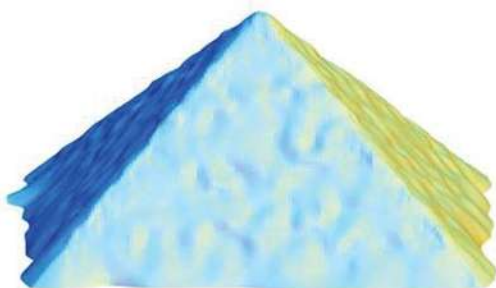


(a) Estimación de normales con el método de Hoppe et al.



(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.12: Reconstrucciones agregando ruido local, con un factor de 0.3.



(a) Estimación de normales con el método de Hoppe et al.



(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.13: Reconstrucciones agregando ruido local, con un factor de 0.4.

Cilindro

La Tabla 5.7 muestra el promedio del error angular. En la Figura 5.14 se tiene la gráfica correspondiente. La estimación de las normales empleando el método propuesto es mejor que la del método de Hoppe et al. La reconstrucción final es superior como en los ejemplos anteriores. En las Figuras 5.15-5.18 se muestran las reconstrucciones finales empleando ambos métodos de estimación de normales. La calidad de la reconstrucción cerca de los bordes y esquinas es mucho mejor con la propuesta de este trabajo.

Factor de ruido	Hoppe et al.		Propuesta	
	Promedio	Desviación Estándar	Promedio	Desviación Estándar
0.1	5.0947	9.9072	1.9140	2.6311
0.2	5.3435	9.6576	2.3664	3.3757
0.3	5.6782	9.6272	2.6161	2.3080
0.4	6.0592	9.5558	3.2395	2.9612

Tabla 5.7: Errores de estimación en ambos métodos para el modelo de un cilindro con ruido local.

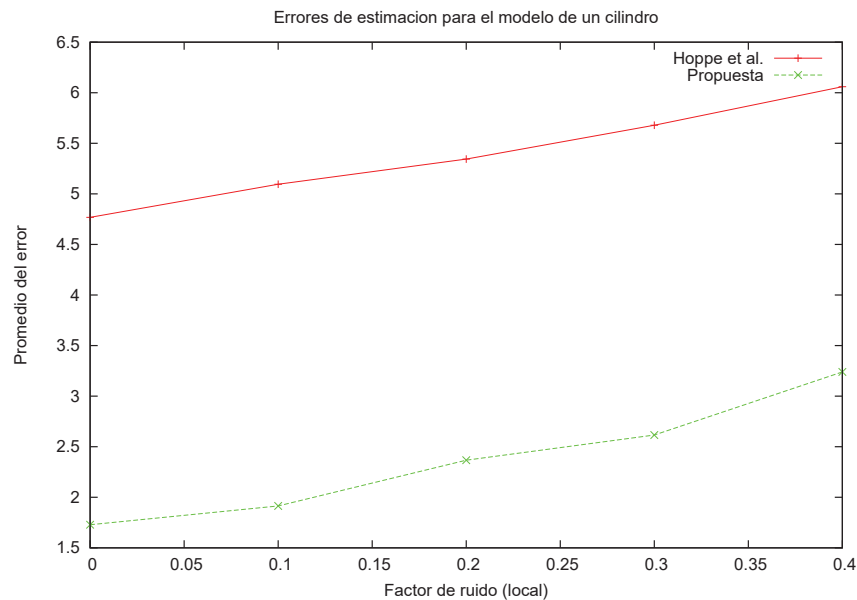
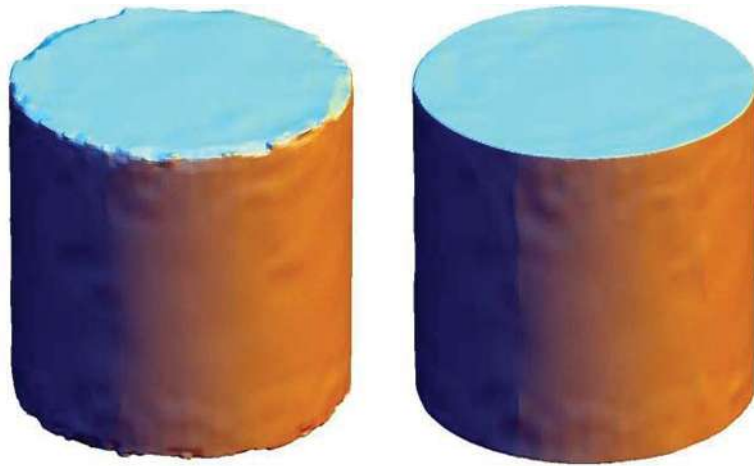


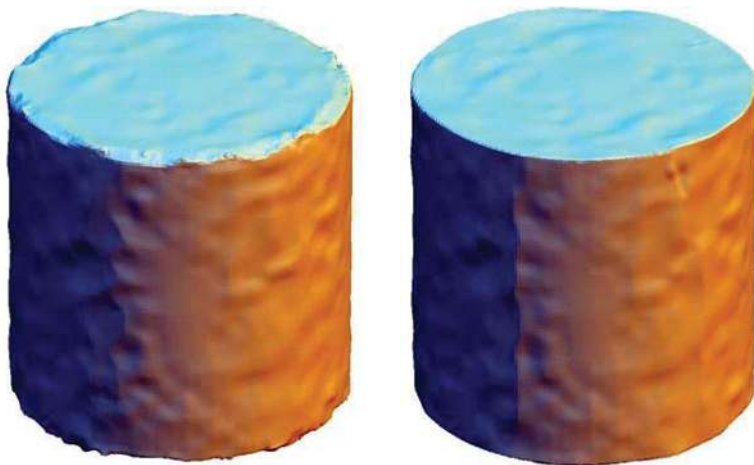
Figura 5.14: Gráfica de los errores de estimación en ambos métodos para el modelo de un cilindro con ruido local.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

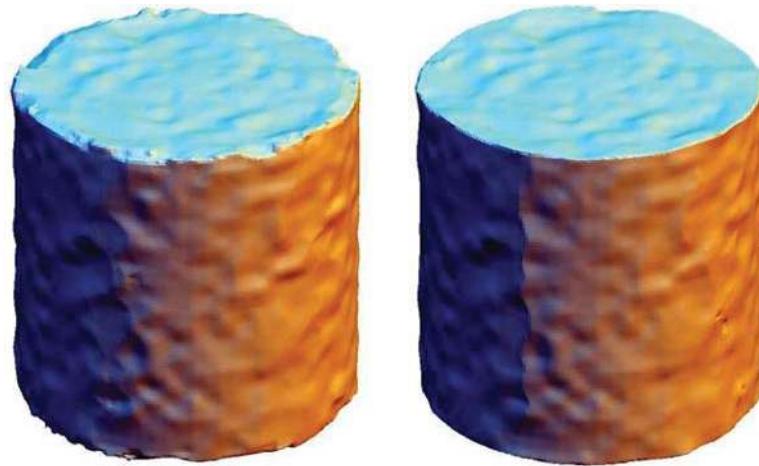
Figura 5.15: Reconstrucciones agregando ruido local, con un factor de 0.1.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

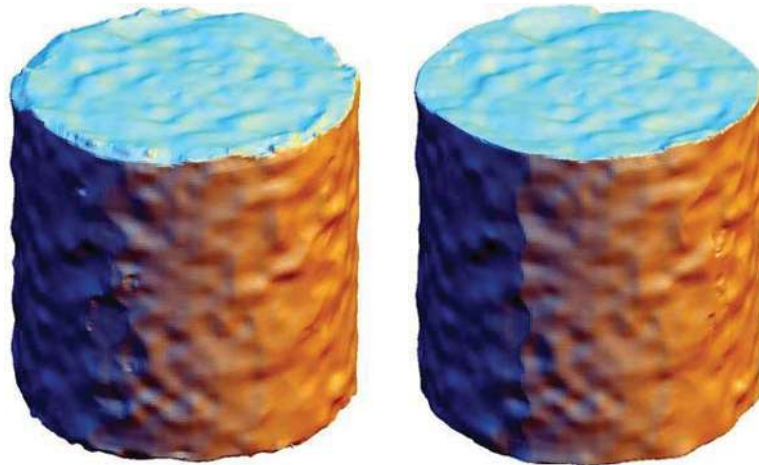
Figura 5.16: Reconstrucciones agregando ruido local, con un factor de 0.2.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.17: Reconstrucciones agregando ruido local, con un factor de 0.3.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.18: Reconstrucciones agregando ruido local, con un factor de 0.4.

5.3.2. Ruido global

Cubo con medias esferas en dos de sus caras

En la Tabla 5.8 se muestran los resultados de la estimación del error angular de ambos métodos cuando se tiene ruido global. Una gráfica con los valores de esta tabla se muestra en la Figura 5.19. Las reconstrucciones correspondientes se presentan en las Figuras 5.20-5.24. Al igual que en el caso anterior, es posible notar una mejoría tanto en la estimación como en la reconstrucción cerca de los bordes y esquinas al emplear la propuesta de este trabajo. Las reconstrucciones muestran bordes más definidos a pesar de tener gran cantidad de ruido en la muestra.

Factor de ruido	Hoppe et al.		Propuesta	
	Promedio	Desviación Estándar	Promedio	Desviación Estándar
0.005	8.0167	13.3623	1.4809	1.9170
0.0075	8.2862	13.2678	2.0153	3.2320
0.01	8.6463	13.2526	2.3381	2.7076
0.0125	9.0244	13.2628	2.7104	2.7545
0.015	9.3773	13.2546	3.1292	3.0472

Tabla 5.8: Errores de estimación en ambos métodos para el modelo de un cubo modificado con ruido global.

Pirámide

Los resultados de la estimación del error angular de ambos métodos cuando se tiene ruido global se muestran en la Tabla 5.9. Para este modelo se eliminó la prueba con el factor de ruido de 0.0125 ya que la orientación de las normales es inconsistente con el resto de los experimentos. En la Figura 5.25 se presenta la gráfica correspondiente. Las reconstrucciones finales se pueden observar en las Figuras 5.26-5.29. Podemos notar una mejoría tanto en la estimación como en la reconstrucción cerca de los bordes y esquinas al emplear la propuesta de este trabajo.

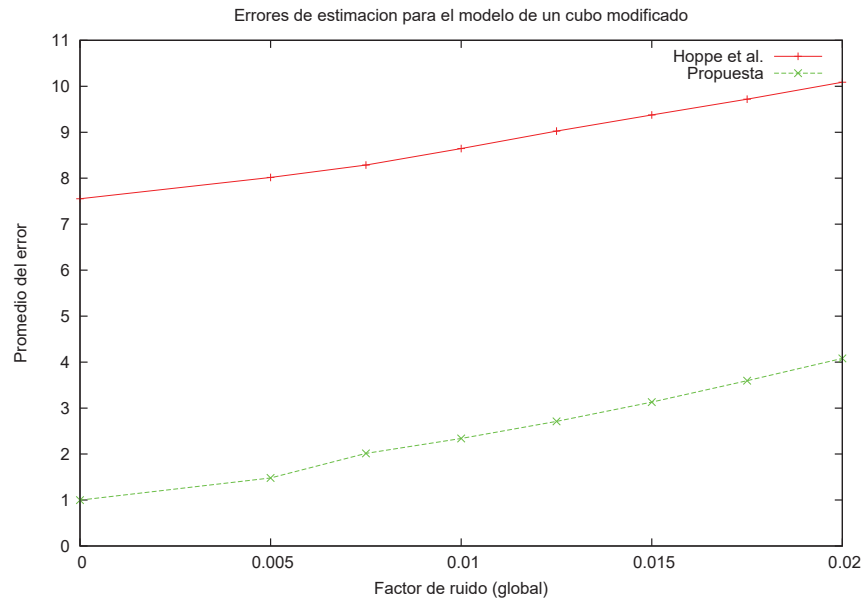
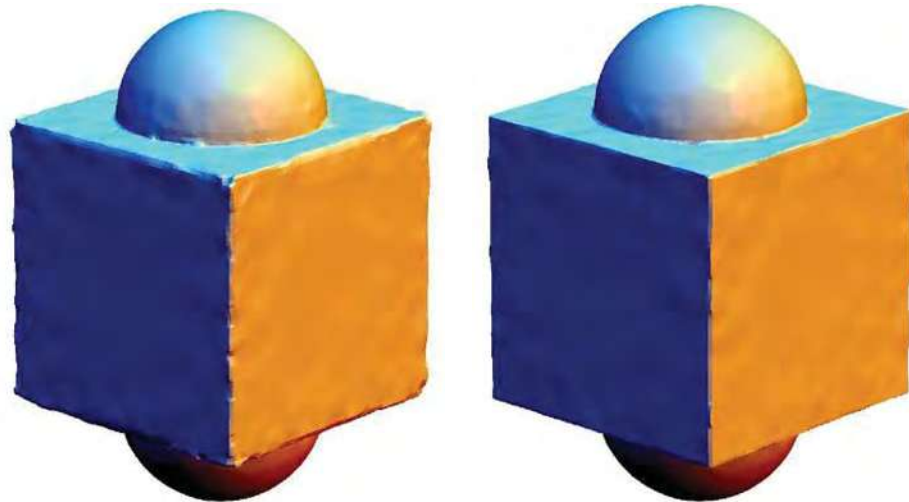


Figura 5.19: Gráfica de los errores de estimación en ambos métodos para el modelo de un cubo modificado con ruido global.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.20: Reconstrucciones agregando ruido global, con un factor de 0.005.

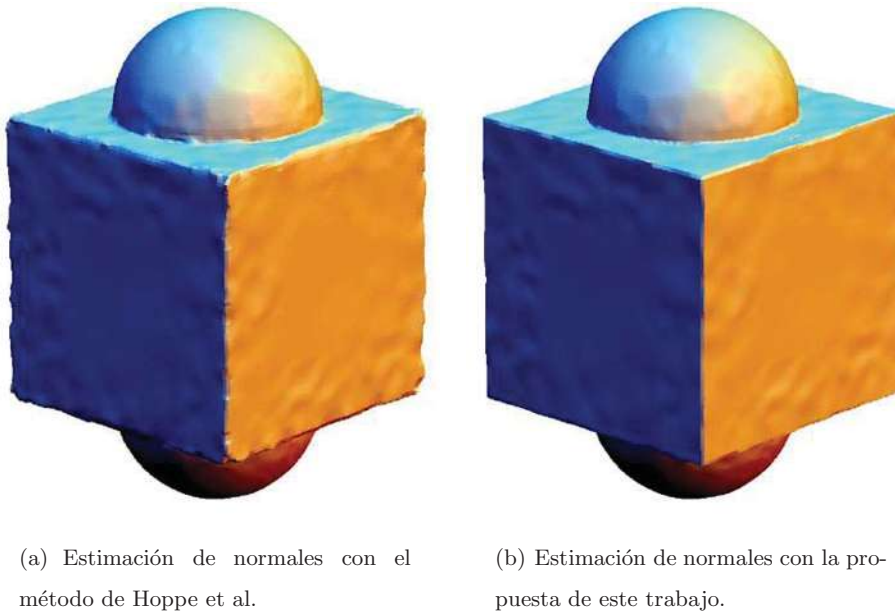


Figura 5.21: Reconstrucciones agregando ruido global, con un factor de 0.0075.

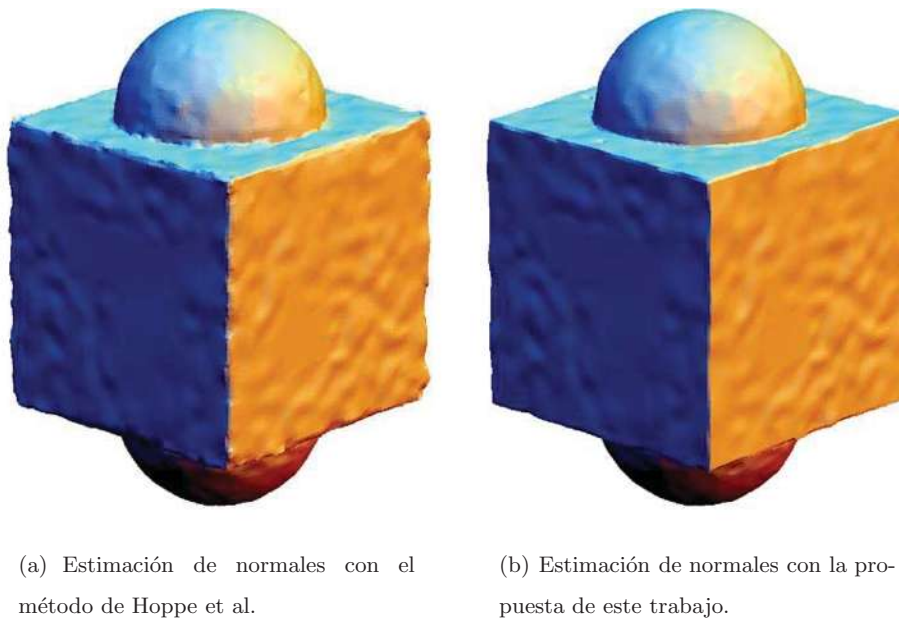
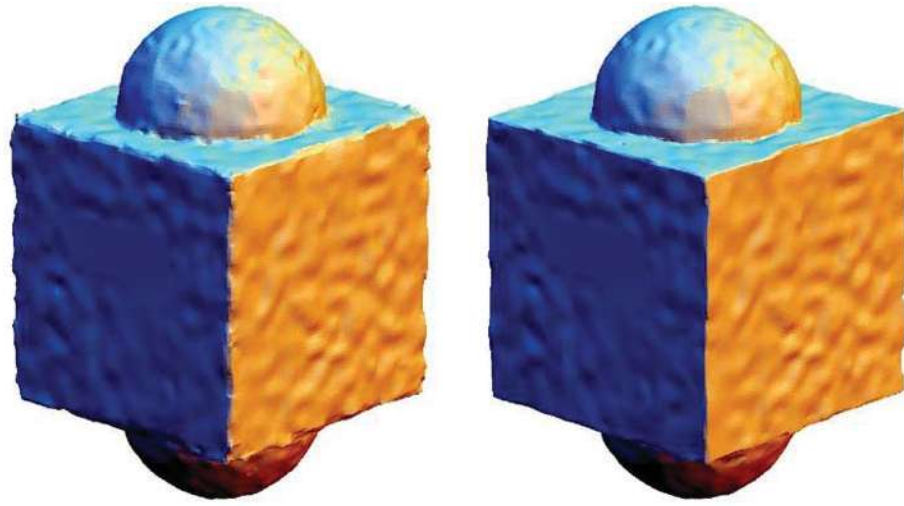


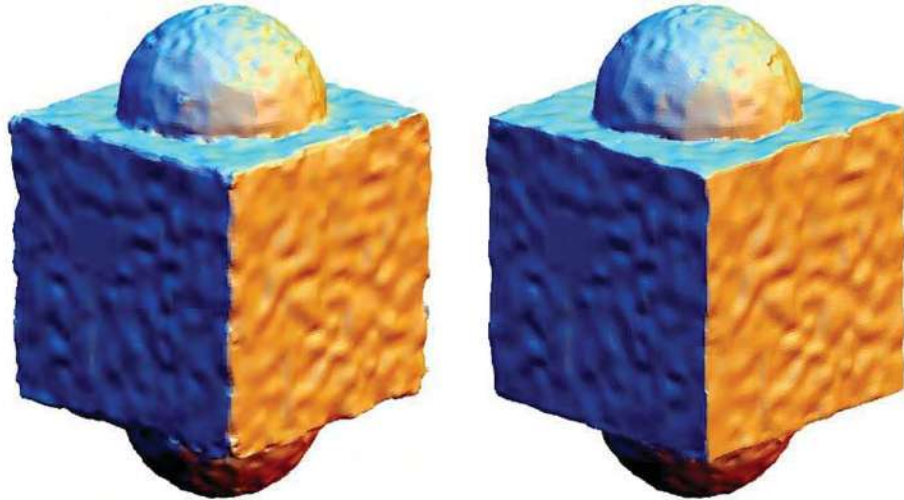
Figura 5.22: Reconstrucciones agregando ruido global, con un factor de 0.01.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.23: Reconstrucciones agregando ruido global, con un factor de 0.0125.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.24: Reconstrucciones agregando ruido global, con un factor de 0.015.

Factor de ruido	Hoppe et al.		Propuesta	
	Promedio	Desviación Estándar	Promedio	Desviación Estándar
0.005	4.2807	7.8364	2.7476	11.1045
0.0075	4.4295	7.7830	2.8354	10.7328
0.01	4.5912	7.7712	3.1703	10.7015
0.015	4.9026	7.6555	3.5451	10.1828

Tabla 5.9: Errores de estimación en ambos métodos para el modelo de una pirámide con ruido global.

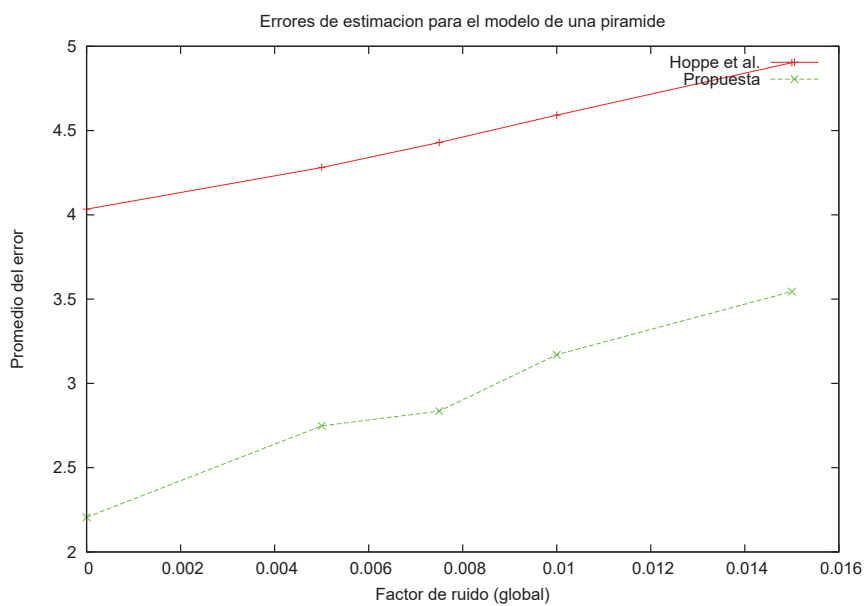
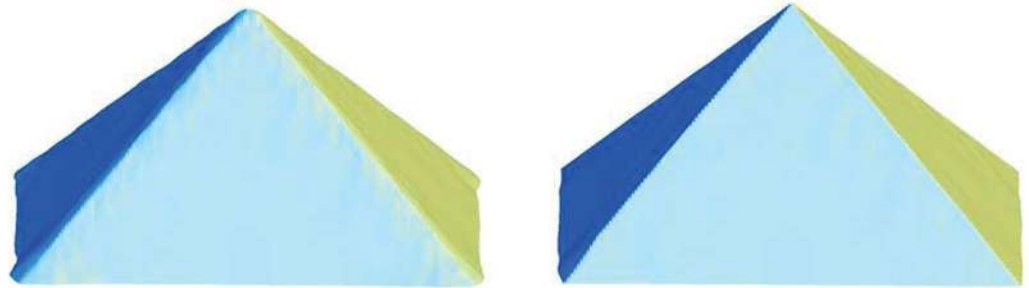


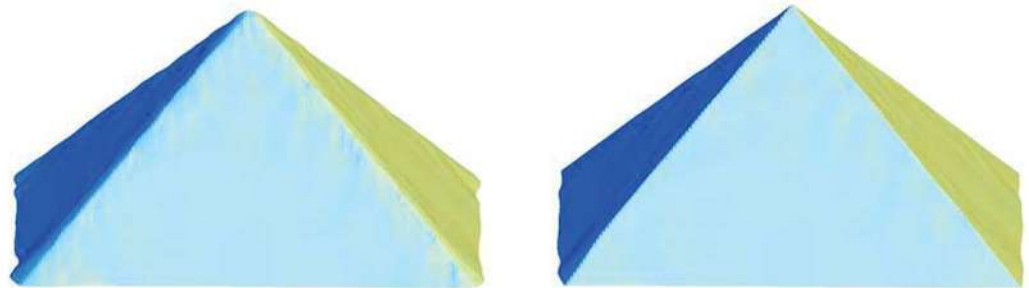
Figura 5.25: Gráfica de los errores de estimación en ambos métodos para el modelo de una pirámide con ruido global.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.26: Reconstrucciones agregando ruido global, con un factor de 0.005.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.27: Reconstrucciones agregando ruido global, con un factor de 0.0075.

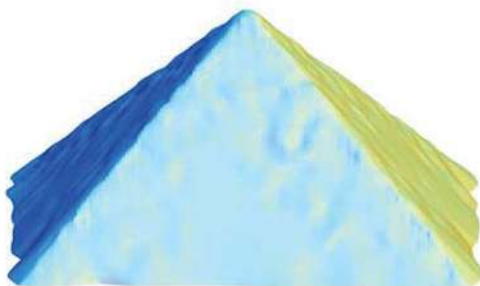


(a) Estimación de normales con el método de Hoppe et al.

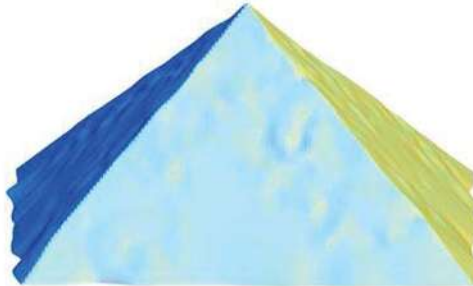


(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.28: Reconstrucciones agregando ruido global, con un factor de 0.01.



(a) Estimación de normales con el método de Hoppe et al.



(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.29: Reconstrucciones agregando ruido global, con un factor de 0.015.

Cilindro

La Tabla 5.10 contiene los resultados de la estimación del error angular de ambos métodos cuando se tiene ruido global. La gráfica con los valores de esta tabla se muestra en la Figura 5.30. Las reconstrucciones correspondientes se presentan en las Figuras 5.31-5.35. Nuevamente, la propuesta de este trabajo produce una mejor estimación de la normal así como una mayor calidad en la reconstrucción cerca de los bordes y esquinas. Las reconstrucciones siguen mostrando bordes más definidos a pesar de la gran cantidad de ruido en la muestra.

Factor de ruido	Hoppe et al.		Propuesta	
	Promedio	Desviación Estándar	Promedio	Desviación Estándar
0.005	5.0209	9.9636	1.7138	1.9882
0.0075	5.093	9.9107	1.7899	1.9556
0.01	5.1670	9.8642	1.8846	2.0326
0.0125	5.2514	9.8166	1.9807	1.9987
0.015	5.3340	9.76	2.0833	1.9997

Tabla 5.10: Errores de estimación en ambos métodos para el modelo de un cilindro con ruido global.

5.4. Conclusiones

Los resultados de los experimentos anteriores, muestran que la propuesta de este trabajo mejora considerablemente la estimación de las normales, permitiendo obtener reconstrucciones de mayor calidad, si la superficie contiene bordes y esquinas. Un beneficio adicional es la tolerancia al ruido, ya es que es posible obtener bordes más definidos aún bajo altos niveles de ruido, superando la estimación del método tradicional. Por lo tanto, podemos concluir que la propuesta viene a incrementar la capacidad de los algoritmos para manejar los problemas que ocurrían tradicionalmente ante la presencia de superficies con bordes agudos y ruido en las muestras.

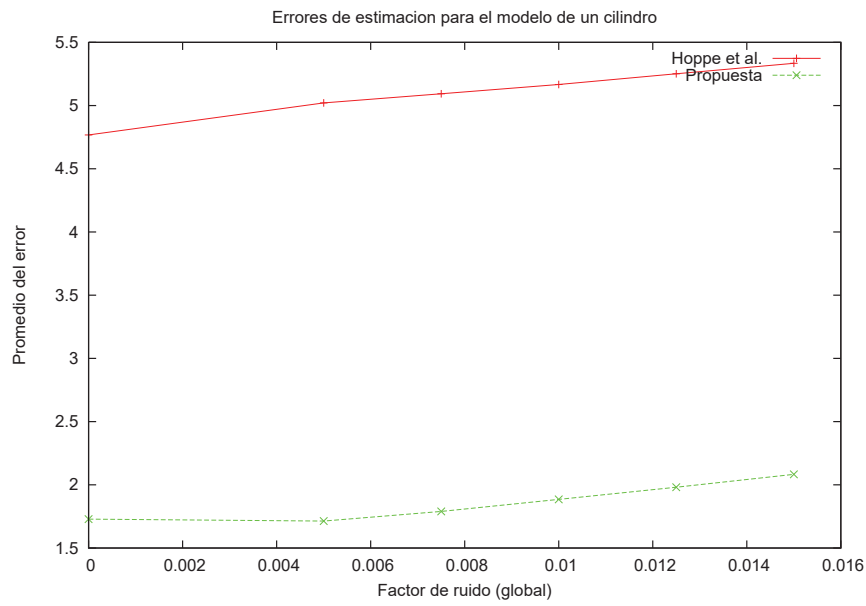


Figura 5.30: Gráfica de los errores de estimación en ambos métodos para el modelo de un cilindro con ruido global.



(a) Estimación de normales con el método de Hoppe et al.



(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.31: Reconstrucciones agregando ruido global, con un factor de 0.005.



(a) Estimación de normales con el método de Hoppe et al.



(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.32: Reconstrucciones agregando ruido global, con un factor de 0.0075.

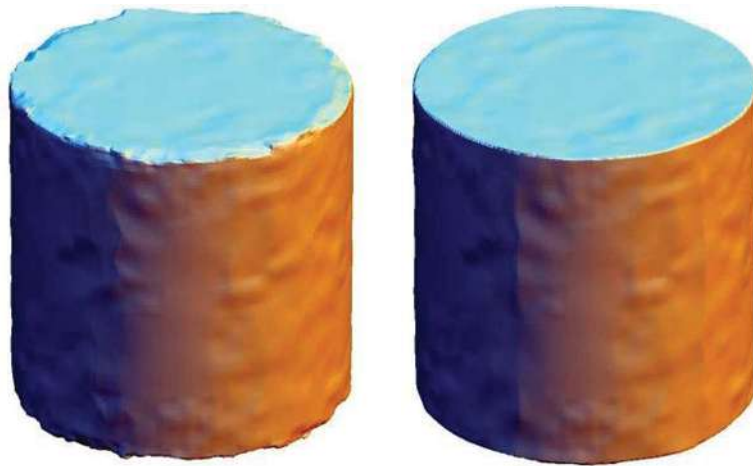


(a) Estimación de normales con el método de Hoppe et al.



(b) Estimación de normales con la propuesta de este trabajo.

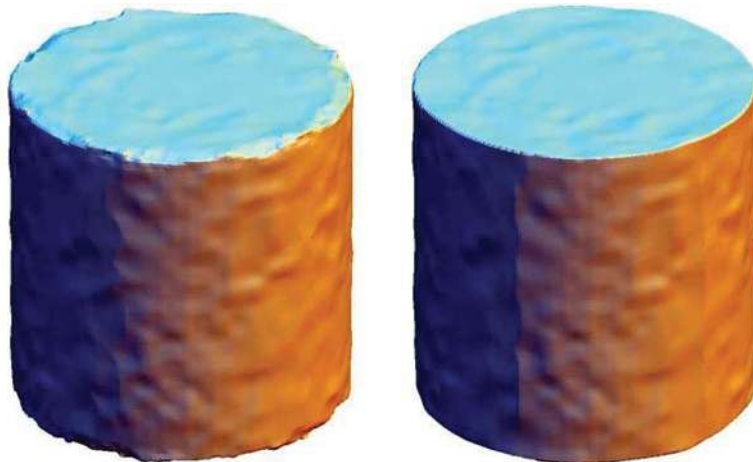
Figura 5.33: Reconstrucciones agregando ruido global, con un factor de 0.01.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.34: Reconstrucciones agregando ruido global, con un factor de 0.0125.



(a) Estimación de normales con el método de Hoppe et al.

(b) Estimación de normales con la propuesta de este trabajo.

Figura 5.35: Reconstrucciones agregando ruido global, con un factor de 0.015.

Capítulo 6

Conclusiones

A continuación se exponen las conclusiones generales de la tesis y algunos de los trabajos que pueden desarrollarse a futuro, en el problema de la estimación de normales a partir de conjuntos de puntos para reconstrucción 3D.

6.1. Conclusiones generales

En este trabajo se presentó un algoritmo de regularización para el cálculo de normales a partir de conjuntos de puntos en 3D. La propuesta permite mejorar las estimaciones cerca de bordes y esquinas, regiones que tradicionalmente causan problemas con los algoritmos existentes.

El algoritmo de regularización produce una reducción en el tamaño de la vecindad, descartando aquellos elementos cuyas normales tienen grandes diferencias. Esto permite que los vecinos tengan las mismas condiciones de suavidad entre ellos.

Ante la carencia de modelos reales que incluyan un conjunto de normales de referencia, el algoritmo de regularización se probó con tres modelos sintéticos que combina regiones con discontinuidades (bordes y esquinas) y regiones suaves. Los experimentos presentaron mejores resultados cuantitativos y cualitativos empleando la propuesta de este trabajo. Se efectuaron experimentos adicionales para conocer el comportamiento del método propuesto ante la presencia de ruido. El modelo sintético fue contaminado con dos tipos de ruido, bajo condiciones controladas. Los experimentos muestran una mejor estimación

de la normal para ambos tipos de ruidos en comparación a la estimación por el método tradicional. En los resultados es posible observar como el método propuesto permite obtener reconstrucciones con bordes definidos en datos con niveles altos de ruido.

6.2. Trabajo futuro

La selección de los parámetros del algoritmo depende del conjunto de puntos. Actualmente esta selección se realiza de forma manual y la elección de los parámetros que conducen a la mejor estimación de las normales se hace visualmente. Una contribución interesante es desarrollar un método para calcularlos de manera automática.

Una aplicación adicional de la propuesta podría utilizar las normales calculadas para diseñar algún tipo de filtrado en conjuntos de puntos con ruido. Existen algunos trabajos relacionados que funcionan sobre mallas triangulares, en las cuales se tiene la conectividad entre los vértices (los elementos en cada vecindad son conocidos). En la mayoría de esos trabajos primero se efectúa un filtrado sobre las normales en los vértices de la malla. Las normales filtradas son utilizadas en el cálculo de las nuevas posiciones de los vértices. Un experimento interesante es utilizar las normales y vecindades calculadas con el algoritmo de este trabajo en combinación con alguna de las técnicas existentes para calcular las nuevas posiciones de los puntos.

El algoritmo de orientación consistente de las normales presentó fallas para un modelo particular. Las normales se pueden orientar consistentemente únicamente para ciertos tamaños de la vecindad. En [Uhercik05] proponen una modificación al algoritmo estándar. Es necesario verificar si esa modificación resuelve los problemas que fueron observados o se debe desarrollar una mejora al algoritmo tradicional.

Apéndice A

Conceptos básicos

A.1. Geometría

En esta sección se presentan algunas de las definiciones y construcciones geométricas básicas ampliamente utilizadas en el área de la reconstrucción de superficies a partir de conjuntos de puntos en 3D.

A.1.1. Diagrama de Voronoi

Dado un conjunto finito de puntos en el plano, la idea es asignar a cada punto una región de influencia de tal manera que las regiones descompongan al plano. Una forma específica de hacerlo es la siguiente:

Sea $S \subseteq \mathbb{R}^2$ un conjunto de n puntos. La región (celda) de Voronoi V_p de $p \in S$ se define como el lugar geométrico de los puntos $x \in \mathbb{R}^2$ que están a una distancia menor o igual a p que a cualquier otro punto en S . Esto es:

$$V_p = \{x \in \mathbb{R}^2 \mid \|x - p\| \leq \|x - q\|, \forall q \in S\} \quad (\text{A.1})$$

La definición anterior se ilustra en la Figura A.1.

Consideresen los medios-planos de puntos al menos tan cerca a p como a q :

$$H_{pq} = \{x \in \mathbb{R}^2 \mid \|x - p\| \leq \|x - q\|\}. \quad (\text{A.2})$$

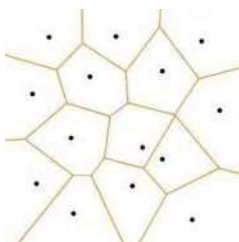


Figura A.1: Diagrama de Voronoi. Fuente: Wikipedia.

La región de Voronoi de p es la intersección de los medios-planos H_{pq} , para todo $q \in S - \{p\}$. Se tiene que V_p es una región poligonal convexa, posiblemente acotada, con a lo más $n - 1$ lados. Cada punto $x \in \mathbb{R}^2$ tiene al menos un punto cercano en S , por lo tanto, al menos cae dentro de una región de Voronoi. Dos regiones de Voronoi caen en lados opuestos de la bisectriz separando los puntos generadores. Se tiene que las regiones de Voronoi no comparten puntos interiores y si un punto x pertenece a dos regiones de Voronoi, entonces está sobre la bisectriz de los dos generadores. Las regiones de Voronoi junto con sus lados y vértices forman el diagrama de Voronoi de S .

A.1.2. Triangulación de Delaunay

Se obtiene un diagrama dual si se dibuja una línea (lados Delaunay) conectando los puntos $p, q \in S$ si y solo si sus regiones de Voronoi se intersectan a lo largo de un segmento común, ver Figura A.2.

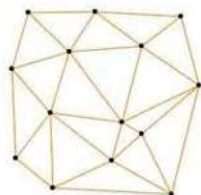


Figura A.2: Triangulación de Delaunay. Fuente: Wikipedia.

En general, los lados Delaunay descomponen la envolvente convexa de S en regiones

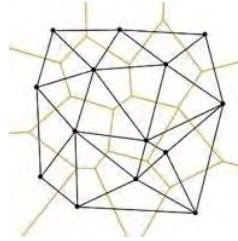


Figura A.3: Diagrama de Voronoi (naranja) y Triangulación de Delaunay (negro). Fuente: Wikipedia.

triangulares, que son conocidas como triángulos Delaunay.

A.1.3. Complejo simplicial

Los complejos simpliciales se emplean como la herramienta fundamental para modelar las formas geométricas o espacios. Estos generalizan y formalizan las nociones geométricas de triangulación. Debido a su naturaleza combinatoria, los complejos simpliciales son estructuras de datos perfectas para los algoritmos de modelado geométrico.

Simplejos

Una colección finita de puntos es afinmente independiente si ningún espacio afín de dimensión i contiene más de $i + 1$ de los puntos. Un k -simplejo es la envolvente convexa de una colección de $k + 1$ puntos afinmente independientes, $\sigma = \text{conv } S$. La dimensión de σ es $\dim \sigma = k$. En \mathbb{R}^d , el número más grande de puntos afinmente independientes es $d + 1$ y se tienen simplejos de dimensión $-1, 0, \dots, d$. El (-1) -simplejo es el conjunto vacío. La Figura A.4 muestra los tipos de simplejos no vacíos en \mathbb{R}^3 .

La envolvente convexa de cualquier subconjunto $T \subseteq S$ es un simplejo. Este es un subconjunto de $\text{conv } S$ y es llamado una cara de σ , la cual se denota por $\tau \leq \sigma$. Si la $\dim \tau = l$ entonces τ es llamada una l -cara. $\tau = \emptyset$ y $\tau = \sigma$ reciben el nombre de caras impropias, las restantes son caras propias de σ . El número de l caras de σ es igual al número de formas en que podemos elegir $l + 1$ puntos de un conjunto de $k + 1$, que es:

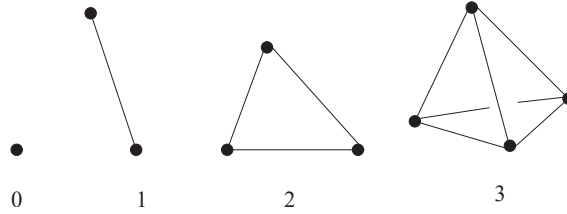


Figura A.4: Un 0-simplejo es un punto o vértice, un 1-simplejo es un lado, un 2-simplejo es un triángulo y un 3-simplejo es un tetraedro. Adaptado de [Edelsbrunner01].

$$\binom{k+1}{l+1} \quad (\text{A.3})$$

El número total de caras es:

$$\sum_{l=-1}^k \binom{k+1}{l+1} = 2^{k+1} \quad (\text{A.4})$$

Complejo simplicial

Un complejo simplicial es la colección de caras de un conjunto finito de simplejos, donde dos cualesquiera de ellos son disjuntos o coinciden en una cara común. De manera formal, es una colección K tal que

1. $\sigma \in K \wedge \tau \leq \sigma \implies \tau \in K$
2. $\sigma, v \in K \implies \sigma \cap v \leq \sigma, v$

Nótese que \emptyset es una cara de todo simplejo y por lo tanto pertenece a K por la condición 1. La condición 2 permite la posibilidad de que σ y v sean disjuntos. Un subcomplejo es un subconjunto que es un conjunto simplicial por si mismo. Debe observarse que todo subconjunto de un complejo simplicial satisface la condición 2. Para reforzar la condición 1, pueden añadirse caras y simplejos al subconjunto. De manera formal, la cerradura de un subconjunto $L \subseteq K$ es el subcomplejo más pequeño que contiene L .

A.1.4. Forma α

Dada una triangulación de Delaunay D de un conjunto de puntos S , uno puede asignar a cada simplejo $\sigma \in D$ un tamaño definido de la siguiente manera. Sea Θ_σ la bola más pequeña cuya frontera contiene todos los vértices de σ . El tamaño de σ se define igual al radio de Θ_σ y σ se dice que está libre de conflicto si Θ_σ no contiene ningún punto de S distinto a los vértices de σ . El subcomplejo Σ_α de simplejos $\sigma \in D$ es llamado una forma α de S si tiene alguna de las siguientes propiedades: El tamaño de σ es menor que α y σ es libre de conflicto o σ es una cara de τ y $\tau \in \Sigma_\alpha$.

A.2. Espacios topológicos

Las nociones topológicas usadas en el área de la reconstrucción de superficies son predominantemente combinatorias. Para entender la conexión a los fenómenos continuos son necesarios algunos conceptos básicos de topología de conjuntos. A continuación se presenta una breve introducción a los espacios topológicos, homeomorfismos, triangulaciones y variedades.

A.2.1. Topología

El concepto fundamental en topología de conjuntos es el de espacio topológico, que es un conjunto de puntos \mathbb{X} con una colección X de subconjuntos $A \subseteq \mathbb{X}$ que satisfacen:

1. $\emptyset, \mathbb{X} \in X$
2. $Z \subseteq X$ implica que $\cup Z \in X$
3. $Z \subseteq X$ y Z implica que $\cap Z \in X$

La colección X es una topología y sus subconjuntos reciben el nombre de conjuntos abiertos en \mathbb{X} . Uno de los ejemplos más importantes es el espacio euclidiano d -dimensional, denotado como \mathbb{R}^d . Haciendo uso de la función de distancia euclidiana se define una bola abierta como el conjunto de todos los puntos más cercanos que una distancia dada a un punto. La topología de \mathbb{R}^d es la colección de conjuntos abiertos, donde cada conjunto abierto es la unión de bolas abiertas. Todos los espacios topológicos en el área de la reconstrucción de superficies son subconjuntos de \mathbb{R}^d . Un subespacio topológico del par \mathbb{X}, X es un subconjunto $\mathbb{Y} \subseteq \mathbb{X}$ junto con la topología del subespacio que consiste en todas las intersecciones entre \mathbb{Y} y los conjuntos abiertos, $Y = \{\mathbb{Y} \cap A \mid A \in X\}$. Un ejemplo es la d -bola, definida como el conjunto de puntos a una distancia 1 o menor del origen,

$$\mathbb{B}^d = \{x \in \mathbb{R}^d \mid \|x\| \leq 1\} \quad (\text{A.5})$$

Sus conjuntos abiertos son las intersecciones de \mathbb{B}^d con los conjuntos abiertos en \mathbb{R}^d . Nótese que un conjunto abierto en \mathbb{B}^d no es necesariamente abierto en \mathbb{R}^d .

A.2.2. Homeomorfismo

Dos espacios topológicos se consideran el mismo o del mismo tipo si están conectados de la misma manera. El significado de *conectados de la misma manera* tiene que ser definido, existiendo varias posibilidades, donde la más importante está basada en homeomorfismos, los cuales son funciones entre espacios topológicos. En este caso una función es continua si la preimagen de cada conjunto abierto es abierta. Un homeomorfismo es una función $f : \mathbb{X} \rightarrow \mathbb{Y}$ que es biyectiva, continua y tiene una inversa continua. Si existe un homeomorfismo entonces se dice que \mathbb{X} y \mathbb{Y} son homeomorfos, en este caso, esto será denotado como $\mathbb{X} \approx \mathbb{Y}$. Si se quiere enfatizar que \approx es una relación de equivalencia, se dice \mathbb{X} y \mathbb{Y} son topologicamente equivalentes o que tienen el mismo tipo topológico. La Figura A.5 muestra ejemplos de espacios en una dimensión con diferentes tipos topológicos por parejas. Como ejemplo adicional, considere el disco abierto unitario, que es el conjunto de puntos en \mathbb{R}^2 a una distancia menor que uno al origen. Este disco puede ser estirado sobre todo el plano. Sea $f(x) = x/(1 - \|x\|)$, la función es biyectiva y continua, además su inversa es continua. De lo anterior, se tiene que el disco abierto es homeomorfo a \mathbb{R}^2 . De forma general, toda bola abierta k -dimensional es homeomorfa a \mathbb{R}^k .

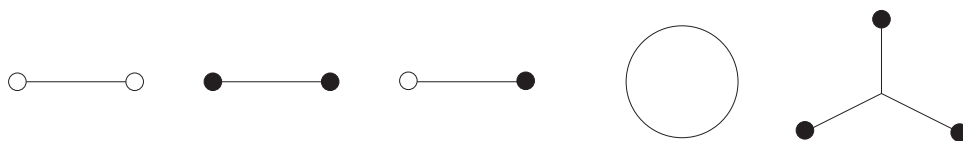


Figura A.5: De izquierda a derecha, el intervalo abierto, el intervalo cerrado, el intervalo medioabierto, el círculo y una bifurcación. Adaptado de [Edelsbrunner01].

A.2.3. Triangulación

Este término cambia de significado de un área a otra. En geometría, no existe una definición aceptada generalmente, sin embargo, en la mayoría de los casos significa un complejo simplicial. En topología, una triangulación tiene un significado preciso, el cual es similar a la idea de una malla que da una estructura combinatoria al espacio. Sea K un complejo simplicial en \mathbb{R}^d . Su espacio subyacente es la unión de sus simplejos junto con la

topología del subespacio heredada de \mathbb{R}^d .

$$|K| = \{x \in \mathbb{R}^d \mid x \in \sigma \in K\} \quad (\text{A.6})$$

Un poliedro es el espacio subyacente de un complejo simplicial. Se puede pensar en K como una estructura combinatoria impuesta sobre $|K|$. Existe algunas otras. Empleando homeomorfismos, se puede imponer la misma estructura sobre espacios que no son poliedros. Una triangulación de un espacio topológico \mathbb{X} es un complejo simplicial K cuyo espacio subyacente es homeomorfo a \mathbb{X} , $|K| \approx \mathbb{X}$. El espacio \mathbb{X} es triangulable si este tiene una triangulación. Un ejemplo, es la triangulación del disco cerrado \mathbb{B}^2 con nueve triángulos, como se muestra en la Figura A.6.

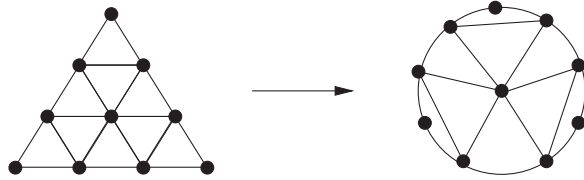


Figura A.6: Triangulación de un disco cerrado. El homeomorfismo mapea cada vértice, lado y triángulo a un subconjunto homeomorfo del disco. Adaptado de [Edelsbrunner01].

A.2.4. Variedades

Las variedades son un tipo de espacio topológico y están definidas localmente. Una vecindad de un punto $x \in \mathbb{X}$ es un conjunto abierto que contiene a x . Existen una gran cantidad de vecindades, pero usualmente basta tomar una que es lo suficientemente pequeña. Un espacio topológico \mathbb{X} es una k -variedad si cada $x \in \mathbb{X}$ tiene una vecindad homeomorfa a \mathbb{R}^k . En ocasiones resulta más intuitivo sustituir \mathbb{R}^k por una k -bola abierta, algo que no causa diferencia debido a que ambos son homeomorfos. Un ejemplo simple de una k -variedad es la k -esfera, que es el conjunto de puntos a una distancia unitaria del origen en el espacio euclidiano $(k + 1)$ -dimensional.

$$\mathbb{S}^k = \{x \in \mathbb{R}^{k+1} \mid \|x\| = 1\} \quad (\text{A.7})$$

Algunos ejemplos se muestran en la Figura A.7. La triangulación más pequeña de \mathbb{S}^k es el complejo frontera de un $(k + 1)$ -simplejo σ . Para construir un homeomorfismo, se coloca σ de tal forma que contenga el origen en su interior y se proyecta centralmente cada punto de la frontera de σ sobre la esfera.

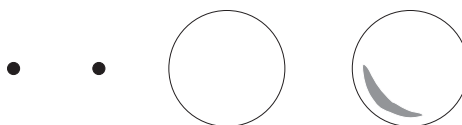


Figura A.7: La 0-esfera es un par de puntos, la 1-esfera es un círculo y la 2-esfera es lo que usualmente se conoce como una esfera. Adaptado de [Edelsbrunner01].

A.2.5. Variedades con fronteras

Todos los puntos de una variedad tienen la misma vecindad. Se obtiene una clase más general de espacios si se permite tener dos tipos de vecindades. El segundo tipo es la media bola abierta. Nuevamente se puede estirar este espacio, en esta ocasión sobre la mitad del espacio euclidiano de la misma dimensión. Formalmente, el medio plano k -dimensional es

$$\mathbb{H}^k = \{x = (x_1, x_2, \dots, x_k) \in \mathbb{R}^k \mid x_1 \geq 0\} \quad (\text{A.8})$$

Un espacio \mathbb{X} es una k -variedad con frontera si todo punto $x \in \mathbb{X}$ tiene una vecindad homeomorfa a \mathbb{R}^k o a \mathbb{H}^k . La frontera es el conjunto de puntos con una vecindad homeomorfa a \mathbb{H}^k y está denotada como $\text{bd } \mathbb{X}$. Un ejemplo de una k -variedad con una frontera (no vacía) es la k -bola, ver Figura A.8. Su frontera es la $(k - 1)$ -esfera, $\text{bd } \mathbb{B}^k = \mathbb{S}^{k-1}$.

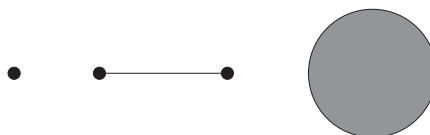


Figura A.8: La 0-bola es un punto, la 1-bola es un intervalo cerrado y la 2-bola es un disco cerrado. Adaptado de [Edelsbrunner01].

A.2.6. Orientabilidad

Las variedades con o sin frontera pueden ser orientables o no orientables, la distinción es una propiedad global que no puede ser observada localmente. De manera intuitiva, imagínese una hormiga $(k + 1)$ -dimensional caminando sobre una k -variedad. En cualquier momento, la hormiga está en un lado de la vecindad local con la cual está en contacto. La variedad es no orientable si existe un recorrido que trae a la hormiga de vuelta a la misma vecindad pero ahora del otro lado y es orientable si esta ruta no existe. Un par de ejemplos de variedades no orientables con y sin fronteras se presentan en la Figura A.9. Imagínese la frontera de una forma sólida en nuestro espacio tres dimensional. Esta frontera es una 2-variedad y acota el interior de la forma por una lado y el exterior por el otro. La 2-variedad por lo tanto será orientable. Resulta que toda 2-variedad embebida en \mathbb{R}^3 separa el interior del exterior y es por lo tanto orientable.

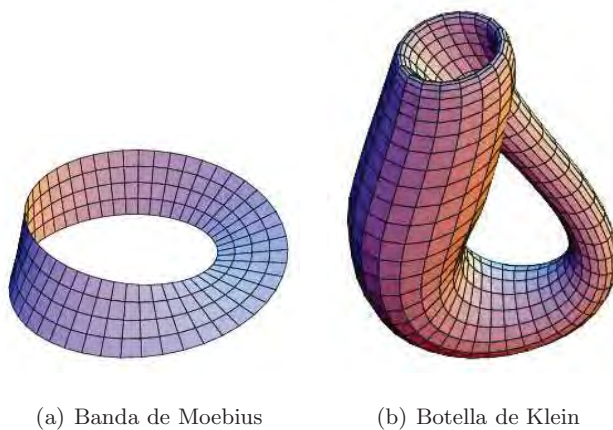


Figura A.9: Ejemplos de variedades no orientables con y sin fronteras. Fuente: Wikipedia.

Referencias

- [Adamson03] Adamson, A. y Alexa, M. Approximating and intersecting surfaces from points. *En SGP '03: Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing*, págs. 230–239. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, 2003. ISBN 1-58113-687-0.
- [Amenta98a] Amenta, N. y Bern, M. Surface reconstruction by voronoi filtering. *En SCG '98: Proceedings of the fourteenth annual symposium on Computational geometry*, págs. 39–48. ACM Press, New York, NY, USA, 1998. ISBN 0-89791-973-4. doi:<http://doi.acm.org/10.1145/276884.276889>.
- [Amenta98b] Amenta, N., Bern, M., y Eppstein, D. The crust and the β -skeleton: combinatorial curve reconstruction. *Graph. Models Image Process.*, 60(2):125–135, 1998. ISSN 1077-3169. doi:<http://dx.doi.org/10.1006/gmip.1998.0465>.
- [Amenta98c] Amenta, N., Bern, M., y Kamvysselis, M. A new voronoi-based surface reconstruction algorithm. *En SIGGRAPH '98: Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, págs. 415–421. ACM Press, New York, NY, USA, 1998. ISBN 0-89791-999-8. doi:<http://doi.acm.org/10.1145/280814.280947>.
- [Amenta00] Amenta, N., Choi, S., Dey, T. K., y Leekha, N. A simple algorithm for homeomorphic surface reconstruction. *En SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry*, págs.

- 213–222. ACM Press, New York, NY, USA, 2000. ISBN 1-58113-224-7. doi:<http://doi.acm.org/10.1145/336154.336207>.
- [Amenta01] Amenta, N., Choi, S., y Kolluri, R. K. The power crust. *En SMA '01: Proceedings of the sixth ACM symposium on Solid modeling and applications*, págs. 249–266. ACM Press, New York, NY, USA, 2001. ISBN 1-58113-366-9. doi:<http://doi.acm.org/10.1145/376957.376986>.
- [Amenta04] Amenta, N. y Kil, Y. J. Defining point-set surfaces. *En SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, págs. 264–270. ACM Press, New York, NY, USA, 2004. doi:<http://doi.acm.org/10.1145/1186562.1015713>.
- [Attali98] Attali, D. r-regular shape reconstruction from unorganized points. *Comput. Geom. Theory Appl.*, 10(4):239–247, 1998. ISSN 0925-7721. doi:[http://dx.doi.org/10.1016/S0925-7721\(98\)00013-3](http://dx.doi.org/10.1016/S0925-7721(98)00013-3).
- [Attene00] Attene, M. y Spagnuolo, M. Automatic surface reconstruction from point sets in space. *Computer Graphics Forum*, 19(3):457–465, 2000. URL citeseer.ist.psu.edu/attene00automatic.html
- [Bajaj95a] Bajaj, C. L., Bernardini, F., y Xu, G. Automatic reconstruction of surfaces and scalar fields from 3d scans. *En SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, págs. 109–118. ACM Press, New York, NY, USA, 1995. ISBN 0-89791-701-4. doi:<http://doi.acm.org/10.1145/218380.218424>.
- [Bajaj95b] Bajaj, C. L., Chen, J., y Xu, G. Modeling with cubic a-patches. *ACM Trans. Graph.*, 14(2):103–133, 1995. ISSN 0730-0301. doi:<http://doi.acm.org/10.1145/221659.221662>.
- [Bernardini97] Bernardini, F. y Bajaj, C. L. Sampling and reconstructing manifolds using alpha-shapes. *En Proc. 9th Canadian Conf. Computational Geometry*, págs. 193–198. 1997. URL <http://citeseer.ist.psu.edu/187879.html>

- [Blinn82] Blinn, J. F. A generalization of algebraic surface drawing. *ACM Trans. Graph.*, 1(3):235–256, 1982. ISSN 0730-0301. doi:<http://doi.acm.org/10.1145/357306.357310>.
- [Bloomenthal94] Bloomenthal, J. An implicit surface polygonizer. págs. 324–349. Academic Press Professional, Inc., San Diego, CA, USA, 1994. ISBN 0-12-336155-9.
- [Boissonnat84] Boissonnat, J.-D. Geometric structures for three-dimensional shape representation. *ACM Trans. Graph.*, 3(4):266–286, 1984. ISSN 0730-0301. doi:<http://doi.acm.org/10.1145/357346.357349>.
- [Boissonnat00] Boissonnat, J.-D. y Cazals, F. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *En SCG '00: Proceedings of the sixteenth annual symposium on Computational geometry*, págs. 223–232. ACM Press, New York, NY, USA, 2000. ISBN 1-58113-224-7. doi:<http://doi.acm.org/10.1145/336154.336208>.
- [Brigger00] Brigger, P., Hoeg, J., y Unser, M. B-Spline snakes: A flexible tool for parametric contour detection. *IEEE Transactions on Image Processing*, 9(9):1484–1496, September 2000.
- [Buhmann03] Buhmann, M. D. *Radial Basis Functions: theory and implementations*. Cambridge University Press, New York, NY, USA, 2003. ISBN 0521633389.
- [Calderon03] Calderon, F. y Marroquin, J. L. Un nuevo algoritmo para calculo de flujo optico y su aplicacion al registro de imagenes. *Computación y Sistemas*, 6(3):213–225, 2003.
- [Calderon06a] Calderon, F. Notas de procesamiento digital de señales. Inf. téc., UMSNH, 2006.
URL <http://lc.fie.umich.mx/calderon>

- [Calderon06b] Calderon, F., Romero, L., y Flores, J. J. GA-SSD-ARC-NLM for parametric image registration. *En Progress in Pattern Recognition, Image Analysis and Applications, 11th Iberoamerican Congress in Pattern Recognition, CIARP 2006, Cancun, Mexico, November 14-17, 2006, Proceedings*, tomo 4225 de *Lecture Notes in Computer Science*, págs. 227–236. Springer, 2006. ISBN 3-540-46556-1.
- [Carr01] Carr, J. C., Beatson, R. K., Cherrie, J. B., Mitchell, T. J., Fright, W. R., McCallum, B. C., y Evans, T. R. Reconstruction and representation of 3d objects with radial basis functions. *En SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, págs. 67–76. ACM Press, New York, NY, USA, 2001. ISBN 1-58113-374-X. doi:<http://doi.acm.org/10.1145/383259.383266>.
- [Cazals04] Cazals, F. y Giesen, J. Delaunay based surface reconstruction algorithms: Ideas and algorithm. Research Report 5393, INRIA, 2004. URL <http://cgall.inria.fr/Publications/2004/CG04>
- [Cazals06] Cazals, F. y Giesen, J. Delaunay triangulation based surface reconstruction. Inf. téc., INRIA, 2006. URL <http://cgall.inria.fr/Publications/2006/CG06>
- [Chapra97] Chapra, S. C. y Canale, R. P. *Numerical Methods for Engineers: With Programming and Software Applications*. McGraw-Hill, Inc., New York, NY, USA, 1997. ISBN 0070109389.
- [Chen95] Chen, Y. y Medioni, G. Description of complex objects from multiple range images using an inflating balloon model. *Comput. Vis. Image Underst.*, 61(3):325–334, 1995. ISSN 1077-3142. doi:<http://dx.doi.org/10.1006/cviu.1995.1026>.
- [Curless96] Curless, B. y Levoy, M. A volumetric method for building complex models from range images. *En SIGGRAPH '96: Proceedings of the*

- 23rd annual conference on Computer graphics and interactive techniques*, págs. 303–312. ACM Press, New York, NY, USA, 1996. ISBN 0-89791-746-4. doi:<http://doi.acm.org/10.1145/237170.237269>.
- [Curless98] Curless, B. L. *New methods for surface reconstruction from range images*. Tesis Doctoral, Stanford, CA, USA, 1998.
- [Dahmen93] Dahmen, W. y Thamm-Schaar, T.-M. Cubicoids: Modeling and visualization. *Comput. Aided Geom. Des.*, 10(2):89–108, 1993. ISSN 0167-8396. doi:[http://dx.doi.org/10.1016/0167-8396\(93\)90013-S](http://dx.doi.org/10.1016/0167-8396(93)90013-S).
- [Dey99a] Dey, T. K. y Kumar, P. A simple provable algorithm for curve reconstruction. *En SODA '99: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms*, págs. 893–894. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1999. ISBN 0-89871-434-6.
- [Dey99b] Dey, T. K., Mehlhorn, K., y Ramos, E. A. Curve reconstruction: connecting dots with good reason. *En SCG '99: Proceedings of the fifteenth annual symposium on Computational geometry*, págs. 197–206. ACM Press, New York, NY, USA, 1999. ISBN 1-58113-068-6. doi:<http://doi.acm.org/10.1145/304893.304972>.
- [Dey01] Dey, T. K., Giesen, J., y Hudson, J. Delaunay based shape reconstruction from large data. *En PVG '01: Proceedings of the IEEE 2001 symposium on parallel and large-data visualization and graphics*, págs. 19–27. IEEE Press, Piscataway, NJ, USA, 2001. ISBN 0-7803-7223-9.
- [Dey03] Dey, T. K. y Goswami, S. Tight cocone: a water-tight surface reconstructor. *En SM '03: Proceedings of the eighth ACM symposium on Solid modeling and applications*, págs. 127–134. ACM Press, New York, NY, USA, 2003. ISBN 1-58113-706-0. doi:<http://doi.acm.org/10.1145/781606.781627>.

- [Dey04] Dey, T. K. y Goswami, S. Provable surface reconstruction from noisy samples. *En SCG '04: Proceedings of the twentieth annual symposium on Computational geometry*, págs. 330–339. ACM Press, New York, NY, USA, 2004. ISBN 1-58113-885-7. doi: <http://doi.acm.org/10.1145/997817.997867>.
- [Dey05] Dey, T., Li, G., y Sun, J. Normal estimation for point clouds : A comparison study for a voronoi based method. *En Point-Based Graphics, 2005. Eurographics/IEEE VGTC Symposium Proceedings*, págs. 39–46. IEEE Computer Society, 2005.
- [Dey06] Dey, T. K. *Curve and Surface Reconstruction : Algorithms with Mathematical Analysis*. Cambridge University Press, 2006.
- [Edelsbrunner83] Edelsbrunner, H., Kirkpatrick, D. G., y Seidel, R. On the shape of a set of points in the plane. *IEEE Transactions on Information Theory*, 29(4):551–558, 1983.
- [Edelsbrunner92] Edelsbrunner, H. y Mücke, E. P. Three-dimensional alpha shapes. *En VVS '92: Proceedings of the 1992 workshop on Volume visualization*, págs. 75–82. ACM Press, New York, NY, USA, 1992. ISBN 0-89791-527-5. doi:<http://doi.acm.org/10.1145/147130.147153>.
- [Edelsbrunner01] Edelsbrunner, H. *Geometry and topology for mesh generation*. Cambridge University Press, New York, NY, USA, 2001. ISBN 0-521-79309-2.
- [Giesen99] Giesen, J. Curve reconstruction, the traveling salesman problem and menger's theorem on length. *En SCG '99: Proceedings of the fifteenth annual symposium on Computational geometry*, págs. 207–216. ACM Press, New York, NY, USA, 1999. ISBN 1-58113-068-6. doi: <http://doi.acm.org/10.1145/304893.304973>.
- [Girosi93] Girosi, F., Jones, M., y Poggio, T. Priors stabilizers and basis functions:

- From regularization to radial, tensor and additive splines. Inf. téc., Cambridge, MA, USA, 1993.
- [Gopi00] Gopi, M., Krishnan, S., y Silva, C. T. Surface reconstruction based on lower dimensional localized delaunay triangulation. *Comput. Graph. Forum*, 19(3), 2000.
- [Greengard87] Greengard, L. y Rokhlin, V. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987. ISSN 0021-9991. doi: [http://dx.doi.org/10.1016/0021-9991\(87\)90140-9](http://dx.doi.org/10.1016/0021-9991(87)90140-9).
- [Gumhold01] Gumhold, S., Wang, X., y McLeod, R. Feature extraction from point clouds. *En Proc. 10th Int. Meshing Roundtable, 2001*. 2001.
URL citeseer.ist.psu.edu/gumhold01feature.html
- [Guo91] Guo, B. Surface generation using implicit cubics. *En Scientific visualization of physical phenomena*, págs. 485–503. Springer-Verlag New York, Inc., New York, NY, USA, 1991. ISBN 0-387-70081-1.
- [Guo93] Guo, B. Nonsplitting macro patches for implicit cubic spline surfaces. *Comput. Graph. Forum*, 12(3):433–445, 1993.
- [Hart96] Hart, J. C. Sphere tracing: A geometric method for the antialiased ray tracing of implicit surfaces. *The Visual Computer*, 12(10):527–545, December 1996.
- [Hoppe92] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., y Stuetzle, W. Surface reconstruction from unorganized points. *En SIGGRAPH '92: Proceedings of the 19th annual conference on Computer graphics and interactive techniques*, págs. 71–78. ACM Press, New York, NY, USA, 1992. ISBN 0-89791-479-1. doi:<http://doi.acm.org/10.1145/133994.134011>.
- [Hoppe93] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., y Stuetzle, W. Mesh optimization. *En SIGGRAPH '93: Proceedings of the 20th annual*

- conference on Computer graphics and interactive techniques*, págs. 19–26. ACM Press, New York, NY, USA, 1993. ISBN 0-89791-601-8. doi: <http://doi.acm.org/10.1145/166117.166119>.
- [Hoppe94] Hoppe, H., DeRose, T., Duchamp, T., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., y Stuetzle, W. Piecewise smooth surface reconstruction. *En SIGGRAPH '94: Proceedings of the 21st annual conference on Computer graphics and interactive techniques*, págs. 295–302. ACM Press, New York, NY, USA, 1994. ISBN 0-89791-667-0. doi: <http://doi.acm.org/10.1145/192161.192233>.
- [Hoppe95] Hoppe, H. *Surface reconstruction from unorganized points*. Tesis Doctoral, Seattle, WA, USA, 1995.
- [Kobbelt99] Kobbelt, L., Vorsatz, J., Labsik, U., y Seidel, H.-P. A shrink wrapping approach to remeshing polygonal surfaces. *Comput. Graph. Forum*, 18(3):119–130, 1999.
- [Kobbelt04] Kobbelt, L. y Botsch, M. A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(6):801–814, 2004.
- [Kolluri04] Kolluri, R., Shewchuk, J. R., y O'Brien, J. F. Spectral surface reconstruction from noisy point clouds. *En Symposium on Geometry Processing*, págs. 11–21. ACM Press, jul. 2004.
- [Krishnamurthy96] Krishnamurthy, V. y Levoy, M. Fitting smooth surfaces to dense polygon meshes. *En SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, págs. 313–324. ACM Press, New York, NY, USA, 1996. ISBN 0-89791-746-4. doi: <http://doi.acm.org/10.1145/237170.237270>.
- [Levin98] Levin, D. The approximation power of moving least-squares. *Math. Comput.*, 67(224):1517–1531, 1998. ISSN 0025-5718. doi: <http://dx.doi.org/10.1090/S0025-5718-98-00974-0>.

- [Lewiner03] Lewiner, T., Lopes, H., Vieira, A. W., y Tavares, G. Efficient implementation of marching cubes' cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003.
- [Liao95] Liao, C.-W. y Medioni, G. Surface approximation of a cloud of 3d points. *Graph. Models Image Process.*, 57(1):67–74, 1995. ISSN 1077-3169. doi:<http://dx.doi.org/10.1006/gmip.1995.1007>.
- [Lim95] Lim, C. T., Turkiyyah, G. M., Ganter, M. A., y Storti, D. W. Implicit reconstruction of solids from cloud point sets. *En SMA '95: Proceedings of the third ACM symposium on Solid modeling and applications*, págs. 393–402. ACM Press, New York, NY, USA, 1995. ISBN 0-89791-672-7. doi:<http://doi.acm.org/10.1145/218013.218090>.
- [Lorensen87] Lorensen, W. E. y Cline, H. E. Marching cubes: A high resolution 3d surface construction algorithm. *En SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, págs. 163–169. ACM Press, New York, NY, USA, 1987. ISBN 0-89791-227-6. doi:<http://doi.acm.org/10.1145/37401.37422>.
- [Mencl95] Mencl, R. A graph-based approach to surface reconstruction. *Comput. Graph. Forum*, 14(3):445–456, 1995.
- [Mitra03] Mitra, N. J. y Nguyen, A. Estimating surface normals in noisy point cloud data. *En SCG '03: Proceedings of the nineteenth annual symposium on Computational geometry*, págs. 322–328. ACM Press, New York, NY, USA, 2003. ISBN 1-58113-663-3. doi:<http://doi.acm.org/10.1145/777792.777840>.
- [Morse01] Morse, B. S., Yoo, T. S., Chen, D. T., Rheingans, P., y Subramanian, K. R. Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions. *En SMI '01: Proceedings of the International Conference on Shape Modeling & Applications*,

- pág. 89. IEEE Computer Society, Washington, DC, USA, 2001. ISBN 0-7695-0853-7.
- [Muraki91] Muraki, S. Volumetric shape description of range data using *Blobby Model*. En *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, págs. 227–235. ACM Press, New York, NY, USA, 1991. ISBN 0-89791-436-8. doi: <http://doi.acm.org/10.1145/122718.122743>.
- [Neumaier98] Neumaier, A. Solving ill-conditioned and singular linear systems: A tutorial on regularization. *SIAM Rev.*, 40(3):636–666, 1998. ISSN 0036-1445. doi:<http://dx.doi.org/10.1137/S0036144597321909>.
- [Ohtake03a] Ohtake, Y. MPUI implementation. In http://www.mpi-inf.mpg.de/~ohtake/mpu_implicit/, 2003.
- [Ohtake03b] Ohtake, Y., Belyaev, A., Alexa, M., Turk, G., y Seidel, H.-P. Multi-level partition of unity implicits. En *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, págs. 463–470. ACM Press, New York, NY, USA, 2003. ISBN 1-58113-709-5. doi:<http://doi.acm.org/10.1145/1201775.882293>.
- [Ohtake03c] Ohtake, Y., Belyaev, A., y Seidel, H.-P. A multi-scale approach to 3d scattered data interpolation with compactly supported basis functions. En *SMI '03: Proceedings of the Shape Modeling International 2003*, págs. 292. IEEE Computer Society, Washington, DC, USA, 2003. ISBN 0-7695-1909-1.
- [Ohtake04] Ohtake, Y., Belyaev, A. G., y Seidel, H.-P. 3d scattered data approximation with adaptive compactly supported radial basis functions. En *SMI*, págs. 31–39. 2004.
- [O'Rourke00] O'Rourke, J. *Computational geometry in C*. Cambridge University Press, New York, NY, USA, 2000. ISBN 0521649765.

- [Pauly03] Pauly, M., Keiser, R., Kobbelt, L. P., y Gross, M. Shape modeling with point-sampled geometry. *En SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, págs. 641–650. ACM Press, New York, NY, USA, 2003. ISBN 1-58113-709-5. doi:<http://doi.acm.org/10.1145/1201775.882319>.
- [Pentland91] Pentland, A. y Sclaroff, S. Closed-form solutions for physically based shape modeling and recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(7):715–729, 1991. ISSN 0162-8828. doi:<http://dx.doi.org/10.1109/34.85660>.
- [Pratt87] Pratt, V. Direct least-squares fitting of algebraic surfaces. *En SIGGRAPH '87: Proceedings of the 14th annual conference on Computer graphics and interactive techniques*, págs. 145–152. ACM Press, New York, NY, USA, 1987. ISBN 0-89791-227-6. doi:<http://doi.acm.org/10.1145/37401.37420>.
- [Rivera00] Rivera, M. y Marroquin, J. L. The adaptive rest-condition spring system: An edge-preserving regularization technique. *IEEE Int. Conf. on Image Processing (ICIP-2000), IEEE Signal Processing Society, Vancouver, BC, Canada*, II:805–807, September 2000.
- [Rusinkiewicz01] Rusinkiewicz, S. y Levoy, M. Efficient variants of the ICP algorithm. *En Proceedings of the Third Intl. Conf. on 3D Digital Imaging and Modeling*, págs. 145–152. 2001.
- [Sakkalis99] Sakkalis, T. y Charitos, C. Approximating curves via alpha shapes. *CV-GIP: Graph. Models Image Process.*, 61(3):165–176, 1999. ISSN 1049-9652.
- [Saleem04] Saleem, W. *A Flexible Framework for Learning-based Surface Reconstruction*. Proyecto Fin de Carrera, Computer Science Department, University of Saarland, Postfach 15 11 50, 66041 Saarbrücken, dic. 2004.
- [Schall05] Schall, O. y Samozino, M. Surface from scattered points: A brief survey

- of recent developments. *En* B. Falcidieno y N. Magnenat-Thalmann, eds., *1st International Workshop towards Semantic Virtual Environments*, págs. 138–147. MIRALab, Villars, Switzerland, 2005.
- [Sethian99] Sethian, J. A. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [Söderkvist99] Söderkvist, I. Introductory overview of surface reconstruction methods. Inf. Téc. 1999-10, Department of Mathematics, Lulea University of Technology, Lulea, Sweden., 1999. Can be retrieved from <http://www.sm.luth.se/inge/publications/surfrec.ps>.
URL citeseer.ist.psu.edu/soderkvist99introductory.html
- [Taubin91] Taubin, G. Estimation of planar curves, surfaces, and nonplanar space curves defined by implicit equations with applications to edge and range image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13(11):1115–1138, 1991. ISSN 0162-8828. doi: <http://dx.doi.org/10.1109/34.103273>.
- [Tobor04] Tobor, I., Reuter, P., y Schlick, C. Efficient reconstruction of large scattered geometric datasets using the partition of unity and radial basis functions. *Journal of WSCG 2004*, 12(3):467–474, 2004.
- [Uhercik05] Uhercik, M. *Implicit Surface Reconstruction using Local Approximations*. Tesis Doctoral, 2005.
- [Watt99] Watt, A. H. y Watt, A. *3d Computer Graphics with Cdrom*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1999. ISBN 0201398559.
- [Zhao00] Zhao, H.-K., Osher, S., Merriman, B., y Kang, M. Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. *Comput. Vis. Image Underst.*, 80(3):295–314, 2000. ISSN 1077-3142. doi:<http://dx.doi.org/10.1006/cviu.2000.0875>.

Índice

- Árbol Euclidiano de Expansión Mínima,
 - 9, 34
- Árbol de Expansión Mínima, 36
- Algoritmo de cubos en marcha, 19
- Algoritmos geométricos, 7, 8, 14
 - Algoritmo de coconos, 13, 57
 - Algoritmo de corteza, 9, 11, 13
 - Garantías teóricas, 7, 9, 13, 14
 - Restricciones, 7, 14
 - Tratamiento de bajas densidades en la muestra, 14
- Algoritmos volumétricos, 7, 8, 16
 - Funciones cuadráticas, 24
 - Funciones de base radial, 20, 24
 - Métodos basados en conceptos físicos, 26
 - Métodos de conjunto de nivel, 26
 - Mínimos cuadrados móviles, 25, 26
 - Partición de Unidad, 22, 23, 27
 - Reconstrucción de bordes y esquinas, 8, 22, 24, 27
 - Técnicas basadas en modelos deformables, 26
 - Técnicas implícitas, 16
 - Tolerancia al ruido, 8
 - Ventajas, 8
- Cámaras de rango, 1
- Conjunto de nivel cero, 8, 16
- Conjunto de puntos, 1, 2, 7–10, 16, 17, 20–22, 25–27, 29, 30, 57, 91
- Distancia signada, 17–19, 23
- Estimación de normales, 4, 16, 26, 27, 29, 55, 57, 81
 - Ajuste adaptativo, 37
 - Ajuste con peso, 37
 - Comparación cualitativa, 55, 64, 67, 71, 74, 81
 - Comparación cuantitativa, 55, 59, 61, 62, 64, 68, 71, 74, 78, 81
 - Error, 58
 - Estimación cerca de bordes y esquinas, 6, 55, 59, 61, 74
 - Experimento
 - Errores, 59, 61, 64, 67, 71
 - Factores de ruido, 58
 - Mínimos cuadrados, 17, 26, 29, 30
 - Matriz de covarianza, 29, 30, 37, 55

- Mejorar el cálculo, 6, 39, 55, 64, 67, 74, 81
- Nivel de ruido, 58
- Global, 58
 - Local, 58
- Orientación consistente, 4, 31, 34
- Plano tangente, 4, 17–19, 29, 30, 34, 35
- Centro, 17, 30, 35
 - Vector normal unitario, 17, 29, 30, 37
- Planos tangentes, 19
- Planteamiento alternativo, 33
- Problema en bordes y esquinas, 31, 37, 41, 81
- Propuesta geométrica, 29
- Propuesta numérica, 29
- Regiones suaves, 6, 59, 61
- Reorganización de vecindades, 41, 47, 54, 57, 59, 61
- Tolerancia al ruido, 6, 64, 67, 71, 74, 81
- Forma alfa, 8, 91
- Geometría, 87
- Cara, 89, 90
 - Impropia, 89
 - Propia, 89
- Complejo simplicial, 8, 16, 17, 89–91, 93–95
- Diagrama de Voronoi, 8, 10, 11, 13, 19, 29, 88, 89
- Envolverte convexa, 11, 88, 89
- Lado Delaunay, 10, 88
- Región de Voronoi, 12, 87, 88
- Simplejo, 8, 89–91
- Triangulación, 8, 10, 89
- Triangulación de Delaunay, 7–11, 13, 14, 16, 88, 89, 91
- Geometría computacional, 7
- Grafo de Gabriel, 9
- Grafo Riemanniano, 34, 36
- Modelado geométrico, 1, 89
- Adquisición de datos, 1
 - Ajuste de la superficie, 2
 - Registro e integración, 1
- Modelo sintético, 57
- Ruido, 64
- Objetivos de la tesis, 6
- Planteamiento del problema, 4
- Polo, 11
- Propuesta, 6, 41, 55
- Reconstrucción de superficies, 2, 7–9, 41, 92
- Problema en bordes y esquinas, 13
- Representación digital, 1, 5
- Aplicaciones, 5
 - Ambientes virtuales, 5

- Diseño industrial, 5
- Ingeniería inversa, 5
- Desventajas, 5
- Ventajas, 5

- Sistema de resortes basado en ARC, 42,
50
 - Eliminar contribución al error, 44
 - Relajar la condición, 44
 - Término de penalización, 46
 - Término de pertenencia, 46, 55

- Topología, 2, 9, 16, 92–94
 - Conjuntos abiertos, 92–94
 - Espacio topológico, 92–95
 - Homeomorfismo, 12, 93–95
 - Subespacio topológico, 92
 - Triangulación, 93–95
 - Variedad, 12, 18, 94–96
 - Frontera, 95
 - Orientable, 96
 - Sin fronteras, 18

- Vecindad, 5, 6, 13, 25, 29, 30, 38, 39, 41,
51, 54, 59, 61, 94–96