



UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN

MEJORAMIENTO AUTOMÁTICO DE MALLAS COMPUESTAS DE UNIDADES DE
RESPUESTA HIDROLÓGICA (URHS) IMPLEMENTADO EN QGIS

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN COMPUTACIÓN

SERGIO ARIEL VILLARROEL GARCÉS

PROFESOR GUÍA:
PEDRO SANZANA CUEVAS

PROFESORA CO-GUÍA:
NANCY HITSCHFELD KAHLER

MIEMBROS DE LA COMISIÓN:
MARISA ERNST ELIZALDE
MATÍAS TORO IPINZA

SANTIAGO DE CHILE
2023

Resumen

En el contexto del modelamiento hidrológico, el modelo PUMMA (Peri-Urban Model for landscape MAnagment) representa áreas urbanas a través de EHU (Elementos Hidrológicos Urbanos) y zonas naturales y periurbanas a través de URHs (Unidades de Respuesta Hidrológica). Las URHs pueden ser representadas en un modelo vectorial, en donde estas son representadas por polígonos, los cuales deben cumplir ciertas condiciones para su correcta aplicación en el modelo hidrológico. Entre las condiciones geométricas que definen si un polígono es de buena calidad para su uso en el modelo hidrológico, están que sus centroides se encuentren dentro del polígono, que no sea un polígono largo y delgado, entre otras.

Durante este trabajo de memoria, se analizan las problemáticas y posibles soluciones para poder mejorar las URHs, entre las cuales destacan la generación de una triangulación y posterior disolución de triángulos para lograr poder subdividir las URHs en polígonos más pequeños y así obtener polígonos que puedan ser aplicados en el modelo hidrológico correctamente.

Se describe cómo se determina si un polígono cumple con cierta calidad para su aplicación en el modelo hidrológico, en base a criterios geométricos llamados descriptores de forma.

También se detalla el cómo el algoritmo de mejoramiento es implementado e integrado en el Sistema de Información Geográfico QGIS, describiendo de forma general el sistema geográfico, y también detallando el algoritmo implementado.

Para validar la solución propuesta, se aplican los algoritmos de mejoramiento sobre varios casos de prueba, los cuales fueron obtenidos desde una URH de un área periurbana de Lo Barnechea-La Dehesa. Se evalúa el efecto que tiene los distintos parámetros de triangulación y de disolución en distintos casos, ya sea modificando los ángulos mínimos en la triangulación, como también en cuanto al efecto de los descriptores de forma. Finalmente, se evalúan los tiempos de ejecución del algoritmo de disolución.

Este trabajo está dedicado para todas las personas que me apoyaron y me han tenido paciencia en este proyecto, en especial a mi familia, a la profesora Nancy y a Pedro quienes me han guiado en este proyecto.

Tabla de Contenido

1. Introducción	1
1.1. Motivación	2
1.2. Objetivos	3
1.2.1. Objetivo general	3
1.2.2. Objetivos específicos	3
1.3. Descripción general de la solución	3
1.4. Contenido de la Memoria	4
2. Antecedentes	5
2.1. Descriptores de forma	6
2.2. Implementación actual	6
2.2.1. Tecnologías usadas	6
2.2.2. Algoritmos actualmente implementados	7
3. Análisis	11
3.1. Problema	11
3.2. Análisis de la solución e implementación	12
3.2.1. Triangulación	14
3.2.2. Disolución	14
3.3. Métricas	15
3.4. Estructura de Datos QGIS	16

4. Diseño e Implementación	19
4.1. Implementación del Plugin	19
4.2. Implementación de la Triangulación	20
4.3. Implementación de la Disolución	21
4.3.1. Interfaz Gráfica y restricciones de entrada	24
5. Validación	26
5.1. Triangulación	29
5.2. Disolución	33
5.3. Evaluación de proceso completo	38
5.3.1. Efecto de la triangulación inicial	38
5.3.2. Aplicación sobre polígonos delgados y alargados	43
5.3.3. Índice de Solidez versus Índice de Convexidad	45
5.4. Evaluación de casos particulares	49
5.4.1. Aplicación repetida de algoritmo de disolución	49
5.4.2. Casos en el que el algoritmo de disolución no obtiene el resultado deseado	51
5.4.3. Diferencias en el orden de disolución	52
5.5. Tiempos de ejecución	54
6. Conclusiones	56
6.1. Trabajo a Futuro	57
Bibliografía	59

Índice de Tablas

5.1. URHs Casos de Estudio	29
5.2. URHs Casos de Estudio: Triangulados sin restricción	29
5.3. URHs Casos de Estudio: Triangulados con restricción de ángulo 5°	29
5.4. URHs Casos de Estudio: Triangulados con restricción de ángulo 25°	30
5.5. URHs Casos de Estudio: Triangulados con restricción de 1% de área máxima	30
5.6. URHs Casos de Estudio: Disolución con restricción de Factor de forma de 0,3	33
5.7. URHs Casos de Estudio: Disolución con restricción de Compacidad de 0,5 . .	33
5.8. URHs Casos de Estudio: Disolución con restricción de Índice de Solidez de 0,85	33
5.9. URHs Casos de Estudio: Disolución con restricción de Índice de Convexidad de 0,75	34
5.10. URHs Casos de Estudio: Disolución con restricción de 10% de área máxima	34
5.11. Disolución en base a triangulación ángulo mín. 20° y distintas restricciones de área	39
5.12. Disolución en base a triangulación distintos ángulos mínimos	39
5.13. Disolución en base a triangulación distintos ángulos mínimos con $FF = 0,3$.	40
5.14. Disolución de un polígono delgado y alargado triangulado en 25°	43
5.15. Disolución de un polígono unión de calles triangulado en 25°	43
5.16. Diferencia en disolución polígonos con agujero y centroide fuera	46
5.17. Disolución Centro Deportivo con índices de solidez y convexidad	47
5.18. Disolución URH Grande triangulado en área máxima $1.000\ m^2$	52
5.19. Tiempo de ejecución proceso de disolución [s]	54

Índice de Ilustraciones

2.1. Descomposición de un polígono con un polígono en su interior.	8
2.2. Descomposición de un polígono con alta variabilidad en la pendiente.	9
2.3. Descomposición de un polígono cuyo centroide puede encontrarse fuera de éste.	10
3.1. Ejemplo de descriptores de forma aplicados sobre URHs	13
3.2. Triangulaciones de una URH	15
3.3. URH con centroides fuera	16
3.4. Referencia de Clases de QgsMapLayer	17
3.5. Representación de un polígono en QGIS	18
4.1. Interfaz gráfica de ambos pasos de triangulación y disolución	24
5.1. URH sector Lo Barnechea-La Dehesa	27
5.2. URHs Casos de estudio	28
5.3. URHs Casos de estudio triangulados con ángulo 25°	31
5.4. URHs Casos de estudio triangulados con restricción de 1 % de area máxima	32
5.5. URHs Casos de estudio disueltos con restricción de Factor de Forma 0,30	35
5.6. URHs Casos de estudio disueltos con restricción de Índice de Convexidad 0,75	36
5.7. URHs Casos de estudio disueltos con restricción de 10 % de área máxima	37
5.8. Disolución en base a triangulación ángulo mín. 20° y distintas restricciones de área	41
5.9. Disolución en base a triangulaciones con distintos ángulos mínimos	42
5.10. Disolución Calle simple con distintos Factores de Forma	43

5.11. Disolución Varias Calles con distintos Factores de Forma	44
5.12. Disolución en polígonos con agujero y centroide fuera	46
5.13. Disolución Calle comparando índices de solidez y convexidad	47
5.14. Disolución Centro Deportivo para distintos índices de solidez y convexidad .	48
5.15. Disolución de polígono en varios pasos $IS = 0,8$	50
5.16. Disolución de polígono con $IS = 0,83$	51
5.17. Disolución de polígono con $FF = 0,5$	52
5.18. Disolución por área en distintos ordenes	53
5.19. Regresión lineal sobre tiempo de ejecución de la disolución en orden	55

Índice de algoritmos

1.	Triangle Features	21
2.	Build Neighbours	22
3.	Dissolve Features	23

Capítulo 1

Introducción

La aplicación de modelos hidrológicos permite cuantificar y evaluar los impactos hidrológicos que genera la introducción de elementos artificiales en zonas de estado natural, tales como calles, canales, zanjas y colectores de aguas lluvias entre otros (Gregory, 2006[1]; Gironás et al., 2007[2]). Dentro de este contexto, se ha desarrollado un modelo hidrológico distribuido llamado PUMMA (Peri-Urban Model for landscape MAnagment) (Jankowsky et al., 2010[3]; Jankowsky, 2011[4]), en el cual el área urbana se representa mediante el enfoque de EHU (Elementos Hidrológicos Urbanos) (Rodríguez et al., 2008[5]), mientras que las zonas periurbanas y naturales se representan mediante URHs (Unidades de Respuesta Hidrológica) (Flügel, 1995[6]). Diferentes trabajos de investigación han permitido probar hipótesis de segmentación en ambientes urbanos (Sanzana, 2019[7]) y cuantificar el efecto de segmentar distintas unidades en un modelo hidrológico (Sanzana, 2017[8]).

Las URHs son el resultado de la intersección entre distintos mapas, representados a través de capas vectoriales, tanto por polígonos como por polilíneas. Las capas representadas por polígonos corresponden a los mapas de uso y tipo de suelo, subcuencas y geología, mientras que las capas de polilíneas se asocian a información de ríos, canales artificiales y colectores de aguas lluvias. De esta manera, las URHs son polígonos simples de formas muy irregulares que buscan representar sectores de características homogéneas.

Para la correcta aplicación de las URHs en el modelo PUMMA, éste requiere que la malla hidrológica (compuesta de URHs) cumpla con una correcta conectividad y con ciertos criterios numéricos. Es por esto que es necesario un preprocesamiento de esta malla que permita corregir los posibles problemas numéricos y topológicos que aparecen al generar la malla hidrológica, y de esta forma permitir su uso en simulaciones hidrológicas.

Dentro de este contexto, Sanzana, 2011[9] propuso e implementó un preprocesamiento de la malla de entrada para que cumpla con los requerimientos del modelo, tales como pendiente homogénea, convexidad de los elementos y una distribución homogénea del área de los elementos. Más precisamente, los problemas resueltos fueron los siguientes:

- **Suavizado de bordes de los polígonos:** El suavizado de bordes es necesario dado que las mallas de entrada son el resultado de digitalizaciones de imágenes, y por lo tanto pueden poseer polígonos cuyos perímetros se encuentran sobrestimados, lo que

no es deseable en el modelo.

- **Descomposición de polígonos con otros polígonos en su interior:** La existencia de polígonos dentro de otros debe ser tratada, ya que puede distorsionar la dirección de los flujos en el modelo hidrológico.
- **Descomposición de polígonos con alta variabilidad en cierta propiedad física, por ejemplo la pendiente:** En la modelación hidrológica, es deseable incluir otras propiedades físicas no consideradas en el concepto de URHs. Dentro de estas propiedades, resulta interesante incluir la pendiente, ya que afecta múltiples procesos hidrológicos. Es por esto que es necesario descomponer las URHs en las cuales se presenta una alta heterogeneidad en cuanto a dicha propiedad en URHs más homogéneas.
- **Descomposición de polígonos cuyos centroides se encuentran fuera de éste:** Los centroides de cada polígono son usados para el cálculo de flujos entre URHs, por lo que es necesario para el modelo que los centroides de estos se encuentren en su interior para una correcta representación.
- **Descomposición de polígonos muy grandes:** Cuando polígonos muy grandes son conectados con polígonos pequeños se pueden provocar problemas numéricos al calcular los flujos en el modelo, por lo que es necesario descomponer dichos polígonos en algunos más pequeños.

1.1. Motivación

Si bien el trabajo realizado en Sanzana, 2011[9] dio resultados satisfactorios en la modelación hidrológica, éste se encuentra implementado en distintos scripts, los cuales interactúan con GRASS GIS ¹(Sistema de Información Geográfica) y Triangle (Software que genera triangulaciones). La implementación de estos distintos scripts no permitió abordar en forma integra todos los problemas geométricos que la malla inicial presentaba, por ese motivo resulta interesante automatizar y complementar el procesamiento de las mallas integrándolo en el software QGIS ² (Sistema de Información Geográfica más amplio que GRASS-GIS) como un *plugin*.

Por otro lado, resulta necesario buscar una generalización y optimización de los algoritmos presentados en Sanzana, 2011[9], ya que al realizar el pre-procesamiento de mallas en cuencas peri-urbanas, el tiempo de computación es considerable (alrededor de unos días). Junto con esta optimización de algoritmos, es necesario además revisar algunos casos particulares en los que los algoritmos fallan y otros que requieren intervención manual.

Adicionalmente, es necesario implementar nuevos algoritmos de descomposición de polígonos que busquen resolver algunos de los problemas que quedaron propuestos en Sanzana, 2011[9]. Particularmente, uno de los problemas tiene que ver con los polígonos delgados y alargados (por ejemplo los que buscan representar calles), ya que estos elementos distorsionan el modelo hidrológico (funcionando como barreras) y requieren algún tipo de descomposición en polígonos más pequeños que no generen dichos problemas.

¹<https://grass.osgeo.org/>

²<https://www.qgis.org/>

Finalmente, resulta interesante investigar e implementar cuantificadores que permitan evaluar la calidad de estas nuevas mallas en comparación con la malla original desde un punto de vista hidrológico.

1.2. Objetivos

1.2.1. Objetivo general

Automatizar y optimizar el proceso de mejoramiento de una malla hidrológica realizado en Sanzana, 2011[9], junto con la implementación de criterios para evaluar la calidad de las mallas generadas.

1.2.2. Objetivos específicos

- Automatizar el proceso de mejoramiento de la malla implementado en Sanzana, 2011[9], incorporándolo como *plugin* a QGIS.
- Buscar una optimización de los algoritmos implementados en Sanzana, 2011[9], junto con revisar casos particulares en los cuales el algoritmo implementado falla, como es el caso de los polígonos con otros polígonos en su interior.
- Implementar un método para corregir los problemas generados por polígonos delgados y alargados, ya sea a través de un proceso de descomposición basado en el índice de compacidad³ o mediante otros factores.
- Evaluar criterios que permitan comparar la calidad de las mallas generadas por los distintos métodos de descomposición, desde un punto de vista de los resultados obtenidos en la simulación hidrológica.

1.3. Descripción general de la solución

Para la resolución de los objetivos principales, el trabajo se enfocó en el estudio y desarrollo de un *plugin* en QGIS, el cual se enfoca en dos procesos principales:

- Integración de Triangle⁴ en QGIS y su aplicación bajo ciertos parámetros.
- Disolución de polígonos, ya sean productos del paso anterior de triangulación o de otro estado, en polígonos que cumplan ciertos parámetros.

³El Índice de Compacidad se define como $C = \sqrt{4\pi A}/P$, en donde A y P son el área y el perímetro del polígono respectivamente.

⁴<https://www.cs.cmu.edu/~triangle.html>

La aplicación consecutiva de ambos procesos de triangulación y su posterior disolución de polígonos busca lograr los objetivos planteados y lograr obtener URHs de la calidad requerida por el sistema hidrológico.

Finalmente, se buscan cuáles son los parámetros de triangulación y disolución que logran obtener polígonos de la calidad deseada. Para la determinación de la calidad de la solución, se evaluarán por criterios netamente geométricos, pero de relevancia en el contexto del modelo hidrológico.

1.4. Contenido de la Memoria

Este trabajo de memoria está compuesto de la siguiente manera: En el capítulo 2 se describen algunas propuestas de soluciones al problema, junto con la implementación propuesta por Sanzana, 2011[9]. En el capítulo 3 se describe el problema en mayor detalle, junto con la propuesta de la solución y sus distintos alcances. En el capítulo 4 se describe la solución propuesta en esta memoria, detallando los algoritmos más importantes y la interfaz gráfica principal. El capítulo 5 presenta la validación del algoritmo implementado, evaluando el algoritmo en URHs obtenidas de datos reales y que no tenían la calidad deseada por el sistema hidrológico. Finalmente, en el capítulo 6 se resume el trabajo realizado, se concluye sobre los diversos resultados y se describe el trabajo futuro.

Capítulo 2

Antecedentes

En el contexto de modelamiento hidrológico, existen distintas maneras de representar el terreno y sus características, tales como los modelos ráster y vectorial. Ambas representaciones poseen distintos algoritmos para el cálculo de flujos y cada una tiene sus ventajas y desventajas en su aplicación para distintos problemas. En el caso de zonas peri-urbanas, Sanzana, 2011[9] determinó que la mejor manera de resolver este problema es usando una representación vectorial la cual se realiza a través de la herramienta semi automática Geo-PUMMA.

La representación vectorial de una zona peri-urbana se hace por medio de una malla de polígonos, los cuales buscan representar zonas relevantes en el concepto hidrológico y que comparten ciertos parámetros de interés, tales como uso y tipo de suelo, etc. Sin embargo, esta malla suele contener polígonos que no son útiles para el modelo hidrológico, tales como polígonos con sus centroides fuera del polígono, o polígonos alargados, los cuales distorsionan totalmente los cálculos de flujos.

La descomposición de estos polígonos de mala calidad en polígonos que cumplan ciertas condiciones de forma, y que al mismo tiempo sean significativos, resulta completamente necesaria. Es posible descomponer el polígono a través de métodos de triangulación tales como Triangle (Shewchuk 1996)[10], sin embargo el resultado no es satisfactorio, tanto en la cantidad de polígonos (triángulos) resultantes, lo cual afecta el tiempo de computación del sistema hidrológico notablemente, como en la representabilidad de la malla resultante, la cual pierde información relevante en cuanto a la forma de la malla inicial.

Para la resolución del problema de división de polígonos 2D en formas significativas, existen distintos métodos, tales como DuDE (Liu et al. 2014)[11] y ACD (Lien & Amato 2005)[12]. DuDE (Dual-space Decomposition) y ACD (Approximate Convex Decomposition) son algoritmos eficientes que buscan descomponer el polígono original, segmentándolo en partes aproximadamente convexas, con lo cual se logra el objetivo de obtener polígonos con sus centroides en su interior. Sin embargo, esto solamente resuelve uno de los problemas, por lo que resulta necesario buscar e implementar una descomposición en polígonos distinta, pero que logre todos o la mayor parte de los objetivos planteados anteriormente.

2.1. Descriptores de forma

El concepto de forma no es un fenómeno fácil de definir, ya que no todos percibimos las formas de la misma manera. Describir un objeto a través de unas formas es difícil poner en palabras, y es mucho más difícil describir una forma a un tercero y que tengamos la misma imagen o interpretación del objeto. Muchas veces, uno puede definir las formas en base a otro tipo de formas, por ejemplo, para describir un furgón uno lo describiría con forma más parecida a una caja que a un auto convencional.

Dentro de este contexto, Russ, 2002[13], destaca la importancia de definir criterios numéricos que capturen distintas características de un objeto, a los cuales llama descriptores de forma. Los descriptores de forma son variados y han existido en la literatura incluso con distintos nombres. Dentro de estos, los más interesantes son el Factor de Forma $FF = (4\pi A)/P^2$, Índice de Compacidad $C = \sqrt{4\pi A}/P$, Índice de Solidez $SI = A/A_{Convex}$ e Índice de Convexidad $CI = P_{Convex}/P$, en donde P y A representan el perímetro y el área del polígono, y P_{Convex} y A_{Convex} , representan el perímetro y el área de la cerradura convexa del polígono.

2.2. Implementación actual

2.2.1. Tecnologías usadas

Un Sistema de Información Geográfica (SIG) es un sistema computacional diseñado para capturar, almacenar, manipular, analizar y desplegar en todas sus formas la información geográficamente referenciada con el fin de resolver problemas complejos de planificación y gestión geográfica. Su uso en los modelos hidrológicos se ha convertido en una necesidad debido al gran volumen de información geográfica que debe ser manejada. El uso de un SIG además permite representar distintos elementos geográficos, ya sea usando un enfoque de modelación ráster o uno de modelación vectorial, siendo este último el de interés para el presente trabajo.

GRASS GIS (*Geographic Resources Analysis Support System*)¹ es un SIG de uso libre y de código abierto que otorga las funcionalidades básicas de un SIG, pero con funcionalidades topológicas necesarias para el modelo PUMMA. Por otro lado, QGIS² es otro SIG de uso libre y código abierto que nos provee de una GUI (*Graphical User Interface*) para GRASS GIS, junto con la posibilidad de integrar con otros sistemas SIG o bases de datos y también de incluir *plugins* creados por usuarios.

Dentro de las características de estos SIG está la posibilidad de representar los datos espaciales de dos maneras:

- Representación Vectorial: cada objeto de interés puede ser representado por puntos, líneas o polígonos.

¹<http://grass.osgeo.org/>

²<http://www.qgis.org/es/site/>

- Representación Ráster: se hace una representación mediante píxeles, en donde cada uno de estos posee la característica del suelo u objeto que busca representar.

Para la representación de la URHs se usa exclusivamente la representación vectorial. Sin embargo, el sistema permite realizar la transformación entre representación vectorial y ráster, y viceversa, en caso de ser necesario. Durante este trabajo sólo será de nuestro interés la representación vectorial.

La implementación actual realizada por Sanzana, 2011[9] se encuentra implementada principalmente en *Python*³, debido a que mediante el uso de librerías fue posible usar rutinas presentes en GRASS-GIS. También fue necesario el uso de Triangle⁴ para poder realizar triangulaciones de polígonos cuando fuese necesario. Finalmente, se usó un script en R⁵ para transformar los polígonos desde el formato usado en GRASS-GIS, al formato requerido en Triangle.

2.2.2. Algoritmos actualmente implementados

A continuación se describe la forma en la que Sanzana, 2011[9] resolvió los problemas a través de varios scripts:

2.2.2.1. Suavizado de bordes de los polígonos

Para la implementación de un método de suavizado de bordes se estudiaron los algoritmos de Douglas-Peucker (Douglas and Peucker, 1973)[14] y de Snakes (Kass et al., 1988)[15], de los cuales se escogió este último, ya que se obtenía una representación más acorde a la realidad, a diferencia del algoritmo de Douglas-Peucker que, a pesar de que efectivamente reducía el perímetro de los polígonos, éste tendía a sobre simplificar los bordes a líneas rectas, con lo cual se perdía información relevante de la forma del polígono. GRASS-GIS permitió el fácil uso del algoritmo de Snakes gracias a que ya se encuentra implementada mediante el comando *v.generalize* con la opción de Snakes.

2.2.2.2. Descomposición de polígonos con otros polígonos en su interior

En el caso de los polígonos con otros polígonos en su interior, se implementó un algoritmo simple el cual busca dividir el polígono exterior, de tal manera que el polígono interior se elimine. El algoritmo implementado calcula las distancias de los vértices del polígono interior con respecto al polígono exterior, luego escoge la distancia más pequeña y la más grande, y finalmente utiliza estas distancias como líneas de borde de los nuevos polígonos.

³<https://www.python.org/>

⁴<http://www.cs.cmu.edu/~quake/triangle.html>

⁵<http://www.r-project.org/>

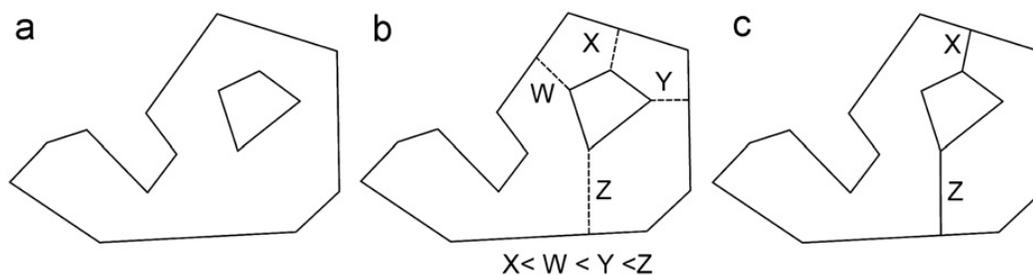


Figura 2.1: Descomposición de un polígono con un polígono en su interior.

2.2.2.3. Descomposición de polígonos con alta variabilidad en cierta propiedad física, por ejemplo la pendiente

Como se explicó anteriormente, es necesario mantener cierta homogeneidad dentro de cada URH. Para determinar cuáles polígonos deben ser tratados, el usuario debe especificar un valor límite para la desviación estándar de una cierta propiedad física de la URH. Cuando se detecta que cierto polígono presenta una mayor desviación estándar en dicha propiedad que el máximo permitido, éste es descompuesto de la siguiente manera:

- Se identifica dentro del polígono el valor de los cuartiles Q_1 y Q_3 (primer y tercer cuartiles respectivamente).
- Se crean líneas de contorno que separen las partes del polígono que pertenezcan a distintos cuartiles (ver Figura 2.2b).
- Se disuelven las áreas más pequeñas conservando sólo las tres áreas más grandes (ver Figura 2.2c).
- Se suavizan y luego se simplifican los contornos generados, usando los algoritmos de Snake y de Douglas-Peucker respectivamente (ver Figura 2.2, pasos d y e).
- Se disuelve finalmente el polígono usando los contornos generados. (ver Figura 2.2f).

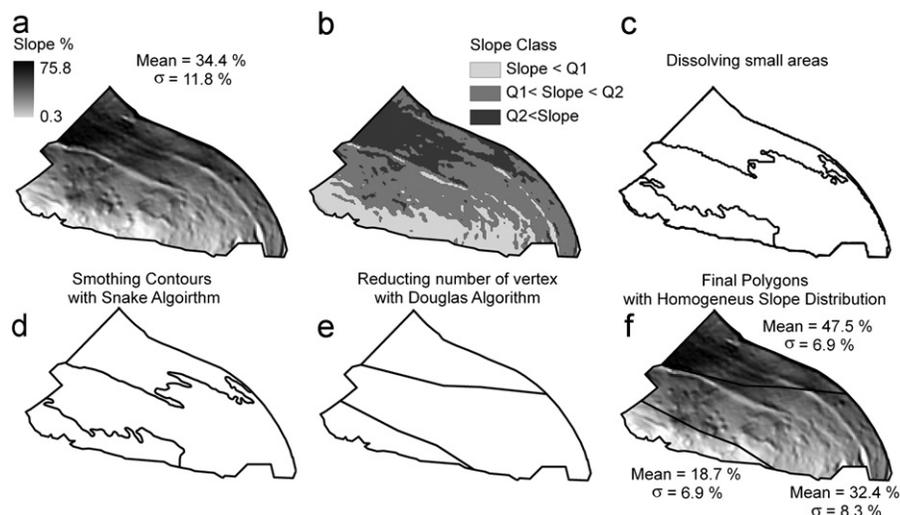


Figura 2.2: Descomposición de un polígono con alta variabilidad en la pendiente.

2.2.2.4. Descomposición de polígonos cuyos centroides se encuentran fuera de éste

Para la resolución de este problema, se propuso eliminar estos polígonos por medio de una descomposición de estos, en polígonos más pequeños, pero que cumplan ciertos parámetros geométricos que nos determine que su centroide esté adentro del polígono. Para determinar esto, se estudió el uso de diferentes descriptores de formas propuestos por Russ (2002), de los cuales se usó el Índice de Convexidad, descrito como $IC = P_{Convex}/P$, en donde P es el perímetro del polígono y P_{Convex} es el perímetro de la cerradura convexa de los vértices del polígono. Empíricamente, se determinó un valor límite para el Índice de Convexidad igual a 0.75, por lo que cualquier polígono cuyo Índice de Convexidad sea menor a este debe ser reparado.

Para descomponer estos polígonos, se propuso generar una triangulación de Delaunay (realizada mediante Triangle) sobre estos. El problema de usar solo la triangulación, es que dan como resultado una gran cantidad de polígonos que, además de aumentar el tiempo de computación, no son una representación deseable de una URH. Es por esto que luego del proceso de triangulación, se reduce el número de polígonos mediante un método de disolución. Este proceso de disolución toma un triángulo y lo combina con su vecino de mayor área y así sucesivamente, hasta que no se pueda disolver sin exceder el índice de convexidad límite fijado anteriormente. Este procedimiento de disolución es aplicado sobre todos los triángulos que no han sido ya disueltos, con lo que, como resultado final, se obtienen polígonos que son más deseables para el modelo hidrológico.

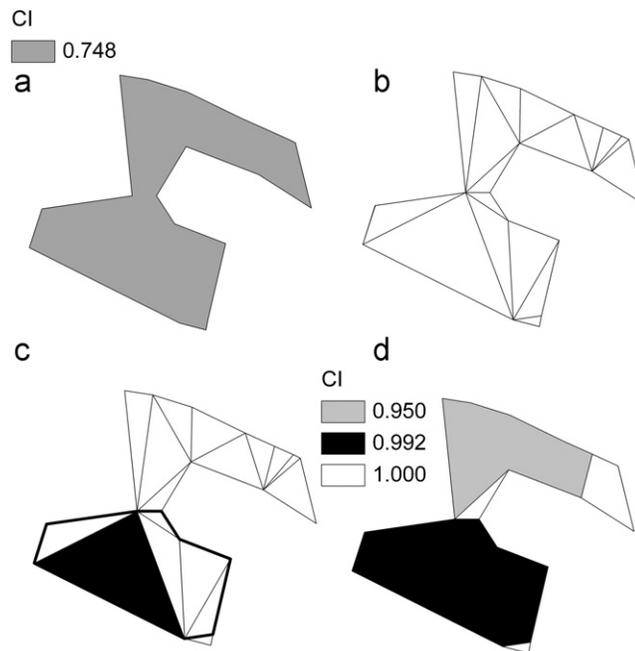


Figura 2.3: Descomposición de un polígono cuyo centroide puede encontrarse fuera de éste.

2.2.2.5. Descomposición de polígonos por área máxima

La descomposición de estos polígonos, cuyas áreas exceden cierto valor máximo definido por el usuario, se realiza de manera similar a la descomposición de polígonos con centroides fuera de estos, con la restricción adicional de área máxima al realizar la disolución de triángulos.

Capítulo 3

Análisis

En este capítulo se describen los problemas a resolver, analizando las soluciones y complejidades que cada uno de estos presenta. Además, se describen algunas métricas para poder comparar la calidad de los resultados.

Finalmente, se muestra de manera general las estructuras de datos usadas en QGIS, así como una breve descripción sobre el cómo se representan los polígonos en el sistema geográfico.

3.1. Problema

La implementación actual de mejoramiento de polígonos en Sanzana, 2011[9] busca abordar 5 distintas problemáticas en cuanto a la forma de los polígonos:

- Suavizado de bordes de los polígonos.
- Descomposición de polígonos con otros polígonos en su interior.
- Descomposición de polígonos con alta variabilidad en cierta propiedad física.
- Descomposición de polígonos cuyos centroides se encuentran fuera de éste.
- Descomposición de polígonos por área máxima.

Con respecto al proceso de suavizado de bordes de polígonos, éste está actualmente implementado en GRASS-GIS a través de un comando *v.generalize*, el cual toma una polilínea y permite aplicar distintos algoritmos de simplificación, tales como Douglas-Peucker y Snakes. Si bien este preprocesamiento de polígonos es sumamente necesario para el modelo hidrológico, se determinó no incluir este proceso en un nuevo *plugin* en QGIS, ya que este comando ya se encuentra implementado en QGIS en el menú Vector, opción Generalizer, el cual posee todas las funcionalidades del comando *v.generalize* de GRASS-GIS.

Con respecto a la descomposición de polígonos con alta variabilidad en cierta propiedad física, se decidió por no incorporar como un *plugin* en QGIS, ya que el proceso es más complejo e involucra no solo la representación vectorial de las URHs, sino también la representación ráster para obtener las propiedades físicas. Dada su complejidad para su incorporación al *plugin*, se deja pendiente y se deja propuesto como trabajo a futuro.

Finalmente, para abordar los demás casos, vale decir la descomposición de polígonos con polígonos en su interior, polígonos con centroides fuera y polígonos de área muy grande, se decidió por resolverlos mediante un proceso de triangulación y posterior disolución de acuerdo a ciertos parámetros, en donde la correcta elección de los parámetros determinará si se solucionan estas problemáticas.

3.2. Análisis de la solución e implementación

Para la implementación del *plugin* en QGIS se decidió realizar en dos pasos separados: triangulación y disolución, cada uno con sus propios parámetros ajustables. Se optó por esta separación para dar una mayor flexibilidad al usuario y también para poder ir comparando el efecto que tienen los parámetros en cada paso.

El principal parámetro geométrico que se decidió usar es el descriptor de forma de los polígonos (Russ (2002)[13]). Estos corresponden a medidas que solamente consideran la forma geométrica de los polígonos, dentro de los cuales se implementaron los siguientes:

- Factor de Forma basado en un círculo : $FF = (4\pi A)/P^2$
- Compacidad basada en un círculo: $C = \sqrt{4\pi A}/P$
- Índice de Convexidad: $CI = P_{Convex}/P$
- Índice de Solidez: $SI = A/A_{Convex}$

Se escogieron estos descriptores de forma, ya que son los que más ayudan a detectar los polígonos que pueden tener una forma no deseada.

Por ejemplo, el Factor de Forma y Compacidad, cuyos valores son valores positivos entre 0 y 1, siendo 1 para cuando el polígono es un círculo perfecto. Valores muy menores a 1 significan que el polígono está completamente alejado de la forma de un círculo y que es posible que sea un polígono que se deba tratar. La diferencia más notable entre ambos índices es la sensibilidad con respecto al perímetro del polígono, en donde es el Factor de Forma el que se va más afectado y da valores más bajos con respecto a la Compacidad a medida que el perímetro dista más a la de un círculo.

Por otro lado, los Índices de Solidez y de Convexidad buscan comparar las áreas y perímetros con respecto a los de su cerradura convexa, respectivamente. Estos índices son valores positivos que van desde 0 a 1, siendo un valor de 1 para cuando el polígono es totalmente convexo. Si bien ambos índices logran el objetivo de cuantificar la convexidad del

polígono de cierta manera, resulta interesante evaluar ambos índices. Por ejemplo, el Índice de Convexidad, el cual se basa puramente en los perímetros, puede arrojar valores notablemente más bajos que el Índice de Solidez cuando se trata de polígonos con contornos irregulares que aumentan notablemente el perímetro del polígono, pero no así el área (ver Figura 3.1c).

Finalmente, si bien se evaluó al principio del trabajo hacer que los procesos de triangulación y disolución sean efectuados en la misma ejecución, se decidió por separar los procesos para lograr una mayor flexibilidad en cuanto a la disolución de polígonos, ya que es deseable poder diluir los polígonos no solamente desde la triangulación inicial, sino también desde cualquier tipo de disolución anterior y así poder ir refinando el resultado, ya sea mediante los mismos parámetros de disolución, o mediante la elección de otros parámetros (diferente descriptor de forma o ir relajando las restricciones, de acuerdo a lo que se requiera).

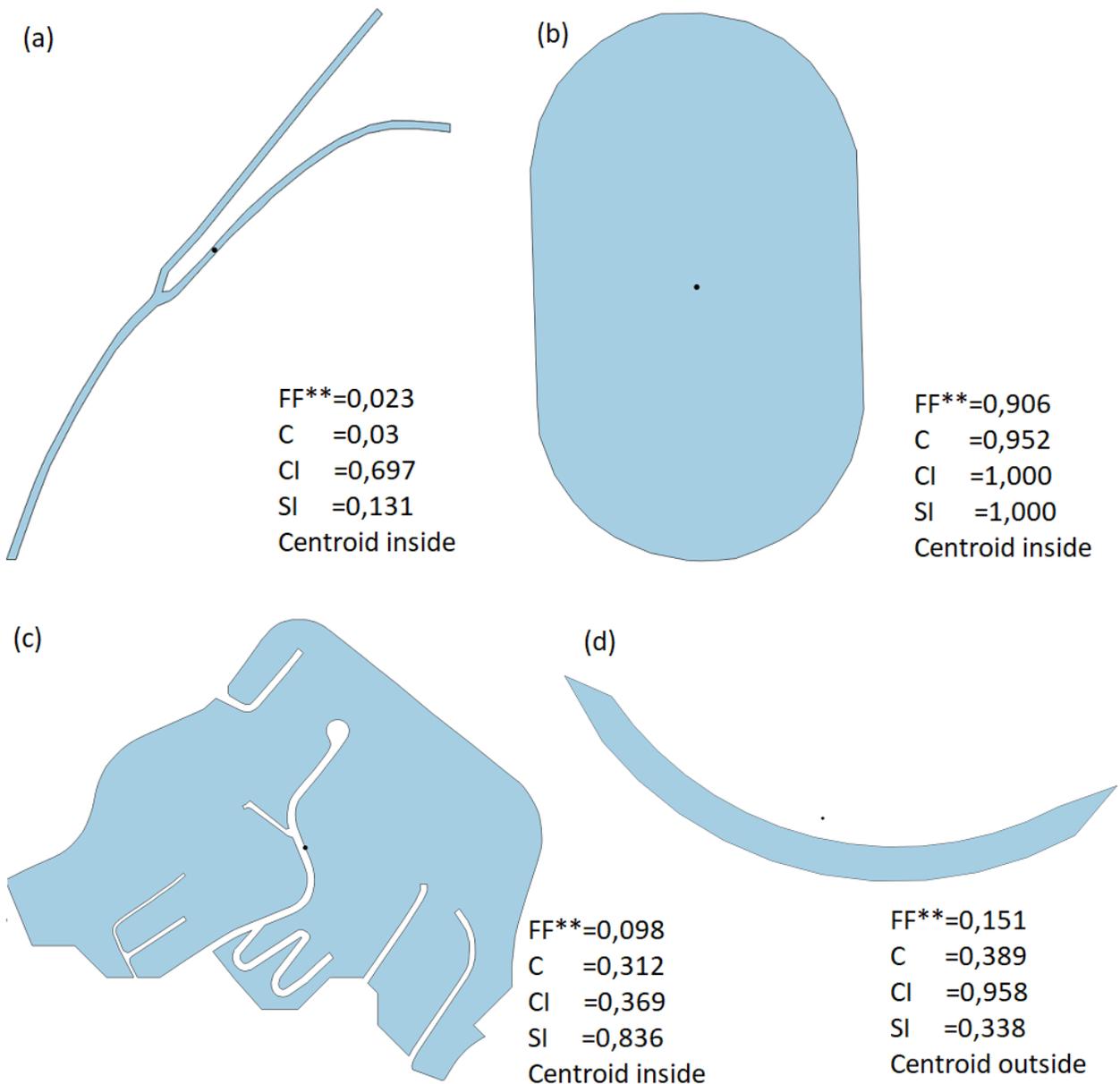


Figura 3.1: Ejemplo de descriptores de forma aplicados sobre URHs

3.2.1. Triangulación

Para el proceso de triangulación, el sistema QGIS solo presenta como solución una Triangulación de Delaunay, la cual no cumple con nuestro primer objetivo ya que no mantiene la forma del polígono original. Para lograr la triangulación correspondiente, y que respete la forma del polígono inicial, incluyendo los segmentos exteriores e interiores (agujeros), se optó por buscar una librería en *Python* que implemente la solución propuesta en Triangle¹.

Para la triangulación, se determinó que ciertos parámetros como ángulo mínimo y área máxima de las triangulaciones, eran necesarios para dar control sobre la calidad de las triangulaciones generadas, ya que el no incluirlas solo genera triangulaciones entre los puntos originales del polígono (triangulación de Delaunay restringida) y generaría triángulos de dimensiones muy irregulares entre los sectores con menor densidad de vértices, versus los sectores con mayores cantidad de vértices, tales como curvas muy detalladas. Por ejemplo, en la figura 3.2c, en donde se hace una triangulación sin ninguna restricción de calidad, es posible ver que los sectores centrales, donde hay una curva con alta densidad de vértices, hay una mayor cantidad de triángulos que en los extremos, y además son de un tamaño muy menor.

Se decidió además que resultaría interesante en este paso dar la opción de filtrar y triangular, o solamente filtrar, la malla de polígonos de entrada dependiendo si cumplen o no los factores de forma determinados anteriormente.

3.2.2. Disolución

Para el proceso de disolución, no existe en QGIS una solución que satisfaga ninguno de los objetivos planteados inicialmente. La disolución que ya está implementada en QGIS permite disolver un polígono totalmente o incluir una restricción en algún campo, por ejemplo diluir URHs que poseen el mismo tipo de suelo, pero ninguna restricción con respecto a formas geométricas, lo cual es lo que se requiere.

Los parámetros a considerar en este algoritmo son el área máxima de los polígonos disueltos y el descriptor de forma que debe cumplir. De esta forma, se espera que los polígonos obtenidos en el proceso de disolución cumplan con las restricciones geométricas propuestas, y que esto mismo signifique que estos polígonos cumplan con los objetivos planteados para su uso en el modelo hidrológico.

En cuanto al algoritmo para la disolución, se propone recorrer todos los componentes del polígono original triangulado e ir uniendo cada polígono con uno o más polígonos adyacentes, siempre y cuando se cumplan tantos los parámetros de forma, como los de área. Este proceso se repite para cada polígono hasta llegar a un estado en que no se pueda hacer uniones de polígonos sin incumplir los parámetros geométricos establecidos.

¹<https://www.cs.cmu.edu/~triangle.html>

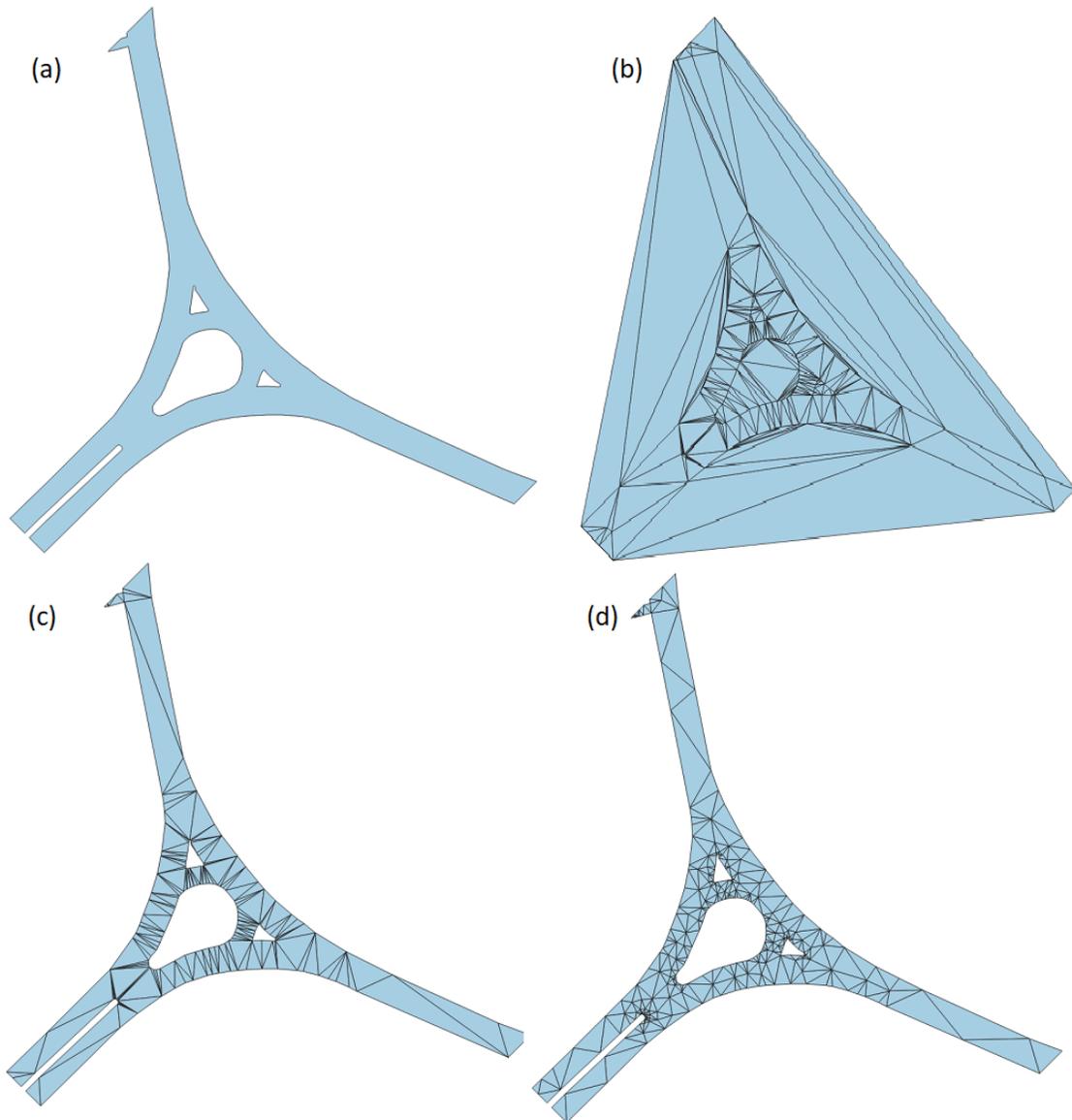


Figura 3.2: Triangulaciones de una URH: (a)URH sin triangulación (b) Triangulación de Delaunay (c) Triangulación sin criterios de calidad (d) Triangulación con ángulo mínimo 20°

3.3. Métricas

En cuanto al tema de métricas y poder corroborar si los algoritmos están cumpliendo o no lo requerido, estos se centrarán en las siguientes características:

- Los polígonos resultantes no deben tener agujeros en su interior.
- El centroide de los polígonos se debe encontrar dentro del polígono, ya que es fundamental para el modelamiento hidrológico.
- Los polígonos deben cumplir con el factor de forma requerido.

Estas métricas, puramente geométricas, dependen no tan solo de la implementación del

algoritmo, sino que también de los parámetros entregados en la ejecución. Como modo de ejemplo, los índices de solidez y de convexidad son los que buscan medir la convexidad del polígono, y mientras sea convexo el centroide se encontrará dentro del polígono. Sin embargo, cuando se establecen estos índices a valores menores a 1, pero cercanos, existe la posibilidad de que el centroide se encuentre fuera del polígono, tal como se puede ver en la Figura 3.3.

Por último, otra métrica básica a considerar va a ser el número de polígonos resultantes luego de la disolución. Si bien esta es una métrica bastante básica, es importante tener en cuenta ya que estos polígonos deben pasar por el modelo hidrológico, en donde tener una enorme cantidad de polígonos no tan solo sobrecarga el modelo, sino también se estaría perdiendo información sobre la forma del polígono original, el cual representa una zona física de ciertas características.

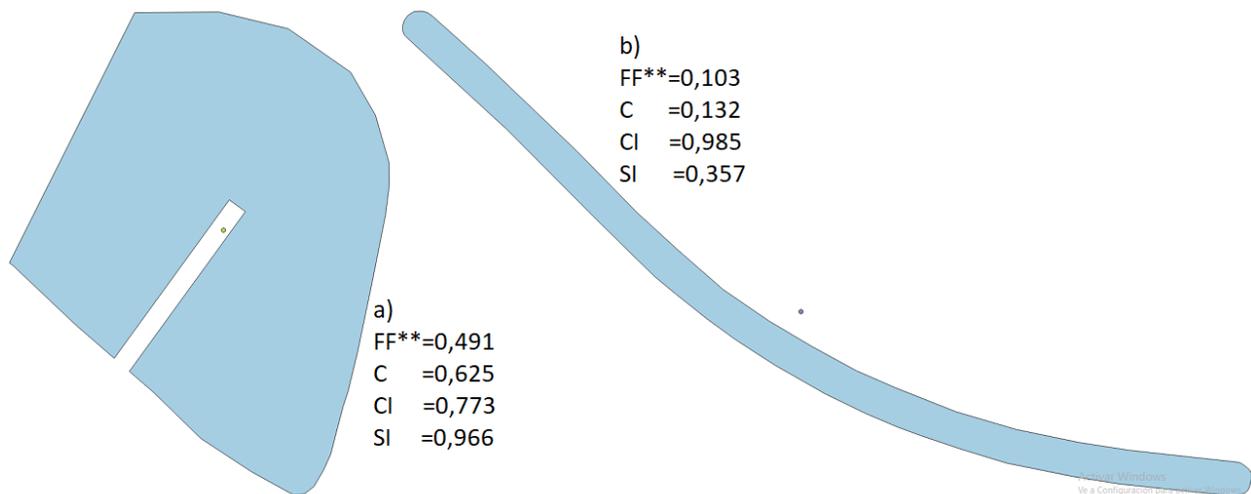


Figura 3.3: URH con centroides fuera: (a)URH cuyo índice de solidez es cercano a 1 (b) URH cuyo índice de convexidad es cercano a 1

3.4. Estructura de Datos QGIS

Para la representación espacial de los objetos, como se mencionó con anterioridad, existen 2 principales tipos de representación : Vectorial y Ráster, dentro de las cuales solamente se verá la representación Vectorial. Internamente, esto lo representa a través de la clases *QgsMapLayer*, la cual puede ser de ambos tipos: *QgsVectorLayer* o *QgsRasterLayer*, e incluso otros tipos de representaciones (Figura 3.4).

Solo nos centraremos en la clase *QgsVectorLayer*, la cual es la representación poligonal que necesitamos. Para representar cada punto, línea o polígono, QGIS usa la clase *QgsFeature*. Como en este trabajo se manejará principalmente URHs, las cuales son representadas por polígonos, todas las *Feature* tendrán este tipo, por lo que desde acá en adelante se usará el concepto de *Feature* y polígono indistintamente.

Dentro de la clases *QgsFeature*, lo más relevante que encontraremos son su id, su geometría (*QgsGeometry*) y sus atributos (*QgsField*).

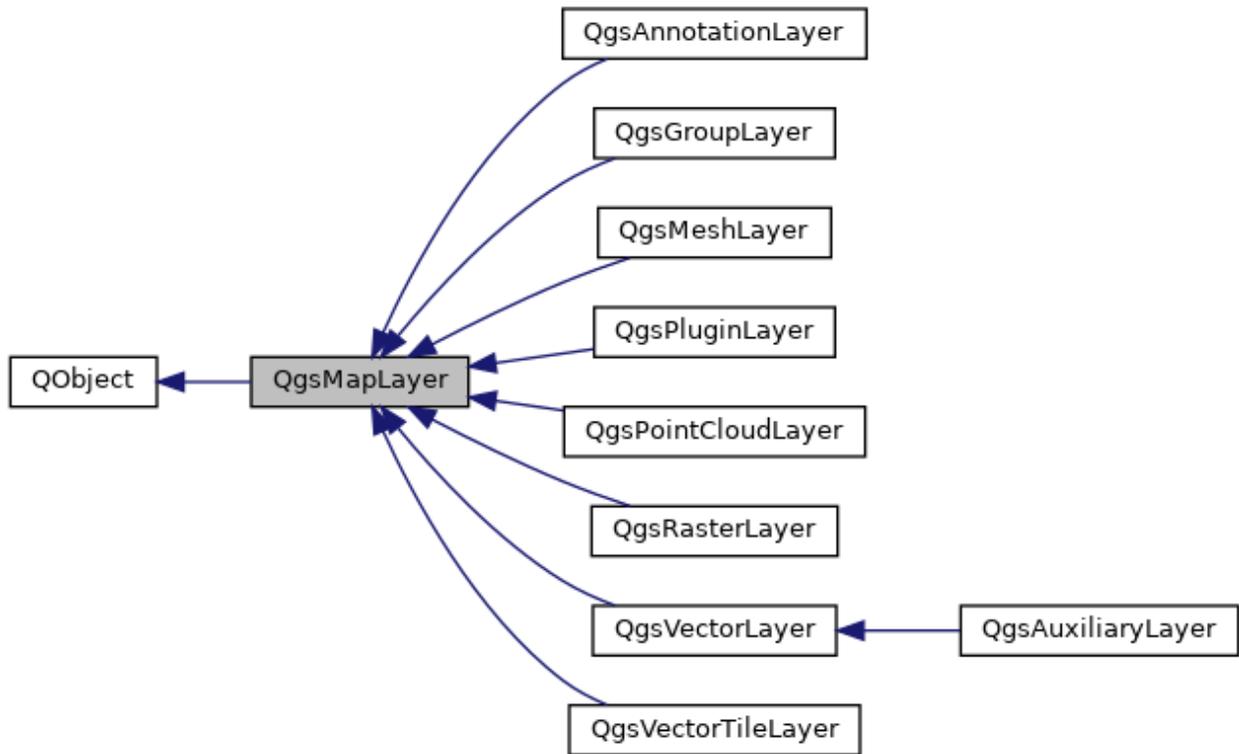


Figura 3.4: Referencia de Clases de QgsMapLayer

Finalmente, en la clase *QgsGeometry* es donde se encuentran la geometría del objeto y en donde QGIS permite realizar operaciones geométricas, tales como obtener el área, perímetro, calcular cerraduras convexas, computar uniones e intersecciones entre geometrías, etcétera. Vale la pena mencionar que es en esta clase en donde se debe verificar el tipo de geometría del objeto, el cual en nuestro caso debiese ser un polígono para cada URH.

Dentro de *QgsGeometry*, los polígonos pueden ser representados como un Polígono o un Multipolígono. En ambos casos, la representación es la misma, siendo la única diferencia el cómo fueron creadas (multipolígono permite representar más de un polígono, totalmente desconectados uno de otro). Ambos se representan como una lista de *boundaries* o perímetros. En el caso de un Polígono, la representación es la siguiente:

- La primera lista de puntos representa el perímetro exterior del polígono. Se representa por un listado de puntos, listados en sentido horario, en donde el primer punto debe ser repetido al final del listado para cerrar el perímetro.
- Las demás listas de puntos representan los perímetros interiores o agujeros del polígono. El listado de puntos es igual que con el perímetro interior, con la diferencia en el orden que debe ser en sentido antihorario.

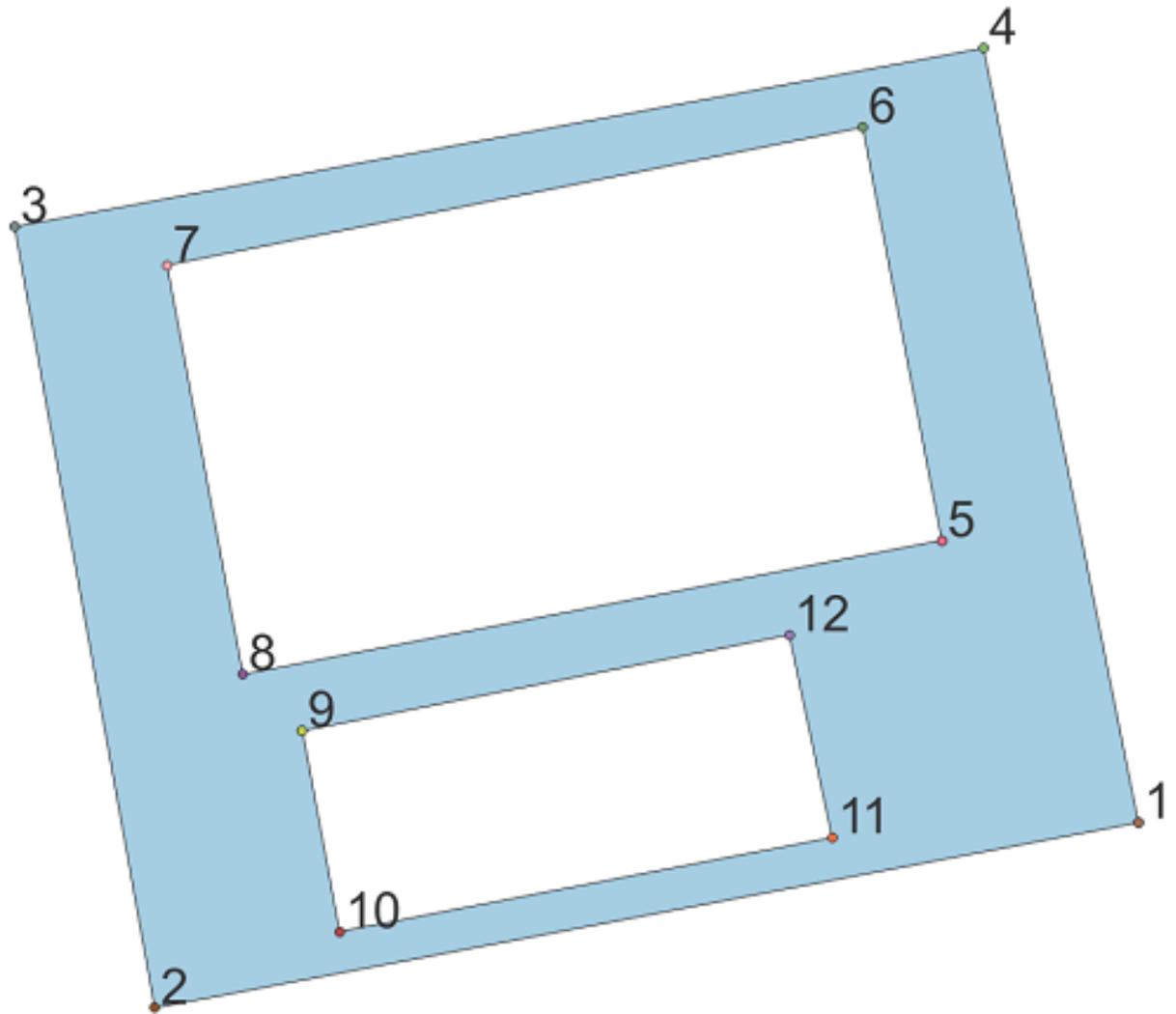


Figura 3.5: Representación de un polígono en QGIS: Perímetro exterior representado por puntos 1,2,3,4 (sentido horario); perímetros interiores por puntos 5,6,7,8 y 9,10,11,12 (ambos sentido antihorario)

Capítulo 4

Diseño e Implementación

En este capítulo se abordarán de manera concreta la implementación de la solución. Antes que todo, se detalla cómo se construye un plugin en QGIS y se da la estructura general que se implementará. Luego, se detalla la implementación de las etapas de triangulación y disolución, describiendo tanto el paso a paso de los algoritmos, como en forma de un pseudo código. Finalmente, se describe la interfaz gráfica implementada y los distintos parámetros que permite.

4.1. Implementación del Plugin

Para la implementación de plugins en QGIS existen diversas herramientas dentro del mismo software que ayudan en esta labor. Dentro de ellas, la más notable es el plugin “Plugin Builder”, el cual crea de manera automática los archivos básicos necesarios para la implementación de un plugin.

De esta manera, la estructura básica de la implementación actual del plugin es la siguiente:

- `__init__.py`:
Es el punto inicial del plugin. Su función es exclusivamente inicializar y retornar el plugin principal.
- `metadata.txt`:
Contiene información sobre el plugin, tales como el nombre del plugin, versión de QGIS requerida y autor, entre otros metadatos.
- `resources.qrc`:
Este archivo contiene la referencia a los archivos externos del plugin, tales como imágenes u otros tipos de archivos. Debe ser compilado mediante *pyrcc4* para producir el archivo `resources.py` y así facilitar el uso de los archivos externos dentro del plugin.
- `shapeDescriptors.py`:
Implementación y cálculo de los distintos descriptores de forma a utilizar.

- `trianglePlugin.py`:
Esta clase es la clase principal del plugin en la cual se implementan la inicialización y la funcionalidad del plugin.
- `trianglePlugin_dialog.py`:
Esta es la clase que está encargada de la correcta implementación de la interfaz gráfica. Si bien su única función es inicializar la interfaz gráfica ya determinada en otro archivo, es en esta clase en la que se debe implementar cualquier inicialización dinámica y que no esté en la interfaz gráfica ya implementada.
- `trianglePlugin_dialog_base.ui`
Este es el archivo en el cual se implementa la interfaz gráfica base.

Dentro de esta estructura, es en `trianglePlugin.py` en donde están implementadas tanto la triangulación, como la disolución.

4.2. Implementación de la Triangulación

El proceso de triangulación no presentó mayores desafíos una vez entendido la estructura de datos usada por QGIS para representar sus elementos geométricos.

La triangulación se hizo a través de la librería en Python llamada “Triangle”¹, la cual funciona como *wrapper* del algoritmo implementado por J. Shewchuk “Triangle”².

Para el uso de la librería Triangle se requieren entregar los datos de la misma forma que lo requiere la implementación de J. Shewchuk:

- **Vértices:** Se debe entregar un *array* que contenga todos los vértices del polígono a triangular.
- **Segmentos:** Se debe entregar un *array* que contenga los segmentos del polígono original, los cuales la triangulación respetará. Cada segmento está definido como una dupla de vértices, los cuales son identificados por su índice en el *array* Vértices.
- **Agujeros (Holes):** Cuando el polígono original posee un agujero o un polígono al interior, se debe entregar esta información para que la triangulación mantenga este agujero como tal. Lo que se pide en esto, es que para cada agujero se debe entregar un punto que está al interior de cada uno de ellos.
- **Opciones:** Se pueden entregar opciones adicionales de triangulación, las cuales son principalmente opciones que tienen que ver con la calidad de la triangulación entregada. Dentro de las opciones que nos interesa está la opción ‘p’, la cual obliga a que la triangulación respete los segmentos; la opción ‘q’ la cual permite imponer un ángulo mínimo para cada triángulo; y la opción ‘a’ la cual nos permite imponer un área máxima para cada triángulo.

¹<https://rufat.be/triangle/index.html>

²<https://www.cs.cmu.edu/~quake/triangle.html>

Por lo tanto, la implementación de la triangulación consistió en básicamente los siguientes pasos:

- Obtener los datos desde la interfaz del plugin, tales como las *Features* (URHs) y los parámetros para la ejecución.
- Extraer la información geométrica de las *Features*, la cual consiste en los vértices del polígono.
- Comprobar si la geometría cumple con los parámetros de descriptor de forma. Si los cumple, entonces en ese caso no corresponde triangular, por lo que se continúa con la siguiente *Feature*.
- Construir los segmentos desde los vértices, para su posterior ingreso en Triangle, ya que la geometría obtenida desde QGIS no contiene estos datos.
- Para cada agujero interior, buscar un punto dentro de éste, y guardar para su uso en Triangle.
- Triangular la geometría con los datos obtenidos, más los parámetros iniciales, tales como área máxima y ángulo mínimo.
- Repetir este proceso para cada *Feature* que no cumpla con el *Shape Descriptor* que se fijó.
- Obtener el resultado final y crear la geometría a retornar a la interfaz gráfica QGIS

Algoritmo 1: Triangle Features

Data: Features

Result: Triangled features

$featuresToAdd \leftarrow []$;

for *Feature* in *Features* **do**

$geometry \leftarrow Feature.getGeometry()$;

if *geometry* is well shaped **then**

 | continue ;

$polyVertices \leftarrow$ Points from *geometry*;

$polySegments \leftarrow$ Build segments from *geometry*;

$polyHoles \leftarrow$ Build holes from *geometry*;

$triangledFeature \leftarrow$ Call Triangle (*polyVertices*, *polySegments*, *polyHoles*);

 Add *triangledFeature* to *featuresToAdd*;

Return and Draw *featuresToAdd*

4.3. Implementación de la Disolución

A diferencia del proceso de triangulación, la disolución de polígonos es algo más complicada, ya que requiere información adicional sobre cada polígono, tales como los vecinos de cada polígono, cálculos de área y perímetro, y determinación del valor de su descriptor de forma.

El primer paso para este proceso es recorrer todos los polígonos y para cada uno de ellos, detectar y determinar cuáles son los vecinos. Para determinar los polígonos vecinos, estos deben necesariamente compartir un segmento, en otras palabras, compartir al menos en 2 vértices contiguos.

Es importante destacar que Triangle, y su implementación en Python, permiten obtener un listado de vecinos a través de la opción 'n'. Sin embargo, al momento de la implementación resultó interesante implementar el algoritmo de disolución de una manera general que no requiera que su entrada sea solamente obtenida desde Triangle. Es por esto que fue necesario implementar un algoritmo de vecinos general, ya que la entrada no necesariamente puede tener sus vecinos definidos, y es más, no necesariamente pueden ser triángulos.

Se tomó como decisión de diseño el mantener como estructuras de datos para la disolución un diccionario de *Features* y uno de vecinos, en donde el diccionario de *Features* tiene como *key* el id del *Feature*, y el diccionario de vecinos contiene como *key* el id del *Feature*, y entrega como información el id de los vecinos de dicho *Feature*.

Para determinar los vecinos de cada *Feature*, se optó por el siguiente algoritmo:

- Para cada *Feature*, construir su *Bounding Box*. Vale decir, para cada *Feature* se construye el cuadrado más pequeño que contenga completamente la geometría.
- Luego se hace una búsqueda de todos los *Features* que intersecten dicha *Bounding Box*, los cuales serán los candidatos a vecinos.
- Para cada candidato de vecino se verifica si efectivamente son polígonos vecinos y si comparten al menos un segmento en común. Para esto, se comparan los polígonos y se verifica si comparten al menos 2 vértices en común.
- Se actualizan los vecinos del *Feature* y se prosigue con los demás.

Algoritmo 2: Build Neighbours

```

Data: dictFeatures
Result: dictNeighbours
dictNeighbours ← [];
for feature in dictFeatures do
    boundingBox ← feature.getBoundingBox();
    candidateNeighbours ← [];
    for candidate in dictFeatures do
        if candidate intersects boundingBox then
            └ Add candidate to candidateNeighbours;
    for candidate in candidateNeighbours do
        if feature shares 2 vertices with possibleNeighbour then
            └ Add candidate as neighbour of feature in dictNeighbours

```

Lamentablemente, esta implementación nos entrega un algoritmo $O(n^2)$, lo cual es sumamente ineficiente, en especial en el caso de que la triangulación inicial resultase en una cantidad

enorme de triángulos o una gran cantidad de polígonos con muchos vértices. Si bien el hecho de que chequear el *bounding box* del polígono inicialmente nos permite ahorrar chequeos en vértices innecesariamente, sigue siendo en el peor de los casos de $O(n^2)$.

Una vez determinados los vecinos de cada polígono que compone la URH inicial, es posible implementar el algoritmo de disolución:

- Ordenar el diccionario de *Features* de acuerdo a lo que solicite el usuario. En el caso de la implementación actual, se permite ordenar de acuerdo al área.
- Mientras existan *Features* que procesar, obtener el primer polígono del diccionario ordenado.
- Ordenar los vecinos del *Feature* e iterar sobre estos.
- Por cada vecino, determinar si la unión de ambos *Features* cumple con los parámetros de Descriptor de Forma y de área. En el caso que no cumpla, se busca a otro de los vecinos y se realiza esta misma operación.
- Si la unión del *Feature* con uno de los vecinos cumple con lo requerido, se remueve el vecino del diccionario de *Features*, se agrega la unión de *Features* al diccionario, se actualizan los vecinos de acuerdo a la nueva unión, y finalmente se ordenan nuevamente el diccionario, para volver al ciclo inicial solamente realizando la unión de un solo vecino por iteración.
- En el caso de que se recorrieran todos los vecinos y ninguno de ellos cumple con las restricciones impuestas, se agrega el *Feature* a la lista de *Features* procesados y se elimina toda referencia de vecinos que tenga el *Feature* actual.

Algoritmo 3: Dissolve Features

Data: dictFeatures, dictNeighbours
Result: dissolvedFeatures
sort *dictFeatures*;
while *dictFeatures* is not empty **do**
 feature \leftarrow *dictFeatures.pop()*;
 sortedNeighbours \leftarrow sort *dictNeighbours.get(feature)*;
 for *neighbour* in *sortedNeighbours* **do**
 union \leftarrow *feature* \cup *neighbour*;
 if *union* follow restrictions **then**
 add *union* to *dictFeatures*;
 remove *neighbour* from *dictFeatures*;
 refresh neighbours;
 sort *dictFeatures*;
 break;
 if no available neighbours **then**
 add *feature* to *dissolvedFeatures*;
 remove all neighbours references to *feature*;
Return and draw *dissolvedFeatures*;

Como se puede ver, este algoritmo tiene un peor caso y caso promedio de $O(n^3)$, ya que cada vez que se une el *Feature* con uno de sus vecinos y se cumplen las restricciones, el listado de *Features* disminuye en 1, por lo que con esto quedaría un algoritmo de $O(n^2)$. Como al final de cada unión de *Features* se debe ordenar el listado nuevamente, lo cual es $O(n \log n)$, el algoritmo final se convierte en uno de $O(n^3)$, lo cual es bastante ineficiente.

Si bien en el algoritmo original presentado por Sanzana, 2011[9] se optó por obtener los *Features* y vecinos de área mayor, en la implementación actual se optó por dejar al usuario elegir entre ordenar por área de manera ascendente, descendente o sin orden alguno. Se espera que si el usuario elige un orden ascendente, vale decir ir diluyendo los *Features* desde el más pequeño al más grande, la disolución resultará con *Features* de un área mínima mayor a que las otras alternativas. Además, se espera ver si el no elegir un orden consigue resultados satisfactorios, ya que esto mejoraría el algoritmo a uno de $O(n^2)$.

4.3.1. Interfaz Gráfica y restricciones de entrada

Como se decidió realizar el proceso en 2 pasos, se hizo una interfaz gráfica sencilla separada para cada paso.

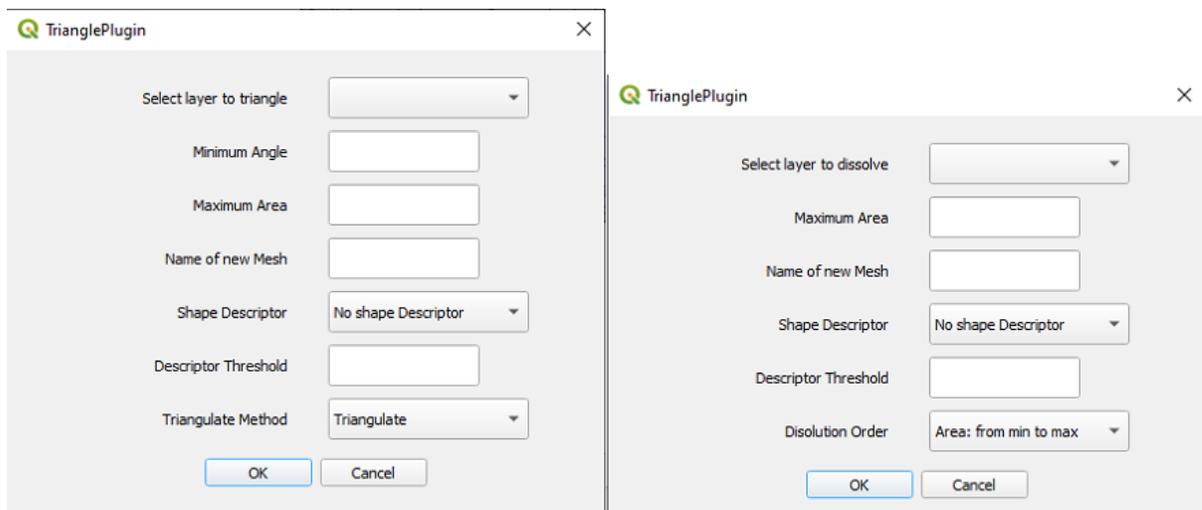


Figura 4.1: Interfaz gráfica de ambos pasos de triangulación y disolución

Como se ve en la Figura 4.1, los parámetros a ingresar son bastante sencillos y existen varios en común. A continuación se describen para qué sirve cada parámetro y qué restricciones tienen cada uno:

- **Select layer to triangle:** En este campo se selecciona el *layer* a procesar. Con esto se refiere a la URH a triangular o disolver. Estos son listados por nombre y son llenados automáticamente cada vez que se accede al plugin.
- **Minimum Angle:** Este campo está solamente incluido en la triangulación, y tiene como restricción principal que el ángulo mínimo debe ser estrictamente mayor a 0° y menor a

34°. Se escogió la restricción de 34° ya que el algoritmo implementado por J. Shewchuk explícitamente detalla que para ángulos mayores a 34°, la triangulación usualmente no termina.

- Maximum Area: La única restricción es que debe ser un número positivo. La unidad de área es la misma que la del layer original.
- Name of new Mesh: Nombre de la malla a retornar. Si no existe nombre, entonces se escoge por defecto el nombre “New Mesh”.
- Shape Descriptor: Se escoge el descriptor que se quiere imponer, ya sea uno de los 4 descriptores de forma implementados, o simplemente no imponer ninguna restricción de este tipo.
- Descriptor Threshold: Se debe escoger un número positivo entre 0 y 1, la cual junto con el descriptor de forma permiten restringir el resultado.
- Triangulate Method: Se permite escoger si se requiere triangular los *Features* que no cumplen las restricciones, o si simplemente se quieren filtrar y retornar estos.
- Disolution Order: Se permite escoger el orden de disolución. Como opciones están las de disolver en cuanto al área de manera ascendente, descendente o sin orden alguno.

Capítulo 5

Validación

Para el proceso de validación, se estudiará el caso de URHs situados en el sector de Lo Barnechea-La Dehesa, al norte de Cerro Alvarado.

Este caso en particular representa un sector periurbano, en el cual no es posible aplicar ni el modelo hidrológico que modela los sectores totalmente rurales ni tampoco los urbanos. Dentro de este caso de estudio, podemos ver los siguientes casos que resulta interesante observar:

- Polígonos con agujeros al interior, con centroides dentro y fuera.
- Polígonos alargados, los cuales representan una calle, con centroides dentro y fuera.
- Polígonos complejos y delgados, los cuales representan varias calles unidas.
- Polígonos relativamente bien formados, pero de gran área que requieren ser disueltos.

Se han identificado los siguientes URHs dentro del caso de estudio para evaluar los algoritmos:

- a) Calle con Agujero en medio. Altos índices de convexidad y solidez, pero centroide fuera del polígono.
- b) Calle haciendo retorno en loop. Bajo índice de convexidad y solidez, con centroide fuera.
- c) Calle alargada. Alto índice de convexidad, pero con centroide fuera.
- d) Varias calles unidas en una URH. Centroide fuera del polígono y bajísimos índices en los distintos factores de forma
- e) Centro deportivo, en donde hay varios agujeros, pero con centroide dentro.
- f) URH que representa un sector rural, cercano a la periferia. Relativamente altos índices en los distintos factores de forma y con centroide dentro, pero de un área considerablemente alta.

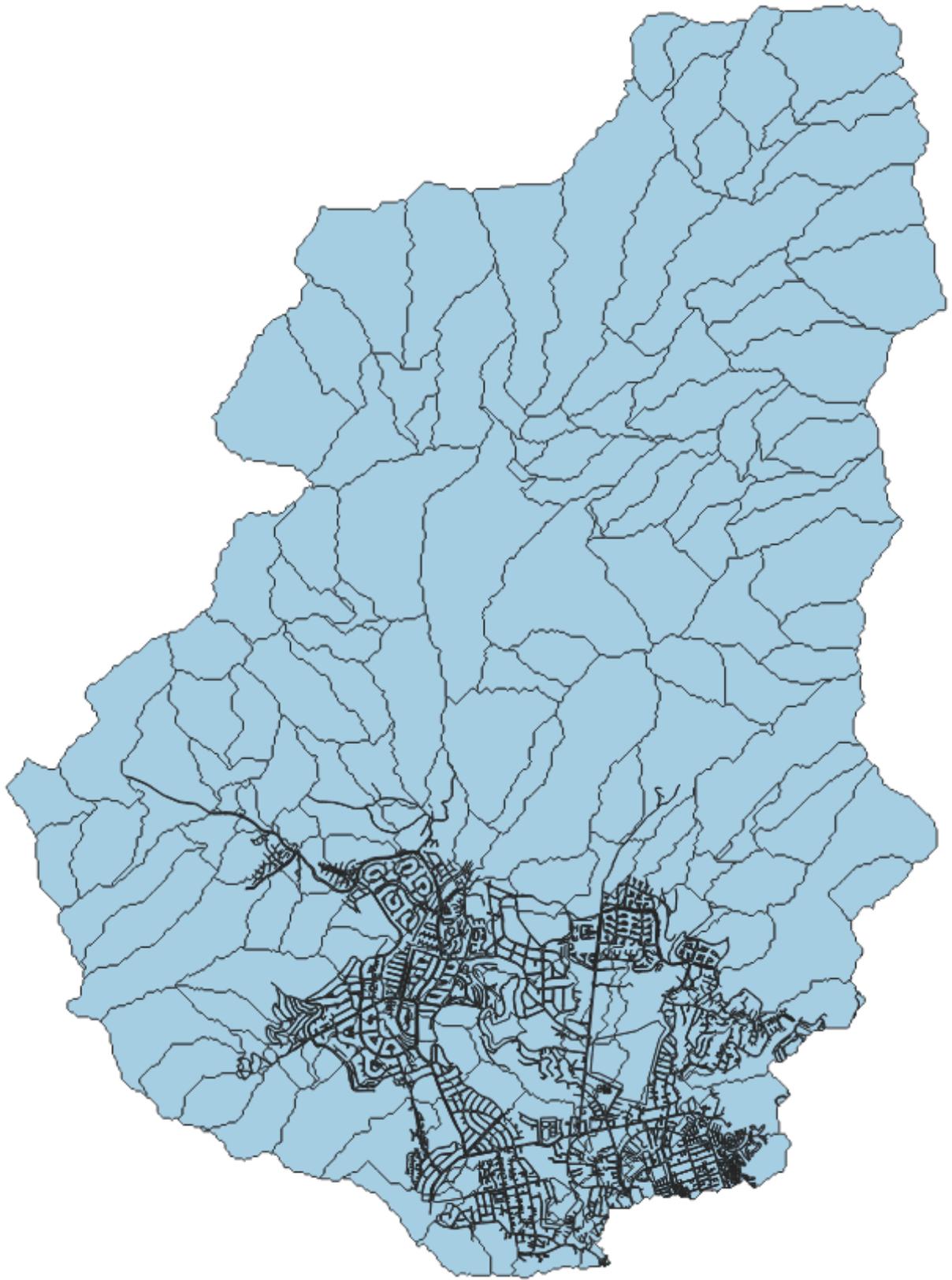
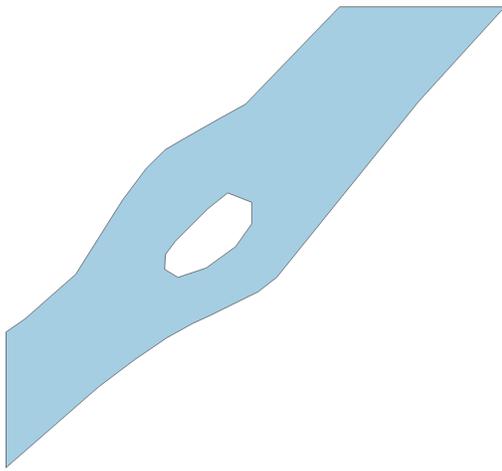
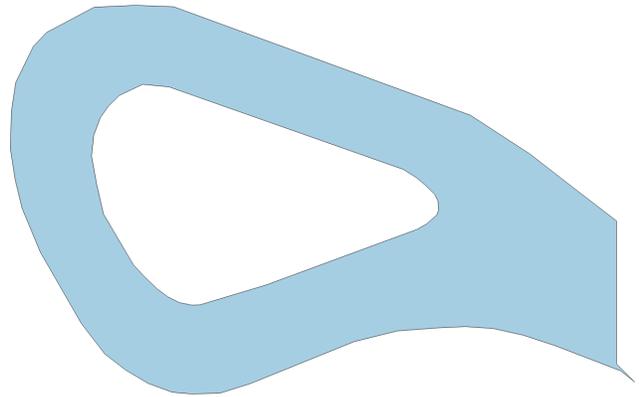


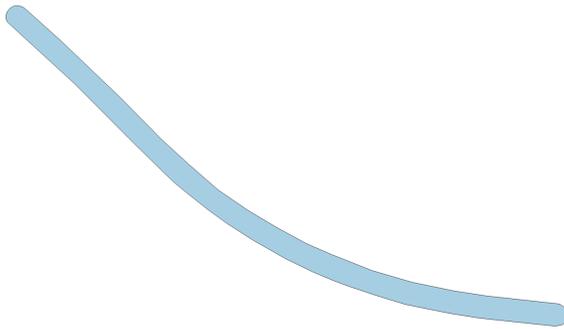
Figura 5.1: URH sector Lo Barnechea-La Dehesa



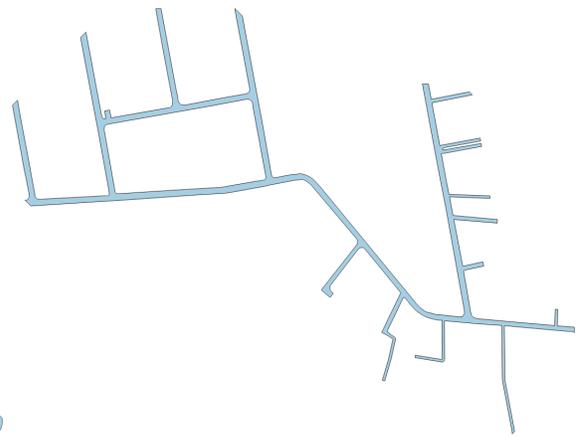
(a) Calle con agujero



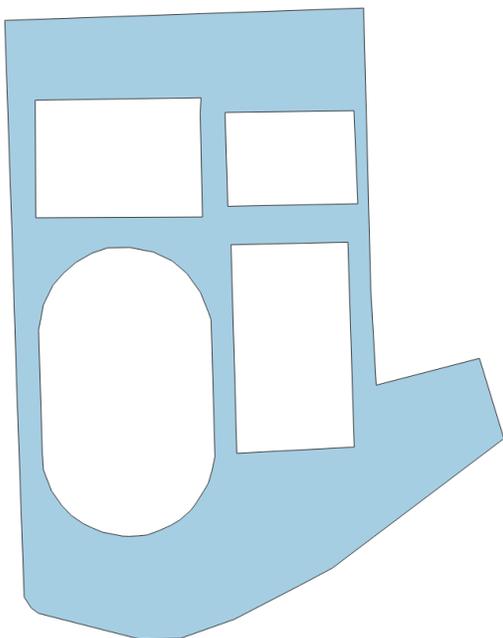
(b) Calle retorno



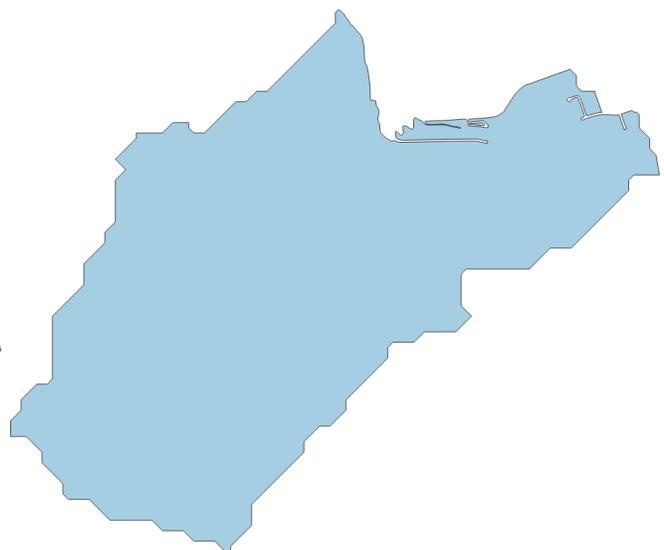
(c) Calle simple



(d) Varias calles



(e) Centro deportivo



(f) URH de gran área

Figura 5.2: URHs Casos de estudio

URHs Casos de Estudio						
Figura	Área(m^2)	Perímetro(m)	FF*	C	IC	IS
a)Calle con agujero	210,657	94,637	0,296	0,544	0,845	0,863
b)Calle retorno	453,939	163,770	0,213	0,461	0,649	0,638
c)Calle simple	1.720,751	457,561	0,103	0,321	0,985	0,357
d)Varias calles	15.362,573	5.043,351	0,008	0,087	0,361	0,076
e)Centro deportivo	42.782,643	2.607,542	0,079	0,281	0,432	0,515
f)URH de gran área	1.342.951,145	7.517,16	0,299	0,546	0,679	0,807

Tabla 5.1: URHs Casos de Estudio

5.1. Triangulación

El primer paso en este proceso de mejoramiento de mallas es la triangulación de estas. La implementación de Triangle como plugin permite obtener la triangulación directamente desde QGIS.

Para efectos de validar la triangulación, se considerarán como variables a medir: Número de triángulos, número de vértices originales y luego de la triangulación y el área de la triangulación más grande.

Para efectos de validación, se aplicó el paso de Triangulación sobre los 6 casos de estudio con la siguientes restricciones : Sin restricción alguna, restricción de ángulo de 5° , restricción de ángulo de 25° y restricción de área total dividido por 100.

URHs Casos de Estudio: Triangulados sin restricción					
Figura	N° Triángulos	N°Vértices Orig	N° Vértices Tria	Área máx(m^2)	
a)Calle con agujero	31	31	31	25,706	
b)Calle retorno	59	59	59	46,299	
c)Calle simple	49	51	51	113,431	
d)Varias calles	394	394	394	506,515	
e)Centro deportivo	77	71	71	5.079,655	
f)URH de gran área	620	622	622	151.910,283	

Tabla 5.2: URHs Casos de Estudio: Triangulados sin restricción

URHs Casos de Estudio: Triangulados con restricción de ángulo 5°					
Figura	N° Triángulos	N°Vértices Orig	N° Vértices Tria	Área máx(m^2)	
a)Calle con agujero	31	31	31	25,706	
b)Calle retorno	64	59	62	46,299	
c)Calle simple	49	51	51	113,431	
d)Varias calles	473	394	456	410,893	
e)Centro deportivo	88	71	78	5.079,655	
f)URH de gran área	813	622	721	85.786,811	

Tabla 5.3: URHs Casos de Estudio: Triangulados con restricción de ángulo 5°

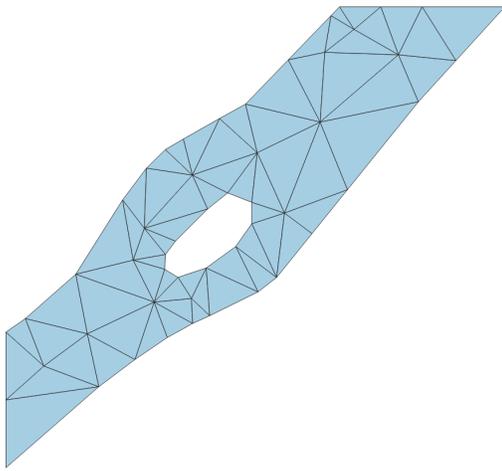
URHs Casos de Estudio: Triangulados con restricción de ángulo 25°					
Figura	N° Triángulos	N°Vértices Orig	N° Vértices Tria	Área máx(m ²)	
a)Calle con agujero	64	31	51	10,883	
b)Calle retorno	143	59	112	18,761	
c)Calle simple	78	51	70	111,583	
d)Varias calles	1.371	394	1.114	105,168	
e)Centro deportivo	234	71	179	1.245,808	
f)URH de gran área	2.499	622	1.645	54.647,459	

Tabla 5.4: URHs Casos de Estudio: Triangulados con restricción de ángulo 25°

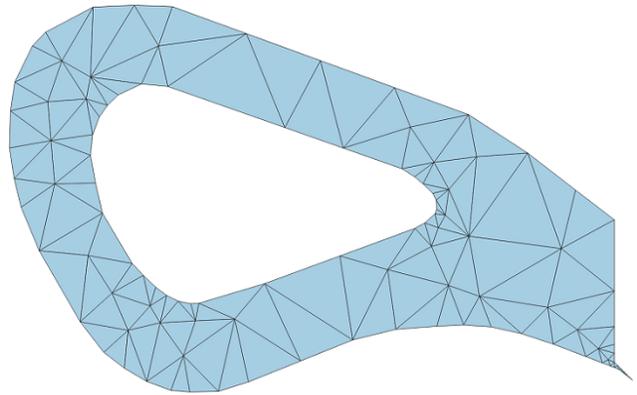
URHs Casos de Estudio: Triangulados con restricción de área 1 % de área máxima					
Figura	N° Triángulos	N°Vértices Orig	N° Vértices Tria	Área máx(m ²)	
a)Calle con agujero	162	31	102	2,082	
b)Calle retorno	174	59	122	4,537	
c)Calle simple	168	51	122	16,950	
d)Varias calles	444	394	444	151,655	
e)Centro deportivo	175	71	132	423,144	
f)URH de gran área	732	622	678	13.101,554	

Tabla 5.5: URHs Casos de Estudio: Triangulados con restricción de 1 % de área máxima

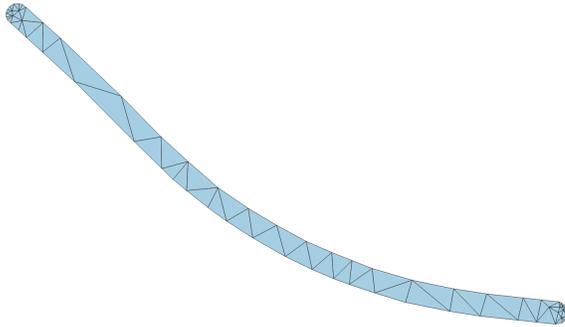
En estos resultados es posible ver que las triangulaciones fueron realizadas con éxito y que a medida que se aumenta la calidad en cuanto al ángulo mínimo, tanto el número de triángulos como de vértices aumentan, ya que la triangulación estima necesario agregar vértices que no estaban el polígono original.



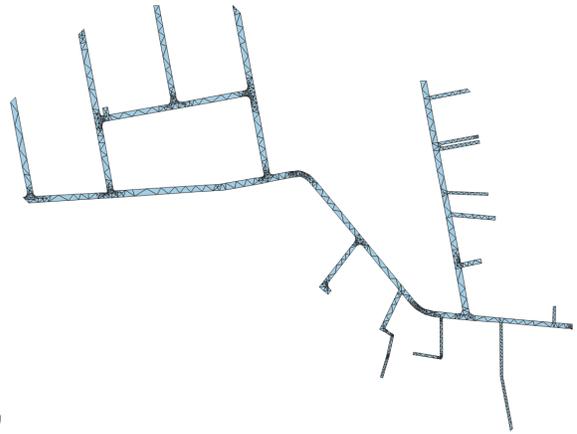
(a) Calle con agujero



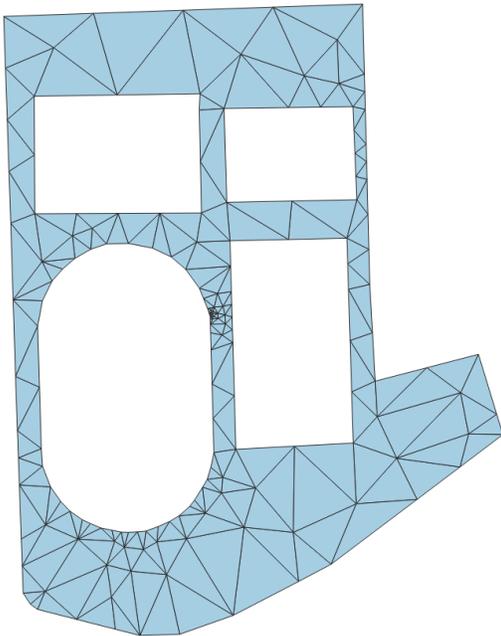
(b) Calle retorno



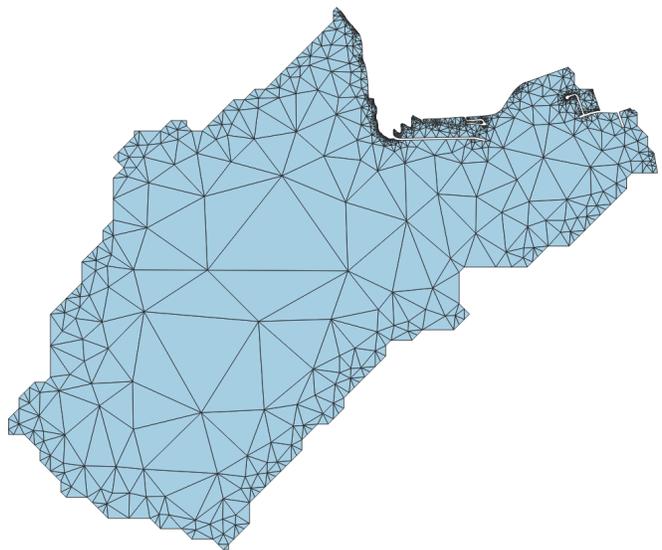
(c) Calle simple



(d) Varias calles

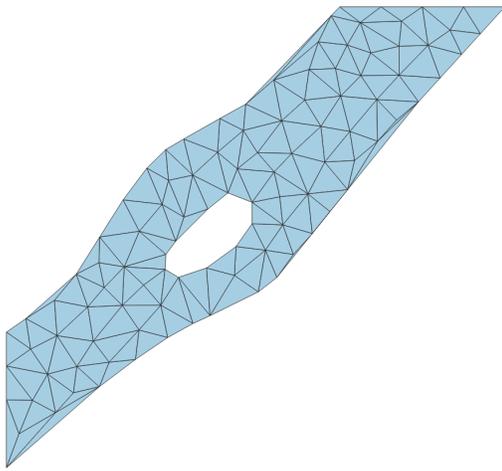


(e) Centro deportivo

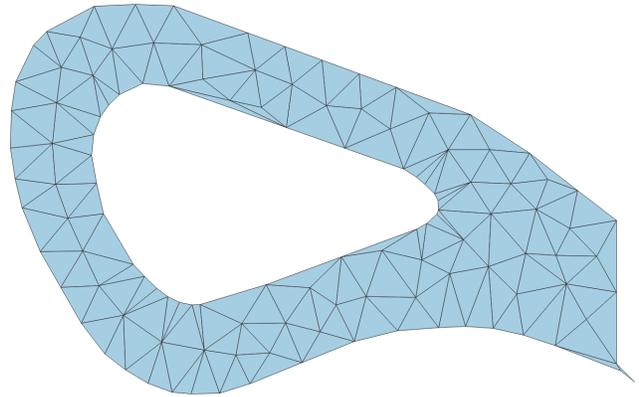


(f) URH de gran área

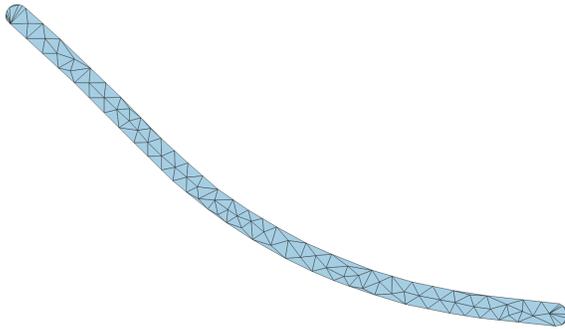
Figura 5.3: URHs Casos de estudio triangulados con ángulo 25°



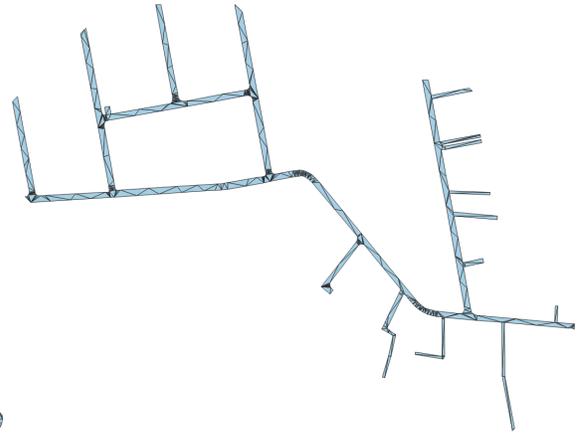
(a) Calle con agujero



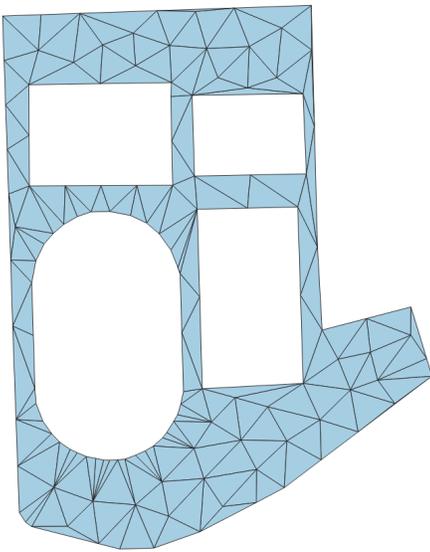
(b) Calle retorno



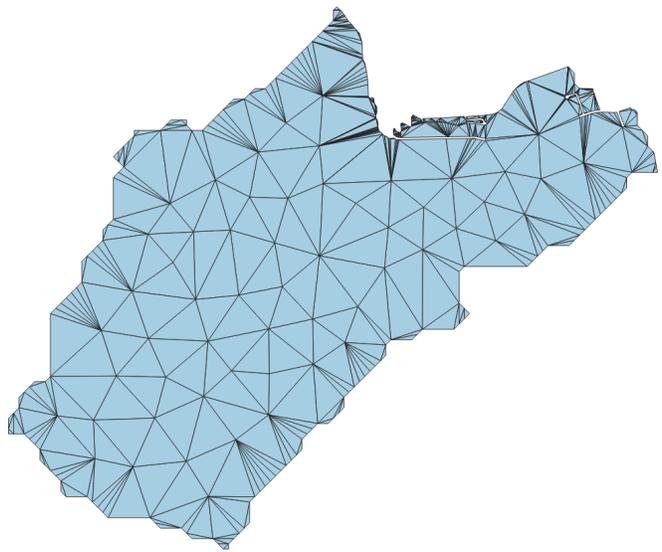
(c) Calle simple



(d) Varias calles



(e) Centro deportivo



(f) URH de gran área

Figura 5.4: URHs Casos de estudio triangulados con restricción de 1% de area máxima

5.2. Disolución

Para la disolución, se muestran los resultados aplicados sobre los 6 casos de estudio triangulados con una restricción de ángulo de 25° .

Las variables a medir en este caso serán el número de triángulos originales, número de polígonos resultantes, factor de forma promedio de los polígonos resultantes y número de polígonos con su centroide fuera.

En cuanto a los factores de disolución, se aplicarán los siguientes criterios: Factor de forma de 0,3, Compacidad de 0,5, Índice de solidez de 0,85, Índice de convexidad de 0,75 y disolución a un área máxima de un 10% del área original.

URHs Casos de Estudio: Disolución con restricción de Factor de forma de 0,3					
Figura	N° Triángulos	N° Polígonos Final	FF promedio	N° centroides fuera	
a)Calle con agujero	64	2	0,377	0	
b)Calle retorno	143	2	0,327	2	
c)Calle simple	78	5	0,373	0	
d)Varias calles	1.371	98	0,384	2	
e)Centro deportivo	234	9	0,363	1	
f)URH de gran área	2.499	2	0,342	0	

Tabla 5.6: URHs Casos de Estudio: Disolución con restricción de Factor de forma de 0,3

URHs Casos de Estudio: Disolución con restricción de Compacidad de 0,5					
Figura	N° Triángulos	N° Polígonos Final	C promedio	N° centroides fuera	
a)Calle con agujero	64	1	0,544	1	
b)Calle retorno	143	2	0,524	2	
c)Calle simple	78	4	0,580	0	
d)Varias calles	1.371	77	0,571	4	
e)Centro deportivo	234	6	0,540	0	
f)URH de gran área	2.499	1	0,546	0	

Tabla 5.7: URHs Casos de Estudio: Disolución con restricción de Compacidad de 0,5

URHs Casos de Estudio: Disolución con restricción de Índice de Solidez de 0,85					
Figura	N° Triángulos	N° Polígonos Final	IS promedio	N° centroides fuera	
a)Calle con agujero	64	5	0,917	0	
b)Calle retorno	143	9	0,898	0	
c)Calle simple	78	4	0,926	0	
d)Varias calles	1.371	63	0,919	0	
e)Centro deportivo	234	17	0,909	0	
f)URH de gran área	2.499	37	0,903	0	

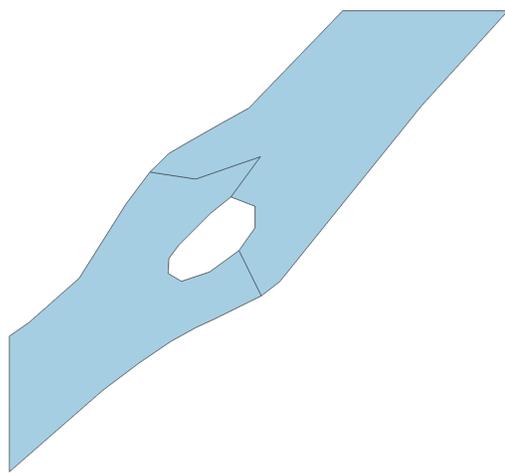
Tabla 5.8: URHs Casos de Estudio: Disolución con restricción de Índice de Solidez de 0,85

URHs Casos de Estudio: Disolución con restricción de Índice de Convexidad de 0,75					
Figura	N° Triángulos	N° Polígonos Final	<i>IC</i> promedio	N° centroides fuera	
a)Calle con agujero	64	1	0,845	1	
b)Calle retorno	143	2	0,825	2	
c)Calle simple	78	1	0,985	1	
d)Varias calles	1.371	14	0,843	11	
e)Centro deportivo	234	5	0,793	3	
f)URH de gran área	2.499	3	0,835	0	

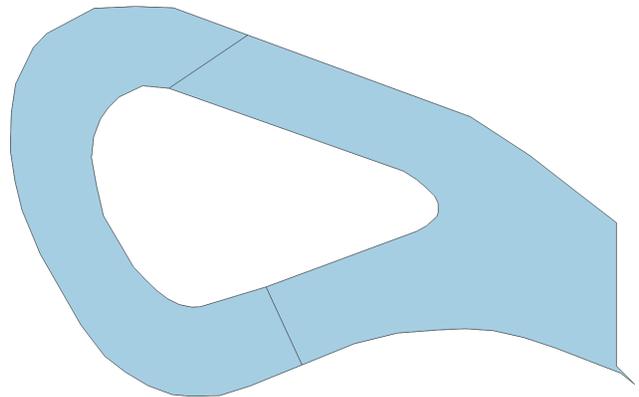
Tabla 5.9: URHs Casos de Estudio: Disolución con restricción de Índice de Convexidad de 0,75

URHs Casos de Estudio: Disolución con restricción de 10% de área máxima					
Figura	N° Triángulos	N° Polígonos Final	Área mínima	N° centroides fuera	
a)Calle con agujero	64	15	9,066	1	
b)Calle retorno	143	14	20,375	1	
c)Calle simple	78	15	82,187	0	
d)Varias calles	1.371	14	374,245	10	
e)Centro deportivo	234	15	2.026,259	2	
f)URH de gran área	2.499	14	58.007,039	2	

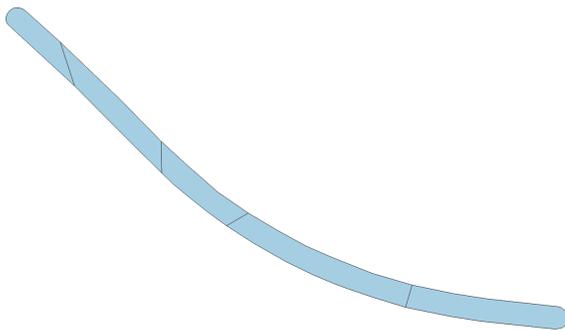
Tabla 5.10: URHs Casos de Estudio: Disolución con restricción de 10% de área máxima



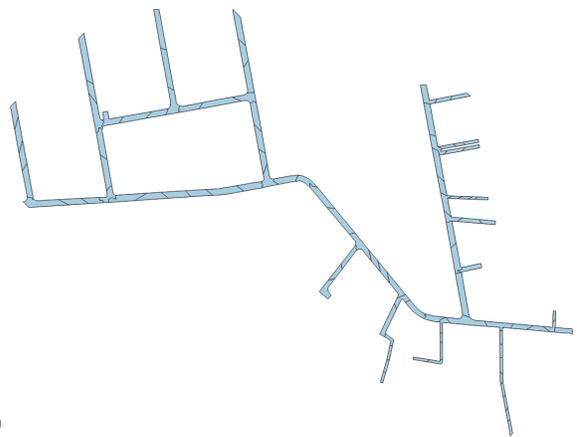
(a) Calle con agujero



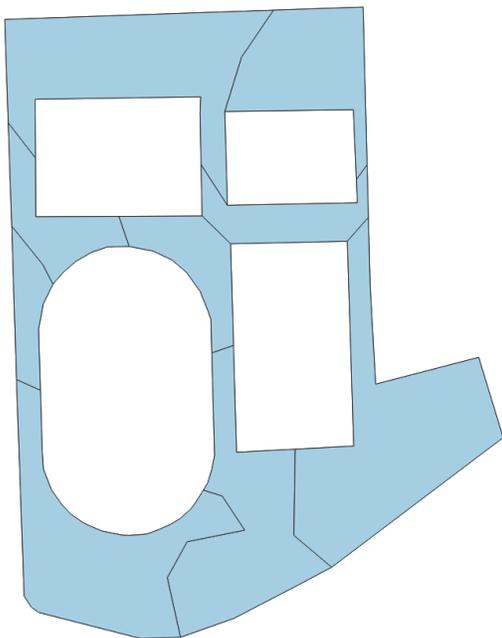
(b) Calle retorno



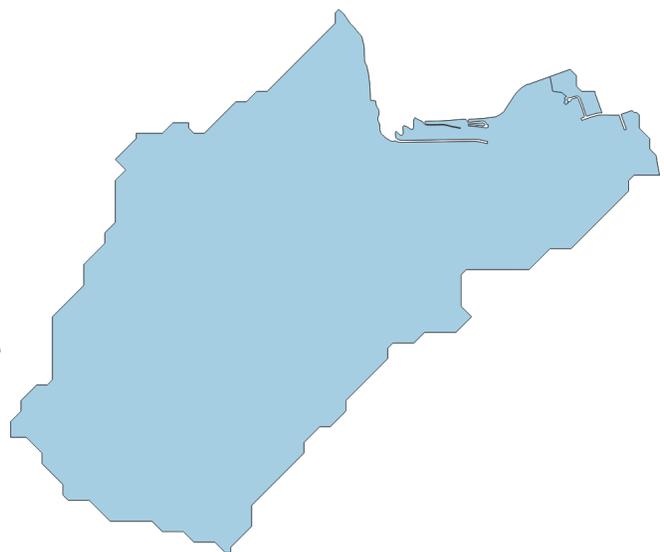
(c) Calle simple



(d) Varias calles

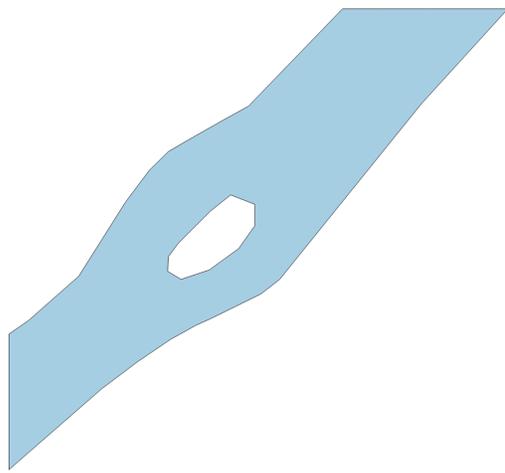


(e) Centro deportivo

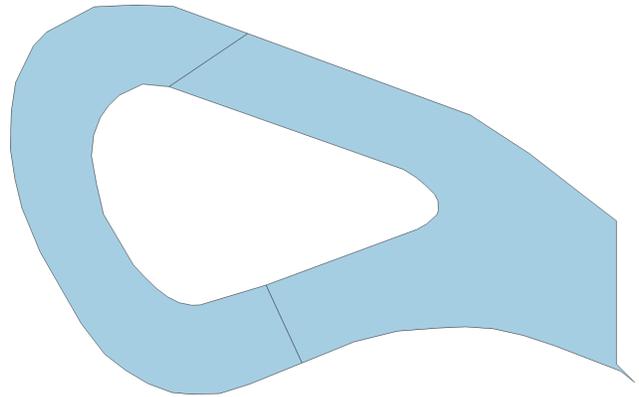


(f) URH de gran área

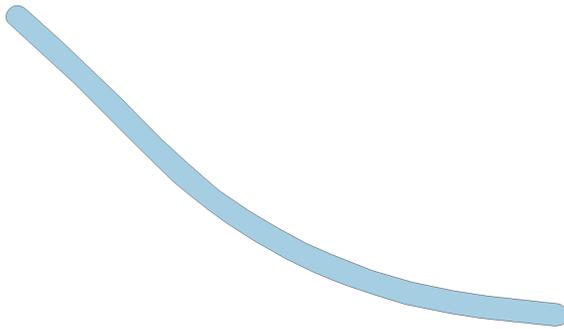
Figura 5.5: URHs Casos de estudio disueltos con restricción de Factor de Forma 0, 30



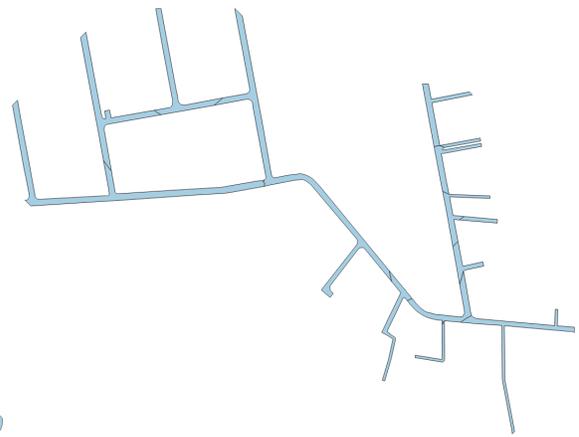
(a) Calle con agujero



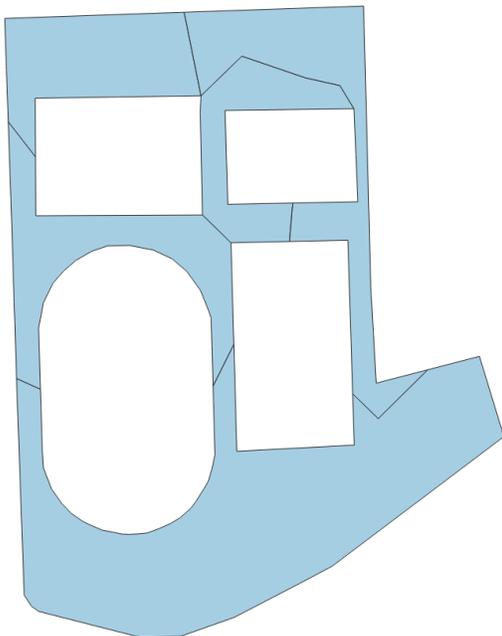
(b) Calle retorno



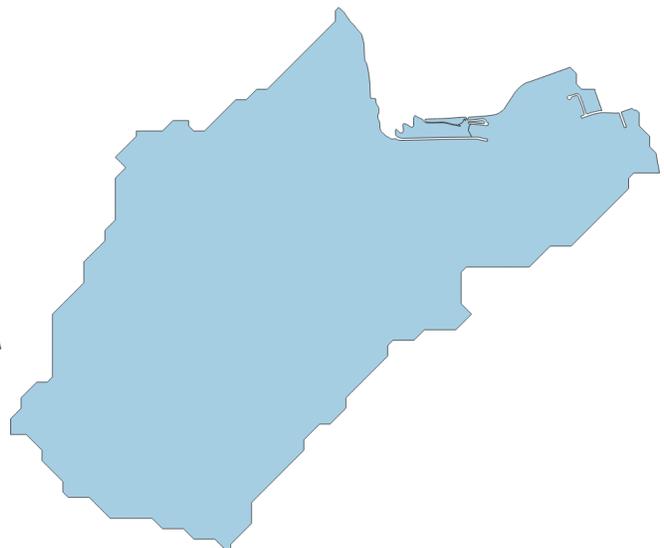
(c) Calle simple



(d) Varias calles

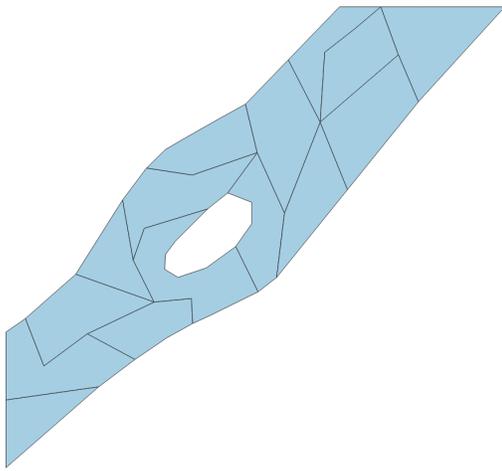


(e) Centro deportivo

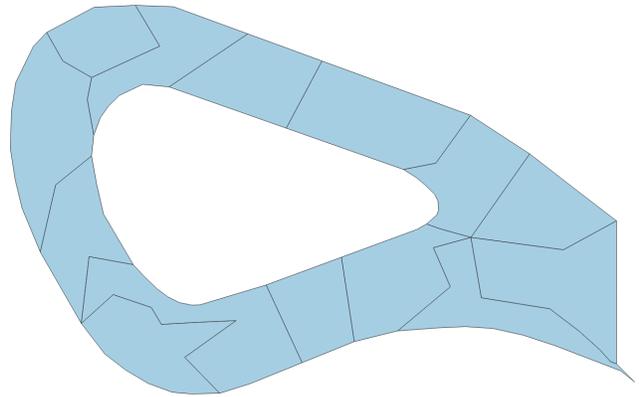


(f) URH de gran área

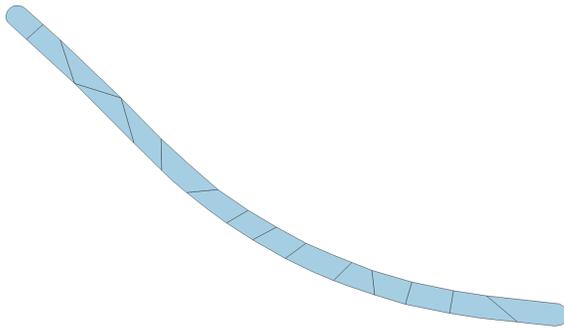
Figura 5.6: URHs Casos de estudio disueltos con restricción de Índice de Convexidad 0,75



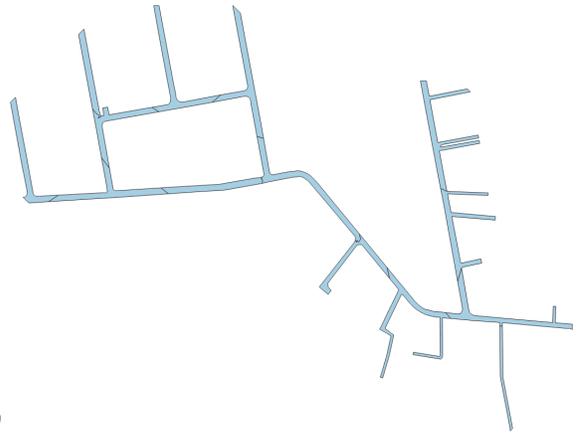
(a) Calle con agujero



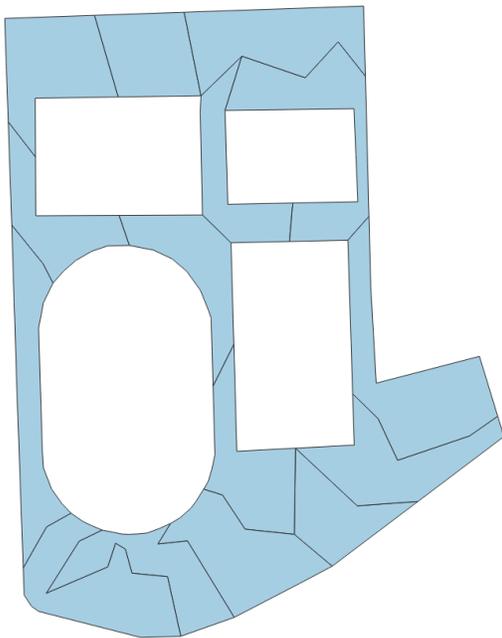
(b) Calle retorno



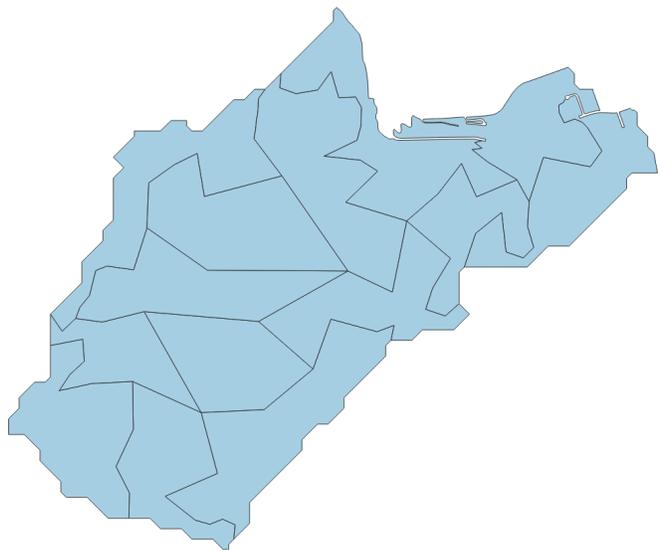
(c) Calle simple



(d) Varias calles



(e) Centro deportivo



(f) URH de gran área

Figura 5.7: URHs Casos de estudio disueltos con restricción de 10% de área máxima

Dentro de los resultados de estos casos de estudio, se ven resultados diversos por la naturaleza de estos casos y por los distintos descriptores de forma a considerar.

Por ejemplo, en los casos de restricciones por factores de forma y compacidad, los cuales comparan la figura inicial con respecto a un círculo, estos algoritmos efectivamente cumplen su objetivo de eliminar los polígonos de forma extremadamente alargadas, como el caso de la calle simple y de las varias calles. También permite, en cierta medida, obtener polígonos cuyo centroide están mayormente dentro de la figura original, tal como se ve en el caso de varias calles, en donde la cantidad de polígonos con centroide fuera rondan entre un 2% y un 5% respecto a los polígonos totales. Sin embargo, esto no es su objetivo principal, e incluso hay casos como en calle retorno en los cuales ambas soluciones no son suficientes para asegurar centroides dentro con los parámetros dados (se debe subir la restricción, pero con el costo de aumentar la cantidad de polígonos resultantes).

En los casos de restricciones por índices de solidez y convexidad vemos resultados muy distintos, los cuales son dados mayormente por el factor de restricción elegido, pero también tiene relación con el tipo de índice. Si bien vemos que en el caso de un índice de solidez igual a 0,85 se logra obtener solamente polígonos con centroide en su interior, tiene el costo de que la cantidad de polígonos es bastante mayor, lo que puede ser un factor deseado o no deseado por el usuario: por ejemplo el caso de varias calles es necesario separar totalmente, ya que corresponden a un conjunto de calle ,pero en el caso del centro deportivo, el cual representa una zona definida, no es necesario separar en tantos polígonos. Por otro lado, se ve que el índice de convexidad igual a 0,75 logra un número de polígonos bajo, y en algunos casos retorna el polígono original ya que este ya cumplía con la condición, pero estos polígonos no logran el objetivo de tener su centroide interior. Esto se ve mayormente en varias calles, en donde la mayor cantidad de polígonos resultantes corresponden a polígonos largos y alargados, los cuales suelen tener una curvatura que empuja el centroide fuera, pero que posee un efecto un efecto mayor en el índice de solidez que en el índice de convexidad. Incluso aumentando el índice de convexidad a un 0,9 no logra eliminar del todo los polígonos con centroide fuera en los que se logran 4 polígonos con centroide fuera de un total de 29 polígonos, vale decir un 13,8%; e incluso un factor de 0,95, en donde se logran 2 polígonos con centroide fuera de un total de 49 polígonos, un 4,08%.

5.3. Evaluación de proceso completo

Si bien el proceso de triangulación y posterior disolución pueden ser aplicados separadamente y requieren ser analizados por separado, es necesario evaluar el proceso completo sobre los casos de prueba propuestos.

5.3.1. Efecto de la triangulación inicial

Uno de los puntos a evaluar, es el cómo afecta la calidad de la triangulación inicial al proceso de disolución. En particular, se medirá el efecto que tiene el número de triángulos que se dan en la triangulación inicial y además el efecto del ángulo mínimo de los triángulos.

Para ambos casos, se hará la evaluación sobre el caso de prueba del Centro Deportivo, y en el caso de la evaluación por número de polígonos, se impondrá un ángulo mínimo de 20° para mantener una calidad mínima de los triángulos y sólo se modificará el parámetro de área máxima para lograr la cantidad de triángulos deseada.

En cuanto a los factores de forma, se decidió evaluar un Factor de Forma de 0,3 y un Índice de Convexidad igual a 0,95.

Disolución en base a triangulación ángulo mín. 20° y distintas restricciones de área				
Restricción	N° Tri. Inic.	N° Polígonos	$FF = 0,3$	N° Polígonos $IC = 0,95$
Sin Restricción	167	12		11
Área máx $100m^2$	707	9		15
Área máx $50m^2$	1338	10		44
Área máx $20m^2$	3360	10		184
Área máx $10m^2$	6636	10		319

Tabla 5.11: Disolución en base a triangulación ángulo mín. 20° y distintas restricciones de área

Disolución en base a triangulación distintos ángulos mínimos				
Restricción	N° Tri. Inic.	N° Polígonos	$FF = 0,3$	N° Polígonos $IC = 0,95$
Sin Restricción	77	11		9
Ángulo mínimo 10°	114	12		11
Ángulo mínimo 20°	167	12		11
Ángulo mínimo 30°	331	9		17

Tabla 5.12: Disolución en base a triangulación distintos ángulos mínimos

Como se puede ver, los resultados son bastante dispares en cuanto al número de triángulos iniciales. Para el proceso de triangulación, se impuso un ángulo mínimo de 20° para todos los casos, y distintas restricciones de área. En cuanto a las restricciones de forma de los polígonos, no se vio una diferencia notable en cuanto al factor de forma, el cual se probó con un valor de 0,3, vale decir una restricción lo suficientemente relajada para permitir ciertos polígonos finales.

Por otro lado, la restricción más estricta de un Índice de Convexidad igual a 0,95 fue notablemente afectada por la cantidad de triángulos iniciales, resultando en una cantidad de polígonos finales considerable, tal como se puede ver en las Figuras 5.8c y 5.8d. Es importante mencionar que este resultado no es debido al uso de un factor de forma distinto, sino que a que la restricción de forma es más estricta. Resultados similares ocurren para el caso $FF = 0,6$.

Otro factor a considerar es que el aumento en la cantidad de triángulos iniciales tiene otro efecto indeseado en la disolución final. Como se puede ver en las Figuras 5.8a y 5.8b, la disolución final resulta en una cantidad similar de polígonos, sin embargo, en el segundo caso se ve que las divisiones del polígono tienen una forma más irregular. Si bien esto no presenta

un problema mayor en la disolución, vale la pena mencionar ya que esto obligaría a tener que usar nuevamente el algoritmo de suavizado de bordes de polígonos.

En cuanto al efecto del ángulo mínimo en la triangulación inicial, no se ven cambios sustanciales en el número de polígonos finales. La única diferencia se produce principalmente por la cantidad de triángulos iniciales debido a la restricción del ángulo mínimo, lo cual afecta directamente la cantidad de polígonos finales.

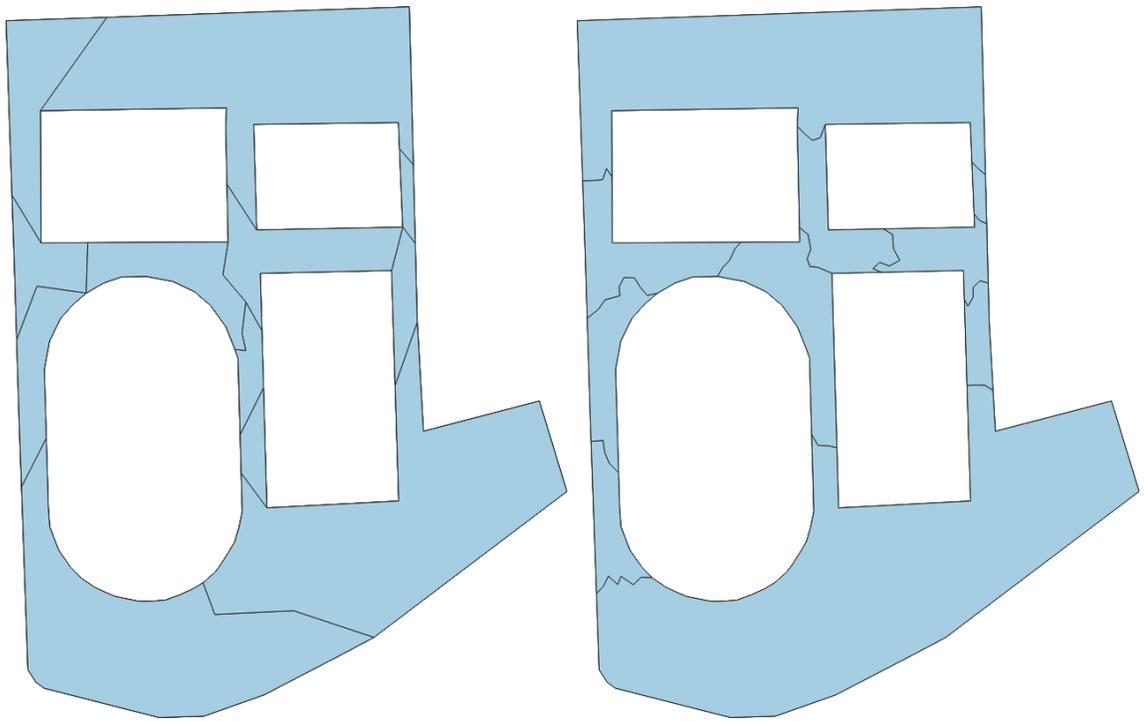
Sin embargo, en cuanto a la calidad de los polígonos finales la restricción de ángulo resulta de suma importancia en cuanto a evitar la aparición de triángulos que no cumplan la condición inicial de descriptor de forma, particularmente en cuanto a los Factores de Forma y Compacidad.

Disolución en base a triangulación distintos ángulos mínimos con $FF = 0,3$			
Restricción	N° Tri. Inic.	N° Polígonos Final	N° Polígonos $FF < 0,3$
Sin Restricción	77	11	4
Ángulo mínimo 10°	114	12	3
Ángulo mínimo 20°	167	12	0
Ángulo mínimo 30°	331	9	0

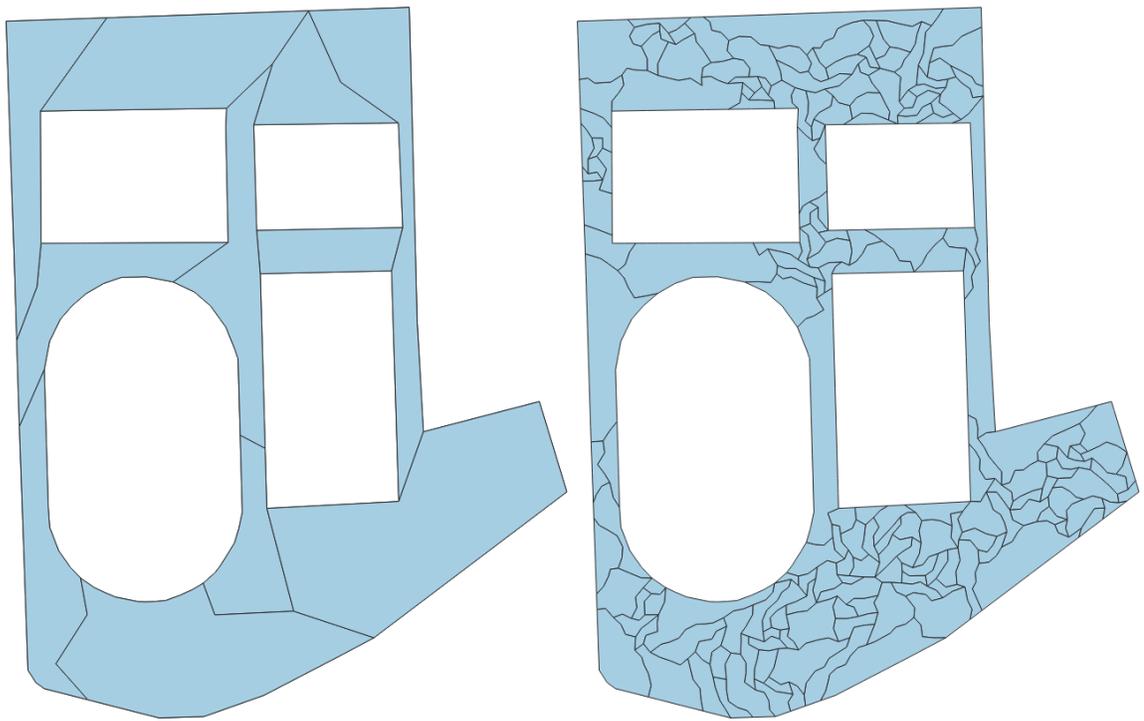
Tabla 5.13: Disolución en base a triangulación distintos ángulos mínimos con $FF = 0,3$

Como se ve en la Tabla 5.13, la restricción de ángulos mínimos de 0° y 10° obtuvo como resultado final polígonos que no cumplieron la restricción de descriptor de forma impuesta. Esto se puede explicar porque la cantidad de triángulos iniciales generados con restricciones de 20° y 30° tienden a generar triángulos con un Factor de Forma más cercano a 1 que con respecto a los generados con restricciones de 0° y 10°, los cuales tienden a generar menos triángulos, pero más alargados. Si bien el algoritmo de disolución busca mejorar estos polígonos mediante su fusión con otros polígonos, hay casos en los que no existe vecino directo con el cual se pueda disolver cumpliendo las restricciones, por lo que el algoritmo termina sin solución para estos polígonos alargados. Es posible visualizar esto en la Figura 5.9a, en donde todos los triángulos que aparecen en la disolución final no cumplen con el Factor de Forma impuesto.

En base a estos resultados, es recomendable entonces generar triangulaciones con restricciones de ángulo de 20° o mayor, para reducir estos problemas.

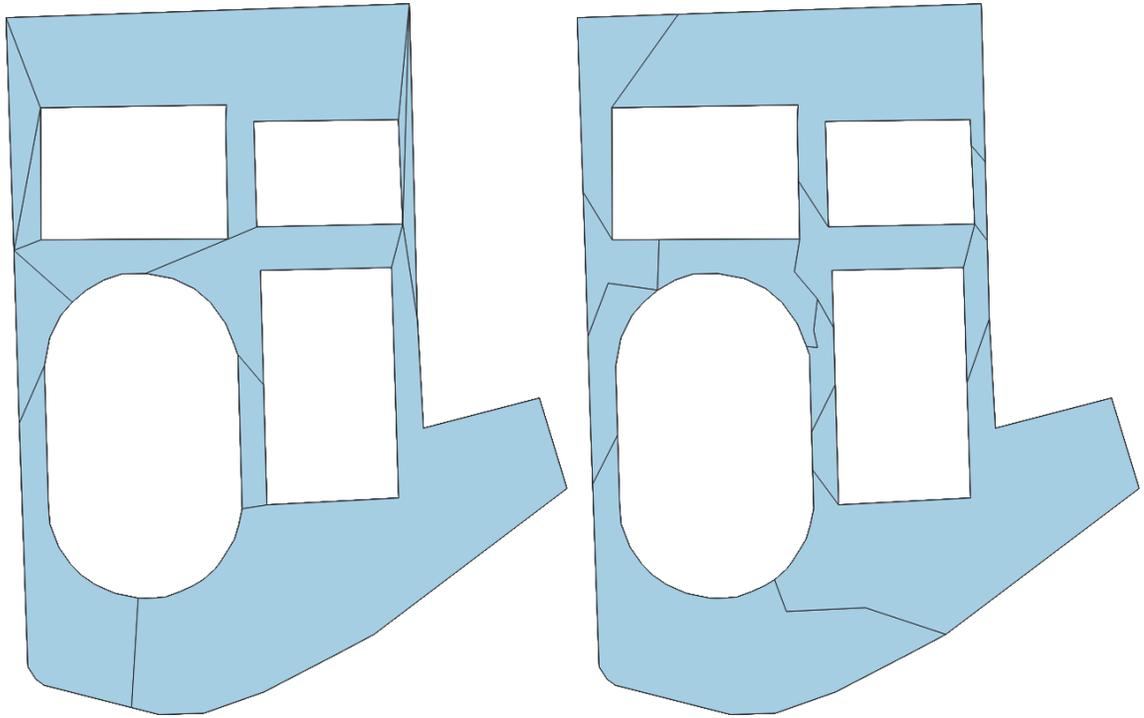


(a) Triang. Inic. 167, Disolución $FF = 0,3$ (b) Triang. Inic. 3360, Disolución $FF = 0,3$

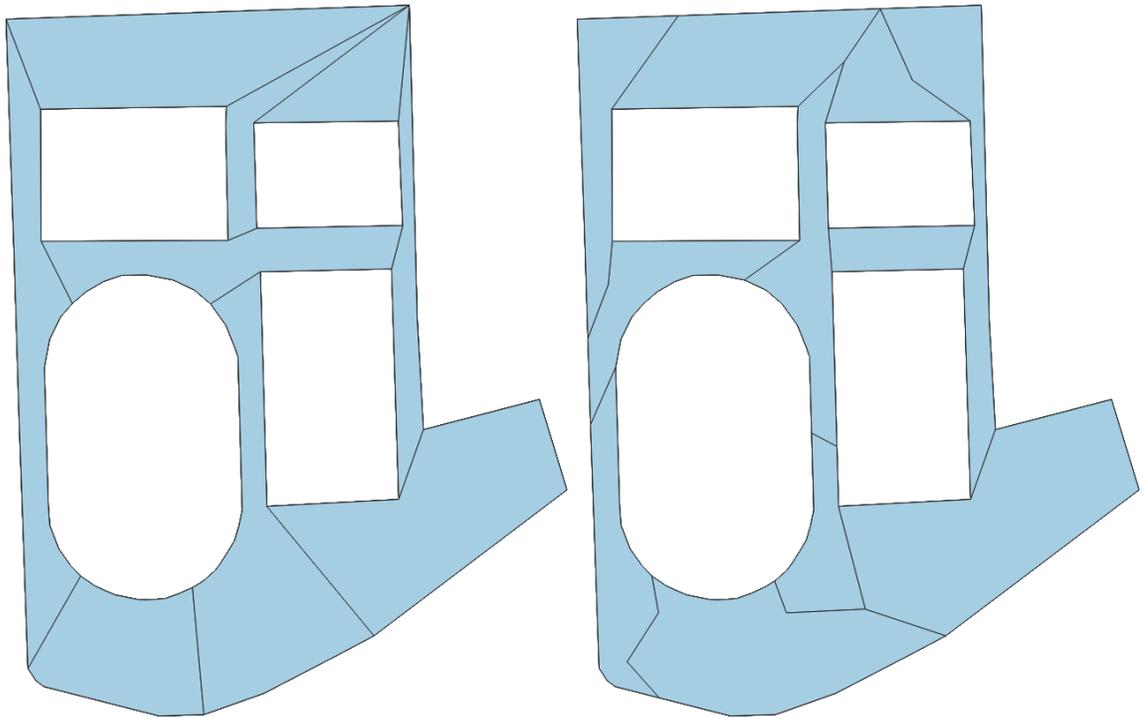


(c) Triang. Inic. 167, Disolución $IC = 0,95$ (d) Triang. Inic. 3360, Disolución $IC = 0,95$

Figura 5.8: Disolución en base a triangulación ángulo mín. 20° y distintas restricciones de área



(a) Ángulo mín. 0° , Disolución $FF = 0,3$ (b) Ángulo mín. 20° , Disolución $FF = 0,3$



(c) Ángulo mín. 0° , Disolución $IC = 0,95$ (d) Ángulo mín. 20° , Disolución $IC = 0,95$

Figura 5.9: Disolución en base a triangulaciones con distintos ángulos mínimos

5.3.2. Aplicación sobre polígonos delgados y alargados

Uno de los objetivos planteados inicialmente es el poder detectar y corregir los polígonos delgados y alargados. Para esto, se plantearon dos descriptores de forma a evaluar: Factor de Forma y Compacidad, los cuales solo se diferencian en la sensibilidad con respecto al perímetro, pero esencialmente miden lo mismo: qué tan parecido es el polígono a la forma de un círculo, en base al área y el perímetro.

A continuación, se muestran los resultados de su aplicación en dos tipos de URHs que poseen este problema: uno que representa una calle y otro que representa una unión de calles.

Disolución de un polígono delgado y alargado triangulado en 25°			
Restricción	N° Polígonos Final	FF Promedio	N° Polígonos $FF < \text{restricción}$
$FF = 0,2$	3	0,27733	0
$FF = 0,3$	5	0,373	0
$FF = 0,4$	9	0,44344	2
$FF = 0,5$	13	0,551	3
$FF = 0,6$	22	0,60745	7

Tabla 5.14: Disolución de un polígono delgado y alargado triangulado en 25°

Disolución de un polígono unión de calles triangulado en 25°			
Restricción	N° Polígonos Final	FF Promedio	N° Polígonos $FF < \text{restricción}$
$FF = 0,2$	55	0,282491	0
$FF = 0,3$	98	0,384673	0
$FF = 0,4$	147	0,467646	6
$FF = 0,5$	269	0,549922	54
$FF = 0,6$	521	0,57561	253

Tabla 5.15: Disolución de un polígono unión de calles triangulado en 25°

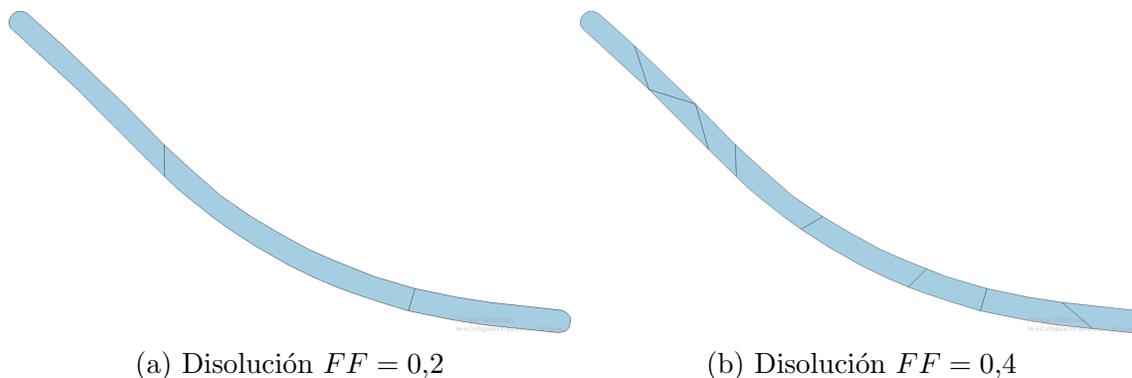
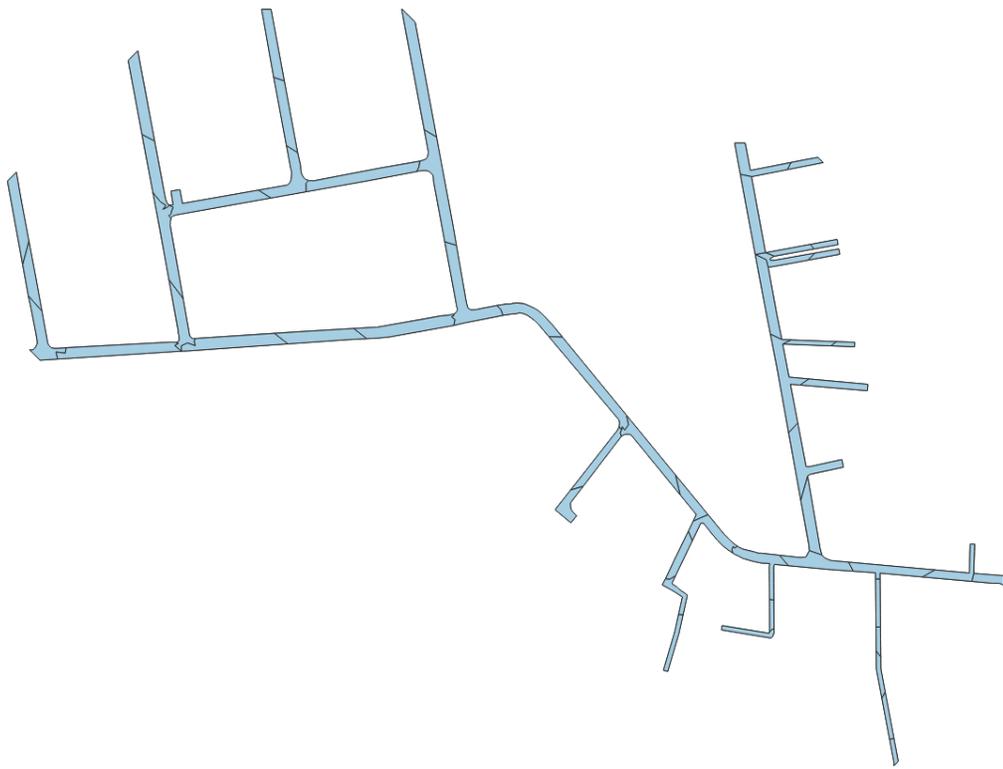
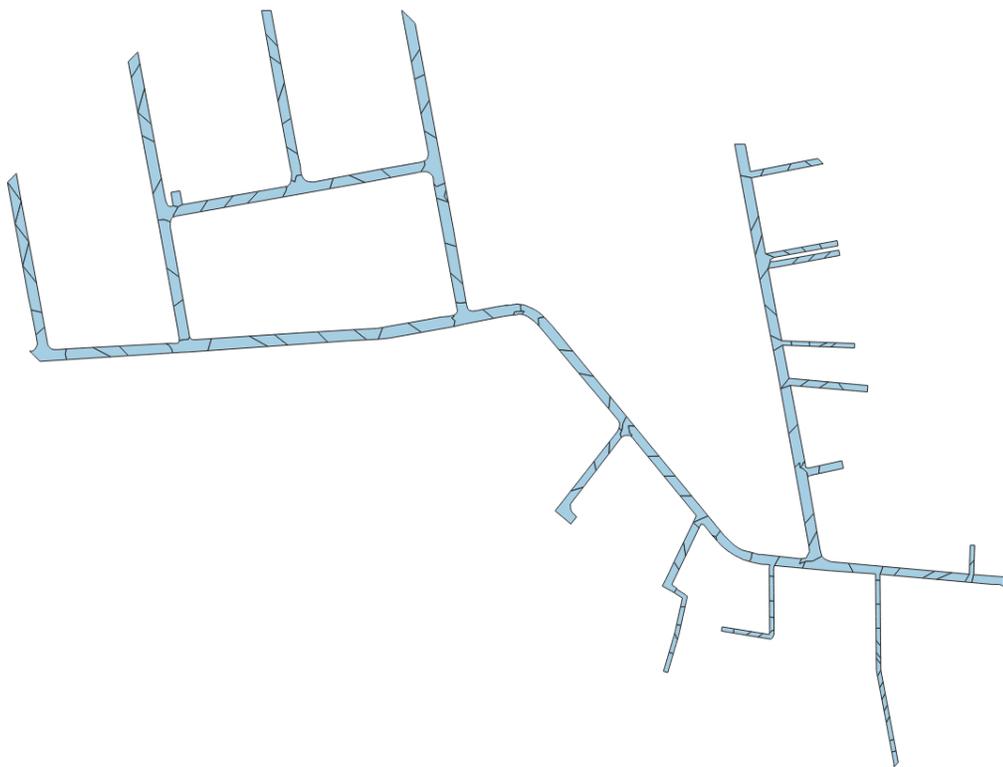


Figura 5.10: Disolución Calle simple con distintos Factores de Forma



(a) Disolución $FF = 0,2$



(b) Disolución $FF = 0,4$

Figura 5.11: Disolución Varias Calles con distintos Factores de Forma

Como se puede ver en los resultados, los polígonos delgados y alargados fueron correctamente divididos en base a la restricción impuesta. Por ejemplo, en el polígono de la calle simple, se ve que el polígono es dividido en subsecciones en el que cada una cumple con el factor de forma solicitado, salvo en algunos casos en el que el factor de forma es lo suficientemente estricto para que los propios triángulos resultantes de la triangulación inicial no cumplan con el factor de forma y el algoritmo no encuentra forma de eliminarlo por disolución.

En la Figura 5.10 se muestran dos casos representativos, en los que uno muestra la disolución por $FF = 0,2$ que es bastante más permisiva, pero que aún así logra el objetivo principal, mientras que por otra parte la disolución por $FF = 0,4$ divide la calle en secciones más pequeñas, incluso dejando un par de triángulos originales que no cumplían la restricción de forma y no pudieron ser disueltos por el algoritmo.

Por otro lado, en el caso del polígono que representa una unión de calles, este es un tema un poco más complicado, pero que de todas maneras el algoritmo logra sus objetivos. Existen casos como con $FF > 0,5$ en los que el algoritmo no logra realizar la disolución en una cantidad considerable de polígonos, e incluso con $FF = 0,6$ se ve que el promedio de FF de los polígonos resultantes es incluso menor al mínimo que se trató de imponer, de hecho un poco menos de la mitad de los polígonos finales no cumplían la restricción.

Sin embargo, como se puede apreciar en la Figura 5.11, la disolución para factores de forma más flexibles logra obtener resultados deseables, solo con algunos casos de borde que no pudieron cumplir la restricción.

En el caso de que sea necesario imponer un factor de forma más estricto como por ejemplo $FF = 0,6$ y lograr obtener resultados con una menor cantidad de polígonos que no cumplan la restricción, se estima necesario realizar la triangulación inicial con un mayor número de polígonos y/o un ángulo mínimo mayor a 25° , lo que tiene consecuencias principalmente en el tiempo de ejecución de la disolución. Por ejemplo, el imponer un ángulo de 30° , aumentando la cantidad de triángulos iniciales desde 1.371 a 2.038, se logró obtener una disolución en 359 polígonos finales, con un promedio de $FF = 0,624451$, de los cuales 83 son los que no cumplen la condición de forma impuesta. Una mejora sustancial con respecto al caso anterior con 25° , pero con el costo en tiempo y forma de los polígonos finales que el usuario debe considerar de acuerdo a lo que necesite.

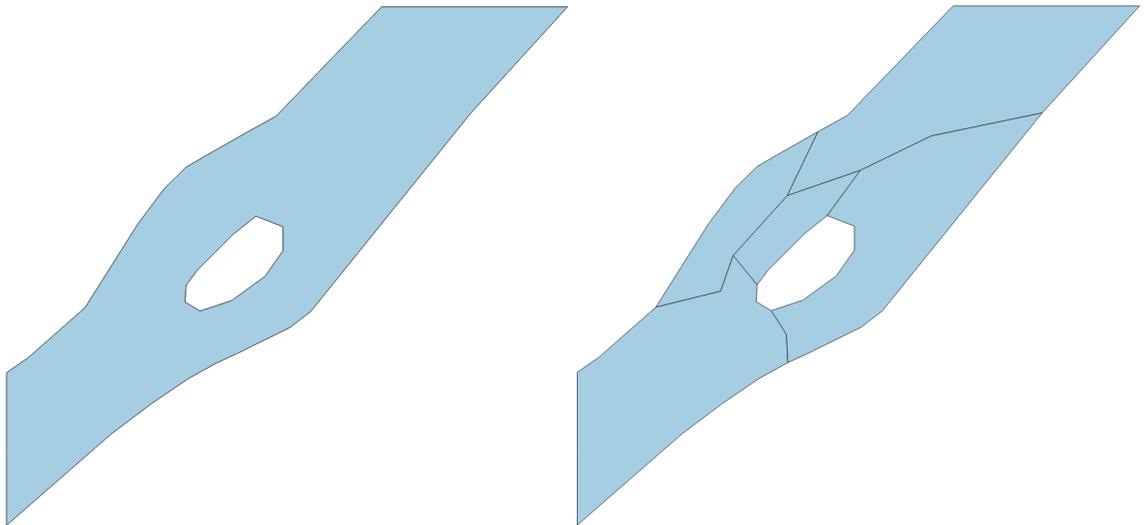
5.3.3. Índice de Solidez versus Índice de Convexidad

Otro de los puntos a evaluar es el cómo afectan los índices de solidez y de convexidad en la disolución. Ambos descriptores comparan la forma del polígono con respecto a la de su cerradura convexa, siendo comparados a través de las áreas y los perímetros respectivamente. Con estos descriptores de forma lo que se busca es obtener polígonos finales que no estén tan alejados en forma de su cerradura convexa y con esto los centroides tienden a quedar dentro del polígono original.

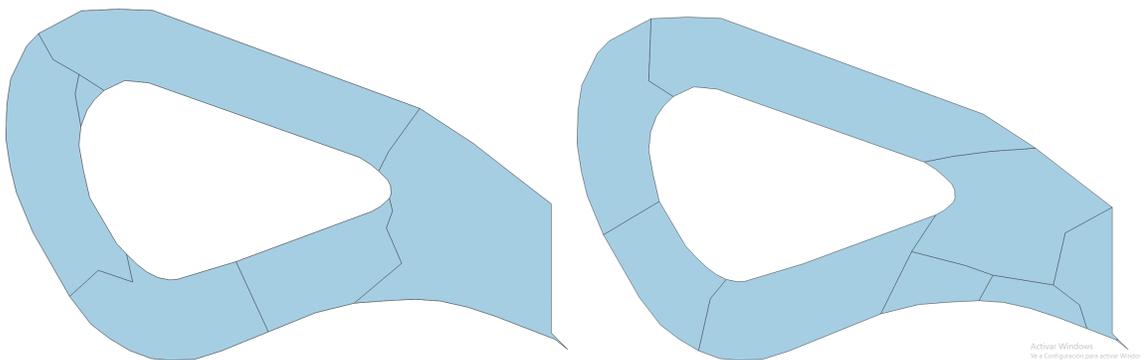
Intuitivamente, se entiende que los polígonos que tiene un gran agujero en su interior se ven más afectado en el índice de solidez y de convexidad con respecto a uno que tiene un pequeño agujero en su interior.

Diferencia en disolución polígonos con agujero y centroide fuera			
Polígono Entrada	Descriptor de Forma	N° Polígonos	N° Centroides fuera
Calle con agujero	$IS = 0,80$	1	1
Calle con agujero	$IS = 0,85$	5	0
Calle retorno	$IS = 0,80$	6	0
Calle retorno	$IS = 0,85$	9	0

Tabla 5.16: Diferencia en disolución polígonos con agujero y centroide fuera



(a) Calle con agujero , Disolución $IS = 0,80$ (b) Calle con agujero , Disolución $IS = 0,85$



(c) Calle retorno , Disolución $IS = 0,80$ (d) Calle retorno , Disolución $IS = 0,85$

Figura 5.12: Disolución en polígonos con agujero y centroide fuera

Como se puede ver en estos casos de prueba, en los que ambos representan polígonos de calles con agujero y su centroide fuera, los resultados de aplicar los distintos índices de solidez se comportan tal como se esperaba. En el primer caso, de la calle con agujero dentro, se ve que se requiere un índice de solidez mayor a 0,8, ya que el polígono original ya cumplía este índice y al realizar el proceso de triangulación y disolución se obtuvo como resultado el polígono original, con su centroide fuera. En el resto de resultados, se ve que con los índices adecuados el polígono es efectivamente dividido en polígonos más pequeños que no solo logran cumplir con los índices impuestos, sino que también logran mantener sus centroides dentro.

Si bien en estos casos se muestra que el índice de solidez puede fallar en el caso de que el centroide se encuentre en un agujero pequeño al interior del polígono, esto se puede calificar como un caso puntual y que se debe ajustar por parte del usuario. A pesar de esto, el índice de solidez resulta ser un índice muy útil para detectar polígonos con su centroide posiblemente fuera. Particularmente, este índice es bastante sensible con respecto a polígonos delgados y alargados que pueden tener curvaturas y posiblemente su centroide fuera.

En la Figura 5.13, el polígono original era una calle, un polígono delgado y alargado, el cual tenía su centroide fuera debido a su curvatura. En este caso, el polígono original tenía un índice de solidez de 0,357 y un índice de convexidad de 0,985, con lo que el índice de solidez requirió una restricción menos estricta para poder dividir el polígono, mientras que el índice de convexidad requería una restricción tan estricta, que no tiene sentido aplicar en otros casos.

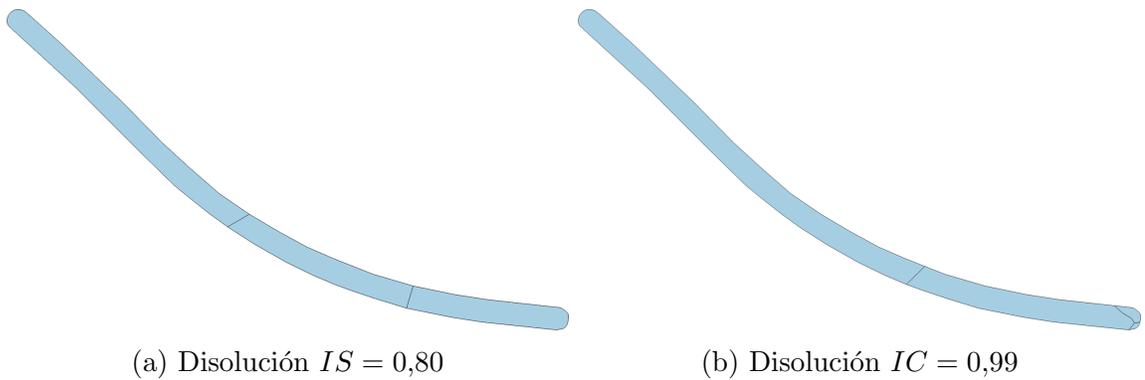


Figura 5.13: Disolución Calle comparando índices de solidez y convexidad

Como ejemplo final, se evaluó los índices de convexidad y de solidez en un polígono con varios agujeros como lo es el del Centro Deportivo.

Disolución Centro Deportivo con índices de solidez y convexidad			
Polígono Entrada	Descriptor de Forma	N° Polígonos	N° Centroides fuera
Centro Deportivo	$IS = 0,80$	15	0
Centro Deportivo	$IS = 0,90$	23	0
Centro Deportivo	$IC = 0,80$	6	1
Centro Deportivo	$IC = 0,90$	13	1

Tabla 5.17: Disolución Centro Deportivo con índices de solidez y convexidad

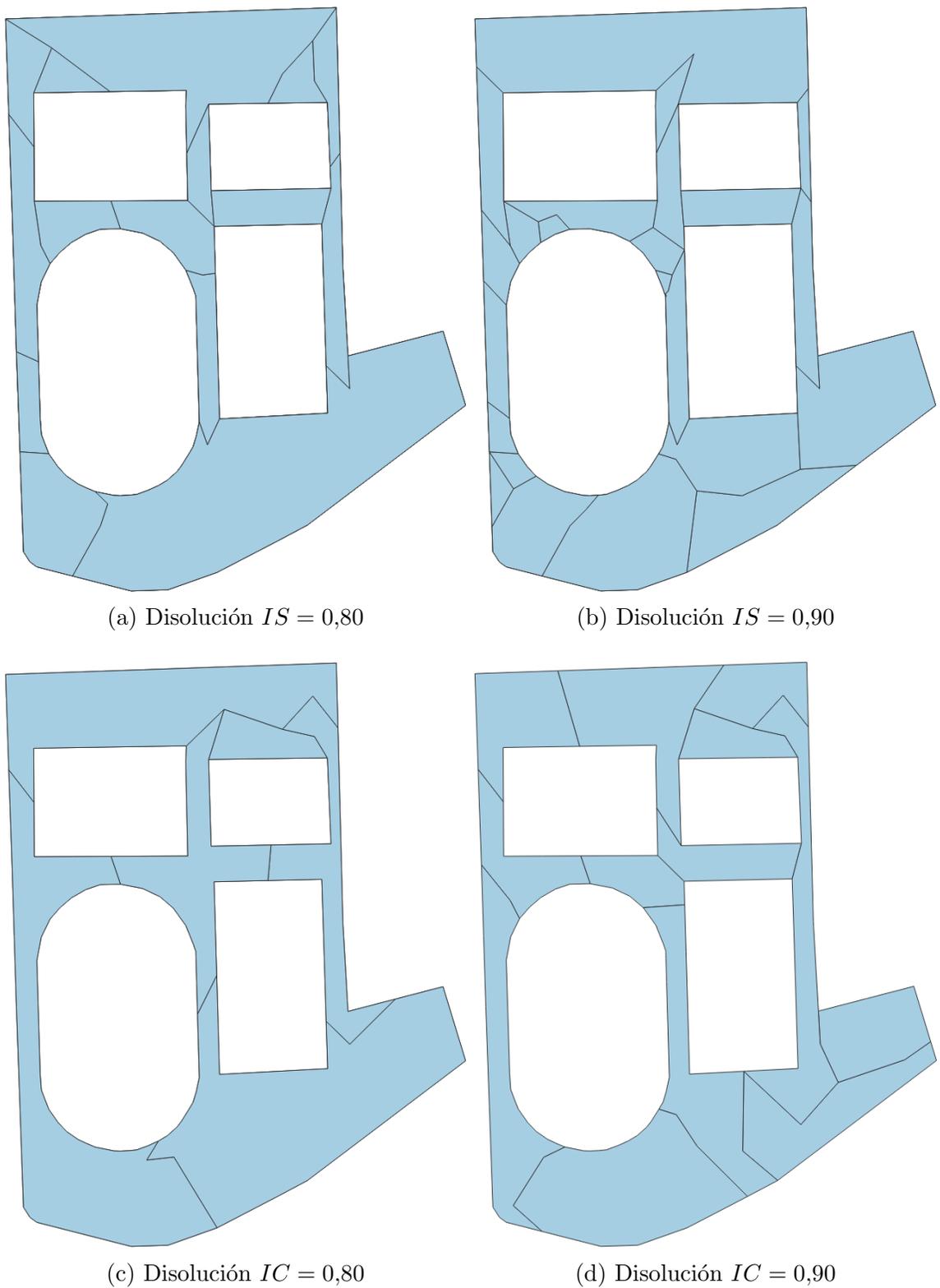


Figura 5.14: Disolución Centro Deportivo para distintos índices de solidez y convexidad

Como se puede ver, la disolución por índice de solidez tiende a generar una mayor cantidad de polígonos, pero al mismo tiempo logró evitar generar polígonos con centroides fuera. Por otro lado, la restricción de índice de convexidad genera una cantidad menor de polígonos

pero no logra evitar generar polígonos con centroide fuera. Si bien en este caso se observa que el mejor resultado en cuanto a número de polígonos generados finales y los que contienen centroide fuera es la disolución por índice de solidez de 0,80, hay que tener en cuenta que ambos índices tienen distintos usos y no es posible dar una conclusión definitiva para determinar el mejor índice en un caso general, ya que depende de lo que necesite el usuario finalmente, al cual le puede bastar el resultado de $IC = 0,8$ ya que le generaba una menor cantidad de polígonos finales y solo uno de estos con su centroide fuera, el cual puede ser refinado mediante este mismo proceso.

Otro de los aspectos a destacar es que, a diferencia del factor de forma y compacidad, la disolución por índices de solidez y de convexidad nunca obtienen polígonos que sean menores al descriptor de forma impuesto. Esto se debe a que para los casos probados todos provienen desde una triangulación de una URH, por lo tanto por definición el polígono a disolver está compuesto de polígonos convexos. Esto se cumple siempre que la disolución sea aplicada sobre una triangulación, ya que al aplicarla sobre otro tipo de entrada (por ejemplo refinar una disolución anterior) es posible encontrar índices de solidez y convexidad menores a 1.

5.4. Evaluación de casos particulares

Si bien el algoritmo de triangulación y disolución logró obtener los resultados deseados, existen algunos casos particulares que hay resulta de interés evaluar.

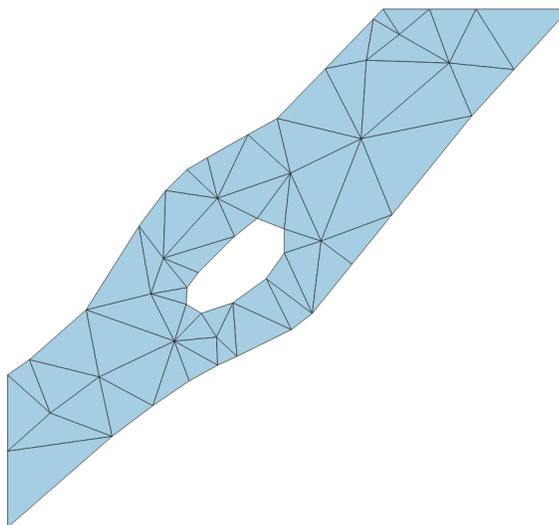
5.4.1. Aplicación repetida de algoritmo de disolución

La separación del algoritmo en un paso de triangulación y posterior disolución permite aplicar la disolución sobre el resultado con parámetros distintos o iguales. Si bien el algoritmo permite aplicar la disolución con los mismos parámetros, lo que se esperaba era que esto no cambie el resultado final.

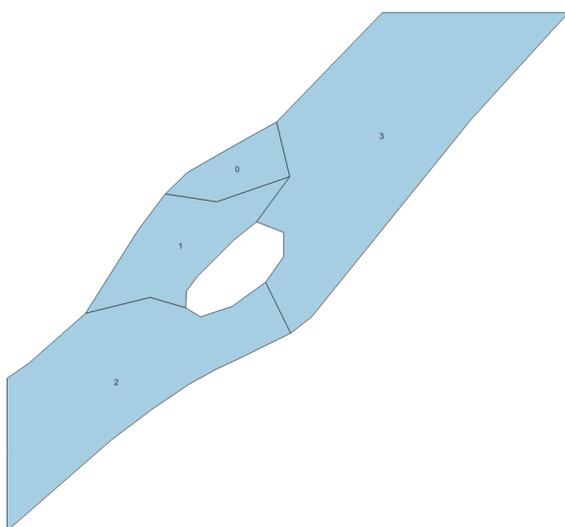
Sin embargo, luego de probar el algoritmo múltiples veces, resulta que para las disoluciones en base a descriptores de forma, el resultado que se obtiene al aplicar la disolución sobre la triangulación original no es el resultado final que se pensaba, ya que si se aplica el algoritmo de disolución con los mismos parámetros, se logra refinar el resultado. En algunos casos es necesario aplicar el algoritmo hasta 6 veces para obtener un resultado final que no puede ser refinado. Es importante destacar que todos los resultados de disolución mostrados anteriormente son los resultados definitivos, luego de aplicar la disolución hasta que no fue posible refinar más.

Como es posible ver en la Figura 5.15, se intentó hacer la disolución del polígono con restricción en el índice de solidez de 0,8. Como el polígono original tiene un índice de solidez de 0,863, se espera que el resultado de la disolución sea el polígono original que se trianguló. Sin embargo, se ve que al aplicar la disolución se obtiene como resultado el polígono dividido en 4 polígonos, en el que cada uno cumple con la restricción impuesta. Si bien este fue el resultado del algoritmo de disolución, se ve claramente que el algoritmo no realizó una

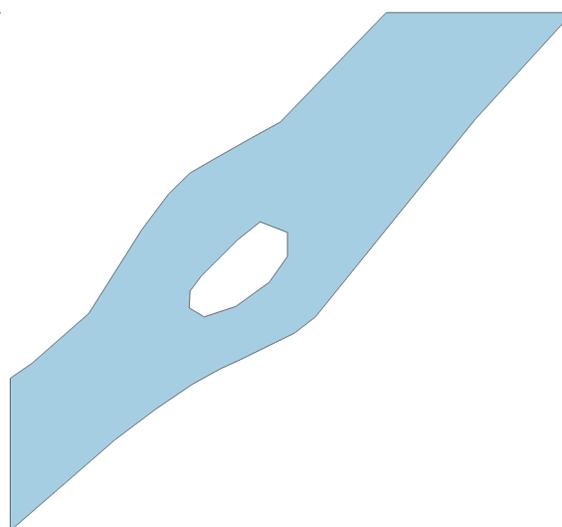
disolución completa, por ejemplo los polígonos 0 y 1 pueden ser combinados y no debiese haber ningún problema en cuanto a su índice de solidez. Luego de aplicar el algoritmo de disolución nuevamente sobre este primer resultado, se logra obtener el resultado que se esperaba.



(a) Triangulación original 25°



(b) Disolución N°1



(c) Disolución N°2

Figura 5.15: Disolución de polígono en varios pasos $IS = 0,8$

La razón de este comportamiento se debe a que el algoritmo de disolución cuando procesa un polígono y no es posible combinar con ninguno de sus vecinos por no cumplir el descriptor de forma, éste lo saca de la lista de polígonos a procesar y queda como un polígono final. Vale decir, que si después del proceso de disolución uno de sus vecinos se combina y ahora sí es posible combinarlo con el polígono que ya se procesó, el algoritmo no puede hacerlo ya que el primer polígono fue descartado como un polígono final. Esta situación no ocurre para la disolución con restricción de área exclusivamente, en donde la disolución es final en la primera ejecución.

5.4.2. Casos en el que el algoritmo de disolución no obtiene el resultado deseado

Existen casos en los que el algoritmo de disolución no logra la disolución esperada. Al igual que el caso anterior, esto ocurre exclusivamente en disoluciones en base a algún descriptor de forma.

Usando el mismo ejemplo de la Figura 5.15, al aumentar la restricción en el índice de solidez a 0,83, se espera que el algoritmo de disolución logre obtener como resultado el polígono original, el cual cumple con esta condición. Sin embargo, como se ve en la Figura 5.16, luego de aplicar un par de veces el algoritmo de disolución, se logra obtener como resultado final 5 polígonos que cumplen la condición en vez de 1, y es el resultado final de la disolución, imposible de refinar con los mismos parámetros de $IS = 0,83$.

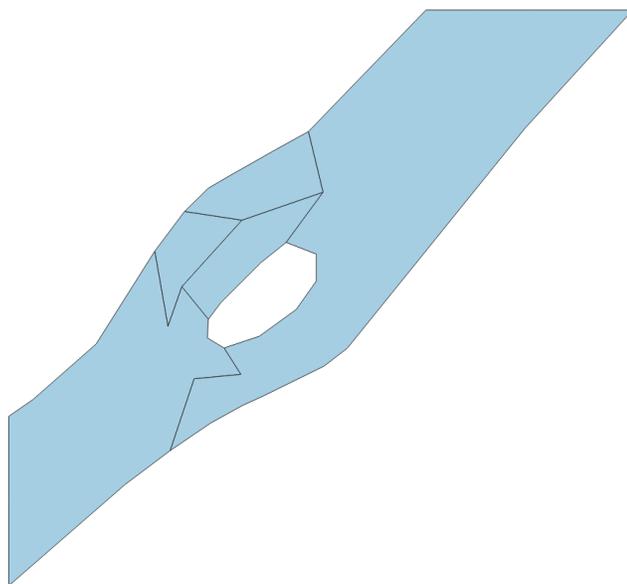


Figura 5.16: Disolución de polígono con $IS = 0,83$

Como ejemplo más claro, se puede ver en el caso teórico mostrado en la Figura 5.17. Si tomamos un cuadrado perfecto, con $FF = 0,7854$, y lo triangulamos de la misma manera que se ve en la figura, si se realiza una disolución con $FF = 0,5$ el algoritmo no puede realizar ninguna disolución y da como resultado la triangulación inicial. Esto se debe a que el algoritmo solamente considera como posibles candidatos a disolución a los vecinos directos, pero solamente uno a la vez, y como en este caso es imposible encontrar un vecino que cumpla el descriptor de forma indicado, no es posible realizar ninguna disolución por parte del algoritmo.

Estos casos de borde no son abordados por el algoritmo, el cual en su proceso de disolución realiza las uniones de polígonos uno por uno. Una alternativa puede ser permitir que cuando no se encuentren vecinos con los cuales realizar la disolución, explorar la posibilidad de unir múltiples vecinos al mismo tiempo (y vecinos de los vecinos), de tal manera que la nueva unión de polígonos sí cumpla con las restricciones dadas. La implementación de este algoritmo o una solución alternativa resulta interesante para evaluar a futuro, ya que no

solo permitirá resolver este tipo de casos de borde, sino también es posible que mejore los resultados obtenidos en la Figura 5.8d. Queda propuesto ésta, u otra alternativa de solución, para su implementación a futuro.

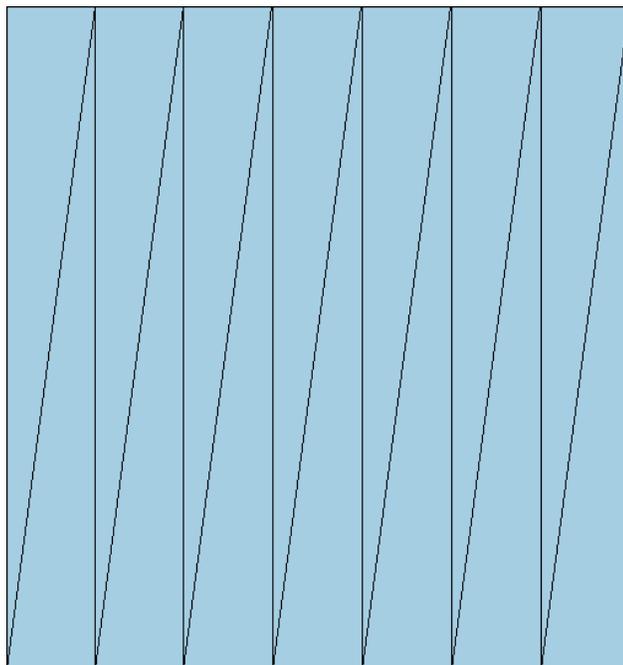


Figura 5.17: Disolución de polígono con $FF = 0,5$

5.4.3. Diferencias en el orden de disolución

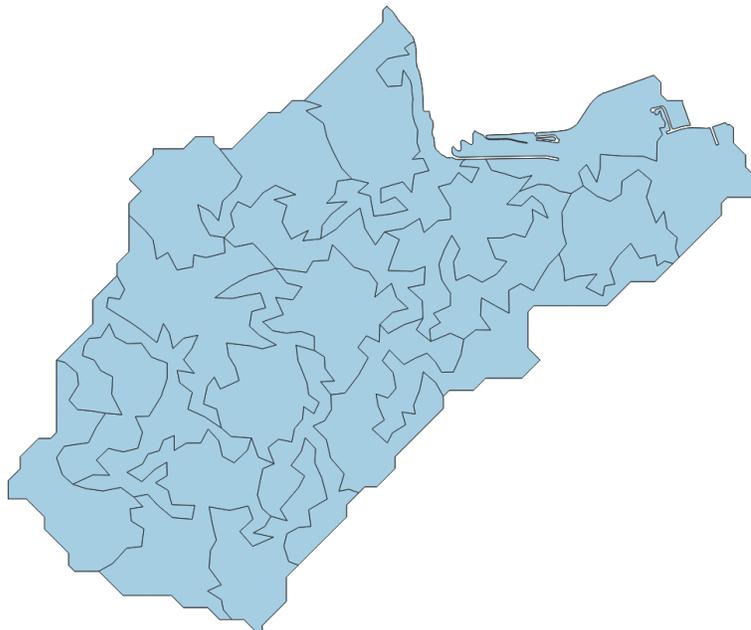
Dentro de las opciones que se da en la disolución, una de ellas es permitir al usuario elegir el orden en el que el algoritmo procesa la disolución, ya sea por orden de área de los polígonos ascendente, descendente o sin orden alguno.

Se analizarán las diferencias entre el orden de área ascendente o descendente, ya que el algoritmo sin orden no es posible analizar en cuanto a resultados porque estos no son consistentes y dependen de la entrada. Sin embargo, lo que sí se puede decir del algoritmo sin orden es que el tiempo de ejecución mejora ya que no realiza ningún ordenamiento de los polígonos en base a área.

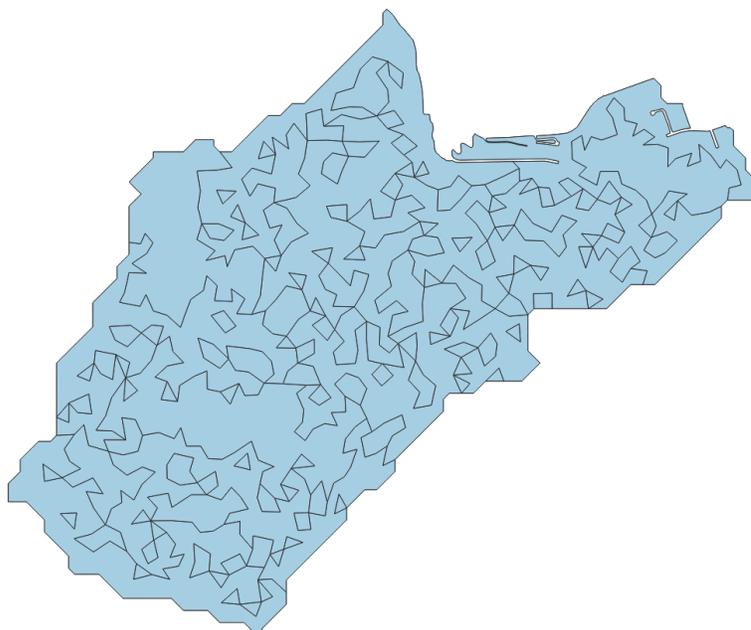
Disolución URH Grande triangulado en área máxima 1.000 m^2			
Restricción	Orden	N° Polígonos	N° Pasos
Área máxima 100.000 m^2	Ascendente	21	1
Área máxima 100.000 m^2	Descendente	131	1
$FF = 0,5$	Ascendente	53	4
$FF = 0,5$	Descendente	50	4

Tabla 5.18: Disolución URH Grande triangulado en área máxima 1.000 m^2

Como se puede ver, el mayor cambio se produce en cuanto a la restricción por área. En los casos de restricciones por descriptores de forma la diferencia entre el orden no es sustancial, aunque en otros casos de prueba se ha experimentado una mejora mínima en el orden descendente en cuanto a cantidad de polígonos resultantes y los pasos para llegar a la disolución final, sin embargo no es tan considerable.



(a) Disolución orden ascendente



(b) Disolución orden descendente

Figura 5.18: Disolución por área en distintos ordenes

Como se puede ver en la Figura 5.18, el algoritmo de disolución en orden descendente

presenta resultados no deseados. La explicación a este fenómeno es porque el algoritmo en orden ascendente al ir diluyendo desde el polígono más pequeño, va aumentando el mínimo de los polígonos y creando polígonos cada vez más grande, evitando que queden polígonos pequeños. Mientras tanto, el algoritmo en orden descendente comienza desde el más grande y lo va uniendo con los más pequeños, buscando maximizar los polígonos de mayor área. En este proceso de maximización, es muy posible que al llegar a su límite de área máxima, queden polígonos en su interior, los cuales no pueden ser diluidos por la restricción de área, tal como se ven la Figura 5.18b.

El orden por defecto implementado es el ascendente, para evitar estas situaciones, pero el usuario puede ir cambiando a un orden descendente para ver los resultados, en especial en el caso de disolución por factores de forma en donde el orden puede incluso resultar mejor, o sin orden para lograr un algoritmo más rápido.

5.5. Tiempos de ejecución

Uno de los aspectos a tener en cuenta en este algoritmo es el tiempo de ejecución del proceso. Dentro de este aspecto, se optó por solamente medir el algoritmo de disolución, ya que el tiempo de triangulación depende solamente del algoritmo Triangle implementado por Shewchuk 1996)[10], el cual en estos casos es considerablemente pequeño.

Se consideró medir las siguientes variables dentro del algoritmo de disolución:

- Tiempo en determinar los vecinos.
- Tiempo en el proceso de disolución de polígonos con los vecinos ya determinados. El algoritmo de disolución a medir considerará un orden de polígonos desde área más pequeña a más grande (el tiempo es similar para el orden inverso).
- Tiempo en el proceso de disolución con vecinos ya determinados, pero sin realizar ningún orden de polígonos.

A continuación se muestran los tiempos de disolución tomando como variable el número de polígonos de la triangulación y el tiempo medido en segundos¹:

Tiempo de ejecución proceso de disolución [s]					
Proceso	$N = 10$	$N = 100$	$N = 1008$	$N = 10002$	$N = 43271$
Determinar Vecinos	0,000862	0,015864	0,189549	1,956119	8,430345
Disolución en orden	0,000894	0,022464	1,553728	177,3327	3944,590
Disolución sin orden	0,000673	0,007596	0,078988	0,859648	4,809923

Tabla 5.19: Tiempo de ejecución proceso de disolución [s]

¹El proceso fue ejecutado en un computador con procesador *AMD Ryzen 5 1600*, en la versión *QGIS 3.22.11-Białowieża, Python 3.9.5*, usando la librería *timeit* para medir el tiempo de ejecución.

Como se puede ver en la Tabla, hay tiempos de ejecución muy dispares, en donde los tiempos de encontrar vecinos y de disolución sin orden se encuentran dominados por un comportamiento lineal con respecto al número de polígonos de entrada, mientras que el tiempo de disolución con orden tiene comportamiento cuadrático e incluso cúbico.

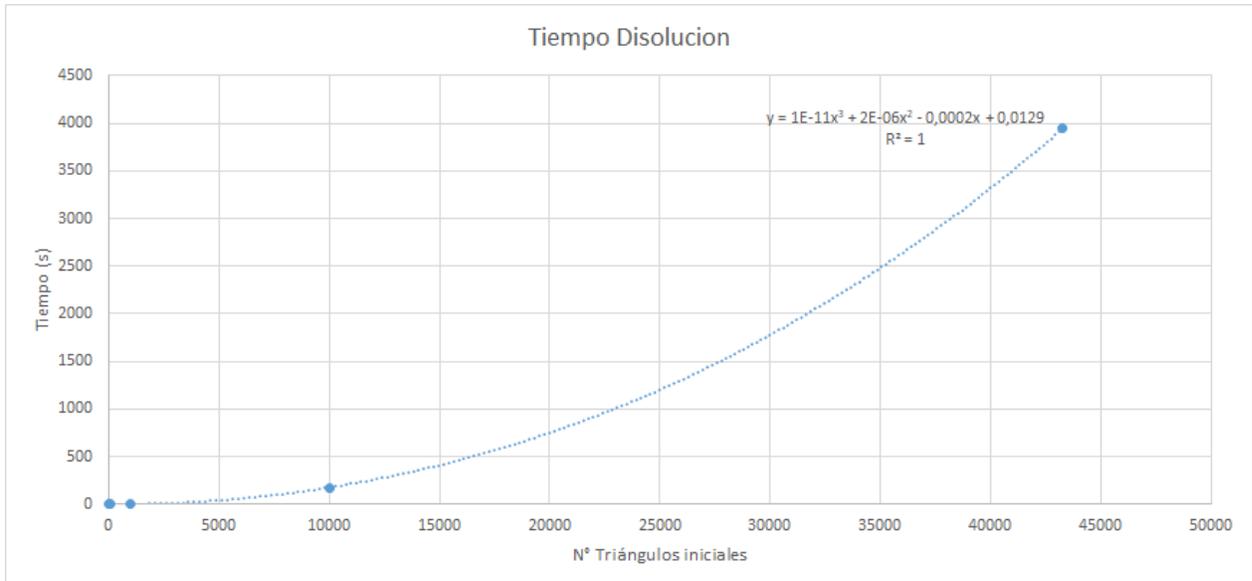


Figura 5.19: Regresión lineal sobre tiempo de ejecución de la disolución en orden

Si bien el tiempo de ejecución de las disoluciones en orden y sin orden se encuentran dentro de lo esperado, es el tiempo obtenido en encontrar vecinos el que llama más la atención, ya que viendo el algoritmo se esperaba un comportamiento cuadrático o cúbico. La razón de esto puede ser porque los tiempos se midieron tomando como entrada una triangulación, ya que de esta manera se planea usar sobre las URHs, y por lo tanto al verificar si un polígono es efectivamente vecino de otro, se comparan una menor cantidad de vértices a otro tipo de polígono.

Para confirmar la variable del tiempo de determinar vecinos, se puede tomar como caso particular el hacer una disolución sobre el conjunto de URH original de Lo Barnechea, el cual está conformada por 2.906 polígonos de distintas formas. Al aplicar una disolución con orden, se encuentra el comportamiento inverso, en donde el tiempo en determinar vecinos es mucho mayor al de disolución. Particularmente, el tiempo en encontrar vecinos fue de 195,79708 segundos, mientras que el tiempo en realizar la disolución es de 4,153508 segundos.

Capítulo 6

Conclusiones

Durante este trabajo de memoria se estudió y diseñó un plugin en QGIS cuyo objetivo principal es detectar polígonos que no cumplieran con ciertos criterios importantes para el modelamiento hidrológico, y posteriormente descomponer dicho polígono en polígonos más pequeños que cumplan con dichos criterios. Dentro del contexto general, se puede decir que los objetivos propuestos fueron mayormente cumplidos, salvo la integración de la descomposición en base a propiedades físicas, la cual fue descartada en el análisis debido a su complejidad.

Si bien durante el trabajo se había considerado en su diseño realizar el proceso completo de triangulación y disolución, luego de un análisis y conversaciones con el profesor guía, se optó por mantener estos procesos separados e independientes, ya que se estimó que no era necesario que el sistema fuera completamente automático en el proceso completo. Tomando en cuenta este antecedente, se puede concluir que fue una excelente decisión, ya que no solo permite al usuario interactuar con estos procesos por separado, sino que también corrige ciertas particularidades del algoritmo de disolución que se discutieron en la sección 5.4, las cuales no se tenían consideradas al principio del trabajo.

Dentro de los resultados obtenidos y el ajuste de los parámetros, se puede concluir que existen diversos factores y no es posible dar una configuración que resuelva todos los casos. Sin embargo, estos resultados nos dan una idea general de qué parámetros pueden servir.

En el caso de la triangulación, se observa que el parámetro de ángulo mínimo es importante indicar, en especial en los casos que se intente una disolución por algún descriptor de forma. Dentro de los resultados obtenidos, es posible determinar que un ángulo de 20° sería una restricción razonable para alcanzar una triangulación de calidad. Dependiendo del caso, incluso sería recomendable aumentar a 25° e incluso 30° , si se requiere imponer una disolución por descriptor de forma más estricta.

En cuanto a la disolución, se observó que los distintos descriptores de forma buscaban resolver un problema diferente, pero que funcionaron mayormente de la manera requerida, en donde el Factor de Forma y Compacidad lograban detectar y disolver un polígono largo y alargado, mientras que los Índices de Solidez y Convexidad lograban dividir las URHs en polígonos más pequeños y convexos, y de esta manera obtener polígonos más propensos a tener su centroide dentro.

En cuanto a los tiempos de ejecución, se obtuvieron los resultados esperados, con la excepción del algoritmo de encontrar y formar vecinos, el cual se estimó de orden cuadrático, pero en la práctica fue mayormente lineal. Esto nos muestra que para medir el tiempo de un algoritmo no solo es necesario ver el algoritmo y evaluar de acuerdo a este en su caso general, sino que también hay que considerar en qué condiciones se aplicará este algoritmo. Como en este caso la entrada del algoritmo es generalmente una triangulación, el efecto cuadrático del algoritmo de encontrar vecinos no fue tan considerable como se estimó en un principio.

6.1. Trabajo a Futuro

Como trabajo a futuro quedarían por ver distintas maneras de abordar el problema, tales como evaluar otros factores de forma que puedan ser útiles o interesantes para la caracterización de los polígonos, o cambiar el orden en la disolución e ir diluyendo en base a otro criterio, como por ejemplo, por el vecino con borde más largo en vez de el de área más grande, u otros factores.

Un punto importante a considerar como trabajo a futuro, es ver la eficiencia de los algoritmos planteados, en especial el algoritmo de disolución. Como se vio en el trabajo, para un caso general de triangulación y disolución, el mayor tiempo de ejecución se vio en el proceso de disolución con algún orden de área. También quedaría por ver si es posible mejorar la eficiencia del algoritmo de encontrar vecinos, el cual si bien resultó lineal en la práctica ya que tomó como entrada una triangulación, este podría no ser siempre el caso. Junto con esto, también quedaría por realizar la comparación en cuanto a tiempo entre el algoritmo implementado en este trabajo y el implementado en Sanzana, 2011[9].

En cuanto a la calidad de la disolución, también queda pendiente el cómo resolver los temas planteados en 5.4, en los que la disolución por descriptores de forma no siempre llegó a resultados satisfactorios en su primera aplicación, e incluso hay casos particulares en los que este falla. Como se indicó, una posible solución sería permitir la evaluación de 1 o más vecinos al mismo tiempo, y determinar si estas uniones cumplen con las restricciones impuestas.

Otro de los puntos a ver que pueden quedar como trabajo a futuro es la descomposición de URHs en base a la diferencia en las propiedades físicas, algo que no se abordó en este trabajo.

La primera versión del Plugin QGIS que sirvió de base para el desarrollo del actual trabajo de memoria de título fue publicado en un libro de análisis espacial y riesgos en ambientes hídricos (Sanzana, 2018[16]). Por lo que se recomienda en el futuro implementar los resultados de las segmentaciones geométricas en un modelo hidrológico para analizar la contribución de este trabajo en el área de recursos hídricos. En el futuro se recomienda llevar a cabo este trabajo de manera conjunta entre Ciencias de la Tierra y Ciencias de la Computación. Si bien el trabajo realizado se hizo en base a URHs reales del sector de Lo Barnechea, el foco de este fue principalmente en integrar e implementar estos algoritmos en *QGIS*, y probar si estos algoritmos lograron sus objetivos de mejoramiento de las URHs en base a los objetivos propuestos, quedando pendiente probar si el mejoramiento teórico de URHs en este trabajo se traduce a un mejoramiento en el modelo hidrológico.

Bibliografía

- [1] K. Gregory. The human role in changing river channels. *Geomorphology*, 79:172–191, 09 2006.
- [2] Jorge Gironás, Larry Roesner, and Herve Andrieu. Morphologic approach in studying urbanized and suburbanizing watersheds. pages 1–10, 05 2007.
- [3] Branger F. Braud I. Debionne S. Viallet P. Rodríguez F. Jankowfsky, S. Development of a suburban catchment model within the LIQUID® framework. In *Proceedings of the International congress on Environmental Modeling and Software*, Ontario, Ottawa, Canada, 2010.
- [4] Sonja Jankowfsky. Understanding and modelling of hydrological processes in small peri-urban catchments using an object-oriented and modular distributed approach. Application to the Chaudanne and Mercier sub-catchments (Yzeron catchment, France). 12 2011.
- [5] Fabrice Rodriguez, Hervé Andrieu, and Floriane Morena. A distributed hydrological model for urbanized areas – Model development and application to case studies. *Journal of Hydrology*, 351(3):268–287, 2008.
- [6] Wolfgang-Albert Flügel. Delineating hydrological response units by geographical information system analyses for regional hydrological modelling using PRMS/MMS in the drainage basin of the River Bröl, Germany. *Hydrological Processes*, 9(3-4):423–436, 1995.
- [7] P. Sanzana, J. Gironás, I. Braud, N. Hitschfeld, F. Branger, F. Rodriguez, M. Fuamba, J. Romero, X. Vargas, J. F. Muñoz, S. Vicuña, and A. Mejía. Decomposition of 2D polygons and its effect in hydrological models. *Journal of Hydroinformatics*, 21(1):104–122, 09 2018.
- [8] P. Sanzana, J. Gironás, I. Braud, F. Branger, F. Rodriguez, X. Vargas, N. Hitschfeld, J.F. Muñoz, S. Vicuña, A. Mejía, and S. Jankowfsky. A gis-based urban and peri-urban landscape representation toolbox for hydrological distributed modeling. *Environmental Modelling Software*, 91:168–185, 2017.
- [9] Pedro Sanzana. Automatización del procesamiento de unidades de respuesta hidrológica (URHs) con GRASS para un modelo hidrológico distribuido. Tesis para optar al grado de Magíster en Ciencias de la Ingeniería Mención en recursos y medioambiente hídrico, Universidad de Chile, 2011.

- [10] Jonathan Richard Shewchuk. *Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator*. Springer, Berlin, Heidelberg, 1996.
- [11] Guilin Liu, Zhonghua Xi, and Jyh-Ming Lien. Dual-space decomposition of 2d complex shapes. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4154–4161, 2014.
- [12] Jyh-Ming Lien and Nancy M. Amato. Approximate convex decomposition of polygons. *Computational Geometry*, 35(1):100–123, 2006. Special Issue on the 20th ACM Symposium on Computational Geometry.
- [13] John Russ and John C. *Image Processing Handbook, Fourth Edition*. 01 2002.
- [14] David H. Douglas and Thomas K. Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122, 1973.
- [15] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.
- [16] Pedro Sanzana, S. Villaroel, Isabelle Braud, Nancy Hitschfeld, Jorge Gironás, F. Branger, F. Rodriguez, Ximena Vargas, Tomás Gómez, N. Bagdhadi, and Mehrez Zribi. *Representation du réseau de drainage en zones urbaines et périurbaines à l'aide d'un maillage polygonal 2D formé d'éléments pseudo-convexes*. 01 2018.